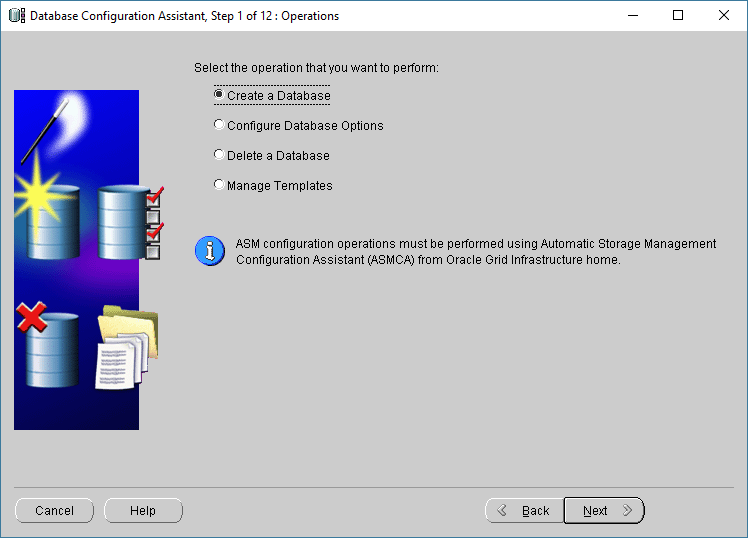
**Practical No : 1**

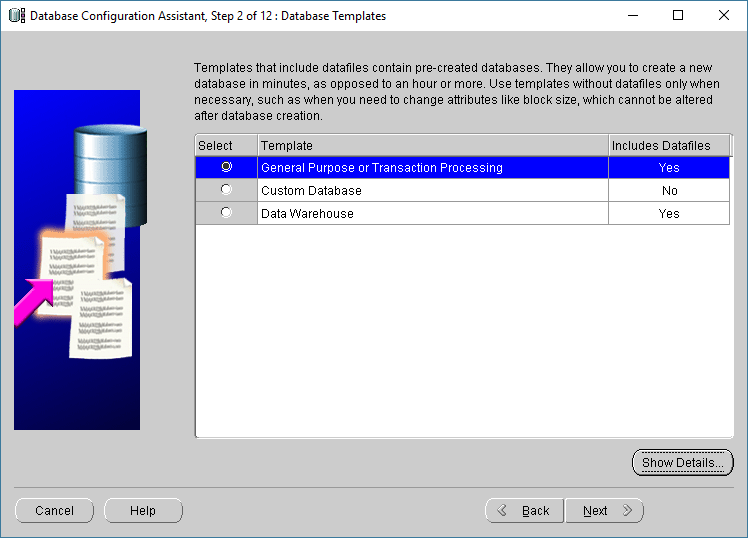
**Aim :**For a given a global conceptual schema, divide the schema into vertical fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.

**Step 1:**Goto->start->all program->oracle-oraHome92->Configuration and migration tools->Database Configuration Assistant

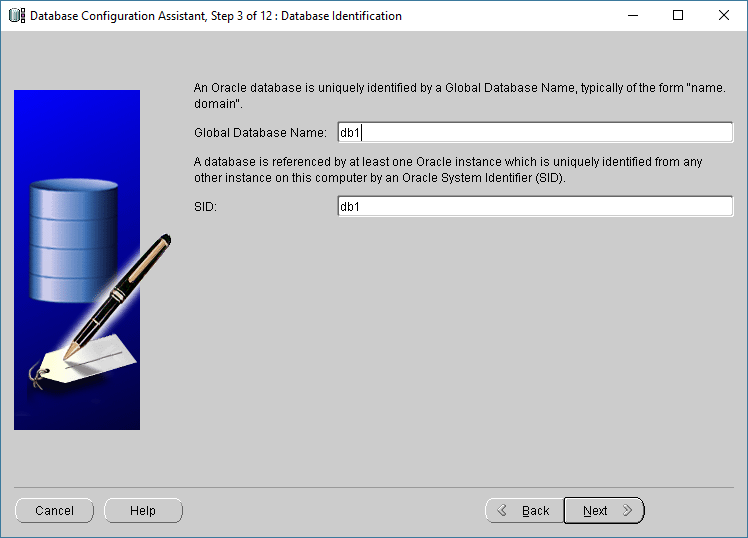
**Step 2:** Launch the launcher file from the menu, which look like below figure.



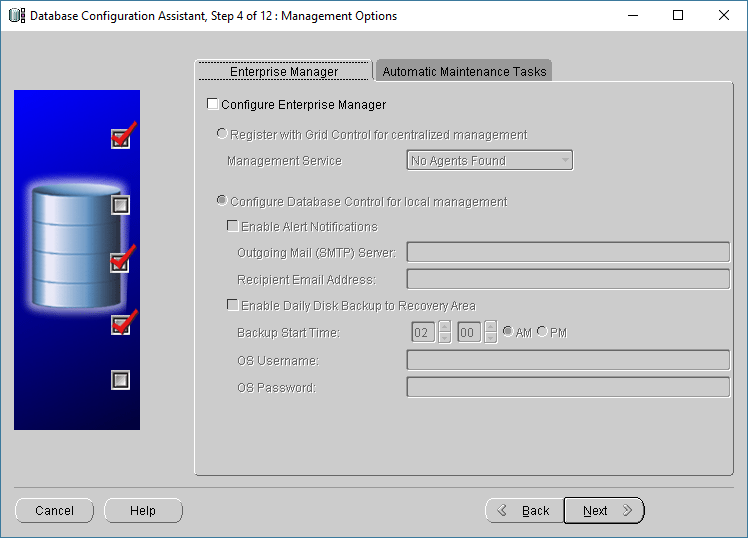
**Step 3:** First step is to select the option of creating a database.



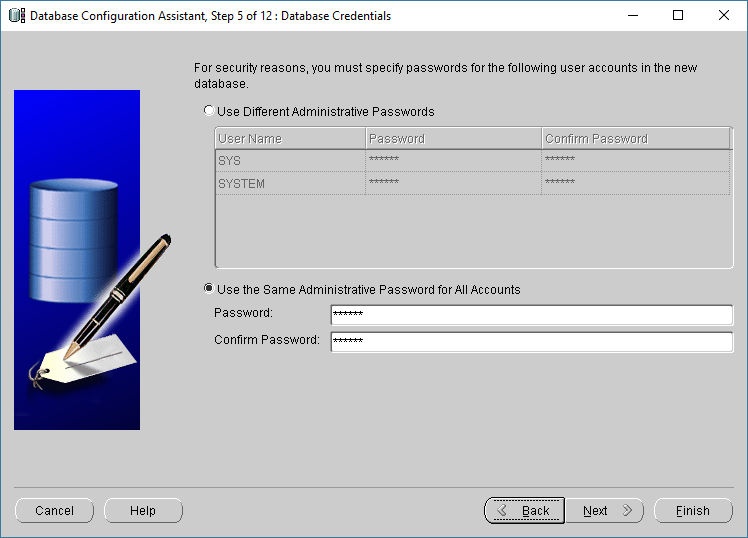
**Step 4:** Second step is to select the general purpose database option.



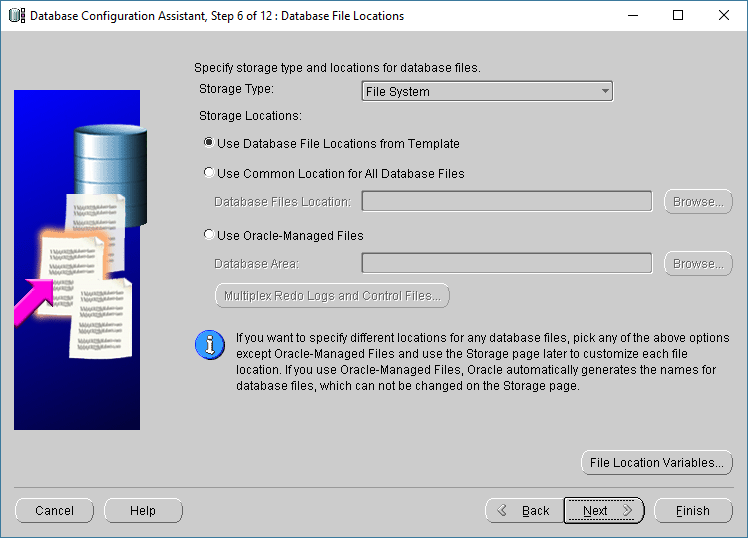
**Step 5:** Select the dedicated Management Options depends on your choice.



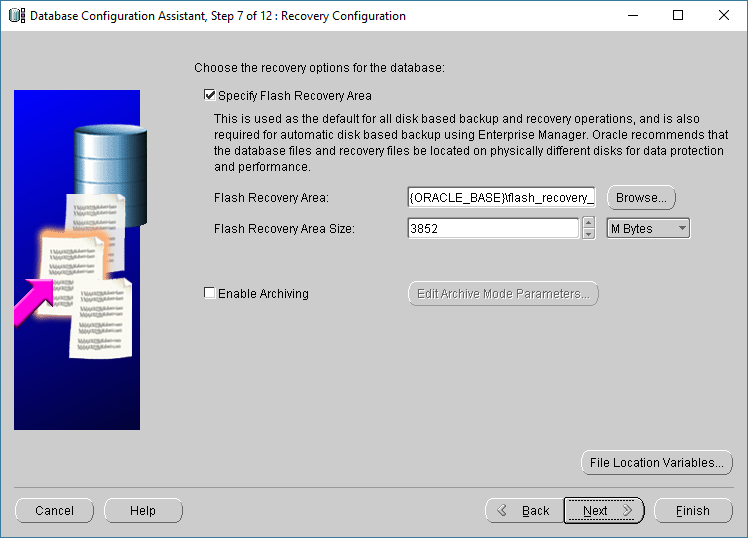
**Step 6:** Assign the same password to all accounts by selecting option.



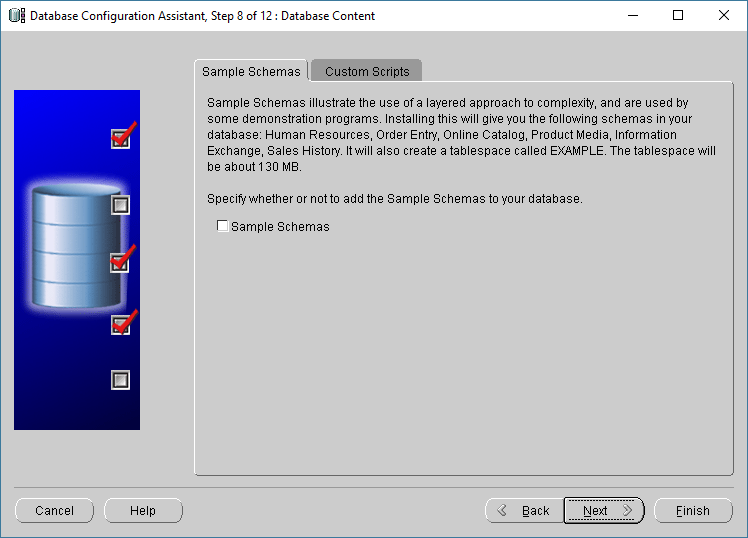
**Step 7:**Specify storage type and locations for database files.



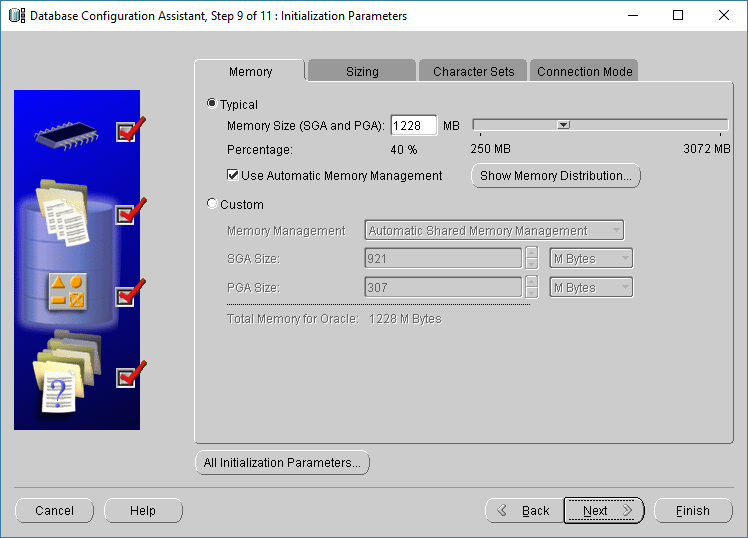
**Step 8:**Choose the recovery options for the database.



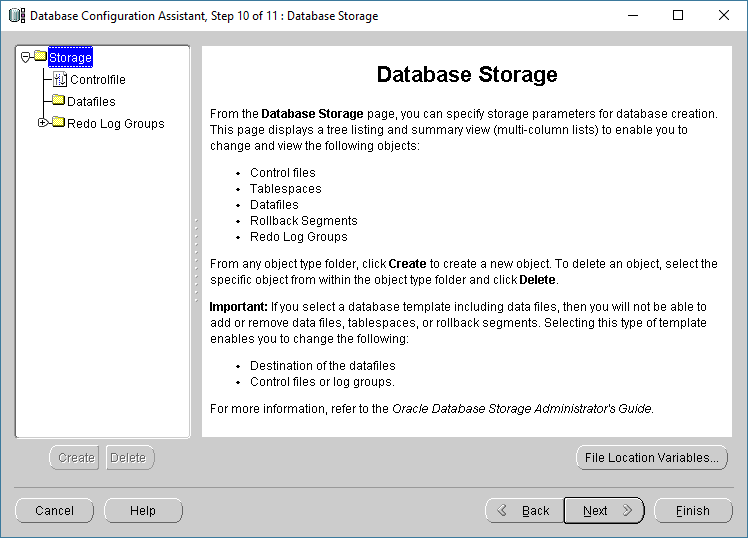
**Step 9:** Click on Next.



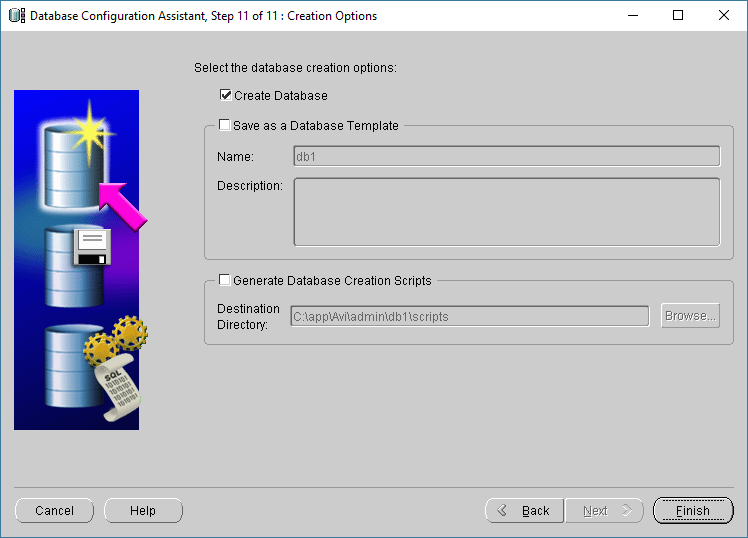
**Step 10:** Memory allocation in Initialization Parameters, Click on the next.



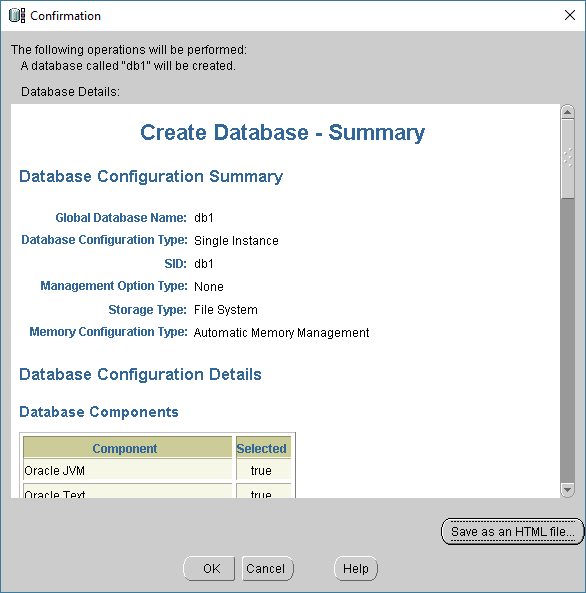
**Step 11:** Click on Next.



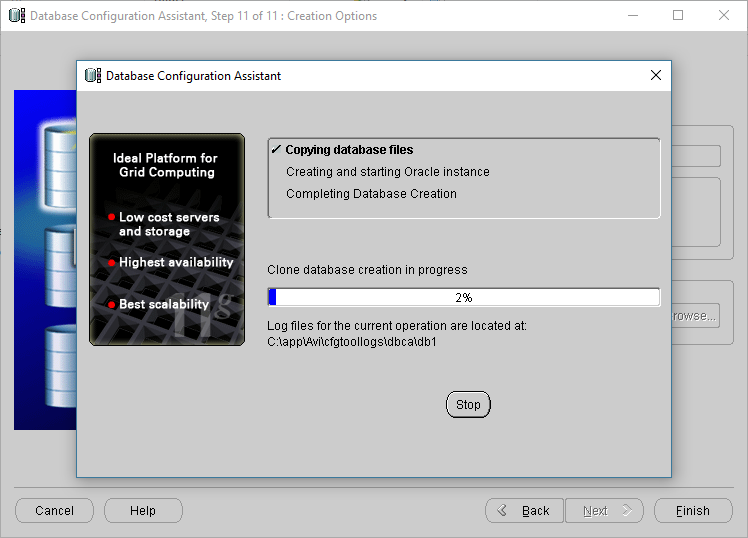
**Step 12:** Click on Finish.



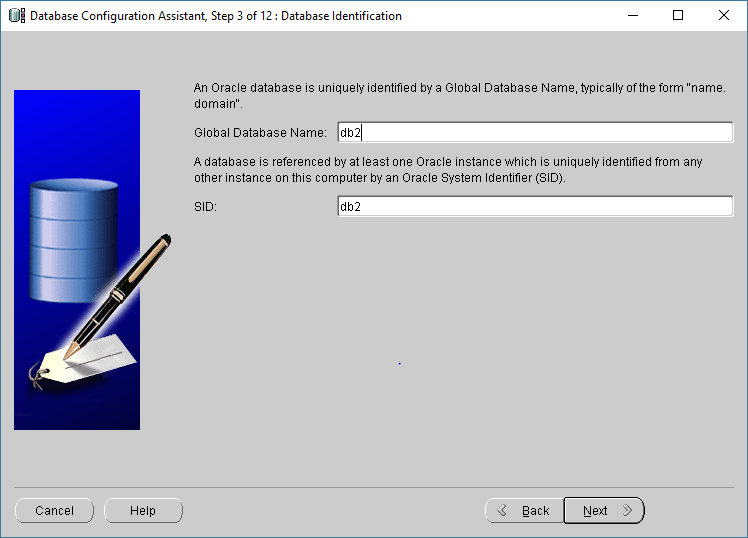
**Step 13:** Click on ok, to start the creation of database .



**Step 14:** Wait for some time till system create database.



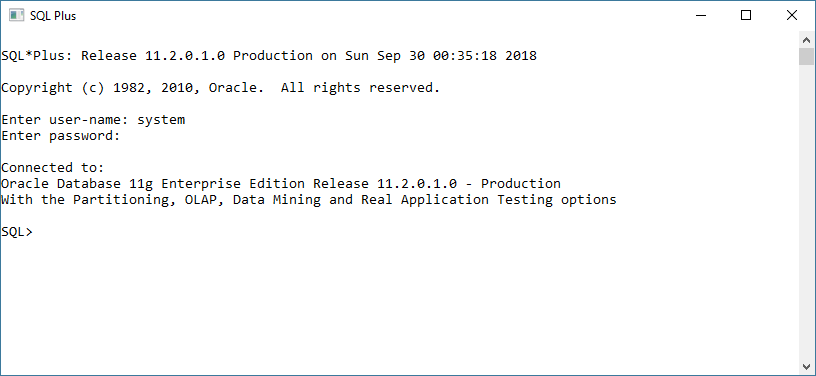
**Step 15:** Similarly, create the another database named as db2.



**Step 16:** Open SQL Plus.



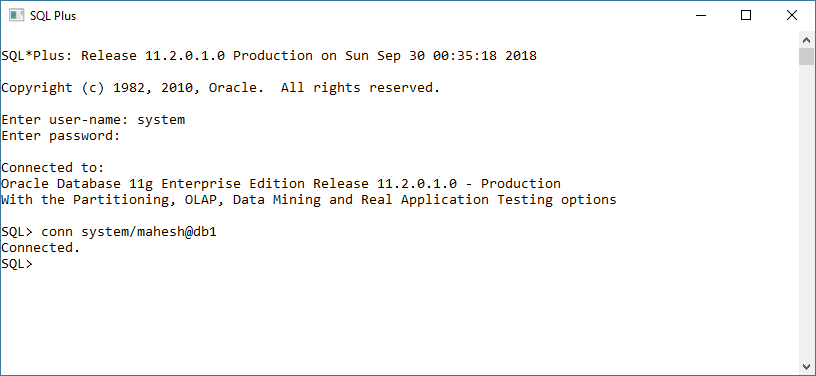
**Step 17:** Provide the username and password into sql plus.



**Step 18:**Goto db1 database.

**SQL>** conn system/mahesh@db1

Connected.



**Step 19:** Create one table as employee table in database db1.

**SQL>** create table employee (enumber int Primary key, enamevarchar(20), address varchar(30), email varchar(20), salary int);

**Table created.**

**SQL>** insert into employee values(1,'Mahesh','borivali','mm@gmail.com',20000);

**1 row created.**

**SQL>** insert into employee values(2,'Gangasagar','malad','gsi@gmail.com',15000);

**1 row created.**

**SQL>** insert into employee values(3,'Jay','goregoan','jk@gmail.com',18000);

**1 row created.**

**SQL>** insert into employee values(4,'Sandeep','kandivali','nova@gmail.com',9000);

**1 row created.**

**SQL>** insert into employee values(5,'Ranjeet','dadar','rk@gmail.com',7000);

**1 row created.**

**SQL>** select \*from employee;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

1 Mahesh borivali mm@gmail.com 15000

2 Gangasagarmalad gsi@gmail.com 15000

3 Jay goregoan jk@gmail.com 18000

4 Sandeep kandivali nova@gmail.com 9000

5 Ranjeet dadar rk@gmail.com 7000

**Step 20:** Enter following command to create link between two databases.

**SQL>** create database link db1todb2 connect to system identified by mahesh using 'db2';

**Database link created.**

**SQL>** create database link db2todb1 connect to system identified by mahesh using 'db1';

**Database link created.**

**Step 21:** Create 4 fragmentation

**SQL>** create table emp1 as select enumber,ename,salary from employee@db2todb1;

**Table created.**

**SQL>** select \*from emp1;

ENUMBER ENAME SALARY

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

1 Mahesh 15000

2 Gangasagar 15000

3 Jay 18000

**SQL>** create table emp11 as select enumber,ename,email from employee@db2todb1;

**Table created.**

**SQL>** select \*from emp11;

ENUMBER ENAME EMAIL

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

1 Mahesh mm@gmail.com

2 Gangasagar gsi@gmail.com

3 Jay jk@gmail.com

4 Sandeep nova@gmail.com

5 Ranjeet [rk@gmail.com](mailto:rk@gmail.com)

**SQL>** create table emp2 as select enumber,ename,address from employee@db2todb1;

**Table created.**

**SQL>** select \*from emp2;

ENUMBER ENAME ADDRESS

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

1 Mahesh borivali

2 Gangasagarmalad

3 Jay goregoan

4 Sandeep kandivali

5 Ranjeet dadar

**SQL>** create table emp22 as select enumber,ename,email,salary from employee@db2todb1;

**Table created.**

**SQL>** select \*from emp22;

ENUMBER ENAME EMAIL Salary

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

1 Mahesh [mm@gmail.com](mailto:mm@gmail.com) 15000

2 Gangasagar[gsi@gmail.com](mailto:gsi@gmail.com) 15000

3 Jay [jk@gmail.com](mailto:jk@gmail.com) 18000

4 Sandeep [nova@gmail.com](mailto:nova@gmail.com) 9000

5 Ranjeet [rk@gmail.com](mailto:rk@gmail.com) 7000

**Step 22:** Run Following Commands.

**Output :-**

**1)** Find the salary of all employee

**SQL>** select salary from emp1;

SALARY

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

15000

15000

18000

9000

7000

**2)** Find email of all employee where salary is greater than 8000

**SQL>** select email from emp4 where salary>8000;

EMAIL

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

mm@gmail.com

gsi@gmail.com

jk@gmail.com

nova@gmail.com

rk@gmail.com

**3)** Find the employee name ,email, where id is known

**SQL>** select ename,email from emp2 where enumber=3;

ENAME EMAIL

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

Jay [jk@gmail.com](mailto:jk@gmail.com)

**4)** Find the employee name , address where id is known

**SQL>** select ename,address from emp3 where enumber=3;

ENAME ADDRESS

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

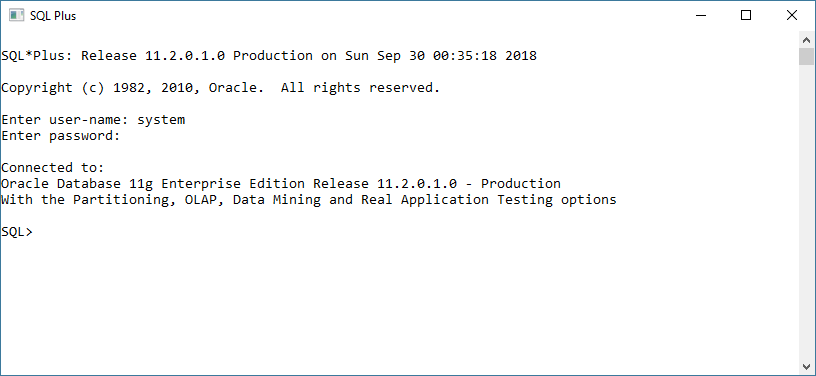
Jay malad

**Practical No : 2**

**Aim :**For a given a global conceptual schema, divide the schema into horizontal fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.

**Step 1:** Create Database db1 and db2

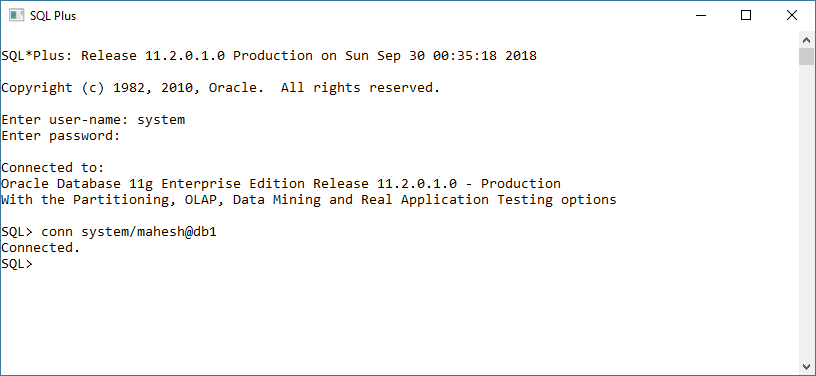
**Step 2:** Provide the username and password into sql plus.



**Step 3:**:Goto db1 database.

**SQL>** conn system/mahesh@db1

Connected.



**Step 4:** Create one table as employee table in database db1.

**SQL>** create table employee (enumber int Primary key, enamevarchar(20), address varchar(30), email varchar(20), salary int);

**Table created.**

**SQL>** insert into employee values(2,'Gangasagar','malad','gsi@gmail.com',15000);

**1 row created.**

**SQL>** insert into employee values(3,'Jay','goregoan','jk@gmail.com',18000);

**1 row created.**

**SQL>** insert into employee values(4,'Sandeep','kandivali','nova@gmail.com',9000);

**1 row created.**

**SQL>** insert into employee values(5,'Ranjeet','dadar','rk@gmail.com',7000);

**1 row created.**

**SQL>** select \*from employee;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

1 Mahesh borivali mm@gmail.com 15000

2 Gangasagarmalad gsi@gmail.com 15000

3 Jay goregoan jk@gmail.com 18000

4 Sandeep kandivali nova@gmail.com 9000

5 Ranjeet dadar rk@gmail.com 7000

**Step 5:** Enter following command to create link between two databases.

**SQL>** create database link db1todb2 connect to system identified by mahesh using 'db2';

**Database link created.**

**SQL>** create database link db2todb1 connect to system identified by mahesh using 'db1';

**Database link created.**

**Step 6:** Create table in database db2 using ‘db2todb1’ link.

**SQL>** create table emp3 as select \* from employee@db2todb1 where salary<=10000;

**Table created.**

**SQL>** select \*from emp3;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

4 Sandeep kandivali nova@gmail.com 9000

5 Ranjeet dadar rk@gmail.com 7000

**Step 7:** Create table in database db2 using ‘db2todb1’ link.

**SQL>** create table emp4 as select \* from employee@db2todb1 where address='borivali';

**Table created.**

**SQL>** select \* from emp4;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

1 Mahesh borivali mm@gmail.com 15000

**Step 8:**

**SQL>** Conn system/mahesh@db2;

**Connected.**

**SQL>** select salary from employee@db2todb1;

SALARY

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

15000

15000

18000

9000

8000

**SQL>** select email,salary from employee@db2todb1 where salary<=9000;

EMAIL SALARY

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

nova@gmail.com 9000

rk@gmail.com 7000

**SQL>** select ename, email from employee@db2todb1 where enumber=2;

ENAME EMAIL

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

Gangasagar gsi@gmail.com

**Step 9:** Creating table in ‘db1’ using table of ‘’db2’ using ‘db1todb2’ link

**SQL>** conn system/mahesh@ db1

**Connected.**

**SQL>** Create table emp3 as select \* from emp3@db1todb2 where address='borivali';

**Table created.**

**SQL>** select \* from emp3;

**SQL>** select \*from employee;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

4 Sandeep kandivali nova@gmail.com 9000

**Practical No : 3**

**Aim :** Place the replication of global conceptual schema on different nodes and execute queries that will demonstrate distributed databases environment.

**Query :**

**Creating Tables :-**

**In DB5**

**SQL>**create table emp (enumber number primary key, ename varchar2(10), addr varchar2(15), email varchar2(10),salary float)

**Table Created**

**In DB6**

**SQL>** create table emp (enumber number primary key, ename varchar2(10), addr varchar2(15), email varchar2(10),salary float)

**Table Created**

**Creating Link :-**

**SQL>**connect system/mahesh@db6

**Connected**

**SQL>**create public database link db5todb6 connect to system identified by mahesh using db6;

**Database link created**

**Creating Tiggers :-**

**SQL>**create or replace trigger insert\_data

after insert on emp

for each row

begin

insert into emp@db5todb6 values(:new.enumber,:new.ename,:new.address,:new.email,:new.salary);

end;

/

**Tigger created**

**SQL>**create or replace tiggerinsert\_data

before delete on emp

for each row

begin

delete from emp@db5todb6 where enumber=:old.enumber;

end;

/

**Tigger created**

**SQL>**create or replace tiggerinsert\_data

After update on emp

for each row

begin

update emp@db5todb6 set

enumber=:new.enumber,

ename=:new.ename,

addr=:new.addr,

email=:new.email,

salary=:new.salary

where enumber=old.enumber;

end;

/

**Tigger created**

**Inserting Values:-**

**SQL>** insert into employee values(1,'Mahesh','borivali','mm@gmail.com',20000);

**1 row created.**

**SQL>** insert into employee values(2,'Gangasagar','malad','gsi@gmail.com',15000);

**1 row created.**

**SQL>** insert into employee values(3,'Jay','goregoan','jk@gmail.com',18000);

**1 row created.**

**SQL>** insert into employee values(4,'Sandeep','kandivali','nova@gmail.com',9000);

**1 row created.**

**SQL>** insert into employee values(5,'Ranjeet','dadar','rk@gmail.com',7000);

**1 row created.**

**SQL>**connect system/mahesh@db5

**Connected**

**SQL>** select \*from employee;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

1 Mahesh borivali mm@gmail.com 15000

2 Gangasagarmalad gsi@gmail.com 15000

3 Jay goregoan jk@gmail.com 18000

4 Sandeep kandivali nova@gmail.com 9000

5 Ranjeet dadar rk@gmail.com 7000

**Updating Values:-**

**SQL>**connect system/mahesh@db6

**Connected**

**SQL>** update employee set enumber = 3, ename='Jay', address='malad', email='jk@gmail.com', salary=18000 where= enumber = 3;

**1 row updated.**

**SQL>**connect system/mahesh@db5

**Connected**

**SQL>** select \*from employee;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

1 Mahesh borivali mm@gmail.com 15000

2 Gangasagarmalad gsi@gmail.com 15000

3 Jay malad jk@gmail.com 18000

4 Sandeep kandivali nova@gmail.com 9000

5 Ranjeet dadar rk@gmail.com 7000

**SQL>**connect system/mahesh@db6

**Connected**

**SQL>**delete from employee where enumber=5;

**1 row deleted.**

**SQL>**connect system/mahesh@db5

**Connected**

**SQL>** select \*from employee;

ENUMBER ENAME ADDRESS EMAIL SALARY

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

1 Mahesh borivali mm@gmail.com 15000

2 Gangasagarmalad gsi@gmail.com 15000

3 Jay malad jk@gmail.com 18000

4 Sandeep kandivali nova@gmail.com 9000

1. Ranjeet dadar rk@gmail.com 7000

**Query :-**

1. **Find the salary of all employees.**

**SQL>**select enumber, ename, eesalary from employee;

ENUMBER ENAME SALARY

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

1 Mahesh 15000

2 Gangasagar 15000

3 Jay 18000

4 Sandeep 9000

5 Ranjeet 7000

1. **Find the email of all employees where salary = 15000.**

**SQL>** select email from employee where salary=15000;

EMAIL

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

gsi@gmail.com

1. **Find the employee name and email where employee number is known.**

**SQL>** select ename, email from employee where enumber=3;

ENAME EMAIL

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

Jay [jk@gmail.com](mailto:jk@gmail.com)

1. **Find the employee name and address where employee number is known.**

**SQL>** select ename, address from employee where enumber=3;

ENAME ADDRESS

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

Jay malad

**Practical No : 4**

**Aim :**Create different types that include attributes and methods. Define tables for these types by adding sufficient number of tuples. Demonstrate insert, update and delete operations on these tables. Execute queries on them.

Using Object Oriented databases create the following types:

a) AddrType1 (PinQuery: number, Street :char, City : char, state :char)

b) (ii)BranchType (address: AddrType1, phone1: integer,phone2: integer )

c) AuthorType (name:char,,addr AddrType1)

d) PublisherType (name: char, addr: AddrType1, branches: BranchTableType

e) AuthorListType as varray, which is a reference to AuthorType

Next create the following tables:

f) BranchTableType of BranchType

g) authors of AuthorType

h) books(title: varchar, year : date,

published\_by ref PublisherType,authorsAuthorListType)

i) Publishers of PublisherType

Insert 10 records into the above tables and fire the following queries:

a) List all of the authors that have the same pin Query as their publisher:

b) List all books that have 2 or more authors:

c) List the name of the publisher that has the most branches

d) Name of authors who have not published a book

e) List all authors who have published more than one book:

f) Name of authors who have published books with at least two different publishers

g) List all books (title) where the same author appears more than once on the list of authors (assuming that an integrity constraint requiring that the name of an author is unique in a list of authors has not been specified).

**Query:**

**Create table:**

**SQL>**Create or replace type AddrType1 as object (PinQuery number (5), Street char(20), City varchar2(50), state varchar2(40), no number(4) );

**Type created.**

**SQL>**create or replace type BranchType as object (address AddrType1, phone1 integer,phone2 integer );

**Type created.**

**SQL>**create or replace type BranchTableType as table of BranchType;

**Type created.**

**SQL**> create or replace type AuthorType as object (name varchar2 (50), addr AddrType1);

**Type created.**

**SQL>**create table Authors of AuthorType;

**Table created.**

**SQL>**create or replace type AuthorListType as varray(10) of ref AuthorType ;

**Type created.**

**SQL>**create or replace type PublisherType as object(name varchar2(50), addr AddrType1, branches BranchTableType);

**Type created.**

**SQL>**create table Publishers of PublisherType NESTED TABLE branches STORE as branchtable;

**Table created.**

**SQL>**create table books(title varchar2(50), year date, published\_by ref PublisherType, authors AuthorListType);

**Table created.**

**Inserting rows :**

**SQL**> insert into Authors values('Vigil', AddrType1(7000,'AT street', 'mumbai', 'maharashtra' ,1007));

**1 row created.**

**SQL>**insert into Authors values('Rohan', AddrType1(7007,'VT street','mumbai','maharashtra' ,1006));

**1 row created.**

**SQL>**insert into Authors values('Ameya',AddrType1(7003,'PL street','mumbai','maharashtra' ,1003));

**1 row created.**

**SQL>**insert into Authors values('Vigil',AddrType1(7008,'AT street', 'mumbai', 'maharashtra' ,1007));

**1 row created.**

**SQL>**insert into Authors values ('DonBox',AddrType1 (7006,'Nehrut','mumbai','maharashtra' ,1005));

**1 row created.**

**SQL>**insert into Authors values ('Haseeb', AddrType1(8002,'TH street','pune', 'maharashtra' ,13));

**1 row created.**

**SQL>**insert into Authors values('Ninad',AddrType1(7008,'TT street', 'Nasik','maharashtra' ,1008));

**1 row created.**

**SQL>**insert into Authors values('Richard',AddrType1(7002,'FL street','pune', 'maharashtra' ,03));

**1 row created.**

**SQL>**insert into Publishers values ('Vipul', AddrType1 (4002,'PK street', 'mumbai','maharashtra',03), BranchTableType(BranchType (AddrType1(5002,'PL street', 'mumbai', 'maharashtra', 03), 23406,69896)));

**1 row created.**

**SQL>**insert into Publishers values('McGraw',AddrType1(7007,'LJstreet','mumbai' ,'maharashtra',07), BranchTableType (BranchType ( AddrType1 (7007,'K street','mumbai', 'maharashtra',1007), 4543545,8676775)));

**1 row created.**

**SQL>**insert into Publishers values ('Tata',AddrType1(7008,'JW street','mumbai', 2 'maharashtra',27), BranchTableType (BranchType (AddrType1(1002,'DM street','nasik', 'maharashtra',1007), 456767,7675757)));

**1 row created.**

**SQL>**insert into Publishers values ('Nurali', AddrType1(7002,'ST street','pune','maharashtra' ,1007), BranchTableType (BranchType (AddrType1(1002,'SG street','pune', 'maharashtra', 1007), 4543545,8676775)));

**1 row created.**

**SQL>**insert into Publishers values('Tata', AddrType1(6002,'Gold street','nasik','maharashtra' ,1007), BranchTableType(BranchType(AddrType1(6002,'South street', 'nasik','mha',1007), 4543545,8676775)));

**1 row created.**

**SQL>**insert into books select 'IP','28-may-1983', ref (pub), AuthorListType(ref(aut)) from Publishers pub,Authorsaut where pub.name='Tata' and aut.name='Richard';

**2 rows created.**

**SQL>**insert into books select 'ADBMS','09-jan-1890',ref(pub), AuthorListType(ref(aut)) from Publishers pub,Authorsaut where pub.name='McGraw' and aut.name='Rohan';

**1 row created.**

**SQL**> insert into books select 'c prog','25-may-1983', ref (pub),AuthorListType(ref(aut)) from Publishers pub,Authorsaut where pub.name='Vipul' and aut.name='Haseeb';

**1 row created.**

**Firing Queries on the tables.**

1. **List all of the authors that have the same pin Query as their publisher:**

**Query:**

**SQL**>select a.name from Authors a, Publishers p where a.addr.pinQuery = p.addr.pinQuery;

**Output:**

NAME

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

Richard

Rohan

Ninad

Vigil

1. **List all books that have 2 or more authors**

**Query:**

**SQL**>Select title from books b where 2 <= (select count(\*) from table(b.authors));

**Output:**

TITLE

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

c prog

1. **List the name of the publisher that has the most branches**

**Query:**

**SQL**>Select p.name from publishers p, table (p.branches)

group by p.name having count(\*)> = all (select count(\*)from publishers p, table(p.branches) group by name);

**Output:**

NAME

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

Tata

1. **Name of authors who have not published a book**

**Query:**

**SQL**>select a.name from authors a where not exists(select b.title from books b,table(select authors from books b1 where b.title=b1.title)where a.name=name);

**Output:**

NAME

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

Rohan

1. **List all authors who have published more than one book**

**Query:**

**SQL**>select a.name from authors a, books b, table (b.authors) v where v.column\_value = ref(a) group by a.name having count(\*) > 1;

**Output:**

NAME

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

Vigil

Haseeb

Richard

1. **Name of authors who have published books with at least two different publishers**

**Query:**

**SQL**>select a.name from authors a, books b, table (b.authors) v where v.column\_value = ref(a) group by a;

**Output:**

NAME

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

Vigil

Haseeb

Richard

1. **List all books (title) where the same author appears more than once on the list of authors (assuming that an integrity constraint requiring that the name of an author is unique in a list of authors has not been specified).**

**Query:**

**SQL**>select title from authors a, books b, table (b.authors) v where v.column\_value = ref(a) group by title having count(\*) > 1;

**Output:**

TITLE

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

ADBMS

IP

c prog

**Practical No : 5**

**Aim :**Create a temporal database and issue queries on it.

**Query:**

**Create table:**

**SQL>**create table Emp\_Appnt(Acc\_No number(10),Name varchar2(10),RECDate date,

RETDate date);

**Table created.**

**Inserting rows :**

**SQL>**insert into Emp\_Appnt values(2025,'Mahesh','12-feb-2005','12-oct-2011') ;

**1 row created.**

**SQL>**insert into Emp\_Appnt values(2211,'Naveen','16-march-2008','16-sep-2010') ;

**1 row created.**

**SQL>**insert into Emp\_Appnt values(2221,'Jay’,'18-june-2004','18-july-2006') ;

**1 row created.**

**SQL>**insert into Emp\_Appnt values(2221,'Ragini','18-june-2004','21-july-2008') ;

**1 row created.**

**SQL>** insert into emp\_appnt values(2000,'Meeta','16-oct-2003','16-sep-2010');

**1 row created.**

**SQL>**select \* from emp\_appnt ;

ACC\_NO NAME RECDATE RETDATE

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

2025 Mahesh 12-FEB-05 12-OCT-11

2211 Naveen 16-MAR-08 16-SEP-10

2221 Jay 18-JUN-04 18-JUL-06

2221 Ragini 18-JUN-04 21-JUL-08

2000 Meeta 16-OCT-03 16-SEP-10

**Queries:**

**SQL>**select \* from emp\_appnt where RECDate='18-june-2004';

ACC\_NO NAME RECDATE RETDATE

---------------- ---**------------**---------------- --**------------------------**

2221 Ragini 18-JUN-04 18-JUL-06

**SQL>**select \* from emp\_appnt where RETDate='16-sep-2010';

ACC\_NO NAME RECDATE RETDATE

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

2211 Naveen 16-MAR-08 16-SEP-10

2000 Meeta 16-OCT-03 16-SEP-10

**SQL>**create table tbl\_shares(C\_Name varchar2(10),No\_Share Number(10),Price number(10),TransTime varchar2(10) Default To\_char(sysdate,'HH:MI'));

**Table created.**

**Inserting rows :**

**SQL>**insert into tbl\_shares(C\_Name,No\_Share,Price) values('Mahesh',123,500) ;

**1 row created.**

**SQL>**insert into tbl\_shares(C\_Name,No\_Share,Price) values(‘Naveen’,121,810,) ;

**1 row created.**

**SQL>**insert into tbl\_shares(C\_Name,No\_Share,Price) values(‘Jay’,233,600) ;

**1 row created.**

**SQL>**insert into tbl\_shares(C\_Name,No\_Share,Price) values(‘Ragini’,203,650) ;

**1 row created.**

**SQL>** insert into tbl\_shares(C\_Name,No\_Share,Price) values(‘Prasad’,212,880);

**1 row created.**

**SQL>**select \* from tbl\_shares;

C\_NAME NO\_SHARE PRICE TRANSTIME

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

Mahesh 123 500 02:03

Naveen 121 810 02:04

Jay 233 600 02:05

Ragini 203 650 02:06

Prasad 212 880 02:06

**SQL>**select \* from tbl\_shares where price>100 and TransTime='02:05';

C\_NAME NO\_SHARE PRICE TRANSTIME

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

Jay 233 600 02:05

**SQL>**select \* from tbl\_shares where price=(select max(price) from tbl\_shares where TransTime='02:04');

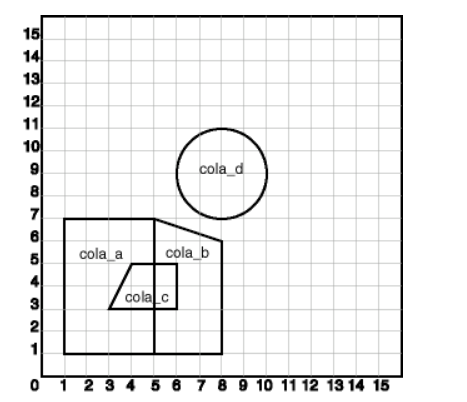
C\_NAME NO\_SHARE PRICE TRANSTIME

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

Naveen 121 810 02:04

**Practical No : 6**

**Aim :** Create a table that stores spatial data and issue queries on it.

****

Create a spatial database table that stores the number, name and location, which consists of four different areas say abc, pqr, mno and xyz.

Fire the following queries:

a)Find the topological intersection of two geometries.

b)Find whether two geometric figures are equivalent to each other.

c)Find the areas of all different locations.

d)Find the area of only one location.

e) Find the distance between two geometries.

**Query for Creating Table :**

**SQL>** create table cola\_mrp (mkt\_id number primary key,name varchar(20), shape MDSYS.SDO\_Geometry);

**Queries for inserting rows :**

**SQL>** insert into cola\_mrp values (1,'cola\_a',MDSYS.SDO\_GEOMETRY(2003,NULL,NULL,

MDSYS.SDO\_ELEM\_INFO\_ARRAY(1,1003,3), MDSYS.SDO\_ORDINATE\_ARRAY(1,1,5,7)))

/

**SQL>**insert into cola\_mrpvalues(2,'cola\_b',MDSYS.SDO\_GEOMETRY(2003,NULL,NULL,

MDSYS.SDO\_ELEM\_INFO\_ARRAY(1,1003,1),

MDSYS.SDO\_ORDINATE\_ARRAY(5,1,8,1,8,6,5,7,5,1)))

/

**SQL>**insert into cola\_mrpvalues(3,'cola\_c',MDSYS.SDO\_GEOMETRY( 2003,NULL,NULL,

MDSYS.SDO\_ELEM\_INFO\_ARRAY(1,1003,1),

MDSYS.SDO\_ORDINATE\_ARRAY(3,3,6,3,6,5,4,5,3,3)))

/

**SQL>**insert into cola\_mrpvalues(4,'cola\_d',MDSYS.SDO\_GEOMETRY(2003,NULL,NULL,

MDSYS.SDO\_ELEM\_INFO\_ARRAY(1,1003,4),

MDSYS.SDO\_ORDINATE\_ARRAY(7,9,10,9,8,11)))

/

**Creating Metadata information:**

insert into user\_SDO\_GEOM\_METADATAvalues('cola\_mrp','shape',

MDSYS.SDO\_DIM\_ARRAY(

MDSYS.SDO\_DIM\_ELEMENT('X',0,20,0.005),

MDSYS.SDO\_DIM\_ELEMENT('Y',0,20,0.005)),NULL);

**Query for creating index :**

create index cola\_spatial\_idx on cola\_market(location) Indextype Is mdsys.spatial\_index;

**Queries :**

**1) Find the topological intersection of two geometries.**

select SDO\_GEOM.SDO\_INTERSECTION (c\_a.shape,c\_c.shape,0.005)

from cola\_mrpc\_a,cola\_mrpc\_c

where c\_a.name='cola\_a' AND c\_c.name='cola\_c';

**Output :-**

, SDO\_SRID, SDO\_PO

SDO\_GEOM.SDO\_INTERSECTION(C\_A.SHAPE,C\_C.SHAPE,0.005)(SDO\_GTYPE --------------------------------------------------------------------------------

SDO\_GEOMETRY(2003, NULL, NULL, SDO\_ELEM\_INFO\_ARRAY(1, 1003, 1), SDO\_ORDINATE\_ARRAY(4, 5, 3, 3, 5, 3, 5, 5, 4, 5))

**2) Find whether two geometric figures are equivalent to each other.**

SELECT SDO\_GEOM.RELATE(c\_c.shape, 'EQUAL', c\_a.shape,0.005)

FROM cola\_mrpc\_c, cola\_mrpc\_a

WHERE c\_c.name='cola\_c' AND c\_a.name = 'cola\_a';

**Output :-**

SDO\_GEOM.RELATE(C\_C.SHAPE,'EQUAL',C\_A.SHAPE,0.005)

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

FALSE

**3) Find the areas of all different locations**

select name,SDO\_GEOM.SDO\_AREA(shape,0.005) from cola\_mrp;

**Output :-**

NAME SDO\_GEOM.SDO\_AREA(SHAPE,0.005)

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

cola\_a 24

cola\_b 16.5

cola\_c 5

cola\_d 7.85398163

4) Find the area of only one location.

select c.name,SDO\_GEOM.SDO\_AREA(c.shape,0.005) from cola\_mrp c

where c.name='cola\_a';

**Output :**

NAME SDO\_GEOM.SDO\_AREA(C.SHAPE,0.005)

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

cola\_a 24

**5) Find the distance between two geometries.**

select SDO\_GEOM.SDO\_DISTANCE(c\_b.shape,c\_d.shape,0.005)

from cola\_mrpc\_b,cola\_mrpc\_d

where c\_b.name= 'cola\_b' AND c\_d.name ='cola\_d';

**Output :-**

SDO\_GEOM.SDO\_DISTANCE(C\_B.SHAPE,C\_D.SHAPE,0.005)

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

1.8973666

**Practical No : 7**

**Aim :** Formulate a database using active rules with row and statement level.

**Create table:**

**SQL>**create table Project1 (pname varchar2(10), pnonumber(5) primary key, thrs number(5),

head\_nonumber(5));

**Table created.**

**SQL>**create table Employee1 (enonumber(5) primary key, ename varchar2(10), hrs number(5), super\_no number(5),pno number(5));

**Table created.**

**SQL>**alter table Employee1 add constraint et\_1 foreign key(pno) references Project1(pno);

**Table altered.**

**Queries :**

1. **Inserting into Project1:-**

**SQL>**insert into Project1 values('prj1',001,5,1);

**1 row created.**

**SQL>**insert into Project1 values('prj2',002,10,2);

**1 row created.**

**SQL>**insert into Project1 values('prj3',003,10,3);

**1 row created.**

**SQL>**insert into Project1 values('prj4',004,8,4);

**1 row created.**

**SQL>**insert into Project1 values('prj5',005,5,5);

**1 row created.**

1. **create or replace a trigger to insert a new employee tuple and display the new total hours from project table.**

**SQL>**create or replace trigger empinsert

after insert on Employee1 for each row

when (new.pno is not NULL)

update Project1

set thrs=thrs+:new.hrs

where pno=:new.pno

/

**Trigger created.**

1. **Inserting into Employee1:-**

**SQL>**insert into Employee1 values(0001,'Mahesh',5,2,001);

**1 row created.**

**SQL>**insert into Employee1 values(0002,'Naveen',4,3,002);

**1 row created.**

**SQL>**insert into Employee1 values(0003,'Ragini',6,4,004);

**1 row created.**

**SQL>**insert into Employee1 values(0004,'Aakash',6,2,002) ;

**1 row created.**

**SQL>**insert into Employee1 values(0005,'Swapnil',5,3,005);

**1 row created.**

1. **Creating a trigger to change the hrs of existing employee and display the new total hours from project table.**

**SQL>**create or replace trigger emphrs

after update of hrs on Employee1

for each row

when(new.pno is not NULL)

update Project1

set thrs=thrs+:new.hrs-:old.hrs

where pno=:new.pno

/

**Trigger created**

**Output:-**

Before Trigger :-

**SQL> select \* from Employee1;**

ENO ENAME HRS SUPER\_NO PNO

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

1 Mahesh 5 2 1

2 Naveen 4 3 2

3 Ragini 6 4 4

4 Aakash 6 2 2

5 Swapnil 5 3 5

**5 rows selected.**

**SQL> select \* from Project1;**

PNAME PNO THRS HEAD\_NO

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

prj1 1 5 1

prj2 2 10 2

prj3 3 10 3

prj4 4 13 4

prj5 5 5 5

**5 rows selected.**

After trigger :-

**SQL>**update Employee1 set hrs=2 where eno=2;

**1 row updated.**

**SQL> select \* from Employee1;**

ENO ENAME HRS SUPER\_NO PNO

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

1 Mahesh 5 2 1

2 Naveen 2 3 2

3 Ragini 6 4 4

4 Aakash 6 2 2

5 Swapnil 5 3 5

**5 rows selected**.

**SQL> select \* from Project1;**

PNAME PNO THRS HEAD\_NO

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

prj1 1 5 1

prj2 2 8 2

prj3 3 10 3

prj4 4 13 4

prj5 5 5 5

**5 rows selected.**

1. **Creating a trigger to change the project of an employee and display the new total hours from project table.**

**SQL>**create or replace trigger empproj

after update on Employee1

for each row

update Project1

set thrs= thrs - :old.hrs

where pno=:old.pno ;

update Project1

set thrs=thrs + :new.hrs

where pno=:new.pno

/

**Table created.**

**Output:-**

Before Trigger**:-**

**SQL>**select \* from Employee1;

ENO ENAME HRS SUPER\_NO PNO

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

1 Mahesh 5 2 1

2 Naveen 2 3 2

3 Ragini 6 4 4

4 Aakash 6 2 2

5 Swapnil 5 3 5

**5 rows selected.**

**SQL>**select \* from Project1;

PNAME PNO THRS HEAD\_NO

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

prj1 1 5 1

prj2 2 10 2

prj3 3 10 3

prj4 4 13 4

prj5 5 5 5

**5 rows selected.**

After Trigger:-

**SQL>**Update Employee1 Setpno=2 where eno=3;

**1 row updated.**

**SQL>**select \* from Employee1;

ENO ENAME HRS SUPER\_NO PNO

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

1 Mahesh 5 2 1

2 Naveen 2 3 2

3 Ragini 6 4 2

4 Aakash 6 2 2

5 Swapnil 5 3 5

**5 rows selected.**

**SQL>**select \* from Project1;

PNAME PNO THRS HEAD\_NO

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

prj1 1 5 1

prj2 2 8 2

prj3 3 10 3

prj4 4 13 4

prj5 5 5 5

**5 rows selected.**

1. **Creating a trigger to deleting the project of an employee.**

**SQL>**create or replace trigger delemp

after update of pno on Employee1

for each row

update Project1

set thrs=thrs-:old.hrs

where pno=:old.pno

/

**Table created.**

**Output:-**

Before Trigger :-

**SQL>**select \* from Employee1;

ENO ENAME HRS SUPER\_NO PNO

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

1 Mahesh 5 2 1

2 Naveen 2 3 2

3 Ragini 6 4 2

4 Aakash 6 2 2

5 Swapnil 5 3 5

**5 rows selected.**

**SQL**> select \* from Project1;

PNAME PNO THRS HEAD\_NO

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

prj1 1 5 1

prj2 2 8 2

prj3 3 10 3

prj4 4 13 4

prj5 5 5 5

**5 rows selected.**

After Trigger :-

**SQL>**update Employee1 set pno=NULL where eno=2;

**1 row updated.**

**SQL>**select \* from Employee1;

ENO ENAME HRS SUPER\_NO PNO

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

1 Mahesh 5 2 1

2 Naveen 2 3

3 Ragini 6 4 2

4 Aakash 6 2 2

5 Swapnil 5 3 5

**5 rows selected.**

**SQL>**select \* from Project1;

PNAME PNO THRS HEAD\_NO

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

prj1 1 5 1

prj2 2 6 2

prj3 3 10 3

prj4 4 13 4

prj5 5 5 5

**5 rows selected.**

**Practical No : 8**

**Aim :**Create a XML data base and demonstrate insert, update and delete operations on these tables. Issue queries on it.

**Query:**

**Creating Employee table:**

**SQL>**CREATE TABLE employee (Dept\_idnumber(5),emp\_specification XMLTYPE);

**Table created.**

**Inserting data for XML:**

**SQL>**insert into employee values

(1,XMLTYPE('<emp>

<e\_id>1</e\_id>

<ename>Mahesh</ename>

<email>Mahesh@yahoo.com</email>

<acc\_no>101</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 jan 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(2,XMLTYPE('<emp>

<e\_id>2</e\_id>

<ename>Naveen</ename>

<email>Naveen@yahoo.com</email>

<acc\_no>102</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 feb 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(3,XMLTYPE('<emp>

<e\_id>3</e\_id>

<ename>Pornima</ename>

<email>pornima@yahoo.com</email>

<acc\_no>103</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 mar 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(4,XMLTYPE('<emp>

<e\_id>4</e\_id>

<ename>shreya</ename>

<email>shreya@yahoo.com</email>

<acc\_no>104</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 april 2011</doj>

</emp>'));

**1 row created .**

**SQL>**insert into employee values

(5,XMLTYPE('<emp>

<e\_id>5</e\_id>

<ename>ketal</ename>

<email>ketal@yahoo.com</email>

<acc\_no>105</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 may 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(6,XMLTYPE('<emp>

<e\_id>6</e\_id>

<ename>aakanksha</ename>

<email>aakanksha@yahoo.com</email>

<acc\_no>106</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 june 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(7,XMLTYPE('<emp>

<e\_id>7</e\_id>

<ename>aakash</ename>

<email>aakash@yahoo.com</email>

<acc\_no>107</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 july 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(8,XMLTYPE('<emp>

<e\_id>8</e\_id>

<ename>nishit</ename>

<email>nishit@yahoo.com</email>

<acc\_no>108</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 aug 2011</doj></emp>'));

insert into employee values

(9,XMLTYPE('<emp>

<e\_id>9</e\_id>

<ename>swapnil</ename>

<email>swapnil@yahoo.com</email>

<acc\_no>109</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 sept 2011</doj>

</emp>'));

**1 row created.**

**SQL>**insert into employee values

(10,XMLTYPE('<emp>

<e\_id>10</e\_id>

<ename>anthony</ename>

<acc\_no>110</acc\_no>

<email>anthony@yahoo.com</email>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 oct 2011</doj>

</emp>'));

**1 row created.**

**QUERIES:**

**1) Retrieve the names of employee:**

**SQL>**Select e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal() from employee e;

E.EMP\_SPECIFICATION.EXTRACT('/EMP/ENAME/TEXT()').GETSTRINGVAL()

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

Mahesh

Pornima

shreya

ketal

aakanksha

aakash

nishit

swapnil

anthony

Naveen

**2) Retrieve the acc\_no of employees:**

**SQL>**Select e.emp\_specification.EXTRACT('/emp/acc\_no/text()').getStringVal() from employee e;

E.EMP\_SPECIFICATION.EXTRACT('/EMP/ACC\_NO/TEXT()').GETSTRINGVAL()

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

101

102

103

104

105

106

107

108

109

110

**3) Retrieve the names, acc\_no, and email of employees:**

**SQL>**Select e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal() "Name", e.emp\_specification.EXTRACT('/emp/acc\_no/text()').getStringVal() "Account\_no", e.emp\_specification.EXTRACT('/emp/email/text()').getStringVal() "Email" from employee e;

Name Account\_no Email

Mahesh 101 Mahesh@yahoo.com

Naveen 102 Naveen@yahoo.com

Pornima 103 pornima@yahoo.com

shreya 104 shreya@yahoo.com

ketal 105 ketal@yahoo.com

aakanksha 106 aakanksha@yahoo.com

aakash 107 aakash@yahoo.com

nishit 108 nishit@yahoo.com

swapnil 109 swapnil@yahoo.com

Anthony 110 anthony@yahoo.com

**4) Update the 4th record from the table and display the name of an employee.**

**SQL>**Update employee e set

e.emp\_specification=XMLTYPE('<emp>

<e\_id>4</e\_id>

<ename>shree</ename>

<email>shreya@yahoo.com</email>

<acc\_no>104</acc\_no>

<mngr\_email>aditi@yahoo.com</mngr\_email>

<doj>22 april 2011</doj>

</emp>')

where e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal()='shreya';

**1 row updated.**

**Before updation :-**

**SQL>**Select e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal() from

employee where e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal()='shreya';

Name

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

shreya

**After updation :-**

**SQL>**Select e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal() “Name”

from employee e

where e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal()='shree';

Name

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

Shree

**5) Delete 10th record from the table:**

**SQL>**delete from employee e

where e.emp\_specification.EXTRACT('/emp/ename/text()').getStringVal()='anthony';

**1 row deleted.**

**SQL> select \* from employee;**

DEPT\_ID

----------

EMP\_SPECIFICATION

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

1

<emp>

<e\_id>1</e\_id>

<ename>Mahesh</ename>

<email>[Mahesh@yahoo.com</email](mailto:priya@yahoo.com%3c/email)>

2

<emp>

<e\_id>2</e\_id>

<ename>Naveen</ename>

DEPT\_ID

----------

EMP\_SPECIFICATION

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

<email>Naveen@yahoo.com</email>

3

<emp>

DEPT\_ID

----------

EMP\_SPECIFICATION

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

<e\_id>3</e\_id>

<ename>Pornima</ename>

<email>pornima@yahoo.com</emai

4

<emp>

<e\_id>4</e\_id>

<ename>shree</ename>

<email>shreya@yahoo.com</email>

DEPT\_ID

----------

EMP\_SPECIFICATION

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

5

<emp>

<e\_id>5</e\_id>

<ename>ketal</ename>

<email>ketal@yahoo.com</email>

6

DEPT\_ID

----------

EMP\_SPECIFICATION

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

<emp>

<e\_id>6</e\_id>

<ename>aakanksha</ename>

<email>aakanksha@yahoo.com</

7

<emp>

<e\_id>7</e\_id>

<ename>aakash</ename>

DEPT\_ID

----------

EMP\_SPECIFICATION

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

<email>aakash@yahoo.com</email>

8

<emp>

<e\_id>8</e\_id>

<ename>nishit</ename>

<email>nishit@yahoo.com</email>

9

DEPT\_ID

----------

EMP\_SPECIFICATION

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

<emp>

<e\_id>9</e\_id>

<ename>swapnil</ename>

<email>swapnil@yahoo.com</emai

9 rows selected.