

#1

Ans:

### **Primary Key:**

A primary key is a unique identifier for each record in a database table. It ensures that each row can be uniquely identified and serves as a reference point for other related tables. Here's an example:

Let's say we have a table called "Students" with the following columns:

- StudentID (Primary Key)
- Name
- Age
- Address

In this case, "StudentID" would be the primary key as it uniquely identifies each student in the table.

### **Composite Primary Key:**

A composite primary key is a primary key that consists of multiple columns, where each column, by itself, may not be unique, but their combination creates a unique identifier. Here's an example:

Consider a table called "Orders" with the following columns:

- OrderID (Primary Key)
- ProductID (Primary Key)
- Quantity

In this case, both "OrderID" and "ProductID" together form the composite primary key. It means each order can have multiple rows with different products, but the combination of OrderID and ProductID will be unique for each row.

### **Foreign Key:**

A foreign key is a field or a set of fields in a table that links to the primary key of another table. It establishes a relationship between two tables, allowing data in one table to refer to data in another table. Here's an example:

We have two tables: "Orders" and "Customers."

- Orders table:
  - OrderID (Primary Key)
  - OrderDate
  - CustomerID (Foreign Key)
  - TotalAmount

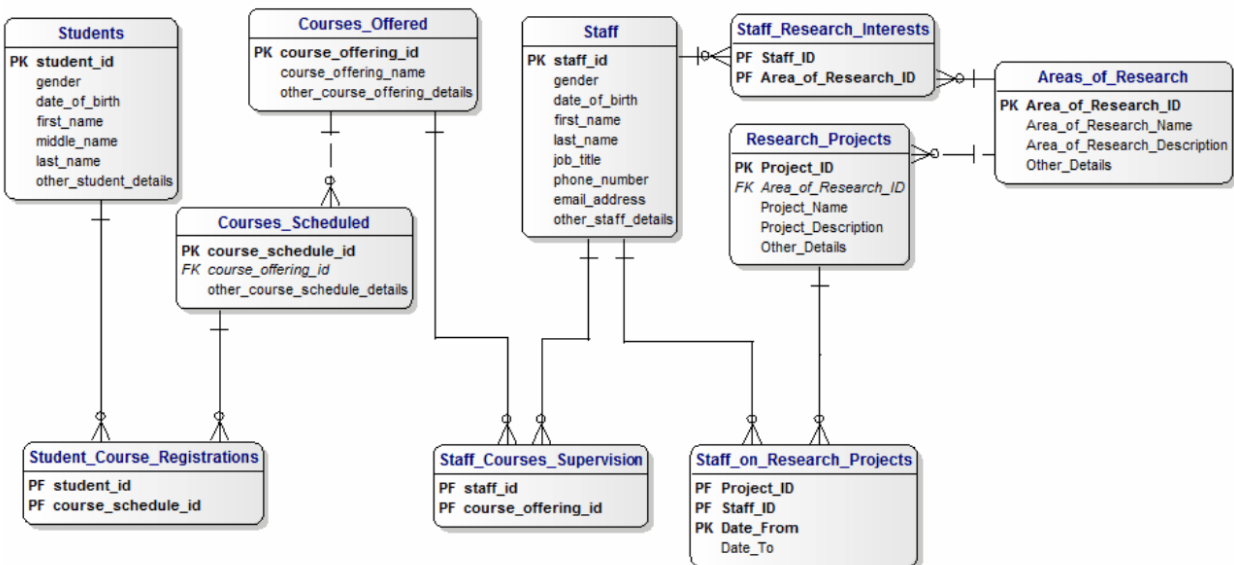
- Customers table:
  - CustomerID (Primary Key)
  - Name
  - Email
  - Address

In this example, the "CustomerID" in the "Orders" table is a foreign key that references the primary key "CustomerID" in the "Customers" table. It creates a link between orders and their respective customers. Each order is associated with a customer by referring to their unique "CustomerID" from the "Customers" table.

These are some basic examples of primary keys, composite primary keys, and foreign keys in a relational database. They are fundamental concepts in database design, ensuring data integrity and establishing relationships between different entities.

#2

Ans:



#3

Ans:

CREATE TABLE User (

Username VARCHAR(50) PRIMARY KEY,

Email VARCHAR(100),

```
First Name VARCHAR(50),  
Last Name VARCHAR(50),  
Phone No VARCHAR(20),  
Address VARCHAR(200),  
HSC Result FLOAT,  
Date of Birth DATE,  
Age INT,  
Password VARCHAR(100)  
);
```

#4

Ans:

```
CREATE TABLE User (  
    Username VARCHAR(50) PRIMARY KEY,  
    Email VARCHAR(100) UNIQUE NOT NULL,  
    First Name VARCHAR(50) NOT NULL,  
    Last Name VARCHAR(50) NOT NULL,  
    Phone No VARCHAR(20) UNIQUE,  
    Address VARCHAR(200),  
    HSC Result FLOAT CHECK ( HSC Result >= 0 AND HSC Result <= 100),  
    Date of Birth DATE,  
    Age INT,
```

Password VARCHAR(100) NOT NULL

);

#5

Ans:

#### **Disadvantages of Redundancy in Database Design:**

1. **Wasted Storage Space:** Redundant data consumes additional storage space, leading to increased storage costs and inefficiency.
2. **Data Inconsistency:** Redundancy can cause data inconsistencies when the same information is stored in multiple places. If the data is updated in one location but not in another, it can lead to conflicting or outdated information.
3. **Increased Maintenance Complexity:** Redundant data requires more effort to maintain and keep consistent across the database, leading to higher maintenance complexity.
4. **Data Integrity Issues:** Redundancy can lead to data integrity problems, as there is a higher chance of data being inaccurately duplicated or deleted.
5. **Slow Performance:** With redundant data, more data needs to be retrieved and updated, which can lead to slower performance in data retrieval and processing.

#### **Disadvantages of Incompleteness in Database Design:**

1. **Missing Information:** Incomplete data means missing critical information, which can result in incomplete or inaccurate analysis and decision-making.
2. **Reduced Data Quality:** Incompleteness can lead to lower data quality, making it difficult to trust and rely on the data for making informed decisions.
3. **Limited Analysis:** Incomplete data may limit the scope of analysis and reporting, hindering comprehensive insights and understanding of the underlying processes.
4. **Inability to Enforce Constraints:** Incomplete data may prevent the application of certain constraints, making it challenging to ensure data integrity and validity.
5. **Difficulty in Validation:** Incompleteness can make it harder to validate and verify data accuracy, leading to uncertainties and doubts about the data's reliability.
6. **Inefficient Data Retrieval:** Incomplete data may require additional efforts to retrieve or complete missing information, causing delays in data retrieval processes.

Overall, both redundancy and incompleteness can negatively impact data management, data quality, and the overall efficiency and effectiveness of a database system. Proper database design practices, such as normalization and data validation, are essential to minimize these disadvantages and ensure a well-structured and reliable database.

#6

Ans:

```
SELECT *  
FROM HRDatabase  
WHERE LastName LIKE 'k%' OR LastName LIKE '%k';
```

#7

Ans:

```
SELECT E1.*  
FROM Employees E1  
INNER JOIN Employees E2 ON E1.ManagerID = E2.EmployeeID  
WHERE E1.Salary > E2.Salary;
```

#8

Ans:

```
SELECT E.EmployeeName, D.DepartmentName  
FROM Employees E  
INNER JOIN Departments D ON E.DepartmentID = D.DepartmentID;
```