

Name; Avijit Chowdhury  
Email: chowdhuryavijit333@gmail.com

**Answer to the Quest. No: 1**

Primary Key	Composite key
1) Single column	1) Multiple columns
2) Unique row identification with single-column setup.	2) Unique row identification with multiple-column setup.
3) Uses single-column for declaration	3) Uses multiple columns for declaration.
4) Example: employee_id	4) Example: employee_id, order_id

**Answer to the Quest. No: 2**

Join query time complexity is less than the query not using join query. A join query retrieves data simultaneously from different tables based on a common relationship. In contrast, a normal query can recover data from individual tables separately. Join queries can be complex, but more efficient than normal queries not using join queries.

**Answer to the Quest. No: 3**

```
CREATE DATABASE MIDTERM;
-- DROP DATABASE MIDTERM
USE MIDTERM;
CREATE TABLE DEPARTMENTS(
    DEPARTMENT_ID INT PRIMARY KEY,
    DEPARTMENT_NAME VARCHAR(20)
);
CREATE TABLE EMPLOYEES(
    EMPLOYEE_ID INT ,
    FIRST_NAME VARCHAR(20) NOT NULL,
    LAST_NAME VARCHAR(10) NOT NULL,
    DATE_OF_BIRTH DATE NOT NULL,
    DEPARTMENT_ID INT NOT NULL PRIMARY KEY,
    SALARY INT,
    FOREIGN KEY EMPLOYEES(DEPARTMENT_ID) REFERENCES
DEPARTMENTS(DEPARTMENT_ID));
```

**Answer to The Quest. No: 4**

```
USE DUMMYDB;
SELECT *
FROM EMPLOYEES
WHERE SALARY = (SELECT SALARY AS SECOND_HIGEST
                FROM EMPLOYEES
```

ORDER BY SALARY DESC

LIMIT 1  
OFFSET 1);

#### **Answer to the Quest. No: 5**

```
SELECT DEPARTMENT_NAME, (SELECT AVG(SALARY) FROM EMPLOYEES WHERE  
DEPARTMENT_ID=EMPLOYEES.DEPARTMENT_ID) AS AVG_SALARY  
FROM DEPARTMENTS;
```

#### **Answer To The Quest No: 6**

##### **INNER JOIN:**

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

##### **LEFT JOIN:**

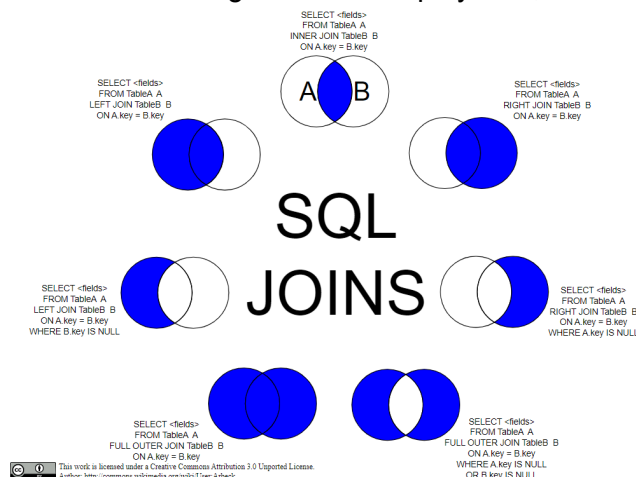
```
SELECT Customers.CustomerID, Customers.CompanyName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

##### **RIGHT JOIN:**

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

##### **SELF JOIN:**

```
SELECT e1.EmployeeID, e1.LastName, e2.LastName AS Manager  
FROM Employees e1  
LEFT JOIN Employees e2 ON e1.ManagerID = e2.EmployeeID;
```



Inner join means join in between two sets. Left means joining with a left and middle edge. Right means joins with the right edge and middle edge. Self means to participate with a self table.

**Answer to the Quest. No: 7**

Subquery means query between another query. More specifically, A subquery is a query that is nested within another query.

**Example:**

```
-- THOSE EMPLOYEES GET THE SALARY GRETER THAN MANAGER
SELECT *
FROM EMPLOYEES AS EMP
WHERE SALARY > (SELECT SALARY
                  FROM EMPLOYEES AS MGR
                  WHERE EMP.MANAGER_ID = MGR.EMPLOYEE_ID
                  );
```

**Answer to the Quest. No: 8**

```
SELECT *
FROM EMPLOYEES
WHERE SALARY < (SELECT SALARY
                  FROM EMPLOYEES
                  WHERE FIRST_NAME ='STEVEN' AND LAST_NAME = 'KING');
```

**Answer to the Quest. No: 9**

```
SELECT COUNT(*)
FROM EMPLOYEES;
```

**Answer to the Quest. No: 10**

```
SELECT DEPARTMENTS.DEPARTMENT_NAME
FROM DEPARTMENTS
LEFT JOIN EMPLOYEES
ON DEPARTMENTS.DEPARTMENT_ID = EMPLOYEES.DEPARTMENT_ID
WHERE EMPLOYEES.DEPARTMENT_ID IS NULL;
```