EXPERIMENT: 2

Aim: Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET.

SOLUTION:

To create employee table

SQL> create table employee(Fname varchar2(20),Lname varchar2(20),Ssn number(4) primary key,B_date date,Address varchar2(30),Gender char(1),Salary number(7),Dno number(4));

Table created.

SQL> SELECT *from employee;

FNAME	LNAME	SSN	B_DATE	ADDRESS	G	SALARY	DNO
SMITH		1111	03-NOV-16	BJD	M	2000	10
ALLEN		2222	03-NOV-16	SBC	M	3000	20
MARTIN		3333	03-NOV-16	HYD	M	4000	30
JONES		4444	28-SEP-15	TNU	M	2500	10
BLAKE		5555	04-SEP-24	VZA	M	2500	10
TURNER		6666	21-OCT-99	GNT	M	6000	20

6 rows selected.

To create dependent table

SQL> create table dependent (essn number (4),dependent_name varchar2 (20),gender char (1),b_date date,relationship varchar2 (20),primary key (essn, dependent_name));

Table created.

SQL> Select *from dependent;

ESSN	DEPENDENT_NAME	G	B_DATE	RELATIONSHIP
1111	SMITH	M		
2222	POOJA	F		
3333	MARTIN	M		
3333	RAJA	M		

ALL

The ALL comparison condition is used to compare a value to a list or subquery. It must be preceded by =, ! =, >, <, <=, >= and followed by a list or subquery.

Retrieve the names of employees whose salary is greater than the salary of all the employees in department 10

SQL> Select Fname, Lname From Employee Where Salary> All (Select Salary From Employee Where Dno=10);

FNAME LNAME

---------- ----------

ALLEN

MARTIN

TURNER

Find Employees data whose employee salary should above 2000 and 3000 and 4000

SQL> SELECT fname, salary FROM employee WHERE salary > ALL (2000, 3000, 4000);

FNAME SALARY

---------- ----------

TURNER 6000

ANY

The ANY comparison condition is used to compare a value to a list or subquery. It must be preceded by =, ! =, >, <, <=, >= and followed by a list or subquery.

Retrieve the names of employees whose salary is greater than the salary of any one of the employees in department 10

SQL> Select Fname, Lname From Employee Where Salary> Any(Select Salary From Employee Where Dno=10);

FNAME LNAME

---------- ----------

ALLEN

MARTIN

JONES

BLAKE

TURNER

Find Employees data whose employee salary having more than 2000 or 3000 or 4000

SQL> SELECT fname, salary FROM employee WHERE salary > Any (2000, 3000, 4000);

FNAME	SALARY
ALLEN	3000
MARTIN	4000
JONES	2500
BLAKE	2500
TURNER	6000

IN

The IN operator allows you to specify multiple values in a WHERE clause.

Retrieve the name of each employee who has a dependent with the firstname and same gender as the employee

SQL> select e.fname, e.lname from employee e where e.ssn in (select essn from dependent where e.gender=gender and e.fname = dependent_name);

FNAME	LNAME
SMITH	
MARTIN	

To find employees data whose empno is 3000 or 60000

SQL> select * from employee where salary in(3000, 6000);

FNAME	LNAME	SSN	B_DATE	ADDRESS	G	SALARY	DNO
ALLEN		2222	03-NOV-16 S	BC	M	3000	20
TURNER		6666	21-OCT-99	GNT	M	6000	20

EXISTS and NOT EXISTS

The EXISTS condition is used in combination with a subquery and is considered to be met, if the subquery returns at least one row. It can be used in a SELECT, INSERT, UPDATE, or DELETE statement.

Retrieve the name of each employee who has a dependent with the firstname and same gender as the employee

SQL> select e.fname, e.lname from employee e where exists (select*from dependent where e.ssn=essn and e.gender=gender and e.fname = dependent_name);

FNAME LNAME

---------- ----------

SMITH

MARTIN

Retrieve the names of employees who have no dependents

SQL> select fname, lname from employee where not exists (select * from dependent where ssn=essn);

FNAME LNAME

---------- ----------

JONES

BLAKE

TURNER

UNION

UNION is used to combine the results of two or more Select statements. However it will eliminate duplicate rows from its result set. In case of union, number of columns and datatype must be same in both the tables.

To combine fnames whose employee gender is M and whose dependent dependent_name is SMITH.

SQL> (select fname from employee e, dependent d where e.ssn=d.essn and d.gender='M') union (select fname from employee e, dependent d where e.ssn=d.essn and d.dependent_name='SMITH');

FNAME

MARTIN

SMITH

To find common fnames whose employee gender is M and whose dependent dependent_name is SMITH.

Intersect

Intersect operation is used to combine two SELECT statements, but it only retuns the records which are common from both SELECT statements. In case of **Intersect** the number of columns and datatype must be same.

SQL> select fname from employee e, dependent d where e.ssn=d.essn and d.gender='M' INTERSECT (select fname from employee e, dependent d where e.ssn=d.essn and d.dependent_name='SMITH');

FNAME

SMITH