EXP NO:

NESTED QUERIES

DATE:

AIM:

To implement **nested queries (subqueries)** in SQL for data retrieval

Description:

1.Subquery in the WHERE Clause:

Filters rows based on a condition that depends on the result of another query. Commonly used with operators like >, <, IN, and EXISTS.

2. Subquery in the FROM Clause:

Uses a subquery to create a temporary result set, which is then used as a table for the outer query.

3. Subquery in the SELECT Clause:

Calculates or retrieves derived values for each row by using a subquery in the SELECT statement.

4. Correlated Subquery:

A subquery that refers to a column from the outer query. It is executed for each row in the outer query.

5. Subquery in the HAVING Clause:

Filters groups after aggregation by comparing the aggregate result to the result of a subquery.

6. Using EXISTS with a Subquery:

Checks whether a subquery returns any rows. If the subquery returns at least one row, the outer query includes those records.

7. Using IN with Subquery:

Filters rows by checking if a column value matches any value returned by a subquery.

8. Subquery in the INSERT Statement:

Inserts rows into a table based on the result of a subquery.

9. Subquery in the UPDATE Statement:

Updates column values based on the result of a subquery.

10. Subquery in the DELETE Statement:

Deletes rows from a table based on a condition derived from another query.

<u>SYNTAX:</u>

1.Subquery in the WHERE Clause:

```
SELECT column1, column2, ...

FROM table_name

WHERE column_name operator (SELECT column_name FROM table_name
WHERE condition);
```

2. Subquery in the FROM Clause:

```
SELECT column1, column2, ...

FROM (SELECT column1, column2, ... FROM table_name WHERE condition) AS
subquery

WHERE condition;
```

3. Subquery in the SELECT Clause:

```
SELECT column1, column2, ...,

 (SELECT aggregate_function(column) FROM table_name WHERE condition) AS alias_name

FROM table_name;
```

4. Correlated Subquery:

```
SELECT column1, column2, ...

FROM table1 t1

WHERE column_name operator (SELECT column_name FROM table2 t2 WHERE t2.column_name = t1.column_name);
```

5. Subquery in the HAVING Clause:

```
SELECT column1, aggregate_function(column) AS alias_name

FROM table_name

GROUP BY column1

HAVING aggregate_function(column) operator (SELECT aggregate_function(column) FROM table_name WHERE condition);
```

6. Using EXISTS with a Subquery:

```
SELECT column1, column2, ...

FROM table_name t1

WHERE EXISTS (SELECT 1 FROM table_name t2 WHERE t2.column_name = t1.column_name);
```

7. Using IN with Subquery:

```
SELECT column1, column2, ...

FROM table_name

WHERE column_name IN (SELECT column_name FROM table_name WHERE condition);
```

8. Subquery in the INSERT Statement:

```
INSERT INTO table_name (column1, column2, ...)

SELECT column1, column2, ...

FROM table_name

WHERE condition;
```

9. Subquery in the UPDATE Statement:

```
UPDATE table_name

SET column1 = value, column2 = value

WHERE column_name = (SELECT column_name FROM table_name WHERE condition);
```

10. Subquery in the DELETE Statement:

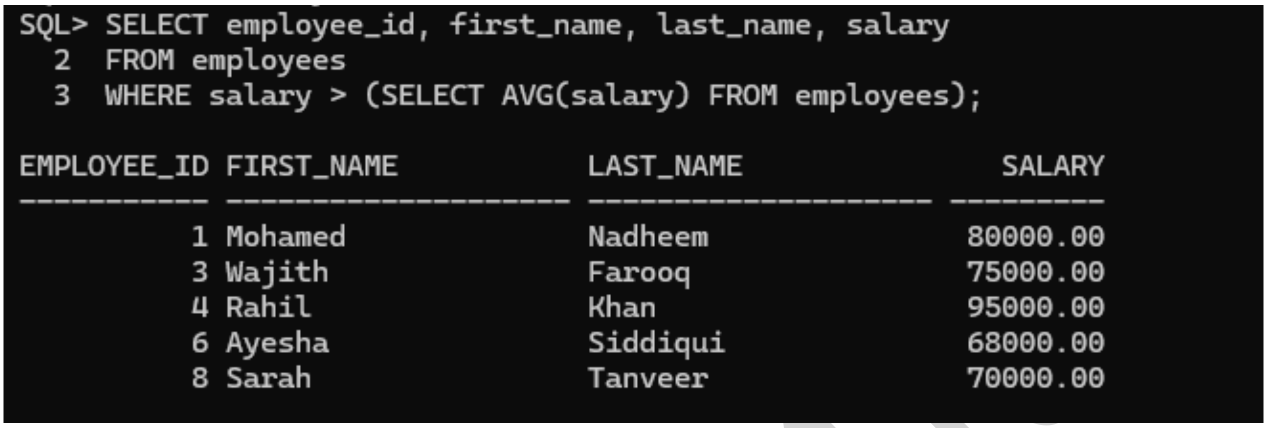
```
DELETE FROM table_name

WHERE column_name NOT IN (SELECT column_name FROM table_name WHERE condition);
```

PROGRAM & OUTPUT:

SQL> SELECT employee_id, first_name, last_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);



```
SQL> SELECT employee_id, first_name, last_name, salary
  2  FROM employees
  3  WHERE salary > (SELECT AVG(salary) FROM employees);

EMPLOYEE_ID FIRST_NAME           LAST_NAME                SALARY
----------- -------------------- -------------------- ----------
          1 Mohamed              Nadheem                80000.00
          3 Wajith               Farooq                 75000.00
          4 Rahil                Khan                   95000.00
          6 Ayesha               Siddiqui               68000.00
          8 Sarah                Tanveer                70000.00
```

SQL> SELECT department_id, avg_salary

FROM (

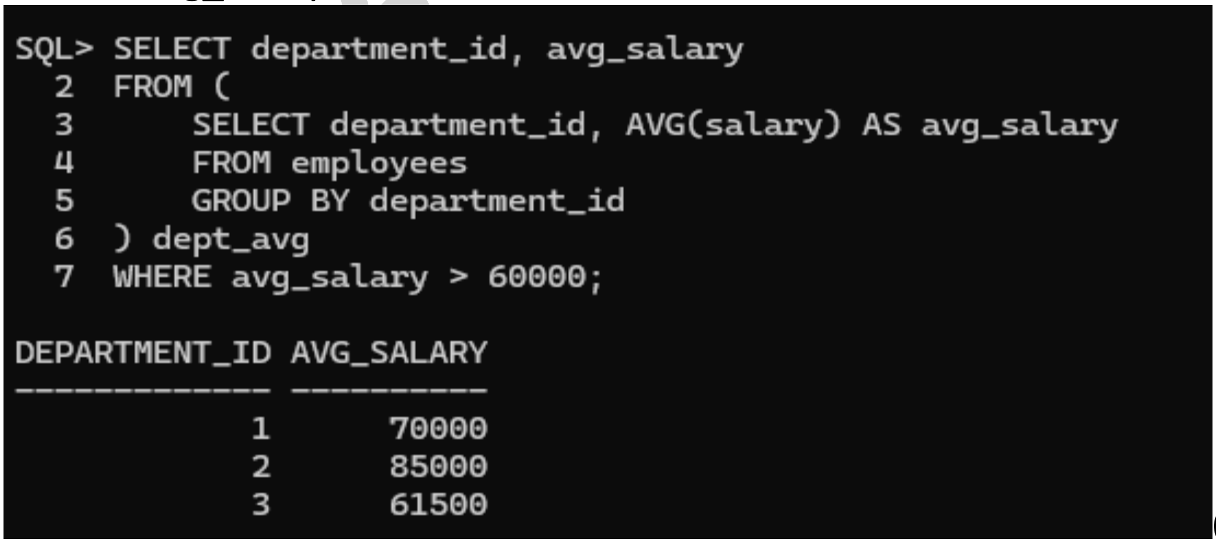
   SELECT department_id, AVG(salary) AS avg_salary

   FROM employees

   GROUP BY department_id

) dept_avg

WHERE avg_salary > 6000



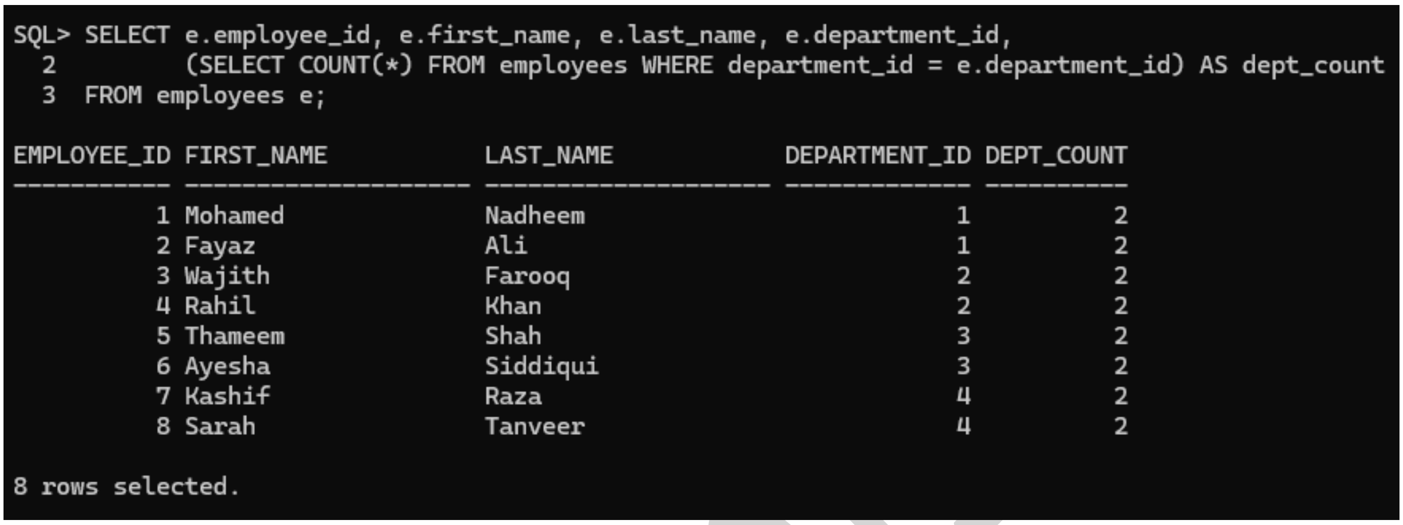
```
SQL> SELECT department_id, avg_salary
  2  FROM (
  3      SELECT department_id, AVG(salary) AS avg_salary
  4      FROM employees
  5      GROUP BY department_id
  6  ) dept_avg
  7  WHERE avg_salary > 60000;

DEPARTMENT_ID AVG_SALARY
------------- ----------
            1      70000
            2      85000
            3      61500
```

0;

SQL> SELECT e.employee_id, e.first_name, e.last_name, e.department_id,

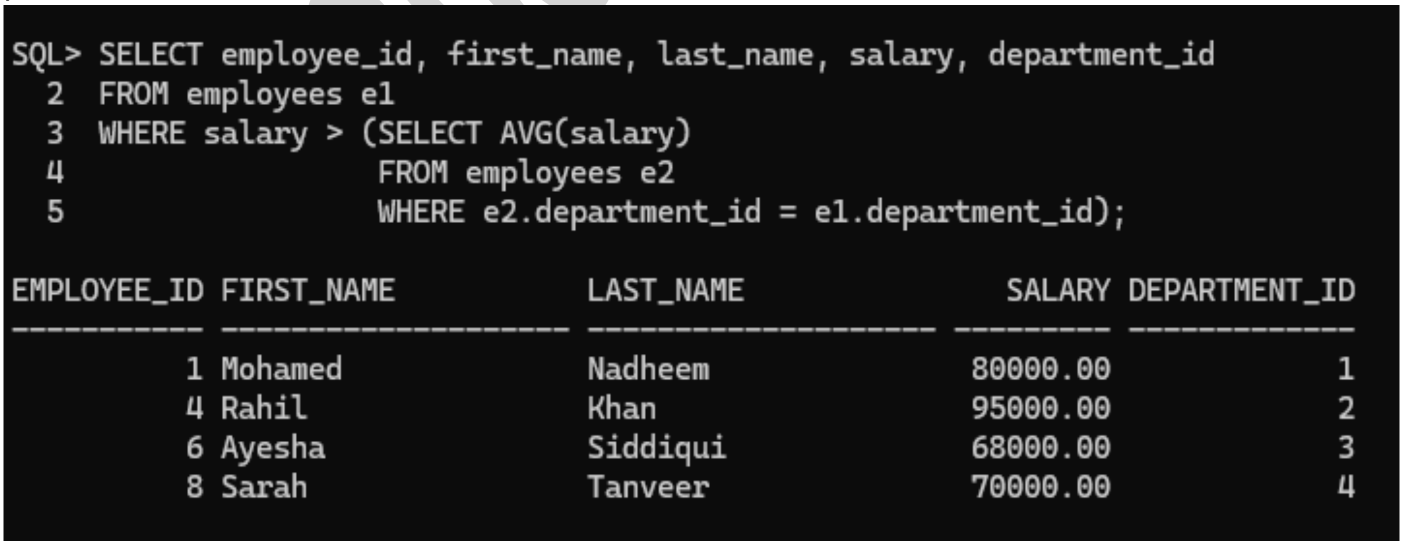
(SELECT COUNT(*) FROM employees WHERE department_id = e.department_id) AS dept_count

FROM employees e;



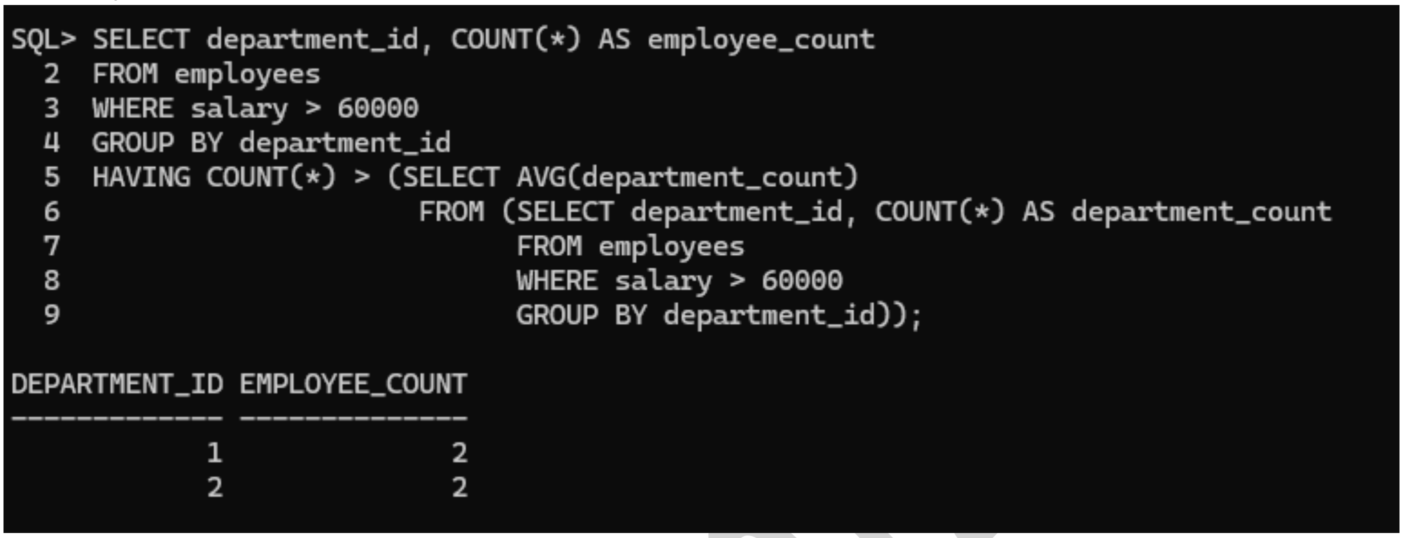
```
SQL> SELECT e.employee_id, e.first_name, e.last_name, e.department_id,
  2        (SELECT COUNT(*) FROM employees WHERE department_id = e.department_id) AS dept_count
  3  FROM employees e;

EMPLOYEE_ID FIRST_NAME           LAST_NAME            DEPARTMENT_ID DEPT_COUNT
----------- -------------------- -------------------- ------------- ----------
          1 Mohamed              Nadheem                          1          2
          2 Fayaz                Ali                              1          2
          3 Wajith               Farooq                           2          2
          4 Rahil                Khan                             2          2
          5 Thameem              Shah                             3          2
          6 Ayesha               Siddiqui                         3          2
          7 Kashif               Raza                             4          2
          8 Sarah                Tanveer                          4          2

8 rows selected.
```

SQL> SELECT employee_id, first_name, last_name, salary, department_id
FROM employees e1
WHERE salary > (
    SELECT AVG(salary)
    FROM employees e2
    WHERE e2.department_id = e1.department_id
);



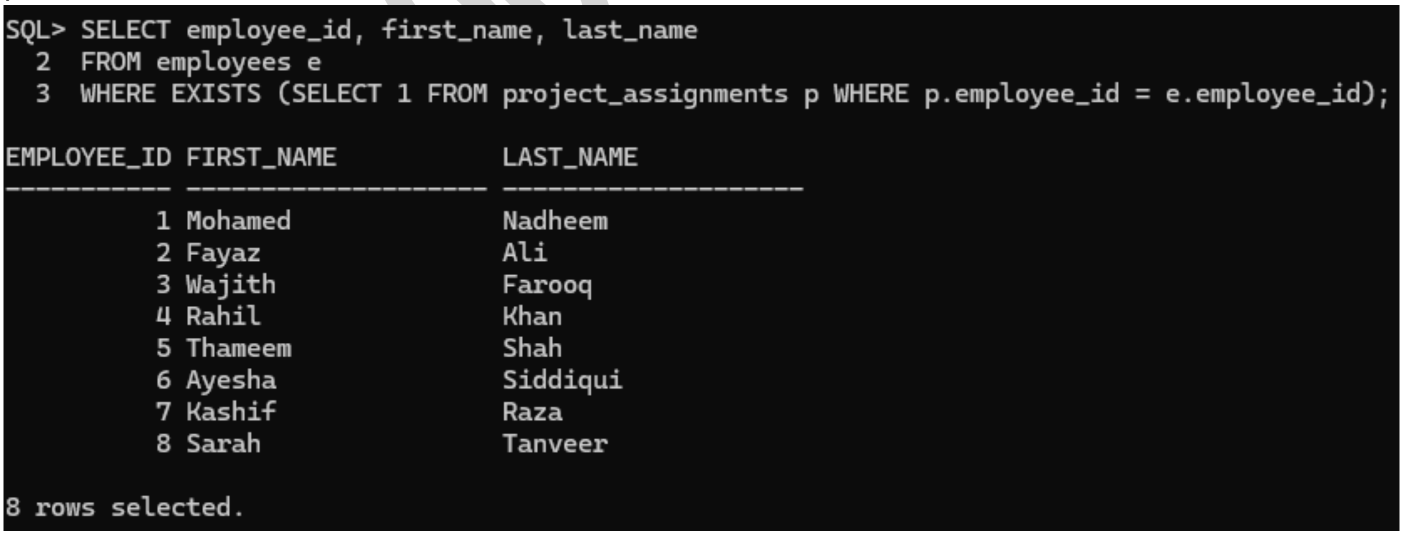
```
SQL> SELECT employee_id, first_name, last_name, salary, department_id
  2  FROM employees e1
  3  WHERE salary > (SELECT AVG(salary)
  4                  FROM employees e2
  5                  WHERE e2.department_id = e1.department_id);

EMPLOYEE_ID FIRST_NAME           LAST_NAME                SALARY DEPARTMENT_ID
----------- -------------------- -------------------- ---------- -------------
          1 Mohamed              Nadheem                80000.00             1
          4 Rahil                Khan                   95000.00             2
          6 Ayesha               Siddiqui               68000.00             3
          8 Sarah                Tanveer                70000.00             4
```

SQL> SELECT department_id, COUNT(*) AS employee_count
FROM employees
WHERE salary > 60000
GROUP BY department_id
HAVING COUNT(*) > (SELECT COUNT(*) FROM employees WHERE salary > 60000);



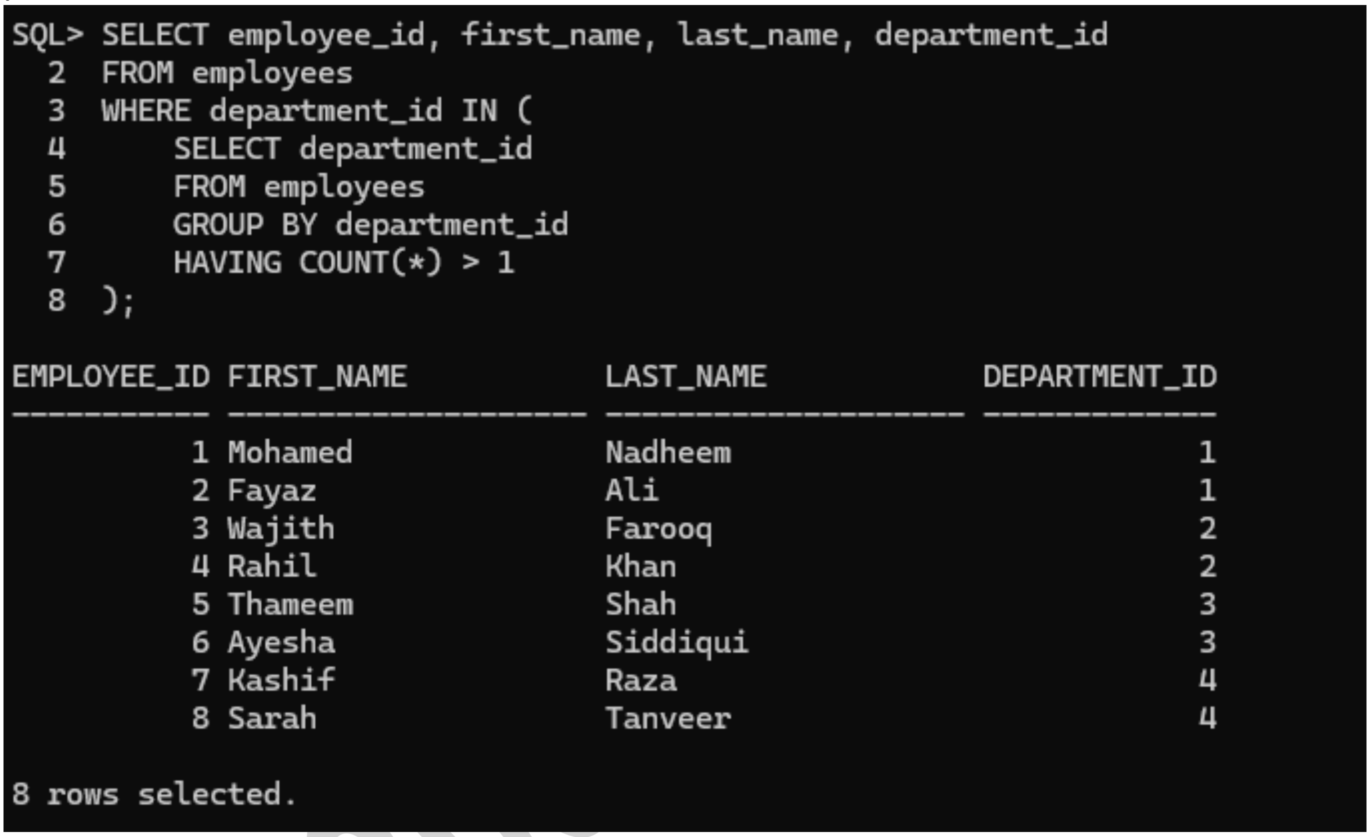
```
SQL> SELECT department_id, COUNT(*) AS employee_count
  2  FROM employees
  3  WHERE salary > 60000
  4  GROUP BY department_id
  5  HAVING COUNT(*) > (SELECT AVG(department_count)
  6                     FROM (SELECT department_id, COUNT(*) AS department_count
  7                           FROM employees
  8                           WHERE salary > 60000
  9                           GROUP BY department_id));

DEPARTMENT_ID EMPLOYEE_COUNT
------------- --------------
            1              2
            2              2
```

SQL> SELECT employee_id, first_name, last_name
FROM employees e
WHERE EXISTS (
   SELECT 1 FROM project_assignments p WHERE p.employee_id = e.employee_id
);



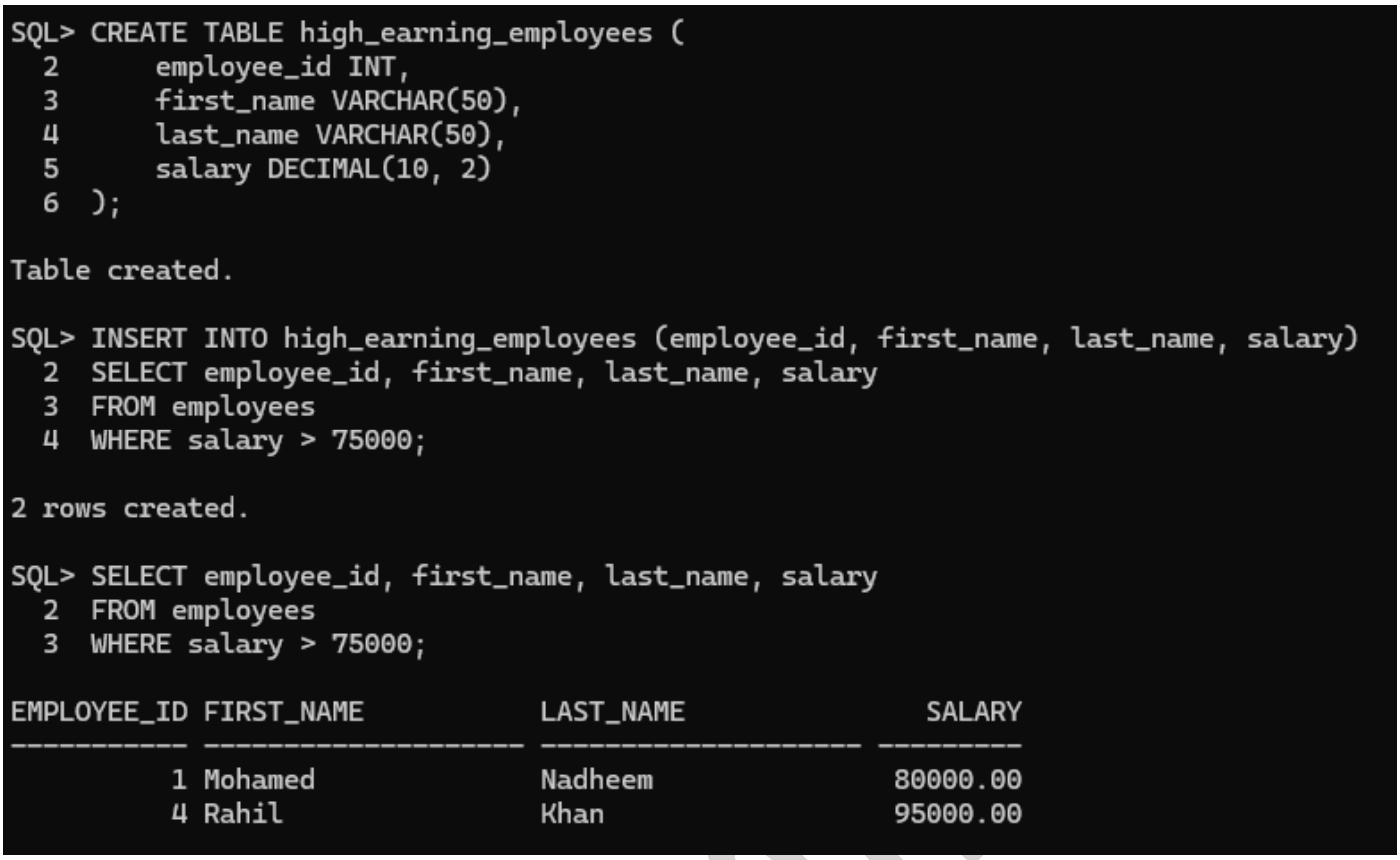
```
SQL> SELECT employee_id, first_name, last_name
  2  FROM employees e
  3  WHERE EXISTS (SELECT 1 FROM project_assignments p WHERE p.employee_id = e.employee_id);

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- --------------------
          1 Mohamed              Nadheem
          2 Fayaz                Ali
          3 Wajith               Farooq
          4 Rahil                Khan
          5 Thameem              Shah
          6 Ayesha               Siddiqui
          7 Kashif               Raza
          8 Sarah                Tanveer

8 rows selected.
```

```
SQL> SELECT employee_id, first_name, last_name, department_id
FROM employees
WHERE department_id IN (
   SELECT department_id
   FROM employees
   GROUP BY department_id
   HAVING COUNT(*) > 1
);
```



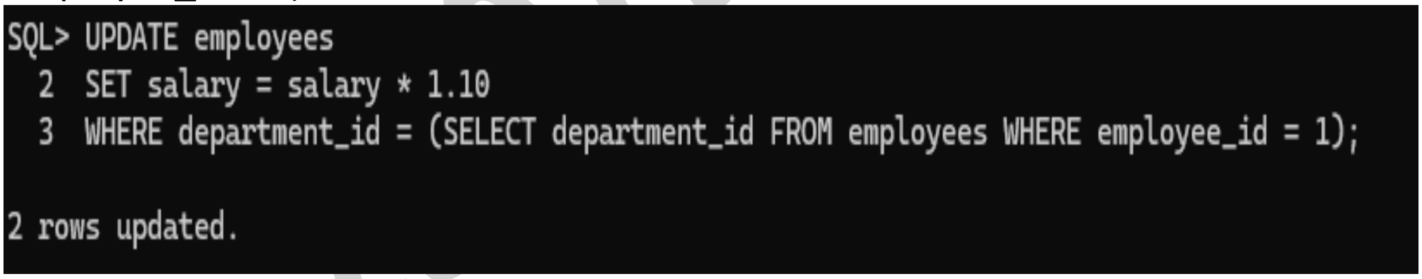
```
SQL> SELECT employee_id, first_name, last_name, department_id
  2  FROM employees
  3  WHERE department_id IN (
  4      SELECT department_id
  5      FROM employees
  6      GROUP BY department_id
  7      HAVING COUNT(*) > 1
  8  );

EMPLOYEE_ID FIRST_NAME           LAST_NAME            DEPARTMENT_ID
----------- -------------------- -------------------- -------------
          1 Mohamed              Nadheem                          1
          2 Fayaz                Ali                              1
          3 Wajith               Farooq                           2
          4 Rahil                Khan                             2
          5 Thameem              Shah                             3
          6 Ayesha               Siddiqui                         3
          7 Kashif               Raza                             4
          8 Sarah                Tanveer                          4

8 rows selected.
```

```
SQL> -- Create table
CREATE TABLE high_earning_employees (
   employee_id INT,
   first_name VARCHAR(50),
   last_name VARCHAR(50),
   salary DECIMAL(10, 2)
);
-- Insert query
INSERT INTO high_earning_employees (employee_id, first_name, last_name,
salary)
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE salary > 75000;
```



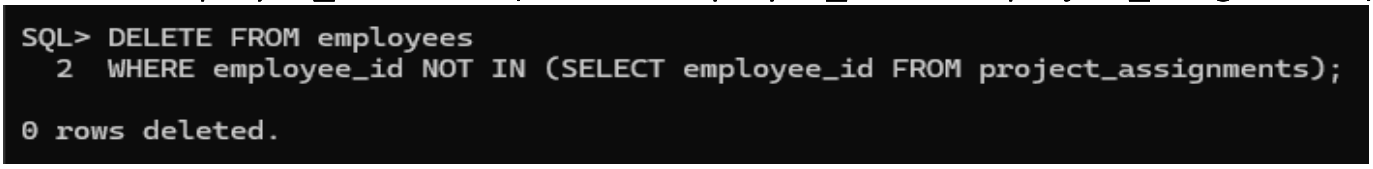
```
SQL> CREATE TABLE high_earning_employees (
  2      employee_id INT,
  3      first_name VARCHAR(50),
  4      last_name VARCHAR(50),
  5      salary DECIMAL(10, 2)
  6  );

Table created.

SQL> INSERT INTO high_earning_employees (employee_id, first_name, last_name, salary)
  2  SELECT employee_id, first_name, last_name, salary
  3  FROM employees
  4  WHERE salary > 75000;

2 rows created.

SQL> SELECT employee_id, first_name, last_name, salary
  2  FROM employees
  3  WHERE salary > 75000;

EMPLOYEE_ID FIRST_NAME           LAST_NAME                SALARY
----------- -------------------- -------------------- ----------
          1 Mohamed              Nadheem                80000.00
          4 Rahil                Khan                   95000.00
```

SQL> UPDATE employees
SET salary = salary * 1.10
WHERE department_id = (SELECT department_id FROM employees WHERE employee_id = 1);



```
SQL> UPDATE employees
  2  SET salary = salary * 1.10
  3  WHERE department_id = (SELECT department_id FROM employees WHERE employee_id = 1);

2 rows updated.
```

SQL> DELETE FROM employees
WHERE employee_id NOT IN (SELECT employee_id FROM project_assignments);



```
SQL> DELETE FROM employees
  2  WHERE employee_id NOT IN (SELECT employee_id FROM project_assignments);

0 rows deleted.
```

RESULT:

Thus the implementation of Nested queries has been executed successfully