Structuur DWH - ETL

Herakles

Herakles – Eos - Prométhée (PR1 – PR2)

Release 1 – Persoon
Release 2 – Marchandise

Accepté par :			
Client:	SPF Finances, Decision Maker	Siemens	Siemens, Decision Maker
Représentant SPF Finances Date + Signature:	Veronique Vandamme – Manager de Programme	Représentant Siemens: Date + Signature:	Stefan Verplaetse – Manager de Programme

INHOUDSOPGAVE

Historique des modifications			
Version	Statut / Modifications	Date	Auteur
00.01	Draftversie ter interne revisie	09/06/2006	A. D'haene, C. Scheers
00.01B	Commentaren aangebracht door FodFin QC/ICT	19/06/2006	K. Berton
00.01B	Commentaar aangebracht door FodFin ICT-DWH	20/06/2006	T. Beerens
00.02	Vorige commentaren opgenomen	24/07/2006	A. D'haene
01.00	Versie ter validatie	04/08/2006	A. D'haene
01.01	Herziene versie	17/08/2006	A. D'haene
01.02	Precisering van de logische stappen van het ETL proces en van het feit dat de structuur van de beheerstabellen maar een insteek is, die in technische documentatie en implementatie verfijnd zal worden.	22/02/2008	A. D'haene

1. Document aanduidingen

1.1 Doel en opzet

Dit document bevat een hoog niveau beschrijving van de structuur en de processen van het datawarehouse voor het Herakles project. Dit document bevat de volgende onderwerpen, telkens opgedeeld in aparte hoofdstukken.

- Een overzicht van de volledige ketting van de datawarehouse vanaf de bronnen tot de constructie van de datamarts.
- Een beschrijving van de structuur van het ETL-gedeelte, ook hier wordt er gestreefd naar een zo hoog mogelijke uniformiteit en herbruikbaarheid tussen de processen, niet enkel binnen één release maar ook overheen de releases. Daarnaast wordt er ook naar gestreefd om door middel van een aangepaste structuur van het ETL-gedeelte een zo goed mogelijke ontkoppeling te krijgen tussen de bronnen enerzijds en de ETL-jobs en het DWH anderzijds om de impact bij wijzigingen of toevoeging van een bron te beperken.
- Een beschrijving van de wijze waarop het domeinmodel gepersisteerd zal worden in het DWH. Hierbij worden een aantal standaard structuren voorgesteld volgens dewelke het DWH opgezet zal worden om een zekere uniformiteit te bewaren doorheen het volledige project.

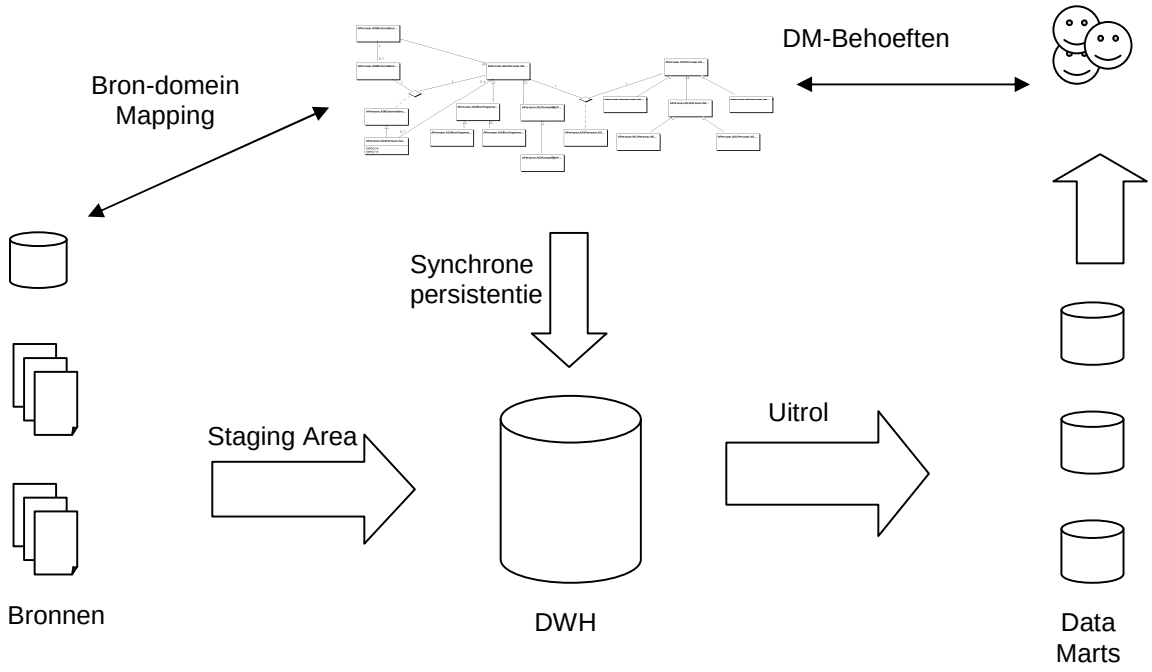
Tijdens elk van deze beschrijvingen moeten een aantal punten in acht genomen te worden. Het gaat hier enerzijds om vereisten die reeds geformuleerd zijn in het lastenboek en de offerte en anderzijds om een aantal beschouwingen die voortvloeien uit het uiteindelijke doel van deze omgeving zijnde risicobeheer.

Aangezien dit document is opgevat als opleveringsdocument voor fase PR113 zullen de concepten voorgesteld worden aan de hand van concrete voorbeelden uit deze fase. Voor een exhaustieve oplijsting van alle veschillende componenten (bronnen, mappings, tabellen) wordt er verwezen naar de gedetailleerde documenten die bij de oplevering van deze fase horen (mappingslijst, signatuurlijst, logisch datamodel DWH,…).

2. Het Datawarehouse

2.1 Overzicht

Het DWH is de persistentie in een IBM-DB2 databank van het domeinmodel dat tijdens de voorstudie werd opgesteld in het kader van de voorstudie. Het laat toe al de Subjecten, Facetten, en Klassen (S-F-C) op te slaan in een gestructureerde vorm.



Figuur 1: Overzicht DWH

Samen met het SFC-model vormt deze persistentie de centrale zuil van de omgeving en gezien het iteratieve karakter van het project zal het DWH-model in iedere iteratie aangepast worden.

Het is echter niet enkel het logische model van de DWH-persistentie dat zal wijzigen doorheen de verschillende iteraties en releases, ook het conceptuele model (SFC) kan onderhevig zijn aan wijzigingen. Dit is voornamelijk het geval wanneer er binnen een bron informatie gevonden wordt die oorspronkelijk niet in het domeinmodel van de voorstudie aanwezig was (vb. het niveau vestiging in het facet A01_PERSOON_PERSONNE. Het is onder meer hierom dat het van het grootste belang is dat het logische (en derhalve ook het fysische) model van de persistentie vormelijk gelijk is aan het domeinmodel. Dit laat toe om zowel voor het incorporeren van een nieuwe databron als voor het vastleggen van de informatiebehoeften voor een nieuwe data mart de communicatie met de business gebruikers strikt kan gebeuren in termen van het conceptueel model (business domein) zonder hen te moeten confronteren met technische issues met betrekking tot databanken, ETL-jobs en dergelijke. Hieruit volgt ook dat het incorporeren van nieuwe databronnen en het definiëren en uitrollen van bijkomende data-marts volledig van elkaar ontkoppeld kunnen worden in termen van ontwerp en implementatie.

2.2 De Bronnen

Voor de eerste iteratie zijn er vijf bronnen geïdentificeerd, binnen elk van deze bronnen wordt er een onderscheid gemaakt tussen de zogenaamde basistabellen enerzijds en referentieeltabellen anderzijds. Tussen de basistabellen bestaat er binnen elke bron telkens één primaire terwijl de andere basistabellen kunnen gezien worden als afhankelijk hiervan. Een voorbeeld hiervan is de tabel NPP_T_RP binnen de bron Signaletiek Natuurlijk Persoon waarin de natuurlijke sleutel, rijksregisternummer in dit geval, uniek is terwijl in de andere basistabellen zoals NPP_T_RP_ADRES_020 deze natuurlijke sleutel voorkomt als vreemde sleutel naar de primaire tabel en niet noodzakelijk uniek is.
Naast de gegevens die rechtstreeks afkomstig zijn uit brontabellen worden er in de basistabellen ook coderingen gebruikt (bv. Belastingsregime in de BTW-signaletiek) waarvan de omschrijvingen niet expliciet als tabel bestaan in de bron maar waarvan deze enkel beschikbaar zijn in de documentatie (en/of hard-gecordeerd in de applicatie zelf). Om deze coderingen op te kunnen nemen worden er zogenaamde manuele referentiebronnen voorzien waarin de informatie beschikbaar in de documentatie wordt overgenomen en opgeladen in het DWH.

Onderstaande tabel geeft een overzicht van het aantal basistabellen en brontabellen voor het referentieel die in iteratie 1 gebruikt worden per bron.

Bron	Aantal Basistabellen	Aantal referentiebronnen
KBO	15	25
Signaletiek Natuurlijk Persoon	4	5
Signaletiek Rechtspersoon	3	4
Signaletiek BTW	1	0
TP490	1	0
Manuele referentiebronnen	0	8

Indien men in bovenstaande tabel het aantal referentiebronnen optelt merkt men dat men ver boven de 27, in het conceptueel model, vastgelegde referentietabellen uitkomt. De reden hiervoor is tweeledig. Enerzijds is het zo dat een aantal van de referentietabellen, voornamelijk afkomstig van het KBO, in de bronnen opgesplitst zijn in meerdere tabellen en ook zo worden aangeleverd. Anderzijds wil de codering en granulariteit van een referentietabel ook wel eens verschillen tussen de verschillende bronnen zodanig dat, om deze te uniformiseren, eenzelfde referentietabel verschillende onafhankelijke brontabellen kan hebben. Dit laatste is onder meer het geval voor Y0212_CODE_SITUATION_JURIDIQUE waarvan de gegevens zowel afkomstig kunnen zijn van het KBO als van een interne bron van de FOD Financiën.

2.3 Het ETL-proces

2.3.1 Algemeen overzicht

De ETL-processen hebben tot doel de gegevens, komende van verschillende bronnen (intern en extern), en in verschillende formaten (relationeel in DB of flat-file, full of incrementeel), op te nemen in het DWH. Dit gebeurt in 8 etappes die aanschouwelijk weergegeven zijn in Figure 2: Overzicht ETL en waarbinnen er nog verschillende jobtypes bestaan.
Deze verschillende etappes zijn, in volgorde:

- Voorverwerking
- Extractie
- Identificatie
- Beheer van het referentieel
- Transformatie van decoratieve gegevens
- Hergroepering in termen van brontabellen
- Hergroepering in termen van het domeinmodel (bronoverkoepelend)

- Laden van de gegevens in het DWH.

Deze etappes en bijhorende jobtypes zullen in dit hoofdstuk overlopen worden met en korte omschrijving van de voornaamste verantwoordelijkheden. Voor elk van de jobtypes werd een “mapping-template” voorzien die dan voor elke effectief te implementeren job werd ingevuld en deel uitmaakt van de oplevering PR113.

De logische definitie van de communicatie tussen de verschillende etappes en de jobs gebeurt aan de hand van vastgelegde signaturen. Een signatuur is de verzameling van karakteristieken (velden, datatypes) van een tabel of bestand op een logisch niveau, onafhankelijk van de effectieve persistentie van de gegevens.

Naast deze verticaal afgelijnde functionaliteiten bestaan er nog een aantal functionele aspecten die in elke etappe opnieuw opduiken zoals, beheer van fouten (zowel logische als technische), genereren van rejects, opvolgen van gestarte en beëindigde jobs, opvolgen van de datakwaliteit, enz… Deze horizontale functionaliteiten worden besproken in sectie 2.5 Beheer en opvolging van de gegevens.

2.3.2 Voorverwerking

Onder de verantwoordelijkheden van de voorverwerking worden de volgende jobtypes begrepen:

- ***Opschoning van de gegevens***. Er is gebleken dat bepaalde bronnen die als tekstbestand worden aangeleverd nog steeds, door datastage, onverwerkbare karakters bevatten zoals een binaire 0x00 of waarin karakters zoals 0x0A (LF) voorkomen binnen een tekstveld (b.v. omschrijving) wat zorgt voor moeilijkheden bij het verwerken met datastage. Deze speciale karakters worden in deze voorverwerkingsstap verwijderd door een klein programma geschreven in C dat “in-line” in DataStage zelf geïntegreerd wordt.
- ***Laden van de gegevens in databank-tabellen voor analyse***.
    - Voor de analyse met profile-stage enerzijds en om op een eenvoudige manier de referentiële integriteit van een bron te valideren via SQL-statements is het interessant om de gegevens in een databank geladen te hebben. Dit type van job verzorgt dit laadproces voor de gegevens die onder de vorm van tekstbestanden binnenkomen. ;
    - Deze jobs worden ook gebruikt om reeds een rudimentaire kwaliteitscontrole uit te voeren met betrekking tot de aanwezigheid en het ingevuld zijn van identificatiegegevens.
- ***Splitsen van bestanden in individuele signaturen***: Een aantal brongegevens worden aangereikt in de vorm van tekstbestanden waarin verschillende tabellen van de bron in éénzelfde bestand voorkomen met een aanduiding van het informatietype. Hieruit volgt een variabele lengte en signatuur van de verschillende records binnen hetzelfde bestand. Dit jobtype splist een dergelijk bestand in een aantal nieuwe bestanden waarin telkens de records met overeenkomende betekenis en signatuur vervat zitten.
- ***Delta-creatie***: Een aantal van de bronnen worden telkens opnieuw aangeleverd in de vorm van een volledig extract. Dit jobtype zal zo’n volledig extract omzetten naar een “delta-extract” dat enkel de wijzigingen ten opzichte van de vorige toestand bevat.
- ***Bestandspivotering***: Binnen een aantal bronnen (voornamelijk de referentietabellen van het KBO) is het het geval dat de meertaligheid van de omschrijvingen geïmplementeerd wordt door de taalcode “uit te normaliseren” waardoor er voor een bepaalde code, 3 records voorkomen in de bron voor de drie verschillende landstalen. Hoewel dit systeem zeer flexibel is met betrekking tot het bijvoegen van nieuwe talen werd er binnen de modelisatie van het DWH voor geopteerd om deze taalcode expliciet te denormaliseren zodanig dat de informatie in de drie talen in hetzelfde record beschikbaar is. Om dit te bewerkstelligen werd dit type job voorzien.
- ***Bestandssamenvoeging***: Binnen het KBO-referentieel zijn er nog drie entiteiten die door middel van meer dan 1 bestand worden aangeleverd. Dit jobtype verzorgt de JOIN tussen deze tabellen.

2.3.3 Extractie

Aan de uitgang van de vorige stap worden, ten gevolge van de delta-extractie, enkel de gegevens die een wijziging in het DWH als gevolg kunnen hebben weerhouden. Het in deze etappe enige voorkomende jobtype "***Split***" zal de binnenkomend records splitsen in identificerende gegevens, decoratieve gegevens en links met het referentieel. De voornaamste reden voor deze split is het hergebruik en verregaande uniformisering van de jobs voor identificatie en beheer van het referentieel. Een andere reden is het feit dat indien een bron verdwijnt of vervangen wordt door een andere bron met gelijkaardige informatie (bv. signaletiek NPP vervangen door het rijksregister) dit slechts een beperkte of zelfs geen impact heeft op de identificatie- en refentieelbeheersjobs.
Per element van een bepaald type wordt er slechts één uitgang voorzien. Indien een bepaald element zoals een identificatie of een link naar een referentieel meermaals voorkomt (in verschillende rollen) worden deze achter elkaar gezet en naar dezelfde uitgang gestuurd. Om de records daarna op een ondubbelzinnige wijze terug te kunnen samenstellen wordt er een veld toegevoegd dat de rol aanduidt die dit element speelt binnen het record. Een voorbeeld hiervan is zijn de activiteitscodes die binnen eenzelfde basistabel meermaals kunnen voorkomen in verschillenden rollen zoals "hoofdactiviteit" of "nevenactiviteit". Een ander voorbeeld, ditmaal met betrekking tot de identificatiegegevens, zijn de voorkomens van de rijksregisternummers van "Titularis" en "Partner" binnen de basistabel NPP_T_AJ.
Om achteraf deze records weer samen te kunnen stellen worden zij voor het splitsen voorzien van een load- en een record-ID. Naast de wedersamenstelling verzorgen deze ook de mogelijkheid om een record doorheen de volledige ETL-ketting op te volgen.

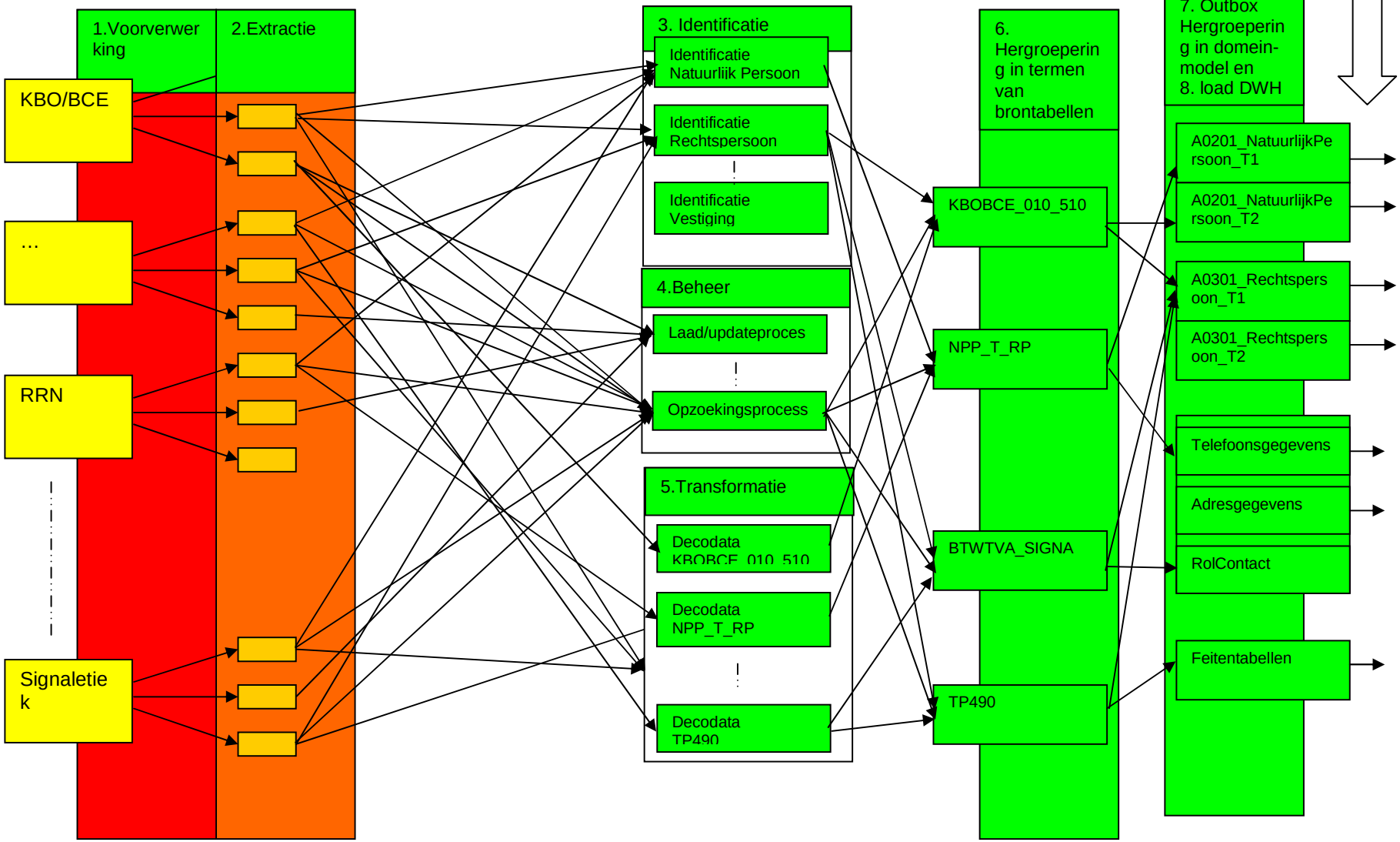


Figure 2: Overzicht ETL

2.3.4 Identificatie

2.3.4.1 Definities

Een aantal van de domein-entiteiten zullen op zeer uiteenlopende wijze geïdentificeerd worden door de natuurlijke sleutels van de verschillende bronnen. Het beheer van deze sleutels en de mapping op de interne technische sleutels (TK's) is van het hoogste belang voor identificatie en matching van identiteit enerzijds en het anonimiseren van de gegevens anderzijds. Deze sectie bespreekt enkel de identificatie, voor anonimisatie worden dezelfde structuren gebruikt aangezien er vanuit gegaan wordt dat gegevens die toelaten een persoon te identificeren per definitie ook die gegevens zijn die, met het oog op bescherming van de privacy, beperkt toegankelijk dienen gemaakt te worden.
Om externe identificatie en anonimisering op een formelere wijze te ondersteunen werd in het domeinmodel de klasse A0102_ExterneIdentificatie_IndentificationExterne voorzien. Met behulp van deze entiteit worden de externe natuurlijke sleutels vertaald naar hun interne technische sleutel. Binnen de natuurlijke sleutels dienen we een onderscheid te maken tussen twee types:

- de "witte" sleutels: Dit zijn velden die binnen de externe bron de functie van primaire (of alternatieve) sleutel vervullen en derhalve als unieke identificatie van de betrokken entiteit kunnen beschouwd worden. Indien het hier een samen gestelde sleutel betreft kunnen deze samenstellende velden geconcateneerd worden en als dusdanig ook gebruikt worden binnen hetzelfde mechanisme.
- de "grijze" sleutels: Dit zijn velden of combinaties van velden die een domein-entiteit niet noodzakelijk uniek identificeren binnen een bepaalde bron maar eventueel wel (met een bepaalde kwaliteitsindicatie) kunnen gebruikt worden voor secundaire identificatie, bijvoorbeeld indien primaire, witte identificatie niet mogelijk is. Welke combinaties van velden in aanmerking komen als grijze sleutel wordt bepaald tijdens de analyse van de bron.

Binnen de grijze sleutels wordt er een onderscheid gemaakt tussen verschillende datakwaliteitsniveaus (DQ-niveaus)

- DQ1 en DQ2: gereserveerd voor respectievelijk de witte en manuele identificatie.
- DQ3: Zuivere grijze identificatie (100% match). Een combinatie van grijze sleutelvelden die tijdens de analyse van de bron naar voren kwam als bruikbaar voor een unieke identificatie. Twee combinaties van deze velden die identiek zijn beschrijven ook noodzakelijk dezelfde entiteit. Een voorbeeld van een combinatie van grijze velden die eventueel (te verifiëren tijdens de analyse) zo'n een DQ3 identificatie zou kunnen definiëren is de combinatie "maatschappelijke benaming, oprichtingsdatum, rijksregisternummer oprichter" voor de identificatie van een rechtspersoon. Dit geeft geen theoretische zekerheid maar indien de analyse uitwijst dat deze combinatie uniek is en dit door de business kan gevalideerd worden kan deze combinatie gebruikt worden alsof het een witte sleutel was. Daarnaast kan dit kwaliteitsniveau en mechanisme ook gebruikt worden voor samengestelde natuurlijke sleutels aangezien men in dat geval ook zeker is van de uniciteit van de identificatie.
- DQ4: Een zeer sterk benaderende match (doch geen 100%). Een combinatie van grijze sleutelvelden waarvan tijdens de analyse blijkt dat zij een quasi zekere match voor identificatie. Dit geeft echter geen zekerheid dat bijvoorbeeld twee personen die op basis van een DQ4-identificatie als dezelfde worden aanzien ook daadwerkelijk dezelfde zijn.
- DQ5 tot DQ8: Combinaties van grijze sleutelvelden die resulteren in benaderende matches met dalende kwaliteit (te bepalen tijdens de analysefase en eventueel te evalueren door QualityStage stages geïntegreerd in DataStage). Identificaties die in dit pad gebeuren dienen met de nodige omzichtigheid behandeld te worden. Het effectieve kwaliteitsniveau (5 tot 8) wordt toegekend in functie van het gewicht van de match zoals toegekend door QualityStage.
- DQ9: Deze gegevens konden niet worden geïdentificeerd tegenover een reeds bestaande instantie van een entiteit, dit kan enerzijds zijn ten gevolge van een mindere kwaliteit van de gegevens maar het is even goed mogelijk dat het hier gewoon over een nieuwe instantie van deze entiteit handelt.

Uit deze problematiek van verschillende kwaliteitsniveaus voor identificatie blijkt dat het toekennen van slechts één technische sleutel niet volstaat. Indien voor identificaties van type DQ4 dezelfde technische sleutel zou toegekend

worden als het oorspronkelijke record zou dit betekenen dat, indien achteraf blijkt dat deze identificatie niet juist was, het onmogelijk is om de gegevens behorende tot deze 2 (blijkbaar verschillende) personen terug van elkaar te scheiden en te re-alloceren aan de nieuwe (correcte) TK's. Om deze reden wordt er voor identificaties vanaf het type DQ4 naast de primaire technische sleutel (PTK) ook een tweede technische sleutel (TK) bijgehouden die wel uniek is overheen de verschillende wijzen van identificatie van een instantie van een entiteit. Het is dank zij deze TK dat het mogelijk is om achteraf, indien blijkt dat de DQ4-identificatie foutief was, de gegevens terug uit elkaar te halen zonder dat het nodig is deze uit het datawarehouse.
Structureel is deze oplossing gelijkaardig aan een traag veranderende dimensie van type 2 met dit verschil dat het onderscheid tussen records met dezelfde primaire technische sleutel en verschillende secundaire technische sleutels niet gemaakt wordt op basis van wijzigingen aangebracht aan bepaalde velden maar op basis van informatie met betrekking tot kwaliteit van match van de grijze velden.

Voor de identificatie maken we verder nog een onderscheid tussen enerzijds het laden van de gegevens vanuit de primaire basistabel van een bron en het opzoeken van de technische sleutels voor een bepaalde binnenkomende natuurlijke sleutel vanuit een afhankelijke basistabel (zie sectie 2.2 voor verduidelijking omtrent het verschil tussen primaire en afhankelijke basistabellen). In het algemeen wordt er gesteld dat enkel bij het laden vanuit de primaire basistabellen nieuwe identiteiten worden toegevoegd in het datawarehouse aangezien indien het geval zich voordoet dat een instantie van een entiteit voor het eerst wordt waargenomen in een afhankelijke basistabel dit betekend dat er problemen zijn met de referentiële integriteit van de bron. De waargenomen instantie zal hoe dan ook opgenomen worden in de identificatietabellen maar dit zal gepaard gaan met genereren van een foutmelding in de foutenlog.
Voorgaande stelling laat toe de volledige identificatie (basistabel per basistabel) in batchmodus uit te voeren zonder dat het tijdens het verwerken van een batch noodzakelijk is de look-up signaturen te verversen, wat de performantie enorm ten goede zal komen.

2.3.4.2 Automatische identificatie

Voor de automatische identificatie dient er tijdens de analysefase, naast een identificatie van de primaire en secundaire basistabellen per bron, een oplijsting te gebeuren van de witte en grijze (met bijhorende kwaliteit) sleutelvelden voor elke bron. Voor elke entiteit uit het domeinmodel voor dewelke er een identificatie noodzakelijk is worden de volgende look-up signaturen voorzien. Deze worden gegenereerd vanuit de tabellen voorzien voor externe identificatie (zie sectie 2.4.2 Sleutelbeheer van de identificatie) en gebruikt als look-up signaturen tijdens de identificatiejobs.

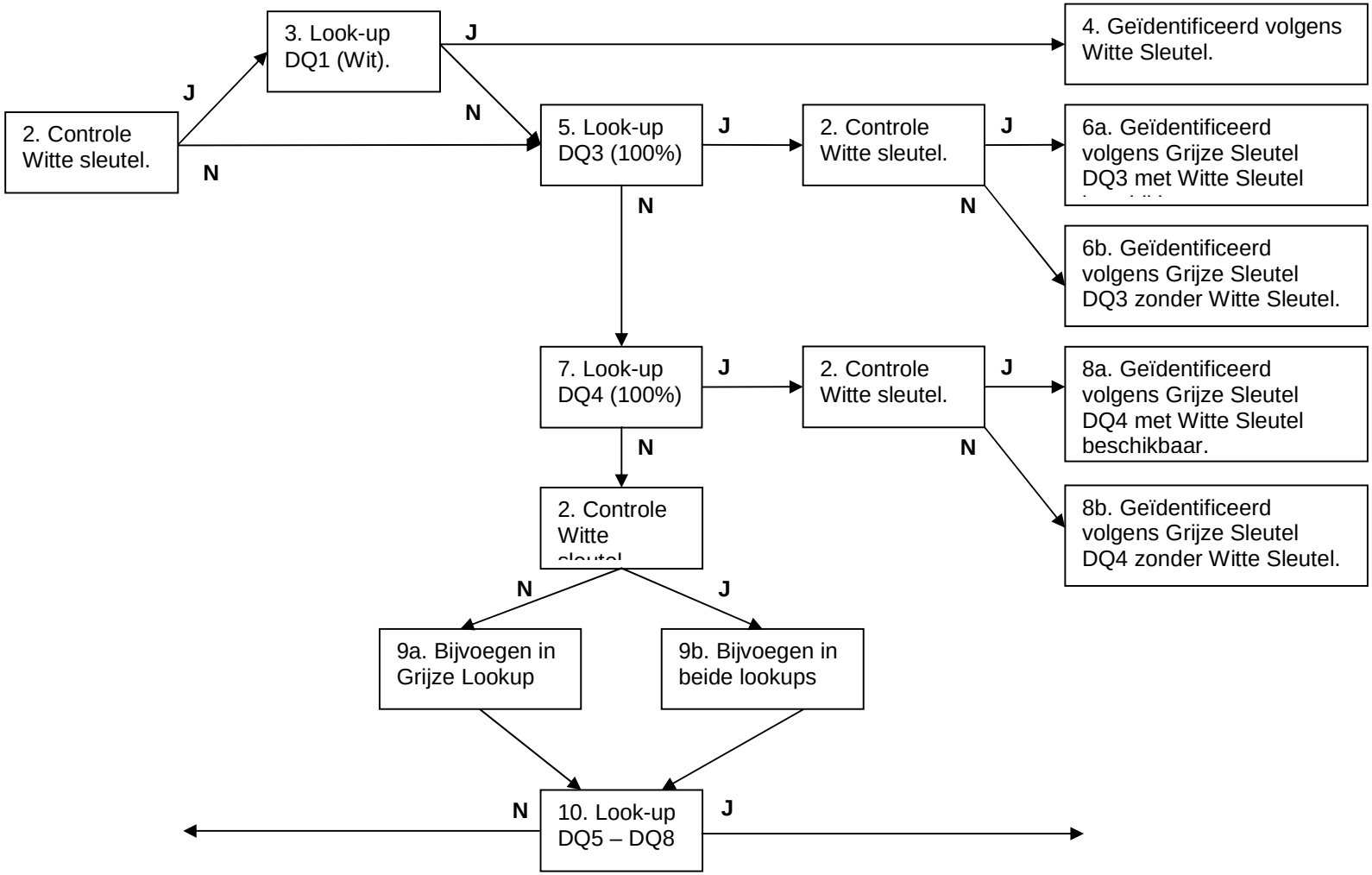
- LKP_WIT_
  - NKw – Natuurlijke witte sleutel
  - PTK – Initiële functionele sleutel
- LKP_GRIJS_
  - NKg – Natuurlijke grijze sleutelvelden. Dit kunnen meerdere velden zijn afhankelijk van de entiteit voor dewelke dit identificatiemechanisme opgezet is.
  - PTK – Initiële functionele sleutel
  - TK – Technische sleutel
  - DQ – Data kwaliteitsniveau

Om performantieredenen maken we een onderscheid tussen een initiele full-load en een incrementeel laadproces. Tijdens het initiële laadproces zijn de look-up tabellen nog leeg (nog geen gegevens in het DWH geladen) en dient er dus ook geen identificatie te gebeuren wat resulteert in een hogere performantie. Alle binnenkomende (unieke) witte sleutels met bijhorende decoratieve velden die dienst doen als samenstellende velden voor grijze sleutels kunnen zonder meer geladen worden in bovenstaande tabellen terwijl hen de nodige PTK's, en TK's worden toegekend.

Aangezien het incrementele laadproces erop voorzien moet zijn om zowel gegevens voorzien van een witte sleutel als gegevens waarvoor de witte sleutel niet gekend is, en die enkel kunnen geïdentificeerd zijn door grijze sleutels,

te laden zal de datastroom heel wat complexer zijn. Deze stroom staat voorgesteld in Figuur 3: Overzicht identificatie en bestaat uit de volgende componenten waarvan de eerste enkel betrekking heeft op de analyse en dus niet voorgesteld is in de “run-time”-stroom.

1. **“Analyse fase”** Identificeer in de bron (of deel/signatuur van een bron) de witte en grijze sleutelvelden, ken aan de verschillende combinaties van grijze sleutelvelden een kwaliteitsniveau toe met bijzondere aandacht voor sleutels van kwaliteitsniveau DQ3.
2. **“Controle wit sleutelveld”** Controleer of er een wit sleutelveld in de binnenkomende signatuur aanwezig is en of deze ingevuld (geen null-waarde) is. Indien het antwoord op één of beide vragen negatief is kan er geen witte identificatie (stap 3) uitgevoerd worden en dient er direct overgegaan te worden tot de grijze identificatie.
3. **“Look-up DQ1”** Zoek de waarde van het witte sleutelveld op in de look-up tabel ID_LOOKUP_WIT.
4. **“Geïdentificeerd volgens witte sleutel”** De waarde van de witte sleutel werd gevonden in ID_LOOKUP_WIT, aan het binnenkomende record worden dus de in die tabel gevonden waarden voor PTK, PTKM, en TK toegekend. Het behandelde veld krijgt in het DWH de kwaliteitswaarde DQ1 en de bijhorende grijze sleutelvelden worden samen met de gevonden PTKM en TK toegevoegd aan de tabel ID_LOOKUP_GRIJS en krijgen ook hier kwaliteitsindicatie DQ1.



Figuur 3: Overzicht identificatie

5. **“Look-up DQ3”** Match de waarden van de grijze sleutelvelden van het binnenkomende record tegenover de (eventueel meerdere) grijze sleutelcombinaties van kwaliteitsniveau DQ3. Indien er meerdere sleutelcombinaties bestaan worden zij geëvalueerd volgens dalende performantie en in geval van match worden de overige niet meer gecontroleerd (kritisch pad).
6. **“Geïdentificeerd volgens Grijze sleutel DQ3”**
    a. **Witte Sleutel beschikbaar”** De combinatie van grijze sleutelvelden werd gevonden in de ID_LOOKUP_GRIJS. Aan het binnenkomende record worden de in die tabel gevonden waarden voor PTK en PTKM en TK toegekend. Het behandelde record krijgt in het DWH de kwaliteitswaarde DQ1. De in het binnenkomende record aanwezige Witte Sleutel wordt samen met de gevonden PTK, PTKM en TK toegevoegd in de look-up tabel ID_LOOKUP_WIT.
    b. **“Witte Sleutel niet beschikbaar”** De combinatie van grijze sleutelvelden werd gevonden in de ID_LOOKUP_GRIJS. Aan het binnenkomende record worden de in die tabel gevonden waarden voor PTK, PTKM en TK toegekend. Het behandelde record krijgt in het DWH de kwaliteitswaarde DQ3. Er worden geen records toegevoegd aan de reeds bestaande look-up tabellen.
7. **“Look-up DQ4”** Match de waarden van de grijze sleutelvelden van het binnenkomende record tegenover de (eventueel meerdere) grijze sleutelcombinaties van kwaliteitsniveau DQ4. Indien er meerdere sleutelcombinaties bestaan worden zij geëvalueerd volgens dalende performantie en in geval van match worden de overige niet meer gecontroleerd (kritisch pad).
8. **“Geïdentificeerd volgens Grijze Sleutel DQ4”**
    a. **“Witte Sleutel beschikbaar”** De combinatie van grijze sleutelvelden werd gevonden in de ID_LOOKUP_GRIJS. Aan het binnenkomende record worden de in die tabel gevonden waarden voor PTK en PTKM toegekend, er wordt echter een nieuwe waarde voor TK gegenereerd (aangezien men niet 100% zeker kan zijn van de identiteit. Het binnenkomende record krijgt in het DWH datakwaliteitsniveau DQ1. De in het binnenkomende record beschikbare Witte Sleutel wordt samen met de gevonden PTK en PTKM en de nieuw gegenereerde TK opgeslagen in de look-up tabel ID_LOOKUP_WIT.
    b. **“Witte Sleutel niet beschikbaar”** De combinatie van grijze sleutelvelden werd gevonden in de ID_LOOKUP_GRIJS. Aan het binnenkomende record worden de in die tabel gevonden waarden voor PTK en PTKM toegekend, er wordt echter een nieuwe waarde voor TK gegenereerd (aangezien men niet voor 100% zeker kan zijn van de identiteit. Het binnenkomende record krijgt in het DWH kwaliteitsniveau DQ4. Er worden geen records toegevoegd aan de aan de reeds bestaande look-up tabellen.
9. **“Bijvoegen van niet eerder gevonden identiteit”**
    a. **“Bijvoegen in ID_LOOKUP_GRIJS”** Er kon niet met voldoende zekerheid een PTK bepaald worden tegenover de reeds bestaanden. Er zal een nieuwe PTK, PTKM en TK worden aangemaakt en samen met de grijze sleutelvelden toegevoegd worden in ID_LOOKUP_GRIJS met kwaliteit DQ3.
    b. **“Bijvoegen in beide lookups”** Er kon niet met voldoende zekerheid een reeds bestaande PTK bepaald worden tegenover de reeds bestaanden. Er zal een nieuwe PTK, PTKM en TK worden aangemaakt, samen met de grijze sleutelvelden toegevoegd worden in ID_LOOKUP_GRIJS met kwaliteit DQ1 en dezelfde technische sleutels worden gebruikt om samen met de Witte Sleutel toegevoegd te worden aan ID_LOOKUP_WIT.
10. **“Lookup DQ5-DQ8”** Hier worden met behulp van QualityStage een aantal stages gedefinieerd die kandidaat-matches kunnen genereren die niet naar boven kwamen tijdens de DQ3 en DQ4 matches.

2.3.4.3 Manuele Matching

Onder manuele matching kunnen 2 verschillende mechanismen verstaan worden. Enerzijds is dit het analyseren en proberen te matchen met behulp van datastage van alle records die beneden de DQ4 treshold vallen. Anderzijds gebruiken we de term manuele match ook voor het manueel wijzigen, door een expert, van een manuele PTK (PTKM) bij een bestaande PTK zodanig dat bestaande entiteiten kunnen gehergroepeerd worden als één enkele entiteit.

In het eerste geval worden de gegevens door de bovenstaande jobs in de grijze look-up tabel ingebracht met een eigen nieuwe PTK en TK. Deze gegevens worden echter ook doorgestuurd naar QualityStage expert die verantwoordelijk is voor het genereren van QualityStage Stages waarin gevorderde matchings met gespecialiseerde algoritmes gedefinieerd en geautomatiseerd uitgevoerd kunnen worden zodanig dat ook deze, met bijhorende kwaliteit, geïdentificeerd kunnen worden tegenover reeds bestaande instanties. De koppeling van deze “a posteriori matching” aan de oorspronkelijk geïdentificeerde gebeurt, technisch gezien, op dezelfde wijze als de koppeling van PTK’s d.m.v. PTKM’s. Concreet verschillende deze mechanismen enkel door het feit dat in het eerste geval deze koppeling gebeurt door middel van een automatische update van de manuele mappingtabel vanuit de QualityStage jobs terwijl in het tweede geval dit gebeurt door het opladen van een matching-tabel, manueel opgesteld met behulp van een Office tool (bv. Excell in .csv-formaat) en waarvoor de expert zich baseert op rapporten getrokken uit de beveiligde (geannonimiseerde) gedeelte van het data-warehouse (waarin de grijze sleutelvelden ook voorkomen.) Deze gegevens worden in het DWH geladen door een speciaal hiervoor voorziene DataStage-job.

2.3.5 Beheer van het referentieel

Naast identificerende elementen bevatten de brontabellen ook links naar een aantal referentietabellen. Voor elk van deze referentietabellen werd er minstens één bron geïdentificeerd van waaruit deze referentietabel opgevuld wordt. Voor het beheer van elk van deze referentietabellen werden er telkens 2 jobs gedefinieerd die voor de 27 opgenomen referentietabellen quasi uniform zijn:

- ***Laden van de referentietabel***: Deze job dient voor het updaten van de referentietabel en ondersteunt zowel het initieel opvullen van deze tabel als het doorvoeren van wijzigingen tijdens de incrementele loads. Elke in de bron voorkomende referentiecode (ook te beschouwen als natuurlijke sleutel) wordt hier vertaald naar een technische sleutel.
- ***Opzoeken van een sleutel***: Voor alle voorkomens van een verwijzing naar een referentietabel vanuit een basistabel dient er een vertaling te gebeuren die de natuurlijke sleutel, van toepassing binnen die welbepaalde bron, omzet naar een technische sleutel geldig binnen de DWH-omgeving.

Naast de 27 referentietabellen zijn er nog 2 bijkomende tabellen die op dezelfde manier behandeld worden. Dit zijn de A0118_LIJST_PERSONNE en de A0102_EXTERNE_IDENTIFICATIE_TYPE_WITTE _SLEUTEL. Deze hebben beide een manueel samengesteld bronbestand. Daarnaast heeft de A0102 ook een brontabel als input voor het laden, afkomstig van het KBO. Voor A0118 in het manueel (hardcoded) bestand voor de eerste iteratie enkel de 4 lijsten van de bron TP490 voor terwijl er voor de A0102 voor iteratie 1 slechts 2 manuele records heeft die respectievelijk het ondernemingsnummer en het rijksregisternummer definiëren.

Vanuit een hoog niveau logica en met het oog op hergebruik van mechanismen kan men stellen dat de 2 hierboven gedefinieerde jobtypes logisch gezien overeen komen met enerzijds de identificatie vanuit de primaire basistabel en de anderzijds de identificatie van identiteiten vanuit de afhankelijke basistabellen. De referentieelbron neemt hier de rol over van de primaire basistabel terwijl de andere basistabellen, die de verwijzingen naar het referentieel bevatten, gezien kunnen worden als de afhankelijke tabellen. Ook hier kan men stellen, indien men de parallel doortrekt, dat het 2de jobtype in principe op geen enkel moment een toevoeging aan het referentieel kan doen aangezien dit weer betekend dat er problemen zijn met de structurele integriteit van de bron. Er wordt, net als bij de identificatie, evenwel toch voorzien dat het opzoeken van een sleutel die nog niet voorkomt in de referentietabel resulteert in het toevoegen van een lijn in deze tabel waarbij een nieuwe technische sleutel gegenereerd wordt en

waarvan de omschrijving per definitie op “ONBEKEND” wordt gezet. Deze situatie heeft echter ook weer (cfr. de identificatie) tot gevolg dat een foutmelding wordt toegevoegd aan de foutenlog. Men kan stellen dat het mechanisme voor het beheer van de referentietabellen een vereenvoudigde vorm is van het mechanisme voor identificatie.

Er bestaat momenteel één toevoeging aan dit mechanisme van referentieelbeheer zijnde de identificatie van adressen en het toekennen van technische sleutels van deze contactgegevens. Hoewel de mechanismen voor de samenstellende componenten van een adres (Landcodes, gemeentecodes, straatcodes) dezelfde blijven worden deze bij de identificatie van een adres samengebracht in één mapping die het binnenkomend adres opsplitst in deze componenten, de individuele opzoekingen doet en daarna de resultaten hiervan behandelt op een manier gelijkaardig aan de grijze-sleutel-identificatie.

2.3.6 Transformatie van de decoratieve gegevens

Onder decoratieve gegevens worden gegevens begrepen die noch bijdragen tot de identificatie van een individu of onderneming, noch een link naar een referentietabel voorstellen. Het zijn gegevens die typisch bijkomende informatie geven over het individu of de onderneming aan dewelke zij gekoppeld zijn. Aangezien in dit project het merendeel van deze gegevens worden gemodelleerd via links naar referentietabellen blijven er voor deze transformatie voornamelijk datumgegevens over zoals bijvoorbeeld “Datum van oprichting”, “Datum inschrijving in KBO”, “Datum stopzetting onderneming” voor gegevens omtrent ondernemingen.

Er wordt voorzien in 1 job van dit type per basistabel wat resulteert in 24 jobs van dit type. Deze jobs hebben als voornaamste verantwoordelijkheid het “casten” van het veldtype dat in de bron voorkomt naar het veldtype van het DWH. Daarnaast zorgen zij ook voor de vertaling van datum naar het formaat dat gebruikt wordt als functionele sleutel van de tijdsdimensie (zie 2.4.5 Beheer en opbouw van de tijdsdimensie voor de definitie van deze functionele sleutel).

2.3.7 Hergroepering in termen van basistabellen

Nadat in de voorgaande 3 etappes (die dankzij de “split” die tijdens de extractie werd gedefinieerd onafhankelijk van elkaar uitgevoerd kunnen worden) alle natuurlijke sleutels extern aan het DWH vertaald werden in interne technische sleutels worden in dit jobtype in eerste instantie de basistabellen terug samengesteld op basis van het record-ID dat in de split werd toegekend. Hierna wordt dit record, binnen deze zelfde job, opnieuw gesplitst maar ditmaal in termen van het domeinmodel. In feite worden de basistabellen hier geprojecteerd tegen de domeinentiteiten waarop zij een impact hebben. De basistabel “Signaletiek BTW” bijvoorbeeld, heeft impact op meerdere entiteiten in het domeinmodel zoals de klassen A0301_Rechtspersoon_PersonneMorale, A0201_NatuurlijkePersoon_PersonnePhysique, A0202_BedrijfNatuurlijkePersoon_PersonnePhysiqueEntreprise, A0108_Activiteitsdomein_DomainDActivité, en A0109_Contract_Contrat waarin telkens een gedeelte van de informatie in het binnenkomende BTW-record in het DWH gepersisteerd wordt.

Aangezien verschillende basistabellen een impact kunnen hebben op dezelfde domein-entiteit werd er geopteerd om doorheen de verschillende jobs van dit type “Hergroepering” ervoor te zorgen dat de gegevens bestemd voor eenzelfde entiteit ook dezelfde signatuur hebben teneinde deze gegevens in de volgende stap op een uniforme manier te kunnen verwerken.

Indien een bepaalde basistabel slechts gedeeltelijk binnen 1 iteratie wordt opgenomen worden de gegevens aan de uitgang van deze job bewaard om te kunnen voorzien in een historische load van deze gegevens tijdens een volgende iteratie.

2.3.8 Hergroepering in termen van het domeinmodel

In de vorige etappe werden de gegevens opnieuw gesplitst in termen van de domein-entiteiten. De uniformisering van de signaturen heeft als gevolg dat de jobs in deze etappe telkens maar 1 signatuur aan de ingang hebben. Er

worden echter wel meerdere bestanden gegenereerd met deze signatuur afkomstig van de verschillende bronnen. Elk van deze bestanden heeft een verschillende prioriteit van laden en is, afhankelijk van de business-rules, mogelijk ook enkel van toepassing op instanties van entiteiten die aan bepaalde voorwaarden voldoen.
De primaire verantwoordelijkheid van dit jobtype is dus feitelijk het implementeren van de business-rules met betrekking tot bronprioretisering en het voorwaardelijk incorporeren van gegevens. Hier dienen mechanismen voorzien te worden die erop toezien dat tijdens een incrementele load enkel die records worden uitgestuurd voor te laden (of updaten) waarvoor er ook werkelijk nieuwe gegevens toegekomen zijn.

2.3.9 Laden van het Datawarehouse

Na toepassing van de business-rules (prioretisatie) zijn de gegevens klaar om effectief geladen te worden in het Datawarehouse. In deze stap worden zowel de referentieeltabellen en identificatie als de dimensies en brugtabellen gelanden in het DWH.

2.4 De datamodellen

2.4.1 Algemeen

Het uitwerken van het databankmodel van het DWH dat het domeinmodel volledig dekt en tegelijkertijd de noodzakelijke performantie kan behalen is typisch een manuele stap (ontwerp) die moeilijk (of eerder niet) te automatiseren valt.

Volgens Inmon is een datawarehouse een geïntegreerde, subject-georiënteerde persistentie (non-volatile) van gehistoriseerde gegevens. Hieraan kan men volgens Kimball nog toevoegen dat een DWH geoptimiseerd is voor rapportering in tegenstelling tot operationele systemen die geoptimiseerd zijn voor transactionele verwerking.
Het is vanuit deze laatste definitie dat er binnen datawarehouseprojecten normaal gezien niet gemodelleerd wordt in een 3de normaalvorm gezien dit de complexiteit van de rapporteringsqueries enorm de hoogte in zou drijven met bijhorende negatieve impact op de performantie. Een 3de normaalvorm is op zich wel zeer interessant is voor transactionele verwerking gezien de gegevens op geen enkel moment ontdubbeld worden (geen redundantie) waardoor eenzelfde gegeven niet op meerdere plaatsen moet ingevoegd worden met de nodige integriteitsproblemen tot gevolg.
De definitie van Kimball gaat er echter wel vanuit dat de rapporteringsvereisten strikt vastgelegd zijn zodanig dat het model hierop getuned kan worden. In dit project echter zijn de rapporteringsvereisten niet op voorhand vastgelegd waardoor de aanpak meer in de richting van Inmon drijft. Er worden echter een aantal mechanismen gebruikt die hun oorsprong vinden in het werk van Kimball. Zo wordt er bijvoorbeeld voor de referentieeltabellen geen 3de normaalvorm gebruikt voor wat betreft de verschillende talen voor de beschrijvingen. De taal wordt in deze gevallen in de tabel “ingedenormaliseerd.” Dit geeft beperkingen naar het toevoegen van talen toe maar de afweging werd gemaakt (en door de business goedgekeurd) dat het model wordt beperkt tot de drie landstalen.

Tussen deze twee uitersten (Kimball en Inmon) zal een “gulden middenweg” gevonden moeten worden die toelaat de massa aan gegevens binnen dit project op een performante en onderhoudbare manier te stokkeren.

Om het datamodel desondanks toch een bepaalde uniformiteit te geven (het wiel niet iedere iteratie opnieuw uit te vinden) zullen er in de volgende secties een aantal standaard structuren gedefinieerd worden die binnen de verschillende releases en iteraties regelmatig zullen hergebruikt worden.

Deze standaardstructuren worden opgebouwd vanuit het oogpunt de volgende basisfunctionaliteiten te ondersteunen:

- Voor elke dimensie, de mogelijkheid om zowel een type 1 als type 2 slowly changing mechanisme te voorzien door de business te bepalen op het niveau van kolom.

- De mogelijkheid om feiten- en brugtabellen transparant te koppelen aan verschillende dimensies (vb. Natuurlijk Persoon en Rechtspersoon) die beiden logische extenties zijn van een gemeenschappelijke ouder en beiden aangeroepen moeten kunnen worden door een feiten- of brugtabel (bv. de brugtabel GebruikContact.
- De mogelijkheid om een zelfde instantie van een entiteit op meerdere wijzen en met onderscheiden kwaliteitsniveaus te kunnen identificeren uit verschillende onafhankelijke bronnen met de mogelijkheid om verbeteringen in de brongegevens in te brengen.

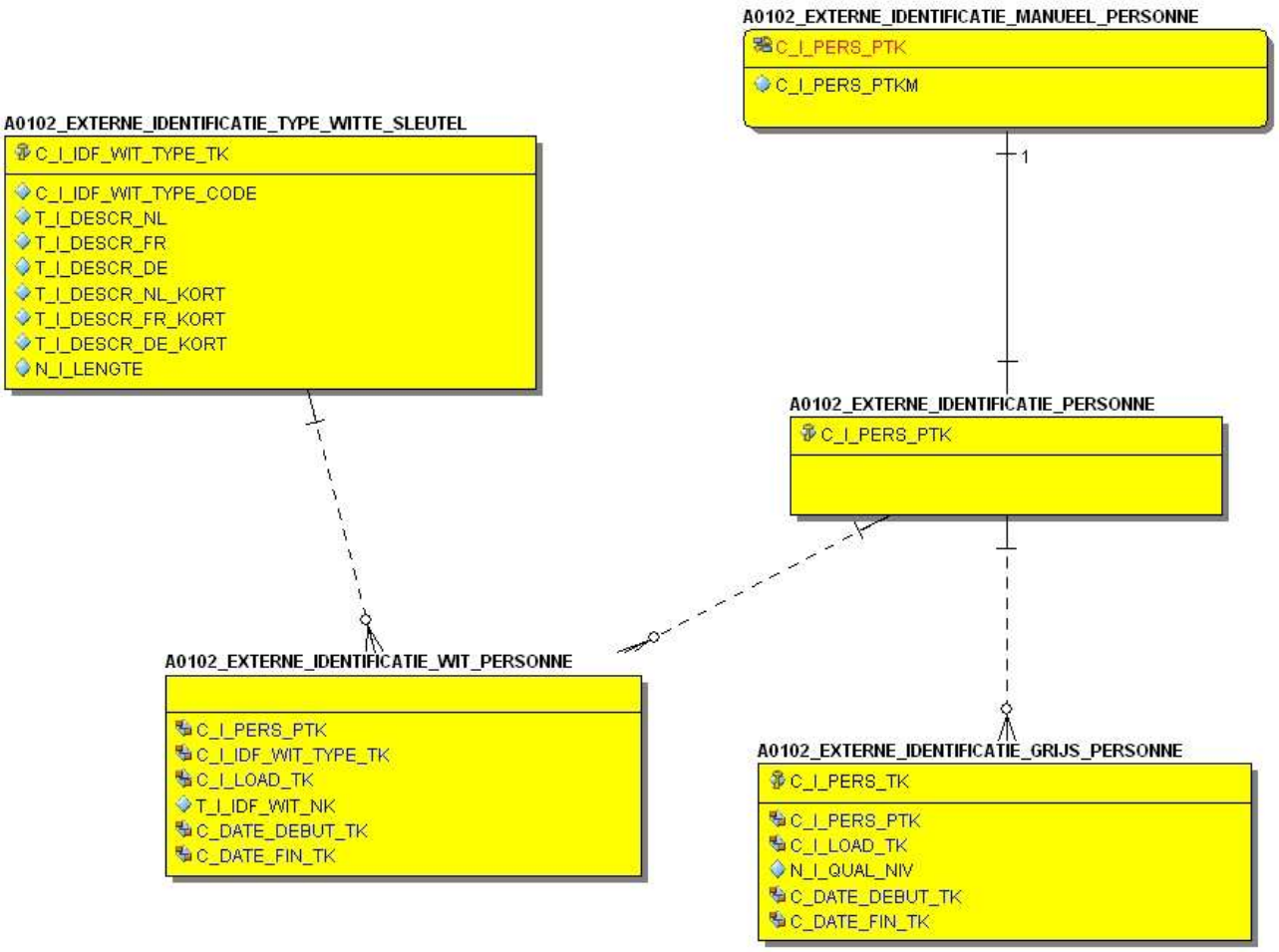
Voor elk van deze gewenste functionaliteiten kan men zich mechanismen inbeelden die een dergelijke functionaliteit ondersteunen, deze mechanismen zijn telkens, hetzij reeds algemeen gekend als “best practice” binnen het datawarehouse domein, hetzij een uitbreiding of andere toepassing van dergelijke best practices. De uitdaging zit het er echter in om deze mechanismen op elkaar te superponeren teneinde de verschillende functionaliteiten tegelijk te kunnen ondersteunen.
In de komende secties zullen op incrementele wijze de verschillende mechanismen met elkaar worden geïntegreerd om via een logisch, begrijpbaar pad te komen tot een globale structuur die de verschillende vereisten ondersteunt.

2.4.2 Sleutelbeheer van de identificatie

De mechanismen en processtromen van de identificatieprocedure zijn beschreven in sectie 1.1.1. De tabelstructuren nodig voor deze procedures zullen echter hier reeds beschreven worden aangezien zij de structurele basis vormen van de volledige versleuteling.

De logische vorm van de tabellen is voorgesteld in onderstaande Figuur 4: Identificatietabellen. Hierin vinden we de volgende tabellen terug:

- A0102_EXTERNE_IDENTIFICATIE_MANUEEL_PERSONNE: Bewaart de hercombinaties van initiële technische sleutels onder een nieuwe technische sleutel teneinde automatische identificaties manueel te kunnen herclassificeren.
    - PTK: Initiële primaire technische sleutel (aangemaakt door de automatische identificatie).
    - PTKM: Technische sleutel die initieel dezelfde waarde heeft als de PTK maar waarin hij de mogelijkheid heeft verschillende voorkomens van eenzelfde entiteit te hergroeperen.
- A0102_EXTERNE_IDENTIFICATIE_TYPE_WITTE_SLEUTEL: Deze tabel bevat de beschrijvingen van de verschillende witte sleutels die in de bronsystemen voorkomen (Rijksregisternummer, ondernemingsnummer, sociale zekerheidsnummer,…) Samen met de volgende tabel laten deze structuur toe om primaire en alternatieve sleutels die betrekking hebben op dezelfde persoon met elkaar in relatie te brengen.
- A0102_EXTERNE_IDENTIFICATIE_WIT_PERSONNE: Tabel waarin de verschillende witte sleutels bewaard worden voor alle personen en alle types van witte sleutel. Met betrekking tot persoon is deze versleuteld op PTK (Primaire Technische Sleutel), dit komt overeen met een strikte identificatie (DQ1) van de persoon.
- A0102_EXTERNE_IDENTIFICATIE_PERSONNE: Centrale unieke persoonsidentificatietabel, primair versleuteld op PTK en het centrale deel van de identificatie.
- A0102_EXTERNE_IDENTIFICATIE_GRIJS_PERSONNE: Tabel waarin de mapping tussen de technische sleutel en de grijze sleutelvelden gemaakt wordt voor een bepaalde datakwaliteitsniveau. Voor kwaliteitsniveaus DQ4 en hoger (slechtere kwaliteit) wordt ook telkens een nieuwe technische sleutel aangemaakt bij eenzelfde technische sleutel teneinde geen onherroepelijke koppelingen in het DWH in te brengen.
    - TK: Technische versleuteling van de identificatie om eventuele verkeerdelijk toegekende PTK’s naderhand nog van elkaar te kunnen onderscheiden zonder dat hiervoor gegevens in het DWH opnieuw moeten geladen worden.
    - PTK: Initiële primaire technische sleutel (aangemaakt door de automatische identificatie).
    - N_I_QUAL_NIV: Datakwaliteitsniveau van de identificatie waarop deze PTK en TK werden toegekend.



Figuur 4: Identificatietabellen

Om te voorzien in het slechts tijdelijk geldig zijn van bepaalde sleutelwaarden, word in de sleutelhoudende tabellen telkens C_DATE_DEBUT_TK en C_DATE_FIN_TK toegevoegd.
De tabel A0102_EXTERNE_IDENTIFICATIE_GRIJS_PERSONNE bevat geen grijze sleutelvelden, zij heeft hier de vorm van een abstracte (of dummy) tabel en de functionaliteit die zij moet ondersteunen (het bewaren van de grijze sleutelvelden) wordt overgeërfd door de 2 tabellen
A0102_EXTERNE_IDENTIFICATIE_GRIJS_PERSONNE_PHYSIQUE en
A0102_EXTERNE_IDENTIFICATIE_GRIJS_PERSONNE_MORALE die bijgevoegd zullen worden in sectie 2.4.4 Sleutelbeheer van Klasse-extenties.

Structureel is deze versleuteling een toepassing van surrogate key mechanismen zoals deze standaard in datawarehousing gebruikt worden met dit onderscheid dat de verschillende records met eenzelfde PTK maar een verschillende TK zich niet van elkaar onderscheiden op basis van het wijzigen van een veld (zoals overwegend gebruikt in een DWG) maar op basis van verschillende grijze sleutelvelden en een verschillende kwaliteitsniveau van de identificatie.

2.4.3 Sleutelbeheer van traag wijzigende dimensies

Volgens de business-vereisten moet het mogelijk zijn om historiek verschillend bij te kunnen houden per veld. Op het gebied van informatie-inhoud kunnen we deze terugbrengen tot een systeem van de type 1 en 2 traag wijzigende dimensies waarmee de drie mogelijke historiektypes ondersteund worden

- Type 1: Er wordt geen historiek bijgehouden, het bestaande record wordt overschreven met de nieuwe waarden.
- Type 2: Er wordt een volledige historiek bijgehouden, elke wijziging resulteert in een nieuw record, versleuteld door een nieuwe technische sleutel en met indicatie van de geldigheid doorheen de tijd van dit record.
- Type 3: Er kunnen ook nog vereisten bestaan waarin men specificeert dat enkel de laatste 2 of 3 waarden bijgehouden dienen te worden of enkel de oorspronkelijke en de laatste maar de informatie noodzakelijk om een dergelijke behoefte te ondersteunen zit ook vervat in een type 2 historiek zodanig dat deze type 3 structuren niet in het DWH zullen opgenomen worden. Indien gewenst kunnen deze in de datamarts opgebouwd worden vanuit de type 2 historiek.

In traditionele DWH-modelleringen wordt er voor elke dimensie één van deze types gekozen. Gezien de iteratieve aanpak van dit project en het niet strikt vastliggen van de informatiebehoeften worden er hier voor geopteerd om in het geval dat er een bepaalde historiek vereist is, deze te implementeren als een type 2 dimensie (T2). Om te vermijden dat er voor de velden waarvoor er geen historiek vereist is te veel plaats wordt verloren door ze telkens te kopiëren in een type 2 record wordt er voor elke entiteit ook een type 1 tabel (T1) voorzien.

Voor de volledige tijdsversleuteling worden er voor elke entiteit (vb. Persoon) 3 tabellen voorzien:

- T1: Tijdsonafhankelijke tabel waarin de velden worden opgeslagen waarvoor er geen historiek vereist is.
- T2: Tijdsafhankelijke tabel waarin de velden worden opgeslagen waarvoor er een historiek vereist is.
- ACT: Tabel gegenereerd vanuit de vorige 2 waarin de laatste toestand wordt opgeslagen. Deze bevat zowel de velden uit T1 als T2 maar voor deze laatste enkel de huidige waarde.

Wat betreft versleuteling zou de T1-tabel, in normale omstandigheden, als primaire sleutel de PTK van de entiteit krijgen. In dit geval, ten gevolge van de meervoudige identificaties wordt deze rol overgenomen door de technische sleutel TK uit de vorige stap wat niet wegneemt dat het toch interessant is om hier ook de functionele sleutel (PTK) mee te nemen maar men moet er zich dan wel van bewust zijn dat deze dan niet meer uniek is over deze tabel.

De T2-tabel, die de volledige historiek van de entiteit bijhoudt, heeft als primaire sleutel een TKT2 (Technical Key Type 2) en als vreemde sleutel de TK. Ook hier wordt (teneinde “shortcut joins” te voorzien) PTK mee opgenomen door ze eenvoudigweg doorheen de structuur naar beneden te propageren. Deze tabel heeft bovendien een vreemde sleutel naar de datumdimensie D_Datum om de geldigheid van het record aan te duiden. Er wordt maar 1 link naar deze dimensie voorzien aangezien dit vanuit een informatietechnisch standpunt voldoende is (start- en einddatum aangeven is redundant).

De Actual-tabel heeft een 1-op-1 relatie met de T1-tabel en ook dezelfde primaire sleutel (TK). Daarnaast worden hier ook weer de PTK’s (volledig dekkend) en de TKT2 (niet dekkend) opgenomen voor het implementeren van de “shortcut joins.” Dit levert in feite geen actual op niveau van persoon maar wel op niveau van de doublures van de grijze sleutels.

Dit model voor de opbouw en naamgeving van de verschillende tabellen die bij een klasse behoren wordt niet voor alle klassen in zijn volledigheid geïmplementeerd. De regels hier zijn:

- Voor echte dimensies, hiermee wordt bedoeld met een echte eigen identiteit (Persoon, NatuurlijkPersoon, Rechtspersoon, Vestiging) worden de drie tabellen geïmplementeerd.
- Voor referentietabellen beperken we ons tot de implementatie van de T2-tabel aangezien er voor deze klassen geen T1 behoefte bestaat. Zij worden dus geïmplementeerd als klassieke SCD’s van type 2.

- Voor brug en feitentabellen (A0108_ActiviteitsDomein, A0104_GebruikContact, …) waarvan de records op zich geen eigen identiteit hebben wordt afgestapt van bovenstaand principe aangezien het niet opportuun lijkt hier een primaire sleutel voor te voorzien aangezien dit een identiteit zou impliceren.

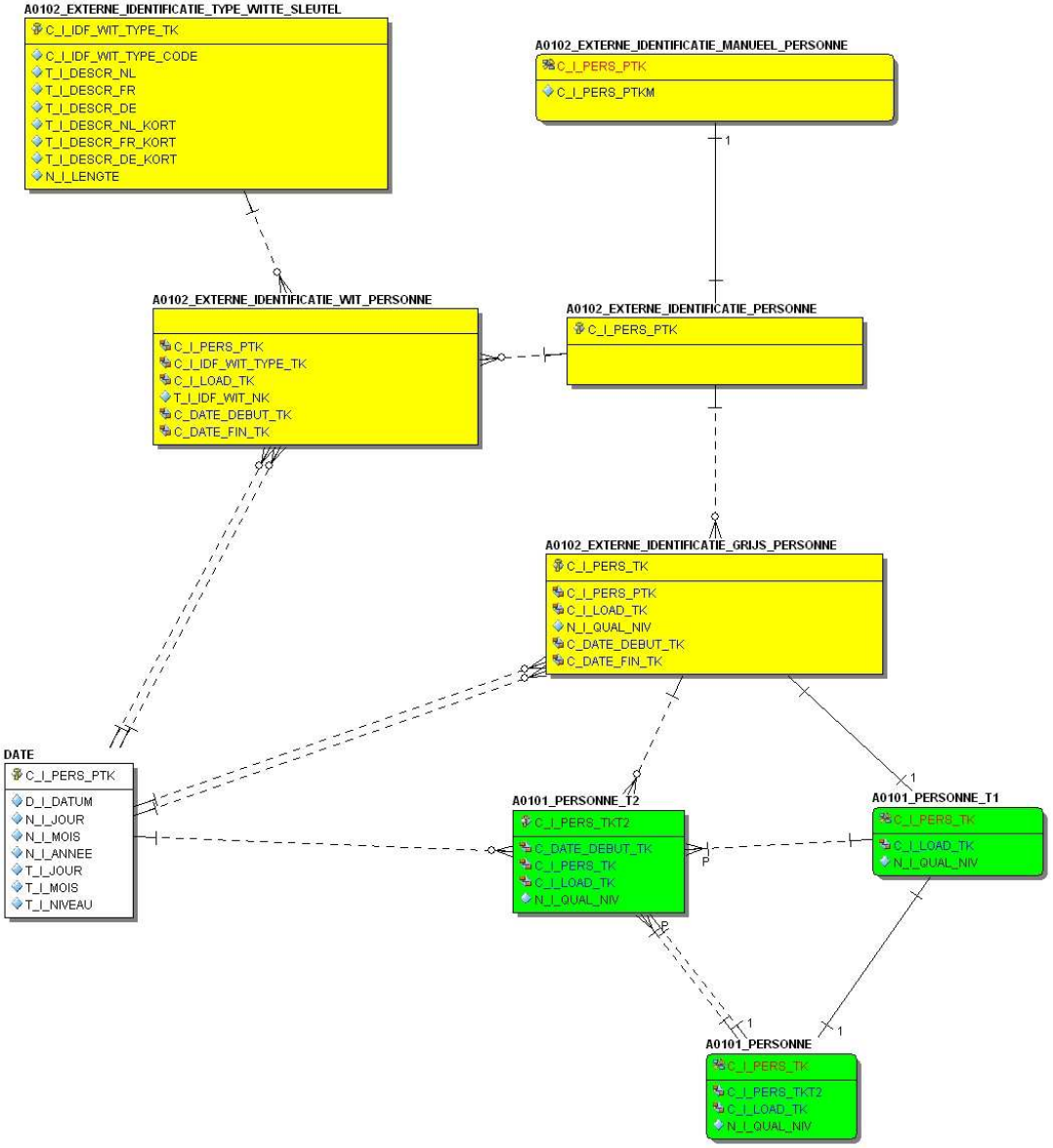


Figure 5: Historiektabellen

2.4.4 Sleutelbeheer van Klasse-extenties

Binnen het model wordt er op een aantal plaatsen gebruik gemaakt van Klasse-Extensies (of specialisaties in zuivere OO termen). Dit onder meer op het gebied van de klassen Persoon, NatuurlijkePersoon, Rechtspersoon en de volledige opsplitsing van contacttypes. Een instantie van de klasse persoon kan dus maar ofwel een

Rechtspersoon, ofwel een Natuurlijk Persoon zijn elk met zijn eigen signatuur waartussen er zo goed als geen overlap bestaat op attribuut-niveau (Persoon kan in OO-termen gezien worden als abstracte klasse).
Het is echter wel zo dat er een hoop entiteiten bestaan die zowel betrekking kunnen hebben op een Natuurlijk Persoon als op een Rechtspersoon en dus in het domeinmodel gerelateerd zijn aan de entiteit Persoon. Een voorbeeld hiervan is de brugtabel GebruikContact die de persoonsgegevens van zowel Natuurlijke Personen als Rechtspersonen koppelt aan contactgegevens.

Om deze entiteiten transparant met elkaar te kunnen koppelen zonder verschillende tabellen te moeten voorzien voor bijvoorbeeld contacten van natuurlijke dan wel rechtspersonen, worden de technische sleutels voor klassen die extensies zijn van éénzelfde parent-klasse gegenereerd vanuit dezelfde sleutelruimte.

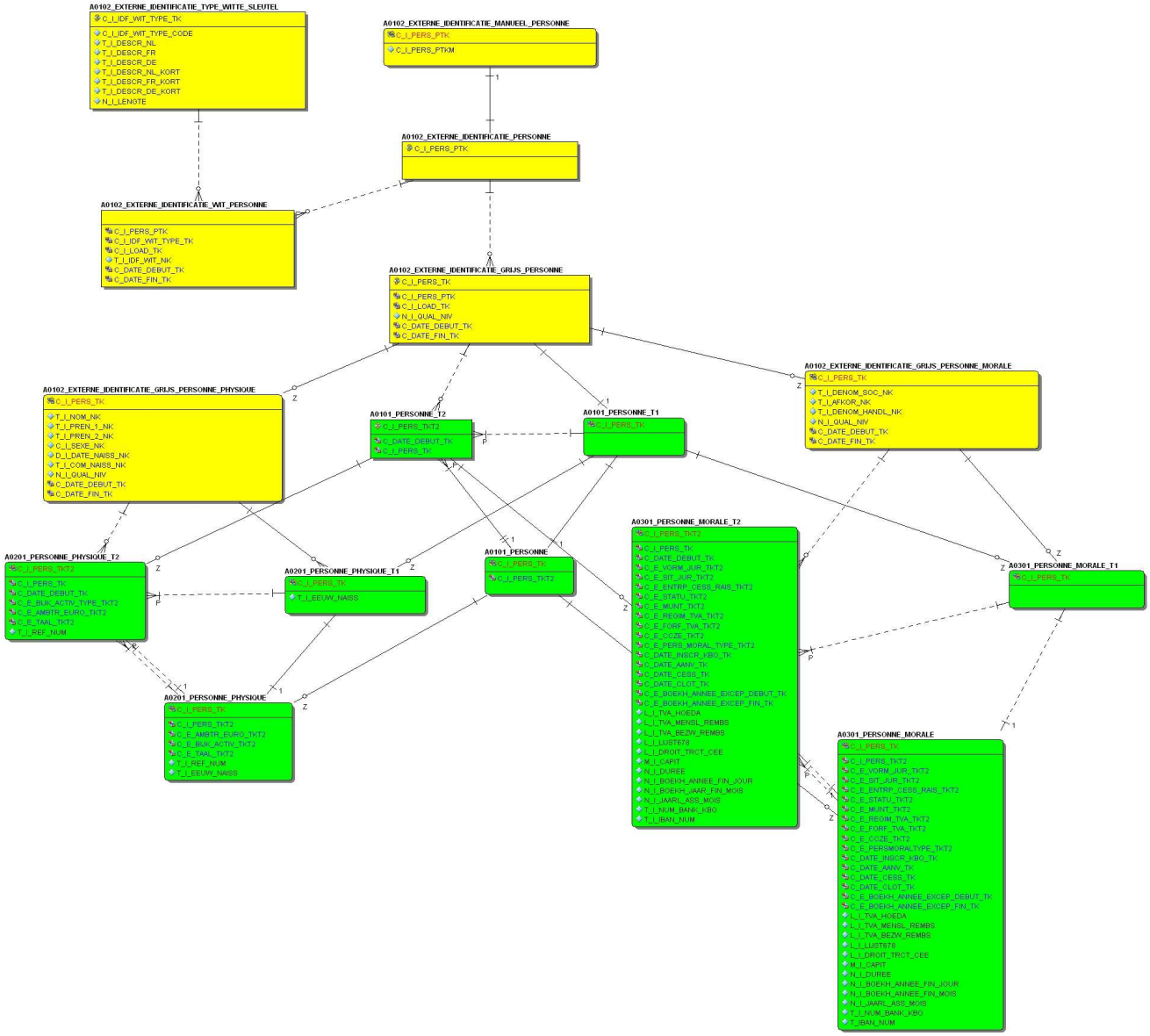


Figure 6: Overzicht Sleutelbeheer

Dit betekent dat er telkens tussen de Parent-tabellen en hun extensies een “one to zero or one” relatie bestaat die exclusief is tussen de verschillende subtabellen (een TK of TKT2 die in een natuurlijk persoon-tabel voorkomt mag niet in een rechtspersoontabel voorkomen en omgekeerd).
Het is ook hier dat de tabel A0102_EXTERNE_IDENTIFICATIE_GRIJS_PERSOON wordt ontdubbeld in de 2 tabellen voor natuurlijke en rechtspersonen die dan wel de respectievelijke grijze sleutelvelden bevatten die niet aanwezig zijn in de parent-tabel.

Het is ook hier waar de anonimisatie geïmplementeerd wordt. Zelfs wanneer de toegang tot de A0102-tabellen afgesloten is is men nog steeds in staat uitgebreide analyses te doen op de in het DWH opgenomen gegevens. Zonder toegang tot de A0102_EXTERNE_IDENTIFICATIE_GRIJS_* tabellen kan men deze echter nooit herleiden tot effectieve personen.

2.4.5 Beheer en opbouw van de tijdsdimensie

In de bovenstaande secties werd de tabel D_Datum geïntroduceerd. In datawarehouses is het gangbaar om de tijd als aparte dimensie te zien in plaats van tijdsaspecten als attributen van een tabel te zien. Dit heeft als voornaamste voordelen dat men enerzijds verschillende feitentabellen makkelijker en structureel uniformer met elkaar in relatie kan brengen en men anderzijds in deze tijdsdimensie een aantal decoratieve velden (d.w.z. velden die een instantie van een entiteit verder omschrijven zonder dat ze als identificiatieveld kunnen dienen of als meetwaarde geïnterpreteerd kunnen worden) expliciet kan opnemen zodanig dat er tijdens bevraging geen datum-transformatie functies van de databank moeten aangeroepen worden, wat de performantie zeer ten goede komt. Deze decoratieve velden laten ook toe andere interpretaties dan kalenderjaar (vb. boekjaar, aanslagjaar) transparant op te nemen. Een nadeel is echter dat tijdens het laadproces alle datum vertaald (via lookup) moeten worden naar de technische sleutel van de tijdsdimensie.

Om zo veel mogelijk voordelen van een tijdsdimensie te bewaren en de nadelen te omzeilen zal in dit project de datum-dimensie gecodificeerd worden met een functionele sleutel wat wil zeggen dat de primaire sleutel van de dimensie is geen pure (interpretatieloze) technische sleutel. Deze wordt opgebouwd als JJJJMMDD en heeft dus een functionele interpretatie. Dit is de enige uitzondering in datawarehousing waar een dergelijke functionele betekenis aan een technische sleutel gegeven wordt met als belangrijkste voordeel dat tijdens het laadproces de versleuteling van de datum rechtstreeks kan gebeuren, zonder look-up in de (grote) dimensietabel.
Daarnaast wordt deze tabel geïmplementeerd als een hybride dimensie waarin de verschillende niveaus (jaar, maand, dag) van de tijdshiërarchie transparant en door elkaar in de dimensie opgenomen worden. Dit heeft als voordeel dat alle feitentabellen ongeacht hun niveau van aggregatie naar dezelfde tabel kunnen verwijzen en er dus geen voorkennis nodig is van op welke granulariteit bepaalde gegevens beschikbaar zijn.

De volgende velden zullen al zeker opgenomen worden in de datumdimensie maar het is niet uitgesloten dat er hier in functie van de business-vereisten nog zullen bijgevoegd worden.

Veldnaam	Domeintype	Omschrijving
C_I_DATE_TK	Numeriek	Technische (functionele) primaire sleutel
niveau	Numeriek	Duidt aan op welk niveau dit record zich bevindt (0: dag, 1:maand, 2:jaar)
datum	Datum	De effectieve datum in datumformaat voor niveaus 1 en 2 wordt de eerste dag van de periode genomen.
jaar	Numeriek	Het jaar in cijfers
Kwartaal	Numeriek	Kwartaal in het kalenderjaar
maand	Numeriek	De maand in cijfers
dag	Numeriek	De dag in cijfers
dagenSinds19000101	Numeriek	Het aantal dagen sinds 1 januari1900 voor tijdsverschilberekeningen

		in kalenderdagen.
dagInDeWeek	Numeriek	0: zondag, 1: maandag etc...

Voor redenen van opslag worden de drietalige beschrijvingen van de dagen van de week en de maanden opgeslagen in 2 aparte tabellen (snow-flakes) gecodeerd op de velden maand en dagInDeWeek. Deze tabellen hebben geen geldigheidsdata of expliciete technische versleuteling zoals andere look-ups.

Indien het noodzakelijk blijkt een lager niveau van granulariteit in de tijd te bewaren voor bepaalde gegevens dan zullen deze in een onafhankelijke tabel D_Tijd opgeslagen worden opgeslagen die weer volgens de zelfde principes opgebouwd zal worden.

2.4.6 Datatypes

Om een uniformiteit te bewaren overheen het DWH zal er een onderscheid gemaakt worden tussen het logische datatype van een veld (ook wel domeintype genoemd) en het fysische datatype. De bedoeling is het aantal verschillende fysische datatypes te beperken om geen onoverzichtelijke wildgroei te bekomen. Als domeintypes kunnen reeds de volgende definiëren:

- Datum
- Datum-Tijd
- Lange String
- Middellange String
- Korte String
- Bedrag
- Klein natuurlijk getal
- Groot natuurlijk getal
- Reëel getal
- Technische sleutel

Deze lijst is nog niet exhaustief, verdere onderverdeling en keuze van het bijhorende fysische type kan slechts gebeuren na de gedetailleerde analyse, wanneer de exacte inhoud van de brongegevens bekend is.

2.4.7 Keuze van de velden

Welke velden er geïmplementeerd dienen te worden en of er hiervan een historiek dient bijgehouden te worden wordt vastgelegd door de busines-gebruikers op basis van de brondata-analysedocumenten. Dit bepaalt de uiteindelijke plaatsing van de gegevens (T1 of T2) en de benodigde referentietabellen. Velden die reeds in de voorstudie gedefinieerd werden maar die niet gevonden werden in één van de beschikbare bronnen worden niet meegenomen.

2.5 Beheer en opvolging van de gegevens.

2.5.1 Overzicht

Dit hoofdstuk probeert een overzicht te geven van de mechanismen die gebruikt zullen worden voor het beheer van de gegevens in het DWH. Hierbij zullen een aantal doelstellingen geformuleerd worden (eventueel uitgaande van een aantal reële probleemstellingen) waarna deze mechanismen zullen worden voorgesteld en uitgediept.

We kunnen de doelstellingen voor beheer opdelen in 3 grote categorieën:

- Beheer van jobs, sequenties, en events.

- Welke events (binnenkomende data, gefaalde jobs of sequenties, …) hebben zich voorgedaan en wat was de actie die erop ondernomen werd? Hierin wordt ook begrepen welke regelmatig terugkerende taken (evt. voor onderhoud) er gepland zijn.
- Welke sequenties zijn er gestart t.g.v. welke events (t.g.v. binnenkomende gegevens of gescheduled)
- Welke jobs werden er gestart vanuit deze sequenties en zijn deze correct afgerond.

- Beheer van datastromen
  - In de ETL-ketting worden er een aantal signaturen gedefinieerd van verschillende types (bron-, communicatie- , look-up- en target-signaturen). Om het DWH en de ETL-ketting op ieder moment een recovery en restartability is het noodzakelijk een volledig overzicht te bewaren welke jobs en sequences op welk moment welke nieuwe gegevens hebben geïntroduceerd om ervoor te zorgen dat wanneer zich een fout voordoet in één van de laadprocessen, deze gegevens quasi chirurgisch kunnen verwijderd worden zonder dat het noodzakelijk is het warehouse van nul terug op te bouwen. Indien er zich een fout voordoet tijdens één van de laadprocessen zelf laat het beschreven mechanisme ook toe deze, slechts gedeeltelijk geladen, gegevens te verwijderen uit het DWH alvorens het laadproces opnieuw op te starten.

- Beheer van datakwaliteit
  - Aangezien het DWH in de eerste plaats opgezet wordt voor risicobeheer zullen er geen (of zo weinig mogelijk) gegevens geweigerd worden, hoe slecht hun kwaliteit ook is, aangezien het feit dat de kwaliteit van de gegevens slecht is op zich ook reeds een indicatie voor een bepaald risico is. Om niet te hervallen in een "Garbage In Gargbage Out" systeem zal er aan alle opgenomen gegevens een kwaliteitsindicator aangegeven worden zodanig dat de eindgebruiker goed op de hoogte is van de betrouwbaarheid van de ter beschikking gestelde gegevens. Hierin kunnen we verschillende types van indicatoren onderscheiden:
    - Indicatoren met betrekking tot de kwaliteit van de identificatie. Hiervoor verwijzen we naar de verschillende DQ's zoals gedefinieerd in sectie 2.3.4 Identificatie. Deze kwaliteitsindicatoren worden vanuit de identificatietabellen ook verder gepropageerd naar de onderliggende (dochter-) tabellen om deze kwaliteitsindicatoren makkelijk ter beschikking te stellen. Indien er gegevens geaggregeerd worden is het aan de business-gebruikers om te definiëren hoe deze indicatoren dienen te propageren. Het meest veilige voorstel is de indicator van de slechtste kwaliteit over te nemen
    - Indicatoren met betrekking tot de volledigheid van informatie. Het is mogelijk dat gegevens rond een bepaalde entiteit in een eerste iteratie enkel beschikbaar zijn vanuit een secundaire bron van waaruit enerzijds geen witte identificatie mogelijk of anderzijds niet alle decoratieve velden voor de bijhorende dimensie beschikbaar zijn (vb. BIS-register gegevens die eerst worden waargenomen in het KBO en pas in een latere iteratie worden verrijkt met gegevens van het BIS-register zelf). Om dit aan te duiden wordt er in de T1, T2 en ACT tabellen ook een vlag "Indicatie van volledigheid" opgenomen. Deze indicator zal de verhouding weergeven tussen de ingevulde (not NULL) velden tegen het totaal aantal velden.

2.5.2 Traceren van de gegevens, jobs en sequences.

Omwille van de enorme volumes die typisch in een DWH voorkomen en de vereisten van beschikbaarheid is het onaanvaardbaar om, in het geval er zich een fout voordoet in de ETL-ketting, de gegevens volledig opnieuw te laden. Dit geldt niet alleen voor fouten zoals het niet juist beïndigen van een job maar omwille van de complexe technische versleuteling ook voor de gevallen waarin een logische fout zich voordoet zonder dat dit als resultaat heeft dat een job crasht maar waardoor de integriteit van de versleuteling verloren gaat. Aangezien er, omwille van performantievereisten, geen roll-back op databank-niveau wordt voorzien zou men in zo'n geval al bijna moeten teruggrijpen naar het volledig restoren van de databank. Om dit te vermijden wordt restartability ook voorzien op het inhoudelijke niveau van de gegevens door middel van traceren van alle binnenkomende data.

Voor het traceren van de gegevens wordt er gestart vanuit het concept van de “load.” Onder een load wordt verstaan: De volledige verzameling van enerzijds brongegevens en communicatie- en targetsignaturen, en anderzijds de uitvoeringen (runs) van de jobs en sequences die deze gegevens verwerken voor een bepaalde batch van binnenkomende brondata van één bron. Dit wil zeggen, het nieuw binnenkomende gegevensbestand voor een bron en alle achterliggende verwerkingen en tussentijds en finaal opgeslagen gegevens, hetzij in DataSets, hetzij in db2-tabellen.

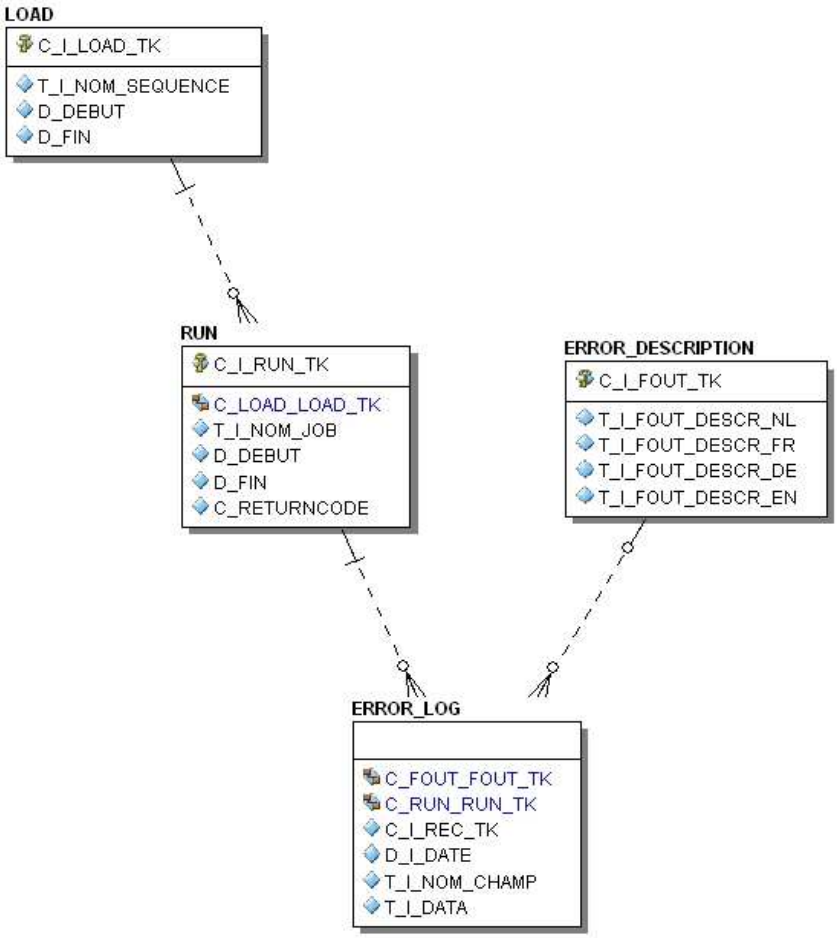


Figure 7: Beheerstabellen

Om alle gegevens betreffende uitgevoerde loads, volumes van signaturen en foutmeldingen van jobs en sequences op te slaan wordt er een systeem voorzien van beheerstabellen waarin exact bijgehouden wordt welke job op welk moment welke signatuur heeft gegenereerd met daarbij bijkomende gegevens m.b.t. het aantal lijnen en/of gegenereerde foutmeldingen. Figure 7: Beheerstabellen geeft een insteek naar deze structuur, die in de technische documentatie en implementatie verfijnd zal worden.

De gegevens zelf zullen in dit kader bij het binnenkomen in de ETL-ketting direct een C_I_LOAD_TK krijgen die identificeert waar ze van afkomstig zijn. Daarnaast zullen de gegevens binnen een signatuur ook een C_I_REC_TK krijgen waardoor ze op verschillende wijzen gesplitst (voor transformatie of verschillende types van identificatie) kunnen worden om naderhand terug samen te voegen voor het laden. Eens de gegevens terug samengevoegd (op het moment van het laden) verdwijnt het C_I_REC_TK weer maar de C_I_LOAD_TK wordt

uiteindelijk weerhouden in het warehouse. Het beheren van deze C_I_LOAD_TK en de mechanismen noodzakelijk om dit geheel consistent te houden komen overeen met de aspecten rond het toekennen van kwaliteitsindicatoren en worden beschreven in de volgende sectie. De gegevens (fouten) verzameld in deze tabellen dienen als uitgangspunt voor het genereren van feedback naar de oorspronkelijke bron van de gegevens.

2.5.3 Toekennen en beheren van kwaliteitsindicatoren.

Zoals vermeld in sectie 2.5.1 zullen alle gegevens die in het DWH belanden een kwaliteitsindicator toegewezen krijgen. Het toewijzen van deze indicatoren gebeurt enerzijds op basis van de kwaliteit van de matching zoals uiteengezet in sectie 2.3.4.2 rond Automatische identificatie en anderzijds kan dit nog gebeuren op basis van de volledigheid van de gegevens.

Om deze kwaliteitsniveaus (en volgens hetzelfde systeem de C_I_LOAD_TK's) te kunnen beheren en om ook op consistente wijze enerzijds loads te kunnen verwijderen uit het datawarehouse en anderzijds datamarts te bouwen met een opgelegde minimum kwaliteit moeten we de verschillende soorten gegevens in acht nemen:

- Identificatiegegevens: Hier wordt er geen probleem verwacht aangezien er geen updates noch deletes van bestaande records uit de identificatietabellen gebeuren tijdens het verwerken van identificaties (enkel inserts). Zodoende kunnen deze tabellen in hun vorige vorm hersteld worden door enkel de gegevens van een bepaald load-ID te verwijderen.
- Decoratieve gegevens (waaronder in deze fase ook de links naar de referentietabellen):
    - Voor de type 1 gegevens, die dus telkens overschreven worden door een nieuwe waarde dient men erop te letten dat er hier een strenge back-up policy wordt toegepast teneinde, in het geval oude gegevens overschreven worden het nog steeds mogelijk blijft het warehouse, in het geval dat de load onsuccesvol was, in zijn oorspronkelijke staat terug te brengen.
    - Voor type 2 gegevens is het mogelijk zoals voorzien telkens een nieuw record te inserteren. Dit record kan dan een lagere kwaliteitswaarde hebben en tijdens het bouwen van de DataMarts kan er dan nog gekozen worden welke records al dan niet worden meegenomen. Het feit dat er hier in een meer Inmon-stijl gewerkt wordt laat ook toe om van een T2-record enkel de datum van waarneming mee te nemen ipv daarnaast ook nog eens de einddatum (datum van volgende waarneming) aangezien het mogelijk is deze twee te hergenereren (de informatie is aanwezig) bij het uitrollen van de datamart. Dit laat ons toe een type 2 dimensie op te bouwen waarin er gegevens van verschillende kwaliteit gestockeerd zijn en waarbij we bij uitrol de vereiste kwaliteit kunnen kiezen. Ook laat het ons toe in een volledig insert-of-delete scenario te werken zonder dat er voor het onderhoud van deze tabel update-statements dienen uitgevoerd te worden. Een derde voordeel van deze aanpak is dat deze het zonder meer toelaat om aanpassingen te doen aan gegevens uit het verleden (ook weer door eenvoudigweg te inserteren zonder de noodzaak aan 2 trage selects en 2 updates en "between"-queries).
- Feitengegevens. Ook hier wordt een kwaliteitsindicatie bepaald op basis van de kwaliteit van de match aangezien er in het geval van feiten nooit andere gegevens overschreven worden. Ook hier kan er gewerkt worden in een insert-of-delete scenario zonder ons zorgen te moeten maken over de impact van updates.

Indien verscheidene kwaliteitsindicatoren gecombineerd dienen te worden in één veld is het steeds de business die het laatste woord heeft. Er word echter voorgesteld om voor de veiligheid telkens de slechtste van de twee kwaliteitsindicatoren over te nemen als kwaliteit van het samengestelde record.