Q: 01 - Write the difference between Primary Key and Composite Primary Key

Answer :: Difference between Primary Key and Composite Primary Key

Primary Key: A primary key is a single field (or column) in a table that uniquely identifies each row in that table. It cannot accept null values and must contain unique values. A primary key is used to ensure that each record in the table is unique and to establish a link between tables in a relational database.

Composite Primary Key: A composite primary key, also called a composite key, is a combination of two or more columns to form a primary key for a table. It is used when no single field can uniquely identify each record. A composite key ensures uniqueness by the combination of values in these fields.

Q: 02 - Write the difference between using JOIN Query and not using JOIN query

Answer :: Difference between using JOIN Query and not using JOIN query

Using JOIN Query: JOIN queries are used to retrieve data from two or more related tables based on a related column between them. It allows you to combine rows from two or more tables, based on a related column between them, which helps in fetching data that is distributed across multiple tables.

Not Using JOIN Query: Without using JOIN, you can still retrieve data from multiple tables, but it would require separate queries or subqueries for each table and then manually combining the results. This approach is less efficient and can lead to more complex queries when trying to retrieve related data from multiple tables.

Q: 03 - Answer:: Employees Table

```
CREATE TABLE Employees (
    First_Name VARCHAR(255),
    Last_Name VARCHAR(255),
    Date_of_Birth DATE,
    Department_Id INT,
    Salary DECIMAL(10, 2),
    PRIMARY KEY (First_Name, Last_Name),
    FOREIGN KEY (Department_Id) REFERENCES Departments(Department_Id)
);
```

Departments table:

```
CREATE TABLE Departments (
    Department_Id INT AUTO_INCREMENT,
    Department_Name VARCHAR(255),
    PRIMARY KEY (Department_Id)
);
```

Q: 04 - Write SQL Query to get the second max salary
Answer ::

```
SELECT MAX(Salary) AS SecondMaxSalary
FROM Employees
WHERE Salary < (SELECT MAX(Salary) FROM Employees);
```

Q: 05 - Write SQL Query to show the department names and the average salary of the departments
Answer ::

```
SELECT D.Department_Name, AVG(E.Salary) AS AverageSalary
FROM Departments D
JOIN Employees E ON D.Department_Id = E.Department_Id
GROUP BY D.Department_Name;
```

Q: 06 - Illustrate the INNER, LEFT, RIGHT, SELF Joins
Answer ::

INNER JOIN CODE::

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

LEFT JOIN (LEFT OUTER JOIN)

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
LEFT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

RIGHT JOIN (RIGHT OUTER JOIN)

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
RIGHT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

SELF JOIN

```
SELECT A.EmployeeName AS EmployeeName1, B.EmployeeName AS EmployeeName2
FROM Employees A, Employees B
WHERE A.EmployeeID <> B.EmployeeID AND A.DepartmentID = B.DepartmentID;
```

Q: 07 - What is a subquery? Write with an example

Answer ::

```
SELECT EmployeeName
FROM Employees
WHERE DepartmentID = (SELECT DepartmentID
                      FROM Departments
                      WHERE DepartmentName = 'IT');
```

Q: 08 - Show the names of the employees who get less salary than Steven

Answer ::

```
SELECT E.First_Name, E.Last_Name
FROM Employees E, (SELECT Salary FROM Employees WHERE First_Name = 'Steven') AS S
WHERE E.Salary < S.Salary;
```

Q: 09 - Count the number of employees of each job type

Answer ::

```
SELECT job_id, COUNT(*) AS NumberOfEmployees
FROM Employees
GROUP BY job_id;
```

Q: 10 - Show the names of Departments which doesn’t have any employees

Answer ::

```
SELECT D.Department_Name
FROM Departments D
LEFT JOIN Employees E ON D.Department_Id = E.Department_Id
WHERE E.Department_Id IS NULL;
```