Ex. No: 2	**DATABASE QUERYING – SIMPLE QUERIES, NESTED QUERIES, SUB QUERIES AND JOINS**
Date:	

AIM:

To Study and Practice Constraints, Nested Queries and joining records based on conditions in RDBMS.

A) CONSTRAINTS :

1. NOT NULL:

Used to set the not null constraint to the specified column name which will not allow null values.

Syntax:

Create table tablename(fieldname1 datatype(constraint)not null,fieldname2 datatype,

…   fieldnamen datatype);

Example:

```
SQL> create table notnull (eno varchar2(10) not null,ename varchar2(10),esalary

number(20)); Table created.


SQL> insert into notnull values(‘&eno’,’&ename’,’&esalary’);

Enter value for eno: 1

Enter value for ename: arul

Enter value for esalary: 20000

old 1: insert into notnull values(‘&eno’,’&ename’,’&esalary’)

new 1: insert into notnull values(‘1’,’arul’,’20000’)

1 row created.


SQL> /

Enter value for eno:

Enter value for ename: raj
Enter value for esalary: 30000

old 1: insert into notnull values('&eno','&ename','&esalary')

new 1: insert into notnull values('','raj','30000')

insert into notnull values('','raj','30000')
                      *
ERROR at line 1:

ORA-01400: cannot insert NULL into ("SCOTT"."NOTNULL"."ENO")
```

2. CHECK:

Check constraint specify conditions that each tuple must satisfy.

Syntax:

Create table tablename(Fieldname1 datatype(constraint),Fieldname2 datatype,

…   Fieldname3 datatype);

Example:

```
SQL> create table con( empid varchar2(20) not null,empname
varchar2(20),empsalary number(10) check(empsalary>10000));

Table created.

SQL> insert into con

values('&empid','&empname','&empsalary'); Enter value for

empid: 1

Enter value for empname: kumar

Enter value for empsalary: 20000

old 1: insert into con values('&empid','&empname','&empsalary')

new 1: insert into con values('1','kumar','20000')

1 row
created.

SQL> /

Enter value for empid: 2

Enter value for empname:

raja
```

```
Enter value for empsalary: 9000
old 1: insert into con values('&empid','&empname','&empsalary')
new 1: insert into con values('2','raja','9000')
insert into con values('2','raja','9000')
*
ERROR at line 1:
ORA-02290: check constraint (SCOTT.SYS_C0010283) violated
```

3. UNIQUE:

Used to set unique constraint to the specified column name which will not allow redundant values

Syntax:

```
Create table tablename(fieldname1 datatype(constraint)unique,fieldname2 datatype,
…                Fieldname3 datatype);
```

Example:

```
SQL> create table conn(eno varchar2(10) unique,ename varchar2(20));

Table created.

SQL> insert into conn values('&eno','&ename');
Enter value for eno: 1
Enter value for ename: hello
old 1: insert into conn values('&eno','&ename')
new 1: insert into conn values('1','hello')

1 row created.

SQL> /
Enter value for eno: 1
Enter value for ename: hi

old 1: insert into conn values('&eno','&ename')

new 1: insert into conn values('1','hi')

insert into conn values('1','hi')

*

ERROR at line 1:

ORA-00001: unique constraint (SCOTT.SYS_C0010285) violated
```

4. PRIMARY KEY:

Primary key is a constraint for both unique and not null.

Syntax:

Create table tablename(Fieldname1 datatype(constraint)unique,fieldname2 datatype,
…   Fieldname3 datatype);

Example:

```
SQL> create table con(empid varchar2(10),empname varchar2(20)

primary key); Table created.


SQL> insert into con

values('&empid','&empname'); Enter value for

empid: 1

Enter value for empname: kumar

old 1: insert into con values('&empid','&empname')

new 1: insert into con values('1','kumar')

1 row created.
```

```
SQL> /
Enter value for empid: 2
Enter value for empname: kumar
old 1: insert into con values('&empid','&empname')
new 1: insert into con values('2','kumar')
insert into con values('2','kumar')
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.SYS_C0010286) violated
```

5. ADDING CONSTRIANT:

Used to set any constraint to the specified column at the last by specifying the constraint type and field name.

Syntax:

Create table tablename(Fieldname1 datatype(constraint)unique,fieldname2 datatype, constraint constraintname constrainttype(fieldname));

Example:

```
SQL> create table con(empid varchar2(10),empname varchar2(10),constraint c1
primary key(empid));
Table created.

SQL> insert into con
values('&empid','&empname'); Enter value for
empid: 1
Enter value for empname: anand
old 1: insert into con values('&empid','&empname')
new 1: insert into con values('1','anand')
1 row created.
```

```
SQL> /
Enter value for empid: 1
Enter value for empname: vijay
old 1: insert into con values('&empid','&empname')
new 1: insert into con values('1','vijay')
insert into con values('1','vijay')
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.C1) violated
```

6. ADD CONSTRAINT(ALTER)

Used to set the constraint for the table already created by using alter command.

Syntax:

Alter table tablename add constraint constraintname (fieldname)datatype,primary key.

Example:

```
SQL> create table con(empid varchar2(10),empname
varchar2(10)); Table created.
SQL> alter table con add constraint c1 primary
key(empid); Table altered.

SQL> desc con;
 Name                          Null? Type
 ----------------------------------------------------
 ---
                                                  N
 EMPID                                            O
                                                  T
                                                  N
                                                  U
                                                  L
                                                  L
                                                  V
                                                  A
                                                  R
                                                  C
                                                  H
                                                  A
```

R		1
2		0
(		)
EMPNAME	VARCHAR2(10)	

7. DROP CONSTRAINT:

Used to drop the constraint.

Syntax:

Alter table tablename drop constraint constraintname.

Example:

```
SQL> alter table con drop constraint c1;

Table altered.


SQL> desc con;

 Name                          Null? Type

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

 EMPID                               VARCHAR2(10)

 EMPNAME                                VARCHAR2(10)
```

8. REFERENTIAL INTEGRITY:

Used to refer the primary key of the parent table from the childtable.

Syntax:

a)Create table tablename(Fieldname1 datatype primary key,fieldname2 datatype,

…   Fieldname3 datatype);

b) Create table tablename(Fieldname1 datatype references,Parent tablename(fieldname)

…   Fieldname n datatype);

Example:

```
SQL> create table parent(eno varchar2(10),ename varchar2(10)

primary key); Table created.

SQL> insert into parent values(‘&eno’,’&ename’);
Enter value for eno: 1
Enter value for ename: ajay

old 1: insert into parent values(‘&eno’,’&ename’)

new 1: insert into parent values(‘1’,’ajay’)
```

```
1 row created.

SQL> /
Enter value for eno: 2
Enter value for ename:
bala
old 1: insert into parent values(‘&eno’,’&ename’)
new 1: insert into parent values(‘2’,’bala’)
1 row created.

SQL> create table child (eno varchar2(10),ename varchar2(10) references
parent(ename)); Table created.

SQL> insert into child values(‘&eno’,’&ename’);
Enter value for eno: 1
Enter value for ename: ajay
old 1: insert into child values(‘&eno’,’&ename’)
new 1: insert into child values(‘1’,’ajay’)

1 row created.

SQL> /
Enter value for eno: 2
Enter value for ename: balaji
old 1: insert into child values(‘&eno’,’&ename


new 1: insert into child values(‘2’,’balaji’)
insert into child values(‘2’,’balaji’)
ERROR at line 1:
ORA-02291: integrity constraint (SCOTT.SYS_C0010290) violated - parent key not
Found
```

9. ON DELETE CASCADE:

The changes done in parent table is reflected in the child table when references are made .

Syntax:

```
Create table tablename(Fieldname1 datatype references,Parent

tablename(fieldname), On delete cascade);
```

Example:

```
SQL> create table parent(eno varchar2(10),ename varchar2(10)

primary key); Table created.

SQL> insert into parent values(‘&eno’,’&ename’);

Enter value for eno: 1

Enter value for ename: a

old 1: insert into parent values(‘&eno’,’&ename’)

new 1: insert into parent values(‘1’,’a’)

1 row created.


SQL> create table child(eno varchar2(10),ename varchar2(10) references
parent(ename) on delete cascade);

Table created.


SQL> insert into child values(‘&eno’,’&ename’);

Enter value for eno: 2

Enter value for ename: a

old 1: insert into child values(‘&eno’,’&ename’)

new 1: insert into child values(‘2’,’a’)

1 row created.
```

```
SQL> select * from parent;

ENO      ENAME

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

1        a
```

```
SQL> select * from

child; ENO  ENAME

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

2        a

SQL> delete from parent where eno=1;

1 row deleted.

SQL> select * from parent;

no rows selected

SQL> select * from child;

no rows selected
```

B) SUBQUERY:

<u>SINGLE ROW SUB-QUERY:</u>

Syntax:

Select <fieldname> from <tablename1> where <fieldname>= (select<fieldname>from<fieldnam e2>where (condition);

1. UNION:

Syntax:

Select <fieldlist> from <tablename1> where (condition) union

select<fieldlist> from<tablename2> where (condition);

2. INTERSECT:

Syntax:

Select <fieldlist> from <tablename1> where (condition) intersect

select<fieldlist> from<tablename2> where (condition);

3. IN:

Syntax:

Select <fieldlist> from <tablename1> where (condition) in

select<fieldlist> from<tablename2> where (condition);

4. BETWEEN:

Syntax:

Select <fieldlist> from <tablename1> where (condition) between

select<fieldlist> from<tablename2> where (condition);

5. LIKE:

Syntax:

Select <fieldlist> from <tablename> where <fieldname> like <expression>;

6. NOT LIKE:

Syntax:

Select <fieldlist> from <tablename> where <fieldname> not like <expression>;

7. ALL:

Syntax:

Select <fieldlist> from <tablename1>where

<fieldname> all Select <fieldlist> from <tablename2>

where (condition);

8. ANY:

Syntax:

Select <fieldlist> from <tablename1> where

(condition) any Select <fieldlist> from <tablename2>

where(condition);

C) JOINS:

1. EQUI JOIN:

It retrieves rows from two tables having a common column using '=' operators.

Syntax:

Select <tablename1.fieldlist,tablename2.fieldlist> from
<tablename1><tablename2> where
<tablename1.keyfield>=<tablename2.keyfield>;

2. INNER JOIN :

Inner join returns the matching rows from the tables that are being joined.

Syntax:

Select tablename1.fieldlist,tablename2.fieldlist from tablename1 inner join
tablename2 on tablename1.keyfield = tablename2.keyfield;

3. OUTER JOIN :

Outer join returns the both matching and non-matching rows for the tables that are being joined. An outer join is an extended form of the inner join. The rows in one table having no matching rows in the other table will also appear in the result table with nulls.

LEFT OUTER JOIN :

Syntax:

Select tablename1.fieldlist,tablename2.fieldlist from tablename1 left outer join
tablename2 on tablename1.keyfield = tablename2.keyfield;

RIGHT OUTER JOIN :

Syntax:

Select tablename1.fieldlist,tablename2.fieldlist from tablename1 right outer join
tablename2 on tablename1.keyfield = tablename2.keyfield;

FULL OUTER JOIN :

Syntax:

Select tablename1.fieldlist,tablename2.fieldlist from tablename1 full outer join
tablename2 on tablename1.keyfield = tablename2.keyfield;

OUTPUT / EXAMPLE (NESTED QUERIES):

```
SQL> create table employee(empno varchar2(10),empname
varchar2(10),empsalary number(10));

Table created.

SQL> drop table employee1;

Table dropped.
```

```
SQL> create table employee1(empno varchar2(10),empname
varchar2(10),empsalary number(10));

Table created.

SQL> insert into employee values(‘&empno’,’&empname’,’&empsalry’);

Enter value for empno: 1

Enter value for empname: arun

Enter value for empsalry: 10000

old 1: insert into employee values(‘&empno’,’&empname’,’&empsalry’)

new 1: insert into employee values(‘1’,’arun’,’10000’)

1 row
created.

SQL> /

Enter value for empno: 2

Enter value for empname:

bala

Enter value for empsalry: 20000

old 1: insert into employee values(‘&empno’,’&empname’,’&empsalry’)

new 1: insert into employee values(‘2’,’bala’,’20000’)

1 row created.

SQL> insert into employee1 values(‘&empno’,’&empname’,’&empsalry’);

Enter value for empno: 1

Enter value for empname: arun

Enter value for empsalry: 10000

old 1: insert into employee1 values(‘&empno’,’&empname’,’&empsalry’)

new 1: insert into employee1 values(‘1’,’arun’,’10000’)

1 row created.
```

```
SQL> /

Enter value for empno: 3

Enter value for empname: chitra

Enter value for empsalry: 40000

old 1: insert into employee1 values('&empno','&empname','&empsalry')

new 1: insert into employee1 values('3','chitra','40000')

1 row created.


SQL> select * from employee; EMPNO

           EMPNAME EMPSALARY

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

1       arun        10000

2       bala       20000


SQL> select * from employee1;


EMPNO EMPNAME
EMPSALARY

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

1       arun        10000

3       chitra      40000
```

Output for Subquery:

```
SQL> select * from employee where empsalary=(select min(empsalary)from

employee1); EMPNO      EMPNAME EMPSALARY

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

1       arun        10000
```

UNION:

```
SQL> select empname,empno from employee where(empsalary>10000)
union select empname,empno from employee1 where(empsalary>10000);

EMPNAME EMPNO

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

bala     2

chitra   3
```

INTERSECT:

```
SQL> select empname,empno from employee where(empsalary>9000)
intersect select empname,empno from employee1 where(empsalary>9000);

EMPNAME EMPNO

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

arun     1
```

IN:

```
SQL> select empname from employee where empsalary in(select empsalary from
employee1);


EMPNAME

-------

arun

SQL> select empno,empname from employee where empno in(select
empsalary from employee1);

no rows selected


SQL> select empno,empname from employee where empno in(select empno from
employee1);

EMPNO    EMPNAME

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

    1        Arun
```

BETWEEN:

```
SQL> select empno,empname from employee where empsalary between 10000 and 30000;

EMPNO    EMPNAME

--------------
     1           arun

     2bala
```

LIKE:

```
SQL> select empname,empno from employee where empname like ‘b%’;


EMPNAME EMPNO

--------------
bala      2
```

NOT LIKE:

```
SQL> select empname,empno from employee where empname not like ‘b%’;

EMPNAME EMPNO

--------------
arun      1
```

ALL:

```
SQL> select empname,empsalary from employee1 where empsalary > all (select
empsalary from employee where empsalary>10000);

EMPNAME EMPSALARY

--------------
chitra        40000
```

ANY:

```
SQL> select empname,empsalary from employee1 where empsalary>any
(select min(empsalary) from employee);

EMPNAME EMPSALARY
------- -------
chitra     40000
```

OUTPUT / EXAMPLE(JOINS):

```
SQL> create table employee (empid varchar2(10),empname varchar2(10),deptid
varchar2(10) primary key);
Table created.

SQL> insert into employee values('&empid','&empname','&deptid');
Enter value for empid: 10cse01
Enter value for empname:
anand Enter value for deptid:
cse
old 1: insert into employee values('&empid','&empname','&deptid')
new 1: insert into employee values('10cse01','anand','cse')
1 row created.

SQL> /
Enter value for empid: 10ece02
Enter value for empname: bala
Enter value for deptid: ece
old 1: insert into employee values('&empid','&empname','&deptid')
new 1: insert into employee values('10ece02','bala','ece')
1 row created.
```

```
SQL> /

Enter value for empid: 10mech03

Enter value for empname: karthi

Enter value for deptid: MECH

old 1: insert into employee values(‘&empid’,’&empname’,’&deptid’)

new 1: insert into employee values(‘10mech03’,’karthi’,’MECH’)

1 row created.


SQL> create table department(deptid varchar2(10) primary key,deptname

varchar2(20)); Table created.


SQL> insert into department

values(‘&deptid’,’&deptname’); Enter value for deptid:

cse

Enter value for deptname: computerscience

old 1: insert into department values(‘&deptid’,’&deptname’)

new 1: insert into department values(‘cse’,’computerscience’)

1 row created.


SQL> /

Enter value for deptid: ece

Enter value for deptname: electronics

old 1: insert into department values(‘&deptid’,’&deptname’)

new 1: insert into department values(‘ece’,’electronics’)

1 row
created.

SQL> /

Enter value for deptid: mech

Enter value for deptname: mechanical
```

```
old 1: insert into department values(‘&deptid’,’&deptname’)
new 1: insert into department values(‘mech’,’mechanical’)
1 row created.
```

```
SQL> select * from employee;

EMPID    EMPNAME DEPTID

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

10cse01 anand      cse

10ece02 bala       ece

10mech03 karthi MECH


SQL> select * from department;

DEPTID DEPTNAME

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

ece       electronics

cse

computerscience mech

            mechanical
```

EQUI JOIN

```
SQL> select
employee.empid,employee.empname,department.deptid,department.deptname from
employee,department where employee.deptid = department.deptid;

EMPID EMPNAME DEPTID DEPTNAME ---

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

10ece02 bala       ece       electronics

10cse01 anand      cse

computerscience
```

INNER JOIN:

```
SQL> select employee.empid,department.deptname from employee inner join
department on employee.deptid = department.deptid;

EMPID    DEPTNAME

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

10ece02 electronics

10cse01 computerscience

SQL> select * from employee inner join department on

employee.deptid=department.deptid; EMPID EMPNAME DEPTID DEPTID

DEPTNAME ___________________________________________
```

```
10ece02 bala     ece     ece     electronics
10cse01 anand     cse     cse
computerscience
```

LEFT OUTER JOIN:

```
SQL> select employee.empname,department.deptname from
employee left outer join department on
employee.deptid=department.deptid;


EMPNAME DEPTNAME

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

anand

computerscience

bala

electronics karthi
```

RIGHT OUTER JOIN:

```
SQL> select employee.empname,department.deptname from
employee right outer join

department on employee.deptid=department.deptid;

EMPNAME DEPTNAME

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

bala      electronics

anand

          comput

          erscienc

          e

          mechani

          cal
```

FULL OUTER JOIN:

```
SQL> select employee.empname,department.deptname from
employee full outer join department on
employee.deptid=department.deptid;


EMPNAME DEPTNAME
```

```
---------------------
anand

computerscience

bala

electronics karthi


mechanical
```