Konzepte, Regeln und Architekturen für das Oracle Data Warehouse

Alfred Schlaucher
Oracle
Hamburg

Schlüsselworte
Data Warehouse, Architektur, Regeln

Einleitung

Heute gehören DWH –Systeme zum Unternehmensalltag. Sie sind strategisch und oft mit wichtigen operativen Anwendungen verzahnt. Aber in kaum einem anderen Bereich gibt es so viele Baustellen, unzufriedene Anwender und klagende Kostenträger. Genannt werden: Zu geringe Umsetzungsgeschwindigkeit neuer Analyseanforderungen, Komplexität und Kosten für Betrieb und Weiterentwicklung, zu geringe Informationsausbeute für die Benutzer und zu wenig Anwenderkomfort (Performance, Flexibilität). Auf der anderen Seite boomt der Tools-Markt und es lösen sich die Wellen vermeintlicher Innovationen gegenseitig ab: Appliance, Realtime Data Warehouse, In Memory, BigData... Doch die Gründe sind nicht schlechte Technik oder Tools, sondern oft werden wichtige Methoden und Architekturgrundsätze nicht angewandt.

Deshalb an dieser Stelle eine Zusammenfassung von einfachen Regeln und Methoden um Anwendung und Betrieb des Oracle Data Warehouse effizienter zu machen - o*hne dass es teuer.*

Theoretische Grundlagen

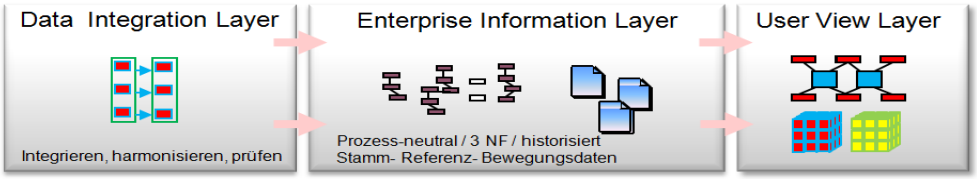
In unseren Data Warehouse Seminaren legen wir schon seit Jahren Wert darauf, zunächst Sinn, Zweck und die Funktionsweise eines Data Warehouse Systems zu verstehen, bevor wir Technologien erklären. Daher zunächst ein paar Grundlagen:

Seit Einführung des Data Warehouse-Konzepts in den 1990er Jahren gelten 4 Prinzipien für ein erfolgreiches DWH:

1. *Unternehmensweite* DWHs halten Informationen an *zentraler* Stelle in einer integrierten und abgestimmten Form vor.
2. Die Informationen sind für *alle verständlich* abgelegt, ohne Fachjargon und mit zusätzlichen fachlichen Erklärungen und Referenzinformationen.
3. Daten sind historisiert, was Vorhersagen für die Zukunft erlaubt.
4. Die Daten sind aus den OLTP-Systemen herauskopiert und ermöglichen eine neutrale, von operativen Zwängen losgelöste und Zeitpunkt-bezogene Analyse.

Damit OLTP-Daten diesen Zustand erreichen, durchlaufen sie in dem DWH einen notwendigen *Informationsbeschaffungsweg*, den man in 3 Phasen (Schichten / Layer) unterteilen kann:

A) Herauslösen der Daten aus den Vorsystemen (OLTP), integrieren, harmonisieren und qualitätssichern (Stage-Schritt, Data Integration Layer).
B) Granularisiertes Ablegen der Informationen (nahezu 3 NF), anreichern mit Referenzdaten und historisieren (Enterprise Information Layer).
C) Endbenutzergerechtes, fachthemenorientiertes Aufbereiten (meist multidimensional organisierte Daten, User View Layer).



Dieses 3-Schichtenmodell hat sich als Standard etabliert. Auf das Oracle Data Warehouse angewendet gelten folgende Konzepte bzw. Best Practices:

Konzepte, Methoden und Best Practices

Die **Konzentration des Schichtenmodells auf eine (!) einheitliche, zusammenhängende physische Ablage** ist gerade für das Oracle Data Warehouse eine der wichtigsten Forderungen und zielt auf eine Reihe von Vorteilen (s. u.). Alle oben genannten Layer sind am besten in einer (!) Datenbank auf einem (!) Server (oder Server Cluster) aufgehoben.

Ein wichtiges Prinzip ist das **datenzentrierte Vorgehen** bei dem Management der Warehouse-Daten. Das ist das Nutzen der technischen Mittel der Datenbank als primärer „Host" der Daten und betrifft Security, ETL, Vorbereiten von Kennzahlen und auch fachspezifische Sichten. Die Vorteile liegen in dem „gemeinsamen" Nutzen von Hardware und Lizenzen, einmalige Aufwende und kurze Wege.

Es gilt **Neutralität des DWH-Systems** gegenüber den Vorsystemen, vorallem aber gegenüber den nachfolgenden Business Intelligence – Tools (BI-Tools) zu wahren. Das bedeutet, dass nur *die* Datenmodellformen genutzt werden, die für die Aufgabenstellung in den jeweiligen DWH-Layern am passendsten sind:

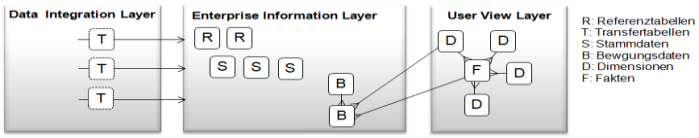
- nahezu 3NF in Enterprise Layer
- kompakte multidimensionale Strukturen – Star Schema (!) - im End User Layer.

DWH-Daten sind langlebig (10 Jahre und länger), Vorsysteme ändern sich meist viel häufiger und BI-Tools werden ebenfalls öfter ausgetauscht oder es befinden sich sogar unterschiedliche BI-Tools gleichzeitig im Einsatz. Wer diese Regel nicht berücksichtigt, transportiert alle Änderungen in den Vorsystemen und bei den BI-Tools automatisch auch in das DWH.

Durchgängigkeit des Schichtenmodells für Benutzerzugriffe. Um Informationen nicht unnötig zwischen den Schichten zu kopieren und zu verdoppeln, sollte Endbenutzern Lesezugriffe bis in den Enterprise Information Layer hinein erlaubt sein. Hier sind neben Stammdaten auch viele interessante Referenzdaten abgelegt. Das macht das DWH zu einem echten *Information-Repository* und wird für die Endanwender noch wertvoller. Meist haben die DWH-Administratoren ihre Hände auf dieser zentralen Schicht und sie argumentieren z. B. mit Security und Performance. Mit dieser restriktiven Haltung schmälern sie jedoch nur den Nutzen des DWH für die Anwender. Die vorgebrachten meist technischen Punkte lassen sich jedoch mit anderen Mitteln lösen.

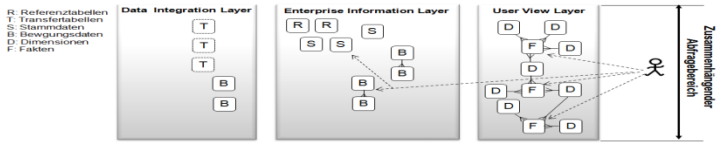
Es gilt der Grundsatz: Eine Kopie der Daten im End User Layer ist nur sinnvoll, wenn Daten in eine kompaktere Modellform (z. B. Starschema) oder in einer für Endbenutzer leichter verstehbaren Darstellung überführt werden.

Große Bewegungsdatenbestände (Fakten-Daten) sollte es nur einmal im gesamten System geben und nicht vom Enterprise Layer in den User View Layer (Data Marts) 1 zu 1 kopiert werden. Das bedeutet, dass diese in dem Enterprise Layer angesiedelt sind, um sie über den End User Layer zu referenzieren. Eventuell können die Bestände sogar mehrfach wieder verwendet werden. In Starschemen referenzieren bei dieser Vorgehensweise die Dimensions-Tabellen im User View Layer (Data Mart) als Parent Table „ihre" Faktentabellen im Enterprise Layer. Ohne technische Einschränkungen zu erleiden (z. B. Star Transformation-Feature) funktioniert das jedoch nur, wenn alle Objekte in der selben Datenbank liegen.



Verbundene-Kennzahlen

In dem User View Layer sind **Verknüpfungen in der multidimensionalen Struktur** zu schaffen, um Sachverhalte aus unterschiedlichen Kontexten und Sichten zusammenhängend abfragen zu können. Die verbindenden Elemente sind die Dimensionen (Geschäftsobjekte) über die die Kennzahlen (Fakten, Measures) in einen Bezug gebracht werden können. Das verhindert „Äpfel/Birne"-Vergliche schon bei dem Design des Datenmodells, macht sach-übergreifende Abfragen gefahrlos möglich und schafft zusätzliche Flexibilität für die Anwender. Extrem formuliert: *Es sollten alle Faktentabellen mit passenden Joins über Ihre Dimensionen in einer Abfrage zugegriffen werden können*. Das ist in der Praxis nicht immer machbar, weil Sachgebiete u. U. zu unterschiedlich sind.



Vorgedachte Aggregate und Kennzahlen

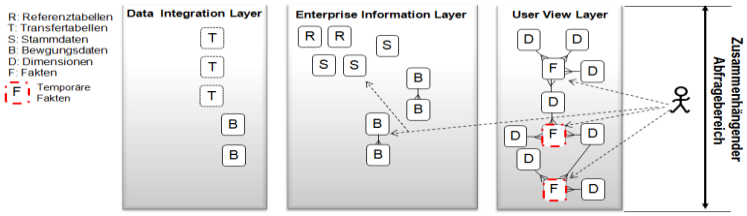
Die meisten Kennzahlen sind bereits bekannt. Man bereitet sie schon in der DWH-Datenbank als **vorgedachte Aggregate** vor und nicht erst in dem BI-Tool. Das mindert den Aufwand in den aufsetzenden BI-Tools und schafft zusätzliche Flexibilität bzw. Angebote für die Anwender. Hier helfen **Kennzahlenbäume**, die technisch als Nested-Materialized Views (aufeinander aufbauende Materialized Views) realisiert sind. Die Verwaltung erfolgt durch die Datenbank transparent, völlig automatisiert und ressourcenschonend. Der Wiederverwendungseffekt ist beachtlich.

Die klassischen **Aggretationstabellen** sind out. Man realisiert sie heute grundsätzlich mit Materialized Views. Das spart Verwaltungs- und ETL-Aufwand.

Generell wird man den **Feinheitsgrad** der Daten in User und Enterprise Layer so granular wie möglich halten, denn aggregieren kann man später immer noch, aber etwas Aggregiertes lässt sich im Nachhinein nicht mehr granularer gestalten. Granulare Daten bieten mehr Auswerteoptionen und Flexibilität für die Anwender. Datenbanken und Hardware sollte so ausgelegt sein, dass auch große Datenbestände „on the Fly“ aggregiert werden können. Die Technologie hierzu ist längst vorhande.

Dynamische Auswertestrukturen

Die Strukturen im User View Layer (Data Marts) sollten so gewählt werden, dass man ihre Inhalte jederzeit *dynamisch* aus den Daten der Vorschicht wieder herleiten kann, wenn diese durch die Anwender angefordert werden. So muss man nicht permanent alle User View Daten bereithalten oder aktualisieren. Das spart Plattenplatz, ETL-Aufwand und auch ein Backup der Data Mart –Daten ist nicht mehr nötig.



Das Security-Konzept des Analyse-Systems sollte in der DWH-Datenbank gelöst sein und nicht in dem nachgelagerten BI-Tool. Damit ist eine Offenheit für flexible BI-Strategien gegeben. Unterschiedliche BI-Tools, aber auch z. B. Excel oder sonstige OCI-Anwendungen partizipieren von dem „Einmal-Security“-Aufwand in der Datenbank. Ausserdem vermeidet man peinliche Security-Pannen. Security- oder **Mandanten-Anforderungen** dürfen *nicht* zu einer Verdoppelung von Strukturen und Daten führen.

ETL-Prozesse sollten innerhalb der Datenbank stattfinden, um unnötigen Datentransport zwischen Datenbank- und ETL-Server zu vermeiden. Zudem nutzt man die Hardware-Ressource der Datenbank und Datenbank – Lizenzen besser aus. Hinzu kommen die vielen technischen Mittel der Datenbank wie Table Functions, Partitioning und die Load-Features, die extrem schnelles Laden möglich machen.

1 zu 1 Transport-Vorgänge, also Datenbewegungen ohne Mehrwert-liefernde Transformationen, sind zu vermeiden.

Alle ETL-Prozesse sollte man **Set-Based**, d. h. mengen-basiert und nicht satzweise, durchführen. **Constraints und Indexe** sind auszuschalten. Syntaktische Datenqualität wird nicht mit Datenbank-Constraints sondern aufgrund der besseren Performance mit SQL-Mitteln gelöst.

Die **Parallelisierung (parallele Datenbankprozesse)** wird im ETL-Prozess nicht pauschal, sondern gezielt für einzelne Objekte gesteuert. ETL-Prozesse sind kontrollierte Vorgänge. Hier weiss man i. d. R. was passiert und kann bezogen auf Parallelisierung aber auch bzgl. Re-Indizierung, Verwenden von Partitioning oder Aktualisierung der Statistiken kontrolliert vorgehen. „Künstliches“ d. h. manuell programmiertes Parallelisieren ausserhalb der Datenbank ist zu aufwendig, führt zu Lock-Situationen und sollte vermieden werden.

Kommen dennoch **Engine-gestützte ETL-Werkzeuge** zur Anwendung, die Transformatiosprozesse nicht in der Datenbank durchführen, sollte man die Datenleitung zwischen ETL- und Datenbank-Server möglichst leistungsstark wählen (10Gb). Einfache Datenkopien sollte man nicht mit dem ETL-Werkzeug, sondern mit Datenbankmitteln lösen und stattdessen einen graphischen Platzhalter für diese Datenbank-Operation in dem graphischen Modell einfügen.

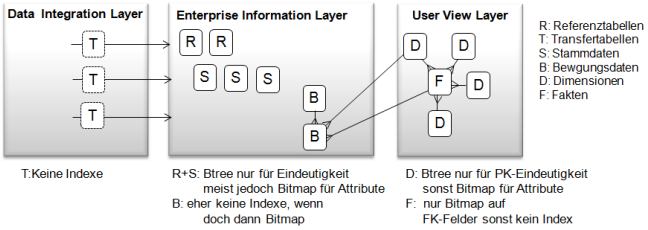
Dokumentation, Metadaten und Kontrolle: Kaum ein anderer Bereich wird in dem DWH stärker vernachlässigt als der der Metadaten. Wahrscheinlich weil die Verantwortlichen hier keinen unmittelbaren Nutzen sehen. Aber die Folgen fehlender Metadatendokumentation entstehen einige Zeit später umso drastischer, wenn Chaos Einzug hält: Sachverhalte sind mehrfach und auch noch unterschiedlich gespeichert. Die meisten Nutzer (und Administratoren) wissen nur z. T. was gespeichert ist. Große Datenmengen liegen nutzlos brach.

Ein DWH braucht neben einer sauberen Datenmodellierung auch ein **neutrales Metadaten-Repository**, in dem neben den technischen Objekten der Datenbank (Tabellen, Spalten, Views, Mappings) auch fachliche Beschreibungen der Inhalte, eine Dokumentation über Herkunft (Transformationen) und Verwendung (Nutzungsstatistik) der Daten und eine Dokumentation über Benutzer und ihre Erwartungen vorhanden ist. Es muss jederzeit bekannt sein: *Wer benutzt welche Daten und wo sind welche Daten wie gespeichert.*

Diese Information sollte für alle (!) Beteiligten z. B. in einer Web-Oberfläche zur Verfügung stehen (und nicht nur dem einen Administrator).

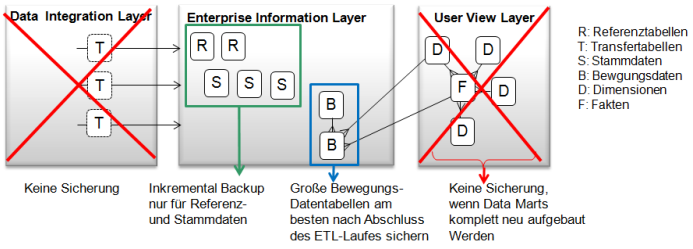
Indizierung im DWH

Im DWH nutzt man eher **Bitmap-Indizierung** als Btrees. Bitmaps benötigen weniger Platz und sind bei mengenorientierten Abfragen schneller. Btrees nutzt man oft nur zur Verwaltung der Eindeutigkeit bei Stamm-, Referenz- und Dimensionstabellen (Primary Key-Funktion).



Daten-Backup des Data Warehouse

DWH Systeme benötigen ein eigenes, also von OLTP-Systemen unabhängiges, Backup-Verfahren, denn nicht alles muss man regelmäßig sichern. Integration und User View Layer sind nicht zu sichern. Referenz- und Stammdaten muss man nur dann sichern, wenn sie sich geändert haben (Inkremental Backup des RMAN). Große Bewegungsdatentabellen sichert man gekoppelt an den ETL-Prozess, weil man hier die geänderten Datenbestände kennt, z. B. in Verbindung mit Partitioning. Backups nur über das Storage System (Image-Copy) sind zu vermeiden (Gefahr von Block-Corruption und zu teuer).



Regeln aus Hardware-Sicht

Data Warehouse-Daten sollten in einem separaten DWH-Storage-System liegen. Eine gemeinsame Storage-Nutzung z. B. mit OLTP-Systemen ist zu vermeiden. Das DWH liest meist größere und zusammenhängende Datenmengen oft sogar über einen „Full table Scan“, OLTP – Systeme eher kleinere Daten-Häppchen und dazu oft noch über einen Index.

Das gilt analog auch für das Netzwerk zwischen DWH-Storage und Datenbank-Server. Direkt angeschlossener Datenspeicher ist sicher das beste. (Exadata – Appliance – Prinzip).

Bei der Wahl des Storage-Systems setzt man eher auf mehr und dafür kleine Platten. Ideal ist es, wenn diese nur zur Hälfte beschrieben werden (Datenablage nur auf den äußeren Spindeln). Der Hauptspeicher sollte mindestens mit 8 GB / Core dimensioniert sein, um aufwendige Star-Join-Operationen und analytische Funktionen besser zu unterstützen.

Die Menge der CPU-Cores richtet sich an der zu lesenden Datenmenge pro Sekunde (MB / sec). Man geht bei heutigen CPUs (>2GHz) von etwa 200 MB/sec Verarbeitungskapazität pro Core aus.

Bei einem optimierten 30TB DWH mit etwa 5000 MB/sec Datendurchsatz (5GB/sec) sind das ~ 25 Cores und etwa 100 Platten (ohne Spiegelung).

Wer weniger als 5 GB/ Sec Leseperformance in einem solchen DWH hat, sollte sich damit nicht zufrieden geben und Alternativen suchen.

Die Kosten für Storage-Systeme variieren sehr stark. Hat man „privaten“ Storage für das DWH gewählt, so kann man unterschiedlich teuere Platten für unterschiedliche wichtige/aktuelle Daten im DWH einsetzen. Meist wird nur ein geringer Teil der DWH-Daten intensiv genutzt und der große „Rest“ liegt für Monate brach.

Kontaktadresse:
Alfred Schlaucher
Oracle
Kühnehöfe 5
22761 Hamburg

Telefon: +49 (0) 40 89091 132
E-Mail Alfred.Schlaucher@oracle.com