DAY 8

```
PRN : 200243020003
```

Sub-queries and Co-related subqueries

1. The HR department wants to determine the names of all the employees who were hired after Davies. Create a query to display the name and hire date of any employee hired after employee Davies.

```
SELECT
    LAST_NAME,
    HIRE_DATE
FROM
    EMPLOYEES
WHERE
    HIRE_DATE > (
        SELECT
            HIRE_DATE
        FROM
            EMPLOYEES
        WHERE
            LAST_NAME = 'Davies')
    ORDER BY
        HIRE_DATE;
```

2. Display employee details who earns more than employee 'Abel'

```
SELECT
    LAST_NAME,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY > (
        SELECT
            SALARY
        FROM
            EMPLOYEES
        WHERE
            LAST_NAME = 'Abel');
```

3. Display empid,last_name,salary,job for emloyees working in department same as 'Taylor' and earns less than him

```
SELECT
    EMPLOYEE_ID,
    LAST_NAME,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY < (
        SELECT
            SALARY
        FROM
            EMPLOYEES
        WHERE
            LAST_NAME = 'Taylor')
        AND DEPARTMENT_ID = (
            SELECT
                DEPARTMENT_ID
            FROM
                EMPLOYEES
            WHERE
                LAST_NAME = 'Taylor');
```

4. Display employee details who earns minimum salary

```
SELECT
    DEPARTMENT_ID,
    MIN(SALARY)
FROM
    EMPLOYEES
GROUP BY
    DEPARTMENT_ID;
```

5. Display department-wise dept_id,salary for those departments whose max salary is more than 10000

```
SELECT
    DEPARTMENT_ID,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY > ALL (
        SELECT
            MAX(SALARY)
        FROM
            EMPLOYEES
        WHERE
            SALARY = 10000);
```

6. Display empid,dep-id,salary for those employees who earn maximum in his department. Eg: king earns highest salary ($24000) in his department(90)

```
SELECT
    concat(concat(to_char(MAX(SALARY), '$99,999'), ' in his department '),
DEPARTMENT_ID)
FROM
    EMPLOYEES
GROUP BY
    DEPARTMENT_ID
HAVING
    MAX(SALARY)
    in(
        SELECT
            max(SALARY)
            FROM EMPLOYEES
        GROUP BY
            DEPARTMENT_ID)
ORDER BY
    DEPARTMENT_ID;
```

7. display second highest salary

```
SELECT
    LAST_NAME,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY = (
        SELECT
            MAX(SALARY)
        FROM
            EMPLOYEES
        WHERE
            SALARY < (
                SELECT
                    MAX(SALARY)
                FROM
                    EMPLOYEES));
```

8. Display duplicate employee names

```
SELECT
    first_name,
    last_name
FROM
    employees
    OUTER
WHERE
    2 <= (
        SELECT
            count(last_name)
        FROM
            employees
            INNER
        WHERE
            inner.last_name = outer.last_name);
```

9. Display n^th highest salary where value of 'n' is given by user (i.e if user enters 3 then, 3 highest salaries should be displayed)

```
SELECT
    salary
FROM
    employees
    OUTER
WHERE (& n) >= (
    SELECT
        count(DISTINCT inner.salary)
    FROM
        employees
        INNER
    WHERE
        outer.salary <= inner.salary);
```

10. The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

```
SELECT
    DEPARTMENT_ID,
    LAST_NAME,
    HIRE_DATE
FROM
    EMPLOYEES
WHERE
    JOB_ID in(
        SELECT
            JOB_ID FROM EMPLOYEES
        WHERE
```

```
            LAST_NAME = '&name')
    AND LAST_NAME <> 'name';
```

11. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

```
SELECT
    EMPLOYEE_ID,
    LAST_NAME,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY > (
        SELECT
            AVG(SALARY)
        FROM
            EMPLOYEES)
    ORDER BY
        SALARY;
```

12. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains the letter ?u.?

```
SELECT
    EMPLOYEE_ID,
    LAST_NAME,
    DEPARTMENT_NAME
FROM
    EMPLOYEES,
    DEPARTMENTS
WHERE
    DEPARTMENT_NAME LIKE '%u%';
```

13. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

```
SELECT DISTINCT
    EMPLOYEE_ID,
    LAST_NAME,
    LOCATION_ID
FROM
    EMPLOYEES,
    DEPARTMENTS
```

```
WHERE
    LOCATION_ID = 1700;
```

14. Create a report for HR that displays the last name and salary of every employee who reports to King.

```
SELECT
    SALARY,
    DEPARTMENT_ID
FROM
    EMPLOYEES
WHERE
    DEPARTMENT_ID = ANY (
        SELECT
            DEPARTMENT_ID
        FROM
            EMPLOYEES
        WHERE
            LAST_NAME = 'King');
```

15. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

```
SELECT
    DEPARTMENT_ID,
    LAST_NAME,
    JOB_ID
FROM
    EMPLOYEES
WHERE
    DEPARTMENT_ID = (
        SELECT
            DEPARTMENT_ID
        FROM
            DEPARTMENTS
        WHERE
            DEPARTMENT_NAME = 'Executive');
```

16. Display only stateid and statename for location under location Toronto

```
SELECT
    stateid,
    statename
FROM
    states
WHERE
    stateid = (
        SELECT
            stateid
        FROM
            states
        WHERE
            locationid = (
                SELECT
                    locationid
                FROM
                    LOCATION
                WHERE
                    locationname = 'TORONTO'));
```

17. Display statename for city "Fredericktown"

```
SELECT
    STATE_PROVINCE AS statename
FROM
    locations
WHERE
    city = 'Fredericktown';
```

18. Write a query to display the last name, department number, and salary of any employee whose department number and salary both match the department number and salary of any employee who earns a commission

```
SELECT
    LAST_NAME,
    SALARY,
    DEPARTMENT_ID
FROM
    EMPLOYEES
WHERE
    COMMISSION_PCT IS NOT NULL;
```

19. Display the last name, department name, and salary of any employee whose salary and commission match the salary and commission of any employee l ocated in location ID 1700

```
SELECT
    last_name,
    department_name,
    salary
FROM
    employees
    JOIN departments USING (department_id)
```

```
WHERE
    salary in(
        SELECT
            salary FROM employees
            JOIN departments
            USING (department_id)
            JOIN locations
            USING (location_id)
        WHERE
            location_id = 1700);
```

20. Create a query to display the last name, hire date, and salary for all employees who have the same salary and commission as Kochhar. Note: Do not display Kochhar in the result set.

```
SELECT
    LAST_NAME,
    HIRE_DATE,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY = (
        SELECT
            SALARY
        FROM
            EMPLOYEES
        WHERE
            LAST_NAME = 'Kochhar')
        AND LAST_NAME <> 'Kochhar';
```

21. Create a query to display the employees who earn a salary that is higher than the salary of all of the sales managers (JOB_ID = 'SA_MAN'). Sort the results on salary from highest to lowest

```
SELECT
    LAST_NAME,
    SALARY
FROM
    EMPLOYEES
WHERE
    SALARY > ALL (
        SELECT
            SALARY
        FROM
            EMPLOYEES
        WHERE
            JOB_ID = 'SA_MAN')
    ORDER BY
        SALARY DESC;
```

22. Display the details of the employee ID, last name, and department ID of those employees who live in cities whose name begins with T.

```
SELECT
    EMPLOYEE_ID,
    LAST_NAME,
    DEPARTMENT_ID
FROM
    EMPLOYEES
WHERE
    DEPARTMENT_ID in(
        SELECT
            DEPARTMENT_ID FROM DEPARTMENTS
        WHERE
            LOCATION_ID in(
                SELECT
                    LOCATION_ID FROM LOCATIONS
                WHERE
                    CITY LIKE 'T%'));
```

23. Write a query to find all employees who earn more than the average salary in their departments. Display last name, salary, department ID, and the average salary for the department. Sort by average salary. Use aliases for the columns retrieved by the query as shown in the sample output.

```
SELECT
    employee_id,
    last_name,
    DEPARTMENT_ID,
    salary
FROM
    employees
    OUTER
WHERE
    salary > (
        SELECT
            avg(salary)
        FROM
            employees
            INNER
        GROUP BY
            department_id
        HAVING
            inner.department_id = outer.department_id)
    ORDER BY
        department_id;
```

24. Write a query to display the employee ID, last names, and department names of all employees. Note: Use a scalar subquery to retrieve the department name in the SELECT statement.

```
SELECT
    employee_id,
    last_name,
    department_name
FROM
    employees
    JOIN departments USING (department_id)
WHERE
    department_id in(
        SELECT
            department_id FROM departments);
```

25. Write a query to display the last names of the employees who have one or more coworkers in their departments with later hire dates but higher salaries.

```
SELECT
    last_name
FROM
    employees
    OUTER
WHERE
    department_id in(
        SELECT
            department_id FROM employees
            INNER
        WHERE
            inner.department_id = outer.department_id
            and(inner.salary > outer.salary
                AND inner.hire_date > outer.hire_date));
```