Data Warehouse and Data Mining

Dhruv Gupta

dhruv.gupta@ntnu.no

17-January-2022

NTNU | Norwegian University of Science and Technology

1. QA and References
   - Answers to Previous Questions
   - References for Today's Lecture
2. Data Cube
   - Concept Hierarchy Examples
   - Measures
3. Data Warehouse Concepts
   - Definition
   - Applications
   - Database Comparision
   - Data Warehouse Architecture
   - Data Warehouse Models
   - ETL Process
   - Metadata Repository
4. Data Warehouse Implementation
   - Data Cube Computation
   - Indexing OLAP Data
   - OLAP Server Architectures
5. Summary

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Answers

1. Lecture Plan.
   - Corrected and posted on BlackBoard.
   - Lecture plan and assignment deadlines tentative — may change over semester.
   - For the coming weeks, lectures only on Tuesday at 0815-1000 hrs.
2. TDT4100 exam date clash?
   - Nothing can be done as the course does not belong to the third year study program.
3. Lecture on MDX next Tuesday (one hour).
4. First Assignment
   - Hopefully, available by Wednesday and due by 10.February.2021.
5. Voice clear?

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

References for "Origins of Data Warehouses"

1. Conference Article:
   Gray et al. *"Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals"*. ICDE'96.
2. Extended Article:
   `https://arxiv.org/abs/cs/0701155`
3. All text and images for "Origins of Data Warehouse" are based on the arXiv preprint.

Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals

Jim Gray	Microsoft	Gray@Microsoft.com
Adam Bosworth	Microsoft	AdamB@Microsoft.com
Andrew Layman	Microsoft	AndrewL@Microsoft.com
Hamid Pirahesh	IBM	Pirahesh@Almaden.IBM.com

Abstract: *Data analysis applications typically aggregate data across many dimensions looking for unusual patterns. The SQL aggregate functions and the* `GROUP BY` *operator produce zero-dimensional or one-dimensional answers. Applications need the N-dimensional generalization of these operators. This paper defines that operator, called the* data cube *or simply* cube. *The cube operator generalizes the histogram, cross-tabulation, roll-up, drill-down, and sub-total constructs found in most report writers. The cube treats each of the N aggregation attributes as a dimension of N-space. The aggregate of a particular set of attribute values is a point in this space. The set of points forms an N-dimensional cube. Super-aggregates are computed by aggregating the N-cube to lower dimensional spaces. Aggregation points are represented by an "infinite value",* `ALL`, *so the point (*`ALL,ALL,...,ALL`, *sum(*)) represents the global sum of all items. Each* `ALL` *value actually represents the set of values contributing to that aggregation.*

1. Introduction

Data analysis applications look for unusual patterns in data. They summarize data values, extract statistical information, and then contrast one category with another. There are two steps to such data analysis:

- **extracting** the aggregated data from the database into a file or table, and
- **visualizing** the results in a graphical way.

Visualization tools display trends, clusters, and differences. The most exciting work in data analysis focuses on presenting new graphical metaphors that allow people to discover data trends and anomalies. Many tools represent the dataset as an *N*-dimensional space. Two and three-dimensional sub-slabs of this space are rendered as 2D or 3D objects. Color and time (motion) add two more dimensions to the display giving the potential for a 5D display.

How do traditional relational databases fit into this picture? How can flat files (SQL tables) possibly model an *N*-dimensional problem? Relational systems model *N*-dimensional data as a relation with *N*-attribute domains. For example, 4-dimensional earth-temperature data is typically represented by a `Weather` table shown below. The first four columns represent the four dimensions: x, y, z, t. Additional columns represent measurements at the 4D points such as temperature, pressure, humidity, and wind velocity. Often these measured values are aggregates over time (the hour) or space (a measurement area).

Table 1: `Weather`

Time (UCT)	Latitude	Longitude	Altitude (m)	Temp (c)	Pres (mb)
27/11/94:1500	37:58:33N	122:45:28W	102	21	1009
27/11/94:1500	34:16:18N	27:05:55W	10	23	1024

The SQL standard provides five functions to aggregate the values in a table: `COUNT()`, `SUM()`, `MIN()`, `MAX()`, and `AVG()`. For example, the average of all measured temperatures is expressed as:

```
SELECT   AVG(Temp)
FROM     Weather;
```

In addition, SQL allows aggregation over distinct values. The following query counts the distinct number of reporting times in the `Weather` table:

```
SELECT   COUNT(DISTINCT Time)
FROM     Weather;
```

Many SQL systems add statistical functions (median, standard deviation, variance, etc.), physical functions (center of mass, angular momentum, etc.), financial analysis (volatility, Alpha, Beta, etc.), and other domain-specific functions.

Some systems allow users to add new aggregation functions. The Illustra system, for example, allows users to add aggregate functions by adding a program with the following three callbacks to the database system [Illustra]:

`Init (&handle)`: Allocates the handle and initializes the aggregate computation.

`Iter (&handle, value)`: Aggregates the next value into the current aggregate.

`value = Final(&handle)`: Computes and returns the resulting aggregate by using data saved in the handle. This invocation deallocates the handle.

Consider implementing the `Average()` function. The `handle` stores the `count` and the `sum` initialized to `zero`. When passed a new non-null value, `Iter()` increments the `count` and adds the `sum` to the value. The `Final()` call deallocates the `handle` and returns `sum` divided by `count`.

1063-6382/96 $5.00 © 1996 IEEE

References for "Origins of Data Warehouses"

1. Conference Article:
   Gray et al. *"Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals"*. ICDE'96.
2. Extended Article:
   `https://arxiv.org/abs/cs/0701155`
3. All text and images for "Origins of Data Warehouse" are based on the arXiv preprint.

Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals

Jim Gray
Surajit Chaudhuri
Adam Bosworth
Andrew Layman
Don Reichart
Murali Venkatrao
Frank Pellow
Hamid Pirahesh[1]

May 1997

Technical Report
MSR-TR-97-32

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

This paper appeared in *Data Mining and Knowledge Discovery* 1**(1): 29-53 (1997)**

[1] IBM Research, 500 Harry Road, San Jose, CA. 95120

References for "Data Warehouse Concepts & Impl."

1. Book: Han et al. *"Data Mining Concepts and Techniques"*, 3rd Edition, 2012, Morgan Kaufmann Publishers.
2. All text and images for "Data Warehouse Concepts and Implementation" are based on the book by Han et al.

References for "Bitmap and Join Indexing"

1. Book: Jensen et al. *"Multidimensional Databases and Data Warehousing"*, 2010, Morgan & Claypool Publishers.
2. All text and images for "Bitmap and Join Indexing" are based on the book by Jensen et al.

1. QA and References
   - Answers to Previous Questions
   - References for Today's Lecture
2. Data Cube
   - Concept Hierarchy Examples
   - Measures
3. Data Warehouse Concepts
   - Definition
   - Applications
   - Database Comparision
   - Data Warehouse Architecture
   - Data Warehouse Models
   - ETL Process
   - Metadata Repository
4. Data Warehouse Implementation
   - Data Cube Computation
   - Indexing OLAP Data
   - OLAP Server Architectures
5. Summary

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Concept Hierarchy Examples

Figure 4.10 Hierarchical and lattice structures of attributes in warehouse dimensions: (a) a hierarchy for *location* and (b) a lattice for *time*.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Measures

SALES			
Model	Year	Color	Sales
Chevy	1990	red	5
Chevy	1990	white	87
Chevy	1990	blue	62
Chevy	1991	red	54
Chevy	1991	white	95
Chevy	1991	blue	49
Chevy	1992	red	31
Chevy	1992	white	54
Chevy	1992	blue	71
Ford	1990	red	64
Ford	1990	white	62
Ford	1990	blue	63
Ford	1991	red	52
Ford	1991	white	9

All text from: Gray et al. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals". ICDE'96.

Measures

- CUBE operator proposed by Gray et al. allows aggregation function to be expressed over attributes or dimensions in the argument list.
- These aggregate functions or measures that are computed over dimensions or attributes can be of three types so that they can be efficiently computed over multi-dimensional data:
  1. Distributive.
  2. Algebraic.
  3. Holistic.

All text from: Gray et al. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals". ICDE'96.

Measures — Distributive Functions

- Distributive Functions are aggregate function F() is distributive if there is a function G() such that:

$$\mathtt{F}(\{X_{i,j}\}) = \mathtt{G}(\{\mathtt{F}(\{X_{i,j} \mid i = 1, ..., I\}) \mid j = 1, ...J\}).$$

- COUNT(), MIN(), MAX(), SUM() are all distributive.
- In fact, F = G for all but COUNT(). G= SUM() for the COUNT() function. Once order is imposed, the cumulative aggregate functions also fit in the distributive class.

All text from: Gray et al. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals". ICDE'96.

Measures — Alegbraic Functions

- An aggregate function F() is algebraic if there is an M-tuple valued function G() and a function H() such that:

$$\mathtt{F}(\{\mathtt{X}_{i,j}\}) = \mathtt{H}(\{\mathtt{G}(\{\mathtt{X}_{i,j}|i = 1, .., I\})|j = 1, ..., J\}).$$

- AVERAGE(), STD_DIV, MaxN(), MinN(), CENTER_OF_MASS() are all algebraic.
- For AVERAGE, the function G() records the sum and count of the subset. The H() function adds these two components and then divides to produce the global average.
- Similar techniques apply to finding the N largest values, the center of mass of group of objects, and other algebraic functions. The key to algebraic functions is that a fixed size result (an M-tuple) can summarize the sub-aggregation.

All text from: Gray et al. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals". ICDE'96.

Measures — Holistic Functions

- Aggregate function `F()` is holistic if there is no constant bound on the size of the storage needed to describe a sub-aggregate.
- That is, there is no constant `M`, such that an M-tuple characterizes the computation $\mathtt{F}(\{\mathtt{X}_{i,j} \mid i = 1, ..., I\})$.
- `Median()`, `MostFrequent()` (also called the `Mode()`), and `Rank()` are common examples of holistic functions.

All text from: Gray et al. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals". ICDE'96.

1. QA and References
   - Answers to Previous Questions
   - References for Today's Lecture
2. Data Cube
   - Concept Hierarchy Examples
   - Measures
3. Data Warehouse Concepts
   - Definition
   - Applications
   - Database Comparision
   - Data Warehouse Architecture
   - Data Warehouse Models
   - ETL Process
   - Metadata Repository
4. Data Warehouse Implementation
   - Data Cube Computation
   - Indexing OLAP Data
   - OLAP Server Architectures
5. Summary

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Definition

- A semantically consistent data store that serves as a physical implementation of a decision support data model.
- It stores the information an enterprise needs to make strategic decisions.
- It is often viewed as an architecture, constructed by integrating data from multiple heterogeneous sources.
- It supports structured and/or ad hoc queries, analytical reporting, and decision making.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Definition

"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process."

— Wiliam H. Inmon

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Definition

- Data warehouses are subject-oriented:
  - Organized around major subjects
    (e.g., customer, supplier, product, and sales).
  - Focus on the modeling and analysis of data for decision makers
    (as opposed to day-to-day operations and transaction processing of an organization)
  - Provide a simple and concise view of particular subject issues by excluding data that are not useful in the decision support process.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Definition

- Data warehouses are integrated:
  - Usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and online transaction records.
  - Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Definition

- Data warehouses are time-variant:
  - Data are stored to provide information from an historic perspective (e.g., the past 5–10 years).
  - Every key structure in the data warehouse contains, either implicitly or explicitly, a time element.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Definition

- Data warehouses are nonvolatile:
  - Always a physically separate store of data transformed from the application data found in the operational environment.
  - A data warehouse does not require transaction processing, recovery, and concurrency control mechanisms.
  - It usually requires only two operations in data accessing:
    - Initial loading of data.
    - Access of data.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Business Intelligence & Decision Making

- Organizations use this information to support business decision-making activities:
  1. Increasing customer focus (i.e., analysis of customer buying patterns)
     - Buying preference,
     - Buying time,
     - Budget cycles,
     - Appetites for spending
  2. Comparing the performance of sales by quarter, by year, and by geographic regions in order to fine-tune production strategies.
  3. Analyzing operations and looking for sources of profit.
  4. Managing customer relationships.
  5. Making environmental corrections.
  6. Managing the cost of corporate assets.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Databases versus Data Warehouses

Table 4.1 Comparison of OLTP and OLAP Systems

Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements decision support
DB design	ER-based, application-oriented	star/snowflake, subject-oriented
Data	current, guaranteed up-to-date	historic, accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	GB to high-order GB	$\geq$ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

Note: Table is partially based on Chaudhuri and Dayal [CD97].

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Multi-Tier Architecture

Figure 4.1 A three-tier data warehousing architecture.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Bottom Tier

- Always a relational database system.
- Back-end tools and utilities are used to feed data from operational databases or other external sources.
- Back-end tools and utilities perform data extraction, cleaning, and transformation.
- Also contains a metadata repository, which stores information about the data warehouse and its contents.

Figure 4.1 A three-tier data warehousing architecture.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Middle Tier

- Is an OLAP server.
- Can be a relational OLAP (ROLAP) model
  - An extended relational DBMS that maps operations on multidimensional data to standard relational operations.
- Can be a a multi-dimensional OLAP (MOLAP) model
  - A special-purpose server that directly implements multidimensional data and operations.

Middle tier:
OLAP server

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Top Tier

- Is a front-end client layer that contains:
  - Query and reporting tools.
  - Analysis tools.
  - Data mining tools (e.g., trend analysis, prediction, and so on).

Top tier:
Front-end tools

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Data Warehouse Models

- Three data warehouse models:
  - Enterprise Warehouse
  - Data Mart
  - Virtual Warehouse

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Enterprise Warehouse

- Contains information about subjects spanning the entire organization.
- Provides corporate-wide data integration from
  1. Operational systems
  2. External information providers
- Scope: cross-functional.
- Contains detailed data as well as summarized data.
- Size: between gigabytes to terabytes (or beyond).
- Implementation on: traditional mainframes, computer superservers, or parallel architecture platforms.
- Implementation cycle: years (requires extensive business modeling).

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Data Mart

- Contains a subset of corporate-wide data that is of value to a specific group of users.
- Scope: specific selected subjects.
  - For example, a marketing data mart may confine its subjects to customer, item, and sales.
- Data contained tends to be summarized.
- Implementation on: low-cost departmental servers.
- Implementation cycle: weeks.
- Independent data marts:
  - Data captured from:
    - Operational systems
    - External information providers
    - Data generated locally within a department or geographic area.
- Dependent data marts:
  - Data captured from: enterprise data warehouses.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Virtual Warehouse

- Is a set of views over operational databases.
- For efficient query processing, only some of the possible summary views may be materialized.
- Easy to build but requires excess capacity on operational database servers.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Data Warehouse Development

- Implement the warehouse in an incremental and evolutionary manner.
- High-level corporate data model is defined.
- Independent data marts can be implemented in parallel.
- Distributed data marts can be constructed to integrate different data marts via hub servers.

Figure 4.2 A recommended approach for data warehouse development.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Extraction, Transformation, and Loading

- Data warehouse systems use back-end tools and utilities to populate and refresh their data.
  1. Data Extraction: gathers data from multiple, heterogeneous, and external sources.
  2. Data Cleaning: detects errors in the data and rectifies them when possible.
  3. Data Transformation: converts data from legacy or host format to warehouse format.
  4. Load: sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions.
  5. Refresh: propagates the updates from the data sources to the warehouse.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Metadata

- Metadata are data about data.
- Metadata are the data that define warehouse objects.
- Metadata are created and captured for:
  1. Timestamping any extracted data .
  2. Source of the extracted data.
  3. Missing fields that have been added by data cleaning or integration processes.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Metadata Repository

- Description of the data warehouse structure:
  1. Warehouse schema.
  2. View.
  3. Dimensions.
  4. Hierarchies.
  5. Derived data definitions.
  6. Data mart locations and contents.
- Operational metadata:
  1. Data lineage (history of migrated data and the sequence of transformations applied to it).
  2. Currency of data (active, archived, or purged).
  3. Monitoring information (warehouse usage statistics, error reports, and audit trails).

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Metadata Repository

- Algorithms used for summarization:
  1. Measure and dimension definition algorithms.
  2. Data on granularity.
  3. Partitions.
  4. Subject areas.
  5. Aggregation.
  6. Summarization.
  7. Predefined queries and reports.
- Mapping from the operational environment to the data warehouse:
  1. Source databases and their contents.
  2. Gateway descriptions.
  3. Data partitions.
  4. Data extraction, cleaning, transformation rules and defaults.
  5. Data refresh and purging rules.
  6. Security (user authorization and access control).

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Metadata Repository

- Data related to system performance:
  1. Indices and profiles that improve data access and retrieval performance.
  2. Rules for the timing and scheduling of refresh, update, and replication cycles.
- Business metadata:
  1. Business terms and definitions.
  2. Data ownership information.
  3. Charging policies.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1. QA and References
   - Answers to Previous Questions
   - References for Today's Lecture
2. Data Cube
   - Concept Hierarchy Examples
   - Measures
3. Data Warehouse Concepts
   - Definition
   - Applications
   - Database Comparision
   - Data Warehouse Architecture
   - Data Warehouse Models
   - ETL Process
   - Metadata Repository
4. Data Warehouse Implementation
   - Data Cube Computation
   - Indexing OLAP Data
   - OLAP Server Architectures
5. Summary

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Cube Operator

- A data cube is a lattice of cuboids.
- Lets analyze sales contained in the relation:

```
AllElectronics(item,year,city,sales)
```

- How many GROUP BY clauses do we need to generate the data cube for sales as a measure according to (item, sales, city) as dimensions?

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator

- A data cube is a lattice of cuboids.
- Lets analyze sales contained in the relation:

AllElectronics (item,year,city,sales)

- How many GROUP BY clauses do we need to generate the data cube for sales as a measure according to (item, sales, city) as dimensions?
- Answer: {(city, item, year), (city, item), (city, year), (item, year), (city), (item), (year), (∅)}

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator

Figure 4.14 Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains *city, item*, and *year* dimensions.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator

- Base cuboid contains all three dimensions, city, item, and year. Can return the total sales for any combination of the three dimensions.
- Apex cuboid, or 0-D cuboid contains the total sum of all sales.
- Base cuboid is the least generalized (most specific) of the cuboids.
- Apex cuboid is the most generalized (least specific) of the cuboids, and is often denoted as `ALL`.
- Drill Down within Data Cube: start at the apex cuboid and explore downward in the lattice.
- Roll Up within Data Cube: start at the base cuboid and explore upward.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator

- A cube operator on n dimensions is equivalent to a collection of group-by statements, one for each subset of the n dimensions.
- Therefore, the cube operator is the n-dimensional generalization of the group-by operator.
- The data cube for the example can be defined as:

```
DEFINE CUBE sales_cube [city, item, year]: SUM(sales)
```

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator

- For a cube with n dimensions, there are a total of 2^n cuboids, including the base cuboid.
- `COMPUTE CUBE sales_cube` would explicitly instruct the system to compute the sales aggregate cuboids for all eight subsets of the set `city`, `item`, `year`, including the empty subset.
- Good idea to compute in advance all or at least some of the cuboids in a data cube.
- Precomputation leads to fast response time and avoids some redundant computation.
- Most OLAP products resort to some degree of precomputation of multidimensional aggregates.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator Precomputations

- For a cube with n dimensions, there are a total of 2^n cuboids, including the base cuboid.

- How many cuboids are there for n dimensions where each ith dimension (e.g., `date`) has L_i levels (e.g., {`year`, `month`, `day`})?

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator Precomputations

- For a cube with n dimensions, there are a total of 2^n cuboids, including the base cuboid.

- How many cuboids are there for n dimensions where each ith dimension (e.g., `date`) has L_i levels (e.g., {`year`, `month`, `day`})?

$$\text{Total Number of Cubiods} = \prod_{i=1}^{n} \left(L_i + 1\right).$$

- L_i is the number of levels in dimension i and 1 is added to account for `ALL` value.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator Precomputations

- For a cube with n dimensions, there are a total of 2^n cuboids, including the base cuboid.

- How many cuboids are there for n dimensions where each ith dimension (e.g., `date`) has L_i levels (e.g., {`year`, `month`, `day`})?

$$\text{Total Number of Cubiods} = \prod_{i=1}^{n} \left(L_i + 1\right).$$

- L_i is the number of levels in dimension i and 1 is added to account for `ALL` value.
- For example, $n = 10$, $|L_i| = 5 \quad \forall 1 \leq i \leq 10$, then total number of cubiods = $5^{10} \approx 9.8 \times 10^6$!

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Operator Precomputations

- It is unrealistic to precompute and materialize all of the cuboids that can possibly be generated for a data cube (i.e., from a base cuboid).
- If there are many cuboids, and these cuboids are large in size, a more reasonable option is partial materialization.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Cube Materialization

- No Materialization: compute expensive multidimensional aggregates on-the-fly (very slow).
- Full Materialization: precompute all of the cuboids (full cube). Requires large memory space.
- Partial Materialization: selectively compute a proper subset of the whole set of possible cuboids.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Partial Cube Materialization

- Selection of subset of cuboids to materialize may depend on:
  - Query Workload: frequencies and access cost.
  - Workload Characteristics: update cost and storage requirements.
  - Physical Database Design: generation and selection of indices.
  - Heuristics: based on frequently referenced cuboids.
- Iceberg Cube: which is a data cube that stores only those cube cells with an aggregate value (e.g., count) that is above some minimum support threshold.
- Shell Cube: precomputes cuboids for a small number of dimensions (e.g., three to five). Queries on additional combinations of the dimensions can be computed on-the-fly.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

Indexing Overview

- Fact tables are typically very large and are used in the majority of queries.
- Therefore, a fact table often has many indices such that most queries can benefit from one or more of them.
- Typically, a separate index is built on each dimension key.

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Indexing Overview

- If the DBMS supports index intersection, the indices can then be used in combination when answering a query.
- It can also be a good idea, however, to create indices on combinations of dimension keys for those combinations that are often used together by queries.
- The combinations for which to create indices thus have to be chosen carefully based on the typical usage.
- In general, the key referencing the Date dimension should be put first in combinations since most queries refer to dates.

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Indexing Overview

- Virtually any DBMS supports the B-tree index which is a tree structure where the leaves contain lists of row IDs (where a row ID can be a physical location of the row or something else the system can use to identify and find a certain row).
- To find rows that have a given value for the indexed attribute, the system traverses the tree and obtains a list of row IDs. The row IDs are then used to retrieve the rows.
- The B-tree is an efficient index.
- However, when the indexed attribute has low cardinality (i.e., holds few different values), another representation can be a better choice.

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Bitmap Indexing

- When there are few values in a column, we can create a position bitmap for each value instead of maintaining lists of row IDs for each value (as in the B-tree).
- We call such a collection of position bitmaps a bitmap index.

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Bitmap Indexing

- When there are few values in a column, we can create a position bitmap for each value instead of maintaining lists of row IDs for each value (as in the B-tree).
- We call such a collection of position bitmaps a bitmap index.

Base table

RID	*item*	*city*
R1	H	V
R2	C	V
R3	P	V
R4	S	V
R5	H	T
R6	C	T
R7	P	T
R8	S	T

item bitmap index table

RID	H	C	P	S
R1	1	0	0	0
R2	0	1	0	0
R3	0	0	1	0
R4	0	0	0	1
R5	1	0	0	0
R6	0	1	0	0
R7	0	0	1	0
R8	0	0	0	1

city bitmap index table

RID	V	T
R1	1	0
R2	1	0
R3	1	0
R4	1	0
R5	0	1
R6	0	1
R7	0	1
R8	0	1

Note: H for "home entertainment," C for "computer," P for "phone," S for "security," V for "Vancouver," T for "Toronto."

Bitmap Vector

Figure 4.15 Indexing OLAP data using bitmap indices.

All from: 1. Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.
2. Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Bitmap Indexing

RowID	BookID	Title	Binding	Language	...
1	9436	Winnie the Pooh	Hardcover	English	...
2	1029	Le Petit Prince	Paperback	French	...
3	8733	Alice in Wonderland	Paperback	English	...
4	2059	Wind in the Willows	Hardcover	English	...
5	5995	A Bear Called Paddington	Paperback	English	...
6	1031	Pierre Lapin	Hardcover	French	...
7	3984	Le avventure di Pinocchio	Hardcover	Italian	...

Figure 4.2: Book dimension table also showing the special system-maintained attribute RowID, which normally cannot be seen by the user

Bitmap for `Binding`

```
Hardcover: 1001011
Paperback: 0110100
```

Bitmap for `Language`

```
English:  1011100
French:   0100010
Italian:  0000001
```

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Bitmap Indexing

- A bitmap index makes it very fast to locate rows with a certain value.
- It is also possible to combine bitmap indices.
- Example 1: To find all hardcover books in English can be done by computing the logical AND of the position bitmaps for hardcover in Binding and for English in Language. In other words, we compute:

```
        1001011
AND     1011100
=       1001000
```

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Bitmap Indexing

- A bitmap index makes it very fast to locate rows with a certain value.
- It is also possible to combine bitmap indices.
- Example 2: To find books written in French or Italian, we compute logical OR of the position bitmaps for French and Italian in Language:

```
        0100010
OR      0000001
  =     0100011
```

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Bitmap Indexing

- A bitmap index makes it very fast to locate rows with a certain value.
- It is also possible to combine bitmap indices.
- Example 2: To find books written in French or Italian, we compute logical OR of the position bitmaps for French and Italian in Language:

```
        0100010
OR      0000001
  =     0100011
```

- The logical AND can be computed very efficiently as a single CPU operation can compare 32 or 64 bits on a modern CPU.

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Join Indexing

- Traditional indexing maps the value in a given column to a list of rows having that value.
- In contrast, join indexing registers the joinable rows of two relations from a relational database.
- For example, if two relations `R(RID, A)` and `S(B, SID)` join on the attributes A and B, then the join index record contains the pair (RID, SID), where RID and SID are record identifiers from the R and S relations, respectively.
- Join index records can identify joinable tuples without performing costly join operations.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Join Indexing

- Consider star schema for AllElectronics of the form sales_star [time, item, branch, location]: dollars_sold = SUM(sales_in_dollars)"

Figure 4.16 Linkages between a *sales* fact table and *location* and *item* dimension tables.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Join Indexing

- Consider star schema for `AllElectronics` of the form `sales_star [time, item, branch, location]: dollars_sold = SUM(sales_in_dollars)`"

Join index table for *location/sales*

location	*sales_key*
. . .	. . .
Main Street	T57
Main Street	T238
Main Street	T884
. . .	. . .

Join index table for *item/sales*

item	*sales_key*
. . .	. . .
Sony-TV	T57
Sony-TV	T459
. . .	. . .

Join index table linking *location* and *item* to *sales*

location	*item*	*sales_key*
. . .	. . .	. . .
Main Street	Sony-TV	T57
. . .	. . .	. . .

Figure 4.17 Join index tables based on the linkages between the *sales* fact table and the *location* and *item* dimension tables shown in Figure 4.16.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Join Indexing

RowID	BookID	Title	Genre	...
1	7493	Tropical Food	Cooking	...
2	9436	Winnie the Pooh	Childrens' books	...
3	9948	Gone With the Wind	Fiction	...
4	9967	Italian Food	Cooking	...

Book (dimension table)

RowID	BookID	ShopID	SalesID	DayID	Count	Price
1	9436	854	1021	2475	2	30
2	7493	854	1021	2475	1	20
3	9948	876	2098	3456	1	20
4	7493	876	2231	3456	2	40
5	7493	876	3049	2475	1	20
6	9436	854	3362	3569	2	30
7	9967	731	3460	3569	1	35
8	7493	731	3460	3569	1	15
9	9948	731	3460	3569	1	15

Sales (fact table)

Book_RowID	*Sales_RowID*
1	⟨*2, 4, 5, 8*⟩
2	⟨*1, 6*⟩
3	⟨*3, 9*⟩
4	⟨*7*⟩

Join index for Book and Sales

Figure 4.3: A fact table, a dimension table, and a join index with a list of pointers

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.

Join Indexing

- Join indexing is especially useful for maintaining the relationship between a foreign key and its matching primary keys, from the joinable relation.
- Star schema model of data warehouses makes join indexing attractive for cross-table search, because the linkage between a fact table and its corresponding dimension tables comprises the fact table's foreign key and the dimension table's primary key.
- Join indices may span multiple dimensions to form composite join indices. We can use join indices to identify subcubes that are of interest.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Efficient Processing of OLAP Queries

- The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes.
- Given materialized views, query processing should proceed as follows:
  1. Determine which operations should be performed on the available cuboids:
     - Transform any selection, projection, roll-up (group-by), and drill-down operations specified in the query into corresponding SQL and/or OLAP operations.
     - For example, slicing and dicing a data cube corresponds to selection and/or projection operations on a materialized cuboid.
  2. Determine to which materialized cuboid(s) the relevant operations should be applied:
     - Identifying materialized cuboids with query answering potential and selecting the cuboid with the least cost.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Efficient Processing of OLAP Queries

- Example OLAP query processing: Suppose a data cube for AllElectronics of the form sales cube [time, item, location]: sum (sales in dollars).
- Dimension hierarchies:
  - time: day < month < quarter < year
  - item: item name < brand < type
  - location: street < city < province_or_state < country
- Query needs to be processed on {brand, province_or_state} with the selection constant year = 2010
- Four materialized cuboids:
  - cuboid 1: {year, item name, city}
  - cuboid 2: {year, brand, country}
  - cuboid 3: {year, brand, province_or_state}
  - cuboid 4: {item name, province_or_state}, where year = 2010
- Which of the four cuboids should be selected for query processing?

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1 QA and References
- Answers to Previous Questions
- References for Today's Lecture

2 Data Cube
- Concept Hierarchy Examples
- Measures

3 Data Warehouse Concepts
- Definition
- Applications
- Database Comparision
- Data Warehouse Architecture
- Data Warehouse Models
- ETL Process
- Metadata Repository

4 Data Warehouse Implementation
- Data Cube Computation
- Indexing OLAP Data
- OLAP Server Architectures

5 Summary

ROLAP

- Relational OLAP (ROLAP) servers:
  - Intermediate servers that stand in between a relational back-end server and client front-end tools.
  - Use a relational or extended-relational DBMS to store and manage warehouse data, and OLAP middleware to support missing pieces.
  - ROLAP servers include optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services.
  - ROLAP technology tends to have greater scalability than MOLAP technology.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

MOLAP

- Multidimensional OLAP (MOLAP) servers:
  - These servers support multidimensional data views through array-based multidimensional storage engines.
  - They map multi-dimensional views directly to data cube array structures.
  - The advantage of using a data cube is that it allows fast indexing to precomputed summarized data.
  - Notice that with multidimensional data stores, the storage utilization may be low if the data set is sparse.
  - In such cases, sparse matrix compression techniques should be explored.
  - MOLAP servers adopt a two-level storage representation to handle dense and sparse data sets: Denser subcubes are identified and stored as array structures, whereas sparse subcubes employ compression technology for efficient storage. utilization.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

HOLAP

- Hybrid OLAP (HOLAP) servers:
  - The hybrid OLAP approach combines ROLAP and MOLAP technology.
  - Benefiting from the greater scalability of ROLAP and the faster computation of MOLAP.
  - For example, a HOLAP server may allow large volumes of detailed data to be stored in a relational database, while aggregations are kept in a separate MOLAP store.
- Specialized SQL servers:
  - Provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

1. QA and References
   - Answers to Previous Questions
   - References for Today's Lecture
2. Data Cube
   - Concept Hierarchy Examples
   - Measures
3. Data Warehouse Concepts
   - Definition
   - Applications
   - Database Comparision
   - Data Warehouse Architecture
   - Data Warehouse Models
   - ETL Process
   - Metadata Repository
4. Data Warehouse Implementation
   - Data Cube Computation
   - Indexing OLAP Data
   - OLAP Server Architectures
5. Summary

Summary

Figure 4.1 A three-tier data warehousing architecture.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Summary

Figure 4.14 Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains *city, item*, and *year* dimensions.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Summary

RowID	BookID	Title	Binding	Language	...
1	9436	Winnie the Pooh	Hardcover	English	...
2	1029	Le Petit Prince	Paperback	French	...
3	8733	Alice in Wonderland	Paperback	English	...
4	2059	Wind in the Willows	Hardcover	English	...
5	5995	A Bear Called Paddington	Paperback	English	...
6	1031	Pierre Lapin	Hardcover	French	...
7	3984	Le avventure di Pinocchio	Hardcover	Italian	...

Figure 4.2: Book dimension table also showing the special system-maintained attribute RowID, which normally cannot be seen by the user

Bitmap for `Binding`

```
Hardcover: 1001011
Paperback: 0110100
```

Bitmap for `Language`

```
English:  1011100
French:   0100010
Italian:  0000001
```

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

Summary

RowID	BookID	Title	Genre	...
1	7493	Tropical Food	Cooking	...
2	9436	Winnie the Pooh	Childrens' books	...
3	9948	Gone With the Wind	Fiction	...
4	9967	Italian Food	Cooking	...

Book (dimension table)

RowID	BookID	ShopID	SalesID	DayID	Count	Price
1	9436	854	1021	2475	2	30
2	7493	854	1021	2475	1	20
3	9948	876	2098	3456	1	20
4	7493	876	2231	3456	2	40
5	7493	876	3049	2475	1	20
6	9436	854	3362	3569	2	30
7	9967	731	3460	3569	1	35
8	7493	731	3460	3569	1	15
9	9948	731	3460	3569	1	15

Sales (fact table)

Book_RowID	*Sales_RowID*
1	⟨*2, 4, 5, 8*⟩
2	⟨*1, 6*⟩
3	⟨*3, 9*⟩
4	⟨*7*⟩

Join index for Book and Sales

Figure 4.3: A fact table, a dimension table, and a join index with a list of pointers

All from: Jensen et al. "Multidimensional Databases and Data Warehousing", 2010, Morgan Claypool Publishers.