CSE5334 DATA MINING
Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
  - A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
What is Data Warehouse?

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

Data warehousing: The process of constructing and using data warehouses
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.
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—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
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 isolated</td>
<td>historical, summarized, multidimensional 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>
Why Separate Data Warehouse?

- Different functions and different data:

- Note: There are more and more systems which perform OLAP analysis directly on relational databases

- There is no absolute boundary.
Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
Data Cube

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

- A data cube contains **aggregates of measure values**, on various combinations of dimensions, and furthermore, with various levels of aggregation on individual dimension.

- In data warehousing literature, an n-D base cube is called a **base cuboid**. The top most 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**. The lattice of cuboids forms a data cube.
A 3-D Cuboid

- Sales volume as a function of product, month, and region

Dimensions: Product, Location, Time

Hierarchical summarization paths:
- Industry
- Region
- Year
- Category
- Country
- Quarter
- Item
- City
- Month
- Office
- Day
- Week

Lecture 3: Data Warehousing, OLAP
An Example of Data Cube

Total annual sales of TV in U.S.A.

Product (category)

TV
PC
VCR

sum

Time (Quarter)

1Qtr
2Qtr
3Qtr
4Qtr
sum

Location (country)

U.S.A
Canada
Mexico
sum

All, All, All
Data Cube: A Lattice of Cuboids

- 0-D (apex) cuboid
- 1-D cuboids
- 2-D cuboids
- 3-D (base) cuboid
Cuboids

Lecture 1: Introduction

Total annual sales of TV in U.S.A.
Another 4-D Data Cube

- 0-D (apex) cuboid
- 1-D cuboids
- 2-D cuboids
- 3-D cuboids
- 4-D (base) cuboid
A Concept Hierarchy on Location Dimension

- all
- region
  - Europe
    - Germany
      - Frankfurt
    - Spain
      - Vancouver
    - Canada
      - Toronto
  - North_America
    - Mexico
  - ...
Concept Hierarchy in Data Cube
Conceptual Schema Design

- **Dimensions & Measures**
  - **Dimension tables**, such as product (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
Conceptual Modeling of Data Warehouses

- **Star schema**: A fact table in the middle connected to a set of dimension tables
Example of Star Schema

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

Measures

Time
- time_key
day
day_of_the_week
month
quarter
year

Location
- location_key
street
city
state_or_province
country

Branch
- branch_key
branch_name
branch_type

Item
- item_key
item_name
brand
type
supplier_type
Conceptual Modeling of Data Warehouses

- **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.

  - It provides explicit support of hierarchy
    - Easier to manage the dimension
    - Can be less efficient (due to join) than star schema
Example of Snowflake Schema

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

- **branch**
  - branch_key
  - branch_name
  - branch_type

- **Measures**
  - units_sold
  - dollars_sold
  - avg_sales

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

- **item**
  - item_key
  - item_name
  - brand
  - type
  - supplier_key

- **supplier**
  - supplier_key
  - supplier_type

- **location**
  - location_key
  - street
  - city_key

- **city**
  - city_key
  - city
  - state_or_province
  - country
Conceptual Modeling of Data Warehouses

- **Fact constellations**: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation
Example of Fact Constellation

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_type

Time:
- time_key
day
day_of_the_week
month
quarter
year

Branch:
- branch_key
- branch_name
- branch_type

Location:
- location_key
- street
- city
- province_or_state
- country

Measures:
- time_key
- item_key
- branch_key
- location_key
- units_sold
- dollars_sold
- avg_sales

Shipping Fact Table:
- time_key
- item_key
- shipper_key
- from_location
- to_location
- dollars_cost
- units_shipped

Shipper:
- shipper_key
- shipper_name
- location_key
- shipper_type

Lecture 3: Data Warehousing, OLAP
Measures of Data Cube: Three Categories

- **Distributive**: if the result derived by applying the function to \( n \) aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., \( \text{count}(), \text{sum}(), \text{min}(), \text{max}() \)

- **Algebraic**: if it can be computed by an algebraic function with \( M \) arguments (where \( M \) is a bounded integer), each of which is obtained by applying a distributive aggregate function
  - E.g., \( \text{avg}(), \text{min}_N(), \text{standard_deviation}() \)

- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., \( \text{median}(), \text{mode}(), \text{rank}() \)
Typical OLAP 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

- **Pivot (rotate):**
  - reorient the cube, visualization, 3D to series of 2D planes
Roll up and Drill Down

- Roll up: increasing the level of aggregation
  - further aggregating along one more dimension
  - or further aggregating along the hierarchy of one dimension

- Drill down: decreasing the level of aggregating

It is like traversing in the lattice of cuboids.
Fig. 3.10 Typical OLAP Operations

- **Dice for** (location = Toronto or Vancouver) and (time = Q1 or Q2) and (item = Home entertainment or Computer)
- **Slice for** time = Q2
- **Pivot**
- **Roll-up on location** (from cities to countries)
- **Drill-down on time** (from quarters to months)
Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
OLAP Server Architectures

- **Relational OLAP (ROLAP)**
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - Greater scalability

- **Multidimensional OLAP (MOLAP)**
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data

- **Hybrid OLAP (HOLAP)** (e.g., Microsoft SQLServer)
  - Flexibility, e.g., low level: relational, high-level: array

- **Specialized SQL servers** (e.g., Redbricks)
  - Specialized support for SQL queries over star/snowflake schemas
Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L levels?

\[
T = \prod_{i=1}^{n} (L_i + 1)
\]

- Materialization of data cube
  - Materialize every (cuboid) (full materialization), none (no materialization), or some (partial materialization)
  - Selection of which cuboids to materialize
    - Based on size, sharing, access frequency, etc.
### Indexing OLAP Data: Bitmap Index

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The $i$-th bit is set if the $i$-th row of the base table has the value for the indexed column
- not suitable for high cardinality domains

#### Base table

<table>
<thead>
<tr>
<th>Cust</th>
<th>Region</th>
<th>Type</th>
<th>RecID</th>
<th>Asia</th>
<th>Europe</th>
<th>America</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Asia</td>
<td>Retail</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C2</td>
<td>Europe</td>
<td>Dealer</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C3</td>
<td>Asia</td>
<td>Dealer</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C4</td>
<td>America</td>
<td>Retail</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C5</td>
<td>Europe</td>
<td>Dealer</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Index on Region

<table>
<thead>
<tr>
<th>RecID</th>
<th>Asia</th>
<th>Europe</th>
<th>America</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Index on Type

<table>
<thead>
<tr>
<th>RecID</th>
<th>Retail</th>
<th>Dealer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Indexing OLAP Data: Join Indices

- Join index: JI(R-id, S-id) where R (R-id, ...) ▷◁ S (S-id, ...)
- Traditional indices map the values to a list of record ids
  - It materializes relational join in JI file and speeds up relational join
- In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
  - E.g. fact table: Sales and two dimensions city and product
    - A join index on city maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  - Join indices can span multiple dimensions
Efficient Processing OLAP Queries

- Determine which operations should be performed on the available cuboids
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection

- Determine which materialized cuboid(s) should be selected for OLAP op.
  - Let the query to be processed be on \{brand, province_or_state\} with the condition “year = 2004”, and there are 4 materialized cuboids available:
    1) \{year, item_name, city\}
    2) \{year, brand, country\}
    3) \{year, brand, province_or_state\}
    4) \{item_name, province_or_state\}  where year = 2004

Which should be selected to process the query?

- Explore indexing structures and compressed vs. dense array structures in MOLAP
Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Summary
Summary: Data Warehouse and OLAP Technology

- Why data warehousing?
- A **multi-dimensional model** of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures
- **OLAP** operations: drilling, rolling, slicing, dicing and pivoting
- Data warehouse architecture
  - OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Indexing OALP data: Bitmap index and join index
  - OLAP query processing