Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- Data Warehouse Implementation
- Summary
What is a Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization’s operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.
- “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—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**
## 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 index/hash on prim. key</td>
<td>lots of scans</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 a 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
- Note: There are more and more systems which perform OLAP analysis directly on relational databases
Data Warehouse: A Multi-Tiered Architecture

Data Sources

- Other sources
- Operational DBs

Data Storage

- Data Warehouse
- Metadata
- Extract Transform Load Refresh

OLAP Engine

- OLAP Server
- Serve

Front-End Tools

- Analysis Query Reports
- Data mining

Data Marts
Three Data Warehouse 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 group 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
Extraction, Transformation, and Loading (ETL)

- **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 defn, 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
  - The **mapping** from operational environment to the data warehouse
  - Data related to **system performance**
    - warehouse schema, view and derived data definitions
  - **Business data**
    - business terms and definitions, ownership of data, charging policies
Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- Data Warehouse Implementation
- Summary
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

**Example of Star Schema**

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

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

- **item**
  - item_key
  - item_name
  - brand
  - type
  - supplier_type

- **location**
  - location_key
  - street
  - city
  - state_or_province
  - country

**Sales Fact Table**

- **time_key**
- **item_key**
- **branch_key**
- **location_key**
- **units_sold**
- **dollars_sold**
- **avg_sales**

**Measures**
Example of Snowflake 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

branch
- branch_key
  - branch_key
  - branch_name
  - branch_type

item
- item_key
  - item_name
  - brand
type
  - supplier_key

supplier
- supplier_key
  - supplier_type

location
- location_key
  - location
  - street
  - city_key

city
- city_key
city
state_or_province
country
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**

**Location**
- **location_key**
- **street**
- **city**
- **province_or_state**
- **country**

**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**
A Concept Hierarchy: **Dimension** (location)

- **Region**
  - Europe
    - Germany
      - Frankfurt
        - L. Chan
    - Spain
    - Canada
      - Vancouver
        - Toronto
      - Mexico
  - North_America

- **Office**
  - all
  - Toronto
  - Frankfurt
  - Vancouver
  - Spain
  - Germany
  - North_America
Data Cube Measures: 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., `count()`, `sum()`, `min()`, `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., `avg()`, `min_N()`, `standard_deviation()`
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., `median()`, `mode()`, `rank()`
View of Warehouses and Hierarchies

Specification of hierarchies

- Schema hierarchy
  
  day < {month < quarter; week} < year

- Set_grouping hierarchy
  
  {1..10} < inexpensive
Multidimensional Data

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

Dimensions: *Product, Location, Time*

Hierarchical summarization paths:

- Industry
- Region
- Year
- Category
- Country
- Quarter
- Product
- City
- Month
- Office
- Week
- Day
A Sample Data Cube

- **Product**: TV, PC, VCR
- **Date**: 1Qtr, 2Qtr, 3Qtr, 4Qtr, sum
- **Country**: U.S.A, Canada, Mexico, sum

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

- **0–D** (apex) cuboid
- **1–D** cuboids
- **2–D** cuboids
- **3–D** (base) cuboid
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
- Other operations
  - drill across: involving (across) more than one fact table
  - drill through: through the bottom level of the cube to its back-end relational tables (using SQL)
Fig. 3.10 Typical OLAP Operations
A Star-Net Query Model

Each circle is called a footprint
Browsing a Data Cube

- Visualization
- OLAP capabilities
- Interactive manipulation
Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- Data Warehouse Implementation
- Summary
Four views regarding the design of a data warehouse

- **Top-down view**
  - allows selection of the relevant information necessary for the data warehouse

- **Data source view**
  - exposes the information being captured, stored, and managed by operational systems

- **Data warehouse view**
  - consists of fact tables and dimension tables

- **Business query view**
  - sees the perspectives of data in the warehouse from the view of end-user
Data Warehouse 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)

- **From software engineering point of view**
  - **Waterfall**: structured and systematic analysis at each step before proceeding to the next
  - **Spiral**: rapid generation of increasingly functional systems, short turn around time, quick turn around

- **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
Data Warehouse Development: A Recommended Approach

Define a high-level corporate data model

Multi-Tier Data Warehouse

Enterprise Data Warehouse

Data Mart

Distributed Data Marts

Model refinement

Model refinement
Data Warehouse Usage

- Three kinds of data warehouse applications
  - Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - Analytical processing
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - Data mining
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools
From On-Line Analytical Processing (OLAP) to On Line Analytical Mining (OLAM)

Why online analytical mining?
- High quality of data in data warehouses
  - DW contains integrated, consistent, cleaned data
- Available information processing structure surrounding data warehouses
  - ODBC, OLEDB, Web accessing, service facilities, reporting and OLAP tools
- OLAP-based exploratory data analysis
  - Mining with drilling, dicing, pivoting, etc.
- On-line selection of data mining functions
  - Integration and swapping of multiple mining functions, algorithms, and tasks
Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- Data Warehouse Implementation
- Summary
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.
The “Compute Cube” Operator

- Cube definition and computation in 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`, introduced by Gray et al.’96)

  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),
  ()
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
- A recent bit compression technique, Word-Aligned Hybrid (WAH), makes it work for high cardinality domain as well [Wu, et al. TODS’06]

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

<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, ...) \(\bowtie\bowtie\) 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 \( \text{"year} = 2004\" \), and there are 4 materialized cuboids available:
    1) \( \{\text{year}, \text{item}\_\text{name}, \text{city}\} \)
    2) \( \{\text{year}, \text{brand}, \text{country}\} \)
    3) \( \{\text{year}, \text{brand}, \text{province}\_\text{or}\_\text{state}\} \)
    4) \( \{\text{item}\_\text{name}, \text{province}\_\text{or}\_\text{state}\} \) where \( \text{year} = 2004 \)

  Which should be selected to process the query?

- Explore indexing structures and compressed vs. dense array structs in MOLAP
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 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- Data Warehouse Implementation
- Summary
Summary

- **Data warehousing**: A multi-dimensional model of a data warehouse
  - A data cube consists of *dimensions* & *measures*
  - Star schema, snowflake schema, fact constellations
  - OLAP operations: drilling, rolling, slicing, dicing and pivoting

- **Data Warehouse Architecture, Design, and Usage**
  - Multi-tiered architecture
  - Business analysis design framework
  - Information processing, analytical processing, data mining, **OLAM** (Online Analytical Mining)

- **Implementation**: Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Indexing OALP data: Bitmap index and join index
  - OLAP query processing
  - OLAP servers: ROLAP, MOLAP, HOLAP