M.Tech Geoinformatics

Database Management Systems

Introduction

Prakash. K
Guest Faculty
Department of Geography
Bharathidasan University, Tiruchirappalli.

Terminologies

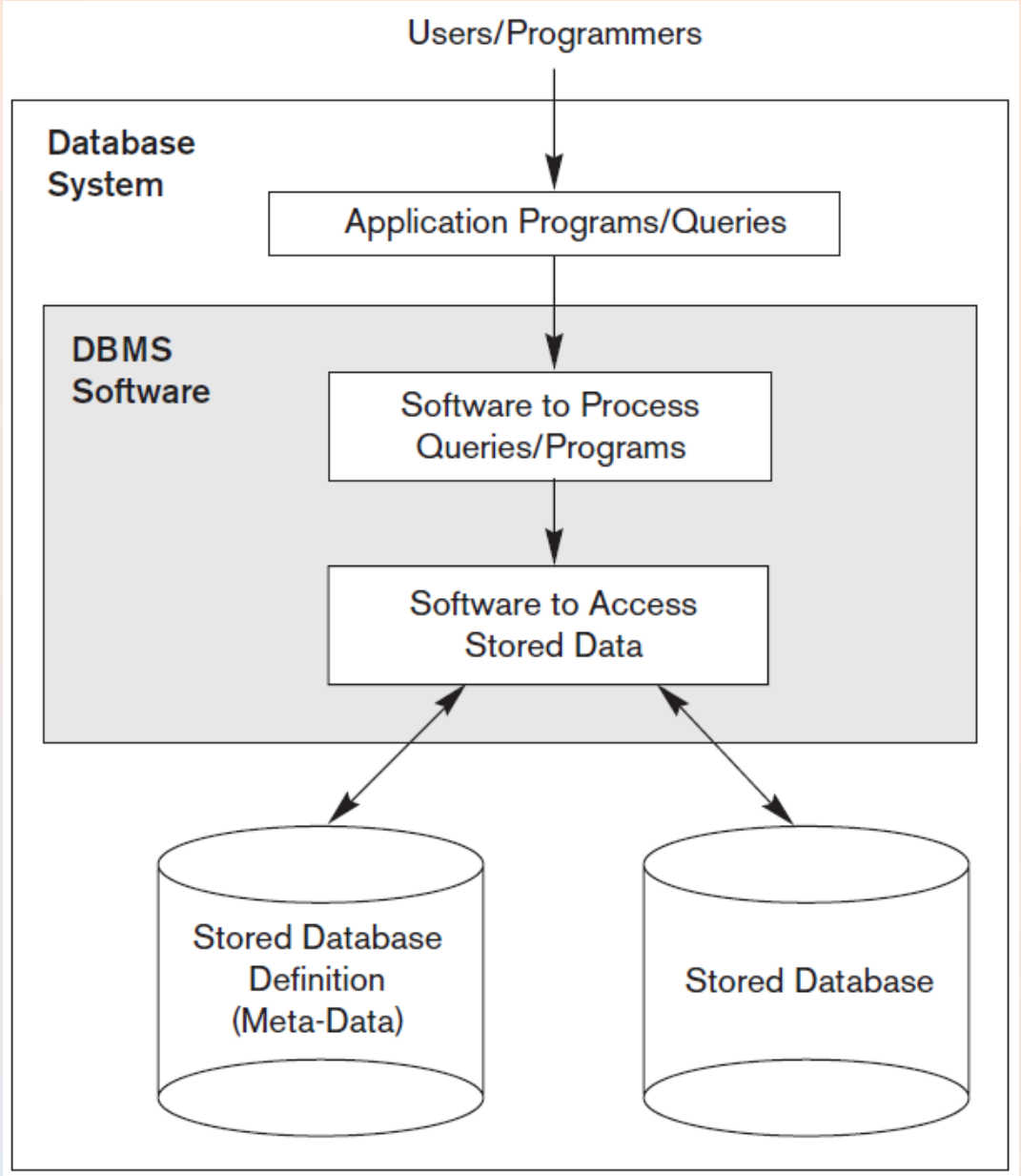
- Data: Known facts that can be recorded and have implicit meaning· Database is a collection of related data.
- Information: When the data is processed, organized, and presented in a structured way then it is called information.
- Database System: Composed of 5 major parts: Hardware, Software (DBMS), People, Procedures and Data.
- Database Management System (DBMS): Collection of components that support data acquisition, dissemination, storage, maintenance, retrieval, and formatting

Data in a database can be added, deleted, changed, sorted or searched all using a DBMS.

Example usage of Database System:

- Membership and subscription mailing lists
- Accounting and bookkeeping information
- The data obtained from scientific research

Simplified database system



Characteristics of DBMS

- Real world entity
- Self explaining nature
- Atomicity of operations (Transactions)
- Concurrent access without anomalies
- Stores any kind of structured data
- Integrity
- Ease of access (The DBMS Queries)
- SQL and No-SQL Databases
- Security

Characteristics of DBMS

Real World Entity

- One of the most important and easy-to-understand characteristics of DBMS is that it is realistic.
- For example, we can have a Database Management System for a School or a big MNC, and the data is stored in the form of real-world entities. Any student that is stored in a Student database is like a real-world student (object/entity) and has properties

Self Explaining Nature

- Metadata is the data about data. For example: In a DBMS for a particular School, the total Number of rows in the database and what is the name of each column of the database table, and all such information about the data is metadata.

Atomicity of Operations (Transactions)

- Atomicity means that either the operation should not be performed or it should be performed in its entirety i.e. it must be either 0% or 100% completed.
- DBMS provides us with atomic operations (i.e. any operation which is either complete 100% or not complete at all). This is very useful and important. You can understand the importance of atomic operations with the help of the example given below.

Concurrent Access without Anomalies

- Multiple users can access the database at the same time without any anomalies (problems: like 2 people trying to access the same data might restrict one from accessing it).

Stores Any Kind of Structured Data

- A database can store any kind of structured data.

Integrity

- Integrity means that the data which comes into the Database should be correct as well as consistent. Let us see an example to understand the meaning of correct and consistent data with respect to DBMS.

Ease of Access (The DBMS Queries)

- DBMS provides ease of access to the data inside the database. We can run a search query to find any data and the process is way faster than manual searching and is more reliable.
- The CRUD (Create, Read, Update & Delete) operations are very easy to perform with a database because of the DBMS queries.

SQL and No-SQL Databases

- There are 2 types of databases (not DBMS): SQL and No-SQL.
- The SQL databases store the data in the form of Tables i.e. rows and columns. The No-SQL databases can store data in any other form than a table. For instance: the very popular MongoDB stores the data in the form of JSON (JavaScript Object Notation).

Security

- The database should be accessible to the users in a limited way.
- The access to make changes to a database by the user should be limited and the users must not be given complete access to the entire database.
- Unauthorized users should not be allowed to access the database.

Types of DBMS

Server DBMS

- Oracle
- SQL Server
- DB2
- MySQL
- PostgreSQL

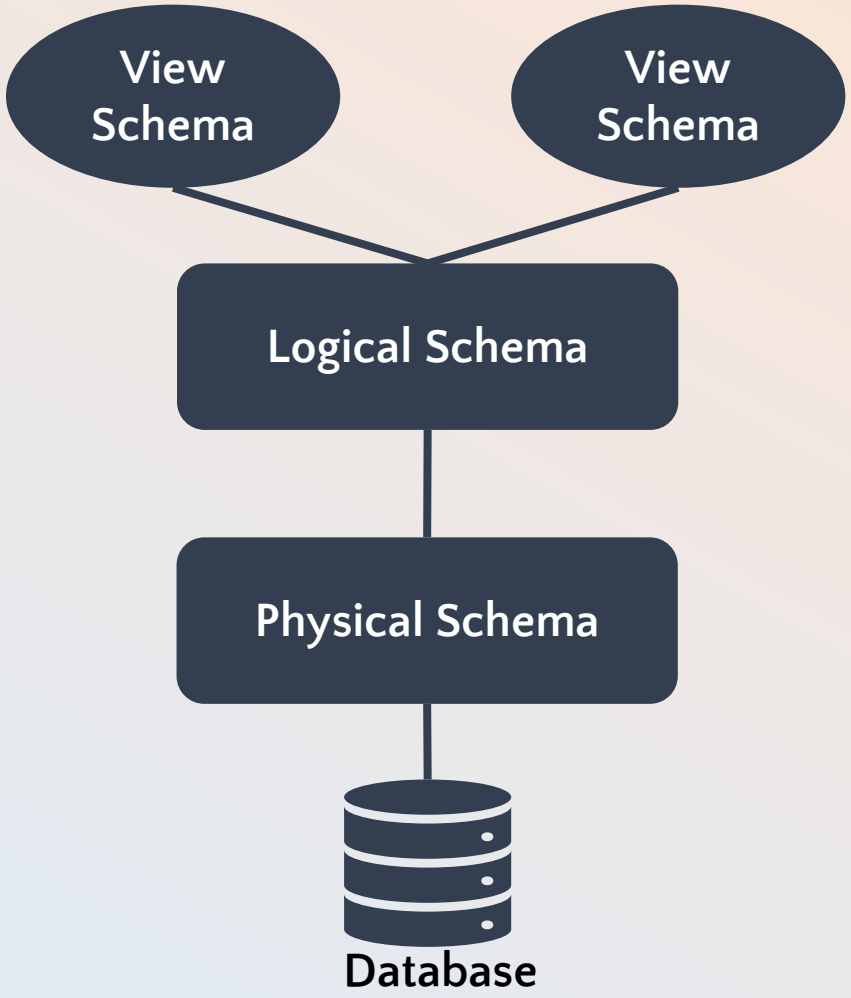
Desktop DBMS

- Microsoft Access
- FoxPro
- File Maker Pro

Database Schema

- A database schema is a logical representation of data that shows how the data in a database should be stored logically. It shows how the data is organized and the relationship between the tables.
- Database schema contains table, field, views and relation between different keys like primary key, foreign key.
- Data are stored in the form of files which is unstructured in nature which makes accessing the data difficult. Thus to resolve the issue the data are organized in structured way with the help of database schema.
- Database schema provides the organization of data and the relationship between the stored data.
- Database schema defines a set of guidelines that control the database along with that it provides information about the way of accessing and modifying the data.

Difference between Logical and Physical Database Schema



Physical Schema	Logical Schema
Physical schema describes the way of storage of data in the disk.	Logical schema provides the conceptual view that defines the relationship between the data entities.
Having Low level of abstraction.	Having a high level of abstraction.
The design of database is independent to any database management system.	The design of a database must work with a specific database management system or hardware platform.
Changes in Physical schema effects the logical schema	Any changes made in logical schema have minimal effect in the physical schema
Physical schema does not include attributes.	Logical schema includes attributes.
Physical schema contains the attributes and their data types.	Logical schema does not contain any attributes or data types.

Physical/Internal Schema

- A Physical schema defines, how the data are stored physically in the storage systems in the form of files & indices. This is the actual code or syntax needed to create the structure of a database, we can say that when we design a database at a physical level, it's called physical schema.
- The Database administrator chooses where and how to store the data in different blocks of storage.

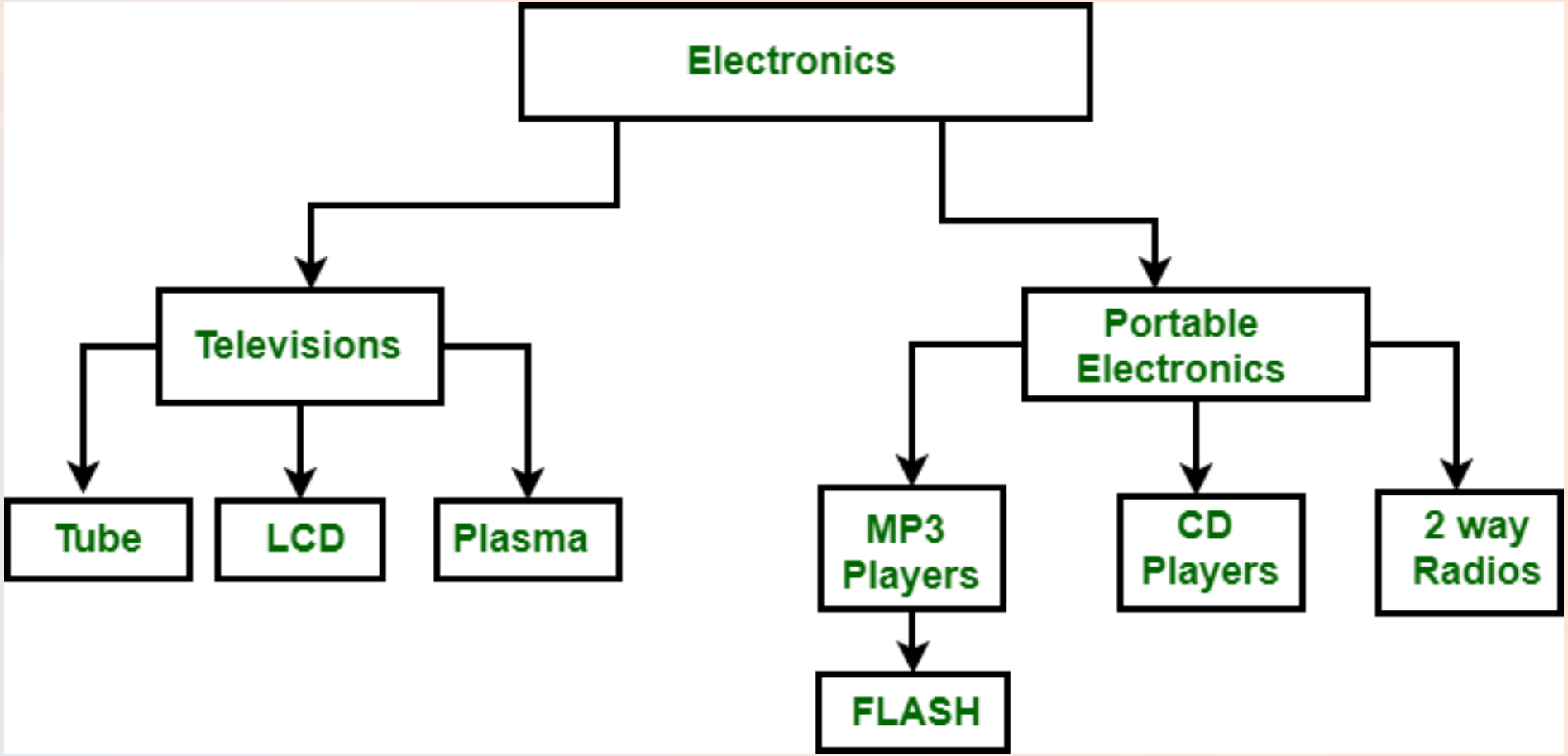
Logical/Conceptual Schema

- A logical database schema defines all the logical constraints that need to be applied to the stored data, and also describes tables, views, entity relationships, and integrity constraints.
- The Logical schema defines how the data is stored & how the attributes of a table are connected.
- Using ER modelling the relationship between the components of the data is maintained.
- In logical schema, different integrity constraints are defined to maintain the quality of insertion and update the data.

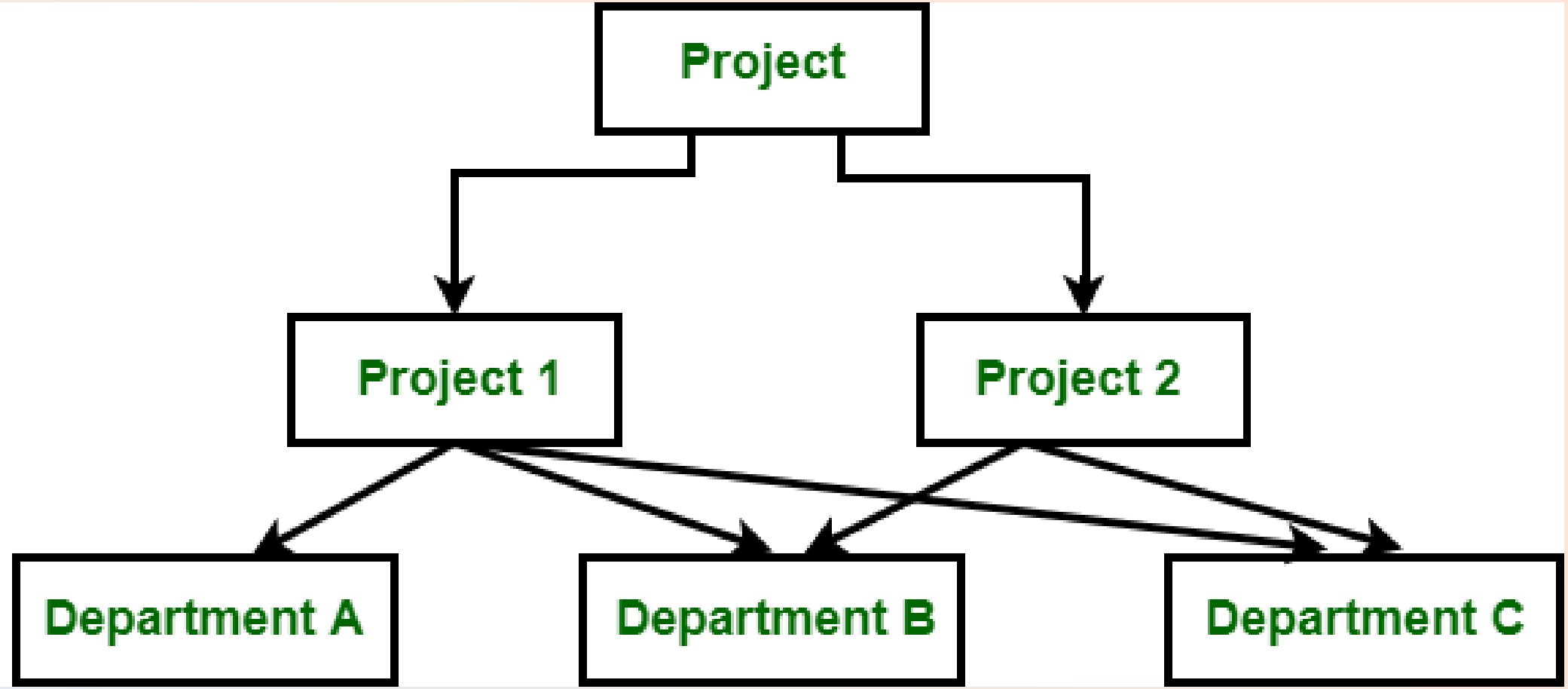
View/External Schema

- It is a view level design which is able to define the interaction between end-user and database.
- User is able to interact with the database with the help of the interface without knowing much about the stored mechanism of data in database.

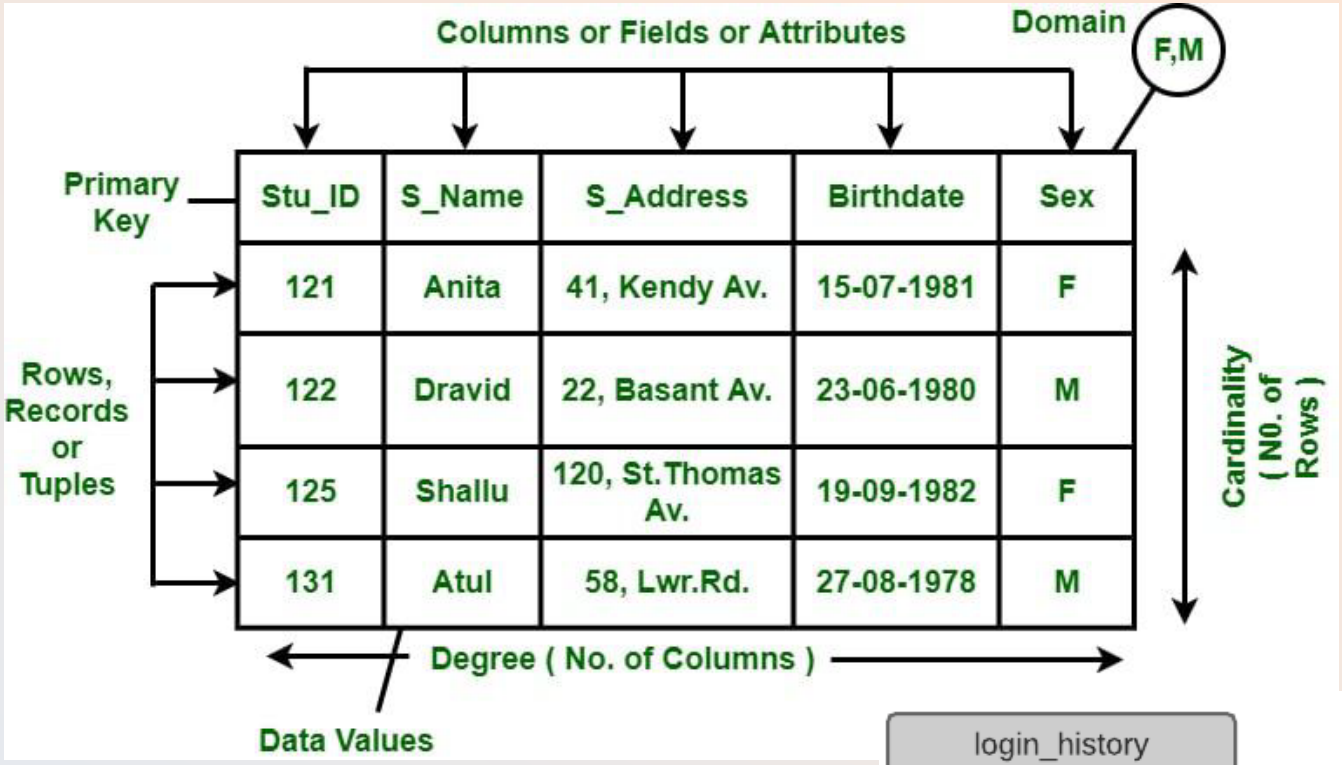
Data Models: Hierarchical Model

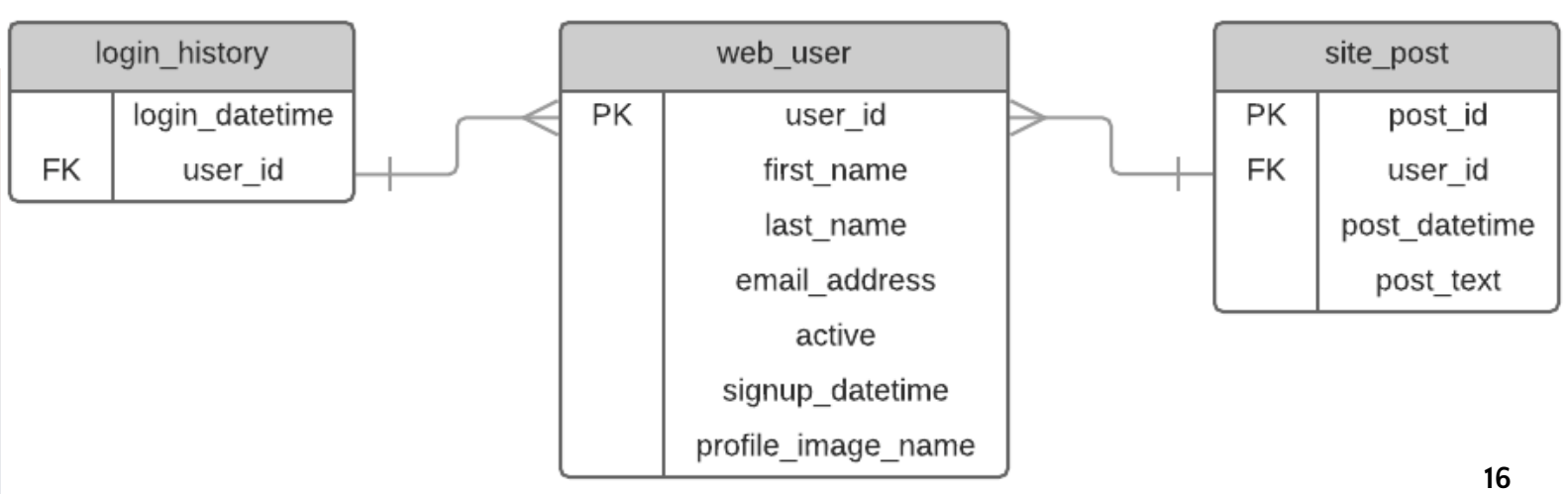


Data Models: Network Model



Data Models: Relational Model





Database Languages

Data Definition Language (DDL):

- When no strict separation of levels/schema, DDL is used by the DBA and by database designers to define both physical/internal and logical/conceptual schema
- The DBMS will have a DDL compiler whose function is to process DDL statements in order to identify descriptions of the schema constructs and to store the schema description in the DBMS catalog
- In DBMS where a clear separation is maintained between the conceptual and internal levels, the DDL is used to specify the ***conceptual schema*** only

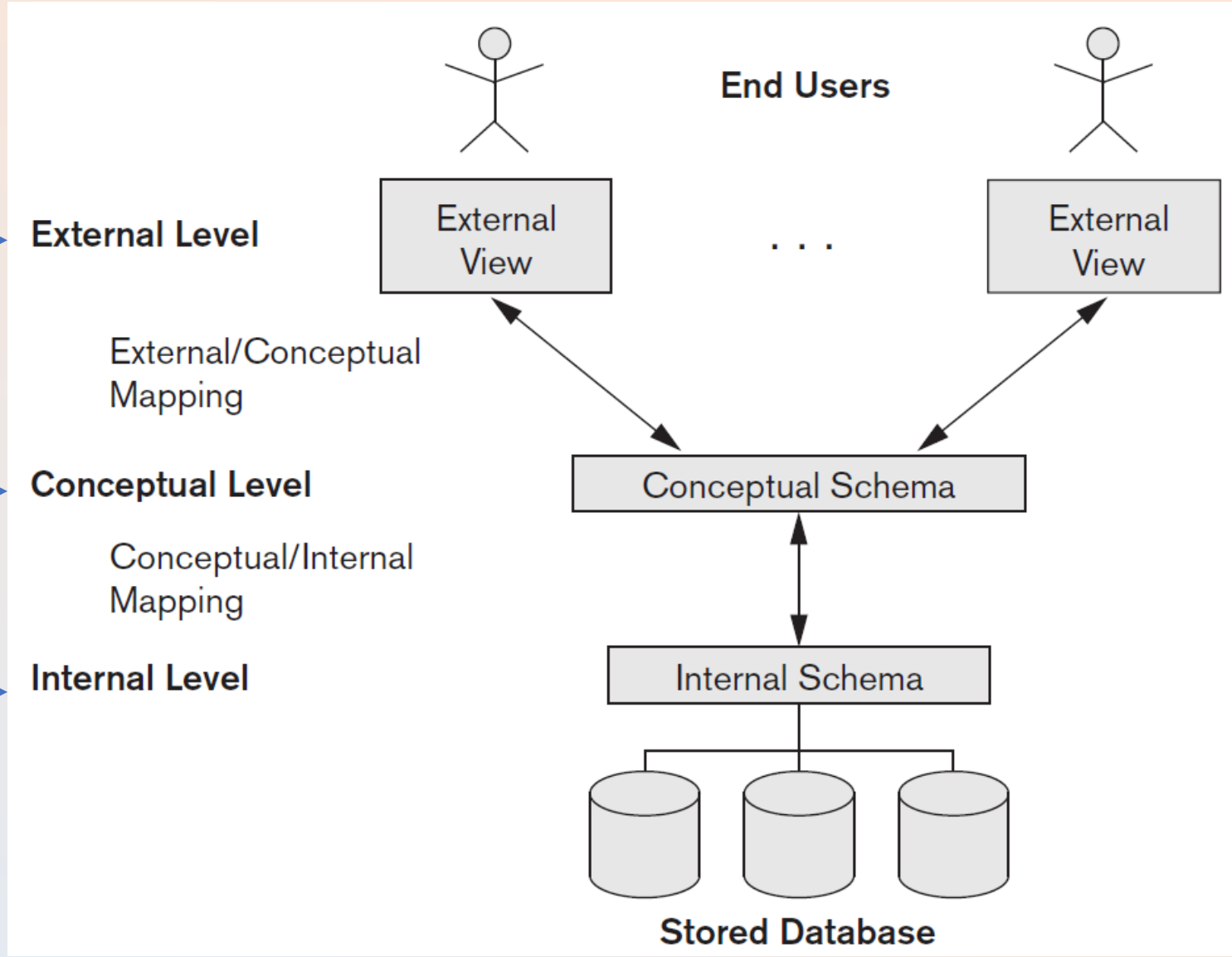
Storage Definition Language (SDL):

- In DBMS where a clear separation is maintained between the conceptual and internal levels, the SDL is used to specify the ***internal schema***
- In most relational DBMSs today, there is no specific language that performs the role of SDL. Instead, the internal schema is specified by a combination of functions, parameters, and specifications related to storage of files.

View Definition Language (VDL):

- To specify user views and their mappings to the conceptual schema
- In most DBMSs the DDL is used to define both conceptual and external schemas
- In relational DBMSs, SQL is used in the role of VDL to define user or application views as results of predefined queries

Database Languages



Data Manipulation Language (DML):

- Once the database schemas are compiled and the database is populated with data, users must have some means to manipulate the database.
- Typical manipulations include (CRUD) Create, Read, Update, and Delete data.
- The DBMS provides a set of operations or a language called the data manipulation language (DML) for these purposes.

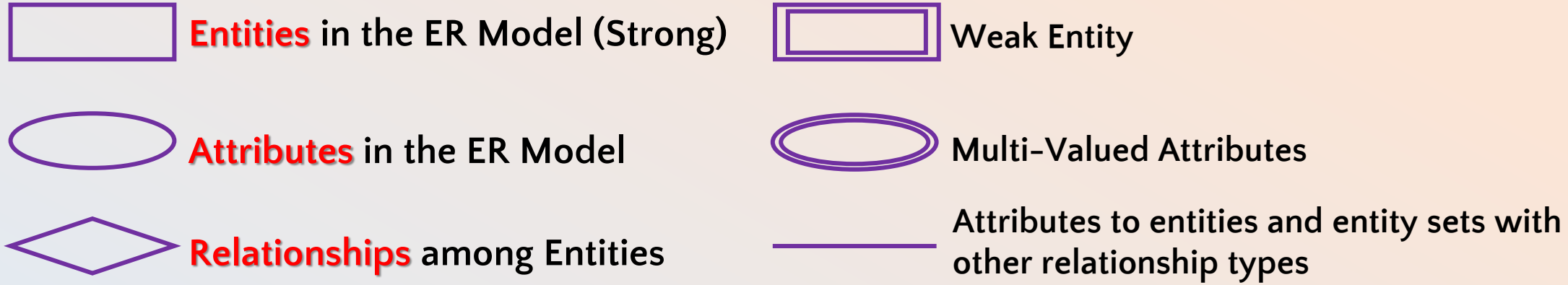
There are two main types of DML

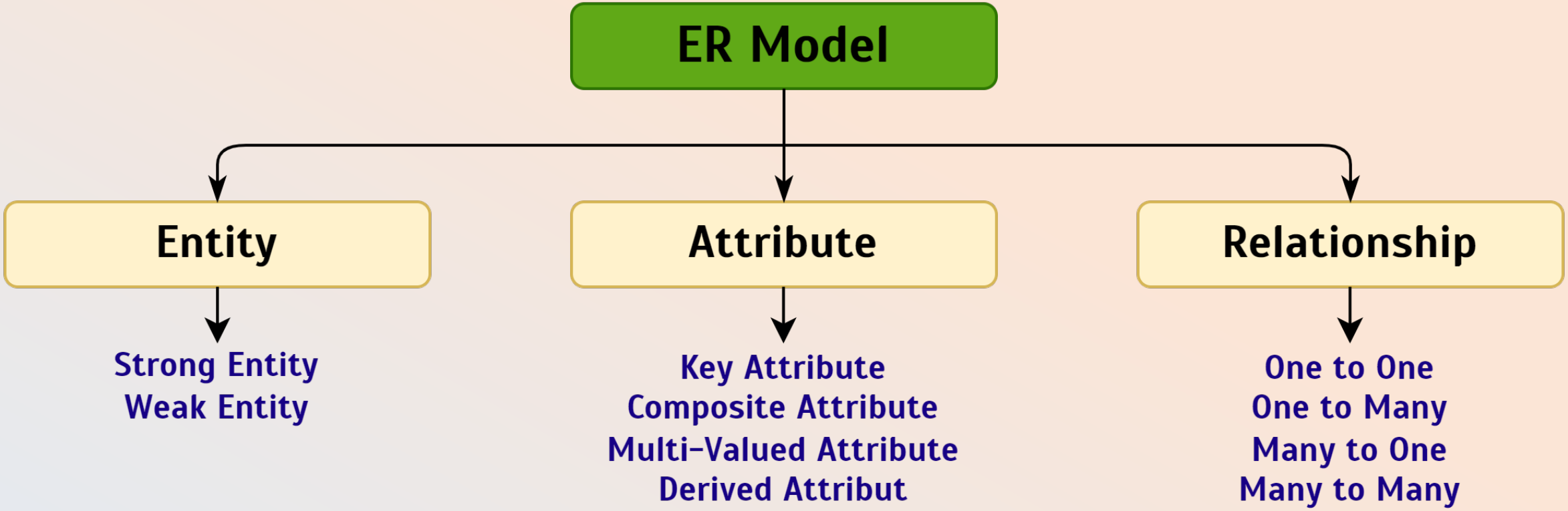
- High-level or Nonprocedural DML
- Lowl-evel or Procedural DML

- A nonprocedural DML can be used on its own to specify complex database operations concisely. Many DBMSs allow high-level DML statements either to be entered interactively from a display or terminal or to be embedded in a general-purpose programming language.
- A procedural DML must be embedded in a general-purpose programming language. This type of DML typically retrieves individual records or objects from the database and processes each separately. Therefore, it needs to use programming language constructs, such as looping, to retrieve and process each record from a set of records.

Entity Relationship Model (ER Model)

- The ER model describes data as entities, relationships, and attributes.
- A model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.





	ID	Name	Address	Phone
Strong Entity	PD201803	Aman	586, City Hall, Trichy	1234567890
Weak Entity	---	ABC Corp	2/15, KK Nagar, Trichy	0422-1234567

Entity

- ER model represents an entity, which is an object in the real world with an independent existence.
- An entity may be an object with
  - a physical existence (for example, a particular person, car, house, or employee) or
  - an object with a conceptual existence (for instance, age, color, education)

1. Strong Entity

- A type of entity that has a key attribute. Strong Entity does not depend on other Entity in the Schema.

2. Weak Entity

- Some entity type exists for which key attributes can’t be defined. These are called Weak Entity types.

Entity Types, Entity Sets

Entity Type : It refers to the category that a particular entity belongs to.

Example :

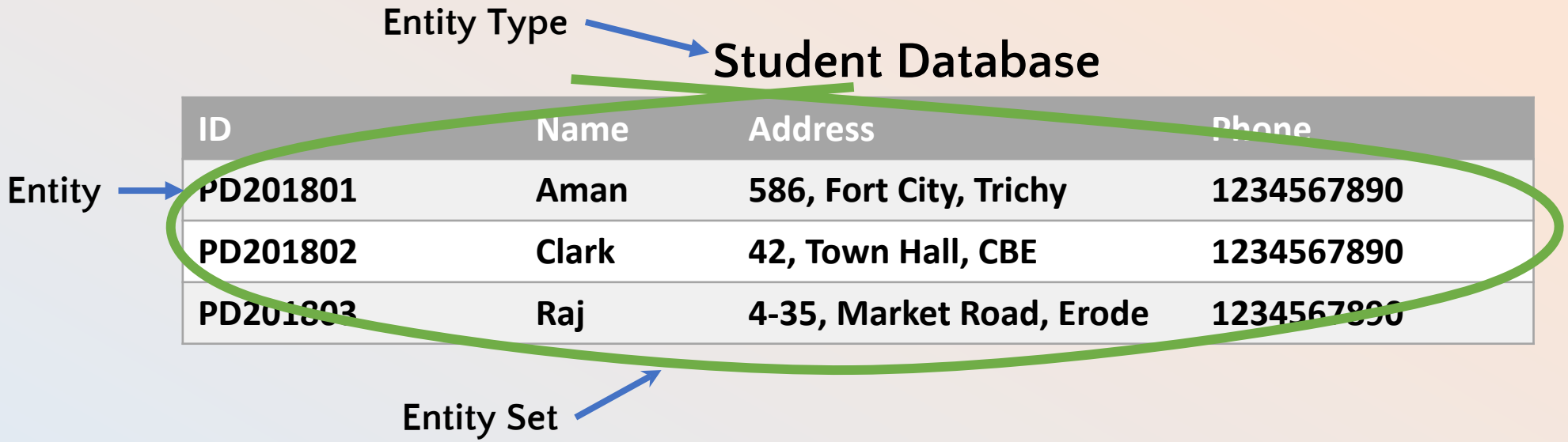
- A table named student in a university database.
- A table named employee in a company database.

Entity Set : An entity set is a collection or set of all entities of a particular entity type at any point in time. The type of all the entities should be the same.

Example :

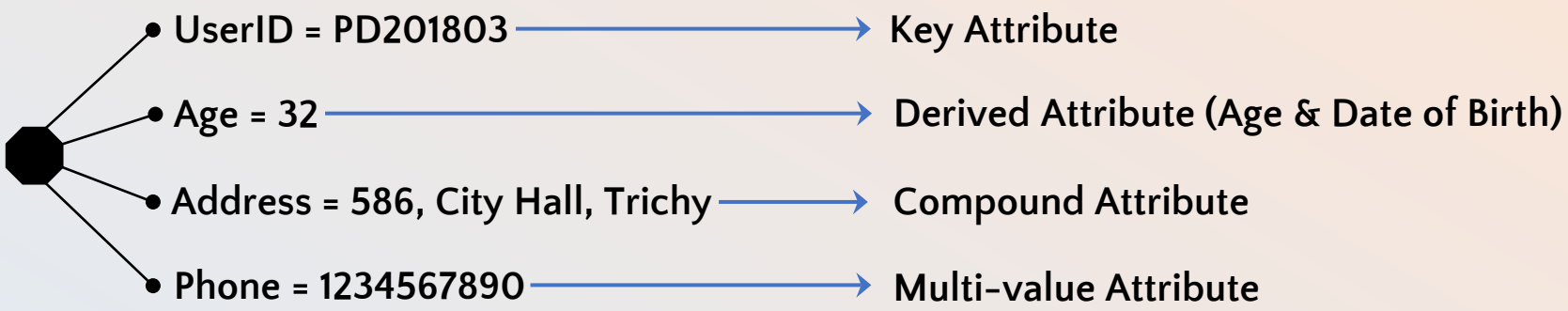
- The collection of all the students from the student table at a particular instant of time is an example of an entity set.
- The collection of all the employees from the employee table at a particular instant of time is an example of an entity set.

Entity	Entity Type	Entity Set
A thing in the real world with independent existence	A category of a particular entity	Set of all entities of a particular entity type.
Any particular row (a record) in a relation (table) is known as an entity.	The name of a relation (table) in RDBMS is an entity type	All rows of a relation (table) in RDBMS is entity set

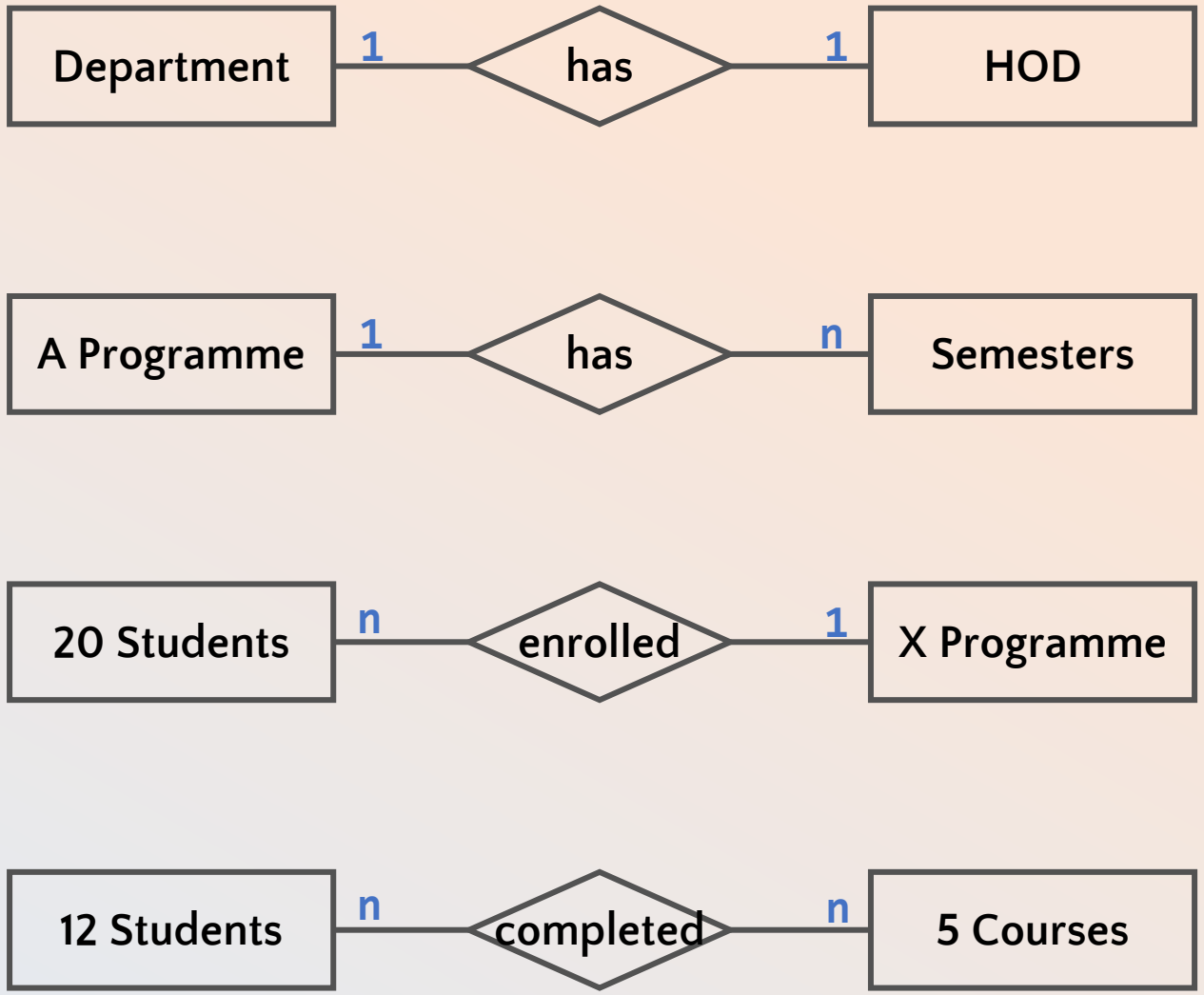


Attributes

- Each entity has attributes—the particular properties that describe it.
- The attribute values that describe each entity become a major part of the data stored in the database.
- For example, an EMPLOYEE entity may be described by the employee's name, age, address, salary, and job.



Relationships



Thank you!