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')
```

```
create database company;
show databases;
use company;

-- Create Tables: employee and
employee_detail

create table employee(
emp_id int not null,
emp_name varchar(25),
gender
char(20),
salary int,
city char(20)
);

insert into employee(emp_id, emp_name, gender, salary,
city)
values
(1,'Arjun','M',75000,'Pune'),
(2,'Ekadanta','M',125000,'Bangalore'),
(3,'Lalita','F',150
000,'Mathura'),
(4,'Madhav','M',250000,'Delhi'),
(5,'Visakha','F',120000,'Mathura');

create
table employee_detail(
emp_id int not null,
project varchar(50),
emp_position char(20),
DOJ
date
);

insert into employee_detail(emp_id, project, emp_position, DOJ)
values
(1,'P1','Executive','2019-01-26'),
(2,'P2','Executive','2020-05-04'),
(3,'P1','Lead','20
21-10-21'),
(4,'P3','Manager','2019-11-29'),
(5,'P2','Manager','2020-08-01');

show
tables;
select * from employee_detail;
select * from employee;

-- Find the list of employees
whose salary ranges between 2L to 3L.
select emp_name, salary from employee where
salary>200000 and salary<300000;
-- or
select emp_name, salary from employee where salary
between 200000 and 300000;

-- Find the list of employees whose salary ranges between 2L to
3L.
select e1.emp_id, e1.emp_name, e1.city
from employee e1, employee e2
where e1.city=e2.city
and e1.emp_id!=e2.emp_id;

-- Query to find the null values in the Employee table.
select *
from employee where emp_id is null;

-- Query to find the cumulative sum of employee’s
salary.
select emp_id, salary, sum(salary) over(order by emp_id) as cumulative_sum from
employee;

-- What’s the male and female employees ratio.
```

```
SELECT
    (COUNT(*) * 100.0 /
(SELECT COUNT(*) FROM employee)) AS male_ratio
FROM employee WHERE gender = 'M';
SELECT

(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM employee)) AS female_ratio
FROM employee WHERE gender
= 'F';

-- Write a query to fetch 50% records from the Employee table.
select * from
employee
where emp_id <= (select count(emp_id)/2 from employee);

select * from
        (select *,
Row_Number() over(order by emp_id) as RowNumber
        from employee) as emp
where emp.RowNumber
<= (select count(emp_id)/2 from employee);

-- 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(CAST(salary AS CHAR), 1, LENGTH(salary) - 2), 'XX') AS masked_number
from
employee;

-- Write a query to fetch even and odd rows from Employee table.
-- even
select *
from
        (select *, Row_Number() over(order by emp_id) as
    RowNumber
    from employee) as
emp
where emp.RowNumber % 2 = 0;

-- odd
select * from
        (select *, Row_Number() over(order by
emp_id) as
    RowNumber
    from employee) as emp
where emp.RowNumber % 2 = 1;
-- or
select *
from employee where mod(emp_id,2) = 0;

-- Write a query to find all the Employee names whose
name:
-- • Begin with ‘A’
select * from employee where emp_name like 'A%';

-- •
Contains ‘A’ alphabet at second place
select * from employee where emp_name like '_a%';

--
• Contains ‘Y’ alphabet at second last place
select * from employee where emp_name like
'%y_';

-- • Ends with ‘L’ and contains 4 alphabets
select * from employee where emp_name
like '____l';

-- • Begins with ‘V’ and ends with ‘A’
select * from employee where
emp_name like 'V%a';
```

```
-- Write a query to find the list of Employee names which is:
-- •
starting with vowels (a, e, i, o, or u), without duplicates
select distinct emp_name
from
employee
where lower(emp_name) regexp '^[aeiou]';

-- • ending with vowels (a, e, i, o, or
u), without duplicates
select distinct emp_name
from employee
where lower(emp_name) regexp
'[aeiou]$';

-- • starting & ending with vowels (a, e, i, o, or u), without
duplicates
select distinct emp_name
from employee
where lower(emp_name) regexp
'^[aeiou].*[aeiou]$';

-- Find Nth highest salary from employee table with and without using
the LIMIT keywords.
-- without limit
select salary from employee as e1
where N-1 = (
        select
count(distinct(e2.salary))
    from employee as e2
    where e2.salary>e1.salary
    );

--
using limit
select salary from employee
order by salary desc
limit 1 offset N;

-- Write a
query to find and remove duplicate records from a table.
-- finding duplicates
select emp_id,
emp_name, gender, salary, city,
count(*) as duplicate_count
from employee
group by emp_id,
emp_name, gender, salary, city
having count(*) > 1;

-- removing duplicates
delete from
employee
where emp_id in (
    select emp_id
    from (
        select emp_id
        from
employee
        group by emp_id
        having count(*) > 1
    ) as duplicates
);
SET
SQL_SAFE_UPDATES=0;  -- this statement is use for remove the warning when we use delete, so the
above query
                        -- will execute fine.

-- Query to retrieve the list
of employees working in same project.
with CTE as(
        select e.emp_id, e.emp_name, ed.project

from employee as e
    inner join employee_detail as ed
```

```
    on e.emp_id = ed.emp_id

)
select c1.emp_name, c2.emp_name, c1.project
from CTE c1, CTE c2
where c1.project = c2.project
and c1.emp_id != c2.emp_id and c1.emp_id < c2.emp_id;

/* Show the employee with the highest
salary for each project and also find the total salary spend on a
particular project */
select
ed.project, max(salary) as project_max_sal, sum(e.salary)
as project_total_sal from employee as
e
inner join employee_detail as ed
on e.emp_id = ed.emp_id
group by project
order by
project_max_sal desc;

-- Alternative, more dynamic solution: here you can fetch
EmpName, 2nd/3rd highest value, etc
WITH CTE AS (
        SELECT project, emp_name,
salary,
        Row_Number() OVER (PARTITION BY project ORDER BY salary Desc) AS row_rank
        FROM
employee AS e
        INNER JOIN employee_detail AS ed
        ON e.emp_id = ed.emp_id
    )
SELECT project,
emp_name, salary
FROM CTE
WHERE row_rank = 1;

-- Query to find the total count of employees
joined each year
select year(DOJ) AS join_year, count(*) AS emp_count
from employee as e
inner
join
employee_detail as ed on e.emp_id = ed.emp_id
group by join_year
order by join_year
asc;

/* Create 3 groups based on salary col, salary less than 1L is low, between
1-2L is
medium and above 2L is High   */
select emp_name, salary,
        case
                when salary > 200000 then
'High'
                when salary >= 100000 and salary <= 200000 then 'Medium'
                else 'Low'
        end as
salary_status
from employee;


/* 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 emp_id, emp_name,
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 emp_id,
emp_name;
```