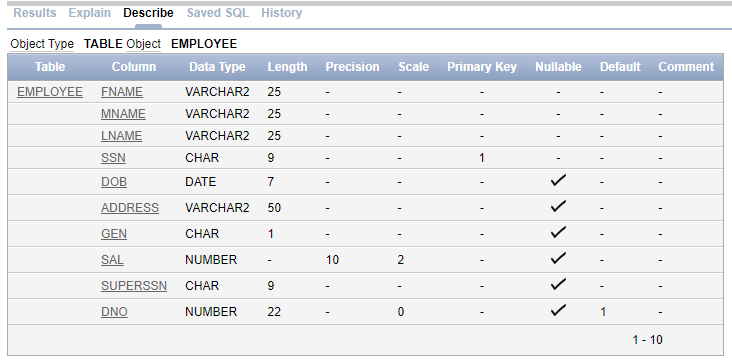
**ASSIGNMENT-1**

**Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) using CREATE, ALTER, DROP, INSERT statements.**

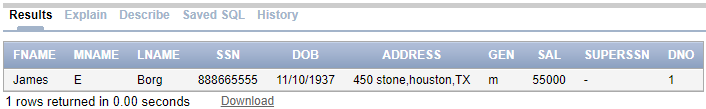
1. **EMPLOYEE: 1)Fname,Lname,SSN,Dno-NOT NULL 2)Dno-Default value 1 and dno between(1-10) 3)SSN-Primary key 4)Super\_ssn ,Dno –Foreign key on delete set to null**

create table employee(fname varchar(25) not null,mname varchar(25) not null,lname varchar(25) not null,ssn char(9) not null,dob date,address varchar(50),gen char(1) check (gen in('m','f','o')),sal decimal(10,2),superssn char(9),dno int default 1,constraint ssn\_pk primary key(ssn));

alter table employee add constraint dno\_ch check(dno>=1 and dno<=10);

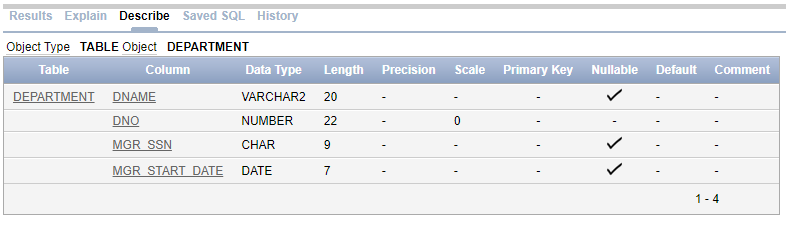


insert into employee (fname,mname,lname,ssn,dob,address,gen,sal) values('James','E','Borg',888665555,'11-10-1937','450 stone,houston,TX','m',55000);



1. **DEPARTMENT:1)Dname,Dnum,Mgr\_ssn-NOT NULL,set default 2)Dnumber- Primary key 3)Dname-Unique 4) Mgr\_SSN-Foreign Key**

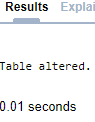
create table department(dname varchar(20),dno int not null,mgr\_ssn char(9),mgr\_start\_date date,constraint mgr\_ssn\_fk foreign key(mgr\_ssn)references employee(ssn)on delete set null)



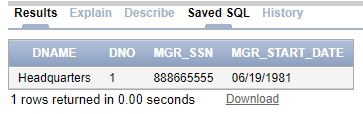
alter table department add constraint dno\_ch\_dept check(dno>=1 and dno<=10)

alter table department add constraint dno\_pk primary key(dno)

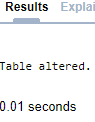
alter table department add constraint dname\_uni unique(dname)



insert into department values('Headquarters',1,888665555,'6-19-1981')

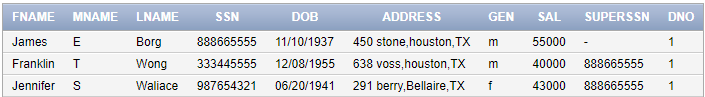


alter table employee add constraint fk\_superssn foreign key(superssn) references employee(ssn)on delete set null



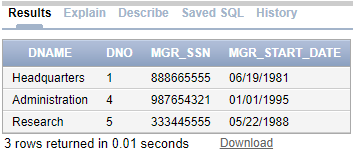
insert into employee (fname,mname,lname,ssn,dob,address,gen,sal,superssn) values('Franklin','T','Wong',333445555,'12-8-1955','638 voss,houston,TX','m',40000,888665555);

insert into employee (fname,mname,lname,ssn,dob,address,gen,sal,superssn) values('Jennifer','S','Waliace',987654321,'6-20-1941','291 berry,Bellaire,TX','f',43000,888665555);

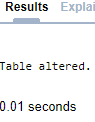


insert into department values('Administration',4,987654321,'1-1-1995')

insert into department values('Research',5,333445555,'5-22-1988')

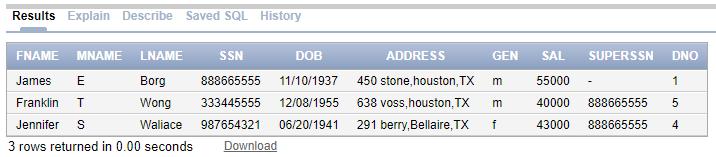


alter table employee add constraint dno\_fk foreign key(dno)references department(dno) on delete cascade;



update employee set dno=5 where ssn=333445555

update employee set dno=4 where ssn=987654321



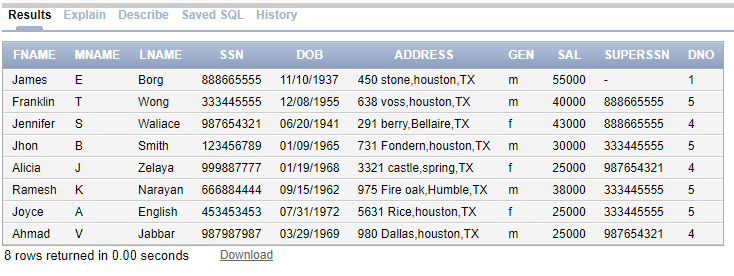
insert into employee values('Jhon','B','Smith',123456789,'1-9-1965','731 Fondern,houston,TX','m',30000,333445555,5)

insert into employee values('Alicia','J','Zelaya',999887777,'1-19-1968','3321 castle,spring,TX','f',25000,987654321,4)

insert into employee values('Ramesh','K','Narayan',666884444,'9-15-1962','975 Fire oak,Humble,TX','m',38000,333445555,5)

insert into employee values('Joyce','A','English',453453453,'7-31-1972','5631 Rice,houston,TX','f',25000,333445555,5)

insert into employee values('Ahmad','V','Jabbar',987987987,'3-29-1969','980 Dallas,houston,TX','m',25000,987654321,4)

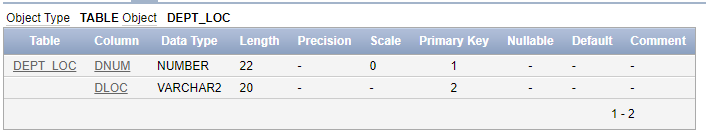


1. **DEPT\_LOCATIONS:1)Dnumber,Dloc-Not null 2)(Dnum,Dloc)-primary key 4)Dnumber-foreign key on delete cascade**

create table dept\_loc (dnum integer,dloc varchar(20) not null,primary key(dnum,dloc));

alter table dept\_loc add constraint fk\_dnum foreign key (dnum) references department(dno) on delete cascade

desc dept\_loc



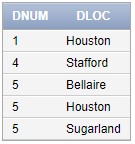
insert into dept\_loc values(1,'Houston')

insert into dept\_loc values(4,'Stafford')

insert into dept\_loc values(5,'Bellaire')

insert into dept\_loc values(5,'Sugarland')

insert into dept\_loc values(5,'Houston')



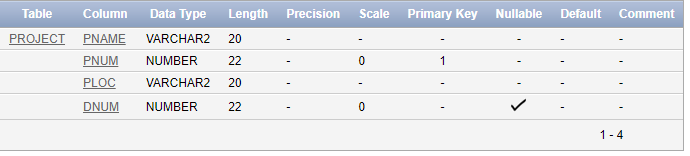
1. **PROJECT:1)Pname,Pnumber,Dnum-NOT NULL 2)Pnumber-Primary key 3)Pname-Unique 3)Dnum-Foreign Key**

create table project(pname varchar(20) not null,pnum integer,ploc varchar(20) not null,dnum integer,primary key(pnum))

alter table project add constraint fk\_pdnum foreign key (dnum) references department(dno) on delete cascade

alter table project add constraint uni\_dno unique(pname)

desc project



insert into project values('ProductX',1,Bellaire,5)

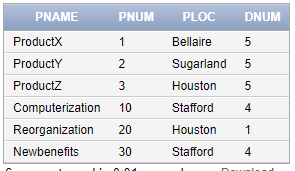
insert into project values('ProductY',2,Sugarland,5)

insert into project values('ProductZ',3,'Houston',5)

insert into project values('Computerization',10,'Stafford',4)

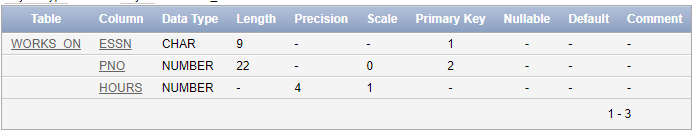
insert into project values('Reorganization',20,'Houston',1)

insert into project values('Newbenefits',30,'Stafford',4)



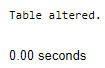
1. **WORKS\_ON:1)Essn,Pno,Hours:NOT NULL 2)(Essn,Pno)-primary key 3)Essn-Foreign key,Pno-Foreign key**

create table works\_on(essn char(9),pno integer,hours decimal(4,1)not null,primary key(essn,pno))

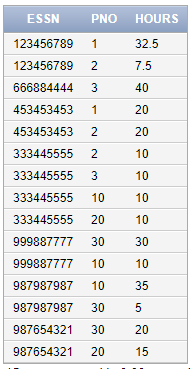


alter table works\_on add constraint fk\_pno foreign key (pno)references project(pnum) on delete cascade

alter table works\_on add constraint fk\_essn foreign key (essn) references employee(ssn) on delete cascade



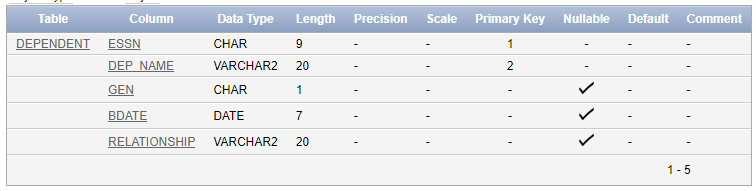
insert into works\_on values(:essn,:pno,:hrs)



1. **DEPENDENT:1)Essn, Depenedent\_Name-NOT NULL 2)( Essn, Depenedent\_Name)-primary key 3)Essn-foreign key**

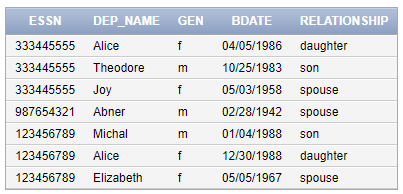
create table dependent(essn char(9),dep\_name varchar(20),gen char(1),bdate date,relationship varchar(20),check (gen in('m','f')),primary key(essn,dep\_name))

desc dependent



alter table dependent add constraint fk\_dep\_essn foreign key(essn) references employee(ssn) on delete cascade

insert into dependent values(:essn,:name,:gen,:bday,:relation)



**ASSIGNEMENT-2**

**Implementing the queries for Insertion, Updation, Deletion operations. Use ROLL BACK, COMMIT & SAVE POINTS Concepts with INSERT, UPDATE, DELETE statements.**

1. **Write a query to insert new department Prodcuction to existing DEPARTMENT table and commit and rollback**

SQL> connect

Enter user-name: 2BA22CS025

Enter password:

Connected.

SQL> select \* from department;

DNAME DNO MGR\_SSN MGR\_START

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

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

SQL> insert into department values('production',3,123456789,'12-oct-2010');

1 row created.

SQL> update employee set dno=3 where ssn=123456789;

1 row updated.

SQL> commit

2

SQL> commit;

Commit complete.

SQL> rollback;

Rollback complete.

SQL> select \* from department;

DNAME DNO MGR\_SSN MGR\_START

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

production 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

1. **Delete department where dnum=5 and undo the changes.**

delete department no 5;

SQL> delete from department where dno=5;

1 row deleted.

SQL> select \* from department;

DNAME DNO MGR\_SSN MGR\_START

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

production 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

SQL> select \* from employee;

FNAME MNAME LNAME

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

SSN DOB ADDRESS G

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

SAL SUPERSSN DNO

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

James E Borg

888665555 10-NOV-37 450 stone,houston,TX m

55000 1

Jennifer S Waliace

987654321 20-JUN-41 291 berry,Bellaire,TX f

43000 888665555 4

Jhon B Smith

123456789 09-JAN-65 731 Fondern,houston,TX m

30000 3

Alicia J Zelaya

999887777 19-JAN-68 3321 castle,spring,TX f

25000 987654321 4

Ahmad V Jabbar

987987987 29-MAR-69 980 Dallas,houston,TX m

25000 987654321 4

SQL> rollback;

Rollback complete.

SQL> select \* from department;

DNAME DNO MGR\_SSN MGR\_START

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

production 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

SQL> select \* from employee;

FNAME MNAME LNAME

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

SSN DOB ADDRESS G

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

SAL SUPERSSN DNO

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

James E Borg

888665555 10-NOV-37 450 stone,houston,TX m

55000 1

Franklin T Wong

333445555 08-DEC-55 638 voss,houston,TX m

40000 888665555 5

Jennifer S Waliace

987654321 20-JUN-41 291 berry,Bellaire,TX f

43000 888665555 4

Jhon B Smith

123456789 09-JAN-65 731 Fondern,houston,TX m

30000 333445555 3

Alicia J Zelaya

999887777 19-JAN-68 3321 castle,spring,TX f

25000 987654321 4

Ramesh K Narayan

666884444 15-SEP-62 975 Fire oak,Humble,TX m

38000 333445555 5

Joyce A English

453453453 31-JUL-72 5631 Rice,houston,TX f

25000 333445555 5

Ahmad V Jabbar

987987987 29-MAR-69 980 Dallas,houston,TX m

25000 987654321 4

Prajwal R Shettar

125125125 10-JUL-55 123 abc,xyz m

50000 888665555 5

1. rows selected.
2. **Create two savepoints for deleting department 5 and 3 and also undo the changes from ssavepoint which is first create**

SQL> savepoint s1;

Savepoint created.

SQL> delete from newdepartment where dno=5;

1 row deleted.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Production 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

SQL> select \* from newemployee;

FNAME MNAME LNAME

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

SSN DOB ADDRESS G

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

SAL SUPERSSN DNO

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

James E Borg

888665555 10-NOV-37 450 stone,houston,TX m

55000 1

Jennifer S Waliace

987654321 20-JUN-41 291 berry,Bellaire,TX f

43000 888665555 4

Jhon B Smith

123456789 09-JAN-65 731 Fondern,houston,TX m

30000 333445555 3

Alicia J Zelaya

999887777 19-JAN-68 3321 castle,spring,TX f

25000 987654321 4

Ahmad V Jabbar

987987987 29-MAR-69 980 Dallas,houston,TX m

25000 987654321 4

SQL> savepoint sp2;

Savepoint created.

SQL> delete from newdepartment where dno=3;

1 row deleted.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

SQL> select \* from newemployee;

FNAME MNAME LNAME

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

SSN DOB ADDRESS G

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

SAL SUPERSSN DNO

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

James E Borg

888665555 10-NOV-37 450 stone,houston,TX m

55000 1

Jennifer S Waliace

987654321 20-JUN-41 291 berry,Bellaire,TX f

43000 888665555 4

Alicia J Zelaya

999887777 19-JAN-68 3321 castle,spring,TX f

25000 987654321 4

Ahmad V Jabbar

987987987 29-MAR-69 980 Dallas,houston,TX m

25000 987654321 4

SQL> rollback to s1;

Rollback complete.

SQL> select \* from newemployee;

FNAME MNAME LNAME

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

SSN DOB ADDRESS G

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

SAL SUPERSSN DNO

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

James E Borg

888665555 10-NOV-37 450 stone,houston,TX m

55000 1

Franklin T Wong

333445555 08-DEC-55 638 voss,houston,TX m

40000 888665555 5

Jennifer S Waliace

987654321 20-JUN-41 291 berry,Bellaire,TX f

43000 888665555 4

Jhon B Smith

123456789 09-JAN-65 731 Fondern,houston,TX m

30000 333445555 3

Alicia J Zelaya

999887777 19-JAN-68 3321 castle,spring,TX f

25000 987654321 4

Ramesh K Narayan

666884444 15-SEP-62 975 Fire oak,Humble,TX m

38000 333445555 5

Joyce A English

453453453 31-JUL-72 5631 Rice,houston,TX f

25000 333445555 5

Ahmad V Jabbar

987987987 29-MAR-69 980 Dallas,houston,TX m

25000 987654321 4

8 rows selected.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Production 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

1. **Create three save points and undo after second savepoint**
   * 1. **Update the Dname as Finance for Dnum=3**
     2. **Update the Dname as Marketing for Dnum=1**
     3. **Update the Dname as HR for Dnum=5**

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Production 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

SQL> savepoint a1;

Savepoint created.

SQL> update newdepartment set dname='Finance' where dno=3;

1 row updated.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Finance 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

SQL> savepoint a2;

Savepoint created.

SQL> update newdepartment set dname='Marketing' where dno=1;

1 row updated.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Finance 3 123456789 12-OCT-10

Marketing 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

Research 5 333445555 22-MAY-88

SQL> savepoint A2;

Savepoint created.

SQL> update newdepartment set dname='HR' where dno=5;

1 row updated.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Finance 3 123456789 12-OCT-10

Marketing 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

HR 5 333445555 22-MAY-88

SQL> rollback to a2;

Rollback complete.

SQL> select \* from newdepartment;

DNAME DNO MGR\_SSN MGR\_START

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

Finance 3 123456789 12-OCT-10

Headquarters 1 888665555 19-JUN-81

Administration 4 987654321 01-JAN-95

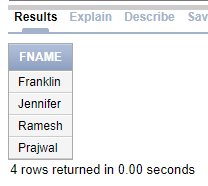
Research 5 333445555 22-MAY-88

**ASSIGNEMENT-3**

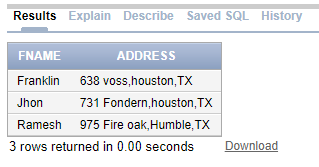
**Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS,UNION, EXCEPT ,INTERSECT,BETWEEN,LIKE,ORDER BY.**

1. **Retrieve the Fname and salary of all employee who is working for department 4 and make over salary 30000 or working for department 5 make over salary 25000.**

select fname from employee where (dno=4 and sal>30000) or (dno=5 and sal>25000)

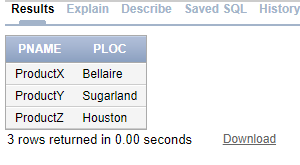


1. **Retrieve the Fname and address of all employee whose salary is in the range of 30000 to 40000.**

select fname,address from employee where (sal>=30000 and sal<=40000) 

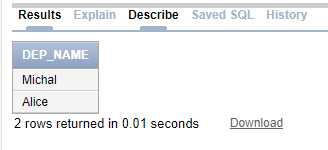
1. **Retrieve the Pname and plocation of all project controlled by department with number 5**

select pname,ploc from project where (dnum=5)



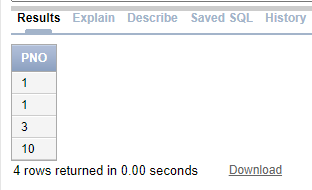
1. **Retrieve the Names of dependent born in the year 1988**

select dep\_name from dependent where bdate like '\_\_\_\_\_\_1988'



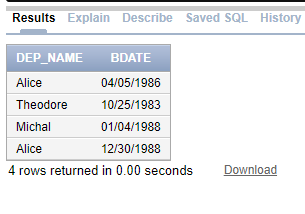
1. **Retrieve the pnumber for which employee worked for more than 30 hours**

select pno from works\_on where hours>30



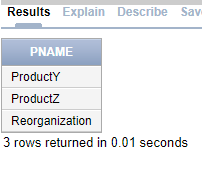
1. **Retrieve the Name and birth date of dependents who have daughter or son relationship with the employees.**

select dep\_name,bdate from dependent where relationship='son' or relationship='daughter'



1. **Retrieve the name of project which are located at a place which contains a letter u**

select pname from project where ploc like '%u%'



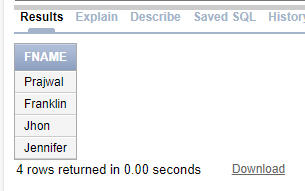
1. **Retrieve the names of employee who are mangers of the department**

select fname from employee,department where ssn=mgr\_ssn



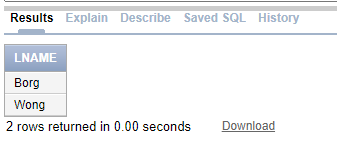
1. **Retrieve the names of each employee who has dependent**

select distinct fname from employee,dependent where ssn=essn



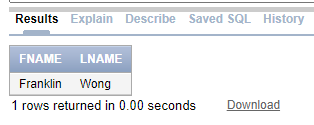
1. **For every department located ad Houston retrieve the Manger Lname and salary if Lname of employee contains only four characters.**

select lname from employee,department,dept\_loc where dloc='Houston' and dept\_loc.dnum=department.dno and mgr\_ssn=ssn



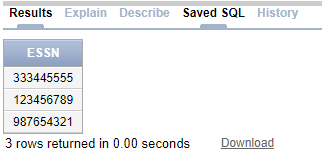
1. **Retrieve the Fname and Lname of employee who is a supervisor for a employee with first name as Ramesh**

select s.fname,s.lname from employee s where ssn in(select e.superssn from employee e where e.fname='Ramesh')



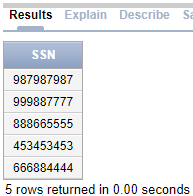
1. **List the ssn of all employee who are manger and also works on project**

select distinct essn from department,works\_on where mgr\_ssn=essn



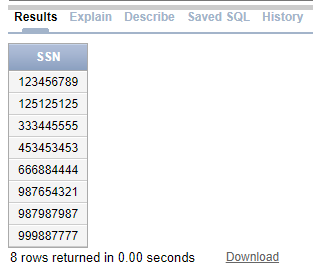
1. **Retrieve the ssn of employee who have no dependent**

select ssn from employee where ssn not in(select essn from dependent)



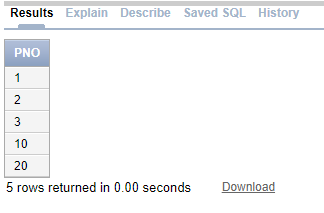
1. **List the ssn of all employee who either works on project or having the dependent**

select ssn from employee where ssn in(select essn from works\_on) or ssn in(select essn from dependent)



1. **Make a list of project numbers for projects that involve an employee whose last name is franklin either as worker or as manager of the department that controls the project.**

select distinct pno from employee,works\_on where essn=ssn and fname='Franklin' union select distinct pnum from project,employee,department where dnum=department.dno and fname='Franklin' and ssn=mgr\_ssn



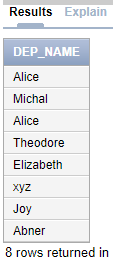
1. **Retrieve the Fname of employee in the increasing order of their salary and decreasing order of department number**

select fname from employee order by dno desc,sal asc



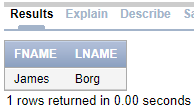
1. **Retrieve the Dependent name in the decreasing order of their birth date**

select dep\_name from dependent order by bdate desc



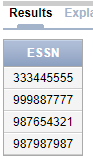
1. **Retrieve the firstname and lastname of all employee whose salary is more than the salary of employees working for department 5.**

select fname,lname from employee where sal>(select max(sal) from employee where dno=5)



1. **Retrieve the ssn of all employee who work on any of the project controlled by department 4**

select distinct essn from project,works\_on where pno=pnum and dnum=4



1. **Retrieve the name of each employee who has a dependent with same first name and is the same gender as the employee**

select fname from employee e,dependent d where essn=ssn and fname=dep\_name and e.gen=d.gen

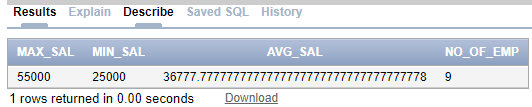


### ASSIGNMNENT-4

**Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING**

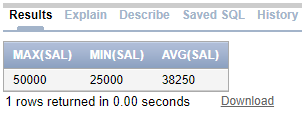
1. **Retrieve the maximum salary,the minimum salary and the average salary among all the employees also total number of employees in the company**

select max(sal) as max\_sal,min(sal) as min\_sal,avg(sal) as avg\_sal,count(\*) as no\_of\_emp from employee



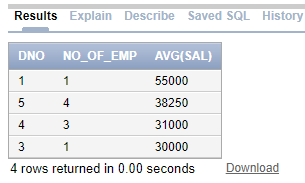
1. **Find the maximum salary, the minimum salary, and the average salary among employees who work for the 'Research' department.**

select max(sal),min(sal),avg(sal) from employee e,department d where e.dno=d.dno and dname='Research'



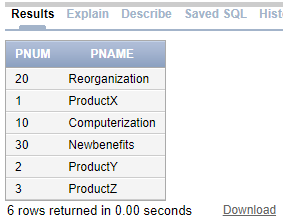
1. **For each department, retrieve the department number, the number of employees in the department, and their average salary.**

select dno,count(\*) as no\_of\_emp,avg(sal) from employee group by dno



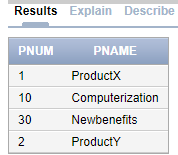
1. **For each project, retrieve the project number, project name, and the number of employees who work on that project.**

select pnum,pname from (project join works\_on on pnum=pno) group by pnum,pname



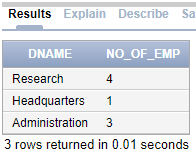
1. **For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.**

select pnum,pname from (project join works\_on on pnum=pno) group by pnum,pname having count(\*)>2



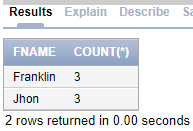
1. **For each department whose average employee salary is more than Rs 30000, retrieve the department name and number of employees working for that department.**

select dname,count(\*) as no\_of\_emp from employee e,department d where d.dno=e.dno group by e.dno,dname having avg(sal)>30000



1. **For each employee who have more than two dependents,retrieve the name of employee and number of dependents**

select fname,count(\*) from employee,dependent where ssn=essn group by fname,essn having count(\*)>2

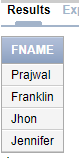


**ASSIGNMNENT-5**

**TO CREATE AND DROPPING OF VIEW**

1. **Create view that has the list of names of employees who have dependent and also display the view;**

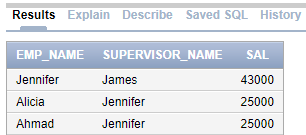
create view emp\_dep as select distinct fname from employee,dependent where ssn=essn



1. **Create a view that has the employee name,supervisor name and employee salary for each employee who works for administration department and also display the view**

create view emp\_sup as select e.fname as emp\_name,s.fname as Supervisor\_name,e.sal from employee e,employee s,department where e.superssn=s.ssn and department.dno=e.dno and dname='Administration'

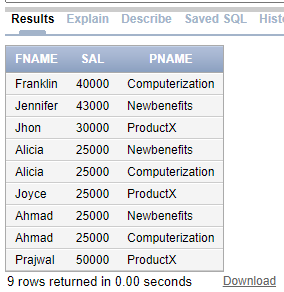
select \* from emp\_sup



1. **Create a view that has employee name, salary and project name which are located at Bellarie, Stafford use explicit set values.**

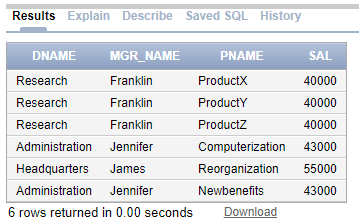
create view emp\_pro as select fname,sal,pname from employee,project,works\_on where (ploc='Bellaire' or ploc='Stafford') and pnum=pno and ssn=essn

select \* from emp\_pro



1. **Create view that has department name, manager name,project name and salary for every projectand also display the view.**

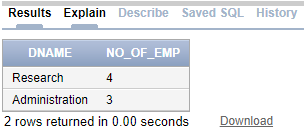
create view dpt\_pro as select dname,fname as mgr\_name,pname,sal from employee,project,department where dnum=department.dno and mgr\_ssn=ssn



1. **Create view that has the department name,number of employees working in the department, for each department with more than one employee working in the department display the view.**

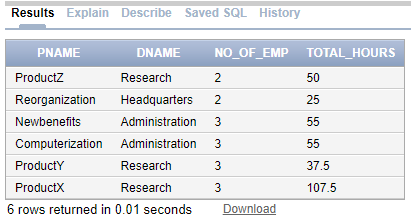
create view emp\_dept as select dname,count(\*) as no\_of\_emp from employee e,department d where e.dno=d.dno group by e.dno,dname having count(\*)>1

select \* from emp\_dept



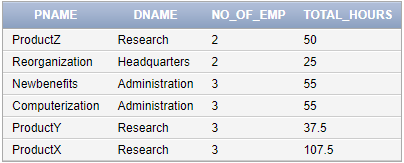
1. **Create view that has the project name,controlling department name,number of employees and total number of hours worked on the project for each projectand also display the view.**

create view pro\_dep as select pname,dname,count(\*) as no\_of\_emp,sum(hours) as total\_hours from department,project,works\_on where pno=pnum and dno=dnum group by pname,dname



1. **Create view that has the project name,controlling department name,number of employees and total number of hours worked on the project for each project with more than one employee working on itand also display the view.**

create view pro\_dep\_c as select pname,dname,count(\*) as no\_of\_emp,sum(hours) as total\_hours from department,project,works\_on where pno=pnum and dno=dnum group by pname,dname having count(\*)>1



1. **Create a view that has dnumber that has more than five employees retrieve the department number and number of its employees who are making more than Rs 40000also display the view**

create view dno\_emp\_sal as select dno,count(\*) as emp\_more\_than\_40000 from employee where dno in(select dno from employee group by dno having count(dno)>2) and sal>40000 group by dno



### ASSIGNMNENT-6

**Develop PL/SQL program using PROCEDURES.**

1. **Write a stored procedure to add new department record accept the input from user.**

create or replace procedure dep\_add(dna in department.dname%type,dn in department.dno%type,mssn in department.mgr\_ssn%type,mgrd in department.mgr\_start\_date%type) is

begin

insert into department values(dna,dn,mssn,mgrd);

dbms\_output.put\_line('Row successfully added!');

end;

/

declare

dna department.dname%type:=:department\_name;

dn department.dno%type:=:department\_number;

mssn department.mgr\_ssn%type:=:Mgr\_sssn;

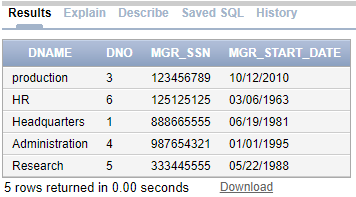
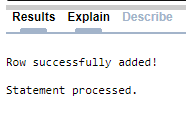
md department.mgr\_start\_date%type:=:Mgr\_start\_date;

begin

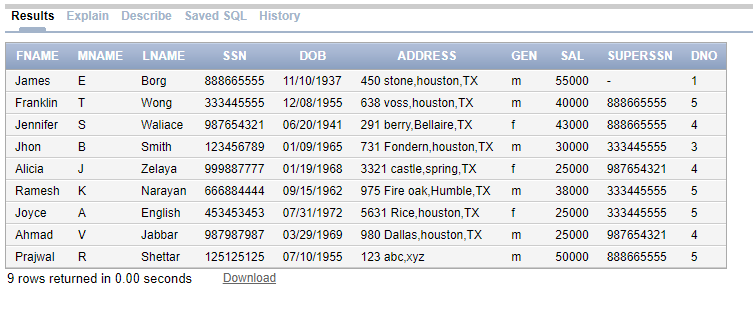
dep\_add(dna,dn,mssn,md);

end;

/



1. **Write a stored procedure to display the employee name and salary if name of the working department and ssn of the employee is passed.**



create or replace procedure disp\_emp(dn in department.dname%type,ss in employee.ssn%type) is

en employee.fname%type;

sa employee.sal%type;

begin

select fname,sal into en,sa from employee e,department d where ssn=ss and dn=dname and e.dno=d.dno;

dbms\_output.put\_line('Name Salary');

dbms\_output.put\_line(en||' '||sa);

end;

/

declare

dn department.dname%type:=:Dname;

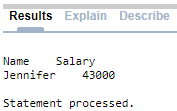
ss employee.ssn%type:=:SSN;

begin

disp\_emp(dn,ss);

end;

/



1. **Write a stored procedure to update the salary (by 10%) of the employee who is working on the project named ProductX and hours more than 30.**

create or replace procedure emp\_sal as

begin

update employee set sal=sal+sal\*0.1 where ssn in(select ssn from employee,project,works\_on where pname='ProductX' and pno=pnum and hours>30 and ssn=essn);

end;

/

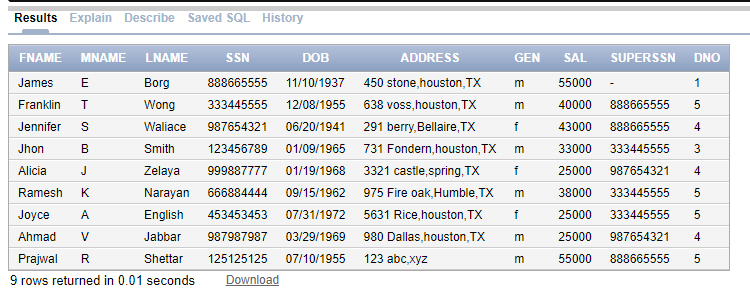
begin

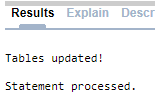
emp\_sal();

dbms\_output.put\_line('Tables updated!');

end;

/





### ASSIGNMNENT-7

**Develop PL/SQL program using FUNCTIONS.**

1. **Write a stored function to display salary of employee if ssn of employee is passed as an argument.**

create or replace function get\_sal(ss in employee.ssn%type)

return employee.sal%type

is

sa employee.sal%type;

begin

select sal into sa from employee where ssn=ss;

return sa;

end;

/

declare

ss employee.ssn%type:=:SSN;

sa employee.sal%type;

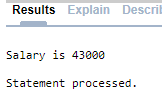
begin

sa:=get\_sal(ss);

dbms\_output.put\_line('Salary is '||sa);

end;

/



1. **Write a stored function to display total salary of all employees who are working on the project, accept the project number from user.**

create or replace function get\_sal\_sum(pn in works\_on.pno%type)

return number

is

ss number;

begin

select sum(sal) into ss from employee,works\_on where ssn=essn and pno=pn;

return ss;

end;

/

declare

pn works\_on.pno%type:=:PNO;

ss number;

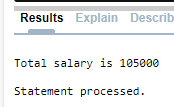
begin

ss:=get\_sal\_sum(pn);

dbms\_output.put\_line('Total salary is '||ss);

end;

/



1. **Write a stored function to display total number of employees working for the project located at particular location. Accept location from user.**

create or replace function get\_count(lo in project.ploc%type)

return number

as

co number;

begin

select count(\*) into co from employee,project,works\_on where ploc=lo and pnum=pno and essn=ssn;

return co;

end;

/

declare

pl project.ploc%type:=:Plocation;

c number;

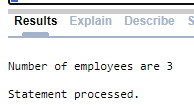
begin

c:=get\_count(pl);

dbms\_output.put\_line('Number of employees are '||c);

end;

/



### ASSIGNMNENT-8

**Develop PL/SQL programs using CURSOR.**

1. Using implicit cursor update the salary of the employee are working on project number (1,10) and also display the number employees salary updated. If number of rows updated more than 5 then print many employees are benefited otherwise few benifited.

declare

ct number;

ss employee.ssn%type;

begin

update employee set sal=sal+sal\*0.01 where ssn in(select distinct essn from works\_on where pno=1 or pno=10);

ct:=sql%rowcount;

dbms\_output.put\_line('Total number of rows effected are : '||ct);

if ct>5 then

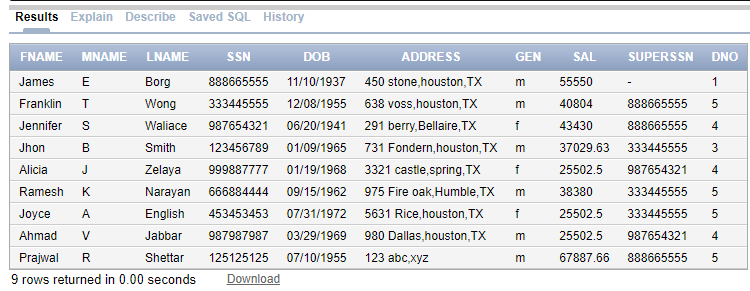
dbms\_output.put\_line('Many employees are benefited');

else

dbms\_output.put\_line('Few employees benefitted');

end if;

end;



1. **Define a EXPLICIT cursor to display the ssn, fname and dependent names of employees.**

declare

cursor ck\_emp is select ssn,fname,dep\_name from employee,dependent where ssn=essn;

ss employee.ssn%type;

fn employee.fname%type;

dn dependent.dep\_name%type;

begin

dbms\_output.put\_line('Emp\_ssn Emp\_name Dep\_name');

open ck\_emp;

loop

fetch ck\_emp into ss,fn,dn;

exit when ck\_emp%notfound;

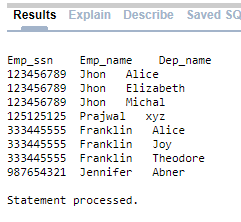
dbms\_output.put\_line(ss||' '||fn||' '||dn);

end loop;

close ck\_emp;

end;

/



### ASSIGNMNENT-9

**Develop PL/SQL programs using Triggers**

1. **Create trigger to display total number of employees working for department research in organization/company before inserting new record .**

create or replace trigger emp\_tri before insert on employee

declare

ct number;

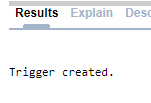
begin

select count(\*) into ct from employee join department on employee.dno=department.dno where dname='Research';

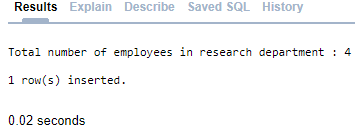
dbms\_output.put\_line('Total number of employees in research department : '||ct);

end;

/



insert into employee values('abc','p','xys',321654987,'06/09/2000','132 sd,TX','m',10000,'888665555','7');



1. **Write a trigger to display the salary difference between the old and new salary before updating.**

create or replace trigger emp\_tri before update on employee for each row

declare

dif number;

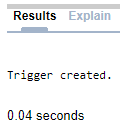
begin

dif:=:new.sal-:old.sal;

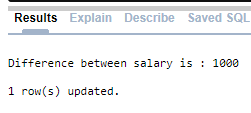
dbms\_output.put\_line('Difference between salary is : '||dif);

end;

/



update employee set sal=sal+sal\*0.1 where ssn=321654987;



**ASSIGNMENT-10**

**Develop Programs using Packages.**

1. **Create a package prj\_info which include one procedure named prj\_insert to insert a new project details controlled by department 1. Package also includes one function prj\_count to find the total number of project controlled by department 1.**

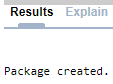
create or replace package prj\_infoa is

procedure prj\_insert(pn in project.pname%type,pnumb in project.pnum%type,pl in project.ploc%type);

function prj\_count return integer;

end;

/



create or replace package body prj\_infoa is

procedure prj\_insert(pn in project.pname%type,pnumb in project.pnum%type,pl in project.ploc%type)

is

begin

insert into project values(pn,pnumb,pl,1);

dbms\_output.put\_line('Row inserted into project');

end prj\_insert;

function prj\_count return integer

is

c integer;

begin

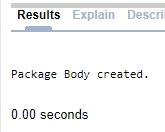
select count(\*) into c from project where dnum=1;

return c;

end prj\_count;

end prj\_infoa;

/



declare

pn project.pname%type:=:Pname;

pnumb project.pnum%type:=:Pnum;

pl project.ploc%type:=:Plocation;

ct integer;

begin

prj\_infoa.prj\_insert(pn,pnumb,pl);

ct:=prj\_infoa.prj\_count();

dbms\_output.put\_line('Number of projects from department 1 is '||ct);

end;

/

