**SQL COURSE END PROJECT 2**

**Q1**: Find all employees whose salaries are equal to the average salary of their department.

**Solve using subquery without using ANY, ALL**

select employee\_id,concat(first\_name, "",LAST\_NAME),department\_id,salary

from employees\_data where (department\_id,salary) in

(select department\_id,avg(salary)

from employees\_data group by department\_id)

**Q1.1**: **Solve the same using ANY OR ALL operator**

select \* from employees\_data a

where salary=(select avg(salary) from employees\_data

where department\_id=a.department\_id)

**Solve the same using join(ALTERNATE)**

select \* from employees\_data a join

(select department\_id,avg(salary) as avg\_salary from employees\_data

group by department\_id) b on a.department\_id=b.department\_id

where a.salary=b.avg\_salary

**Q 2:** Find all employees whose salaries are lesser than the lowest average salary of the department.

select EMPLOYEE\_ID,FIRST\_NAME,DEPARTMENT\_ID,SALARY

from employees\_data where salary<

(select min(sal) from

(select avg(SALARY) sal from employees\_data group by DEPARTMENT\_ID) j)

**Q3**: Find all employees whose salaries are more than the highest average salary of the department.

select EMPLOYEE\_ID,FIRST\_NAME,DEPARTMENT\_ID,SALARY

from employees\_data where salary>

(select max(sal) from

(select avg(SALARY) sal from employees\_data group by DEPARTMENT\_ID) j)

**Q4**: Find all employees whose salaries are greater than the highest salary in the Marketing department.

select \* from employees\_data where salary>

(select max(salary) from employees\_data a inner join department\_data b

on a.DEPARTMENT\_ID=b.DEPARTMENT\_ID

where department\_name="Marketing")

**Q5:** Find the Employee Id and Name of the managers with a minimum of 7 employees directly reporting to them.

Use the `**employees\_data**` table from the Metabase.

select mgr\_id,mgr\_name,count(mgr\_name) from (select a.EMPLOYEE\_ID,a.FIRST\_NAME,b.EMPLOYEE\_ID as "mgr\_id",b.FIRST\_NAME as "mgr\_name" from employees\_data a inner join employees\_data b on a.manager\_id=b.EMPLOYEE\_ID)n group by mgr\_id having count(mgr\_name)>=7

**Q6**: Print the employee's name against each manager, as shown below.

select b.FIRST\_NAME as "mgr\_name",a.FIRST\_NAME as "Employee\_name" from employees\_data a inner join employees\_data b on a.manager\_id=b.EMPLOYEE\_ID

**Q7**: Write a SQL query that finds out employees who earn more than their managers

Use the `**employees\_data**` table from the Metabase.

select a.EMPLOYEE\_ID,a.FIRST\_NAME,a.salary,b.EMPLOYEE\_ID as "mgr\_id", b.FIRST\_NAME as "mgr\_name",b.salary from employees\_data a inner join employees\_data b on a.manager\_id=b.EMPLOYEE\_ID where a.salary>b.salary

**Q8**: Print the person's name with consecutive years (twice).

DB -<https://www.db-fiddle.com/f/8Y61oCnyAkz64v3qFgVV4k/0>

select Name, count(nextyear-Year) from (select Name,Year,Lead(Year,1,"N/A")over(partition by Name) as nextyear from t) h where (nextyear-Year)=1 group by Name

Alternative

SELECT DISTINCT(a.Name), a.Year

FROM t AS a

JOIN t AS b

ON a.Name = b.Name AND (a.Year = b.Year+1 OR a.Year+1 = b.Year)

ORDER BY a.Name, a.Year;

**Q9**: We have a table **employee\_data** which has fields first\_name and last\_name. Some or Both of them can be NULL.

We would like to select the full name, one of them, or if both are not set, then default to **No Name**

SELECT FIRST\_NAME,LAST\_NAME, CASE WHEN FIRST\_NAME IS NOT NULL AND LAST\_NAME IS NOT NULL THEN concat(FIRST\_NAME,' ',LAST\_NAME) WHEN FIRST\_NAME IS NOT NULL THEN FIRST\_NAME WHEN LAST\_NAME IS NOT NULL THEN LAST\_NAME ELSE 'NO NAME' END AS 'FULL\_NAME' FROM employees\_data

**10.As a programmer write a SQL query to display details of all the trainee (traineeid, traineename, cgpa and technology trained) for the trainees who are eligible for .NET project but aren't allocated with any. Display the result in the increasing order of traineeid. DB -** [**https://www.db-fiddle.com/f/vQepCw3A4RnpZMF3tP6rQm/0**](https://www.db-fiddle.com/f/vQepCw3A4RnpZMF3tP6rQm/0)

select a.traineeid, a.traineename, a.cgpa,a.technology from trainee a left join projectallocation b on a.traineeid=b.traineeid where a.technology=".NET" and b.traineeid is null order by a.traineeid