

SQL Case Study – 2

Simple Queries

--1. List all the employee details.

```
SELECT * FROM EMPLOYEE
```

--2. List all the department details.

```
SELECT * FROM DEPARTMENT
```

--3. List all job details.

```
SELECT * FROM JOB
```

--4. List all the locations.

```
SELECT * FROM LOCATION
```

--5. List out the First Name, Last Name, Salary, Commission for allEmployees.

```
SELECT  
LAST_NAME,  
FIRST_NAME,  
COMM
```

FROM EMPLOYEE

--6. List out the Employee ID, Last Name, Department ID for all employees and alias Employee ID as "ID of the Employee", Last Name as "Name of the Employee", Department ID as "Dep_id".

```
SELECT  
EMPLOYEE_ID AS "ID OF EMPLOYEE",  
LAST_NAME AS "EMPLOYEE NAME",  
DEPARTMENT_ID AS "DEPARTMENT ID"  
FROM EMPLOYEE
```

--7. List out the annual salary of the employees with their names only.

```
SELECT  
LAST_NAME AS "EMPLOYEE NAME",  
SALARY*12 AS "ANUAL SALARY"  
FROM EMPLOYEE
```

WHERE Condition

--1. List the details about "Smith".

```
SELECT * FROM EMPLOYEE  
WHERE LAST_NAME='SMITH'
```

--2. List out the employees who are working in department 20.

```
SELECT * FROM EMPLOYEE  
WHERE DEPARTMENT_ID=20
```

--3. List out the employees who are earning salaries between 3000 and 4500.

```
SELECT * FROM EMPLOYEE  
WHERE SALARY BETWEEN 3000 AND 4500
```

--4. List out the employees who are working in department 10 or 20.

```
SELECT * FROM EMPLOYEE  
WHERE DEPARTMENT_ID IN(20,30)
```

--5. Find out the employees who are not working in department 10 or 30.

```
SELECT * FROM EMPLOYEE  
WHERE DEPARTMENT_ID NOT IN(10,30)
```

--6. List out the employees whose name starts with 'S'

```
SELECT * FROM EMPLOYEE  
WHERE LAST_NAME LIKE 'S%'
```

--7. List out the employees whose name starts with 'S' and ends with 'H'.

```
SELECT * FROM EMPLOYEE  
WHERE LAST_NAME LIKE 'S%H'
```

--8. List out the employees whose name length is 4 and start with 'S'.

```
SELECT * FROM EMPLOYEE  
WHERE LAST_NAME LIKE 'S____'
```

--9. List out employees who are working in department 10 and draw salaries more than 3500.

```
SELECT * FROM EMPLOYEE  
WHERE DEPARTMENT_ID=10  
AND SALARY >3500
```

--10. List out the employees who are not receiving commission.

```
SELECT * FROM EMPLOYEE  
WHERE COMM IS NULL
```

ORDER BY Clause

--1. List out the Employee ID and Last Name in ascending order based on the Employee ID.

```
SELECT EMPLOYEE_ID, LAST_NAME  
FROM EMPLOYEE  
ORDER BY EMPLOYEE_ID
```

--2. List out the Employee ID and Name in descending order based on salary.

```
SELECT EMPLOYEE_ID, LAST_NAME  
FROM EMPLOYEE  
ORDER BY SALARY DESC
```

--3. List out the employee details according to their Last Name in ascending-order.

```
SELECT * FROM EMPLOYEE  
ORDER BY LAST_NAME, SALARY DESC
```

--4. List out the employee details according to their Last Name in ascending order and then Department ID in descending order.

```
SELECT * FROM EMPLOYEE  
ORDER BY LAST_NAME,DEPARTMENT_ID DESC
```

GROUP BY and HAVING Clause

--1. How many employees are in different departments in the organization?

```
SELECT DEPARTMENT_ID,COUNT(*) AS "NUMBER OF EMPLOYEE"  
FROM EMPLOYEE  
GROUP BY DEPARTMENT_ID
```

--2. List out the department wise maximum salary, minimum salary and average salary of the employees.

```
SELECT DEPARTMENT_ID,  
MAX(SALARY) MAXUM_SALARY,  
MIN(SALARY) MINMUM_SALARY,  
AVG(SALARY) AVERAGE_SALARY  
FROM EMPLOYEE  
GROUP BY DEPARTMENT_ID
```

--3. List out the job wise maximum salary, minimum salary and average salary of the employees.

```
SELECT JOB_ID,  
MAX(SALARY) MAXUM_SALARY,  
MIN(SALARY) MINMUM_SALARY,  
AVG(SALARY) AVERAGE_SALARY
```

```
FROM EMPLOYEE  
GROUP BY JOB_ID
```

--4. List out the number of employees who joined each month in ascending order.

```
SELECT DATENAME(MM,HIRE_DATE) MONTH,COUNT(*) NoOfEmployee  
FROM EMPLOYEE  
GROUP BY DATENAME(MM,HIRE_DATE)
```

--5. List out the number of employees for each month and year in ascending order based on the year and month.

```
SELECT DATEPART(YYYY,HIRE_DATE) YEAR,  
       DATENAME(MM,HIRE_DATE) MONTH,  
       COUNT(*) NoOfEmployee  
FROM EMPLOYEE  
GROUP BY DATEPART(YYYY,HIRE_DATE),DATENAME(MM,HIRE_DATE)
```

--6. List out the Department ID having at least four employees.

```
SELECT DEPARTMENT_ID,COUNT(*) "NUMBER OF EMPLOYEES"  
FROM EMPLOYEE  
GROUP BY DEPARTMENT_ID  
HAVING COUNT(*)>4
```

--7. How many employees joined in the month of January?

```
SELECT DATENAME(MM,HIRE_DATE) MONTH,COUNT(*) "NUMBER OF EMPLOYEES"  
FROM EMPLOYEE  
WHERE DATENAME(MM,HIRE_DATE)='JANUARY'
```

GROUP BY DATENAME(MM,HIRE_DATE)

--8. How many employees joined in the month of January or September?

```
SELECT DATENAME(MM,HIRE_DATE) MONTH,COUNT(*) "NUMBER OF EMPLOYEES"
FROM EMPLOYEE
WHERE DATENAME(MM,HIRE_DATE)IN ('JANUARY','SEPTEMBER')
GROUP BY DATENAME(MM,HIRE_DATE)
```

--9. How many employees joined in 1985?

```
SELECT DATEPART(YY,HIRE_DATE) YEAR,COUNT(*) "NUMBER OF EMPLOYEES"
FROM EMPLOYEE
WHERE DATEPART(YY,HIRE_DATE)=1985
GROUP BY DATEPART(YY,HIRE_DATE)
```

--10. How many employees joined each month in 1985?

```
SELECT DATEPART(YY,HIRE_DATE) YEAR,
       DATENAME(MM,HIRE_DATE) MONTH,
       COUNT(*) "NUMBER OF EMPLOYEES"
FROM EMPLOYEE
WHERE DATEPART(YY,HIRE_DATE)=1985
GROUP BY DATEPART(YY,HIRE_DATE),DATENAME(MM,HIRE_DATE)
```

--11. How many employees joined in March 1985?

```
SELECT DATEPART(YY,HIRE_DATE) YEAR,
       DATENAME(MM,HIRE_DATE) MONTH,
       COUNT(*) "NUMBER OF EMPLOYEES"
```

```
FROM EMPLOYEE
WHERE DATEPART(YEAR,HIRE_DATE) = 1985
```

--12. Which is the Department ID having greater than or equal to 3 employees joining in April

```
SELECT DEPARTMENT_ID,
COUNT(*) "NUMBER OF EMPLOYEES"
FROM EMPLOYEE
WHERE DATEPART(YEAR,HIRE_DATE)=1985
AND
DATENAME(MONTH,HIRE_DATE)='APRIL'
GROUP BY DEPARTMENT_ID
HAVING COUNT(*)>=3
```

Joins

--1. List out employees with their department names.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY, E.DEPARTMENT_ID, NAME
FROM EMPLOYEE E
INNER JOIN DEPARTMENT D
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID
```

--2. Display employees with their designations.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY, E.DEPARTMENT_ID, Designation
FROM EMPLOYEE E
INNER JOIN JOB J
ON E.JOB_ID=J.JOB_ID
```


--3. Display the employees with their department names and regional groups.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY, E.DEPARTMENT_ID, NAME, City
FROM EMPLOYEE E
INNER JOIN DEPARTMENT D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID
INNER JOIN LOCATION1 L ON D.LOCATION_ID=L.LOCATION_ID
```

--4. How many employees are working in different departments? Display with department names.

```
SELECT NAME AS "DEPARTMENT NAME", COUNT(*) AS "NUMBER OF EMPLOYEES"
FROM DEPARTMENT D
INNER JOIN EMPLOYEE E
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID
GROUP BY NAME
```

--5. How many employees are working in the sales department?

```
SELECT NAME, COUNT(*) AS "NUMBER OF EMPLOYEES"
FROM DEPARTMENT D
INNER JOIN EMPLOYEE E
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID
GROUP BY NAME
HAVING NAME='SALES'
```

--6. Which is the department having greater than or equal to 5 employees? Display the department names in ascending order.

```
SELECT NAME AS "DEPARTMENT NAME", COUNT(*) AS "NUMBER OF EMPLOYEES"
FROM DEPARTMENT D
```

```
INNER JOIN EMPLOYEE E
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID
GROUP BY NAME
HAVING COUNT(*)>=5
```

--7. How many jobs are there in the organization? Display with designations.

```
SELECT Designation,COUNT(*) AS "NO OF JOBS"
FROM JOB J
INNER JOIN EMPLOYEE E
ON E.JOB_ID=J.JOB_ID
GROUP BY Designation
```

--8. How many employees are working in "New York"?

```
SELECT City,COUNT(*) "NUMBER OF EMPLOYEES"
FROM EMPLOYEE E, DEPARTMENT D, LOCATION1 L
WHERE E.DEPARTMENT_ID=D.DEPARTMENT_ID
AND D.LOCATION_ID=L.LOCATION_ID
AND L.City='NEW YORK'
GROUP BY CITY
```

--9. Display the employee details with salary grades. Use conditional statement to create a grade column.

```
SELECT
EMPLOYEE_ID,
FIRST_NAME,
SALARY,
CASE
WHEN SALARY > 2500 THEN 'GRADE 1'
```

```
WHEN SALARY BETWEEN 1500 AND 2500 THEN 'GRADE 2'
```

```
WHEN SALARY < 1500 THEN 'GRADE 3'
```

```
END AS GRADE
```

```
FROM EMPLOYEE
```

--10. List out the number of employees grade wise. Use conditional statement to create a grade column.

```
SELECT
```

```
GRADE,
```

```
COUNT(GRADE) AS 'NO OF EMP WITH GRADE'
```

```
FROM EMPLOYEE
```

```
GROUP BY GRADE
```

--11. Display the employee salary grades and the number of employees between 2000 to 5000 range of salary.

```
SELECT
```

```
SALARY ,
```

```
GRADE
```

```
FROM EMPLOYEE
```

```
WHERE SALARY BETWEEN 2000 AND 5000
```

--12. Display all employees in sales or operation departments.

```
SELECT
```

```
EMPLOYEE_ID,
```

```
FIRST_NAME,
```

```
LAST_NAME,
```

```
Name as 'Departments'
```

```
FROM EMPLOYEE E
```

```
INNER JOIN DEPARTMENT D
```

```
ON E.DEPARTMENT_ID = D.Department_Id  
WHERE Name = 'Sales' OR NAME = 'Operations'
```

```
select * from EMPLOYEE  
SELECT * FROM DEPARTMENT  
SELECT * FROM JOB
```

SET Operators

--1. List out the distinct jobs in sales and accounting departments.

```
SELECT  
Designation  
FROM JOB  
WHERE Job_ID  
IN  
(SELECT Job_ID FROM EMPLOYEE WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM  
DEPARTMENT WHERE NAME = 'SALES'))  
UNION  
SELECT  
Designation  
FROM JOB  
WHERE Job_ID  
IN  
(SELECT Job_ID FROM EMPLOYEE WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM  
DEPARTMENT WHERE NAME = 'ACCOUNTING'))
```

```
--SELECT Designation  
--FROM JOB WHERE JOB_ID IN(SELECT JOB_ID FROM EMPLOYEE  
--WHERE DEPARTMENT_ID=(SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE NAME='SALES'))
```

--UNION

--SELECT Designation

--FROM JOB WHERE JOB_ID IN(SELECT JOB_ID FROM EMPLOYEE

--WHERE DEPARTMENT_ID=(SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE
NAME='ACCOUNTING'))

--2. List out all the jobs in sales and accounting departments.

SELECT Designation

FROM JOB WHERE JOB_ID IN(SELECT JOB_ID FROM EMPLOYEE

WHERE DEPARTMENT_ID=(SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE NAME='sales'))

UNION

SELECT Designation

FROM JOB WHERE JOB_ID IN(SELECT JOB_ID FROM EMPLOYEE

WHERE DEPARTMENT_ID=(SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE
NAME='ACCOUNTING'))

--3. List out the common jobs in research and accounting departments in ascending

SELECT Designation

FROM JOB WHERE JOB_ID IN(SELECT JOB_ID FROM EMPLOYEE

WHERE DEPARTMENT_ID=(SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE
NAME='RESEARCH'))

UNION

SELECT Designation

FROM JOB WHERE JOB_ID IN(SELECT JOB_ID FROM EMPLOYEE

WHERE DEPARTMENT_ID=(SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE
NAME='ACCOUNTING'))

order. Subqueries

--1. Display the employees list who got the maximum salary.

```
SELECT *  
FROM EMPLOYEE  
WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEE)
```

--2. Display the employees who are working in the sales department.

```
SELECT *  
FROM EMPLOYEE  
WHERE DEPARTMENT_ID IN (SELECT Department_Id FROM DEPARTMENT WHERE Name ='SALES')
```

--3. Display the employees who are working as 'Clerk'.

```
SELECT *  
FROM EMPLOYEE  
WHERE JOB_ID = (SELECT JOB_ID FROM JOB WHERE Designation = 'CLERK' )
```

--4. Display the list of employees who are living in "New York".

```
SELECT * FROM EMPLOYEE,DEPARTMENT  
WHERE Location_ID = (SELECT Location_Id FROM LOCATION1 WHERE City = 'NEW YORK')
```

-- QUERY IS R8 BUT NOT FETCHING DATA

-- SELECT * FROM EMPLOYEE

```
-- WHERE
-- DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM DEPARTMENT
--      WHERE Location_Id IN (SELECT Location_Id FROM LOCATION1
--      WHERE City = 'NEW YORK'))
```

--5. Find out the number of employees working in the sales department.

```
SELECT COUNT(DEPARTMENT_ID) AS 'PEOPLES WORK SALES DEPT' FROM EMPLOYEE
WHERE DEPARTMENT_ID = (SELECT Department_Id FROM DEPARTMENT WHERE Name = 'SALES')
```

--6. Update the salaries of employees who are working as clerks on the basis of 10%.

```
UPDATE EMPLOYEE SET SALARY = (SALARY + SALARY * 10 / 100)
WHERE JOB_ID = (SELECT JOB_ID FROM JOB WHERE Designation = 'CLERK')
```

--7. Delete the employees who are working in the accounting department.

```
DELETE FROM EMPLOYEE
WHERE DEPARTMENT_ID = (SELECT Department_Id FROM DEPARTMENT WHERE Name =
'Accounting' )
```

--8. Display the second highest salary drawing employee details.

```
--SELECT * FROM EMPLOYEE
--WHERE SALARY = (SELECT MAX (SALARY) FROM EMPLOYEE
--      WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE))
```

```
;WITH T AS
```

```
(
```

```
SELECT *,
```

```

DENSE_RANK() OVER (ORDER BY SALARY DESC) AS RNK
FROM EMPLOYEE
)
SELECT
EMPLOYEE_ID,
FIRST_NAME,
LAST_NAME,
SALARY
FROM T
WHERE RNK = 2;

```

--9. Display the nth highest salary drawing employee details.

```

SELECT A.SALARY FROM EMPLOYEE A
WHERE (SELECT COUNT(DISTINCT(B.SALARY)) FROM EMPLOYEE B
WHERE A.SALARY<B.SALARY)IN(1,2,4)

```

--10. List out the employees who earn more than every employee in department 30.

```

SELECT e.DEPARTMENT_ID, e.FIRST_NAME, e.SALARY
FROM EMPLOYEE e
WHERE e.SALARY = (SELECT MAX(e2.SALARY)
FROM EMPLOYEE e2
WHERE e2.DEPARTMENT_ID = e.DEPARTMENT_ID
) AND
e.DEPARTMENT_ID = 30;

```


--11. List out the employees who earn more than the lowest salary in department. Find out whose department has no employees.

```
SELECT e.DEPARTMENT_ID, e.FIRST_NAME, e.SALARY
FROM EMPLOYEE e
WHERE e.SALARY > (SELECT MIN(e2.SALARY)
FROM EMPLOYEE e2
WHERE e2.DEPARTMENT_ID = e.DEPARTMENT_ID
)
```

--12. Find out which department has no employees.

```
SELECT
Department_Id,
Name
FROM DEPARTMENT
WHERE Department_Id NOT IN (SELECT DEPARTMENT_ID FROM EMPLOYEE)
```

--13. Find out the employees who earn greater than the average salary for their department.

```
SELECT * FROM JOB
SELECT * FROM LOCATION1
SELECT * FROM DEPARTMENT
SELECT * FROM EMPLOYEE
```