Information Management Systems (CSET201L)

(Introduction to DBMS –Part 2)

Data Abstraction

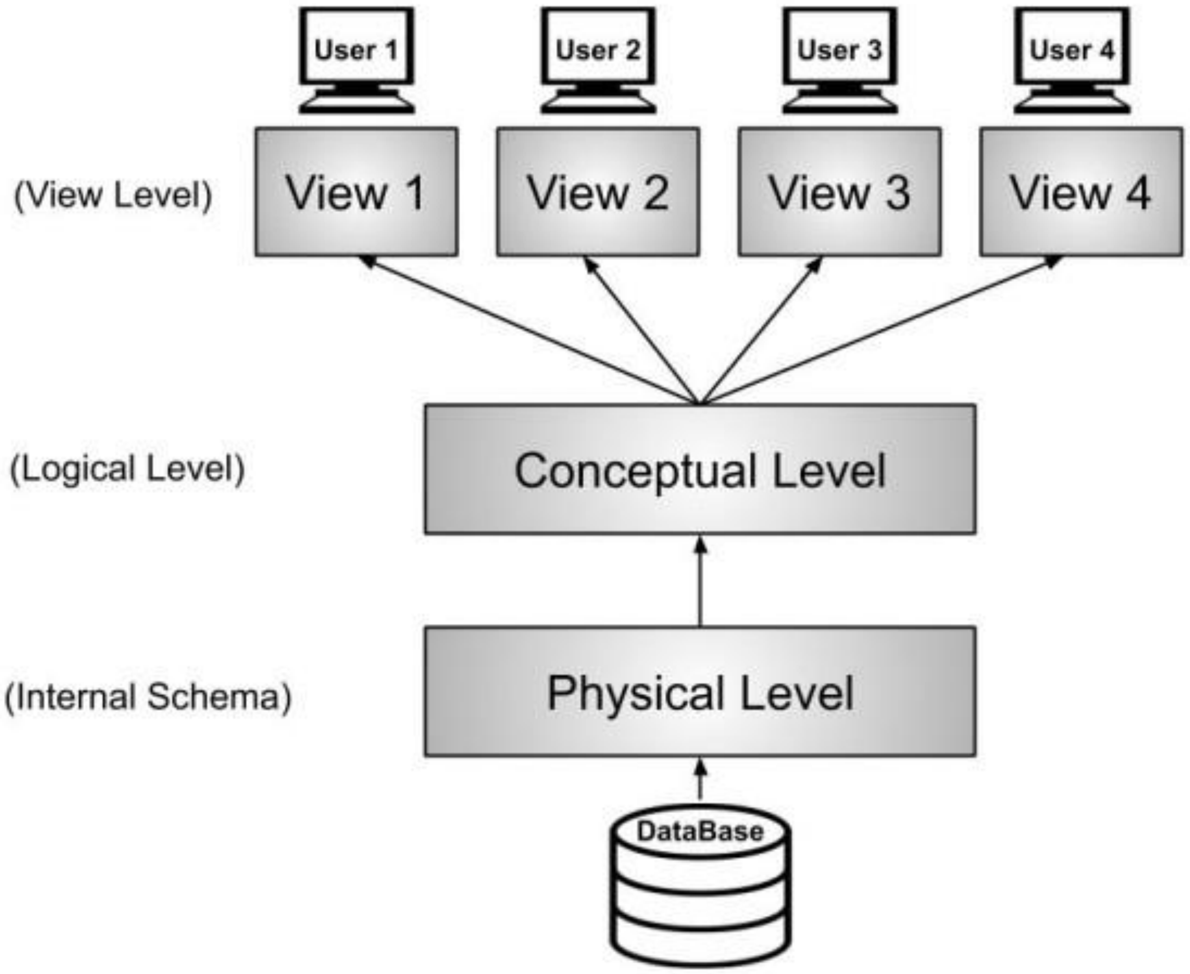
Data Abstraction refers to the process of **hiding irrelevant details** from the user.

Database systems are made-up of **complex data structures**. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

Hiding certain details of how the data are stored.

There are mainly **three levels of data abstraction** and we divide it into three levels in order **to achieve Data Independence.**

Level of Abstraction



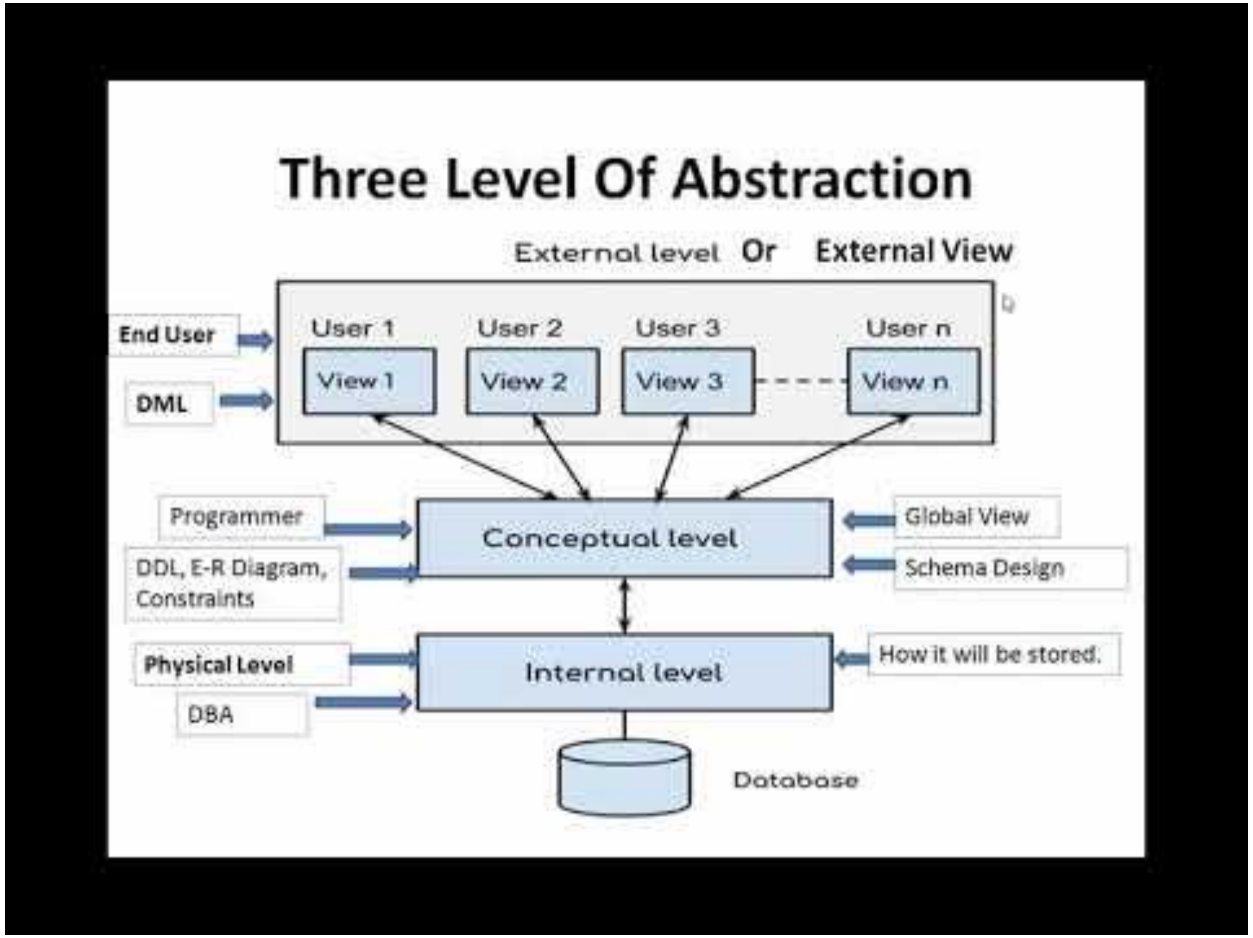
Levels of Data Abstraction

Data Levels and their Roles (Data Abstraction)

Physical level– This is the lowest level of data abstraction. It describes **how data is actually stored in database**. You can get the complex data structure details at this level. Implementation details such as indexing methods like B+ trees or hashing and access methods such as sequential or random access used are described in this level.

Conceptual/logical level At this level, we can see **what data is stored** in the database without knowing the implementation details such as the data structures and tree implementations. This level also tells us about the relations between the different fields and database tables.

3. View level – This is the highest level from the three levels of data abstraction. In this, only the part of the data which are relevant to the users is accessible. **What part of the data is seen** by a specific application. This level tells the application about how the data should be shown to the user.



Three Level Of Abstraction

Example of Data abstraction

Type of Schema	Implementation
External Schema	View 1: Course info(cid:int,cname:string) View 2: studeninfo(id:int. name:string)
Conceptual Shema	Students(id: int, name: string, login: string, age: integer) Courses(id: int, cname.string, credits:integer) Enrolled(id: int, grade:string)
Physical Schema	• Relations stored as unordered files. • Index on the first column of Students.

Schema and Instance

- The data which is stored in the database at a particular moment of time is called an instance of the database.
- The overall design of a database is called schema. A schema contains schema objects like table, views, stored procedure, etc.
- Similar to types and variables in programming languages
- Schema
  - **Logical Schema**-the overall logical structure of the database
  - Analogous to type information of a variable in a program
  - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
  - **Customer Schema**

Name	Customer ID	Account #	Aadhar ID	Mobile #

  - **Account Schema**

Account #	Account type	Interest rate	Min Bal.	Balance

  - **Physical Schema:** the overall physical structure of database.

Schema and Instance

- Instance
  - The actual content of the database at a particular point in time
  - Analogous to values of a variable

- **Customer Schema**

Name	Customer ID	Account #	Aadhar ID	Mobile #
Pawan	1234	943234	223454321654	9934567378
Rahul	4321	343242	342323542342	9823234522

- **Account Schema**

Account #	Account type	Interest rate	Min Bal.	Balance
943234	Savings	4%	5000	6543
343242	Current	0.0%	0	234455

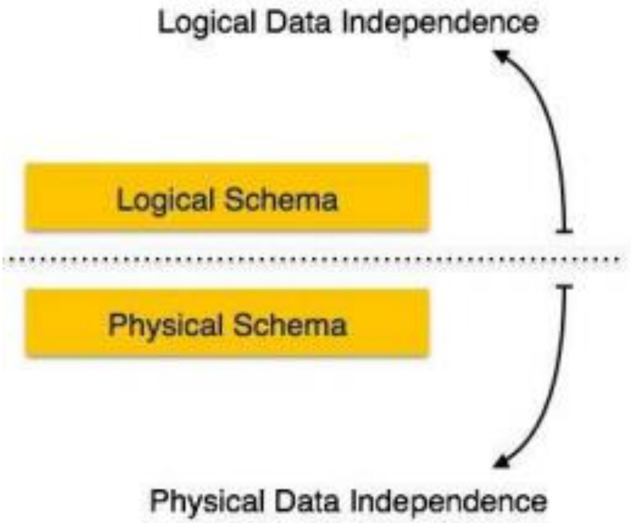
Data Independence

- Data Independence is defined as a property of DBMS that helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level. Data independence helps you to keep data separated from all programs that make use of it.

- A database system normally contains a lot of data in addition to users’ data. For example, it stores data about data, known as metadata, to locate and retrieve data easily . It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.

Physical and Logical Data Independence

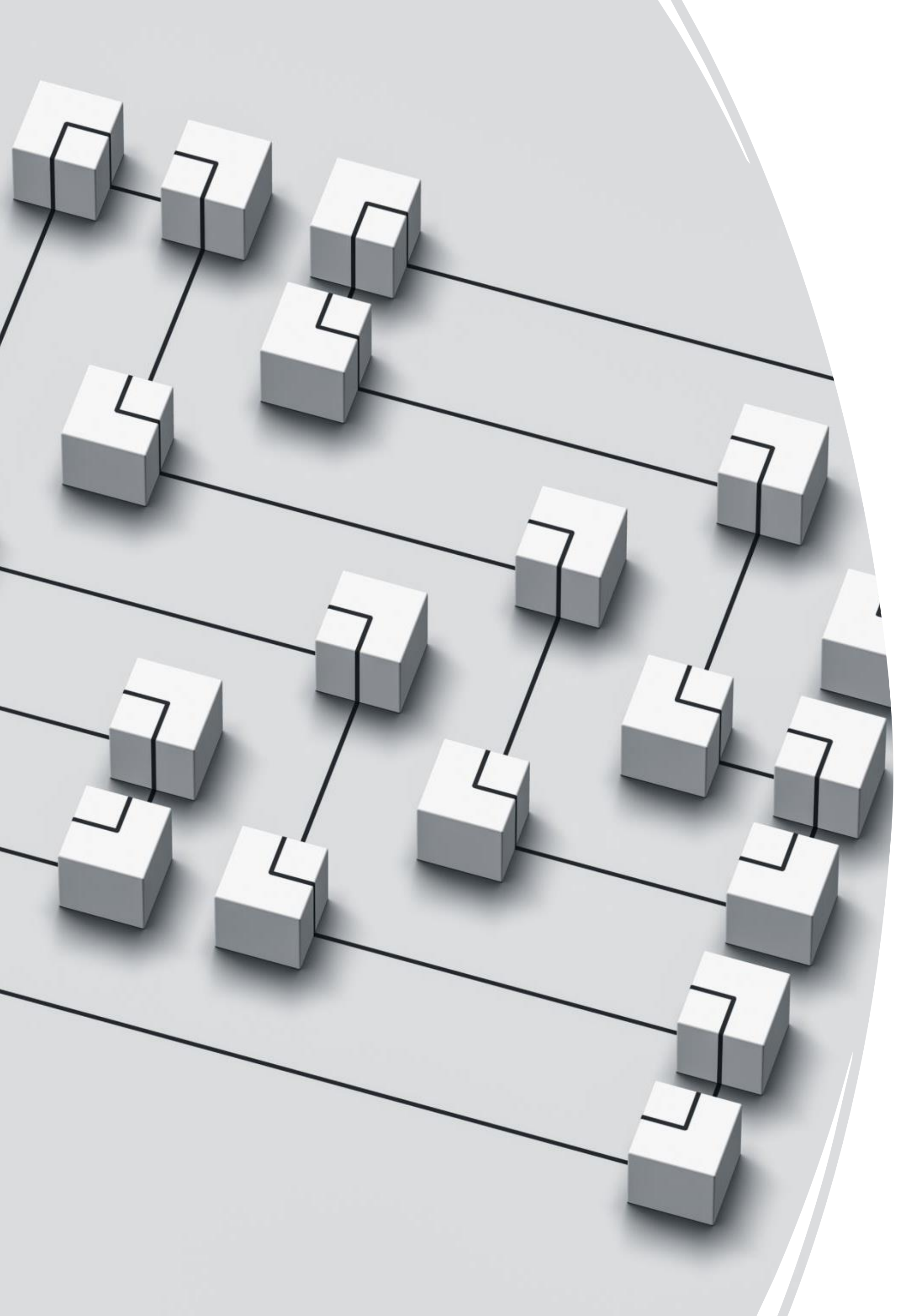
- **Physical data independence** It is defined as the ability to make changes in the structure of the lowest level of the Database Management System (DBMS) without affecting the higher-level schemas. Hence, modification in the Physical level should not result in any changes in the Logical or View levels.
- **Examples of changes under Physical Data Independence**
- Using a new storage device like Hard Drive or Magnetic Tapes
- Modifying the file organization technique in the Database
- Switching to different data structures.
- Changing the access method.
- Modifying indexes.
- Changes to compression techniques or hashing algorithms.
- Change of Location of Database from say C drive to D Drive



Physical and Logical Data Independence

- **Logical Data Independence** is the ability to change the conceptual schema without changing
  - External views
  - External API or programs
- Any change made will be absorbed by the mapping between external and conceptual levels.

- **Examples of changes under Logical Data Independence**
- Due to Logical independence, any of the below change will not affect the external layer.
  - Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
  - Merging two records into one
  - Breaking an existing record into two or more records



Data Models

- A collection of tools for describing the structure of data base,
    - Data
    - Data relationship
    - Data constraints
- Relational model (**we focus in this course**)
- Entity relationship data model(mainly for database design)
- Object based data model (object oriented and object relational)
- Semi- structure data model (XML)
- Other older models
    - Network model
    - Hierarchical model

Relational model

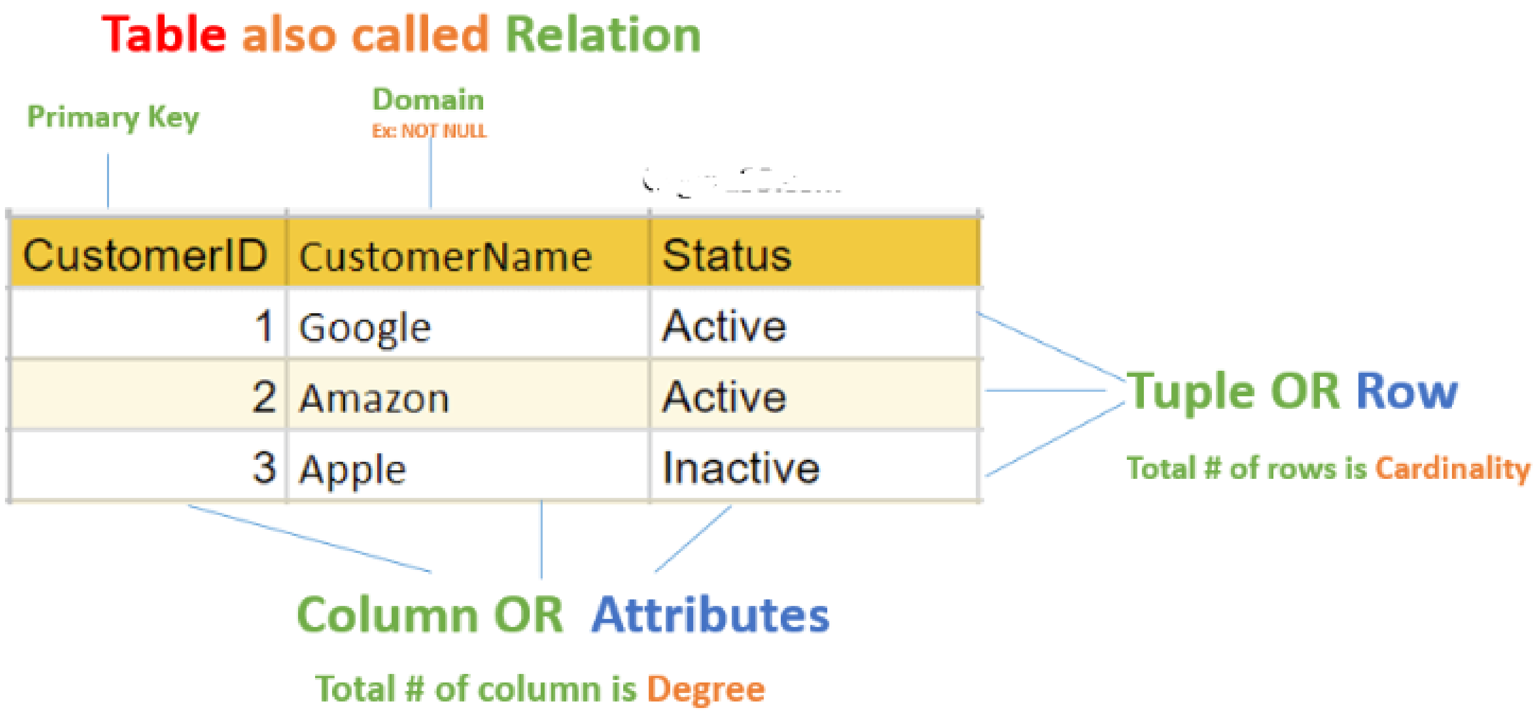


Table also called Relation

Primary Key

Domain

Ex: NOT NULL

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row

Total # of rows is Cardinality

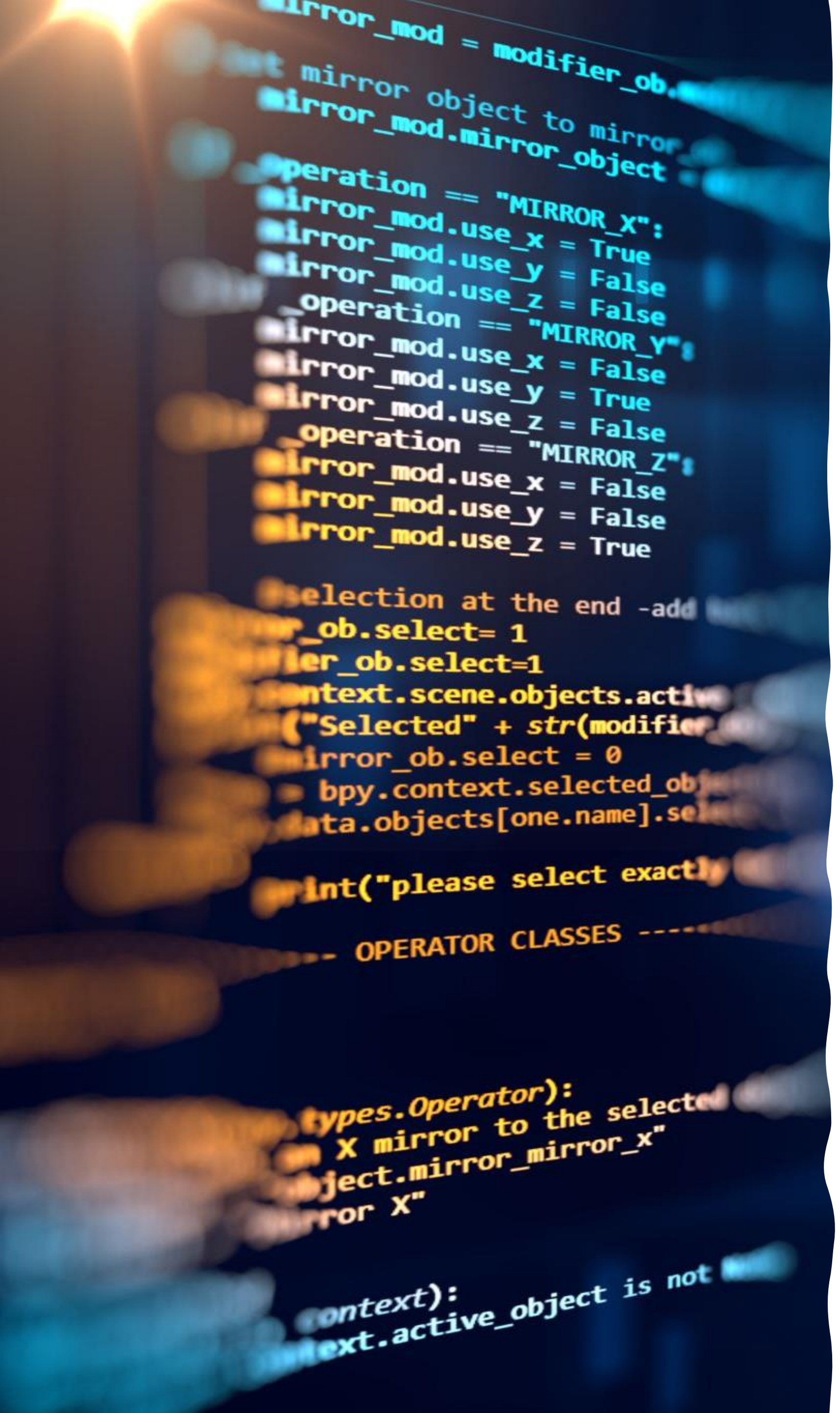
Column OR Attributes

Total # of column is Degree

DBMS Languages

DBMS system provides a Data Definition Language to specify the database schema and a data-manipulation language to express database queries and updates.

In Practice, the DDL and DML are not to separate languages; instead they simply form parts of a single database language, such as the widely used SQL language.



Data Definition Language

Specification notation for defining the database schema

- E.g.

  create table account (account-number char(10),balance integer)

- DDL compiler generates a set of tables stored in a data dictionary
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data storage and definition language
    - language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

Data Manipulation Language:

- **Language for accessing and manipulating the data organized by the appropriate data model**

- **Two classes of languages**

  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data

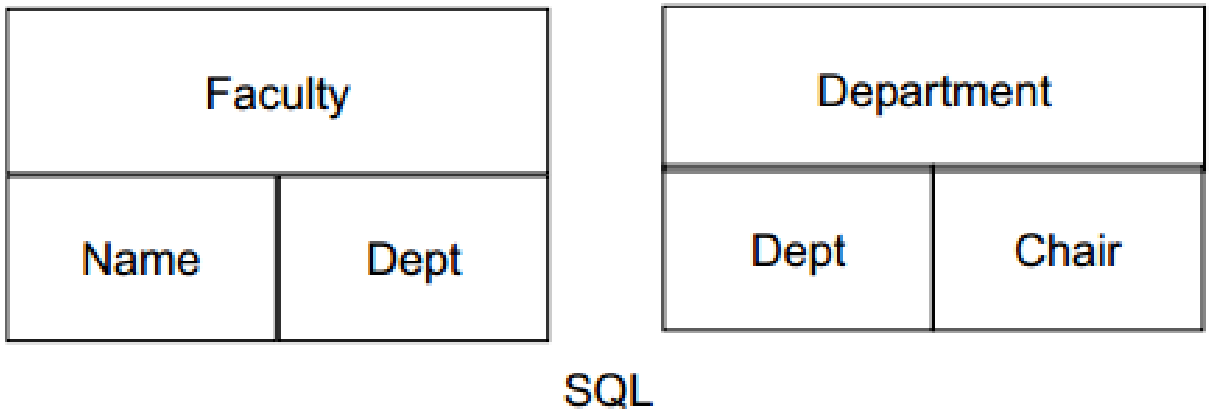
- **SQL is the most widely used query language**

Database Languages

- SELECT Chair
- FROM Faculty, Department
- WHERE Faculty.name = “Ken Noname”
- AND Faculty.Dept = Department.Dept

- Data definition language (DDL) ~ like type definitions in C or C++

- Data Manipulation Language (DML)
- Query (SELECT)
- UPDATE < relation name >
- SET <attribute> = < new-value>
- WHERE <condition>



Faculty	
Name	Dept

Department	
Dept	Chair

SQL

Thank you