



1. Define the following terms.

a) Entity,

b) entity set.

c) attribute

d) tuple.

e) domain.

f) key

g) null

### **a) Entity**

An entity is a thing or object that exists, like a student or a product. It's like a single item in a database.

### **b) Entity Set**

An entity set is a group of similar entities, like all students or all products. It's like a table in a database that contains many similar items.

### **c) Attribute**

An attribute is a characteristic or feature of an entity, like a student's name or age. It's like a column in a database table.

### **d) Tuple**

A tuple is a single row in a database table, representing one entity or item.

### **e) Domain**

A domain is the set of possible values for an attribute, like all possible ages or names.

### **f) Key**

A key is a unique identifier for each tuple or row in a database table, like a student ID.

### **g) Null**

Null means a value is missing or unknown, like if someone's age or address isn't known.

## 2. What is different types of keys, and what is the true

1. **Primary Key (PK)**
2. **Candidate Key (CK)**
3. **Foreign Key (FK)**
4. **Super Key (SK)**
5. **Composite Key (CoK)**
6. **Alternate Key (AK)**

## **1. Primary Key (PK)**

- Unique key in a table
- No null values
- No duplicate values
- Eg: Student ID

## **2. Candidate Key (CK)**

- Unique key in a table
- Multiple candidate keys possible
- Eg: Student ID, Roll Number

## **3. Foreign Key (FK)**

- Primary key of another table in this table
- Establishes relationship
- Eg: Order table's Customer ID (Customer table's PK)

#### **4. Super Key (SK)**

- Unique key in a table
- Includes primary key and foreign key
- Eg: Student ID, Student Name + Student ID

#### **5. Composite Key (CoK)**

- Combination of multiple columns as unique key
- Eg: Order table's Order ID + Product ID

#### **6. Alternate Key (AK)**

- Alternative to candidate key
- Candidate key other than primary key
- Eg: Roll Number (alternative to Student ID).

### **3. Discuss different types of relationship and example.**

#### **1. One-to-One Relationships (1:1)**

- One row links to only one row**
- Eg: Customer - Address**
- RDBMS:**

**Customers(CustID)**

**Addresses(AddrID, CustID)**



## **2. \*One-to-Many (1:N)\***

- One row links to many rows**
- Eg: Customer - Orders**
- RDBMS:**

**Customers(CustID)**

**Orders(OrderID, CustID)**

### **3. \*Many-to-One (N:1)\***

- Many rows link to one row**
- Eg: Orders - Customer**
- RDBMS:**

**Customers(CustID)**

**Orders(OrderID, CustID)**

#### **4. \*Many-to-Many (N:N)\***

- Many rows link to many rows**
- Eg: Students - Courses**
- RDBMS:**

**Students(StudID)**

**Courses(CourseID)**

**Enrollment(StudID, CourseID)**

## **4.State the difference.**

- \* Union and intersection**
- \* project and join**
- \* selection and rejection**

## **1. Union vs Intersection**

**Union: Combines 2 tables (allows duplicates)**

**Intersection: Common rows between tables (no duplicates)**

## **2. Project vs Join**

**Project: Selects specific columns**

**Join: Combines tables based on related columns**

### **3. Selection vs Rejection**

**Selection: Picks rows based on conditions**

**Rejection: Opposite, picks rows NOT matching condition.**

## 5. Define the following terms

**a) partial dependency**

**b) Transitive dependency**

**c) Normalization**

**d) Data anomalies**

**e) Cardinality**

## **1. Partial Dependency**

**When a column depends on only part of a table's key**

## **2. Transitive Dependency**

**When a column depends on another column, not the table's key**

## **3. Normalization**

**Organizing data to reduce duplicates and dependencies**



## **4. Data Anomalies**

**Errors in data due to bad design (insertion, deletion, update)**

## **5. Cardinality**

**Number of rows in a relationship between tables (1:1, 1:N, N:N)**