**EX NO : 04 NESTED QUERIES**

**AIM:**

To execute Nested Query using IN,NOT IN, Some, ALL, Exists and Not Exists keywords.

Also to execute scalar sub query in select and insert clause.

**TYPES:**

* Independent sub-query – execution of Inner query is independent of Outer query.
* Correlated sub-query - execution of Inner query is dependent of Outer query.
* Scalar sub query- selects only one column or expression and returns one value or one row.

**To create a relation:**

SQL> create table student(Rollno number(6),Name varchar(5),Mark number(4),DID number(4));

Table created.

SQL> select \* from student;

ROLLNO NAME MARK DID

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

101 MUTHU 57 2

102 KRISH 47 1

103 DHONI 75 1

104 VIRAT 85 3

105 SUGAN 59 1

**Independent sub-query:**

**Using in keyword**

Ex: Find the Rollno and Name of the student who got marks above 60

SQL> select Rollno,Name from student where Rollno in(select Rollno from student where mark>60);

ROLLNO NAME

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

103 DHONI

104 VIRAT

**Using not in keyword**

EX: Find the Rollno and Name of the student who got marks other than 60

SQL> select Rollno,Name from student where Rollno not in(select Rollno from student where mark=60);

ROLLNO NAME

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

101 MUTHU

102 KRISH

103 DHONI

104 VIRAT

105 SUGAN

**Using some keyword**

Ex: Find the names of students who have scored a mark greater than some other student's mark

SQL> select Name from student where mark>some(select mark from student);

NAME

-----

VIRAT

DHONI

SUGAN

MUTHU

**Using All keyword**

Ex: Find the name of student who have scored second highest mark

SQL> select Name from student where mark>all(select Mark from student where mark<(select max(Mark) from student));

NAME

-----

DHONI

**Using Exists keyword**

EX: Find the names of students who have received a DID value of 1:

SQL> select Name from student where exists (select Name from student where DID=1);

NAME

-----

KRISH

DHONI

SUGAN

**Using not Exists keyword**

EX: Find the names of students who have a mark less than or equal to 60.

SQL> select name from student s where not exists (select name from student where Mark>60 and s.Rollno=Rollno);

NAME

-----

KRISH

MUTHU

SUGAN

**Correlated sub-query:**

Ex: Find the Name of the student who got marks above 60

SQL> SELECT Name FROM student O WHERE Mark IN (SELECT Mark FROM student I WHERE I.Rollno = O.Rollno AND Mark > 60);

NAME

-----

DHONI

VIRAT

**Using Exists keyword**

Ex: Find the Rollno and Name of the student who got marks above 60

SQL> select Rollno,Name from student O where exists (select Mark from student I where O.Rollno=I.Rollno and Mark>60);

ROLLNO NAME

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

103 DHONI

104 VIRAT

**Scalar Sub query:**

SQL> create table department(did number(5),deptname varchar(20));

SQL> select \* from department;

DID DEPTNAME

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

1 CSE

2 IT

1. ECE

SQL> create table faculty(fid number(4),fname varchar(20),deptname varchar(10));

SQL> select \* from faculty;

FID FNAME DEPTNAME

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

1 MUTHU IT

2 DHONI IT

3 KRISH CSE

4 Akhil ECE

5 VIRAT ECE

1. Karthik ECE

**Scalar subquery using select clause:**

SQL> select deptname,(select count(\*) from faculty where faculty.deptname=de

partment.deptname) as no\_of\_faculty from department;

DEPTNAME NO\_OF\_FACULTY

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

CSE 1

IT 2

ECE 3

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

SQL> create table emp(emp\_name varchar(10),dept\_name varchar(10),salary number(6),bonus number(5));

Table created.

SQL> select \* from emp;

EMP\_NAME DEPT\_NAME SALARY BONUS

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

MUTHU IT 50000 5000

KRISH CSE 40000 4500

VIRAT ECE 30000 4000

SUGAN AIDS 30000 3500

Praveen CHEM 28000 3000

SQL> create table emp\_salary\_summary(sum\_salaries number(6),max\_salary number(6),min\_salary number(6),avg\_salary number(6));

Table created.

**To see the records in emp\_salary\_summary:**

SQL>select \* from emp\_salary\_summary;

no rows selected.

**Scalar subquery using insert clause:**

SQL> insert into emp\_salary\_summary(sum\_salaries,max\_salary,min\_salary,avg\_salary) values( (select sum(salary) from emp),( select max(salary) from emp),(select min(salary) from emp),(select avg(salary) from emp));

1 row created.

SQL> select \* from emp\_salary\_summary;

SUM\_SALARIES MAX\_SALARY MIN\_SALARY AVG\_SALARY

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

178000 50000 28000       35600

|  |  |  |
| --- | --- | --- |
| CONTENTS | MARKS ALLOTED | MARKS OBTAINED |
| Aim, Algorithm, SQL, PL/SQL | 30 |  |
| Execution and Result | 20 |  |
| Viva | 10 |  |
| Total | 60 |  |

**Result:**

Thus,the execution of Nested query using in,not in,some,all,exists and not exists key words,also the execution of scalar sub query using select and insert clause are exected successfully.