Data Warehousing

Databases

- Databases are developed on the IDEA that DATA is one of the critical materials of the Information Age
- Information, which is created by data, becomes the bases for decision making

Decision Support Systems

- Created to facilitate the decision making process
- So much information that it is difficult to extract it all from a traditional database
- Need for a more comprehensive data storage facility
  - Data Warehouse

Decision Support Systems

- Extract Information from data to use as the basis for decision making
- Used at all levels of the Organization
- Tailored to specific business areas
- Interactive
- Ad Hoc queries to retrieve and display information
- Combines historical operation data with business activities

4 Components of DSS

- Data Store – The DSS Database
  - Business Data
  - Business Model Data
  - Internal and External Data
- Data Extraction and Filtering
  - Extract and validate data from the operational database and the external data sources

4 Components of DSS

- End-User Query Tool
  - Create Queries that access either the Operational or the DSS database
- End User Presentation Tools
  - Organize and Present the Data

Differences with DSS

- Operational databases
  - Stored in Normalized Relational Database
  - Support transactions that represent daily operations (Not Query Friendly)
- 3 Main Differences
  - Time Span
  - Granularity
  - Dimensionality

Time Span

- Operational databases
  - Real Time
  - Current Transactions
  - Short Time Frame
  - Specific Data Facts
- DSS
  - Historic
  - Long Time Frame (Months/Quarters/Years)
  - Patterns

Granularity

- Operational databases
  - Specific Transactions that occur at a given time
- DSS
  - Shown at different levels of aggregation
  - Different Summary Levels
  - Decompose (drill down)
  - Summarize (roll up)

Dimensionality

- Most distinguishing characteristic of DSS databases
- Operational
  - Represents atomic transactions
- DSS
  - Data is related in Many ways
  - Develop the larger picture
  - Multi-dimensional view of data

DSS Database Requirements

- DSS Database Scheme
  - Support Complex and Non-Normalized data
    - Summarized and Aggregate data
    - Multiple Relationships
    - Queries must extract multi-dimensional time slices
    - Redundant Data

DSS Database Requirements

- Data Extraction and Filtering
  - DSS databases are created mainly by extracting data from operational databases combined with data imported from external source
    - Need for advanced data extraction & filtering tools
    - Allow batch / scheduled data extraction
    - Support different types of data sources
    - Check for inconsistent data / data validation rules
    - Support advanced data integration / data formatting conflicts

DSS Database Requirements

- End User Analytical Interface
  - Must support advanced data modeling and data presentation tools
  - Data analysis tools
  - Query generation
  - Must Allow the User to Navigate through the DSS
- Size Requirements
  - VERY Large – Terabytes
  - Advanced Hardware (Multiple processors, multiple disk arrays, etc.)

Data Warehousing Concepts

- DSS – friendly data repository for the DSS is the DATA WAREHOUSE
- A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing.
- It usually contains historical data derived from transaction data,
- but it can include data from other sources.

Data Warehousing Concepts

- It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.
- In addition to a relational database, a data warehouse environment includes

Data Warehousing Concepts

- an extraction, transportation, transformation, and loading (ETL) solution,
- an online analytical processing (OLAP) engine,
- client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.

Data Warehousing Concepts

- A common way of introducing data warehousing is to refer to the characteristics of a data warehouse as follows
- Subject Oriented
- Integrated
- Nonvolatile
- Time Variant

Subject Oriented

- Data warehouses are designed to help you analyze data.
- For example, to learn more about your company's sales data,
- you can build a warehouse that concentrates on sales.

Subject Oriented

- Using this warehouse, you can answer questions like
- "Who was our best customer for this item last year?“
- This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

Integrated

- Integration is closely related to subject orientation.
- Data warehouses must put data from disparate sources into a consistent format.
- They must resolve such problems as naming conflicts and inconsistencies among units of measure.
- When they achieve this, they are said to be integrated.

Nonvolatile

- Nonvolatile means that, once entered into the warehouse, data should not change.
- This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.

Time Variant

- In order to discover trends in business,
- analysts need large amounts of data.
- This is very much in contrast to online transaction processing (OLTP) systems,
- where performance requirements demand that historical data be moved to an archive.
- A data warehouse's focus on change over time is what is meant by the term time variant.

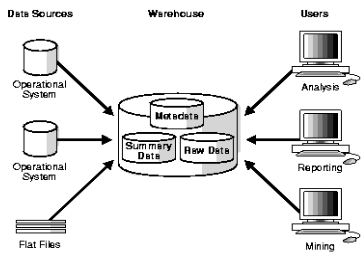
Data Warehouse Architectures

- Three common architectures are:
- Data Warehouse Architecture (Basic)
- Data Warehouse Architecture (with a Staging Area)
- Data Warehouse Architecture (with a Staging Area and Data Marts)

Data Warehouse Architecture (**Basic)**

- Figure below shows a simple architecture for a data warehouse.
- End users directly access data derived from several source systems through the data warehouse.

Architecture of a Data Warehouse



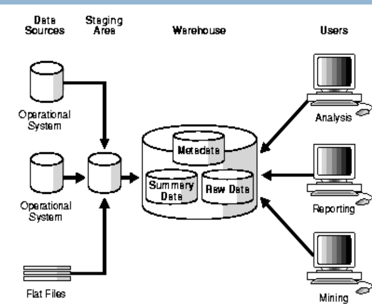
Data Warehouse Architecture (**Basic**)

- In Figure above, the metadata and raw data of a traditional OLTP system is present,
- as is an additional type of data, summary data.
- Summaries are very valuable in data warehouses because they pre-compute long operations in advance.
- For example, a typical data warehouse query is to retrieve something like August sales.
- A summary in Oracle is called a materialized view

with a Staging Area

- In basic architecture, you need to clean and process your operational data before putting it into the warehouse.
- You can do this programmatically, although most data warehouses use a staging area instead.
- A staging area simplifies building summaries and general warehouse management.
- Figure below illustrates this typical architecture.

Architecture of a Data Warehouse with a Staging Area



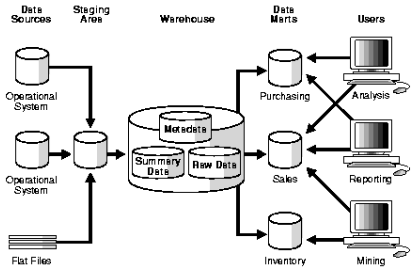
with a Staging Area and Data Marts

- Although the architecture in Figure above is quite common,
- you may want to customize your warehouse's architecture for different groups within your organization.
- You can do this by adding **data marts**,

with a Staging Area and Data Marts

- which are systems designed for a particular line of business.
- Figure below illustrates an example where purchasing, sales, and inventories are separated.
- In this example, a financial analyst might want to analyze historical data for purchases and sales.

Architecture of a Data Warehouse with a Staging Area and Data Marts



Logical Design in Data Warehouses

- Logical Versus Physical Design in Data Warehouses
- Creating a Logical Design
- Data Warehousing Schemas
- Data Warehousing Objects

Logical Versus Physical Design in Data Warehouses

- The logical design is more conceptual and abstract than the physical design.
- In the logical design, you look at the logical relationships among the objects.
- In the physical design, you look at the most effective way of storing and retrieving the objects

Logical Versus Physical Design in Data Warehouses

- as well as handling them from a transportation and backup/recovery perspective.
- Orient your design toward the needs of the end users.
- End users typically want to perform analysis and look at aggregated data,
- rather than at individual transactions.

Logical Versus Physical Design in Data Warehouses

- However, end users might not know what they need until they see it.
- In addition, a well-planned design allows for growth and changes as the needs of users change and evolve.
- By beginning with the logical design, you focus on the information requirements and save the implementation details for later.

Creating a Logical Design

- A logical design is conceptual and abstract.
- You do not deal with the physical implementation details yet.
- You deal only with defining the types of information that you need.
- One technique you can use to model your organization's logical information requirements is entity-relationship modeling.

Creating a Logical Design

- Entity-relationship modeling involves identifying the things of importance (entities),
- the properties of these things (attributes),
- and how they are related to one another (relationships).
- The process of logical design involves arranging data into a series of logical relationships called entities and attributes.

Creating a Logical Design

- An entity represents a chunk of information.
- In relational databases, an entity often maps to a table.
- An attribute is a component of an entity that helps define the uniqueness of the entity.
- In relational databases, an attribute maps to a column.

Creating a Logical Design

- To be sure that your data is consistent, you need to use unique identifiers.
- A unique identifier is something you add to tables so that you can differentiate between the same item when it appears in different places.
- In a physical design, this is usually a primary key.

Creating a Logical Design

- While entity-relationship diagramming has traditionally been associated with highly normalized models such as OLTP applications,
- the technique is still useful for data warehouse design in the form of dimensional modeling.
- In dimensional modeling, instead of seeking to discover atomic units of information (such as entities and attributes) and

Creating a Logical Design

- all of the relationships between them,
- you identify which information belongs to a central fact table and which information belongs to its associated dimension tables.
- You identify business subjects or fields of data, define relationships between business subjects, and name the attributes for each subject.

Creating a Logical Design

- Your logical design should result in
- (1) a set of entities and attributes corresponding to fact tables and dimension tables and
- (2) a model of operational data from your source into subject-oriented information in your target data warehouse schema.

Creating a Logical Design

- You can create the logical design using a pen and paper,
- or you can use a design tool such as Oracle Warehouse Builder (specifically designed to support modeling the ETL process)
- or Oracle Designer (a general purpose modeling tool).

Data Warehousing Schemas

- A schema is a collection of database objects, including tables, views, indexes, and synonyms.
- You can arrange schema objects in the schema models designed for data warehousing in a variety of ways.
- Most data warehouses use a dimensional model.

Data Warehousing Schemas

- The model of your source data and the requirements of your users help you design the data warehouse schema.
- You can sometimes get the source model from your company's enterprise data model and reverse-engineer the logical data model for the data warehouse from this.

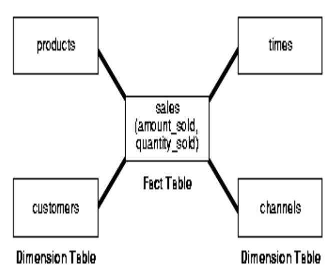
Data Warehousing Schemas

- The physical implementation of the logical data warehouse model may require some changes to adapt it to your system parameters-
- size of machine, number of users, storage capacity, type of network, and software

Data Warehousing Schemas

- The star schema is the simplest data warehouse schema.
- It is called a star schema because the diagram resembles a star,
- with points radiating from a center.
- The center of the star consists of one or more fact tables and the points of the star are the dimension tables, as shown in Figure below.

The star schema



The star schema

- The most natural way to model a data warehouse is as a star schema,
- only one join establishes the relationship between the fact table and any one of the dimension tables.
- A star schema optimizes performance by keeping queries simple and providing fast response time.
- All the information about each level is stored in one row

Data Warehousing Objects

- Fact tables and dimension tables are the two types of objects commonly used in dimensional data warehouse schemas.
- Fact tables are the large tables in your warehouse schema that store business measurements.
- Fact tables typically contain facts and foreign keys to the dimension tables.

Data Warehousing Objects

- Fact tables represent data, usually numeric and additive, that can be analyzed and examined.
- Examples include sales, cost, and profit.
- Dimension tables, also known as lookup or reference tables, contain the relatively static data in the warehouse.

Data Warehousing Objects

- Dimension tables store the information you normally use to contain queries.
- Dimension tables are usually textual and descriptive and you can use them as the row headers of the result set.
- Examples are customers or products.

Fact Tables

- A fact table typically has two types of columns:
- those that contain numeric facts (often called measurements), and those that are foreign keys to dimension tables.
- A fact table contains either detail-level facts or facts that have been aggregated.

Fact Tables

- Fact tables that contain aggregated facts are often called summary tables.
- A fact table usually contains facts with the same level of aggregation.
- Though most facts are additive, they can also be semi-additive or non-additive.

Fact Tables

- Additive facts can be aggregated by simple arithmetical addition.
- A common example of this is sales.
- Non-additive facts cannot be added at all.

Fact Tables

- An example of this is averages.
- Semi-additive facts can be aggregated along some of the dimensions and not along others.
- An example of this is inventory levels, where you cannot tell what a level means simply by looking at it.

Creating a New Fact Table

- You must define a fact table for each star schema.
- From a modeling standpoint, the primary key of the fact table is usually a composite key that is made up of all of its foreign keys.