Evolution of Database Systems

Krzysztof Dembczyński

Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland

Software Development Technologies
Master studies, second semester
Academic year 2018/19 (winter course)

Review of the Previous Lecture

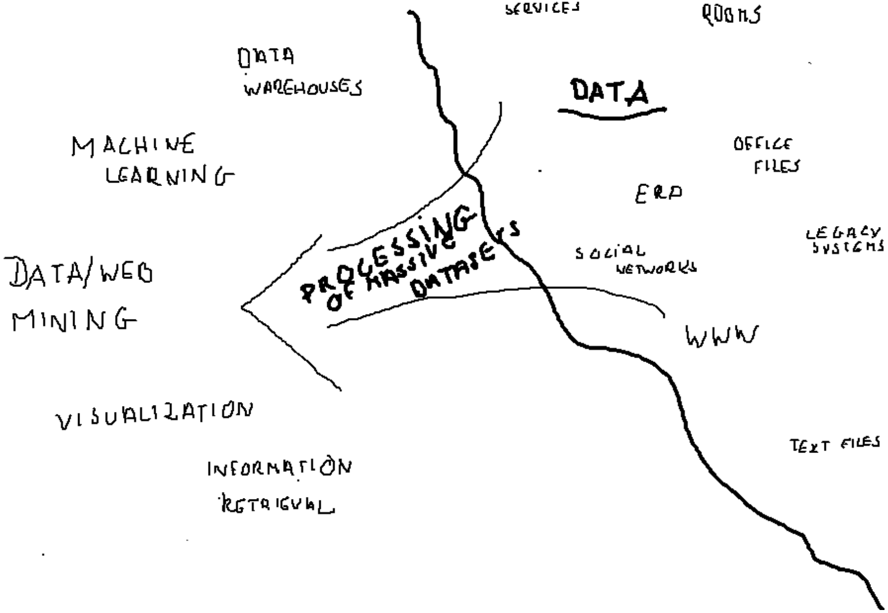
- Mining of massive datasets.
- Classification and regression.

Outline

1. Evolution of database systems
2. Analytical database systems
3. Processing of massive datasets
4. Summary

Outline

1. Evolution of database systems
2. Analytical database systems
3. Processing of massive datasets
4. Summary


SERVICES
DATA WAREHOUSES
DATA
MACHINE LEARNING
OFFICE FILES
ERP
LEGACY SYSTEMS
PROCESSING OF MASSIVE DATASETS
SOCIAL NETWORKS
DATA/WEB MINING
WWW
VISUALIZATION
TEXT FILES
INFORMATION RETRIEVAL

Data is the new oil (?)

Database management system

- A database is a collection of information that exists over a long period of time.

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.
- The DBMS is expected to:

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.
- The DBMS is expected to:
  - Allow users to create new databases and specify their schemas (logical structure of data),

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.
- The DBMS is expected to:
  - Allow users to create new databases and specify their schemas (logical structure of data),
  - Give users the ability of query the data and modify the data,

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.
- The DBMS is expected to:
  - Allow users to create new databases and specify their schemas (logical structure of data),
  - Give users the ability of query the data and modify the data,
  - Support the storage of very large amounts of data, allowing efficient access to data for queries and database modifications,

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.
- The DBMS is expected to:
    - Allow users to create new databases and specify their schemas (logical structure of data),
    - Give users the ability of query the data and modify the data,
    - Support the storage of very large amounts of data, allowing efficient access to data for queries and database modifications,
    - Enable durability, the recovery of the database in the face of failures,

Database management system

- A database is a collection of information that exists over a long period of time.
- A database management system (DBMS) is specialized software responsible for managing the database.
- The DBMS is expected to:
  - Allow users to create new databases and specify their schemas (logical structure of data),
  - Give users the ability of query the data and modify the data,
  - Support the storage of very large amounts of data, allowing efficient access to data for queries and database modifications,
  - Enable durability, the recovery of the database in the face of failures,
  - Control access to data from many users at once in isolation and ensure the actions on data to be performed completely.

Data models

- **Data model** is an abstract model that defines how data is represented and accessed.
  - **Logical data model** – from a user's point of view
  - **Physical data model** – from a computer's point of view.
- Data model defines:
  - Data objects and types, relationships between data objects, and constraints imposed on them.
  - Operations for defining, searching and updating data.

Approaches to data management

- File management system

Approaches to data management

- File management system
- Database management system

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems
  - Post-relational database systems

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems
  - Post-relational database systems
  - Object-based database systems

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems
  - Post-relational database systems
  - Object-based database systems
  - Multi-dimensional database systems

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems
  - Post-relational database systems
  - Object-based database systems
  - Multi-dimensional database systems
- NoSQL and BigData

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems
  - Post-relational database systems
  - Object-based database systems
  - Multi-dimensional database systems
- NoSQL and BigData
- *NewSQL*

Approaches to data management

- File management system
- Database management system
  - Early database management systems (e.g. hierarchical or network data models)
  - Relational database systems
  - Post-relational database systems
  - Object-based database systems
  - Multi-dimensional database systems
- NoSQL and BigData
- *NewSQL*
- **The choice of the approach strongly depends on a given application!**

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather "Not only" and **SQL** states for "traditional relational DBMS".

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather “Not only” and **SQL** states for “traditional relational DBMS”.
- NoSQL systems are alternative to traditional relational DBMS

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather "Not only" and **SQL** states for "traditional relational DBMS".
- NoSQL systems are alternative to traditional relational DBMS
  - Flexible schema (less restricted than typical RDBMS, but may not support join operations)

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather "Not only" and **SQL** states for "traditional relational DBMS".
- NoSQL systems are alternative to traditional relational DBMS
  - Flexible schema (less restricted than typical RDBMS, but may not support join operations)
  - Quicker/cheaper to set up

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather "Not only" and **SQL** states for "traditional relational DBMS".
- NoSQL systems are alternative to traditional relational DBMS
  - Flexible schema (less restricted than typical RDBMS, but may not support join operations)
  - Quicker/cheaper to set up
  - Massive scalability (scale-out instead of scale-up)

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather “Not only” and **SQL** states for “traditional relational DBMS”.
- NoSQL systems are alternative to traditional relational DBMS
  - Flexible schema (less restricted than typical RDBMS, but may not support join operations)
  - Quicker/cheaper to set up
  - Massive scalability (scale-out instead of scale-up)
  - Relaxed consistency $\rightarrow$ higher performance and availability, but fewer guarantees (like ACID)

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather "Not only" and **SQL** states for "traditional relational DBMS".
- NoSQL systems are alternative to traditional relational DBMS
  - Flexible schema (less restricted than typical RDBMS, but may not support join operations)
  - Quicker/cheaper to set up
  - Massive scalability (scale-out instead of scale-up)
  - Relaxed consistency → higher performance and availability, but fewer guarantees (like ACID)
  - Not all operations supported (e.g., join operation)

What is NoSQL?

- Not every data management/analysis problem is best solved exclusively using a traditional relational DBMS
- **No** means rather "Not only" and **SQL** states for "traditional relational DBMS".
- NoSQL systems are alternative to traditional relational DBMS
  - Flexible schema (less restricted than typical RDBMS, but may not support join operations)
  - Quicker/cheaper to set up
  - Massive scalability (scale-out instead of scale-up)
  - Relaxed consistency $\rightarrow$ higher performance and availability, but fewer guarantees (like ACID)
  - Not all operations supported (e.g., join operation)
  - No declarative query language (requires more programming, but new paradigms like MapReduce appear)

NoSQL

- Different types of models:

NoSQL

- Different types of models:
  - MapReduce frameworks,

NoSQL

- Different types of models:
  - MapReduce frameworks,
  - key-values stores,

NoSQL

- Different types of models:
  - MapReduce frameworks,
  - key-values stores,
  - column stores and BigTable implementations,

NoSQL

- Different types of models:
  - MapReduce frameworks,
  - key-values stores,
  - column stores and BigTable implementations,
  - document-oriented databases,

NoSQL

- Different types of models:
  - MapReduce frameworks,
  - key-values stores,
  - column stores and BigTable implementations,
  - document-oriented databases,
  - graph database systems.

NoSQL

- Different types of models:
  - MapReduce frameworks,
  - key-values stores,
  - column stores and BigTable implementations,
  - document-oriented databases,
  - graph database systems.
- Design for different purposes.

BigData – a lot of Vs[1]

- **Volume**: the quantity of generated and stored data.
- **Variety**: the type and nature of the data.
- **Velocity**: the speed at which the data is generated and processed.
- **Variability**: inconsistency of the data.
- **Value**: the value of the data.

[1] https://en.wikipedia.org/wiki/Big_data

Two types of systems

- Operational systems:

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).
  - **Main tasks**: processing of a huge number of concurrent transactions, and insuring data integrity.

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).
  - **Main tasks**: processing of a huge number of concurrent transactions, and insuring data integrity.
- Analytical systems:

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).
  - **Main tasks**: processing of a huge number of concurrent transactions, and insuring data integrity.
- Analytical systems:
  - support knowledge workers (e.g., manager, executive, analyst) in decision making.

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).
  - **Main tasks**: processing of a huge number of concurrent transactions, and insuring data integrity.
- Analytical systems:
  - support knowledge workers (e.g., manager, executive, analyst) in decision making.
  - Also referred to as **on-line analytical processing** (OLAP).

Two types of systems

- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).
  - **Main tasks**: processing of a huge number of concurrent transactions, and insuring data integrity.
- Analytical systems:
  - support knowledge workers (e.g., manager, executive, analyst) in decision making.
  - Also referred to as **on-line analytical processing** (OLAP).
  - **Main tasks**: effective processing of multidimensional queries concerning huge volumes of data.

Two types of systems

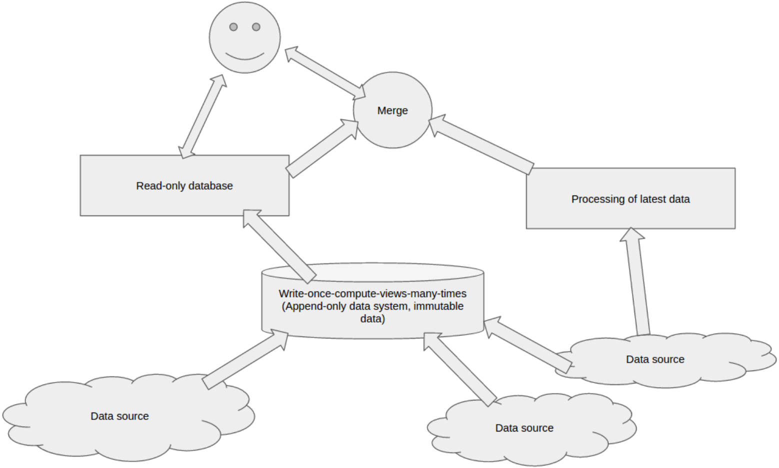
- Operational systems:
  - Support day-to-day operations of an organization.
  - Also referred to as **on-line transaction processing** (OLTP).
  - **Main tasks**: processing of a huge number of concurrent transactions, and insuring data integrity.
- Analytical systems:
  - support knowledge workers (e.g., manager, executive, analyst) in decision making.
  - Also referred to as **on-line analytical processing** (OLAP).
  - **Main tasks**: effective processing of multidimensional queries concerning huge volumes of data.
  - Database systems of a **write-once-read-many-times** type.

Outline

Analytical database systems

- Data warehouses,
- Business intelligence,
- Computational and analytical tools,
- Scientific databases,
- Analytics engines for large-scale data processing.

Analytical database systems



Analytical database systems

- The old and still good definition of the data warehouse:

Analytical database systems

- The old and still good definition of the data warehouse:
  - **Data warehouse** is defined as a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.

Analytical database systems

- The old and still good definition of the data warehouse:
  - **Data warehouse** is defined as a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.
    - **Subject oriented**: oriented to the major subject areas of the corporation that have been defined in the data model.

Analytical database systems

- The old and still good definition of the data warehouse:
  - **Data warehouse** is defined as a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.
    - **Subject oriented**: oriented to the major subject areas of the corporation that have been defined in the data model.
    - **Integrated**: there is no consistency in encoding, naming conventions, etc., among different data sources that are heterogeneous data sources (when data is moved to the warehouse, it is converted).

Analytical database systems

- The old and still good definition of the data warehouse:
  - **Data warehouse** is defined as a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.
    - **Subject oriented**: oriented to the major subject areas of the corporation that have been defined in the data model.
    - **Integrated**: there is no consistency in encoding, naming conventions, etc., among different data sources that are heterogeneous data sources (when data is moved to the warehouse, it is converted).
    - **Non-volatile**: warehouse data is loaded and accessed; update of data does not occur in the data warehouse environment.

Analytical database systems

- The old and still good definition of the data warehouse:
  - **Data warehouse** is defined as a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.
    - **Subject oriented**: oriented to the major subject areas of the corporation that have been defined in the data model.
    - **Integrated**: there is no consistency in encoding, naming conventions, etc., among different data sources that are heterogeneous data sources (when data is moved to the warehouse, it is converted).
    - **Non-volatile**: warehouse data is loaded and accessed; update of data does not occur in the data warehouse environment.
    - **Time-variant**: the time horizon for the data warehouse is significantly longer than that of operational systems.

Life-cycle of analytical database systems

- Logical design of the database
- Design and implementation of ETL (extraction, transformation, load) process
- Deployment of the system
- Optimization of the system
- Refreshing of the data

Logical design of the database

- University authorities decided to analyze teaching performance by using the data collected in databases owned by the university containing information about students, instructors, lectures, faculties, etc.
- They would like to get answers for the following queries:

Logical design of the database

- University authorities decided to analyze teaching performance by using the data collected in databases owned by the university containing information about students, instructors, lectures, faculties, etc.
- They would like to get answers for the following queries:
  - What is the average score of students over academic years?
  - What is the number of students over academic years?
  - What is the average score by faculties, instructors, etc.?
  - What is the distribution of students over faculties, semesters, etc.?
  - ...

Example

- An exemplary query could be the following:

```
SELECT Instructor, Academic_year, AVG(Grade)
FROM Data_Warehouse
GROUP BY Instructor, Academic_year
```

- And the result:

Academic_year	Name	AVG(Grade)
2013/14	Stefanowski	4.2
2014/15	Stefanowski	4.5
2013/14	Słowiński	4.1
2014/15	Słowiński	4.3
2014/15	Dembczyński	4.6

Motivation

- The result is also commonly given as a pivot table:

AVG(Grade)	Academic_year	
Name	2013/2014	2014/2015
Stefanowski	4.2	4.5
Słowiński	4.1	4.3
Dembczyński		4.6

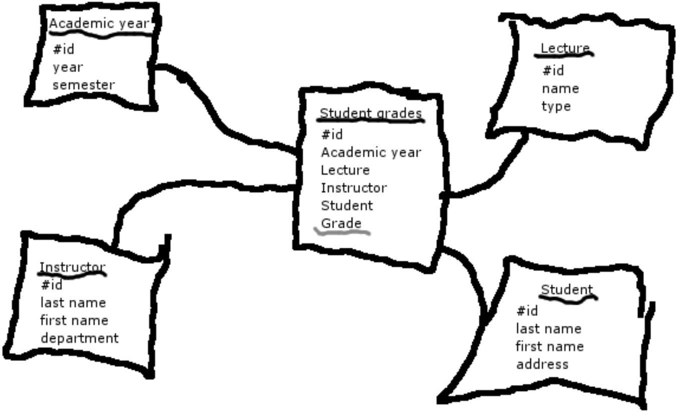
- Three main goals for logical design:
  - Simplicity:
    - Users should understand the design,
    - Data model should match users' conceptual model,
    - Queries should be easy and intuitive to write.
  - Expressiveness:
    - Include enough information to answer all important queries,
    - Include all relevant data (without irrelevant data).
  - Performance:
    - An efficient physical design should be possible to apply.

Three basic conceptual schemes

- Star schema,
- Snowflake schema,
- Fact constellations.

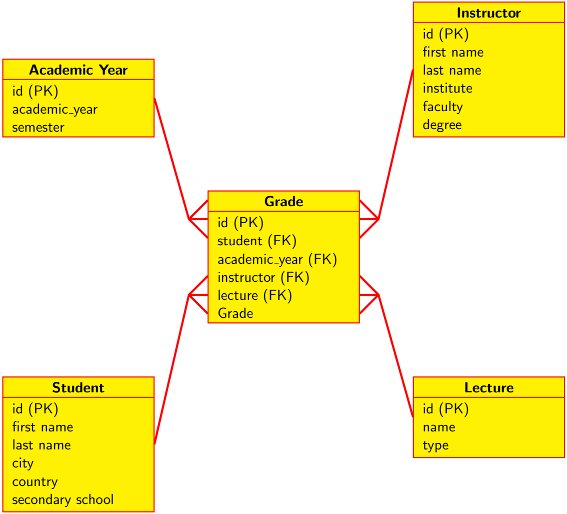
Star schema

- A single table in the middle connected to a number of dimension tables.



Star schema

- A single table in the middle connected to a number of dimension tables.



Star schema

- **Measures**, e.g. grades, price, quantity.

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.
  - Measures depend on a set of dimensions, e.g. student grade depends on student, course, instructor, faculty, academic year, etc.

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.
  - Measures depend on a set of dimensions, e.g. student grade depends on student, course, instructor, faculty, academic year, etc.
- **Fact table**

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.
  - Measures depend on a set of dimensions, e.g. student grade depends on student, course, instructor, faculty, academic year, etc.
- **Fact table**
  - Relates the dimensions to the measures.

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.
  - Measures depend on a set of dimensions, e.g. student grade depends on student, course, instructor, faculty, academic year, etc.
- **Fact table**
  - Relates the dimensions to the measures.
- **Dimension tables**

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.
  - Measures depend on a set of dimensions, e.g. student grade depends on student, course, instructor, faculty, academic year, etc.
- **Fact table**
  - Relates the dimensions to the measures.
- **Dimension tables**
  - Represent information about dimensions (student, academic year, etc.).

Star schema

- **Measures**, e.g. grades, price, quantity.
  - Measures should be aggregative.
  - Measures depend on a set of dimensions, e.g. student grade depends on student, course, instructor, faculty, academic year, etc.
- **Fact table**
  - Relates the dimensions to the measures.
- **Dimension tables**
  - Represent information about dimensions (student, academic year, etc.).
  - Each dimension has a set of descriptive attributes.

Fact table

- Each fact table contains measurements about a process of interest.

Fact table

- Each fact table contains measurements about a process of interest.
- Each fact row contains foreign keys to dimension tables and numerical measure columns.

Fact table

- Each fact table contains measurements about a process of interest.
- Each fact row contains foreign keys to dimension tables and numerical measure columns.
- Any new fact is added to the fact table.

Fact table

- Each fact table contains measurements about a process of interest.
- Each fact row contains foreign keys to dimension tables and numerical measure columns.
- Any new fact is added to the fact table.
- The aggregated fact columns are the matter of the analysis.

Dimension tables

- Each dimension table corresponds to a real-world object or concept, e.g. customer, product, region, employee, store, etc..

Dimension tables

- Each dimension table corresponds to a real-world object or concept, e.g. customer, product, region, employee, store, etc..
- Dimension tables contain many descriptive columns.

Dimension tables

- Each dimension table corresponds to a real-world object or concept, e.g. customer, product, region, employee, store, etc..
- Dimension tables contain many descriptive columns.
- Generally do not have too many rows (in comparison to the fact table).

Dimension tables

- Each dimension table corresponds to a real-world object or concept, e.g. customer, product, region, employee, store, etc..
- Dimension tables contain many descriptive columns.
- Generally do not have too many rows (in comparison to the fact table).
- Content is relatively static.

Dimension tables

- Each dimension table corresponds to a real-world object or concept, e.g. customer, product, region, employee, store, etc..
- Dimension tables contain many descriptive columns.
- Generally do not have too many rows (in comparison to the fact table).
- Content is relatively static.
- The attributes of dimension tables are used for filtering and grouping.

Dimension tables

- Each dimension table corresponds to a real-world object or concept, e.g. customer, product, region, employee, store, etc..
- Dimension tables contain many descriptive columns.
- Generally do not have too many rows (in comparison to the fact table).
- Content is relatively static.
- The attributes of dimension tables are used for filtering and grouping.
- Dimension tables describe facts stored in the fact table.

Fact table vs. Dimension tables

- Fact table:

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),
  - dynamic (growing over time).

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),
  - dynamic (growing over time).
- Dimension table

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),
  - dynamic (growing over time).
- Dimension table
  - wide,

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),
  - dynamic (growing over time).
- Dimension table
  - wide,
  - small (few rows),

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),
  - dynamic (growing over time).
- Dimension table
  - wide,
  - small (few rows),
  - descriptive (rows are described by descriptive attributes),

Facts contain numbers, dimensions contain labels

Fact table vs. Dimension tables

- Fact table:
  - narrow,
  - big (many rows),
  - numeric (rows are described by numerical measures),
  - dynamic (growing over time).
- Dimension table
  - wide,
  - small (few rows),
  - descriptive (rows are described by descriptive attributes),
  - static.

Facts contain numbers, dimensions contain labels

Denormalization

- Denormalization is the process of attempting to optimize the performance of a database by adding redundant data or by grouping data.

Denormalization

- Denormalization is the process of attempting to optimize the performance of a database by adding redundant data or by grouping data.
- Denormalization helps cover up the inefficiencies inherent in relational database software.

Denormalization

- Denormalization is the process of attempting to optimize the performance of a database by adding redundant data or by grouping data.
- Denormalization helps cover up the inefficiencies inherent in relational database software.
- **Normalize until it hurts, denormalize until it works** :)

Important aspects of dimensional modeling

- Four step procedure:

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,
- Surrogate keys,

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,
- Surrogate keys,
- Degenerate dimension,

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,
- Surrogate keys,
- Degenerate dimension,
- Role-playing dimensions,

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,
- Surrogate keys,
- Degenerate dimension,
- Role-playing dimensions,
- Slowly changing dimensions,

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,
- Surrogate keys,
- Degenerate dimension,
- Role-playing dimensions,
- Slowly changing dimensions,
- Mini dimension,

Important aspects of dimensional modeling

- Four step procedure:
  - Select the business process to model (e.g. sales).
  - Determine the grain of the business process (e.g. single transaction in a market identified by bar-code scanners at cash register).
  - Choose the dimensions describing the business process (e.g. localization, product, data, promotion, etc.).
  - Identify the numeric measures for the facts (e.g. price, quantity).
- Date and time dimension,
- Surrogate keys,
- Degenerate dimension,
- Role-playing dimensions,
- Slowly changing dimensions,
- Mini dimension,
- Factless fact tables.

Multidimensional data model

- Retail sales data:

Location:Vancouver				
Time (quarters)	Items			
	TV	Computer	Phone	Security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Multidimensional data model

- Similar information for other cities:

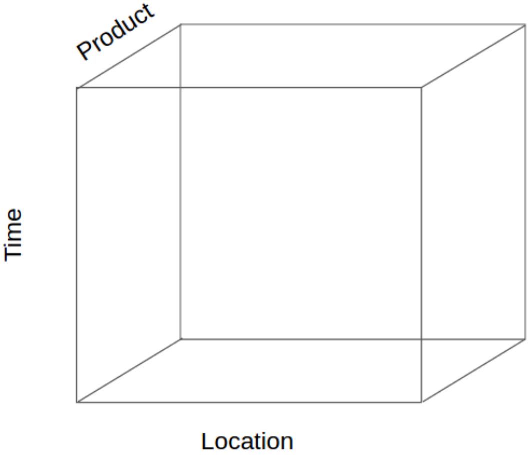
Location:Vancouver				
Time	Items			
(quarters)	TV	Computer	Phone	Security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Location:Toronto				
Time	Items			
(quarters)	TV	Computer	Phone	Security
Q1	1087	968	38	872
Q2	1130	1024	41	952
Q3	1034	1048	45	1002
Q4	1142	1091	52	984

Location:Chicago				
Time	Items			
(quarters)	TV	Computer	Phone	Security
Q1	854	882	89	623
Q2	943	890	64	698
Q3	1023	924	59	789
Q4	1129	992	63	870

Location:New York				
Time	Items			
(quarters)	TV	Computer	Phone	Security
Q1	818	746	43	591
Q2	894	769	52	682
Q3	940	795	58	728
Q4	978	864	59	784

Multidimensional cube



- More dimensions possible.

Different levels of aggregation

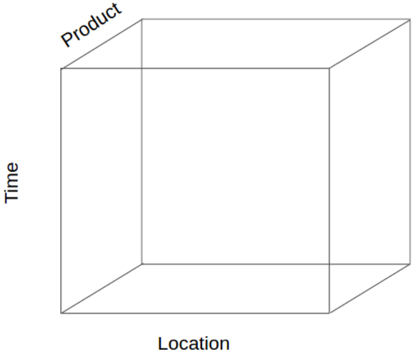
- Sales(time, product, *)

Time (quarters)	Items			
	TV	Computer	Phone	Security
Q1	3364	3421	184	2486
Q2	3647	3635	188	2817
Q3	3809	3790	186	3020
Q4	4176	3985	212	3218

- Sales(time, *, *); Sales(*, *, *)

Operators in multidimensional data model

- Roll up – summarize data along a dimension hierarchy.
- Drill down – go from higher level summary to lower level summary or detailed data.
- Slice and dice – corresponds to selection and projection.
- Pivot – reorient cube.
- Raking, Time functions, etc.



Exploring the cube

Time (quarters)	Items			
	TV	Computer	Phone	Security
Q1	3364	3421	184	2486
Q2	3647	3635	188	2817
Q3	3809	3790	186	3020
Q4	4176	3985	212	3218

$\Leftrightarrow$

Time		Items			
		TV	Computer	Phone	Security
Q1		3364	3421	184	2486
Q2		3647	3635	188	2817
Q3		3809	3790	186	3020
Q4	October	1172	960	105	1045
	November	1005	1340	45	987
	December	1999	1685	62	1186

Outline

Processing of massive datasets

Processing of massive datasets

- Physical data organization:

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access:

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.
- Summarization, materialization, and denormalization.

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.
- Summarization, materialization, and denormalization.
- Data compression.

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.
- Summarization, materialization, and denormalization.
- Data compression.
- Approximate query processing.

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.
- Summarization, materialization, and denormalization.
- Data compression.
- Approximate query processing.
- Probabilistic data structures and algorithms.

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.
- Summarization, materialization, and denormalization.
- Data compression.
- Approximate query processing.
- Probabilistic data structures and algorithms.
- Data schemas:

Processing of massive datasets

- Physical data organization: row-based, column-based, key-values stores, multi-dimensional arrays, etc.
- Partitioning and sharding (Map-Reduce, distributed databases).
- Data access: hashing and sorting ($\rightarrow$ tree-based indexing).
- Advanced data structures: multi-dimensional indexes, inverted lists, bitmaps, special-purpose indexes.
- Summarization, materialization, and denormalization.
- Data compression.
- Approximate query processing.
- Probabilistic data structures and algorithms.
- Data schemas: star schema, flexible schemas.

Outline

Summary

- Significant difference between operational and analytical systems.
- Different data models dedicated to particular applications.
- NoSQL = "Not only traditional relational DBMS."
- OLAP vs. OLTP.
- Star schema.
- Multidimensional data model.
- Processing of massive datasets.

Bibliography

- H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database Systems: The Complete Book. Second Edition*.
  Pearson Prentice Hall, 2009
- R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition*.
  John Wiley & Sons, 2013
- Nathan Marz and James Warren. *Big Data: Principles and best practices of scalable real-time data systems*.
  Manning Publications Co., 2015