Data Warehouses

Anne Denton

Department of Computer Science
North Dakota State University

Outline

1. Concepts of Data Warehouses
   - General properties of data warehouses
2. Multi-dimensional data model
   - Data cubes
   - Rollup and drilldown
   - Star schema

Table of Contents

Motivations for data warehouses

- Read-only transactions faster
  - Read-only transactions cannot block each other
- Integrated, subject-oriented collection of data
  - Items related to the same subject may be spread over multiple transactional databases
  - Data cleaning may be necessary to perform analytical processing
- Time variant, but non-volatile
  - Snapshots that don't change but keep track of historical development

OLAP vs. OLTP

- On-Line Transaction Processing (OLTP)
  - Objective of conventional transactional relational databases
  - Managing of day-to-day operations, like purchasing, inventory, accounting, etc.
- On-Line Analytical Processing (OLAP)
  - Objective of data warehouse systems
  - Data analysis and decision-making
- Distinct features

	OLTP	OLAP
Data	current and detailed	historical, consolidated
Design	relational design	star-schema + subject knowledge
View	current and detailed	evolutionary and integrated
Access	updates	read-only but complex queries

Reasons for a separate data warehouse

- Improves performance for both systems
  - Read-only queries with table scope would slow down transactional system
  - Small updates would slow down read-only queries with table scope
  - Similar to creating snapshots and using those for read-only queries
- Data representation
  - Multi-dimensional views
  - Data consolidation from multiple transactional databases
  - Filling in missing data and reconciling formats

Question 1

Could there be a reason to have a data warehouse that uses the same type of RDBMS as the transactional databases of a business

1. Yes
2. No

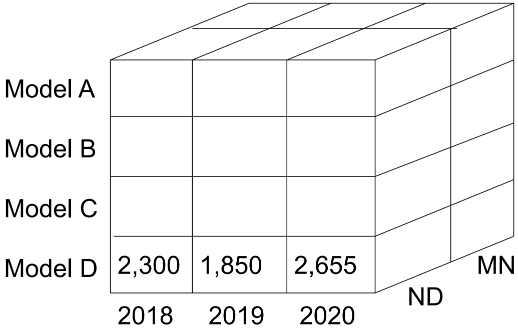
Table of Contents

Data cubes

- Generalization of spreadsheets to arbitrarily many dimensions
  - Based on representation of one “fact", for example, “sales", ie. the dollar-value of combined sales
  - Each explanatory attribute considered one dimension
  - Example: “sales" depending on location, time, and model
- Dimensions greater than three?
  - Fundamentally, one “fact" can depend on arbitrarily many dimensions
  - For the above example, “sales" could also depend on customer type, like educational vs. commercial
  - Also called hyper-cube, considering that cubes conventionally have 3 dimensions
  - Dimensions don’t have to have same number of entries, and they do not have to have numerical data
  - In the following “cube" will be used regardless of dimension

Data cubes

- 2 or 3 dimensional cubes can still be graphically represented
- For higher dimensions, slices can be shown



Question 2

How many dimensions would a data “cube" have that represents the following fact table of rainfall in various regions have?

Month	Region	Rainfall in inches
Jan	NW	5.4
Feb	NW	6.7
Jan	NE	12.1

1. 1
2. 2
3. 3
4. 4

Question 3

If the following data “cube" was represented as a relational table, how many columns would it have

	NW	NE
Jan	5.4	12.1
Feb	6.7	8.9
Mar	7.1	13.3

1. 2
2. 3
3. 4
4. 6

Question 4

If the following data “cube" was represented as a relational table, how many rows would it have

	NW	NE
Jan	5.4	12.1
Feb	6.7	8.9
Mar	7.1	13.3

1. 2
2. 3
3. 4
4. 6

Table of Contents

Rollup and drilldown

- Rollup along one dimension results in a cube of one fewer dimension
  - For example, the 3-dimensional “sales" cube can be rolled up along the model dimension to result in a 2-dimensional “cube" of sales values depending on location and time
  - Alternatively, it can be rolled up along the time dimension to result in a cube of sales depending on model and location
  - It can also be rolled-up along the model and time dimension to result in a 1-dimensional “cube" depending on location
- Rollup normally implemented using aggregate functions and group by statements
- Materialized views used for storing aggregates
- Which cubes should be materialized is question of physical design of the data warehouse

Question 5

If the following data “cube" was rolled up along the region dimension, what would be the dimensionality of the result?

	NW	NE
Jan	5.4	12.1
Feb	6.7	8.9
Mar	7.1	13.3

1. 1
2. 2
3. 3
4. 4

Question 6

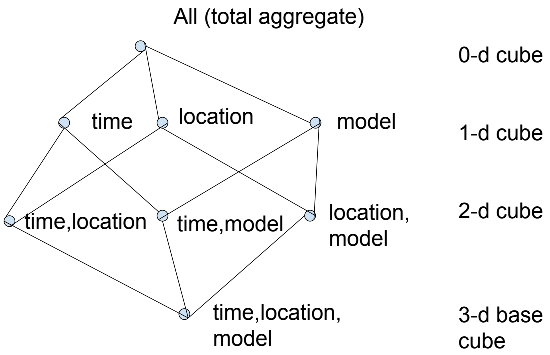
If the following data “cube" was rolled up along the region dimension, how many elements would the result have?

	NW	NE
Jan	5.4	12.1
Feb	6.7	8.9
Mar	7.1	13.3

1. 1
2. 2
3. 3
4. 4

Lattice of cubes

- The alternatives of roll-up can be represented as a lattice



Typical OLAP operations

- **Roll up:** summarize data through dimension reduction
- **Drill down:** reverse roll-up by going back to less aggregated data
- **Slice:** select elements matching a slice of the cube (i.e. reduced by one dimension)
- **Dice:** select elements matching cube of lower dimensionality
- **Pivot:** reorient the cube, visualization, 3D to series of 2D planes (i.e. rotate)
- **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)

Question 7 (Multiple answers can be correct)

Which of the following reduces the volume of results

1. Roll up
2. Drill down
3. Slice
4. Dice
5. Pivot

Question 8 (Multiple answers can be correct)

Which of the following involves involves multiple levels of aggregates

1. Roll up
2. Drill down
3. Slice
4. Dice
5. Pivot

Support for data cube functions in PostgreSQL

- Support for data cube functions used to be limited to dedicated data warehouse software
- Initially even a physical cube organization was considered as storage model
- Now, standard RDBMS offer adequate functionality
- Even open source RDBMSs like PostgreSQL `https://www.postgresqltutorial.com/postgresql-cube/`

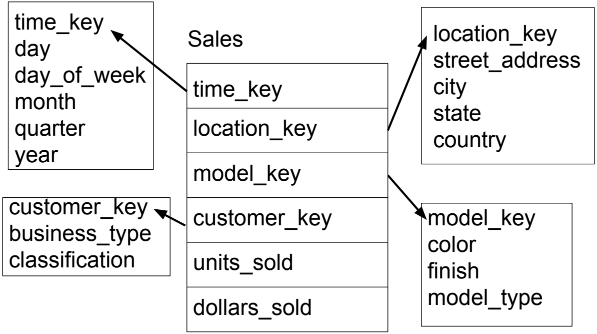
Table of Contents

Motivation for star schema

- Typically there is additional information to be represented for each dimension
  - Time may be broken down in years, months, and days
  - Location information may be complex
- This information may be needed for queries

Example of a star schema

- The dimensions may contain extensive information



Concept hierarchies

- Dimensions can have different granularities
- Examples
  - A year has multiple months, which has multiple days
  - A region has multiple districts with multiple stores
- Implementing functions that recognize concept hierarchies adds implementation challenges

Further extensions

- Snowflake schema
  - Branches may have further branches
- Constellation schema
  - Branches may interconnect
- Benefits of original star schema
  - Corresponds to normalizing dimension tables
  - Not always desired or necessary in data warehouses because there are no updates
  - Data warehouses typically show some level of “denormalization"
  - Little space wasted because of small size of dimension tables

Question 9 (Multiple answers can be correct)

A star schema is often preferred over a snowflake schema because

1. It is properly normalized
2. Storage takes less space
3. It involves fewer join operations

Summary

- Some functionality of data warehouses is now available as standard even in some open source databases like PostgreSQL
- Data warehouses are still useful for analytical processing
  - For performance reasons
  - For logistic reasons (allow data aggregation, cleaning, storing of snapshots of data)