

### 3 OLAP und OLTP

Daten eines DWH stehen in Kontrast zu Daten in einer operativen Datenbank. In einem DWH geht es ausschliesslich um die Analyse der Daten. Operative Datenbanken hingegen behandeln Daten Transaction-Orientiert. Man spricht deshalb von

- **Online Analytical Processing (OLAP)** in DWHs, gegenüber
- **Online Transaction Processing (OLTP)** in traditionellen operativen Datenbanken.

Eine typische OLAP Abfrage ist: „gebe mir den Verkauf pro Monat und pro Filiale an“ während eine OLTP Abfrage wäre „gebe mir den aktuellen Stand des Kontos.“

Bei DWHs geht es um **wenige** Abfragen, die sehr komplex sind und recht **viel Zeit** in Anspruch nehmen. Bei OLTP geht es um viele, relativ kurze Update-Transaktionen, die wir aus dem relationalen DBMS-Welt kennen. Tabelle 1 fasst die Unterschiede zusammen.

**Tabelle 1: Unterschiede zwischen OLTP und OLAP**

OLTP	OLAP
Transaktionsorientiert	Analysisorientiert
Anwendungsorientiert	Themenorientiert
Abfrageresultate detailliert	Abfrageresultate zusammengefasst
Zeitlich genau	Zeitlich ungenau (snapshots)
Für die Administration	Für Managers
Wird ständig modifiziert (z.B. Jede Millisekunde)	Wird nicht oder selten (im Vergleich mit operativen Datenbanken) modifiziert (z.B. jeden Tag)
Läuft repetitiv	Läuft heuristisch
Untersuchte Prozesse sind a priori verstanden	Untersuchte Prozesse sind a priori nicht verstanden
Performanz-sensitiv	Nicht Performanz-sensitiv
Zugriff Datensatz-orientiert	Zugriff Mengenorientiert

## 4 Drei komplementäre Trends

Bei der Anwendung einer DWH werden drei verbunden Themenbereiche betrachtet:

- **Data Warehousing** bezeichnet die Integration [Engl: consolidation] von Daten aus verschiedenen Quellen in einem einzigen, sehr grossen Archiv:

- Laden von Daten, periodisches Aktualisieren/Synchronisieren aus anderen Systemen;
- Semantische Integration.

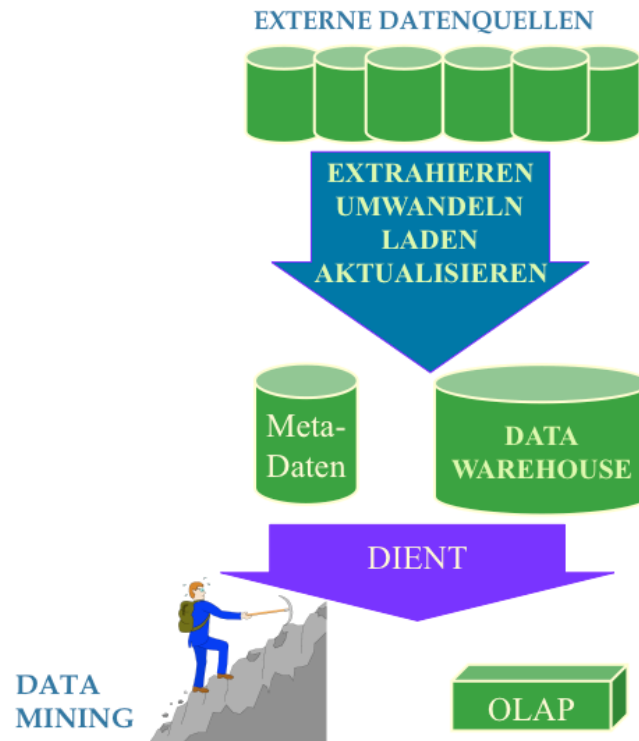
- **OLAP**:

- Komplexe SQL-Abfragen und Views;
- Abfragen basieren auf Spreadsheet-Stil. Operationen werden auf „multidimensionale“ Sichten der Daten durchgeführt;
- Interaktive und online Abfragen, durchgeführt von Endbenutzern.

- **Data Mining**: Entdeckungsartige Suche nach interessanten Trends und Anomalien

Zuerst konzentrieren wir uns auf Data Warehousing und OLAP und werden später auf Data Mining eingehen, da DWH und OLAP meistens Grundsteine für Data Mining Operationen darstellen. Data Warehousing kann mit dem Design des Datenmodells assoziiert werden, welches dann für die OLAP Abfragen angewendet wird.

## 5 Herausforderungen beim Data Warehousing



**Abbildung 2: Workflow bei Data Mining**

Daten werden aus einer operativen Datenbank (oder allgemeiner: aus externen Datenquellen) integriert. Mit Integration bezeichnen wir den Prozess, um die Daten in ein einheitliches Format und Schema zu bringen. Oft werden sie noch mit zusammengefasster Information ergänzt (Metadaten).

Data Warehouses nehmen heute schnell Grössen von mehreren Terabytes an. Bei solchen Datenmengen stossen Datenbanksysteme langsam an ihre Grenzen.

Interaktive Antwortzeiten sind für Abfragen nötig, da die Daten „on the fly“ von den Benutzern (speziell Manager, die aus Prinzip keine Zeit haben) analysiert werden.

## 6 Extrahieren, transformieren, laden

Ein DWH erfordert die integrierte Sammlung von unterschiedlich strukturierten Daten aus verschiedenen Quellen. Dies ist bei weitem keine triviale Aufgabe: Die Integration der Daten erfordert Entscheide, die sich im gesamten Design widerspiegeln. Der Prozess der Integration von heterogenen Daten aus (operativen) Umgebungen wird mit ETL bezeichnet:

- **EXTRACT:** Daten werden aus externen, operativen Datenbanken ausgelesen und vorbereitet [cleaned], um Fehler zu minimieren und um fehlende Informationen zu ergänzen.
- **TRANSFORM:** Es geht um Semantische Integration: Wenn man Daten aus verschiedenen Quellen verarbeitet, muss man unangepasste Informationen eliminieren (z.B. unterschiedliche Währungen). Die Transformation kann über die Definition einer Sicht [View] erledigt werden.
- **LOAD:** Laden, Aktualisieren, Entfernen [purge]:
  - Daten müssen regelmässig geladen, periodisch aktualisiert, und veraltete Daten entfernt werden. Das Laden kann z.B. mit Materialisation der in der „Transform“-Stufe erstellten View realisiert werden. Die Materialisation wird aber in einer anderen Datenbank gespeichert, als diejenige, die ihre Definition enthält. Zusätzliche Vorbereitungen sind nötig: Sortieren, Generierung von zusammenfassenden Information (Monats-Summen, usw.), Partitionierung, Generierung von Indizes. Bei Datenmengen von mehreren Terabytes kann das Einfügen von Daten in das DWH sehr lange dauern!
  - Aktualisieren: Ähnlich wie Asynchronous Replication in operativen DBMS.

Für alle Informationen im DWH müssen die Datenquelle, Ladezeit und andere Information registriert werden, so dass später rekonstruiert werden kann, welche Daten im DWH überhaupt abgelegt sind! (Stichwort: Metadata Repository)

## 7 Multidimensionales Datenmodell

Sammlung von numerische Messungen, die von einer Menge Dimensionen abhängen.

- Beispiel: Messung **Verkauf**, Dimensionen **Produkt** (key: **pid**), **Ort** (**ortid**), und **Zeit** (**zeitid**).

Hier sehen wir die Tranche [slice] **ortid=1**

pid	zeitid	verkauf
11	1	8
11	2	10
11	3	10
12	1	30
12	2	20
12	3	50
13	1	25
13	2	8
13	3	15

pid	zeitid	ortid	verkauf
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

Abbildung 3: Beispiel eines multidimensionalen Datenmodell

Im Multidimensionalen Datenmodell unterscheidet man zwischen **Messungen** und **Dimensionen**. Messungen sind die Werte, die von den Benutzern überwacht sein sollen. Messungen hängen von den Dimensionen ab. Dimensionen fassen den Kontext der Messungen in verschiedenen Auflösungen.

In der Figur 2 werden Messungen **Verkauf**, Dimensionen **Produkt** (key: **pid**), **Ort** (**ortid**), and **Zeit** (**zeitid**) dargestellt.

## 8 MOLAP vs. ROLAP

Multidimensionale Daten können physisch als eine (Disk-resident, persistent) Array gespeichert werden. Die **MOLAP**-Systeme sind auf diese Art von Datenverwaltung aufgebaut.

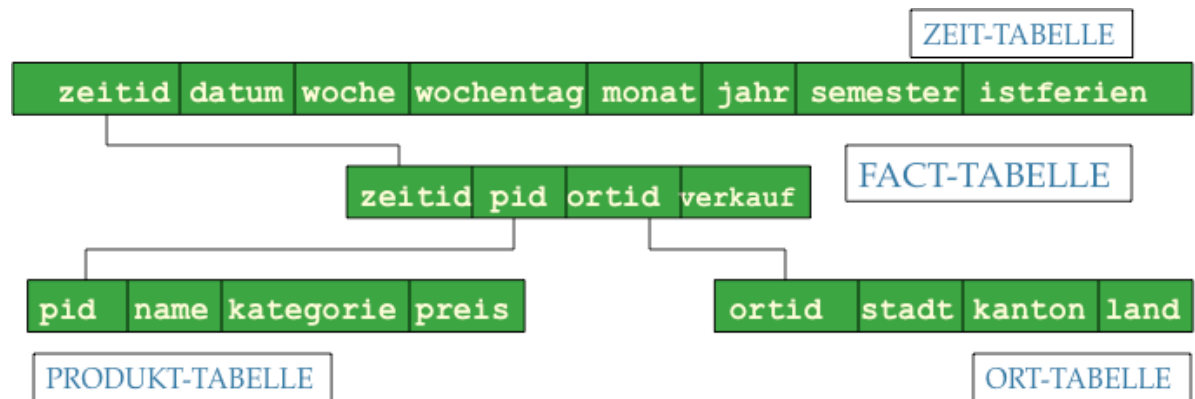
Multidimensionale Daten können aber auch als Relationen gespeichert werden. In diesem Fall spricht man von ROLAP-Systemen. Dies hat den grossen Vorteil für Unternehmen, dass sie die DWH mit der gleichen Software verwalten können als für die operative Aufgaben schon benutzt wird.

In ROLAP ist die Hauptrelation, die den Messwert mit den Dimensionen assoziiert, die **Fact Table**. Jede Dimension kann zusätzliche Attribute haben, die in der assoziierten **Dimension Table** vorkommen.

Z.B. eine Dimension wäre `Products(pid, pname, categorie, preis)`

Fact tables sind viel grösser als Dimension Tables.

## 9 OLAP Design: Star Schema



**Abbildung 4: Beispiel eines Star Schemas**

Im Star Schema werden die Beziehung zwischen Fact Table und Dimension Tables ersichtlich. Die Beziehung erfolgt über den Hauptschlüssel der entsprechenden Dimensionstabelle, die dann als Fremdschlüssel in der Fact Table vorkommt.

Fact Table ist in BCNF, aber Dimensionstabellen sind nicht normalisiert (zur Erinnerung BCNF: „each attribute must describe an entity or relationship identified by the key, the whole key, and nothing but the key“).

Dimensionstabellen sind klein; Modifikationen/Einfügen/Löschen sind seltene Operationen. So sind die Anomalien, die wegen der Nicht-Normalisierung entstehen, weniger wichtig als die gute Leistung der Abfragen.

Star Join: Join auf alle Relationen des Star-Schemas. Mit dem Star Schema wird recht ersichtlich, dass die Joins in OLAP abfragen eine wesentliche Rolle spielen.

Andere Schemata: Snowflake Schema ist eine Erweiterung des Star Schemas, um die Dimensionstabellen des Star-Schemas zu normalisieren. Obwohl von theoretischen Interesse, wird es in Praxis nicht (oft) angewendet, und deshalb betrachten wir es hier nicht weiter.

## 10 Hierarchien von Dimensionen

Wie vorher erwähnt werden Dimensionen so aufgebaut, dass sie erlauben, Messwerte mit verschiedenen „Auflösungen“ darzustellen. Diese Auflösungen werden im DWH Jargon als Hierarchie bezeichnet. Die Hierarchie ordnet die verschiedenen Dimensionsauflösungen in einer logischen Reihenfolge.

Figur 3 zeigt drei Beispiele von Dimensionen. Für die Dimension „Ort“ ist Stadt in Kanton enthalten, und Kanton ist in Land enthalten. Für die Zeit Dimension (die übrigens meistens die wichtigste Dimension ist) ist Woche aber nicht im Monat enthalten, deshalb gibt es zwei „Branchen“.

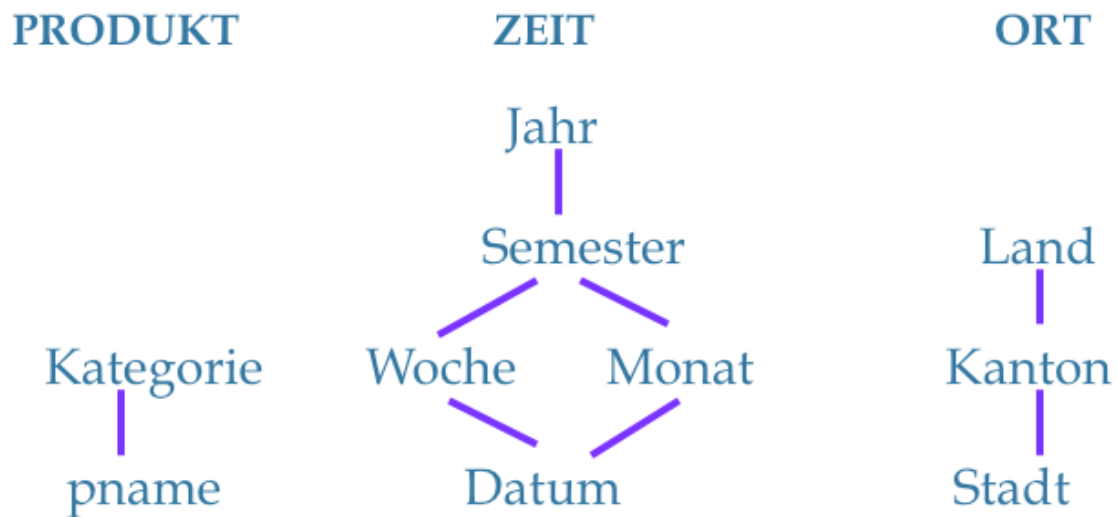


Abbildung 5: Beispiel von Dimensionshierarchien

## 11 OLAP Abfragen und typische Operationen auf DWHs

- Beeinflusst von SQL und von Spreadsheets
- Eine gewöhnliche Operation ist, die Messwerte über eine oder mehrere Dimensionen aufzusummieren [Aggregate]:
  - Finde den gesamten Verkauf [verkauf].
  - Finde den gesamten Verkauf von einem bestimmten Produkt [pname] für jede Stadt [stadt], oder für jeden Kanton [kanton]
  - Finde die 5 meist verkauften Produkte (geordnet mit gesamten Verkauf).
- Roll-up: „Wandern“ auf einer bestimmten Dimension von meist detailliert zu meist zusammengefasst.
  - Z.B: finde den gesamten Verkauf von pname per Stadt, dann finde den gesamten Verkauf per Kanton
- Drill-down: das Gegenteil von Roll-up.
  - Z.B., Aus der gesamten Verkauf per Kanton einen Drill-down ausführen, um den gesamten Verkauf per Stadt zu ansehen.
  - Z.B. Man kann auch auf einer anderen Dimension den Drill-down führen, z.B. um den gesamten Verkauf per Produkt zu berechnen.
- Pivoting: Aufsummieren auf ausgewählte Dimensionen.
  - Z.B. Pivoting auf Ort und Zeit ergibt diese Kreuztabulation [cross-tabulation]
- Slicing and Dicing: Gleichheits- bzw. Bereichsselektion auf einer oder mehreren Dimensionen.

Es ist zu merken, dass Roll-up, Drill-down, usw. neue Namen für alte Operationen sind. Die Namen wurde eher aus Marktspezifische Gründe eingeführt, aber repräsentieren Operationen die schon lange in der Datenbankwelt angewendet worden sind.

## 12 OLAP mit SQL Abfragen

Die Kreuztabulation, die das Pivoting von der vorherige Abbildung entspricht, wird mit folgenden SQL-Abfragen erhalten:

```
SELECT SUM(S.verkauf)
FROM verkauf S, Zeit T, Ort L
WHERE S.zeit_id=T.zeit_id AND S.ort_id=L.ort_id
GROUP BY T.jahr, L.kanton
---
```

```
SELECT SUM(S.verkauf)
FROM verkauf S, Zeit T
WHERE S.zeit_id=T.zeit_id
GROUP BY T.jahr
---
```

```
SELECT SUM(S.verkauf)
FROM verkauf S, Ort L
WHERE S.ort_id=L.ort_id
GROUP BY L.kanton
---
```

```
SELECT SUM(S.verkauf)
FROM verkauf S, Ort L
WHERE S.ort_id=L.ort_id
---
```