Mid Term Exam (Solution)

1. **Primary key:** Uniquely identifies each record in a table. Example: ID, Phone number, email etc.
   Foreign key: Creates a relationship between two tables by referencing the primary key in another table.
2. **Self join:** Self Join in SQL is a join where a table is joined with itself. It is used when rows compare within the same table.

```
SELECT e.FIRST_NAME, m.MANAGER_ID
FROM EMPLOYEES AS e
    JOIN EMPLOYEES AS m
        ON e.EMPLOYEE_ID = m.MANAGER_ID;
```

3.
```
CREATE TABLE Employees(
   Employee_ID CHAR(10) PRIMARY KEY,
   First_Name VARCHAR(30),
   Last_Name VARCHAR(30),
   Date_of_Birth VARCHAR(30),
   Department_Id CHAR(10),
   Salary NUMERIC(10,3)
);
CREATE TABLE Projects(
   Project_ID CHAR(10) PRIMARY KEY,
   Project_Name VARCHAR(60),
   Start_Date DATE,
   End_Date DATE,
   Budget NUMERIC(10,3)
);
CREATE TABLE Employee_Projects(
   Employee_ID CHAR(10),
   Project_ID CHAR(10),
   PRIMARY KEY(Employee_ID, Project_ID),
   FOREIGN KEY(Employee_ID) REFERENCES
                                    Employees(Employee_ID),
   FOREIGN KEY(Project_ID) REFERENCES Projects(Project_ID));
```

4.
```
SELECT SALARY
FROM EMPLOYEES
WHERE SALARY = (
        SELECT MAX(SALARY)
        FROM EMPLOYEES
    WHERE SALARY < (
                SELECT MAX(SALARY)
                FROM EMPLOYEES
                WHERE SALARY < (
                        SELECT MAX(SALARY)
                        FROM EMPLOYEES
                )
        )
);
```
5.
```
SELECT D.DEPARTMENT_NAME,
        COUNT(E.EMPLOYEE_ID) AS EMPLOYEE_COUNT
FROM DEPARTMENTS AS D
        JOIN EMPLOYEES AS E
                ON D.DEPARTMENT_ID = E.DEPARTMENT_ID
GROUP BY D.DEPARTMENT_NAME;
```
6. **INNER JOIN:** Returns only the rows that have matching values in both tables.

   EX:
   ```
   SELECT E.FIRST_NAME, D.DEPARTMENT_NAME
   FROM EMPLOYEES AS E
           INNER JOIN DEPARTMENTS AS D
                   ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;
   ```
   LEFT JOIN: Returns all rows from the left table and matched rows from the right table. If there is no match, return NULL.

   EX:
   ```
   SELECT E.FIRST_NAME, D.DEPARTMENT_NAME
   FROM EMPLOYEES AS E
           LEFT JOIN DEPARTMENTS AS D
                   ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;
   ```
   RIGHT JOIN: Returns all rows from the right table and matched rows from the left table. If there is no match, return NULL.

EX: SELECT E.FIRST_NAME, D.DEPARTMENT_NAME
FROM EMPLOYEES AS E
RIGHT JOIN DEPARTMENTS AS D
ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;

CROSS JOIN: Returns the Cartesian product of the two tables

EX: SELECT E.FIRST_NAME, D.DEPARTMENT_NAME
FROM EMPLOYEES AS E
CROSS JOIN DEPARTMENTS AS D
ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;

7. **Common Table Expression (CTE):** Save the output of a query under any name is called CTE.

   EX:
```
WITH CTE_EX AS (
    SELECT FIRST_NAME
    FROM EMPLOYEES
    WHERE SALARY > (
        SELECT AVG(SALARY)
        FROM EMPLOYEES
    )
)
SELECT *
FROM CTE_EX;
```

8.
```
SELECT FIRST_NAME, LAST_NAME
FROM EMPLOYEES
WHERE SALARY < (
    SELECT SALARY
    FROM EMPLOYEES
    WHERE FIRST_NAME = 'Steven' AND LAST_NAME = 'King'
);
```

9.
```
SELECT D.DEPARTMENT_NAME, E.FIRST_NAME
                                    AS MANAGER_NAME
FROM DEPARTMENTS AS D
    JOIN EMPLOYEES AS E
        ON D.MANAGER_ID = E.EMPLOYEE_ID;
```

10.
```
SELECT DISTINCT L.CITY
FROM DEPARTMENTS AS D
        JOIN LOCATIONS AS L
            ON D.LOCATION_ID = L.LOCATION_ID;
```