ASSIGNMENT: SQL – CASE STUDY 2

```
-- Case Study 2

-- TABLE 1 - LOCATION

CREATE TABLE LOCATION(Location_ID INT PRIMARY KEY, City VARCHAR(50));

INSERT INTO LOCATION(Location_ID, City) VALUES
                    (122, 'New York'),
                    (123, 'Dallas'),
                    (124, 'Chicago'),
                    (167, 'Boston');

SELECT * FROM LOCATION;


-- TABLE 2 - DEPARTMENT

CREATE TABLE DEPARTMENT(
                     Department_Id INT PRIMARY KEY,
                     Name VARCHAR(50),
                     Location_Id INT,
                     FOREIGN KEY (Location_Id) REFERENCES LOCATION(Location_ID)
                     );

INSERT INTO DEPARTMENT (Department_Id, Name, Location_Id) VALUES
              (10, 'Accounting', 122),
              (20, 'Sales', 124),
              (30, 'Research', 123),
              (40, 'Operations', 167);

SELECT * FROM DEPARTMENT;


-- TABLE 3 - JOB

CREATE TABLE JOB(JOB_ID INT PRIMARY KEY, DESIGNATION VARCHAR(20));

INSERT  INTO JOB VALUES
                    (667, 'CLERK'),
                    (668,'STAFF'),
                    (669,'ANALYST'),
                    (670,'SALES_PERSON'),
                    (671,'MANAGER'),
                    (672, 'PRESIDENT')

SELECT * FROM JOB;

-- TABLE 4 - EMPLOYEE

DROP TABLE EMPLOYEE

CREATE TABLE EMPLOYEE(
                    EMPLOYEE_ID INT,
                    LAST_NAME VARCHAR(20),
```

```
                    FIRST_NAME VARCHAR(20),
                    MIDDLE_NAME CHAR(1),
                    JOB_ID INT FOREIGN KEY
                    REFERENCES JOB(JOB_ID),
                    MANAGER_ID INT,
                    HIRE_DATE DATE,
                    SALARY INT,
                    COMM INT,
                    DEPARTMENT_ID  INT FOREIGN KEY
                    REFERENCES DEPARTMENT(DEPARTMENT_ID))

INSERT INTO EMPLOYEE VALUES
(7369,'SMITH','JOHN','Q',667,7902,'17-DEC-84',800,NULL,20),
(7499,'ALLEN','KEVIN','J',670,7698,'20-FEB-84',1600,300,30),
(7505,'DOYLE','JEAN','K',671,7839,'04-APR-85',2850,NULl,30),
(7506,'DENNIS','LYNN','S',671,7839,'15-MAY-85',2750,NULL,30),
(7507,'BAKER','LESLIE','D',671,7839,'10-JUN-85',2200,NULL,40),
(7521,'WARK','CYNTHIA','D',670,7698,'22-FEB-85',1250,500,30)

SELECT * FROM EMPLOYEE;

-- 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 all Employees.

       SELECT FIRST_NAME, LAST_NAME, SALARY, 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 THE EMPLOYEE" , LAST_NAME AS "NAME OF THE EMPLOYEE",
       DEPARTMENT_ID AS Dept_id FROM EMPLOYEE;

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

       SELECT FIRST_NAME, LAST_NAME, 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 =10 OR DEPARTMENT_ID = 20
-- 5. Find out the employees who are not working in department 10 or 30.
       SELECT * FROM EMPLOYEE WHERE NOT DEPARTMENT_ID =10 AND DEPARTMENT_ID = 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 LEN(FIRST_NAME) = 4 AND 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 EMPLOYEE_ID 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 LAST_NAME ASC;
-- 2. List out the Employee ID and Name in descending order based on salary.
       SELECT EMPLOYEE_ID, LAST_NAME, SALARY  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 ASC;
```

```
-- 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 ASC, DEPARTMENT_ID DESC;

       SELECT * FROM EMPLOYEE ORDER BY DEPARTMENT_ID DESC;

-- GROUP BY AND HAVING CLAUSE

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

       SELECT COUNT(EMPLOYEE_ID) FROM EMPLOYEE;

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

       SELECT MAX(SALARY), MIN(SALARY), AVG(SALARY) FROM EMPLOYEE GROUP BY DEPARTMENT_ID;

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

       SELECT MAX(SALARY), MIN(SALARY), AVG(SALARY) FROM EMPLOYEE GROUP BY JOB_ID;

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

       SELECT MONTH(HIRE_DATE) AS MONTH, COUNT(EMPLOYEE_ID) AS NO_OF_EMPLOYEES FROM
       EMPLOYEE GROUP BY MONTH(HIRE_DATE)

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

       SELECT MONTH(HIRE_DATE) AS MONTH,YEAR(HIRE_DATE) AS YEAR,COUNT(EMPLOYEE_ID) AS
       NO_OF_EMPLOYEES FROM EMPLOYEE
       GROUP BY MONTH(HIRE_DATE), YEAR(HIRE_DATE)ORDER BY YEAR(HIRE_DATE),
       MONTH(HIRE_DATE) ASC;

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

       SELECT DEPARTMENT_ID FROM EMPLOYEE GROUP BY DEPARTMENT_ID HAVING
       COUNT(DEPARTMENT_ID) = 4;

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

       SELECT COUNT(EMPLOYEE_ID) AS NO_OF_EMPLOYEES FROM EMPLOYEE WHERE MONTH(HIRE_DATE)
       = 1;

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

       SELECT COUNT(EMPLOYEE_ID) AS NO_OF_EMPLOYEES FROM EMPLOYEE WHERE MONTH(HIRE_DATE)
       IN (1, 9);

-- 9. How many employees joined in 1985?

       SELECT COUNT(EMPLOYEE_ID) AS NO_OF_EMPLOYEES FROM EMPLOYEE WHERE YEAR(HIRE_DATE) =
       1985;

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

```
SELECT COUNT(EMPLOYEE_ID) AS NO_OF_EMPLOYEES,MONTH(HIRE_DATE) AS
MONTH,YEAR(HIRE_DATE) AS YEAR FROM EMPLOYEE
GROUP BY MONTH(HIRE_DATE), YEAR(HIRE_DATE)HAVING YEAR(HIRE_DATE) = 1985;


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

SELECT COUNT(EMPLOYEE_ID) AS NO_OF_EMPLOYEES, YEAR(HIRE_DATE) AS YEAR FROM
EMPLOYEE GROUP BY YEAR(HIRE_DATE)HAVING YEAR(HIRE_DATE) = 1985;



-- 12. Which is the Department ID having greater than or equal to 3 employees joining in
April 1985?

SELECT DEPARTMENT_ID,MONTH(HIRE_DATE) AS MONTH,YEAR(HIRE_DATE) AS YEAR FROM
EMPLOYEE GROUP BY Department_ID, MONTH(HIRE_DATE), YEAR(HIRE_DATE)HAVING
MONTH(HIRE_DATE) = 4
AND YEAR(HIRE_DATE) = 1985 AND COUNT(*) >= 3;

-- JOINS

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

SELECT E.EMPLOYEE_ID, E.LAST_NAME, E.FIRST_NAME, D.Name AS DepartmentName, L.City
AS LocationCity
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
JOIN LOCATION L ON D.LOCATION_ID = L.Location_ID;

-- 2. Display employees with their designations.

SELECT E.EMPLOYEE_ID, E.LAST_NAME, E.FIRST_NAME, J.DESIGNATION
FROM EMPLOYEE E
JOIN JOB J ON E.JOB_ID = J.JOB_ID;


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

SELECT E.EMPLOYEE_ID, E.LAST_NAME, E.FIRST_NAME, D.Name AS DepartmentName, L.City
AS LocationCity
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
JOIN LOCATION L ON D.LOCATION_ID = L.Location_ID;

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

SELECT D.Department_Id, D.Name AS DepartmentName, COUNT(E.EMPLOYEE_ID) AS
EmployeeCount
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DEPARTMENT_ID = E.DEPARTMENT_ID
GROUP BY D.Department_Id, D.Name;


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

SELECT COUNT(E.EMPLOYEE_ID) AS EmployeeCount
```

```
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPARTMENT_ID = E.DEPARTMENT_ID
WHERE D.Name = 'Sales';
```

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

```
SELECT D.Name AS DepartmentName, COUNT(E.EMPLOYEE_ID) AS EmployeeCount
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPARTMENT_ID = E.DEPARTMENT_ID
GROUP BY D.Name HAVING COUNT(E.EMPLOYEE_ID) >= 5 ORDER BY DepartmentName ASC;
```

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

```
SELECT COUNT(DISTINCT JOB_ID) AS JobCount, DESIGNATION FROM JOB GROUP BY
DESIGNATION;
```

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

```
SELECT COUNT(E.EMPLOYEE_ID) AS EmployeeCount FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
JOIN LOCATION L ON D.LOCATION_ID = L.Location_ID
WHERE L.City = 'New York';
```

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

```
SELECT E.EMPLOYEE_ID, E.LAST_NAME, E.FIRST_NAME, E.SALARY,
CASE
        WHEN E.SALARY >= 5000 THEN 'A'
        WHEN E.SALARY >= 3000 AND E.SALARY < 5000 THEN 'B'
        WHEN E.SALARY >= 2000 AND E.SALARY < 3000 THEN 'C'
        ELSE 'D'
 END AS SalaryGrade
FROM EMPLOYEE E;
```

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

```
SELECT
CASE
        WHEN SALARY >= 5000 THEN 'A'
        WHEN SALARY >= 3000 AND SALARY < 5000 THEN 'B'
        WHEN SALARY >= 2000 AND SALARY < 3000 THEN 'C'
        ELSE 'D'
END AS Salary, COUNT(*) AS EmployeeCount FROM EMPLOYEE GROUP BY Salary;
```

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

```
SELECT
        CASE
                WHEN SALARY >= 5000 THEN 'A'
```

```
                WHEN SALARY >= 3000 AND SALARY < 5000 THEN 'B'
                WHEN SALARY >= 2000 AND SALARY < 3000 THEN 'C'
            ELSE 'D'
            END AS Salary, COUNT(*) AS EmployeeCount FROM EMPLOYEE WHERE SALARY
           BETWEEN 2000 AND 5000 GROUP BY Salary;


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

      SELECT E.EMPLOYEE_ID, E.LAST_NAME, E.FIRST_NAME, D.Name AS DepartmentName
      FROM EMPLOYEE E JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID WHERE
      D.Name IN ('Sales', 'Operations');

-- SET OPERATORS

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


      SELECT DISTINCT JOB.DESIGNATION FROM EMPLOYEE
      JOIN DEPARTMENT ON EMPLOYEE.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
      JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID WHERE DEPARTMENT.Name = 'Sales'
      UNION
      SELECT DISTINCT JOB.DESIGNATION FROM EMPLOYEE
      JOIN DEPARTMENT ON EMPLOYEE.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
      JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID WHERE DEPARTMENT.Name = 'Accounting';

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

      SELECT DISTINCT JOB.DESIGNATION FROM EMPLOYEE
      JOIN DEPARTMENT ON EMPLOYEE.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
      JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID WHERE DEPARTMENT.Name IN ('Sales',
      'Accounting');


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

      SELECT DISTINCT JOB.DESIGNATION FROM EMPLOYEE
      JOIN DEPARTMENT ON EMPLOYEE.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
      JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID WHERE DEPARTMENT.Name = 'Research'
      INTERSECT
      SELECT DISTINCT JOB.DESIGNATION FROM EMPLOYEE
      JOIN DEPARTMENT ON EMPLOYEE.DEPARTMENT_ID = DEPARTMENT.DEPARTMENT_ID
      JOIN JOB ON EMPLOYEE.JOB_ID = JOB.JOB_ID
      WHERE DEPARTMENT.Name = 'Accounting' ORDER BY JOB.DESIGNATION ASC;


-- SUBQUERIES

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

      SELECT EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY FROM EMPLOYEE
      WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEE);

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

      SELECT EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY FROM EMPLOYEE
      WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE Name = 'Sales');
```

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

	SELECT EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY 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 EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY FROM EMPLOYEE
	WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE Location_ID IN
	(SELECT Location_ID FROM LOCATION WHERE City = 'New York'));

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

	SELECT COUNT(*) AS EmployeeCount 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 * 1.1
	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 TOP 1 * FROM EMPLOYEE WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE)
	ORDER BY SALARY DESC;


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

	SELECT * FROM EMPLOYEE ORDER BY SALARY DESC OFFSET 4 ROWS FETCH NEXT 1 ROWS ONLY;

	WITH RankedEmployees AS (
	 SELECT
		EMPLOYEE_ID,
		LAST_NAME,
		FIRST_NAME,
		SALARY,
		ROW_NUMBER() OVER (ORDER BY SALARY DESC) AS SalaryRank
	FROM EMPLOYEE
	)
	SELECT * FROM RankedEmployees WHERE SalaryRank = 5;

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

	SELECT * FROM EMPLOYEE WHERE SALARY > ALL (SELECT SALARY FROM EMPLOYEE WHERE
	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 * FROM EMPLOYEE E1 WHERE SALARY > (SELECT MIN(SALARY) FROM EMPLOYEE E2
        WHERE E1.DEPARTMENT_ID = E2.DEPARTMENT_ID);

-- Departments with no employees
        SELECT DISTINCT D.DEPARTMENT_ID, D.Name AS DepartmentName FROM DEPARTMENT D
        WHERE NOT EXISTS (SELECT 1 FROM EMPLOYEE E WHERE E.DEPARTMENT_ID =
        D.DEPARTMENT_ID);


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

        SELECT DEPARTMENT_ID, Name AS DepartmentName FROM DEPARTMENT WHERE DEPARTMENT_ID
        NOT IN (SELECT DISTINCT DEPARTMENT_ID FROM EMPLOYEE);


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

        SELECT * FROM EMPLOYEE E1 WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEE E2
        WHERE E1.DEPARTMENT_ID = E2.DEPARTMENT_ID);
```