Database Management Systems

Introduction to Databases

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning
Indian Statistical Institute, Kolkata

January, 2023

Let's play a game!!!

What is the maximum marks (so far) in Computing Lab?

Let's play a game!!!

What is the maximum marks (so far) in Computing Lab?

Think why someone was ahead of others ... probably because the data was

- kept at a right place (storage)
- updated last time properly (modification)
- examined with a fast strategy (analysis)

Let's play a game!!!

What is the maximum marks (so far) in Computing Lab?

Think why someone was ahead of others ... probably because the data was

- kept at a right place (storage)
- updated last time properly (modification)
- examined with a fast strategy (analysis)

As a whole, we can say that the data was organized (management) properly by the winner.

Introduction

DBMS deals with the management of data

Introduction

DBMS deals with the management of data

Management of data refers to
– *storing* data,
– *modifying* (add, edit, delete) data, and
– *analyzing* (extract data/information) data

Note: A database is a collection of data.

Think about the past

Before DBMS, the typical file-processing systems were supported by conventional operating systems. The system stored permanent records in various files, and it needed different application programs to extract records from, and add records to, the appropriate files.

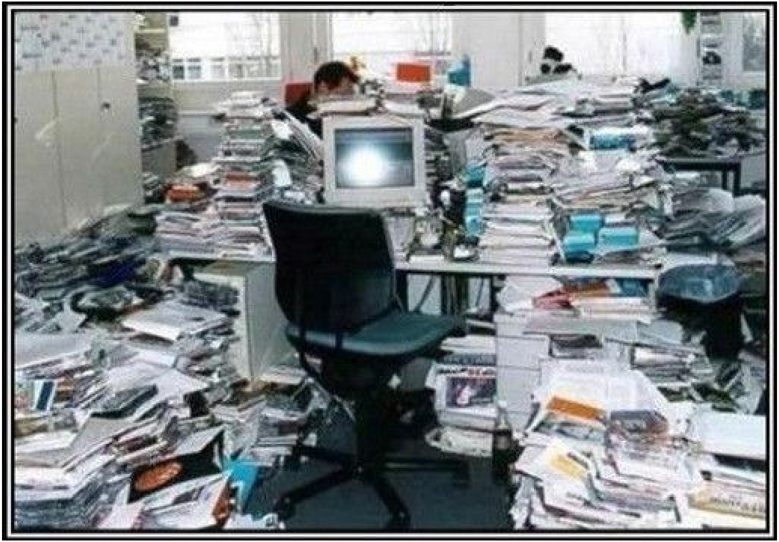
Think about the past

Before DBMS, the typical file-processing systems were supported by conventional operating systems. The system stored permanent records in various files, and it needed different application programs to extract records from, and add records to, the appropriate files.

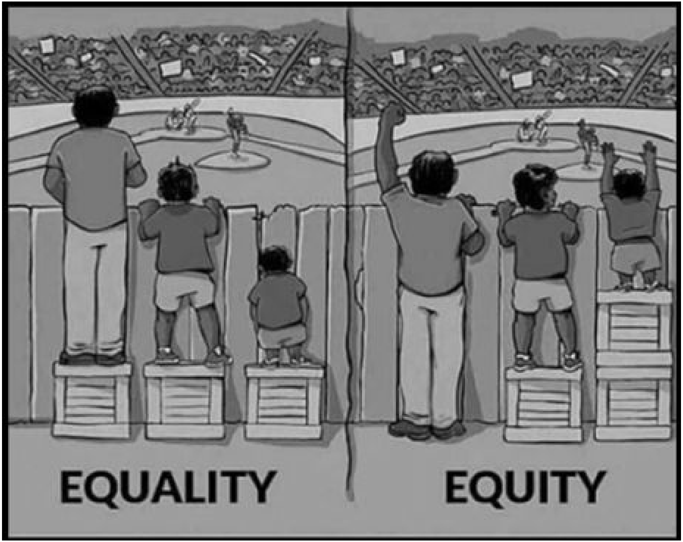
1. Data redundancy and inconsistency – *repeated copies*
2. Difficulty in accessing data – *time complexity*
3. Data isolation – *changes reflected for all*
4. Integrity problems – *accuracy and consistency*
5. Atomicity problems – *everything or nothing*
6. Concurrent-access anomalies – *simultaneous access*
7. Security problems – *privacy*

Data redundancy and inconsistency

Difficulty in accessing data



Data isolation

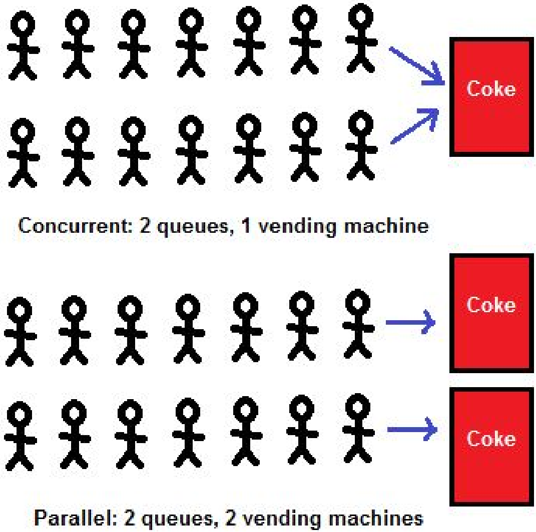


Integrity problems

"WE'RE ALL ABOUT INTEGRITY HERE. BY THE WAY, IF MY WIFE CALLS, TELL HER I'M NOT IN."

Atomicity problems

Concurrent-access anomalies



Security problems

History

"Data matures like wine, applications like fish" – Andy Todd.

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL
1980s: Object-oriented DBMS

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL
1980s: Object-oriented DBMS
1990s: Parallel and distributed DBMS

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL
1980s: Object-oriented DBMS
1990s: Parallel and distributed DBMS
Early 2000s: XML, XQuery

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL
1980s: Object-oriented DBMS
1990s: Parallel and distributed DBMS
Early 2000s: XML, XQuery
Late 2000s: Google BigTable, Yahoo PNuts

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL
1980s: Object-oriented DBMS
1990s: Parallel and distributed DBMS
Early 2000s: XML, XQuery
Late 2000s: Google BigTable, Yahoo PNuts
2010s: NoSQL

History

"Data matures like wine, applications like fish" – Andy Todd.

1950s: Storage on magnetic tapes
Early 1960s: Hierarchical database systems
Late 1960s: Network database systems
1970s: Relational DBMS
End of 1970s: SQL
1980s: Object-oriented DBMS
1990s: Parallel and distributed DBMS
Early 2000s: XML, XQuery
Late 2000s: Google BigTable, Yahoo PNuts
2010s: NoSQL
2020s: NewSQL

Data abstraction

External level

$\Updownarrow$

Logical level

$\Updownarrow$

Physical level

Data abstraction

External level

$\Updownarrow$

Logical level

$\Updownarrow$

Physical level

The collection of information stored in the database at a particular moment is called an *instance* of the database.

Data abstraction

External level
$\Updownarrow$
Logical level
$\Updownarrow$
Physical level

The collection of information stored in the database at a particular moment is called an *instance* of the database.

The overall design of the database is called the database *schema*.
– *Physical schema* reflects database design at the physical level
– *Logical schema* reflects database design at the logical level

Let us brainstorm!!!

Suppose we wish to create a public repository to keep songs in three different raw formats – the video only, the audio, and the lyrics. The purpose is to allow the users to download these three types of files as and when required. Each of the aforementioned triplet (video, audio, text) is also associated with some metadata like the singer, year, album/movie, lyricist, etc.

Conceptualize a physical design (schema) to store the necessary data files and metadata together.

Let us brainstorm!!!

Suppose we wish to create a public repository to keep songs in three different raw formats – the video only, the audio, and the lyrics. The purpose is to allow the users to download these three types of files as and when required. Each of the aforementioned triplet (video, audio, text) is also associated with some metadata like the singer, year, album/movie, lyricist, etc.

Conceptualize a physical design (schema) to store the necessary data files and metadata together.

Note: `Polyglot Persistence` is a concept that encourages employing multiple data storage technologies, chosen based on the way data is being used by an application or its component, while storing data.

Idea I

The concept: Use a hierarchical structure to organize the files and their metadata and a hierarchical structure to store the raw files.

```
                          header
                 /                       \        \
               video                     audio    text
        /        |          \              :        :
     type 1    type 2       ...
     singer    singer       ...
      year      year        ...
        :         :          :
```

Advantages: Quick access

Disadvantages: Impractical with respect to consistency; One way searching is only possible

Idea II

The concept: Use a networked structure to organize the files and their metadata and store the raw files.

year album/movie

\ /

singer – (video)

/ \

lyricist ...

Advantages: Easy access

Disadvantages: One way searching is only possible

Idea III

The concept: Use a table to store the metadata and a hierarchical structure to store the raw files.

Song	singer	year	album/movie	lyricist	...	path
...	...	...	...	...	...	./...

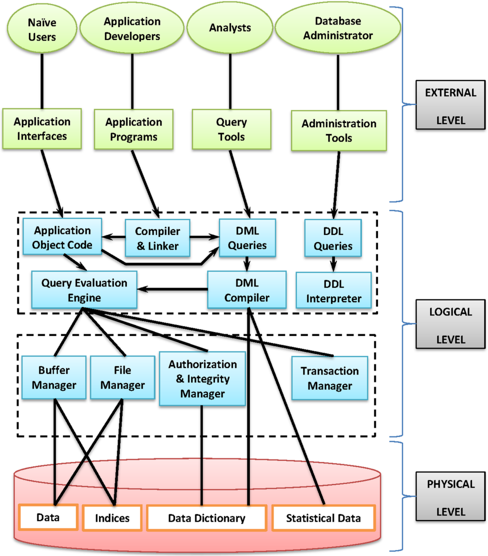
Advantages: Both way searching is possible

Disadvantages: Complex design that blends a relational and hierarchical schema

Languages

- **Data-definition language (DDL):** It specifies the database schema
- **Data-manipulation language (DML):** It expresses database queries and updates for the following tasks.
  1. The retrieval of information stored in the database
  2. The insertion of new information into the database
  3. The deletion of information from the database
  4. The modification of information stored in the database

DBMS System Components



Basics of Data Integration

Each local database management system may use a different data model. For instance, some may employ the relational model, whereas others may employ older data models, such as the network model or the hierarchical model.

Since the multi-database system is supposed to provide the illusion of a single, integrated database system, a common data model must be used.

A commonly used choice is the relational model, with SQL as the common query language. Indeed, there are several systems available today that allow SQL queries to a nonrelational database management system.

Schema integration

There should be the provision of a common conceptual schema. Each local system provides its own conceptual schema. The multi-database system must integrate these separate schema into one common schemata. Schema integration is a complicated task, mainly because of the semantic heterogeneity.

Challenges of data integration

Schema integration is not simply straightforward translation between data-definition languages.

1. The same attribute names may appear in different local databases but with different meanings.
2. The data types used in one system may not be supported by other systems.
3. Translation between data types may not be simple.
4. Even for identical data types, problems may arise from the physical representation of data.
   - One system may use ASCII, another EBCDIC,
   - Floating-point representations may differ;
   - Integers may be represented in big-endian or little-endian form.
   - At the semantic level, an integer value for length may be inches in one system and millimeters in another.

Aim of data integration

Users will be provided with homogeneous logical view of data physically distributed over heterogeneous data sources.

Purpose of data integration

The data integration helps to perform the following toward different applications.

- Statistical analysis
- Data mining
- Online analytical processing (OLAP)
- and so on

Approaches to data integration

There are three major approaches of data integration as listed below.

- Ad-hoc programming
- Data warehousing
- Virtual integration

Ad-hoc programming

Create custom solutions for each application.

Disadvantages:

- The customization is time consuming.
- The customization is not generic.

Data warehousing

Extract all the data into a single data source.

Disadvantages:

- Data has to be cleaned from different formats.
- All sorts of data are to be stored irrespective of their usefulness.
- Data needs to be periodically updated.

Virtual integration

Keep data in the local data sources and for every query over the mediated schema find the results (probably more than one) and combine them, if necessary. Wrappers are custom-built programs that transform data from the source native format to something acceptable to the mediator.

Disadvantages:

- Designing a single mediated schema is not easy.
- Translation of queries over the mediated schema to queries over the source schemas.
- Query optimization and execution are hard.
- Incomplete data sources cannot be handled.

Limitations

1. The developments largely depend on the size of the data
2. Design depends on applications
3. Management complexity
4. Vulnerability to system failure
5. Conversion
6. Increased costs

The concluding remark

The concepts we can acquire as advanced DBMS will soon become conventional

Resources

Books:

1. C. J. Date, An Introduction to Database Systems, Pearson Education, Inc., 8th Edition, 2006.
2. A. Silberschatz, H. F. Korth and S. Sudarshan, Database System Concepts, Tata McGraw-Hill, 6th Edition, 2011.
3. R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, Pearson Education, Inc., 4th Edition, 2004.
4. R. Ramakrishnan and J. Gehrke, Database Management Systems, McGraw-Hil, 3rd Edition, 2007.
5. H. Garcia-Molina, J. D. Ullman and J. Widom, Database Systems: The Complete Book, Pearson Education, Inc., 2nd Edition, 2009.
6. G. Harrison and S. Feuerstein, MySQL stored procedure programming. O'Reilly Media, Inc., 2006.

Resources

Books (contd...):

7. K. Loney, Oracle Database 11g - The Complete Reference, McGraw-Hill, Inc., 2008.
8. I. Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publications, 6th Edition, 2010.
9. G. Fritchey, SQL Server Query Performance Tuning, Apress, 4th Edition, 2011.
10. P. J. Sadalage and M. Fowler, NoSQL distilled: a brief guide to the emerging world of polyglot persistence, Pearson Education, Inc., 1st Edition, 2013.
11. C. J. Date and H. Darwen, Database Explorations: Essays on The Third Manifesto and Related Topics, Trafford Publishing, 2010.

Resources

Journals:

1. ACM Transactions on Database Systems.
2. The VLDB Journal.
3. SIGKDD Explorations.

Conferences:

1. ACM KDD.
2. ACM SIGMOD/PODS.
3. IEEE ICDE.
4. IEEE ICDM.
5. VLDB.

Resources

Similar courses:

1. MIT: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-830-database-systems-fall-2010.
2. Stanford: http://web.stanford.edu/class/cs245.
3. Harvard: http://daslab.seas.harvard.edu/classes/cs165
4. Princeton: http://www.cs.princeton.edu/courses/archive/spr96/cs425.

Advanced courses:

1. Cornell: http://www.cs.cornell.edu/courses/cs632/2001sp.
2. CMU: https://15721.courses.cs.cmu.edu/spring2019.

Webpage: https://www.isical.ac.in/~malaybhattacharyya/Courses/DBMS/Spring2023