Database Management System

Unit 6:Database Management System

- Introduction to Database, components and structure of DBMS – logical structure – the 3 level architecture and mapping among them. Comparison between traditional file based system and DBMS. Advantages and drawbacks of DBMS.

- Relational Model - What is relational model, Relational key constraints – candidate key, primary key, foreign key.ER Model – entities, attributes, relationship, and cardinality. Entity types, Entity sets Attributes and Keys Relationship types, Relationship Sets, converting ER diagram to relational tables. Database Schema

Unit 6:Database Management System

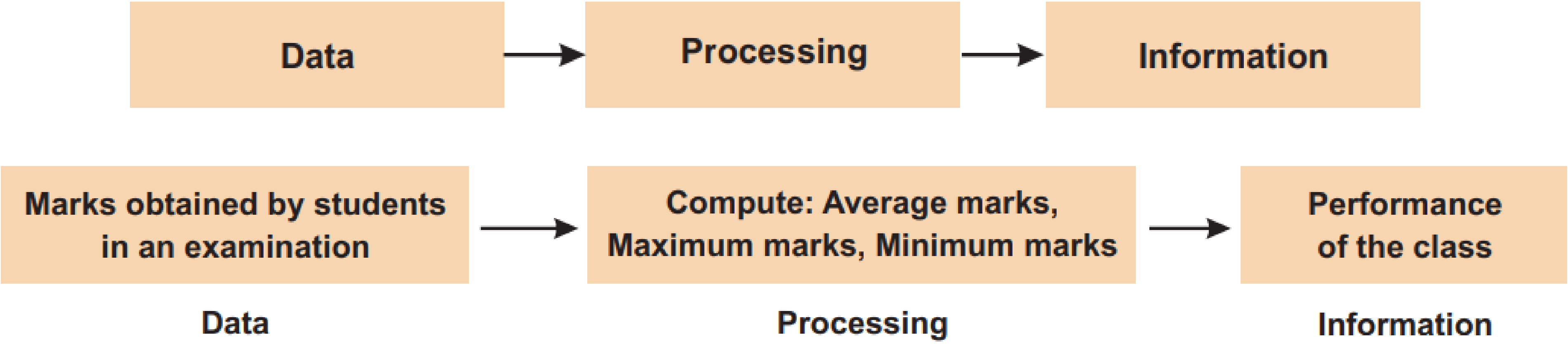
- Database Anomalies, CODD Rules and Normalization theory, 1 NF, 2 NF, 3 NF and BCNF.

- Introduction to transaction and concept of concurrency control. Transaction and system concepts, desirable properties of transactions, transaction support in SQL. Concurrency control techniques, the locking protocol, serializable schedules, locks, 2 phase commit. Techniques, concurrency control based on timestamp ordering

Database Concept

- Data is a collection of raw facts which have not been processed to reveal useful information.

- Information is produced by processing data.



Data → Processing → Information

Marks obtained by students in an examination → Compute: Average marks, Maximum marks, Minimum marks → Performance of the class

Data

Processing

Information

DATABASE PROPERTIES

1) A database is a representation of some aspect of the real world also called mini-world.

Whenever there are changes in this mini-world they are also reflected in the database.

2) It is designed, built and populated with data for specific purpose.

3) It can be of any size and complexity.

4) It can be maintained manually or it may be computerized.

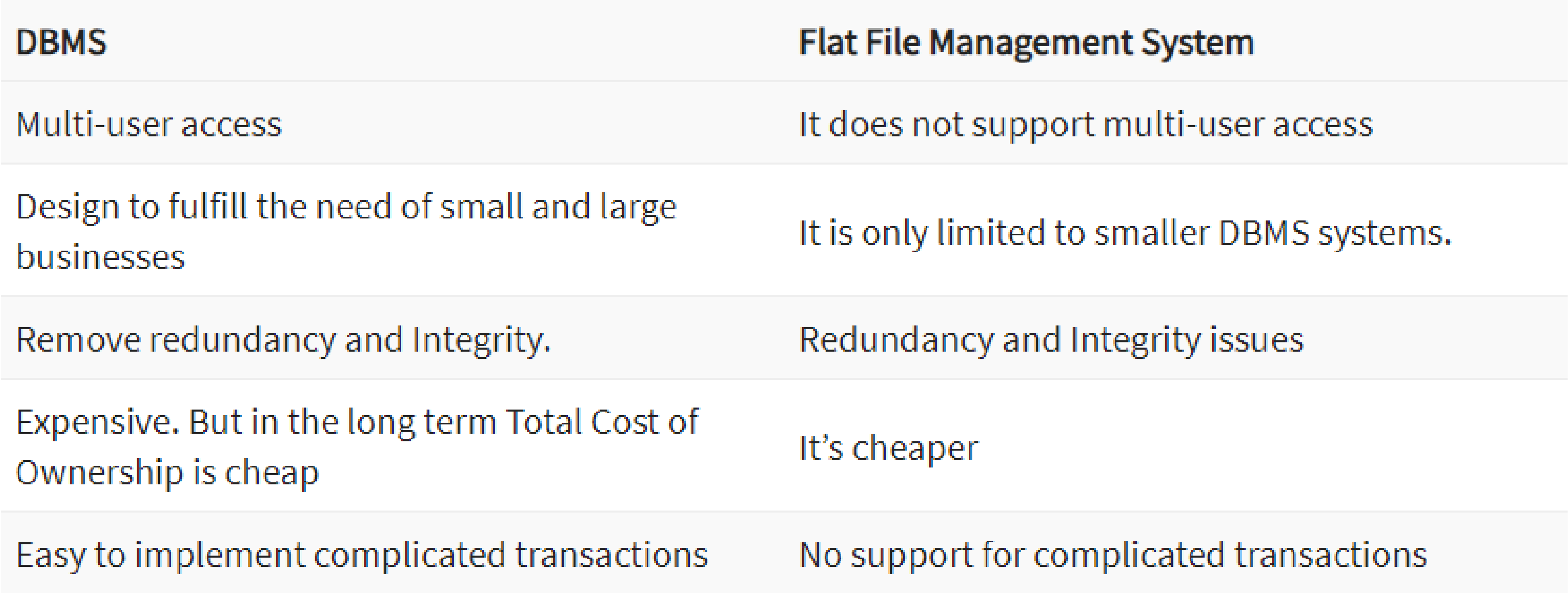
DATA PROBLEMS

1. Data Redundancy: Same information is stored in more than one file. This would result in wastage of space.

2. Data Inconsistency: If a file is updated then all the files containing similar information must be updated else it would result in inconsistency of data.

3. Lack of Data Integration: As data files are independent, accessing information out of multiple files becomes very difficult.

Data vs. Flat File



DBMS	Flat File Management System
Multi-user access	It does not support multi-user access
Design to fulfill the need of small and large businesses	It is only limited to smaller DBMS systems.
Remove redundancy and Integrity.	Redundancy and Integrity issues
Expensive. But in the long term Total Cost of Ownership is cheap	It's cheaper
Easy to implement complicated transactions	No support for complicated transactions

DATABASE MANAGEMENT SYSTEM (DBMS)

- A database management system is a collection of programs that enables users to create, maintain and use a database.

- It enables creation of a repository of data that is defined once and then accessed by different users as per their requirements.

- Thus there is a single repository of data which is accessed by all the application programs

VARIOUS OPERATIONS THAT NEED TO BE PERFORMED ON A DATABASE

1. Defining the Database: It involves specifying the data type of data that will be stored in the database and also any constraints on that data.

2. Populating the Database: It involves storing the data on some storage medium that is controlled by DBMS.

3. Manipulating the Database: It involves modifying the database, retrieving data or querying the database, generating reports from the database etc.

VARIOUS OPERATIONS THAT NEED TO BE PERFORMED ON A DATABASE

4. Sharing the Database: Allow multiple users to access the database at the same time.

5. Protecting the Database: It enables protection of the database from software/hardware failures and unauthorized access.

6. Maintaining the Database: It is easy to adapt to the changing requirements.

Characteristics of Database Management Systems

1. **Self-describing Nature of a Database System: DBMS contains not only the database but also the description of the data that it stores. This description of data is called metadata.**

2. **Insulation Between Programs and Data: Since the definition of data is stores separately in a DBMS, any change in the structure of data would be done in the catalogue and hence programs which access this data need not be modified. This property is called Program-Data Independence.**

CHARACTERISTICS OF DATABASE MANAGEMENT SYSTEMS

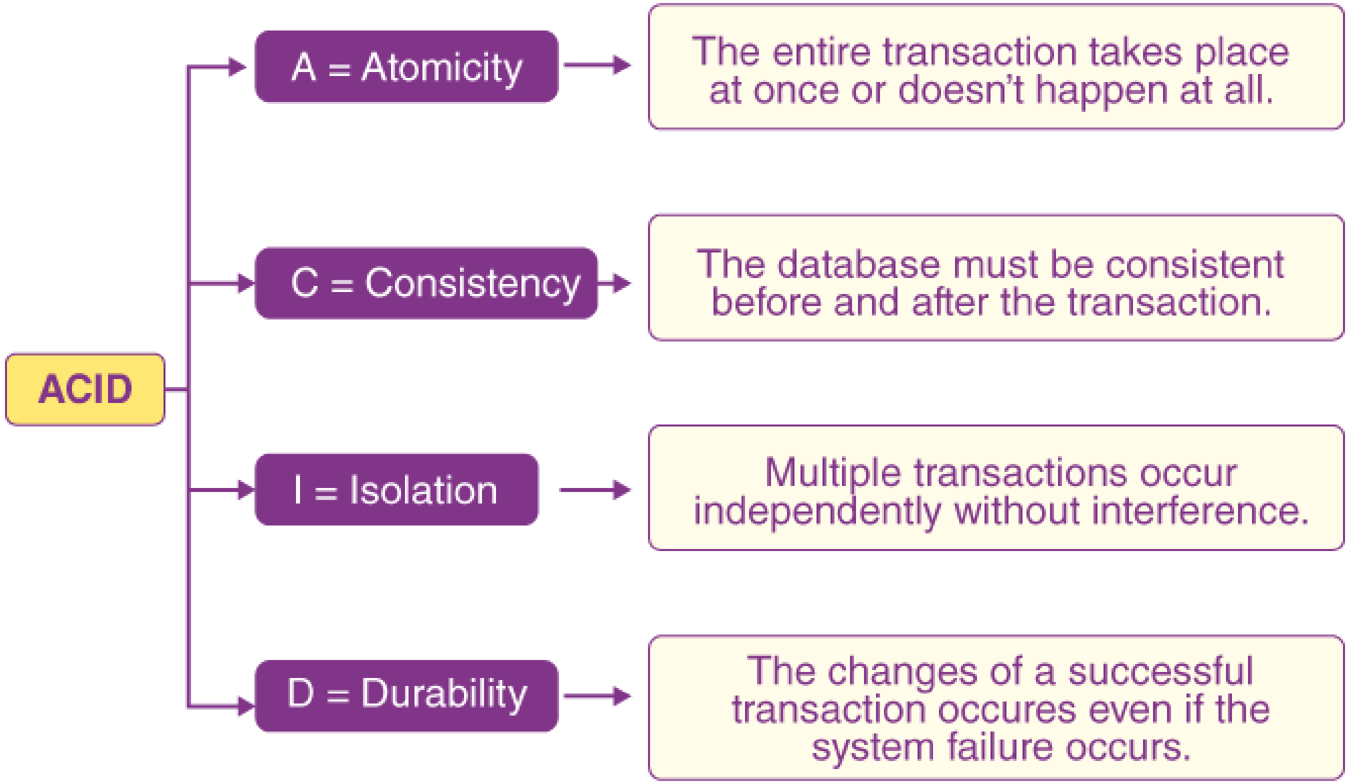
3. Sharing of Data: A multiuser environment allows multiple users to access the database simultaneously.

Limitations of using DBMS Approach

1. High Cost: The cost of implementing a DBMS system is very high. It is also a very time consuming process which involves analyzing user requirements, designing the database specifications, writing application programs and then also providing training.

2. Security and Recovery Overheads: Unauthorized access to a database can lead to Database Management Applications threat to the individual or organization depending on the data stored.

ACID Properties in DBMS



ACID

A = Atomicity → The entire transaction takes place at once or doesn't happen at all.

C = Consistency → The database must be consistent before and after the transaction.

I = Isolation → Multiple transactions occur independently without interference.

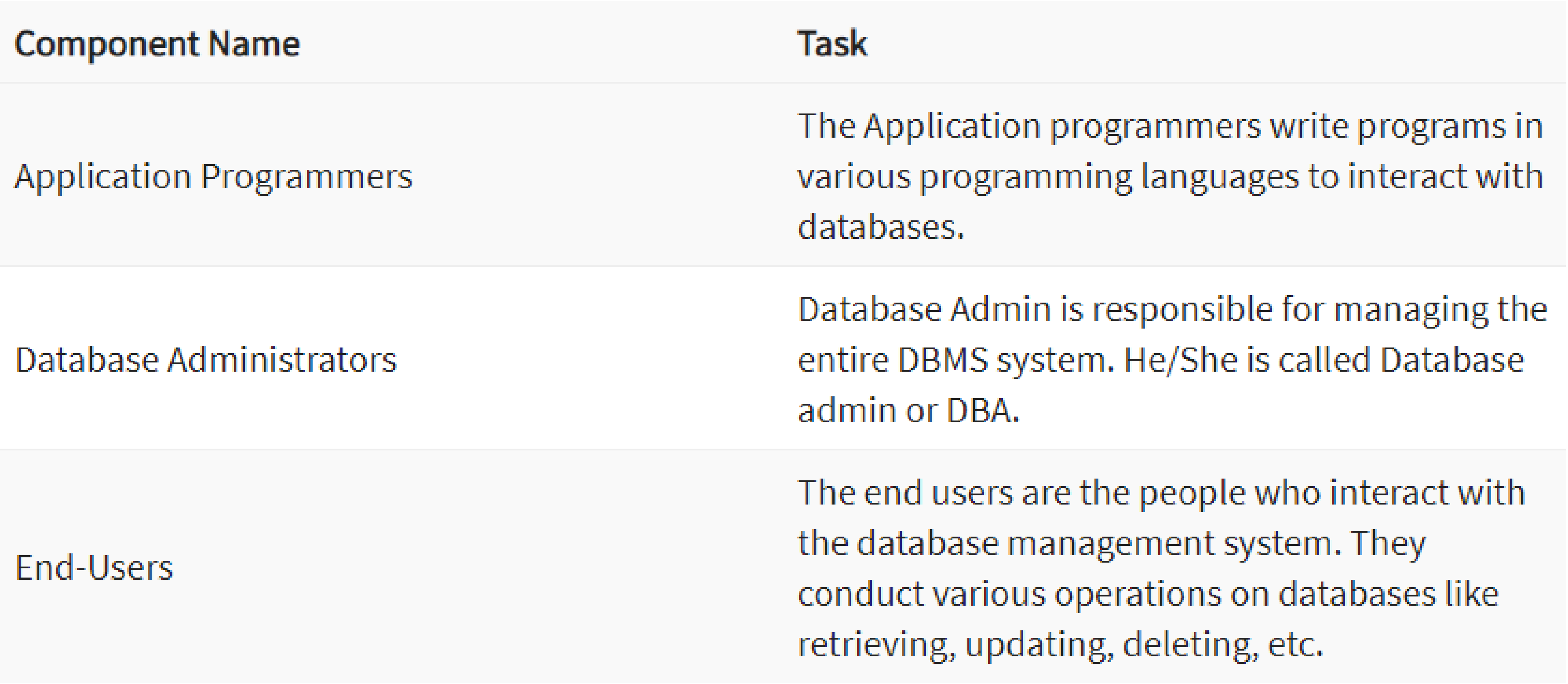
D = Durability → The changes of a successful transaction occures even if the system failure occurs.

ACID Properties

To ensure integrity of data, the database system must maintain:

- **Atomicity.** Either all operations of the transaction are properly reflected in the database or none are.
- **Consistency.** Execution of a transaction in isolation preserves the consistency of the database.
- **Isolation.** Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions.
- **Durability.** After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

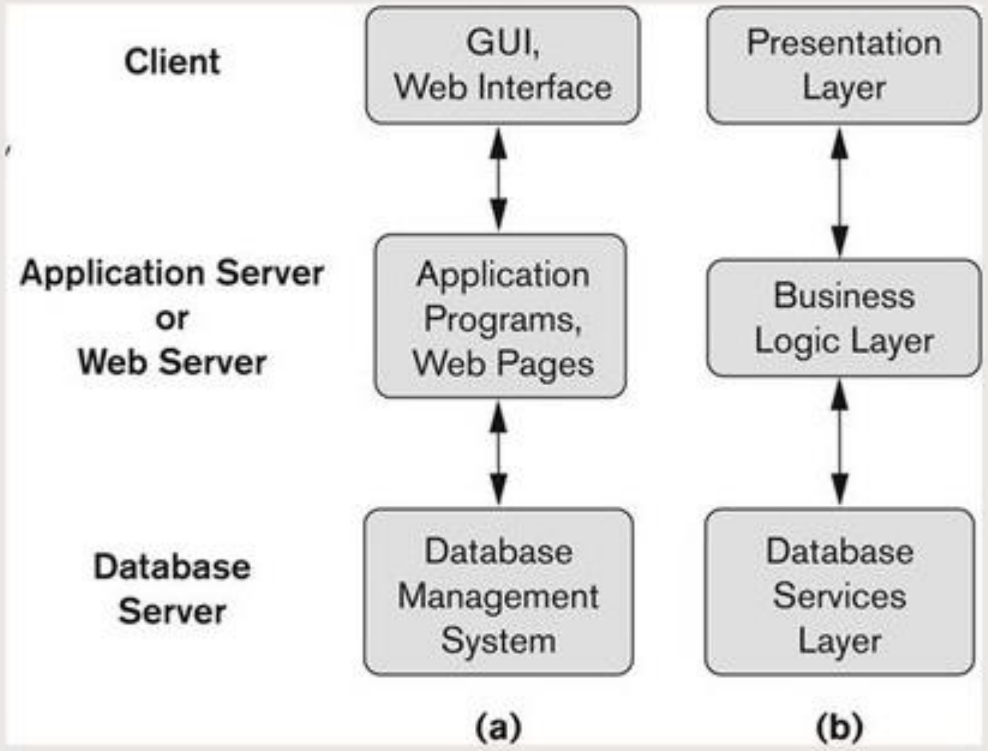
Users of DBMS



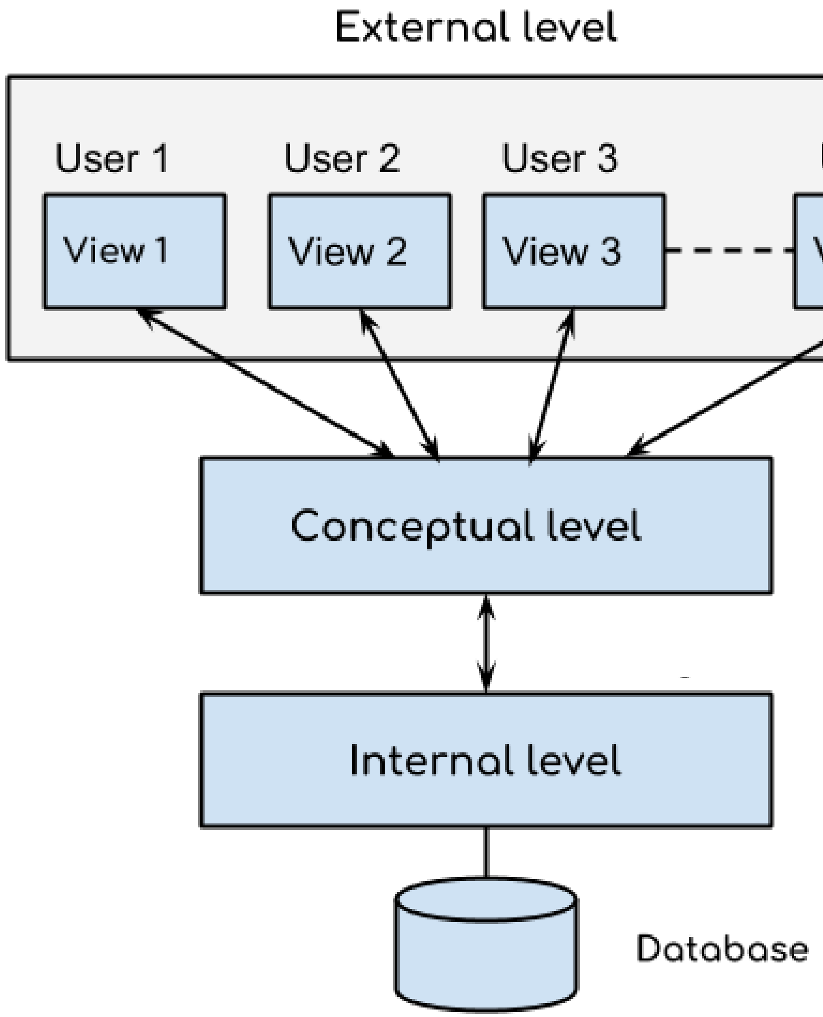
Component Name	Task
Application Programmers	The Application programmers write programs in various programming languages to interact with databases.
Database Administrators	Database Admin is responsible for managing the entire DBMS system. He/She is called Database admin or DBA.
End-Users	The end users are the people who interact with the database management system. They conduct various operations on databases like retrieving, updating, deleting, etc.

THREE TIER ARCHITECTURE IN DBMS

Three-tier client-server architecture



THREE TIER ARCHITECTURE IN DBMS



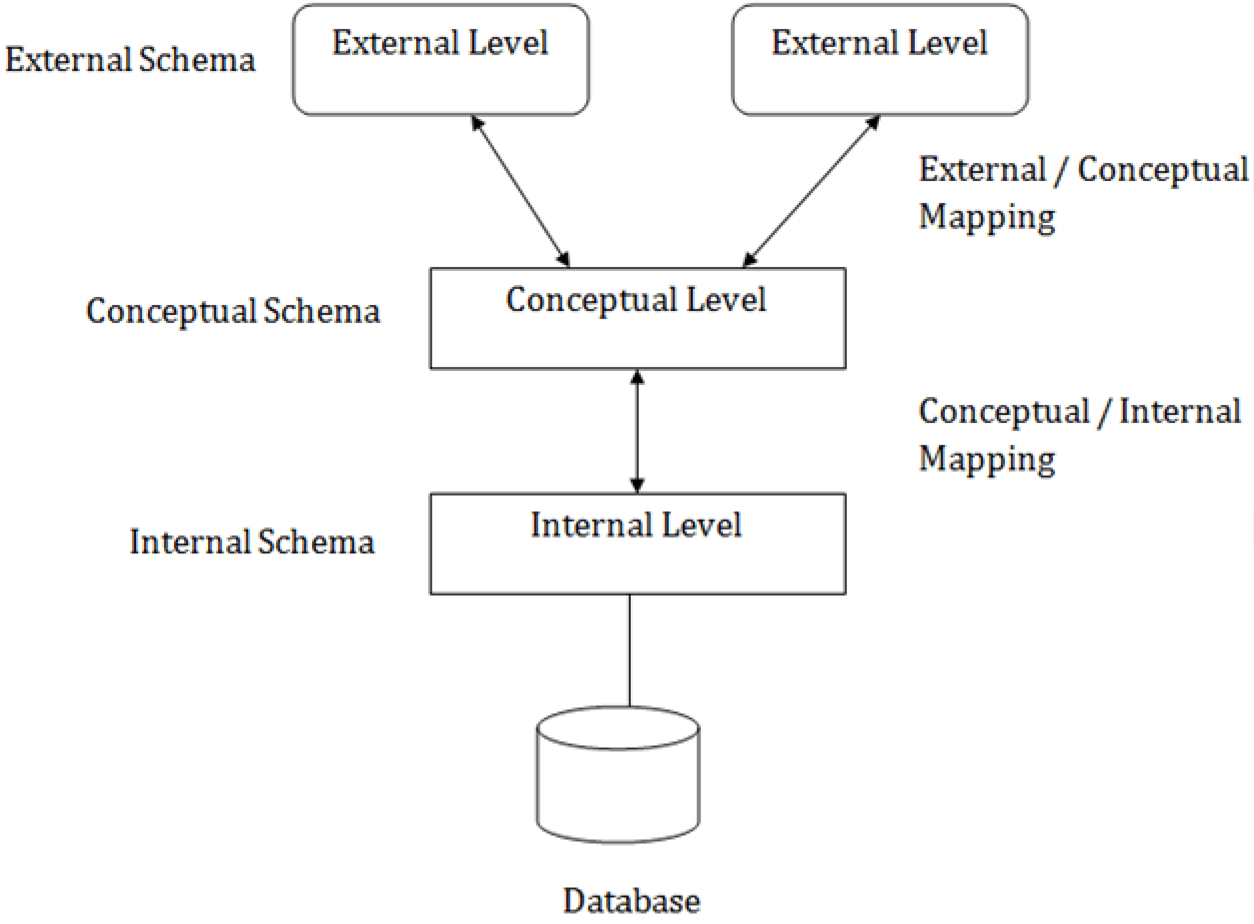
Three schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.

- This framework is used to describe the structure of a specific database system.

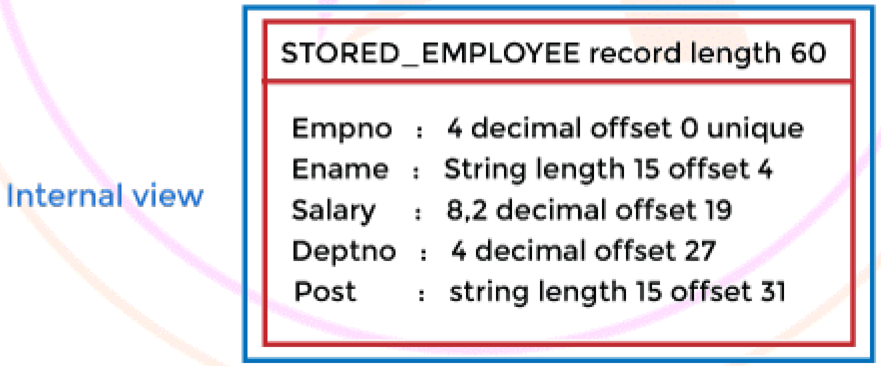
- The three schema architecture is also used to separate the user applications and physical database.

THREE TIER ARCHITECTURE IN DBMS



INTERNAL LEVEL

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.



Internal view

STORED_EMPLOYEE record length 60

```
Empno   :  4 decimal offset 0 unique
Ename   :  String length 15 offset 4
Salary  :  8,2 decimal offset 19
Deptno  :  4 decimal offset 27
Post    :  string length 15 offset 31
```

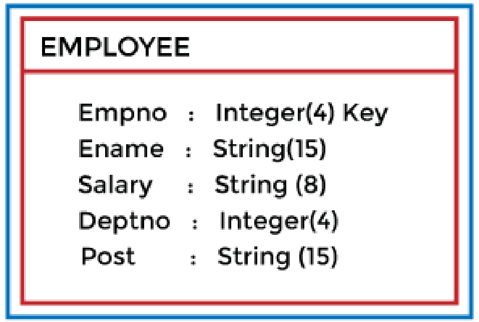
INTERNAL LEVEL

- The internal level is generally is concerned with the following activities:

- Storage space allocations.
- For Example: B-Trees, Hashing etc.
- Access paths.
- For Example: Specification of primary and secondary keys, indexes, pointers and sequencing.
- Data compression and encryption techniques.
- Optimization of internal structures.
- Representation of stored fields.

CONCEPTUAL SCHEMA

Global view



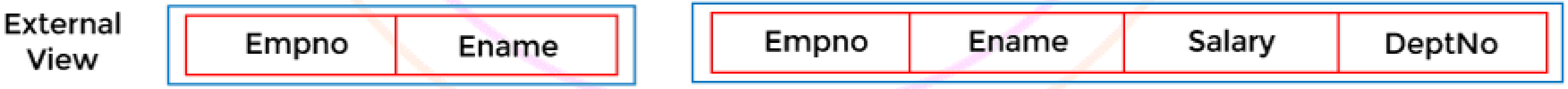
```
EMPLOYEE

Empno   :  Integer(4) Key
Ename   :  String(15)
Salary  :  String (8)
Deptno  :  Integer(4)
Post    :  String (15)
```

- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

EXTERNAL LEVEL

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.



External View

Empno	Ename

Empno	Ename	Salary	DeptNo

RELATIONAL DATABASE (RDBMS)

- DBMS stands for Relational Database Management System.

- All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS.

- It is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd.

RELATIONAL DATABASE (RDBMS)

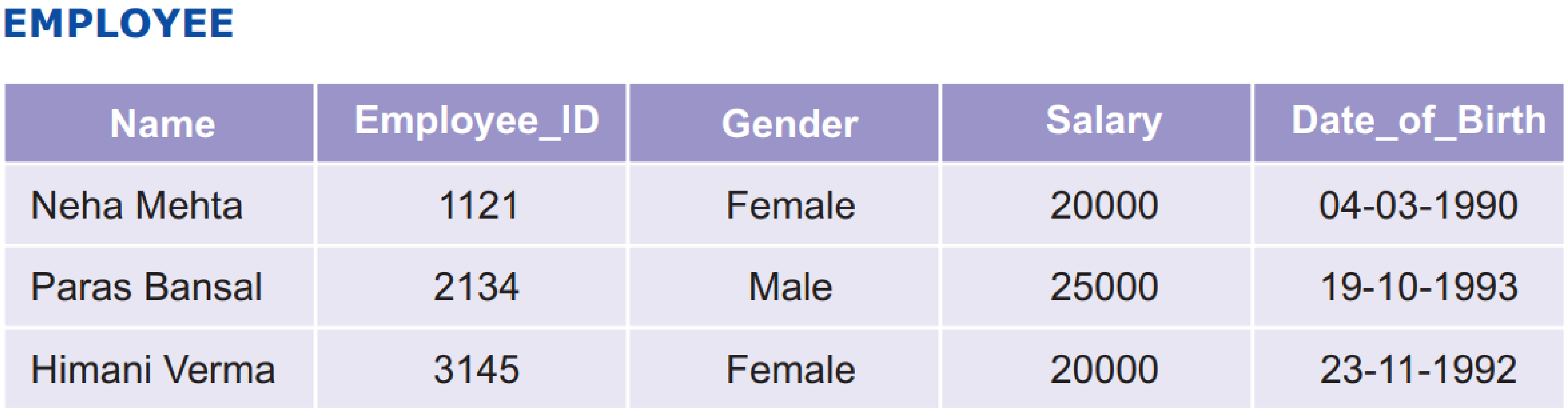
- It is used to organize collection of data as a collection of relations where each relation corresponds to a table of values.

- Each row in the table corresponds to a unique instance of data and each column name is used to interpret the meaning of that data in each row.

Relational Database (RDBMS)

- For example, consider EMPLOYEE table in Figure

- Each row in this table represents facts about a particular employee.



EMPLOYEE

Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992

IN RELATIONAL MODEL,

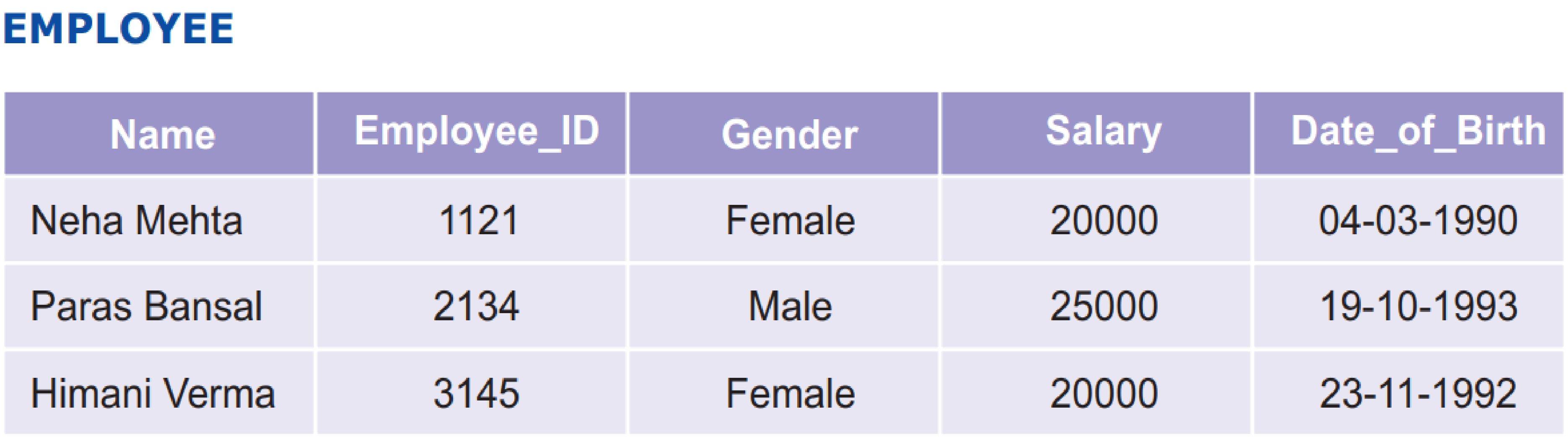
- A row is called a Tuple.
- A column is called an Attribute.
- A table is called as a Relation.
- The data type of values in each column is called the Domain.
- The number of attributes in a relation is called the Degree of a relation.
- The number of rows in a relation is called the Cardinality of a relation.

In relational model,

- Relation Schema R is denoted by R (A_1 , A_2 , A_3 …, A_n) where R is the relation name and A_1 , A_2 , A_3 …, A_n is the list of attributes.

- Relation State is the set of tuples in the relation at a point in time. A relation state r of relation schema R (A_1 , A_2 , A_3 …, A_n), denoted r(R) is a set of n-tuples r = {t_1, t_2,...., t_m}, where each n-tuple is an ordered list of values t = <v_1, v_2, ...,v_n>, where v_i is in domain of A or is NULL. Here n is the degree of the relation and m is the cardinality of I the relation.

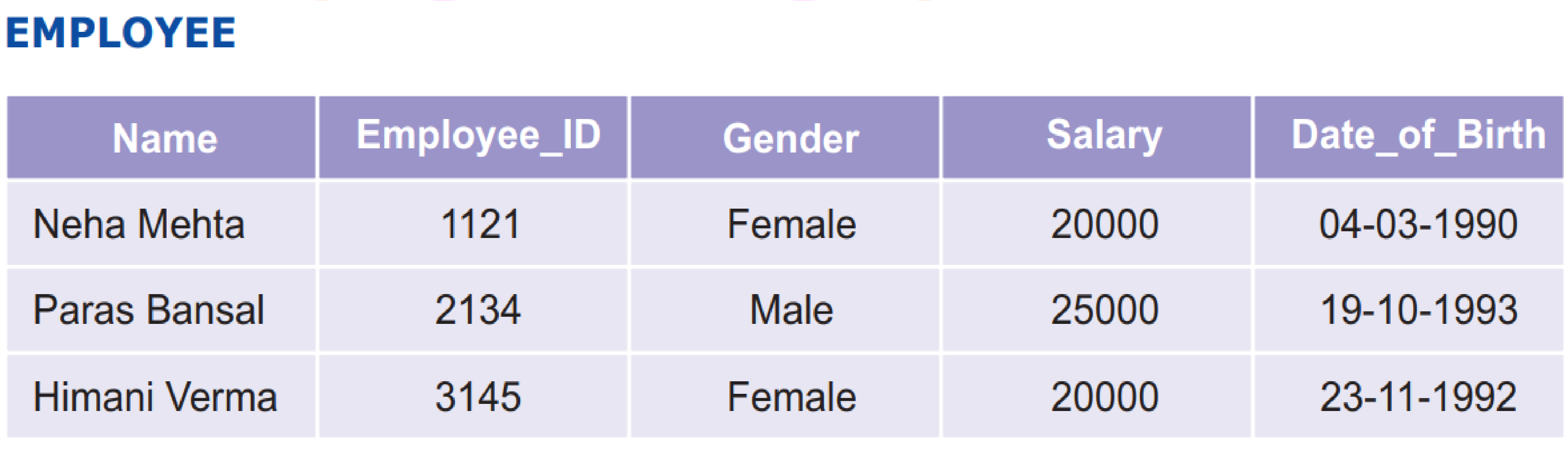
Relational Database (RDBMS)



EMPLOYEE

Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992

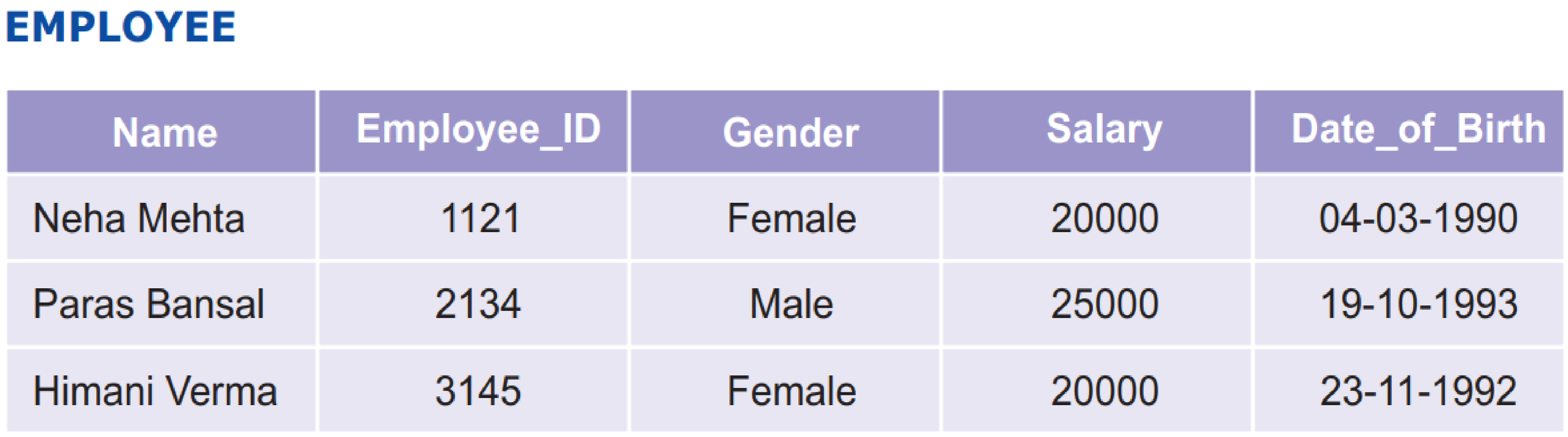
Relational Database (RDBMS)



EMPLOYEE

Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992

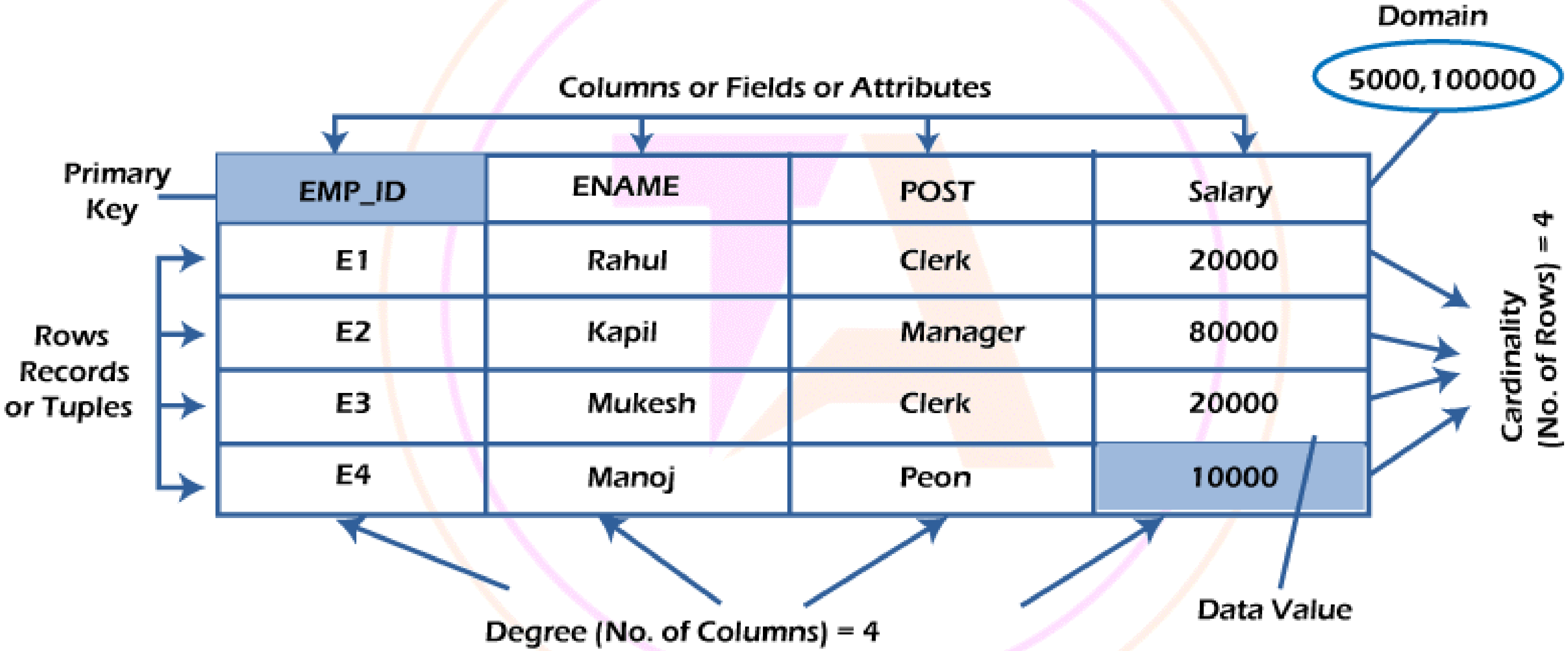
Relational Database (RDBMS)



EMPLOYEE

Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992

RELATIONAL DATABASE (RDBMS)



Domain

5000,100000

Columns or Fields or Attributes

Primary Key

Rows Records or Tuples

EMP_ID	ENAME	POST	Salary
E1	Rahul	Clerk	20000
E2	Kapil	Manager	80000
E3	Mukesh	Clerk	20000
E4	Manoj	Peon	10000

Cardinality (No. of Rows) = 4

Degree (No. of Columns) = 4

Data Value

MORE CHARACTERISTICS OF RELATIONS

- Ordering of tuples is not important in a Relation.

- The ordering of attributes is also unimportant.

- No two tuples of relation should be identical i.e. given any pair of two tuples, value in at least one column must be different.

- The value in each tuple is an atomic value (indivisible).

- If the value of an attribute in a tuple is not known or not applicable or not available, a special value called null is used to represent them

Relational Model Constraints

- Constraints, are restrictions on the values, stored in a database based on the requirements.

- For example, in the relation EMPLOYEE, the Employee_ID must be a 4-digit number, the Date_of_Birth must be such that the birth year > 1985.

RELATIONAL MODEL CONSTRAINTS

- **Domain Constraint:** The domain refers to the possible values each attribute can contain. It can be specified using standard data types such as integers, floating numbers, etc.

- For example, the Employee_ID must be a 4-digit number.

- An attribute entitled Marital_Status may be limited to married or unmarried values.

Relational Model Constraints

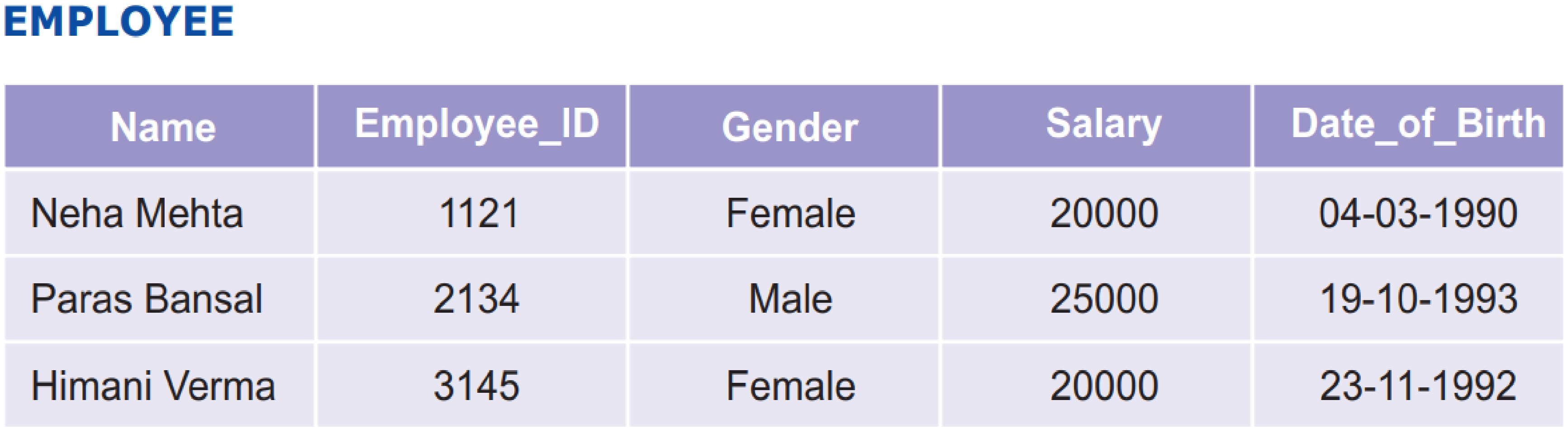
- **NULL Values**

- **The NULL value of the table specifies that the field has been left blank during record creation. It is different from the value filled with zero or a field that contains space.**

RELATIONAL MODEL CONSTRAINTS

- **Key Constraint:**

- **Super-key** is a set of attributes in a relation, for which no two tuples in a relation state have the same combination of values. Every relation must have at least one super-key which is the combination of all attributes in a relation.



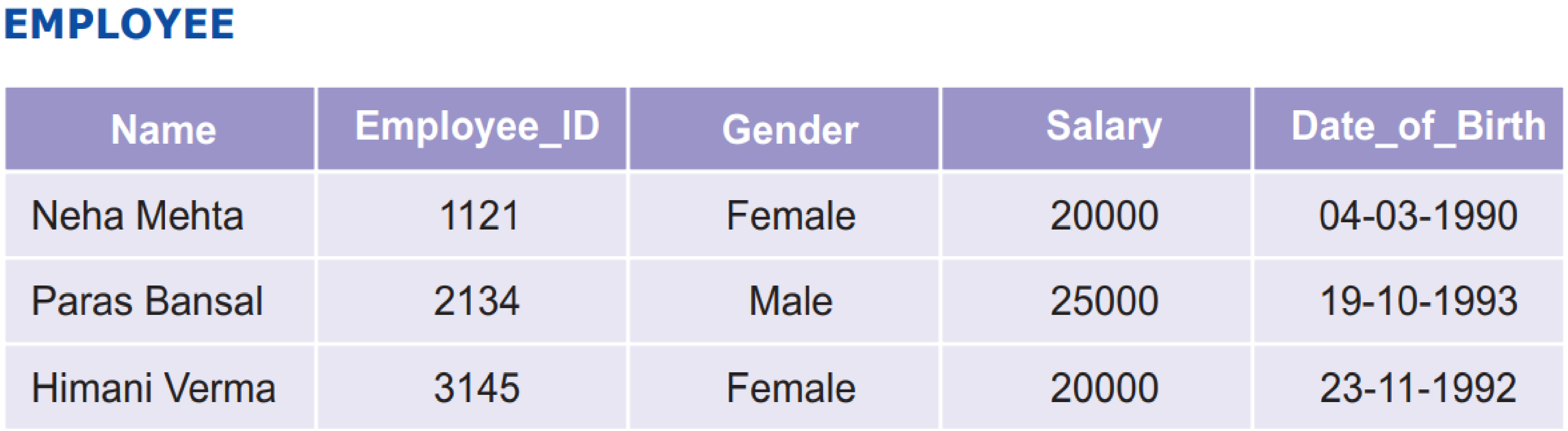
EMPLOYEE

Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992

Relational Model Constraints

- **Candidate key:**

- **Key is the minimal superkey.**

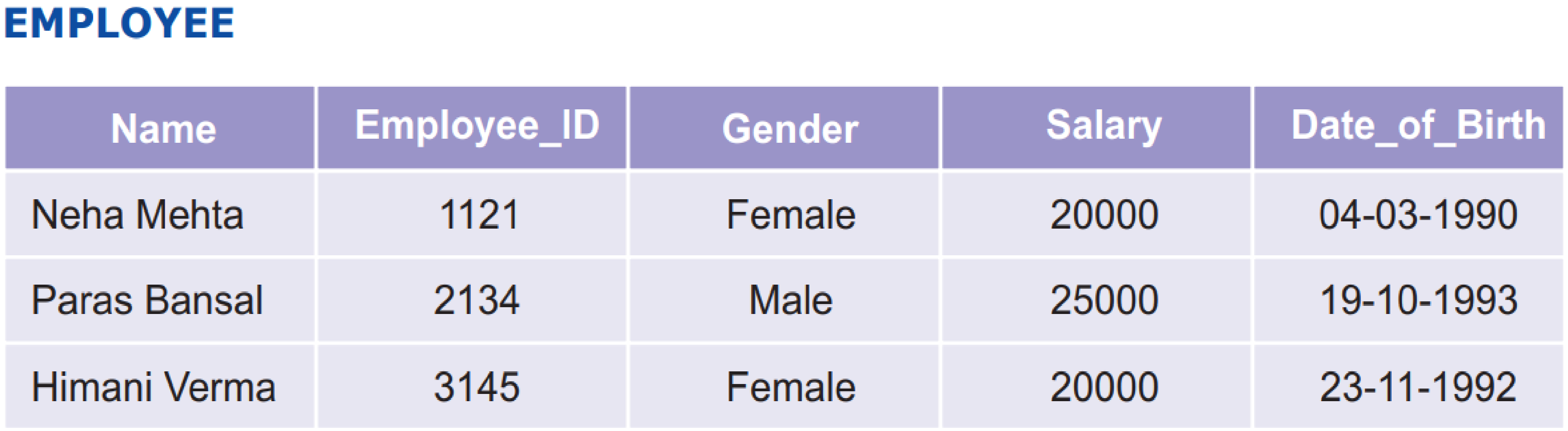


EMPLOYEE

Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992

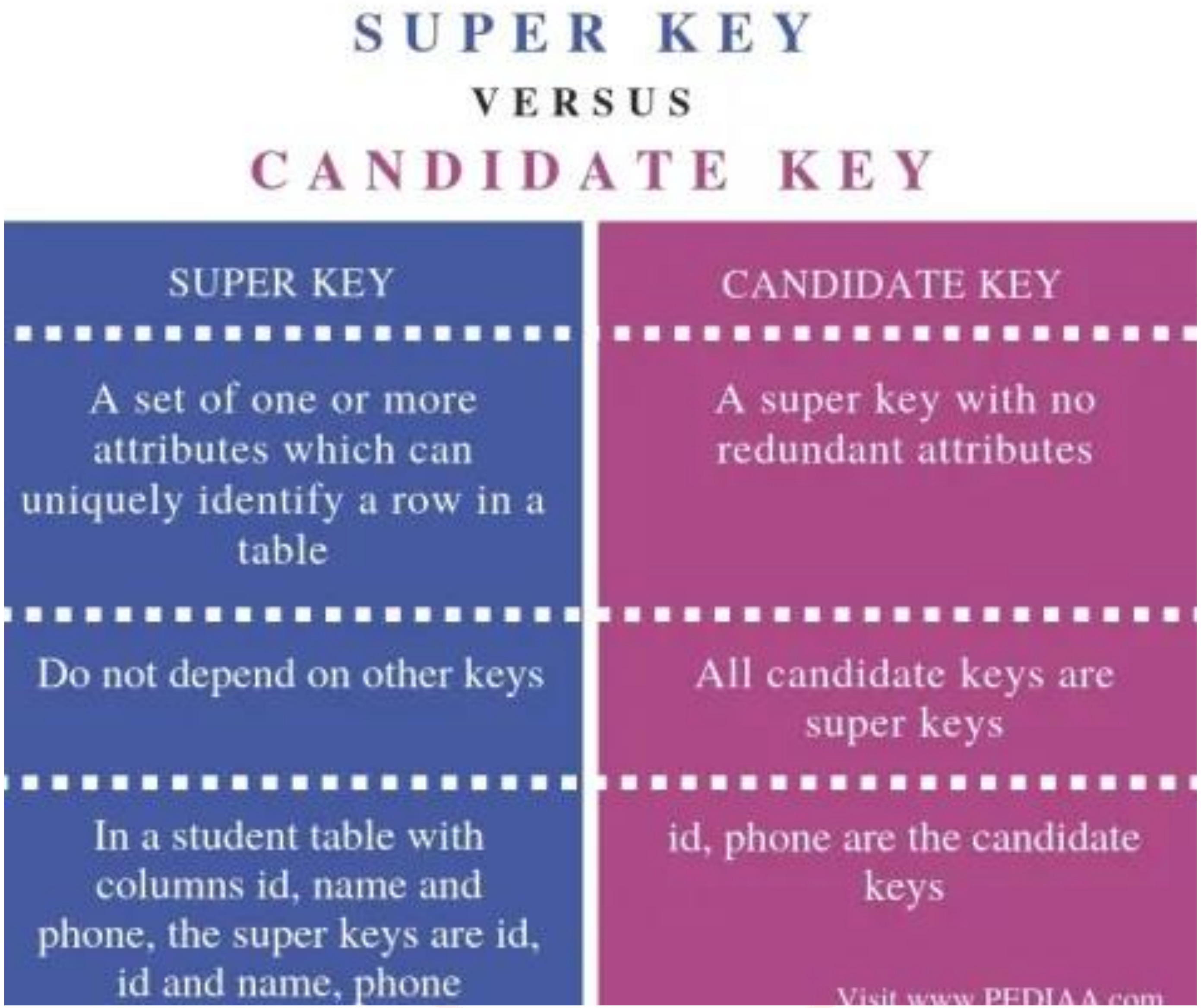
Relational Model Constraints

➢ Primary Key:



EMPLOYEE

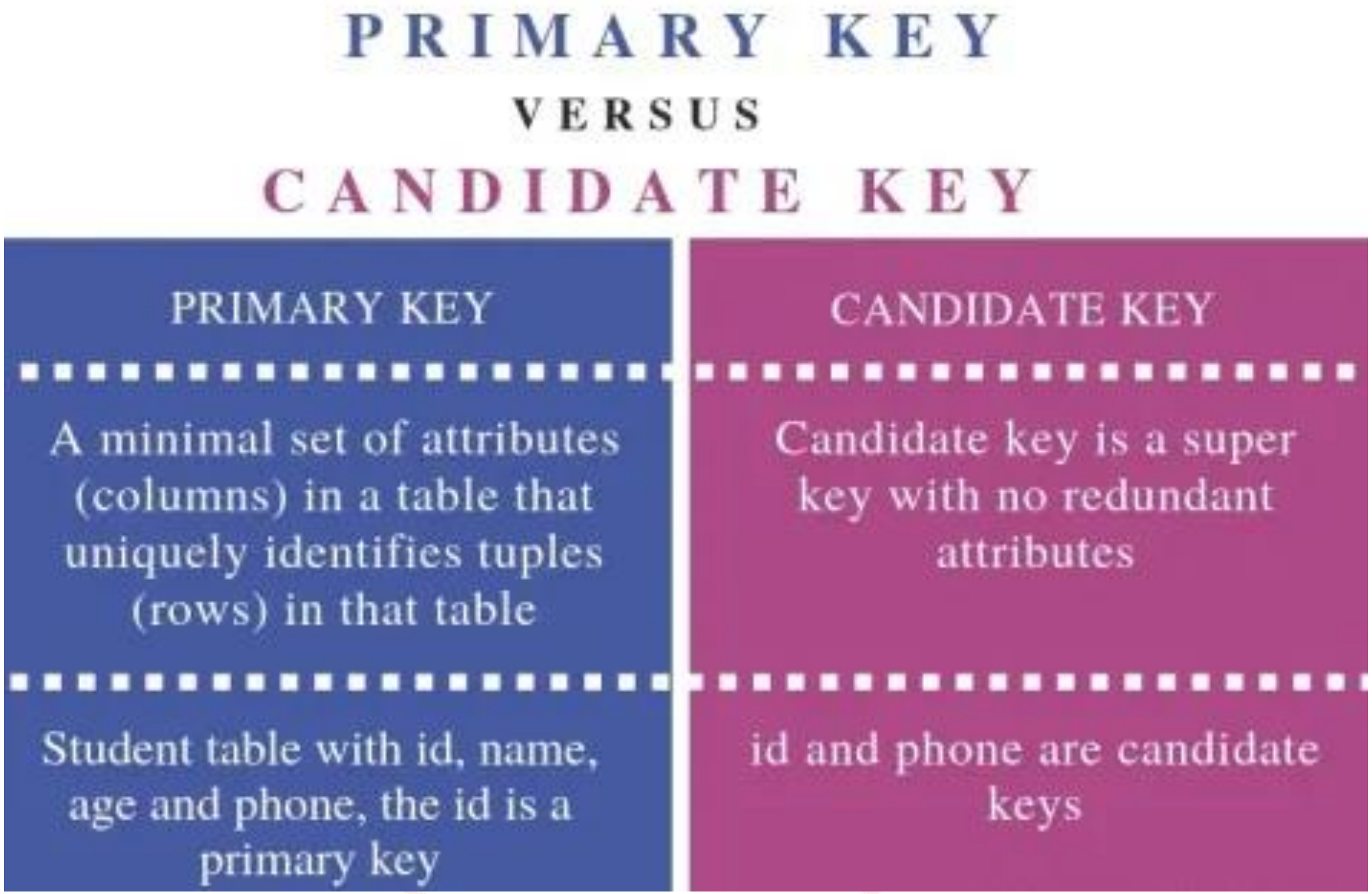
Name	Employee_ID	Gender	Salary	Date_of_Birth
Neha Mehta	1121	Female	20000	04-03-1990
Paras Bansal	2134	Male	25000	19-10-1993
Himani Verma	3145	Female	20000	23-11-1992



SUPER KEY VERSUS CANDIDATE KEY

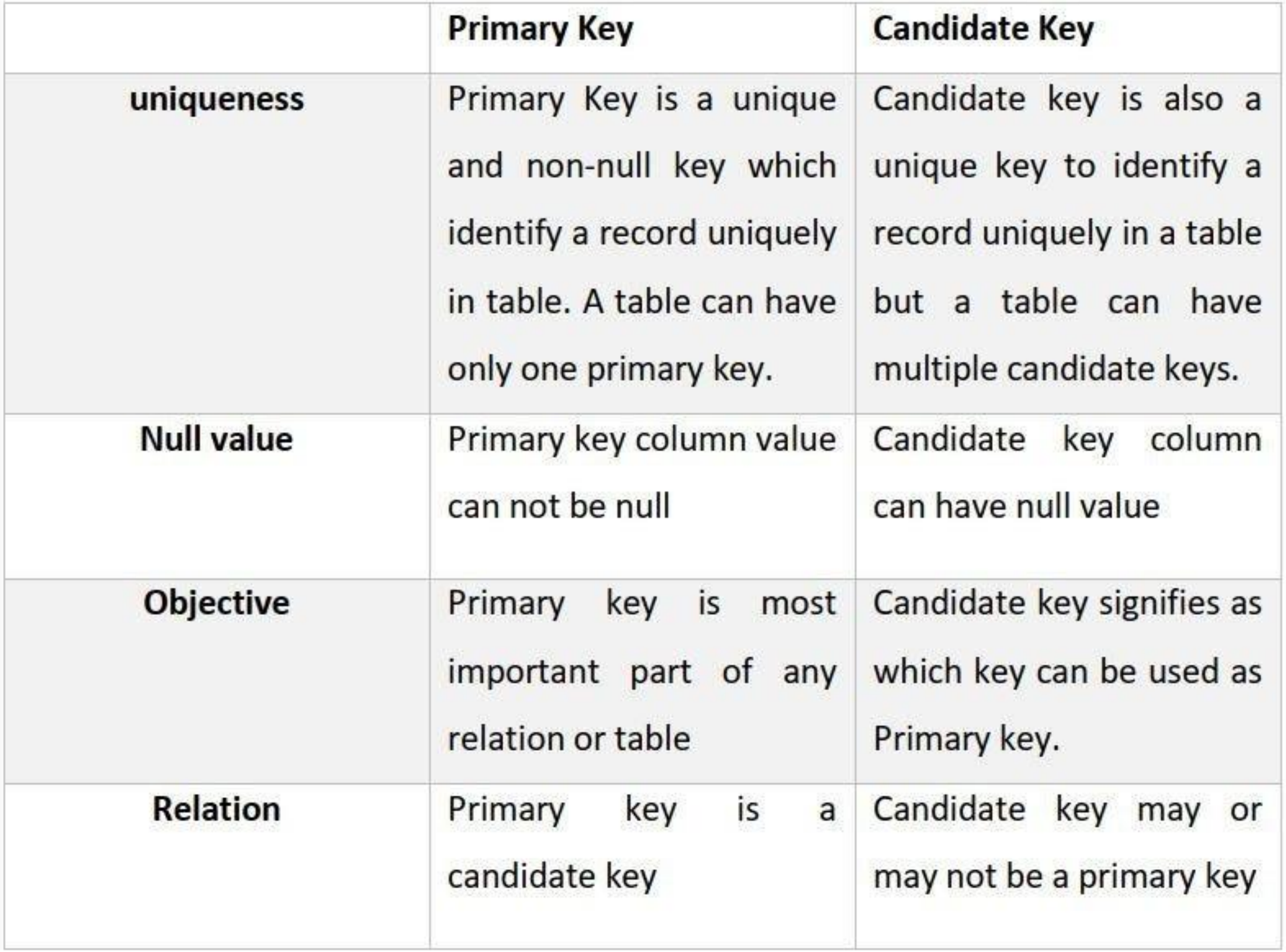
SUPER KEY	CANDIDATE KEY
A set of one or more attributes which can uniquely identify a row in a table	A super key with no redundant attributes
Do not depend on other keys	All candidate keys are super keys
In a student table with columns id, name and phone, the super keys are id, id and name, phone	id, phone are the candidate keys

Visit www.PEDIAA.com



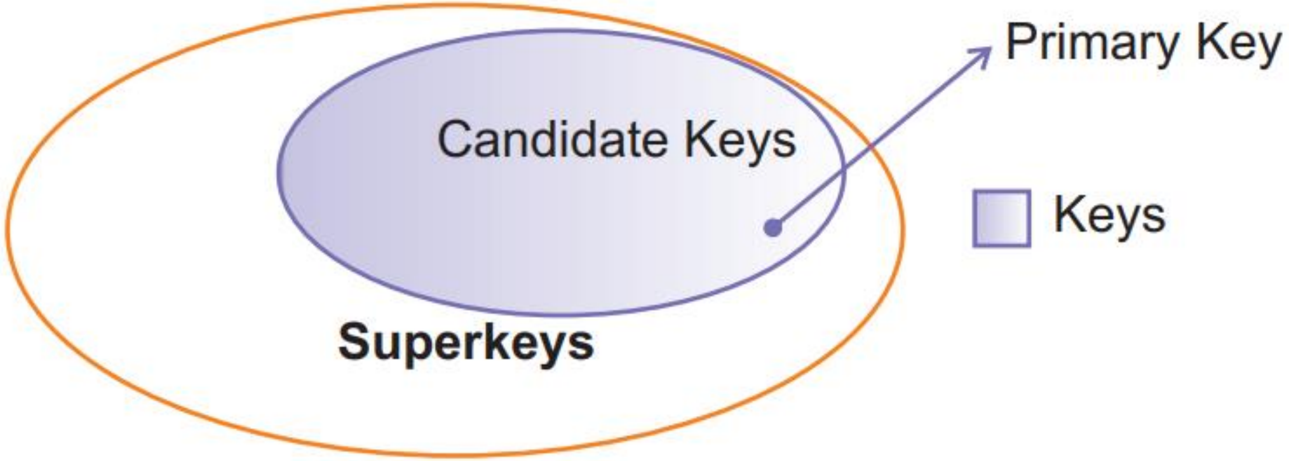
PRIMARY KEY VERSUS CANDIDATE KEY

PRIMARY KEY	CANDIDATE KEY
A minimal set of attributes (columns) in a table that uniquely identifies tuples (rows) in that table	Candidate key is a super key with no redundant attributes
Student table with id, name, age and phone, the id is a primary key	id and phone are candidate keys



	Primary Key	Candidate Key
uniqueness	Primary Key is a unique and non-null key which identify a record uniquely in table. A table can have only one primary key.	Candidate key is also a unique key to identify a record uniquely in a table but a table can have multiple candidate keys.
Null value	Primary key column value can not be null	Candidate key column can have null value
Objective	Primary key is most important part of any relation or table	Candidate key signifies as which key can be used as Primary key.
Relation	Primary key is a candidate key	Candidate key may or may not be a primary key

Relational Model Constraints



Q. What is an RDBMS?

a) Database that stores data elements that are not linked
b) Database that accesses data elements that are not linked
c) Database that stores and allows access to data elements that are linked
d) None of the mentioned

Q. What is a relation in RDBMS?

a) Key
b) Table
c) Row
d) Data Types

Q. The term attribute refers to a _____________ of a table.

A. Record
B. Column
C. Tuple
D. Key

Q. For each attribute of a relation, there is a set of permitted values, called the _________ of that attribute.

A. Domain
B. Relation
C. Set
D. Schema

Entity

An entity is a real-world thing which can be distinctly identified like a person, place or a concept. It is an object which is distinguishable from others. If we cannot distinguish it from others then it is an object but not an entity. An entity can be of two types:

Tangible Entity: Tangible Entities are those entities which exist in the real world physically. Example: Person, car, etc.

Intangible Entity: Intangible Entities are those entities which exist only logically and have no physical existence. Example: Bank Account, etc.

Example: If we have a table of a Student (Roll_no, Student_name, Age, Mobile_no) then each student in that table is an entity and can be uniquely identified by their Roll Number i.e Roll_no.

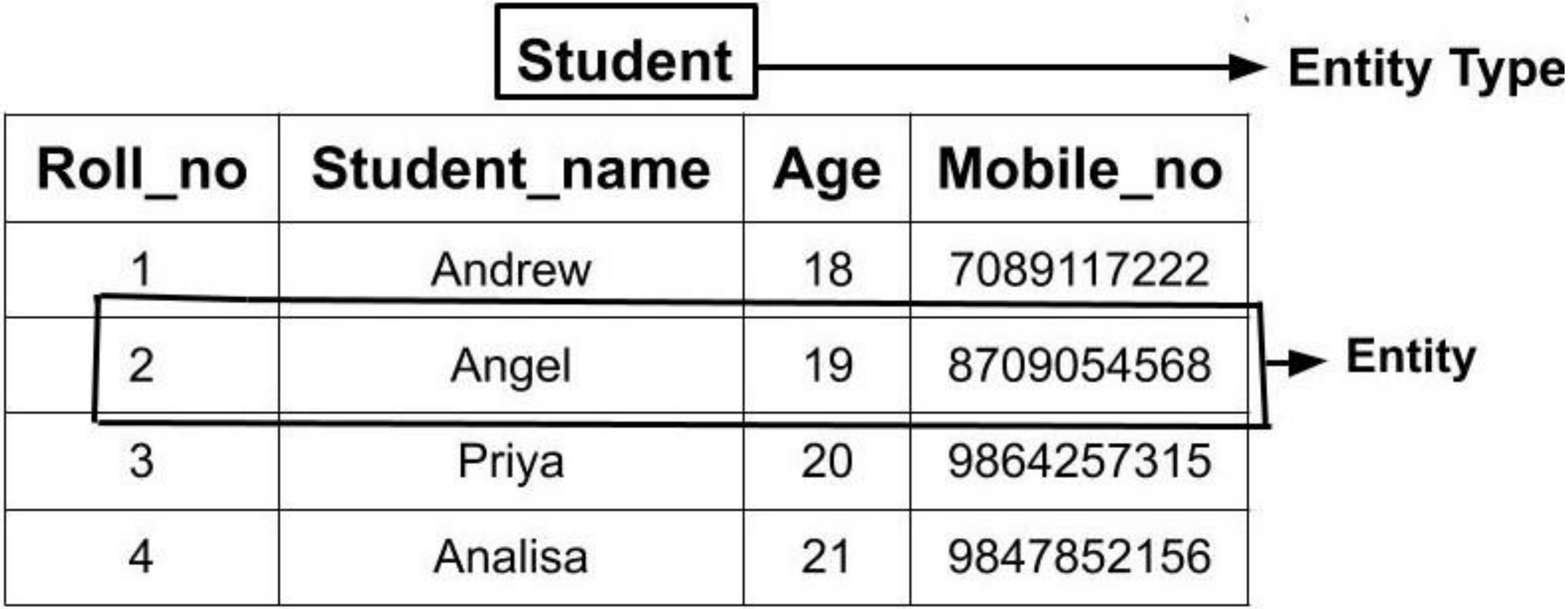
Student

Roll_no	Student_name	Age	Mobile_no
1	Andrew	18	7089117222
2	Angel	19	8709054568
3	Priya	20	9864257315
4	Analisa	21	9847852156

→ Entity

Entity Type

The entity type is a collection of the entity having similar attributes. In the above Student table example, we have each row as an entity and they are having common attributes i.e each row has its own value for attributes Roll_no, Age, Student_name and Mobile_no. So, we can define the above STUDENT table as an entity type because it is a collection of entities having the same attributes. So, an entity type in an ER diagram is defined by a name(here, STUDENT) and a set of attributes(here, Roll_no, Student_name, Age, Mobile_no).



Student → Entity Type

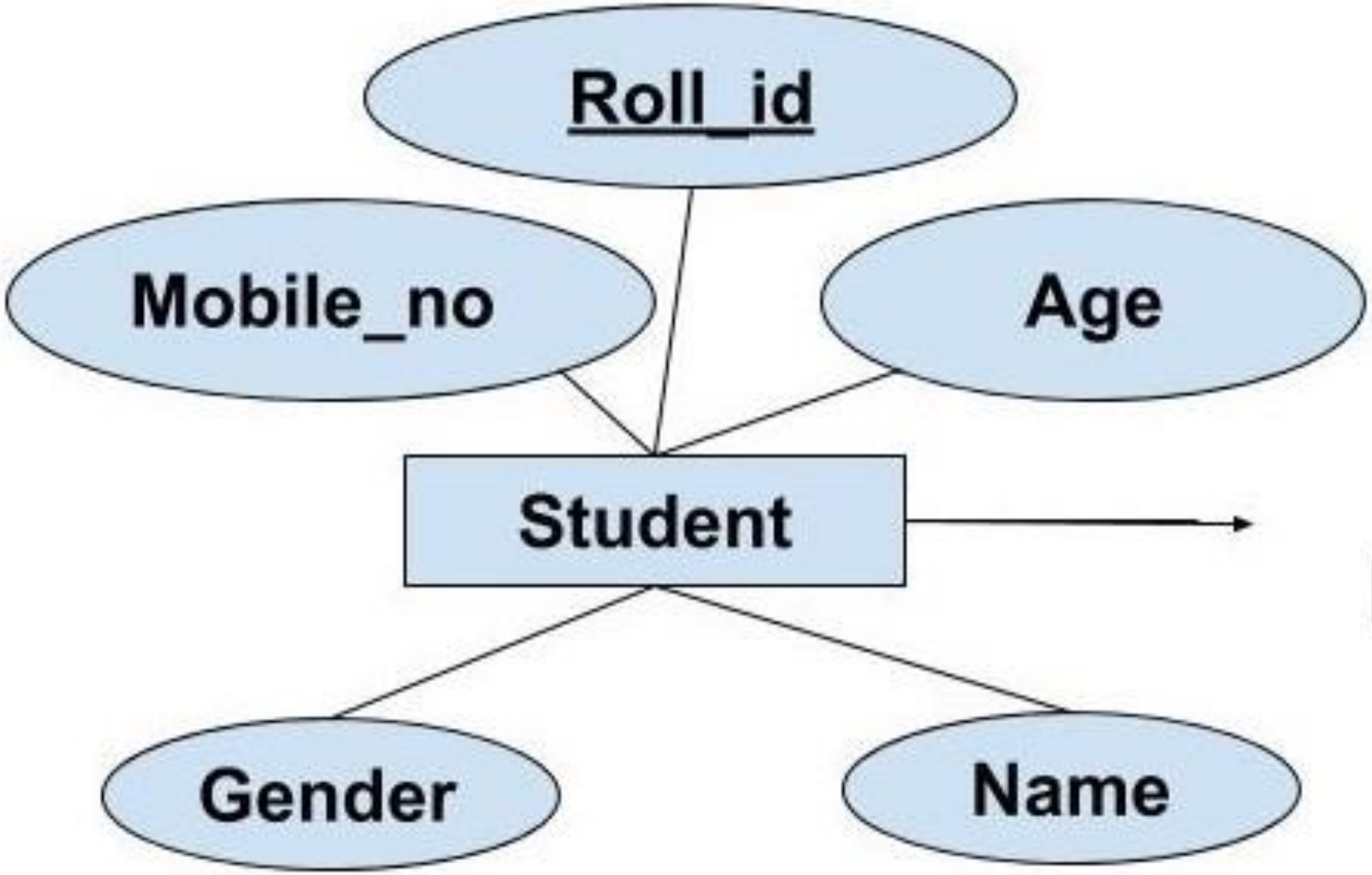
Roll_no	Student_name	Age	Mobile_no
1	Andrew	18	7089117222
2	Angel	19	8709054568
3	Priya	20	9864257315
4	Analisa	21	9847852156

→ Entity

Types of Entity type

1. Strong Entity Type
2. Weak Entity Type

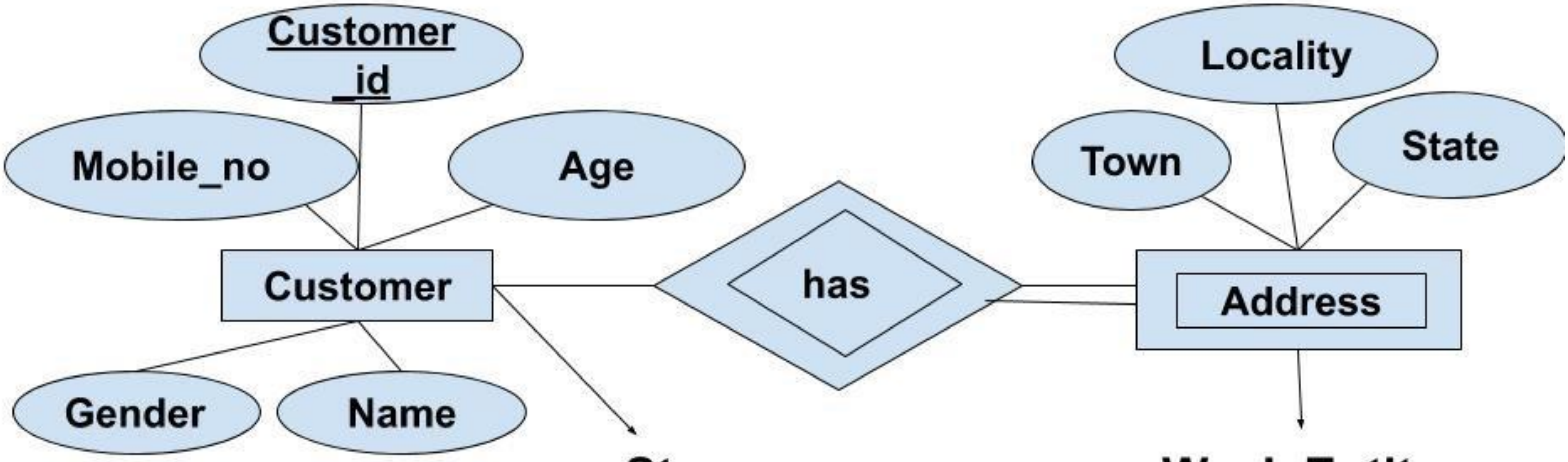
- Strong Entity Type: Strong entity are those entity types which has a key attribute.
- The primary key helps in identifying each entity uniquely. It is represented by a rectangle. In the above example, Roll_no identifies each element of the table uniquely and hence, we can say that STUDENT is a strong entity type.



Strong
EntityType

Weak Entity Type : Weak entity type doesn't have a key attribute. Weak entity type can't be identified on its own. It depends upon some other strong entity for its distinct identity.

- ✓ A weak entity is represented by a double outlined rectangle.
- ✓ The relationship between a weak entity type and strong entity type is called an identifying relationship and shown with a double outlined diamond instead of a single outlined diamond.



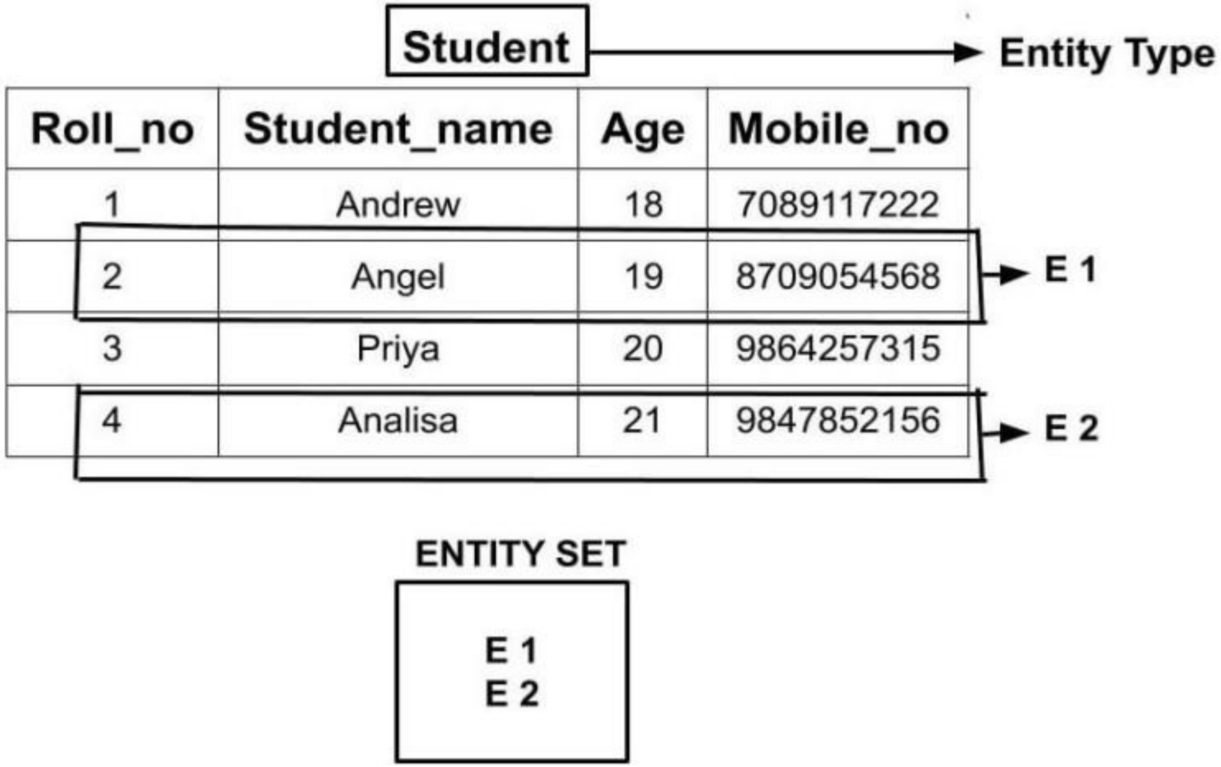
Strong EntityType

Weak Entity Type

Entity Set

Entity Set is a collection of entities of the same entity type. In the above example of STUDENT entity type, a collection of entities from the Student entity type would form an entity set. We can say that entity type is a superset of the entity set as all the entities are included in the entity type.

Example : In the below example, two entities E1 (2, Angel, 19, 8709054568) and E2(4, Analisa, 21, 9847852156) form an entity set.

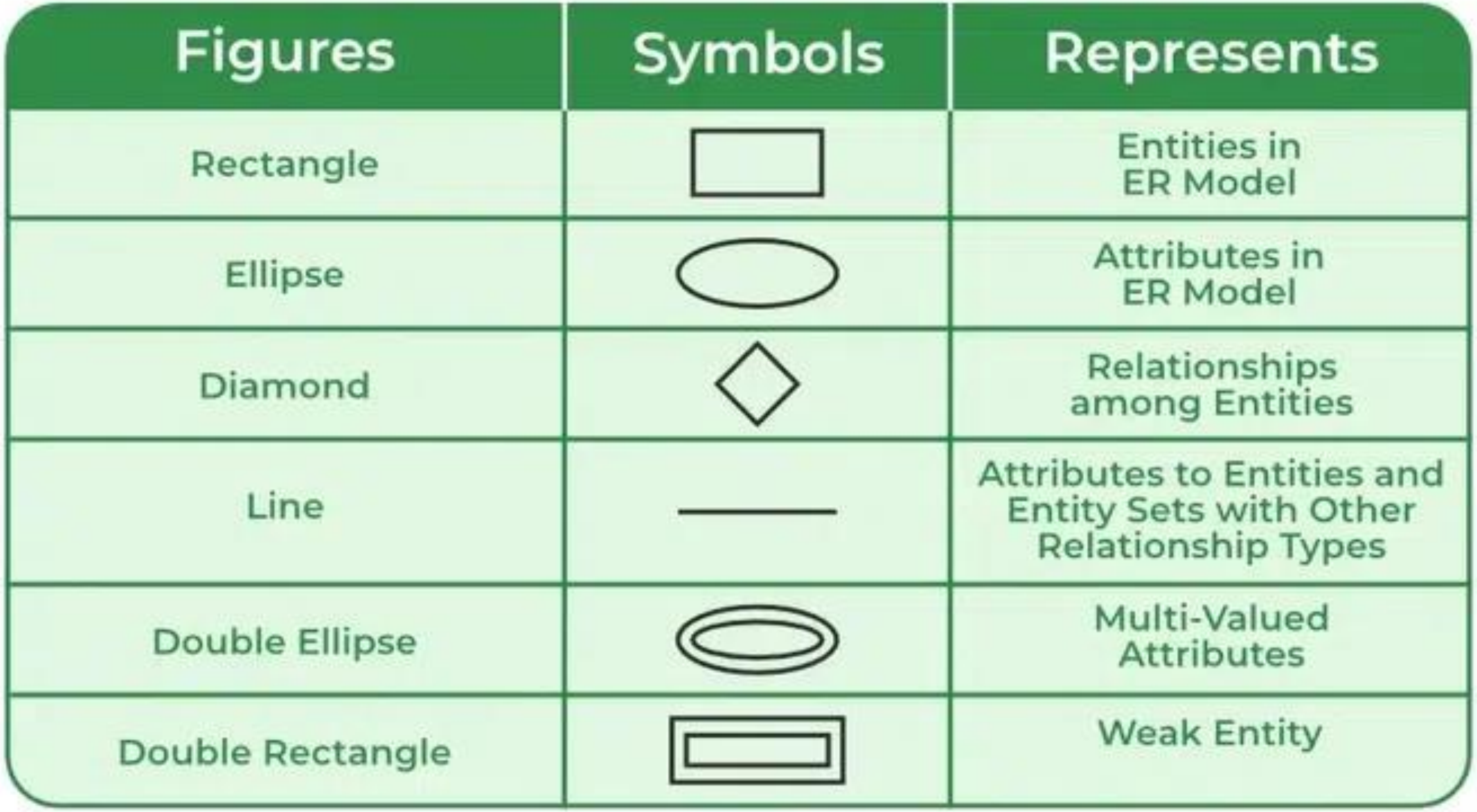


Roll_no	Student_name	Age	Mobile_no
1	Andrew	18	7089117222
2	Angel	19	8709054568
3	Priya	20	9864257315
4	Analisa	21	9847852156

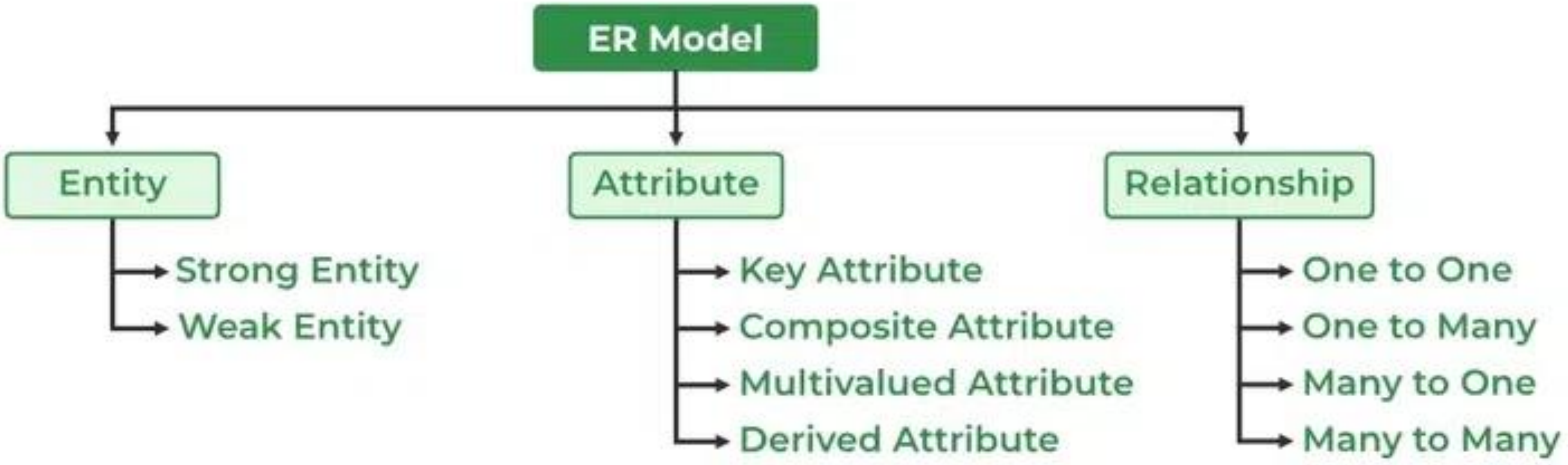
Introduction of ER Model

The Entity Relational Model is 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.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.



Figures	Symbols	Represents
Rectangle	▭	Entities in ER Model
Ellipse	⬭	Attributes in ER Model
Diamond	◇	Relationships among Entities
Line	—	Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse	◎	Multi-Valued Attributes
Double Rectangle	▣	Weak Entity



ER Model

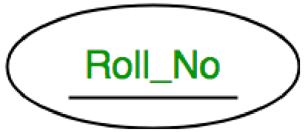
- Entity
  - Strong Entity
  - Weak Entity
- Attribute
  - Key Attribute
  - Composite Attribute
  - Multivalued Attribute
  - Derived Attribute
- Relationship
  - One to One
  - One to Many
  - Many to One
  - Many to Many

Attributes

Attributes are the properties that define the entity type. For example, Roll_No, Name, DOB, Age, Address, and Mobile_No are the attributes that define entity type Student. In ER diagram, the attribute is represented by an oval.

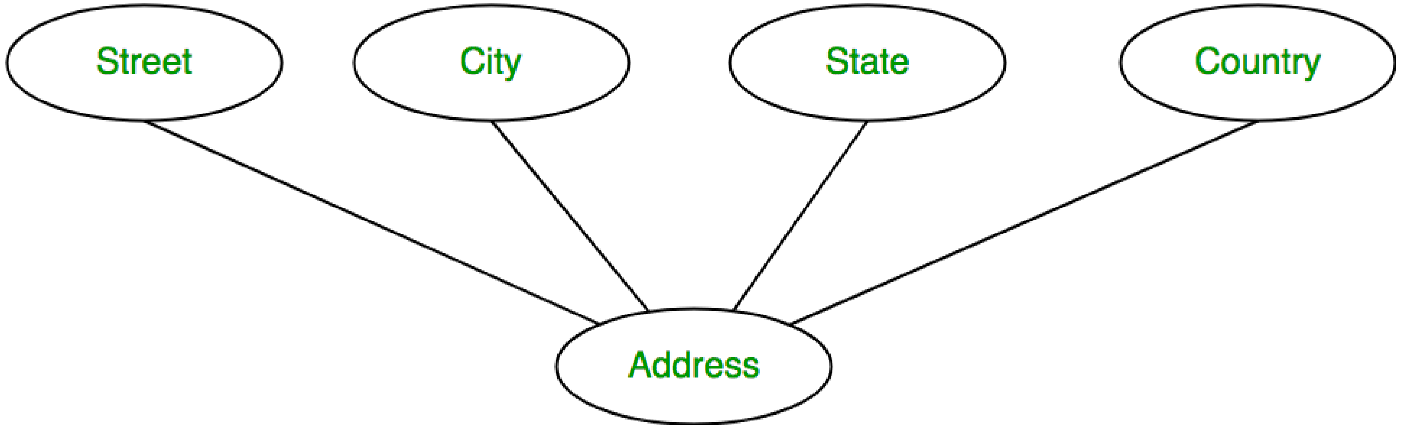
1. Key Attribute

The attribute which uniquely identifies each entity in the entity set is called the key attribute. For example, Roll_No will be unique for each student. In ER diagram, the key attribute is represented by an oval with underlying lines.



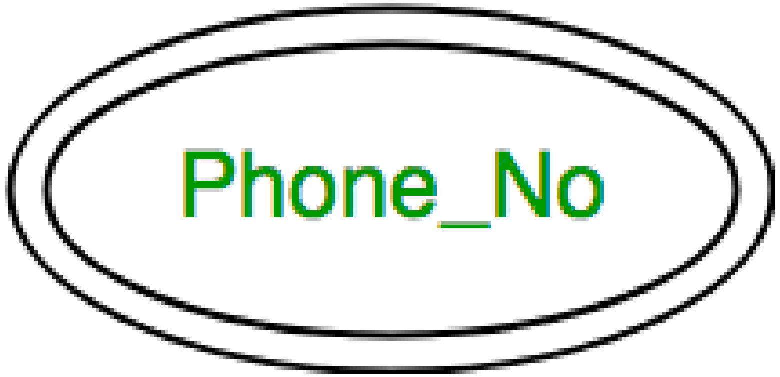
2. Composite Attribute

An attribute composed of many other attributes is called a composite attribute. For example, the Address attribute of the student Entity type consists of Street, City, State, and Country. In ER diagram, the composite attribute is represented by an oval comprising of ovals.



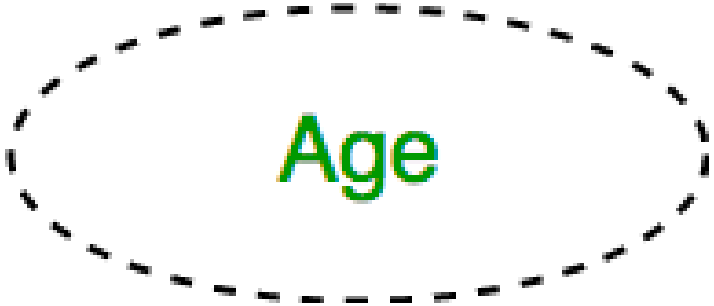
3. Multivalued Attribute

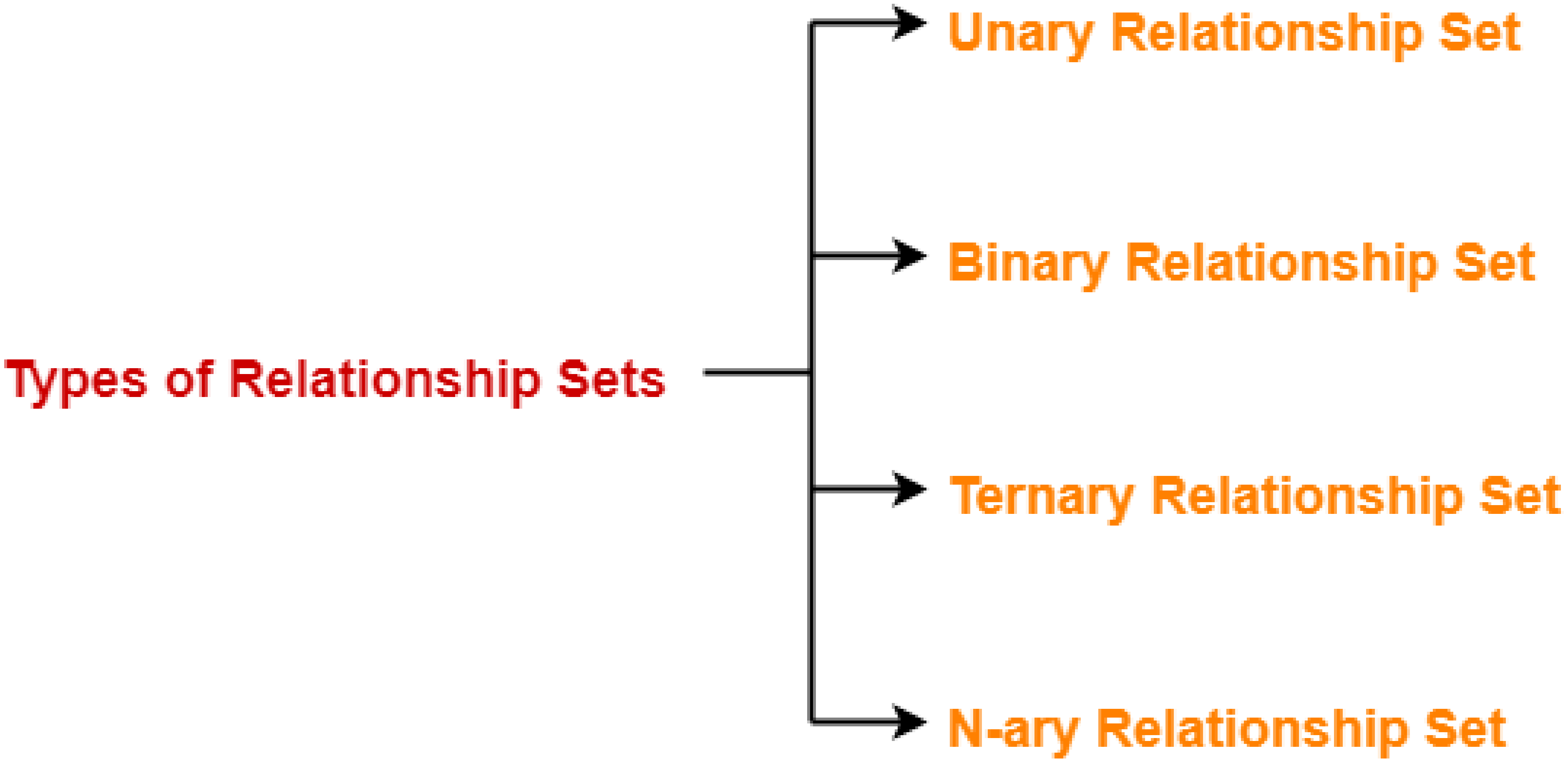
An attribute consisting of more than one value for a given entity. For example, Phone_No (can be more than one for a given student). In ER diagram, a multivalued attribute is represented by a double oval.



4. Derived Attribute

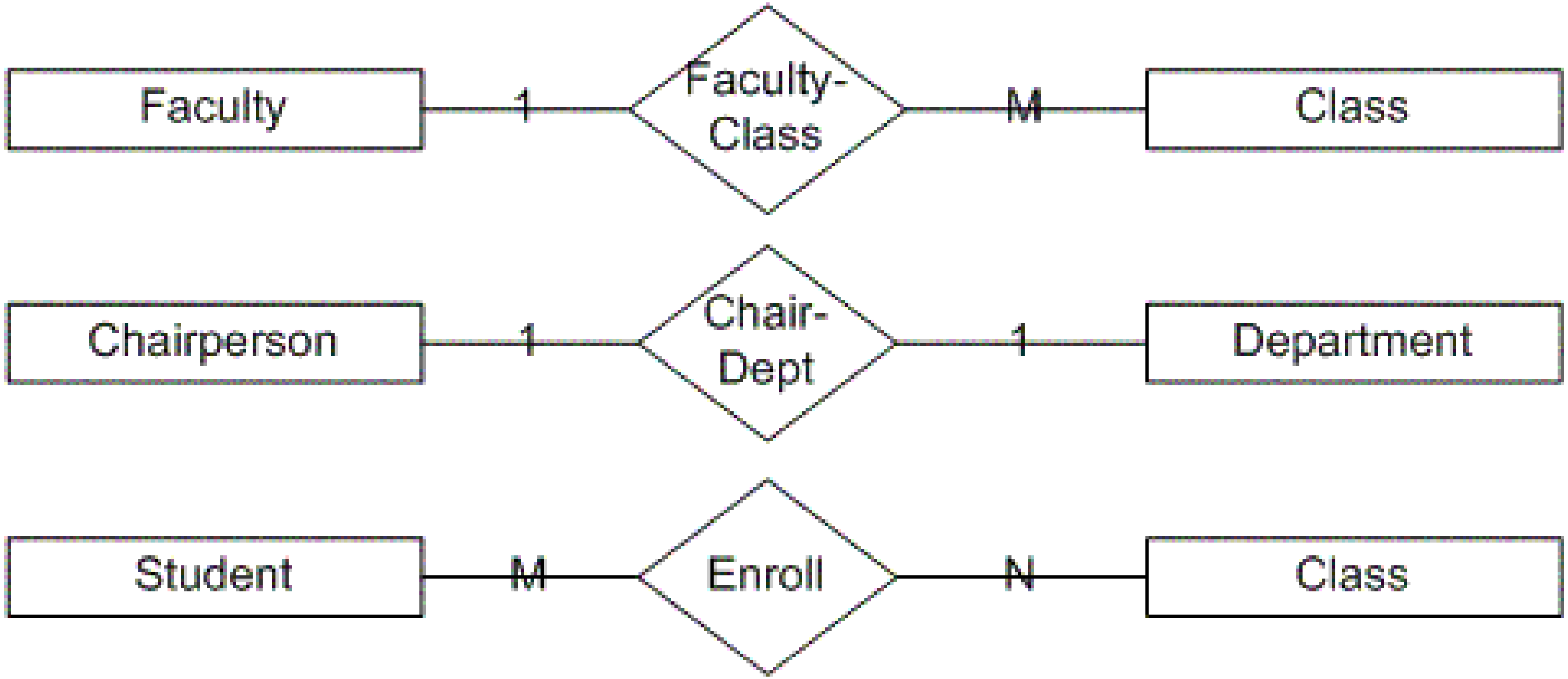
An attribute that can be derived from other attributes of the entity type is known as a derived attribute. e.g.; Age (can be derived from DOB). In ER diagram, the derived attribute is represented by a dashed oval.





Types of Relationship Sets

- Unary Relationship Set
- Binary Relationship Set
- Ternary Relationship Set
- N-ary Relationship Set



Faculty 1 Faculty-Class M Class

Chairperson 1 Chair-Dept 1 Department

Student M Enroll N Class

Method 1: One: 1

Many M,N