Date: 17/9/25

EXERCISE-9

Sub queries

AIM: To define the use of sub queries

Objectives

After completing this lesson, you should be able to do the following:

- Define subqueries
- Describe the types of problems that subqueries can solve
- List the types of subqueries
- Write single-row and multiple-row subqueries

Using a Subquery to Solve a Problem
Who has a salary greater than Abel's?

Main query:
Which employees have salaries greater than Abel's salary?

Subquery:
What is Abel's salary?

Subquery Syntax

SELECT *select_list* FROM *table* WHERE *expr operator* (SELECT *select_list* FROM *table*);

- The subquery (inner query) executes once before the main query (outer query).

- The result of the subquery is used by the main query.

A subquery is a SELECT statement that is embedded in a clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries. They can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself.

You can place the subquery in a number of SQL clauses, including the following:

- WHERE clause
- HAVING clause
- FROM clause

In the syntax:
operator includes a comparison condition such as >, =, or IN

Note: Comparison conditions fall into two classes: single-row operators (>, =, >=, <, <>, <=) and multiple-row operators (IN, ANY, ALL). statement. The subquery generally executes first, and its output is used to complete the query condition for the main (or outer) query

Using a Subquery

SELECT last_name FROM employees WHERE salary > (SELECT salary FROM employees WHERE last_name = 'Abel');

Displays employees who are not IT programmers and whose salary is less than that of any IT programmer. The maximum salary that a programmer earns is $9,000.

< ANY means less than the maximum. >ANY means more than the minimum. =ANY is equivalent to IN.

Using the ALL Operator in Multiple-Row Subqueries

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL (SELECT salary FROM employees WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

Displays employees whose salary is less than the salary of all employees with a job ID of IT_PROG and whose job is not IT_PROG.

➢ ALL means more than the maximum, and <ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators.

Null Values in a Subquery

```
SELECT emp.last_name FROM employees emp
WHERE emp.employee_id NOT IN (SELECT mgr.manager_id FROM employees mgr);
```

Notice that the null value as part of the results set of a subquery is not a problem if you use the IN operator. The IN operator is equivalent to =ANY. For example, to display the employees who have subordinates, use the following SQL statement:

```
SELECT emp.last_name
FROM employees emp
WHERE emp.employee_id IN (SELECT mgr.manager_id FROM employees mgr);
```

Display all employees who do not have any subordinates:

```
SELECT last_name FROM employees
WHERE employee_id NOT IN (SELECT manager_id FROM employees WHERE manager_id IS
NOT NULL);
```

Find the Solution for the following:

1. 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 l_name, hire_date from employees where dept_id =
(Select dept_id from employees where l_name ='Zlotkey')
AND l_name != 'Zlotkey';

2. 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 emp_number, l_name, salary from employees
where salary >(Select AVG(salary) from employees)
order by salary ASC;

2)

emp-ID	l-name	salary
100	Priyan	100000
101	Kumar	120000
102	Prashanth	1300000

3)

emp-id	~~Last~~ l-name
104	Naruto

4)

l-name	DepartmentNumber	Job ID
Gates	90	1003

3. 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 a *u*.

```
Select emp - number, l - name FROM employees where
l_name like '%u%';
```

4. 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 e.l-name, e.dept-id, e.job-id from employees e
JOIN departments d on e.dept-id = d.dept-id
Where d.location-id =1700;
```

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

```
Select e.l-name, e.salary from employees e
JOIN employees manager ON e.manager-id = manager.emp-id
Where manager.l-name = 'King';
```

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

```
Select e.dept-id, e.l-name, e.job-id FROM employees
e JOIN departments d ON e.dept-id = d.dept-id
Where d.dept-name = 'Executive';
```

7. Modify the query 3 to display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a *u*.

```
Select emp-number, l-name, salary from employees
Where salary > (Select AVG(salary) FROM employees)
AND last-name LIKE '%u%';
```

5)

l_name	salary
Devi	130000

6)

Dept Number	l_name	Job_id
10	Naruto	1004

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	

RESULT:

Thus the concept of Sub queries is studied using mySQL.

Practice Questions

1. Ellen Abel is an employee who has received a $2,000 raise. Display her first name and last name, her current salary, and her new salary. Display both salaries with a $ and two decimal places. Label her new salary column AS New Salary.

```
Select f-name, l-name, salary, TO-CHAR(salary +2000,
'$99,999.00') AS "New Salary" FROM employees WHERE
f-name = 'Ellen' AND l-name = 'Abel';
```

2. On what day of the week and date did Global Fast Foods' promotional code 110 Valentine's Special begin?

```
Select TO-CHAR(start-date, 'Day, Month DD, YYYY') AS
"Promotional Date" FROM promotional-menus WHERE
promotional-code = 110;
```

3. Create one query that will convert 25-Dec-2004 into each of the following (you will have to convert 25-Dec-2004 to a date and then to character data):

December 25th, 2004
DECEMBER 25TH, 2004
25th december, 2004

```
TO-DATE('25-Dec-2004', 'DD-MON-YYYY')
TO-CHAR(TO-DATE('25-Dec-2004', 'DD-MON-YYYY'),
'Month DDth, YYYY'). TO-CHAR(TO-DATE('25-Dec-2004',
'DD-MON-YYYY'), 'month DD, YYYY'). TO-CHAR(TO-DATE('25-Dec
, ~~'DD-MON-YYYY~~ , -2004', 'DD-MON-YYYY'), 'DDth "of" Month YYYY')
```

4. Create a query that will format the DJs on Demand d_packages columns, low-range and high- range package costs, in the format $2500.00.

```
Select TO-CHAR(low-range, '$9999.00') AS "Low
Range", TO-CHAR(high-range, '$9999.00') AS "High
Range" FROM d-packages;
```

5. Convert JUNE192004 to a date using the fx format model.

```
Select TO-DATE('JUNE19 2004', 'fx MONTHDD YYYY')
FROM DUAL;
```

6. What is the distinction between implicit and explicit datatype conversion? Give an example of each.

Explicit conversion is when a user manually uses a conversion function to change the data type of an expression. Eg) TO-CHAR(SYSDATE, 'YYYY-MM-DD').

Implicit conversion is when the database automatically converts a datatype to another without the use of function. (Eg) Select '10'+5 FROM DUAL.

7. Why is it important from a business perspective to have datatype conversions?

Data type conversion is crucial for business for several reasons ~~that~~ like ensuring data integrity, report generation, data Integration and performance.

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	5
Total (10)	10
Faculty Signature	