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

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

```
SELECT *
FROM employee
WHERE salary BETWEEN 200000 AND 300000;
```

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

```
SELECT a.empid, a.empname, a.city, a.salary
 FROM employee as a
 JOIN employee as b ON a.empid != b.empid
 WHERE a.city = b.city ;
```

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 Male,
(COUNT(*) FILTER (WHERE Gender = 'F') * 100.0 / COUNT(*) ) AS Female
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);
```

Q3: Show the employee with the highest salary for each project.

```
WITH cte AS (
SELECT *, ROW_NUMBER() OVER (PARTITION BY ed.project ORDER BY e.salary DESC) AS row_no
FROM employee as e
JOIN employeedetail as ed ON e.empid = ed.empid
)
SELECT project, salary, empname
FROM cte
WHERE row_no <=1;
```

Q4: Query to find the total count of employees joined each year.

```
SELECT EXTRACT(year from doj) AS joining_year, COUNT(empid)
FROM employeedetail
GROUP BY 1
ORDER BY 1 ASC;
```

Q5: Create 3 groups based on salary columns, salary less than 1 Lakh is low, between 1 to 2 lakh is medium and above 2 lakh is high.

```
SELECT   *,
   CASE
     WHEN salary > 200000 THEN 'High’
     WHEN salary BETWEEN 100000 AND 200000 THEN 'Medium’
     WHEN salary < 100000 THEN 'Low’
   END
FROM employee;
```

Q6: 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 ed.empid = e.empid
)
SELECT a.empid, a.empname, b.empid, b.empname, a.project
FROM cte as a
JOIN cte as b ON a.empid != b.empid
WHERE a.empid < b.empid AND a.project = b.project
```

Q7: Write a query to fetch even and odd rows from the employee table.

```
--for even rows--
SELECT *
FROM employee
WHERE empid % 2 = 0;

--for odd rows--
SELECT *
FROM employee
WHERE empid % 2 = 1;
```

Q8: Query to fetch the employee's salary but replace the LAST 2 digits with 'XX'
i.e 12345 will be 123XX.

```
SELECT *, CONCAT(SUBSTRING(salary::text, 1, LENGTH(salary::text)-2),'XX') AS masked_numbers
FROM employee;
```

Q9(a): Write a query to find all the Employee names whose names:

- **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’**

```
1) SELECT * FROM employee WHERE empname LIKE 'A%’;
2) SELECT * FROM employee WHERE empname LIKE '_a%’;
3) SELECT * FROM employee WHERE empname LIKE '%y_’;
4) SELECT * FROM employee WHERE empname LIKE ‘_ _ _l’;
5) SELECT * FROM employee WHERE empname LIKE 'V%a';
```

Q9(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.**

```
1. SELECT DISTINCT * FROM employee
   WHERE empname SIMILAR TO '[A,E,I,O,U]%';

2. SELECT DISTINCT * FROM employee
   WHERE empname SIMILAR TO '%[a,e,i,o,u]';

3. SELECT DISTINCT * FROM employee
    WHERE empname SIMILAR TO '[A,E,I,O,U]%[a,e,i,o,u]';
```