Fundamentals of Database Systems
[Introduction to Databases]

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

August, 2019

What is a database?

A database is a collection of data

What is a database?

A database is a collection of data

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

Note: Database Management Systems (DBMS) deals with the management 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
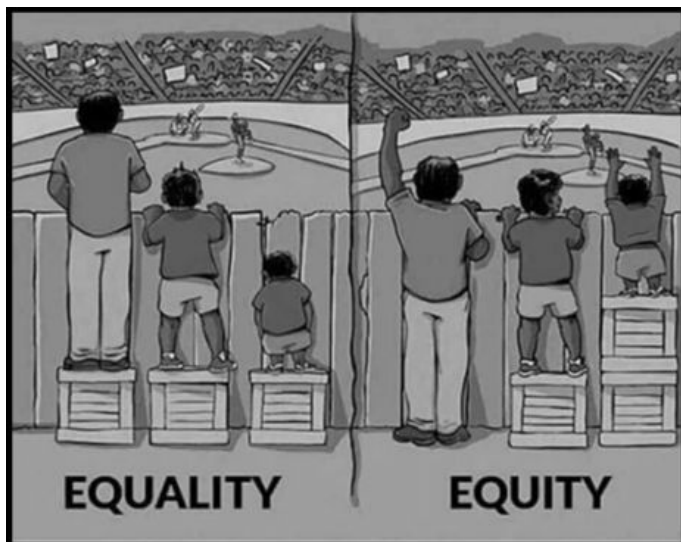
Data redundancy and inconsistency

Difficulty in accessing data
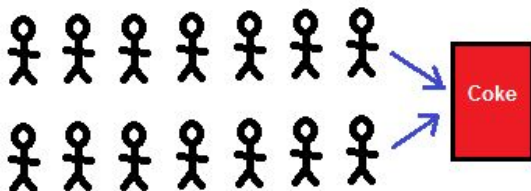
Data isolation

Integrity problems

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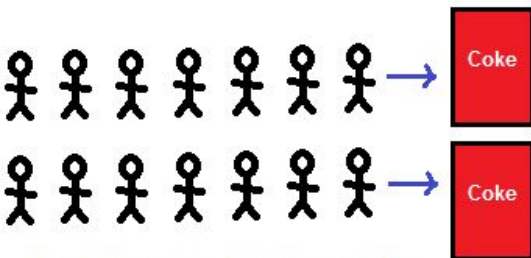
Atomicity problems

Concurrent-access anomalies

Concurrent: 2 queues, 1 vending machine

Parallel: 2 queues, 2 vending machines

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
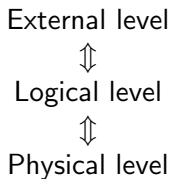
Data Abstraction

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
⇕
Logical level
⇕
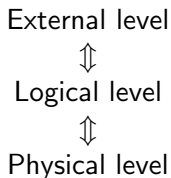Physical level

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
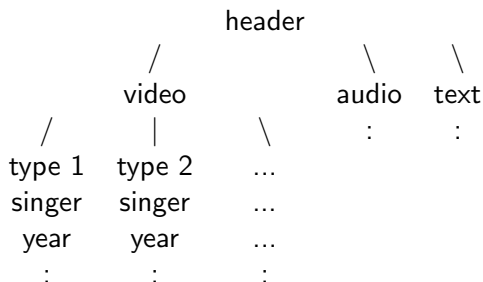– *Logical schema* reflects database design at the logical level

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.

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    ...
      :       :       :
```
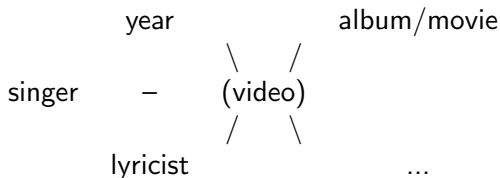
Advantages: Quick access

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
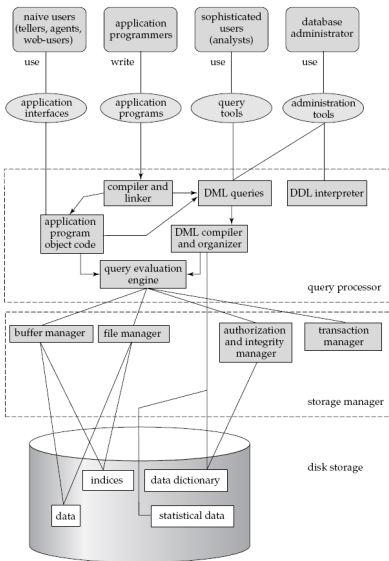
Limitations

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