NAME: Atharva Chavan
TE-A IT
ROLL NO: T1851010
PRN: 71901316L

<u>Group B: SQL & PL/SQL</u>
Assignment No. 7

Aim: Implement nested sub queries. Perform a test for set membership (in, not in), set comparison (<some, >=some, <all etc.) and set cardinality (unique, not unique).

Objective:
- To learn different types ofJoins.
- To implement different subqueries.

Theory :

MySQL JOINS are used with SELECT statement. It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.

There are three types of MySQL joins:

- MySQL INNER JOIN (or sometimes called simplejoin)
- MySQL LEFT OUTER JOIN (or sometimes called LEFTJOIN)
- MySQL RIGHT OUTER JOIN (or sometimes called RIGHTJOIN)

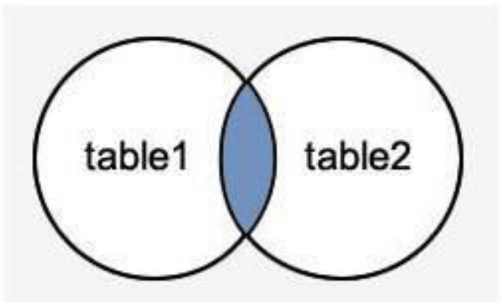
MySQL Inner JOIN (Simple Join)

The MySQL INNER JOIN is used to return all rows from multiple tables where the join condition is satisfied. It is the most common type of join.

Syntax:

```
SELECT columns
FROM table1

INNER JOIN table2
ON table1.column = table2.column;
```

Image representation:



Let's take an example:

Consider two tables "officers" and "students", having the following data.
Execute the following query:

```
SELECT officers.officer_name, officers.address, students.course_name
FROM officers
INNER JOIN students
ON officers.officer_id = students.student_id;
```

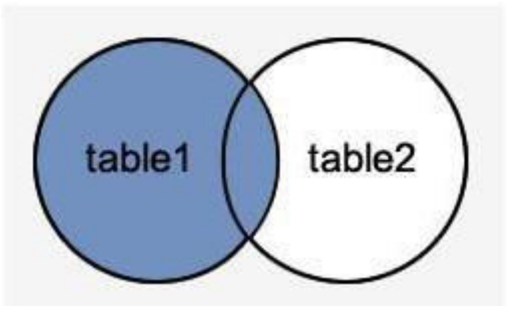
MySQL Left Outer Join

The LEFT OUTER JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

Syntax:

```
SELECT columns
FROM table1
LEFT [OUTER] JOIN table2
ON table1.column = table2.column;
```

Imagerepresentation:



Let's take anexample:

Consider two tables "officers" and "students", having the following data.
Execute the following query:

```
SELECT officers.officer_name, officers.address, students.course_name
FROM officers
LEFT JOIN students
ON officers.officer_id = students.student_id;
```

MySQL Right Outer Join

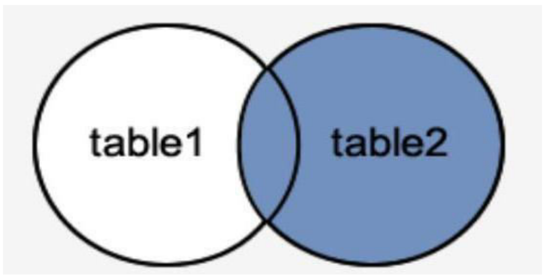
The MySQL Right Outer Join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where he join condition is fulfilled.

Syntax:

```
SELECT columns
```

```
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

Image representation:



Let's take an example:

Consider two tables "officers" and "students", having the following data.
Execute the following query:

```
SELECT      officers.officer_name,      officers.address,
            students.course_name,
students.student_name
FROM officers
RIGHT JOIN students
ON officers.officer_id = students.student_id;
```

SPECIAL OPERATOR:

<u>MySQL IN Condition</u>

The MySQL IN condition is used to reduce the use of multiple OR conditions in a SELECT, INSERT, UPDATE and DELETE statement.

Syntax:

expression IN(value1, value2,...... value_n);

Parameters:

expression: It specifies a value to test.
value1,value2, ... or value_n: These are the values to test against expression. If any ofthese values matches expression, then the IN condition will evaluate to true. This is a quick method to test if any one of the values matches expression.
Execute the following query:

```
SELECT *

FROM officers
WHERE officer_name IN ('Ajeet', 'Vimal', 'Deepika');
```

MySQL NOT Condition

The MySQL NOT condition is opposite of MySQL IN condition. It is used to negate a condition in a SELECT, INSERT, UPDATE or DELETE statement.

Syntax:

```
NOT condition
```

Parameter:

condition: It specifies the conditions that you want to negate.

MySQL NOT Operator with IN condition

Consider a table "officers", having the following data.
Execute the following query:

```
SELECT *

FROM officers
WHERE officer_name NOT IN ('Ajeet','Vimal','Deepika');
```

MySQL IS NULL Condition

MySQL IS NULL condition is used to check if there is a NULL value in the expression. It is used with SELECT, INSERT, UPDATE and DELETE statement.

Syntax:

```
expression IS NULL
```

Parameter:

expression: It specifies a value to test if it is NULL
Execute the following query:

```
SELECT *
FROM officers
WHERE officer_name IS NULL;
```

MySQL IS NOT NULL Condition

MySQL IS NOT NULL condition is used to check the NOT NULL value in the expression. It is used with SELECT, INSERT, UPDATE and DELETE statements.

Syntax:

expression IS NOT NULL

Parameter:

expression: It specifies a value to test if it is not NULL value.
Execute the following query:

```
SELECT *
FROM officers
WHERE officer_name IS NOT NULL;
```

SET OPERATORS:

The Set operator combines the result of 2 queries into a single result. The following are the operators:

- Union

Unionall

- Intersect

- Minus

LAB PRACTICE ASSIGNMENT:

Consider the following table structure for this assignment:

- Location(Location_Id integer, Reginal_Groupvarchar(20))
- Department (Department_Id, Name,Location_Id)
- Job(Job_IdInteger,FunctionVarchar(30))
- Employee(Employee_Id, Lastname ,Firstname, Middlename, Job_Id, Manager_Id, Hiredate, Salary,Department_Id)
- Loan(Employee_Id, Firstname , Loan_Amount)

LOCATION TABLE

LOCATION_ID	REGINAL_GROUP
122	New York
123	Dallas
124	Chicago
167	Boostan

DEPARTMENT TABLE

DEPARTMENT_ID	NAME	LOCATION_ID
10	Accounting	122

20	Research	124
30	Sale	123
40	Operation	164

JOB TABLE

JOB_ID	FUNCTION
667	Cleark
668	Staff
669	Analyst
670	Saleperson
671	Manager
672	President

EMPLOYEE TABLE

EMPLOYEE_ID	LASTNAME	FIRSTNAME	MIDDLENAME	JOB_ID	MANAGER_ID	HIREDATE	SALARY	DEPARTMENT_ID
7369	Smith	Jon	Q	667	7902	17-DEC-84	800	10
7499	Allen	Kevin	J	670	7698	20-FEB-85	1600	20
7505	Doyle	Jean	K	671	7839	04-APR-85	2850	20
7506	Dennis	Lynn	S	671	7839	15-MAY-85	2750	30
7507	Baker	Leslie	D	671	7839	10-JUN-85	2200	40
7521	wark	cynthia	D	670	7698	22-FEB-85	1250	10

Perform the following queries on the above table:

1) Perform all types of JOIN operations on Employee and Loantables.

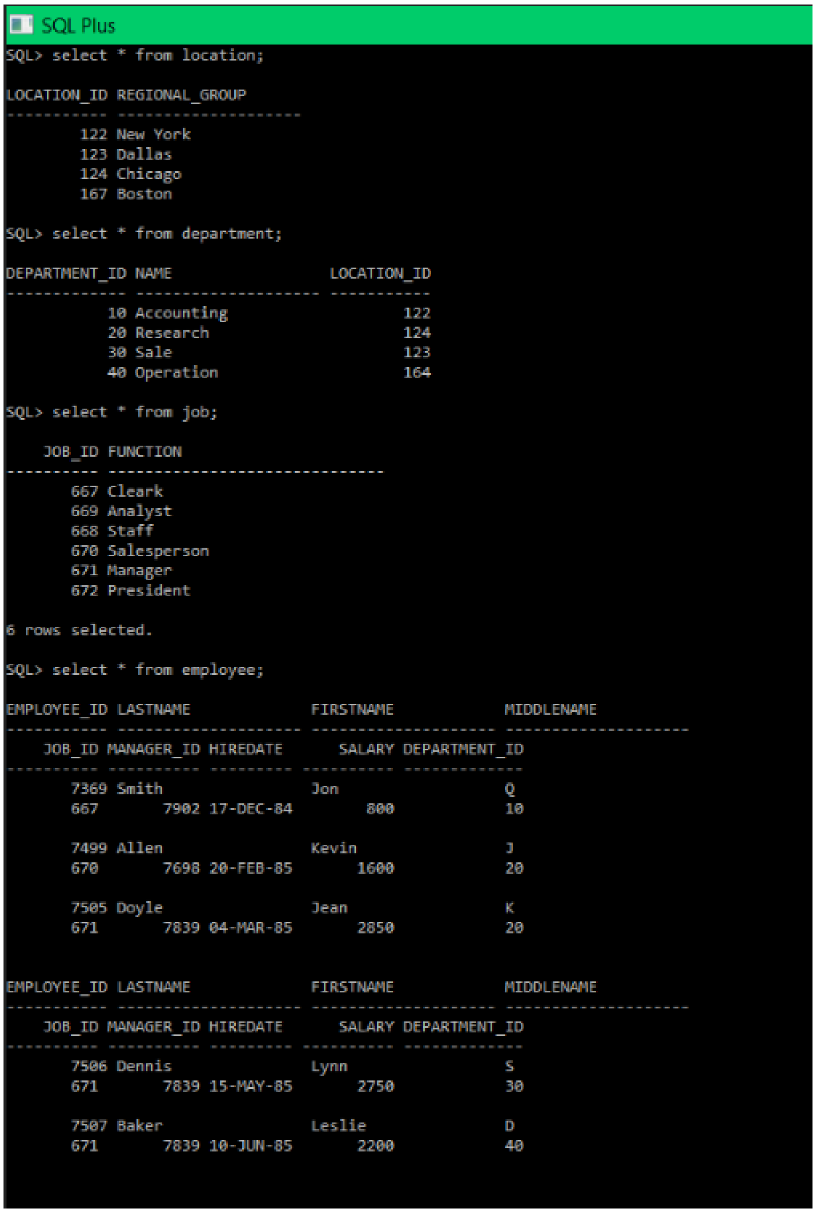
2) Perform all types of set operations on Employee and Loantables.
3) Find out no.of employees working in “Sales”department
4) Find out the employees who are not working in department 10 or30.
5) List out employee id, last name in descending order based on the salarycolumn.
6) How many employees who are working in different departments wise in the organization
7) List out the department id having at least fouremployees
8) Display the employee who got the maximumsalary.
9) Update the employees’ salaries, who are working as Clerk on the basis of10%.
10) Delete the employees who are working in accountingdepartment.
11) Find out whose department has notemployees.
12) List out the department wise maximum salary, minimum salary, average salary ofthe employees
13) How many employees who are joined in1985.
14) Display the employees who are working in “NewYork”
15) List our employees with their departmentnames

Conclusion:

We have implemented join, set operations, set cardinalities and nested sub queries.

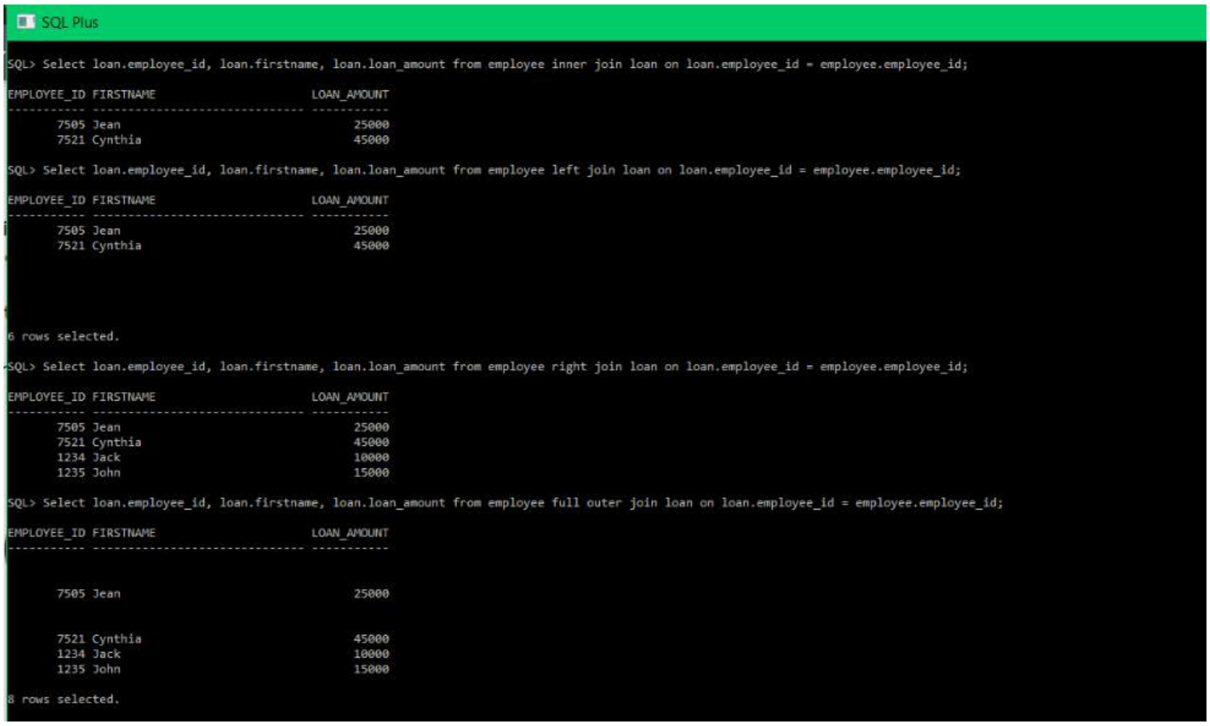
<u>Code & Output: -</u>

Tables:



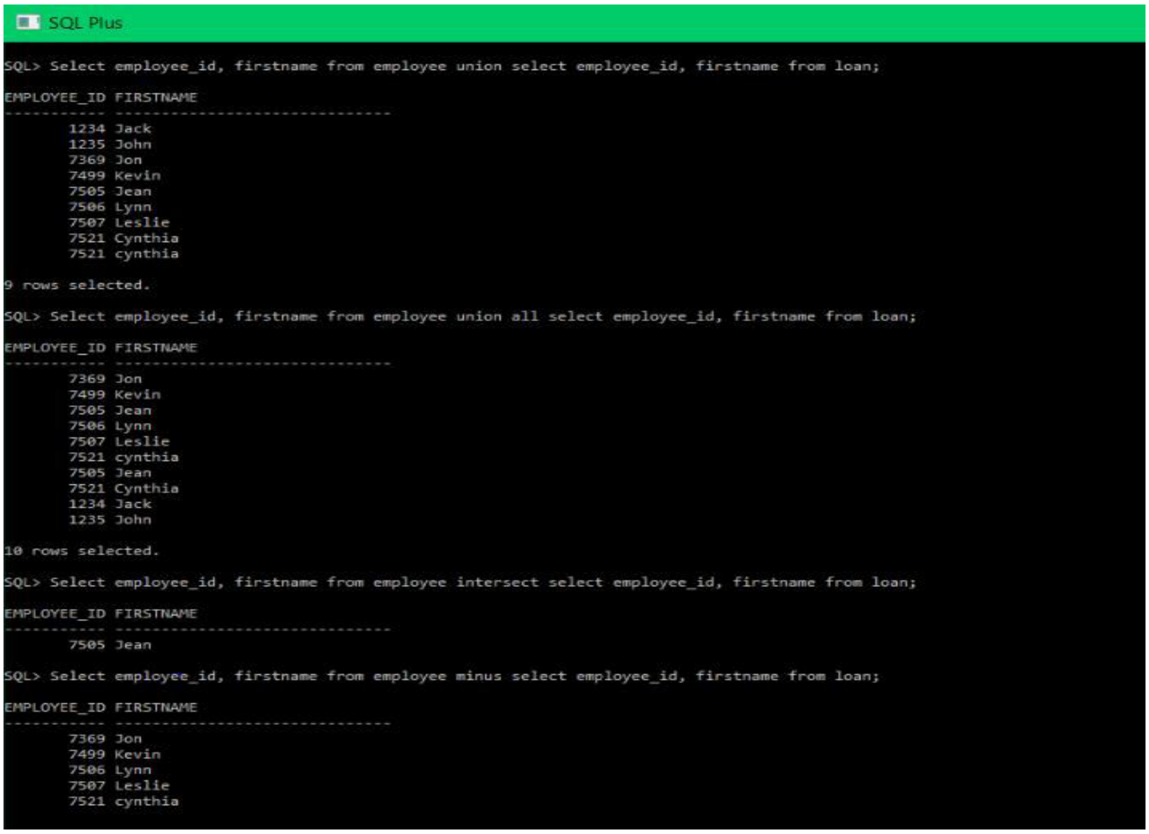
```
SQL> select * from location;

LOCATION_ID REGIONAL_GROUP
----------- --------------------
        122 New York
        123 Dallas
        124 Chicago
        167 Boston

SQL> select * from department;

DEPARTMENT_ID NAME                 LOCATION_ID
------------- -------------------- -----------
           10 Accounting                   122
           20 Research                     124
           30 Sale                         123
           40 Operation                    164

SQL> select * from job;

    JOB_ID FUNCTION
---------- ------------------------------
       667 Clearк
       669 Analyst
       668 Staff
       670 Salesperson
       671 Manager
       672 President

6 rows selected.

SQL> select * from employee;

EMPLOYEE_ID LASTNAME             FIRSTNAME            MIDDLENAME
----------- -------------------- -------------------- --------------------
    JOB_ID MANAGER_ID HIREDATE      SALARY DEPARTMENT_ID
---------- ---------- --------- ---------- -------------
       7369 Smith                Jon                  Q
       667       7902 17-DEC-84        800            10

       7499 Allen                Kevin                J
       670       7698 20-FEB-85       1600            20

       7505 Doyle                Jean                 K
       671       7839 04-MAR-85       2850            20


EMPLOYEE_ID LASTNAME             FIRSTNAME            MIDDLENAME
----------- -------------------- -------------------- --------------------
    JOB_ID MANAGER_ID HIREDATE      SALARY DEPARTMENT_ID
---------- ---------- --------- ---------- -------------
       7506 Dennis               Lynn                 S
       671       7839 15-MAY-85       2750            30

       7507 Baker                Leslie               D
       671       7839 10-JUN-85       2200            40
```

Q.1→



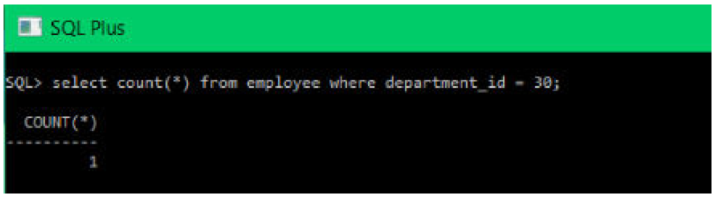
```
SQL> Select loan.employee_id, loan.firstname, loan.loan_amount from employee inner join loan on loan.employee_id = employee.employee_id;

EMPLOYEE_ID FIRSTNAME                      LOAN_AMOUNT
----------- ------------------------------ -----------
       7505 Jean                                 25000
       7521 Cynthia                              45000

SQL> Select loan.employee_id, loan.firstname, loan.loan_amount from employee left join loan on loan.employee_id = employee.employee_id;

EMPLOYEE_ID FIRSTNAME                      LOAN_AMOUNT
----------- ------------------------------ -----------
       7505 Jean                                 25000
       7521 Cynthia                              45000





6 rows selected.

SQL> Select loan.employee_id, loan.firstname, loan.loan_amount from employee right join loan on loan.employee_id = employee.employee_id;

EMPLOYEE_ID FIRSTNAME                      LOAN_AMOUNT
----------- ------------------------------ -----------
       7505 Jean                                 25000
       7521 Cynthia                              45000
       1234 Jack                                 10000
       1235 John                                 15000

SQL> Select loan.employee_id, loan.firstname, loan.loan_amount from employee full outer join loan on loan.employee_id = employee.employee_id;

EMPLOYEE_ID FIRSTNAME                      LOAN_AMOUNT
----------- ------------------------------ -----------


       7505 Jean                                 25000


       7521 Cynthia                              45000
       1234 Jack                                 10000
       1235 John                                 15000

8 rows selected.
```

Q.2→



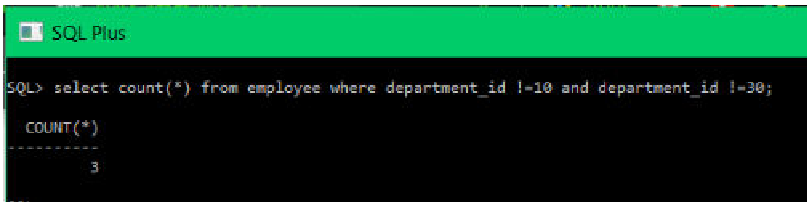
```
SQL> Select employee_id, firstname from employee union select employee_id, firstname from loan;

EMPLOYEE_ID FIRSTNAME
----------- ------------------------------
       1234 Jack
       1235 John
       7369 Jon
       7499 Kevin
       7505 Jean
       7506 Lynn
       7507 Leslie
       7521 Cynthia
       7521 cynthia

9 rows selected.

SQL> Select employee_id, firstname from employee union all select employee_id, firstname from loan;

EMPLOYEE_ID FIRSTNAME
----------- ------------------------------
       7369 Jon
       7499 Kevin
       7505 Jean
       7506 Lynn
       7507 Leslie
       7521 cynthia
       7505 Jean
       7521 Cynthia
       1234 Jack
       1235 John

10 rows selected.

SQL> Select employee_id, firstname from employee intersect select employee_id, firstname from loan;

EMPLOYEE_ID FIRSTNAME
----------- ------------------------------
       7505 Jean

SQL> Select employee_id, firstname from employee minus select employee_id, firstname from loan;

EMPLOYEE_ID FIRSTNAME
----------- ------------------------------
       7369 Jon
       7499 Kevin
       7506 Lynn
       7507 Leslie
       7521 cynthia
```

Q.3→



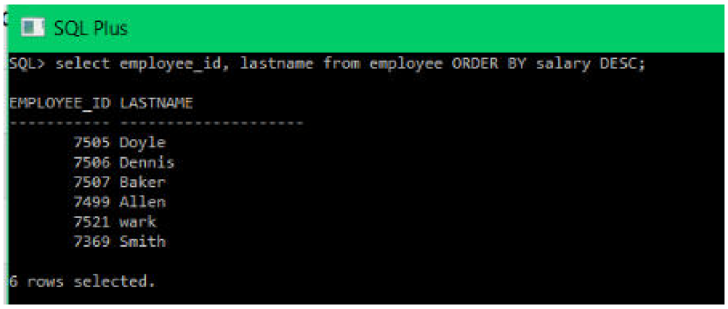
```
SQL> select count(*) from employee where department_id = 30;

  COUNT(*)
----------
         1
```

Q.4→



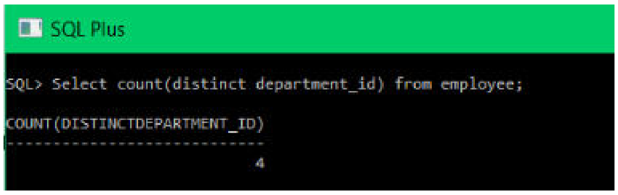
```
SQL> select count(*) from employee where department_id !=10 and department_id !=30;

  COUNT(*)
----------
         3
```

Q.5→



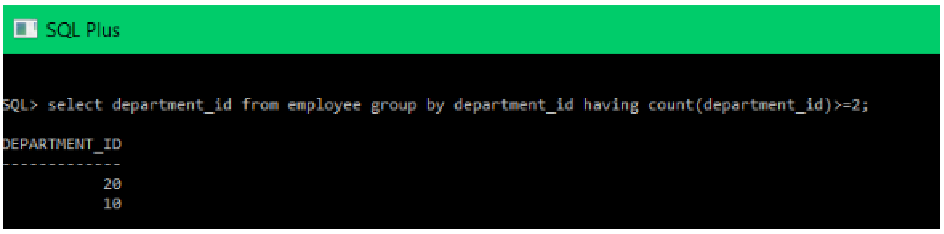
```
SQL> select employee_id, lastname from employee ORDER BY salary DESC;

EMPLOYEE_ID LASTNAME
----------- --------------------
       7505 Doyle
       7506 Dennis
       7507 Baker
       7499 Allen
       7521 wark
       7369 Smith

6 rows selected.
```

Q.6→



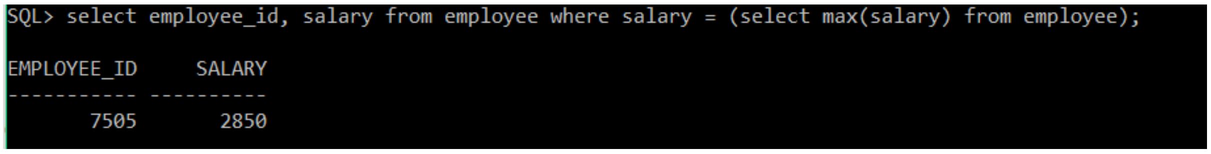
```
SQL> Select count(distinct department_id) from employee;

COUNT(DISTINCTDEPARTMENT_ID)
----------------------------
                           4
```

Q.7→



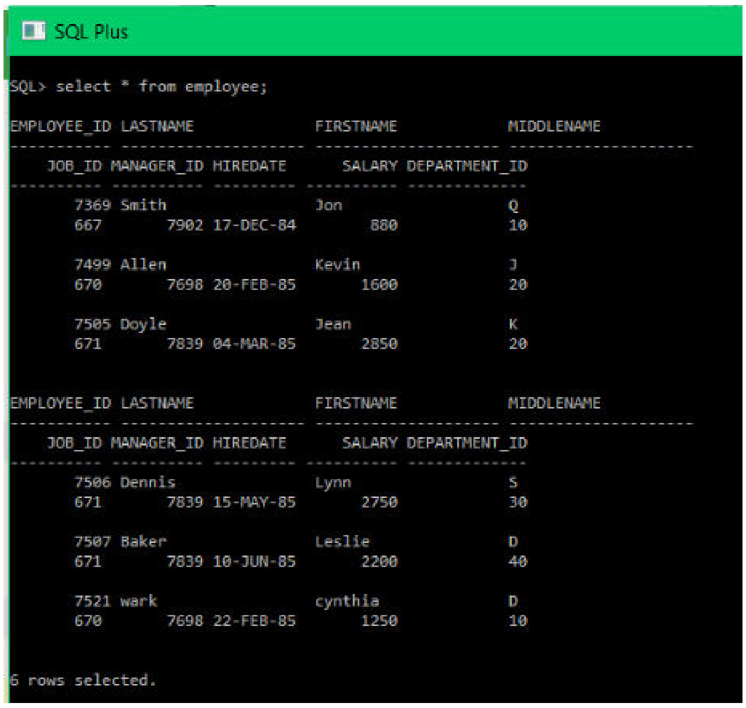
```
SQL> select department_id from employee group by department_id having count(department_id)>=2;

DEPARTMENT_ID
-------------
           20
           10
```

Q.8→



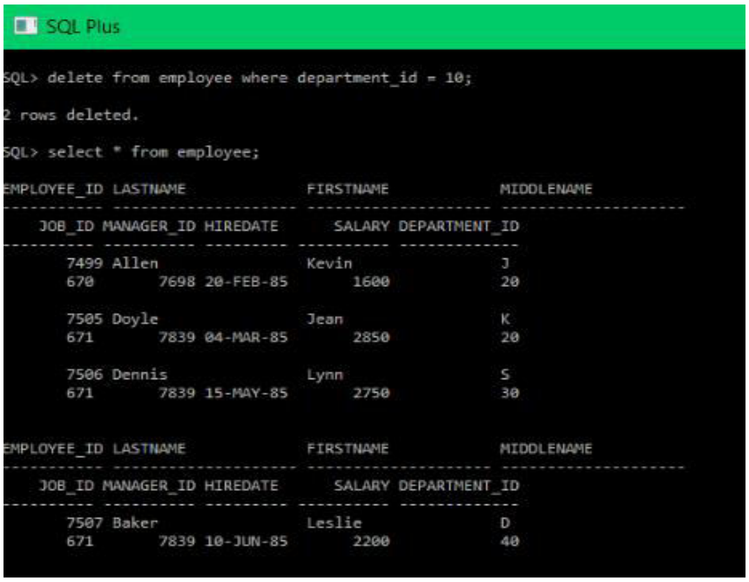
```
SQL> select employee_id, salary from employee where salary = (select max(salary) from employee);

EMPLOYEE_ID     SALARY
----------- ----------
       7505       2850
```

Q.9→



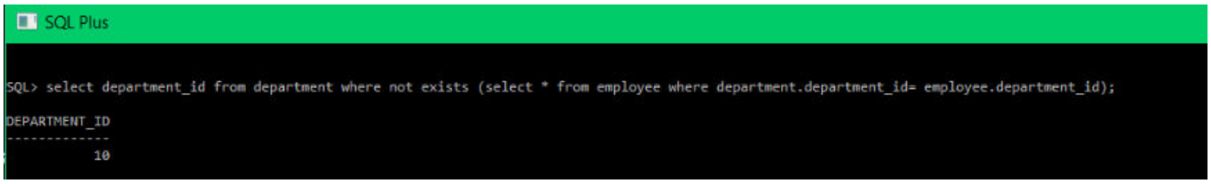
```
SQL> select * from employee;

EMPLOYEE_ID LASTNAME             FIRSTNAME            MIDDLENAME
----------- -------------------- -------------------- --------------------
    JOB_ID MANAGER_ID HIREDATE      SALARY DEPARTMENT_ID
---------- ---------- --------- ---------- -------------
       7369 Smith                Jon                  Q
       667        7902 17-DEC-84        880            10

       7499 Allen                Kevin                J
       670        7698 20-FEB-85       1600            20

       7505 Doyle                Jean                 K
       671        7839 04-MAR-85       2850            20

EMPLOYEE_ID LASTNAME             FIRSTNAME            MIDDLENAME
----------- -------------------- -------------------- --------------------
    JOB_ID MANAGER_ID HIREDATE      SALARY DEPARTMENT_ID
---------- ---------- --------- ---------- -------------
       7506 Dennis               Lynn                 S
       671        7839 15-MAY-85       2750            30

       7507 Baker                Leslie               D
       671        7839 10-JUN-85       2200            40

       7521 wark                 cynthia              D
       670        7698 22-FEB-85       1250            10


6 rows selected.
```

Q.10→



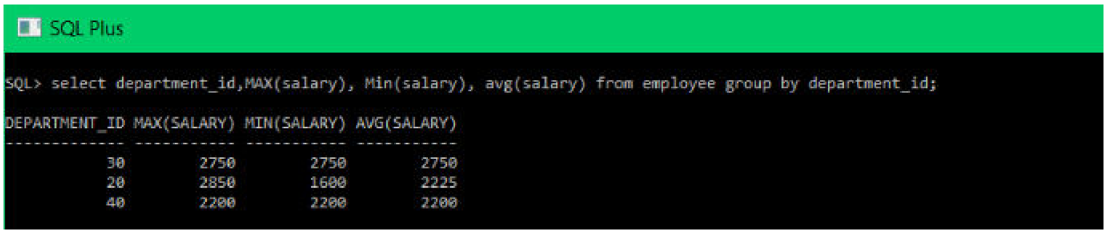
```
SQL> delete from employee where department_id = 10;

2 rows deleted.

SQL> select * from employee;

EMPLOYEE_ID LASTNAME             FIRSTNAME            MIDDLENAME
----------- -------------------- -------------------- --------------------
    JOB_ID MANAGER_ID HIREDATE      SALARY DEPARTMENT_ID
---------- ---------- --------- ---------- -------------
       7499 Allen                Kevin                J
       670        7698 20-FEB-85       1600            20

       7505 Doyle                Jean                 K
       671        7839 04-MAR-85       2850            20

       7506 Dennis               Lynn                 S
       671        7839 15-MAY-85       2750            30

EMPLOYEE_ID LASTNAME             FIRSTNAME            MIDDLENAME
----------- -------------------- -------------------- --------------------
    JOB_ID MANAGER_ID HIREDATE      SALARY DEPARTMENT_ID
---------- ---------- --------- ---------- -------------
       7507 Baker                Leslie               D
       671        7839 10-JUN-85       2200            40
```

Q.11→



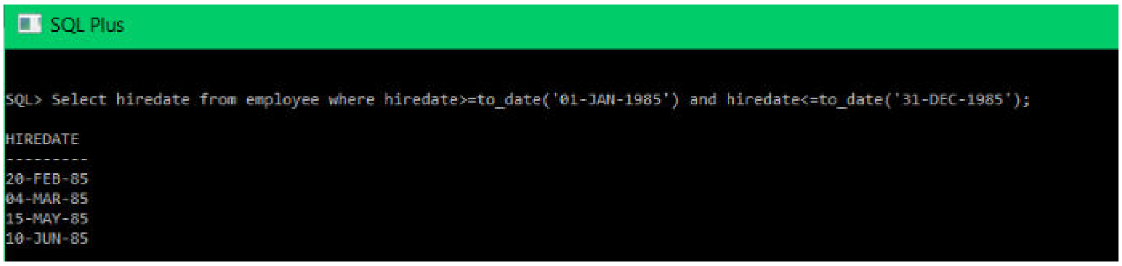
```
SQL> select department_id from department where not exists (select * from employee where department.department_id= employee.department_id);

DEPARTMENT_ID
-------------
           10
```

Q.12→



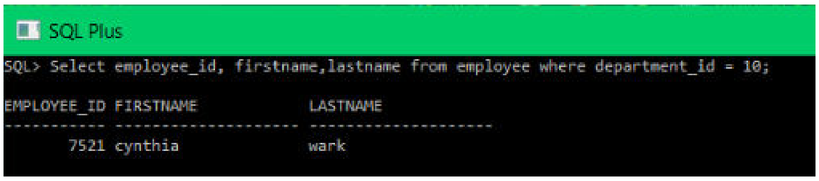
```
SQL> select department_id,MAX(salary), Min(salary), avg(salary) from employee group by department_id;

DEPARTMENT_ID MAX(SALARY) MIN(SALARY) AVG(SALARY)
------------- ----------- ----------- -----------
           30        2750        2750        2750
           20        2850        1600        2225
           40        2200        2200        2200
```

Q.13→



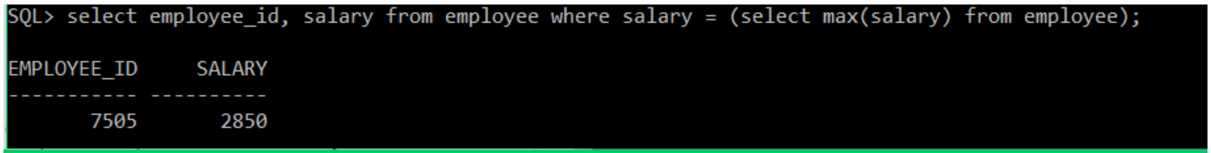
```
SQL> Select hiredate from employee where hiredate>=to_date('01-JAN-1985') and hiredate<=to_date('31-DEC-1985');

HIREDATE
---------
20-FEB-85
04-MAR-85
15-MAY-85
10-JUN-85
```

Q.14→

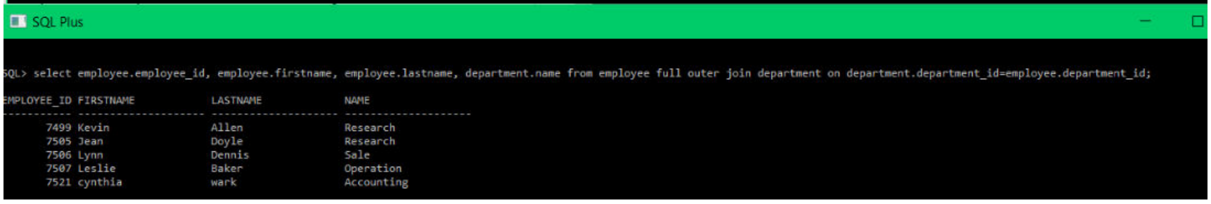


```
SQL> Select employee_id, firstname,lastname from employee where department_id = 10;

EMPLOYEE_ID FIRSTNAME            LASTNAME
----------- -------------------- --------------------
       7521 cynthia              wark
```

Q.15→



```
SQL> select employee_id, salary from employee where salary = (select max(salary) from employee);

EMPLOYEE_ID     SALARY
----------- ----------
       7505       2850
```



```
SQL> select employee.employee_id, employee.firstname, employee.lastname, department.name from employee full outer join department on department.department_id=employee.department_id;

EMPLOYEE_ID FIRSTNAME            LASTNAME             NAME
----------- -------------------- -------------------- --------------------
       7499 Kevin                Allen                Research
       7505 Jean                 Doyle                Research
       7506 Lynn                 Dennis               Sale
       7507 Leslie               Baker                Operation
       7521 cynthia              wark                 Accounting
```