BUILDING BLOCKS OF DATAWAREHOUSE

G.Lakshmi Priya & Razia Sultana.A
Assistant Professor/IT
Data Warehouse—Subject-Oriented

- Organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.
Data Warehouse—Subject-Oriented

• Organized around major subjects, such as customer, product, sales.
• Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
• Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.
Data Warehouse—Subject-Oriented

• Organized around major subjects, such as customer, product, sales.

• Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.

• Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.
Data Warehouse - Integrated

• Constructed by integrating multiple, heterogeneous data sources
  – relational databases, flat files, on-line transaction records
• Data cleaning and data integration techniques are applied.
  – Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
    • E.g., Hotel price: currency, tax, breakfast covered, etc.
  – When data is moved to the warehouse, it is converted.
Data Warehouse -Time Variant

• The time horizon for the data warehouse is significantly longer than that of operational systems.
  – Operational database: current value data.
  – Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)

• Every key structure in the data warehouse
  – Contains an element of time, explicitly or implicitly
  – But the key of operational data may or may not contain “time element”.
Data Warehouse - Non Updatable

• A physically separate store of data transformed from the operational environment.

• Operational update of data does not occur in the data warehouse environment.
  – Does not require transaction processing, recovery, and concurrency control mechanisms.
  – Requires only two operations in data accessing:
    • *initial loading of data* and *access of data*. 
Need for Data Warehousing

• Integrated, company-wide view of high-quality information (from disparate databases)
• Separation of operational and informational systems and data (for improved performance)

Table 11-1: comparison of operational and informational systems

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Operational Systems</th>
<th>Informational Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary purpose</td>
<td>Run the business on a current basis</td>
<td>Support managerial decision making</td>
</tr>
<tr>
<td>Type of data</td>
<td>Current representation of state of the business</td>
<td>Historical point-in-time (snapshots) and predictions</td>
</tr>
<tr>
<td>Primary users</td>
<td>Clerks, salespersons, administrators</td>
<td>Managers, business analysts, customers</td>
</tr>
<tr>
<td>Scope of usage</td>
<td>Narrow, planned, and simple updates and queries</td>
<td>Broad, ad hoc, complex queries and analysis</td>
</tr>
<tr>
<td>Design goal</td>
<td>Performance throughput, availability</td>
<td>Ease of flexible access and use</td>
</tr>
<tr>
<td>Volume</td>
<td>Many, constant updates and queries on one or a few table rows</td>
<td>Periodic batch updates and queries requiring many or all rows</td>
</tr>
</tbody>
</table>
Need to separate operational and information systems

Three primary factors:

– A data warehouse centralizes data that are scattered throughout disparate operational systems and makes them available for DS.

– A well-designed data warehouse adds value to data by improving their quality and consistency.

– A separate data warehouse eliminates much of the contention for resources that results when information applications are mixed with operational processing.
Data Warehouse Architectures

- 1. Generic Two-Level Architecture
- 2. Independent Data Mart
- 3. Dependent Data Mart and Operational Data Store
- 4. Logical Data Mart and Active Warehouse
- 5. Three-Layer architecture

All involve some form of *extraction, transformation* and *loading* (ETL)
Generic two-level architecture

Periodic extraction → data is not completely current in warehouse

One, company-wide warehouse
Independent Data Mart

**Data marts:**
Mini-warehouses, limited in scope

Separate ETL for each *independent* data mart

Data access complexity due to *multiple* data marts
Independent Data mart

- Independent data mart: a data mart filled with data extracted from the operational environment without benefits of a data warehouse.
Dependent data mart with operational data store

**ODS** provides option for obtaining **current** data.

Single ETL for **enterprise data warehouse** *(EDW)*

Dependent data marts loaded from EDW
Dependent data mart-Operational data store

• **Dependent data mart**: A data mart filled exclusively from the enterprise data warehouse and its reconciled data.

• **Operational data store (ODS)**: An integrated, subject-oriented, updatable, current-valued, enterprisewise, detailed database designed to serve operational users as they do decision support processing.
Logical data mart

ODS and data warehouse are one and the same.

Data marts are NOT separate databases, but logical views of the data warehouse
⇒ Easier to create new data marts

Near real-time ETL for active Data Warehouse
**Active data warehouse**

- **Active data warehouse**: An enterprise data warehouse that accepts near-real-time feeds of transactional data from the systems of record, analyzes warehouse data, and in near-real-time relays business rules to the data warehouse and systems of record so that immediate actions can be taken in response to business events.
## Table 11-2: Data Warehouse Versus Data Mart

<table>
<thead>
<tr>
<th>Data Warehouse</th>
<th>Data Mart</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope</strong></td>
<td><strong>Scope</strong></td>
</tr>
<tr>
<td>• Application independent</td>
<td>• Specific DSS application</td>
</tr>
<tr>
<td>• Centralized, possibly enterprise-wide</td>
<td>• Decentralized by user area</td>
</tr>
<tr>
<td>• Planned</td>
<td>• Organic, possibly not planned</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td><strong>Data</strong></td>
</tr>
<tr>
<td>• Historical, detailed, and summarized</td>
<td>• Some history, detailed, and summarized</td>
</tr>
<tr>
<td>• Lightly denormalized</td>
<td>• Highly denormalized</td>
</tr>
<tr>
<td><strong>Subjects</strong></td>
<td><strong>Subjects</strong></td>
</tr>
<tr>
<td>• Multiple subjects</td>
<td>• One central subject of concern to users</td>
</tr>
<tr>
<td><strong>Sources</strong></td>
<td><strong>Sources</strong></td>
</tr>
<tr>
<td>• Many internal and external sources</td>
<td>• Few internal and external sources</td>
</tr>
<tr>
<td><strong>Other Characteristics</strong></td>
<td><strong>Other Characteristics</strong></td>
</tr>
<tr>
<td>• Flexible</td>
<td>• Restrictive</td>
</tr>
<tr>
<td>• Data-oriented</td>
<td>• Project-oriented</td>
</tr>
<tr>
<td>• Long life</td>
<td>• Short life</td>
</tr>
<tr>
<td>• Large</td>
<td>• Start small, becomes large</td>
</tr>
<tr>
<td>• Single complex structure</td>
<td>• Multi, semi-complex structures, together complex</td>
</tr>
</tbody>
</table>

Adapted from Strange (1997)
Three-layer architecture
Three-layer architecture
Reconciled and derived data

• *Reconciled data*: detailed, current data intended to be the single, authoritative source for all decision support.

• *Derived data*: Data that have been selected, formatted, and aggregated for end-user decision support application.

• *Metadata*: technical and business data that describe the properties or characteristics of other data.
Data Characteristics

Status vs. Event Data

Example of DBMS log entry:

Status

Event = a database action (create/update/delete) that results from a transaction
Data Characteristics
Transient vs. Periodic Data

Figure 11-8:
Transient operational data

Changes to existing records are written over previous records, thus destroying the previous data content
Data Characteristics
Transient vs. Periodic Data

Figure 11-9:
Periodic warehouse data

Data are never physically altered or deleted once they have been added to the store
Other data warehouse changes

• New descriptive attributes
• New business activity attributes
• New classes of descriptive attributes
• Descriptive attributes become more refined
• Descriptive data are related to one another
• New source of data
Data Reconciliation

• Typical operational data is:
  – Transient – not historical
  – Not normalized (perhaps due to denormalization for performance)
  – Restricted in scope – not comprehensive
  – Sometimes poor quality – inconsistencies and errors

• After ETL, data should be:
  – Detailed – not summarized yet
  – Historical – periodic
  – Normalized – 3rd normal form or higher
  – Comprehensive – enterprise-wide perspective
  – Quality controlled – accurate with full integrity
Record-level functions & Field-level functions

- **Record-level functions**
  - Selection: data partitioning
  - Joining: data combining
  - Normalization
  - Aggregation: data summarization

- **Field-level functions**
  - Single-field transformation: from one field to one field
  - Multi-field transformation: from many fields to one, or one field to many
In general – some transformation function translates data from old form to new form

Algorithmic transformation uses a formula or logical expression

Table lookup – another approach
Multifield transformation

M:1 – from many source fields to one target field

1:M – from one source field to many target fields
Derived Data

• **Objectives**
  – Ease of use for decision support applications
  – Fast response to predefined user queries
  – Customized data for particular target audiences
  – Ad-hoc query support
  – Data mining capabilities

• **Characteristics**
  – Detailed (mostly periodic) data
  – Aggregate (for summary)
  – Distributed (to departmental servers)

Most common data model = **star schema**
(also called “dimensional model”)
The Star Schema

• *Star schema*: is a simple database design in which dimensional (describing how data are commonly aggregated) are separated from fact or event data.

• A star schema consists of two types of tables: *fact tables* and *dimension table*. 
Components of a star schema

**Fact tables** contain factual or quantitative data.

**Dimension tables** contain descriptions about the subjects of the business.

1:N relationship between dimension tables and fact tables.

Dimension tables are denormalized to maximize performance.

Excellent for ad-hoc queries, but bad for online transaction processing.
Star schema example

**Fact table** provides statistics for sales broken down by product, period and store dimensions
Star schema with sample data

<table>
<thead>
<tr>
<th>Product Code</th>
<th>Description</th>
<th>Color</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Sweater</td>
<td>Blue</td>
<td>40</td>
</tr>
<tr>
<td>110</td>
<td>Shoes</td>
<td>Brown</td>
<td>10</td>
</tr>
<tr>
<td>125</td>
<td>Gloves</td>
<td>Tan</td>
<td>1/2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Period Code</th>
<th>Year</th>
<th>Quarter</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>1999</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>002</td>
<td>1999</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>003</td>
<td>1999</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Code</th>
<th>Period Code</th>
<th>Store Code</th>
<th>Units Sold</th>
<th>Dollars Sold</th>
<th>Dollars Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>002</td>
<td>S1</td>
<td>30</td>
<td>1500</td>
<td>1200</td>
</tr>
<tr>
<td>125</td>
<td>003</td>
<td>S2</td>
<td>50</td>
<td>1000</td>
<td>600</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
<td>S1</td>
<td>40</td>
<td>1600</td>
<td>1000</td>
</tr>
<tr>
<td>110</td>
<td>002</td>
<td>S3</td>
<td>40</td>
<td>2000</td>
<td>1200</td>
</tr>
<tr>
<td>100</td>
<td>003</td>
<td>S2</td>
<td>30</td>
<td>1200</td>
<td>750</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Store Code</th>
<th>Store Name</th>
<th>City</th>
<th>Telephone</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Jan’s</td>
<td>San Antonio</td>
<td>683-192-1400</td>
<td>Burgess</td>
</tr>
<tr>
<td>S2</td>
<td>Bill’s</td>
<td>Portland</td>
<td>943-681-2135</td>
<td>Thomas</td>
</tr>
<tr>
<td>S3</td>
<td>Ed’s</td>
<td>Boulder</td>
<td>417-195-8037</td>
<td>Perry</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Issues Regarding Star Schema

• Dimension table keys must be *surrogate* (non-intelligent and non-business related), because:
  – Keys may change over time
  – Length/format consistency

• Granularity of Fact Table – what level of detail do you want?
  – Transactional grain – finest level
  – Aggregated grain – more summarized
  – Finer grains ➔ better *market basket analysis* capability
  – Finer grain ➔ more dimension tables, more rows in fact table
• **Duration of the database**
  - Ex: 13 months or 5 quarters
  - Some businesses need for a longer durations.

• **Size of the fact table**
  - Estimate the number of possible values for each dimension associated with the fact table.
  - Multiply the values obtained in the first step after making any necessary adjustments.
Modeling dates

Fact tables contain time-period data

Date dimensions are important
Variations of the Star Schema

1. Multiple fact tables
2. Factless fact tables
3. Normalizing Dimension Tables
4. Snowflake schema
Multiple Fact tables

• More than one fact table in a given star schema.
• Ex: There are 2 fact tables, one at the center of each star:
  – *Sales* – facts about the sale of a product to a customer in a store on a date.
  – *Receipts* - facts about the receipt of a product from a vendor to a warehouse on a date.
  – Two separate *product* dimension tables have been created.
  – One *date* dimension table is used.
Factless Fact Tables

• There are applications in which fact tables do not have nonkey data but that do have foreign keys for the associated dimensions.

• The two situations:
  – To track events
  – To inventory the set of possible occurrences (called coverage)
Factless fact table showing occurrence of an event.
Factless fact table showing coverage
Normalizing dimension tables

• Dimension tables may not be normalized. Most data warehouse experts find this acceptable.

• In some situations in which it makes sense to further normalize dimension tables.

• Multivalued dimensions:
  – Ex: Hospital charge/payment for a patient on a date is associated with one or more diagnosis.
  – N:M relationship between the Diagnosis and Finances fact table.
  – Solution: create an associative entity (helper table) between Diagnosis and Finances.
Multivalued dimension
Snowflake schema

• *Snowflake schema* is an expanded version of a star schema in which dimension tables are normalized into several related tables.

• Advantages
  – Small saving in storage space
  – Normalized structures are easier to update and maintain

• Disadvantages
  – Schema less intuitive
  – Ability to browse through the content difficult
  – Degraded query performance because of additional joins.
Example of snowflake schema

**Time**
- time_key
- day
- day_of_the_week
- month
- quarter
- year

**Sales Fact Table**
- time_key
- item_key
- branch_key
- location_key
- units_sold
- dollars_sold
- avg_sales

**Item**
- item_key
- item_name
- brand
- type
- supplier_key

**Supplier**
- supplier_key
- supplier_type

**Location**
- location_key
- street
- city_key

**City**
- city_key
- city
- province_or_state
- country
The User Interface

- A variety of tools are available to query and analyze data stored in data warehouses.
  - 1. Querying tools
  - 2. On-line Analytical processing (OLAP, MOLAP, ROLAP) tools
  - 3. Data Mining tools
  - 4. Data Visualization tools
Role of Metadata (data catalog)

• Identify subjects of the data mart
• Identify dimensions and facts
• Indicate how data is derived from enterprise data warehouses, including derivation rules
• Indicate how data is derived from operational data store, including derivation rules
• Identify available reports and predefined queries
• Identify data analysis techniques (e.g. drill-down)
• Identify responsible people
Querying Tools

• SQL is not an analytical language
• SQL-99 includes some data warehousing extensions
• SQL-99 still is not a full-featured data warehouse querying and analysis tool.
• Different DBMS vendors will implement some or all of the SQL-99 OLAP extension commands and possibly others.
On-Line Analytical Processing (OLAP)

• OLAP is the use of a set of graphical tools that provides users with multidimensional views of their data and allows them to analyze the data using simple windowing techniques

• **Relational OLAP (ROLAP)**
  – OLAP tools that view the database as a traditional relational database in either a star schema or other normalized or denormalized set of tables.

• **Multidimensional OLAP (MOLAP)**
  – OLAP tools that load data into an intermediate structure, usually a three or higher dimensional array. (Cube structure)
From tables to data cubes

- A data warehouse is based on a **multidimensional data model** which views data in the form of a **data cube**

- A **data cube**, such as **sales**, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as **item** *(item_name, brand, type)*, or **time** *(day, week, month, quarter, year)*
  - Fact table contains measures (such as **dollars_sold**) and keys to each of the related dimension tables
MOLAP Operations

- Roll up (drill-up): summarize data
  - by climbing up hierarchy or by dimension reduction
- Drill down (roll down): reverse of roll-up
  - from higher level summary to lower level summary or detailed data, or introducing new dimensions
- Slice and dice:
  - project and select
Figure 11-22: Slicing a data cube
Figure 11-23: Example of drill-down

Summary report

<table>
<thead>
<tr>
<th>Brand</th>
<th>Package size</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>SofTowel</td>
<td>2-pack</td>
<td>$75</td>
</tr>
<tr>
<td>SofTowel</td>
<td>3-pack</td>
<td>$100</td>
</tr>
<tr>
<td>SofTowel</td>
<td>6-pack</td>
<td>$50</td>
</tr>
</tbody>
</table>

Drill-down with color added

<table>
<thead>
<tr>
<th>Brand</th>
<th>Package size</th>
<th>Color</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>SofTowel</td>
<td>2-pack</td>
<td>White</td>
<td>$30</td>
</tr>
<tr>
<td>SofTowel</td>
<td>2-pack</td>
<td>Yellow</td>
<td>$25</td>
</tr>
<tr>
<td>SofTowel</td>
<td>2-pack</td>
<td>Pink</td>
<td>$20</td>
</tr>
<tr>
<td>SofTowel</td>
<td>3-pack</td>
<td>White</td>
<td>$50</td>
</tr>
<tr>
<td>SofTowel</td>
<td>3-pack</td>
<td>Green</td>
<td>$25</td>
</tr>
<tr>
<td>SofTowel</td>
<td>3-pack</td>
<td>Yellow</td>
<td>$25</td>
</tr>
<tr>
<td>SofTowel</td>
<td>6-pack</td>
<td>White</td>
<td>$30</td>
</tr>
<tr>
<td>SofTowel</td>
<td>6-pack</td>
<td>Yellow</td>
<td>$20</td>
</tr>
</tbody>
</table>
Data Mining

• Data mining is knowledge discovery using a blend of statistical, AI, and computer graphics techniques

• Goals:
  – Explain observed events or conditions
  – Confirm hypotheses
  – Explore data for new or unexpected relationships

• Techniques
  – Case-based reasoning
  – Rule discovery
  – Signal processing
  – Neural nets
  – Fractals
Data Visualization

• Data visualization is the representation of data in graphical/multimedia formats for human analysis
OLAP tool Vendors

• IBM
• Informix
• Cartelon
• NCR
• Oracle (Oracle Warehouse builder, Oracle OLAP)
• Red Brick
• Sybase
• SAS
• Microsoft (SQL Server OLAP)
• Microstrategy Corporation
Data Warehouse vs. Operational DBMS

• OLTP (on-line transaction processing)
  – Major task of traditional relational DBMS
  – Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.

• OLAP (on-line analytical processing)
  – Major task of data warehouse system
  – Data analysis and decision making

• Distinct features (OLTP vs. OLAP):
  – User and system orientation: customer vs. market
  – Data contents: current, detailed vs. historical, consolidated
  – Database design: ER + application vs. star + subject
  – View: current, local vs. evolutionary, integrated
  – Access patterns: update vs. read-only but complex queries
# OLTP vs. OLAP

<table>
<thead>
<tr>
<th></th>
<th>OLTP</th>
<th>OLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>users</strong></td>
<td>clerk, IT professional</td>
<td>knowledge worker</td>
</tr>
<tr>
<td><strong>function</strong></td>
<td>day to day operations</td>
<td>decision support</td>
</tr>
<tr>
<td><strong>DB design</strong></td>
<td>application-oriented</td>
<td>subject-oriented</td>
</tr>
<tr>
<td><strong>data</strong></td>
<td>current, up-to-date detailed, flat relational</td>
<td>historical, summarized, multidimensional</td>
</tr>
<tr>
<td></td>
<td>isolated</td>
<td>integrated, consolidated</td>
</tr>
<tr>
<td><strong>usage</strong></td>
<td>repetitive</td>
<td>ad-hoc</td>
</tr>
<tr>
<td><strong>access</strong></td>
<td>read/write</td>
<td>lots of scans</td>
</tr>
<tr>
<td></td>
<td>index/hash on prim. key</td>
<td></td>
</tr>
<tr>
<td><strong>unit of work</strong></td>
<td>short, simple transaction</td>
<td>complex query</td>
</tr>
<tr>
<td><strong># records accessed</strong></td>
<td>tens</td>
<td>millions</td>
</tr>
<tr>
<td><strong>#users</strong></td>
<td>thousands</td>
<td>hundreds</td>
</tr>
<tr>
<td><strong>DB size</strong></td>
<td>100MB-GB</td>
<td>100GB-TB</td>
</tr>
<tr>
<td><strong>metric</strong></td>
<td>transaction throughput</td>
<td>query throughput, response</td>
</tr>
</tbody>
</table>