
SMARTHOUSE

DATAWAREHOUSE PLATFORM VOOR SMARTINSTALL



De Haagse Hogeschool – IT & Design
Afstudeerverslag Informatica
SmartInstall

Jan Bongers
09057633

Versie: 1.0
Examinatoren: H.J. Middendorp & H.C. van den Bosch

REFERAAT

Titelbeschrijving

Jan Bongers, "SmartHouse, data warehouse platform voor SmartInstall". Afstudeerverslag opleiding Informatica, Haagse Hogeschool, 2016.

Samenvatting

SmartInstall verzorgt het proces van meterwisselingen en metercontroles voor diverse netbeheerders en biedt daarbij hun planning- en administratiesoftware, SmartWork 3.0, bij aan. Eén van de diensten die zij hiervoor aanbieden, is het rapporten over en het analyseren van de data omtrent de meterwisseling die een monteur registreert tijdens zijn werkzaamheden. Voor het leveren van deze dienst is echter nog geen goede oplossing, waardoor het SmartInstall veel tijd en kostbare resources kost. Er is de wens voor een oplossing dat dient functioneren als een warehouseomgeving. Alvorens de start van het project is een globale planning opgesteld en is er gekozen om te werken volgens Scrum. In de opstartfase van het project is er onderzoek gedaan omtrent warehousing en is er middels een pakketselectie advies uitgebracht over de te gebruiken software voor de DBMS, ETL en BI-modules. De gebruikte softwarepakketten betreffen PostgreSQL, Pentaho Kettle en Pentaho BI-suite. Voor de realisatie van de warehouseomgeving is gebruikgemaakt van technieken van de The Kimball Group. De scheduling van de ETL-jobs en het automatiseren van deployen en testen is gedaan met behulp van Jenkins.

Descriptor

Data warehouse, Pakketselectie, ETL, Business Intelligence, Pentaho, PostgreSQL, Afstudeerverslag, Informatica, Jenkins, Liquibase, The Kimball group

VOORWOORD

Het rapport wat voor u ligt is geschreven in het kader van verslaglegging betreft de werkzaamheden van het afstudeerproject tijdens de afstudeerperiode binnen de opleiding Informatica aan de Haagse hogeschool in Den Haag. Het afstudeerproject is gemaakt in opdracht van SmartInstall, gelegen in Capelle aan den IJssel, in de periode van 8 februari tot en met 3 juni 2016.

Hierbij wil ik de directie van SmartInstall bedanken voor de kans die zij mij hebben gegeven en voor de sfeervolle, motiverende werkomgeving waarin ik mijn afstudeeropdracht heb kunnen uitvoeren.

Ook wil ik mijn begeleiders Henk Middendorp en Henk van den Bosch bedanken voor de begeleiding, feedback en adviezen die ik heb ontvangen tijdens mijn afstudeerperiode. Mede dankzij de expertise van Henk Middendorp kon ik een goede start maken aan mijn afstudeerproject.

Tot slot wil ik mijn vriendin Wendy Vos bedanken voor de steun tijdens een moeilijke periode door overlijden in de familie dat plaats heeft gevonden tijdens het uitvoeren van het afstudeerproject.

Jan Bongers

3 juni 2015, 's-Gravenhage

INHOUDSOPGAVE

1	Inleiding.....	3
2	SmartInstall	4
2.1	SmartWork.....	5
2.2	Organisatiestructuur	5
3	Opdrachtoomschrijving.....	7
3.1	Aanleiding & probleemstelling	7
3.2	Doelstelling	8
3.3	Resultaat & deliverables	8
4	Aanpak.....	9
4.1	Scrum methodiek.....	9
4.2	Risico's.....	11
4.3	Sprint planning & uitvoer.....	12
5	Sprint 0 - Onderzoek en globale analyse	13
5.1	The Kimball group	13
5.2	Key requirements.....	16
5.3	Pakketselectie	17
5.4	Advies software pakketten	20
6	Sprint 1 - Warehouseprototype.....	22
6.1	Gegevensmodel	22
6.2	Ontwerpen database	24
6.3	Mapping van gegevens	28
6.4	Implementeren van database ontwerpen	30
6.5	Pentaho Kettle ETL.....	32
6.6	Prototype	33
7	Sprint 2 – Realisatie warehouse in staging.....	34
7.1	Demonstratie prototype.....	34
7.2	Wijzigen en uitbreiden database ontwerpen.....	35
7.3	Stagingserver	37
7.4	Uitbreiden ETL-Module.....	39
7.5	Pentaho Business Intelligence	50
8	Sprint 3 – Realisatie Bèta-versie	53
8.1	Extra requirements	53
8.2	Liquibase	54
8.3	Uitbreiden en afronden warehouse designs.....	56
8.4	Uitbreiden en Afronden ETL-Module	60

8.5	Jenkins finetuning en geautomatiseerd Testen	61
9	Conclusie en aanbevelingen	62
9.1	Conclusie.....	62
9.2	Aanbevelingen	63
10	Evaluatie.....	64
10.1	Producten.....	64
10.2	Proces.....	64
10.3	Kennis.....	65
10.4	Verbeteringen	65
11	Verantwoording Beroepstaken.....	66
11.1	Selecteren methoden, technieken en tools	66
11.2	Uitvoeren analyse door definitie van requirements	66
11.3	Opstellen gegevensmodel voor een database	66
11.4	Ontwerpen, bouwen en bevragen van een database	66
11.5	Bouwen applicatie	67
11.6	Beheren van configuraties.....	67
12	Bibliografie	68
	Bijlagen	70
	Bijlage A - Afstudeerplan.....	i
	Bijlage B - Pakketselectie	ii
	Bijlage C - Requirements	iii
	Bijlage D - Gegevensmodel	iv
	Bijlage E - Data warehousedesign.....	v
	Bijlage F - ETL & SQL.....	vi
	Bijlage G – Tussentijdse assessment.....	vii
	Bijlage H – Evaluatie SmartInstall.....	viii

CODE SNIPPETS LIJST

Code Snippet 1 – Invulling van dimensies	31
Code Snippet 2 - Physical design implementatie	31
Code Snippet 3 - PostgreSQL installatie.....	38
Code Snippet 4 - Pentaho Bi-suite installatie	38
Code Snippet 5 - GIT versiebeheer command.....	39
Code Snippet 6 - Work ended query	44
Code Snippet 7 - Workorder finished query.....	45
Code Snippet 8 - Custom JavaScript	45
Code Snippet 9 - Liquibase master-XML-bestand	54
Code Snippet 10 - Liquibase changelog.....	55

TABELLENLIJST

Tabel 1 - Risico analyse	11
Tabel 2 - SCD-type beschrijvingen	15
Tabel 3 - RequirementNummering.....	16
Tabel 4 - LongLists warehousemodules.....	19
Tabel 5 - Score toekenning	19
Tabel 6 - BI selectie matrix.....	20
Tabel 7 - Shortlists warehousemodules.....	20
Tabel 8 - Geadviseerde pakketten	21
Tabel 9 - Key requirements voor prototype	24
Tabel 10 - Prototype user stories.....	25
Tabel 11 - Asset inventory mapping	29
Tabel 12 - Sprint 2 user stories	35
Tabel 13 - Mapping van sprint 2 wijzigingen	43
Tabel 14 - Sprint 3 user stories	53
Tabel 15 - Meter forecast requirements	56
Tabel 16 - Mapping van sprint 3 wijzigingen	58

FIGURENLIJST

Figuur 1 - Organigram SmartInstall.....	5
Figuur 2 - Jira product backlog.....	10
Figuur 3 - Jira sprint flow.....	10
Figuur 4 - Tijdverdeling	12
Figuur 5 - Sprint cyclus	12
Figuur 6 - The Kimball Group	13
Figuur 7 - Starschema	14
Figuur 8 - BI Magix Quadrant.....	18
Figuur 9 – SmartWork 3.0 database	23
Figuur 10 - Conceptual design inventory fact.....	25
Figuur 11 - Logical design inventory fact	26
Figuur 12 - Conceptual design asset placed fact	27
Figuur 13 - Logical design asset placed fact	28
Figuur 14 - Spoon update employee dimesion.....	32
Figuur 15 – Configuratie SCD type	32
Figuur 16 - Kettle job flow.....	33
Figuur 17 - VoorbeeldRapportage	33
Figuur 18 - Employee dimension wijzigingen	36
Figuur 19 - Globale warehouse architectuur.....	37
Figuur 20 - Jenkins broncodebeheer	40
Figuur 21 - Jenkins bouwactiveerders	40
Figuur 22 - Bitbucket Jenkins service	41
Figuur 23 - Jenkins dashboard	41
Figuur 24 - Jenkins project configuratie	41
Figuur 25 - Jenkins E-mailNotificatie.....	42
Figuur 26 - Spoon timeliness fact job	42
Figuur 27 - Load timeliness stap	46
Figuur 28 - Spoon Timeliness Transformatie	47
Figuur 29 - Spoon overkoepelende timeliness job	47
Figuur 30 - Spoon extact timeliness fact data	47
Figuur 31 - Spoon load timeliness data	48
Figuur 32 - Spoon timeliness test job	48
Figuur 33 - Overkoepelende testjob.....	49
Figuur 34 - Pentaho connection wizard.....	50
Figuur 35 - Pentaho cube wizard	50
Figuur 36 - Pentaho join wizard	51
Figuur 37 - Pentaho properties wizard	51
Figuur 38 - Pentaho main dashboard	52
Figuur 39 - Liquibase Jenkins configuratie.....	55
Figuur 40 - Liquibase changelogoverzicht	55
Figuur 41 - Physical design meter forecast.....	57
Figuur 42 - Jenkins dashboard opsplitsing.....	61

1 INLEIDING

Dit document beschrijft het opzetten van een warehouseomgeving om de levering van de diensten van SmartInstall te verbeteren. Een warehouseomgeving bestaat uit een databasemanagementsysteem (DBMS), een extract, transform & loadmodule (ETL-module) en een business intelligence (BI) tool. In de eerste plaats worden de aanleiding en probleemstelling in kaart gebracht, gevolgd door een beschrijving van de gekozen ontwikkelmethodiek Scrum. Vervolgens wordt de opstartfase beschreven, waarin kort de theorie van The Kimball Group, een groep van senior consultants die gespecialiseerd is in het ontwerpen van effectieve datawarehouses, en het proces van de totstandkoming van de pakketselectie worden besproken. In de pakketselectie komen diverse softwarepakketten aan bod voor DBMS, ETL en BI. Deze pakketten worden afgewogen tegen een aantal criteria en in detail besproken.

Het bouwen van de warehouseomgeving is opgedeeld in drie uitvoeringsprints. In de uitvoeringsprints is de warehouseomgeving gebouwd. De uitvoeringsprints zijn per sprint beschreven. Per sprint worden alle aspecten van de sprintcyclus beschreven, van requirements tot testen. In de eerste uitvoerende sprint is het prototype tot stand gekomen dat gebruikt is om meer requirements omtrent de warehouseoplossing te achterhalen.

In de tweede sprint wordt beschreven hoe het aanpassen van de ETL-jobs ervoor heeft gezorgd dat de transformaties te testen zijn middels testjobs. Ook wordt in deze sprint besproken hoe versiebeheer is toegepast in de ETL-module en hoe de Jenkins-server is opgezet en waar deze voor dient. Verder worden de opzet en configuratie van de Pentaho BI-suite uitgebreid besproken.

De derde sprint staat in het teken van afronding en verbetering van de warehouseomgeving en automatisering van testen. Tevens wordt er beschreven hoe de Liquibase-technologie is geïntegreerd ten behoeve van het in versiebeheer zetten van de database. Tot slot wordt er evaluerend teruggekeken op de uitvoer van afstudeerperiode. Het tussentijdse assessment en de bedrijfsevaluatie zijn in te zien in bijlage I en J.

2 SMARTINSTALL

SmartInstall is een relatief jong bedrijf dat is opgericht in 2011. Het bedrijf houdt zich hoofdzakelijk bezig met de grootschalige uitrol van slimme meters. Slimme meters vervangen de huidige meters voor elektriciteit en gas. Deze digitale meter meet net als de huidige traditionele meter het energieverbruik in huis. Het verschil tussen de slimme meter en de traditionele meter is dat een slimme meter automatisch meterstanden verstuurt naar de netbeheerder en maakt de meter het mogelijk om zelf het energieverbruik in te zien [1]. Voor de uitrol van de slimme meters biedt SmartInstall verschillende diensten aan om netbeheerders en installatiebedrijven te ondersteunen. Het bedrijf wordt geleid door een directieteam van drie werkgevers met veel ervaring op het gebied van informatietechnologie (IT) en de nutssector.

De Europese Unie heeft besloten dat er in het jaar 2020 bij 80 procent van de huishouden een slimme meter aanwezig moet zijn [2]. SmartInstall heeft hierop ingespeeld door diensten aan te bieden die het proces van de uitrol ondersteunen en eenvoudiger maken. Daarnaast houdt SmartInstall zich bezig met opdrachten van onder andere de plaatsing van gas- en watermeters, kaifa-projecten en kwaliteitsinspecties. De diensten die SmartInstall aanbiedt om bovengenoemde opdrachten te realiseren worden hieronder beschreven.

Monteurs opleiden

SmartInstall selecteert lokale monteurs uit de regio waar de slimme meters worden geplaatst om de plaatsing uit te voeren. Om de slimme meters te mogen plaatsen, dienen monteurs te beschikken over bedrijfsvoering elektrische installaties- (BEI) en veiligheidsinstructie aardgas (VIAG) certificatie [3]. Indien monteurs niet over de benodigde certificaten beschikken, leidt SmartInstall deze monteurs op met een BEI- en VIAG-certificatietraining. Na de opleiding zijn de monteurs in het bezit van de certificaten en kunnen ze aan de slag met de VOP-M7 of VP-metervaarigheden voor de plaatsing van de slimme meter.

Planning

Met behulp van het intern ontwikkelde SmartWorkplatform zorgt SmartInstall voor een efficiënte, geautomatiseerde planning van de uit te voeren werkzaamheden. Het platform is inzetbaar voor diverse installatiewerkzaamheden. De efficiënte planning zorgt ervoor dat monteurs zo min mogelijk reistijd op een dag hebben, waardoor ze meer meters op een dag kunnen plaatsen. Consumenten kunnen met het SmartWorkplatform zelf hun afspraak beheren en verzetten. De netbeheerders hebben te allen tijde online een real-time overzicht van de planning en voortgang.

Uitvoering & Datamanagement

Tijdens de uitvoering van de installatiewerkzaamheden worden de monteurs ondersteund met een Android-app, genaamd SmartApp. De monteur wordt met behulp van SmartApp stap voor stap door het proces van de plaatsing geleid en voert gegevens van de installatiewerkzaamheden in SmartApp in. Alle gegevens worden direct gevalideerd en verstuurd naar SmartWork. SmartApp geeft ook de agenda en routes van de monteur per dag weer.

Logistiek beheer

Naast de uitvoering verzorgt SmartInstall ook het logistiek beheer van de te plaatsen en geplaatste meters. Het SmartWorkplatform houdt bij hoeveel meters en welk type meters er in voorraad zijn en hoeveel er in bezit van monteurs zijn voor de plaatsing. Ook houdt het bij waar en wanneer een meter geplaatst is, zodat deze te allen tijde traceerbaar is.

Rapporteren

SmartInstall rapporteert diverse werkzaamheden aan de netbeheerders. Deze rapportages betreffen onder andere de projectvoortgang, monteur productiviteit en logistiekoverzichten. Op basis van deze rapportages worden de netbeheerders op de hoogte gesteld van de voortgang van de werkzaamheden en kunnen ze monteurs bij- en aansturen in hun werkzaamheden.

2.1 SMARTWORK

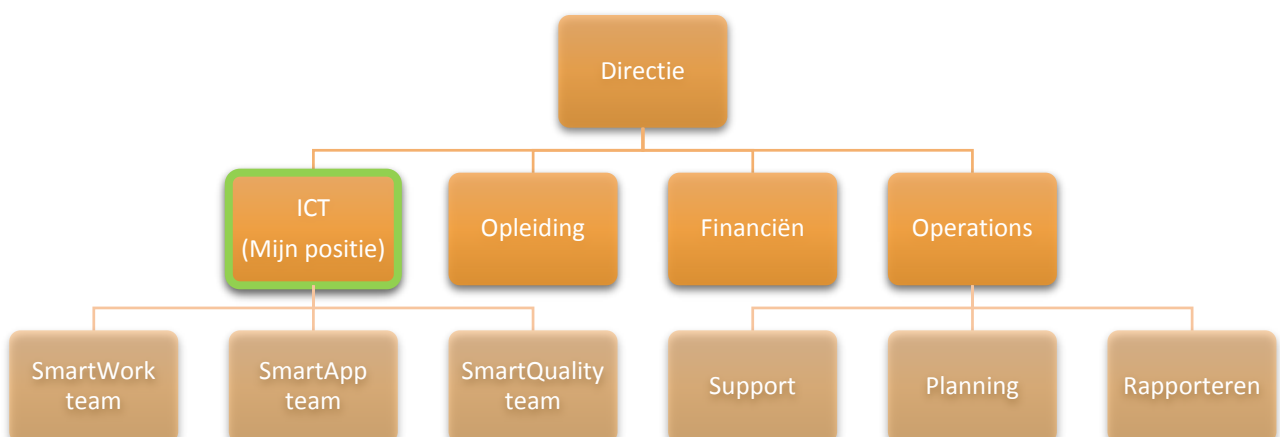
Om de eerder genoemde diensten te kunnen leveren heeft SmartInstall een innovatief ICT-platform opgezet en op de markt gebracht waar tot op heden veel belangstelling voor is. SmartInstall stelt het platform, dat ondersteunend is voor de diensten, ter beschikking aan de klanten. Het platform betreft SmartWork 3.0.

Zoals de naam doet suggereren, is SmartWork 3.0 de derde iteratie van het platform. SmartWork 3.0 is ontwikkeld in het Spring-framework, in combinatie met AngularJS voor de front-end. Voor de planningsmodule is de opensource OptaPlanner-technologie toegepast.

De kern van SmartWork 3.0 is om de capaciteit automatisch in te plannen en deze planning continu te optimaliseren. Ook verwerkt en controleert SmartWork 3.0 alle binnenkomende data die geregistreerd wordt met SmartApp en functioneert het als tool voor het logistieke beheer. Momenteel wordt het platform uitgebreid zodat het inzetbaar is voor een bredere selectie van branches.

2.2 ORGANISATIESTRUCTUUR

SmartInstall is een kleine organisatie, bestaande uit drie directeuren en veertien werknemers. Binnen het bedrijf is geen sprake van een strenge hiërarchie. De werknemers en directie zijn altijd beschikbaar en werken nauw samen om de verschillende afdelingen operationeel te houden. Figuur 1 geeft een organigram van SmartInstall weer, waarin de verschillende afdelingen afgebeeld zijn.



FIGUUR 1 - ORGANIGRAM SMARTINSTALL

De directie bestaat uit drie personen met ieder één of twee afdeling(en) onder zich. De afdelingen ICT en Financiën hebben ieder een directeur en de afdelingen Opleiding en Operations hebben samen één directeur. De directie houdt zich, naast de leidinggevende functie, vooral bezig met het verkrijgen van projecten en opdrachten en het werven van nieuwe klanten. Tussen de directie en de afdelingen zitten geen tussenpersonen. Alle werknemers van de verschillende afdelingen leggen hun verantwoording af aan de directie.

De afdeling ICT ontwikkelt en beheert alle intern ontwikkelde software. SmartWork betreft de administratie- en planningsapplicatie, SmartApp de Android-app die monteurs in het veld gebruiken en SmartQuality, een softwarepakket ter ondersteuning van kwaliteitsinspecties dat momenteel nog in ontwikkeling is. SmartQuality bestaat uiteindelijk uit een back-endserver en een Android-app. Voor elk van de eerdergenoemde producten is een ontwikkelingsteam gevormd, bestaande uit drie tot vijf personen. Bij de planning van een sprint kunnen ontwikkelaars van team wisselen, afhankelijk van de hoeveelheid werk en focus van de producten. De teams zijn vernoemd naar de producten zoals weergegeven in Figuur 1. Gedurende mijn afstudeeropdracht ben ik werkzaam op de ICT-afdeling.

De afdeling Operations zorgt voor de planning, het support voor de monteurs en klanten en de rapportages. Voor deze werkzaamheden maakt de afdeling gebruik van SmartWork 3.0. Het grootste gedeelte van de planningswerkzaamheden is geautomatiseerd met het SmartWorkstelsel. De afdeling biedt nu voornamelijk hulp aan klanten die hun afspraak willen afzeggen en controleert de planning op bijzonderheden.

De afdeling Opleidingen houdt zich vier keer per jaar vier weken lang bezig met het opleiden van lokale monteurs. In de overige tijd is deze afdeling non-actief. De afdeling financiën is verantwoordelijk voor alle financiële zaken en maakt verder geen gebruik van de intern ontwikkelde softwarepakketten.

3 OPDRACHTOMSCHRIJVING

3.1 AANLEIDING & PROBLEEMSTELLING

De diensten die SmartInstall aanbiedt zijn onder andere analyse van de installatiewerkzaamheden en rapportage over diverse gegevens. Met behulp van deze analyses en rapportages is het mogelijk om de installatiewerkzaamheden en het uitvoeringsproces aan te sturen en te verbeteren. Ook worden de rapporten als communicatiemiddel gebruikt voor de netbeheerders met betrekking tot de voortgang en knelpunten.

SmartWork 3.0 is nog niet voorzien van een degelijke oplossing voor de analyses en rapportages. Alle analyses en rapportages worden momenteel gemaakt met behulp van MySQL Workbench en Microsoft Excel. Naast een aantal standaardrapportages moet SmartInstall vaak ad-hocanalyses en rapportages maken op verzoek van de netbeheerders. Deze ad-hocverzoeken kunnen wekelijks worden gevraagd en de gewenste informatie in een dergelijk verzoek is afwisselend. Door een gebrek aan een degelijke oplossing kost het SmartInstall veel tijd en dus veel geld om deze ad-hocverzoeken te verwerken in MySQL Workbench en Microsoft Excel. Ook hebben de netbeheerders de wens om zelf analyses uit te kunnen voeren en statistieken te kunnen inzien, wat op dit moment niet mogelijk is.

Er is vraag naar een platform om data te kunnen analyseren en rapporten te kunnen genereren. De oplossing dient aan te sluiten op het huidige softwarepakket. Men wil onder andere de efficiëntie en effectiviteit van monteurs kunnen bijhouden, voortgangsstatistieken kunnen inzien en factuurrapportages kunnen maken. Het platform dient met dynamische, afwisselende data en variabelen om te kunnen gaan en te functioneren als een datawarehouse. SmartInstall wil uiteindelijk gebruikers van het softwarepakket de mogelijkheid bieden om zelf analyses uit te voeren. Voor SmartWork 2.0, de voorganger van SmartWork 3.0, is ook een rapportageomgeving gemaakt. Deze bevatte een database met dezelfde structuur als de database van SmartWork 2.0. Dagelijks werden de data van SmartWork 2.0 en de rapportageomgeving gesynchroniseerd. Naast het feit dat deze synchronisatie zes uur duurde, was de performance van queries ook niet goed. Gemiddeld moesten gebruikers tien tot vijftien minuten wachten voordat er een rapport getoond werd. De complexiteit van de database zorgde ervoor dat gebruikers slechts met veel moeite nieuwe rapporten konden maken of ad-hocvragen konden beantwoorden. Om deze redenen wil de opdrachtgever een warehouseomgeving realiseren. Om aan de vraag te kunnen voldoen wordt er een datawarehouse met bijbehorende BI-omgeving opgezet. Het platform wordt in eerste instantie alleen intern gebruikt, maar dient later ook extern gebruikt te kunnen worden.

De realisatie van het warehouseplatform bespaart SmartInstall veel tijd en kostbare resources. Per sprint van twee weken wordt er gemiddeld aan twee rapportages gewerkt. Dit betreft het wijzigen van bestaande of het implementeren van nieuwe rapportages. Het wijzigen of maken van rapportages kost SmartInstall per rapportage gemiddeld veertig manuren, oftewel tachtig manuren per sprint, wat ongeveer €1200 per twee weken is. Naast een geldbesparing kan SmartInstall de kostbare manuren beter besteden aan de ontwikkeling van andere projecten of de verdere ontwikkeling van SmartWork 3.0.

Met het realiseren van een warehouseplatform kunnen netbeheerders zelf wijzigingen in de rapporten aanmaken of zelf nieuwe rapporten maken op basis van in de warehouse aanwezige data. Indien er informatie ontbreekt, blijkt uit ervaring van derden dat de implementatie van de nieuwe informatie ongeveer 24 manuren kost. Uitgaand van de meest extreme situatie moet er per sprint nieuwe informatie geïmplementeerd worden. Dit houdt in dat er per maand 48 manuren worden besteed aan het warehouseplatform, in plaats van de 160 manuren in de huidige situatie. Uitgedrukt in financiële kosten is dit minimaal een besparing van $(160 * 15) - (48 * 15) = €1680$ per maand. Indien er geen nieuwe informatie geïmplementeerd dient te worden en de netbeheerders zelf de wijzigingen kunnen uitvoeren kunnen er meer manuren en geld bespaard worden.

3.2 DOELSTELLING

De doelstelling van de afstudeeropdracht luidt als volgt: 'Het verbeteren van het analyse- en rapportageproces met het opzetten van een data warehouseplatform'. Deze doelstelling wordt gerealiseerd met behulp van onderstaande deeldoelstellingen:

- **Onderzoek**
 - requirementsrapport opleveren;
 - selectie van bestaande oplossingen maken;
 - adviseren over bestaande oplossingen.
- **Ontwerpen en bouwen**
 - gegevensmodel SmartWork 3.0 beschrijven;
 - datawarehousedesign beschrijven;
 - datawarehousedesign implementeren;
 - ETL-module opzetten;
 - prototype warehouseplatform opleveren;
 - bètaversie warehouseplatform opleveren.

3.3 RESULTAAT & DELIVERABLES

Het resultaat van de afstudeeropdracht betreft een bètaversie van het warehouseplatform in een stagingomgeving. Indien succesvol gerealiseerd bespaart dit op zowel korte als lange termijn tijd, manuren en geld. SmartInstall kan bij de realisatie beter aan de ad-hocvragen voldoen en heeft de mogelijkheid om het softwarepakket uit te breiden en de rapportagedienst beter te leveren.

Bij de afronding van het project worden onderstaande producten opgeleverd:

- pakketselectie;
- requirementsrapport;
- gegevensmodel;
- databasedesign;
- database datawarehouse;
- prototype warehouseplatform;
- bètaversie warehouseplatform.

4 AANPAK

4.1 SCRUM METHODIEK

Tijdens de ontwikkeling van de warehouseomgeving is de Scrummethodiek toegepast. Deze methode wordt toegepast in elk project binnen SmartInstall en was ook een goede toepassing voor de afstudeeropdracht. Scrum is een framework voor Agile-development en kan worden ingezet om in teamverband op een effectieve, flexibele manier software te ontwikkelen [4]. Een team bouwt een product in een opeenvolging van iteraties, genaamd sprints. Een sprint duurt gemiddeld twee tot vier weken. Bij de afronding van een sprint wordt er een stuk werkende software en de eventuele bijbehorende tussenproducten opgeleverd. Het increment tussen twee opeenvolgende sprints zorgt voor een duidelijk overzicht van de voortgang van een project. Hieronder staan de voordelen zoals vermeld door Prowareness [4], de voornaamste redenen waarom SmartInstall de Scrummethodiek in elk project toepast:

- Scrum verhoogt de effectiviteit van het team.
- Scrum zorgt ervoor dat er iedere twee tot vier weken een stuk werkende software kan worden opgeleverd.
- Scrum zorgt voor inzicht in de voortgang van het project.
- Scrum zorgt ervoor dat enkel wordt gebouwd wat er nodig is.

Bij de start van het afstudeerproject bestond er alleen een algemeen idee over de oplossingsrichting voor het probleem zoals beschreven in hoofdstuk 3.1. Echter, de opdrachtgevers en de stakeholders hadden geen beeld van het uiteindelijke product. In de globale planning waren alleen de hoofdlijnen van het project in kaart gebracht. Daarnaast kon het door de verdere ontwikkeling van SmartWork 3.0 voorkomen dat de sourcedatabase zou wijzigen tijdens de ontwikkeling van de warehouseomgeving. Er moest flexibel kunnen worden omgegaan met dergelijke wijzigingen, zodat de warehouse voor de release van de SmartWork 3.0-wijziging aangepast kon worden. Daarom was een watervalmethodiek geen geschikte toepassing voor het project. Doordat de stakeholders niet wisten wat het eindproduct moest zijn en welke functionaliteiten het moest ondersteunen was er veel contact met hen nodig. Ook moesten de mogelijke oplossingen aan de stakeholders getoond worden. Het gehele project was op die manier tegelijkertijd een leerproces voor mij en de stakeholders van SmartInstall. Het product diende eerst te worden gevormd zodat de stakeholders wisten wat ze konden verwachten, wat de mogelijkheden ervan waren en hoe het uiteindelijke product eruit moest zien. Scrum leent zich goed voor dergelijke situaties, want binnen Scrum is contact met de stakeholders van groot belang, wat zorgt voor snelle feedback van de stakeholders. Met behulp van de dagelijkse stand-upmeetings waren de stakeholders te allen tijde op de hoogte van de voortgang van het project en met behulp van de sprint retrospective konden we het leermoment toepassen in de aankomende sprint. Tevens konden de stakeholders ook tijdig bijsturen indien het project niet de gewenste richting inging.

Voor het afstudeerproject functioneerde de directeur van financiën als de product owner. Dit hield in dat hij de prioriteiten toewees aan de user stories en op die manier bepaalde welke user stories eerst opgepakt zouden worden. Ook is binnen SmartInstall de product owner degene die onder andere requirements accepteert of afwijst. Alle user stories staan in een productbacklog. De user story die bij het invullen van een sprint bovenaan staat heeft de hoogste prioriteit en de user story die onderaan de backlog staat heeft de laagste prioriteit. Aan het eind van de sprint kijkt de product owner of de user story naar wens is geïmplementeerd en accepteert hij de gemaakte functionaliteiten met behulp van een acceptatietest. Indien de implementatie van een user story wordt afgewezen beschrijft de product owner de reden van afwijzing en wordt de user story weer in de productbacklog geplaatst.

Backlog 20 issues	
↑ SMAR-11 SH11: Als gebruiker kan ik de filters voor een ad-hoc rapportage opslaan zodat ik deze op een later tijdstip kan hergebruiken	
↑ SMAR-14 SH14: Als gebruiker kan ik de filters en instellingen van visualisaties opslaan zodat ik deze op een later tijdstip kan hergebruiken	
↑ SMAR-8 SH8: Als beheerder kan ik bepalen naar wie de automatisch gegenereerde rapporten gemaïld worden zodat ik flexibel kan inspelen op de wensen van de gebruikers	
✓ ↑ SMAR-41 Liquibase implementeren	
↑ SMAR-36 SH28: Als beheerder kan ik per feiten tabel zien of de uitvoering van het ETL proces succesvol is verlopen zodat ik tijdig de fouten kan opsporen	
↑ SMAR-37 SH29: Als gebruiker heb ik van elk type meter inzicht in de meest recente schatting van het aantal benodigde meters voor de aankomende periode	
↑ SMAR-38 SH30: Als gebruiker kan ik de schatting van het aantal benodigde meters per regio en per project achterhalen zodat ik de project en regio's kan toewijzen	
✓ ↑ SMAR-42 Jenkins ETL job opsplitsen	
↑ SMAR-39 SH31: Als gebruiker kan ik de provincie van een locatie achterhalen zodat ik weet waarom een regio niet bepaald kan worden	
↑ SMAR-40 SH32: Als gebruiker kan ik het land van een locatie achterhalen zodat ik weet waarom een provincie niet bepaald kan worden	
✓ ↑ SMAR-20 BI server export configureren	
↑ SMAR-13 SH13: Als gebruiker kan ik data visualisaties exporteren in JPEG formaat zodat ik deze buiten het platform kan gebruiken	
↑ SMAR-7 SH7: Als beheerder kan ik de scheduling van de automatisch gegenereerde rapporten beheren zodat ik kan inspelen op de wensen van de gebruikers	
↓ SMAR-32 SH27: Als gebruiker wil ik de geregistreerde telwerken kunnen inzien en of deze binnen de gedefinieerde marge valt zodat ik eventueel correcties kan aanbrengen	
↓ SMAR-29 SH24: Als gebruiker wil ik per project, per dag en per monteur kunnen inzien hoeveel correcties op de geregistreerde werkzaamheden zijn aangebracht	

FIGUUR 2 - JIRA PRODUCT BACKLOG

Ter ondersteuning van de Scrummethodiek wordt gebruikgemaakt van Atlassian Jira software. Met de Jira tool kunnen user stories en sprints worden aangemaakt, sprints worden ingepland en taken worden toegewezen en verdeeld over een team [5]. Alle user stories uit het requirementsdocument (bijlage C) zijn verwerkt in Jira en staan in de Jira backlog. Figuur 2 geeft de backlog weer zoals die er in Jira uitziet. De backlog is te allen tijde geprioriteerd volgens eerder genoemde methode. Tijdens het maken en plannen van een sprint worden de bovenste user stories van de backlog in de sprint geplaatst. De user stories behouden hun prioriteit in de sprint, zodat de belangrijkste user story eerst wordt opgepakt.

Na het plannen en invullen van de sprint wordt per issue een gedefinieerde flow gevolgd. De flow voor het afstudeerproject is als volgt:

- To do → In progress → Acceptance test → Done

Issues in 'to do' zijn de issues die gerealiseerd moeten worden voor het einde van de sprint. Zodra de werkzaamheden aan een van deze issues beginnen komt het issue in 'in progress' te staan. Bij het beëindigen van de werkzaamheden wordt een issue in 'acceptance test' geplaatst. Issues in acceptance test kunnen terug naar in progress of door naar 'done', afhankelijk van het wel of niet accepteren van het issue tijdens de acceptatietest. Een voorbeeld van hoe de flow er in Jira uitziet staat weergegeven in figuur 3.

To Do	In Progress	Acceptance Test	Done
↑ SMAR-37 SH29: Als gebruiker heb ik van elk type meter inzicht in de meest recente schatting van het aantal benodigde meters voor de aankomende periode ↑ SMAR-38 SH30: Als gebruiker kan ik de schatting van het aantal benodigde meters per regio en per project achterhalen zodat ik de project en regio's kan toewijzen ↑ SMAR-13 SH13: Als gebruiker kan ik data visualisaties exporteren in JPEG formaat zodat ik deze buiten het platform kan gebruiken	↑ SMAR-36 SH28: Als beheerder kan ik per feiten tabel zien of de uitvoering van het ETL proces succesvol is verlopen zodat ik tijdig de fouten kan opsporen ✓ ↑ SMAR-42 Jenkins ETL job opsplitsen	↑ SMAR-39 SH31: Als gebruiker kan ik de provincie van een locatie achterhalen zodat ik weet waarom een regio niet bepaald kan worden ↑ SMAR-40 SH32: Als gebruiker kan ik het land van een locatie achterhalen zodat ik weet waarom een provincie niet bepaald kan worden	✓ ↑ SMAR-41 Liquibase implementeren ↑ SMAR-8 SH8: Als beheerder kan ik bepalen naar wie de automatisch gegenereerde rapporten gemaïld worden zodat ik flexibel kan inspelen op de wensen van de gebruikers ↑ SMAR-14 SH14: Als gebruiker kan ik de filters en instellingen van visualisaties opslaan zodat ik deze op een later tijdstip kan hergebruiken

FIGUUR 3 - JIRA SPRINT FLOW

4.2 RISICO'S

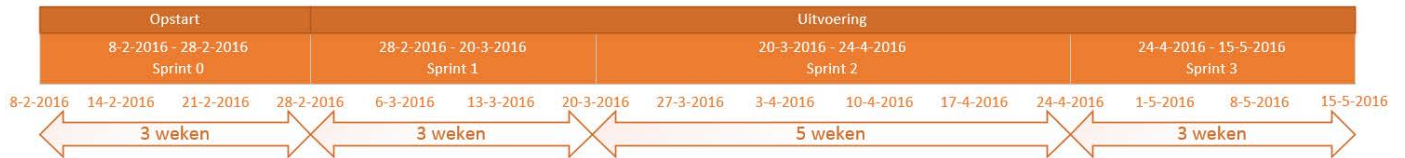
Om te voorkomen dat de voortgang van het project wordt gehinderd door onvoorziene omstandigheden is er voorafgaand aan de werkzaamheden een risicoanalyse opgesteld. Per risico is aangegeven hoe groot de kans is dat het risico zich voordoet, wat de impact is van het risico en wat de maatregelen zijn om het risico te voorkomen.

Risico	Kans	Impact	Kans reductie	Impact reductie/Alternatief
Hardware van development omgeving gaat onverwacht kapot	L	H	Development PC niet verplaatsen voor demo's en dergelijke, maar gebruik maken van remote desktop	Image maken van de development omgeving zodat de development omgeving eenvoudig kan worden opgezet op een reserve PC
Verliezen van project bestanden door falen van hardware	L	H	Het project onder versiebeheer zetten	Tijdig documenteren van proces en oplossingen op confluence
Slechte communicatie met de product owner waardoor niet het gewenste gemaakt wordt	L	M	Het houden van dagelijkse scrum meetings. Tijdens de meetings kunnen onduidelijke requirements helder worden gemaakt	Goed gebruik maken van Jira software voor het managen van het proces en het bijhouden van user stories
Scope creep. Ongeïdentificeerde requirements tijdens de sprint review en planning komen in de sprint naar boven en bedreigen de tijdigheid van het project	H	M	Afbakening van de scope voorafgaand aan de werkzaamheden	Sprint planning voor het bepalen van de scope van de aankomende sprint
De afstudeerder heeft te weinig kennis voor het realiseren van een warehouse omgeving	M	H	Kennis opdoen door middel van het lezen van vak boeken	Tijdig aan de bel trekken en hulp vragen aan begeleiders of externe partijen
Opgestelde sprint planning heeft gebreken	M	H	Samen met de product owner realistische inschattingen maken voor de te realiseren functionaliteiten in de backlog	User stories met de hoogste prioriteit (boven aan de backlog) als eerst realiseren

TABEL 1 - RISICO ANALYSE

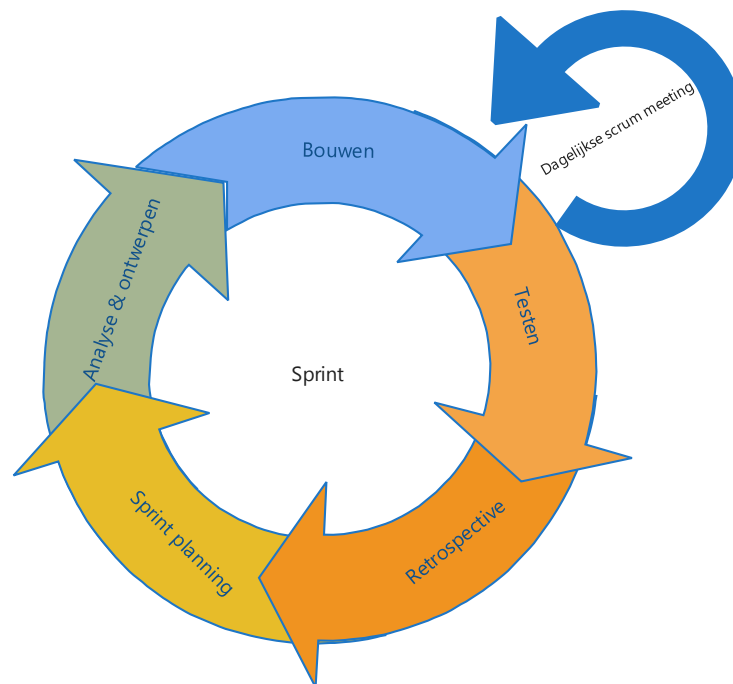
4.3 SPRINT PLANNING & UITVOER

Voorafgaand aan de afstudeerperiode is er een globale planning opgesteld voor het project. Deze is terug te vinden in bijlage A. De globale planning bestaat uit een opstartfase, sprint 0, en drie uitvoeringsprints. In totaal bedraagt de gehele projectplanning veertien weken. In figuur 4 is de tijdverdeling van het project visueel weergegeven.



FIGUUR 4 - TIJDVERDELING

Voorafgaand aan de uitvoeringsprints vindt een sprint planningmeeting plaats. In deze meeting wordt bepaald welke user stories verwerkt worden. Met een geprioriteerde product backlog worden de user stories die bovenaan de product backlog staan in de sprint geplaatst. Indien de tijd van de sprint naar schatting gevuld is wordt de sprint definitief gemaakt en kunnen de werkzaamheden gestart worden. In elke sprint worden dezelfde fases doorlopen, namelijk planning, analyse, ontwerpen, bouwen en testen. Figuur 5 geeft de cyclus van een sprint weer.



FIGUUR 5 - SPRINT CYCLUS

5 SPRINT 0 - ONDERZOEK EN GLOBALE ANALYSE

Sprint 0, de opstartfase, stond in het teken van kennis opdoen over datawarehousing, een pakketselectie maken en de huidige architectuur van het SmartWorkplatform in kaart brengen. Bij de afsluiting van deze sprint werd duidelijk welke softwarepakketten voor de warehouseomgeving gebruikt worden en waarom deze pakketten zijn gekozen.

5.1 THE KIMBALL GROUP

Bij de opstart van het project had ik geen tot weinig kennis van datawarehousing. Daarom leek het mij verstandig om voorafgaand aan alle werkzaamheden kennis te vergaren door me te oriënteren op het onderwerp en boeken te lezen. Tijdens de oriëntatie kwam ik al snel bij verschillende boeken en artikelen terecht. Een vaak genoemde naam in de artikelen is The Kimball Group. Vanwege het feit dat deze naam steeds terugkeerde in de diverse informatiebronnen besloot ik mij erin te gaan verdiepen.



FIGUUR 6 - THE KIMBALL GROUP

The Kimball Group is een groep van senior consultants die gespecialiseerd is in het ontwerpen van effectieve datawarehouses. Met meer dan dertig jaar ervaring in het veld is de groep in het bezit van veel kennis en knowhow [6]. Deze heeft zij gedeeld door verschillende boeken te schrijven. Binnen het warehousevakgebied worden de methodieken van The Kimball Group gezien als 'best-practice'. Alle verwijzingen naar deze groep, de status van 'best-practice' en het feit dat zij meer dan dertig jaar ervaring hebben, maakten duidelijk dat de boeken konden worden gebruikt als fundering voor de kennis met betrekking tot het datawarehouse project.

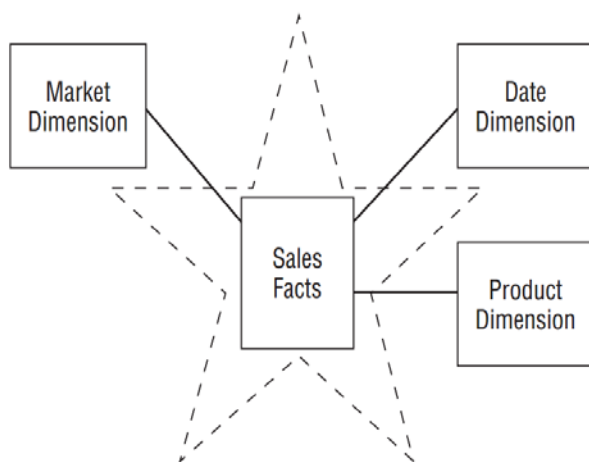
Om de datawarehouse te begrijpen en te kunnen ontwikkelen is het boek 'The datawarehouse toolkit' [7] gelezen en om de ETL-modules te kunnen bouwen is het boek 'The datawarehouse ETL toolkit' gebruikt [8]. Voor de BI-module heeft The Kimball Group alleen een boek geschreven over Microsoft BI. Op het moment van oriënteren wist ik nog niet welk pakket ik zou gaan gebruiken voor de BI-module, dus besloot ik dit boek niet te lezen. De andere twee boeken bespreken softwareonafhankelijke technieken die altijd toepasbaar zijn, waardoor deze wel bruikbaar waren.

5.1.1 DIMENSIONAL MODELING

Door 'The datawarehouse toolkit' te lezen vergaarde ik kennis over dimensional modeling, het ontwerpen van een warehousedatabase. De technieken van The Kimball Group worden gezien als best-practice binnen dit vakgebied. Dit was de reden om haar technieken toe te passen op het project.

De structuur van een warehousedatabase verschilt sterk van die van de normalized database die bij een operationeel systeem hoort. Een normalized database is gemodelleerd in derde normaalvorm (3NF). Een normalized model verwijderd redundantie door data op te splitsen in veel discrete entiteiten, waarvan ieder een relationele tabel vormt. Dergelijke normalized modellen kunnen uitlopen tot een complex, onoverzichtelijk spinnenwebdiagram. Derde normaalvormstructuren zijn nuttig voor operationele processen omdat een insert of een update maar één plaats van de database raakt. Ze zijn echter lastig te begrijpen en te ingewikkeld voor BI-queries. Gebruikers kunnen normalized structuren niet of lastig navigeren. Doordat ze de samenhang tussen de tabellen niet goed begrijpen schrijven gebruikers vaak queries die minder goed performen en wegens de complexiteit is de performance van normale queries ook vaak niet optimaal. Om eerder genoemde redenen is het nuttig om een dimensional model op te stellen voor een warehouseomgeving.

Een dimensional model heeft een simpele structuur die eenvoudig te begrijpen is voor gebruikers en biedt hoge query performance. Ondanks de simpele structuur bevat een dimensional model dezelfde data als een normalized model. Het boek spreekt bij dimensional modeling over starschema's, omdat het model op een ster lijkt. Figuur 7 geeft weer hoe een starschema er volgens The Kimball Group uit ziet.



FIGUUR 7 - STARSHEMA

Een starschema bestaat uit een feitentabel en meerdere dimensietabellen. Er bestaan verschillende soorten feitentabellen, zoals transaction-, periodicsnapshot- of factlesstabellen. Alle soorten feitentabellen bevatten de feiten, ook wel de metingen, en de vreemde sleutels naar de dimensietabellen. De dimensietabellen bevatten alle gedetailleerde informatie omtrent het feit van een starschema.

Een feit in een feitentabel kan additief, semi-additief of non-additief zijn. Een additieve meting is het meest flexibele en bruikbare feit. Deze meting kan over alle dimensies worden opgeteld (SUM, AVG). Bij een semi-additieve meting is dit

mogelijk bij sommige dimensies, maar niet alle dimensies. Een voorbeeld van een veel voorkomende semi-additieve meting zijn balansbedragen. Bij non-additieve metingen zoals verhoudingen is het optellen van de meting over de dimensies niet mogelijk. Een goede benadering voor non-additieve metingen is om deze te berekenen op basis van additieve metingen. Verder wijst het boek erop dat null-values als vreemde sleutel in een feitentabel vermeden dienen te worden tijdens het modelleren.

Een attribuut van een dimensie betreft een van de zeven types 'slowly changing dimensions' (SCD). Binnen één dimensie kunnen meerdere SCD-types voorkomen. In tabel 2 zijn de SCD-types beschreven die tijdens de uitvoering van de afstudeeropdracht aan bod zijn gekomen.

Type	Beschrijving
Type 0: Behoud origineel	Het attribuut van de dimensie wijzigt nooit
Type 1: Overschrijven	Bij een wijziging wordt de oude waarde van het attribuut overschreven met de nieuwe waarde. Een type 1 attribuut representeert altijd de meest recente data. De geschiedenis wordt hierbij niet bewaard
Type 2: Record toevoegen	Bij een wijziging wordt er een nieuw record toegevoegd met de nieuwe attribuut waarde. Vanaf moment van wijzigen wordt het nieuwe record gebruikt bij het updaten van de feit tabel. De geschiedenis wordt met dit type bewaard. Het gebruik van type 2 eist een toevoegen van de kolom versie en datum van en datum tot kolommen voor het aanduiden van de geldigheidsperiode

TABEL 2 - SCD-TYPE BESCHRIJVINGEN

Het ontwerpproces van een dimensional model bestaat uit vier stappen. De vier stappen van het proces zijn als volgt:

1. Selecteer het business proces.
2. Bepaal de granulariteit.
3. Identificeer de dimensies.
4. Identificeer de feiten.

Een business proces is een operationele activiteit dat wordt uitgevoerd door het bedrijf of de organisatie, zoals het beheren van voorraad, het registreren van studenten voor een les, het verwerken van een verzekeringsclaim en dergelijke activiteiten. Deze business processen genereren metingen die vertaald kunnen worden naar een feit in een feitentabel. Het selecteren van een business proces zorgt ervoor dat de granulariteit kan worden bepaald. De granulariteit, ook wel de korrel, stelt vast wat één record van een feitentabel representeert. De granulariteit wordt het bindende contract voor het ontwerp. De feiten en de dimensies worden afgeleid van de bepaalde granulariteit en moeten consistent zijn met deze granulariteit. Verschillende vormen van granulariteit mogen niet worden samengevoegd in één feitentabel. De dimensies verschaffen de context van wie, wat, waar, wanneer, waarom en hoe rondom een business proces. Dimensies zijn ook wel de 'ziel' van een data warehouse en bevatten de gedetailleerde informatie van een feitentabel. De feiten zijn de metingen die resulteren uit een business proces en zijn bijna altijd numeriek.

Na de theorie rondom dimensional modeling geeft het boek een aantal voorbeelden en oefeningen om dimensional models te maken. Het leek mij verstandig om de voorbeelden te volgen en de oefeningen te doen, zodat ik zeker wist dat ik minimaal over de basisvaardigheden om een warehouse te ontwerpen beschikte.

5.1.2 ETL TOOLKIT

Het tweede boek van The Kimball Group, 'The Datawarehouse ETL toolkit', bespreekt ETL-technieken. Voor de extractie van data geeft het boek aan dat het van belang is om de source van de data goed te begrijpen en in kaart te brengen. Aangezien ik aan de ontwikkeling van SmartWork 3.0 en de achterliggende database heb gewerkt, had ik het voordeel dat ik de source al goed begreep. De inhoud van alle entiteiten en de relaties tussen de entiteiten waren voor mij al duidelijk. De scope van het project is beperkt tot de SmartWork 3.0-omgeving, waardoor het niet nodig was om andere sources te doorgronden en in kaart te brengen.

Voor de transformaties binnen het ETL-proces is het nodig om op de hoogte te zijn van de businessrules en impliciete kennis. Ook hier had ik het voordeel dat ik al bekend was met SmartInstall en daardoor alle businessrules en interne codes al kende. Een van de toegepaste methodieken uit het boek is logical datamapping als input om de ETL-jobs te kunnen bouwen. De logical datamapping is terug te vinden in bijlage E. Beide boeken boden een goede fundering om aan de slag te kunnen gaan met het warehouseproject.

5.2 KEY REQUIREMENTS

Voorafgaand aan de pakketselectie hebben er een brainstormsessie en een interview met de stakeholders plaatsgevonden. Deze sessie had als doel om de keyrequirements te eliciteren en om een goed beeld te krijgen wat er van de warehouse oplossing verwacht werd. Met een sessie en interviews konden de globale requirements worden achterhaald. De meer gedetailleerde requirements zijn achterhaald op basis van een demonstratie van het prototype in een later stadium van het afstudeerproject. Zo konden de stakeholders de mogelijkheden zien en een beeld krijgen van de uiteindelijke oplossing. Het volledige requirementsdocument is terug te vinden in bijlage C. Aan de hand van de keyrequirements was het mogelijk om de pakketselectiecriteria voor de modules op te stellen.

Nummering	Beschrijving
KXX	Key requirements die bij de eerste meeting zijn verkregen.
KXXA	Key requirements die zijn verkregen naar aanleiding van de uitwerking van de key requirements. Dit kunnen requirements zijn die in eerste instantie niet naar voren zijn gekomen of die niet genotuleerd zijn.
PXX	Requirements die zijn verkregen aan de hand van een demonstratie van het datawarehouseprototype.
PXXA	Requirements die zijn verkregen naar aanleiding van de uitwerking van de prototyperequirements. Dit kunnen requirements zijn die in eerste instantie niet naar voren zijn gekomen of niet genotuleerd zijn.
EXX	Extra requirements die zijn verkregen na de oplevering van de datawarehouseomgeving aan het einde van een sprint.
EXXA	Requirements die zijn verkregen naar aanleiding van de uitwerking van de extra requirements. Dit kunnen requirements zijn die in eerste instantie niet naar voren zijn gekomen of niet genotuleerd zijn.

TABEL 3 - REQUIREMENTNUMMERING

Om de requirements traceerbaar te houden en het moment van achterhalen vast te leggen is een nummeringsmethodiek opgesteld om op alle requirements toe te passen. De nummering van de requirement geeft aan op welk moment de requirement is verkregen en zorgt voor een degelijke traceerbaarheid in de latere hoofdstukken van het requirementsdocument. In tabel 3 staat weergegeven wat de nummering van een requirement inhoudt, waarbij 'XX' het nummer van een requirement representeert.

Na de sessie en interviews zijn de requirements uitgewerkt in een document. Na de uitwerking is een vervolgsessie gepland om de uitgewerkte requirements te bespreken. Het doel daarvan was om de requirements door de stakeholders te laten valideren en accepteren en eventueel nog bij stellen. Aan de hand van de requirementsbespreking kwamen er requirements naar boven die in de eerdere sessies nog niet aan het licht waren gekomen. Deze werden meteen uitgewerkt in het document zodat ze ook direct gevalideerd en geaccepteerd konden worden. Een overzicht van de keyrequirements is te vinden in hoofdstuk 2 van bijlage C.

5.3 PAKKETSELECTIE

Om een advies uit te kunnen brengen met betrekking tot de te gebruiken softwarepakketten voor de warehouseoplossing, is er een pakketselectie opgesteld. De pakketselectie is gemaakt voor alle drie de modules: DBMS, ETL en BI. Ondanks dat de producten van SmartInstall al gebruikmaken van een DBMS, is er besloten om ook een pakketselectie te doen voor de DBMS. De DBMS, MySQL, die bij de huidige producten zit, is gekozen wegens het bekend zijn met de DBMS. Er is toentertijd niet onderzocht of dit ook een goede keuze was. Ook was de vraag of de huidige DBMS voldoet aan de eisen voor een warehousedatabase. Om deze redenen is er besloten om ook voor de DBMS een pakketselectie op te stellen. Het volledige document van de pakketselectie is terug te vinden in bijlage B. In de pakketselectie zijn eerst de huidige en gewenste situatie geanalyseerd. Vervolgens zijn er een aantal criteria opgesteld aan de hand van de eerder verkregen requirements en technieken van het boek 'The datawarehouse ETL toolkit'. Daarna zijn er longlists gemaakt van de potentiële softwarepakketten voor alle drie de modules. Ten slotte zijn deze longlists afgewogen tegen de criteria om de shortlists te vormen.

5.3.1 ANALYSE HUIDIGE & GEWENSTE SITUATIE

SmartInstall verzorgt de planning en het datamanagement van energieleveranciers, zoals Enduris en Westland. De planning en het datamanagement zijn geautomatiseerd en gemoderniseerd door middel van de intern ontwikkelde systemen, SmartWork 3.0 en SmartApp. SmartWork 3.0 is de operationele software die alle diensten van SmartInstall ondersteunt en SmartApp is een mobiele Android-app die de monteurs ondersteunt tijdens hun werkzaamheden.

Een van de diensten die SmartInstall biedt is het rapporteren van de data richting de klanten en het analyseren van de data om de werkzaamheden aan te kunnen sturen en eventueel verbanden te leggen tussen de variabelen. Dit gebeurt door middel van queries op de transactionele database die in productie draait en door statische, geautomatiseerde rapporten die in SmartWork 3.0 zijn gebouwd. Om in te kunnen spelen op de vraag van de klanten, worden de rapporten in SmartWork 3.0 vaak aangepast en worden er regelmatig nieuwe rapporten geïmplementeerd.

Het statische kenmerk van de rapportages zorgt ervoor dat er vaak een compleet nieuw rapport geïmplementeerd dient te worden zodra een bestaande rapportagevraag wijzigt of dusdanig afwijkt, dat deze niet beantwoord kan worden met de bestaande rapportages. Van tevoren staat namelijk al vast welke specifieke vraag de rapportage beantwoordt. Het wijzigen of implementeren van rapportages kost SmartInstall kostbare IT-resource en, indien het een interne vraag betreft, veel geld.

Door bovengenoemde problemen heeft SmartInstall de wens om een oplossing te implementeren die de huidige methode van rapporteren en analyseren kan vervangen. De oplossing dient los te staan van SmartWork 3.0. De huidige rapportages moeten uit SmartWork 3.0 worden gefaseerd en worden ondersteund door de nieuwe oplossing. De oplossingsrichting die SmartInstall voor zich ziet is het implementeren van een datawarehouse. Gezien de omvang van SmartInstall speelt het budget een cruciale rol. Het maximaal te besteden budget voor de oplossing in zijn totaliteit bedraagt €1000 per maand of €12.000 per jaar.

Het volgende betreft een lijst van eisen die in overweging moeten worden genomen tijdens de pakketselectie:

- Het moet een 'bewezen technologie' zijn.
- Het moet een flexibel platform zijn dat om kan gaan met Agile-development.
- De oplossing moet op een 64-bit Linuxsysteem kunnen draaien.
- Er moet volledige documentatie aanwezig zijn voor het gekozen pakket, zodat overdracht relatief snel en eenvoudig plaats kan vinden.
- Het is bij voorkeur een opensource oplossing.

5.3.2 LONGLISTS

Met behulp van de eisen zoals genoemd in hoofdstuk 5.3.1 begon ik mijn oriëntatie op de aangeboden softwarepakketten. Daarbij werd duidelijk dat er veel pakketten op de markt zijn voor alle modules. Vooral voor de BI-module zijn veel softwarepakketten beschikbaar. Dat maakte het lastig om een selectie te maken voor de longlists.

Tijdens de oriëntatie kwam Gartner naar voren, een bedrijf dat technologische onderzoeken doet, onder andere naar datawarehousingsoftware [9]. Gartner heeft voor alle warehousemodules een Gartner Magic Quadrant opgesteld. Figuur 8 toont een Gartner Magic Quadrant voor BI-softwarepakketten. Ook beschikt de website van Gartner over communityreviews van verschillende warehousesoftwarepakketten. De reviews staan los van de Gartner Magic Quadrant.



FIGUUR 8 - BI MAGIX QUADRANT

De leveranciers van de softwarepakketten verwijzen allemaal naar hun positie volgens Gartner. Dit bevestigde dat de pakketten die genoemd zijn door Gartner een goede start zijn voor de longlist selecties. Gezien de voorkeur van opensource software was het besluit snel gemaakt om open source software in de longlist selectie op te nemen. Echter, het doel was om mij niet beperken tot alleen opensource software, zodat ik een goede selectie kon maken uit zowel commerciële als opensource software. Daarbij werd gestreefd om zes tot acht pakketten te selecteren voor de longlists, die zo veel mogelijk aan de eerder genoemde criteria voldeden. Het relatief lage budget dat beschikbaar was gesteld door SmartInstall maakte al dat veel pakketten afvielen voor de longlist. Indien duidelijk was dat een pakket boven het budget lag werd het niet meegenomen in de longlist. Tabel 4 geeft de longlist van de drie modules weer.

DBMS Product	ETL product	BI Product
Infobright	PowerCenter Informatica	Sap Web Intelligence
Greenplum	Pentaho Data Integrator	Tableau Software
Oracle database	CloverETL	Qlik BI
Microsoft SQL server	IBM InfoSphere Information Server	IBM Cognos
Teradata	Oracle Data Integrator	Pentaho Business Analytics
IBM DB2	Talend Open Studio	Birst
PostgreSQL	-	Board

TABEL 4 - LONGLISTS WAREHOUSEMODULES

5.3.3 SHORTLISTS

Om tot de shortlists te komen zijn de softwarepakketten van de longlists per module tegen de criteria afgewogen en gescoord met behulp van een selectiematrix. De volledige lijst van criteria en de beschrijvingen ervan is terug te vinden in hoofdstuk 4 van bijlage B. De score die een softwarepakket heeft gekregen voor een criterium is afhankelijk van in hoeverre een pakket voldoet aan de criteria. Wanneer een bepaalde score wordt toegekend aan een criterium is weergegeven in tabel 5. Deze toekenning van scores is toegepast op alle drie de longlists van de drie modules.

Score	Toekenning
0	Toegekend wanneer een pakket niet voldoet aan de criteria
1	Toegekend wanneer een pakket gedeeltelijk voldoet aan de criteria of wanneer er eventueel alternatieve mogelijkheden zijn
2	Toegekend wanneer een pakket volledig aan de criteria voldoet

TABEL 5 - SCORE TOEKENNING

De criteria voor de selectiematrix bestaan uit een zestal algemene criteria en zes tot tien modulespecifieke criteria. De algemene criteria zijn gebruikt voor alle drie de selectiematrices. Tabel 6 geeft de selectiematrix voor de BI-module weer. De andere twee selectiematrices zijn terug te vinden in bijlage B. De drie pakketten met de hoogste totaalscore in de selectiematrix vormen samen de shortlist voor de betreffende module. Voor de BI-module is er gekozen om het Pentaho- en het Qlik-pakket in de shortlist op te nemen. De pakketten SAP, Tableau en IBM delen de derde positie, maar zijn niet opgenomen in de shortlist. De reden hiervoor is dat deze pakketten uiteindelijk niet binnen het budget vallen of niet op CentOS draaien. Daarom zijn deze pakketten niet bruikbaar geacht voor de warehouseoplossing en bleven alleen Pentaho en Qlik over.

Met het toepassen en invullen van de selectiematrices ontstond voor alle modules een shortlist. De shortlists per module zijn weergegeven in tabel 7. Zoals is af te lezen in de tabel heeft ook de ETL-module slechts twee pakketten voor de shortlist. Het derde pakket van de top drie betreft Oracle Data Integrator. Deze scoorde hetzelfde als Pentaho Data Integrator, maar lag uiteindelijk boven het beschikbare budget. Om deze reden is besloten het Oracle-pakket niet op te nemen in de shortlist.

Software pakket	SAP	Tableau	Qlik	IBM	Pentaho	Birst	Board
Algemene criteria							
1.1 Draait op CentOS	1	0	2	1	2	0	0
1.2 Binnen budget	0	1	1	0	2	0	0
1.3 Documentatie	2	2	2	2	2	1	2
1.4 Community	2	2	2	2	2	1	1
1.5 Upgrade mogelijkheden	2	2	2	2	2	2	2
1.6 Betrouwbaarheid	2	2	2	2	2	2	2
BI criteria							
4.1 Web-based client	2	2	2	2	2	2	2
4.2 Security	1	2	2	2	2	2	1
4.3 Filteren / Drill-down	2	2	2	2	2	2	2
4.4 Visualisaties	2	2	2	2	2	2	2
4.5 Exporteren	2	2	2	2	2	2	2
4.6 Rapporten / Grafieken opslaan	2	1	1	2	1	1	1
4.7 Gebruiksvriendelijkheid	2	2	2	1	2	2	2
4.8 Dashboard	1	2	1	1	1	2	2
4.9 DBMS Ondersteuning	2	2	2	2	2	2	2
4.10 Scheduling	2	1	2	2	2	1	1
Totaal	27	27	29	27	30	24	24

TABEL 6 - BI SELECTIE MATRIX

Alle pakketten uit de shortlists zijn kort beschreven en benaderd met betrekking tot gebruikersgemak, betrouwbaarheid, flexibiliteit, licenties en externe tools. Ook is gekeken naar de mogelijkheid om te kunnen combineren met eerder geadviseerde pakketten. Met behulp van deze methodes en nader onderzoek kon een advies opgesteld worden voor de te gebruiken softwarepakketten.

DBMS Product	ETL product	BI Product
Oracle database	Pentaho Data Integrator	Qlik BI
IBM DB2	Talend Open Studio	Pentaho Business Analytics
PostgreSQL	-	-

TABEL 7 - SHORTLISTS WAREHOUSEMODULES

5.4 ADVIES SOFTWARE PAKETTEN

Met de selectiematrices en het onderzoek kon een advies worden opgesteld voor de te gebruiken softwarepakketten. De conclusie voor de DBMS-selectie is dat PostgreSQL een volwaardige keus is voor de datawarehouseoplossing binnen SmartInstall. Rekening houdend met het te besteden budget valt het Oracle-pakket af. IBM DB2 is binnen het budget te houden, maar leek dan te gelimiteerd. Bovendien betreffen de data binnen SmartInstall een kleine hoeveelheid en komt deze, gebaseerd op de afgelopen vijf jaar, de aankomende tien jaar niet aan een terabyte. De kracht van IBM DB2 komt pas naar voren zodra het om terabytes aan data gaat. PostgreSQL met datawarehousing configuraties is voor SmartInstall een toekomstbestendige en flexibele oplossing.

Talend en Pentaho zijn uitgebreide ETL-tools met mogelijkheden voor extra functionaliteiten via plug-ins. Pentaho heeft als voordeel dat er geen beperking zit op het aantal developers. Beide pakketten zijn relatief jong, maar kunnen desondanks betrouwbaar worden geacht vanwege de grote en/of bekende namen die de pakketten met succes in gebruik hebben [10] [11].

Het advies voor de ETL-module is om het Pentaho-pakket te gebruiken voor de warehouseoplossing. Pentaho heeft veel handigheden die missen bij Talend, zoals de interne scheduler en de administratieconsole. De mogelijkheid om het pakket van Pentaho uit te breiden met business analysis was een voordeel ten opzichte van Talend. Uiteindelijk is hier ook voor gekozen, al stond dit wel los van de keuze voor Pentaho als ETL-pakket.

Qlik en Pentaho zijn beide intuïtief en eenvoudig in gebruik. De functionaliteiten die Qlik aanbiedt zitten echter niet in één pakket. QlikSense is de meest recente BI-oplossing die Qlik aanbiedt, maar mist veel functionaliteiten die QlikView wel heeft. De e-mail scheduler van rapporten voor QlikSense zit niet standaard in het pakket en dient tegen extra licentiekosten aangeschaft te worden. Zo zijn er meer functionaliteiten in QlikSense die achteraf aangeschaft dienen te worden, waardoor het pakket uiteindelijk buiten het budget valt. Het advies is dan ook om het Pentaho-pakket te gebruiken als BI-software voor de warehouseoplossing.

Voor zowel ETL als BI is gekozen om de software van Pentaho te gebruiken. Voor beide modules is een gratis community-editie beschikbaar. De community-edities kunnen geüpgraded worden naar de Enterprise-edities indien de functionaliteiten van de Enterprise-edities nodig blijken te zijn. Op het moment van schrijven is het nog steeds mogelijk om de warehouse oplossing te realiseren met de community-editie. Het voordeel van het feit dat beide modules van hetzelfde bedrijf afkomstig zijn is dat de aanschaf van een Enterpriselicentie voor beide modules geldt en dus een eenmalige uitgave betreft.

De geadviseerde DBMS is PostgreSQL. Dit is een gratis opensource softwarepakket dat geen extra kosten oplevert voor SmartInstall. Naast het feit dat PostgreSQL het best uit de selectie komt, is de keuze voor PostgreSQL voordelig omdat SmartInstall werknemers in dienst heeft met veel PostgreSQL-ervaring. De nodige kennis voor configuratie is dus aanwezig en de overdracht van DBMS-onderhoud kan snel plaatsvinden.

Zowel Pentaho Data Integration als Pentaho BI-suite ondersteunt PostgreSQL en beide hebben standaard meegeleverde PostgreSQL-functionaliteiten. De drie geadviseerde pakketten kunnen dus gecombineerd worden. Het uiteindelijke advies voor de softwarepakketten is weergegeven in tabel 8. Deze pakketten zijn gebruikt om het prototype en de bètaversie te bouwen.

Module	Product	Versie
DBMS	PostgreSQL	9.5.1
ETL	Pentaho Data Integration	6.0.1.0
BI	Pentaho BI Suite	6.0.1.0

TABEL 8 - GEADVISEERDE PAKKETTEN

6 SPRINT 1 - WAREHOUSEPROTOTYPE

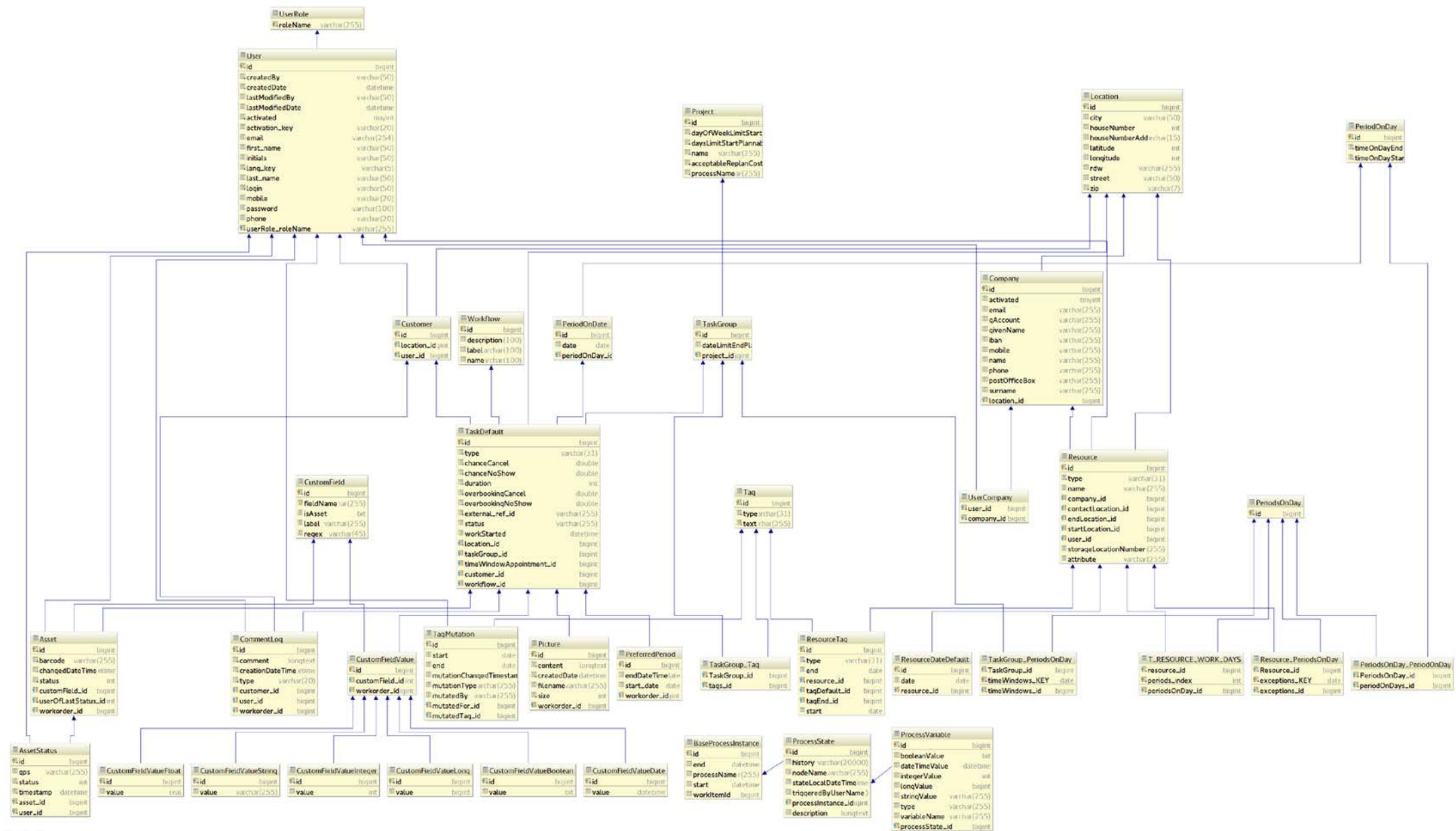
Sprint 1 stond in het teken van het bouwen van een prototype. Dat prototype is gemaakt aan de hand van de eerder opgestelde requirements. Voorafgaand aan het bouwen is de sourcedatabase in kaart gebracht en zijn er databaseontwerpen gemaakt voor de warehouse. De databaseontwerpen bestaan uit conceptual, logical en physical ontwerpen gebaseerd op de Kimball-technieken. Bij de afsluiting van de sprint is er een prototype gebouwd die op een lokale omgeving draaide.

6.1 GEGEVENSMODEL

Om een beeld te krijgen van hoe de database op dat moment in productie stond heb ik een gegevensmodel van de SmartWork 3.0-database gegenereerd met behulp van IntelliJ IDEA. Het model van de SmartWork 3.0-database is weergegeven in figuur 9. Het gegevensmodel is terug te vinden in bijlage D. Een aantal tabellen van de database zijn niet relevant voor de warehouse. Deze tabellen zijn dan ook uit het model weggelaten. Het gaat onder andere om tabellen voor Spring Batch, Liquibase, importitems, exportitems en SmartQuality-gerelateerde tabellen.

Hoewel de CustomField-types wel relevant zijn voor de warehouse, zijn ook deze weggelaten uit het model. Ze geven namelijk geen extra informatie. CustomField-types kunnen uit zes verschillende datatypes bestaan. Met de huidige implementatie houdt dat in dat er zes databasetabellen bestaan die aangeven welk datatype een CustomField is. CustomFieldValue heeft dezelfde implementatie, maar dit is niet weggelaten. In tegenstelling tot CustomField heeft CustomFieldValue een kolom genaamd 'value'. In deze kolom wordt de data van een CustomField opgeslagen. Een CustomField-tabel bevat alleen een ID, die is overgeërfd van de CustomField-tabel.

Voor alle relevante tabellen van het gegevensmodel, is per tabel een gedetailleerde beschrijving gemaakt. Deze beschrijvingen dienen om duidelijk te maken wat een tabel van de SmartWork 3.0-database inhoudt. Indien een tabel van de SmartWork 3.0-database op zichzelf geen of weinig betekenis heeft, is deze gegroepeerd met andere tabellen die gezamenlijk wel een betekenis hebben. Alle gedetailleerde beschrijvingen zijn terug te vinden in bijlage D.



FIGUUR 9 – SMARTWORK 3.0 DATABASE

6.2 ONTWERPEN DATABASE

Voor het prototype is in overleg met de product owner gekozen om de user stories rondom assets te implementeren. De betreffende user stories zijn weergegeven in tabel 10. De requirements behorende bij deze user stories zijn weergegeven in tabel 9. Uit de user stories en de requirements is af te lezen dat het de inventaris van de assets en de geplaatste assets betreft. De business processen betreffen het beheren van de metervoorraad en het controleren van meterplaatsingen. Aan de hand van de geselecteerde business processen en de bepaalde granulariteit van deze processen zijn de requirements opgesteld. Na het opstellen van de requirements kwamen er twee feiten naar voren. Dit zijn de twee feiten die gerealiseerd moesten worden. De twee feiten zijn niet gecombineerd in één zogenoemde ‘consolidated fact table’. De tabellen kunnen niet worden uitgedrukt in dezelfde granulariteit en zullen nooit samen geanalyseerd worden. De feiten dienen ieder een ander business proces en hebben andere doeleinden. De granulariteit en de requirements wijzen er op dat een inventaris per monteur beschikbaar dient te zijn. Buiten de gehele inventaris, moet een gebruiker zelf een periode kunnen selecteren, met een minimale selectie van één dag. Verder wilt een gebruiker de inventaris kunnen filteren op de status van een asset. Het overzicht van de geplaatste meters dient per monteur, per dag en/of per project getoond te worden en moet er gefilterd kunnen worden op locatie. Verder moet in het overzicht duidelijk zijn voor welke werkopdracht de asset geplaatst is. Met dit gegeven konden de dimensies voor de twee feiten bepaald worden.

Nr.	Requirement
K13	Een gebruiker moet een dagelijkse snapshot van de <u>inventaris</u> van alle meters per monteur kunnen inzien
K14	Een gebruiker moet de mogelijkheid hebben om de snapshot van de inventaris van alle meters te filteren op monteur en status
K15	Een gebruiker kan de periode van de inventaris snapshot van alle meters selecteren, waarbij de minimale selectie een dag en de maximale jaren is.
K16	Een gebruiker moet het <u>aantal meters</u> kunnen inzien die een medewerker op zijn naam heeft <u>geregistreerd</u>
K19	Een gebruiker moet een overzicht van alle <u>geplaatste meters</u> per monteur , per dag kunnen inzien
K20	Een gebruiker kan het <u>aantal meters</u> inzien die een monteur heeft <u>geplaatst</u>
K21	Een gebruiker moet een overzicht van alle geplaatste meters per project kunnen inzien
K22	Een gebruiker moet de geplaatste meters kunnen filteren op locatie
K23	Een gebruiker kan is het geplaatste meters overzicht achterhalen bij welke werkopdracht een meter geplaatst is

TABEL 9 - KEY REQUIREMENTS VOOR PROTOTYPE

ID	User story	Req. Nr.
SH3	Als gebruiker wil de inventaris van meters per dag, per medewerker kunnen inzien zodat ik weet hoeveel meters een medewerker op een dag op zijn naam heeft gescand	K13,K14 K16
SH4	Als gebruiker wil ik kunnen selecteren over welke periode de inventaris van de meters getoond moet worden zodat ik een wekelijks, maandelijks of jaarlijks overzicht heb	K15
SH5	Als gebruiker wil ik de geplaatste meters per dag, per monteur, per locatie en per werkopdracht kunnen inzien zodat ik een overzicht heb van de locaties en werkopdrachten waar een meter is geplaatst	K19,K20 K21,K22 K23

TABEL 10 - PROTOTYPE USER STORIES

6.2.1 ASSET INVENTORY FACT

Requirement K13 van tabel 10 noemt twee entiteiten die als dimensie opgesteld kunnen worden. De dikgedrukte woorden van de requirement geven aan welke woorden de entiteiten aangeven. Voor het inventaris feit zijn er dus gegevens van de monteur en de asset nodig. In requirement K14 wordt er gesproken over status en in requirement K16 over dag (datum). Een gebruiker moet de status van een asset kunnen achterhalen en de inventaris over een geselecteerde dag of periode kunnen inzien. Met deze gegevens, gebaseerd op de granulariteit, weten we welke dimensies en feiten nodig zijn. De dimensies betreffen monteur, asset en datum. Schuingedrukte onderstreepte woorden van de requirements uit tabel 10 geven de gegevens van de feiten aan en welk feit het betreft. Met deze informatie is het conceptual design zoals weergegeven in figuur 10 van het inventaris feit opgesteld.



FIGUUR 10 - CONCEPTUAL DESIGN INVENTORY FACT

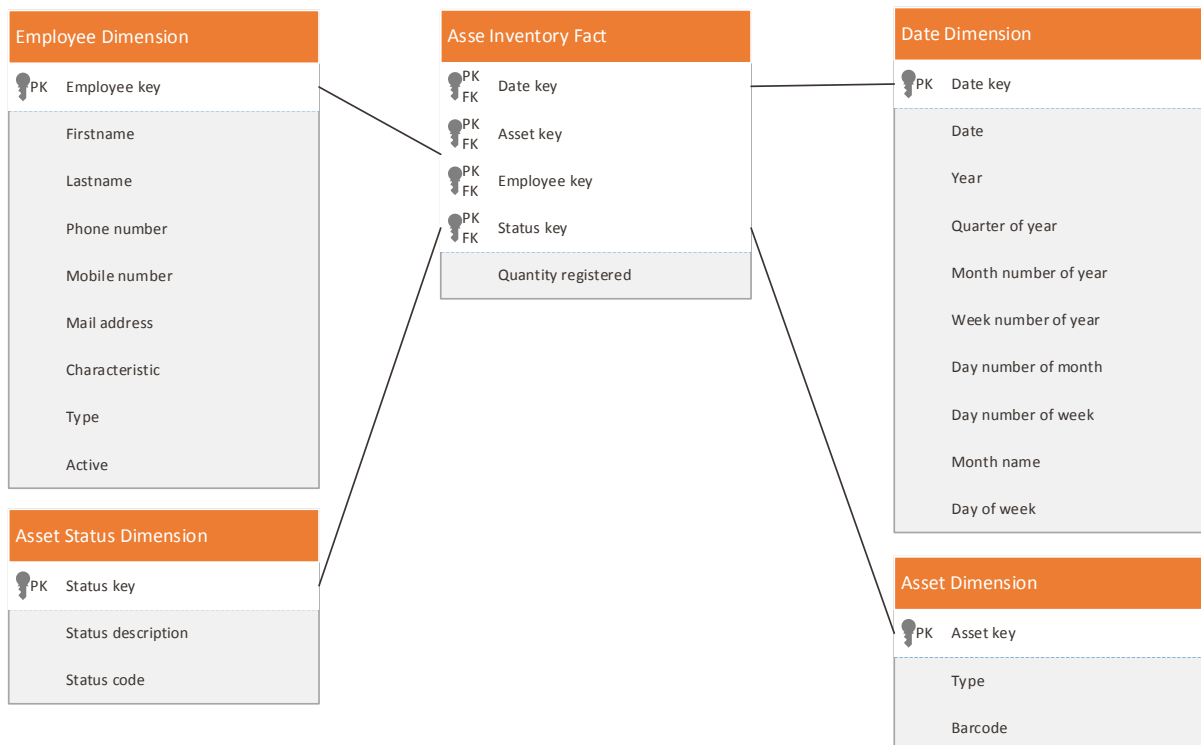
De gegevens van een asset wijzigen nooit. Het enige wijzigende element van een asset is de status van een asset. De statussen van een asset zijn vastgelegd en zullen nooit veranderen. Omdat de gegevens van assets nooit wijzigen, besloot ik het wijzigende element van assets als aparte dimensie op te nemen in de warehousedatabase. Het bijkomende voordeel van een aparte asset status dimensie, is dat deze bij aanmaak van de tabel kan worden gevuld en nooit zal wijzigen. De geschiedenis van een asset wordt tevens ook automatisch correct bewaard in de feitentabel.

Van de vier dimensies, is de asset status dimensie als eerst ingevuld. Een asset status bestaat uit een twee cijferige code en een status beschrijving. Een asset kan één van de zeven vastgelegde status codes toegewezen krijgen. Deze zijn als kenmerken aan de asset status dimensie toegevoegd. De asset dimensie kon ook eenvoudig worden ingevuld. Een asset heeft alleen een barcode en een kenmerk om aan te geven om wat voor type asset het betreft.

Met het invullen van de monteur dimensie kwam ik er achter dat deze dimensie niet voldoende is. Verschillende systeemgebruikers kunnen een asset registreren als defect, vermist of retour. De overweging voor het opnemen van de systeemgebruikers in monteur dimensie in plaats van het maken van een nieuwe dimensie voor systeemgebruikers, is gemaakt wegens het feit dat beide dimensies identiek zouden zijn bij opsplitsing. Alle medewerkers en gebruikers in de SmartWork 3.0 database bevatten dezelfde informatie. Het opsplitsen van de dimensie in twee identieke dimensies, is een overbodige stap. Het toevoegen van klanten aan deze dimensie in een later stadium van het project, waarmee de employee dimension uiteindelijk de naam person dimension kreeg, is om dezelfde reden gedaan. Ook klanten bevatten dezelfde informatie waardoor het maken van een customer dimension wederom zou resulteren in twee identieke dimensies.

Om onderscheid te kunnen maken tussen de soorten medewerkers, is het kenmerk 'type' opgenomen in employee dimensie. De keus om het type werknemer niet te bepalen aan de hand van verschillende vreemde sleutels die refereren naar employee dimension, is omdat de types relatief vaak wijzigen. Met de afronding van het SmartQuality project, komt er een nieuw type bij, namelijk het type 'Inspecteur'. Dit zou resulteren in een database structuur wijziging. Met het opnemen van het kenmerk 'type' in employee dimension, worden database structuur wijzigingen voorkomen en kan het opgelost worden in het ETL proces. Tevens bleek later in het project bij het configureren van de business intelligence omgeving dat het in Pentaho BI suite niet mogelijk is meerdere joins op één dimensie te maken. Door deze beperking kon het type niet bepaald worden met de vreemde sleutels.

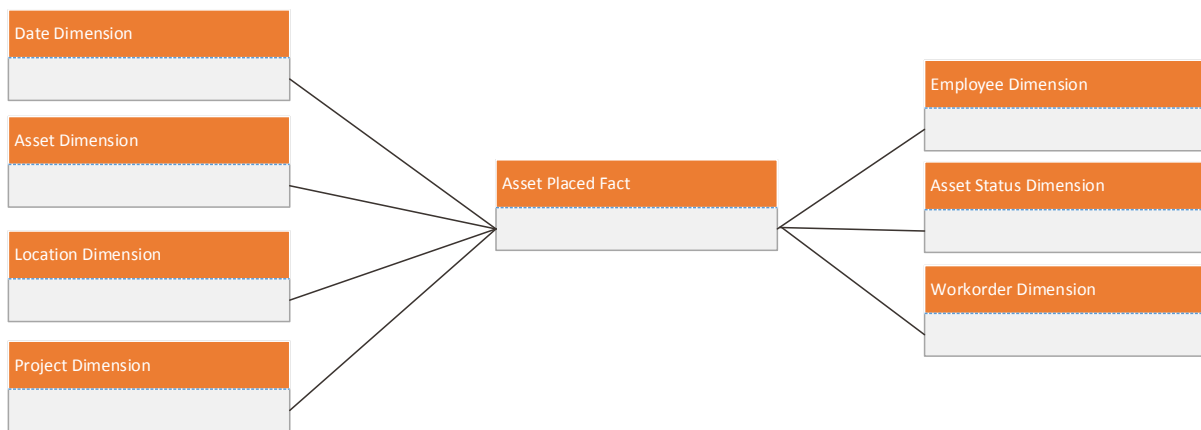
De kenmerken van de date dimensie is gebaseerd op de requirement dat een geselecteerde periode minimaal een dag en maximaal een jaar moet kunnen zijn. De woorden 'minimaal' en 'maximaal' geven tevens aan dat alle kenmerken tussen een dag en een jaar ook mogelijk moeten zijn. Met het invullen van de kenmerken in het conceptual design, kwam uiteindelijk het logical design tot stand zoals weergegeven in figuur 11. Op het moment van modelleren was er nog geen sprake van een SCD.



FIGUUR 11 - LOGICAL DESIGN INVENTORY FACT

6.2.2 ASSET PLACED FACT

Het bepalen van de dimensies en feiten voor asset placed is op dezelfde manier ontstaan als bij asset inventory. In de requirements geven de dikgedrukte woorden de dimensies aan en de schuingedrukte onderstreepte woorden de feiten aan. Voor het geplaatste assets feit zijn er wederom gegevens van de monteur en de asset nodig. Naast een geplaatste assets overzicht per monteur, dient het overzicht ook per project getoond te kunnen worden. Het overzicht betreft een door een gebruiker geselecteerde periode. Verder moet het overzicht te filteren zijn op locatie, moet het te achterhalen zijn waar voor welke werkopdracht een asset is geplaatst en wilt de gebruiker de plaatsing status kunnen zien. In totaal betreft het zeven dimensies rondom het geplaatste assets feit, namelijk medewerker, asset, asset status, datum, project, locatie en werkopdracht, waarbij asset status afkomstig is van het modelleren van asset inventory. Hiermee is het conceptual design zoals weergegeven in figuur 12 opgesteld.



FIGUUR 12 - CONCEPTUAL DESIGN ASSET PLACED FACT

De invulling van de dimensies datum, asset, asset status en medewerker was al gedaan met het opstellen van het inventaris feit. Momenteel heeft een project alleen een project naam en een SmartWork3.0 proces. Beide kenmerken zijn ingevuld in de project dimensie. De locatie dimensie moest op moment van modelleren in ieder geval dezelfde informatie bevatten die ook beschikbaar is in SmartWork3.0. Een werkopdracht bevat alleen de informatie van externe referentie en welke status de werkopdracht heeft. In tegenstelling tot de statussen van een asset, zijn de statussen van een werkopdracht dynamisch en niet van te voren vastgelegd. Overigens kan elk SmartWork proces weer een andere lijst met statussen bevatten. Om deze reden is status als kenmerk in de werkopdracht opgenomen.

Het feit dat de status van een werkopdracht kan wijzigen, zorgde er voor dat de werkopdracht dimensie een slowly changing dimension zou zijn. Om de geschiedenis kloppend te houden bij een status wijziging van een werkopdracht, dient het een type 2 SCD te zijn. Dit houdt in dat de status kenmerk van de werkopdracht dimensie bij een wijziging niet overschreven wordt, maar dat er een nieuwe record van de werkopdracht in de dimensie wordt aangemaakt met de nieuwe status. Voor de realisatie van een SCD type 2, is het kenmerk 'versie' nodig. Bij het opzoeken van de werkopdracht in de dimensie moet er altijd gebruik gemaakt worden van de hoogste versie. Voor meer complete informatie van de geschiedenis, zijn ook de kenmerken datum van en datum tot opgenomen in de dimensie. Dit geeft aan tot wanneer een versie van de werkopdracht geldig was of vanaf wanneer een versie van de werkopdracht geldig is. Met al deze gegevens is het logical design zoals weergegeven in figuur 13 opgesteld.



FIGUUR 13 - LOGICAL DESIGN ASSET PLACED FACT

6.3 MAPPING VAN GEGEVENS

Na het opstellen van de database ontwerpen, konden de beschrijvingen voor de ETL module, ook wel de mapping, worden opgesteld. Het opstellen van de mapping had als doel het in kaart brengen waar de data uit de source vandaan komt en waar de data naar de target heen gaat. Voor het opstellen van de mapping heb ik de methodiek van het boek 'The data warehouse ETL toolkit' [8] gebruikt. De mapping is opgesteld met behulp van het naast elkaar leggen van de gedetailleerde gegevensmodel beschrijvingen zoals beschreven in hoofdstuk 6.1 en het physical design van desbetreffende feit. De mapping voor asset inventory is weergegeven in tabel 11.

Target					Source				Transformation
Table	Column	Data Type	Table Type	SCD	Database	Table	Column	Data Type	-
dim_employee	employee_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_employee	first_name	varchar (50)	dimension	1	Smartwork3	User	first_name	varchar (50)	-
dim_employee	last_name	varchar (50)	dimension	1	Smartwork3	User	last_name	varchar (50)	-
dim_employee	phone	varchar (20)	dimension	1	Smartwork3	User	phone	varchar (20)	-
dim_employee	mobile	varchar (20)	dimension	1	Smartwork3	User	mobile	varchar (20)	-
dim_employee	email	varchar (20)	dimension	1	Smartwork3	User	email	varchar (254)	-
dim_employee	characteristic	varchar (50)	dimension	0	Smartwork3	User	login	varchar (50)	-
dim_employee	type	varchar (50)	dimension	1	Smartwork3	User	userRole_roleName	varchar (255)	Op basis van de rol dient het type (monteur, front office e.d.) bepaald te worden. Waarde 'onbekend' indien het type niet kan worden bepaald
dim_employee	active	boolean	dimension	1	Smartwork3	User	activated	tinyint	-
dim_asset	asset_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_asset	barcode	varchar(250)	dimension	0	Smartwork3	Asset	barcode	varchar (255)	select a.barcode from asset a. Check op de geldigheid van de barcode middels business rules
dim_asset	type	varchar (15)	dimension	0	Smartwork3	Asset	barcode	varchar (255)	Type van asset bepalen aan de hand van de barcode. Waarde 'onbekend' indien het type niet kan worden bepaald.
fact_asset_inventory	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_asset_inventory	asset_key	integer	fact	N/A	DWH	dim_asset	asset_key	integer	lookup op barcode
fact_asset_inventory	employee_key	integer	fact	N/A	DWH	dim_employee	employee_key	integer	lookup op characteristic
fact_asset_inventory	status_key	integer	fact	N/A	DWH	dim_asset_status	status_key	integer	lookup op status code
fact_asset_inventory	quantity_registered	integer	fact	N/A	Smartwork3	Asset	status	integer	Waarde '1' indien status 10, anders waarde '0'

TABEL 11 - ASSET INVENTORY MAPPING

De mappingtabel is onderverdeeld in verschillende componenten. De componenten zijn ingevuld zoals beschreven in hoofdstuk 3 van het boek 'The datawarehouse ETL toolkit' [8]. De component transformation is de enige die gedeeltelijk afwijkt van het boek. De betekenis van de componenten, zoals ze ook zijn terug te vinden in het boek, is als volgt:

- **Target Table:** De fysieke naam van de tabel zoals die is terug te vinden in de warehouse.
- **Target Column:** De naam van de kolom in de warehousetabel.
- **Target Data Type:** Het datatype van de kolom in de warehousetabel.
- **Table Type:** Geeft aan of de tabel een feit of dimensie is.
- **SCD:** Geeft voor dimensietabellen aan of het een type-1-, type-2- of type-3-SCD betreft. Het type kan voor elke kolom in de dimensie anders zijn.
- **Sourcedatabase:** De naam van de database waarin de sourcedata zich bevinden. Het kan ook de naam van een bestand betreffen indien de source een (tekst)bestand is.
- **Source Table:** De naam van de tabel waar de sourcedata vandaan komen.
- **Source Column:** De naam van de kolom van de sourcetabel. Kolommen die nodig zijn uit andere tabellen om de data uit deze kolom te krijgen worden gedocumenteerd in de transformation component met behulp van SQL.
- **Source Data Type:** Het datatype van de kolom in de sourcetabel.
- **Transformation:** Geeft eventueel een extra beschrijving van de acties die nodig zijn om de data te verkrijgen of de SQL om de data uit de source te verkrijgen.

6.4 IMPLEMENTEREN VAN DATABASE ONTWERPEN

Toen alle designs opgesteld waren, konden ze geïmplementeerd worden. Om ze te implementeren in PostgreSQL is gebruikgemaakt van de met PostgreSQL meegeleverde tool pgAdmin III [12]. Voor de realisatie van het ontwerp zijn de physical designs vertaald naar een SQL Script. Eerst zijn alle dimensies vertaald naar SQL en vervolgens de twee feitentabellen. In Code Snippet 2 staan als voorbeeld de SQL-queries van de date dimension en de twee feitentabellen weergegeven.

Er waren dusver twee dimensies die bij het aanmaken van de tabellen ingevuld konden worden. Deze dimensies bevatten altijd dezelfde vooraf vastgelegde data en wijzigen nooit. De dimensies betreffen de asset status dimension en date dimension. Het SQL-script voor het invullen van deze dimensies is terug te vinden in Code Snippet 1. De asset status dimension is ingevuld met alle mogelijke statuscodes en -beschrijvingen die een asset kan hebben. De date dimension is ingevuld met data uit een periode van tien jaar. De data betreffen begin 2015 tot eind 2025. Er is gekozen voor begin 2015 in plaats van de huidige datum, omdat SmartWork 3.0 sinds 2015 in gebruik is. Daarom moeten er ook data uit 2015 in de datawarehouse opgeslagen kunnen worden. Een datum van tien jaar in de toekomst zorgt ervoor dat de date dimension voor een lange tijd niet aangevuld hoeft te worden en dat de datawarehouse met data van de aankomende jaren om kan gaan.

```

CREATE TABLE dim_date
(
    date_key serial,
    date date NOT NULL,
    year integer NOT NULL,
    quarter integer NOT NULL,
    month integer NOT NULL,
    week integer NOT NULL,
    monthday integer NOT NULL,
    weekday integer NOT NULL,
    month_name character varying(30) NOT NULL,
    weekday_name character varying(30) NOT NULL,
    CONSTRAINT pk_dim_date PRIMARY KEY (date_key)
);

CREATE TABLE fact_asset_inventory
(
    date_key integer NOT NULL REFERENCES dim_date (date_key),
    asset_key integer NOT NULL REFERENCES dim_asset (asset_key),
    employee_key integer NOT NULL REFERENCES dim_employee (employee_key),
    status_key integer NOT NULL REFERENCES dim_status (status_key),
    quantity_registered integer NOT NULL,
    CONSTRAINT pk_fact_asset_inventort PRIMARY KEY (date_key, asset_key, employee_key, status_key)
);

CREATE TABLE fact_asset_placed
(
    date_key integer NOT NULL REFERENCES dim_date (date_key),
    employee_key integer NOT NULL REFERENCES dim_employee (employee_key),
    location_key integer NOT NULL REFERENCES dim_location (location_key),
    workorder_key integer NOT NULL REFERENCES dim_workorder (workorder_key),
    asset_key integer NOT NULL REFERENCES dim_asset (asset_key),
    status_key integer NOT NULL REFERENCES dim_asset_status (status_key),
    project_key integer NOT NULL REFERENCES dim_project (project_key),
    quantity_placed integer NOT NULL,
    CONSTRAINT pk_fact_asset_placed PRIMARY KEY (date_key, employee_key, location_key, workorder_key,
    asset_key, status_key, project_key)
);

```

CODE SNIPPET 2 - PHYSICAL DESIGN IMPLEMENTATIE

```

INSERT INTO "dim_date"
("date", "year", "quarter", "month", "week", "monthday", "weekday", "month_name", "weekday_name")
SELECT
    day,
    date_part('year', day),
    date_part('quarter', day),
    date_part('month', day),
    date_part('week', day),
    date_part('day', day),
    date_part('dow', day),
    rtrim(to_char(day, 'Month')),
    rtrim(to_char(day, 'day'))
FROM
    generate_series('2015-01-01'::date, '2025-12-31'::date, '1 day') day;

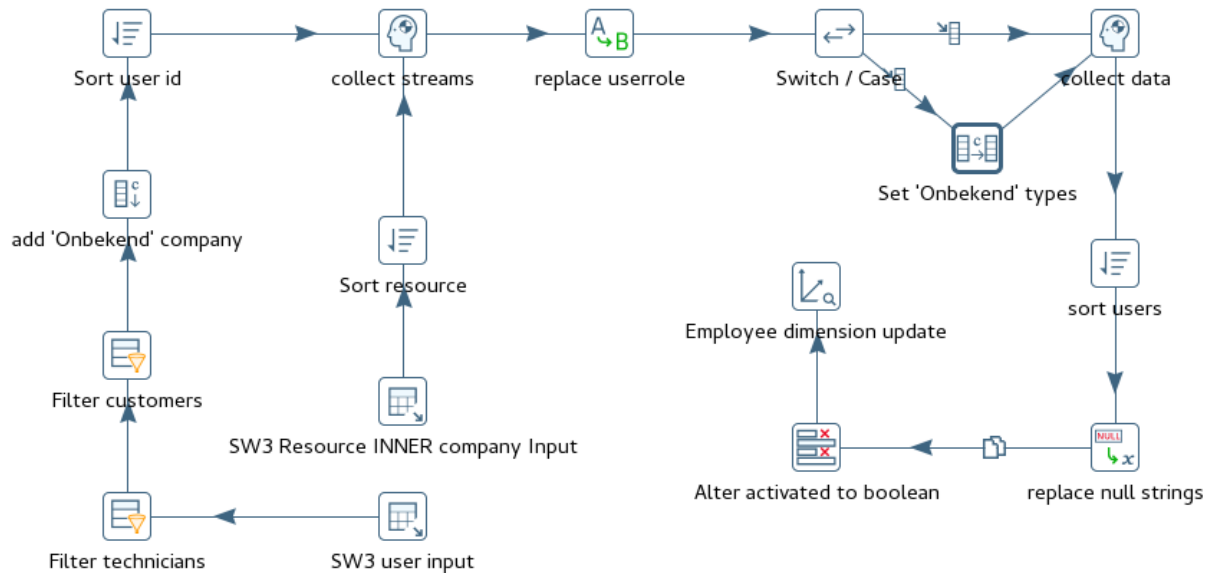
INSERT INTO "dim_asset_status"
(status_code, status_description)
VALUES
    (10, 'In voorraad'),
    (20, 'Defect'),
    (30, 'Retour'),
    (40, 'Vermist'),
    (50, 'Gestolen'),
    (60, 'Geplaatst'),
    (70, 'Geplaatst zonder registratie');

```

CODE SNIPPET 1 – INVULLING VAN DIMENSIES

6.5 PENTAHO KETTLE ETL

Na de implementatie kon de ETL-module gerealiseerd worden. Met behulp van de eerder opgestelde mappings zijn stap voor stap de transformaties gerealiseerd in Kettle Spoon. De flow van de ETL-jobs diende te beginnen met het updaten van alle dimensies, gevolgd door het invullen van de feitentabellen. Figuur 14 geeft de volledige Kettle-transformatie weer van het updaten van de employee dimension zoals weergegeven in Spoon.



FIGUUR 14 - SPOON UPDATE EMPLOYEE DIMESION

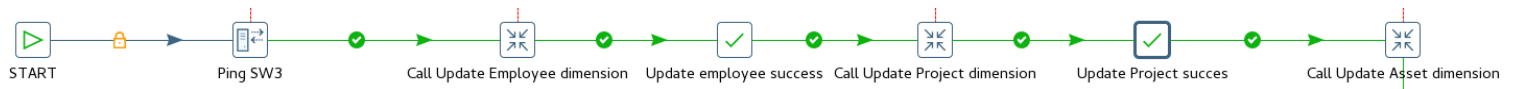
Zoals is af te lezen uit figuur 14 is tijdens het ophalen van de data geen filter toegepast op de rollen, ten behoeve van de performance. Queries die een 'IS NOT' of '<>' bevatten, hebben een impact op de performance, doordat ze per row meer acties moet uitvoeren om aan de query te voldoen. Het ETL-boek van The Kimball Group bevestigt ook dat dergelijke queries ten koste gaan van de performance. Het filteren van de rollen in de Kettle-engine na het ophalen van de users bleek ook sneller te verlopen.

De laatste stap van alle update-jobs is de dimension update. Binnen deze component kan per row bepaald worden welke actie er uitgevoerd dient te worden bij het updaten van de data, oftewel welk type SCD het betreft. Figuur 15 geeft weer hoe dit in Spoon geconfigureerd kan worden. Indien er gekozen wordt voor bijvoorbeeld een type-2-SCD eist Spoon een version kenmerk en de date from en date to worden toegevoegd aan de tabel. De dimensie-updatecomponent houdt zelf de versie en de dates bij.

^	#	Dimension field	Stream field to compare with	Type of dimension update
	1	first_name	first_name	Update
	2	last_name	last_name	Update
	3	phone	phone	Update
	4	mobile	mobile	Update

FIGUUR 15 – CONFIGURATIE SCD TYPE

Na het bouwen van alle dimensie-updatetransformaties en de feiten load jobs, is de overkoepelende Kettle-job aangemaakt. Binnen deze job worden alle eerder opgestelde transformatiejobs aangeroepen. Daarop volgt de flow de errorflow of de succesflow zoals weergegeven in figuur 16.



FIGUUR 16 - KETTLE JOB FLOW

Alle dimensie-updatetransformaties en feiten load jobs worden asynchroon afgehandeld, tenzij anders aangegeven. De overkoepelende Kettle-job die de calls naar de update- en loadjobs maakt, loopt te allen tijde synchroon.

6.6 PROTOTYPE

In het kader van het prototype is de Pentaho BI-suite in een lokale omgeving opgezet. Buiten het opzetten van de BI Suite zijn er geen front-end werkzaamheden of configuratiewerkzaamheden verricht voor de BI-omgeving. Ter voorbereiding van de demonstratie is de connectie naar de lokale warehousedatabase opgezet en zijn de gemaakte feiten voorbereid in de BI-omgeving. Voor beide feiten zijn voorbeeldrapporten aangemaakt zoals is weergegeven in figuur 17. Aan het einde van de sprint was het prototype van het warehouseplatform klaar en draaide het in een lokale omgeving.

Meetwaarden						
Quantity placed						
Kolommen						
Name						
Rijen						
Month name						
Last name						
Filter						
Month name	Last name	Afkeur 2016	GSA 2015	GSA 2016	Reimerswaal	Steekproef 2016
November	van der Kroef		155			
	van der Harst		121			
	van den Heuvel		128			
	van de Voorde		73			
	van Wijk		144			
	van Vliet		213			
	van Rooij		175			
	van Minde		219			
	Walraven		22			
	Tramper		253			
	Toubiana		238			
	Sande van de		18			
	Rombaut		18			
	Raber		178			
	Pouwelse		128			
	Pijpelink		148			
	Phil		80			
	Panneman		1			
	Neels		31			
	Moorman		45			

FIGUUR 17 - VOORBEELDRAPPORTAGE

7 SPRINT 2 – REALISATIE WAREHOUSE IN STAGING

Sprint 2 stond in het teken van uitbreiding van het prototype tot een eerste versie van het uiteindelijke platform. Voorafgaand hieraan zijn er nieuwe requirements vergaard middels een demonstratie van het prototype van de vorige sprint. Ook is in deze sprint de stagingserver opgezet zodat het platform aan het eind van de sprint in een stagingomgeving kon draaien. Verder zijn in sprint 2 ook werkzaamheden verricht aan de BI-omgeving.

7.1 DEMONSTRATIE PROTOTYPE

Voor de start van de sprint is er een demonstratie van het prototype gegeven. Zo kregen de stakeholders een beeld van de mogelijkheden van het warehouseplatform en wisten ze wat ze konden verwachten bij de oplevering. Een van de doelen van de demonstratie van het prototype was om meer requirements te achterhalen. Tijdens de demonstratie kwamen de vragen en wensen van de stakeholders naar voren. De requirements zijn wederom gebaseerd op de granulariteit van de business processen. Achteraf zijn de requirements in het requirementsdocument verwerkt en besproken met de product owner. Hij heeft vervolgens de user stories, weergegeven in tabel 12, geselecteerd die in sprint 2 gerealiseerd moesten worden.

ID	User story	Req. Nr.
SH1	Als gebruiker wil ik de huidige status van een werkopdracht per dag, per project inzien zodat ik de voortgang van een project kan bijhouden	K8
SH2	Als gebruiker kan ik de aantallen van de 'eind' statuses zien zodat ik inzicht heb in de verhouding tussen de werkopdrachten	K9,K10, K11,K12
SH6	Als beheerder kan ik de scheduling van synchronisatie taken beheren zodat ik de taken naar wens van klanten kan configureren	K4
SH9	Als gebruiker wil ik de rapporten binnen het platform kunnen exporteren naar CSV en XLSX formaat zodat ik deze buiten het platform ook kan gebruiken	K25,K26
SH10	Als gebruiker kan ik ad-hoc rapporten maken op basis van bestaande data in de data warehouse zodat ik niet vooraf gedefinieerde vragen kan beantwoorden	K4A,K5A
SH12	Als gebruiker heb ik de mogelijkheid om data te visualiseren met grafieken en diagrammen zodat ik een duidelijk, visueel overzicht heb van mijn data	K28,K29 K30
SH15	Als beheerder kan ik systeemgebruikers beheren zodat ik op aanvraag gebruikers en zijn permissies kan toevoegen, verwijderen of wijzigen	K8A,K9A K10A
SH16	Als gebruiker kan ik de tijdigheid van monteurs per dag per tijdvak per werkopdracht of per project in zien zodat ik weet welke monteurs ik moet aansturen	P1,P4,P5 P6,P7, P8A, P12A
SH17	Als gebruiker kan ik per dag, per project en per werkopdracht zien hoeveel minuten te vroeg of te laat een monteur bij zijn afspraak is zodat ik weet of een monteur aanhoudend te laat of te vroeg is	P2,P3
SH18	Als gebruikers kan ik het aantal geregistreerde en het aantal ongeregistreerde geplaatste meters achterhalen zodat ik kan bijsturen op het logistiek beheer	P5A
SH19	Als gebruiker wil ik de aanleiding en het metertype van een werkopdracht kunnen achterhalen zodat ik weet wat voor soort werkopdracht het betreft	P10,P11
SH20	Als gebruiker kan ik de regio van een locatie achterhalen zodat ik dit intern eenvoudig kan communiceren	P12

SH21	Als gebruiker wil ik per project, per werknemer en per dag het aantal uitgevallen werkopdrachten kunnen inzien zodat ik werknemers en klanten kan bijsturen	P14,P16
SH22	Als gebruiker wil ik de reden van uitval en het commentaar van uitval registratie kunnen zien zodat ik weet of het met een geldige reden als uitval is geregistreerd	P15,P17
SH23	Als beheerder heb ik een overzicht van alle taken en wat de status van deze taken is zodat ik gefaalde taken kan herstellen en taken eventueel kan fine tunen	P19,P20
SH25	Als gebruiker wil ik per project, per dag en per monteur de productiviteit kunnen inzien zodat ik weet welke monteurs wel of niet productief zijn en hierop kan inspelen	P22,P23, P24,P25, P26,P27, P13A
SH26	Als gebruiker kan ik achterhalen bij welke aannemer een monteur momenteel werkzaam is en waar een monteur eventueel werkzaam was zodat ik contact op kan nemen met de aannemers indien dat nodig is	P1A,P2A

TABEL 12 - SPRINT 2 USER STORIES

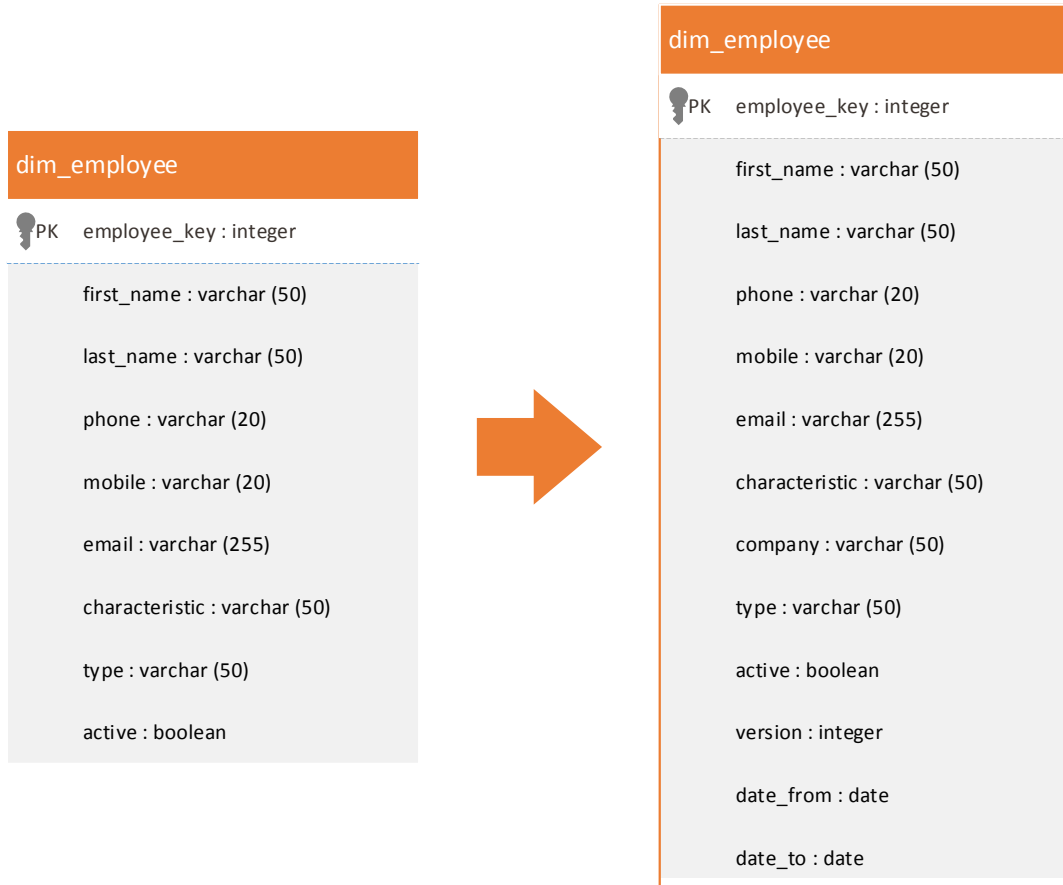
7.2 WIJZIGEN EN UITBREIDEN DATABASE ONTWERPEN

De meeste databasewijzigingen waren een kwestie van kenmerken toevoegen aan de tabellen en deze uitwerken in de mapping. De meest interessante wijziging betreft de mogelijkheid om te achterhalen waar een medewerker werkzaam is. De overweging tussen het opnemen van bedrijf als kenmerk in employee dimension of bedrijf als aparte dimensie was eenvoudig te maken. De enige relevante informatie is de naam van het bedrijf, de rest van de bedrijfsgegevens is niet interessant voor de stakeholders. Daarom is bedrijf als kenmerk opgenomen in employee dimension.

Om de geschiedenis intact te houden moest de employee dimension een type-2-SCD worden. Om dit te realiseren zijn de versie- en datumkenmerken toegevoegd aan de dimensie. In figuur 18 zijn de wijzigingen ten opzichte van eerder het opgestelde model weergegeven.

Het 'workorder failure' feit was in eerste instantie gemodelleerd met de reden en commentaar kenmerken in het feitentabel. Gezien deze combinatie van kenmerken evenveel records zou bevatten in een aparte dimensie als wanneer deze in het feitentabel worden opgenomen, was het overbodig om deze als aparte dimensie te modelleren. Echter, na het toevoegen van het kenmerk 'failure type' was het zinvol om een 'failure reason' dimensie te modelleren. In deze dimensie zijn de kenmerken reden en failure type opgenomen. Deze combinatie van kenmerken blijft beperkt in het aantal records. Het kenmerk commentaar paste niet in de failure reason dimensie, maar functioneert ook niet als meting. Het kenmerk behoort wel tot het feit van het moment van uitvalregistratie en is daarom opgenomen in het feitentabel.

Het uitbreiden van de databaseontwerpen met de feiten 'workorder failure', 'project progress', 'productivity', 'timeliness' en de nieuwe benodigde dimensies zijn op dezelfde wijze gemodelleerd als beschreven in hoofdstuk 6.2. In bijlage D is een compleet overzicht te vinden van de totstandkoming van de eerder genoemde tabellen.



FIGUUR 18 - EMPLOYEE DIMENSION WIJZIGINGEN

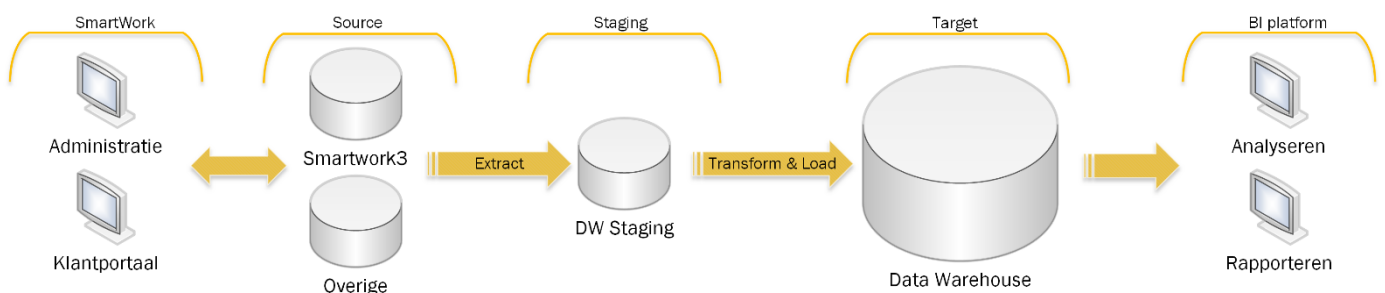
7.3 STAGINGSERVER

Om de stagingserver op te zetten is gebruikgemaakt van een CentOS image die voorafgaand aan het project al was opgezet. In deze image zijn alleen de nodige security configuraties gedaan en hij bevat verder geen installaties van software of packages.

7.3.1 ARCHITECTUUR

Voorafgaand aan het opzetten van de stagingserver heb ik de algemene architectuur gemodelleerd ter verduidelijking voor de stakeholders. De architectuur zoals ik het in eerste instantie in gedachten had is weergegeven in figuur 19. De SmartWork-applicatie leest en schrijft data naar de onderliggende SmartWork 3.0-database, afhankelijk van de acties die een gebruiker uitvoert. Op een gedefinieerd tijdstip wordt de extractie van de data uit de SmartWork 3.0-database naar een stagingdatabase uitgevoerd met behulp van een Kettle-job. De structuur van de stagingdatabase is identiek aan die van SmartWork 3.0. Tijdens dit proces wordt in feite alleen de data van SmartWork 3.0 naar de stagingdatabase gekopieerd. De reden hiervoor is om zo min mogelijk impact te hebben op de SmartWork 3.0-database tijdens het ETL-proces. Indien een ETL-proces fout gaat, hoeft de SmartWork 3.0 database dan niet wederom uitgelezen te worden, maar kan de data uit de stagingdatabase worden gehaald. Na het extractproces worden de transformatie- en loadjobs uitgevoerd. Deze zetten de data om naar het gewenste formaat en de gewenste kwaliteit en laden deze in de warehousedatabase. Het BI-platform leest data uit de warehousedatabase waarover de gebruikers kunnen rapporteren en analyseren.

Voor het realiseren van de warehouseomgeving is de stagingdatabase weggelaten. Het ETL-proces vindt in de avond plaats, waardoor de impact op de SmartWork 3.0-database niet uitmaakt. Op het gewenste tijdstip van de uitvoering zijn er geen gebruikers die gebruikmaken van de SmartWork-applicatie. De stagingdatabase was wel van belang geweest indien het ETL-proces eerder op de dag of vaker dan eenmaal per dag had plaatsgevonden. Als het proces bijvoorbeeld elk uur uitgevoerd moest worden, was het implementeren van een stagingdatabase cruciaal geweest om de gebruikers niet te hinderen tijdens hun werkzaamheden.



FIGUUR 19 - GLOBALE WAREHOUSE ARCHITECTUUR

7.3.2 SERVERCONFIGURATIE

Voor het opzetten en configureren van de stagingserver is eerst Java versie 7 geïnstalleerd. Java 8 is niet gebruikt, omdat de Pentaho-softwarepakketten deze versie nog niet ondersteunen. Java 7 is echter niet meer beschikbaar in de yum repository en moest handmatig worden geïnstalleerd met behulp van de `wget` command. Vervolgens is PostgreSQL op de server geïnstalleerd. Het installatieproces van PostgreSQL is weergegeven in Code Snippet 3. Na de installatie van PostgreSQL is de SmartHouse-database aangemaakt en het SQL-script voor het aanmaken van de tabellen uitgevoerd.

```
yum install postgresql-server postgresql-contrib
sudo systemctl start postgresql
postgresql-setup initdb
sudo systemctl enable postgresql
```

CODE SNIPPET 3 - POSTGRESQL INSTALLATIE

Voor onder andere de scheduling van de ETL-module wilde ik gebruikmaken van Jenkins. Jenkins is een opensource continuous-integration tool geschreven in Java en wordt gezien als een de-facto standaard. Jenkins kan eenvoudig worden geconfigureerd met behulp van de webinterface en is flexibel en uitbreidbaar met plug-ins, zodat het volledig aangepast kan worden aan de wensen van gebruikers [13]. Binnen SmartInstall wordt bij diverse projecten al gebruikgemaakt van Jenkins en ik heb ook al enige ervaring met de tool. Het installeren van Jenkins op de stagingserver is gedaan middels de yum repository. Aangezien Java reeds geïnstalleerd was hoefde er verder niets te gebeuren voor de installatie.

Ten slotte zijn de Pentaho-softwarepakketten op de server geïnstalleerd. In de documentatie van Pentaho staat een duidelijke uitleg van de uit te voeren stappen. Met behulp van het `wget`-command is het meest recente pakket van Pentaho gedownload. De installatie betrof slechts het uitvoeren van het binary-bestand zoals weergegeven in Code Snippet 4. De configuratie van server porten en dergelijke kon worden gedaan tijdens het installatieproces.

```
chmod +x linux-bi-suite-6.1.bin
./linux-bi-suite-6.1.bin
psql -U postgres -p 5432 -a -f create_jcr_postgresql.sql
psql -U postgres -p 5432 -a -f create_quartz_postgresql.sql
psql -U postgres -p 5432 -a -f create_repository_postgresql.sql
```

CODE SNIPPET 4 - PENTAHO BI-SUITE INSTALLATIE

Na de installatie moesten er drie SQL-scripts, meegeleverd met de BI-suite, worden uitgevoerd. De SQL-scripts maken databases en tabellen aan die nodig zijn voor Pentaho BI. In deze tabellen worden de gebruikers, scheduling en meta-data opgeslagen.

Voor de ETL-module diende de stagingserver data uit de SmartWork 3.0-database te lezen met behulp van SSH. Om deze communicatie toe te staan, diende de public SSH-key van de SmartWork-server toegevoegd te worden aan de stagingserver en vice versa. Op de stagingserver moesten de SSH-keys worden gegenereerd en de public SSH-key van SmartWork worden toegevoegd. De public SSH-key heb ik gekregen van de SmartWork-serverbeheerder en toegevoegd aan het bestand `~/.ssh/authorized_keys` op de staging server. De public SSH-key van de stagingserver is gegenereerd met `ssh-keygen -t rsa` en toegevoegd aan de SmartWork-server door de SmartWork-serverbeheerder. Na het uitvoeren van deze acties kon en mocht er communicatie tussen beide servers plaatsvinden.

7.4 UITBREIDEN ETL-MODULE

7.4.1 VERSIEBEHEER

Alvorens de ETL-module uit te breiden zijn de Kettle-jobbestanden onder versiebeheer gezet. Dit maakt de overdracht van het project eenvoudiger en zorgt ervoor dat de bestanden veilig online zijn opgeslagen, waardoor het risico om het product te verliezen door eventuele hardwarebeschadiging wordt weggenomen. Ook maakt dit het eenvoudiger om in teamverband te werken aan het warehouseproduct na de afronding van het afstudeerproject en geeft het de mogelijkheid voor automatisering van het deployment proces.

Voor alle projecten wordt binnen SmartInstall gebruikgemaakt van Git in combinatie met bitbucket. Git is een gratis opensource versiebeheersysteem, gepubliceerd onder de GNU general public license V2. [14]. Bitbucket is een hostingservice voor projecten die gebruikmaken van Git- of Mercurial-versiebeheersystemen en is gepubliceerd door Atlassian. De integratie van bitbucket met andere Atlassian-software zorgt voor een duidelijk overzicht van codewijzigingen en geeft de mogelijkheid om development- en deployment processen te automatiseren. [15].

De ETL-module bestaat uit verschillende Kettle-bestanden. Dit zijn XML tekstbestanden waarin de opbouw en configuratie van de jobs worden opgeslagen. Aangezien de bestanden van Kettle XML-tekstbestanden betreffen, kunnen deze ook onder versiebeheer worden gezet met Git en bitbucket. Na het aanmaken van een repository in bitbucket is van de folder waarin alle bestanden stonden een Git-folder/-branch gemaakt die is gekoppeld aan de bitbucket-repository. Code Snippet 5 geeft weer hoe dit is gedaan met behulp van de Git-command line. Toen het product onder versiebeheer stond is het deployment proces geautomatiseerd met behulp van Jenkins.

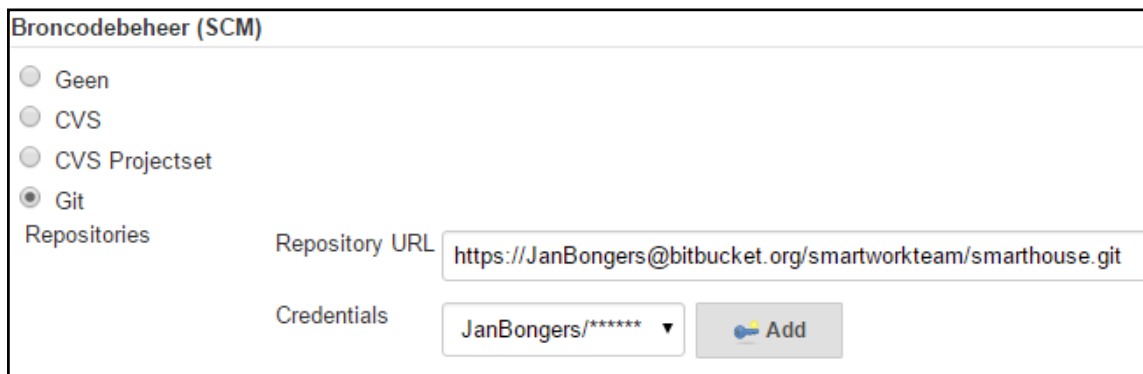
```
cd /home/jay/Pentaho\ Transformations/  
git init  
git remote add origin  
https://JanBongers@bitbucket.org/smartworkteam/smarthouse.git  
git add *  
git commit -m 'initial commit'  
git push -u origin master
```

CODE SNIPPET 5 - GIT VERSIEBEHEER COMMAND

7.4.2 DEPLOYMENT NAAR STAGING

De deployment van de ETL-module was een handmatig proces. Dit zou op langere termijn onhandig zijn en in totaliteit onnodig veel tijd kosten. Het idee was om de bestanden gelijk te deployen zodra er een wijziging werd gepusht naar de bitbucket-repository. Er moet dus een Jenkins-build getriggerd worden bij het pushen van een wijziging. De realisatie hiervan was mogelijk dankzij de Jenkins service management van bitbucket [16].

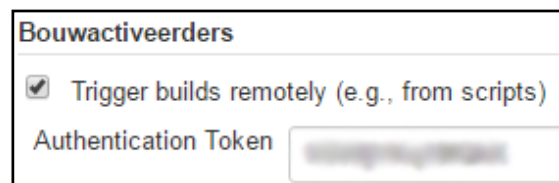
Voor de configuratie van het automatisch deployen moest er een nieuw Jenkinsproject worden aangemaakt. Voorafgaand daaraan zijn de Git- en bitbucket-plugin's geïnstalleerd met behulp van de plug-inmanager van Jenkins. In het Jenkinsproject is aangegeven dat het bij het bouwen van het project een Git repository moet pullen en uitchecken. De configuratie hiervan is weergegeven in figuur 20.



FIGUUR 20 - JENKINS BRONCODEBEHEER

Vervolgens is het project geconfigureerd om geactiveerd te worden op afstand. Hierbij is een authenticatie token gedefinieerd dat gebruikt moest worden om bitbucket te configureren. Dit authenticatie token is een willekeurig gegenereerde string. De configuratie van het op afstand activeren is weergegeven in figuur 21.

Na het aanmaken van het Jenkinsproject kon de configuratie van de bitbucket-service worden gedaan. Jenkinsservice zit standaard in bitbucket. De configuratie hiervan is weergegeven in figuur 22. De invulling van de configuratie is als volgt:



FIGUUR 21 - JENKINS BOUWACTIVEERDERS

- **Endpoint:** `http://username:apitoken@yourjenkinsurl.com`. Waarbij api token afkomstig is van gebruiker authenticatie van Jenkins
- **Project name:** Naam van project zoals gedefinieerd in Jenkins
- **Token:** De in Jenkins aangemaakte authenticatie token, bestaande uit een random gegenereerde string.

Jenkins

Endpoint

Project name

Token

FIGUUR 22 - BITBUCKET JENKINS SERVICE

7.4.3 JENKINS SCHEDULING

Een belangrijk onderdeel van het warehouseplatform is een geautomatiseerd ETL-proces. De realisatie daarvan is ook gedaan met behulp van Jenkins. Jenkins heeft de mogelijkheid om projecten op een gedefinieerd tijdstip te bouwen. Voor de scheduling van de Kettle-jobs zijn er twee Jenkinsprojecten aangemaakt, namelijk 'Kitchen update dimensions' en 'Kitchen update facts'. Het overzicht van de Jenkinsprojecten zoals weergegeven in Jenkins is weergegeven in Figuur 23.

All +			
S	W	Name ↓	Last Success
		Kitchen Update Dimensions	14 min - #7
		Kitchen Update Facts	11 min - #1

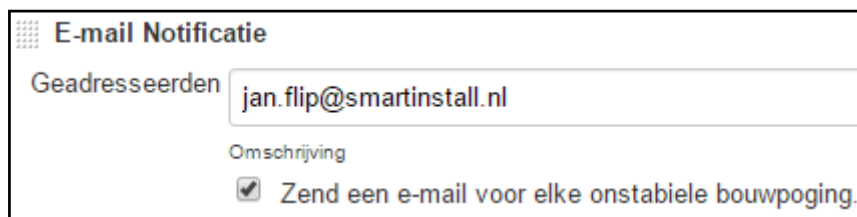
FIGUUR 23 - JENKINS DASHBOARD

Het 'Kitchen update dimensions'-project is dusdanig geconfigureerd dat het dagelijks om 22:00 uur de ETL-job van dimension update uitvoert. Dit gebeurt door middel van een shell script. Het shell script van Jenkins voert het shell script van Kitchen uit met een gedefinieerde Kettle-job. Het 'Kitchen update facts'-project bevat geen scheduling, maar wordt getriggerd zodra de uitvoer van 'Kitchen update dimensions' succesvol is verlopen. Het is namelijk alleen zinvol om de feitentabellen te updaten als de dimensies succesvol zijn geüpdatet. Op deze manier is zeker dat de feitentabellen gevuld worden met de laatste, kloppende informatie van de dimensies. Het activeren van 'Kitchen update facts' is geconfigureerd in de acties na de bouwpoing. De configuratie van de Jenkinsprojecten is weergegeven in figuur 24.

☒ Periodisch bouwen Planning 0 22 * * *	Bouw ander project. Te bouwen projecten Kitchen Update Facts ☒ Trigger only if build is stable
Voer shell-script uit Commando sudo /opt/Pentaho/design-tools/data-integration/kitchen.sh -file=/home/admin/Pentaho/data-integration/job/dimension/update_dimensions.kjb	

FIGUUR 24 - JENKINS PROJECT CONFIGURATIE

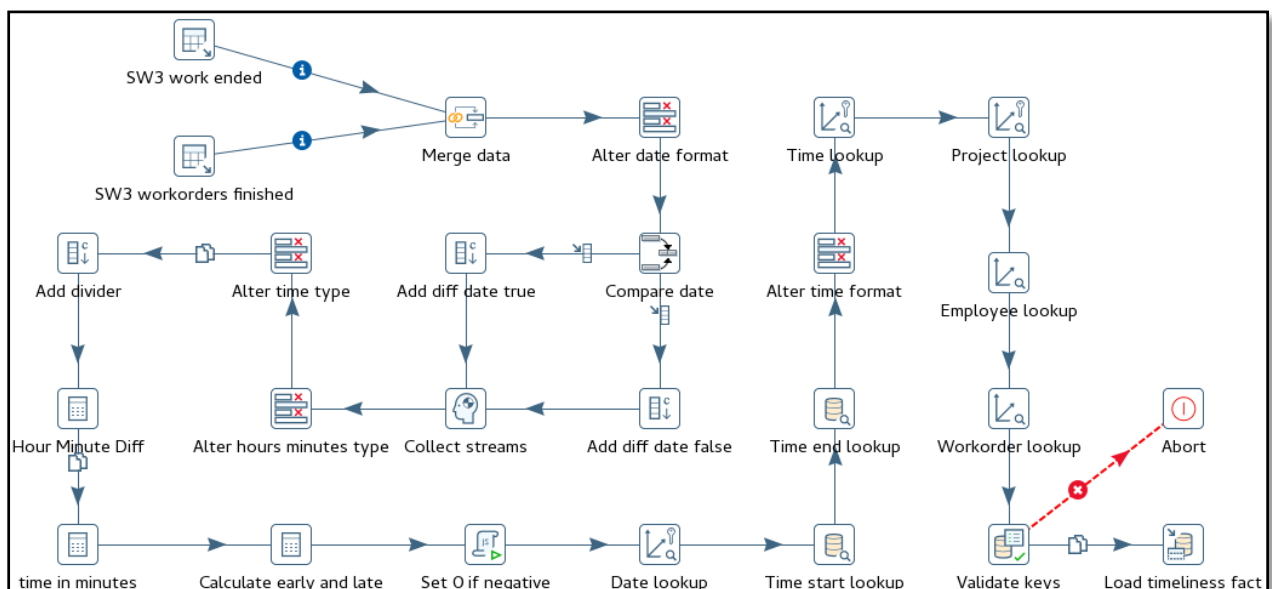
Tijdens het configureren van de Jenkins projecten kwam de mogelijkheid naar voren om een e-mailnotificatie te verzenden na het uitvoeren van een bouwpoing. In het prototype zijn de Kettle-jobs dusdanig geconfigureerd dat ze een e-mail verzenden indien er een fout optreedt bij de uitvoering via de 'Send e-mail'-component. Deze component kan echter slechts één e-mail sturen naar één adres en moet per job opnieuw geconfigureerd worden als een errorflow. Doordat het opgevangen werd als een errorflow, werd in Jenkins weergegeven dat een gefaalde job succesvol was uitgevoerd in plaats van als een rode, instabiele build. Om deze redenen zijn de e-mailnotificaties uit de Kettle-jobs gehaald en wordt dit proces door Jenkins gedaan. De configuratie van e-mailnotificaties is weergegeven in figuur 25. Door deze wijziging werden de Kettle-jobs opgeschoond en gaf Jenkins het correct aan als de uitvoering van een build fout ging. Ook was het nu mogelijk om meer dan één geadresseerde toe te voegen.



FIGUUR 25 - JENKINS E-MAILNOTIFICATIE

7.4.4 UITBREIDEN PENTAHO KETTLE ETL

Voorafgaand aan de bouw en uitbreiding van de Kettle-jobs is de datamapping aangevuld met de nieuwe data. In tabel 13 is de mapping van de wijzigingen ten opzichte van eerder opgestelde tabellen in de vorige sprint weergegeven. In bijlage D is een overzicht te zien van de mapping in zijn totaliteit. De datamapping functioneerde als input voor het bouwen van de ETL-jobs. In figuur 26 is de ETL-transformatiejob van het updaten van het timeliness feit weergegeven. In bijlage F is een overzicht van alle Kettle-ETL-jobs en de gemaakte queries te zien.



FIGUUR 26 - SPOON TIMELINESS FACT JOB

Target					Source				Transformation
Table	Column	Data Type	Table Type	SCD	Database	Table	Column	Data Type	-
dim_employee	company	varchar (50)	dimension	2	Smartwork3	Company	name	varchar (255)	select c.name from user u, resource r, company c u.id = r.user_id and r.company_id = c.id. Controle op naam lengte
dim_location	region	varchar (15)	dimension	1	Smartwork3	Location	zip	varchar (7)	Aan de hand van de business logica dient de regio bepaald te worden voor de postcode
dim_workorder	motive	varchar (50)	dimension	0	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueString cfvs where td.id = cfv.workorder_id and cfv.customField_id = cf.id and cfvs.id = cfv.id
fact_asset_placed	quantity_placed	integer	fact	N/A	Smartwork3	Asset	status	integer	waarde '1' indien status 60, anders waarde '0'
fact_asset_placed	quantity_unregistered	integer	fact	N/A	Smartwork3	Asset	status	integer	waarde '1' indien status 70, anders waarde '0'

TABEL 13 - MAPPING VAN SPRINT 2 WIJZIGINGEN

In Code Snippet 6 staat de query van de inputstap 'SW3 work ended' weergegeven. De data die de query ophaalt betreffen de starttijd en eindtijd van het geplande tijdsblok en het tijdstip waarop de monteur zijn werkzaamheden is begonnen voor een werkopdracht. De werkopdracht wordt alleen meegenomen indien een monteur de werkzaamheden voor de werkopdracht is gestart en de werkzaamheden heeft beëindigd.

```

SELECT
    td.external_ref_id,
    podate.date AS datePlanned,
    poday.timeOnDayStart,
    poday.timeOnDayEnd,
    pv.dateTimeValue AS dateWorked,
    p.name AS project_name,
    p.processName,
    DATE_FORMAT(poday.timeOnDayStart, '%H') hourStart,
    DATE_FORMAT(poday.timeOnDayStart, '%i') minuteStart,
    DATE_FORMAT(poday.timeOnDayEnd, '%H') hourEnd,
    DATE_FORMAT(poday.timeOnDayEnd, '%i') minuteEnd,
    DATE_FORMAT(pv.dateTimeValue, '%H') hourStartWork,
    DATE_FORMAT(pv.dateTimeValue, '%i') minuteStartWork
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON td.id = bpi.workItemId
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
INNER JOIN PeriodOnDate podate ON td.timeWindowAppointment_id = podate.id
INNER JOIN PeriodOnDay poday ON podate.periodOnDay_id = poday.id
INNER JOIN
(
    SELECT tds.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tds
    INNER JOIN BaseProcessInstance bpi ON tds.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'WERKZAAMHEDEN_BEGIN'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id = LatestVariable.workId
AND pv.variableName = 'WERKZAAMHEDEN_BEGIN'
AND pv.dateTimeValue IS NOT NULL
ORDER BY external_ref_id ASC;

```

CODE SNIPPET 6 - WORK ENDED QUERY

Code Snippet 7 geeft de query van de inputstap 'SW3 workorders finished' weer. De query haalt de werkopdrachten waar de werkzaamheden zijn beëindigd op, inclusief de monteur die de werkzaamheden heeft uitgevoerd. De data van deze query worden in de stap 'Merge data' samengevoegd met de data van de query uit Code Snippet 6. De data worden samengevoegd op basis van de externe referentie. Met de samengevoegde data kan de tijdigheid van monteurs worden bepaald in de rest van het ETL-proces.

Alle 'Alter'-stappen betreffen dezelfde component. Met deze component is het mogelijk om kolomnamen te wijzigen, datatypes te wijzigen, dataformaten aan te passen en andere acties uit te voeren. De stap 'Compare date' bepaalt of de start van de werkzaamheden dezelfde datum betreft als de datum van het geplande tijdsblok. In de praktijk komt het voor dat een monteur een of meer dagen eerder dan de geplande dag zijn werkzaamheden uitvoert op persoonlijk verzoek van de klant. Klanten spreken in dat geval de monteur aan die in de buurt werkzaamheden uitvoert en vragen of hij de werkzaamheden die dag al kan uitvoeren in plaats van op de geplande dag. In dit geval is het aantal minuten te vroeg hoog, wat ervoor zorgt dat het berekende gemiddelde niet representatief is.

```

SELECT
    td.external_ref_id,
    pv.stringValue AS login
FROM ProcessVariable pv
INNER JOIN BaseProcessInstance bpi ON pv.processInstance_id = bpi.id
INNER JOIN TaskDefault td ON bpi.workItemId = td.id
WHERE NOT EXISTS
(
    SELECT pvv.processInstance_id FROM ProcessVariable pvv
    WHERE bpi.id = pvv.processInstance_id
    AND pvv.variableName = 'WERKZAAMHEDEN_EIND'
    AND pvv.dateTimeValue IS NULL
)
AND pv.variableName = 'MONTEUR_UTGEVOERD'
AND pv.stringValue IS NOT NULL
ORDER BY external_ref_id;

```

CODE SNIPPET 7 - WORKORDER FINISHED QUERY

In de 'Calculate'-stappen worden onder andere de minuten te vroeg en te laat berekend. Voorafgaand aan deze berekening worden de start- en einduren omgezet naar minuten. Deze worden in minuten opgeteld bij de start- en eindminuten. Vervolgens wordt het aantal minuten te vroeg berekend door de starttijd van de werkzaamheden af te trekken van de geplande starttijd van het tijdsblok. Indien er een positief getal uit de berekening komt, is de monteur te vroeg met zijn werkzaamheden begonnen. De berekening van het aantal minuten te laat wordt uitgevoerd door de geplande eindtijd van het tijdsblok af te trekken van de starttijd van de werkzaamheden. Indien er een positief getal uit de berekening komt, is de monteur te laat met zijn werkzaamheden begonnen.

De stap 'Set 0 if negative' betreft de custom Javascript-component. In deze component is het mogelijk om Javascript-scripts te schrijven omtrent de jobdata. Het script van de 'Set 0 if negative' is weergegeven in Code Snippet 8. Het betreft een eenvoudig script dat de waarde van de kolommen 'minutes_early' en 'minutes_late' op 0 zet als deze door voorgaande berekeningen een negatieve waarde bevatten.

```

var minuteEarly = minutes_early.getInteger();
var minuteLate = minutes_late.getInteger();
var zero = 0;

if(minuteEarly < 0) {
    minutes_early.setValue(zero);
}

if(minuteLate < 0) {
    minutes_late.setValue(zero);
}

```

CODE SNIPPET 8 - CUSTOM JAVASCRIPT

De componenten van de stappen 'Lookup', op de 'Time end lookup' en 'Time start lookup' na, halen de technical key van de betreffende dimensie op, op basis van gedefinieerde zoekvelden. Zo wordt bijvoorbeeld in de stap 'Workorder lookup' de workorder key opgehaald op basis van de externe referentie. Met deze component kan alleen de technical key worden opgehaald. De component van de 'Time end lookup' en 'Time start lookup' werkt hetzelfde als de component van de andere 'Lookup', alleen kan in deze stap geselecteerd worden welke kolommen er teruggegeven worden op basis van de gedefinieerde zoekvelden.

De laatste stap van alle ETL-jobs omtrent een feit betreft een component van de stap 'Load timeliness fact'. In deze stap worden de gegevens van de job toegevoegd aan de tabel. De configuratie van de component is weergegeven in figuur 27. Voor het timeliness-feit worden records opgezocht op basis van 'project_key', 'workorder_key', 'employee_key', 'date_key' en 'time_key'. Indien er geen record bestaat met de waardes van deze combinatie, wordt er een nieuwe toegevoegd. Welke data er aan het nieuwe record moeten worden toegevoegd wordt geconfigureerd onder update fields. Er vinden geen updates plaats als de 'Don't perform any updates'-checkbox is aangevinkt en dan wordt de configuratie alleen gebruikt om records toe te voegen.

Step name: Load timeliness fact

Connection: DWH

Target table: fact_timeliness

Commit size: 1000

Don't perform any updates: ☒

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	project_key	=	project_key	
2	workorder_key	=	workorder_key	
3	employee_key	=	employee_key	
4	date_key	=	date_key	
5	time_key	=	time_key	

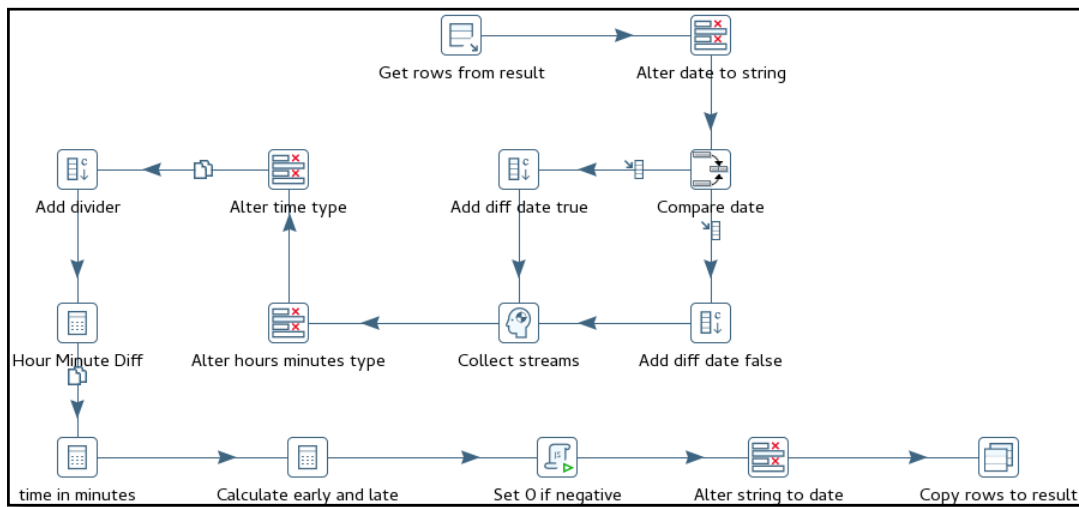
Update fields:

#	Table field	Stream field	Update
1	project_key	project_key	N
2	workorder_key	workorder_key	N
3	employee_key	employee_key	N
4	date_key	date_key	N
5	time_key	time_key	N
6	minutes_early	minutes_early	N
7	minutes_late	minutes_late	N
8	different_date	different_date	N
9	time_window_start	time_window_start	N
10	time_window_end	time_window_end	N

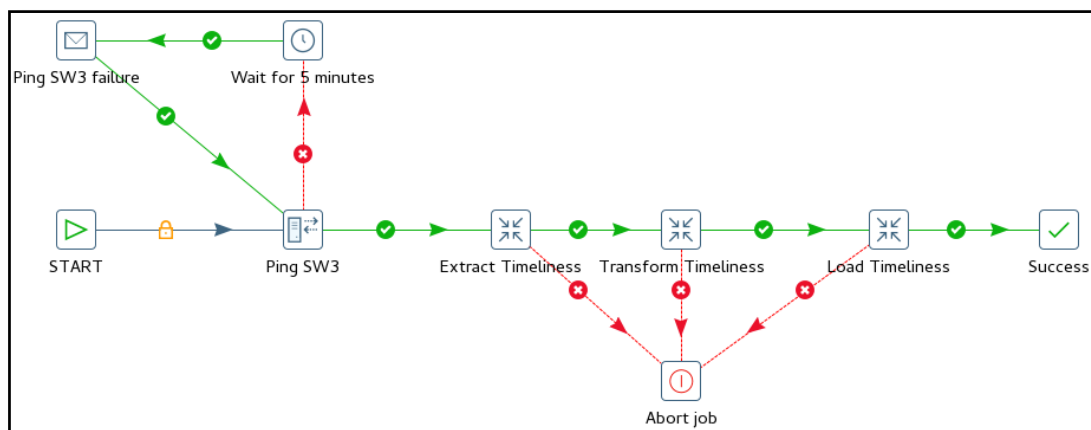
FIGUUR 27 - LOAD TIMELINESS STAP

Ten behoeve van testen voor de waarborging van kwaliteit is de transformatiejob opgesplitst in drie onderdelen, namelijk extract, transform en load. De bestaande ETL-jobs bevatten alle logica, deden de extractie, voerden de transformatie uit en laadden de data in de warehouse. Het leek een logische stap om deze drie acties op te splitsen in verschillende jobs. Dit bood namelijk de mogelijkheid om de transformatie geautomatiseerd te testen. Daarvoor is gebruikgemaakt van de technieken die zijn opgesteld door een Pentaho-Kettle-developer [17].

Met het opsplitsen van de job in drie onderdelen, was de transformatie geïsoleerd. De geïsoleerde transformatiejob is weergegeven in figuur 28. Het verschil ten opzichte van de job weergegeven in figuur 26, is dat de stappen voor de dataextractie en het laden van de data niet meer aanwezig zijn.

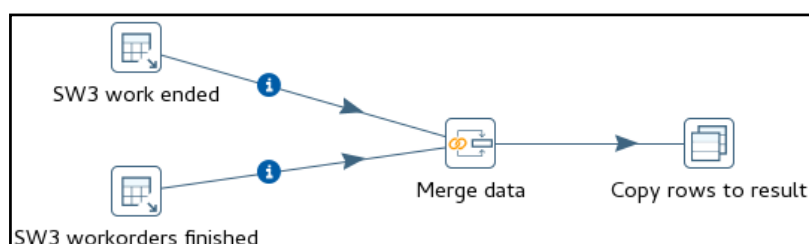


FIGUUR 28 - SPOON TIMELINESS TRANSFORMATIE



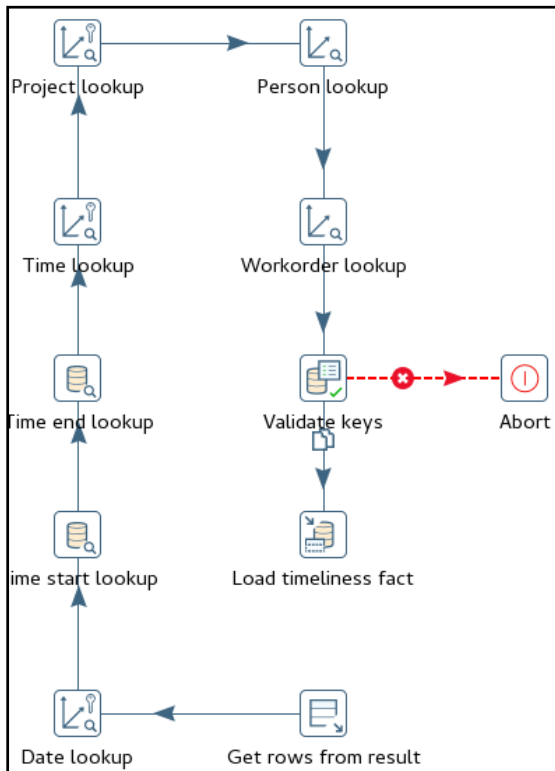
FIGUUR 29 - SPOON OVERKOEPELENDE TIMELINESS JOB

In figuur 29 staat de overkoepelende job van het timeliness feit weergegeven. Zoals is weergegeven in figuur 30 worden in 'Extract assets' eerst alle benodigde data uit de SmartWork3.0-database gelezen. In de extractiestap zit geen transformatielogica. De enige eventuele extra stap in de extractie is het samenvoegen van de opgehaalde data. De opgehaalde data worden met behulp van de component 'Copy rows to result' naar de overkoepelende job gestuurd. 'Transform assets' pakt de data op met behulp van de component 'Get rows from result'. In 'Transform assets' wordt de transformatie gedaan die in figuur 28 is weergegeven.



FIGUUR 30 - SPOON EXTRACT TIMELINESS FACT DATA

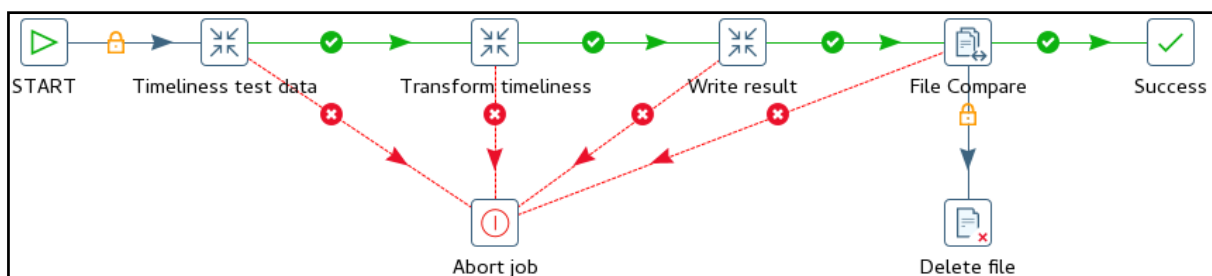
Na de transformatie wordt de data opgepakt door de 'Load timelines'-job zoals weergegeven in figuur 31. In deze job worden de keys van de dimensies opgehaald uit de warehouse zodat deze samen met de feiten kunnen worden geladen in de timeliness feitentabel. Ook in deze job zit geen transformatielogica en worden, in het geval van feitenjobs, alleen de benodigde dimensie keys opgehaald.



FIGUUR 31 - SPOON LOAD TIMELINESS DATA

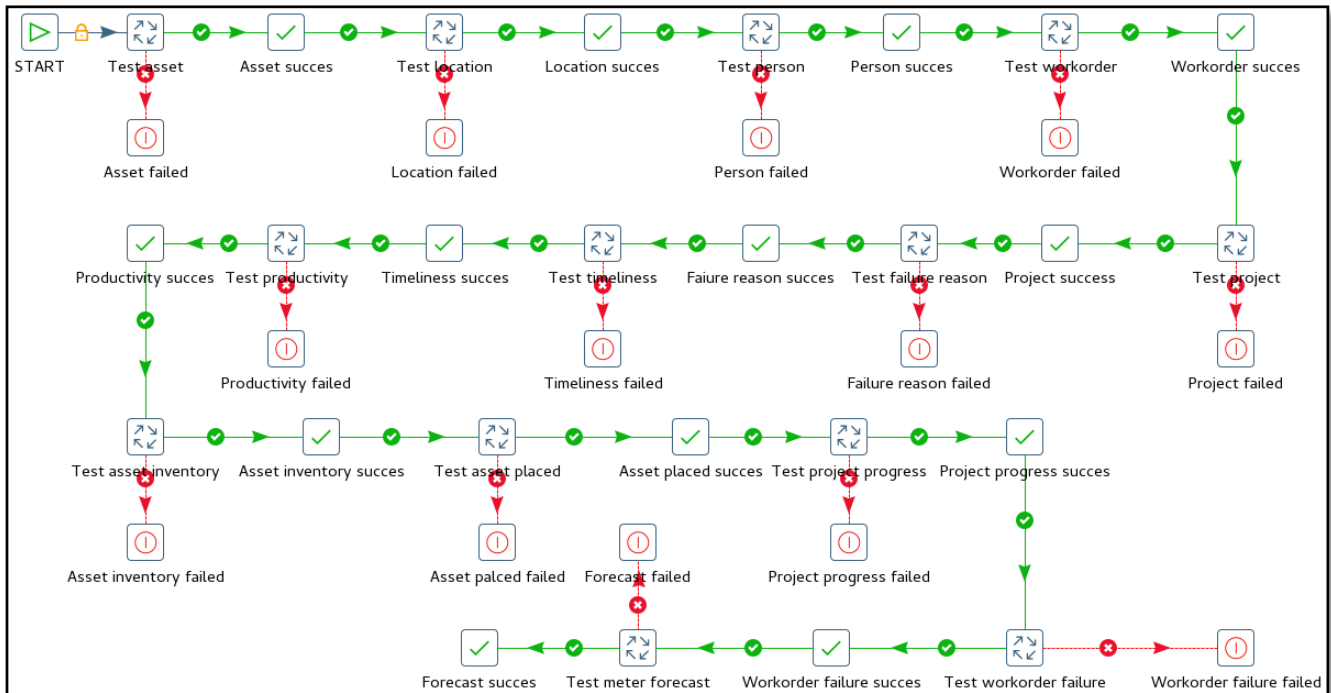
Om de transformatie te testen is een soortgelijke overkoepelende job gemaakt als weergegeven in figuur 29. Hierin zijn de extract en load jobs vervangen. De testjob is weergegeven in figuur 32. De extractjob leest in de test de data uit een met testdata gevuld CSV-bestand. Op deze data wordt dezelfde transformatie uitgevoerd als normaal. De transformatiejob voor het testen betreft dezelfde job als bij de normale job. In plaats van de data te laden in een warehouse, worden de getransformeerde data weggeschreven naar een CSV-bestand als het resultaat. In de testjob worden de data van dit bestand vergeleken met een eerder opgesteld CSV-bestand waar de verwachte outputdata instaan. Indien deze overeenkomen is de job, en dus de test, succesvol. Als ze niet overeenkomen wordt de testjob afgebroken en 'faalt' daarmee. Door de ETL-jobs op te splitsen in drie onderdelen is het mogelijk geworden om de transformatie gecontroleerd en geautomatiseerd te testen. Eerder opgestelde feiten- en dimensiejobs zijn omgebouwd volgens deze logica zodat ook deze getest konden worden.

Het bijkomende voordeel van deze testimplementaties, is dat blackboxtesten opgesteld kunnen worden door bijvoorbeeld de product owner, die bekend is met de SmartWork 3.0-data. Hij kan in een CSV-bestand goede en foute inputdata plaatsen en in een ander CSV-bestand de verwachte output. Deze bestanden kunnen vervolgens worden opgenomen in de testjobs, om aan te tonen of de verwachte output gelijk is aan de daadwerkelijke output of dat de implementatie van de transformatie niet voldoet aan de verwachtingen.



FIGUUR 32 - SPOON TIMELINESS TEST JOB

Alle testjobs zoals weergegeven in figuur 32 worden aangeroepen door een overkoepelende job. Figuur 33 geeft deze overkoepelende job weer. Met behulp van deze job worden alle testjobs uitgevoerd die de transformaties van de dimensies en feiten testen. Hiermee kan na het doorvoeren van wijzigingen eenvoudig worden achterhaald of alle transformaties nog naar verwachting werken. Een soortgelijke opzet van testjobs is gemaakt voor het uitvoeren van de extractie queries. Deze controleren de queries op hun geldigheid op een lokale SmartWork 3.0 database.



FIGUUR 33 - OVERKOEPELENDE TESTJOB

7.5 PENTAHO BUSINESS INTELLIGENCE

The screenshot shows the 'Pentaho Connection Wizard' for a database named 'SmartHouse'. The 'Database Type' is set to 'PostgreSQL'. The 'Settings' section includes: 'Host Name' (127.0.0.1), 'Database Name' (smarthouse), 'Port Number' (5432), 'User Name' (pentaho), and 'Password' (masked with dots). The 'Access' section shows 'Native (JDBC)' selected. A link 'Adding Databases' is visible below the database type list.

FIGUUR 34 - PENTAHO CONNECTION WIZARD

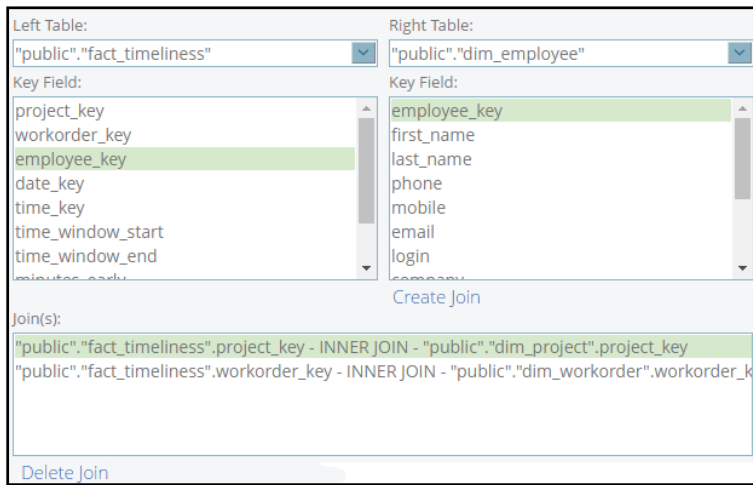
Na het bouwen van alle Kettle-jobs, kon de BI-omgeving worden geconfigureerd en konden de gewenste rapporten worden gemaakt. Alle configuraties zijn gedaan met behulp van de Pentaho-wizards. De connectie met de warehouse database is opgesteld met de connection wizard zoals weergegeven in figuur 34. PostgreSQL wordt standaard ondersteund door Pentaho BI. De configuratie van de connectie bestond uit het invullen van alle velden.

Na het maken van de connectie met de warehouse database, is de configuratie van de cubes gedaan. Om de cubes te kunnen configureren diende er een selectie gemaakt te worden uit alle beschikbare tabellen van de eerder opgestelde databaseconnectie. Zoals in figuur 35 is weergegeven moeten in deze stap de feitentabel en de bijbehorende dimensietabellen worden geselecteerd. Vervolgens moet worden aangegeven welke tabel de feitentabel betreft.

The screenshot shows the 'Pentaho Cube Wizard' with two lists of tables. The 'Available Tables' list includes: 'public"."dim_asset"', 'public"."dim_location"', 'public"."dim_status"', 'public"."fact_asset_inventory"', 'public"."fact_asset_placed"', 'public"."fact_productivity"', 'public"."fact_workorder_failure"', and 'public"."fact_workorder_progress"'. The 'Selected Tables' list includes: 'public"."fact_timeliness"', 'public"."dim_employee"', 'public"."dim_workorder"', 'public"."dim_project"', 'public"."dim_date"', and 'public"."dim_time"'. Below the lists, the 'Fact Table' is set to 'public"."fact_timeliness"'. Navigation arrows are visible between the two table lists.

FIGUUR 35 - PENTAHO CUBE WIZARD

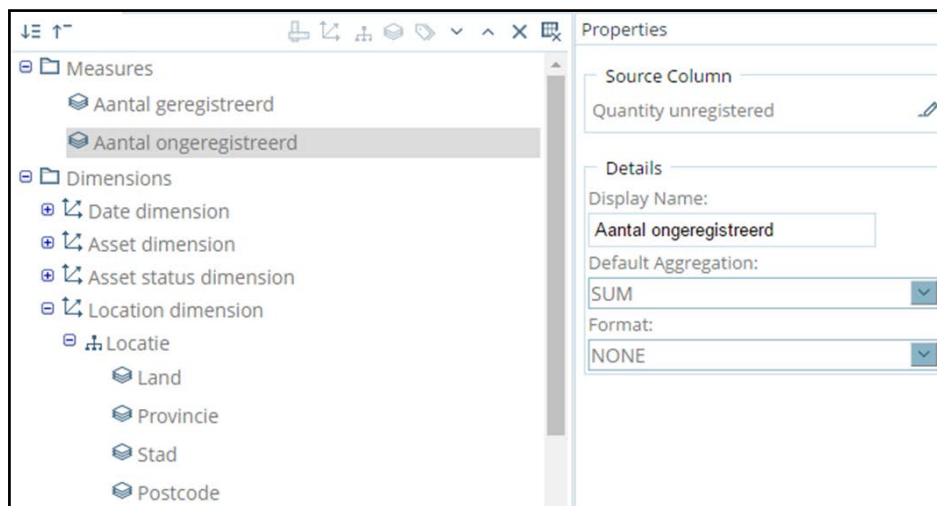
De stap die volgt na het selecteren van de tabellen is het configureren van de joins. Het enige configureerbare jointype is de inner join. De configuratie van de joins is weergegeven in figuur 36. Tijdens het configureren van de joins bleek dat Pentaho BI het niet toestaat om meer dan één join naar dezelfde dimensie te configureren. Dit beïnvloedde onder andere de timeliness feitentabel. Hier zijn de 'time_window_start'- en 'time_window_end'-kenmerken een vreemde sleutel naar de tijd dimensie, maar voor beide kenmerken kan de join niet geconfigureerd worden omdat er al een join naar de tijddimensie is geconfigureerd op het kenmerk 'time_key'. Deze beperking zorgde ook voor een beperking in de vrijheid van databaseontwerpbeslissingen en was een groot nadeel van Pentaho BI.



FIGUUR 36 - PENTAHO JOIN WIZARD

De laatste stap van de configuratiewizard was het definiëren van de aggregaties en het optioneel definiëren van de weergave van namen. Een aggregatie kan worden geconfigureerd als 'sum', 'average', 'minimum', 'maximum' of 'count'. Zowel de weergave van de naam van tabellen als van kenmerken kan worden gedefinieerd. Alle tabellen en kenmerken zijn geconfigureerd met de gewenste Nederlandse benamingen. De wizard van

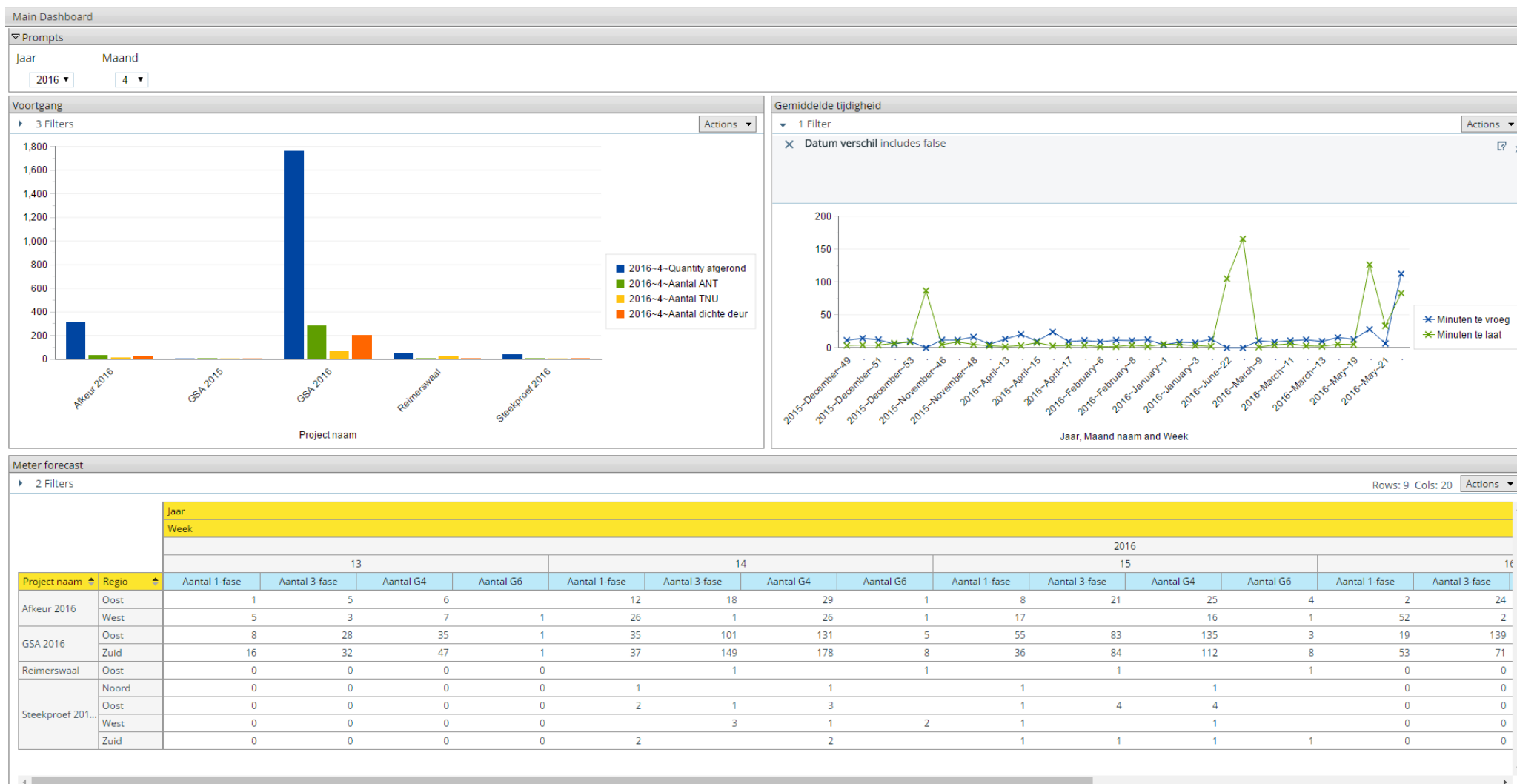
deze configuratie is weergegeven in figuur 37. In dezelfde wizard kan per tabel worden aangegeven of het een datum- of tijddimensie betreft. Ook kunnen kenmerken worden gemarkeerd als geografische kenmerken, waarbij aangegeven moet worden wat voor soort geografische informatie het betreft. Alle stappen, op de configuratie van de connectie na, zijn voor alle feitentabellen doorlopen.



FIGUUR 37 - PENTAHO PROPERTIES WIZARD

Om het hoofddashboard te kunnen realiseren dienden eerst de twee grafieken en het rapport gemaakt te worden. Deze zijn opgesteld met behulp van een drag- en dropinterface. De interface spreekt voor zich en heeft een lage leercurve. Voor zowel de grafieken als het rapport moesten de 'measures' worden geselecteerd en moest gedefinieerd worden welke kenmerken als kolom en welke kenmerken als rij getoond moesten worden. Voor de grafieken en het rapport dienden er filters geconfigureerd te worden voor jaar en maand. Deze filters zijn in het dashboard gekoppeld aan de zogenaamde 'prompts'. De prompts voor het dashboard bestaan uit twee drop-downboxen met daarin de in de database beschikbare jaren en maanden. Het dashboard is weergegeven in figuur 38.

Afstudeerverslag



FIGUUR 38 - PENTAHO MAIN DASHBOARD

8 SPRINT 3 – REALISATIE BÈTAVERSIE

Sprint 3 stond in het teken van de uitbouw van het warehouseplatform tot een bètaversie en de implementatie van de laatste requirements. In deze sprint zijn de databaseontwerpen, het warehouse design en de modules afgerond. Ook is Liquibase geïmplementeerd ten behoeve van automatisering en het onder versiebeheer zetten van de database. Tot slot is aan het eind van de sprint de performance getest en zijn er op basis van de resultaten wijzigingen doorgevoerd voor het verbeteren van de performance.

8.1 EXTRA REQUIREMENTS

Voor de start van de sprint is wederom een demonstratie van het product gegeven. Deze had als doel om de voortgang van het project en de increment ten opzichte van de vorige sprint aan te tonen en om nieuwe of gewijzigde requirements te achterhalen. In deze sprint is gewerkt aan de overgebleven user stories en de user stories die er nog bij zijn gekomen. De user stories die in sprint 3 gerealiseerd moesten worden zijn weergegeven in tabel 14.

ID	User story	Req. Nr.
SH7	Als beheerder kan ik de scheduling van de automatisch gegenereerde rapporten beheren zodat ik kan inspelen op de wens van klanten	K5,K6
SH8	Als beheerder kan ik bepalen naar wie de automatisch gegenereerde rapporten gemaïld worden zodat ik flexibel kan inspelen op de wensen van de klanten	K7
SH11	Als gebruiker kan ik de filters voor een ad-hoc rapportage opslaan zodat ik deze op een later tijdstip kan hergebruiken	K14A
SH13	Als gebruiker kan ik data visualisaties exporteren in JPEG formaat zodat ik deze buiten het platform kan gebruiken	K7A
SH14	Als gebruiker kan ik de filters en instellingen van visualisaties opslaan zodat ik deze op een later tijdstip kan hergebruiken	K6A
SH28	Als beheerder kan ik per feiten tabel zien of de uitvoering van het ETL proces succesvol is verlopen zodat ik tijdig de fouten kan herstellen	E1,E3
SH29	Als gebruiker heb ik van elk type meter inzicht in de meest recente voorspelling van het aantal benodigde meters voor de aankomende twee weken zodat ik weet of er voldoende aantal meters in voorraad zijn	E5,E6,E7 E8,E9, E10, E11, E12, E14
SH30	Als gebruiker kan ik de schatting van het aantal benodigde meters per regio en per project achterhalen zodat ik de project en regio managers op de hoogte kan stellen indien er niet voldoende meter in voorraad zijn	E13
SH31	Als gebruiker kan ik de provincie van een locatie achterhalen zodat ik weet waarom een regio niet bepaald kan worden	E4A
SH32	Als gebruiker kan ik het land van een locatie achterhalen zodat ik weet waarom een provincie niet bepaald kan worden	E5A
SH33	Als gebruiker kan ik het overzicht van de meter forecast exporteren naar een PDF bestand zodat ik dit kan opsturen naar de magazijnbeheerders	E6A

TABEL 14 - SPRINT 3 USER STORIES

8.2 LIQUIBASE

Het toepassen van databasewijzigingen op de stagingserver was een handmatig proces. Het proces bestond uit het uitvoeren van de SQL-scripts die eerder waren opgesteld om de lokale databasestructuur te wijzigen. Om dit te automatiseren is Liquibase toegepast in het project. Het bijkomende voordeel daarvan is dat de database dan tevens onder versiebeheer staat. Zo kunnen foutieve wijzigingen eenvoudig worden teruggedraaid. Liquibase is een opensource databasemanagement tool die is gepubliceerd onder de Apache 2.0-licentie. Liquibase kan worden gebruikt om databasewijzigingen te traceren, te managen en toe te passen en is een waardevolle ondersteuning voor Agile-softwaredevelopment [18]. De Liquibase-technologie is op meerdere projecten binnen SmartInstall toegepast. Echter, het toepassen van Liquibase is altijd gedaan met behulp van Maven en niet als een losstaand onderdeel.

8.2.1 LIQUIBASE-INTEGRATIE

Door de Liquibase-documentatie te lezen kon de Liquibase-technologie worden toegepast in het warehouseproject [19]. Na het downloaden en extraheren van het binary-bestand moest een aantal acties worden uitgevoerd om Liquibase in het project te kunnen toepassen. Om de database volledig onder versiebeheer te kunnen zetten zijn de eerder opgestelde tabellen in Liquibase verwerkt. Voor het toepassen van Liquibase was het opnieuw aanmaken van de warehousedatabase met behulp van liquibase nodig. Met het aanmaken van de database met behulp van Liquibase worden er twee extra tabellen aangemaakt. Deze houden bij welke changes er zijn toegepast en welke nog toegepast moeten worden.

De database wijzigen of uitbreiden met Liquibase wordt gedaan met XML-bestanden. Het master-XML-bestand bevat de verwijzingen naar de changelog-XML-bestanden die de changes bevatten. De volgorde waarin de verwijzingen staan is de volgorde van uitvoering. Code Snippet 9 geeft het master-XML-bestand weer.

```
<?xml version="1.1" encoding="UTF-8" standalone="no"?>
<databaseChangeLog>
  <include file="liquibase/changelog/20160410-initial.xml"/>
  <include file="liquibase/changelog/20160420-ddQuantityUnregisteredColumn.xml"/>
  <include file="liquibase/changelog/20160422-addIndexes.xml"/>
  <include file="liquibase/changelog/20160430-addCountyProvinceColumn.xml"/>
  <include file="liquibase/changelog/20160504-addMeterForecastFactTable.xml"/>
  <include file="liquibase/changelog/20160505-addFailureTypeColumn.xml"/>
  <include file="liquibase/changelog/20160510-renameAssetPlacedColumn.xml"/>
  <include file="liquibase/changelog/20160515-renameDimEmployee.xml"/>
  <include file="liquibase/changelog/20160515-renameEmployeeKeyColumns.xml"/>
</databaseChangeLog>
```

CODE SNIPPET 9 - LIQUIBASE MASTER-XML-BESTAND

In de changelogbestanden staan de uit te voeren wijzigingen. Dit kan in zowel XML als SQL worden gedaan. In Code Snippet 10 is de changelog weergegeven van de aanvulling van de locatiedimensie met de kenmerken land en provincie. De changelogs maken het mogelijk te achterhalen welke changes zijn uitgevoerd en wie de wijzigingen heeft doorgevoerd. Indien een wijziging of aanvulling resulteert in fouten kan de wijziging worden achterhaald en teruggedraaid. In de Liquibase properties zijn configuraties opgenomen voor het uitvoeren van de Liquibase-wijzigingen. In de properties staat wat het masterbestand is, welke database-engine er gebruikt moet worden en in welke database Liquibase de wijzigingen uit moet voeren.

```
<?xml version="1.1" encoding="UTF-8" standalone="no"?>
<databaseChangeLog>
  <changeSet author="Flip" id="1463055399632-1">
    <addColumn tableName="dim_location">
      <column name="country" type="character varying (50)"></column>
    </addColumn>
    <addColumn tableName="dim_location">
      <column name="province" type="character varying (50)"></column>
    </addColumn>
  </changeSet>
</databaseChangeLog>
```

CODE SNIPPET 10 - LIQUIBASE CHANGELOG

8.2.2 LIQUIBASE IN JENKINS

De opgestelde Liquibase-wijzigingen dienen ook uitgevoerd te worden in de database in de stagingomgeving. Dit moet plaatsvinden op het moment dat de nieuwe ETL-versie op staging wordt gezet. Voor de realisatie daarvan is de Liquibase plug-in geïnstalleerd in de Jenkinsomgeving. Vervolgens is het Jenkinsproject 'Deploy to staging' uitgebreid met de bouwstap 'Liquibase changelog' zoals weergegeven in figuur 39. Hierin is wederom geconfigureerd welke database-engine het betreft, wat het masterbestand is en in welke database de wijzigingen doorgevoerd moeten worden.

Evaluate liquibase changesets

Change Log File:
Root changeset file.

Database Engine:

Database URL:

FIGUUR 39 - LIQUIBASE JENKINS CONFIGURATIE

Na het pushen van de master branch met de Liquibase-integratie, werd er in Jenkins een overzicht gegeven van de uitgevoerde Liquibase changes zoals weergegeven in figuur 40. In het overzicht staat weergegeven of de changes succesvol zijn uitgevoerd, wat voor soort wijzigingen het betreft en wie de wijziging heeft doorgevoerd. Per changelog kan bekeken worden wat de door Liquibase gegenereerde SQL code is.

Order	ID	Status	Author	Description	Comments
1	1463054125399-1	Success	Flip	sql	
2	1463054778311-1	Success	Flip	addColumn	
3	1463055369128-1	Success	Flip	sql	
4	1463055399632-1	Success	Flip	addColumn (x2)	
5	1463055619580-1	Success	Flip	sql	

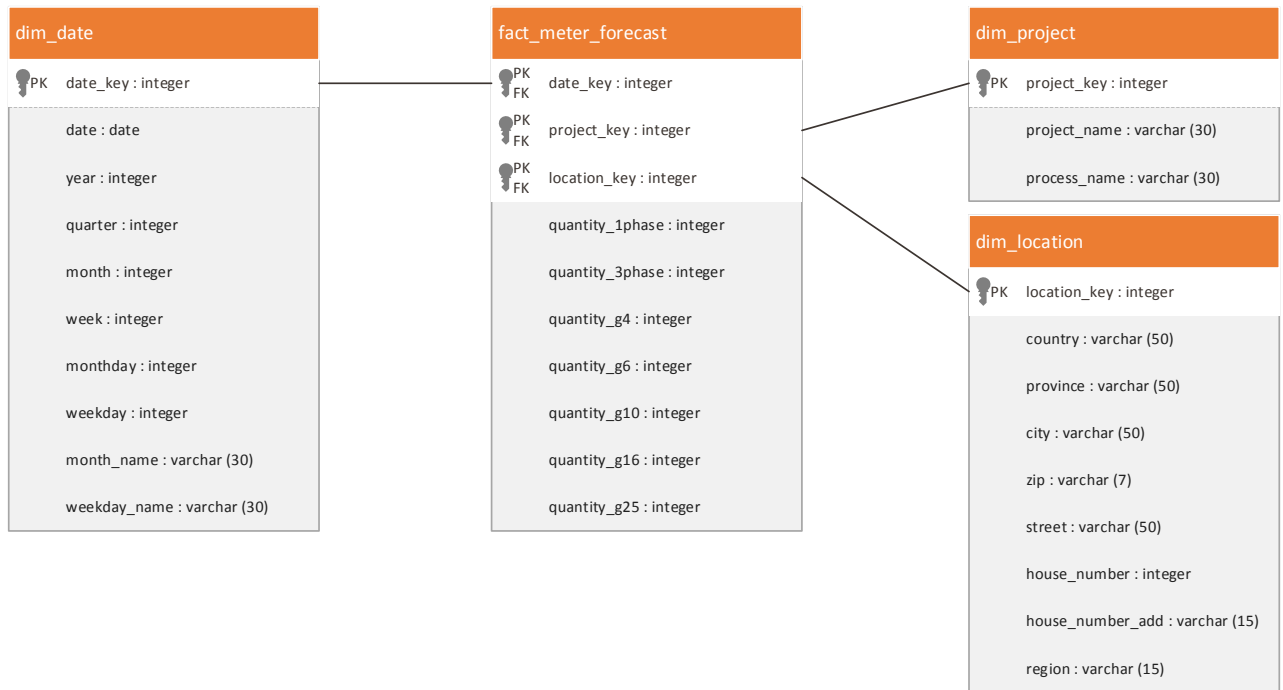
FIGUUR 40 - LIQUIBASE CHANGELOGOVERZICHT

8.3 UITBREIDEN EN AFRONDEN WAREHOUSEDESIGNS

Het uitbreiden van de warehousedesign is op dezelfde wijze gedaan als beschreven in hoofdstuk 7.4.4. Alvorens de ETL-jobs te maken en uit te breiden zijn wederom de databasemodellen en de datamapping aangepast. De mapping van de meter forecast en de dimensie wijzigingen zijn weergegeven in tabel 16. Aan de hand van de requirements zijn de feiten en de dimensies bepaald. In de requirements geven de dikgedrukte woorden de dimensies aan en de schuingedrukte, onderstreepte woorden de feiten. Met behulp van de requirements is net zoals in de voorgaande sprints een conceptual, logical en physical design gemodelleerd. Het physical design van het meter forecast feit is weergegeven in figuur 41. Tevens is de employee dimension gewijzigd naar 'person dimension'. Na de afstudeerperiode is ook data van klanten nodig, waarbij de informatie identiek is aan die van medewerkers. Ten behoeve van een meer algemene term, is de naam van betreffende dimensie gewijzigd naar person dimensie.

Nr.	Requirement
E5	Een gebruiker moet per dag een <u>voorspelling</u> van het <u>aantal benodigde meters</u> voor de aankomende twee weken kunnen inzien
E6	Een gebruiker moet kunnen achterhalen <u>hoeveel 1-fase meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E7	Een gebruiker moet kunnen achterhalen <u>hoeveel 3-fase meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E8	Een gebruiker moet kunnen achterhalen <u>hoeveel G4 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E9	Een gebruiker moet kunnen achterhalen <u>hoeveel G6 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E10	Een gebruiker moet kunnen achterhalen <u>hoeveel G10 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E11	Een gebruiker moet kunnen achterhalen <u>hoeveel G16 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E12	Een gebruiker moet kunnen achterhalen <u>hoeveel G25 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E13	Een gebruiker kan de voorspelling van het aantal benodigde meters per regio en per project inzien

TABEL 15 - METER FORECAST REQUIREMENTS



FIGUUR 41 - PHYSICAL DESIGN METER FORECAST

Target					Source				Transformation
Table	Column	Data Type	Table	SCD	Database	Table	Column	Data Type	-
dim_location	province	varchar (50)	dimension	1	Postcode	Postcode	province	varchar (50)	De provincie behorende bij de postcode dient uit de postcode database op de SmartWork server gehaald te worden. Waarde 'Onbekend' indien null
dim_location	country	varchar (50)	dimension	1	Postcode	Postcode	province	varchar (50)	Waarde 'Nederland' indien de provincie behorende bij de postcode bekend is, anders waarde 'Onbekend'.
fact_meter_forecast	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_meter_forecast	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam
fact_meter_forecast	location_key	integer	fact	N/A	DWH	dim_location	location_key	integer	lookup op zip en huisnummer
fact_meter_forecast	quantity_1phase	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien 1-fase, anders waarde '0'
fact_meter_forecast	quantity_3phase	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien 3-fase, anders waarde '0'
fact_meter_forecast	quantity_g4	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs

TABEL 16 - MAPPING VAN SPRINT 3 WIJZIGINGEN

Afstudeerverslag

fact_meter_forecast	quantity_g6	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G6, anders waarde '0'
fact_meter_forecast	quantity_g10	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G10, anders waarde '0'
fact_meter_forecast	quantity_g16	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G16,
fact_meter_forecast	quantity_g25	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G25, anders waarde '0'

8.4 UITBREIDEN EN AFRONDEN ETL-MODULE

Tijdens een van de dagelijkse Scrummeetings werd ik op de hoogte gesteld van een SmartWork 3.0 databasestructuurwijziging die impact zou hebben op de warehouse. Dit zorgde ervoor dat ik de werkzaamheden voor de meter forecast moest uitstellen tot na het doorvoeren van de nodige wijzigingen. De structuurwijziging had impact op de extractstappen van alle ETL-feitenjobs, op 'Asset placed' en 'Asset inventory' na. Dit zou over zes dagen worden doorgevoerd in productie. De wijziging betrof het verwijderen van de relatie tussen de 'BaseProcessInstance'- en 'ProcessVariable'-tabel. De 'ProcessVariable'-tabel had daardoor een relatie met de 'ProcessState'-tabel, waardoor de databasestructuur complexer werd. De wijziging werd twee dagen later in de SmartWork 3.0-stagingomgeving doorgevoerd, waardoor ik tijdig de queries kon herschrijven naar de nieuwe situatie en deze in de SmartWork 3.0-stagingomgeving kon testen. Op de dag van release van de nieuwe SmartWork 3.0-versie, is de branch met de gewijzigde queries in master gemerged, waarna Jenkins het na het succesvol uitvoeren van de tests gedeployed had op staging. Dankzij het tijdig kunnen wijzigen van de queries en het snel kunnen releasen van een nieuwe ETL-versie, waarbij de kwaliteit gewaarborgd bleef met behulp van de tests, is er geen hinder geweest voor de warehouseomgeving en bleef alles naar behoren werken. Na wijziging van de SmartWork 3.0-queries kon ik mijn werkzaamheden voor de meter forecast hervatten.

Ook heb ik in deze sprint een aantal performanceverbeteringen kunnen doorvoeren. Met het runnen van de Kettle-jobs in debug modus worden er metrics en performance-overzichten weergegeven. Met behulp van deze overzichten kon ik achterhalen in welke stappen de knelpunten zaten en deze verbeteren. Voorafgaand aan het achterhalen van de knelpunten heb ik een dataset gegenereerd met behulp van dbForge. dbForge is een tool voor het managen, onderhouden en monitoren van een MySQL database [20]. Met deze tool is het mogelijk om testdata te genereren volgens geconfigureerde regels. Het is dusdanig geconfigureerd dat er realistische testdata gegenereerd en ingeladen is in de lokale SmartWork 3.0 database dat van tienvoudige omvang is als de huidige productie database. Een groot knelpunt in alle jobs was het sorteren van de resultaten. Na de extractie van de data uit SmartWork 3.0 is de resulterende data gesorteerd op gespecificeerde kenmerken. Uit de metrics bleek echter dat het sorteren van de data in Kettle vrij lang duurde. Ik besloot om het sorteren toe te voegen aan de queries en te verwijderen uit de Kettle-job. Achteraf gezien had ik dit vanaf het begin af aan kunnen toepassen. Dit zorgde voor een performanceboost van gemiddeld vijftig seconden per sorteerstap. Ook de switch-/casestappen leken ten opzichte van andere stappen lang te duren. Ik besloot deze daarom te vervangen door een of meer filterstappen, afhankelijk van hoeveel een switch-/casestap uitvoerde. Indien een switch-/casestap uit twee mogelijke uitkomsten bestond, is deze vervangen door de filter component. Ook dit zorgde voor een performanceboost van gemiddeld 25 seconden per stap. Door deze methode op alle jobs toe te passen is veel transformatietijd bespaard.

8.5 JENKINS FINETUNING EN GEAUTOMATISEERD TESTEN

Voor de realisatie van user story SH28 is het Jenkinsproject 'Kitchen update facts' opgesplitst in meerdere projecten. Voor elk feit is een apart Jenkins project aangemaakt. Alle projecten zijn op dezelfde manier geconfigureerd als het 'Kitchen update facts'-project. De configuratie van het 'Kitchen update dimensions'-project is dusdanig gewijzigd dat het alle update feit projecten activeerde in plaats van alleen het 'Kitchen update facts'-project. Met deze wijziging ontstond er in Jenkins per feit een overzicht waarin werd aangegeven of deze succesvol was uitgevoerd of niet en hoe lang de uitvoering van het project heeft geduurd. Tevens was ook de onafhankelijkheid gerealiseerd. Indien het updaten van een feitentabel faalde had dat geen invloed op de overige updatejobs en werden deze dus gewoon uitgevoerd. In figuur 42 is het dashboardoverzicht weergegeven zoals die is terug te vinden in Jenkins.

All +				
S	W	Name ↓	Laatste geslaagd	Laatst mislukte
		Deploy to staging	4 uren 42 minuten - #62	13 dagen - #27
		Kitchen Update Asset Inventory Fact	5 uren 1 minuut - #62	8 dagen 15 uren - #40
		Kitchen Update Asset Placed Fact	5 uren 1 minuut - #62	8 dagen 15 uren - #40
		Kitchen Update Dimensions	5 uren 2 minuten - #162	7 dagen 17 uren - #140
		Kitchen Update Fact Meter Forecast	5 uren 0 minuten - #61	8 dagen 15 uren - #39
		Kitchen Update Productivity Fact	5 uren 0 minuten - #67	5 dagen 5 uren - #50
		Kitchen Update Project Progress Fact	4 uren 17 minuten - #67	4 uren 49 minuten - #66
		Kitchen Update Timeliness Fact	5 uren 0 minuten - #66	5 dagen 19 uren - #48
		Kitchen Update Workorder Failure Fact	4 uren 59 minuten - #68	15 uren - #64
		Test Kettle ETL	4 uren 43 minuten - #33	16 uren - #26

FIGUUR 42 - JENKINS DASHBOARD OPSPLITSING

Ook is het proces van deployment naar staging verbeterd. Voorafgaand aan het deployen, wilde ik dat Jenkins eerst de tests van Kettle zou uitvoeren. Het kan voorkomen dat er een wijziging wordt gepusht zonder dat de tests eerst ter controle zijn gedraaid. De wijziging kan impact hebben op eerder opgestelde jobs en kan ervoor zorgen dat deze niet meer werken als verwacht. In dat geval wordt een foutieve wijziging doorgevoerd op staging. Om dit te voorkomen is de 'Test Kettle ETL' aangemaakt. Dit project wordt geactiveerd door een wijziging te pushen naar master in plaats van het project 'Deploy to staging'. Het testproject voert bij activatie alle tests uit. Indien alle tests succesvol zijn wordt het project 'Deploy to staging' geactiveerd en worden de wijzigingen doorgevoerd naar staging. Als een of meer tests falen wordt de wijziging niet doorgevoerd naar staging.

9 CONCLUSIE EN AANBEVELINGEN

In dit hoofdstuk wordt de conclusie van het afstudeerproject beschreven en worden er aanbevelingen gedaan voor de vervolgstappen.

9.1 CONCLUSIE

Aan het einde van het afstudeerproject zijn alle producten definitief gemaakt en opgeleverd. Het warehouseplatform draait op moment van schrijven in een stagingomgeving en wordt intern gebruikt ten behoeve van bètatesten. Het project is dusdanig geconfigureerd en gedocumenteerd dat het eenvoudig overdraagbaar moet zijn aan derden door confluence te lezen en de bitbucket-repository uit te checken. Daarna kan de warehousedatabase opgezet worden door het uitvoeren van een Liquibase commando. Het doorvoeren van wijzigingen in de database en ETL-module en het testen van de transformaties in staging zijn geautomatiseerd en worden uitgevoerd bij het pushen van de wijzigingen naar de master branch.

Voorafgaand aan het project is er onderzoek gedaan naar methodieken voor het maken van de warehouseomgeving en de te gebruiken softwarepakketten. De uiteindelijke keuze voor de 'best practice'-methodieken van The Kimball Group zijn toegepast tijdens de uitvoering van het afstudeerproject. De drie gekozen softwarepakketten op basis van de pakketselectie zijn gebruikt om de warehouseomgeving te bouwen en hebben aan de verwachtingen voldaan.

Een aantal opgestelde requirements zijn niet voldaan tijdens het afstudeerproject. De requirements omtrent datacorrecties hadden een lage prioriteit en de relevantie hiervan viel weg door nieuwe functionaliteiten tijdens een release van SmartWork 3.0 die plaatsvond tijdens het afstudeerproject. Verder zijn er requirements die buiten de scope van het project vielen, maar die de stakeholders wel graag gerealiseerd willen zien. Deze worden daarom in de nabije toekomst geïmplementeerd, buiten het afstudeerproject om.

9.2 AANBEVELINGEN

Een aantal verbeteringen dienen te worden doorgevoerd voordat het warehouseplatform in productie kan worden genomen. Een van de verbeteringen betreft het opsplitsen van het 'Kitchen Update Dimension'-project in Jenkins. Het opsplitsen van dit Jenkinsproject zorgt voor een beter overzicht van succesvol uitgevoerde of gefaalde dimension ETL processen. Tevens kan Jenkins dan dusdanig geconfigureerd worden dat fact ETL Jenkins projecten geactiveerd worden indien de benodigde dimensions succesvol geüpdatet zijn. Met deze configuratie zijn de fact ETL Jenkins projecten niet meer afhankelijk van voor betreffende feit irrelevante dimensies.

Gebruikers van de Pentaho BI-suite kunnen sinds de recente update niet meer zelf hun wachtwoord wijzigen. De update van de plug-in ten behoeve van de compatibiliteit, is verkeerd gepubliceerd, waardoor Pentaho de plug-in nu niet toestaat. Deze fout wordt in de aankomende weken hersteld. Indien de plug-in een fouterstel publiceert, dient deze opnieuw geïnstalleerd te worden.

Er dient achterhaalt te worden wat de stakeholders met foutieve, uitgevallen records willen doen. Momenteel worden de records die niet door de validatie van de ETL-jobs heen komt weggeschreven naar een tekstbestand. Wellicht is er de wens dat de foutieve data beter bekend gemaakt wordt met behulp van het sturen van een e-mail naar de beheerders. Deze kunnen dan controleren wat er fout is en hebben de mogelijkheid om de data in SmartWork 3.0 te corrigeren. Het achterhalen van foutieve data kan ook worden ondersteund met een 'audit dimension'. Een audit dimension kan een of meerdere indicatoren bevatten van datakwaliteit. Deze kunnen worden afgeleid van de datakwaliteit schendingen die zijn opgetreden tijdens het ETL-proces. Ook is het mogelijk om in de audit dimensie kenmerken op te nemen die aangeven op welk tijdstip het ETL-proces is uitgevoerd, welke revisie van de ETL het betreft en dergelijke zodat een gebruiker dit in de BI-suite kan achterhalen.

10 EVALUATIE

In dit hoofdstuk wordt evaluerend teruggekeken op de uitvoering van de afstudeeropdracht.

10.1 PRODUCTEN

Ik ben tevreden over de ontwikkelde warehouseomgeving. Het feit dat het product van waarde is voor het bedrijf en het een relatief lang spelend probleem aankaart, geeft mij veel voldoening. Ik ben blij met de oplossing voor het testen van de transformaties, dit was een leuke uitdaging. Het gaf een goed gevoel om te zien dat het isoleren van onderdelen zoals geleerd voor het maken van code, toegepast op de ETL-onderdelen, zorgde voor testbare jobs en daarmee de waarborging van de functionaliteiten.

Aan het begin van het afstudeerproject had ik veel onzekerheden en wist ik niet zeker of ik de gewenste oplossing tijdig zou kunnen leveren. Mede dankzij de met de pakketselectie geselecteerde software werd mij duidelijk dat het project realiseerbaar zou zijn en ik redelijk snel iets werkends kon laten zien. Het onderzoek omtrent warehousing en het opstellen van de pakketselectie nam meer tijd in beslag dan verwacht, maar daardoor heb ik voor mijn gevoel wel een degelijke pakketselectie kunnen maken en kon ik mijn advies baseren op feiten en aan te tonen informatie.

Het meest tevreden ben ik over het geautomatiseerde deployment- en testproces dat opgesteld is met behulp van Jenkins. Dit heeft het onderhouden van de warehouseomgeving eenvoudiger gemaakt en zorgt ervoor dat toekomstige developers zonder problemen hun wijzigingen kunnen doorvoeren naar staging met de waarborging van kwaliteit. Ook de integratie van Liquibase voor het beheren van de database zorgt voor betere onderhoudbaarheid en traceerbaarheid. Ik ben van mening dat het onder versiebeheer zetten van een database een cruciaal onderdeel is van softwareontwikkelingsprojecten. Mede het eenvoudig ongedaan kunnen maken van foutieve wijzigingen in de database is een waardevolle toevoeging voor het warehouseproject.

10.2 PROCES

De Scrummethodiek is goed bevallen. Hiermee kon ik onder andere snel inspelen op de wijzigende sourcedatabase. Aan het begin van het project wist ik nog niet precies welke kant het project op zou gaan. Daarom vond ik de flexibiliteit van Scrum prettig. Het snel en vaak verkrijgen van feedback op de gemaakte onderdelen heeft de ontwikkeling van de warehouse ten goede gedaan. Zonder deze feedback zouden er op sommige vlakken verkeerde keuzes zijn gemaakt en zouden bepaalde onderdelen niet naar wens van de gebruikers zijn. Een van de directieleden van SmartInstall was eerder als stakeholder betrokken bij de ontwikkeling van een warehouse, waarbij het lang duurde voordat er een product getoond werd. Daarom verwachtte hij ook nu geen snelle realisatie. Hij hield zich in het begin op afstand van het project, totdat er na de eerste uitvoerende sprint een werkend product was. De betrokkenheid van de mensen binnen SmartInstall werd groter zodra ze zagen dat hun wensen snel gerealiseerd konden worden. Het nauwe contact met de stakeholders zorgde ervoor dat het product de kant opging die zij wilden en zorgde ervoor dat ze tijdig konden bijsturen. Het mogelijk maken van het testen van de transformaties en daarmee de blackboxtesten zorgde ervoor dat de stakeholders konden zien dat hun functionaliteit wensen daadwerkelijk gerealiseerd werden.

10.3 KENNIS

Gedurende het afstudeerproject heb ik veel kennis opgedaan omtrent warehousing. Bij de start van de afstudeerperiode had ik hier weinig tot geen kennis van en wist ik dat ik veel theorie zou moeten lezen. Ik ben blij dat ik terecht ben gekomen bij The Kimball Group, de best-practice van het warehousevakgebied. Haar boeken hebben mij veel geleerd. Ik had de afstudeeropdracht niet kunnen doen zonder de boeken van The Kimball Group te lezen. Met behulp van de boeken heb ik onder andere geleerd over ETL-technieken en dimensional modeling, wat een andere invalshoek van databasemodelleren is dan wat mij is geleerd op de Haagse Hogeschool. Ook heb ik veel geleerd van Jenkins en de diverse inzetmogelijkheden, wat na mijn afstudeerperiode ook bruikbaar is.

Ook heb ik veel moeten leren voor het gebruiken van de geselecteerde softwarepakketten. Met de PostgreSQL-database-engine had ik tot dusver slechts eenmalig gewerkt. Tijdens de werkzaamheden voor het afstudeerproject ben ik meer bekend geraakt met PostgreSQL, wat in het vervolg van mijn carrière goed van pas kan komen. De kennis die ik heb opgedaan van de ETL-tool Pentaho is in de nabije toekomst bruikbaar voor de migratieprojecten binnen SmartInstall. Voorafgaand aan het afstudeerproject wist ik niet dat dergelijke tools zo krachtig zouden zijn.

10.4 VERBETERINGEN

Warehousing is een breed begrip waar ik nog veel over kan leren. Ik heb nog veel stappen te gaan voordat ik alle theorie volledig beheers. Alle kennis die ik heb opgedaan tijdens het afstudeerproject en alle leermomenten tijdens de werkzaamheden kan ik meenemen in aankomende warehouseprojecten. De communicatie met de stakeholders is goed verlopen, maar ik zie op dit gebied nog wel een verbeterpunt. Ik heb soms nog moeite om achterliggende theorie over te brengen, waardoor mijn uitleg soms onnodig lang is.

11 VERANTWOORDING BEROEPSTAKEN

In dit hoofdstuk worden de beroepstaken verantwoordt die tijdens de afstudeerperiode zijn uitgevoerd.

11.1 SELECTEREN METHODEN, TECHNIEKEN EN TOOLS

In de opstartfase van het afstudeerproject is er onderzocht welke technieken en tools er op de markt omtrent datawarehousing. Er is een pakketselectie gemaakt voor de database, ETL tool en BI tool. Met behulp van de pakketselectie is er een advies opgesteld betreft de te gebruiken pakketten voor de warehouseoplossing. Het advies is besproken met de betrokken stakeholders, die de selectie goed hebben gekeurd. De pakketselectie is te vinden in bijlage B

11.2 UITVOEREN ANALYSE DOOR DEFINITIE VAN REQUIREMENTS

Voorafgaand aan de werkzaamheden van de uitvoeringsprints, zijn er requirements achterhaald middels diverse methoden. De requirement voor het bouwen van een prototype zijn achterhaald middels een brainstormsessie en interview met de stakeholders. Voor de tweede uitvoerende sprint zijn requirements achterhaald met behulp van een prototype. In de derde uitvoerende sprint zijn requirements achterhaald middels een demo van het product in zijn toenmalige staat. Vanaf de tweede sprint is er een impact analyse gemaakt indien een nieuwe requirement impact had op eerder opgestelde requirements. De requirements zijn getoetst op volledigheid en correctheid in samenwerking met de stakeholders. Het requirementsdocument is terug te vinden in bijlage C.

11.3 OPSTELLEN GEGEVENSMODEL VOOR EEN DATABASE

Ter ondersteuning van de warehouse werkzaamheden is er een gegevensmodel opgesteld van de sourcedatabase, smartwork3. De relaties tussen de tabellen zijn in kaart gebracht en per tabel is inhoudelijk beschreven welke informatie het betreft en wat het bestaansrecht van de tabel is. Het gegevensmodel is terug te vinden in bijlage D.

11.4 ONTWERPEN, BOUWEN EN BEVRAGEN VAN EEN DATABASE

Voor het warehouseplatform is een database ontworpen en gebouwd. Het ontwerp van de database bestaat uit een conceptual, logical en physical design. Deze drie designs zijn voor alle feitentabellen opgesteld. Op basis van deze ontwerpen is de database gebouwd in PostgreSQL. Tevens is de database gebruikt in de ETL en BI-modules. Het managen en het onder versiebeheer zetten van de database is gedaan met behulp van de Liquibase-technologie. De ontwerpen van de database zijn terug te vinden in bijlage E.

11.5 BOUWEN APPLICATIE

Voor de realisatie van het platform is gebruikgemaakt van verschillende technologieën. De ETL en BI-modules zijn opgesteld met de tools van Pentaho. De ETL-module is gebouwd in de Pentaho Kettle Engine met behulp van de Pentaho Spoon software. De ETL-module staat onder versiebeheer en wordt met behulp van Jenkins automatisch gedeployed naar een staging server. Het bouwen van de ETL is dusdanig gedaan dat deze getest kan worden middels het opstellen van testjobs. Het kunnen testen van de transformaties zorgt voor waarborging van de kwaliteit.

11.6 BEHEREN VAN CONFIGURATIES

De ETL-module en de database staan onder versiebeheer. Voor versiebeheer van de ETL-module is gebruikgemaakt van Git en voor versiebeheer van de database is gebruikgemaakt van Liquibase. Verder is er een staging server geconfigureerd waarin het warehouseplatform draait. Ook is een Jenkins omgeving geconfigureerd met een integratie van Git en Liquibase. Met behulp van Jenkins wordt met het pushen van een change naar de master branch de nieuwe versie automatisch op de stagingomgeving gedeployed. Voorafgaand aan het deployen worden eerst de tests gedraaid op Jenkins. Indien deze slagen, wordt de nieuwe versie gedeployed. Als de test falen wordt de nieuwe versie niet doorgevoerd op staging. Op de Jenkins-server is de scheduling van de ETL-jobs en het uitvoeren van database wijzingen met Liquibase geconfigureerd. Na het deployen van de nieuwe versie worden de bijbehorende database wijzigingen met behulp van Liquibase automatisch uitgevoerd. Voor de BI omgeving diende de security, database connectie en cubes geconfigureerd te worden voor gebruik van interne gebruikers.

12 BIBLIOGRAFIE

- [1] SlimmeMeters.nl, 2016. [Online]. Available: <http://www.slimmemeters.nl/wat-is-een-slimme-meter-2/>. [Geopend Mei 2016].
- [2] Rijksoverheid, „Slimme meter,” 2016. [Online]. Available: <https://www.rijksoverheid.nl/onderwerpen/slimme-meter>. [Geopend April 2016].
- [3] Enduris, 2016. [Online]. Available: <https://www.enduris.nl/consument/aannemers-en-installateurs/veilig-werken-met-beivag.htm>. [Geopend Mei 2016].
- [4] Prowareness, „Wat is Scrum,” 2016. [Online]. Available: <http://www.scrum.nl/site/Wat-is-Scrum-agile-scrum>. [Geopend April 2016].
- [5] Atlassian, „Jira,” 2016. [Online]. Available: <https://www.atlassian.com/software/jira>. [Geopend April 2016].
- [6] The Kimball Group, „About,” 2015. [Online]. Available: <http://www.kimballgroup.com/about-kimball-group/>. [Geopend April 2016].
- [7] M. R. Ralph Kimball, The Data Warehouse Toolkit, Wiley, 2013.
- [8] R. Kimball, The Data Warehouse ETL Toolkit, Wiley, 2004.
- [9] Gartner, 2016. [Online]. Available: <http://www.gartner.com/technology/home.jsp>. [Geopend Februari 2016].
- [10] Pentaho Corporation, „Pentaho Customers,” [Online]. Available: <http://www.pentaho.com/customers?tid=6&industry=All&product=All>. [Geopend Februari 2016].
- [11] Talend, „Customers,” [Online]. Available: <https://www.talend.com/customers>. [Geopend Februari 2016].
- [12] „pgAdmin,” 2016. [Online]. Available: <http://www.pgadmin.org/>. [Geopend Februari 2016].
- [13] Jenkins, 2016. [Online]. Available: <https://jenkins.io/>. [Geopend Mei 2016].
- [14] Software Freedom Conservancy, „About,” [Online]. Available: <https://git-scm.com/about>. [Geopend 2016].
- [15] Atlassian, „Software,” 2016. [Online]. Available: <https://www.atlassian.com/software>. [Geopend Mei 2016].
- [16] Atlassian, „Jenkins service management,” [Online]. Available: <https://confluence.atlassian.com/bitbucket/jenkins-service-management-251724180.html>. [Geopend 2016].

- [17] Mooreds, [Online]. Available: <http://www.mooreds.com/wordpress/archives/1033>. [Geopend 2016].
- [18] Nathan Voxland, „Liquibase,” [Online]. Available: <http://www.liquibase.org/index.html>. [Geopend April 2016].
- [19] Liquibase, 2016. [Online]. Available: <http://www.liquibase.org/documentation/index.html>. [Geopend 2016].
- [20] Devart, 2016. [Online]. Available: <https://www.devart.com/dbforge/mysql/studio/>. [Geopend Mei 2016].

BIJLAGEN

BIJLAGE A - AFSTUDEERPLAN

Afstudeerplan

Informatica

Versie : 1.5
Datum : 01/02/2016
Auteur : Jan Bongers
Nummer : 09057633

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2016-1.1 (start uiterlijk 8 februari 2016)

Startdatum uitvoering afstudeeropdracht:

Inleverdatum afstudeerdossier volgens jaarrooster: 3 juni 2016

Studentnummer: 09057633
Achternaam: dhr
Voorletters: J.A.B.
Roepnaam: Jan
Adres: Soestdijksekade 724
Postcode: 2574EB
Woonplaats: Den Haag
Telefoonnummer: ---
Mobiel nummer: 0649395052
Privé emailadres: dj_j-flip@live.nl

Opleiding: Informatica
Locatie: Den Haag
Variant: voltijd

Naam studieloopbaanbegeleider: B.M. Derks
Naam begeleidend examinerator: H.J. Middendorp
Naam tweede examinerator: H.C. van den Bosch

Naam bedrijf: SmartInstall
Afdeling bedrijf: ---
Bezoekadres bedrijf: Rietbaan 12
Postcode bezoekadres: 2908LP
Postbusnummer: 5215
Postcode postbusnummer: 2900EE
Plaats: Capelle aan den IJssel
Telefoon bedrijf: 085-2736130
Telefax bedrijf: ---
Internetsite bedrijf: www.smartinstall.nl

Achternaam opdrachtgever: dhr
Voorletters opdrachtgever: J.F.
Titulatuur opdrachtgever: drs
Functie opdrachtgever: Algemeen directeur
Doorkiesnummer opdrachtgever: 06-46077327
Email opdrachtgever: jan.taks@smartinstall.nl

Achternaam bedrijfsmentor: dhr
Voorletters bedrijfsmentor: J.F.
Titulatuur bedrijfsmentor: drs
Functie bedrijfsmentor: Algemeen directeur
Doorkiesnummer bedrijfsmentor: 06-46077327
Email bedrijfsmentor: jan.taks@smartinstall.nl

Doorkiesnummer afstudeerder: ---
Functie afstudeerder (deeltijd/duaal): ---

Titel afstudeeropdracht:

Opzetten van een datawarehouse en een rapportage & analyse platform voor SmartInstall.

Opdrachtomschrijving

1. Bedrijf

SmartInstall heeft een innovatief ICT platform opgezet en op de markt gebracht waar veel belangstelling voor is in verschillende branches. Momenteel word het platform voornamelijk gebruikt door de energie branche en de medische sector. SmartInstall biedt verschillende diensten aan en stellen het ICT platform, wat ondersteunend is voor de diensten, ter beschikking aan hun klanten. Ook stelt SmartInstall de agenda-en registratie app, SmartApp, ter beschikking voor de medewerkers van hun klanten / bedrijven. De medewerkers gebruiken de SmartApp voor het inzien van hun planning en het registreren van uitgevoerde werkzaamheden. Alle operationele en beheer werkzaamheden voor het platform worden door SmartInstall zelf verricht. De diensten die SmartInstall leveren zijn het effectief inplannen van beschikbare capaciteit / werknemers, het registreren en beheren van werkzaamheden, monitoring van effectiviteit en efficiëntie en rapporteren. Deze diensten biedt SmartInstall voornamelijk aan energie leveranciers en aan de medische sector.

Er is recentelijk een nieuw project op de markt gebracht gebaseerd op het vorige project, namelijk SmartWork 3.0. SmartWork 3.0 vervangt SmartWork 2 en is volledig opnieuw gebouwd. SmartWork3 biedt een platform om diensten beter te kunnen ondersteunen en processen meer geautomatiseerd te laten verlopen. Het nieuwe platform maakt gebruik van de nieuwste technologieën. Het is ontwikkeld in het Spring framework in combinatie met AngularJS voor de front-end. De OptaPlanner technologie is toegepast voor de plannings module. De kern van SmartWork 3.0 is het automatisch inplannen van capaciteit en het continue automatisch optimaliseren van deze planning. Ook verwerkt en controleert SmartWork 3.0 automatisch alle binnenkomende data wat geregistreerd is via SmartApp. In de nabije toekomst wil SmartInstall hun softwarepakket dusdanig uitbreiden dat ze de diensten aan kunnen bieden aan een bredere selectie van branches.

2. Probleemstelling

Eén van de diensten die SmartInstall biedt is het analyseren van onder andere de werkzaamheden en het rapporteren van de statistieken van de capaciteit die deze werkzaamheden uitvoeren. Door middel van deze analyses, rapportages en statistieken kunnen ze het uitvoeringsproces verbeteren en beter aansturen. Ook worden de rapportages gebruikt als communicatie richting de klanten voor de voortgang en knelpunten. Alle analyses en rapportages worden door SmartInstall zelf gedaan met behulp van MySQL Workbench.

Het nieuwe softwarepakket is nog niet voorzien van een goede oplossing voor de analyses en rapportages. Naast een aantal 'standaard' rapportages moet SmartInstall vaak 'ad-hoc' analyses en rapportage maken op verzoek van klanten. De 'ad-hoc' analyses en rapportages kunnen wekelijks worden gevraagd en de gewenste informatie in een dergelijk verzoek is heel afwisselend. Met een gebrek aan een goede oplossing kost dit SmartInstall veel tijd en geld. Ook willen klanten graag zelf analyses kunnen uitvoeren en statistieken kunnen inzien, wat momenteel niet mogelijk is.

Bijlage A

Er is vraag naar een oplossing voor het kunnen analyseren van data en het genereren van rapporten wat aansluit op het nieuwe softwarepakket. Met behulp van deze oplossing wil men onder andere efficiëntie en effectiviteit van monteurs kunnen bijhouden en voortgangsstatistieken en factuurrapportages kunnen maken. Ook wilt men aan de hand van de analyses voorspellingen kunnen maken. Het platform moet met dynamische, afwisselende data en variabelen kunnen omgaan en kunnen functioneren als een datawarehouse. Ook wil SmartInstall de gebruikers van het softwarepakket de mogelijkheid bieden om zelf analyses uit te kunnen voeren. Om aan de vraag te kunnen voldoen, zal er een datawarehouse en bijbehorende rapportage & analyse applicatie worden opgezet, wat zowel intern als extern gebruikt kan worden. De datawarehouse en rapportage & analyse applicatie zullen samen het nieuwe rapportage platform vormen die SmartInstall aan kan bieden aan hun klanten. Het platform moet data van meerdere klanten tegelijk kunnen ondersteunen binnen dezelfde omgeving, wat er voor zorgt dat er extra aandacht aan security besteed moet worden. De applicatie dient om data uit de datawarehouse visueel te weergeven en deze eventueel te kunnen exporteren. In de eerste versie zal de applicatie de mogelijkheid bieden om analyses uit te kunnen voeren en rapportages te kunnen genereren van onderdelen waar over het algemeen de meeste vraag naar is, namelijk de data van de energie leveranciers. In een later stadium zal de applicatie verder ontwikkeld worden zodat het om kan gaan met alle gegevens.

3. Doelstelling van de afstudeeropdracht

De afstudeerder zal in een periode van 17 weken een rapportage platform gaan ontwikkelen. De doelstellingen van de opdracht zijn als volgt:

1. Onderzoek

- Requirements rapport opleveren
- Selectie van bestaande oplossingen maken
- Adviseren over bestaande oplossingen

2. Ontwerpen en bouwen

- Beschrijving van het gegevensmodel
- Beschrijving van het Datawarehouse design
- Implementeren van het datawarehouse design
- Module voor data synchronisatie voor SmartWork 3.0
- Prototype analyse & rapportage platform
- Bèta versie analyse & rapportage platform

4. Resultaat

Indien de afstudeeropdracht succesvol is afgerond, zal dit voor SmartInstall op zowel de korte als lange termijn tijd en geld besparen met het analyseren en rapporteren. Ook kan SmartInstall meer functionaliteiten bieden in hun softwarepakket en kunnen ze hun diensten uitbreiden. Met behulp van het platform kan men onder andere inzien of werkzaamheden in een project goed verlopen of dat er meer aansturing nodig is en capaciteit beschikbaar gesteld moet worden. Dit zal uitloop van projecten voorkomen waardoor er tijd en geld bespaard worden voor zowel SmartInstall als voor de klanten van SmartInstall. Met inzicht op de efficiëntie en effectiviteit kan men inzien welke monteurs hun dagelijkse quota's niet halen en dit beter aansturen. Het platform biedt meer ondersteuning voor de diensten die SmartInstall biedt.

De opdracht is succesvol afgerond indien er een datawarehouse is opgezet en een Bèta versie van het platform in een staging omgeving kan draaien.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Binnen SmartInstall wordt er gewerkt volgens SCRUM met het TDD discipline. De uit te voeren werkzaamheden voor de afstudeeropdracht zijn verdeeld in Sprints. Indien een sprint aan het einde niet volledig is afgerond, worden de nog uit te voeren werkzaamheden meegenomen naar de volgende sprint.

Sprint 0: Onderzoek en globale analyse – Duur: 15 dagen

Activiteiten:

- 'Key' requirements van stakeholders achterhalen
- Requirements rapport opstellen
- Bestaande oplossingen / tools voor de datawarehouse zoeken en een selectie maken
- Bestaande oplossingen / tools voor het rapportage platform zoeken en een selectie maken
- Selectie bespreken met stakeholders en requirements aanpassen / verfijnen a.d.h.v. selectie

Sprint 1: Datawarehouse prototype – Duur: 15 dagen

Activiteiten:

- Opstellen database mapping
- Opstellen gegevensmodel voor de datawarehouse
- Opzetten van de datawarehouse
- Prototyping analyse & rapportage platform

Sprint 2: Detail analyse en bouwen – Duur: 25 dagen

Activiteiten:

- Extra en meer gedetailleerde requirements achterhalen m.b.v. een demo prototype
- Verwerken requirements in het requirements rapport
- Ontwerpen aanpassen a.d.h.v. nieuwe en/of gewijzigde requirements
- Platform uitbreiden / wijzigen a.d.h.v. nieuwe en/of gewijzigde requirements
- Testen van het platform

Sprint 3: Testen en opleveren – Duur: 15 dagen

Activiteiten:

- Laatste requirements achterhalen d.m.v. een demo van het platform
- Verwerken requirements in het requirements rapport
- Requirements verwerken in het platform
- Platform ontwikkelen tot een bèta versie
- Testen van het platform in een staging omgeving

Op te leveren (tussen)producten

- *Onderzoeksrapport*
- *Requirementsrapport*
- *Database design en mapping*
- *Datawarehouse*
- *Prototype rapportage platform*
- *Bèta versie rapportage platform*
- *Documentatie m.b.t. platform*

6. Te demonstreren competenties en wijze waarop

1.1 Selecteren methoden, technieken en tools

De afstudeerder dient te onderzoeken welke technieken en tools er op de markt zijn voor de datawarehouse en het rapportage & analyse platform. Met behulp van het onderzoek dien de afstudeerder een selectie te maken voor de te gebruiken oplossing en een advies uit te brengen over de keuze van de oplossing.

1.4 Uitvoeren analyse door definitie van requirements

De afstudeerder dient de wensen en eisen van de stakeholders te achterhalen en deze te verwerken in een requirements rapport. Het requirements rapport zal in de loop van het traject een aantal keer herzien en aangescherpt worden aan de hand van de selectie van bestaande oplossingen en een demo van een prototype.

2.1 Opstellen gegevensmodel voor een database

De afstudeerder zal een gegevensmodel voor de datawarehouse opstellen in de 1^e normaal vorm. Hierbij moet rekening gehouden worden met data die uit verschillende databases komen. De gegevens en zijn overeenkomsten en zullen in kaart worden gebracht door middel van een mapping.

2.2 Ontwerpen, bouwen en bevragen van een database

De afstudeerder zal op basis van het eerder opgestelde gegevensmodel een database opzetten en implementeren in de bestaande SmartWork3.0 omgeving. De database zal worden bevroegd en gemuteerd door meerdere componenten.

3.3 Bouwen applicatie

De afstudeerder zal een synchronisatie module en een prototype voor het platform bouwen. Het platform zal op basis van eerder opgestelde requirements verder worden ontwikkeld tot een bèta versie. Het platform zal aansluiten op het bestaande softwarepakket en zal worden beheerd met het huidige versiebeheer van SmartInstall. Voor het ontwikkelen zal er gebruik gemaakt worden van verschillende tools voor onder andere versie beheer.

4.1 Beheren van configuraties

De afstudeerder dient de configuratie van de synchronisatie module te beheren. De module zal in productie gaan draaien en de data van de SmartWork3.0 database naar de datawarehouse synchroniseren. De configuratie van de module dient om te kunnen gaan met eventuele wijzigingen van de SmartWork3.0 database of wijzigingen van de datawarehouse.

BIJLAGE B - PAKKETSELECTIE

PAKKETSELECTIE

SMARTINSTALL



INHOUDSOPGAVE

1	Inleiding.....	4
2	Analyse huidige situatie	5
3	Analyse gewenste situatie	6
4	Eisen en wensen	7
4.1	Algemene eisen.....	7
4.2	DBMS Eisen	8
4.3	ETL Eisen	9
4.4	BI eisen	10
5	DBMS Selectie	11
5.1	Longlist	11
5.2	Shortlist	12
5.2.1	Oracle database	13
5.2.2	IBM DB2	14
5.2.3	PostgreSQL.....	14
5.3	Conclusie	15
6	ETL Selectie	16
6.1	Longlist	16
6.2	Shortlist	17
6.2.1	Pentaho.....	18
6.2.2	Talend	19
6.3	Conclusie	20
7	BI Selectie	21
7.1	Longlist	21
7.2	Shortlist	22
7.2.1	Qlik	23
7.2.2	Pentaho.....	24
7.3	Conclusie	25
8	Advies	26
9	Bibliografie	27

TABELLENLIJST

Tabel 1 - DBMS Long list.....	11
Tabel 2 - Score toekenning	11
Tabel 3 - DBMS Selectie Matrix.....	12
Tabel 4 - ETL Tool long list.....	16
Tabel 5 - ETL Selectie Matrix.....	17
Tabel 6 - BI Long List	21
Tabel 7 - BI Selectie Matrix	22
Tabel 8 - Geadviseerde pakketten	26

1 INLEIDING

Dit document is geschreven in het kader van pakketselectie. Er wordt beschreven hoe de selectie van de software pakketten tot stand is gekomen en welke pakketten worden geadviseerd voor de data warehouse oplossing binnen SmartInstall.

Er zal een selectie worden gemaakt van softwarepakketten voor de drie modules die samen de data warehouse omgeving zullen vormen. De selectie wordt gedaan met behulp van een selectie matrix. De criteria van de selectie matrix is gebaseerd op de eisen en wensen die de stakeholders hebben voor de data warehouse omgeving. Deze criteria zal tegen de longlist selectie worden gehouden zodat er een shortlist kan worden opgesteld, bestaande uit de best scorende pakketten. Tot slot zal in het laatste hoofdstuk een advies worden gegeven betreft de te gebruiken software pakketten.

2 ANALYSE HUIDIGE SITUATIE

SmartInstall verzorgt de planning en datamanagement van energie leveranciers, zoals onder andere Enduris en Westland. De planning en datamanagement zijn geautomatiseerd en gemoderniseerd door middel van de intern ontwikkelde systemen. De systemen betreft SmartWork3.0 en SmartApp.

SmartApp is een mobiele app die de monteurs ondersteunt tijdens zijn werkzaamheden. De app maakt complete registratie van gegevens in het veld mogelijk, die van cruciaal belang zijn voor het succes van de projecten van de klanten. SmartApp communiceert met SmartWork3.0 om de geregistreerde gegevens op een volledig geautomatiseerde methode over te dragen.

SmartWork3.0 is de operationele software die alle diensten van SmartInstall ondersteunt. De core van SmartWork3.0 is de geautomatiseerde planning module, die conform restricties een efficiënte planningsroute voor de monteur zorgt.

Een van de diensten van SmartInstall is het rapporteren van de data richting de klanten en het analyseren van de data om de werkzaamheden aan te kunnen sturen en eventueel verbanden te leggen tussen de variabelen. Dit gebeurt door middel van query's op de transactionele database die in productie draait en door statische, geautomatiseerde rapporten die in SmartWork3.0 zijn gebouwd. Om in te kunnen spelen op de vraag van de klanten, worden de rapporten in SmartWork3.0 vaak aangepast en worden er regelmatig nieuwe rapporten geïmplementeerd.

SmartWork3.0 draait op een 64bit CentOS Linux omgeving ondersteund door TomCat7. In dezelfde omgeving draait ook de transactionele database. Verder heeft de server beschikking over 5 cores van de Intel Xeon chip, 16GB RAM en 128GB boot disk.

Het statische kenmerk van de rapportages zorgt er voor dat er vaak een compleet nieuw rapport geïmplementeerd dient te worden zodra een bestaande rapportagevraag wijzigt of dusdanig afwijkt, dat deze niet beantwoord kan worden met de bestaande rapportages. Dit komt doordat het van tevoren al vast staat welke specifieke vraag de rapportage beantwoordt. Het wijzigen of implementeren van rapportages kost SmartInstall kostbare IT resource en, indien het een interne vraag betreft, veel geld.

3 ANALYSE GEWENSTE SITUATIE

SmartInstall heeft de wens om een oplossing te implementeren die de huidige oplossingen van rapporteren en analyseren kan vervangen en los staat van SmartWork3.0. De huidige rapportages moeten uit SmartWork3.0 worden gefaseerd en worden ondersteund door de nieuwe oplossing. De oplossingsrichting die SmartInstall voor zich ziet, is het implementeren van een datawarehouse oplossing. Gezien de omvang van SmartInstall, speelt het budget een cruciale rol. Het maximaal te besteden budget voor de oplossing in zijn totaliteit bedraagt €1000 per maand of €12.000 per jaar. De grootte van de data van SmartInstall is zeer klein. Alle data van SW3.0 komt in totaal op nog geen 1 GB, met een totaliteit van alle applicaties bij elkaar van ongeveer 5 GB.

De vragen die SmartInstall aan de datawarehouse moet kunnen vragen, zijn vaak ad-hoc en zullen niet allemaal vast staan, echter zal het antwoord op het grootste gedeelte van de ad-hoc vragen afgeleid kunnen worden van de al bekende vraagstukken of een variatie hiervan zijn. De oplossing moet dus flexibel om kunnen gaan met constant veranderende variabelen.

Het volgende betreft een lijst van eisen die in overweging moeten worden genomen tijdens het selecteren van de pakketten:

- Het moet een 'bewezen technologie' zijn.
- Het moet een flexibel platform zijn dat om kan gaan met agile development
- De oplossing moet op een 64bit Linux systeem kunnen draaien
- Er moet volledige documentatie aanwezig zijn voor de gekozen pakket, zodat overdracht relatief snel en eenvoudig plaats kan vinden

Verder is er binnen SmartInstall nog de voorkeur dat het een open source oplossing betreft conform standaarden ontwikkeld. Indien een commerciële oplossing beter uit de selectie komt dan een open source oplossing, zal deze ook de voorkeur krijgen.

4 EISEN EN WENSEN

De datawarehouse oplossing zal bestaan uit drie modules: Een ETL/ELT¹ tool, een OLAP² ondersteunende DBMS³ en een visualisatie omgeving. Indien een pakket een oplossing biedt voor meer dan 1 van bovengenoemde modules, zal deze de voorkeur hebben boven een gelijkwaardige of iets betere oplossing voor een losstaande module.

Onderstaand overzicht geeft de criteria die worden gesteld aan de te selecteren pakketten weer. De criteria van hoofdstuk 4.1 zijn toepasbaar op alle modules en zal worden gebruikt als knock-out criteria voor de drie modules. Knock-out criteria die specifiek voor een module zijn, zijn bij desbetreffende hoofdstuk van de criteria terug te vinden.

4.1 ALGEMENE EISEN

1.1 Draait op Linux

Kan de software op een Linux omgeving draaien? Kan het op CentOS draaien?

1.2 Binnen budget

De oplossingen mogen in totaliteit niet meer dan €1.000 euro per maand of €12.000 per jaar kosten.

1.3 Documentatie

Beschikt het pakket over een volledige documentatie? Een installatie guide, user guide en technische documentatie moeten minimaal aanwezig zijn.

1.4 Community

Is het pakket een populaire keuze onder de developers en bedrijven? Is er een actieve community betreft het pakket? Is er een officiële forum of iets soortgelijks voor problemen en / of technische hulp.

1.5 Upgrade mogelijkheden

Is het mogelijk om te upgraden naar nieuwe versies met de beschikbare licenties? Is het relatief eenvoudig om te upgraden naar een nieuwe versie? Indien er een migratie dient plaats te vinden, stelt het pakket dan een migratie guide beschikbaar?

1.6 Betrouwbaarheid

Is het pakket een 'bewezen technologie'? Is het in een volwassen staat van ontwikkeling? Is de leverancier van de software een gevestigde naam?

¹ Extract, Transform, Load / Extract, Load, Transform

² Online analytical processing

³ Database management system

4.2 DBMS EISEN

2.1 Maintenance tools

Beschikt het pakket over maintenance tools? Dit houdt in dat er GUI-based maintenance kan worden gedaan / externe configuraties zonder dat er kennis van de pakketten nodig is.

2.2 Security functionaliteiten

Het pakket moet autorisatie en authenticatie bevatten. Data moeten op tabel niveau kunnen worden afgeschermd voor gebruikers door middel van permissies.

2.3 SQL support

Ondersteund het pakket de SQL⁴ taal? Is er volledige ondersteuning of gedeeltelijke ondersteuning? Indien gedeeltelijke ondersteuning, zijn alle basis functies van SQL aanwezig?

2.4 Schaalbaar

Hoe schaalbaar is het pakket? Heeft het de mogelijkheid om zowel horizontaal als verticaal te schalen?

2.5 Data types

Ondersteund het pakket de nodige data types? Heeft het pakket ondersteuning voor datum en tijd types? Bied het pakket de mogelijkheid om Integers en Strings (VARCHAR) lengtes te definiëren? Is het double type volgens de standaard gedefinieerd?

2.6 Exporteren

Bevat het pakket de mogelijkheid om data en tabellen te exporteren? Dit moet zonder externe software mogelijk zijn.

⁴ Structured Query Language

4.3 ETL EISEN

3.1 Version control

Heeft het pakket een versie controle systeem? Heeft het de check-in check-out functionaliteit? Is er in geval van afwezigheid een third-party plug-in voor de tool?

3.2 Business logica

Is het mogelijk om specifieke logica in te bouwen en te testen? Dit mag zowel met externe scripts als met een software matige plug-in.

3.3 Multi table loading

Is het mogelijk om bij het lezen van 1 databron, de data in meerdere tabellen in te laden?

3.4 Slowly changing dimensions

Ondersteund het pakket slowly changing dimensions? Het mag zowel handmatig als automatisch zijn.

3.5 Scheduler

Bevat het pakket een scheduler om ETL jobs te configureren?

3.6 Error handling

Bevat het pakket mogelijkheden om errors af te handelen bij het runnen van een job? Is het mogelijk om een andere flow te triggeren indien er een error optreedt?

3.7 Debugging

Beschikt het pakket over debugging mogelijkheden? Is het mogelijk om stap voor stap te debuggen en eventueel break points te zetten bij interessegebieden?

3.8 Connecties

Ondersteund het pakket meerdere connecties? Ondersteund het de nodige connecties? Ondersteund het pakket input naast databases?

3.9 Data kwaliteit validatie

Biedt het pakket functionaliteiten aan om de kwaliteit van de data te controleren? Zijn er ingebouwde validatie functionaliteiten?

4.4 BI EISEN

4.1 Web-based client

Het pakket moet platform onafhankelijk zijn. Windows, Linux en OS X moeten ondersteund zijn door het pakket. Web-based heeft de voorkeur.

4.2 Security

Het pakket moet authenticatie en autorisatie ondersteuning. Functionaliteiten en data moet voor gebruikers afgeschermd kunnen worden middels het toekennen van permissies en/of rollen. Ook moet het pakket een configuratie console voor gebruikers aanbieden om permissies en rollen te kunnen wijzigen.

4.3 Filteren en drill-down functionaliteit

Data moet kunnen worden gefilterd op basis van de waardes / zoek criteria. Tevens moet het pakket drill-down en drill-up functionaliteiten ondersteunen.

4.4 Visualisaties

Het pakket moet meerdere mogelijkheden tot data visualisatie ondersteunen, onder andere staaf grafieken, lijn grafieken, taarten en dergelijke.

4.5 Exporteren

Het pakket moet de mogelijkheid bieden om rapporten of grafieken te exporteren voor extern gebruik. Het pakket moet minimaal kunnen exporteren naar Excel formaten.

4.6 Rapporten / Grafieken opslaan

Het moet mogelijk zijn om rapporten en grafieken op te kunnen slaan zodat deze later te hergebruiken zijn.

4.7 Gebruiksvriendelijkheid

De gebruikers interface moet intuïtief zijn. Het is snel duidelijk hoe je het pakket moet gebruiken. De voorkeur gaat uit naar drag & drop.

4.8 Dashboard

Het pakket moet dashboard functionaliteiten bevatten. Bij voorkeur kunnen gebruikers zelf hun dashboard samenstellen.

4.9 DBMS Ondersteuning

Het pakket moet ondersteuning bieden voor veel verschillende databases. Minimaal moet het pakket de geselecteerde DBMS ondersteunen. Ook moet het om kunnen gaan met dimensies.

4.10 Scheduling

Het pakket moet de mogelijkheid bieden om rapporten automatisch te genereren op geconfigureerde tijden. Ook moet het pakket de automatisch gegenereerde rapporten per mail kunnen opsturen.

5 DBMS SELECTIE

Dit hoofdstuk beschrijft de pakketselectie voor een DBMS met behulp van een longlist en een shortlist. De drie pakketten met de hoogste score uit de selectie matrix zullen op een dieper niveau worden benaderd en besproken. Met behulp van de shortlist zal het beste pakket naar voren komen. Tot slot wordt er een conclusie getrokken betreft de te gebruiken DBMS voor de data warehouse oplossing.

5.1 LONGLIST

In Tabel 1 wordt het overzicht weergegeven van de geselecteerde software pakketten. Deze selecties vormen samen de longlist. De pakketten zijn geselecteerd op basis van populariteit en architectuur. Er is gekeken of de DBMS OLAP ondersteuning kan bieden en of het geoptimaliseerd is voor betreffende systemen. Systemen die alleen geoptimaliseerd zijn voor of gebouwd zijn rondom OLTP⁵ zijn als niet relevant beschouwd voor de longlist.

Product
Infobright
Greenplum
Microsoft SQL server
Oracle database
Teradata
IBM DB2
PostgreSQL

TABEL 1 - DBMS LONG LIST

De longlist zoals weergegeven in Tabel 1 is opgezet tegen knock-out criteria. De knock-out criteria is gebaseerd op de eisen en wensen die staan beschreven in hoofdstuk 4. Voor de selectie van een DBMS pakket zijn de algemene eisen van hoofdstuk 4.1 en de DBMS eisen van hoofdstuk 4.2 gebruikt als knock-out criteria. Een criteria heeft een score van 2, 1 of 0. Wanneer een criteria welke score krijgt, is weergegeven in Tabel 2. Het resultaat van de score toekenning aan de knock-out criteria en de totale score per pakket is weergegeven in Tabel 3.

Score	Toekenning
0	Toegekend wanneer een pakket niet voldoet aan de criteria
1	Toegekend wanneer een pakket gedeeltelijk voldoet aan de criteria of wanneer er eventueel alternatieve mogelijkheden zijn
2	Toegekend wanneer een pakket volledig aan de criteria voldoet

TABEL 2 - SCORE TOEKENNING

⁵ Online transaction processing

Software pakket	Infobright	Greenplum	Microsoft	Oracle	Teradata	IBM DB2	PostgreSQL
Algemene criteria							
1.1 Draait op CentOS	2	2	0	2	1	2	2
1.2 Binnen budget	1	1	1	1	1	1	2
1.3 Documentatie	1	2	2	2	2	2	2
1.4 Community	2	2	2	2	2	2	2
1.5 Upgrade mogelijkheden	2	2	2	2	2	2	2
1.6 Betrouwbaarheid	1	2	2	2	2	2	2
DBMS criteria							
2.1 Maintenance tools	1	2	2	2	2	2	2
2.2 Security functionaliteiten	2	2	2	2	2	2	2
2.3 SQL support	1	1	2	2	2	2	2
2.4 Schaalbaar	1	2	2	2	2	2	2
2.5 Data types	2	2	2	2	2	2	2
2.6 Exporteren	1	2	2	2	1	2	2
Totaal	17	22	21	23	21	23	24

TABEL 3 - DBMS SELECTIE MATRIX

5.2 SHORTLIST

In dit hoofdstuk worden de top drie pakketten van de selectie matrix van hoofdstuk 5.1 besproken. De top drie van de selectie matrix zijn Oracle database, IBM DB2 en PostgreSQL. Deze drie pakketten worden kort beschreven en benaderd met betrekking op betrouwbaarheid, licenties en schaalbaarheid. Ten slotte zal in de conclusie duidelijk worden welk pakket geadviseerd word om te kiezen voor het opzetten van de data warehouse.

5.2.1 ORACLE DATABASE

Oracle is een van de grootste database software bedrijven met meer dan dertig jaar ervaring in het vakgebied van databases en biedt een breed scala aan mogelijkheden aan. De meest recente versie van Oracle database die momenteel wordt aangeboden is 12c. Oracle database 12c heeft twee edities, namelijk de Enterprise Edition (EE) en de Standard Edition 2 (SE2). De gratis variant, Express Edition (XE), wordt niet meer aangeboden sinds versie 12.

Volgens de Oracle User Manual [1] worden bijna alle extra features die worden aangeboden in EE, niet aangeboden in de SE2 editie. Alle availability, performance en security features zijn niet aanwezig in de SE2 editie. De features die wel aanwezig zijn in SE2, zijn RAC⁶ en Workload management. Verder zijn alle *Windows only* features beschikbaar voor de SE2 editie, maar die zijn voor de selectie niet relevant.

Het licentie model van Oracle is redelijk complex. Ook is het niet gelijk duidelijk wat de licentie overeenkomsten zijn van de aangeboden licenties. De twee aangeboden licenties zijn processor based en user based.

De user based licentie gaat per 10 users. Onder de users vallen echter niet alleen de menselijke gebruikers. Alle devices of applicaties die de database benaderen worden ook gezien als user en dienen ook een licentie te hebben. De user based licentie mag niet gebruikt worden voor web-applicaties en connecties buiten het lokale netwerk. Dit gegeven zorgt ervoor dat de user-based licentie geen optie is.

De processor based licentie kosten liggen een stuk hoger dan de user based licentie kosten. Een processor staat gelijk aan 2 cores van een Intel Xeon CPU zoals deze op onze huidige server is [2]. Dit houdt in dat er een licentie van twee processoren nodig is om te voldoen aan de overeenkomsten. De licentie kosten van processor based licentie met twee processoren voor de EE editie bedraagt bij aankoop eenmalig \$98,515 en jaarlijks \$18,000. Voor de SE2 editie bedraagt dit bij aankoop eenmalig \$34,000 en jaarlijks \$7,500. Voor beide edities geldt een maximaal aantal users van 50, 25 per processor. De SE2 editie is niet uitbreidbaar met extra features en is alleen te upgraden naar EE. De EE editie kan wel uitgebreid worden extra features, echter zitten hier wel extra kosten aan verbonden.

De processor based licentie zorgt voor een beperking betreft schaalbaarheid. Verticaal schalen vervalt met de beperking van maximaal 4 cores. De licentie dient te worden uitgebreid met verticaal schalen. Horizontaal schalen blijft wel mogelijk. Een tweede node (computer systeem) toevoegen aan de bestaande architectuur die ook vier cores bevat, zal in theorie niet tegen de overeenkomst van twee processoren licentie gaan. Het zou echter kunnen zijn dat dit wordt gezien als draaien met 8 cores. Dit is niet duidelijk te achterhalen in de overeenkomsten van Oracle.

⁶ Real Application Clusters

5.2.2 IBM DB2

IBM is eind 19^e eeuw opgericht en is een gevestigde naam op IT gebied. De DBMS die IBM aanbiedt is DB2. Met de eerste release van DB2 in 1983, heeft IBM meer dan dertig jaar ervaring in het vakgebied van databases. De meest recente versie van IBM DB2 die momenteel wordt aangeboden is 10.5. IBM DB2 10.5 heeft in totaal zeven edities, waarvan twee relevant zijn voor de selectie. DB2 Express en DB2 Express-c. Alle overige edities liggen bij voorbaat al ver boven het budget. De editie die na DB2 Express editie komt, zal met de specificaties van de huidige server \$119.000 kosten, exclusief jaarlijkse subscriptie van support en updates.

De IBM Express editie heeft geen kosten overzicht en geen prijs kaart. Wel is te zien dat alle specificaties de helft zijn van bovengenoemde editie prijs, waardoor het aannemelijk is dat de prijs ook de helft hiervan zal zijn. Hiermee zal de prijs rond de \$60.000 liggen.

De meest aantrekkelijke editie voor SmartInstall qua budget is de IBM DB2 Express-c editie. Dit is de gratis community editie die de core database mogelijkheden van DB2 aanbiedt. De IBM Express-c editie heeft een limiet van 2 CPU cores, 16GB ram en 15 TB data per database. Naast de limitatie van 2 cores, zijn de overige specificaties voldoende voor een data warehouse omgeving binnen SmartInstall.

Beide edities van IBM DB2 bevatten geen workload management, geen complex query support en geen InfoSphere Data Architect. Deze drie componenten worden aangeraden door IBM DB2 voor een data warehouse omgeving. De licentie kosten voor DB2 Express liggen boven het budget. DB2 Express-c editie is gratis, maar oogt daarin tegen zeer beperkt in zijn features.

5.2.3 POSTGRESQL

PostgreSQL is de opvolger van Ingres en een gratis open source DBMS. De ontwikkeling van PostgreSQL begon in 1986 en kreeg zijn eerste release in 1997 als versie 6.0. PostgreSQL is momenteel nog net geen 20 jaar op de markt. De meest recente versie van PostgreSQL die momenteel wordt aangeboden is 9.5.1. PostgreSQL is sinds 9.5 geoptimaliseerd om big data en data warehousing beter te kunnen ondersteunen in vergelijking met voorgaande versies.

PostgreSQL is uitgebracht onder de PostgreSQL licentie. Deze licentie is soortgelijk aan de BSD en MIT licentie [3]. De licentie geeft toestemming om de software te gebruiken, te kopiëren, te wijzigen en te verspreiden zonder kosten, zonder schriftelijke overeenkomsten en zonder beperkingen.

De beperkingen van PostgreSQL hebben alleen betrekking op de hoeveelheid data die PostgreSQL momenteel aankan. De totale grootte van de database is ongelimiteerd, echter kan een enkele tabel maximaal 32TB groot zijn, een rij van een tabel 1.6TB en een veld van een rij 1GB. Verder zijn er geen beperkingen betreft de hardware voor PostgreSQL.

Ondanks dat PostgreSQL een decennium jonger is dan Oracle en IBM, is het ook een zeer gevestigde naam en heeft het zichzelf goed bewezen als robuuste en stabiele technologie.

5.3 CONCLUSIE

Met behulp van de shortlist kan er geconcludeerd worden dat PostgreSQL een volwaardige keus is als DBMS voor de data warehouse oplossing binnen SmartInstall. Rekening houdend met het te besteden budget, valt Oracle database af. IBM DB2 is binnen het budget te houden, echter lijkt het dan te gelimiteerd. Bovendien betreft de data binnen SmartInstall een kleine hoeveelheid en zal, gebaseerd op de afgelopen vijf jaar, de aankomende 10 jaar niet aan 1TB komen. PostgreSQL met data warehousing configuraties zal voor SmartInstall een toekomstbestendige en zeer flexibele oplossing zijn.

6 ETL SELECTIE

Dit hoofdstuk beschrijft de pakketselectie voor een ETL tool met behulp van een longlist, selectie matrix en een shortlist. De drie pakketten met de hoogste score uit de selectie matrix zullen op een dieper niveau worden benaderd en besproken. Met behulp van de shortlist zal het beste pakket naar voren komen. Tot slot wordt er een conclusie getrokken betreft de te gebruiken ETL tool voor de data warehouse oplossing.

6.1 LONGLIST

In Tabel 4 is een overzicht weergegeven van de geselecteerde ETL software pakketten. Deze selecties vormen samen de longlist. De pakketten zijn geselecteerd op basis van populariteit, community reviews en Gartner Magic Quadrant selecties [4]. Er is een bewuste keus gemaakt om drie commerciële en drie open source ETL tools te selecteren voor de longlist. Indien een commerciële tool en een open source tool een gelijke score hebben, zal de voorkeur voor de commerciële tool zijn, mits deze support levert bij de aangeboden licenties.

Product
PowerCenter Informatica
Pentaho Data Integrator
CloverETL
IBM InfoSphere Information Server
Oracle Data Integrator
Talend Open Studio

TABEL 4 - ETL TOOL LONG LIST

De longlist zoals weergegeven in Tabel 4 is opgezet tegen knock-out criteria. De knock-out criteria is wederom gebaseerd op de eisen en wensen die staan beschreven in hoofdstuk 4. Voor de selectie van een ETL tool zijn de algemene eisen van hoofdstuk 4.1 en de ETL eisen van hoofdstuk 4.3 gebruikt als knock-out criteria. Een criteria heeft wederom een score van 2, 1 of 0. De toekenning wordt op dezelfde wijze toegepast als in hoofdstuk 5. Tabel 2 weergeeft hoe een criteria gescoord wordt. Het resultaat van de score toekenning aan de knock-out criteria en de totale score per pakket is weergegeven in Tabel 5.

Software pakket	PowerCenter	Pentaho	CloverETL	Oracle	InfoSphere	Talend
Algemene criteria						
1.1 Draait op CentOS	2	2	2	2	2	2
1.2 Binnen budget	0	2	1	0	0	2
1.3 Documentatie	2	2	2	2	2	2
1.4 Community	2	2	2	2	2	2
1.5 Upgrade mogelijkheden	2	2	2	2	2	2
1.6 Betrouwbaarheid	2	2	2	2	2	2
ETL criteria						
3.1 Version control	1	1	0	2	2	1
3.2 Reusable components	2	2	2	2	1	2
3.3 Multi table loading	2	2	2	2	2	1
3.4 Slowly changing dimensions	2	2	2	2	2	2
3.5 Scheduler	2	2	2	2	2	1
3.6 Error handling	2	2	2	2	2	2
3.7 Debugging	2	1	1	2	2	2
3.8 Connections	2	2	2	2	1	2
3.9 Data kwaliteit / validatie	1	2	2	2	2	2
Totaal	26	28	26	28	26	27

TABEL 5 - ETL SELECTIE MATRIX

6.2 SHORTLIST

In dit hoofdstuk worden de top drie pakketten van de selectie matrix van hoofdstuk 6.1 besproken. De top drie van de ETL selectie matrix zijn Oracle, Pentaho en Talend. Ondanks dat met de ETL selectie matrix duidelijk is dat Oracle op elk criteria goed scoort, word Oracle niet meegenomen in de shortlist. Zoals is af te lezen, valt Oracle niet binnen het budget. De kosten voor een ODI⁷ licentie zijn op een soortgelijke wijze te berekenen als voor Oracle database in hoofdstuk 5.2.1. Het is al snel duidelijk dat de prijs voor ODI ver boven het budget uitkomt en daardoor afvalt als keus voor de ETL selectie. Met het afvallen van ODI blijven Pentaho en Talend over. De twee pakketten worden kort beschreven en benaderd met betrekking op gebruikers flexibiliteit, licenties en externe tools. Met PostgreSQL als geadviseerde selectie zoals in hoofdstuk 5 is beschreven, zal er ook worden gekeken of pakketten ondersteuning bieden voor PostgreSQL. Ten slotte zal in de conclusie duidelijk worden welk pakket geadviseerd word voor het opzetten van de data warehouse.

⁷ Oracle Data Integrator

6.2.1 PENTAHO

Pentaho is opgericht in 2004 en ontwikkelt open source software voor data integration en business intelligence. De ontwikkeling van Pentaho's data integration software, Kettle, begon ook in 2004 en kreeg zijn eerste release in 2005. De meest recente versie van het Data Integration pakket die momenteel word aangeboden is 6.0.1.0. Het software pakket voor data integration bestaat uit Kettle in combinatie met Spoon.

Ondanks dat Pentaho en de software van Pentaho relatief jong zijn, zijn ze een snel groeiende, gevestigde naam. Met verschillende succes stories van bedrijven over de hele wereld, inclusief ABN AMRO [5], valt te stellen dat de software die Pentaho aanbiedt betrouwbaar is.

Kettle word gedistribueerd onder de Apache License V2 [6]. De Apache license V2 houdt in dat de software gebruikt mag voor commerciële doeleinden, de software gemodificeerd mag worden, de software gedistribueerd mag worden en de license uitgebreid mag worden. [7].

Kettle en Spoon worden door Pentaho gratis aangeboden als community editie. Pentaho biedt ook verschillende commerciële suite edities aan, waar Kettle en Spoon in zitten. De commerciële pakketten bevatten veel handigheden voor onder andere job scheduling en bied tevens ook een oplossing voor reporting en business analysis. Pentaho stelt geen prijslijst of kosten overzicht beschikbaar, echter maken ze wel duidelijk dat ze een jaarlijkse subscriptie aanbieden voor hun commerciële edities. De professionele editie zou volgens externe bronnen rond de €10.000 per jaar zijn. Dit betreft een ongelimiteerd aantal gebruikers en developers. Op moment van schrijven is er een aanvraag voor een prijsopgave gedaan, maar hier is nog geen reactie op gegeven. Een commercieel pakket van Pentaho zal echter wel binnen het budget liggen.

Pentaho Kettle biedt ondersteuning voor PostgreSQL. PostgreSQL connecties en bulk loader zitten standaard in de software. Ook bied Pentaho ondersteuning voor de alternatieve keuzes DB2 en Oracle. Voor DB2 wordt echter geen bulk loading ondersteund.

Job scheduling kan worden gedaan met behulp van Cron of met de Pentaho software zelf. Job scheduling met Pentaho is echter alleen mogelijk met de commerciële suite. Gebruikers van Pentaho community raden aan om de open source software Jenkins te gebruiken voor job scheduling van Kettle.

6.2.2 TALEND

Talend is opgericht in 2005 en ontwikkelt open source software voor onder andere data integration. Het eerste product, Talend Open Studio, kwam in 2006 op de markt onder de GPL⁸ V2 licentie. Niet veel later bracht talend ook de Enterprise Edition uit onder een commerciële subscriptie licentie [8]. De meest recente versie van Talend Open Studio die momenteel word aangeboden is 6.1.1.

Ook Talend is een bekende naam binnen het data warehouse vakgebied, ondanks dat het ook relatief jong is. Het is een van de open source oplossingen die snel stijgt in populariteit. Talend heeft veel klanten van over de hele wereld, inclusief grote bekende namen als Groupon en Sony [9], en kan daardoor als betrouwbaar worden geacht.

Zoals eerder aangegeven, word Talend Open Studio gedistribueerd onder de GPL V2 licentie en onder een commerciële subscriptie. GPL V2 houdt in dat men met de software mag doen wat hij wilt, mits men de software ook doorgeeft of verkoopt aan derden onder deze licentie. Indien de software wordt aangepast, dient de broncode publiekelijk uitgebracht te worden. De commerciële licentie kosten begint bij ongeveer €3600 per developer per jaar. Afhankelijk van de hoeveelheid developers kan de commerciële editie binnen het budget liggen.

Talend Open Studio biedt ondersteuning voor PostgreSQL. Zowel input connecties als bulk loading componenten voor PostgreSQL worden standaard in de software meegeleverd. Talend biedt deze componenten ook standaard aan voor de alternatieve keuzes DB2 en Oracle.

Talend job scheduling gebeurt middels Cron. Talend biedt geen andere mogelijkheden aan voor job scheduling. In de documentatie van Talend wordt Jenkins wederom aangeraden als tool voor job scheduling.

⁸ GNU General Public License

6.3 CONCLUSIE

Talend en Pentaho zijn beide zeer uitgebreide ETL tools met mogelijkheden voor extra functionaliteiten middels plug-ins. Pentaho heeft als voordeel dat er geen beperking zit op het aantal developers. Beide pakketten zijn relatief jong, maar kunnen desondanks betrouwbaar worden geacht vanwege de grote bekende namen die de pakketten met succes in gebruik hebben.

Het advies is om het Pentaho pakket te gebruiken voor de warehouse oplossing. Pentaho heeft veel handigheden die missen bij Talend, zoals bijvoorbeeld de interne scheduler en de administratie console. Ook is het mogelijk om het pakket van Pentaho uit te breiden met onder andere business analysis componenten, wat een interessante oplossing kan zijn voor SmartInstall. Over het algemeen voelt Pentaho meer compleet aan dan Talend en oogt Pentaho een stuk makkelijker in gebruik dan Talend.

7 BI SELECTIE

Dit hoofdstuk beschrijft de pakketselectie voor een Business Intelligence platform met behulp van een longlist, selectie matrix en een shortlist. De drie pakketten met de hoogste score uit de selectie matrix zullen op een dieper niveau worden benaderd en besproken. Met behulp van de shortlist zal het beste pakket naar voren komen. Tot slot wordt er een conclusie getrokken betreft het te gebruiken Business Intelligence platform voor de data warehouse oplossing.

7.1 LONGLIST

In Tabel 6 is een overzicht weergegeven van de geselecteerde BI software pakketten. Deze selecties vormen samen de longlist. De pakketten zijn wederom geselecteerd op basis van populariteit en de Gartner Magic Quadrant selecties [4]. Community reviews zijn niet meegenomen in de selectie voor BI pakketten voor de longlist. Bijna alle community reviews en verwijzingen hadden een sterk promotie gehalte, waardoor de objectiviteit van dergelijke reviews in twijfel getrokken kan worden.

Product
SAP Web Intelligence
Tableau Software
Qlik BI
IBM Cognos
Pentaho Business Analytics
Birst
Board

TABEL 6 - BI LONG LIST

De longlist zoals weergegeven in Tabel 6 is opgezet tegen knock-out criteria. De knock-out criteria is ook voor de BI selectie gebaseerd op de eisen en wensen die staan beschreven in hoofdstuk 4. Voor de selectie van een BI pakket zijn de algemene eisen van hoofdstuk 4.1 en de BI eisen van hoofdstuk 4.4 gebruikt als knock-out criteria. Een criteria heeft wederom een score van 2, 1 of 0. Wanneer een criteria welke score krijgt, is weergegeven in Tabel 2. Het resultaat van de score toekenning aan de knock-out criteria en de totale score per BI pakket is weergegeven in Tabel 7.

Software pakket	SAP	Tableau	Qlik	IBM	Pentaho	Birst	Board
Algemene criteria							
1.1 Draait op CentOS	1	0	2	1	2	0	0
1.2 Binnen budget	0	1	1	0	2	0	0
1.3 Documentatie	2	2	2	2	2	1	2
1.4 Community	2	2	2	2	2	1	1
1.5 Upgrade mogelijkheden	2	2	2	2	2	2	2
1.6 Betrouwbaarheid	2	2	2	2	2	2	2
BI criteria							
4.1 Web-based client	2	2	2	2	2	2	2
4.2 Security	1	2	2	2	2	2	1
4.3 Filteren / Drill-down	2	2	2	2	2	2	2
4.4 Visualisaties	2	2	2	2	2	2	2
4.5 Exporteren	2	2	2	2	2	2	2
4.6 Rapporten / Grafieken opslaan	2	1	1	2	1	1	1
4.7 Gebruiksvriendelijkheid	2	2	2	1	2	2	2
4.8 Dashboard	1	2	1	1	1	2	2
4.9 DBMS Ondersteuning	2	2	2	2	2	2	2
4.10 Scheduling	2	1	2	2	2	1	1
Totaal	27	27	29	27	30	24	24

TABEL 7 - BI SELECTIE MATRIX

7.2 SHORTLIST

Uit de selectiematrix van Tabel 7 is duidelijk af te lezen dat Qlik en Pentaho het best scoren. In dit hoofdstuk worden deze twee pakketten korter beschreven en benaderd met betrekking op licentie en gebruikersgemak en zal er dieper worden ingegaan op security en gebruikersbeheer. Ook zal voor beide pakketten worden gekeken naar het opzetten van de web-based client. Tot slot wordt er een conclusie getrokken betreft het te gebruiken BI pakket voor de data warehouse oplossing

7.2.1 QLIK

Qlik is opgericht in 1993 onder de naam QlikTech en ontwikkelt software voor business intelligence. De ontwikkeling van Qlik's business intelligence software, QuikView, begon ook in 1993 en kreeg zijn eerste release ergens in 1994. QuikView was een PC gebaseerde desktop tool. De software kreeg in 1996 de nieuwe naam zoals het hedendaags bekend is, namelijk QlikView. De server-based webbrowser tool waar in de selectie naar gekeken is, kreeg zijn eerste release in 2005 en verving de PC gebaseerde desktop tool. De meest recente versie van het Qlik pakket die momenteel word aangeboden is 12.00.

Qlik heeft minimaal twintig jaar ervaring op gebied van business intelligence en hebben veel verschillende klanten van verschillende vakgebieden over de hele wereld. Met succes stories van grote namen als McAfee, Canon en Cisco kan vastgesteld worden dat de software die Qlik aanbiedt betrouwbaar kan worden bevonden [10].

Qlik biedt zowel gratis als betaalde edities van hun software pakketten aan. De software pakketten die aangeboden worden zijn QlikView en QlikSense. Beide pakketten bieden een oplossing voor business intelligence en lijken op het eerste gezicht zeer veel op elkaar. Het lijkt er in eerste instantie op dat QlikSense een nieuwe versie is van QlikView met een andere naam, maar dit is niet correct. QlikSense is een nieuw platform voor BI, maar is geen vervanging van QlikSense. Beide versies worden onderhouden en geüpdate, echter is QlikSense met de nieuwste technologieën ontwikkeld en is tevens ook platform onafhankelijk. Ondanks dat Qlik claimt van niet, lijkt het er op dat QlikSense uiteindelijk de vervanging is voor QlikView. Met dit gegeven kan geconcludeerd worden dat de meest recente versie van het Qlik pakket die momenteel word aangeboden, QlikSense 2.2 is.

De gratis versie van QlikSense bevat alle functionaliteiten die de betaalde versie ook heeft. Het verschil tussen beide versies is het aantal gebruikers, de opslag ruimte en de maximale applicatie grootte. De gratis versie heeft een limiet van vijf gebruikers, heeft een opslag capaciteit van 250 MB en de applicatie mag maximaal 25 MB groot zijn.

De betaalde versie van QlikSense heeft een Cloud en een Enterprise variant. De Cloud variant kost \$20 per gebruiker per maand, heeft een opslag capaciteit van 10 GB en de applicatie mag 50 MB groot zijn. De Enterprise editie heeft een prijskaart van \$1500 per token. 1 token staat gelijk aan 1 gebruiker. Met de Enterprise editie lijkt er geen limiet te zitten op de opslag capaciteit en de applicatie grootte en is de server in eigen beheer.

Na dieper graven in de functionaliteiten van de verschillende Qlik pakketten is het duidelijk dat de aangeboden functionaliteiten niet allemaal in hetzelfde pakket zitten. De score van selectie matrix zoals weergegeven in Tabel 7 voor Qlik lijkt dus voor een combinatie van de pakketten te zijn en niet voor 1 pakket.

7.2.2 PENTAHO

Naast het ETL software pakket zoals beschreven in hoofdstuk 6.2.1, biedt Pentaho ook een software pakket aan voor BI oplossingen. Het pakket dat Pentaho aanbiedt is de Pentaho BI suite en bestaat uit vier onderdelen. De vier onderdelen zijn de BI server, Metadata Editor, Design Studio en Pentaho Administration Console. De drie laatst genoemde pakketten zijn client tools waarmee de BI server omgeving kan worden ingericht. Pentaho BI suite kreeg zijn eerste release in 2006 en werd in hetzelfde jaar uitgebreid met veel andere onderdelen tot wat het nu is. De meest recente versie van Pentaho BI suite die momenteel wordt aangeboden is 6.0.1.0.

Pentaho BI suite wordt gedistribueerd onder de Mozilla Public License V1.1 [6]. De Mozilla Public License houdt in dat de software gebruikt mag worden voor commerciële doeleinden, de software gedistribueerd mag worden en dat de software gemodificeerd mag worden. Indien iemand de software modificeert, dient de source code van de modificatie publiekelijk beschikbaar gesteld te worden en gelden alle bovengenoemde punten van de Mozilla Public License voor de modificatie code [11].

Pentaho BI suite wordt aangeboden als een gratis community editie en een betaalde commerciële editie. De prijs van de commerciële editie is besproken in hoofdstuk 6.2.1 en zal binnen het budget van SmartInstall zijn. De community editie biedt evenveel functionaliteiten als de commerciële editie, echter zijn er wederom een aantal handigheden die de community editie niet bevat. De interne job scheduling voor het automatisch aanmaken en mailen van rapporten wordt niet aangeboden in de community editie. Job scheduling is echter wel mogelijk met de community editie met behulp van plug-ins of third-party add-on. De Administration console geeft de mogelijkheid om gebruikers binnen de BI omgeving te beheren. Aan gebruikers kunnen rollen worden toegewezen die aangeven wat een gebruiker wel en niet mag doen of zien. Ook de rollen kunnen worden beheerd met de Administration console. Aan de rollen kunnen permissies worden toegewezen waarvan de gebruiker zelf bepaald wat deze inhouden.

Rapporten en visualisaties worden aan de gebruiker aangeboden via een webbrowser, wat er voor zorgt dat het pakket platform onafhankelijk is. De rapporten en visualisaties worden gemaakt met de client tools Design Studio. In design studio kunnen rapporten en visualisaties worden gedeployed naar de BI server en met de Administration console is het mogelijk om deze rapporten en visualisaties te configureren. De Administration console is redelijk intuïtief. Het configureren en beheren van de BI omgeving kan daardoor eenvoudig worden overgedragen.

7.3 CONCLUSIE

Qlik en Pentaho zien er beide als zeer intuïtief uit en lijken beide eenvoudig in gebruik. Beide BI pakketten bieden veel mogelijkheden voor rapporten en data visualisatie. Als budget geen criteria is, zijn beide pakketten volgens de selectie matrix identiek qua functionaliteiten en zouden in theorie alle twee net zo goed moeten zijn voor de oplossing binnen SmartInstall.

Zoals in hoofdstuk 7.2.1 beschreven staat, lijken de functionaliteiten die Qlik aanbiedt niet in 1 pakket te zitten. QlikSense is de meest recente BI oplossing die Qlik aanbiedt, maar mist veel functionaliteiten die QlikView wel heeft. De rapport mail scheduler voor QlikSense zit niet standaard in het pakket en dient tegen extra (hoge) licentie kosten aangeschaft te worden. Zo zijn er meer functionaliteiten in QlikSense die achteraf aangeschaft dienen te worden, waardoor het pakket uiteindelijk buiten het budget lijkt te vallen. Het advies is dan ook om het Pentaho pakket te gebruiken als BI software voor de warehouse oplossing.

8 Advies

Op basis van de selecties van hoofdstuk 5, 6 en 7 kan er een advies worden opgesteld voor de te gebruiken pakketten voor de data warehouse oplossing. De drie pakketten zullen samen de data warehouse omgeving vormen. In Tabel 8 zijn de geadviseerde pakketten weergegeven die als beste uit de selecties zijn gekomen.

Module	Product	Versie
DBMS	PostgreSQL	9.5.1
ETL	Pentaho Data Integration	6.0.1.0
BI	Pentaho BI Suite	6.0.1.0

TABEL 8 - GEADVISEERDE PAKKETTEN

Voor zowel ETL als BI is het advies om de software van Pentaho te gebruiken. Voor beide modules is een gratis community editie beschikbaar. De community edities kunnen geüpgraded worden naar de Enterprise edities indien de functionaliteiten van de Enterprise edities nodig blijken te zijn. Het voordeel dat beide modules van hetzelfde bedrijf afkomstig zijn, is dat de aanschaf van een Enterprise licentie voor beide modules geldt en dus een eenmalige uitgave zal zijn.

De geadviseerde DBMS is PostgreSQL. PostgreSQL is een gratis open source software pakket en zal nooit extra kosten opleveren voor SmartInstall. Naast het feit dat PostgreSQL als best uit de selectie komt, heeft de keuze voor PostgreSQL het voordeel dat SmartInstall werknemers in dienst heeft met veel PostgreSQL ervaring. Dit zorgt er voor dat de nodige kennis voor configuratie aanwezig is en dat overdracht van DBMS onderhoud ook sneller plaats kan vinden.

Zowel Pentaho Data Integration als Pentaho BI Suite ondersteunen PostgreSQL en hebben beide standaard meegeleverde PostgreSQL functionaliteiten. De drie geadviseerde pakketten kunnen dus samen gecombineerd worden. Het advies is dan ook om met de geselecteerde pakketten zoals weergegeven in Tabel 8 te gaan gebruiken voor het bouwen van het warehouse prototype.

9 BIBLIOGRAFIE

- [1] Oracle Corporation, „Licensing Information User Manual,” December 2015. [Online]. Available: <https://docs.oracle.com/database/121/DBLIC/E49208-13.pdf>. [Geopend February 2016].
- [2] Oracle Corporation, „Processor core factor table,” 16 Maart 2009. [Online]. Available: <http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>. [Geopend Februari 2016].
- [3] The PostgreSQL Global Development Group, „PostgreSQL Licence,” [Online]. Available: <http://www.postgresql.org/about/licence/>. [Geopend Februari 2016].
- [4] Gartner, [Online]. Available: <http://www.gartner.com/technology/home.jsp>. [Geopend Februari 2016].
- [5] Pentaho Corporation, „Pentaho customers,” [Online]. Available: <http://www.pentaho.com/customers?tid=6&industry=All&product=All>. [Geopend Februari 2016].
- [6] Pentaho Corporation, „Pentaho Open Source Licenses,” [Online]. Available: <http://www.pentaho.com/license>. [Geopend Februari 2016].
- [7] The Apache Software Foundation, „Apache License,” Januari 2004. [Online]. Available: <http://www.apache.org/licenses/LICENSE-2.0.html>. [Geopend Februari 2016].
- [8] J. Browen, Getting Started with Talend Open Studio for Data Integration, Packt Publishing, 2012.
- [9] Talend, „Customers,” [Online]. Available: <https://www.talend.com/customers>. [Geopend Februari 2016].
- [10] „Qlik Customers,” QlikTech International AB, [Online]. Available: <http://global.qlik.com/nl/explore/customers>. [Geopend Februari 2016].
- [11] Mozilla, „Mozilla Public License,” [Online]. Available: <https://www.mozilla.org/en-US/MPL/1.1/>. [Geopend Februari 2016].
- [12] Oracle Corporation, „ODI and OWB Statement of Direction,” Oktober 2013. [Online]. Available: <http://www.oracle.com/us/products/middleware/data-integration/odi-statement-of-direction-1922235.pdf>. [Geopend Februari 2016].

BIJLAGE C - REQUIREMENTS

REQUIREMENTS

SMARTINSTALL



Jan Bongers
09057633
Versie: 1.0
Datum: 6/5/2016

INHOUDSOPGAVE

1	Inleiding.....	4
2	Requirements.....	5
2.1	Key Requirements.....	6
2.2	Prototype Requirements	8
2.2.1	Requirements overzicht	8
2.2.2	wijzigingen en bijzonderheden.....	10
2.3	Extra requirements	12
2.3.1	Requirements overzicht	12
2.3.2	Requirement wijzigingen.....	13
3	User Stories	14

TABELLENLIJST

Tabel 1 - Requirement nummering	5
Tabel 2 - Key requirements	7
Tabel 3 - Prototype requirements	9
Tabel 4 - Extra requirements	12
Tabel 5 - User stories van key requirements	14
Tabel 6 - User stories van prototype requirements	15
Tabel 7 - User Stories van extra requirements	16

1 INLEIDING

Dit document is geschreven in het kader van requirements. Het document geeft een duidelijk overzicht van alle requirements voor de data warehouseoplossing en een overzicht van user stories die gebaseerd zijn op de requirements. De requirements zijn verkregen met behulp van een brainstormsessie met en het interviewen van de betrokken stakeholders. Ook zijn er requirements achterhaald door middel van het geven van een demonstratie van een prototype. De requirements zijn gebaseerd op tijdens de sessie en interviews geselecteerde business processen. Op basis van deze processen is de granulariteit bepaald waaruit de requirements rondom de feiten en dimensies naar voren kwamen. De user stories zijn in de Jira backlog geprioriteerd op basis van de belangen van de stakeholders. De bijbehorende requirements van de user stories die afgeleid zijn van de betreffende requirement zijn terug te vinden met behulp van het requirementnummer.

2 REQUIREMENTS

Tabel 1, Tabel 3 en Tabel 4 geven overzichten van alle requirements weer. De requirements zijn genummerd op basis van moment van verkrijgen. De nummering van de requirement geeft aan op welk moment de requirement is verkregen. In Tabel 1 staat weergegeven wat de nummering van een requirement inhoudt, waarbij XX het nummer van een requirement representeert.

Nummering	Beschrijving
KXX	Key requirements die bij de eerste meeting zijn verkregen.
KXXA	Key requirements die zijn verkregen naar aanleiding van de uitwerking van de key requirements. Dit kunnen requirements zijn die in eerste instantie niet naar voren zijn gekomen of die niet genotuleerd zijn.
PXX	Requirements die zijn verkregen aan de hand van een demonstratie van het datawarehouseprototype.
PXXA	Requirements die zijn verkregen naar aanleiding van de uitwerking van de prototyperequirements. Dit kunnen requirements zijn die in eerste instantie niet naar voren zijn gekomen of niet genotuleerd zijn.
EXX	Extra requirements die zijn verkregen na de oplevering van de datawarehouseomgeving aan het einde van een sprint.
EXXA	Requirements die zijn verkregen naar aanleiding van de uitwerking van de extra requirements. Dit kunnen requirements zijn die in eerste instantie niet naar voren zijn gekomen of niet genotuleerd zijn.

TABEL 1 - REQUIREMENT NUMMERING

2.1 KEY REQUIREMENTS

Tabel 2 geeft een overzicht van requirements weer die zijn verkregen aan de hand van een brainstormsessie en interviews met de stakeholders. Requirements met nummer KXXA zijn verkregen tijdens het bespreken van de uitgewerkte requirements.

Nr.	Requirement
K1	Het systeem moet in een CentOS omgeving draaien
K2	Het systeem moet op database niveau de SQL taal ondersteunen
K3	Het systeem moet dagelijks benodigde data van SmartWork3 naar de warehouse synchroniseren
K4	De beheerder moet het tijdstip van synchronisatie kunnen configureren
K5	Het systeem moet rapporten automatisch kunnen genereren op zelf gedefinieerde datum en tijd
K6	De beheerder kan de tijd en datum van de automatisch te genereren rapporten configureren
K7	Het systeem moet automatisch gegenereerde rapporten kunnen mailen naar gebruikers
K8	Een gebruiker moet per dag en per project kunnen inzien wat de voortgang is, inclusief huidige status van de werkopdrachten
K9	Een gebruiker kan per dag, per project zien hoeveel werkopdrachten er succesvol afgerond zijn
K10	Een gebruiker kan per dag, per project zien hoeveel dichte deuren er zijn gemeld
K11	Een gebruiker kan per dag, per project zien hoeveel adressen niet toegankelijk waren
K12	Een gebruiker kan per dag, per project zien hoeveel werkopdrachten er technisch niet uitvoerbaar waren
K13	Een gebruiker moet een dagelijkse snapshot van de inventaris van alle meters per medewerker kunnen inzien
K14	Een gebruiker moet de mogelijkheid hebben om de snapshot van de inventaris van alle meters te filteren op monteur en status
K15	Een gebruiker kan de periode van de snapshot van de inventaris van alle meters selecteren, waarbij de minimale selectie een dag en de maximale selectie jaren is.
K16	Een gebruiker moet het aantal meters kunnen inzien die een medewerker op zijn naam heeft geregistreerd
K17	Een gebruiker kan een datum van de rapporten filteren op de naam van een maand
K18	Een gebruiker kan de datum van de rapporten filteren op de week van het jaar
K19	Een gebruiker moet een overzicht van alle geplaatste meters per monteur, per dag kunnen inzien
K20	Een gebruiker kan het aantal meters inzien die een monteur heeft geplaatst
K21	Een gebruiker moet een overzicht van alle geplaatste meters per project kunnen inzien
K22	Een gebruiker moet het overzicht van alle geplaatste meters kunnen filteren op locatie
K23	Een gebruiker kan is het geplaatste meters overzicht achterhalen bij welke werkopdracht een meter geplaatst is

K24	Een gebruiker heeft bij een locatie dezelfde informatie tot zijn beschikking als bij de brondata
K25	Een gebruiker moet rapporten kunnen exporteren naar CSV formaat
K26	Een gebruiker moet rapporten kunnen exporteren naar XLSX formaat
K27	Een gebruiker krijgt door middel van een dashboard inzicht in alle belangrijke en meest relevantie informatie
K28	Het systeem moet een staafdiagram visualisatie kunnen weergeven
K29	Het systeem moet een lijngrafiek visualisatie kunnen weergeven
K30	Het systeem moet een taartdiagram visualisatie kunnen weergeven
K1A	Het systeem moet platform onafhankelijk zijn
K2A	Het systeem moet tijdens het synchroniseren alle data controleren en valideren op kwaliteit
K3A	Een gebruiker heeft de mogelijkheid om het dashboard naar eigen wensen in te vullen en vorm te geven
K4A	Een gebruiker kan rapporten maken op basis van de bestaande warehouse data
K5A	Een gebruiker kan visualisaties maken op basis van de bestaande warehouse data
K6A	Een gebruiker kan gemaakte visualisaties opslaan en hergebruiken
K7A	Een gebruiker kan visualisaties exporten in JPEG formaat voor gebruik buiten het platform
K8A	De beheerder moet gebruikers kunnen toevoegen, wijzigen en verwijderen
K9A	De beheerder moet gebruikersrollen kunnen toevoegen, wijzigen en verwijderen
K10A	De beheerder moet permissies kunnen toevoegen, wijzigen en verwijderen
K11A	Het systeem moet functionaliteiten tonen of verbergen aan de hand van rollen en permissies
K12A	Het systeem moet data tonen of verbergen aan de hand van rollen en permissies
K13A	Een gebruiker kan achterhalen aan welk SmartWork3.0 proces een project gekoppeld is
K14A	Een gebruiker moet de gemaakte rapportages kunnen opslaan en hergebruiken
K15A	Een gebruiker heeft bij een medewerker dezelfde informatie tot zijn beschikking als bij de brondata

TABEL 2 - KEY REQUIREMENTS

2.2 PROTOTYPE REQUIREMENTS

Tabel 3 van hoofdstuk 2.2.1 geeft een overzicht van requirements weer die zijn verkregen aan de hand van een demonstratie van het prototype. De kolom 'Impact' geeft aan of de nieuwe requirement eventueel een aanvulling, wijziging of in conflict is met een requirement of de realisatie van een requirement uit hoofdstuk 2.1. Nieuwe requirements met een impact op eerder geformuleerde requirements worden besproken in hoofdstuk 2.2.2

2.2.1 REQUIREMENTS OVERZICHT

Nr.	Requirement	Impact
P1	Een gebruiker moet een dagelijks overzicht van de tijdigheid van per monteur kunnen zien	-
P2	Het systeem moet berekenen hoeveel minuten een monteur eventueel te vroeg bij een afspraak is	-
P3	Het systeem moet berekenen hoeveel minuten een monteur eventueel te laat bij een afspraak is	-
P4	Een gebruiker moet het aantal minuten te vroeg bij afspraken per monteur kunnen zien	-
P5	Een gebruiker moet het aantal minuten te laat bij afspraken per monteur kunnen zien	-
P6	Een gebruiker kan bij een tijdigheid overzicht inzien wat de ingeplande starttijd van het tijdsblok van de afspraak is	-
P7	Een gebruiker kan bij een tijdigheid overzicht inzien wat de ingeplande eindtijd van het tijdsblok van de afspraak is	-
P8	Een gebruiker kan per monteur het aantal minuten te laat visueel inzien met een lijndiagram	-
P9	Een gebruiker kan per monteur het aantal minuten te vroeg visueel inzien met een lijndiagram	-
P10	Een gebruiker moet bij een werkopdracht de aanleiding kunnen inzien	K8,K23
P11	Een gebruiker moet bij een werkopdracht het metertype kunnen inzien	K8,K23
P12	Een gebruiker moet de regio waar de meter is geplaatst kunnen inzien	K22,K24
P13	Een gebruiker kan de aantallen van de voortgang visueel inzien met een taartdiagram	-
P14	Een gebruiker moet per dag, per project een rapport van alle uitgevallen werkopdrachten kunnen inzien	-
P15	Een gebruiker kan in het rapport van uitgevallen werkopdrachten achterhalen wat de reden van uitval is	-
P16	Een gebruiker kan in het rapport van uitgevallen werkopdrachten achterhalen welke werknemer de uitval heeft geregistreerd	-
P17	Een gebruiker kan het commentaar van de uitval registratie bij uitgevallen werkopdrachten inzien	-
P18	Een gebruiker kan per project het aantal uitgevallen werkopdrachten visueel inzien middels een staafdiagram	-
P19	Het systeem moet een e-mail sturen naar de beheerder indien er een fout optreedt bij het geautomatiseerde ETL proces	-
P20	Het systeem moet een status overzicht weergeven van alle synchronisatie jobs	-
P21	Een gebruiker kan inzien hoe vaak een monteur een correctie uit heeft moeten voeren op zijn geregistreerde werkzaamheden	-

Bijlage C - Requirements

P22	Een gebruiker moet de productiviteit per dag van een monteur kunnen inzien	-
P23	Een gebruiker kan bij de productiviteit per dag, per monteur zien wat het aantal beschikbaar gestelde uren was	-
P24	Een gebruiker kan inzien hoeveel solo opdrachten een monteur per dag heeft uitgevoerd	-
P25	Een gebruiker kan inzien hoeveel combi opdrachten een monteur per dag heeft uitgevoerd	-
P26	Een gebruiker kan inzien welke meerwerk werkzaamheden een monteur heeft gerealiseerd	-
P27	Een gebruiker kan per meerwerk inzien hoe vaak een monteur betreffende meerwerk op een dag heeft uitgevoerd	-
P28	Een gebruiker kan per monteur het productiviteitsoverzicht visueel inzien middels een staafdiagram	-
P29	Een gebruiker kan bij het visuele overzicht van de productiviteit aangeven over welke periode hij het overzicht wilt zien	-
P30	Het systeem moet dagelijks benodigde data van SmartQuality naar de warehouse synchroniseren	-
P31	Een gebruiker kan de kwaliteit van de werkzaamheden van een monteur inzien	-
P32	Een gebruiker kan de klant tevredenheid inzien per dag, per werkopdracht inzien	-
P33	Een gebruiker kan inzien hoeveel uitval registraties een monteur heeft gedaan	-
P1A	Een gebruiker kan inzien bij welke aannemer een monteur werkzaam is	K14, K19
P2A	Het systeem moet bijhouden bij welke aannemer een monteur werkzaam was indien een monteur van aannemer wisselt	-
P3A	Een gebruiker kan het tijdstip van registratie van uitval inzien	-
P4A	Een gebruiker kan zien of de geregistreerde telwerken binnen de gedefinieerde marge valt	-
P5A	Een gebruiker moet het aantal ongeregistreerde geplaatste meters kunnen achterhalen	K19, K21
P6A	Het systeem moet het aantal beschikbaar gestelde uren voor productiviteit op de minuut nauwkeurig berekenen	-
P7A	Het systeem moet met het berekenen van het aan beschikbaar gestelde uren rekening houden met eventueel vrije uren of dagen van een monteur	-
P8A	Een gebruiker moet de tijdigheid van monteurs per project kunnen inzien	-
P9A	Een gebruiker moet het tijdstip van de uitval registratie in een uitval rapport kunnen achterhalen	-
P10A	Een gebruiker kan het aantal correcties per project inzien	-
P11A	Een gebruiker kan het aantal correcties over een geselecteerde periode inzien	-
P12A	Een gebruiker kan zien welke werkopdracht het betreft in het tijdigheid overzicht en wat het tijdstip van de afspraak was	-
P13A	Een gebruiker moet kunnen achterhalen voor welk project het meerwerk is uitgevoerd	-
P14A	Een gebruiker moet het tijdstip van uitval registratie kunnen achterhalen	-
P15A	Een gebruiker kan in het tijdigheid overzicht zien of de afspraak op de afgesproken dag is uitgevoerd	-

TABEL 3 - PROTOTYPE REQUIREMENTS

2.2.2 WIJZIGINGEN EN BIJZONDERHEDEN

Uit Tabel 3 van hoofdstuk 2.2.1 is af te lezen dat er een aantal requirements impact hebben op de eerder geformuleerde requirements. Het betreft alleen een wijziging of aanvulling van de realisatie van eerder geformuleerde requirements, er zijn geen requirements die in conflict met elkaar zijn. In de volgende hoofdstukken wordt per requirement beschreven wat de aanvulling of wijziging precies inhoud en wat de bijzonderheden van de requirements zijn.

2.2.2.1 REQUIREMENT P10

Deze requirement heeft een minimale impact op requirements K8 en K23. Voor de realisatie van K8 en K23 is er een werkopdracht dimensie gemaakt. De werkopdracht dimensie dient aangevuld te worden met een kolom voor de data van 'aanleiding'. Het ETL-proces voor het synchroniseren van de werkopdracht dimensie dient ook aangevuld te worden met 'aanleiding'. De data voor 'aanleiding' zal nooit wijzigen.

2.2.2.2 REQUIREMENT P11

Ook deze requirement heeft een minimale impact op requirements K8 en K23. De werkopdracht dimensie dient aangevuld te worden met een kolom voor de data van 'metertype'. Het ETL proces moet worden aangepast zodat het de data van 'metertype' meeneemt. De data voor 'metertype' wijzigt nooit.

2.2.2.3 REQUIREMENT P12

P8 heeft een minimale impact op requirements K22 en K24. Voor de realisatie van K22 en K24 is de locatie dimensie aangemaakt. De locatie dimensie dient aangevuld te worden met een kolom voor 'regio'. Het ETL-proces moet worden aangepast met logica voor het bepalen van de regio op basis van interne business regels rondom de postcode en om de regio in te laden in de warehousedatabase. Een postcode wijzigt nooit, en wijzigt de data voor 'regio' ook nooit.

2.2.2.4 REQUIREMENT P1A & P2A

P1A en P2A hebben een grote impact op requirements K14 en K19 en is een aanvulling op requirements P1 en P16. De werknemer dimensie moet worden aangevuld met de kolom 'Aannemer'. De kolom aannemer zorgt er voor dat de werknemer dimensie van slowly changing dimension type 1 naar type 2 moet wijzigen. De data voor 'Aannemer' kan niet worden overschreven, omdat eerder vastgelegde data dan niet meer correct is. Voor het kloppend houden van eerder vastgelegde data moet de historie van 'Aannemer' bewaard blijven. Voor de realisatie van een SCD type 2 dimensie, dient de dimensie aangevuld te worden met een 'Versie' kolom en effectieve datum kolommen.

2.2.2.5 REQUIREMENT P5A

Requirement P5A heeft impact op de requirements K19 en K21. Voor de realisatie van de requirements K19 en K21, is een feitentabel rondom geplaatste assets gemaakt. Hierbij wordt het aantal geplaatste meters bijgehouden. Het aantal geplaatste meters dient opgesplitst worden in het aantal geregistreerde geplaatste meters en het aantal ongeregistreerde geplaatste meters.

2.2.2.6 BIJZONDERHEDEN

Requirement P30 en P31 vallen buiten de scope van het project. Het betreft data uit andere systemen. Alleen data van SmartWork 3.0 valt binnen de scope van het project. Deze requirements worden dus niet meegenomen en kunnen in de toekomst worden opgepakt.

Requirement P32 betreft data van een module binnen SmartWork 3.0 dat momenteel nog in ontwikkeling is. Gezien de module niet in productie staat en het nog niet vast staat wat deze module precies gaat doen, wordt ook deze requirement niet meegenomen. Aan het einde van de aankomende sprint kan deze requirement wederom benaderd worden om te achterhalen of de module dan volledig ontwikkeld en geïmplementeerd is.

2.3 EXTRA REQUIREMENTS

Tabel 4 van hoofdstuk 2.3.1 geeft een overzicht van requirements weer die zijn verkregen aan de hand van het tonen van de producten die gemaakt zijn tijdens sprint 2. De kolom 'Impact' geeft aan of de nieuwe requirement eventueel een aanvulling, wijziging of in conflict is met een requirement of de realisatie van een requirement uit hoofdstuk 2.1 en 2.2. Nieuwe requirements met een impact op eerder geformuleerde requirements worden besproken in hoofdstuk 2.3.2

2.3.1 REQUIREMENTS OVERZICHT

Nr.	Requirement	Impact
E1	Het systeem moet de feiten tabellen onafhankelijk van elkaar updaten	-
E2	Het systeem moet de feiten tabellen alleen updaten indien het updaten van de dimensies succesvol is uitgevoerd	-
E3	Een gebruiker moet per feiten tabel kunnen zien of deze succesvol is geüpdate of niet.	-
E4	Het systeem moet de uitvoering van ETL proces altijd opnieuw proberen indien de SmartWork omgeving tijdelijk niet bereikbaar is	-
E5	Een gebruiker moet per dag een voorspelling van het aantal benodigde meters voor de aankomende twee weken kunnen inzien	-
E6	Een gebruiker moet kunnen achterhalen hoeveel 1-fase meters er naar schatting voor de aankomende twee weken nodig zijn	-
E7	Een gebruiker moet kunnen achterhalen hoeveel 3-fase meters er naar schatting voor de aankomende twee weken nodig zijn	-
E8	Een gebruiker moet kunnen achterhalen hoeveel G4 meters er naar schatting voor de aankomende twee weken nodig zijn	-
E9	Een gebruiker moet kunnen achterhalen hoeveel G6 meters er naar schatting voor de aankomende twee weken nodig zijn	-
E10	Een gebruiker moet kunnen achterhalen hoeveel G10 meters er naar schatting voor de aankomende twee weken nodig zijn	-
E11	Een gebruiker moet kunnen achterhalen hoeveel G16 meters er naar schatting voor de aankomende twee weken nodig zijn	-
E12	Een gebruiker moet kunnen achterhalen hoeveel G25 meters er naar schatting voor de aankomende twee weken nodig zijn	-
E13	Een gebruiker kan de voorspelling van het aantal benodigde meters per regio en per project inzien	-
E14	Het systeem moet database wijzigingen automatisch uitvoeren	-
E1A	Het systeem moet dagelijks de voorspelling van het aantal benodigde meters uitvoeren	-
E2A	Het systeem moet eerder gemaakte voorspelling van het aantal benodigde meters voor de aankomende twee weken dagelijks updaten	-
E3A	Een gebruik kan het land van een locatie achterhalen	K22,K24,P12
E4A	Een gebruiker kan de provincie van een locatie achterhalen	K22,K24,P12
E5A	Het systeem dient de provincie op te halen uit de bestaande postcode database op basis van de postcode	-
E6A	Een gebruiker moet het meter forecast overzicht kunnen exporteren naar PDF formaat	-
E7A	Een gebruiker heeft bij uitgevallen werkopdracht inzicht in het type uitval	P14,P15 P17
E8A	Een gebruiker moet in het inventaris het aantal defecte meters kunnen inzien die een medewerker op zijn naam heeft staan	K13

TABEL 4 - EXTRA REQUIREMENTS

2.3.2 REQUIREMENT WIJZIGINGEN

Uit Tabel 4 van hoofdstuk 2.3 is af te lezen dat een tweetal requirements impact hebben op eerder geformuleerde requirements. Het betreft alleen een aanvulling op eerder opgestelde requirements. In de volgende hoofdstukken wordt weer per requirement beschreven wat de aanvulling precies inhoud.

2.3.2.1 REQUIREMENT E3A

E3A heeft een minimale impact op requirements K22, K24 en P12. De locatie dimensie dient aangevuld te worden met een kolom voor de data van 'land'. Het ETL-proces moet worden aangepast zodat het de data van 'land' kan bepalen op basis van de postcode. De data voor 'land' wijzigen nooit.

2.3.2.2 REQUIREMENT E4A

E4A heeft ook een minimale impact op requirements K22, K24 en P12. Locatie dimensie dient ook aangevuld te worden met een kolom voor de data van 'provincie'. Het ETL-proces moet worden aangepast om de provincie te kunnen achterhalen aan de hand van de postcode. Een postcode wijzigt nooit en dus wijzigen de data voor 'provincie' ook nooit.

2.3.2.3 REQUIREMENT E7A

E7A heeft impact op requirements P15, P15 en P17. Voor de realisatie zal een nieuwe dimensie worden opgesteld. In deze dimensie zullen de kenmerken uitval type en reden komen. Reden wordt uit workorder failure gehaald. Er moet een ETL proces worden gemaakt voor de nieuwe dimensie en het ETL proces voor workorder failure moet worden aangepast.

2.3.2.4 REQUIREMENT E8A

E8A heeft een minimale impact op requirement K13. Asset inventaris zal worden aangevuld met het feit aantal defecte meters. Het ETL proces van asset inventaris moet ook worden aangepast om betreffende feit te berekenen en te vullen.

3 USER STORIES

ID	User story	Req. Nr.
SH1	Als gebruiker wil ik de huidige status van een werkopdracht per dag, per project inzien zodat ik de voortgang van een project kan bijhouden	K8
SH2	Als gebruiker kan ik de aantallen van de 'eind' statussen zien zodat ik inzicht heb in de verhouding tussen de werkopdrachten	K9,K10, K11,K12
SH3	Als gebruiker wil de inventaris van meters per dag, per medewerker kunnen inzien zodat ik weet hoeveel meters een medewerker op een dag op zijn naam heeft gescand	K13,K14 K16
SH4	Als gebruiker wil ik kunnen selecteren over welke periode de inventaris van de meters getoond moet worden zodat ik een wekelijks, maandelijks of jaarlijks overzicht heb	K15
SH5	Als gebruiker wil ik de geplaatste meters per dag, per monteur, per locatie en per werkopdracht kunnen inzien zodat ik een overzicht heb van de locaties en werkopdrachten waar een meter is geplaatst	K19,K20 K21,K22 K23
SH6	Als beheerder kan ik de scheduling van synchronisatie taken beheren zodat ik de taken naar wens van klanten kan configureren	K4
SH7	Als beheerder kan ik de scheduling van de automatisch gegenereerde rapporten beheren zodat ik kan inspelen op de wensen van klanten	K5,K6
SH8	Als beheerder kan ik bepalen naar wie de automatisch gegenereerde rapporten gemaïld worden zodat ik flexibel kan inspelen op de wensen van de klanten	K7
SH9	Als gebruiker wil ik de rapporten binnen het platform kunnen exporteren naar CSV en XLSX formaat zodat ik deze buiten het platform ook kan gebruiken	K25,K26
SH10	Als gebruiker kan ik ad-hoc rapporten maken op basis van bestaande data in de data warehouse zodat ik niet vooraf gedefinieerde vragen kan beantwoorden	K4A,K5A
SH11	Als gebruiker kan ik de filters voor een ad-hoc rapportage opslaan zodat ik deze op een later tijdstip kan hergebruiken	K14A
SH12	Als gebruiker heb ik de mogelijkheid om data te visualiseren met grafieken en diagrammen zodat ik een duidelijk, visueel overzicht heb van mijn data	K28,K29 K30
SH13	Als gebruiker kan ik data visualisaties exporteren in JPEG formaat zodat ik deze buiten het platform kan gebruiken	K7A
SH14	Als gebruiker kan ik de filters en instellingen van visualisaties opslaan zodat ik deze op een later tijdstip kan hergebruiken	K6A
SH15	Als beheerder kan ik systeemgebruikers beheren zodat ik op aanvraag gebruikers en zijn permissies kan toevoegen, verwijderen of wijzigen	K8A,K9A K10A

TABEL 5 - USER STORIES VAN KEY REQUIREMENTS

ID	User story	Req. Nr.
SH16	Als gebruiker kan ik de tijdigheid van monteurs per dag per tijdvak per werkopdracht of per project in zien en of de monteur de afspraak op de afgesproken dag heeft uitgevoerd zodat ik weet welke monteurs ik moet aansturen	P1,P4,P5, P6,P7, P8A, P12A
SH17	Als gebruiker kan ik per dag, per project en per werkopdracht zien hoeveel minuten te vroeg of te laat een monteur bij zijn afspraak is zodat ik weet of een monteur aanhoudend te laat of te vroeg is	P2,P3
SH18	Als gebruikers kan ik het aantal geregistreerde en het aantal ongeregistreerde geplaatste meters achterhalen zodat ik kan bijsturen op het logistiek beheer	P5A
SH19	Als gebruiker wil ik de aanleiding en het metertype van een werkopdracht kunnen achterhalen zodat ik weet wat voor soort werkopdracht het betreft	P10,P11
SH20	Als gebruiker kan ik de regio van een locatie achterhalen zodat ik dit intern eenvoudig kan communiceren	P12
SH21	Als gebruiker wil ik per project, per werknemer en per dag het aantal uitgevallen werkopdrachten, inclusief het tijdstip van registratie, kunnen inzien zodat ik werknemers en klanten kan bijsturen	P14,P16
SH22	Als gebruiker wil ik de reden van uitval en het commentaar van uitval registratie kunnen zien zodat ik weet of het met een geldige reden als uitval is geregistreerd	P15,P17
SH23	Als beheerder heb ik een overzicht van alle taken en wat de status van deze taken is zodat ik gefaalde taken kan herstellen en taken eventueel kan fine tunen	P19,P20
SH24	Als gebruiker wil ik per project, per dag en per monteur kunnen inzien hoeveel correcties op de geregistreerde werkzaamheden zijn plaatsgevonden zodat ik de monteurs kan bijsturen op de volledigheid van hun werkzaamheden	P12,P10A P11A
SH25	Als gebruiker wil ik per project, per dag en per monteur de productiviteit kunnen inzien zodat ik weet welke monteurs wel of niet productief zijn en hierop kan inspelen	P22,P23, P24,P25, P26,P27, P13A
SH26	Als gebruiker kan ik achterhalen bij welke aannemer een monteur momenteel werkzaam is en waar een monteur eventueel werkzaam was zodat ik contact op kan nemen met de aannemers indien dat nodig is	P1A,P2A
SH27	Als gebruiker wil ik de geregistreerde telwerken kunnen inzien en of deze binnen de gedefinieerde marge valt zodat ik eventuele fraude of fouten kan achterhalen	P4A

TABEL 6 - USER STORIES VAN PROTOTYPE REQUIREMENTS

ID	User story	Req. Nr.
SH28	Als beheerder kan ik per feiten tabel zien of de uitvoering van het ETL proces succesvol is verlopen zodat ik tijdig de fouten kan herstellen	E1,E3
SH29	Als gebruiker heb ik van elk type meter inzicht in de meest recente voorspelling van het aantal benodigde meters voor de aankomende twee weken zodat ik weet of er voldoende aantal meters in voorraad zijn	E5,E6,E7, E8,E9, E10, E11, E12, E14
SH30	Als gebruiker kan ik de voorspelling van het aantal benodigde meters per regio en per project achterhalen zodat ik de project en regio managers op de hoogste kan stellen indien er niet voldoende meter in voorraad zijn	E13
SH31	Als gebruiker kan ik de provincie van een locatie achterhalen zodat ik weet waarom een regio niet bepaald kan worden	E4A
SH32	Als gebruiker kan ik het land van een locatie achterhalen zodat ik weet waarom een provincie niet bepaald kan worden	E5A
SH33	Als gebruiker kan ik het overzicht van de meter forecast exporteren naar een PDF bestand zodat ik dit kan opsturen naar de magazijnbeheerders	E6A
SH34	Als gebruiker kan ik het aantal defecte meters zien die op de naam van een monteur staan zodat ik weet hoeveel foutieve meters er in omloop zijn	E8A

TABEL 7 - USER STORIES VAN EXTRA REQUIREMENTS

BIJLAGE D - GEGEVENSMODEL

GEGEVENSMODEL

SMARTINSTALL



INHOUDSOPGAVE

1	Inleiding.....	3
2	SmartWork3.0 Database.....	3
3	TaskDefault	4
4	Location.....	5
5	TaskGroup	6
6	Project	7
7	Company	8
8	User	9
9	Customer.....	10
10	Resource	11
11	Asset.....	12
12	AssetStatus.....	13
13	CommentLog.....	14
14	Workflow.....	15
15	DeviceRegister	16
16	BaseProcessInstance.....	17
17	ProcessState.....	18
18	ProcessVariable.....	19
19	Tags	20
20	CustomField Groep	21
21	CustomFieldValue Groep	22
22	SmartApp Groep	23
23	Periods groep	25
24	Resource Werktijden Groep	26

TABELLENLIJST

Tabel 1 - TaskDefault Beschrijving	4
Tabel 2 - Location Beschrijving	5
Tabel 3 - TaskGroup beschrijving.....	6
Tabel 4 - Project Beschrijving.....	7
Tabel 5 - Company Beschrijving.....	8
Tabel 6 - User beschrijving.....	9
Tabel 7 - Customer Beschrijving	10
Tabel 8 - Resource Beschrijving	11
Tabel 9 - Asset beschrijving.....	12
Tabel 10 - AssetStatus beschrijving	13
Tabel 11 - CommentLog Beschrijving	14
Tabel 12 - Workflow Beschrijving	15
Tabel 13 - DeviceRegister beschrijving	16
Tabel 14 - BaseProcessInstance Beschrijving	17
Tabel 15 - ProcessState Beschrijving	18
Tabel 16 - ProcessVariable beschrijving	19
Tabel 17 - Tags Beschrijving.....	20
Tabel 18 - CustomField Tabellen Beschrijving	21
Tabel 19 - CustomFieldValue Tabellen Beschrijving.....	22
Tabel 20 - SmartAPp CONfigurtie Tabellen Beschrijving.....	24
Tabel 21 - Period Groep tabellen beschrijving	25

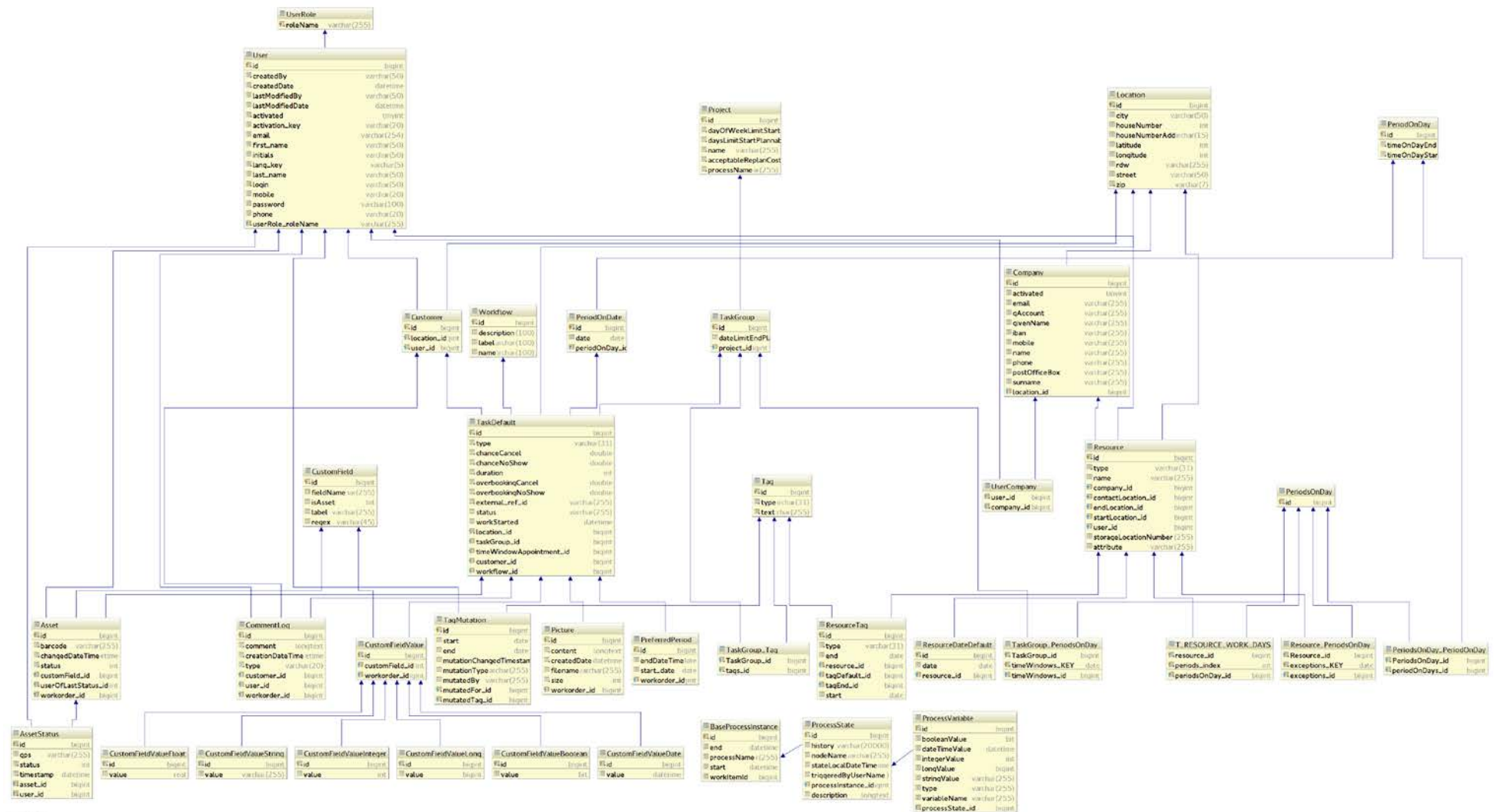
FIGURENLIJST

Figuur 1 - SmartWork3.0 Database	3
Figuur 2 - TaskDefault Tabel	4
Figuur 3 - Location Tabel.....	5
Figuur 4 - TaskGroup Tabel	6
Figuur 5 - Project Tabel	7
Figuur 6 - Company tabel.....	8
Figuur 7 - User Tabel	9
Figuur 8 - Customer Tabel.....	10
Figuur 9 - Resource Tabel	11
Figuur 10 - Asset Tabel.....	12
Figuur 11 - AssetStatus Tabel.....	13
Figuur 12 - CommentLog Tabel.....	14
Figuur 13 - Workflow Tabel.....	15
Figuur 14 - DeviceRegister tabel	16
Figuur 15 - BaseProcessInstance tabel	17
Figuur 16 - ProcessState Tabel.....	18
Figuur 17 - ProcessVariable Tabel.....	19
Figuur 18 - Tags Tabellen	20
Figuur 19 - CustomField Tabellen	21
Figuur 20 - CustomFieldValue Tabellen	22
Figuur 21 - SmartApp Configuratie Tabellen	23
Figuur 22 - Periods Groep Tabellen	25
Figuur 23 - Resource Werktijden Tabellen	26

1 INLEIDING

Dit document is opgesteld in het kader van het in kaart brengen van de sourcedatabase. Van alle relevante tabellen is een model gegenereerd. Per tabel van het opgestelde model wordt beschreven wat de functie van de tabel is en wat de kenmerken van betreffende tabel inhouden. Tot slot worden de tabellen besproken die alleen een betekenis hebben indien ze gegroepeerd zijn met andere samenhangende tabellen.

2 SMARTWORK3.0 DATABASE



FIGUUR 1 - SMARTWORK3.0 DATABASE

Figuur 1 weergeeft een diagram van de SmartWork 3.0 database. Het diagram is gegenereerd met behulp van IntelliJ IDEA. Er zijn een aantal tabellen van de database uit het model weggelaten die niet relevant zijn voor de warehouse. De volgende tabellen en onderdelen zijn uit het diagram weggelaten:

- Spring Batch
- Liquibase logs
- File import Items
- File Export Items
- Persistent Audit Events
- Quality Inspection Triggers
- SmartWork Parameters

Spring Batch wordt gebruikt in SmartWork 3.0 om bestanden te importeren die aangeleverd worden door netbeheerders. Deze bestanden betreffen CS- bestanden met daarin de uit te voeren werkopdrachten, klantinformatie en meterstanden. Om Spring Batch te realiseren zijn er een tiental batchtabellen en een aantal File Import tabellen gemaakt. Deze tabellen zijn bedoeld voor de import logica en zijn niet relevant voor de warehouse.

Voor het maken, wijzigen en onderhouden van de database wordt er gebruik gemaakt van Liquibase. Tevens zorgt Liquibase ervoor dat de database onder versiebeheer staat. Liquibase maakt gebruik van database changelogs en changesets. Om bij te kunnen houden welke changelogs uitgevoerd zijn en welke changesets wel of niet succesvol zijn, zijn er een drietal tabellen in de database aangemaakt. Deze tabellen zijn ter ondersteuning van Liquibase en zijn niet relevant voor de warehouse.

De Persistent Audit Events tabellen zijn er om auditing in SmartWork 3.0 mogelijk te maken. Van alle wijziging die een gebruiker uitvoert wordt bijgehouden door wie, waar, wat en wanneer deze wijziging plaats heeft gevonden. Alle wijzigingen worden bijgehouden om meterstandfraude en dergelijke te voorkomen. Alle audittabellen zijn verder niet relevant voor de warehouse.

Quality Inspection Triggers hebben als doel om inspectieopdrachten in SmartQuality aan te maken. Zodra een opdracht voltooid is, wordt deze als inspectie aangemaakt en naar SmartQuality verstuurd. Dit onderdeel is momenteel nog in ontwikkeling en niet relevant voor de warehouse.

Ondanks dat CustomField types wel relevant zijn voor de warehouse, zijn de bijbehorende tabellen toch weggelaten. Ze zijn weggelaten omdat ze niet veel informatie geven. CustomFields kunnen uit zes verschillende datatypes bestaan. Met de huidige implementatie houdt dat in dat er zes database tabellen bestaan die aangeven welk datatype een CustomField is. CustomFieldValue heeft dezelfde implementatie, echter, deze zijn niet weggelaten. In tegenstelling tot CustomField heeft CustomFieldValue een kolom genaamd 'value'. In deze kolom worden de data van een CustomField opgeslagen. Een CustomField tabel bevat alleen een ID, die is overgeërfd van de CustomField tabel.

3 TASKDEFAULT

TaskDefault	
 PK	id : bigint
 FK	location_id : bigint
 FK	taskGroup_id : bigint
 FK	timeWindowAppointment_id : bigint
 FK	customer_id : bigint
 FK	workflow_id : bigint
	type : varchar (31)
	external_ref_id : varchar (255)
	status : varchar (255)
	duration : integer
	changeCancel : double
	changeNoShow : double
	overbookingCancel : double
	overbookingNoShow : double

De TaskDefault, ook wel de workorder of werkopdracht, is een door een monteur uit te voeren opdracht. Een werkopdracht betreft onder andere de wisseling van de meters op een locatie. De locatie is in dit geval waar de opdracht uitgevoerd moet worden. De customer is de 'klant' die op dit moment in bezit is van de betreffende locatie. Het kan zijn dat een klant woonachtig is op een andere locatie. De workflow is de werkstroom die monteurs volgen tijdens de wisseling met behulp van de Android app, SmartApp. TimeWindowAppointment is het tijdsblok waarin de monteur de opdracht zal uitvoeren. Het tijdsblok kan zowel een vastgelegde gecommuniceerde afspraak zijn of een tijdelijke afspraak die nog kan wijzigen.

FIGUUR 2 - TASKDEFAULT TABEL

Kolom	Beschrijving
location_id	Vreemde sleutel, referentie naar Location tabel (<i>Hoofdstuk 4, Figuur 3</i>)
taskGroup_id	Vreemde sleutel, referentie naar TaskGroup tabel (<i>Hoofdstuk 5, Figuur 4</i>)
timeWindowAppointment_id	Vreemde sleutel, referentie naar PeriodOnDate tabel
customer_id	Vreemde sleutel, referentie naar Customer tabel (<i>Hoofdstuk 9, Figuur 8</i>)
workflow_id	Vreemde sleutel, referentie naar Workflow tabel
type	Geeft aan wat voor type werkopdracht het betreft. Momenteel niet in gebruik en standaard gevuld met de waarde 'workorder'
external_ref_id	De externe referentie. Dit is de referentie, of 'ID', zoals die in de systemen van de netbeheerder bekend is. Moet een unieke waarde zijn.
status	De status van de werkopdracht. Geeft aan waar de werkopdracht zich in het proces bevindt
duration	De installatie tijd in minuten. Dit is het aantal minuten die een monteur krijgt om de werkopdracht uit te voeren
changeCancel	Constraint voor de OptaPlanner logica. Niet relevant voor de warehouse
changeNoShow	Constraint voor de OptaPlanner logica. Niet relevant voor de warehouse
overbookingCancel	Constraint voor de OptaPlanner logica. Niet relevant voor de warehouse
overbookingNoShow	Constraint voor de OptaPlanner logica. Niet relevant voor de warehouse

TABEL 1 - TASKDEFAULT BESCHRIJVING

4 LOCATION

Location	
 PK	id : bigint
	city : varchar (5)
	houseNumber : integer
	houseNumberAdd : varchar (15)
	street : varchar (50)
	zip : varchar (7)
	latitude : integer
	longitude : integer
	rdw : varchar (255)

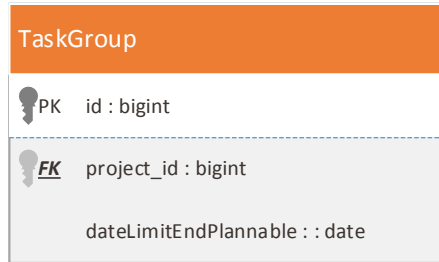
Location is de locatie waar de opdracht uitgevoerd dient te worden of het betreft de woonplaats van een customer. Latitude en longitude zijn integers, die, wanneer door 10.000.000 gedeeld, de positie op aarde representeren.

FIGUUR 3 - LOCATION TABEL

Kolom	Beschrijving
city	De stad van de locatie
houseNumber	Het huisnummer van de locatie
houseNumberAdd	De toevoeging van het huisnummer van de locatie
street	De straat van de locatie
zip	De postcode van de locatie
latitude	WGS84 latitude. Noord-zuid positie van een punt op aarde
longitude	WGS84 longitude. Oost-west positie van een punt op aarde
rdw	Rijksdriehoekcoördinaten. Tot op heden nog ongebruikt

TABEL 2 - LOCATION BESCHRIJVING

5 TASKGROUP



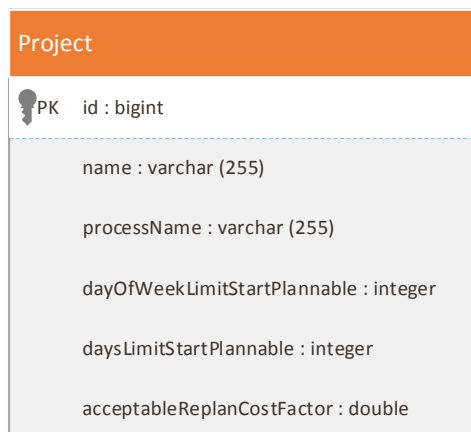
TaskGroup groepeer de werkopdrachten op basis van tijdsperiode en tags. Werkopdrachten van een TaskGroup kunnen aan verschillende projecten gekoppeld zijn.

FIGUUR 4 - TASKGROUP TABEL

Kolom	Beschrijving
project_id	Vreemde sleutel, referentie naar Project tabel (<i>Hoofstuk 6, Figuur 5</i>)
dateLimitEndPlannable	Een datum parameter voor de OptaPlanner logica. Dit bepaald vanaf wanneer er geen automatische planning meer gemaakt moet worden voor werkopdrachten uit de TaskGroup. Niet relevant voor de warehouse.

TABEL 3 - TASKGROUP BESCHRIJVING

6 PROJECT



Project is een uitvoeringsproject gedefinieerd door de netbeheerders. Een project bevat meerdere TaskGroups en dus meerdere werkopdrachten. Een project wijzigt nooit. Een werkopdracht kan echter wel van project wijzigen om diverse redenen.

FIGUUR 5 - PROJECT TABEL

Kolom	Beschrijving
name	Naam van het project, gedefinieerd door de netbeheerder
processName	Naam het SmartWork3.0 proces. Een project volgt 1 van de processen die gedefinieerd zijn in SmartWork3.0. Een proces is ook wel de werkstroom die een werkopdracht van een project kan volgen.
dayOfWeekLimitStartPlannable	De dag van de week waarop brieven met afspraak data worden aangemaakt en verzonden van vastgelegde afspraken.
daysLimitStartPlannable	Het aantal dagen van te voren vanaf vandaag dat de afspraken van het project vastgelegd worden.
acceptableReplanCostFactor	De factor die bepaald hoeveel 'schade' een afspraak herplan actie mag verrichten op de geoptimaliseerde planning.

TABEL 4 - PROJECT BESCHRIJVING

7 COMPANY

Company	
 PK	id : bigint
 FK	location_id : bigint
	email : varchar (255)
	gAccount : varchar (255)
	givenName : varchar (255)
	iban : varchar (255)
	mobile : varchar (255)
	name : varchar (255)
	phone : varchar (255)
	postOfficeBox : varchar (255)
	surname : varchar (255)

Een company is een installatiebedrijf waar monteurs voor werken. SmartInstall huurt monteurs in van deze installatiebedrijven voor de uitvoering van de wissels. Location van de company is de locatie waar het hoofdkantoor van het installatiebedrijf is gevestigd. Dit onderdeel wordt in de aankomende fase, fase 3, van SmartWork3.0 volledig herzien.

FIGUUR 6 - COMPANY TABEL

Kolom	Beschrijving
location_id	Vreemde sleutel, referentie naar Location tabel (<i>Hoofdstuk 4, Figuur 3</i>)
email	Email adres van de contact persoon. Meest de directeur
gAccount	De g-rekening, geblokkeerde bankrekening voor belasting. Tot op heden ongebruikt
givenname	De voornaam naam van de contact person
iban	Het iban rekening nummer van het installatiebedrijf
mobile	Het mobiele nummer van de contact persoon
name	De naam van het installatiebedrijf
phone	Het (hoofd) telefoonnummer van het kantoor van het installatiebedrijf
postOfficeBox	Postbus van het installatiebedrijf
surname	De achternaam van de contact persoon

TABEL 5 - COMPANY BESCHRIJVING

8 USER

User	
	id : bigint
	first_name : varchar (50)
	last_name : varchar (50)
	initials : varchar (50)
	email : varchar (254)
	mobile : varchar (20)
	phone : varchar (20)
	login : varchar (50)
	password : varchar (100)
	lang_key : varchar (5)
	activated : tinyint
	createdBy : varchar (50)
	createdDate : datetime
	lastModifiedBy : varchar (50)
	lastModifiedDate : datetime

Een User is een gebruiker die gebruik maakt van het SmartWork platform. Een gebruiker kan een systeemgebruiker, klant of monteur zijn. Systeemgebruikers zijn front of back office van netbeheerders die een gedeelte van SmartWork3.0 mogen gebruiken. Systeemgebruikers kunnen ook interne administratie of operations medewerkers zijn die toegang hebben tot het volledige systeem. Klanten kunnen alleen gebruik maken van het klantportaal om hun afspraak te herplannen en eventueel hun gegevens te wijzigen. Monteurs maken alleen gebruik van SmartApp en loggen zich via de SmartApp in, in het SmartWork systeem.

FIGUUR 7 - USER TABEL

Kolom	Beschrijving
first_name	Voornaam van de gebruiker
last_name	Achternaam van de gebruiker
initials	Initialen van de gebruiker
email	Email adres van de gebruiker
mobile	Mobiele nummer van de gebruiker
phone	Vaste telefoonnummer van de gebruiker
login	Gebruikersnaam waar gebruikers zich mee inloggen
password	Het encrypted wachtwoord van de gebruiker
lang_key	Language key, de voorkeurstaal van de gebruiker volgens ISO 639-1
activated	Geeft aan of een gebruiker actief of non actief is. Indien non actief, kan de gebruiker geen gebruik meer maken van het systeem. Een gebruiker wordt op non actief gezet bij ontslag of dergelijke.
createdBy	De login van de gebruiker die deze gebruiker heeft aangemaakt. Semi referentie naar zichzelf
createdDate	De datum waarop deze gebruiker aangemaakt is
lastModifiedBy	De login van de gebruiker die deze gebruiker heeft gewijzigd. Semi referentie naar zichzelf
lastModifiedDate	De datum waarop deze gebruiker gewijzigd is

TABEL 6 - USER BESCHRIJVING

9 CUSTOMER

Customer	
 PK	id : bigint
	location_id : bigint
	user_id : bigint

Een Customer is een klant voor wie een monteur de werkzaamheden zal uitvoeren. De locatie van de customer is waar de klant momenteel woonachtig is. Dit kan afwijken van de locatie waar de werkzaamheden worden uitgevoerd. Een Customer bevat een User zodat een klant gebruik kan maken van de klantportaal.

FIGUUR 8 - CUSTOMER TABEL

Kolom	Beschrijving
location_id	Vreemde sleutel, referentie naar Location tabel (<i>Hoofdstuk 4, Figuur 3</i>)
user_id	Vreemde sleutel, referentie naar User tabel (<i>Hoofdstuk 8, Figuur 7</i>)

TABEL 7 - CUSTOMER BESCHRIJVING

10 RESOURCE

Resource	
 PK	id : bigint
 FK	company_id : bigint
 FK	user_id : bigint
 FK	contactLocation_id : bigint
 FK	endLocation_id : bigint
 FK	startLocation_id : bigint
	type : varchar (31)
	storageLocationNumber : varchar (255)
	attribute : varchar (255)

Een Resource is een monteur die onder andere de meter wissels uitvoert. Een monteur is gekoppeld aan een Company, het bedrijf waar de monteur werkzaam is. Contact locatie is de huidige woonplaats van de monteur. Start en eind locatie is de voorkeur van een monteur waar hij zijn dag/route wilt beginnen en eindigen. In de planning wordt hier rekening mee gehouden, maar kan overruled worden.

FIGUUR 9 - RESOURCE TABEL

Kolom	Beschrijving
company_id	Vreemde sleutel, referentie naar Company tabel (<i>Hoofdstuk 7, Figuur 6</i>)
user_id	Vreemde sleutel, referentie naar User tabel (<i>Hoofdstuk 8, Figuur 7</i>)
contactLocation_id	Vreemde sleutel, referentie naar Location tabel (<i>Hoofdstuk 4, Figuur 3</i>)
endLocation_id	Vreemde sleutel, referentie naar Location tabel (<i>Hoofdstuk 4, Figuur 3</i>)
startLocation_id	Vreemde sleutel, referentie naar Location tabel (<i>Hoofdstuk 4, Figuur 3</i>)
type	Tot op heden nog ongebruikt
storageLocationNumber	Voorraad locatie nummer, de locatie waar monteurs hun meters uit de voorraad halen om te plaatsen
attribute	Kenmerk van de monteur. Ook wel de functie die de monteur heeft binnen het installatiebedrijf of de afdeling.

TABEL 8 - RESOURCE BESCHRIJVING

11 ASSET

Asset	
 PK	id : bigint
 FK	customField_id : bigint
 FK	userOfLastStatus_id : bigint
 FK	workorder_id : bigint
	barcode : varchar (255)
	status : int
	changedDateTime : datetime

Een Asset is een elektra, gas of water meter. Dit betreft alleen de nieuwe (nog te plaatsen of geplaatste) meters. Deze worden geregistreerd met de SmartApp ten behoeve van het logistiek beheer. Een Asset is gekoppeld aan een CustomField wat aangeeft om wat voor type meter het betreft. UserOfLastStatus is de gebruiker die als laatst iets met de Asset gedaan heeft. De werkopdracht wordt gekoppeld aan de Asset op moment van plaatsing.

FIGUUR 10 - ASSET TABEL

Kolom	Beschrijving
customField_id	Vreemde sleutel, referentie naar CustomField tabel (<i>Hoofdstuk 20, Figuur 19</i>)
userOfLastStatus_id	Vreemde sleutel, referentie naar User tabel (<i>Hoofdstuk 8, Figuur 7</i>)
workorder_id	Vreemde sleutel, referentie naar TaskDefault tabel (<i>Hoofdstuk 3, Figuur 2</i>)
barcode	De barcode van de Asset
status	De huidige status code van de Asset. De status code betreft de code van in voorraad, defect, retour, vermist, gestolen of geplaatst
changedDateTime	De timestamp waarop de huidige status van de Asset is geregistreerd

TABEL 9 - ASSET BESCHRIJVING

12 ASSETSTATUS

AssetStatus	
 PK	id : bigint
 FK	asset_id : bigint
 FK	user_id : bigint
	gps : varchar (255)
	status : int
	timestamp : datetime

AssetStatus betreft de geschiedenis log van een Asset. Voor elke nieuwe status wordt er een nieuwe AssetStatus geregistreerd. De gekoppelde User betreft de gebruiker die deze status heeft geregistreerd.

FIGUUR 11 - ASSETSTATUS TABEL

Kolom	Beschrijving
asset_id	Vreemde sleutel, referentie naar Asset tabel (<i>Hoofdstuk 11, Figuur 11</i>)
user_id	Vreemde sleutel, referentie naar User tabel (<i>Hoofdstuk 8, Figuur 7</i>)
gps	De gps coördinaten van het moment van registreren.
status	De toegewezen status van het moment van de registratie
timestamp	De timestamp van de toewijzing van de status

TABEL 10 - ASSETSTATUS BESCHRIJVING

13 COMMENTLOG

CommentLog	
 PK	id : bigint
 FK	user_id : bigint
 FK	customer_id : bigint
 FK	workorder_id : bigint
	comment : longtext
	creationDateTime : datetime
	type : varchar (20)

Een CommentLog betreft een klant contact registratie of een monteur instructie. Indien er contact is geweest met de klant, worden het contact en de eventueel gemaakte afspraken geregistreerd. Een monteur instructie kan worden gemaakt door de front- of backoffice om monteur bepaalde instructies voor een werkopdracht te geven. Deze instructies ziet de monteur terug in de SmartApp. Een CommentLog is gekoppeld aan een klant of een werkopdracht, nooit aan beide. De gebruiker die de CommentLog heeft aangemaakt wordt ook geregistreerd.

FIGUUR 12 - COMMENTLOG TABEL

Kolom	Beschrijving
user_id	Vreemde sleutel, referentie naar User tabel (<i>Hoofdstuk 8, Figuur 7</i>)
customer_id	Vreemde sleutel, referentie naar Customer tabel (<i>Hoofdstuk 9, Figuur 8</i>)
workorder_id	Vreemde sleutel, referentie naar TaskDefault tabel (<i>Hoofdstuk 3, Figuur 2</i>)
comment	Het commentaar van contact moment of de monteur instructie
creationDateTime	De timestamp van commentaar registratie
type	Het type commentaar, of contact moment of monteur instructie

TABEL 11 - COMMENTLOG BESCHRIJVING

14 WORKFLOW

Workflow	
 PK	id : bigint
	description : varchar (100)
	label : varchar (100)
	name : varchar (100)

Een Workflow is een werkstroom die een monteur volgt tijdens zijn werkzaamheden. De werkstroom volgt hij met behulp van SmartApp. Een werkstroom bestaat uit schermen in een bepaalde volgorde die aangeven wat een monteur dient te doen en wat hij moet registreren.

FIGUUR 13 - WORKFLOW TABEL

Kolom	Beschrijving
description	Beschrijving van de werkstroom. Vaak een uitleg van de werkzaamheden
label	De label van de workflow. Dit is wat de monteur in SmartApp te zien krijgt
name	De naam van een workflow. Wordt gebruikt voor configuratie in de code

TABEL 12 - WORKFLOW BESCHRIJVING

15 DEVICEREGISTER

DeviceRegister	
 PK	id : bigint
 FK	user_id : bigint
	androidVersion : varchar (255)
	device : varchar (255)
	gps : varchar (255)
	handscanner : varchar (255)
	imeiNumber : varchar (255)
	longSimNumber : varchar (255)
	modelName : varchar (255)
	product : varchar (255)
	sdkversion : varchar (255)
	serialNumber : varchar (255)
	shortSimNumber : varchar (255)
	smartAppVersion : varchar (255)
	date : varchar (255)
	time : varchar (255)

DeviceRegister is de registratie van het apparaat waar de monteur zich mee inlogt. Op het moment van inloggen, wordt er informatie over het apparaat, de Android versie en de SmartApp versie geregistreerd. Indien een monteur na inloggen geen internet verbinding meer heeft en niet traceerbaar is, wordt dit gebruikt om te zien of een monteur wel is op komen dagen.

FIGUUR 14 - DEVICEREGISTER TABEL

Kolom	Beschrijving
user_id	Vreemde sleutel, referentie naar User tabel (<i>Hoofdstuk 8, Figuur 7</i>)
androidVersion	De android versie die op het apparaat geïnstalleerd staat.
device	Het daadwerkelijke apparaat, bijvoorbeeld galaxy tab
gps	De gps coördinaten van het moment van inloggen
handscanner	Het nummer van de gekoppelde handscanner die monteur gebruiken om de barcodes van de meters te scannen
imeiNumber	Het international mobile station equipment identity number van het apparaat
longSimNumber	Het serie nummer van de gebruikte simkaart
modelName	Het model nummer van het apparaat
product	De fabrikant, bijvoorbeeld Samsung
sdkversion	Tot op heden ongebruikt
serialNumber	Het serie nummer van het apparaat
shortSimNumber	Tot op heden ongebruikt
smartAppVersion	De versie van de geïnstalleerde SmartApp
date	De datum van het moment van inloggen
time	Het tijdstip van het moment van inloggen

TABEL 13 - DEVICEREGISTER BESCHRIJVING

16 BASEPROCESSINSTANCE

BaseProcessInstance	
 PK	id : bigint
	workItemId : bigint
	processName : varchar (255)
	start : datetime
	end : datetime

Een BaseProcessInstance is een instantie van het proces waar een werkopdracht in zit. Een proces is ook wel de werkstroom van SmartWork3.0. BaseProcessInstance heeft in principe een koppeling met een werkopdracht, maar dit is niet fysiek geïmplementeerd.

FIGUUR 15 - BASEPROCESSINSTANCE TABEL

Kolom	Beschrijving
workItemId	Het ID van desbetreffende werkopdracht
processName	De naam van het proces waar de werkopdracht zich in bevind
start	De timestamp waarop de werkopdracht begint in het proces. Ook wel de timestamp waarop de werkopdracht in SmartWork3.0 is ingeladen.
end	De timestamp waarop de werkopdracht eindigt in het proces. Ook wel de timestamp waarop de werkopdracht is uitgevoerd, opgeleverd en verwerkt in het systeem van de netbeheerder.

TABEL 14 - BASEPROCESSINSTANCE BESCHRIJVING

17 PROCESSSTATE

ProcessState	
 PK	id : bigint
 FK	processInstance_id : bigint
	history : varchar (20000)
	nodeName : varchar (255)
	description : longtext
	stateLocalDateTime : datetime
	triggeredByUserName : varchar (255)

Een ProcessState is ook wel een status die toegewezen wordt aan een werkopdracht. Welke status een werkopdracht kan hebben, staat niet van te voren vast en wordt in SmartWork3.0 geconfigureerd middels het definiëren van de zogenoemde werkstroom.

FIGUUR 16 - PROCESSSTATE TABEL

Kolom	Beschrijving
processInstance_id	Vreemde sleutel, referentie naar BaseProcessInstance tabel (<i>Hoofdstuk 16, Figuur 15</i>)
history	Tot op heden ongebruikt
nodeName	De naam van de state waar de werkopdracht zich bevind, ook wel de status
description	Een beschrijving van de reden waarom een werkopdracht deze state toegewezen heeft gekregen of wat er is gebeurd met een werkopdracht terwijl die deze state had.
stateLocalDateTime	De timestamp waarop de werkopdracht de state toegewezen krijgt
triggeredByUserName	De login van de User die deze state toegewezen heeft. Waarde 'System' indien het door een automatisch proces is gebeurd.

TABEL 15 - PROCESSSTATE BESCHRIJVING

18 PROCESSVARIABLE

ProcessVariable	
 PK	id : bigint
 FK	processState_id : bigint
	variableName : varchar (255)
	type : varchar (255)
	stringValue : varchar (255)
	integerValue : integer
	longValue : bigint
	dateTimeValue : datetime
	booleanValue : bit

Een ProcessVariable is een dynamische variabele die aan een status van het proces hangt. Een status kan meerdere variabelen bevatten. Medewerkers van operations kunnen middels de ProcessVariables achterhalen wat er met een werkopdracht gebeurd is.

FIGUUR 17 - PROCESSVARIABLE TABEL

Kolom	Beschrijving
processState_id	Vreemde sleutel, referentie naar ProcessState tabel (<i>Hoofdstuk 17, Figuur 16</i>)
variableName	De naam van de variabele
type	Het type van de variabelen. Dit betreft een string, integer, long, datetime of boolean
stringValue	De value van de variabele indien het type string is, anders null
integerValue	De value van de variabele indien het type integer is, anders null
longValue	De value van de variabele indien het type long is, anders null
dateTimeValue	De value van de variabele indien het type datetime is, anders null
booleanValue	De value van de variabele indien het type boolean is, anders null

TABEL 16 - PROCESSVARIABLE BESCHRIJVING

19 TAGS

Tag	
 PK	id : bigint
	type : varchar (31)
	text : varchar (255)

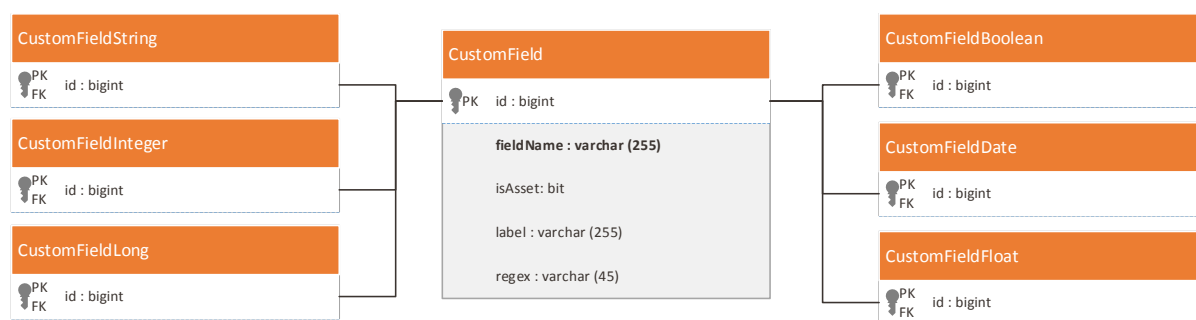
FIGUUR 18 - TAGS TABELLEN

Een tag wordt binnen SmartWork op twee manieren gebruikt. Het kan dienen als een label / kenmerk voor werkopdrachten en monteurs of het dient als het toekennen van vaardigheden (certificering) aan monteurs. Een monteur dient alle vaardigheden en kenmerken te hebben die een werkopdracht heeft om ingepland te mogen worden voor desbetreffende werkopdracht

Kolom	Beschrijving
type	De type van de tag. Dit betreft een label of een monteursvaardigheid
tekst	De daadwerkelijke tag

TABEL 17 - TAGS BESCHRIJVING

20 CUSTOMFIELD GROEP



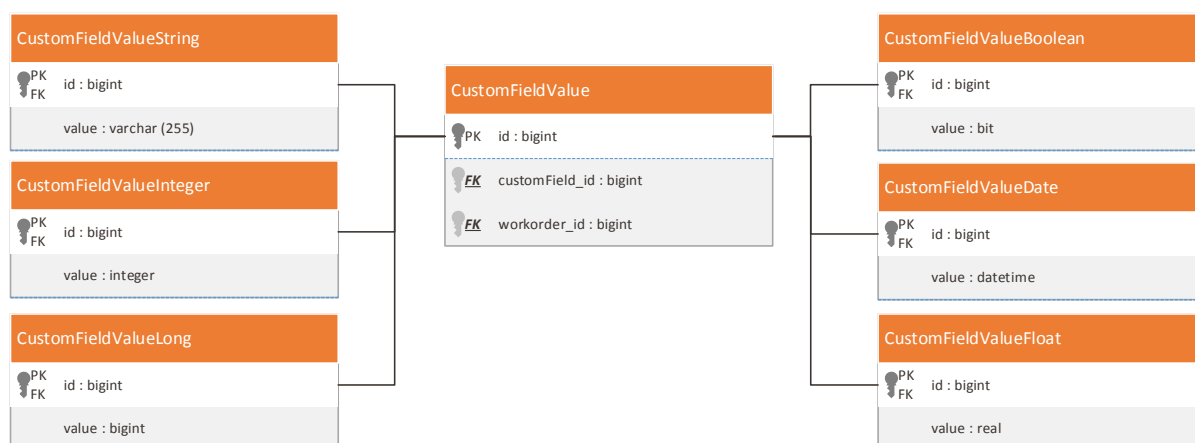
FIGUUR 19 - CUSTOMFIELD TABELLEN

CustomFields zijn dynamische velden die aangemaakt kunnen worden in SmartWork3.0. Een CustomField betreft 1 van de 6 mogelijke data types. CustomFields worden gebruikt om Workflows te realiseren. Workflows hebben ieder een eigen set aan velden die een monteur invult tijdens zijn werkzaamheden. Indien een CustomField aan een workflow gekoppeld is, dient dit veld een waarde te hebben bij het inladen van de werkopdracht. Als een CustomField gekoppeld is aan een ScreenElement of Attribute, wordt deze pas ingevuld tijdens de werkzaamheden van een monteur (Zie hoofdstuk 22)

Tabel	Kolom	Beschrijving
CustomField	fieldName	De naam van een customfield
CustomField	isAsset	Geeft aan of een customfield een asset is
CustomField	label	De label die wordt getoond in SmartApp.
CustomField	regex	De regex waar de waarde van CustomFieldValue aan moet voldoen

TABEL 18 - CUSTOMFIELD TABELLEN BESCHRIJVING

21 CUSTOMFIELDVALUE GROEP



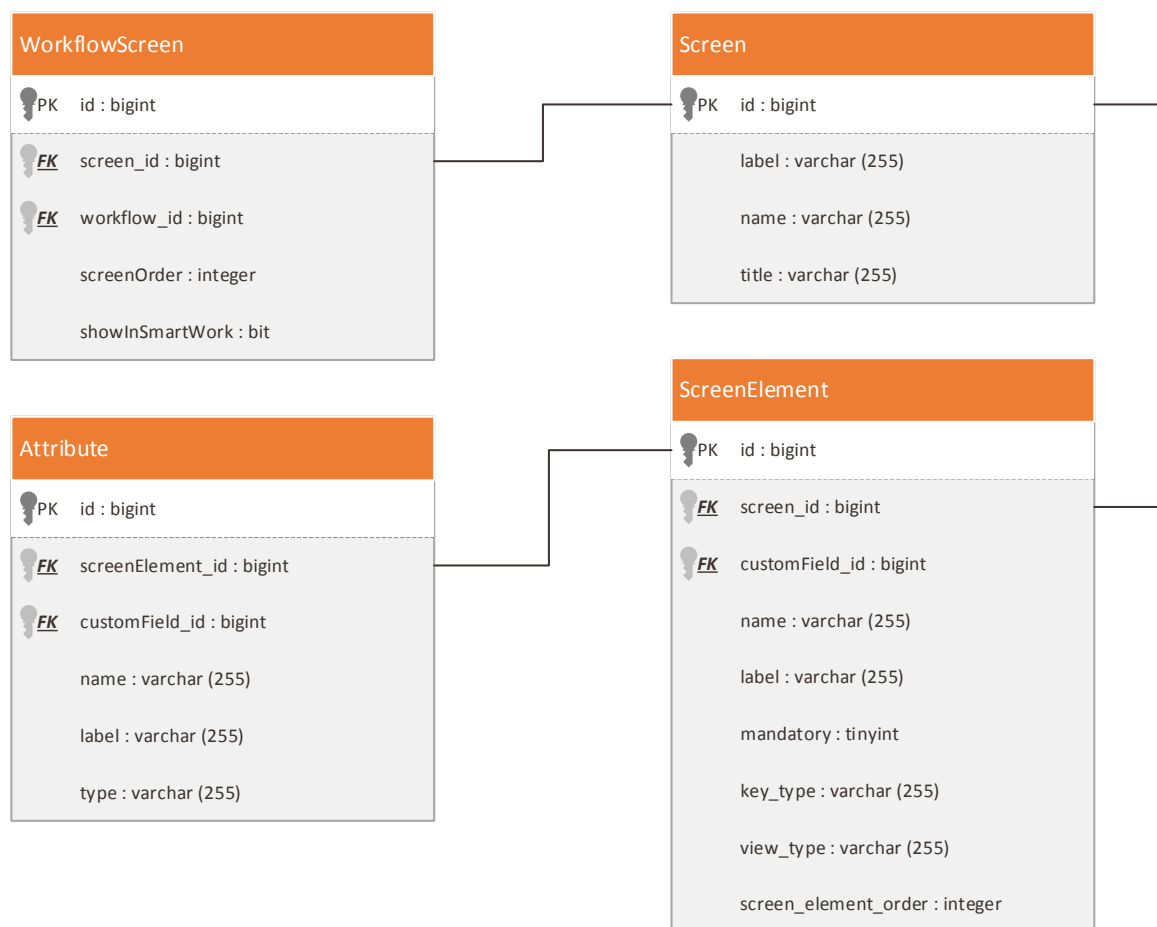
FIGUUR 20 - CUSTOMFIELDVALUE TABELLEN

CustomFieldValues bevatten de waardes van de CustomFields. Net zoals bij CustomFields, bestaat een waarde uit 1 van de 6 data types. Een CustomFieldValue wijzigt nooit van werkopdracht of CustomField, al zou dit in theorie wel kunnen.

Tabel	Kolom	Beschrijving
CustomFieldValue	customField_id	Vreemde sleutel, referentie naar CustomField tabel (Hoofdstuk 20, Figuur 19)
CustomFieldValue	workorder_id	Vreemde sleutel, referentie naar TaskDefault tabel (Hoofdstuk 3, Figuur 4)

TABEL 19 - CUSTOMFIELDVALUE TABELLEN BESCHRIJVING

22 SMARTAPP GROEP



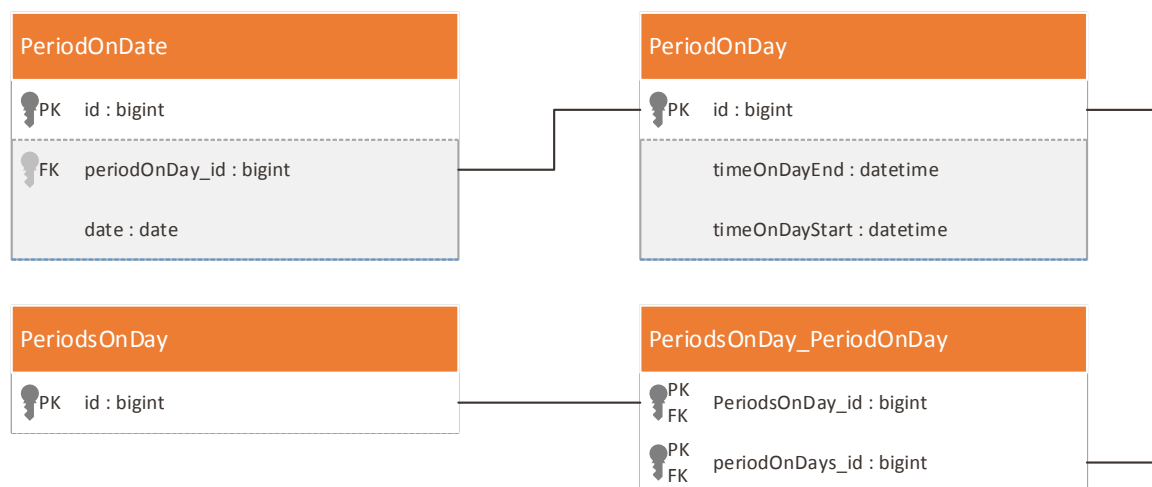
FIGUUR 21 - SMARTAPP CONFIGURATIE TABELLEN

De tabellen van de SmartApp groep zijn de opgeslagen configuratie van de werkstroom voor de SmartApp. De werkstroom van de SmartApp wordt dynamisch opgebouwd. Een werkstroom bestaat uit meerdere schermen. Deze schermen doorloopt een monteur 1 voor 1 per werkopdracht. In de schermen staan stap voor stap wat een monteur tijdens zijn werkzaamheden dient te doen (controleren metergegevens, foto's maken van oude en nieuwe meters e.d.) Een scherm is opgebouwd uit meerdere elementen. Een element betreft een label, tekstveld, checkbox, button, radio button en dergelijke. Aan het element wordt een CustomField gekoppeld waarbij de waarde moet worden ingevuld door de monteur of waarbij de waarde getoond moet worden in de app. Aan een element kunnen 1 of meerdere attributen worden gekoppeld. Een attribuut betreft een minimale of maximale waarde controle die een monteur dient in te vullen, een regex waar de invulde waarde aan moet volden, een lijst van items die getoond moeten worden in de dropdown box en dergelijke.

Tabel	Kolom	Beschrijving
workflowScreen	screen_id	Vreemde sleutel, referentie naar Screen tabel
workflowScreen	workflow_id	Vreemde sleutel, referentie naar Workflow tabel (<i>Hoofdstuk 14, Figuur 13</i>)
workflowScreen	screenOrder	De volgorde van de schermen. Order geeft aan op welke plaats het scherm getoond dient te worden.
workflowScreen	showInSmartWork	Geeft aan of de waardes van het scherm in de front-end van SmartWork3.0 getoond moet worden of niet
Screen	label	Tot op heden nooit gebruikt
Screen	name	De naam van het scherm
Screen	title	De titel van het scherm. De titel wordt getoond in SmartApp
ScreenElement	screen_id	Vreemde sleutel, referentie naar Screen tabel
ScreenElement	customField_id	Vreemde sleutel, referentie naar CustomField tabel (<i>Hoofdstuk 20, Figuur 19</i>)
ScreenElement	name	De naam van de screen element
ScreenElement	label	Tot op heden nooit gebruikt
ScreenElement	mandatory	Geeft aan of een monteur het element verplicht moet invullen
ScreenElement	key_type	Geeft aan welk Android toetsenboard getoond moet worden bij het invullen van de waarde
ScreenElement	view_type	Geeft aan wat voor element er getoond moet worden in SmartApp
ScreenElement	screen_element_order	De volgorde van de elementen. Order geeft aan op welke plaats het element in het scherm getoond wordt.
Attribute	screenElement_id	Vreemde sleutel, referentie naar ScreenElement tabel
Attribute	customField_id	Vreemde sleutel, referentie naar CustomField tabel (<i>Hoofdstuk 20, Figuur 19</i>)
Attribute	name	De naam van het attribuut
Attribute	label	Tot op heden nooit gebruikt
Attribute	type	Geeft aan wat voor type attribuut het betreft

TABEL 20 - SMARTAPP CONFIGURATIE TABELLEN BESCHRIJVING

23 PERIODS GROEP



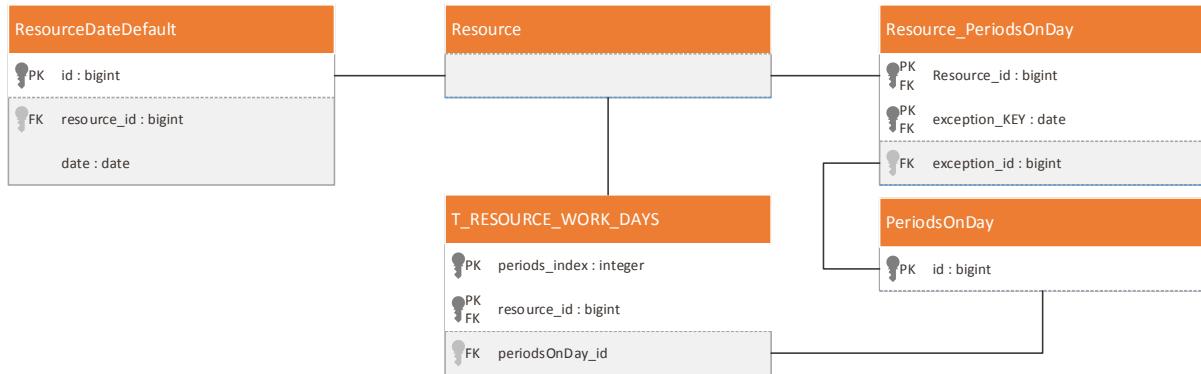
FIGUUR 22 - PERIODS GROEP TABELLEN

De combinatie van PeriodOnDate en PeriodOnDay geeft een tijdsblok op een bepaalde dag aan. Indien er voor een object een dag meer dan 1 tijdsblok moet bevatten, wordt dit afgehandeld met de PeriodsOnDay tabel en de tussentabel PeriodsOnDay_PeriodsOnDay. De datum wordt dan in het object zelf of in een semi tussen tabel opgeslagen met een vreemde sleutel naar PeriodsOnDay.

Tabel	Kolom	Beschrijving
PeriodOnDate	periodOnDay_id	Vreemde sleutel, referentie naar PeriodOnDay tabel
PeriodOnDate	date	De datum waarop het gekoppelde tijdsblok betreft
PeriodOnDay	timeOnDayEnd	Het tijdstip van het einde van een tijdsblok
PeriodOnDay	timeOnDayStart	Het tijdstip van het einde van een tijdsblok

TABEL 21 - PERIOD GROEP TABELLEN BESCHRIJVING

24 RESOURCE WERKTIJDEN GROEP



FIGUUR 23 - RESOURCE WERKTIJDEN TABELLEN

De gegroepeerde tabellen zoals weergegeven in Figuur 23, vormen samen de werktijden van een monteur. Ter verduidelijking is de Resource tabel opgenomen. Zie hoofdstuk 10 voor de beschrijving van de Resource tabel. De ResourceDateDefault tabel geeft aan welke dag de werktijden betreft. T_RESOURCE_WORK_DAYS is de verzameling van alle tijdsblokken voor een dag, waarbij periods_index het nummer van de dag van de week representeert. Resource_PeriodsOnDay bevat alle werktijd uitzonderingen. Een uitzondering kunnen vrije uren of vrije dagen zijn (Een monteur moet 2 uur eerder naar huis op een bepaalde datum of heeft verlof op een bepaalde datum). Exception_KEY is de datum waarop desbetreffende monteur een X aantal uur vrij heeft. Indien een monteur een hele dag vrij is, is dit een tijdsblok van bijvoorbeeld 08:00 – 16:30. Als een monteur op 1 dag twee uur later begint en twee uur later stopt, betreft het twee tijdsblokken van 08:00-10:00 en van 14:30-16:30. De koppeling met PeriodsOnDay, die weer gekoppeld is met PeriodOnDay (zie Figuur 22 van hoofdstuk 23) weergeeft de tijdsblokken van de uren dat een monteur vrij is.

BIJLAGE E - DATA WAREHOUSEDESIGN

DATA WAREHOUSE DESIGN

SMARTINSTALL



Jan Bongers
09057633
Versie: 1.0
Datum: 12/5/2016

INHOUDSOPGAVE

1	Inleiding.....	5
2	Asset Inventory Fact	6
3	Asset Placed Fact	9
4	Project Progress Fact	13
5	Workorder Failure Fact	16
6	Timeliness Fact.....	20
7	Productivity Fact	23
8	Meter Forecast Fact.....	26
9	Data Mapping.....	29

TABELLENLIJST

Tabel 1 - Design Onderdelen	5
Tabel 2 - user Stories Asset Inventory	6
Tabel 3 - Requirements Asset Inventory.....	6
Tabel 4 - Inventory Kenmerken Requirements.....	7
Tabel 5 - User Stories Asset Placed.....	9
Tabel 6 - Requirement Asset Placed	9
Tabel 7 - Asset Placed Kenmerken Requirements.....	10
Tabel 8 - User Stories Project Progress.....	13
Tabel 9 - Project Progress Requirements	13
Tabel 10 - Workorder Progress Kenmerken requirements	14
Tabel 11 - User Stories Workorder Failure	16
Tabel 12 - Workorder Failure Requirements	16
Tabel 13 - Workorder Failure Kenmerken Requirements	17
Tabel 14 - User Stories Timeliness	20
Tabel 15 - Timeliness Requirements.....	20
Tabel 16 - Timeliness Kenmerken Requirements	21
Tabel 17 - User Story productivity	23
Tabel 18 - Productivity Requirements	23
Tabel 19 - Productivity Kenmerken Requirements.....	24
Tabel 20 - User stories meter forecast	26
Tabel 21 - Meter forecast requirements	26
Tabel 22 - Meter forecast kenmerken requirements.....	27

FIGURENLIJST

Figuur 1 - Conceptual Asset Inventory Fact.....	6
Figuur 2 - Locigal Inventory Fact.....	8
Figuur 3 - Physical Asset Inventory Fact	8
Figuur 4 - Conceptual Asset Placed Fact.....	9
Figuur 5 - Logical Asset Placed Fact	11
Figuur 6 - Physical Asset Placed Fact	12
Figuur 7 – Conceptual project progress fact	13
Figuur 8 - Logical project progress fact	14
Figuur 9 - Physical project progress fact.....	15
Figuur 10 - Conceptual Workorder Failure Fact	16
Figuur 11 - Logical Workorder Failure Fact	18
Figuur 12 - Physical Workorder Failure Fact.....	19
Figuur 13 - Conceptual Timeliness Fact.....	20
Figuur 14 - Logical Timeliness Fact	21
Figuur 15 - Physical Timeliness Fact	22
Figuur 16 - Conceptual Productivity Fact	23
Figuur 17 - Logical Productivity Fact.....	24
Figuur 18 - Physical Productivity Fact	25
Figuur 19 - Conceptual meter forecast fact.....	26
Figuur 20 - Logical Meter forecast fact.....	27
Figuur 21 - Physical meter forecast fact	28

1 INLEIDING

Dit document is geschreven in het kader van data warehouseontwerpen. Per feitentabel worden er drie designs weergegeven. De designs betreffen een conceptual, logical en physical design. De complexiteit van het design neemt steeds een stap toe. Dit zorgt voor een duidelijk overzicht van hoe een tabel en zijn attributen, keys en datatypes tot stand zijn gekomen. Tabel 1 geeft weer hoe de complexiteit stapsgewijs toeneemt tussen de verschillende designs van een tabel. Ook wordt er per design aangegeven hoe het ontwerp tot stand is gekomen met behulp van de bijbehorende requirements. Dikgedrukte woorden van de requirements geven de dimensies aan en schuingedrukte onderstreepte woorden geven de feiten aan. Tot slot is er een datamapping weergegeven tussen de source en target tabellen.

Onderdeel	Conceptual	Logical	Physical
Entiteit namen	✓	✓	
Entiteit relaties	✓	✓	
Attributen		✓	
Primaire sleutels		✓	✓
Vreemde sleutels		✓	✓
Tabel namen			✓
Kolom namen			✓
Kolom data typen			✓

TABEL 1 - DESIGN ONDERDELEN

2 ASSET INVENTORY FACT

ID	User story	Req. Nr.
SH3	Als gebruiker wil de inventaris van meters per dag, per medewerker kunnen inzien zodat ik weet hoeveel meters een medewerker op een dag op zijn naam heeft gescand	K13,K14 K16
SH4	Als gebruiker wil ik kunnen selecteren over welke periode de inventaris van de meters getoond moet worden zodat ik een wekelijks, maandelijks of jaarlijks overzicht heb	K15
SH34	Als gebruiker kan ik het aantal defecte meters zien die op de naam van een monteur staan zodat ik weet hoeveel foutieve meters er in omloop zijn	E8A

TABEL 2 - USER STORIES ASSET INVENTORY

Nr.	Requirement
K13	Een gebruiker moet een dagelijkse snapshot van de inventaris van alle meters per medewerker kunnen inzien
K14	Een gebruiker moet de mogelijkheid hebben om de snapshot van de inventaris van alle meters te filteren op medewerker en status
K15	Een gebruiker kan de periode van de snapshot van de inventaris van alle meters selecteren, waarbij de minimale selectie een dag en de maximale selectie jaren is.
K16	Een gebruiker moet het <u>aantal meters</u> kunnen inzien die een medewerker op zijn naam heeft <u>geregistreerd</u>
E8A	Een gebruiker moet in het inventaris het <u>aantal defecte meters</u> kunnen inzien die een medewerker op zijn naam heeft staan

TABEL 3 - REQUIREMENTS ASSET INVENTORY

Het conceptual design van Asset Inventory Fact zoals weergegeven in Figuur 1 is gebaseerd op de user stories van Tabel 2 en de requirements van Tabel 3. Voor het inventaris feit zijn er gegevens van de monteur en de asset nodig. Een gebruiker moet de status van een asset kunnen achterhalen en de inventaris over een geselecteerde periode kunnen inzien. Het enige wijzigende element van een asset is de status van een asset. De statussen van een asset zijn vastgelegd en wijzigen nooit. Gezien een asset zelf nooit wijzigt is besloten om het wijzigende element van assets als aparte dimensie op te nemen. Het voordeel van een aparte asset status dimensie, is dat deze bij de aanmaak van de tabel kan worden gevuld. De geschiedenis van een asset wordt tevens automatisch correct bewaard in de feiten tabel. Met deze gegevens is het duidelijk dat er in totaal vier dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 2. De datatypes zijn gedefinieerd in het physical design van Figuur 3.



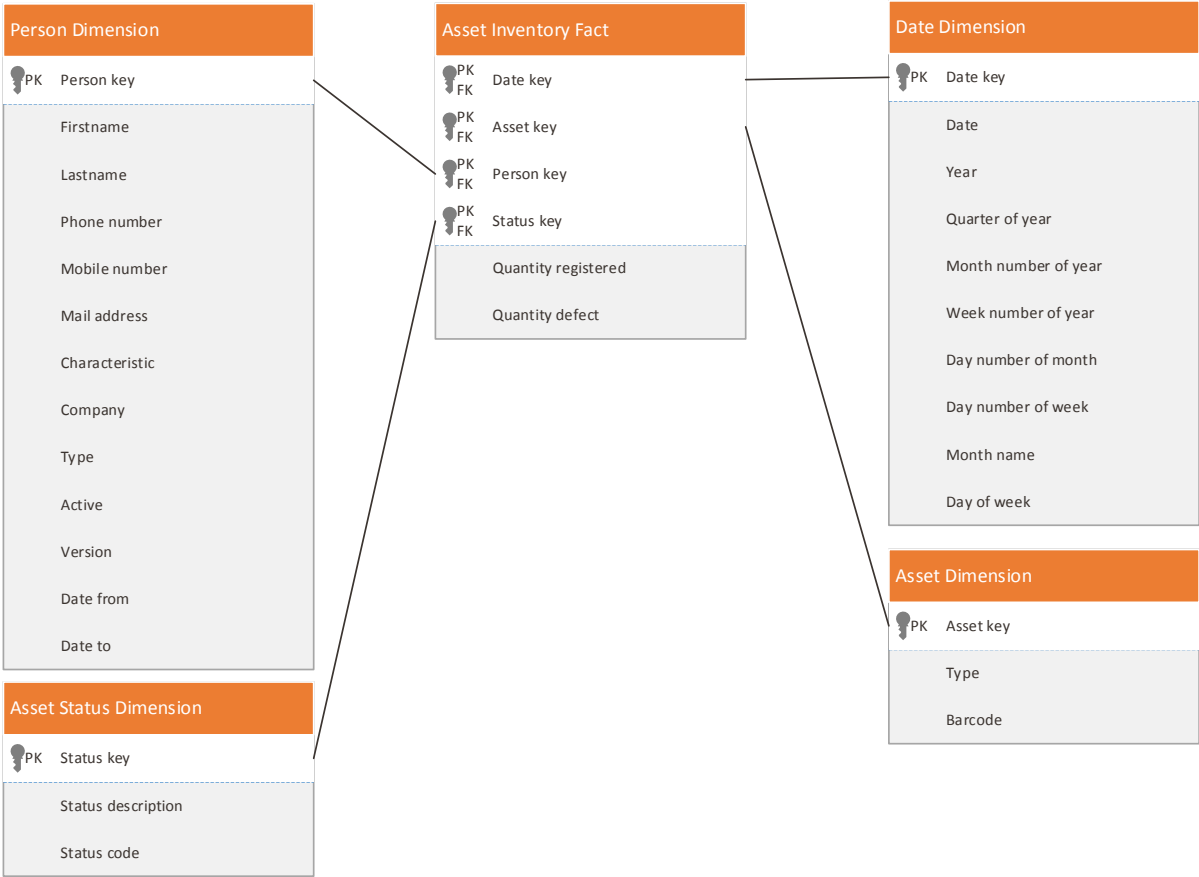
FIGUUR 1 - CONCEPTUAL ASSET INVENTORY FACT

Nr.	Requirement
K15A	Een gebruiker heeft bij een medewerker dezelfde informatie tot zijn beschikking als bij de brondata
K15	Een gebruiker kan de periode van de snapshot van de inventaris van alle meters selecteren, waarbij de minimale selectie een dag en de maximale selectie jaren is.
K17	Een gebruiker kan een datum van de rapporten filteren op de naam van een maand
K18	Een gebruiker kan de datum van de rapporten filteren op de week van het jaar
K15A	Een gebruiker heeft bij een medewerker dezelfde informatie tot zijn beschikking als bij de brondata
P1A	Een gebruiker kan inzien bij welke aannemer een monteur werkzaam is
P2A	Het systeem moet bijhouden bij welke aannemer een monteur werkzaam was indien een monteur van aannemer wisselt

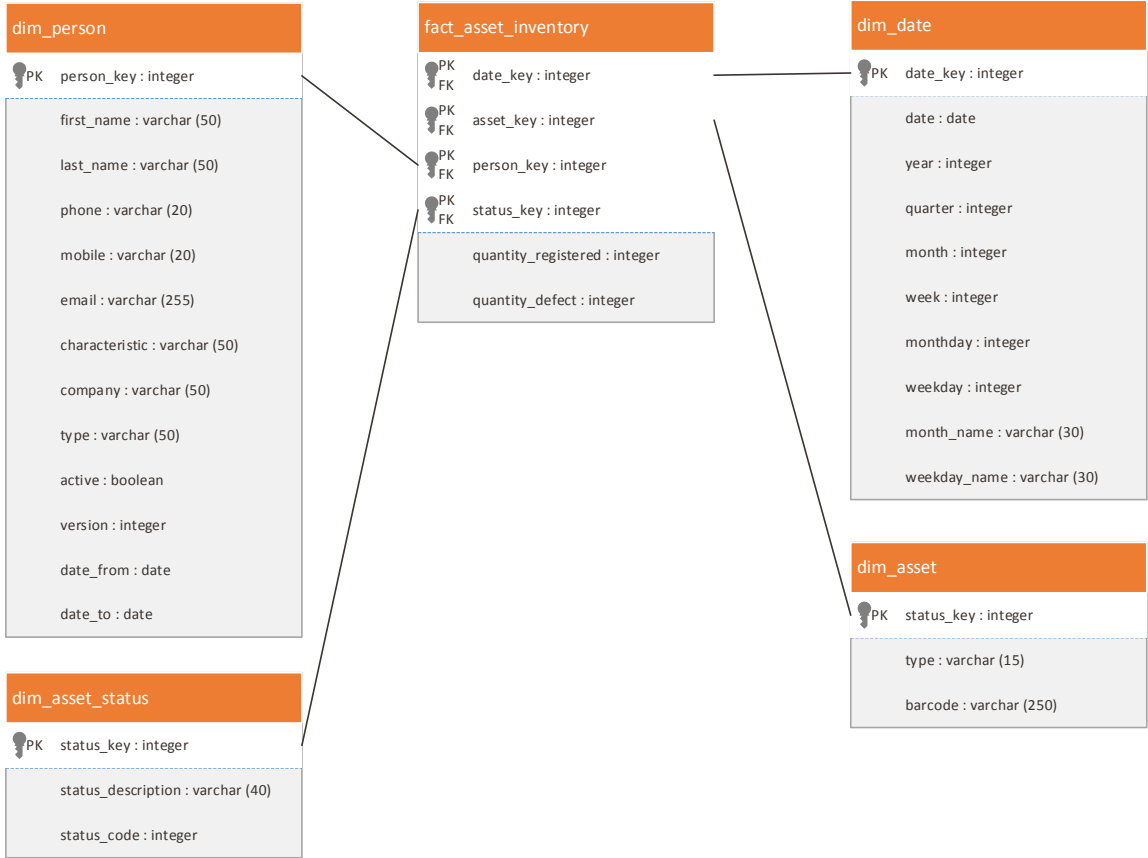
TABEL 4 - INVENTORY KENMERKEN REQUIREMENTS

De invulling van de dimensies en het feit met kenmerken is gebaseerd op de requirements weergegeven in Tabel 4. De date dimension dient minimaal een dag en maximaal een jaar te bevatten. Alle kenmerken tussen de genoemde twee kenmerken in, dienen ook aanwezig te zijn. De naam van de maand en de naam van de weekdag zijn ook opgenomen om aan requirements K17 en K18 te voldoen. Om aan de requirements P1A en P2A te voldoen, moet het bedrijf worden opgenomen in de warehousedatabase. De beslissing om deze als kenmerk bij person op te nemen en niet als dimensie, is omdat het enige relevante van een bedrijf de bedrijfsnaam is. Met het opnemen van company in person, dient de person dimension een SCD type 2 te zijn. Voor de realisatie van SCD type 2 zijn de kenmerken version, date from en date to toegevoegd. Een asset heeft in de broncode alleen een barcode. Met de barcode kan worden achterhaald welk type asset het betreft. Om dit eenvoudiger te maken voor gebruikers is het kenmerk type toegevoegd.

Bijlage E - Data warehouse design



FIGUUR 2 - LOGICAL INVENTORY FACT



FIGUUR 3 - PHYSICAL ASSET INVENTORY FACT

3 ASSET PLACED FACT

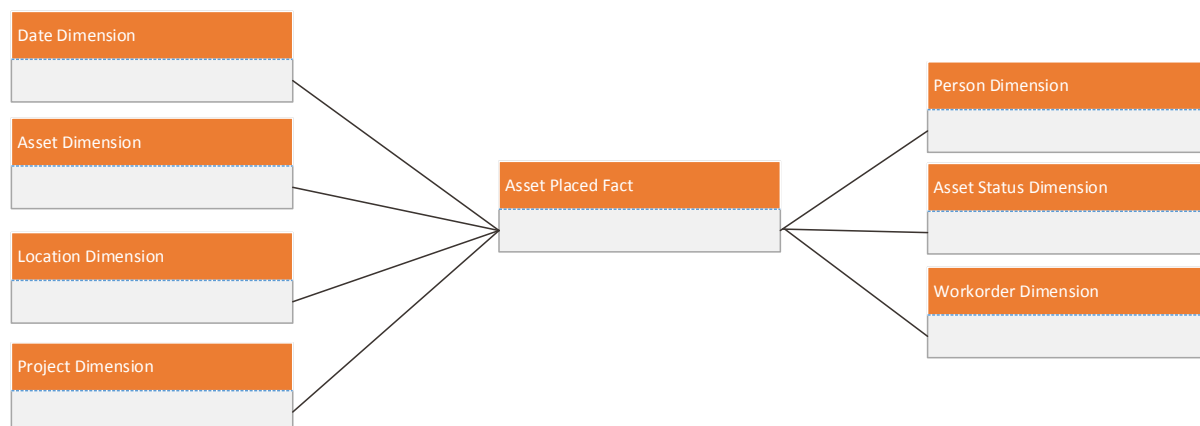
ID	User story	Req. Nr.
SH5	Als gebruiker wil ik de geplaatste meters per dag, per monteur, per locatie en per werkopdracht kunnen inzien zodat ik een overzicht heb van de locaties en werkopdrachten waar een meter is geplaatst	K19,K20 K21,K22 K23
SH18	Als gebruikers kan ik het aantal geregistreerde en het aantal ongeregistreeerde geplaatste meters achterhalen zodat ik kan bijsturen op het logistiek beheer	P5A

TABEL 5 - USER STORIES ASSET PLACED

Nr.	Requirement
K19	Een gebruiker moet een overzicht van <u>alle geplaatste meters</u> per monteur , per dag kunnen inzien
K20	Een gebruiker kan het <u>aantal meters</u> inzien die een monteur heeft <u>geplaatst</u>
K21	Een gebruiker moet een overzicht van alle geplaatste meters per project kunnen inzien
K22	Een gebruiker moet het overzicht van alle geplaatste meters kunnen filteren op locatie
K23	Een gebruiker kan is het geplaatste meters overzicht achterhalen bij welke werkopdracht een meter geplaatst is
P5A	Een gebruiker moet het <u>aantal ongeregistreeerde geplaatste</u> meters kunnen achterhalen

TABEL 6 - REQUIREMENT ASSET PLACED

Het conceptual design van Asset Placed Fact zoals weergegeven in Figuur 4 is gebaseerd op de user stories van Tabel 5 en de requirements van Tabel 6. Voor het geplaatste assets feit zijn er gegevens van asset en de monteur die de meter heeft geplaatst nodig. Een gebruiker moet de status van een asset kunnen achterhalen en de geplaatste meters over een geselecteerde periode kunnen inzien. Ook dient een gebruiker de locatie van plaatsing en de werkopdracht waarvoor de meter is geplaatst te kunnen achterhalen. Het overzicht moet ook het aantal geplaatste meters per project kunnen tonen. Project wordt niet bij werkopdracht opgenomen, maar als dimensie met het oog op de nabije toekomst. Project wordt binnenkort uitgebreid met kenmerken in SmartWork3.0, die ook in de warehouse database aanwezig moeten zijn. Deze gegevens geven aan dat er in totaal zeven dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 5. De datatypes zijn gedefinieerd in het physical design van Figuur 6.

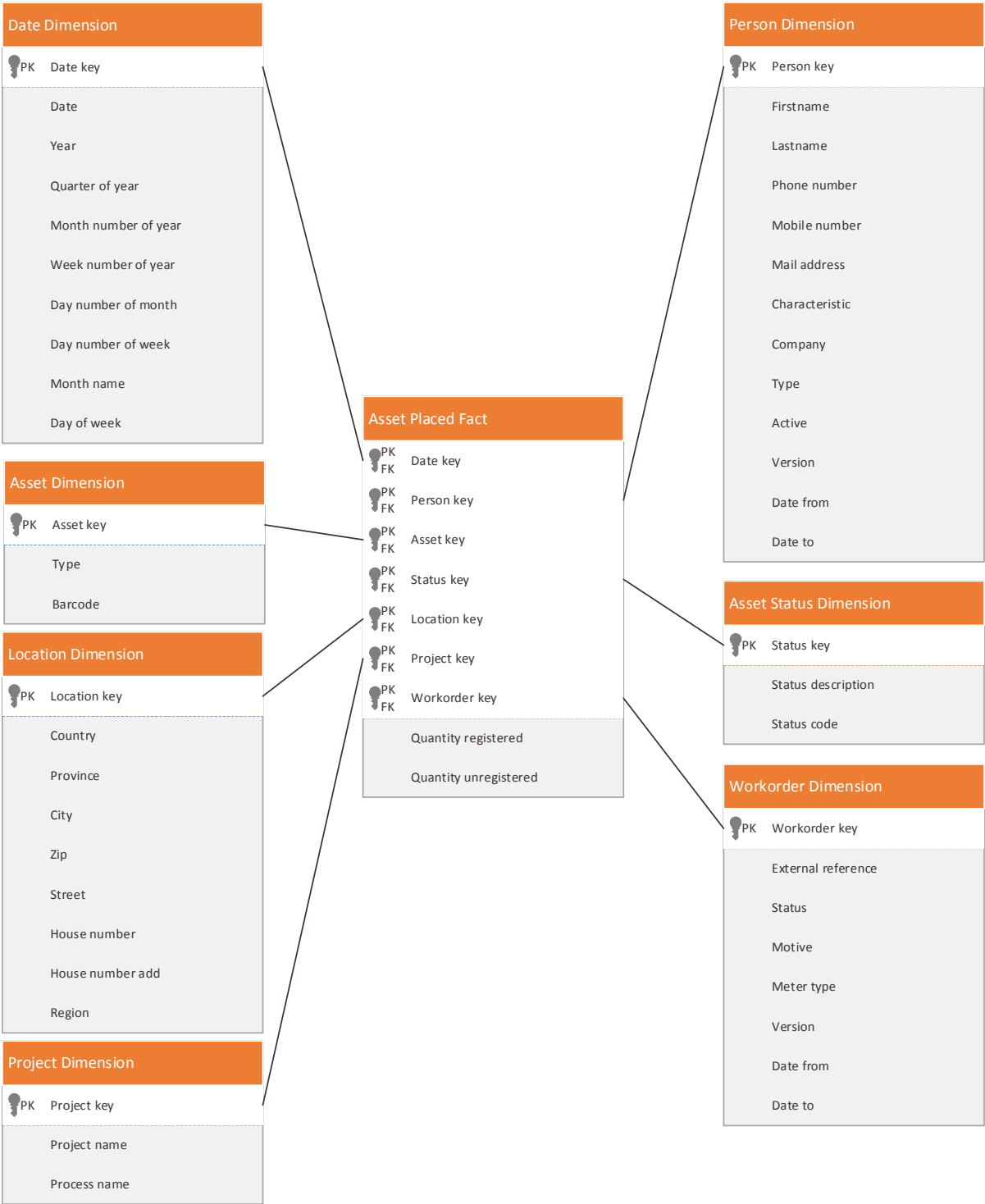


FIGUUR 4 - CONCEPTUAL ASSET PLACED FACT

Nr.	Requirement
K8	Een gebruiker moet per dag en per project kunnen inzien wat de voortgang is, inclusief huidige status van de werkopdrachten
K24	Een gebruiker heeft bij een locatie dezelfde informatie tot zijn beschikking als bij de brondata
K13A	Een gebruiker kan achterhalen aan welk SmartWork3.0 proces een project gekoppeld is
P10	Een gebruiker moet bij een werkopdracht de aanleiding kunnen inzien
P11	Een gebruiker moet bij een werkopdracht het metertype kunnen inzien
P12	Een gebruiker moet de regio waar de meter is geplaatst kunnen inzien
E3A	Een gebruik kan het land van een locatie achterhalen
E4A	Een gebruiker kan de provincie van een locatie achterhalen

TABEL 7 - ASSET PLACED KENMERKEN REQUIREMENTS

De invulling van de dimensies en het feit met kenmerken is gebaseerd op de requirements weergegeven in Tabel 7. De dimensies date, person, asset en asset status zijn in eerdere designs al ingevuld. De requirements voor asset placed fact eisen geen wijzigingen aan deze eerder opgestelde dimensies. Een project bestaat in de bron alleen uit een project naam. Om aan requirement K13A te voldoen, is het kenmerk process name toegevoegd. De locatie dimensie moet dezelfde informatie bevatten als de informatie uit de brondata. Locatie dimensie is aangevuld met de kenmerken regio, provincie en land om aan requirements P12, E3A en E4A te voldoen. Met het invullen van de werkopdracht dimensie is rekening gehouden met requirements K8. Requirement K8 dient niet als requirements voor asset placed, maar heeft wel impact op de werkopdracht dimensie behorende bij asset placed. Een werkopdracht bestaat uit niets anders dan een externe referentie. Het status kenmerk is toegevoegd ten behoeve van K8 en zorgt er voor dat het een SCD type 2 dimensie betreft om de geschiedenis intact te houden bij status wijzigingen. In tegenstelling tot asset status, is bij werkopdracht status gekozen om deze in de werkopdracht dimensie als kenmerk op te nemen. De statussen van een werkopdracht zijn niet van te voren vastgelegd en kunnen niet achterhaald worden met de Smartwork3 database. Alle mogelijke statussen van een werkopdracht zijn in-memory opgeslagen. Met het opnemen van status in werkopdracht hoeft er maar één dimensie bijgehouden te worden in plaats van twee. Verder zijn de kenmerken meter type en aanleiding toegevoegd ten behoeve van requirement P10 en P11.



FIGUUR 5 - LOGICAL ASSET PLACED FACT



FIGUUR 6 - PHYSICAL ASSET PLACED FACT

4 PROJECT PROGRESS FACT

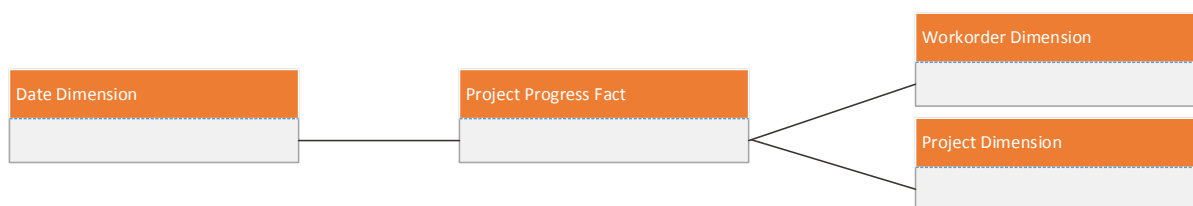
ID	User story	Req. Nr.
SH1	Als gebruiker wil ik de huidige status van een werkopdracht per dag, per project inzien zodat ik de voortgang van een project kan bijhouden	K8
SH2	Als gebruiker kan ik de aantallen van de 'eind' statussen zien zodat ik inzicht heb in de verhouding tussen de werkopdrachten	K9,K10, K11,K12

TABEL 8 - USER STORIES PROJECT PROGRESS

Nr.	Requirement
K8	Een gebruiker moet per dag en per project kunnen inzien wat de voortgang is, inclusief huidige status van de werkopdrachten

TABEL 9 - PROJECT PROGRESS REQUIREMENTS

Het conceptual design van Project Progress Fact zoals weergegeven in Figuur 7 is gebaseerd op de user stories van Tabel 8 en de requirements van Tabel 9. Voor het voortgangsfeit zijn er gegevens van een werkopdracht nodig. Een gebruiker moet de voortgang per dag en/of per project kunnen inzien. Deze gegevens geven aan dat er in totaal drie dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 8. De datatypes zijn gedefinieerd in het physical design van Figuur 9.

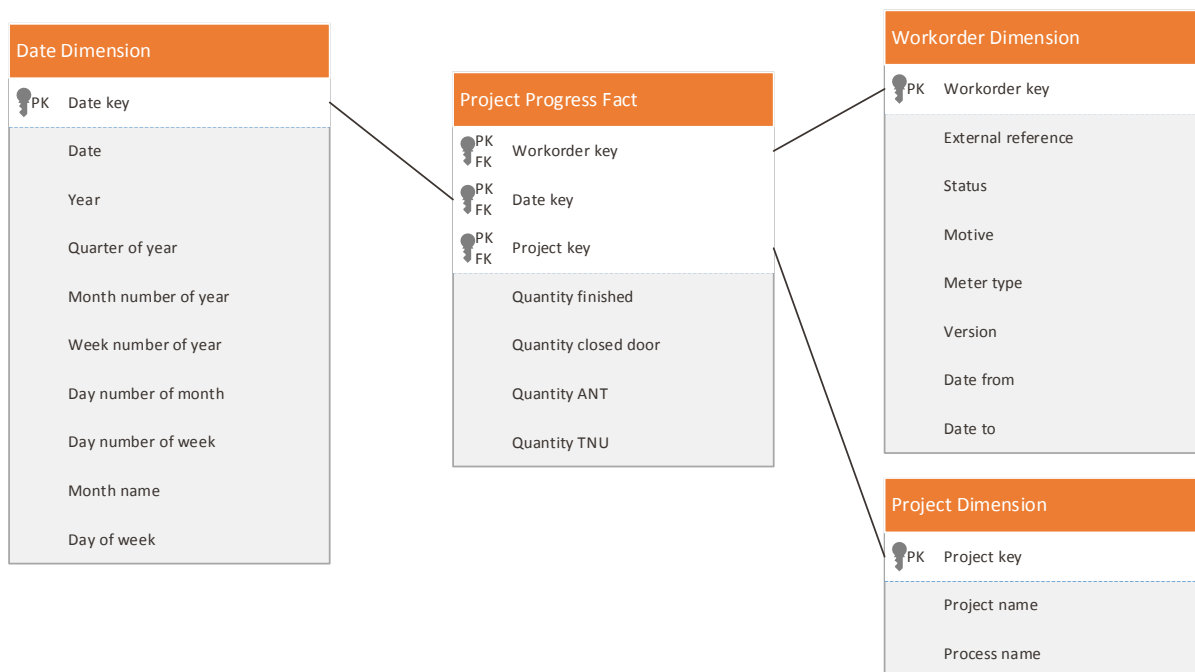


FIGUUR 7 – CONCEPTUAL PROJECT PROGRESS FACT

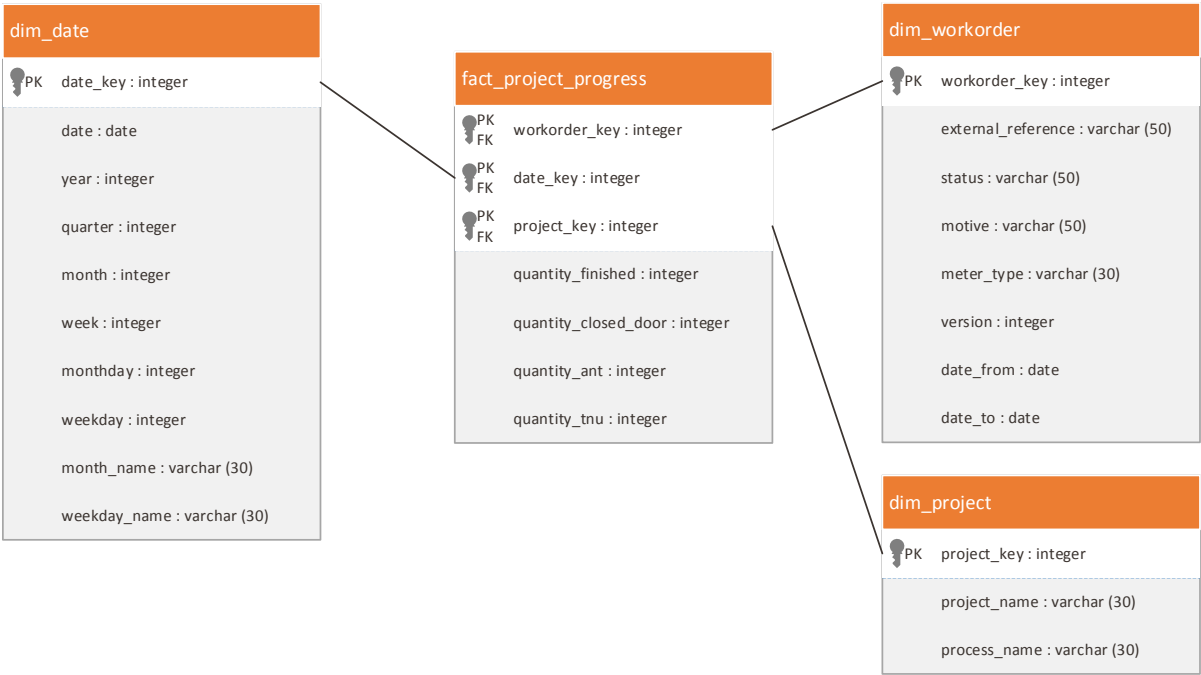
Nr.	Requirement
K9	Een gebruiker kan per dag, per project zien <u>hoeveel</u> werkopdrachten er <u>succesvol afgerond</u> zijn
K10	Een gebruiker kan per dag, per project zien <u>hoeveel dichte deuren</u> er zijn gemeld
K11	Een gebruiker kan per dag, per project zien <u>hoeveel</u> adressen <u>niet toegankelijk</u> waren
K12	Een gebruiker kan per dag, per project zien <u>hoeveel</u> werkopdrachten er <u>technisch niet uitvoerbaar</u> waren

TABEL 10 - WORKORDER PROGRESS KENMERKEN REQUIREMENTS

De invulling van het feit met kenmerken is gebaseerd op de requirements weergegeven in Tabel 10. De dimensies date, workorder en project zijn in eerdere designs al ingevuld. De requirements voor project progress eisen geen wijzigingen aan deze eerder opgestelde dimensies. Om aan de requirements K9, K10, K11 en K12 te voldoen, zijn er vier feit kenmerken aan de feitentabel toegevoegd. Alle vier betreft een aantal van de registraties van de zogenoemde ‘eind’ statussen.



FIGUUR 8 - LOGICAL PROJECT PROGRESS FACT



FIGUUR 9 - PHYSICAL PROJECT PROGRESS FACT

5 WORKORDER FAILURE FACT

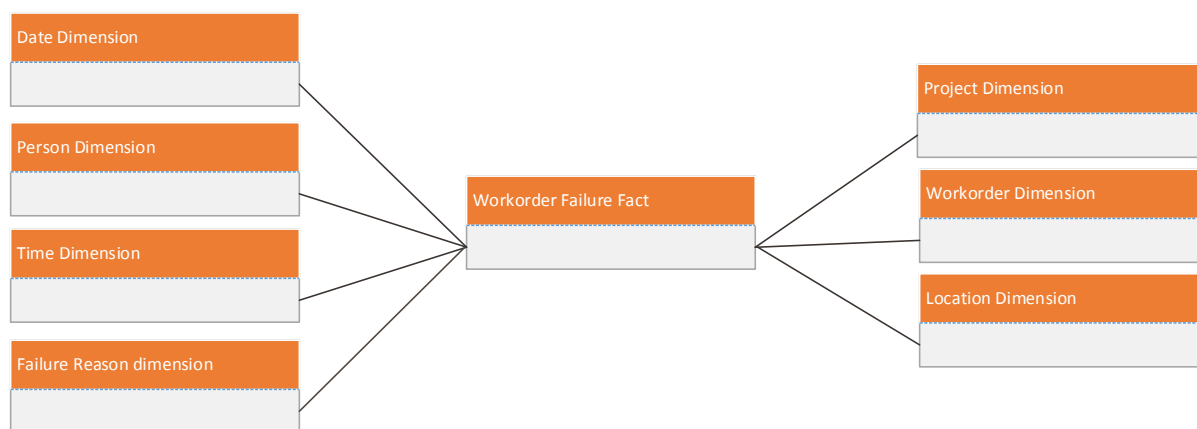
ID	User story	Req. Nr.
SH21	Als gebruiker wil ik per project, per werknemer en per dag het aantal uitgevallen werkopdrachten, inclusief het tijdstip van registratie, kunnen inzien zodat ik werknemers en klanten kan bijsturen	P14,P16 P14A
SH22	Als gebruiker wil ik de reden van uitval en het commentaar van uitval registratie kunnen zien zodat ik weet of het met een geldige reden als uitval is geregistreerd	P15,P17

TABEL 11 - USER STORIES WORKORDER FAILURE

Nr.	Requirement
P14	Een gebruiker moet per dag , per project een rapport van alle uitgevallen werkopdrachten kunnen inzien
P15	Een gebruiker kan in het rapport van uitgevallen werkopdrachten achterhalen wat de reden van uitval is
P16	Een gebruiker kan in het rapport van uitgevallen werkopdrachten achterhalen welke werknemer de uitval heeft geregistreerd
P14A	Een gebruiker moet het tijdstip van uitval registratie kunnen achterhalen
E7A	Een gebruiker heeft bij uitgevallen werkopdracht inzicht in het type uitval

TABEL 12 - WORKORDER FAILURE REQUIREMENTS

Het conceptual design van Workorder Failure Fact zoals weergegeven in Figuur 10 is gebaseerd op de user stories van Tabel 11 en de requirements van Tabel 12. Voor het failure feit zijn er gegevens van een werkopdracht en het type uitval nodig. Een gebruiker moet de failures per dag en/of per project kunnen inzien en kunnen achterhalen welke monteur de failure geregistreerd heeft. Verder is het van belang dat een gebruiker het tijdstip van de failure registratie weet en welke locatie het betreft. Deze gegevens geven aan dat er in totaal zes dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 11. De datatypes zijn gedefinieerd in het physical design van Figuur 12.



FIGUUR 10 - CONCEPTUAL WORKORDER FAILURE FACT

Nr.	Requirement
P17	Een gebruiker kan het commentaar van de uitval registratie bij uitgevallen werkopdrachten inzien

TABEL 13 - WORKORDER FAILURE KENMERKEN REQUIREMENTS

De invulling van het feit met kenmerken is gebaseerd op de requirements weergegeven in Tabel 13. Zoals is te zien in de requirement, is het kenmerk niet als feiten aangegeven. Het dikgedrukte woord geeft in feite een kenmerk aan voor de 'failure reason' dimensie. Commentaar hoort echter niet thuis in de failure reason dimensie en is dus opgenomen in de feit tabel. Met de dimensies date, workorder, project, location en person zijn in eerdere designs al ingevuld. De requirements voor workorder failure eisen geen wijzigingen aan deze eerder opgestelde dimensies. De invulling van de dimensie tijd is gebaseerd op de data van SmartWork3.0. Een tijdstip bestaat in SmartWork3.0 gedeeltelijk uit uur en minuut. In sommige gevallen worden de seconden van registratie niet geregistreerd. Ook is de precisie van seconden niet relevant voor de wenselijke inzichten.



FIGUUR 11 - LOGICAL WORKORDER FAILURE FACT



FIGUUR 12 - PHYSICAL WORKORDER FAILURE FACT

6 TIMELINESS FACT

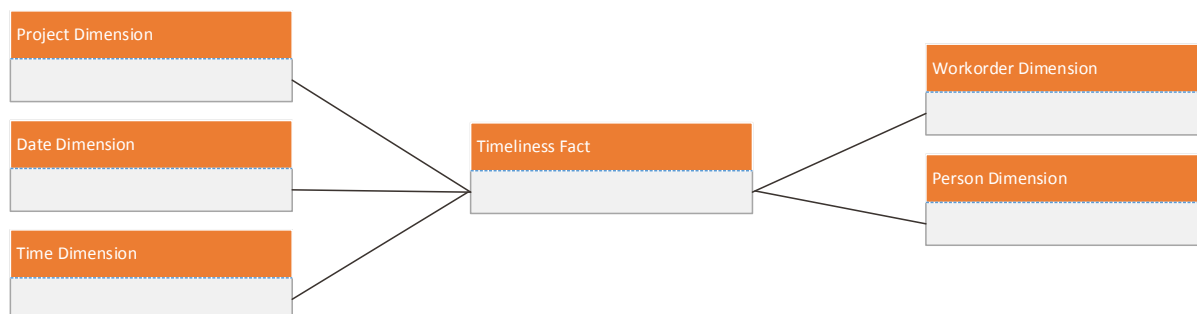
ID	User story	Req. Nr.
SH16	Als gebruiker kan ik de tijdigheid van monteurs per dag per tijdvak per werkopdracht of per project in zien en of de monteur de afspraak op de afgesproken dag heeft uitgevoerd zodat ik weet welke monteurs ik moet aansturen	P1,P4,P5 P6,P7, P8A, P12A
SH17	Als gebruiker kan ik per dag, per project en per werkopdracht zien hoeveel minuten te vroeg of te laat een monteur bij zijn afspraak is zodat ik weet of een monteur aanhoudend te laat of te vroeg is	P2,P3

TABEL 14 - USER STORIES TIMELINESS

Nr.	Requirement
P1	Een gebruiker moet een dagelijks overzicht van de tijdigheid per monteur kunnen zien
P8A	Een gebruiker moet de tijdigheid van monteurs per project kunnen inzien
P12A	Een gebruiker kan zien welke werkopdracht het betreft in het tijdigheid overzicht en wat het tijdstip van de afspraak was

TABEL 15 - TIMELINESS REQUIREMENTS

Het conceptual design van Timeliness Fact zoals weergegeven in Figuur 13 is gebaseerd op de user stories van Tabel 11 en de requirements van Tabel 12. Voor de realisatie van het timeliness feit zijn de gegevens betreft project, datum, tijd, werkopdracht en monteur nodig. Een gebruiker moet kunnen achterhalen wat het ingeplande tijdsblok is en bij welke werkopdracht betreffende afspraak hoort. Deze gegevens geven aan dat er in totaal vijf dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 14. De datatypes zijn gedefinieerd in het physical design van Figuur 15.

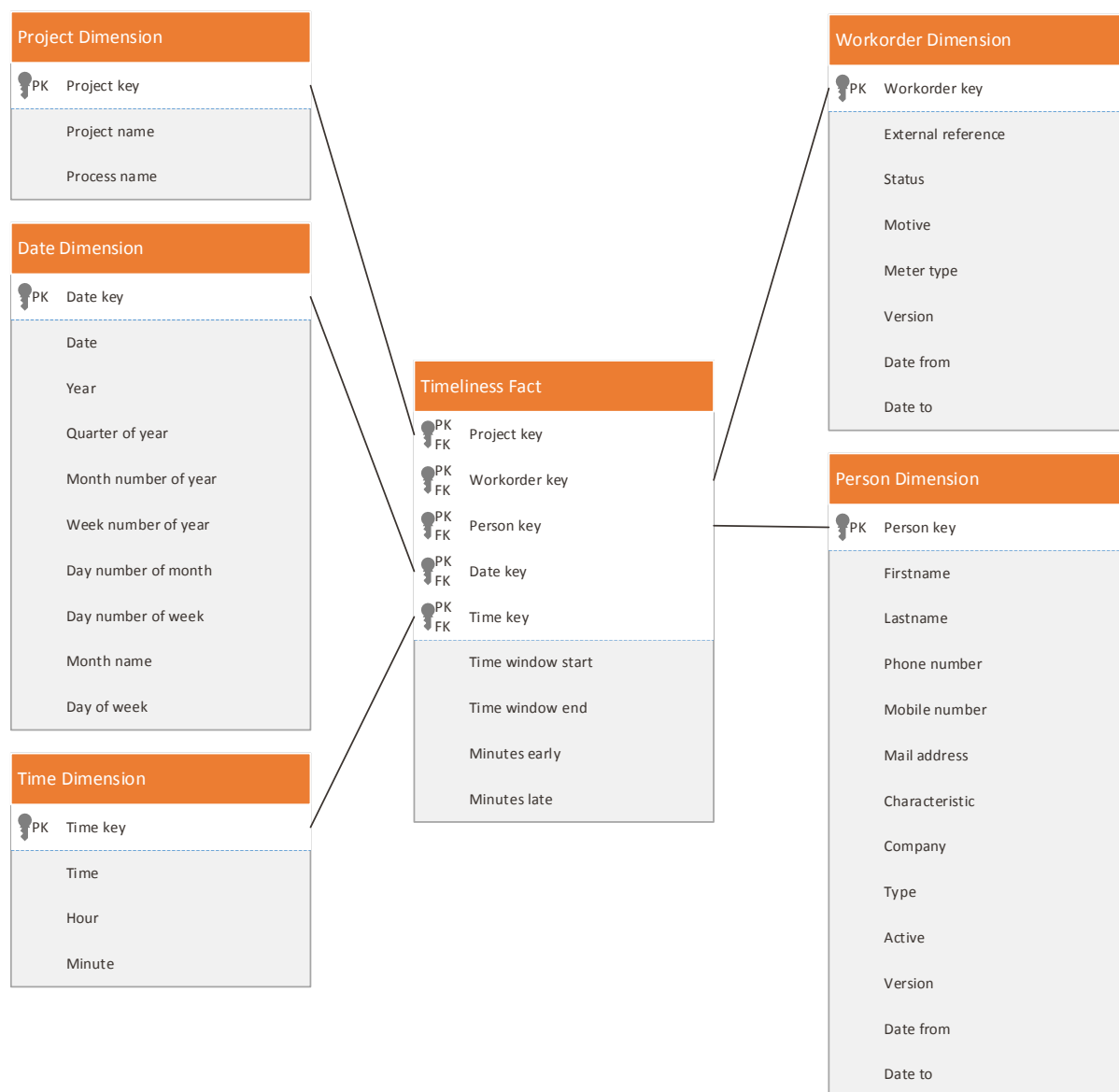


FIGUUR 13 - CONCEPTUAL TIMELINESS FACT

Nr.	Requirement
P4	Een gebruiker moet het <u>aantal minuten te vroeg</u> bij afspraken per monteur kunnen zien
P5	Een gebruiker moet het <u>aantal minuten te laat</u> bij afspraken per monteur kunnen zien
P6	Een gebruiker kan bij een tijdigheid overzicht inzien wat de <u>ingeplande starttijd</u> van het tijdsblok van de afspraak is
P7	Een gebruiker kan bij een tijdigheid overzicht inzien wat de <u>ingeplande eindtijd</u> van het tijdsblok van de afspraak is
P15A	Een gebruiker kan in het tijdigheid overzicht zien of de <u>afpraak</u> op de <u>afgesproken dag</u> is uitgevoerd

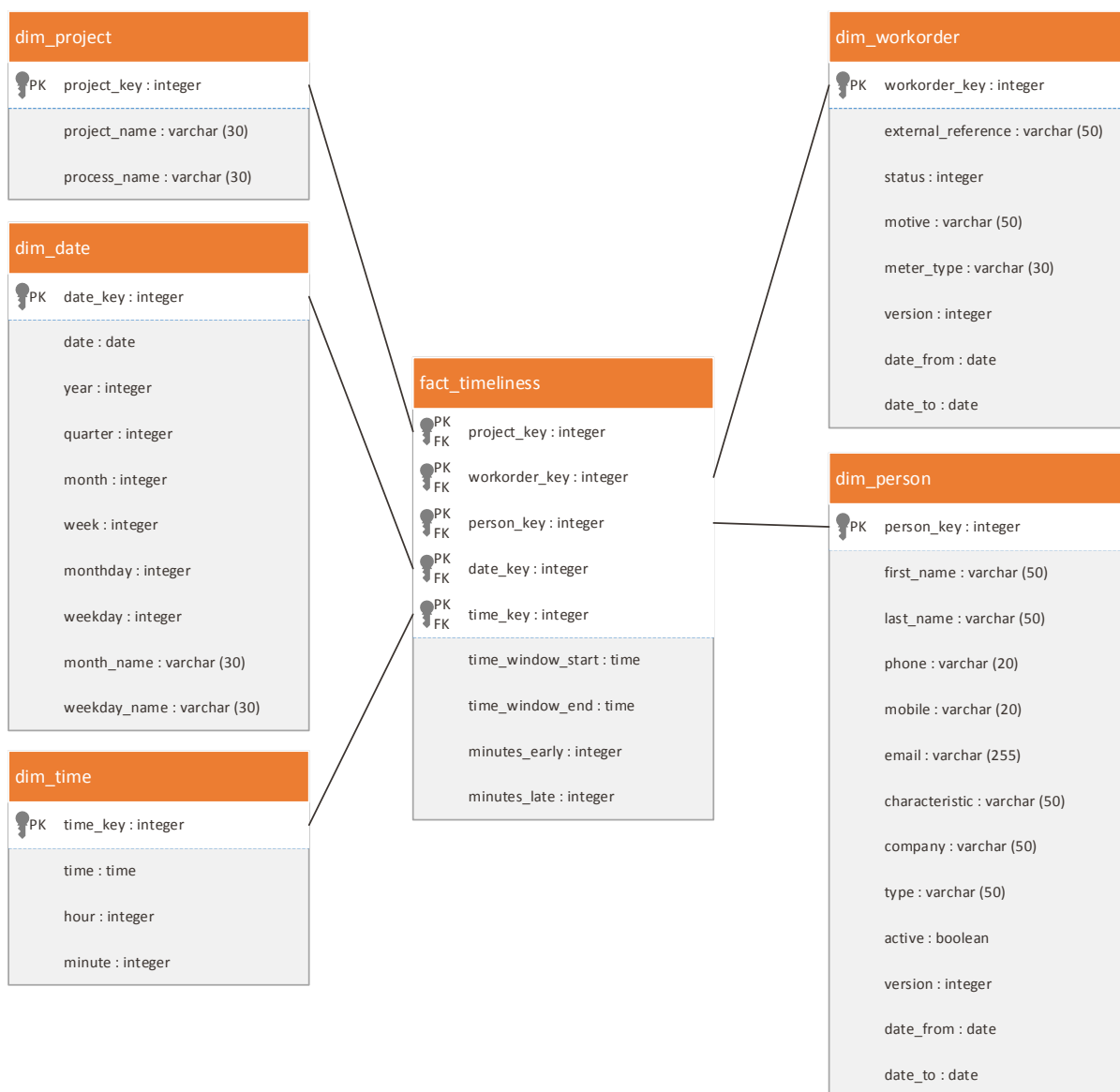
TABEL 16 - TIMELINESS KENMERKEN REQUIREMENTS

Voor de realisatie van de requirements uit Tabel 16 zijn de kenmerken minutes early en minutes late aan het Timeliness fact toegevoegd. Voor het kunnen inzien van de ingeplande start- en eindtijd zijn de kenmerken time window start en time window end toegevoegd. De dimensies date, workorder, project, person en time zijn in eerdere designs al ingevuld. De requirements voor timeliness eisen geen wijzigingen aan deze eerder opgestelde dimensies.



FIGUUR 14 - LOGICAL TIMELINESS FACT

Bijlage E - Data warehouse design



FIGUUR 15 - PHYSICAL TIMELINESS FACT

7 PRODUCTIVITY FACT

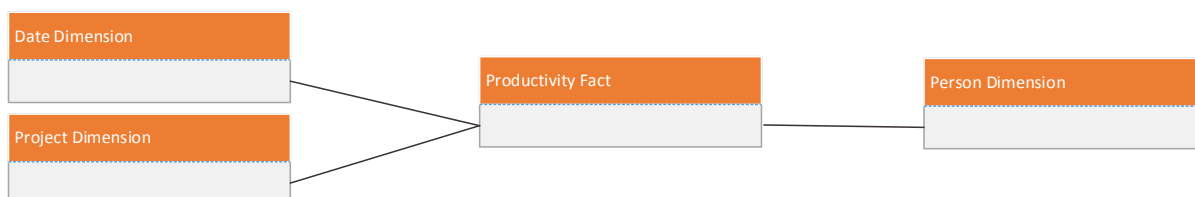
ID	User story	Req. Nr.
SH25	Als gebruiker wil ik per project, per dag en per monteur de productiviteit kunnen inzien zodat ik weet welke monteurs wel of niet productief zijn en hierop kan inspelen	P22,P23, P24,P25, P26,P27, P13A

TABEL 17 - USER STORY PRODUCTIVITY

Nr.	Requirement
P22	Een gebruiker moet per dag de <u>productiviteit</u> van een monteur kunnen inzien
P13A	Een gebruiker moet kunnen achterhalen voor welk project het meerwerk is uitgevoerd

TABEL 18 - PRODUCTIVITY REQUIREMENTS

Het conceptual design van Productivity Fact zoals weergegeven in Figuur 16 is gebaseerd op de user story van Tabel 17 en de requirements van Tabel 18. Om de requirements betreft de Productivity Fact te kunnen realiseren, zijn er gegevens van project en monteur nodig. Een gebruiker dient de productiviteit van een monteur per dag in te zien. Deze gegevens geven aan dat er in totaal drie dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 17. De datatypes zijn gedefinieerd in het physical design van Figuur 18.

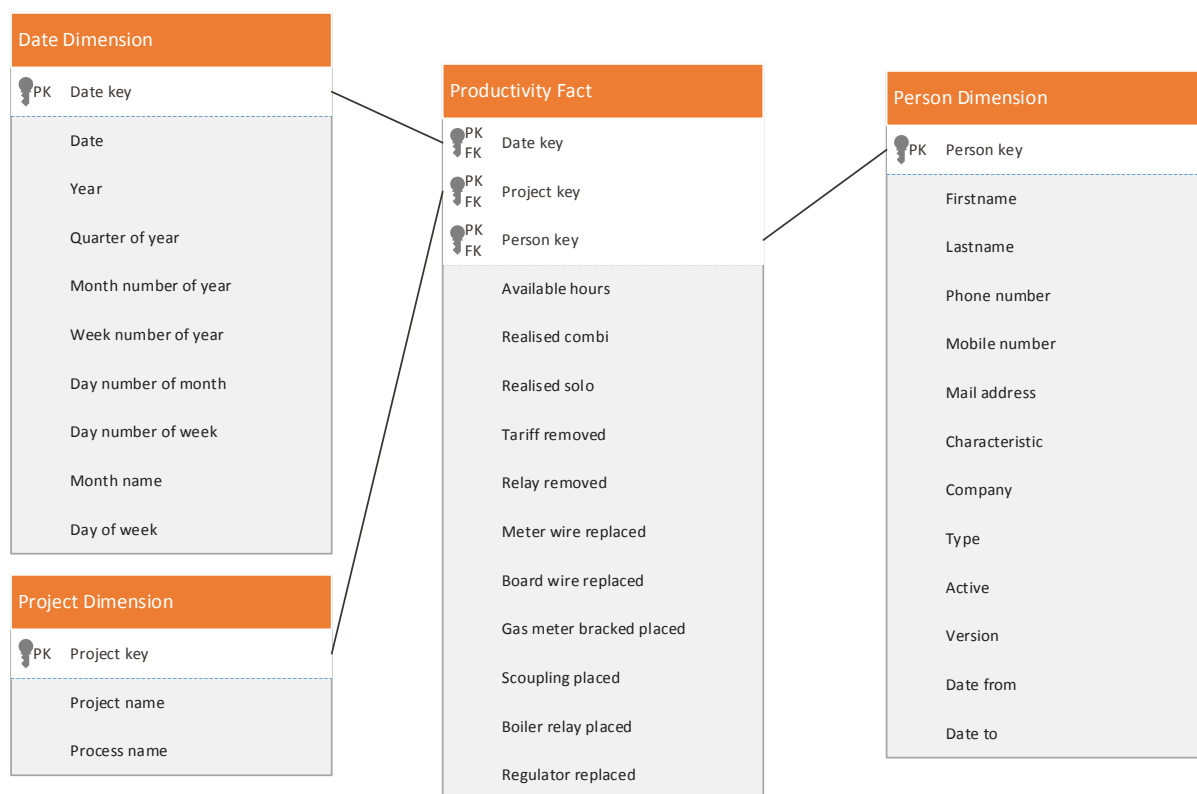


FIGUUR 16 - CONCEPTUAL PRODUCTIVITY FACT

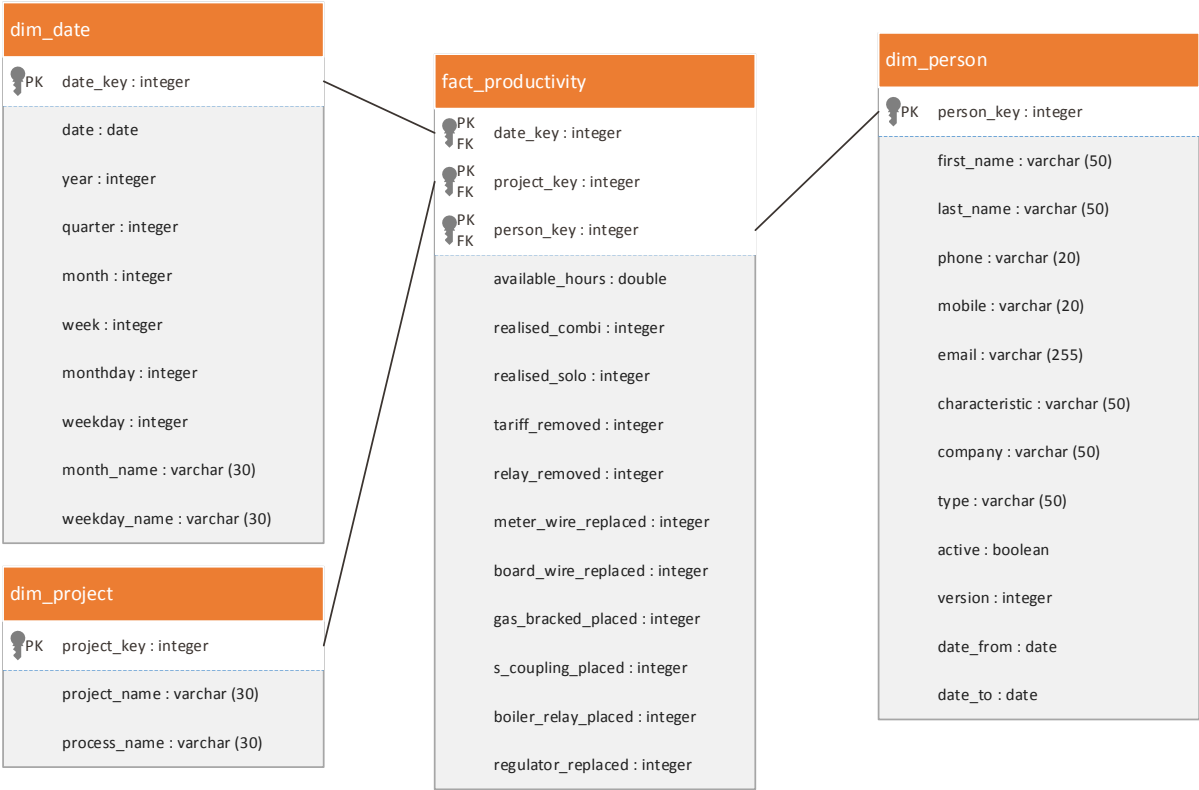
Nr.	Requirement
P23	Een gebruiker kan bij de productiviteit per dag, per monteur zien wat het <u><i>gantal beschikbaar gestelde uren</i></u> was
P24	Een gebruiker kan inzien hoeveel <u><i>solo opdrachten</i></u> een monteur per dag heeft uitgevoerd
P25	Een gebruiker kan inzien hoeveel <u><i>combi opdrachten</i></u> een monteur per dag heeft uitgevoerd
P26	Een gebruiker kan inzien welke <u><i>meerwerk werkzaamheden</i></u> een monteur heeft gerealiseerd
P27	Een gebruiker kan <u><i>per meerwerk</i></u> inzien <u><i>hoe vaak</i></u> een monteur betreffende meerwerk op een dag heeft uitgevoerd

TABEL 19 - PRODUCTIVITY KENMERKEN REQUIREMENTS

De requirements van Tabel 19 geven aan welke kenmerken het productivity fact tabel moet bevatten. De meerwerk kenmerken zijn gebaseerd op de definities zoals ze zijn vastgelegd volgens de business.



FIGUUR 17 - LOGICAL PRODUCTIVITY FACT



FIGUUR 18 - PHYSICAL PRODUCTIVITY FACT

8 METER FORECAST FACT

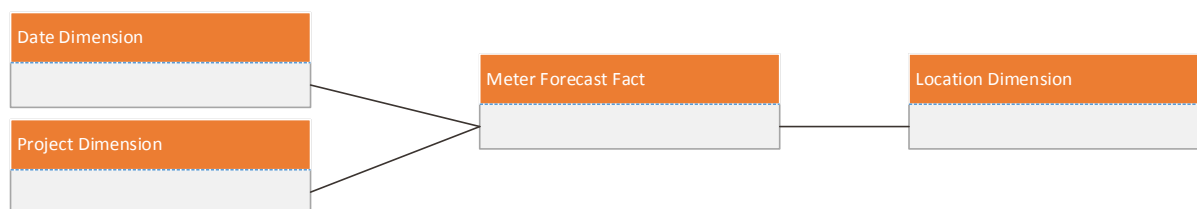
ID	User story	Req. Nr.
SH29	Als gebruiker heb ik van elk type meter inzicht in de meest recente voorspelling van het aantal benodigde meters voor de aankomende twee weken zodat ik weet of er voldoende aantal meters in voorraad zijn	E5,E6,E7 ,E8,E9, E10, E11, E12, E14
SH30	Als gebruiker kan ik de voorspelling van het aantal benodigde meters per regio en per project achterhalen zodat ik de project en regio managers op de hoogte kan stellen indien er niet voldoende meter in voorraad zijn	E13

TABEL 20 - USER STORIES METER FORECAST

Nr.	Requirement
E5	Een gebruiker moet per dag een voorspelling van het <u>aantal benodigde meters</u> voor de aankomende twee weken kunnen inzien
E13	Een gebruiker kan de voorspelling van het aantal benodigde meters per regio en per project inzien

TABEL 21 - METER FORECAST REQUIREMENTS

Het conceptual design van Meter forecast Fact zoals weergegeven in Figuur 19 is gebaseerd op de user stories van Tabel 20 en de requirements van Tabel 21. Zoals uit de dikgedrukte woorden van de requirements is te lezen, zijn er gegevens van project en regio nodig. De voorspelling dient dagelijks voor de aankomende twee weken te worden gedaan. Deze gegevens geven aan dat er in totaal drie dimensies nodig zijn. De toegewezen kenmerken van de tabellen zijn weergegeven in het logical design van Figuur 20. De datatypes zijn gedefinieerd in het physical design van Figuur 21.

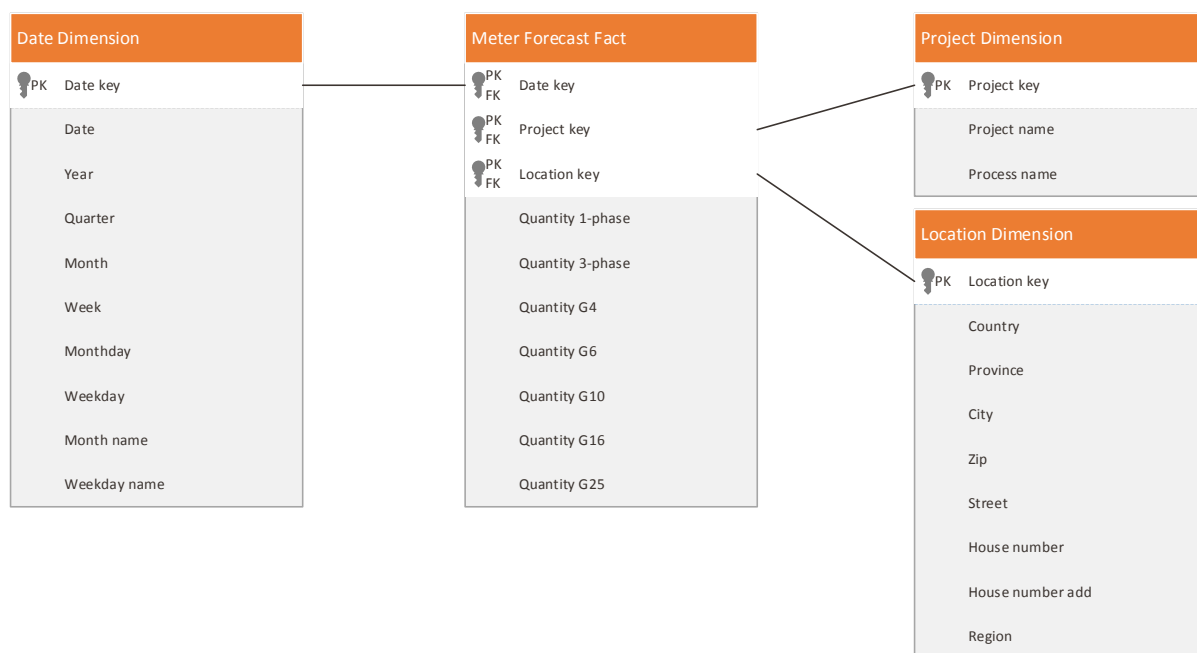


FIGUUR 19 - CONCEPTUAL METER FORECAST FACT

Nr.	Requirement
E6	Een gebruiker moet kunnen achterhalen <u>hoeveel 1-fase meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E7	Een gebruiker moet kunnen achterhalen <u>hoeveel 3-fase meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E8	Een gebruiker moet kunnen achterhalen <u>hoeveel G4 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E9	Een gebruiker moet kunnen achterhalen <u>hoeveel G6 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E10	Een gebruiker moet kunnen achterhalen <u>hoeveel G10 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E11	Een gebruiker moet kunnen achterhalen <u>hoeveel G16 meters</u> er naar schatting voor de aankomende twee weken nodig zijn
E12	Een gebruiker moet kunnen achterhalen <u>hoeveel G25 meters</u> er naar schatting voor de aankomende twee weken nodig zijn

TABEL 22 - METER FORECAST KENMERKEN REQUIREMENTS

De requirements van Tabel 22 geven aan welke kenmerken het meter forecast fact tabel moet bevatten. Voor alle hoeveelheden per metertype is een feit kenmerk toegevoegd aan de tabel.



FIGUUR 20 - LOGICAL METER FORECAST FACT

Bijlage E - Data warehouse design



FIGUUR 21 - PHYSICAL METER FORECAST FACT

9 DATA MAPPING

In onderstaande tabel staat de mapping van source- naar targetdatabase per kolom weergegeven. De mapping dient als input voor het bouwen van de ETL-module. De dimensies time, asset status en date zijn niet in de mapping opgenomen. Deze dimensies worden bij aanmaak gevuld met behulp van een SQL script. Time Dimension wordt gevuld met de uren en minuten van een periode van 24 uur. Date dimension wordt gevuld met data over een periode van 10 jaar, lopend van 1 januari 2015 tot en met 31 December 2025. Asset Status dimensie wordt gevuld met alle va tevoren vastgelegde status codes en bijbehorende beschrijving die een asset kan hebben. De betekenis van de componenten van de mapping tabel is hieronder beschreven.

- **Target table:** De fysieke naam van de tabel zoals die is terug te vinden in de warehouse
- **Target Column:** De naam van de kolom in de warehouse tabel
- **Target Data type:** Het data type van de kolom in de warehouse tabel
- **Table type:** Geeft aan of de tabel een feit of dimensie is
- **SCD:** Weergeeft voor dimensie tabellen of het een type 1, 2 of 3 slowly changing dimensions betreft. Het type kan voor elke kolom in de dimensie anders zijn.
- **Source database:** De naam van de database waar de source data zich bevindt. Het kan ook de naam van een bestand betreffen indien de source een (tekst)bestand is.
- **Source table:** De naam van de tabel waar de source data vandaan komt.
- **Source column:** De naam van de kolom van de source tabel. Kolommen die nodig zijn uit andere tabellen om de data uit deze kolom te verkrijgen, wordt gedocumenteerd in de transformation component middels SQL.
- **Source Data type:** Het data type van de kolom in de source tabel
- **Transformation:** Geeft eventueel een extra beschrijving van de acties die nodig zijn om de data te verkrijgen of de SQL voor het verkrijgen van data uit de source.

Bijlage E - Data warehouse design

Target					Source				Transformation
Table	Column	Data Type	Table	SCD	Database	Table	Column	Data Type	-
dim_person	person_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_person	first_name	varchar (50)	dimension	1	Smartwork3	User	first_name	varchar (50)	-
dim_person	last_name	varchar (50)	dimension	1	Smartwork3	User	last_name	varchar (50)	-
dim_person	phone	varchar (20)	dimension	1	Smartwork3	User	phone	varchar (20)	-
dim_person	mobile	varchar (20)	dimension	1	Smartwork3	User	mobile	varchar (20)	-
dim_person	email	varchar (20)	dimension	1	Smartwork3	User	email	varchar (254)	-
dim_person	characteristic	varchar (50)	dimension	0	Smartwork3	User	login	varchar (50)	-
dim_person	company	varchar (50)	dimension	2	Smartwork3	Company	name	varchar (255)	select c.name from user u, resource r, company c u.id = r.user_id and r.company_id = c.id. Controle op naam lengte
dim_person	type	varchar (50)	dimension	1	Smartwork3	User	userRole_roleName	varchar (255)	Op basis van de rol dient het type (monteur, front office e.d.) bepaald te worden. Waarde 'onbekend' indien het type niet kan worden bepaald
dim_person	active	boolean	dimension	1	Smartwork3	User	activated	tinyint	-
dim_asset	asset_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_asset	type	varchar (15)	dimension	0	Smartwork3	Asset	barcode	varchar (255)	Type van asset bepalen aan de hand van de barcode. Waarde 'onbekend' indien het type niet kan worden bepaald.
dim_asset	barcode	varchar(250)	dimension	0	Smartwork3	Asset	barcode	varchar (255)	select a.barcode from asset a. Check op de geldigheid van de barcode middels business rules
fact_asset_inventory	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_asset_inventory	asset_key	integer	fact	N/A	DWH	dim_asset	asset_key	integer	lookup op barcode
fact_asset_inventory	person_key	integer	fact	N/A	DWH	dim_person	person_key	integer	lookup op characteristic
fact_asset_inventory	status_key	integer	fact	N/A	DWH	dim_asset_status	status_key	integer	lookup op status code

Bijlage E - Data warehouse design

fact_asset_inventory	quantity_registered	integer	fact	N/A	Smartwork3	Asset	status	integer	Waarde '1' indien status 10, anders waarde '0'
dim_location	location_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_location	city	varchar (50)	dimension	0	Smartwork3	Location	city	varchar (50)	-
dim_location	zip	varchar (7)	dimension	0	Smartwork3	Location	zip	varchar (7)	-
dim_location	street	varchar (50)	dimension	0	Smartwork3	Location	street	varchar (50)	-
dim_location	house_number	integer	dimension	0	Smartwork3	Location	houseNumber	integer	-
dim_location	house_number_add	varchar (15)	dimension	0	Smartwork3	Location	houseNumberAdd	varchar (15)	-
dim_location	region	varchar (15)	dimension	1	Smartwork3	Location	zip	varchar (7)	Aan de hand van de business logica dient de regio bepaald te worden voor de postcode
dim_location	province	varchar (50)	dimension	1	Postcode	Postcode	province	varchar (50)	De provincie behorende bij de postcode dient uit de postcode database op de SmartWork server gehaald te worden. Waarde 'Onbekend' indien null
dim_location	country	varchar (50)	dimension	1	Postcode	Postcode	province	varchar (50)	Waarde 'Nederland' indien de provincie behorende bij de postcode bekend is, anders waarde 'Onbekend'.
dim_project	project_name	varchar (30)	dimension	0	Smartwork3	Project	name	varchar (255)	Check op lengte van naam.
dim_project	process_name	varchar (30)	dimension	1	Smartwork3	BaseProcessInstance	processname	varchar (255)	select bpi.processName from TaskDefault td, TaskGroup tg, Project p, BaseProcessInstance bpi where td.taskGroup_id = tg.id and tg.project_id = p.id and bpi.workItemId = td.id.
dim_workorder	workorder_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_workorder	external_reference	varchar (50)	dimension	0	Smartwork3	TaskDefault	external_ref_id	varchar (255)	select td.external_ref_id from ProcessState ps, BaseProcessInstance bpi, TaskDefault td where ps.processInstance_id = bpi.id and td.id = bpi.workItemId.

Bijlage E - Data warehouse design

dim_workorder	status	varchar (50)	dimension	2	Smartwork3	ProcessState	nodeName	varchar (255)	Check op lengte van node
dim_workorder	motive	varchar (50)	dimension	0	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from TaskDefault td, CustomField cf, CustomFieldValueString cfvs where td.id = cfv.workorder_id and cfv.customField_id = cf.id and cfvs.id = cfv.id
dim_workorder	meter_type	varchar (50)	dimension	1	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from TaskDefault td, CustomField cf, CustomFieldValueString cfvs where td.id = cfv.workorder_id and cfv.customField_id = cf.id and cfvs.id = cfv.id
fact_asset_placed	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_asset_placed	person_key	integer	fact	N/A	DWH	dim_person	person_key	integer	lookup op characteristic
fact_asset_placed	asset_key	integer	fact	N/A	DWH	dim_asset	asset_key	integer	lookup op barcode
fact_asset_placed	status_key	integer	fact	N/A	DWH	dim_asset_status	status_key	integer	lookup op status code
fact_asset_placed	location_key	integer	fact	N/A	DWH	dim_location	location_key	integer	lookup op zip en huisnummer
fact_asset_placed	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam
fact_asset_placed	workorder_key	integer	fact	N/A	DWH	dim_workorder	workorder_key	integer	lookup op external referentie
fact_asset_placed	quantity_registered	integer	fact	N/A	Smartwork3	Asset	status	integer	waarde '1' indien status 60, anders waarde '0'
fact_asset_placed	quantity_unregistered	integer	fact	N/A	Smartwork3	Asset	status	integer	waarde '1' indien status 70, anders waarde '0'
fact_productivity	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_productivity	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam
fact_productivity	workorder_key	integer	fact	N/A	DWH	dim_workorder	workorder_key	integer	lookup op external referentie
fact_productivity	person_key	integer	fact	N/A	DWH	dim_person	person_key	integer	lookup op characteristic

Bijlage E - Data warehouse design

fact_productivity	available_hours	double	fact	N/A	Smartwork3	T_REOUSRCE_WORK_DAYS, Resource_PeriodsOnDay, PeriodsOnDay	timeOnDayStart, TimeOnDayEnd	datetime	calculated fact. Sum van timediff pod.timeOnDayEnd – timeOnDayStart voor uur in minuten en minuten min timediff van de uitzondering tijden van exception dates
fact_productivity	realised_combi	integer	fact	N/A	Smartwork3	Workflow	name	varchar (100)	Calculated fact. Op workflow baseren of het combi betreft. Alle waardes op 1 dag per monteur groeperen / optellen
fact_productivity	realized_solo	integer	fact	N/A	Smartwork3	Workflow	name	varchar (100)	Calculated fact. Op workflow baseren of het combi betreft. Alle waardes van 1 dag per monteur groeperen / optellen
fact_productivity	tariff_removed	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waardes op 1 dag per monteur groeperen / optellen
fact_productivity	relay_removed	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waardes op 1 dag per monteur groeperen / optellen

Bijlage E - Data warehouse design

fact_productivity	meter_wire_replaced	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waardes op 1 dag per monteur groeperen / optellen
fact_productivity	board_wire_replaced	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waardes op 1 dag per monteur groeperen / optellen
fact_productivity	gas_bracked_placed	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waardes op 1 dag per monteur groeperen / optellen

Bijlage E - Data warehouse design

fact_productivity	s_coupling_placed	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waarden op 1 dag per monteur groeperen / optellen
fact_productivity	boiler_relay_placed	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waarden op 1 dag per monteur groeperen / optellen
fact_productivity	regulator_replaced	integer	fact	N/A	Smartwork3	CustomFieldValueBoolean	value	boolean	Calculated fact. Select cfvb.value from TaskDefault td, CustomField cf, CustomFieldValue cfv, CustomFieldValueBoolean cfvb where cfv.customField_id = cf.id and cfv.workorkorder_id = td.id and cfv.id = cfvb.id. Alle waarden op 1 dag per monteur groeperen / optellen
fact_timelines	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam
fact_timeliness	workorder_key	integer	fact	N/A	DWH	dim_workorder	workorder_key	integer	lookup op external referentie
fact_timeliness	person_key	integer	fact	N/A	DWH	dim_person	person_key	integer	lookup op characteristic
fact_timeliness	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_timeliness	time_key	integer	fact	N/A	DWH	dim_time	time_key	integer	lookup op uur en minuut

Bijlage E - Data warehouse design

fact_timeliness	time_window_start	time	fact	N/A	Smartwork3	ProcessVariable	dateTimeValue	datetime	select pv.dateTimeValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id
fact_timeliness	time_window_end	time	fact	N/A	Smartwork3	ProcessVariable	dateTimeValue	datetime	select pv.dateTimeValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id
fact_timeliness	minutes_early	integer	fact	N/A	Smartwork3	PeriodOnDay, ProcessVariable	TimeOnDayStart, dateTimeValue	datetime	Calculated fact. Sum timeOnDayStart uur in minuten en minuten min de starttijd van de werkzaamheden uit de dynamische procesvariable. Waarde '0' indien negatief
fact_timeliness	minutes_late	integer	fact	N/A	Smartwork3	PeriodOnDay, ProcessVariable	TimeOnDayEnd, dateTimeValue	datetime	Calculated fact. de eindtijd van de werkzaamheden uit de dynamische procesvariable min de sum timeOnDayStart uur in minuten en minuten. Waarde '0' indien negatief
dim_failure_reason	failure_key	integer	dimension	N/A	-	-	-	-	Surrogate key
dim_failure_reason	failure_type	varchar (3)	dimension	0	Smartwork3	ProcessVariable	stringValue	varchar (255)	select pv.stringValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id

Bijlage E - Data warehouse design

dim_failure_reason	reason	varchar (100)	fact	N/A	Smartwork3	ProcessVariable	stringValue	varchar (255)	select pv.stringValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id Splitten op \$ delimiter
fact_workorder_failure	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_workorder_failure	person_key	integer	fact	N/A	DWH	dim_person	person_key	integer	lookup op characteristic
fact_workorder_failure	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam
fact_workorder_failure	workorder_key	integer	fact	N/A	DWH	dim_workorder	workorder_key	integer	lookup op external referentie
fact_workorder_failure	location_key	integer	fact	N/A	DWH	dim_location	location_key	integer	lookup op zip en huisnummer
fact_workorder_failure	time_key	integer	fact	N/A	DWH	dim_time	time_key	integer	lookup op uur en minuut
fact_workorder_failure	failure_key	integer	fact	N/A	DWH	dim_time	time_key	integer	lookup op reason
fact_workorder_failure	comment	text	fact	N/A	Smartwork3	ProcessVariable	stringValue	varchar (255)	select pv.stringValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id Splitten op \$ delimiter
fact_project_progress	workorder_key	integer	fact	N/A	DWH	dim_workorder	workorder_key	integer	lookup op external referentie
fact_project_progress	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date
fact_project_progress	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam

Bijlage E - Data warehouse design

fact_project_progress	quantity_finished	integer	fact	N/A	Smartwork3	ProcessState	nodeName	varchar (255)	select td.external_ref_id from BaseProcessInstance bpi, ProcessState ps, TaskDefault td where td.id = bpi.workItemId and ps.processInstance_id = bpi.id. Waarde '1' indien het finished nodeName bevat
fact_project_progress	quantity_closed_door	integer	fact	N/A	Smartwork3	ProcessVariable	stringValue	varchar (255)	select pv.stringValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id Waarde '1' indien niet null
fact_project_progress	quantity_ant	integer	fact	N/A	Smartwork3	ProcessVariable	stringValue	varchar (255)	select pv.stringValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id Waarde '1' indien niet null
fact_project_progress	quantity_tnu	integer	fact	N/A	Smartwork3	ProcessVariable	stringValue	varchar (255)	select pv.stringValue from TaskDefault td, BaseProcessInstance bpi, ProcessState ps, ProcessVariable pv where bpi.workItemId = td.id and ps.processInstance_id = bpi.id and pv.processState_id = ps.id Waarde '1' indien niet null
fact_meter_forecast	date_key	integer	fact	N/A	DWH	dim_date	date_key	integer	lookup op date

Bijlage E - Data warehouse design

fact_meter_forecast	project_key	integer	fact	N/A	DWH	dim_project	project_key	integer	lookup op project naam
fact_meter_forecast	location_key	integer	fact	N/A	DWH	dim_location	location_key	integer	lookup op zip en huisnummer
fact_meter_forecast	quantity_1phase	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cf.id = cfvs.id. Waarde '1' indien 1-fase, anders waarde '0'
fact_meter_forecast	quantity_3phase	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien 3-fase, anders waarde '0'
fact_meter_forecast	quantity_g4	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G4, anders waarde '0'
fact_meter_forecast	quantity_g6	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G6, anders waarde '0'

Bijlage E - Data warehouse design

fact_meter_forecast	quantity_g10	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G10,
fact_meter_forecast	quantity_g16	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G16, anders waarde '0'
fact_meter_forecast	quantity_g25	integer	fact	N/A	Smartwork3	CustomFieldValueString	value	varchar (255)	select cfvs.value from CustomFieldValue cfv, CustomField cf, CustomFieldValueString cfvs where cf.id = cfv.customField_id and cfv.id = cfvs.id. Waarde '1' indien G25, anders waarde '0'

BIJLAGE F - ETL & SQL

SQL & ETL

SMARTINSTALL



INHOUDSOPGAVE

1	Jobs.....	5
2	Dimension Transformations	8
2.1	Asset dimension.....	9
2.2	Location dimension.....	10
2.3	Person dimension	11
2.4	Project dimension	12
2.5	workorder dimension	13
2.6	Failure reason dimension	14
3	Fact transformations.....	15
3.1	Asset inventory	16
3.2	Asset placed	17
3.3	Workorder failure	18
3.4	Productivity	22
3.5	Timeliness	25
3.6	Meter forecast	27

FIGURENLIJST

Figuur 1 - Dimension job	5
Figuur 2 - Fact job	5
Figuur 3 - Test job	6
Figuur 4 - Call all dimensions job	7
Figuur 5 - Extract person	8
Figuur 6 - Load person	8
Figuur 7 - Tranform asset.....	9
Figuur 8 - Tranform location	10
Figuur 9 - Transform person	11
Figuur 10 - Transform project.....	12
Figuur 11 - Transform workorder	13
Figuur 12 - Transform failure reason	14
Figuur 13 - Extract timeliness	15
Figuur 14 - Load timeliness	15
Figuur 15 - Transform asset inventory.....	16
Figuur 16 - transform asset placed	17
Figuur 17 - Transform workorder failure.....	18
Figuur 18 - transform productivity	22
Figuur 19 - Transform timeliness	25
Figuur 20 - Transform meter forecast	27

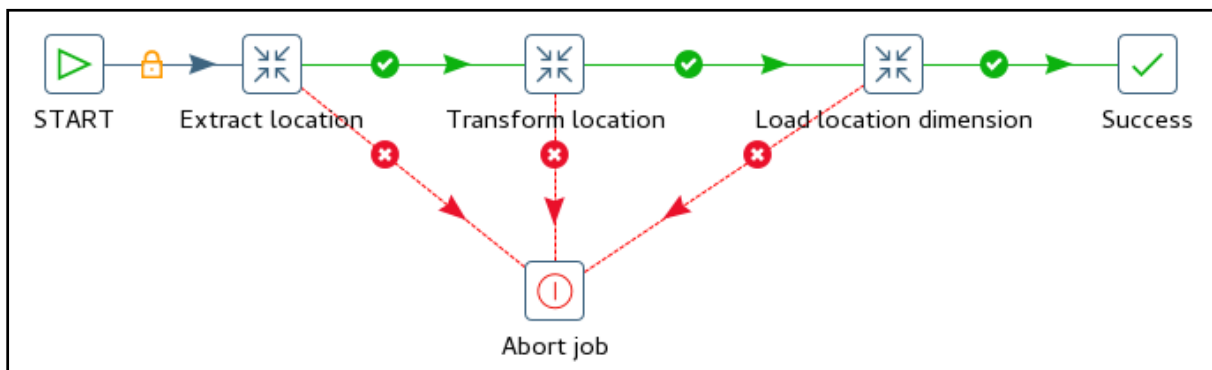
CODE SNIPPET LIJST

Code Snippet 1 - Asset SQL input	9
Code Snippet 2 - Location SQL input	10
Code Snippet 3 - Person SQL input	11
Code Snippet 4 - Project SQL input	12
Code Snippet 5 - Workorder G-meter SQL input	13
Code Snippet 6 - Workorder E-meter SQL input	13
Code Snippet 7 - Workorder SQL input	13
Code Snippet 8 - Workorder motive SQL input	13
Code Snippet 9 - Failure reason DD SQL input	14
Code Snippet 10 - Failure reason TNU SQL input	14
Code Snippet 11 - Failure reason ANT SQL input	14
Code Snippet 12 - Asset inventory SQL input	16
Code Snippet 13 - Asset placed SQL workorder input	17
Code Snippet 14 - Asset placed SQL asset input	17
Code Snippet 15 - Workorder failure SQL workorder data input	19
Code Snippet 16 - Workorder failure SQL DD input	20
Code Snippet 17 - Workorder failure SQL ANT input	20
Code Snippet 18 - Workorder failure SQL TNU input	21
Code Snippet 19 - Productivity SQL technician finished input	22
Code Snippet 20 - productivity SQL state input	23
Code Snippet 21 - Productivity SQL workorder ended input	23
Code Snippet 22 - Productivity SQL tariff removed input	24
Code Snippet 23 - Productivity SQL capacity exception input	24
Code Snippet 24 - Productivity SQL capacity input	24
Code Snippet 25 - Timeliness SQL input	26
Code Snippet 26 - Meter forecast SQL G-meter type	28
Code Snippet 27 - Meter forecast SQL E-meter type	28
Code Snippet 28 - Meter forecast SQL Workorder appointment input	28

1 JOBS

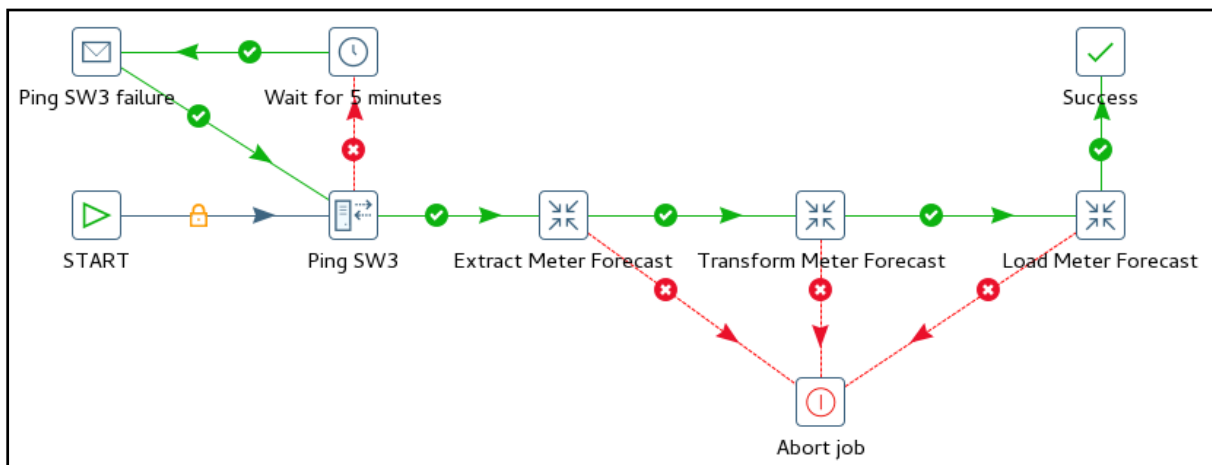
De weergegeven Kettle-jobs betreffen de overkoepelende jobs die de transformaties aanroepen. Voor zowel de dimensies, de feiten als de tests worden er in de overkoepelende jobs drie transformaties aangeroepen. De jobs voor alle feiten, dimensies en tests zijn bijna identiek en worden daarom niet allemaal weergegeven. Voor de feiten, dimensies en tests wordt er één overkoepelende job weergegeven.

Figuur 1 weergeeft de job voor het ETL-proces rondom location dimension. Hierin worden de extract, transform en load onderdelen aangeroepen. Alle dimension jobs zijn op dezelfde wijze gemaakt als weergegeven in Figuur 1.



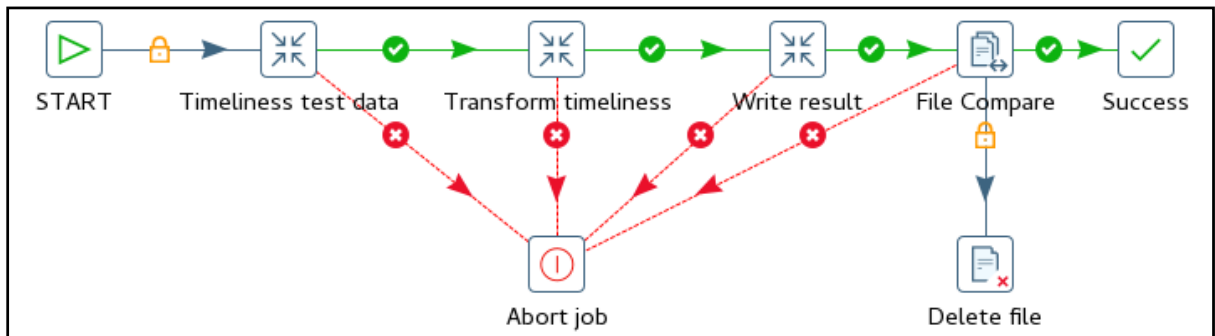
FIGUUR 1 - DIMENSION JOB

Figuur 2 weergeeft de job voor het ETL-proces rondom meter forecast. Het verschil ten opzichte van de dimension jobs, is dat er in de fact jobs gecontroleerd wordt of de SmartWork 3.0 omgeving beschikbaar is. Indien de SmartWork3.0 omgeving niet beschikbaar is, wacht het ETL-proces vijf minuten en stuurt het een e-mail naar de beheerder alvorens het nogmaals proberen. Alle fact jobs zijn op dezelfde wijze gemaakt als weergegeven in Figuur 2.



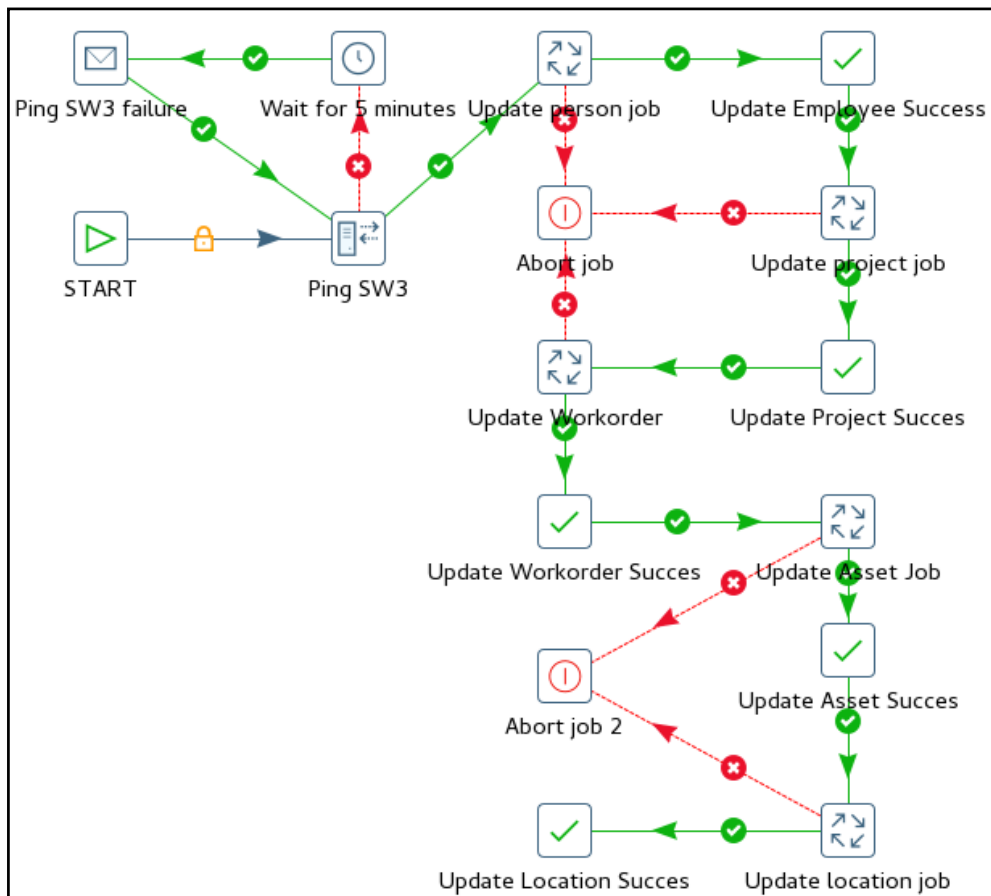
FIGUUR 2 - FACT JOB

Figuur 3 weergeeft de test job voor het ETL-proces rondom timeliness. De extract en load onderdelen zijn vervangen door het inlezen van testdata uit een CSV-bestand en het wegschrijven van het resultaat naar een CSV-bestand. Het CSV-bestand met de resultaten wordt vervolgens vergeleken met een eerder opgesteld CSV-bestand waarin het verwachte resultaat staat. Indien de data van beide bestanden overeenkomen, slaagt de test. Als ze niet overeenkomen faalt de test. Alle test jobs voor zowel het ETL-proces van dimension als het ETL-proces van facts zijn op dezelfde wijze gemaakt als weergegeven in Figuur 3.



FIGUUR 3 - TEST JOB

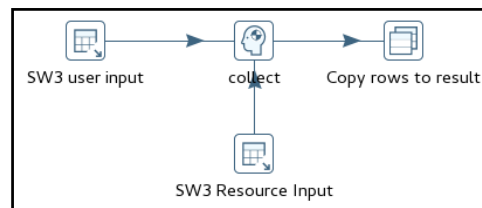
Figuur 4 weergeeft de job die alle dimension jobs als weergegeven in Figuur 1 aanroept. Alvorens het aanroepen van de dimension jobs, wordt er wederom gecontroleerd of de SmartWork 3.0 omgeving beschikbaar is. Gezien de beschikbaarheid gecontroleerd wordt in de job van Figuur 4, wordt de beschikbaarheid niet nogmaals gecontroleerd in de dimension jobs.



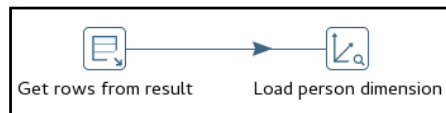
FIGUUR 4 - CALL ALL DIMENSIONS JOB

2 DIMENSION TRANSFORMATIONS

In Figuur 1 van hoofdstuk 1 is weergegeven dat een ETL-proces rondom een dimension bestaat uit extract, transform en load onderdelen. Per dimension wordt het ETL-proces van de transformatie en de SQL code van de extract stap weergegeven. De extract en load onderdelen van alle dimensions zijn vergelijkbaar en worden daarom niet allemaal weergegeven. In Figuur 5 is de extract stap van het ETL-proces rondom person dimension weergegeven. Hier wordt alle benodigde data opgehaald uit de sourcedatabase. Figuur 6 weergeeft de load stap van het person dimension ETL proces. De extract en load stappen van alle overige dimensions zijn op soortgelijke wijze gedaan.

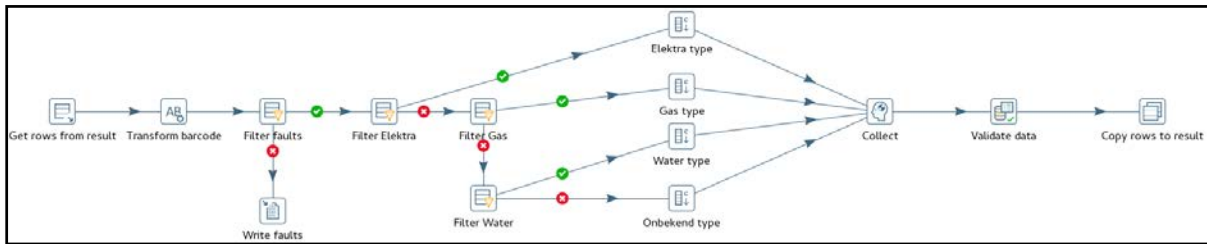


FIGUUR 5 - EXTRACT PERSON



FIGUUR 6 - LOAD PERSON

2.1 ASSET DIMENSION

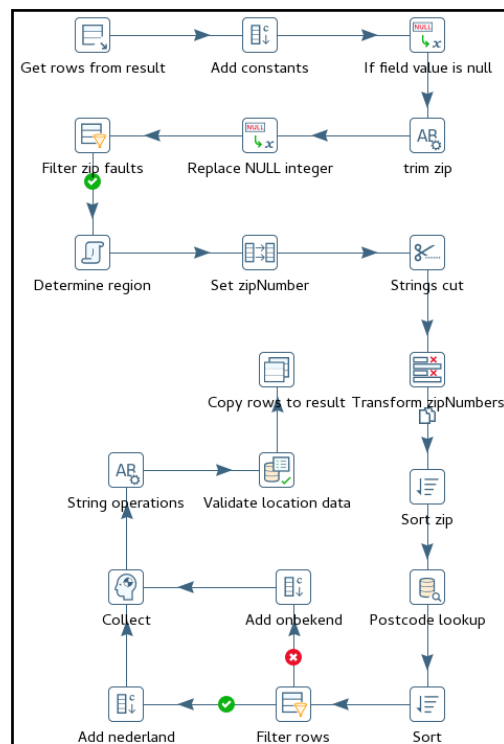


FIGUUR 7 - TRANSFORM ASSET

```
SELECT a.barcode
FROM Asset a;
```

CODE SNIPPET 1 - ASSET SQL INPUT

2.2 LOCATION DIMENSION



FIGUUR 8 - TRANFORM LOCATION

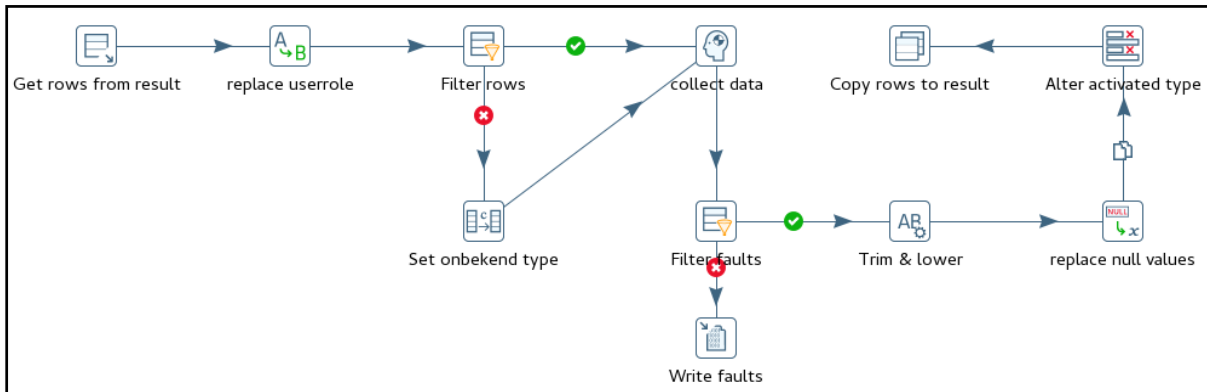
```

SELECT
    l.city,
    l.street,
    l.zip,
    l.houseNumber,
    l.houseNumberAdd
FROM Location l
GROUP BY l.zip, l.houseNumber, l.houseNumberAdd
ORDER BY l.zip;

```

CODE SNIPPET 2 - LOCATION SQL INPUT

2.3 PERSON DIMENSION



FIGUUR 9 - TRANSFORM PERSON

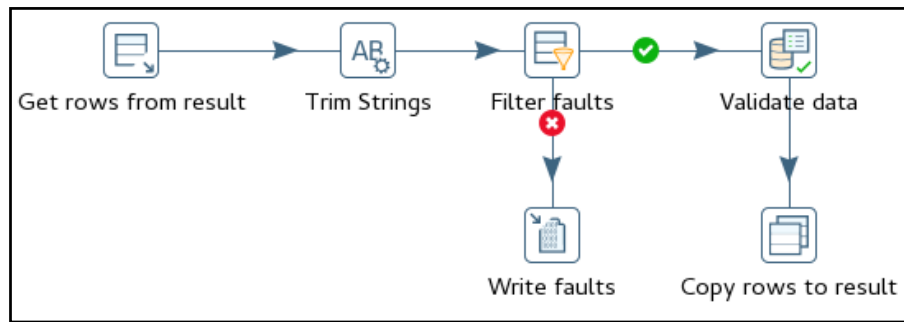
```

SELECT
  u.id AS user_id,
  u.first_name,
  u.last_name,
  u.phone,
  u.mobile,
  u.email,
  u.login,
  u.userRole_roleName,
  u.activated,
  c.name FROM User u
INNER JOIN Resource r ON u.id = r.user_id
INNER JOIN Company c ON r.company_id = c.id
WHERE NOT EXISTS
(
  SELECT *
  FROM User us
  WHERE us.id = u.id
  AND us.userRole_roleName = 'ROLE_EMPLOYEE'
)
AND NOT EXISTS
(
  SELECT *
  FROM User us
  WHERE us.id = u.id
  AND us.userRole_roleName = 'ROLE_CUSTOMER'
)
ORDER BY user_id ASC;

```

CODE SNIPPET 3 - PERSON SQL INPUT

2.4 PROJECT DIMENSION



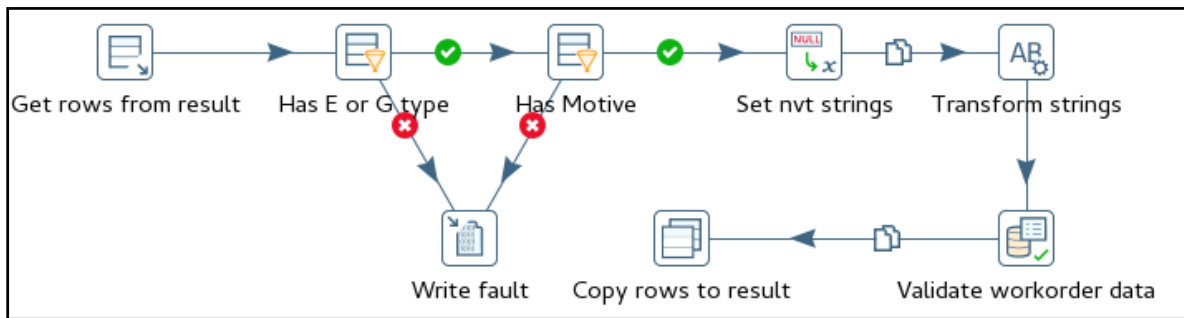
FIGUUR 10 - TRANSFORM PROJECT

```

SELECT
    p.name,
    p.processName
FROM Project p;
    
```

CODE SNIPPET 4 - PROJECT SQL INPUT

2.5 WORKORDER DIMENSION



FIGUUR 11 - TRANSFORM WORKORDER

```

SELECT
    td.external_ref_id,
    td.status
FROM TaskDefault td
ORDER BY td.external_ref_id ASC;

```

CODE SNIPPET 7 - WORKORDER SQL INPUT

```

SELECT
    td.external_ref_id,
    cfvs.value AS motive
FROM TaskDefault td
INNER JOIN CustomFieldValue cfv ON td.id = cfv.workorder_id
INNER JOIN CustomField cf ON cfv.customField_id = cf.id
INNER JOIN CustomFieldValueString cfvs ON cfv.id = cfvs.id
WHERE cf.fieldName = 'Aanleiding'
ORDER BY td.external_ref_id;

```

CODE SNIPPET 8 - WORKORDER MOTIVE SQL INPUT

```

SELECT
    td.external_ref_id,
    cfvs.value AS e_meter_type
FROM TaskDefault td
INNER JOIN CustomFieldValue cfv ON td.id = cfv.workorder_id
INNER JOIN CustomField cf ON cfv.customField_id = cf.id
INNER JOIN CustomFieldValueString cfvs ON cfv.id = cfvs.id
WHERE cf.fieldName = 'MetertypeEmeter'
ORDER BY td.external_ref_id ASC;

```

CODE SNIPPET 6 - WORKORDER E-METER SQL INPUT

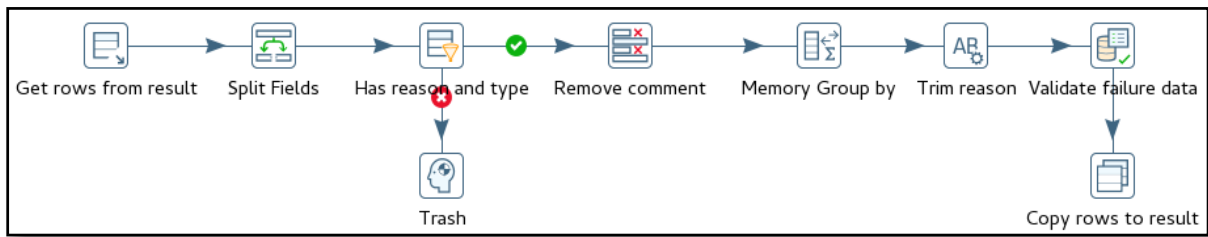
```

SELECT
    td.external_ref_id,
    cfvs.value AS e_meter_type
FROM TaskDefault td
INNER JOIN CustomFieldValue cfv ON td.id = cfv.workorder_id
INNER JOIN CustomField cf ON cfv.customField_id = cf.id
INNER JOIN CustomFieldValueString cfvs ON cfv.id = cfvs.id
WHERE cf.fieldName = 'MetertypeGmeter'
ORDER BY td.external_ref_id ASC;

```

CODE SNIPPET 5 - WORKORDER G-METER SQL INPUT

2.6 FAILURE REASON DIMENSION



FIGUUR 12 - TRANSFORM FAILURE REASON

```

SELECT pv.stringValue as reasoncomment
FROM ProcessVariable pv
WHERE NOT EXISTS
(
  SELECT *
  FROM ProcessVariable pvs
  WHERE pv.id = pvs.id
  AND pvs.variableName = 'REDEN_NIET_TOEGANKELIJK'
  AND pvs.stringValue IS NULL
)
AND pv.variableName = 'REDEN_NIET_TOEGANKELIJK';

```

CODE SNIPPET 11 - FAILURE REASON ANT SQL INPUT

```

SELECT pv.stringValue as reasoncomment
FROM ProcessVariable pv
WHERE NOT EXISTS
(
  SELECT *
  FROM ProcessVariable pvs
  WHERE pv.id = pvs.id
  AND pvs.variableName = 'REDEN_NIET_UITVOERBAAR'
  AND pvs.stringValue IS NULL
)
AND pv.variableName = 'REDEN_NIET_UITVOERBAAR';

```

CODE SNIPPET 10 - FAILURE REASON TNU SQL INPUT

```

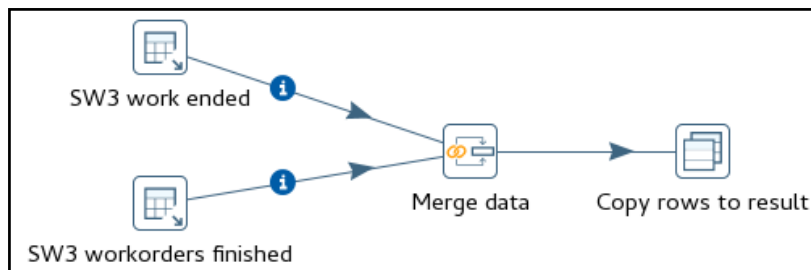
SELECT pv.stringValue as reasoncomment
FROM ProcessVariable pv
WHERE NOT EXISTS
(
  SELECT *
  FROM ProcessVariable pvs
  WHERE pv.id = pvs.id
  AND pvs.variableName = 'REDEN_NIET_REALISEERBAAR'
  AND pvs.stringValue IS NULL
)
AND pv.variableName = 'REDEN_NIET_REALISEERBAAR';

```

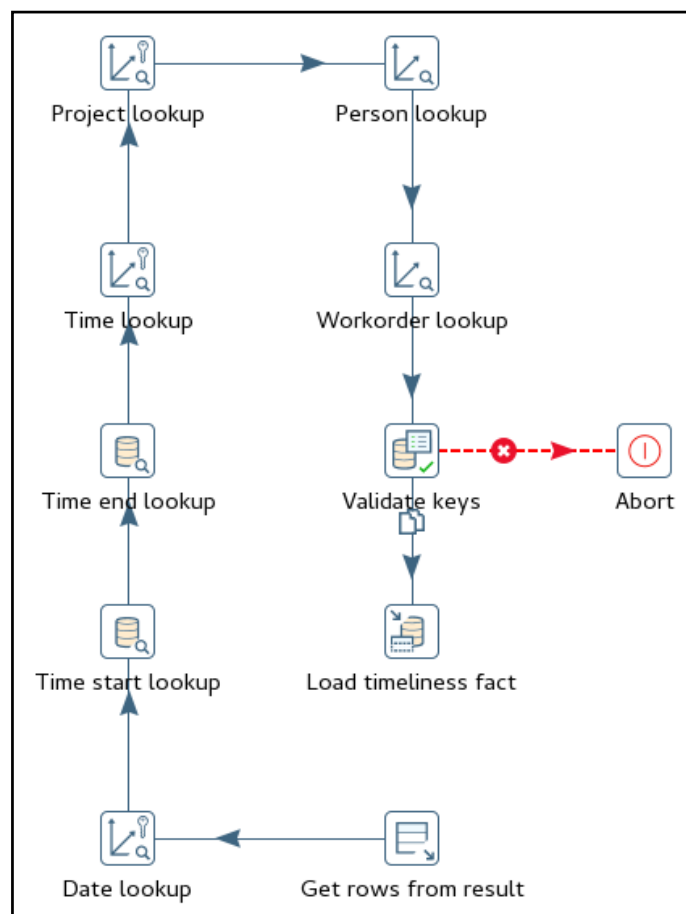
CODE SNIPPET 9 - FAILURE REASON DD SQL INPUT

3 FACT TRANSFORMATIONS

In Figuur 2 van hoofdstuk 1 is weergegeven dat een ETL-proces rondom een fact wederom bestaat uit extract, transform en load onderdelen. Per fact wordt het ETL-proces van de transformatie en de SQL code van de extract stap weergegeven. De extract en load onderdelen van alle facts zijn vergelijkbaar en worden daarom niet allemaal weergegeven. In Figuur 13 is de extract stap van het ETL proces rondom timeliness fact weergegeven. Hier wordt weer alle benodigde data opgehaald uit de sourcedatabase. Figuur 14 weergeeft de load stap van het timeliness fact ETL proces. De lookup componenten halen de surrogate keys op van betreffende dimensions. De extract en load stappen van alle overige dimensions zijn op soortgelijke wijze gedaan.

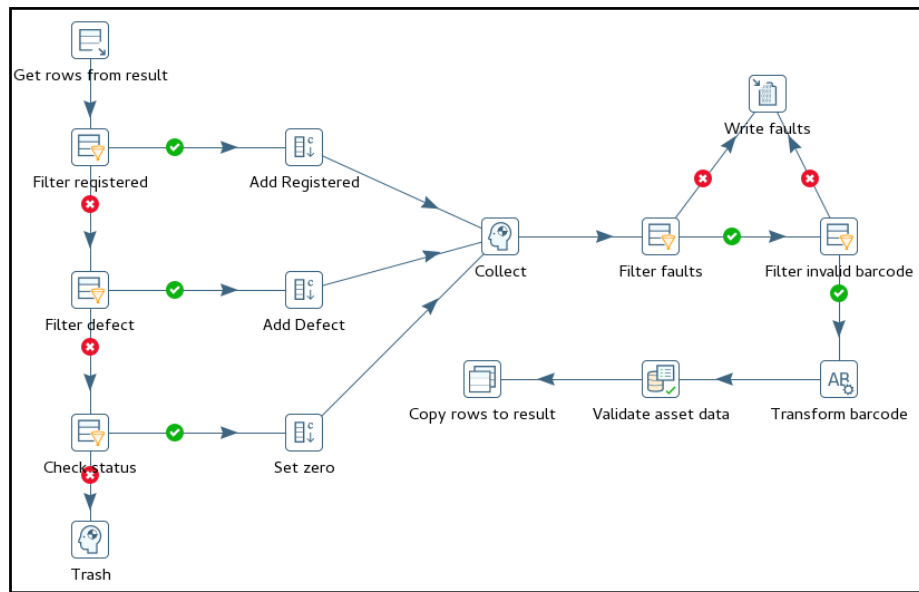


FIGUUR 13 - EXTRACT TIMELINESS



FIGUUR 14 - LOAD TIMELINESS

3.1 ASSET INVENTORY



FIGUUR 15 - TRANSFORM ASSET INVENTORY

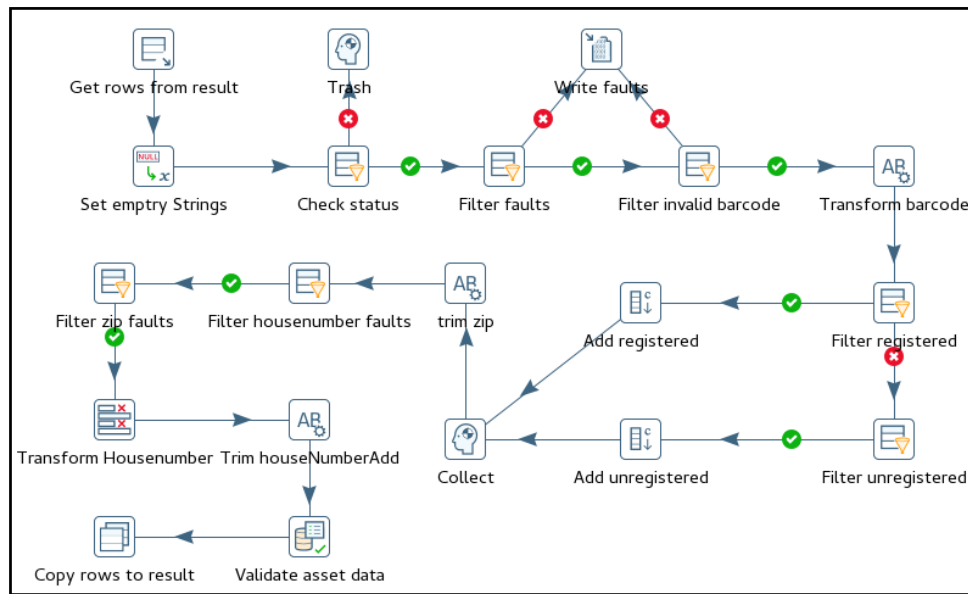
```

SELECT
    ast.status,
    ast.timestamp,
    ast.asset_id,
    ast.user_id,
    a.barcode,
    u.login
FROM AssetStatus ast
INNER JOIN Asset a ON ast.asset_id = a.id
INNER JOIN User u ON ast.user_id = u.id;

```

CODE SNIPPET 12 - ASSET INVENTORY SQL INPUT

3.2 ASSET PLACED



FIGUUR 16 - TRANSFORM ASSET PLACED

```

SELECT
    a.barcode,
    a.changedDateTime,
    a.status,
    a.userOfLastStatus_id,
    a.workorder_id,
    u.login
FROM Asset a
INNER JOIN User u ON a.userOfLastStatus_id = u.id
WHERE a.status > 50
ORDER BY a.workorder_id;

```

CODE SNIPPET 14 - ASSET PLACED SQL ASSET INPUT

```

SELECT
    td.id AS taskdefaultid,
    td.external_ref_id,
    l.zip,
    l.houseNumber,
    l.houseNumberAdd,
    p.name AS projectname,
    p.processName
FROM smartwork3.TaskDefault td
INNER JOIN Location l ON td.location_id = l.id
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
ORDER BY td.id;

```

CODE SNIPPET 13 - ASSET PLACED SQL WORKORDER INPUT


```

SELECT
    td.external_ref_id,
    td.status, ps.processInstance_id,
    MAX(ps.stateLocalDateTime) AS curDateTime,
    DATE_FORMAT(MAX(ps.stateLocalDateTime), '%Y-%m-%d') dateOnly,
    DATE_FORMAT(MAX(ps.stateLocalDateTime), '%H:%i:%s') timeOnly,
    DATE_FORMAT(MAX(ps.stateLocalDateTime), '%H') hourOnly,
    DATE_FORMAT(MAX(ps.stateLocalDateTime), '%i') minuteOnly,
    p.name AS projectName,
    p.processName,
    pv.stringValue AS login,
    l.zip,
    l.houseNumber,
    l.houseNumberAdd
FROM TaskDefault td
INNER JOIN Location l ON td.location_id = l.id
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
INNER JOIN BaseProcessInstance bpi ON td.id = bpi.workItemId
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN
(
    SELECT tdd.id AS workId, MAX(pss.stateLocalDateTime) AS latest
    FROM TaskDefault tdd
    INNER JOIN BaseProcessInstance bpi ON tdd.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'MONTEUR_UITGEVOERD'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest
AND td.id = LatestVariable.workId
WHERE pv.variableName = 'MONTEUR_UITGEVOERD'
AND pv.stringValue IS NOT NULL
ORDER BY td.external_ref_id;

```

CODE SNIPPET 15 - WORKORDER FAILURE SQL WORKORDER DATA INPUT

```

SELECT
    td.external_ref_id,
    pv.stringValue as ant
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON bpi.workItemId = td.id
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN
(
    SELECT tdd.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tdd
    INNER JOIN BaseProcessInstance bpi ON tdd.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'REDEN_NIET_TOEGANKELIJK'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id =
LatestVariable.workId
WHERE pv.variableName = 'REDEN_NIET_TOEGANKELIJK'
ORDER BY td.external_ref_id;

```

CODE SNIPPET 17 - WORKORDER FAILURE SQL ANT INPUT

```

SELECT
    td.external_ref_id,
    pv.stringValue as dd
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON bpi.workItemId = td.id
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN
(
    SELECT tdd.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tdd
    INNER JOIN BaseProcessInstance bpi ON tdd.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'REDEN_NIET_REALISEERBAAR'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id =
LatestVariable.workId
WHERE pv.variableName = 'REDEN_NIET_REALISEERBAAR'
ORDER BY td.external_ref_id;

```

CODE SNIPPET 16 - WORKORDER FAILURE SQL DD INPUT

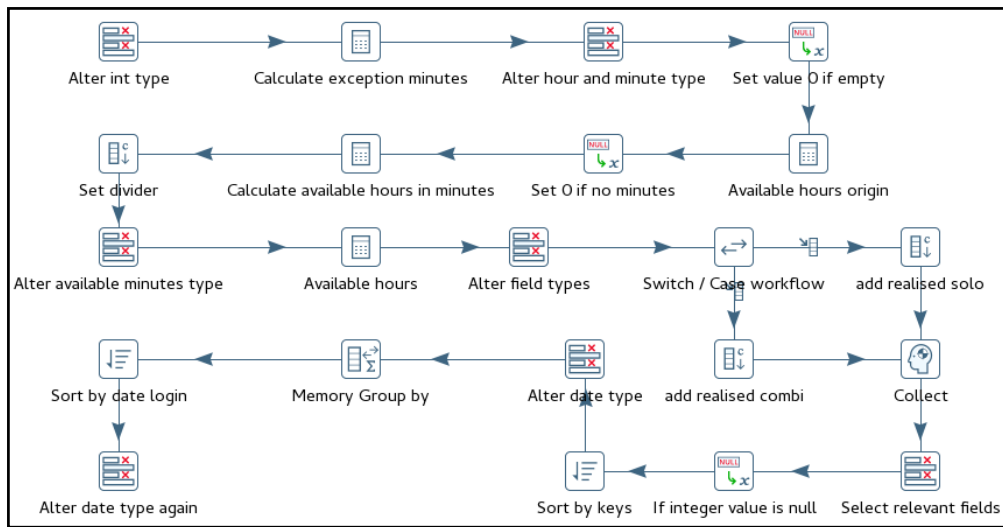

```

SELECT
    td.external_ref_id,
    pv.stringValue as tnu
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON bpi.workItemId = td.id
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN
(
    SELECT tdd.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tdd
    INNER JOIN BaseProcessInstance bpi ON tdd.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'REDEN_NIET_UITVOERBAAR'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id =
LatestVariable.workId
WHERE pv.variableName = 'REDEN_NIET_UITVOERBAAR'
ORDER BY td.external_ref_id;

```

CODE SNIPPET 18 - WORKORDER FAILURE SQL TNU INPUT

3.4 PRODUCTIVITY



FIGUUR 18 - TRANSFORM PRODUCTIVITY

```

SELECT
    td.external_ref_id,
    w.name AS workflow,
    pv.stringValue AS login,
    p.name AS project_name,
    p.processName
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON td.id = bpi.workItemId
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN Workflow w ON td.workflow_id = w.id
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
INNER JOIN
(
    SELECT tds.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tds
    INNER JOIN BaseProcessInstance bpi ON tds.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'MONTEUR_UITGEVOERD'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id =
LatestVariable.workId
WHERE EXISTS
(
    SELECT * FROM ProcessState ps
    WHERE bpi.id = ps.processInstance_id
    AND nodeName = 'Opleveren'
)
AND pv.variableName = 'MONTEUR_UITGEVOERD'
ORDER BY external_ref_id ASC;

```

CODE SNIPPET 19 - PRODUCTIVITY SQL TECHNICIAN FINISHED INPUT

```

SELECT
    td.external_ref_id,
    w.name AS workflow,
    pv.dateTimeValue AS stateLocalDateTime,
    p.name AS project_name,
    p.processName
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON td.id = bpi.workItemId
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN Workflow w ON td.workflow_id = w.id
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
INNER JOIN
(
    SELECT tds.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tds
    INNER JOIN BaseProcessInstance bpi ON tds.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstance_id
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'WERKZAAMHEDEN_EIND'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id =
LatestVariable.workId
WHERE EXISTS
(
    SELECT * FROM ProcessState ps
    WHERE bpi.id = ps.processInstance_id
    AND nodeName = 'Opleveren'
)
AND pv.variableName = 'WERKZAAMHEDEN_EIND'
ORDER BY external_ref_id ASC;

```

CODE SNIPPET 21 - PRODUCTIVITY SQL WORKORDER ENDED INPUT

```

SELECT
    td.external_ref_id,
    ps.stateLocalDateTime as stateDate,
    CASE WEEKDAY(ps.stateLocalDateTime) + 1 WHEN 7 THEN 0 ELSE
WEEKDAY(ps.stateLocalDateTime) + 1 END AS dayOfWeek
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON td.id = bpi.workItemId
INNER JOIN ProcessState ps ON bpi.id = ps.processInstance_id
AND ps.nodeName = 'Opleveren'
ORDER BY external_ref_id ASC;

```

CODE SNIPPET 20 - PRODUCTIVITY SQL STATE INPUT

```

SELECT
    rwd.resource_id,
    rwd.periods_index AS dayOfWeek,
    pod.timeOnDayStart AS hourStart,
    pod.timeOnDayEnd AS hourEnd,
    SUM(DATE_FORMAT(TIMEDIFF(pod.timeOnDayEnd, pod.timeOnDayStart), '%H'))
hoursDiff,
    SUM(DATE_FORMAT(TIMEDIFF(pod.timeOnDayEnd, pod.timeOnDayStart), '%i'))
minuteDiff,
    SUM(DATE_FORMAT(TIMEDIFF(pod.timeOnDayEnd, pod.timeOnDayStart), '%H'))
* 60 hoursDiffInMinutes
FROM T_RESOURCE_WORK_DAYS rwd
INNER JOIN PeriodsOnDay_PeriodOnDay pod_pod ON rwd.PeriodsOnDay_id =
pod_pod.PeriodsOnDay_id
INNER JOIN PeriodOnDay pod ON pod_pod.periodOnDays_id = pod.id
GROUP BY resource_id, dayOfWeek
ORDER BY resource_id, dayOfWeek;

```

CODE SNIPPET 24 - PRODUCTIVITY SQL CAPACITY INPUT

```

SELECT
    r_pod.Resource_id,
    r_pod.exceptions_KEY AS exception_date,
    pod_ex.timeOnDayStart AS exceptionHourStart,
    pod_ex.timeOnDayEnd AS exceptionHourEnd,
    DATE_FORMAT(TIMEDIFF(pod_ex.timeOnDayEnd, pod_ex.timeOnDayStart),
'%H') hoursDiffException,
    DATE_FORMAT(TIMEDIFF(pod_ex.timeOnDayEnd, pod_ex.timeOnDayStart),
'%i') minuteDiffException,
    DATE_FORMAT(TIMEDIFF(pod_ex.timeOnDayEnd, pod_ex.timeOnDayStart),
'%H') * 60 hoursDiffExceptionInMinutes
FROM Resource_PeriodsOnDay r_pod
INNER JOIN PeriodsOnDay_PeriodOnDay pod_pod_ex
ON r_pod.exceptions_id = pod_pod_ex.periodsOnDay_id
INNER JOIN PeriodOnDay pod_ex ON pod_pod_ex.periodOnDays_id = pod_ex.id
ORDER BY r_pod.Resource_id, exception_date;

```

CODE SNIPPET 23 - PRODUCTIVITY SQL CAPACITY EXCEPTION INPUT

```

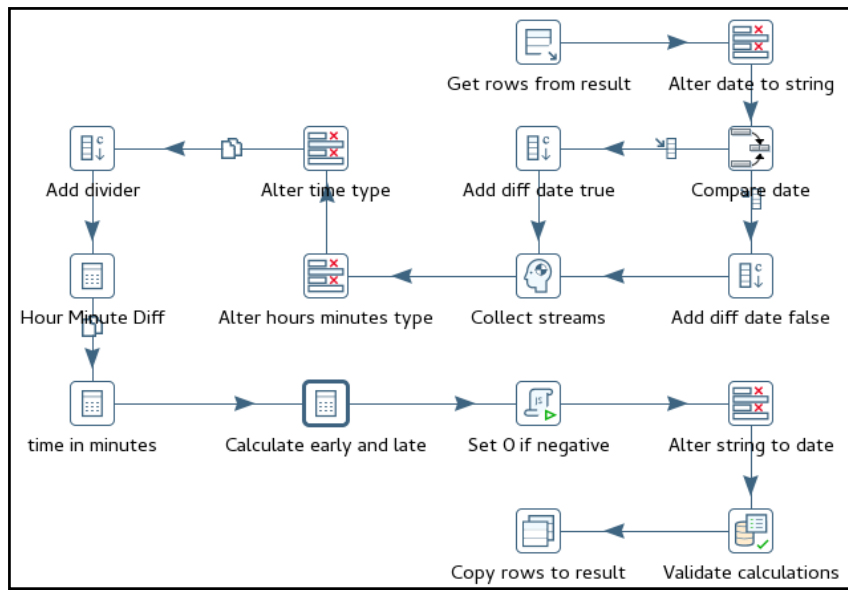
SELECT
    td.external_ref_id,
    cf.fieldName,
    cfvb.value AS tariff_removed
FROM CustomFieldValue cfv
INNER JOIN CustomField cf ON cfv.customField_id = cf.id
INNER JOIN TaskDefault td ON cfv.workorder_id = td.id
INNER JOIN CustomFieldValueBoolean cfvb ON cfv.id = cfvb.id
WHERE cf.fieldName = 'VerwijderenAfdoppentariefdraad'
ORDER BY td.external_ref_id;

-- Zelfde voor overige meerwerken met fieldNames:
-- VerwijderenTFrelais, VervangenMeterbedradingHoofdzekeringElektrameter,
-- VervangenMeterbedradingEmeterGroepenkast,
-- NietGasvoerendeMeterbeugelGeplaatst,
-- AfwijkendeSteekmaatKoppelingGeplaatst, BoilerRelaisGeplaatst,
-- GasdrukregelaarGewisseld

```

CODE SNIPPET 22 - PRODUCTIVITY SQL TARIFF REMOVED INPUT

3.5 TIMELINESS



FIGUUR 19 - TRANSFORM TIMELINESS

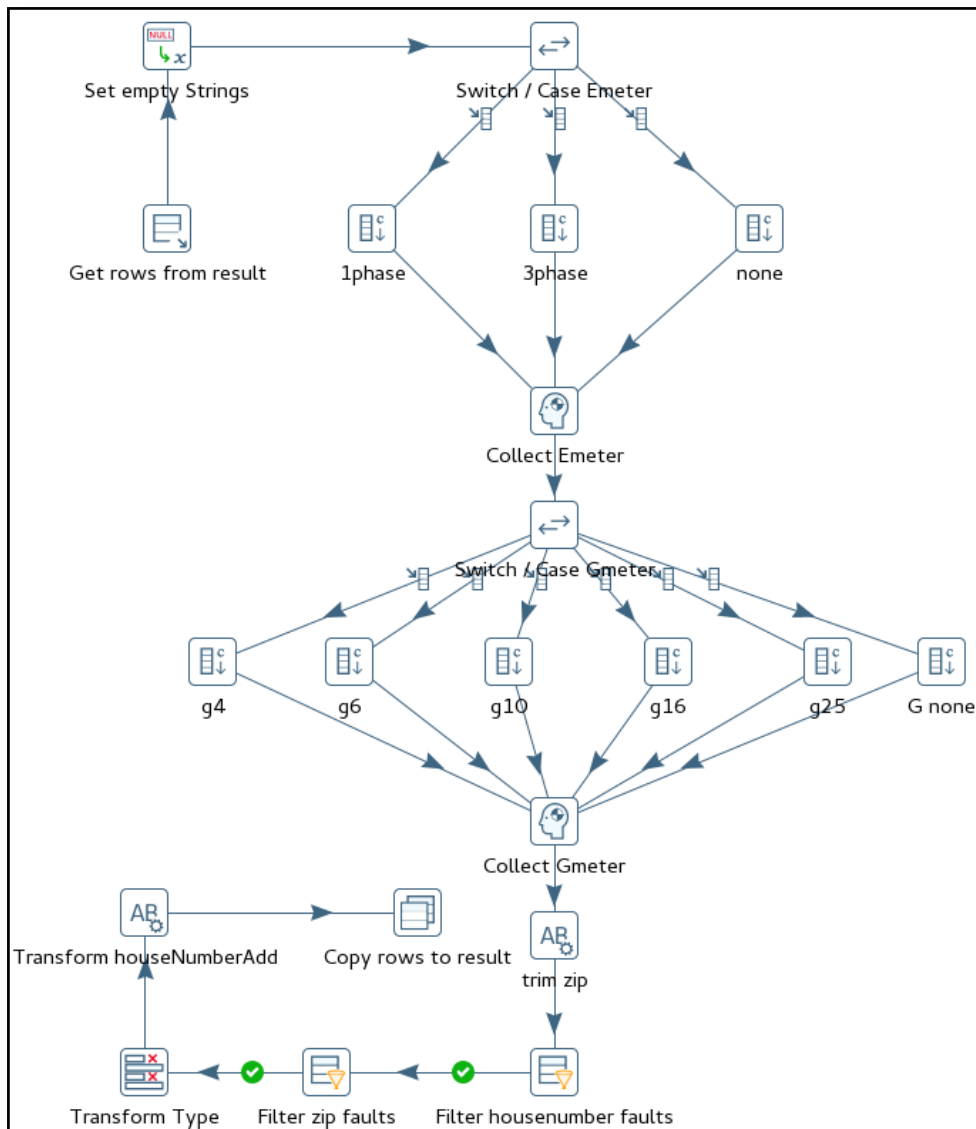
```

SELECT
    td.external_ref_id,
    podate.date AS datePlanned,
    poday.timeOnDayStart,
    poday.timeOnDayEnd,
    pv.dateTimeValue AS dateWorked,
    p.name AS project_name,
    p.processName,
    DATE_FORMAT(poday.timeOnDayStart, '%H') hourStart,
    DATE_FORMAT(poday.timeOnDayStart, '%i') minuteStart,
    DATE_FORMAT(poday.timeOnDayEnd, '%H') hourEnd,
    DATE_FORMAT(poday.timeOnDayEnd, '%i') minuteEnd,
    DATE_FORMAT(pv.dateTimeValue, '%H') hourStartWork,
    DATE_FORMAT(pv.dateTimeValue, '%i') minuteStartWork
FROM TaskDefault td
INNER JOIN BaseProcessInstance bpi ON td.id = bpi.workItemId
INNER JOIN ProcessState ps ON bpi.id = ps.processInstanceId
INNER JOIN ProcessVariable pv ON ps.id = pv.processState_id
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
INNER JOIN PeriodOnDate podate ON td.timeWindowAppointment_id = podate.id
INNER JOIN PeriodOnDay poday ON podate.periodOnDay_id = poday.id
INNER JOIN
(
    SELECT tds.id as workId, MAX(pss.stateLocalDateTime) as latest
    FROM TaskDefault tds
    INNER JOIN BaseProcessInstance bpi ON tds.id = bpi.workItemId
    INNER JOIN ProcessState pss ON bpi.id = pss.processInstanceId
    INNER JOIN ProcessVariable pvs ON pss.id = pvs.processState_id
    WHERE pvs.variableName = 'WERKZAAMHEDEN_BEGIN'
    GROUP BY workId
) AS LatestVariable
ON ps.stateLocalDateTime = LatestVariable.latest AND td.id =
LatestVariable.workId
AND pv.variableName = 'WERKZAAMHEDEN_BEGIN'
AND pv.dateTimeValue IS NOT NULL
ORDER BY external_ref_id ASC;

```

CODE SNIPPET 25 - TIMELINESS SQL INPUT

3.6 METER FORECAST



FIGUUR 20 - TRANSFORM METER FORECAST

```

SELECT
    td.external_ref_id,
    podate.date AS datePlanned,
    p.name AS project_name,
    p.processName,
    l.zip,
    l.houseNumber,
    l.houseNumberAdd
FROM TaskDefault td
INNER JOIN TaskGroup tg ON td.taskGroup_id = tg.id
INNER JOIN Project p ON tg.project_id = p.id
INNER JOIN PeriodOnDate podate ON td.timeWindowAppointment_id = podate.id
INNER JOIN Location l ON td.location_id = l.id
ORDER BY td.external_ref_id;

```

CODE SNIPPET 28 - METER FORECAST SQL WORKORDER APPOINTMENT INPUT

```

SELECT
    td.external_ref_id,
    cf.fieldName,
    cfvs.value AS Emeter
FROM CustomFieldValue cfv
INNER JOIN CustomField cf ON cfv.customField_id = cf.id
INNER JOIN TaskDefault td ON cfv.workorder_id = td.id
INNER JOIN CustomFieldValueString cfvs ON cfv.id = cfvs.id
WHERE cf.fieldName = 'MetertypeEMeter'
ORDER BY td.external_ref_id;

```

CODE SNIPPET 27 - METER FORECAST SQL E-METER TYPE

```

SELECT
    td.external_ref_id,
    cf.fieldName,
    cfvs.value AS Emeter
FROM CustomFieldValue cfv
INNER JOIN CustomField cf ON cfv.customField_id = cf.id
INNER JOIN TaskDefault td ON cfv.workorder_id = td.id
INNER JOIN CustomFieldValueString cfvs ON cfv.id = cfvs.id
WHERE cf.fieldName = 'MetertypeGMeter'
ORDER BY td.external_ref_id;

```

CODE SNIPPET 26 - METER FORECAST SQL G-METER TYPE

BIJLAGE G – TUSSENTIJDSE ASSESSMENT

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	------------------------	--------------------

Formulier tussentijds assessment

Student: Jan Bongers

Studentnummer: 09057633

Datum: 12 mei 2016

eerste / tweede TTA: Middendorp / Bosch

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	<i>Het voortgangsverslag is ontvangen</i>	x	
b	<i>Het afstudeerdossier is digitaal beschikbaar</i>	x	
c	<i>Het afstudeerdossier is opgebouwd conform de richtlijnen</i>	x	
d	<i>Het goedgekeurde afstudeerplan is aanwezig</i>	x	
e	<i>Het plan van aanpak is aanwezig</i>	x	
f	<i>Reeds geleverd commentaar is aanwezig</i>		x
g	<i>Het afstudeerdossier geeft voldoende inzicht in de stand van zaken</i>		x
h	<i>De afstudeeropdracht is tot nu toe naar behoren uitgevoerd</i>		x

Aanpak	O	T	V	G
<i>Passend</i>			x	
<i>Theoretisch verantwoord</i>		x		
<i>Samenhang uitvoering beroepstaken</i>			x	

Beroepstaken op afgesproken niveau uitgevoerd?		O	T	V	G
1	Selectie methode		x		
2	Selectie software			x	
3	Probleem analyse			x	
4	Database ontwerp			x	

Bijlage G – Tussentijds assessment

5	Database bouwen			x	
6	Applicatie bouwen			x	
7	Beheer configuratie		x		
8					

Producten	O	T	V	G
<i>Tussenproducten</i>			x	
<i>Eindproducten</i>	x			

Effectief communiceren	O	T	V	G
<i>Binnen afstudeerbedrijf</i>				
<i>Afstudeerdossier</i>		x		

Reflectie	O	T	V	G
<i>Inzicht in eigen functioneren</i>			x	
<i>Inzicht in eigen leerproces</i>		x		

Toelichting per beoordelingscriterium

Aanpak
Theorie eigen gemaakt en adequaat toegepast. Bruikbare software geselecteerd en werkend prototype opgeleverd.

Beroepstaken op afgesproken niveau uitgevoerd?

Selectie van ontwikkelmethode onvoldoende onderbouwd. Probleem analyse, ontwerp, en bouwen is voldoende maar moet beter beschreven worden, testen ontbreekt vooralsnog.

Producten

Rapportage onvolledig en software producten zijn niet bijgevoegd (code).
--

Effectief communiceren

Beschrijven van probleem, analyse, en ontwerp keuzes kan beter.

Reflectie

Toont inzicht in eigen functioneren maar ziet tekortkomingen over het hoofd.
--

Advies

Aankruisen welk advies de student heeft gekregen

☒	Inleveren (bindend advies)
☐	Verlengen (vrijblijvend advies)
☐	Stoppen (vrijblijvend advies)

Besluit student

Aankruisen welke beslissing de student heeft genomen (alleen na vrijblijvend advies)

☐	Afstudeerdossier wordt op afgesproken datum ingeleverd	Inleverdatum: 3 juni 2016
☐	Afstudeerperiode wordt verlengd	Inleverdatum:
☐	Student stopt met afstudeeropdracht	

Naam begeleidend examiner: Henk Middendorp

Naam tweede examiner: Henk van den Bosch

Datum: 18 mei 2016

BIJLAGE H – EVALUATIE SMARTINSTALL

Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Jan Bongers

Periode: 8/2/2016 t/m 3/6/2016

Bedrijf c.q. instelling: SmartInstall BV

Bedrijfsmentor: Jan Taks

Plaats: Capelle aan den IJssel

Datum: 25 mei 2016

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren

afstudeeropdracht? Ja. Jan werkte al langer bij ons als oproepkracht en kende ons bedrijf zodoende al heel goed. Hij heeft zich in zeer korte tijd op de hoogte gesteld van het onderwerp 'datawarehouses' en wordt nu binnen SmartInstall al als expert gezien.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

Uitstekend.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

• Qua verantwoordelijkheid	goed / voldoende / matig / onvoldoende
• Qua zelfstandigheid	goed / voldoende / matig / onvoldoende
• Qua planmatig werken	goed / voldoende / matig / onvoldoende
• Qua creativiteit	goed / voldoende / matig / onvoldoende
• Qua productiviteit	goed / voldoende / matig / onvoldoende
• Qua samenwerken met collega's	goed / voldoende / matig / onvoldoende
• Qua draagvlakontwikkeling	goed / voldoende / matig / onvoldoende
• Qua inspelen op bedrijfscultuur	goed / voldoende / matig / onvoldoende
• Qua rekening houden met de specifieke context van het bedrijf	goed / voldoende / matig / onvoldoende
• Qua het op gang brengen van de nodige veranderingen	goed / voldoende / matig / onvoldoende

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde

Wij hebben ervaring met afstudeerders van de Haagse Hogeschool. Jan behoort in onze beleving tot de de besten.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?
Uitstekend.

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**
100%
- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**
• 100%
- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**
Zeer hoog.
- **Wat gebeurt er met het opgeleverde (eind)product?**
Wij gaan dit gebruiken als demo voor klanten.
- **Kunt u direct met het opgeleverde product aan de slag?**
Ja.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?
Nee.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Absoluut. Zoals hierboven reeds gesteld hebben wij vaker afstudeerders van de Haagse Hogeschool gehad en zijn wij voornemens ook in de toekomst afstudeerplekken beschikbaar te stellen.

