



# PARIS-release van OWB

## Analyse van een bèta-versie

**Oracle Warehouse Builder (OWB) heeft een wat ongelukkige naam. Het is een tool waarmee zoveel meer kan dan alleen het ontwerpen en genereren van een datawarehouse. In twee artikelen gaan we kennismaken met deze nieuwste versie van OWB, die volgens de huidige planning in augustus in productie gaat. In dit eerste artikel kijken we naar het ontwerp van een eenvoudig ETL-proces (Extract, Transform en Load) om via een klein datawarehouse tot een eenvoudig management-informatiesysteem te komen.**

We gaan uit van brondata in een Oracle-database en een Excel-file. Vanuit dat vertrekpunt ontwerpen we een target-database, de transformaties om de data vanuit de bron naar deze database te laden en transformeren, en een kleine BI Beans-applicatie om enkele analyses uit te voeren en concrete vragen te beantwoorden. De naam is er waarschijnlijk ook de reden van dat OWB vrij onbekend en dus onbemind is gebleven, terwijl het een krachtig tool is voor ontwikkelaars die uiteenlopende taken moeten verrichten: van het maken van een database-ontwerp tot het ontwikkelen van code voor dataconversie, van het ontwerpen van Discoverer of BI Beans-applicaties tot het bouwen van zeer efficiënte processen voor het laden van data uit externe bronnen. Daarnaast bevat de nieuwste release van OWB - 10gR2 - de Data Profiler, een tool voor met name informatie-analisten of technisch ontwerpers, om kwalitatieve en statische analyse te doen van data in een database. Uit de analyse kunnen business-rules geëxtraheerd worden; ook kan OWB code genereren om data op te schonen op basis van de business-rules en correctieregels.

OWB maakt deel uit van de Oracle 9i en 10g Developer Suite, waar ondermeer ook Oracle Designer, Forms, Reports, Discoverer en JDeveloper in zitten. Veel organisaties hebben het tool dus al in huis, zonder het waarschijnlijk te gebruiken of zelfs maar te weten dat het er is. Ooit begonnen onder de naam Data Mart Suite bestaat het product in verschillende vormen al vele jaren. De 9i release in 2002 en de 10gR1 release van vorig jaar hebben OWB naar een heel volwassen niveau

gebracht, vergelijkbaar met concurrerende producten van met name Informatica en Business Objects. Sinds december 2004 is de zogenaamde Paris release van Oracle 10g Warehouse Builder in bèta. Deze nieuwe release is twee jaar lang onderwerp geweest van opgewonden marketing boodschappen vanuit Oracle. Er wordt erg veel interessante nieuwe en verbeterde functionaliteit in het vooruitzicht gesteld.

### Beschrijving casus

Sinds begin juli 2004 publiceert AMIS een Technology Weblog met artikelen rond Oracle- en Java-technologie. Op deze weblog publiceren medewerkers van AMIS over hun ervaringen in de praktijk of in onderzoek met nieuwe tools maar ook bestaande technologie. Allereerst om onderling als collega's kennis uit te wisselen, maar gaandeweg ook om relaties en de rest van de internetgemeenschap te informeren. We hebben inmiddels een kleine 400 artikelen gepubliceerd die bij elkaar meer dan 250.000 keer gelezen zijn door meer dan 50.000 verschillende bezoekers. Het is geen niemendalletje meer en we kunnen niet zomaar besluiten even een upgrade te doen en domweg een dag uit de lucht te zijn. Ook de commerciële



Figuur 1. Sinds begin juli 2004 publiceert AMIS een Technology Weblog met artikelen rond Oracle- en Java-technologie.

	A	B	C	D	E
45635	3563097248	3563097311	DE	DEU	GERMANY
45636	3563097312	3563097343	GB	GBR	UNITED KINGDOM
45637	3563097344	3563097727	DE	DEU	GERMANY
45638	3563097728	3563097791	US	USA	UNITED STATES
45639	3563097792	3563102207	DE	DEU	GERMANY
45640	3563102208	3563110399	CZ	CZE	CZECH REPUBLIC
45641	3563110400	3563118591	RU	RUS	RUSSIAN FEDERATION
45642	3563118592	3563126783	DE	DEU	GERMANY
45643	3563126784	3563134975	HU	HUN	HUNGARY
45644	3563134976	3563143167	IT	ITA	ITALY
45645	3563143168	3563151359	GB	GBR	UNITED KINGDOM
45646	3563151360	3563159551	DE	DEU	GERMANY

Figuur 2. Deze IP-to-country database is een CSV-file die IP-ranges koppelt aan landen.

potentie neemt toe, zeker ook gericht op het buitenland. Daarnaast is het natuurlijk voor AMIS'ers een sport om een artikel zo vaak mogelijk gelezen en zo hoog mogelijk in Google te krijgen.

We willen in deze casus een aantal vragen beantwoord krijgen:

- Wat is het beste moment van de dag om onderhand te doen op de website -wanneer zitten we de minste bezoekers dwars?
- Wat is het beste moment van de week om een artikel te publiceren?
- Uit welke landen komen veel van onze weblog-bezoekers?
- Welke bezoekersaantallen kunnen we over pakweg een half jaar verwachten en is onze hardware daar op berekend?
- Via welke sites komen bezoekers naar onze weblog en welke zoektermen gebruiken ze?

Figuur 3. De table-editor en de data-viewer waarmee de data 'live' in de tabel kunnen worden bekeken.

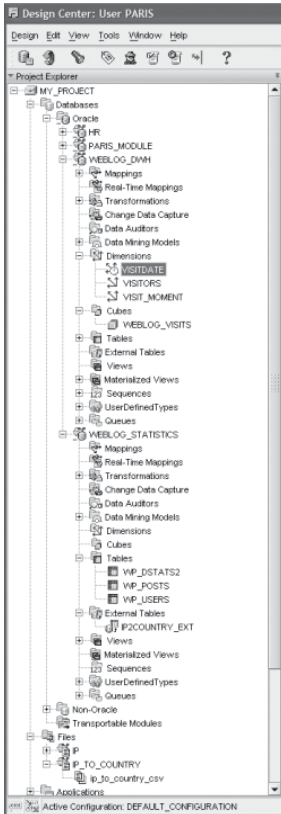
## Registreren van databronnen

Op basis van de operationele statistieken die door de weblog worden bijgehouden kunnen we een kleine datawarehouse vullen waarmee deze query's betrekkelijk eenvoudig kunnen worden uitgevoerd en grafisch gepresenteerd. Inmiddels zijn de aantallen al dusdanig dat een simpele SQL-query direct tegen de statistiek-tabellen niet snel met de antwoorden kan komen. Wel hebben we al enkele live-statistieken tot onze beschikking. We zijn met name geïnteresseerd in de tabellen WP\_POSTS - met alle artikelen, WP\_USERS - met de auteurs van de artikelen - en WP\_DSTATS2 - waarin ieder bezoek aan de weblog wordt geregistreerd, compleet met IP-adres, browsertype, datum en tijdstip. Daarnaast hebben we op internet een IP-to-country database gevonden, een CSV-file die IP-ranges koppelt aan landen. Dat wil zeggen: ieder IP-adres a.b.c.d, bijvoorbeeld 101.54.21.123, wordt omgerekend naar een getal volgens de formule  $256^3 * a + 256^2 * b + 256 * c + d$ . Voor ons voorbeeld is dat dus  $256^3 * 101 + 256^2 * 54 + 256 * 21 + 123$ . Dit getal valt in één van de ranges die met een onder- en bovengrens zijn gedefinieerd in de CSV-file.

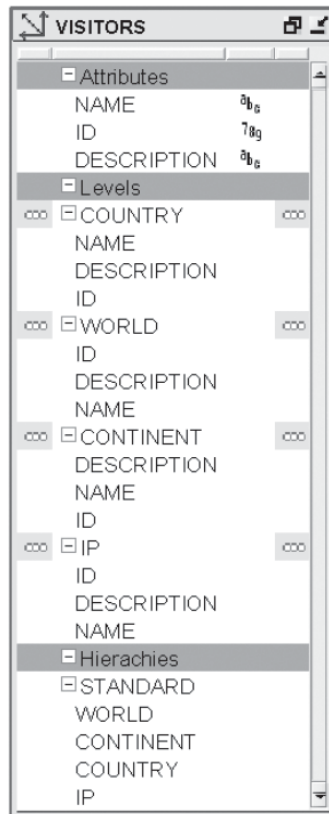
We gaan in Oracle Warehouse Builder onze datasources registreren. Dat is de eerste stap om ze later in het ontwerpproces te kunnen inschakelen in een mapping; een ETL-proces om data uit bronnen via transformaties in doeltabellen te laden. De genoemde tabellen registreren we op normale wijze in OWB door ze te importeren. We kunnen trouwens tabeldefinities ook uit bijvoorbeeld Oracle Designer of andere meta-data repository's inlezen. Andere databronnen in OWB kunnen Views, Advanced Queues, SAP-interfaces, Oracle E-Business Suite-interfaces zijn, maar ook maatwerk-PL/SQL interfaces van eigen applicaties. Daarnaast kunnen files rechtstreeks of via External Tables gelezen worden.

De IP-to-country database wordt als File geregistreerd. Bovendien laten we OWB voor deze file direct een External Table definitie creëren, zodat we SQL-operaties rechtstreeks - zonder tijdelijke tussentabel - tegen de file kunnen uitvoeren. OWB kan de file-definitie samplen. In een eenvoudige wizard geven we configuratiewaarden aan, zoals kolomnamen, records gescheiden door een new-line en velden gescheiden door een komma. Na deze registratie is de bron van ons management-informatie-systeem bekend binnen OWB. We kunnen voor iedere bron gebruikmaken van de Data Viewer om de data rechtstreeks te interpreteren. Ook zouden we de Data Profiler kunnen inzetten om de kwaliteit van de data te analyseren en eventueel te gaan verbeteren. In een volgend artikel zullen we ingaan op dat aspect van OWB. Nu is het tijd om het target te gaan specificeren.

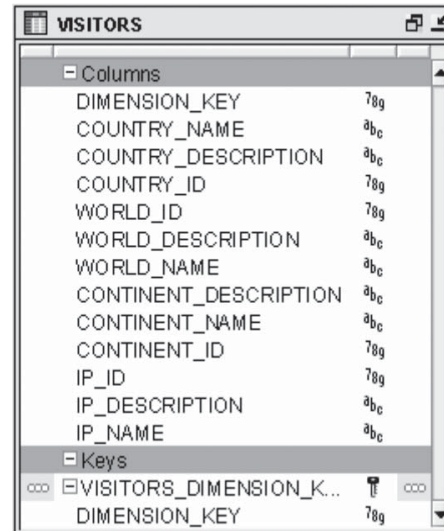
Figuur 3 toont de table-editor en de data-viewer waarmee de data 'live' in de tabel kunnen worden bekeken. In Figuur 4 ziet u de Project Explorer van het OWB Design Center - sterk vergelijkbaar met de Repository Object Navigator in Oracle Designer.



Figuur 4. De Project Explorer van het OWB Design Center.



Figuur 5. Het dimensional ontwerp van de visitors-dimension.



Figuur 6. Het relational ontwerp van de visitors-dimension.

Vanuit deze Explorer zijn alle objecten zichtbaar en kunnen editors, generatoren en tools om te deployen gestart worden.

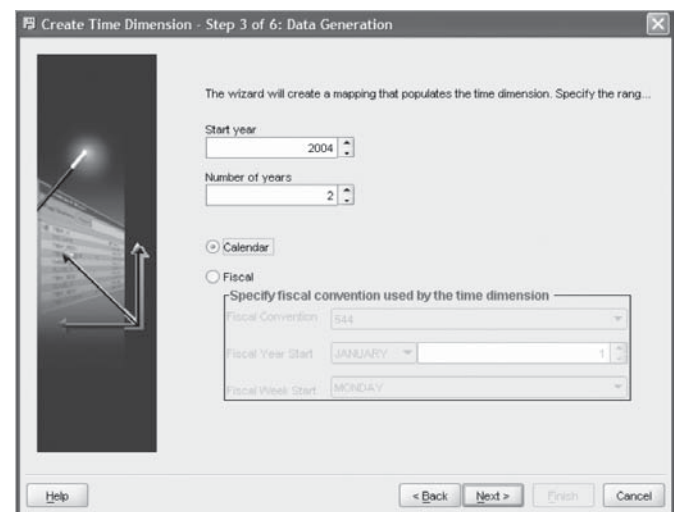
## Ontwerpen van data-targets

Targets in Oracle Warehouse Builder zijn *dimensions*, *cubes* of gewoon relationele tabellen. Om OLAP te ondersteunen - maar ook om *materialized views* effectiever te kunnen benutten via *query rewrite* - is het wenselijk, zo niet noodzakelijk om dimensies en hiërarchieën te definiëren in de doel-omgeving. In ons voorbeeldsysteem hebben we te maken met twee dimensies: de visitor-dimension en de time-dimension. De laatste komt overigens in vrijwel elk systeem voor en kan dan ook door OWB volledig gegenereerd worden, inclusief data. De visitor-dimension kent een hiërarchie met verschillende niveaus. Deze corresponderen met de aggregatieniveaus waarop de rapportage willen gaan doen: *wereld* - *continent* - *land* - *IP*. Met de Dimension Wizard en de Dimension Editor kunnen we deze dimensie ontwerpen in OWB. We geven aan of we een STAR-implementatie (gedenormaliseerd in een tabel) of een SNOWFLAKE-implementatie (genormaliseerd) willen. Vervolgens genereert OWB vanuit de dimensiedefinitie de benodigde tabeldefinitie(s) - als we kiezen voor een ROLAP-implementatie (relationeel). Het alternatief is een MOLAP-implementatie (multi-dimensionaal), in een OLAP Analytical Workspace in de database. In beide gevallen

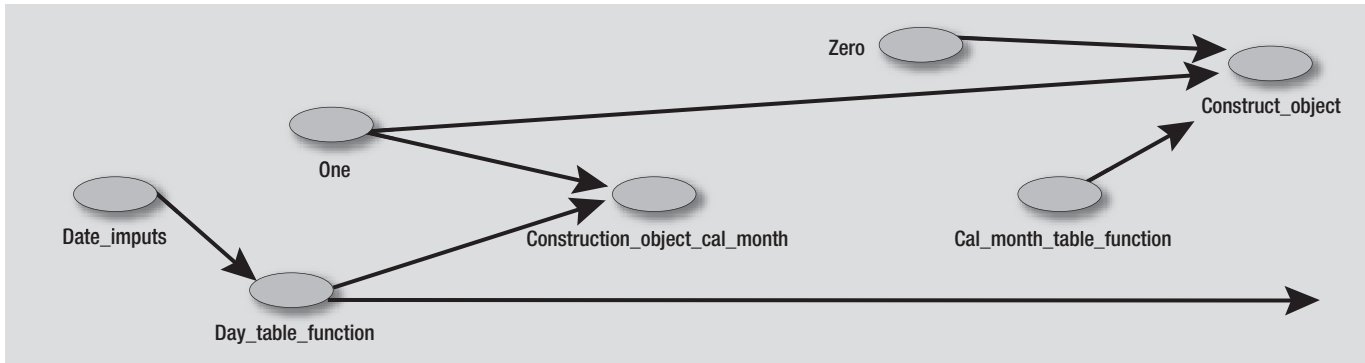
genereert OWB ook de OLAP Catalog meta-data die in de database worden geregistreerd.

Figuur 5 en 6 tonen het *dimensional* en het *relational* ontwerp van de visitors-dimension. De Dimension beschrijving is meer functioneel - deze hebben we in OWB ingebracht. Het tabelontwerp is daarvan afgeleid door OWB zelf en kan naar een DDL-script worden gegenereerd. Met de wizard in figuur 7 kunnen we de time-dimension ontwerpen en genereren. We geven aan om welk soort tijdhiërarchie het gaat, welke niveaus we willen onderkennen (dag, maand, kwartaal, jaar) en welke periode door OWB moet worden geladen. OWB creëert voor deze dimensie niet alleen de tabeldefinitie maar ook een mapping, een PL/SQL-proces om de data te genereren (zie figuur 8).

In onze applicatie kennen we nog een alternatieve 'tijd'-dimensie, namelijk de tijd-van-de-week (VISIT\_MOMENT) dimensie.



Figuur 7. Met de wizard kunnen we de time-dimension ontwerpen en genereren

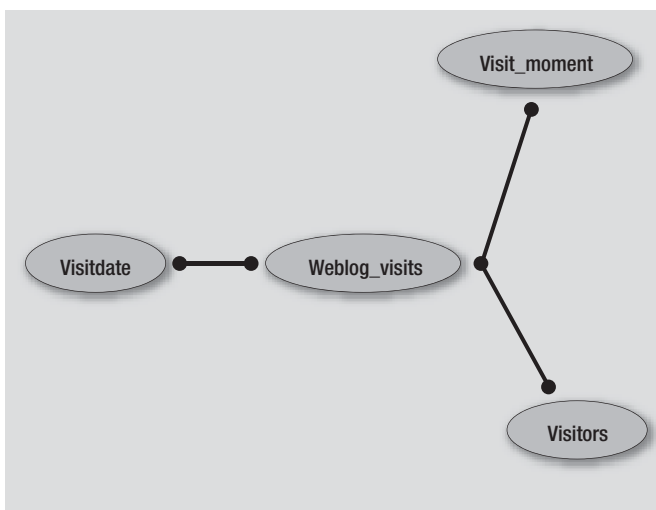


Figuur 8. OWB creëert voor de dimensie een 'mapping': een PL/SQL-proces om de data te genereren.

Hierin kijken we op welk dagdeel van welke dag van de week de bezoeken aan de weblog plaatsvinden. In deze dimensie, die vrijwel niets te maken heeft met de time-dimensie, delen we de etmalen in blokken van twee uur in, deze in dagdelen, de dagdelen in dagen en de dagen in werkdagen en weekend. Naast de dimensies wordt het tweede element in een datawarehouse gevormd door de feiten (*facts*) die via kubussen aan de dimensies worden gerelateerd. Onze facts zijn simpelweg de individuele bezoeken aan de weblog. Onze kubus is toevallig een echte kubus met precies drie dimensies - Visitor, VisitMoment en VisitDate. De default-aggregatie in onze kubus is de *count*: we tellen het aantal bezoeken. Er is geen *measure* in onze facts die zinvol te aggregeren is. Met deze kubus geven we aan dat we geïnteresseerd zijn in de bezoeken aan de weblog per bezoeker (en land en continent), per dag (en maand en jaar) en per moment van de week (dagdeel, weekdag, werkdag of weekend).

## Transformaties ontwerpen

Misschien wel de belangrijkste stap in het ETL-proces is het ontwerpen en de implementatie van de transformatie. In OWB

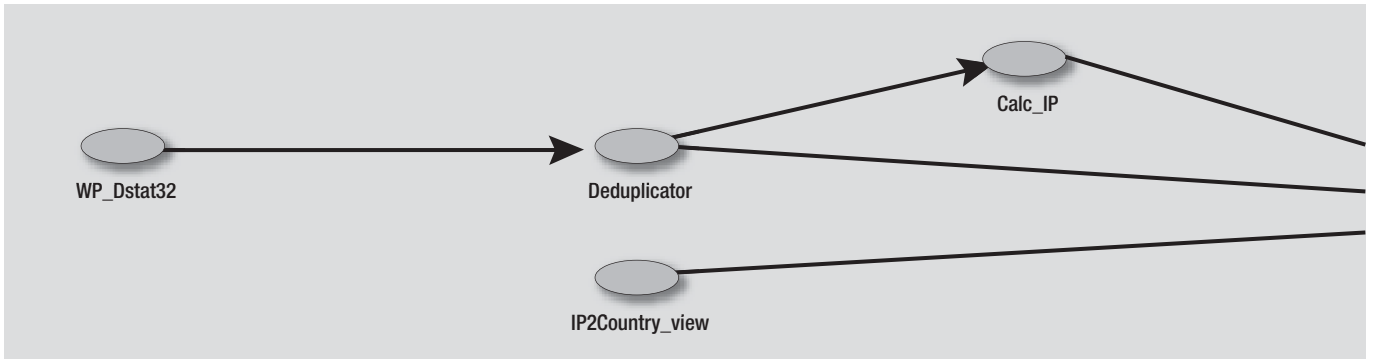


Figuur 9. Het tweede element in een datawarehouse wordt gevormd door de facts, die via kubussen aan de dimensies worden gerelateerd.

wordt deze stap vastgelegd met behulp van een mapping. Simpel gezegd is een mapping een koppeling van databronnen aan datatargets. Tussen bron en target kunnen allerlei operaties plaatsvinden die de data filteren, opschonen, verrijken, converteren, aggregeren, 'pivotten', combineren en splitsen. OWB bevat een groot aantal generieke operatoren die eenvoudig in een mapping gesleept kunnen worden. De operatoren worden met simpele pijltjes gehangen aan bronnen, target en andere operatoren. De ontwikkelaar kan ook zelf operatoren toevoegen, bijvoorbeeld in de vorm van PL/SQL-functies.

Een bijzonder krachtige operator is de 'match name and address'-operator. Deze kan worden gebruikt om NAW-gegevens uit een databron te matchen met een bestaand target en op grond van de match een merge van bepaalde velden uit te voeren. De match kan met behulp van complexe regels en scores afhankelijk van de mate van matching geconstateerd worden. Hierbij kan ondermeer worden gematched op voornamen, initialen, aanspreektitel, (deel van) postcode of telefoonnummer. De mapping in figuur 10 beschrijft in iconen op hoofdlijnen de data uit de bronnen - de tabel WP\_DSTATS, de view op de external table IP2COUNTRY en een hulptabel IP\_COUNTRIES waarin landen aan continenten worden gekoppeld. Als je de mapping in meer detail wilt zien en wilt gaan editen, kun je switchen naar de detailview (zie figuur 11). Hierin kun je door attributen uit een bron of operator te slepen naar een andere operator of een doel, de pijlen tekenen die door OWB vertaald worden in mappings en in SQL- of PL/SQL-code. De mapping start vanuit de WP\_DSTATS2 tabel; hieruit worden alle unieke IP-adressen geselecteerd - de DEDuplicator-operator bewaakt de uniekheid. De CALC\_IP-operator in deze mapping is een PL/SQL-functie die ook is geregistreerd in de OWB-repository. Deze vertaalt een IP-adres in zijn numerieke representatie, zoals hiervoor beschreven. Vervolgens kunnen deze waarden gekoppeld worden met de IP-range die in de IP2COUNTRY\_VIEW wordt geselecteerd uit de External Table en indirect uit de IP-to-country.csv file. De tweede join-operatie tenslotte koppelt de landcode aan de IP\_COUNTRIES tabel waarin bij een land het continent is gedefinieerd. Tenslotte wordt



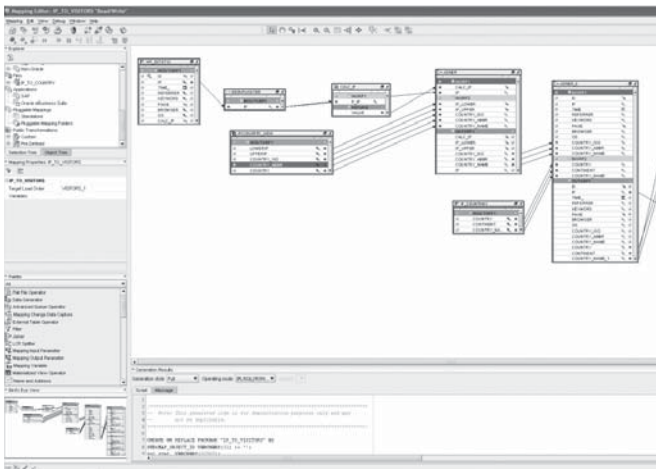


Figuur 10. Deze mapping beschrijft in iconen op hoofdlijnen de data uit de bronnen.

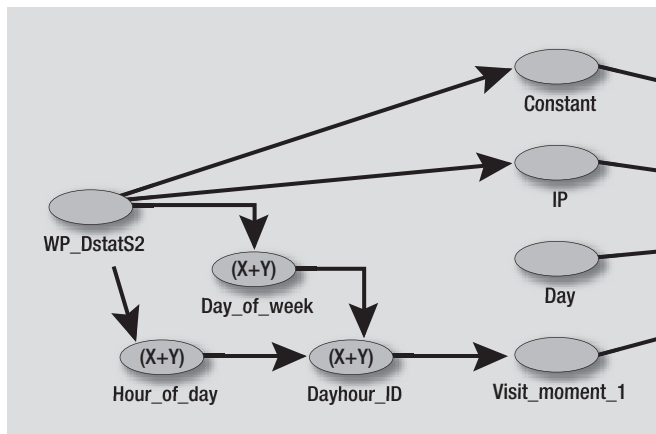
met de zo ontstane resultaatset - met IP, LAND, CONTINENT - de VISITORS (dimension)-tabel bijgewerkt - insert of update. Naast deze mapping worden in onze applicatie ook mappings gecreëerd voor de VisitTime dimension, de VisitMoment Dimension en het laden van de Kubus.

### OWB Repository webbrowser

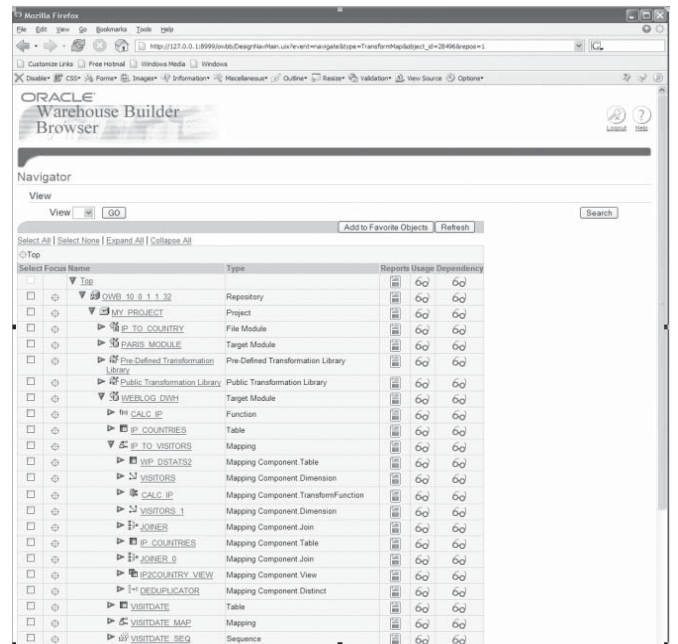
Met behulp van de Oracle Warehouse Builder repository-browser kan iedere geautoriseerde gebruiker via een webbrowser de inhoud van de OWB-repository doorzoeken. Vergelijkbaar met de Repository Object Browser voor Oracle



Figuur 11. In de detailview kun je de pijlen tekenen die door OWB vertaald worden in mappings en in SQL- of PL/SQL-code.



Figuur 12. Mappings voor de VisitTime dimension, de VisitMoment Dimension en het laden van de kubus.

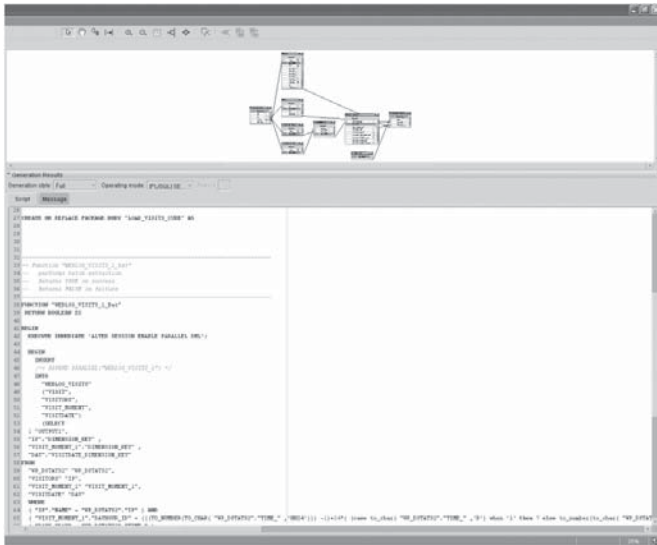


Figuur 13. De OWB repository-browser toont read-only rapporten met de belangrijkste design-gegevens.

Designer toont de OWB-browser ook read-only rapporten met de belangrijkste design-gegevens. Via de browser kunnen geen runtime gegevens worden bekeken, noch kunnen scripts gedownload worden. Maar voor analisten en eventueel eindgebruikers zijn er nuttige rapporten om een overzicht te krijgen zonder de OWB-client te hoeven installeren of te leren gebruiken.

### Laad het target datawarehouse

OWB is - net als Oracle Designer - naast een repository van meta-data met beschrijvingen van objecten en processen ook



Figuur 14. We laden achtereenvolgens de Visitor dimension, de VisitDate dimension en de VisitMoment dimension.

een code-generator. OWB kan de DDL-statements genereren om database-objecten in de database te creëren, bijvoorbeeld tabellen met alle indexen, constraints en partitions. Voor mappings, de transformatie-processen, wordt PL/SQL- en SQL-code gegenereerd, in de vorm van packages. Deze packages maken gebruik van runtime OWB-library's voor ondermeer generieke transformatie-elementen, exception-handling en auditing. Bij de deployment kun je aangeven welke versie van de database in je doelomgeving aanwezig is: 8i, 9i of 10g. OWB genereert code die optimaal gebruik maakt van de Oracle-features in doel-databaseversie. Vanaf 9i kunnen bijvoorbeeld de *merge*- en *multi-table-insert*-statements worden gebruikt. De gegenereerde code kan in verschillende modes worden uitgevoerd: SET-based (SQL-gebaseerd en het meest efficiënt), ROW-based (PL/SQL op basis van cursoren en BULK DML) of SET-based en in geval van exceptions automatisch omschakelen naar ROW-based. Hoewel de gegenereerde code voor ervaren PL/SQL-ontwikkelaars niet heel complex is om te doorgronden, gaat het wel om sterk geoptimaliseerde en uitgebreide PL/SQL- en SQL-routines - honderden tot duizenden regels per mapping. Het genereren levert zowel qua productiviteit als kwaliteit een enorme procesverbetering op. Als het ontwerp correct is vastgelegd in OWB is het genereren van code, het aanmaken van de database-objecten en het uitvoeren van de daadwerkelijke ETL-processen een eenvoudige zaak geworden<sup>1</sup>. De resultaten van de ETL-operaties zijn in de runtime audit-view van OWB op te vragen of in de OWB Audit Browser webinterface te bekijken. OWB geeft de mogelijkheid om direct vanuit de objectnavigator-objecten zoals tabellen, views, dimensies en ook mappings te

<sup>1</sup> Overigens hebben de PL/SQL-packages een heldere API en kunnen ze ook rechtstreeks worden gestart vanuit bijvoorbeeld SQL\*Plus, dbms\_job of dbms\_scheduler of een eigen maatwerk-applicatie!

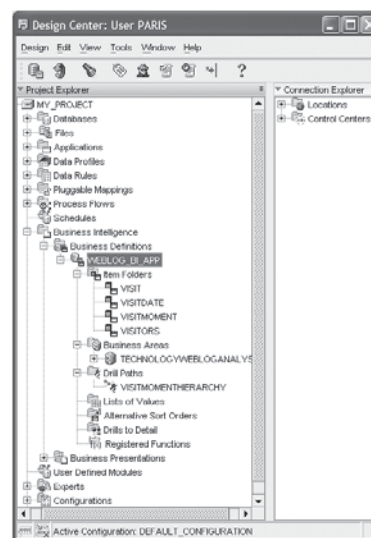
deployen naar een database. Dat gebeurt via het zogenaamde Control Center. Dit Center kan ook worden gebruikt om het uitvoeren van een taak te scheduleren en als achtergrondoperatie te laten uitvoeren. Denk hierbij met name aan het laden van nieuwe data in de targettabellen op basis van de OWB-mappings. Als alternatief kan je ook OWB de scripts laten genereren en deze naar disk schrijven, of ze zelf uitvoeren in SQL\*Plus. We laden achtereenvolgens de Visitor-dimension, de VisitDate-dimension en de VisitMoment-dimension. Dan is het tijd om de Weblog-kubus zelf te vullen. Als alle data geladen zijn in ons datawarehouse kunnen we enkele eenvoudige analyses doen.

## Verkrijg Business Intelligence

Eén van de motto's van Oracle Warehouse Builder is: van data tot informatie. We willen antwoorden op onze vragen hebben. We zitten niet perse te wachten op tientallen megabytes aan data. Oracle Warehouse Builder kan ons nu op twee manieren ondersteunen:

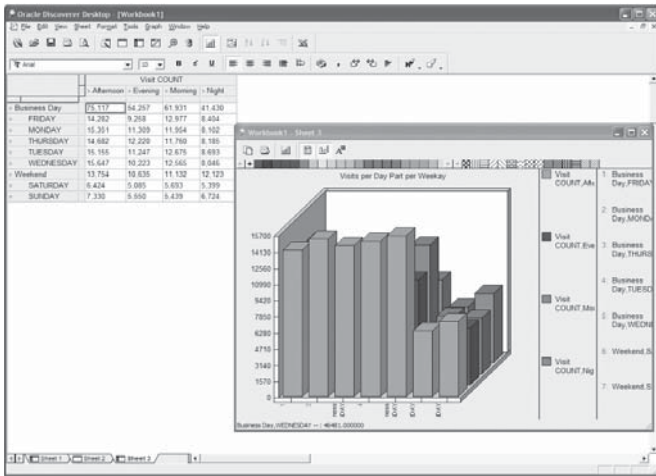
- We kunnen BI-applicaties ontwerpen en genereren vanuit OWB; hierbij gaat het voornamelijk om Oracle Discoverer en Oracle BI Beans-applicaties, maar meer smaken zijn voorzien.
- We kunnen de ingebouwde OWB Cube Data Viewer gebruiken om een eerste indruk te krijgen van onze kubus.

Figuur 15 toont hoe je, binnen OWB, tabellen en een business-definition kunt ontwerpen op basis van de meta-data met betrekking tot dimensies, hiërarchieën en kubussen. Deze kan vervolgens als Oracle Discoverer EUL (End User Layer) worden gegenereerd en geladen. In Oracle Discoverer Administrator kan dan eventuele verdere bewerking van de EUL plaatsvinden. Vervolgens kan in Discoverer Desktop, Plus of Viewer een analyse worden uitgevoerd.

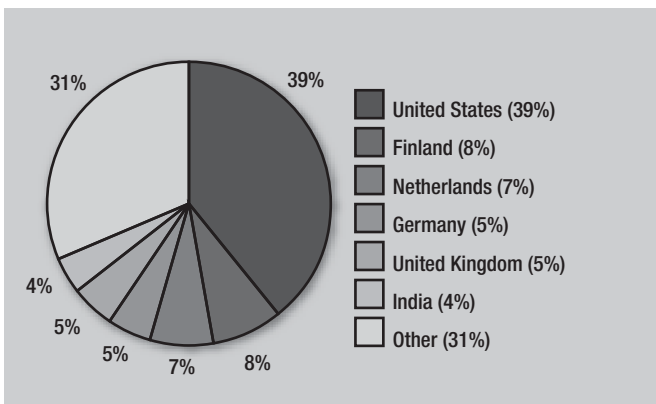


Figuur 15. Binnen OWB kunnen tabellen en een business-definition ontworpen worden op basis van meta-data.

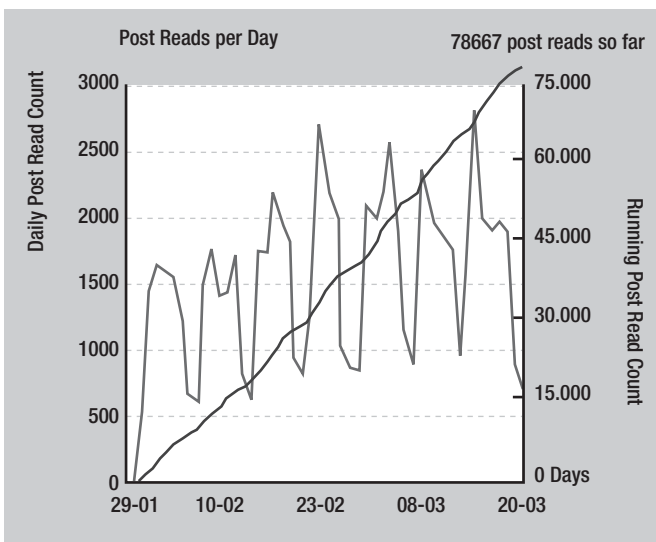
In dit voorbeeld zien we een CrossTab-rapport met bijbehorende grafiek dat een overzicht geeft van het aantal bezoeken aan de weblog tegen de dagen van de week en de dagdelen van die dagen. We zien dat over de gehele periode gerekend, de middag (tussen 12 en 18 uur) het meest populair is - circa 90.000 bezoeken - met de nacht (van 0 tot 6 uur) onderaan bungelend met toch nog 54.000 bezoekers. Het blijkt dat maandag, dinsdag en woensdag de



Figuur 16. Gedetailleerde informatie over spreiding van bezoekers over de week.



Figuur 17. Een simpele query laat snel zien welke landen de meeste bezoekers leveren. In totaal zijn er overigens bezoekers uit meer dan 100 landen geregistreerd.

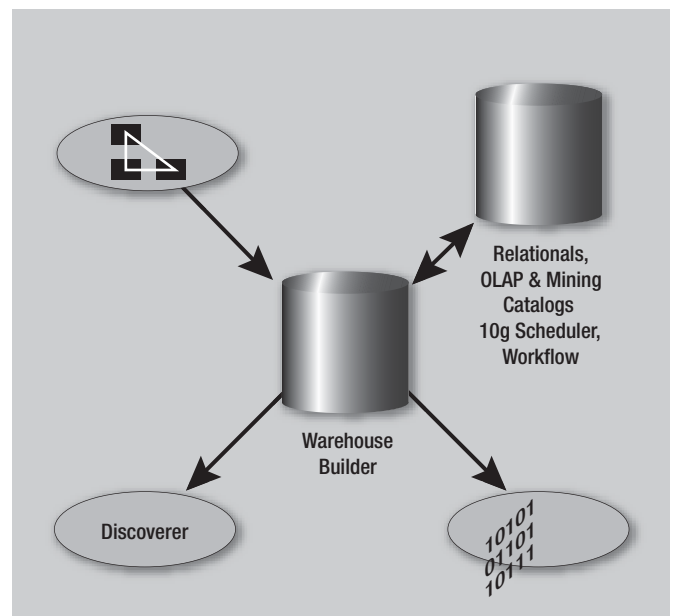


Figuur 18. Een ander overzicht: het aantal post reads per dag tussen 29 januari en 20 maart 2005, als ook het totale aantal over die periode. Uiteraard zijn de dalen in deze figuur de weekenden.

drukste dagen zijn. Als we verder *down-drillen* vinden we nog meer gedetailleerde informatie. Het beste moment in de week voor onderhoud is de nacht van vrijdag op zaterdag tussen middernacht en drie uur in de vroege zaterdagochtend. Het beste moment om een artikel te publiceren lijkt tussen 14.00 en 16.00 uur op dinsdag, woensdag of donderdag - dat zijn de drukst bezochte tijden (zie Figuur 16). Het voert te ver in dit artikel om een complete BI Beans-applicatie te ontwikkelen. Wel zien we in figuur 17 en 18 een paar eerste indrukken van de informatie die nu op basis van de data voor ons beschikbaar is.

### Architectuur, licenties en release-schema

OWB is gebaseerd op een centrale repository in een Oracle Database. Het biedt een perfecte architectuur voor een multi-user tool - vergelijk met Oracle Designer. In deze omgeving kunnen tegelijk meerdere projecten actief zijn. Users en rollen worden gebruikt om rechten op projecten en onderdelen van projecten te organiseren. Ook is een vorm van versiebeheer op de meta-data beschikbaar. Deze is overigens lang niet zo uitgebreid als de Oracle Software Configuration Manager. OWB ondersteunt evenals Oracle Designer faciliteiten om de meta-data in de repository te verrijken op basis van User Defined Attributen en Objecten. Daarnaast biedt OWB een aantal API's - PL/SQL, Java en OMB, de OWB Scripting Language - waarmee op programmatische wijze toegang kan worden gekregen tot inhoud en functionaliteit van OWB. Net als de Repository Object Browser voor Oracle Designer biedt ook OWB een browser-tool om de inhoud van de OWB-repository te bekijken en aan een groter publiek beschikbaar te stellen. OWB kan meta-data definities inlezen uit en wegschrijven naar



Figuur 19. De architectuur van Oracle Warehouse Builder.

een groot aantal externe tools, ondermeer op basis van CWM - het Common Warehouse Model. Ook kan OWB worden geconfigureerd om meta-data en data uit een maatwerk-interface te lezen. OWB kent een Dependency Manager om de analyses te doen van impact en lineage: welke afhankelijkheden in de transformaties en afgeleide objecten bestaan er van een gekozen object en, andersom, uit welke objecten en processtappen wordt een gekozen tabel of kolom afgeleid.

Bij navraag waar de kreet PARIS voor deze 10gR2 release vandaan kwam, bleek dat de afdeling product-development enigszins gestoord raakte van alle versienummers en release-labels. Er is toen besloten om een heel concrete release route uit te zetten door gebruik te maken van plaatsnamen: LONDON (9.0.2) en PARIS (10gR2). Tussentijdse releases zoals 9.0.3 en 9.2 konden eenvoudig worden tussengevoegd: DOVER en CALAIS. Na 10gR2 wordt koersgezet richting OSLO. En dat klinkt ambitieus als de werkelijke afstand maatgevend is voor de wijzigingen van functionaliteit. Het huidige release schema ziet er als volgt uit: in juni 2005 verschijnt bèta 3 van Oracle Warehouse Builder, gevolgd door de productierelease 10gR2 in augustus. Voor gebruik van de OWB Client voor ontwerp en generatie van datawarehouses, code voor ETL-processen en ook inzetten van de Data Profiler, is een licentie op de Oracle 10g Developer Suite benodigd, met ook Oracle Forms, Designer, Reports, Discoverer en JDeveloper. Voor het in een productie-omgeving uitvoeren van de ETL-processen moet de organisatie over een Enterprise Edition van de Oracle 9i of 10g database beschikken.

## Conclusie

De Oracle Warehouse Builder Paris Release is nu nog bèta en dat geeft enige beperkingen. Maar de functionaliteit die al wel beschikbaar en zichtbaar is, toont bijzonder krachtig en veelzijdig. De data profiling bijvoorbeeld biedt functionaliteit die in vrijwel iedere organisatie met databases waardevolle analyses op de kwaliteit van data mogelijk maakt. De database object-editors bieden een krachtige, repository gebaseerde omgeving om database design te doen gevolgd door servergeneratie, met ondersteuning van de nieuwste 10g database-features. De meta-data voor een database - datawarehouse of niet - kunnen via een prima UI worden vastgelegd en ook via API's worden ingelezen, uitgelezen en gemanipuleerd. Het ontwerpen van mappings en het genereren van code voor de ETL-operaties is, na enig wennen, betrekkelijk intuïtief en tamelijk productief. De gegenereerde code maakt gebruik van alle recente verbeteringen voor set-based operaties in de 9i en 10g releases van de Oracle-database en is daarmee erg efficiënt, daar valt bepaald niet tegenop te programmeren. Tenslotte is de proces-besturing, workflow en scheduling ogenschijnlijk erg krachtig - deze heb ik jammer genoeg in de bèta-1 release nog niet werkend kunnen krijgen. Oracle's commitment aan OWB is met deze release nogmaals

onderstreept. Mijn gevoel is dat de complete Oracle BI Toolstack - RDBMS, OLAP en Data Mining Engines, OWB, Discover, BI Beans, Oracle Reports, Oracle Enterprise Planning and Budgetting en als extraatje de Oracle OLAP plugin voor Excel - inmiddels flink volwassen is geworden. De integratie van de tools is goed voor elkaar - via de OLAP-catalog in de database, de BI Beans-ruggengraat van Discoverer, EPB- en custom-applicaties en de geïntegreerde design- en generatie-omgeving die OWB biedt. Oracle Warehouse Builder 10.1.2 komt in augustus 2005 beschikbaar. Organisaties met licenties voor de Oracle Developer Suite kunnen zonder verdere kosten van Oracle Warehouse Builder gebruik maken.

In het tweede deel uit deze serie, dat in de volgende editie van Optimize zal verschijnen, zullen we uitvoerig ingaan op de Data Profiler, Cleanser en Auditor van OWB. Daarnaast kijken we naar het Control Center, de Workflow en de Job Scheduler van OWB 10gR2.

**Lucas Jellema** (jellema@amis.nl) is sinds 2002 werkzaam bij AMIS Service in Nieuwegein, als expertise manager technologie en technisch consultant. Daarvoor werkte hij ruim acht jaar bij Oracle, ondermeer binnen het iDevelopment Center of Excellence. Hij houdt zich onder meer bezig met Java, XML/XSLT en andere webtechnologie als ook de Oracle-database en tools voor applicatie-ontwikkeling.