Data Warehousing and OLAP Technology

Chapter 3
- The Course

DS = Data source
DW = Data warehouse
DM = Data Mining
SD = Staging Database

OLAP
DM

Association
Classification
Clustering
Chapter Outline

- What is a data warehouse?
- How to construct a Data Warehouse
  - What is the Data Model used in data warehouse?
  - 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.
--- Data Warehouse - Integrated

Data Warehouse

Integration System

World Wide Web

Other Sources

Other Databases

Own Databases
-- 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—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
Traditional heterogeneous DB integration: A query driven approach

- Build wrappers/mediators on top of heterogeneous databases
- When a query is posed to a client site, a meta-dictionary is used to translate the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
- Complex information filtering, compete for resources

Data warehouse: update-driven, high performance

- Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis
Query-Driven Approach

Client

Wrapper

Mediator

Wrapper

Source

Client

Wrapper

Source

Source

Wrapper
The Warehousing Approach

Client

Query & Analysis

Metadata

Warehouse

Integration

Source

Source

Source
-- 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?

- High performance for both systems
  - DBMS—tuned for OLTP: access methods, indexing, concurrency control, recovery
  - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation

- Different functions and different data:
  - **missing data**: Decision support requires historical data which operational DBs do not typically maintain
  - **data consolidation**: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - **data quality**: different sources typically use inconsistent data representations, codes and formats which have to be reconciled
Chapter Outline

- What is a data warehouse?
- How to construct a Data Warehouse
  - What is the Data Model used in data warehouse?
  - Data warehouse architecture
  - Data warehouse implementation
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

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.
Cube: A Lattice of Cuboids

0-D (apex) cuboid
1-D cuboids
2-D cuboids
3-D cuboids
4-D (base) cuboid
-- 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
--- Example of Star Schema

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

### Location
- location_key
- street
- city
- state_or_province
- country

### Item
- item_key
- item_name
- brand
- type
- supplier_type

### Branch
- branch_key
- branch_name
- branch_type

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

### Measures
-- Example of Snowflake Schema

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

**Branch**
- branch_key
- branch_name
- branch_type

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

**Measures**

**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
--- Example of Fact Constellation

**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
- units_sold
- riyals_sold

**Measures**
- item_key
- item_name
- brand
- type
- supplier_type

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

**Shipping Fact Table**
- time_key
- item_key
- shipper_key
- from_location
- to_location
- riyals_cost
- units_shipped
- shipper
- shipper_key
- shipper_name
- location_key
- location_key
-- A Concept Hierarchy: Dimension (location)

- All
  - Region
    - Country
      - City
        - Office
  - Europe
    - Germany
      - Frankfurt
      - L. Chan
  - ... Spain
  - North America
    - Canada
      - Vancouver
      - M. Wind
    - Mexico
      - Toronto
-- Multidimensional Data

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

Dimensions: Product, Location, Time
Hierarchical summarization paths
--- A Sample Data Cube

Total annual sales of TV in U.S.A.
Cuboids Corresponding to the Cube

0-D (apex) cuboid

1-D cuboids

2-D cuboids

3-D (base) cuboid
--- Browsing a Data Cube

- Visualization
- OLAP capabilities
- Interactive manipulation
-- 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
What is a data warehouse?

A multi-dimensional data model

Data warehouse architecture

Data warehouse implementation
Data Warehouse: A Multi-Tiered Architecture

Data Sources

Other sources
Operational DBs

Data Storage

Metadata
Monitor & Integrator

Data Warehouse

Extract Transform Load Refresh

OLAP Server

Analysis Query Reports Data mining

OLAP Engine

Front-End Tools

OLAP Server

Data Marts

Analysis Query Reports Data mining

OLAP Engine

Front-End Tools
-- DW Design Process

- Top-down, bottom-up approaches or a combination of both
  - **Top-down**: Starts with overall design and planning (mature)
  - **Bottom-up**: Starts with experiments and prototypes (rapid)

- Typical data warehouse design process
  - Choose a *business process* to model, e.g., orders, invoices, etc.
  - Choose the *grain* (*atomic level of data*) of the business process
  - Choose the *dimensions* that will apply to each fact table record
  - Choose the *measure* that will populate each fact table record
-- Three DW Models

- **Enterprise warehouse**
  - collects all of the information about subjects spanning the entire organization

- **Data Mart**
  - a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart

- **Virtual warehouse**
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized
Define a high-level corporate data model

- Distributed Data Marts
  - Data Mart
  - Data Mart
  - Model refinement
  - Model refinement

- Multi-Tier Data Warehouse
- Enterprise Data Warehouse

-- DW Development: A Recommended Approach
-- Data Warehouse Back-End Tools and Utilities

- Data extraction
  - get data from multiple, heterogeneous, and external sources

- Data cleaning
  - detect errors in the data and rectify them when possible

- Data transformation
  - convert data from legacy or host format to warehouse format

- Load
  - sort, summarize, consolidate, compute views, check integrity, and build indices and partitions

- Refresh
  - propagate the updates from the data sources to the warehouse
-- Metadata Repository …

- Meta data is the data defining warehouse objects. It stores:

- Description of the structure of the data warehouse
  - schema, view, dimensions, hierarchies, derived data definition, data mart locations and contents

- Operational meta-data
  - data lineage (history of migrated data and transformation path),
  - currency of data (active, archived, or purged),
  - monitoring information (warehouse usage statistics, error reports, audit trails)

- The algorithms used for summarization
  - Measure and dimension definition algorithms
  - Data granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports
The mapping from operational environment to the data warehouse
- Source databases and their contents,
- Gateway descriptions, data partitions, data extraction, cleaning, transformation rules, and defaults, data refresh and purge rules
- Security

Data related to system performance
- Indices, profiles
- Timing and scheduling of refresh

Business data
- Business terms and definitions,
- Ownership of data
- Charging policies
-- 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
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

In an n-dimensional cube there are:

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

Cuboids where \( L_i \) is the levels in dimension \( i \)

So the questions is how many cuboids can be materialized

- Materialize every (cuboid) (full materialization)
- some (partial materialization) or
- none (no materialization)
-- Cube Operation

- Cube definition and computation in DMQL

```dmql
define cube sales[item,city,year]:
sum(sales_in_dollars)
compute cube sales
```

- Transform it into a SQL-like language (with a new operator `cube by`)

```sql
SELECT item, city, year, SUM(amount)
FROM SALES
CUBE BY item, city, year
```

- Need compute the following Group-Bys

```
(date, product, customer),
(date, product), (date, customer),
(product, customer), (date), (product),
(customer), ()
```

```graph
A Cube diagram showing combinations of city, item, year.
```
-- 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>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Asia</td>
<td>Retail</td>
</tr>
<tr>
<td>C2</td>
<td>Europe</td>
<td>Dealer</td>
</tr>
<tr>
<td>C3</td>
<td>Asia</td>
<td>Dealer</td>
</tr>
<tr>
<td>C4</td>
<td>America</td>
<td>Retail</td>
</tr>
<tr>
<td>C5</td>
<td>Europe</td>
<td>Dealer</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>
Join index: \( JI(R-id, S-id) \) where \( R (R-id, \ldots) \bowtie \bowtie S (S-id, \ldots) \)

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 structs in MOLAP
End