

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

Lecture 8

## Typical OLAP Operations

### 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)

Lecture 7

- 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

1

Lecture 7

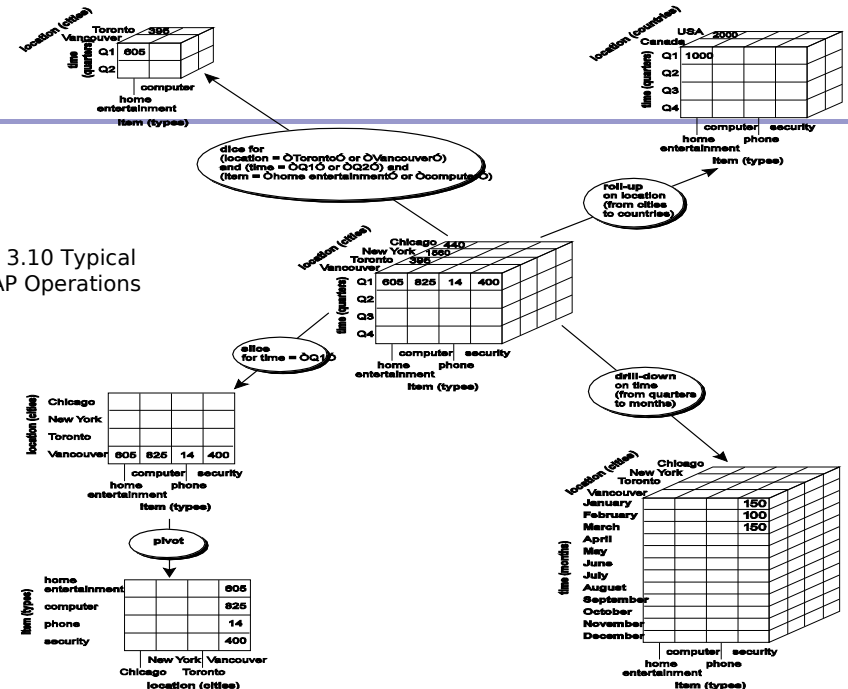


Fig. 3.10 Typical OLAP Operations

3

Lecture 7

2

4

# 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

Lecture 8

5

# Design of Data Warehouse: A Business Analysis Framework

- Four views in the design of 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

Lecture 8

6

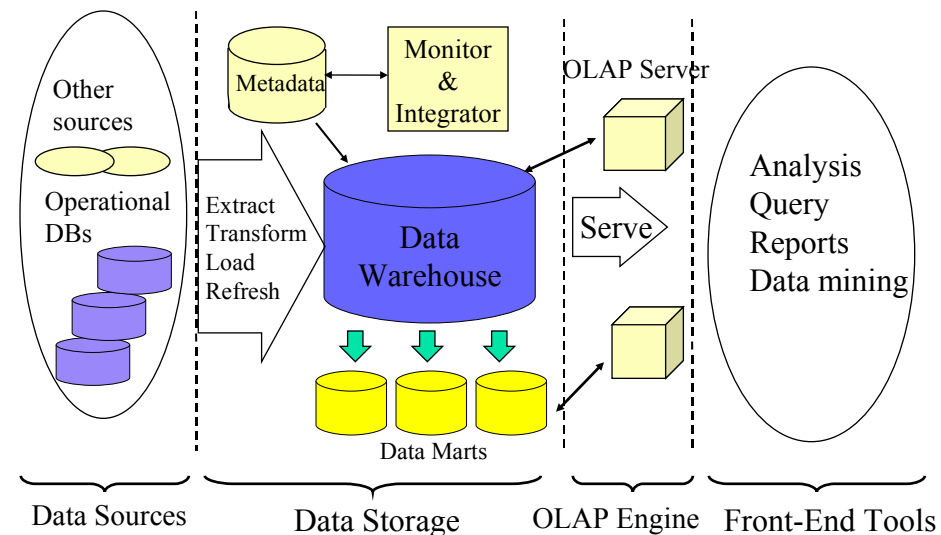
## Data Warehouse Design Process

- 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

Lecture 8

7

## Data Warehouse: A Multi-Tiered Architecture



Lecture 8

8

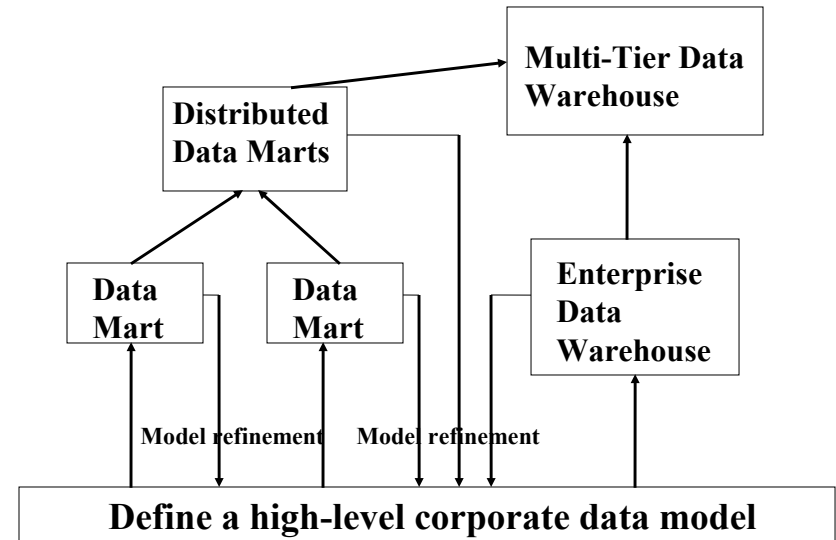
## Three Data Warehouse Models

- **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
- **Virtual warehouse**
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized

Lecture 8

9

## Data Warehouse Development: A Recommended Approach



Lecture 8

10

## 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

Lecture 8

11

## 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

Lecture 8

12

# 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**
- 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.

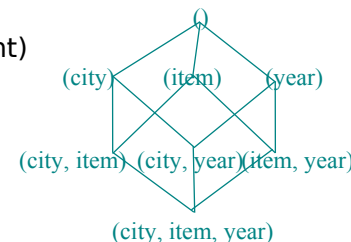
## Cube Operation

- 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)
()
```



## Iceberg Cube



- Computing only the cuboid cells whose count or other aggregates satisfying the condition like  
 $\text{HAVING COUNT}(\ast) \geq \text{minsup}$

### Motivation

- Only a small portion of cube cells may be “above the water” in a sparse cube
- Only calculate “interesting” cells—data above certain threshold
- Avoid explosive growth of the cube
  - Suppose 100 dimensions, only 1 base cell. How many aggregate cells if count  $\geq 1$ ? What about count  $\geq 2$ ?

Lecture 8

17

## 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

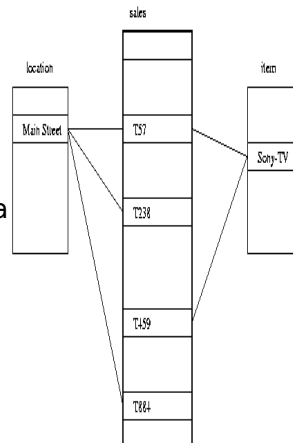
Base table			Index on Region			Index on Type			
Cust	Region	Type	RecID	Asia	Europe	America	RecID	Retail	Dealer
C1	Asia	Retail	1	1	0	0	1	1	0
C2	Europe	Dealer	2	0	1	0	2	0	1
C3	Asia	Dealer	3	1	0	0	3	0	1
C4	America	Retail	4	0	0	1	4	1	0
C5	Europe	Dealer	5	0	1	0	5	0	1

Lecture 8

18

## Indexing OLAP Data: Join Indices

- Join index:  $\text{JI}(\text{R-id}, \text{S-id})$  where  $\text{R}(\text{R-id}, \dots)$   
 $\triangleright \triangleleft \text{S}(\text{S-id}, \dots)$
- 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



Lecture 8

19

## Efficient Processing OLAP Queries

- Determine which operations should be performed on 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  $\{\text{brand}, \text{province\_or\_state}\}$  with the condition “year = 2004”, and there are 4 materialized cuboids available:
    - $\{\text{year}, \text{item\_name}, \text{city}\}$
    - $\{\text{year}, \text{brand}, \text{country}\}$
    - $\{\text{year}, \text{brand}, \text{province\_or\_state}\}$
    - $\{\text{item\_name}, \text{province\_or\_state}\}$  where year = 2004
 Which should be selected to process the query?
- Explore indexing structures and compressed vs. dense array structs in MOLAP

Lecture 8

20

## 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

Lecture 8

21

## 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**

Lecture 8

23

## 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

Lecture 8

22

## 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

Lecture 8

24

## 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
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. ICDE'97
- S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26:65-74, 1997
- E. F. Codd, S. B. Codd, and C. T. Salley. Beyond decision support. *Computer World*, 27, July 1993.
- J. Gray, et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29-54, 1997.
- A. Gupta and I. S. Mumick. *Materialized Views: Techniques, Implementations, and Applications*. MIT Press, 1999.
- J. Han. Towards on-line analytical mining in large databases. *ACM SIGMOD Record*, 27:97-107, 1998.
- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. SIGMOD'96

Lecture 8

25

## References (II)

- C. Imhoff, N. Galemme, and J. G. Geiger. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. John Wiley, 2003
- W. H. Inmon. *Building the Data Warehouse*. John Wiley, 1996
- R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2ed. John Wiley, 2002
- P. O'Neil and D. Quass. Improved query performance with variant indexes. SIGMOD'97
- Microsoft. OLEDB for OLAP programmer's reference version 1.0. In <http://www.microsoft.com/data/oledb/olap>, 1998
- A. Shoshani. OLAP and statistical databases: Similarities and differences. PODS'00.
- S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. ICDE'94
- OLAP council. MDAPI specification version 2.0. In <http://www.olapcouncil.org/research/apily.htm>, 1998
- E. Thomsen. *OLAP Solutions: Building Multidimensional Information Systems*. John Wiley, 1997
- P. Valduriez. Join indices. *ACM Trans. Database Systems*, 12:218-246, 1987.
- J. Widom. Research problems in data warehousing. CIKM'95.

Lecture 8

26