# Database Management Systems (CSE-251)


Presented by

**Md. Atiqul Islam Rizvi**

Assistant Professor, Dept. of CSE, CUET

# Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model  (XML)
- Other older models:
  - Network model
  - Hierarchical model


©Silberschatz, Korth and Sudarshan

# Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model

Columns

Rows

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 22222 | Einstein | Physics | 95000 |
| 12121 | Wu | Finance | 90000 |
| 32343 | El Said | History | 60000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 98345 | Kim | Elec. Eng. | 80000 |
| 76766 | Crick | Biology | 72000 |
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 58583 | Califieri | History | 62000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 15151 | Mozart | Music | 40000 |
| 33456 | Gold | Physics | 87000 |
| 76543 | Singh | Finance | 80000 |

(a) The *instructor* table

# A Sample Relational Database

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 22222 | Einstein | Physics | 95000 |
| 12121 | Wu | Finance | 90000 |
| 32343 | El Said | History | 60000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 98345 | Kim | Elec. Eng. | 80000 |
| 76766 | Crick | Biology | 72000 |
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 58583 | Califieri | History | 62000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 15151 | Mozart | Music | 40000 |
| 33456 | Gold | Physics | 87000 |
| 76543 | Singh | Finance | 80000 |

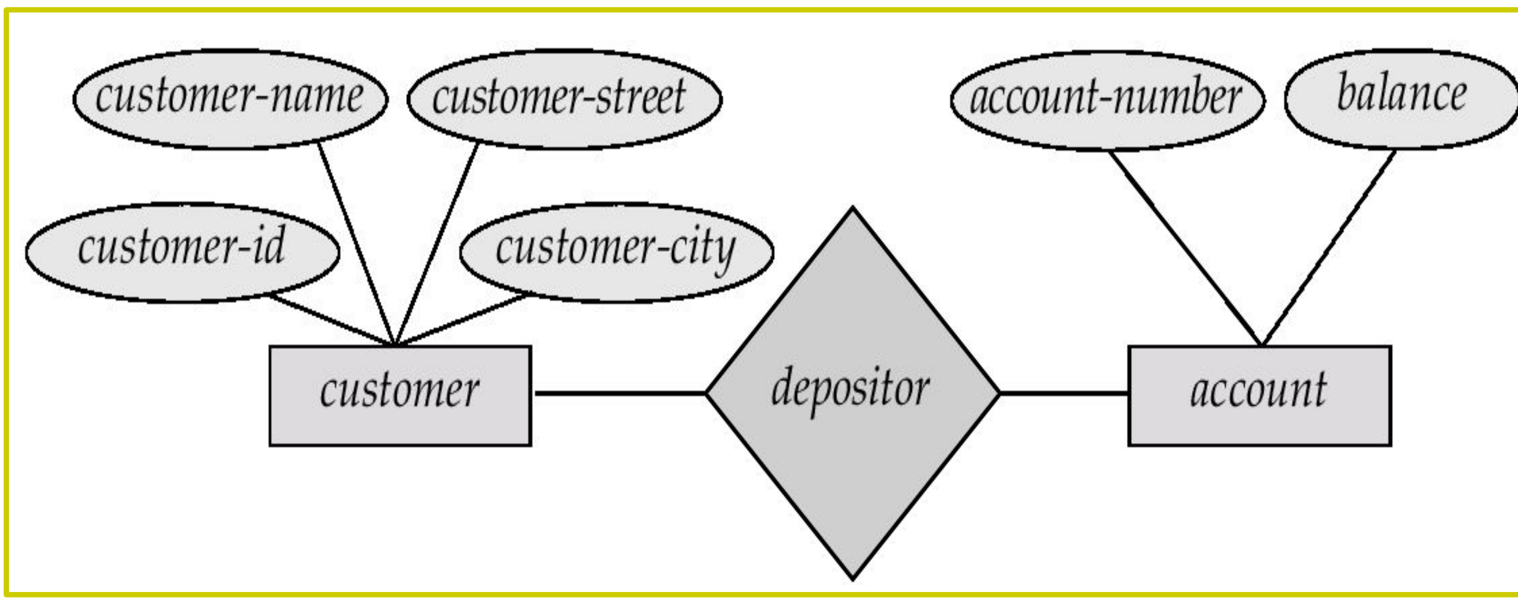
(a) The *instructor* table

| *dept_name* | *building* | *budget* |
|---|---|---|
| Comp. Sci. | Taylor | 100000 |
| Biology | Watson | 90000 |
| Elec. Eng. | Taylor | 85000 |
| Music | Packard | 80000 |
| Finance | Painter | 120000 |
| History | Painter | 50000 |
| Physics | Watson | 70000 |

(b) The *department* table

# Entity-Relationship Model

The entity–relationship (E-R) model is a high-level data model. It is based on a perception of a real world that consists of a collection of basic **objects**, called **entities,** and of **relationships** among these **objects**.
Example of schema in the entity-relationship model





©Silberschatz, Korth and Sudarshan

# Levels of Abstraction

- Hide the complexity of data structures to represent data in the database from users through several levels of data abstraction.

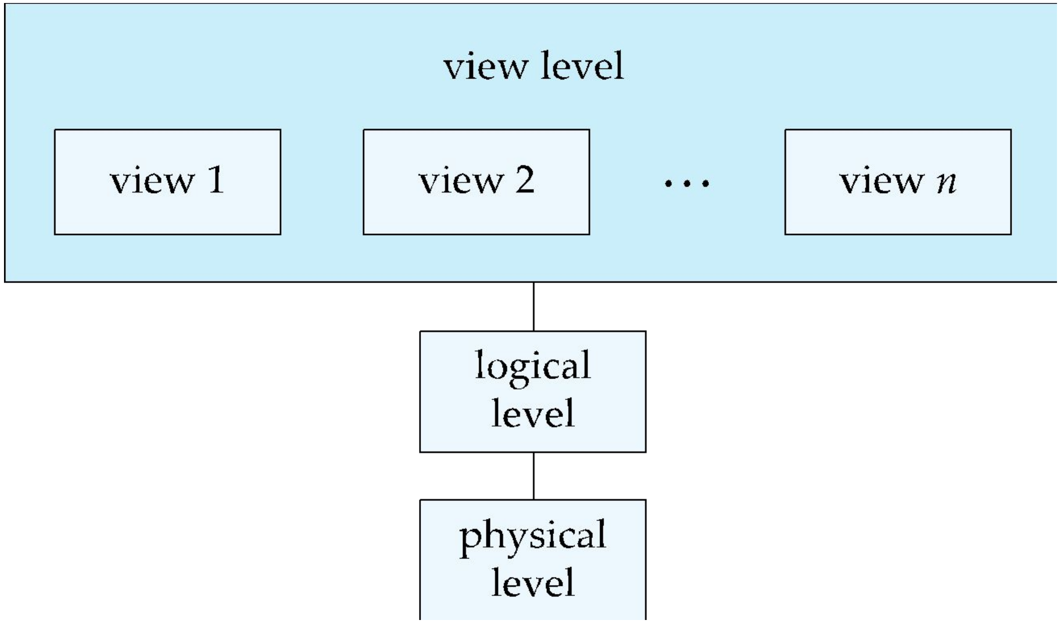
  - **Physical level**: describes how a record (e.g., instructor) is stored.

  - **Logical level**: describes what data stored in database, and what relationships exist among the data.

  - **View level**: application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

# View of Data

An architecture for a database system



# Instances and Schemas

- Similar to types and variables in programming languages
- **Instance** – the actual content of the database at a particular point in time
  - Analogous to the value of a variable at a certain point
- **Schema –** overall design of the database. Analogous to variable declaration with type definitions.
  - **Logical Schema** – the overall logical structure of the database. Includes descriptions of DB structure and constraints that should hold.
    - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
  - **Physical schema** – the overall physical structure of the database.
- Schema changes very infrequently, whereas instance is changed every time the DB is updated


©Silberschatz, Korth and Sudarshan

# Data Independence

- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema.
  - Applications depend on the logical schema

- **Logical Data Independence** – the ability to modify the logical schema without causing application programs to be rewritten.

- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are unchanged.
- In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

# Traditional File-Processing Systems

| <u>Advantages</u> | <u>Disadvantages</u> |
|---|---|
| Simpler to use | Typically does not support multi-user access |
| Less expensive. | Limited to smaller databases |
| Fits the needs of many small businesses and home users | Limited functionality (i.e. no support for complicated transactions, recovery, etc.) |
| Popular FMS's are packaged along with the operating systems of personal computers (i.e. Microsoft Card file and Microsoft Works) | Decentralization of data |
| Good for database solutions for hand held devices such as Palm Pilot | Redundancy and Integrity issues |

# Database Management Systems

| **Advantages** | **Disadvantages** |
|---|---|
| Greater flexibility | Difficult to learn |
| Good for larger databases | Packaged separately from the operating system (i.e. Oracle, Microsoft Access) |
| Greater processing power | Slower processing speeds |
| Fits the needs of many medium to large-sized organizations | Requires skilled administrators |
| Storage for all relevant data | Expensive |
| Provides user views relevant to tasks performed | |
| Ensures data integrity by managing transactions | |
| Supports simultaneous access | |
| Enforces design criteria in relation to data format and structure | |
| Provides backup and recovery controls | |
| Advanced security | |

# Purpose of Database Systems

- Data independence and efficient access.
- Data integrity and security.
- Uniform data administration.
- Concurrent access, recovery from crashes.
- Replication control
- Reduced application development time.
- Increase of dataset in diversity and volume.
- DBMS encompasses several areas of CS such as OS, languages, AI, app development, etc.
- Provide users with an abstract view of the data.


©Silberschatz, Korth and Sudarshan