Data Warehouse and OLAP

Week 5
Midterm I

- Friday, March 4
- Scope
  - Homework assignments 1 – 4
  - Open book
Team Homework Assignment #7

- Do Examples 3.8, 3.10 and Exercise 3.4 (b) and (c). Prepare for the results of the homework assignment.
- Due date
  - beginning of the lecture on Friday March 11th.
Topics

• Definition of data warehouse
• Multidimensional data model
• Data warehouse architecture
• From data warehousing to data mining
What is Data Warehouse? (1)

- A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and that usually resides at a single site.
- A data warehouse is a semantically consistent data store that serves as a physical implementation of a decision support data model and stores the information on which an enterprise need to make strategic decisions.
What is Data Warehouse? (2)

- Data warehouses provide on-line analytical processing (OLAP) tools for the interactive analysis of multidimensional data of varied granularities, which facilitate effective data generalization and data mining

- Many other data mining functions, such as association, classification, prediction, and clustering, can be integrated with OLAP operations to enhance interactive mining of knowledge at multiple levels of abstraction
What is Data Warehouse? (3)

• A decision support database that is maintained separately from the organization’s operational database

• “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process [Inm96].” —W. H. Inmon
Data Warehouse Framework

Figure 1.7 Typical framework of a data warehouse for AllElectronics
Data Warehouse is *Subject-Oriented*

- Organized around major *subjects*, such as customer, product, sales, etc.
- 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 is *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.
Data Warehouse is *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
Data Warehouse is *Nonvolatile*

- 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*
# OLTP vs. OLAP

<table>
<thead>
<tr>
<th>Feature</th>
<th>OLTP</th>
<th>OLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristic</td>
<td>operational processing</td>
<td>informational processing</td>
</tr>
<tr>
<td>Orientation</td>
<td>transaction</td>
<td>analysis</td>
</tr>
<tr>
<td>User</td>
<td>clerk, DBA, database professional</td>
<td>knowledge worker (e.g., manager, executive, analyst)</td>
</tr>
<tr>
<td>Function</td>
<td>day-to-day operations</td>
<td>long-term informational requirements, decision support</td>
</tr>
<tr>
<td>DB design</td>
<td>ER based, application-oriented</td>
<td>star/snowflake, subject-oriented</td>
</tr>
<tr>
<td>Data</td>
<td>current; guaranteed up-to-date</td>
<td>historical; accuracy maintained over time</td>
</tr>
<tr>
<td>Summarization</td>
<td>primitive, highly detailed</td>
<td>summarized, consolidated</td>
</tr>
<tr>
<td>View</td>
<td>detailed, flat relational</td>
<td>summarized, multidimensional</td>
</tr>
<tr>
<td>Unit of work</td>
<td>short, simple transaction</td>
<td>complex query</td>
</tr>
<tr>
<td>Access</td>
<td>read/write</td>
<td>mostly read</td>
</tr>
<tr>
<td>Focus</td>
<td>data in</td>
<td>information out</td>
</tr>
<tr>
<td>Operations</td>
<td>index/hash on primary key</td>
<td>lots of scans</td>
</tr>
<tr>
<td>Number of records accessed</td>
<td>tens</td>
<td>millions</td>
</tr>
<tr>
<td>Number of users</td>
<td>thousands</td>
<td>hundreds</td>
</tr>
<tr>
<td>DB size</td>
<td>100 MB to GB</td>
<td>100 GB to TB</td>
</tr>
<tr>
<td>Priority</td>
<td>high performance, high availability</td>
<td>high flexibility, end-user autonomy</td>
</tr>
<tr>
<td>Metric</td>
<td>transaction throughput</td>
<td>query throughput, response time</td>
</tr>
</tbody>
</table>

**Table 3.1** Comparison between OLTP and OLAP
Why Separate is Data Warehouse Needed? (1)

- Why not perform on-line analytical processing directly on operational databases instead of spending additional time and resources to construct a separate data warehouse?
Why Separate is Data Warehouse Needed? (2)

• High performance for both systems
  – DBMS—tuned for OLTP: searching for particular records, indexing, hashing, concurrency control, recovery
  – Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation (summarization and aggregation)
Topics

- Definition of data warehouse
- Multidimensional data model
- Data warehouse architecture
- From data warehousing to data mining
From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a multidimensional data model
- This model views data in the form of a data cube
- A data cube allows data to be modeled and viewed in multiple dimensions
From Tables and Spreadsheets to Data Cubes (1)

• A data cube is defined by facts and dimensions
  – **Facts** are data which data warehouse focus on
    • Fact tables contain numeric measures (such as dollars_sold) and keys to each of the related dimension tables
  – **Dimensions** are perspectives with respect to fact
    • Dimension tables describe the dimension with attributes. For example, **item** (item_name, brand, type), or **time** (day, week, month, quarter, year)
### Customer Table

<table>
<thead>
<tr>
<th>cust_ID</th>
<th>name</th>
<th>address</th>
<th>age</th>
<th>income</th>
<th>credit_info</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Smith, Sandy</td>
<td>1223 Lake Ave., Chicago, IL</td>
<td>31</td>
<td>$78,000</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

### Item Table

<table>
<thead>
<tr>
<th>item_ID</th>
<th>name</th>
<th>brand</th>
<th>category</th>
<th>type</th>
<th>price</th>
<th>place made</th>
<th>supplier</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3</td>
<td>hi-res-TV</td>
<td>Toshiba</td>
<td>high resolution</td>
<td>TV</td>
<td>$988.00</td>
<td>Japan</td>
<td>NikoX</td>
<td>$600.00</td>
</tr>
<tr>
<td>I8</td>
<td>Laptop</td>
<td>Dell</td>
<td>laptop</td>
<td>computer</td>
<td>$1369.00</td>
<td>USA</td>
<td>Dell</td>
<td>$983.00</td>
</tr>
</tbody>
</table>

### Employee Table

<table>
<thead>
<tr>
<th>empl_ID</th>
<th>name</th>
<th>category</th>
<th>group</th>
<th>salary</th>
<th>commission</th>
</tr>
</thead>
<tbody>
<tr>
<td>E55</td>
<td>Jones, Jane</td>
<td>home entertainment</td>
<td>manager</td>
<td>$118,000</td>
<td>2%</td>
</tr>
</tbody>
</table>

### Branch Table

<table>
<thead>
<tr>
<th>branch_ID</th>
<th>name</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>City Square</td>
<td>396 Michigan Ave., Chicago, IL</td>
</tr>
</tbody>
</table>

### Purchases Table

<table>
<thead>
<tr>
<th>trans_ID</th>
<th>cust_ID</th>
<th>empl_ID</th>
<th>date</th>
<th>time</th>
<th>method_paid</th>
<th>amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>C1</td>
<td>E55</td>
<td>03/21/2005</td>
<td>15:45</td>
<td>Visa</td>
<td>$1357.00</td>
</tr>
</tbody>
</table>

### Item_Sold Table

<table>
<thead>
<tr>
<th>trans_ID</th>
<th>item_ID</th>
<th>qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>I3</td>
<td>1</td>
</tr>
<tr>
<td>T100</td>
<td>I8</td>
<td>2</td>
</tr>
</tbody>
</table>

### Works At Table

<table>
<thead>
<tr>
<th>empl_ID</th>
<th>branch_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>E55</td>
<td>B1</td>
</tr>
</tbody>
</table>

---

Figure 1.6: Fragments of a relational database for AllElectronics.
From Tables and Spreadsheets to Data Cubes (2)

Table 3.2 A 2-D view of sales data for AllElectronics according to the dimensions time and item, where the sales are from branches located in the city of Vancouver. The measure displayed is dollar_sold (in thousands).
### Table 3.3

A 3-D view of sales data for AllElectronics according to the dimensions time, item, and location. The measure displayed is `dollar_sold` (in thousands).

<table>
<thead>
<tr>
<th>time</th>
<th>item</th>
<th>location = “Chicago”</th>
<th>location = “New York”</th>
<th>location = “Toronto”</th>
<th>location = “Vancouver”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>854</td>
<td>882</td>
<td>89</td>
<td>623</td>
<td>1087</td>
</tr>
<tr>
<td>Q2</td>
<td>943</td>
<td>890</td>
<td>64</td>
<td>698</td>
<td>1130</td>
</tr>
<tr>
<td>Q3</td>
<td>1032</td>
<td>924</td>
<td>59</td>
<td>789</td>
<td>1034</td>
</tr>
<tr>
<td>Q4</td>
<td>1129</td>
<td>992</td>
<td>63</td>
<td>870</td>
<td>1142</td>
</tr>
</tbody>
</table>
Figure 3.1 A 3-D data cube representation of the data in Table 3.3, according to the dimensions time, item, and location. The measure displayed is dollar_sold (in thousands).
Figure 3.2 A 4-D data cube representation, according to the dimensions time, item, location, and supplier. The measure displayed is dollar_sold (in thousands).
Cuboid

- A data cube is a lattice of cuboids
- The total number of cuboids
- The apex cuboid
- The base cuboid
Figure 3.14 Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains the three dimensions city, item, and year.
The Curse of Dimensionality

- How many cuboids are there in a n-dimensional data cube?
- How many cuboids are there in a n-dimensional data cube and each dimension (i) has the number of level, \((L_i)\)?
Conceptual Modeling of Data Warehouses

• Modeling data warehouses: dimensions & measures
  – **Star schema**: A fact table in the middle connected to a set of dimension tables
  – **Snowflake schema**: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  – **Fact constellations**: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation
**Figure 3.4** Star schema of a data warehouse for sales.
Figure 3.4 Snowflake schema of a data warehouse for sales.
**Fact Constellation**

**Sales Fact Table**
- time_key
- item_key
- branch_key
- location_key
- dollars_sold
- unit_sold

**Item**
- item_key
- item_name
- brand
- type
- supplier_type

**Location**
- location_key
- street
- city
- province_or_street
- country

**Shipper**
- shipper_key
- shipper_name
- location_key
- shipper_type

**Shipping Fact Table**
- item_key
- time_key
- shipper_key
- from_location
- to_location
- dollars_sold
- unit_shipped

**Figure 3.5** Fact constellation schema of a data warehouse for sales and shipping.
Exercise

• Exercise 3.5 (a) – page 153