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
- Ex. Hotel price: differences on currency, tax, breakfast covered, and parking
- When data is moved to the warehouse, it is converted

**Data Warehouse—Nonvolatile**

- Independence
- A physically separate store of data transformed from the operational environment
- Static: 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—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”

**OLTP vs. OLAP**

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

- Top Tier: Front-End Tools
- Middle Tier: OLAP Server
- Bottom Tier: Data Warehouse Server
- Data

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 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

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
**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

**From Tables and Spreadsheets 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.
- Data cube: A lattice of cuboids.
  - 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.

**Data 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.
Star Schema: An Example

- **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

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

- **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

- **Supplier**
  - supplier_key
  - supplier_name
  - location_key
  - city_key
  - state_or_province
  - country

- **Sales Fact Table**
  - item_key
  - brand
  - type
  - supplier_key

- A Concept Hierarchy for a Dimension (location)

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

Snowflake Schema: An Example

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

- **Location**
  - location_key
  - street
  - city_key
  - city
  - state_or_province
  - country

- **Branch**
  - branch_key
  - branch_name
  - branch_type

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

- **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

- **Supplier**
  - supplier_key
  - supplier_name
  - location_key
  - city_key
  - state_or_province
  - country

- **Sales Fact Table**
  - item_key
  - brand
  - type
  - supplier_key

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

- **Shipper**
  - shipper_key
  - shipper_name
  - location_key
  - city_key
  - city

Fact Constellation: An Example

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

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

- **Branch**
  - branch_key
  - branch_name
  - branch_type

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

- **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

- **Supplier**
  - supplier_key
  - supplier_name
  - location_key
  - city_key
  - state_or_province
  - country

- **Sales Fact Table**
  - item_key
  - brand
  - type
  - supplier_key

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

- **Shipper**
  - shipper_key
  - shipper_name
  - location_key
  - city_key
  - city
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
  - \( \text{avg}(x) = \frac{\text{sum}(x)}{\text{count}(x)} \)
  - Is \( \text{min}_N() \) an algebraic measure? How about \( \text{standard deviation}() \)?
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., median(), mode(), rank()

Multidimensional Data

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

A Sample Data Cube

Cuboids Corresponding to the Cube
**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)

---

**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

---

**Design of Data Warehouse: A Business Analysis Framework**

- 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 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
  - Full materialization: Materialize every (cuboid)
  - No materialization: Materialize none (cuboid)
  - Partial materialization: Materialize some cuboids
  - Which cuboids to materialize?
    - Selection 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

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]

Indexing OLAP Data: Join Indices

- Join index: JI(R-id, S-id) where R (R-id, …) \( \bowtie \) 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 \(year = 2004\), 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)\} \text{ where } year = 2004
  - Which should be selected to process the query?
- Explore indexing structures and compressed vs. dense array structs in MOLAP

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

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

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
- 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
References (I)

- S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. VLDB’96
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. SIGMOD’97
- J. Han. Towards on-line analytical mining in large databases. SIGMOD Record, 1998
- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. SIGMOD’96

References (II)

- P. O’Neil and D. Quass. Improved query performance with variant indexes. SIGMOD’97
- S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. ICDE’94
- J. Widom. Research problems in data warehousing. CIKM’95.