Data Warehousing
Grundbegriffe und Problemstellung

Dr. Andrea Kennel, Trivadis AG, Glattbrugg, Schweiz
Andrea.Kennel@trivadis.com

Schlüsselworte
Data Warehouse, Cube, Data Mart, Bitmap Index, Star Queries, Parallel Queries, Partitioned Tables, Materialized Views

Zusammenfassung
Wer sich mit Data Warehousing beschäftigt, hat zuerst mit verschiedenen Fachbegriffen und Abkürzungen zu tun. Die wichtigsten Begriffe und Abkürzungen werden am Anfang kurz erläutert und Zusammenhänge aufgezeigt.
Da die Anforderungen an ein Data Warehouse anders sind als an konventionelle Datenbanken, ergeben sich auch andere Problemstellungen und Lösungsansätze. Oracle bietet für verschiedene Probleme entsprechende Lösungen. Diese werden nachfolgend erklärt.

Einige Begriffe

Data Warehouse

„Data Warehouse“ kann als Datenlager übersetzt werden. Es dient dazu, Daten über die Zeit zu archivieren und für verschiedene Auswertungen zur Verfügung zu stellen. Ein Data Warehouse ist klar abfrageorientiert. Dabei werden in der Regel nicht einzelne Datensätze gelesen, sondern Summen über viele Datensätze gebildet. Das Datenmodell ist auf das Lesen von vielen Datensätzen optimiert. Es ist denormalisiert, enthält also bewusst Redundanz.

OLAP

OLAP ist die Abkürzung von „Online Analytical Processing“. Im Gegensatz zu OLTP werden bei OLAP Daten nicht mutiert, sondern zum Analysieren gelesen. OLAP ist ein Hilfsmittel zur Analyse und stellt die Daten in verschiedenen Verdichtungsstufen dar. Es können zwei Arten von OLAP unterschieden werden: relational (ROLAP) und multidimensional (MOLAP).
HOLAP steht für Hybrid OLAP und ist die Kombination von ROLAP und MOLAP.

OLTP

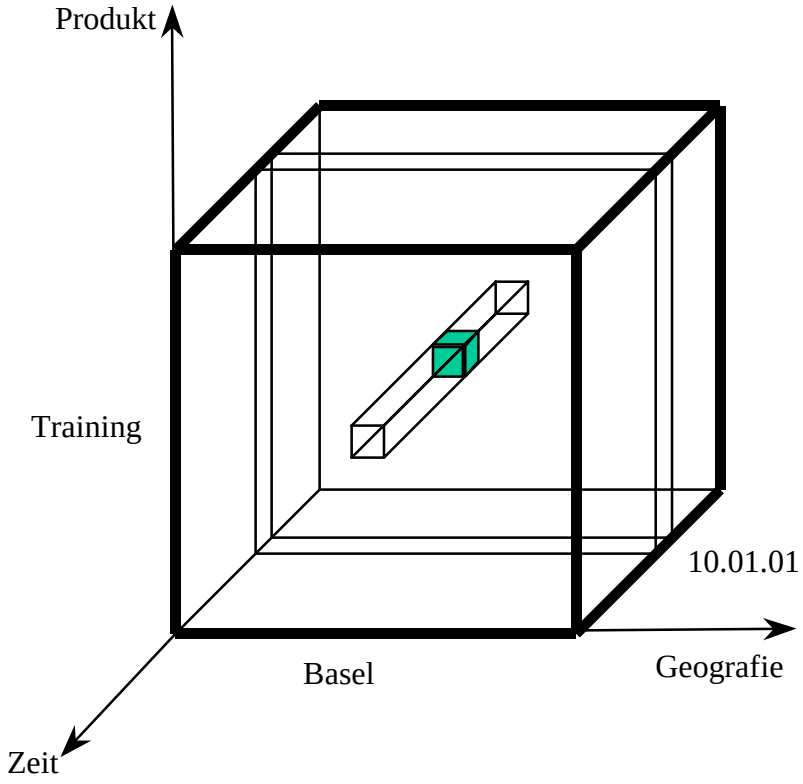
OLTP steht für „Online Transaction Processing“. Dabei werden Daten laufend geschrieben, mutiert und gelesen. Diese Systeme sind im Gegensatz zu OLAP Systemen transaktionsorientiert. Das zugrundeliegende Datenmodell ist relational und auf Schreiben und Lesen einzelner Datensätze optimiert.

Data Mart

Ein „Data Mart“ ist ein Ausschnitt aus den Daten eines Date Warehouses. Die Daten werden als Würfel dargestellt, der die Basisdaten sowie die Verdichtungen darstellt.
Im Bereich Data Mart und Würfel kennt man den Begriff „Slice and Dice“.
Slice bedeutet, dass der Würfel beliebig betrachtet werden kann, indem die Dimensionen ausgetauscht werden oder nur eine Scheibe betrachtet wird.
Dice bedeutet, dass zwischen verschiedenen Verdichtungsstufen gewählt werden kann mit den Möglichkeiten die Stufen über Drill Down und Drill Up zu wechseln.

Würfel (Cube)

Ein Data Mart wird in der Form eines Würfels (Fig. 1) dargestellt. Dabei entsprechen die Kanten den Dimensionen. Die Werte sind im Würfel selber und werden über die Dimensionen gelesen.



Figur 1: Würfel mit drei Dimensionen

DSS

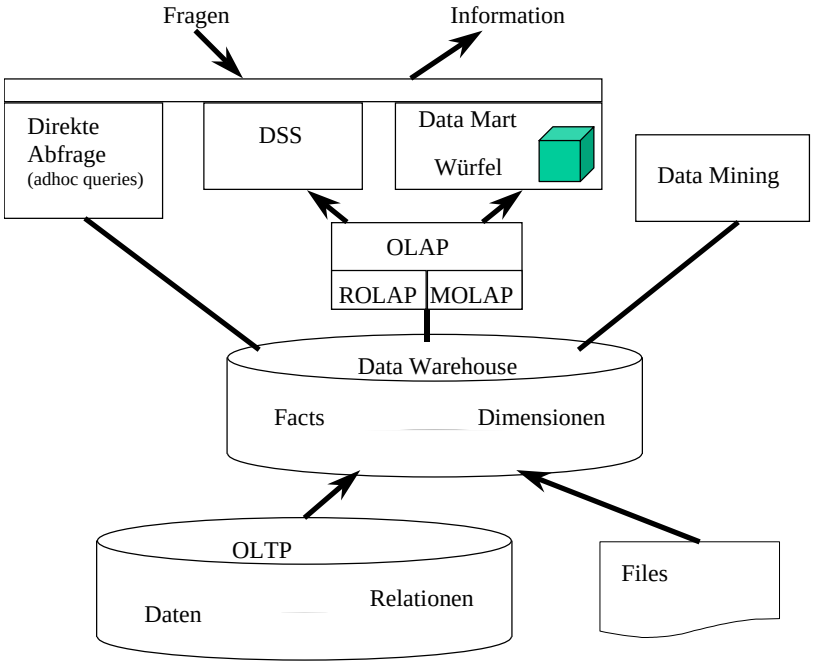
DSS steht für „Decision Support System“ und bezeichnet Abfrageunterstützungen im Allgemeinen.

Data Mining

„Data Mining“ ist die Suche nach auffälligen Mustern. Dabei können Vorgaben und Einschränkungen definiert werden, nach denen gesucht werden soll.

Zusammenfassung

Die verschiedenen Begriffe im Data Warehousing stehen in Abhängigkeit und können wie in Figur 2 dargestellt werden.



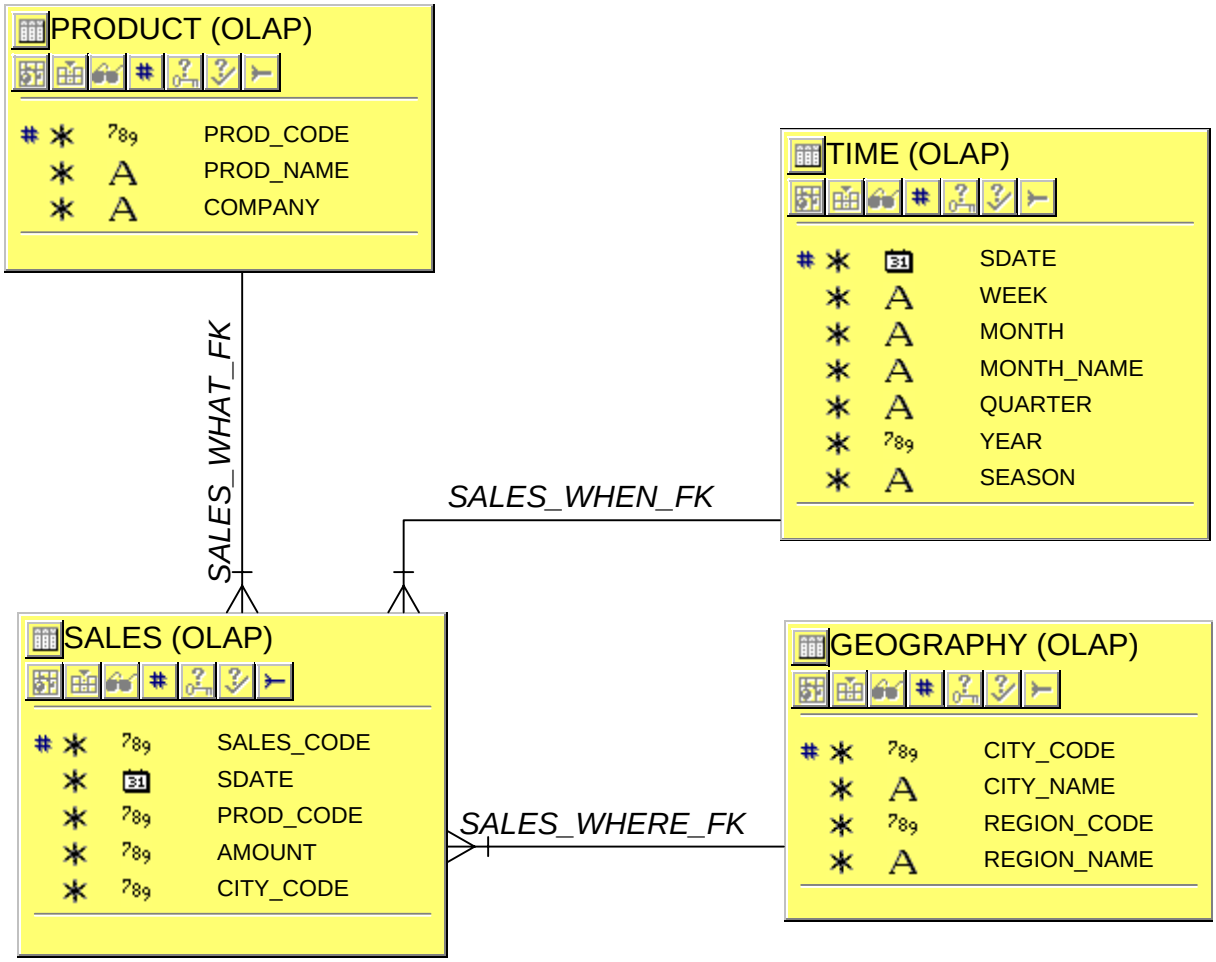
Figur 2: Einige Begriffe und ihr Zusammenspiel

Daten in einem Data Warehouse werden in Form von Facts (Werten) und Dimensionen gespeichert. Die Daten für das Data Warehouse kommen aus OLTP Systemen oder werden über Files geladen. Für die Auswertung werden Data Marts in Form von Würfeln dargestellt, oder es werden andere Decision Support Systeme zur Verfügung gestellt. Diese abfrageorientierten Systeme für die Analyse werden auch OLAP-Systeme genannt. Weiter ist es möglich, Abfagen frei auf dem Data Warehouse auszuführen. Dazu braucht es spezifische Kenntnisse in einer Abfragesprache wie SQL. Data Mining Syteme greifen normalerweise direkt auf das Data Warehouse zu.
Zwei weitere wichtige Begriffe sind Information und Daten. Im Data Warehouse sind die Daten gespeichert. Aufgrund einer Fragestellung, die mit Hilfe einer Abfrage beantwortet wird, entsteht Information. Information ist somit die Antwort auf eine Frage.

Beispiel DWH

Datenmodell

Unser vereinfachtes Datenmodell enthält eine Tabelle mit den Facts (Werten) für Verkäufe und drei Tabellen für die Dimensionen Produkt, Zeit und Geographie.



Figur 3: Datenmodell eines einfachen Data Warehouses

Beispielabfragen

Im obigen Modell können verschiedene Auswertungen über Verkäufe gemacht werden. So kann festgestellt werden, wie viel Stück eines Produkts wo wann verkauft wurden.
In den meisten Abfragen werden viele Datensätze gelesen und gruppiert zusammen gezählt. Als Basistabelle dazu dient immer die Fact-Tabelle SALES. Die Dimensionen können für Einschränkungen und Gruppierungen dazu genommen werden.
Im folgenden werden ein paar Abfragen in SQL Syntax dargestellt.

Abfrage nach Produkt

Welches Produkt wurde wie oft verkauft, über alle Regionen und die ganze erfasste Zeit:

```
SELECT prod_name, SUM(amount)
FROM sales s, product p
WHERE s.prod_code = p.prod_code
GROUP BY prod_name;

PROD_NAME            SUM(AMOUNT)
-------------------- -----------
Consulting                  5415
E-Commerce                  1576
Financial Services          1335
FlexSourcing                3053
Systems-Management          1551
Training                    2648
Verkauf Software            1318
```

Abfrage nach Region

Wie viele Verkäufe fanden je Region statt?

```
SELECT region_name, SUM(amount)
FROM sales s, geography g
WHERE s.city_code = g.city_code
GROUP BY region_name;

REGION_NAME          SUM(AMOUNT)
-------------------- -----------
Baden-Würtemberg            2100
Bern/Mittelland             4232
Nordwestschweiz             2036
Suisse-Romande              2132
Zürich/Aargau               6396
```

Abfrage nach Region für einen Monat

Wie viele Verkäufe fanden je Region im Monat Januar statt?

```
SELECT region_name, month, 'all_products' product,
  SUM(amount)
FROM sales s, geography g, time t
WHERE s.city_code = g.city_code
  AND s.sdate     = t.sdate
  AND t.month     = '01_2001'
GROUP BY region_name, month, 'all_products';

REGION_NAME          MONTH   PRODUCT      SUM(AMOUNT)
-------------------- ------- ------------ -----------
Baden-Würtemberg     01_2001 all_products        2036
Bern/Mittelland      01_2001 all_products        3912
Nordwestschweiz      01_2001 all_products        1940
Suisse-Romande       01_2001 all_products        1972
Zürich/Aargau        01_2001 all_products        6012
```

Problemstellungen

Modellierung

Anzahl Datenbanken

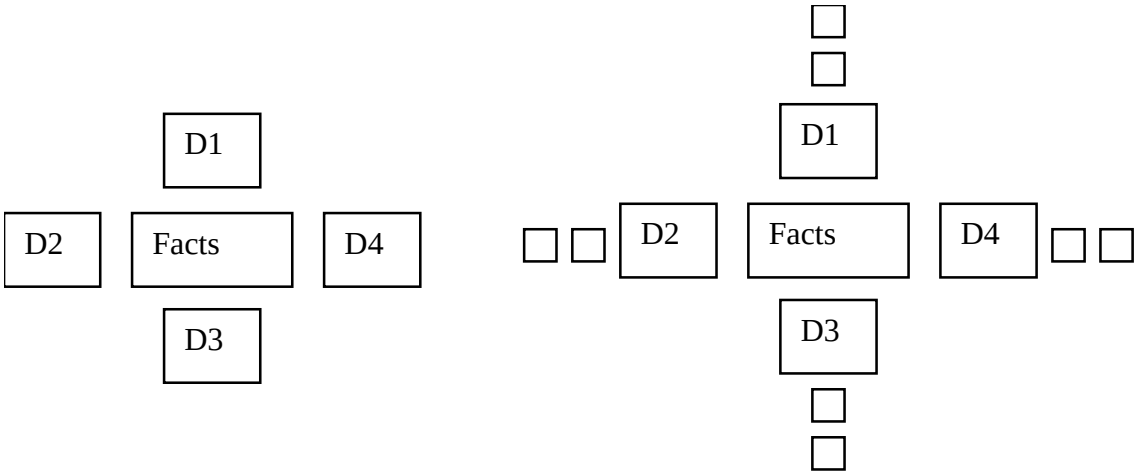
Normalerweise besteht ein Date Warehouse System nicht aus einer einzigen Datenbank. Oft werden die Daten aus Quell Datenbanken gelesen und zuerst in eine sogenannte **Staging Area** geschrieben. Diese Staging Area wird benutz um Daten zu bereinigen. Von der Staging Area gelangen die Daten dann in den **Data Pool**, quasi in den grossen Topf. Hier sind die Daten historisiert. Je nach Auswertung werden die Daten in einem oder mehreren Data Marts zur Verfügung gestellt. Die **Data Marts** sind oft eine weitere Kopie der Daten oder Teilen davon.

Datenmodell

Prinzipiell sind drei Arten von Datenmodellen bekann die im Data Warehousing eingesetzt werden:

- Relationen
- Star Schema
- Snowflake Schema

Das Relationenmodell ist normalisiert. Die Daten werden möglichst redundanzfrei auf mehrere Tabellen verteilt, die zueinander in Beziehung stehen. Dieses Modell wird vor allem für OLTP und für die Staging Area eingesetzt. Je nach Ziel und Aufbau der unterschiedlichen Datenbanken kann dieses Modell auch für den Data Pool gewählt werden.
Im Star Schema wird je Dimension eine Tabelle erstellt. Die Werte werden in den Fact-Tabellen gespeichert (Fig. 4). Die grafische Darstellung dieser Modellierung sieht aus wie ein Stern, daher stammt der Name „Star Schema“.
Das Snowflake Schema ist mit dem Star Schema verwandt. Der Unterschied liegt darin, dass die Dimensionen normalisiert sind. Beide Modelle werden für Data Marts und je nach Anforderung auch für den Data Pool verwendet.



Figur 4: Star Schema und Snowflake Schem

Daten laden

Das Laden der Daten kann in drei Vorgänge unterteilt werden:

- E: Extract, Extrahieren
- T: Transform, Umwandeln
- L: Load, Laden

Von diesen drei Vorgängen stammt auch der Begriff ETL.
Beim Extrahieren werden die Daten in ein File geschrieben oder direkt über einen DB-Link gelesen. Das Extrahieren der Daten kann periodisch, auf Anfrage oder nach einer bestimmten Anzahl Änderungen geschehen.
Beim Umwandeln werden die Daten soweit bereinigt, dass sie geladen werden können. Dazu gehört das Abgleichen von Datentypen, das Umkodieren und Umrechnen in einheitliche Metriken.
Beim Laden werden Schlüssel abgeglichen und die Historisierung nachgeführt.

Das Problem des Datenladens liegt ganz klar bei der Performance. Werden Daten täglich geladen, darf der Ladeprozess sicher nicht mehr als 24 Stunden dauern.

Abfrage beschleunigen

Ein zentrales Problem im Data Warehousing ist klar die Antwortzeit bei Abfragen, da viele Daten gelesen und zusammengezählt werden müssen. Um die Antwortzeiten zu reduzieren, kann mit Tuning oder mit Voraggregation gearbeitet werden.
Im Bereich Tunung kann mit Partitionen, Indexen oder Star Queries gearbeitet werden. All diese Möglichkeiten werden von Oracle unterstützt.
Auch Voraggregation ist mit Oracle möglich. Dabei können spezielle Tabellen generiert werden, oder es kann seit Oracle 8.1.6 mit Materialized Views gearbeitet werden. Die Möglichkeit von Voraggregation wird auch in diversen Tools eingesetzt.

Möglichkeiten von Oracle

Bitmap Index

In einem Bitmap Index wird je möglicher Attributwert ein Bit reserviert. Im Index wird dann je nach Attributwert das entsprechende Bit gesetzt. So kann sehr schnell eine Einschränkung über mehrere Attribute ausgewertet werden.

Tabelle

Rowid	Name	Zivilstand	Mitglied	...
A	Meier	V	J	
B	Müller	L	J	
C	Huber	L	N	
D	Keller	G	N	

Bitmap Index auf Zivilstand

Rowid	A	B	C	D	...
V	1	0	0	0	
L	0	1	1	0	
G	0	0	0	1	

Bitmap Index auf Mitglied

Rowid	A	B	C	D	...
J	1	1	0	0	
N	0	0	1	1	

Abfrage

```
WHERE zivilstand = 'L'
AND mitglied = 'J'
```

Rowid	A	B	C	D	...
L	0	1	1	0	
AND					
J	1	1	0	0	
=					
	0	1	0	0	

Figur 5: Beispiel zu Bitmap Index

Bitmap Indexe werden eingesetzt für Attribute mit kleiner Kardinalität, wenn das System abfrageorientiert ist. Für OLTP Systeme sind Bitmap Indexe nicht geeignet, da sie grosse Locks verursachen.
Abfragen können mit Bitmap Indexen klar beschleunigt werden, weiter sind Bitmap Indexe schnell aufgebaut und brauchen relativ wenig Speicherplatz.

Star Queries

Star Queries können dann eingesetzt werden, wenn mindestens 3 Tabellen (zwei kleine und eine sehr grosse) verknüpft werden und das kartesische Produkt von den zwei kleinen Tabellen kleiner ist als die grosse Tabelle.
Diese Voraussetzung ist gegeben, wenn wir eine sehr grosse Fact-Tabelle mit zwei kleinen Dimensionen verknüpfen. In der Star Query wird dann zuerst das kartesische Produkt der beiden Dimensionen gebildet, dann werden darauf die gewünschten

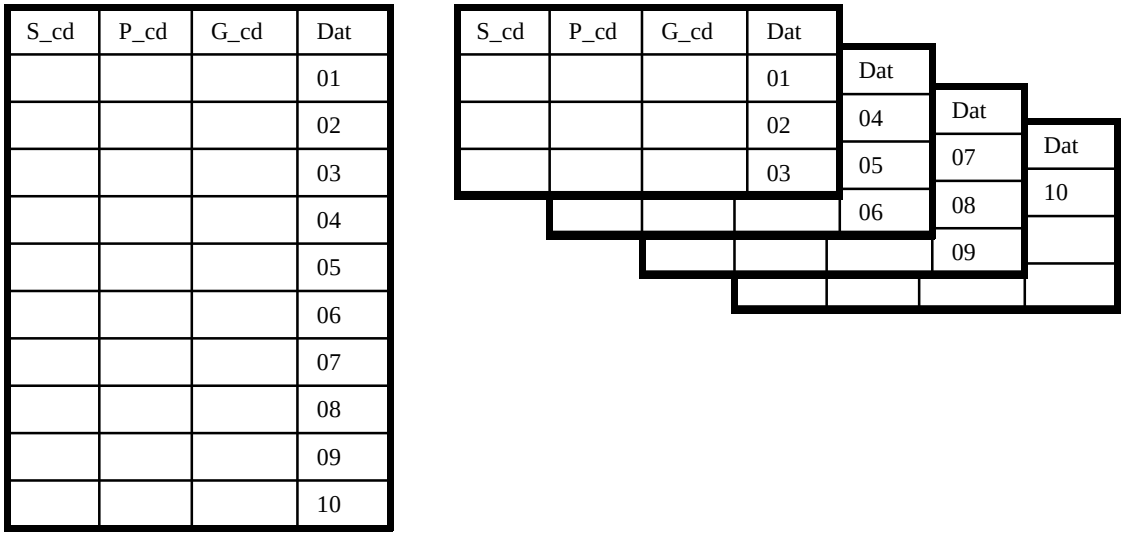
Einschränkungen vorgenommen, bevor das Resultat mit der sehr grossen Tabellen verknüpft wird.

Parallel Queries

Parallelisierung mit Oracle bedeutet, dass ein Query-Koordinator mehrere Sub-Prozesse startet und die Arbeit unter diesen Sub-Prozessen aufteilt.

Partitioned Tables

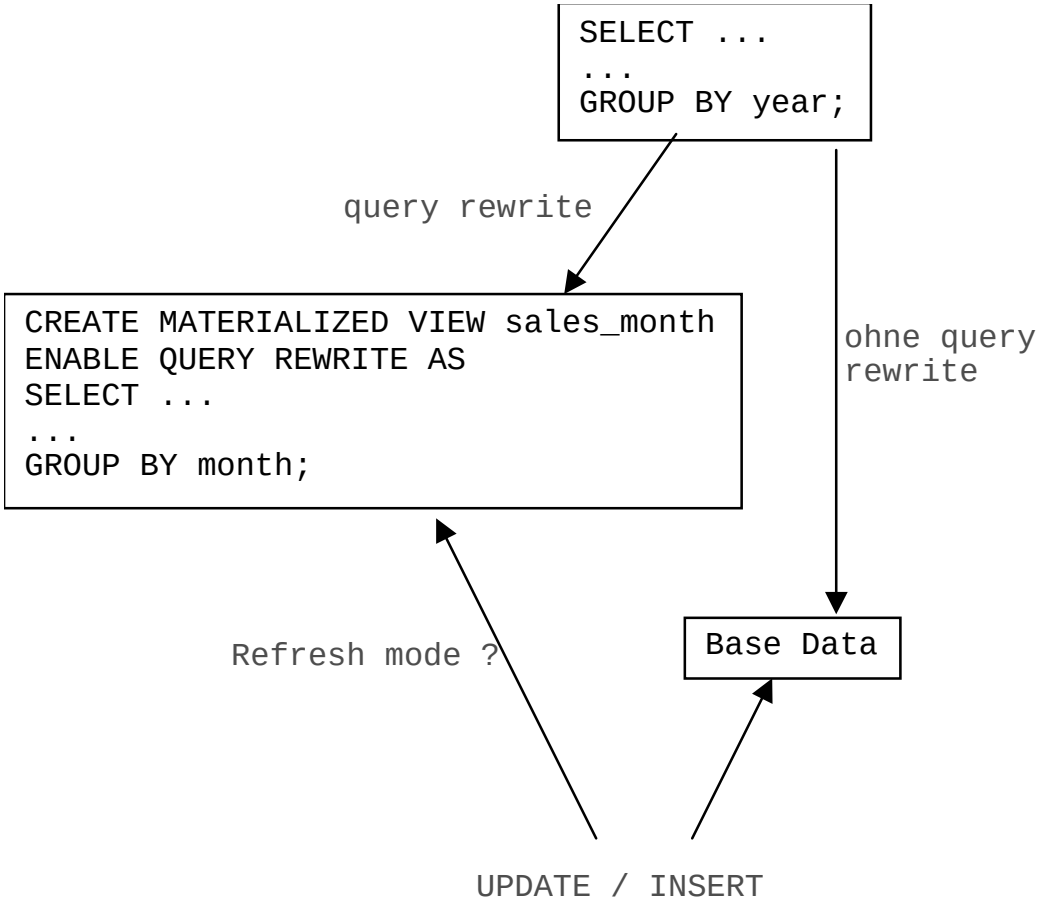
Eine Tabelle wird in physische Partitionen unterteilt. Bei einer Abfrage stellt Oracle selber fest, welche Partitionen gelesen werden müssen. Da je nach Abfrage nur einzelne Partitionen gelesen werden müssen, kann die Abfrage so beschleunigt werden. Weiter kann die Administration der Daten vereinfacht werden, da bei der Administration die Partitionen einzeln verwaltet werden können.



Figur 6: Partitionen

Materialized Views

Eine normale View speichert keine Daten, sondern nur die Definition der Abfrage. Die Materialized View speichert im Gegensatz dazu sowohl die Abfrage als auch die Daten. Werden die Basisdaten verändert, so muss die Materialized View mit einem Refresh nachgeführt werden.
Steht für eine Abfrage eine Materialized View zue Verfügung, so liest Oracle die Daten nicht von der Basistabelle, sondern von der Materialized View. So muss zur Beschleunigung von vorhandenen Abfragen die Abfrage selber nicht verändert werden. Es genügt, wenn eine entsprechende Materialized View definiert wird.



Figur 7: Materialized View

Externe Tabellen (9*i*)

Datenfiles können wie Tabellen angesprochen werden. Das vereifacht das Schreiben der Ladeprozeduren. Externe Tabellen ersetzen den SQL*Loader nicht, es ist aber eine hilfreiche Ergänzung.

Multiple Table Insert (9*i*)

Mit Multipe Table Insert kann mit einem Statement in mehr als eine Tabelle geschrieben werden.

Tools

- Daten laden
  - SQL*Loader
  - Import, Export
- Design
  - Warehouse Builder
- Data Mart, OLAP
  - Oracle Express
  - Discoverer
  - OLAP Services (9*i*)
- Data Mining
  - Darwin

Literatur und Links

Oracle spezifisch:

Oracle8 Data Warehousing, Oracle Press, ISBN 0-07-882511-3

Generelle Grundlagen

The Data Warehouse Toolkit, R. Kimball. Wiley Computer Publishing, ISBN 0-471-15337-0

The Data Warehouse Lifecycle Toolkit, R. Kimball. Wiley Computer Publishing, ISBN 0-471-25547-5

Data Warehouse Systeme, A. Bauer und H. Günzel. Dpunkt.verlag, ISBN 3-932588-76-2

Tool Übersicht

http://www.dw-institute.com/resourceguide2000/index.html

Folien zum Vortrag

http://www.trivadis.com

http://www.trivadis.com/de/services/publikationen/publiste.asp

Dr. Andrea Kennel
Trivadis AG
Kanalstr. 5
CH- 8152 Glattbrugg
Schweiz

Andrea.Kennel@trivadis.com