TOP 10 SQL INTERVIEW QUERIES

Practice Dataset

EmpID	EmpName	Gender	Salary	City
1	Arjun	M	75000	Pune
2	Ekadanta	M	125000	Bangalore
3	Lalita	F	150000	Mathura
4	Madhav	M	250000	Delhi
5	Visakha	F	120000	Mathura

Employee Table

EmployeeDetail Table

EmpID	Project	EmpPosition	DOJ
1	P1	Executive	26-01-2019
2	P2	Executive	04-05-2020
3	P1	Lead	21-10-2021
4	P3	Manager	29-11-2018
5	P2	Manager	01-08-2020

Create Tables: Employee and EmployeeDetail

```
CREATE TABLE Employee (
            EmpID int NOT NULL,
            EmpName Varchar,
            Gender Char,
            Salary int,
            City Char(20)  )
--- first run the above code then below code
INSERT INTO Employee
VALUES (1, 'Arjun', 'M', 75000,  'Pune'),
      (2, 'Ekadanta', 'M', 125000, 'Bangalore'),
      (3, 'Lalita', 'F', 150000 , 'Mathura'),
      (4, 'Madhav', 'M', 250000 , 'Delhi'),
      (5, 'Visakha', 'F', 120000 , 'Mathura')
```

```
CREATE TABLE  EmployeeDetail (
            EmpID int NOT NULL,
            Project Varchar,
            EmpPosition Char(20),
            DOJ date  )
--- first run the above code then below code
INSERT INTO EmployeeDetail
VALUES (1, 'P1', 'Executive', '26-01-2019'),
            (2, 'P2', 'Executive', '04-05-2020'),
            (3, 'P1', 'Lead', '21-10-2021'),
            (4, 'P3', 'Manager', '29-11-2019'),
            (5, 'P2', 'Manager', '01-08-2020')
```

Q1(a): *Find the list of employees whose salary ranges between 2L to 3L.*

```
SELECT EmpName, Salary FROM Employee
WHERE Salary > 200000 AND Salary < 300000
                    --- OR ---
SELECT EmpName, Salary FROM Employee
WHERE Salary BETWEEN 200000 AND 300000
```

Q1(b): *Write a query to retrieve the list of employees from the same city.*

```
SELECT E1.EmpID, E1.EmpName, E1.City
FROM Employee E1, Employee E2
WHERE E1.City = E2.City AND E1.EmpID != E2.EmpID
```

Q1(c): *Query to find the null values in the Employee table.*

```
SELECT * FROM Employee
WHERE EmpID IS NULL
```

Q2(a): *Query to find the cumulative sum of employee's salary.*

```
SELECT EmpID, Salary, SUM(Salary) OVER (ORDER BY EmpID) AS CumulativeSum
FROM Employee
```

Q2(b): *What's the male and female employees ratio.*

```
SELECT
  (COUNT(*) FILTER (WHERE Gender = 'M') * 100.0 / COUNT(*)) AS MalePct,
  (COUNT(*) FILTER (WHERE Gender = 'F') * 100.0 / COUNT(*)) AS FemalePct
FROM Employee;
```

Q2(c): *Write a query to fetch 50% records from the Employee table.*

```
SELECT * FROM Employee
WHERE EmpID <= (SELECT COUNT(EmpID)/2 from Employee)
```

If EmpID is not auto-increment field or numeric, then we can use Row NUMBER function

Q3: *Query to fetch the employee's salary but replace the LAST 2 digits with 'XX'*

i.e 12345 will be 123XX

```
SELECT Salary,
CONCAT(SUBSTRING(Salary::text, 1, LENGTH(Salary::text)-2), 'XX') as masked_number
FROM Employee

                    --- OR ---

SELECT Salary, CONCAT(LEFT(CAST(Salary AS text), LENGTH(CAST(Salary AS text))-2), 'XX')
AS masked_number
FROM Employee
```

```
SELECT Salary,
CONCAT(LEFT(Salary, LEN(Salary)-2), 'XX') as masked_salary
FROM Employee
```

MySQL

Q4: *Write a query to fetch even and odd rows from Employee table.*

General Solution using ROW_NUMBER()

```
---Fetch even rows

SELECT * FROM
        (SELECT *, ROW_NUMBER() OVER(ORDER BY EmpId) AS
        RowNumber
        FROM Employee) AS Emp
WHERE Emp.RowNumber % 2 = 0

---Fetch odd rows

SELECT * FROM
        (SELECT *, ROW_NUMBER() OVER(ORDER BY EmpId) AS
        RowNumber
        FROM Employee) AS Emp
WHERE Emp.RowNumber % 2 = 1
```

Alternative Solution

*If you have an auto-increment field like EmpID then we can use the **MOD()** function:*

```
---Fetch even rows

SELECT * FROM Employee
WHERE MOD(EmpID,2)=0;

---Fetch odd rows

SELECT * FROM Employee
WHERE MOD(EmpID,2)=1;
```

Q5(a): *Write a query to find all the Employee names whose name:*

- *Begin with 'A'*
- *Contains 'A' alphabet at second place*
- *Contains 'Y' alphabet at second last place*
- *Ends with 'L' and contains 4 alphabets*
- *Begins with 'V' and ends with 'A'*

```
SELECT * FROM Employee WHERE EmpName LIKE 'A%';

SELECT * FROM Employee WHERE EmpName LIKE '_a%';

SELECT * FROM Employee WHERE EmpName LIKE '%y_';

SELECT * FROM Employee WHERE EmpName LIKE '____l';

SELECT * FROM Employee WHERE EmpName LIKE 'V%a'
```

Q5(b): *Write a query to find the list of Employee names which is:*

- *starting with vowels (a, e, i, o, or u), without duplicates*
- *ending with vowels (a, e, i, o, or u), without duplicates*
- *starting & ending with vowels (a, e, i, o, or u), without duplicates*

```
SELECT DISTINCT EmpName
FROM Employee
WHERE LOWER(EmpName) SIMILAR TO '[aeiou]%'

SELECT DISTINCT EmpName
FROM Employee
WHERE LOWER(EmpName) SIMILAR TO '%[aeiou]'

SELECT DISTINCT EmpName
FROM Employee
WHERE LOWER(EmpName) SIMILAR TO '[aeiou]%[aeiou]'
```

```
SELECT DISTINCT EmpName
FROM Employee
WHERE LOWER(EmpName) REGEXP '^[aeiou]'

SELECT DISTINCT EmpName
FROM Employee
WHERE LOWER(EmpName) REGEXP '[aeiou]$'

SELECT DISTINCT EmpName
FROM Employee
WHERE LOWER(EmpName) REGEXP
'^[aeiou].*[aeiou]$'
```

MySQL Solution: REGEXP

Q6: *Find Nth highest salary from employee table with and without using the TOP/LIMIT keywords.*

General Solution without using TOP/LIMIT

```
SELECT Salary FROM Employee E1
WHERE N-1 = (
        SELECT COUNT( DISTINCT ( E2.Salary ) )
        FROM Employee E2
        WHERE E2.Salary > E1.Salary  );

              --- OR ---

SELECT Salary FROM Employee E1
WHERE N = (
        SELECT COUNT( DISTINCT ( E2.Salary ) )
        FROM Employee E2
        WHERE E2.Salary >= E1.Salary  );
```

Using LIMIT

```
SELECT Salary FROM Employee
ORDER BY Salary DESC
LIMIT 1 OFFSET N-1
```

Using TOP

```
SELECT TOP 1 Salary
FROM Employee
WHERE Salary < (
    SELECT MAX(Salary) FROM Employee)
    AND Salary NOT IN (
    SELECT TOP 2 Salary
    FROM Employee
    ORDER BY Salary DESC)
ORDER BY Salary DESC;
```

Q7(a): *Write a query to find and remove duplicate records from a table.*

```
SELECT EmpID, EmpName, gender, Salary, city,
COUNT(*) AS duplicate_count
FROM Employee
GROUP BY EmpID, EmpName, gender, Salary, city
HAVING COUNT(*) > 1;
```

```
DELETE FROM Employee
WHERE EmpID IN
(SELECT EmpID FROM Employee
GROUP BY EmpID
HAVING COUNT(*) > 1);
```

Q7(b): *Query to retrieve the list of employees working in same project.*

```
WITH CTE AS
        (SELECT e.EmpID, e.EmpName, ed.Project
        FROM Employee AS e
        INNER JOIN EmployeeDetail AS ed
        ON e.EmpID = ed.EmpID)
SELECT c1.EmpName, c2.EmpName, c1.project
FROM CTE c1, CTE c2
WHERE c1.Project = c2.Project AND c1.EmpID != c2.EmpID AND c1.EmpID < c2.EmpID
```

Q8: *Show the employee with the highest salary for each project*

```
SELECT ed.Project, MAX(e.Salary) AS ProjectSal
FROM Employee AS e
INNER JOIN EmployeeDetail AS ed
ON e.EmpID = ed.EmpID
GROUP BY Project
ORDER BY ProjectSal DESC;
```

Similarly we can find Total Salary for each project, just use SUM() instead of MAX()

Alternative, more dynamic solution: here you can fetch EmpName, 2nd/3rd highest value, etc

```
WITH CTE AS
    (SELECT project, EmpName, salary,
    ROW_NUMBER() OVER (PARTITION BY project ORDER BY salary DESC) AS row_rank
    FROM Employee AS e
    INNER JOIN EmployeeDetail AS ed
    ON e.EmpID = ed.EmpID)
SELECT project, EmpName, salary
FROM CTE
WHERE row_rank = 1;
```

Q9: *Query to find the total count of employees joined each year*

```
SELECT EXTRACT('year' FROM doj) AS JoinYear, COUNT(*) AS EmpCount
FROM Employee AS e
INNER JOIN EmployeeDetail AS ed ON e.EmpID = ed.EmpID
GROUP BY JoinYear
ORDER BY JoinYear ASC
```

Q10: *Create 3 groups based on salary col, salary less than 1L is low, between 1 - 2L is medium and above 2L is High*

```
SELECT EmpName, Salary,
       CASE
           WHEN Salary > 200000 THEN 'High'
           WHEN Salary >= 100000 AND Salary <= 200000 THEN 'Medium'
           ELSE 'Low'
       END AS SalaryStatus
FROM Employee
```

BONUS: ***Query to pivot the data in the Employee table and retrieve the total salary for each city.***
The result should display the EmpID, EmpName, and separate columns for each city (Mathura, Pune, Delhi), containing the corresponding total salary.

```
SELECT
  EmpID,
  EmpName,
  SUM(CASE WHEN City = 'Mathura' THEN Salary END) AS "Mathura",
  SUM(CASE WHEN City = 'Pune' THEN Salary END) AS "Pune",
  SUM(CASE WHEN City = 'Delhi' THEN Salary END) AS "Delhi"
FROM Employee
GROUP BY EmpID, EmpName;
```