Data Warehousing & OLAP

Data Mining: Concepts and Techniques
— Chapter 3 —
Jiawei Han

and

An Introduction to Database Systems
C.J.Date,
What is Data Warehousing?

- What is OLAP?
- What is a Data Cube, what is a Cuboid?
- What is ROLAP, MOLAP, HOLAP

What is 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
Data Warehouse vs. Heterogeneous DBMS

- 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

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</td>
<td>index/hash on prim. key</td>
</tr>
<tr>
<td></td>
<td></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 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

What is OLAP?

- The term OLAP ("online analytical processing") was coined in a white paper written for Arbor Software Corp. in 1993

  - Interactive process of creating, managing, analyzing and reporting on data
  - Analyzing large quantities of data in real-time
OLAP

- Data is perceived and manipulated as though it were stored in a "multi-dimensional array"

- Ideas are explained in terms of conventional SQL-styled tables

Data aggregation

- Data aggregation (agregação) in many different ways

- The number of possible groupings quickly becomes large
  - The user has to consider all groupings
  - Analytical processing problem
Queries for supplier-and-parts database

1) Get the total shipment quantity
2) Get total shipment quantities by supplier
3) Get total shipment quantities by part
4) Get the shipment by supplier and part

<table>
<thead>
<tr>
<th>S#</th>
<th>P#</th>
<th>QTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>P1</td>
<td>300</td>
</tr>
<tr>
<td>S1</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S2</td>
<td>P1</td>
<td>300</td>
</tr>
<tr>
<td>S2</td>
<td>P2</td>
<td>400</td>
</tr>
<tr>
<td>S3</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S4</td>
<td>P2</td>
<td>200</td>
</tr>
</tbody>
</table>
1. SELECT SUM(QTY) AS TOTQTY
   FROM SP
   GROUP BY () ;

<table>
<thead>
<tr>
<th>TOTQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1600</td>
</tr>
</tbody>
</table>

2. SELECT S#, SUM(QTY) AS TOTQTY
   FROM SP
   GROUP BY (S#) ;

<table>
<thead>
<tr>
<th>S#</th>
<th>TOTQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>500</td>
</tr>
<tr>
<td>S2</td>
<td>700</td>
</tr>
<tr>
<td>S3</td>
<td>200</td>
</tr>
<tr>
<td>S4</td>
<td>200</td>
</tr>
</tbody>
</table>
3. SELECT P#, SUM(QTY) AS TOTQTY
FROM SP
GROUP BY (P#);

<table>
<thead>
<tr>
<th>P#</th>
<th>TOTQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>600</td>
</tr>
<tr>
<td>P2</td>
<td>1000</td>
</tr>
</tbody>
</table>

4. SELECT S#, P#, SUM(QTY) AS TOTQTY
FROM SP
GROUP BY (S#,P#),

<table>
<thead>
<tr>
<th>S#</th>
<th>P#</th>
<th>TOTQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>P1</td>
<td>300</td>
</tr>
<tr>
<td>S1</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S2</td>
<td>P1</td>
<td>300</td>
</tr>
<tr>
<td>S2</td>
<td>P2</td>
<td>400</td>
</tr>
<tr>
<td>S3</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S4</td>
<td>P2</td>
<td>200</td>
</tr>
</tbody>
</table>
Drawbacks

- Formulation so many similar but distinct queries is tedious
- Executing the queries is expensive
- Make life easier,
  - more efficient computation
- Single query
  - GROUPING SETS, ROLLUP, CUBE options
  - Added to SQL standard 1999

GROUPING SETS

- Execute several queries simultaneously

SELECT S#, P#, SUM (QTY) AS TOTQTY FROM SP GROUP BY GROUPING SETS ( (S#), (P#) ) ;

<table>
<thead>
<tr>
<th>S#</th>
<th>P#</th>
<th>TOTQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>null</td>
<td>500</td>
</tr>
<tr>
<td>S2</td>
<td>null</td>
<td>700</td>
</tr>
<tr>
<td>S3</td>
<td>null</td>
<td>200</td>
</tr>
<tr>
<td>S4</td>
<td>null</td>
<td>200</td>
</tr>
<tr>
<td>null</td>
<td>P1</td>
<td>600</td>
</tr>
<tr>
<td>null</td>
<td>P2</td>
<td>1000</td>
</tr>
</tbody>
</table>

Single results table
Not a relation !!
null ➔ missing information
SELECT CASE GROUPING (S#)
    WHEN 1 THEN '??'
    ELSE S#
AS S#,
CASE GROUPING (P#)
    WHEN 1 THEN '!!'
    ELSE P#
AS P#,
SUM (QTY) AS TOTQTY
FROM SP
GROUP BY GROUPING SETS ((S#), (P#));

ROLLUP

SELECT S#,P#, SUM (QTY) AS TOTQTY
FROM SP
GROUP BY ROLLUP (S#, P#);

GROUP BY GROUPING SETS ((S#, P#), (S#), ());
ROLLUP

- The quantities have been „roll up“ (estender) for each supplier
- Rolled up „along supplier dimension“

GROUP BY ROLLUP (A,B,...,Z)

⇒

(A,B,...,Z)
(A,B,...)
(A,B)
(A)
()

GROUP BY ROLLUP (A,B) is not symmetric in A and B!

CUBE

```
SELECT S#, P#, SUM ( QTY ) AS TOTQTY
FROM SP
GROUP BY CUBE ( S#, P# ) ;
```

<table>
<thead>
<tr>
<th>S#</th>
<th>P#</th>
<th>TOTQTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>P1</td>
<td>300</td>
</tr>
<tr>
<td>S1</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S2</td>
<td>P1</td>
<td>300</td>
</tr>
<tr>
<td>S2</td>
<td>P2</td>
<td>400</td>
</tr>
<tr>
<td>S3</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S4</td>
<td>P2</td>
<td>200</td>
</tr>
<tr>
<td>S1</td>
<td>null</td>
<td>500</td>
</tr>
<tr>
<td>S2</td>
<td>null</td>
<td>700</td>
</tr>
<tr>
<td>S3</td>
<td>null</td>
<td>200</td>
</tr>
<tr>
<td>S4</td>
<td>null</td>
<td>200</td>
</tr>
<tr>
<td>null</td>
<td>P1</td>
<td>600</td>
</tr>
<tr>
<td>null</td>
<td>P1</td>
<td>1000</td>
</tr>
<tr>
<td>null</td>
<td>null</td>
<td>1600</td>
</tr>
</tbody>
</table>

GROUP BY GROUPING SETS ( (S#, P#), ( S# ), ( P# ), () )
CUBE

Confusing term CUBE (?)
- Derived from the fact that in multidimensional terminology, data values are stored in cells of a multidimensional array or a hypercube
  - The actual physical storage may differ
- In our example
  - cube has just two dimensions (supplier, part)
  - The two dimensions are unequal (no square rectangle...)

Means „group“ by all possible subsets of the set \{A, B, ..., Z\}

Means „group“ by all possible subsets of the set \{A, B, ..., Z\}
- \(M=\{A, B, ..., Z\}\), \(|M|=N\)

- Power Set (Algebra)
- \(P(M)=\{N | N\subseteq M\}\), \(|P(M)|=2^N\)
  - proof by induction

- Subset represent different grade of summarization
- Data Mining: such a subset is called a Cuboid
Cross Tabulations

- Display query results as cross tabulations
  - More readable way
  - Formatted as a simple array
  - Example: two dimensions (supplier and parts)

```
   | P1  | P2  | Total |
---|-----|-----|-------|
S1 | 300 | 200 | 500   |
S2 | 300 | 400 | 700   |
S3 |  0  | 200 | 200   |
S4 |  0  | 200 | 200   |
    | 600 |1000 |1600   |
```

What is 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)
  - time(day, week, month, quarter, year) ...hierarchy
- Fact table contains measures (numerical values, such as dollars_sold) and keys to each of the related dimension tables
2-D view of sales data according to dimensions time and item

<table>
<thead>
<tr>
<th>time (quarter)</th>
<th>location = “Vancouver”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>item (type)</td>
</tr>
<tr>
<td></td>
<td>home</td>
</tr>
<tr>
<td>entertainment</td>
<td>605</td>
</tr>
<tr>
<td></td>
<td>680</td>
</tr>
<tr>
<td></td>
<td>812</td>
</tr>
<tr>
<td></td>
<td>927</td>
</tr>
</tbody>
</table>

3-D view of sales data according to dimensions time, item, location

<table>
<thead>
<tr>
<th>time</th>
<th>location = “Chicago”</th>
<th>location = “New York”</th>
<th>location = “Toronto”</th>
<th>location = “Vancouver”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>item</td>
<td>item</td>
<td>item</td>
<td>item</td>
</tr>
<tr>
<td></td>
<td>home comp. phone sec. ent.</td>
<td>home comp. phone sec. ent.</td>
<td>home comp. phone sec. ent.</td>
<td>home comp. phone sec. ent.</td>
</tr>
<tr>
<td>Q1</td>
<td>854</td>
<td>882</td>
<td>80</td>
<td>823</td>
</tr>
<tr>
<td>Q2</td>
<td>948</td>
<td>800</td>
<td>64</td>
<td>689</td>
</tr>
<tr>
<td>Q3</td>
<td>1022</td>
<td>924</td>
<td>50</td>
<td>789</td>
</tr>
<tr>
<td>Q4</td>
<td>1120</td>
<td>902</td>
<td>65</td>
<td>470</td>
</tr>
</tbody>
</table>
3-D data cube representation

4-D view of sales data according to dimensions time, item, location, supplier
Cuboid

- Names in data warehousing literature:
  - The n-D cuboid, which holds the lowest level of summarization, is called a base cuboid .. \{\{A\},\{B\},\ldots\}
  - The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid .. \{\emptyset\}
  - The lattice of cuboids forms a data cube

Cube: A Lattice of Cuboids ....(Power Set)

\[ 2^4 = 16 \]
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures instead of relational model
- Subject, facilitates on-line data analysis oriented
- Most popular model is the multidimensional model
- Most common modeling paradigm:
  - Star schema
  - Data warehouse contains a large central table (fact table)
    - Contains the data without redundancy
  - A set of dimension tables (each for each dimension)

Example of Star Schema
Snowflake schema

- **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.

**Example of Snowflake 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_key
- **supplier**
  - supplier_key
  - supplier_type
- **location**
  - location_key
  - street
  - city_key
  - city
  - state_or_province
  - country
- **Measures**
  - units_sold
  - dollars_sold
  - avg_sales
- **Sales Fact Table**
Fact constellations

- 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
Measures of Data Cube: Three Categories
(Depending on the aggregate functions)

- **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()

Hierarchies

- Independent variables are often related in hierarchies (taxonomy)
  - Determine ways in which dependent data can be aggregated

- Temporal hierarchy
  - Seconds, minutes, hours, days, weeks, months, years

- Same data can be aggregated in many different ways
  - Same independent variable can belong to different hierarchies
**Hierarchy - Location**

```
all          all
region       region
            Europe ... North America
country     Germany ... Spain ... Canada ... Mexico
            city ...
            office ...
            
            Frankfurt ...
            "..."
            
            Vancouver ...
            "..."
            
            Toronto ...
            "..."
            
            L. Chan ...
            M. Wind ...
```

**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

- Drill up and down

  - Drill up:
    - going from a lower level of aggregation to a higher
  - Drill down:
    - means the opposite

- Possible difference between drill up and roll up
  - Roll up: creating the desired groupings or aggregations
  - Drill up: accessing the aggregations

- Example for drill down:
  - Given the total shipment quantity, get the total quantities for each individual supplier
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)
Multi-dimensional query Language

- No standard yet..
- DMQL, DMX, ...
  - MDX was introduced by Microsoft with Microsoft SQL Server OLAP Services in around 1998, as the language component of the OLE DB for OLAP API. More recently, MDX has appeared as part of the XML for Analysis API. Microsoft proposed that the MDX is a standard, and its adoption among application writers and other OLAP providers is steadily increasing.
Data Warehouse: A Multi-Tiered Architecture

OLAP Server Architectures

- **Relational OLAP (ROLAP)**
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middleware
  - 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
MOLAP

- Multidimensional database (server)
  - Data is stored in cells of a multi-dimensional array
  - Three dimensions: products, customers, time intervals
  - Each individual cell value might then represent the total quantity of the indicated product sold to the indicated customer in the indicated time interval

- Variable dependent or independent
  - Independent: products, customers, time intervals
  - Dependent: quantity

Independent, dependent

- Independent variables form the dimension of the array by which the data is organized
  - addressing scheme of the array
  - Also named: dimensional, location

- Dependent variable values stored in the cells of the array
  - Also named: nondimensional, content
Problems

- Often we do not know which variables are independent and which dependent
- Chosen based on hypothesis, and then tested
- A lot of iteration of trial and error
- Pivoting:
  - Swapping between dimensional and nondimensional variables
  - Array transpose, dimensional reordering, add dimensions

MOLAP

- Array cells often empty
  - The more dimensions, there more empty cells
  - Empty cell $\rightarrow$ Missing information
  - How to treat not present information?
  - How does the system support
    - Information is unknown
    - Has been not captured
    - Not applicable
    - ....
- Arrays are sparse
  - Support techniques to store sparse arrays
HOLAP (hybrid OLAP)

- HOLAP, combine ROLAP and MOLAP
- Controversy: which approach is the best?
  - MOLAP provides faster computation but supports smaller amount of data than ROLAP
  - ROLAP provide scalability, SQL standard has been extended

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
- **OLAP** operations: drilling, rolling, slicing, dicing and pivoting
- OLAP servers: ROLAP, MOLAP, HOLAP
Implementation & Computation of DW and Data Cube