Data Warehouse

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.

What are the terms?

Subject Oriented:

Data that gives information about a particular subject instead of about a company's ongoing operations.

Integrated:

Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.

Time-variant:

All data in the data warehouse is identified with a particular time period.

Non-volatile:

Data is stable in a data warehouse. More data is added but data is never removed. This enables management to gain a consistent picture of the business.

(Source: "What is a Data Warehouse?" W.H. Inmon, Prism, Volume 1, Number 1, 1995).

This definition remains reasonably accurate almost ten years later. However, a single-subject data warehouse is typically referred to as a **data mart**, while data warehouses are generally enterprise in scope.

Also, data warehouses **can be volatile**. Due to the large amount of storage required for a data warehouse, (multi-terabyte data warehouses are not uncommon), only a certain number of periods of history are kept in the warehouse.

> E.g. if three years of data are decided on and loaded into the warehouse, every month the oldest month will be "rolled off" the database, and the newest month added.

Ralph Kimball provided a much simpler definition of a data warehouse. As stated in his book, "The Data Warehouse Toolkit":

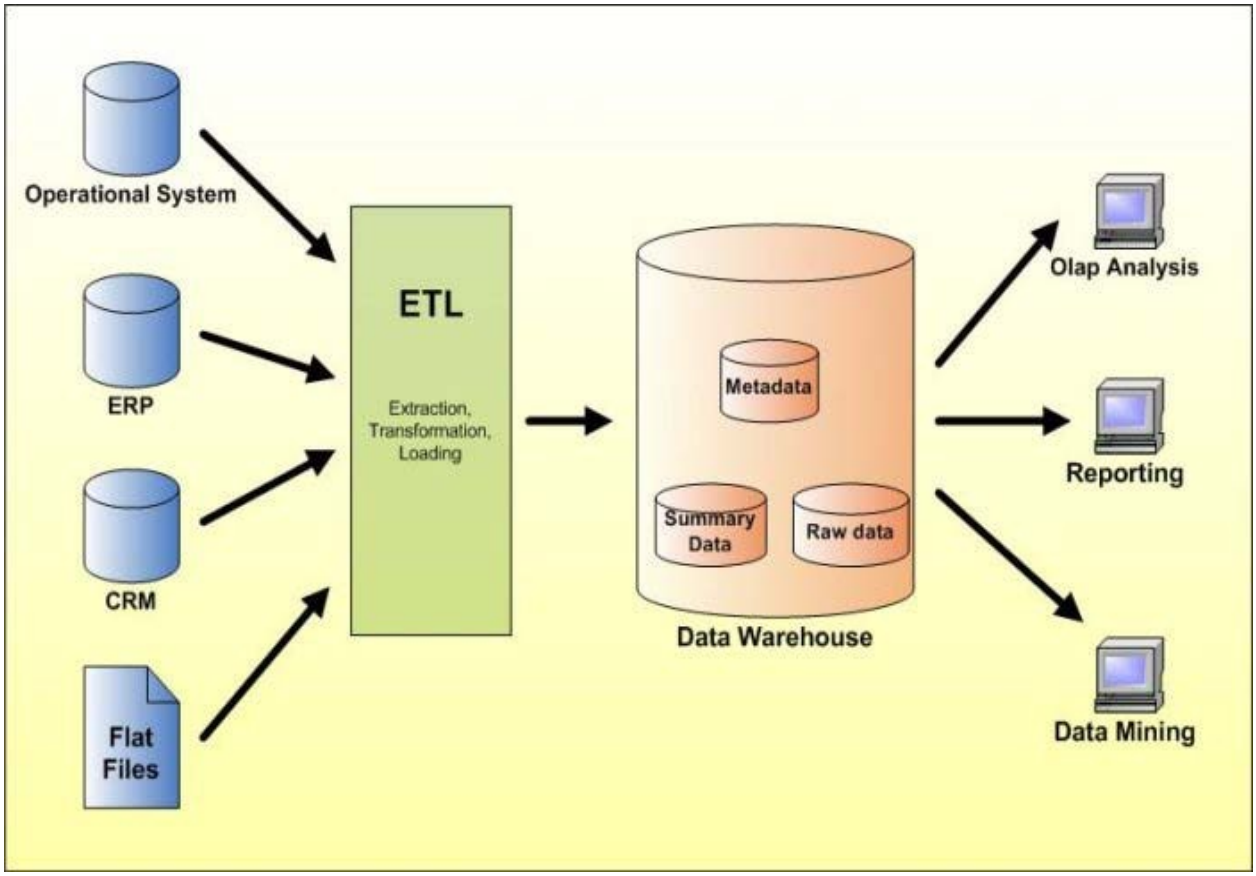
> A data warehouse is a copy of transaction data specifically structured for query and analysis.

This definition provides less insight and depth than Mr. Inmon's, but is no less accurate.

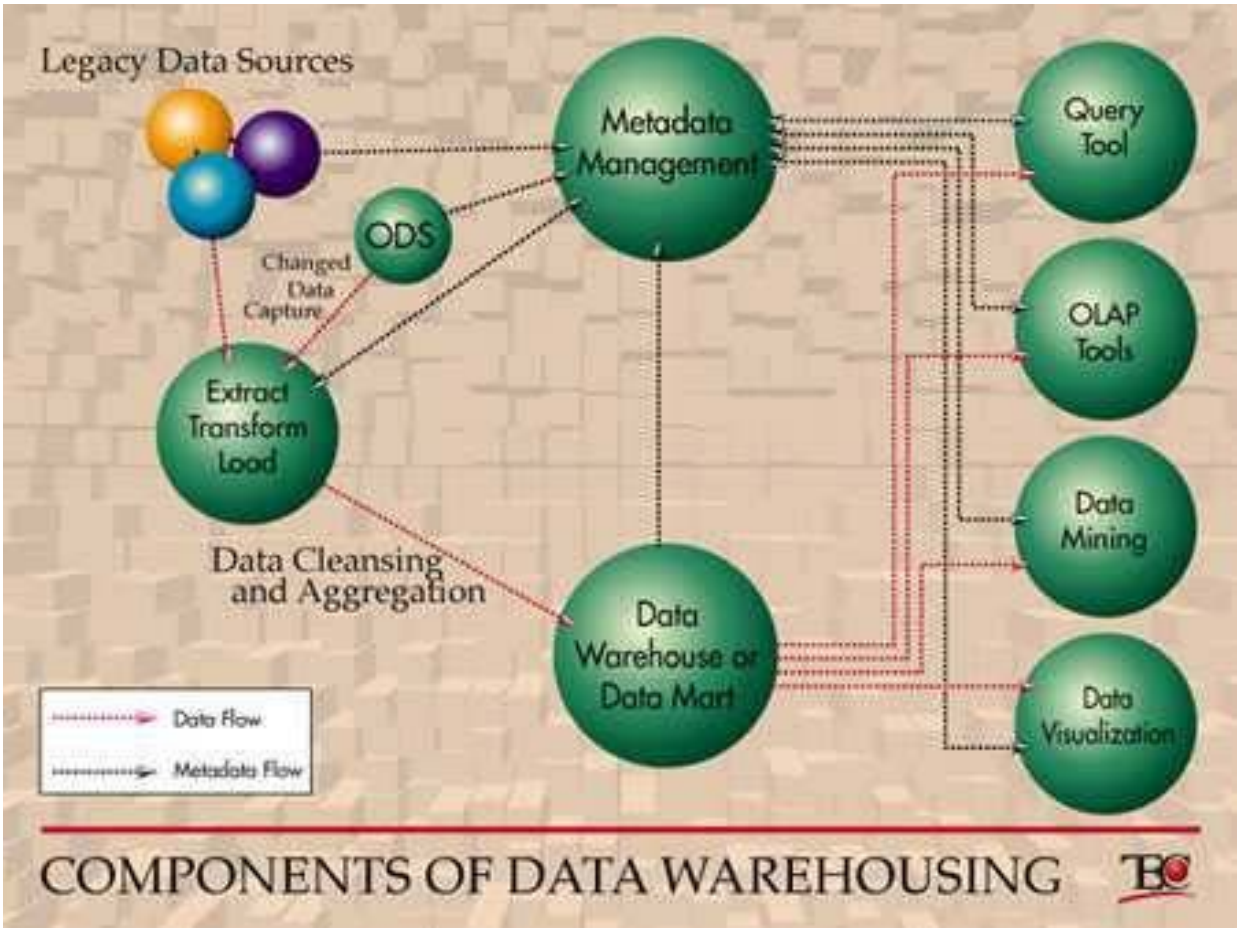
Another definition:

> A data warehouse is a repository (data & metadata) that contains integrated, cleansed, and reconciled data from disparate sources for decision support applications, with an emphasis on online analytical processing. Typically the data is multidimensional, historical, non volatile.

Data Warehouse Architecture



Components of Data Warehousing



Data Warehouse

Decision Support and OLAP

- Information technology to help the knowledge worker (executive, manager) make faster and better decisions.

  e.g. What were the sales volumes by region and product category for the last year?

  e.g. List the top 10 best selling products of each month in 1996

- **On-line analytical processing** (**OLAP**) is an element of **decision support systems** (**DSS**)

reference: VLDB'96 tutorial notes by Chauhuri & Dayal
VLDB'97 tutorial notes by Schneider

OLTP vs OLAP

- On-line transaction processing (OLTP)

	OLTP	OLAP
user	Clerk, IT professional	Knowledge worker
Function	Day to day operations	Decision support
DB design	Application oriented	Subject-oriented
Data	Current, up-to-date Detailed, Flat relational Isolated	Historical Summarized Multi-dimensional Integrated, consolidated
usage	Repetitive	Ad hoc
access	Read/Write Index/hash on Prim Key	Read mostly Lots of scans
unit of work	short, simple transaction	Complex queries
#records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	Trans throughput	Query throughput, response

Data Warehouse

- A decision support database that is maintained separately from the organization's operational databases.

- A data warehouse is

  - subject-oriented
  - integrated
  - time-varying
  - non-volatile

  Collection of data that is used primarily in organizationaldecision making.

Why separate Data Warehouse?

- Special data organization, access methods, and implementation methods are needed to support multi-dimensional views and typical operations of OLAP.

  e.g. total sales volume of beverages for the western region last year.

- Complex OLAP queries would degrade performance for operational transactions.

- Function

  - **Missing data**: DSS requires historical data, whichoperational DBs do not typically maintain.

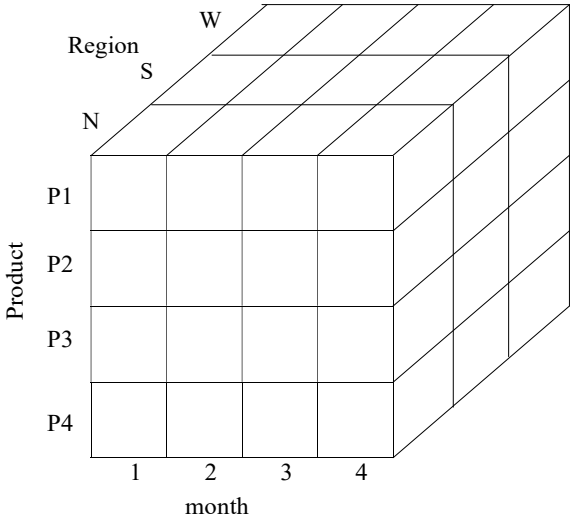
  - **Data consolidation**: DSS requires consolidation of data (aggregation, summarization) from many heterogeneous sources: operational DBs, external sources.

  - **Data quality**: different sources typically use inconsistent data representations, codes, and formats, which have to be reconciled.

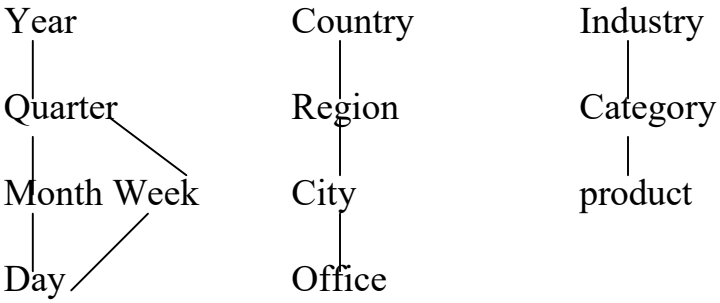
Multidimensional Data

- Sales volumes as a function of product, time, and geography.

- Product, time, and geography are **dimension attributes** and sales volume is a **measure attribute**.



- Dimensions usually have associated with them **hierarchies** that specify aggregation levels and hence granularity of viewing data.



Operations

- **Roll up**: Summarize data
  e.g. total sales volume last year by product category by region.

- **Drill down**, **Roll down**: go from higher level summary to lower level summary or detailed data

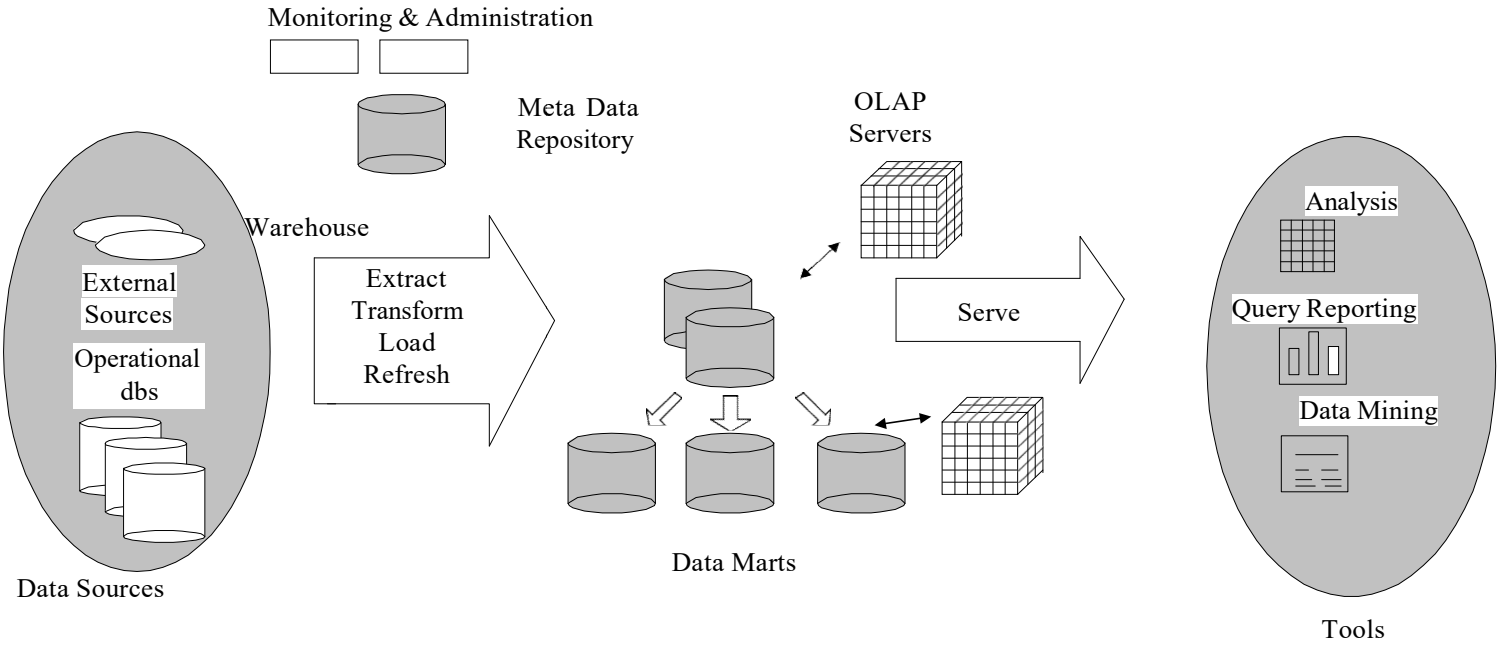
  e.g. For a particular product category, find detailed sales data for each office by date.

- **Slice** and **Dice**: select and project

  e.g. Sales of beverages in the west over the last 6 months.

- **Pivot**: rotate the **cube** to show a particular face

Data Warehousing Architecture



Two /Three – Tier Architecture

- **Warehouse database server**
  * almost always a relational DBMS rarely flat files.

- **OLAP servers**

  * **Relational OLAP** (**ROLAP**) extended relational DBMS that maps operations on multidimensional data to standard relational operations (GROUP BY operator)

  * **Multidimensional OLAP** (**MOLAP**) special purpose server that directly implement multidimensional data and operations

  * **Clients**
    - Query and reporting tools
    - Analysis tools
    - Data mining tools (e.g., trend analysis, prediction)

Warehousing Architecture

- **Enterprise Warehouse**: collects all information about subjects (customers, products, sales, assets, personnel) that span the entire enterprise

  - Requires extensive business modeling

  - May take years to design and build

- **Data Marts**: Departmental subsets that focus on selected subjects:

  e.g. marketing data mart: customer, sales, product

  - faster roll out, but complex integration in the long run

- **Virtual warehouse**: views over operational DBs

  - materialize some views (summaries)

  - easier to build

  - require excess capacity on operational DB servers

Operational Process

- **Data extraction**:
  tools, custom programs (scripts, wrappers)

  - extract data from each source

  - cleanse transform, and integrate data from different sources

- **Data load and refresh**:

  - load data into the warehouse: load utilities

  - periodically refresh warehouse to reflect updates.

  - periodically purge data from warehouse

- **Build derived data and views**

- **Service queries**

- **Monitor the warehouse**

Data Cleaning

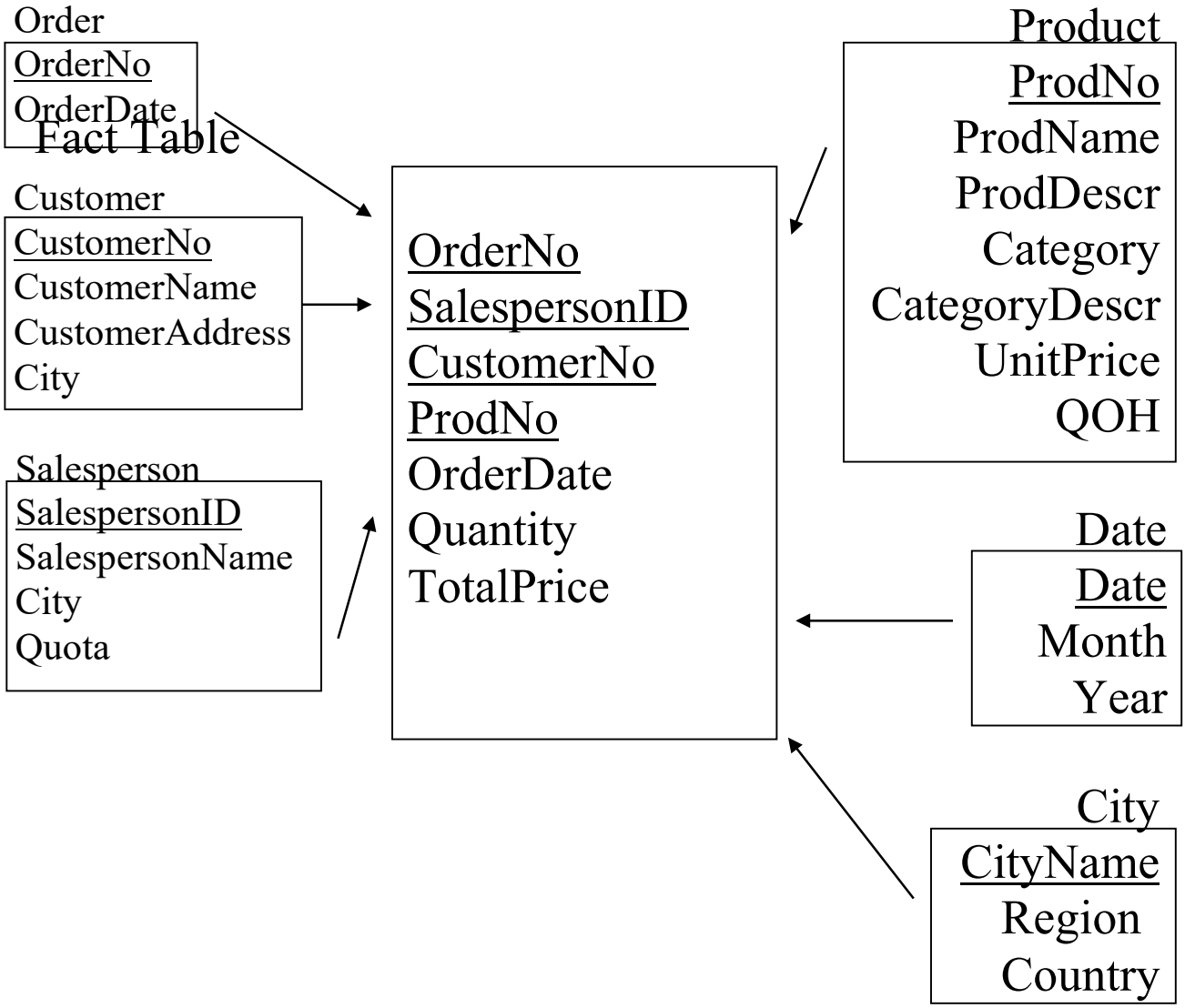
- Why ?
  - data warehouse contains data that is analyzed for business decisions
  - more data and multiple sources could mean more errors in the data and harder to trace such errors
  - Results in incorrect analysis
- Detecting data anomalies and rectifying them early has huge payoffs.
- Example:
  - inconsistent field lengths and orders
  - inconsistent description
  - inconsistent value assignments
  - missing entries
  - violation of integrity constraints

e.g. translate “gender” to sex”.

Warehouse Database Schema

- Star schema
- Snowflake schema
- Fact Constellation schema

Star Schema


Order
OrderNo
OrderDate
Fact Table
Customer
CustomerNo
CustomerName
CustomerAddress
City
Salesperson
SalespersonID
SalespersonName
City
Quota
OrderNo
SalespersonID
CustomerNo
ProdNo
OrderDate
Quantity
TotalPrice
Product
ProdNo
ProdName
ProdDescr
Category
CategoryDescr
UnitPrice
QOH
Date
Date
Month
Year
City
CityName
Region
Country

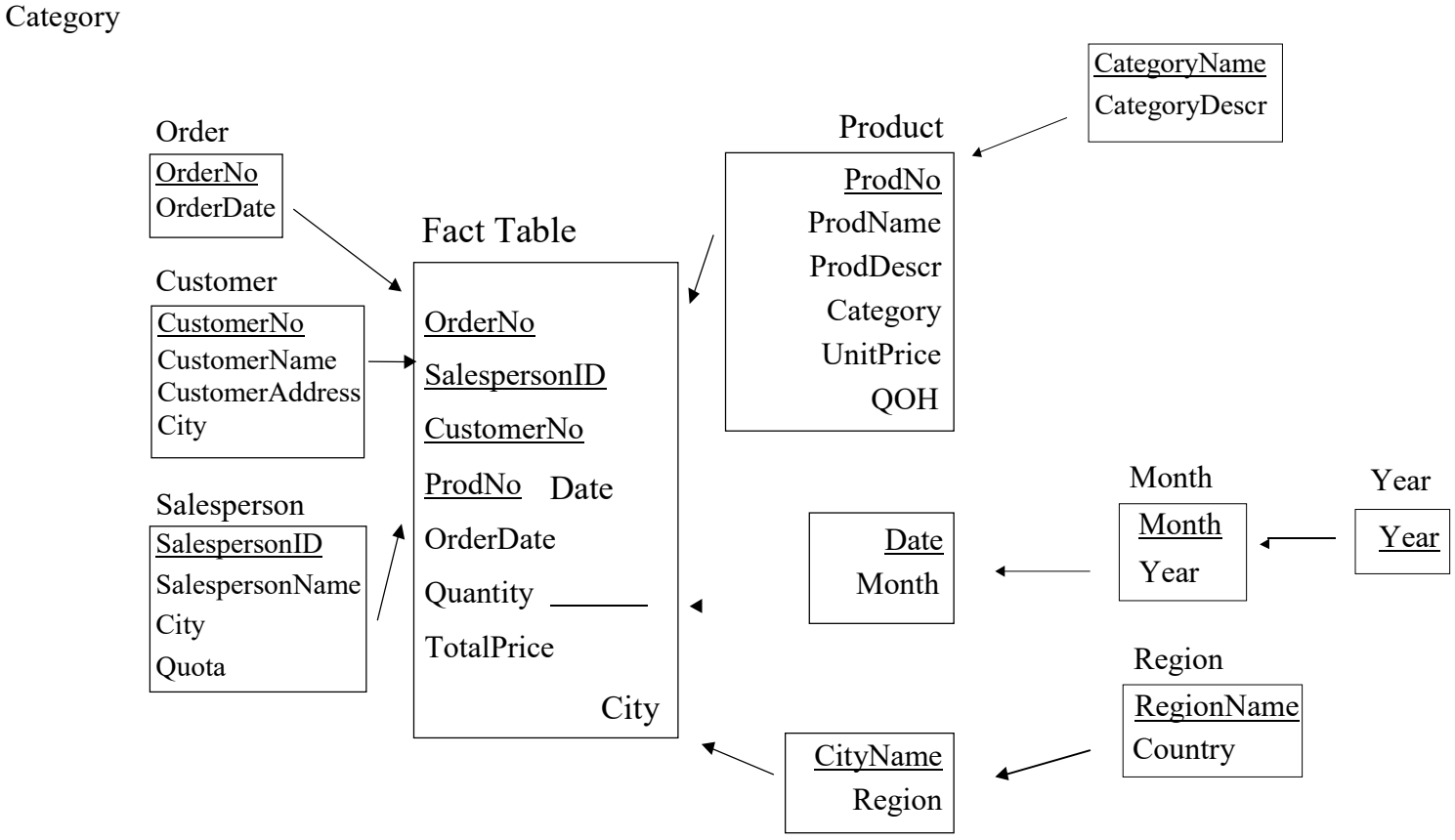
- A single **fact table** and for each dimension one single **dimension table**.

- Every fact points to one tuple in each of the dimension tables and has additional attributes

- Does not capture **hierarchies** directly

- Generated keys are used for performance and maintenance reasons.

Snowflake Schema



- Represent **dimensional hierarchies** directly by normalizing the dimension tables
- Easy to maintain
- Save storage, but it is alleged that it reduces effectiveness of browsing.

Fact Constellation

- multiple fact tables that share many dimension tables

  e.g. Projected expense and the actual expense may share dimension tables.

Aggregated Tables

- In addition to base fact and dimension tables, data warehouses keep aggregated (**summary**) data for efficiency.

- Two approaches:

  (1) store as separate **summary tables**
  - create corresponding "shrunken" dimension tables

  e.g. if a sales is aggregated by category of product, then the shrunken product table will have only the category information.

  (2) add to existing tables
  - use a "level" field to distinguish aggregate dimension - error prone.

Relational OLAP (ROLAP) servers

- Exploits service of relational engine effectively

  e.g. Microstrategy DSS server
  Infomix meta cube

- Key Functionality
  - Needs aggregation navigation logic
  - Ability to generate multi statement SQL
  - Optimize for each individual db backend

- Additional services:
  * cost based query and resource governor
    - detect runaway queries
    - schedule queries for throughput and response
    - cache management
  * design tool for DSS schema
    - storage can increase dramatically if precomputed views are not chosen properly.

    * performance analysis tool to pick aggregates to materialize.

    * data mart creates facilities on scheduled time or triggered by events and exception

    * some ROLAP products use their own storage structures for metadata

- domain specific ROLAP tools over server

- Disadvantages:

    * SQL comes in the way of sequential processing and columnar aggregations

    * such queries are hard to formulate and can often be time consuming to execute.

      e.g. changes in total sales from 1994 to 1995, aggregated by brand.

Multidimensional OLAP (MOLAP) servers

- The storage model is an **n-dimensional array**.
- Direct addressing abilities
- Front end multidimensional queries map to servers capabilities in a straightforward way.
- Problem: handling sparse data in array representation is expensive

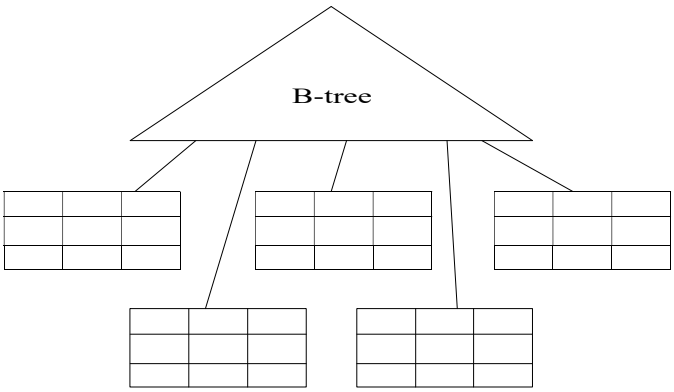
Product \ Date	1	2	3	4	5	6	7	8	sum
sum	30	40	20	20	30	40	10	20	210
P4	20	30				10			60
P3			20		10		10		40
P2	10			20		30		20	80
P1		10			20				30

- A straightforward array representation has good indexing properties but very poor storage utilization when data is sparse.

- A **2-level approach** works better

  - identify one or more two dimensional array structures that are dense.

  - index to these arrays by traditional indexing structures (e.g., B+ tree)



(2 –dimensional dense arrays)

  - 2-level approach increases storage utilization without sacrificing direct addressing capabilities for "most parts"

  - **Time** is often one of the dimensions included in the array structures.

Research Issues

- **Data cleaning**
  focus on data inconsistencies, not on schema inconsistencies

  e.g. Person names: Are the 2 names
  U. Dayal and Umeshwar Dayal
  refer to the same person

- **Data warehouse design**

  - design of summary tables and indexes
  - trade offs in indexing structures
  - business modeling

- **Query processing**

  - selecting appropriate summary tables
  - dynamic optimization with feed back
  - acid test for query optimization:
    estimation, use of transformations, search strategies
  - multi-way join algorithms, StarJoin, parallel hash join

- **Warehouse management**

  - detecting runaway queries
  - resource management
  - process management: scheduling queries, load and refresh
  - increment refresh techniques
    materialized view maintenance
  - failure and checkpoint issues in load and refresh
  - refreshing summary tables during load