

Answer Script

Question No. 1
Write the difference between Primary Key and Composite Primary Key
Answer No. 1
A primary key is a unique identifier based on a single column or field, while a composite primary key is a unique identifier based on a combination of multiple columns or fields.

Question No. 2
Write the difference between using JOIN Query and not using JOIN query
Answer No. 2
Using a JOIN query allows us to retrieve and combine related data from multiple tables in a structured and efficient manner. It provides flexibility and better performance compared to not using JOIN queries, where combinations are done manually. If multiple tables have unique field names we can easily combine data from multiple tables without using JOIN Query otherwise we have to use JOIN Query.

Question No. 3

Create a table of Employees which has the following fields

- a. First Name
- b. Last Name
- c. Date of Birth
- d. Department Id
- e. Salary

Create a table of Departments which has the following fields

- a. Department Id
- b. Department Name

Create both of the tables using proper constraints

Answer No. 3

```
CREATE TABLE departments(  
    dept_id CHAR(4) PRIMARY KEY,  
    department_name VARCHAR(25)  
);
```

```
CREATE TABLE employees(  
    first_name VARCHAR(50),  
    last_name VARCHAR(50) NOT NULL,  
    date_of_birth DATE,  
    dept_id CHAR(4),  
    salary INT NOT NULL,  
    FOREIGN KEY(dept_id) REFERENCES departments(dept_id)  
);
```

Question No. 4

Write SQL Query to get the second max salary.

Answer No. 4

```
SELECT MAX(salary)
FROM employees
WHERE salary < (
    SELECT MAX(salary)
    FROM employees
);
```

Question No. 5

Write SQL Query to show the department names and the average salary of the departments

Answer No. 5

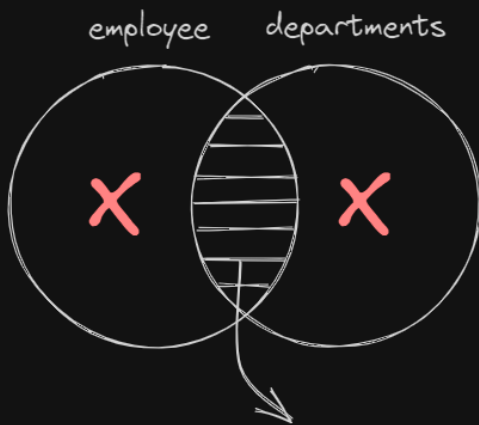
```
SELECT departments.department_name, AVG(employees.salary)
FROM employees JOIN departments
ON employees.department_id = departments.department_id
GROUP BY departments.department_id;
```

Question No. 6

Illustrate the INNER, LEFT, RIGHT, SELF Joins

Answer No. 6

INNER JOIN / JOIN



employee

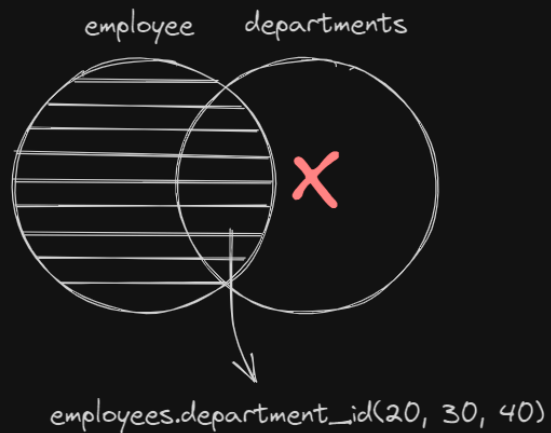
name	department_id
a	20
b	30
c	40

departments

department_id	dept_name
10	w
20	x
30	y

```
SELECT employees.department_id  
FROM employees JOIN departments  
ON employees.department_id = departments.department_id;
```

LEFT JOIN

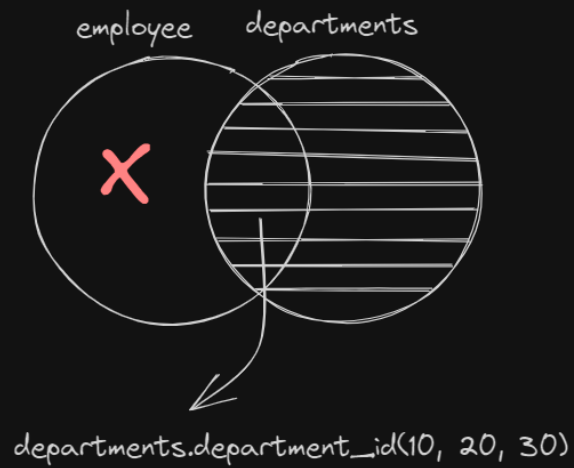


name	department_id
a	20
b	30
c	40

department_id	dept_name
10	w
20	x
30	y

```
SELECT employees.department_id
FROM employees LEFT JOIN departments
ON employees.department_id = departments.department_id;
```

RIGHT JOIN



employee

name	department_id
a	20
b	30
c	40

departments

department_id	dept_name
10	w
20	x
30	y

```
SELECT departments.department_id  
FROM employees RIGHT JOIN departments  
ON employees.department_id = departments.department_id;
```

SELF JOIN \longrightarrow One Table

employee

emp_id	Name	mgr_id
100	Rahim	50
102	Karim	20
104	Sayem	60

```
SELECT m.name  
FROM employees AS n JOIN employees AS m  
ON n.employee_id = m.employee_id;
```

Question No. 7

What is a subquery? Write with an example

Answer No. 7

A subquery is a nested query. It allows us to break down complex data into more manageable data

```
SELECT
    *
FROM
    employees
WHERE
    department_id= (
        SELECT
            department_id
        FROM
            departments
        WHERE
            department_name='Marketing'
    );
```

Question No. 8

Show the names of the employees who get less salary than Steven

Answer No. 8

```
SELECT *
FROM employees
WHERE salary < (
    SELECT salary
    FROM employees
    WHERE first_name = 'STEVEN'
    LIMIT 1
);
```


Question No. 9

Count the number of employees of each job type

Answer No. 9

```
SELECT
  job_id, COUNT(*) AS number_of_emp
FROM
  employees
GROUP BY job_id;
```

Question No. 10

Show the names of Departments which doesn't have any employees

Answer No. 10

```
SELECT department_name
FROM departments LEFT JOIN employees
ON departments.department_id = employees.department_id
GROUP BY department_name
HAVING COUNT(employees.employee_id) = 0;
```