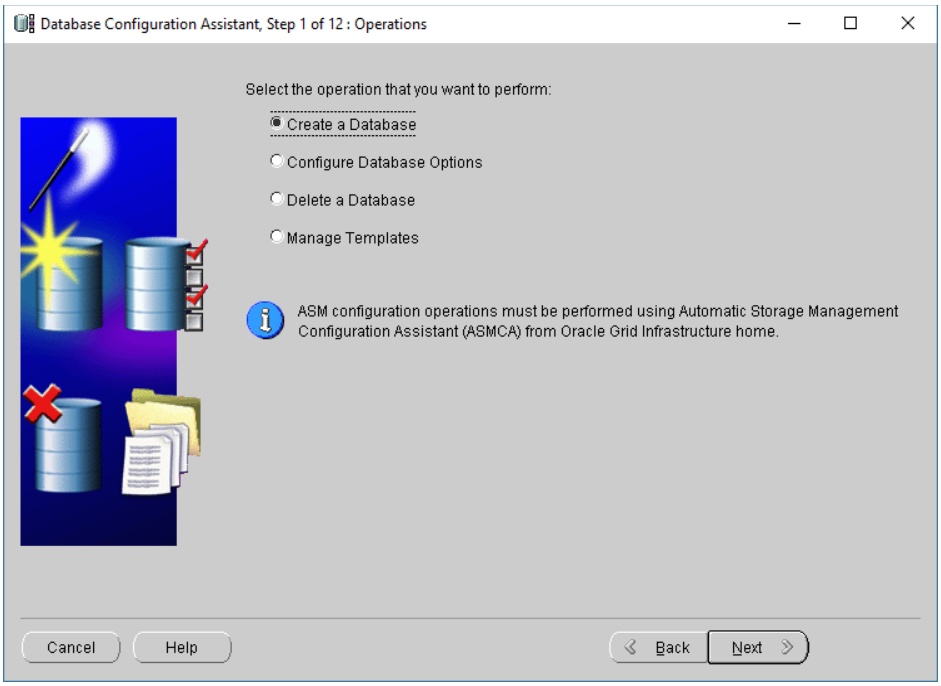
PRACTICAL NO: 1

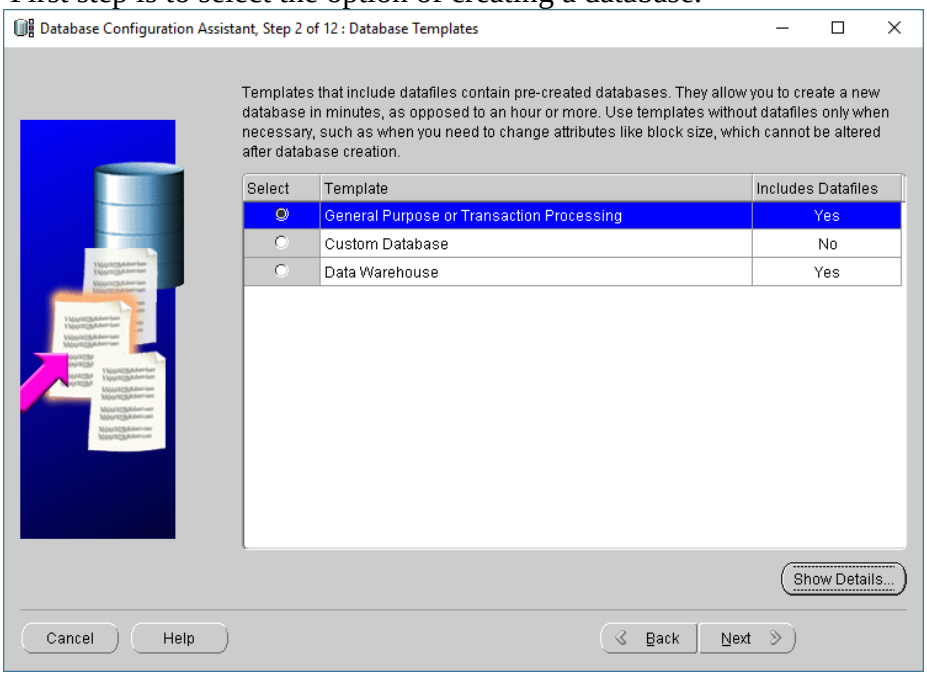
Aim: For a given a global conceptual schema, divide the schema into vertical fragments and place them on different nodes. Execute queries on these fragments that will demonstrate distributed databases environment.

Step1: Go to->start->all programs->oracle-oraHome92->Configuration and migration tools ->Database Configuration Assistant

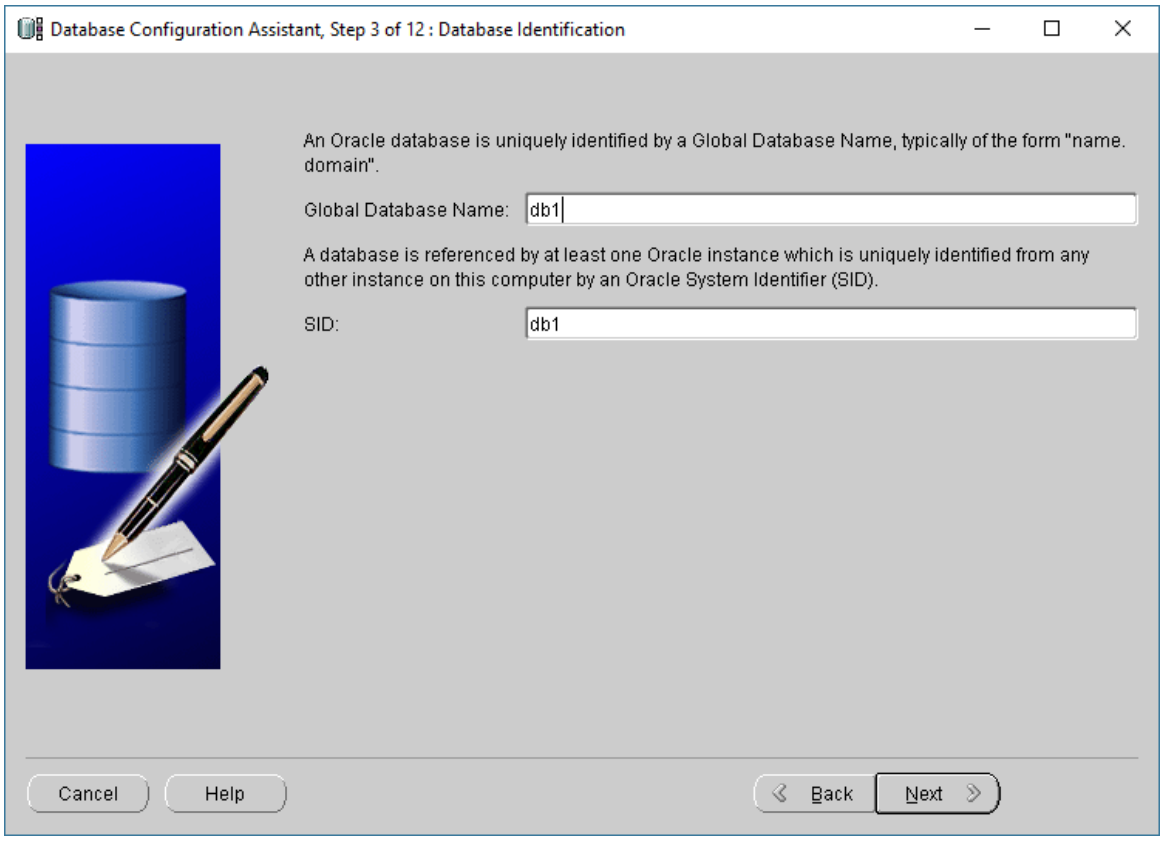
Step 2: Launch the launcher file from the menu, which look like below figure.



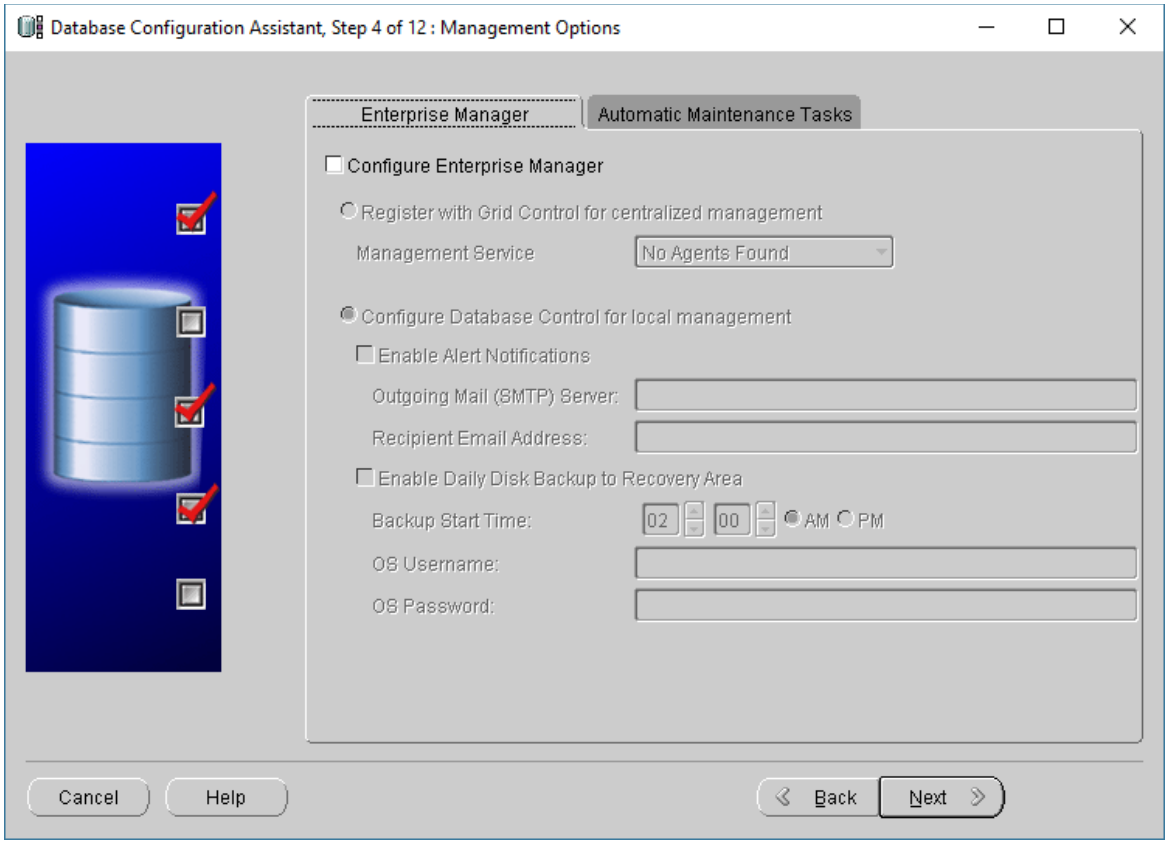
Step 3: First step is to select the option of creating a database.



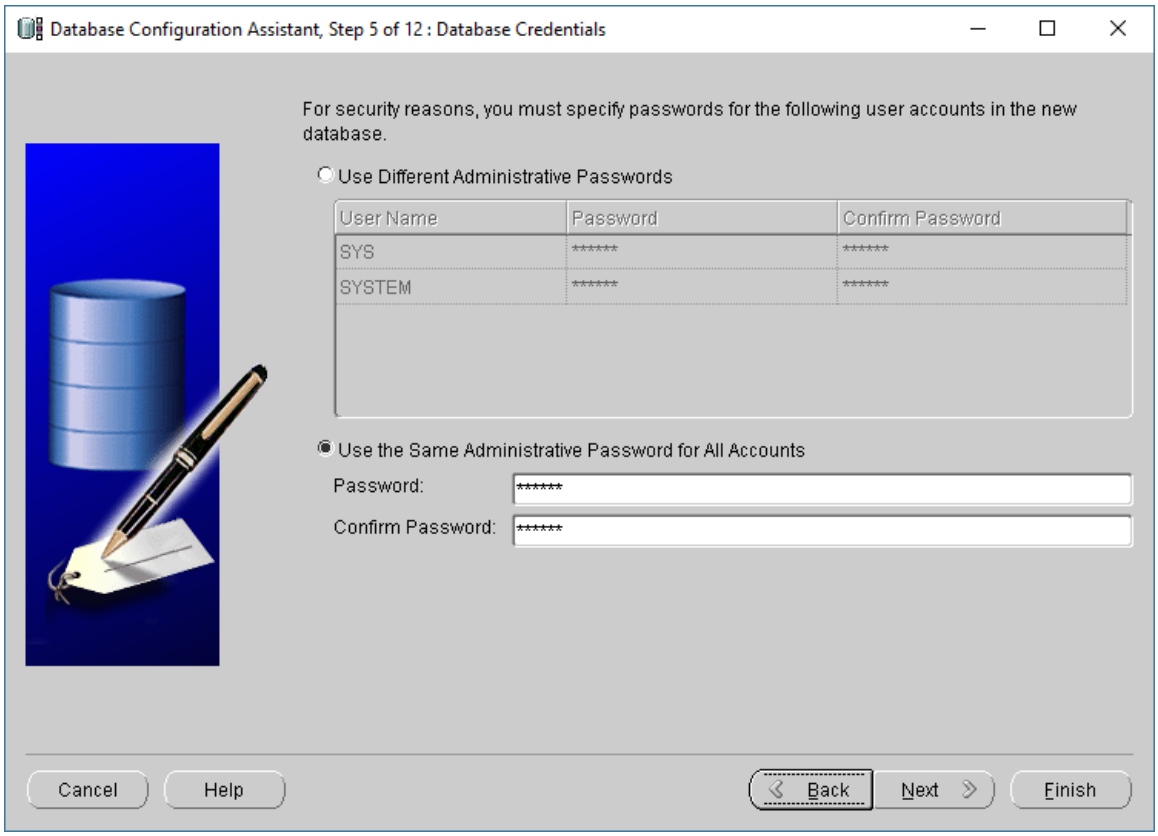
Step 4: Second step is to select the general purpose database option.



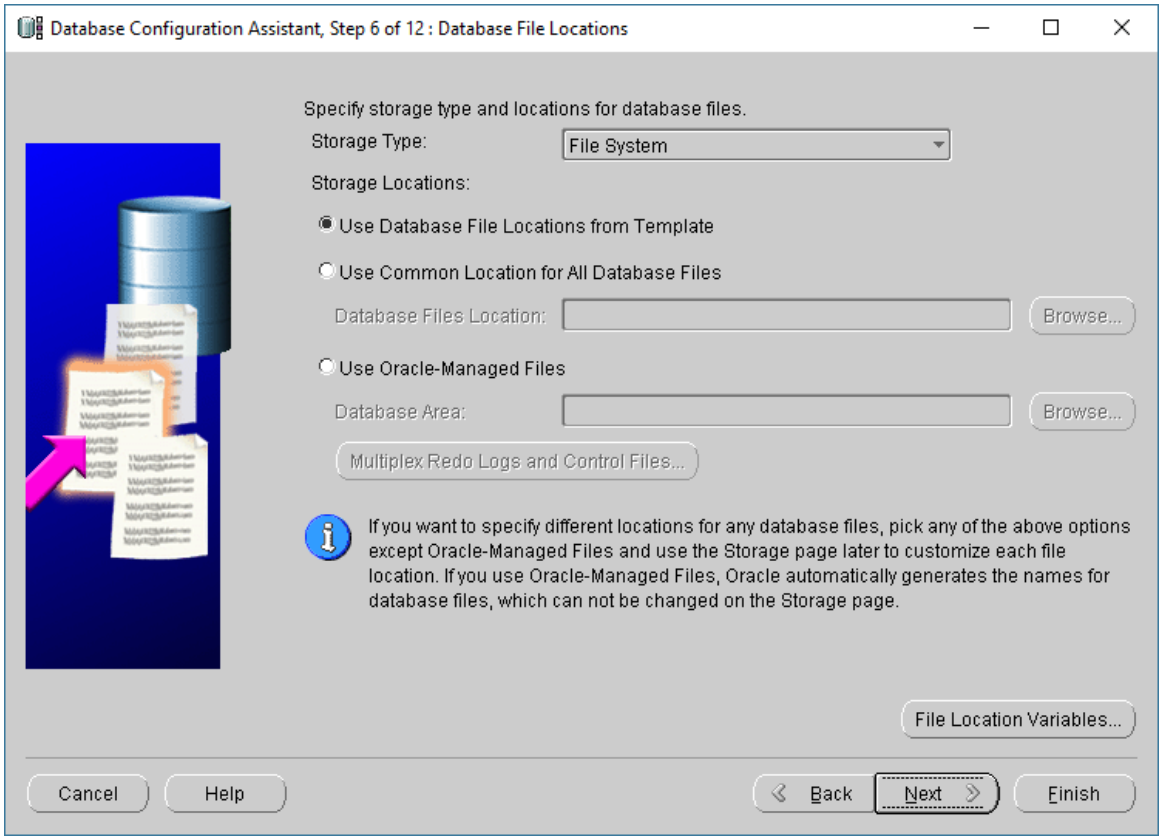
Step 5: Select the dedicated Management Options depends on your choice.



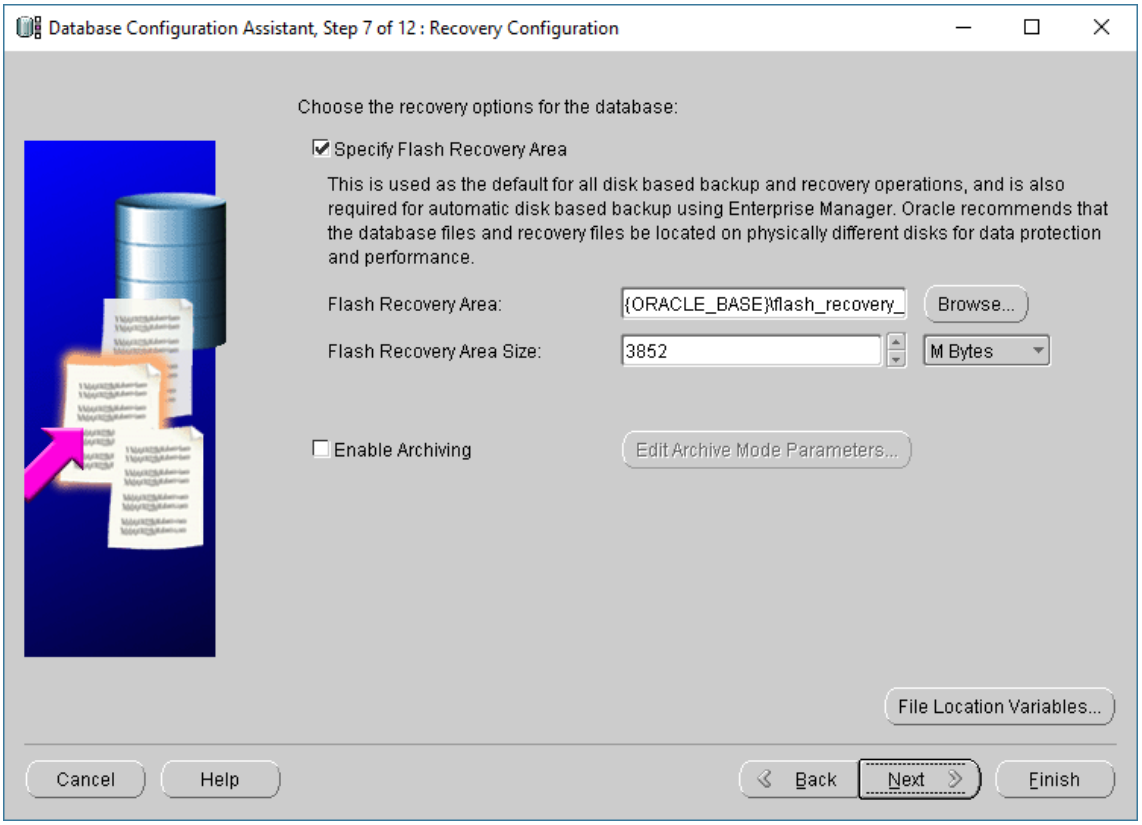
Step 6: Assign the same password to all accounts by selecting option.



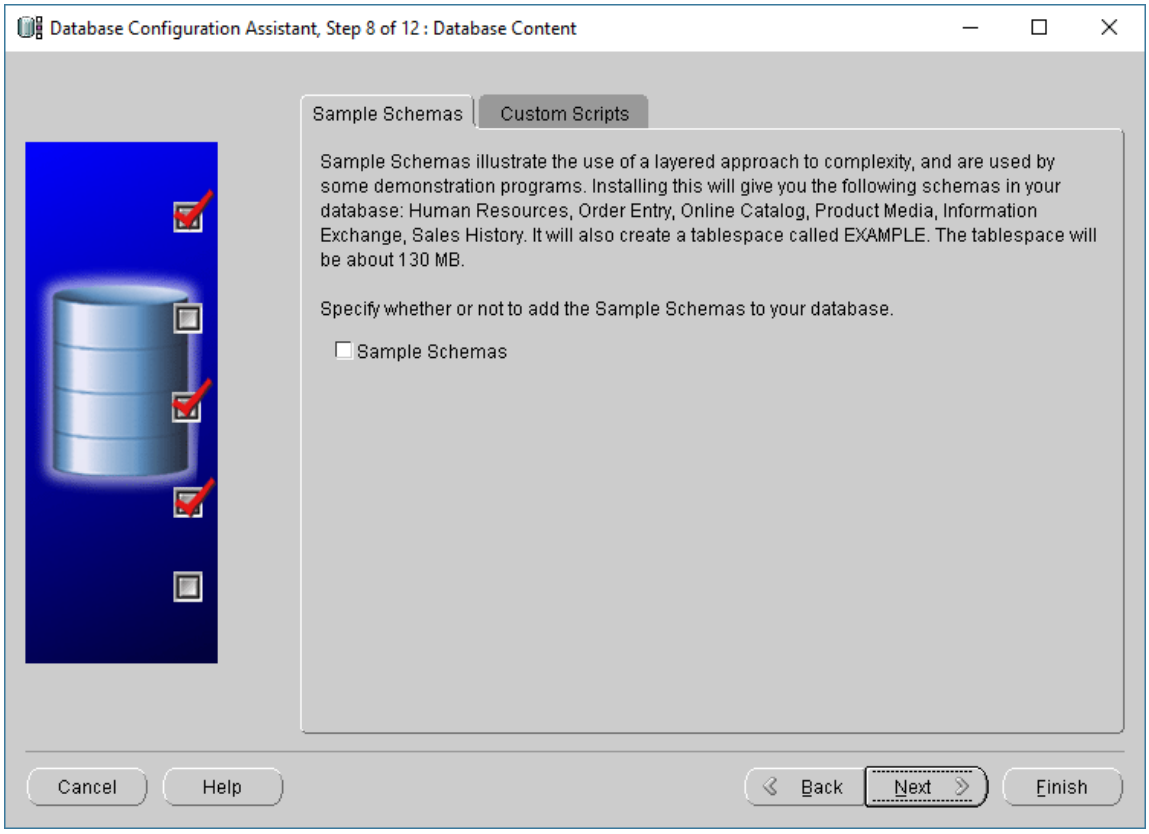
Step 7: Specify storage type and locations for database files.



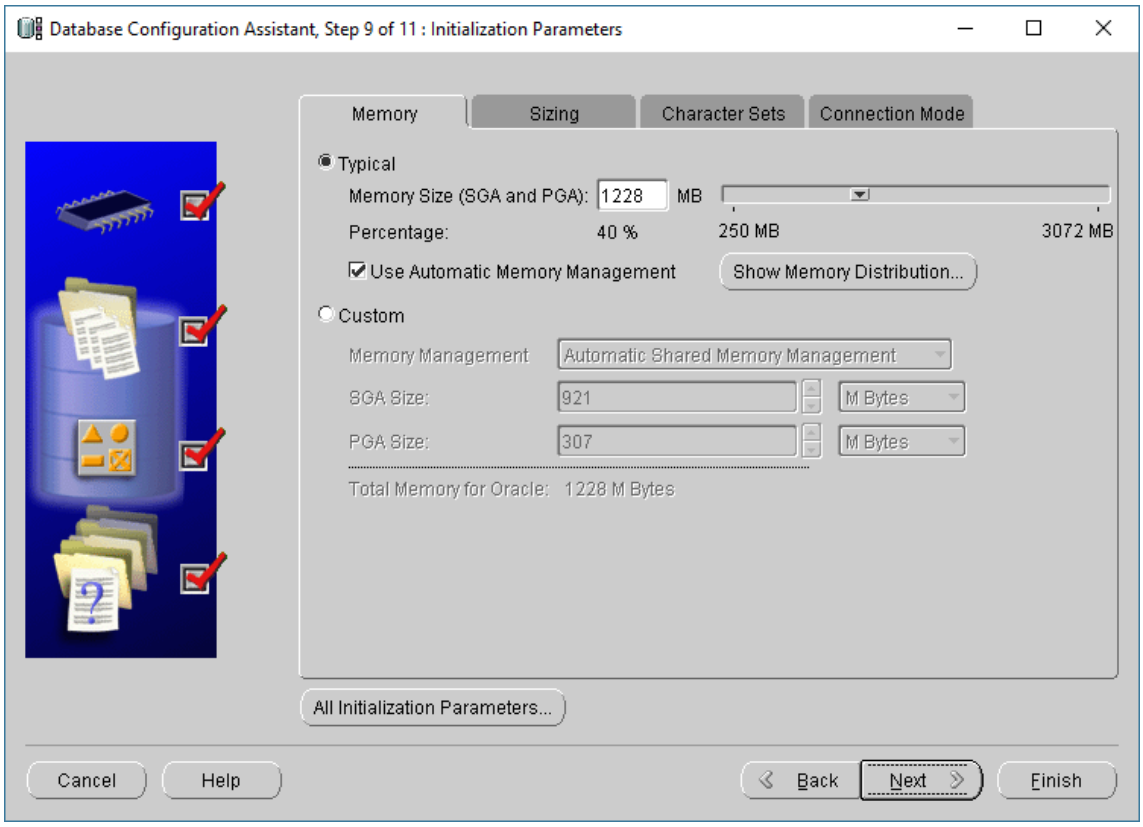
Step 8: Choose the recovery options for the database.



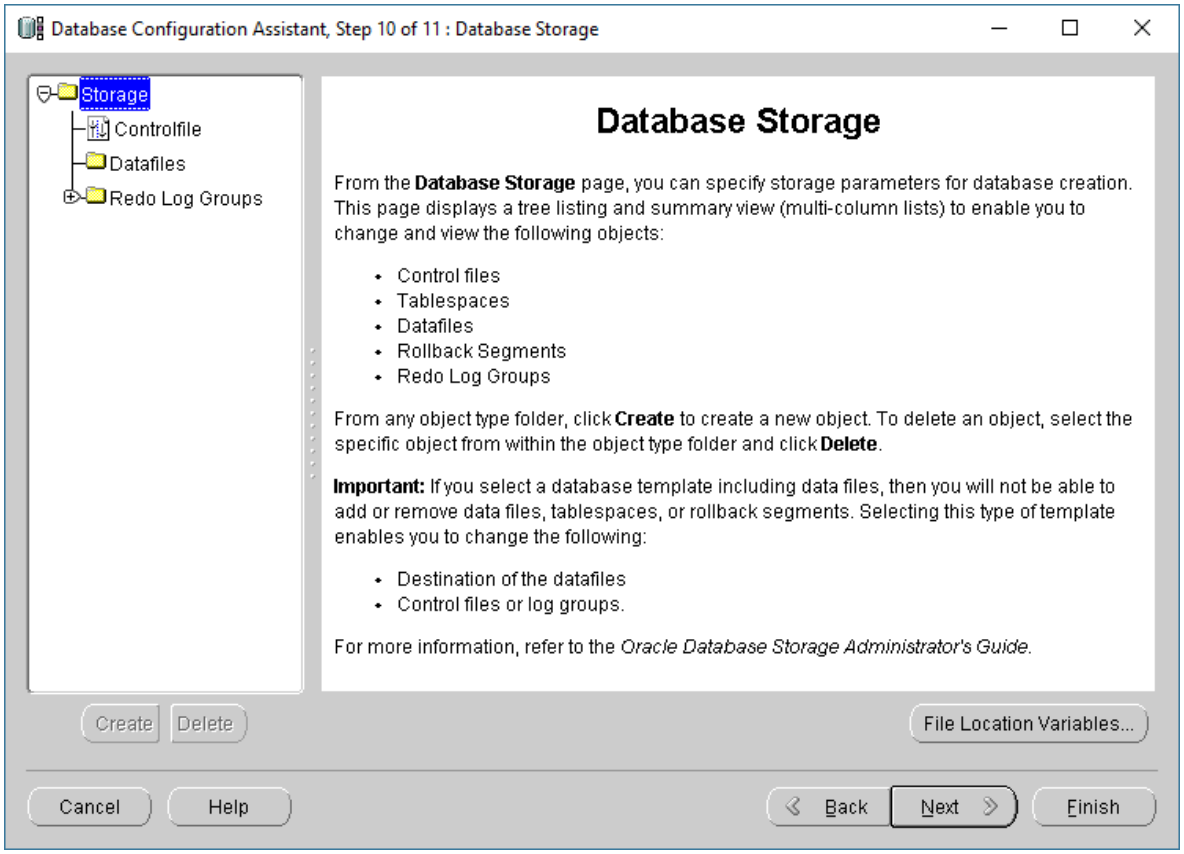
Step 9: Click on Next.



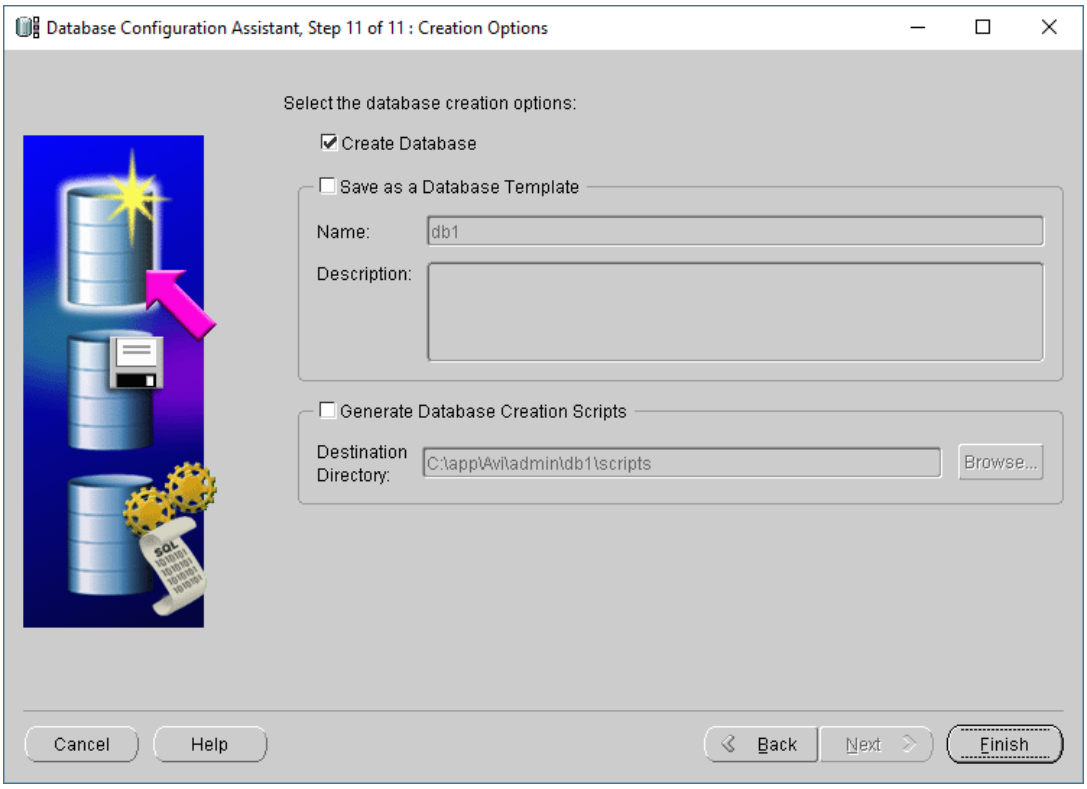
Step 10: Memory allocation in Initialization Parameters, Click on the next.



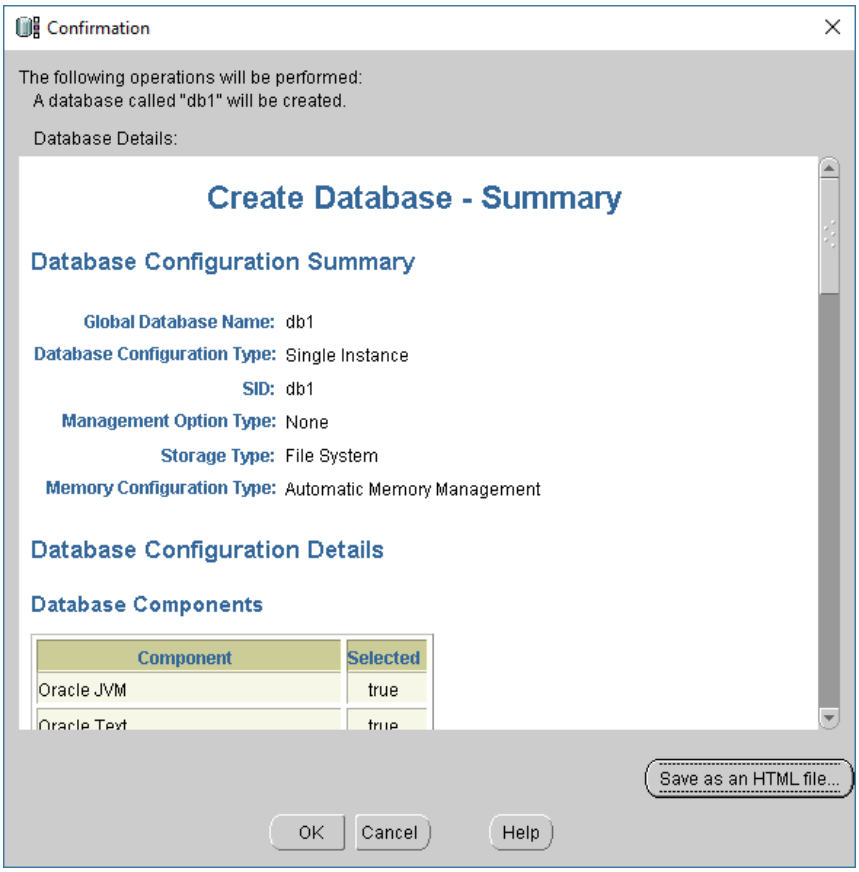
Step 11: Click on Next.



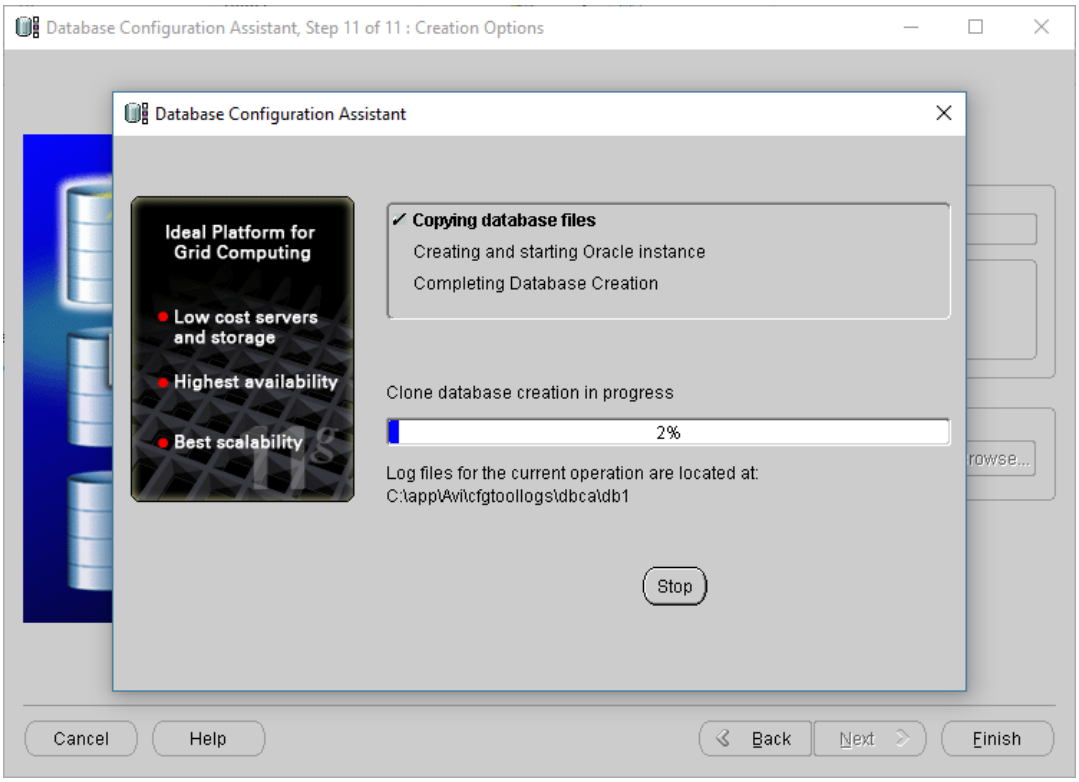
Step 12: Click on Finish.



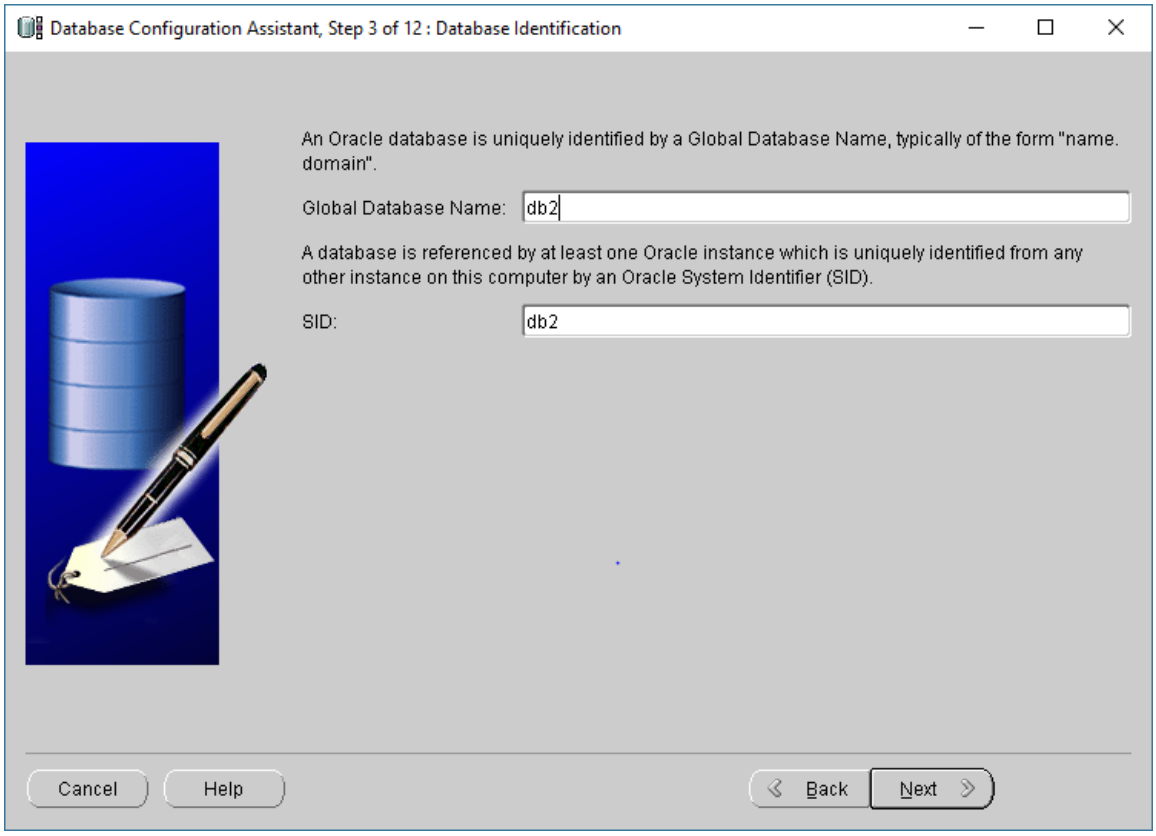
Step 13: Click on ok, to start the creation of database .



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

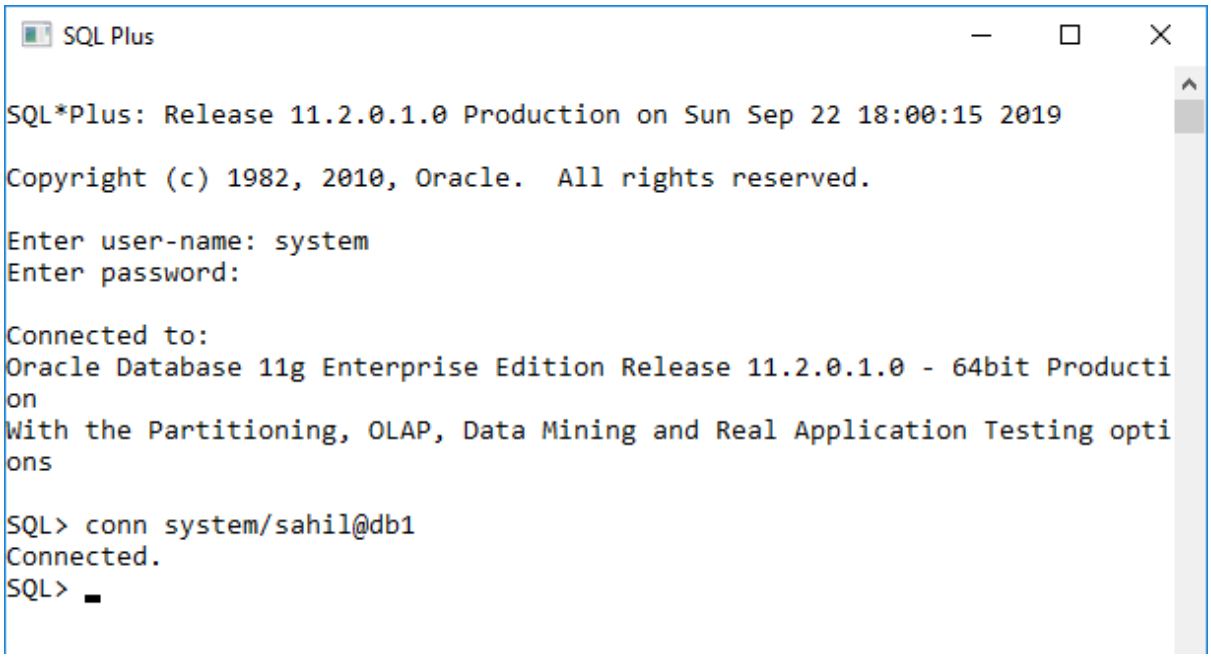


Step 15: Similarly, create another database named as db2.



Step 16: Open SQL Plus.

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



Step 18: Go to db1 database.
SQL> conn system/sahil@db1
Connected.

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

SQL> create table employee (eno int Primary key, ename varchar(20), address varchar(30), email varchar(20), salary int);
Table created.
SQL> insert into employee values(1,'Sahil','Goregaon','sahil@gmail.com',20000);
1 row created.
SQL> insert into employee values(2,'Ankit','Malad','ankit@gmail.com',15000);
1 row created.
SQL> insert into employee values(3,'Rahul','Virar','rahul@gmail.com',18000);
1 row created.
SQL> insert into employee values(4,'Vishal','Vasai','vishal@gmail.com',10000);
1 row created.
SQL> insert into employee values(5,'Rupesh','Parel','rupesh@gmail.com',9000);
1 row created.

SQL> select * from employee;

ENO	ENAME	ADDRESS	EMAIL	SALARY
1	Sahil	Goregaon	sahil@gmail.com	20000
2	Ankit	Malad	ankit@gmail.com	15000
3	Rahul	Virar	rahul@gmail.com	18000
4	Vishal	Vasai	vishal@gmail.com	10000
5	Rupesh	Parel	rupesh@gmail.com	9000

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

SQL> create database link db1todb2 connect to system identified by sahil using 'db2';
Database link created.
SQL> create database link db2todb1 connect to system identified by sahil using 'db1';
Database link created.

Step 21: Create 4 fragmentations
SQL> create table emp1 as select eno,ename,salary from employee@db2todb1;
Table created

SQL> select *from emp1;

ENO	ENAME	SALARY
1	Sahil	20000
2	Ankit	15000
3	Rahul	18000
4	Vishal	10000
5	Rupesh	9000

SQL> create table emp11 as select eno,ename,email from employee@db2todb1;
Table created.

SQL> select *from emp1;

ENO	ENAME	EMAIL
1	Sahil	sahil@gmail.com
2	Ankit	ankit@gmail.com
3	Rahul	rahul@gmail.com
4	Vishal	vishal@gmail.com
5	Rupesh	rupesh@gmail.com

SQL> create table emp2 as select eno,ename,address from employee@db2todb1;
Table created.

SQL> select *from emp2;

ENO	ENAME	ADDRESS
1	Sahil	Goregaon
2	Ankit	Malad
3	Rahul	Virar
4	Vishal	Vasai
5	Rupesh	Parel

SQL> create table emp22 as select eno,ename,email,salary from employee@db2todb1;
Table created.

SQL> select *from emp22;

ENO	ENAME	EMAIL	Salary
1	Sahil	sahil@gmail.com	20000
2	Ankit	ankit@gmail.com	15000
3	Rahul	rahul@gmail.com	18000
4	Vishal	vishal@gmail.com	10000
5	Rupesh	rupesh@gmail.com	9000

Step 22: Run Following Commands.

Output :-

1) Find the salary of all employee

SQL> select salary from emp1;

SALARY
20000
15000
18000
10000
9000

2) Find email of all employee where salary is greater than 10000

SQL> select email from emp4 where salary>10000;

EMAIL
---
sahil@gmail.com
ankit@gmail.com
rahul@gmail.com

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

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

```
ENAME                EMAIL
-------------------- ------------------------------
 Rahul               rahul@gmail.com
```

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

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

```
ENAME                ADDRESS
------------------------ ------------------------------
 Rahul                   Virar
```

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::Go to db1 database.
SQL> conn system/sahil@db1
Connected.

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

SQL> create table employee (eno int Primary key, ename varchar(20), address varchar(30), email varchar(20), salary int);
Table created.
SQL> insert into employee values(1,'Sahil','Goregaon','sahil@gmail.com',20000);
1 row created.
SQL> insert into employee values(2,'Ankit','Malad','ankit@gmail.com',15000);
1 row created.
SQL> insert into employee values(3,'Rahul','Virar','rahul@gmail.com',18000);
1 row created.
SQL> insert into employee values(4,'Vishal','Vasai','vishal@gmail.com',10000);
1 row created.
SQL> insert into employee values(5,'Rupesh','Parel','rupesh@gmail.com',9000);
1 row created.

SQL> select *from employee;

ENO	ENAME	ADDRESS	EMAIL	SALARY
1	Sahil	Goregaon	sahil@gmail.com	20000
2	Ankit	Malad	ankit@gmail.com	15000
3	Rahul	Virar	rahul@gmail.com	18000
4	Vishal	Vasai	vishal@gmail.com	10000
5	Rupesh	Parel	rupesh@gmail.com	9000

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

SQL> create database link db1todb2 connect to system identified by sahil using 'db2';
Database link created.

SQL> create database link db2todb1 connect to system identified by sahil 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;
ENO         ENAME          ADDRESS              EMAIL              SALARY
------------ --------------------- ----------------------- --------------------------- ----------------
         5      Rupesh              Parel                  rupesh@gmail.com            9000
```

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

```
SQL> create table emp4 as select * from employee@db2todb1 where address='Goregaon';
Table created.

SQL> select * from emp4;
ENO         ENAME            ADDRESS              EMAIL              SALARY
--------------- --------------------- ----------------------- --------------------------- -----------------
         1         Sahil               Goregaon               sahil@gmail.com            20000
```

```
SQL> Conn system/sahil@db2;
Connected.

SQL> select salary from employee@db2todb1;

     SALARY
-------------------
      20000
      15000
      18000
      10000
       9000

SQL> select email,salary from employee@db2todb1 where salary<=11000;

 EMAIL                SALARY
----------------------------- ---------------------
vishal@gmail.com          10000
rupesh@gmail.com           9000

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

      ENAME             EMAIL
----------------------- ---------------------------
        Ankit         ankit@gmail.com
```

Step 8: Creating table in ‘db1’ using table of ‘’db2’ using ‘db1todb2’ link

SQL> conn system/sahil@db1
Connected.

SQL> Create table emp3 as select * from emp3@db1todb2 where address='Parel';
Table created.

SQL> select * from emp3;

ENO	ENAME	ADDRESS	EMAIL	SALARY
5	Rupesh	Parel	rupesh@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 DB1

```
SQL>create table emp1 (enumber number primary key, ename varchar2(10), addr
varchar2(15), email varchar2(20),salary float);
Table Created
```

In DB3

```
SQL> create table emp1 (enumber number primary key, ename varchar2(10), addr
varchar2(15), email varchar2(20),salary float);
Table Created
```

Creating Link :-

```
SQL>connect system/sahil@db1
Connected

SQL>create database link db1todb3 connect to system identified by sahil using 'db3';
Database link created
```

Creating Triggers:-

```
SQL>create or replace trigger insert_data1
after insert on emp1
for each row
begin
insert into emp1@db1todb3
values(:new.enumber,:new.ename,:new.addr,:new.email,:new.salary);
end;
/
Tigger created

SQL>create or replace trigger insert_data1
before delete on emp1
for each row
begin
delete from emp1@db1todb3 where enumber=:old.enumber;
end;
/
Tigger created
```

```
SQL>create or replace trigger insert_data1
After update on emp1
for each row
begin
update emp1@db1todb3 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 emp1 values(1,'Sahil','Goregaon','sahil@gmail.com',20000);
1 row created.
SQL> insert into emp1 values(2,'Rahul','RamMandir','rahul@gmail.com',18000);
1 row created.
SQL> insert into emp1 values(3,'Ankit','Malad','ankit@gmail.com',15000);
1 row created.
SQL> insert into emp1 values(4,'Nilesh','Virar','nilesh@gmail.com',10000);
1 row created.
SQL> insert into emp1 values(5,'Rupesh','Aarey','rupesh@gmail.com',9000);
1 row created.
SQL>connect system/sahil@db3
Connected

SQL> select *from emp1;
```

NUMBER	ENAME	ADDR	EMAIL	SALARY
1	Sahil	Goregaon	sahil@gmail.com	20000
2	Rahul	RamMandir	rahul@gmail.com	18000
3	Ankit	Malad	ankit@gmail.com	15000
4	Nilesh	Virar	nilesh@gmail.com	10000
5	Rupesh	Aarey	rupesh@gmail.com	9000

Updating Values:-

SQL>connect system/sahil@db1
Connected

SQL> update employee set enumber = 3, ename='Yogesh', addr='Malad', email='yogesh@gmail.com', salary=18000 where enumber = 3;
1 row updated.

SQL>connect system/sahil@db3
Connected

SQL> select * from emp1;

OUTPUT:

ENUMBER	ENAME	ADDR	EMAIL	SALARY
1	Sahil	Goregaon	sahil@gmail.com	20000
2	Rahul	RamMandir	rahul@gmail.com	18000
3	Yogesh	Malad	yogesh@gmail.com	18000
4	Nilesh	Virar	nilesh@gmail.com	10000
5	Rupesh	Aarey	rupesh@gmail.com	9000

SQL>connect system/mahesh@db1
Connected

SQL>delete from emp1 where enumber=5;
1 row deleted.

SQL>connect system/sahil@db3
Connected

SQL> select *from employee;

OUTPUT:

ENUMBER	ENAME	ADDR	EMAIL	SALARY
1	Sahil	Goregaon	sahil@gmail.com	20000
2	Rahul	RamMandir	rahul@gmail.com	18000
3	Yogesh	Malad	yogesh@gmail.com	18000
4	Nilesh	Virar	nilesh@gmail.com	10000

Query :-

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

SQL>select enumber, ename, eesalary from emp1;

```
ENUMBER ENAME  SALARY
---------- ---------- ----------
     1      Sahil      20000
     2      Rahul      18000
     3      Yogesh     18000
     4      Nilesh     10000
```

2) **Find the email of all employees where salary =18000.**

SQL> select email from employee where salary=18000;

```
EMAIL
------------------------------
rahul@gmail.com
yogesh@gmail.com
```

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

SQL> select ename, email from emp1 where enumber=1;

```
ENAME                EMAIL
---------- ------------------------------
Sahil                sahil@gmail.com
```

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

SQL> select ename, address from emp1 where enumber=1;

```
ENAME                ADDRESS
---------- ------------------------------
Sahil                Goregaon
```

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('Sahil', AddrType1(7000,'AT street', 'mumbai', 'maharashtra' ,1007));

1 row created.

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

1 row created.

SQL> insert into Authors values('Rahul',AddrType1(7003,'PL street','mumbai','maharashtra' ,1003));
1 row created.

SQL> insert into Authors values('Nilesh',AddrType1(7008,'AT street', 'mumbai', 'maharashtra' ,1007));
1 row created.

SQL> insert into Authors values ('Prathmesh',AddrType1 (7006,'Nehrut','mumbai','maharashtra' ,1005));
1 row created.

SQL> insert into Authors values ('Abhay', AddrType1(8002,'TH street','pune', 'maharashtra' ,13));
1 row created.

SQL> insert into Authors values('Rupesh',AddrType1(7008,'TT street', 'Nasik','maharashtra' ,1008));
1 row created.

SQL> insert into Authors values('Hrishikesh',AddrType1(7002,'FL street','pune', 'maharashtra' ,03));
1 row created.

SQL> insert into Publishers values ('Shivaji', 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', 'maharashtra',27), BranchTableType (BranchType (AddrType1(1002,'DM street','nasik', 'maharashtra',1007), 456767,7675757)));

1 row created.

SQL> insert into Publishers values ('Manish', 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,Authors aut where pub.name='Tata' and aut.name='Hrishikesh';

2 rows created.

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

1 row created.

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

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

Ankit
Nilesh
Rupesh
Hrishikesh

2) **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

3) 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

Hrishikesh

4) 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

Hrishikesh
Ankit
Abhay

5) 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

IP

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,'Sahil','12-feb-2005','12-oct-2011') ;
1 row created.
SQL>insert into Emp_Appnt values(2211,'Rahul','16-march-2008','16-sep-2010') ;
1 row created.
SQL>insert into Emp_Appnt values(2221,'Ankit’,'18-june-2004','18-july-2006') ;
1 row created.
SQL>insert into Emp_Appnt values(2221,'Nilesh','18-june-2004','21-july-2008') ;
1 row created.
SQL> insert into emp_appnt values(2000,'Rupesh','16-oct-2003','16-sep-2010');
1 row created.

SQL>select * from emp_appnt;

ACC_NO	NAME	RECDATE	RETDATE
2025	Sahil	12-FEB-05	12-OCT-11
2211	Rahul	16-MAR-08	16-SEP-10
2221	Ankit	18-JUN-04	18-JUL-06
2221	Nilesh	18-JUN-04	21-JUL-08
2000	Rupesh	16-OCT-03	16-SEP-10

Queries:

SQL>select * from emp_appnt where RECDate='18-june-2004';

ACC_NO	NAME	RECDATE	RETDATE
2221	Ankit	18-JUN-04	18-JUL-06

SQL>select * from emp_appnt where RETDate='16-sep-2010';

ACC_NO	NAME	RECDATE	RETDATE
2211	Rahul	16-MAR-08	16-SEP-10
2000	Rupesh	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('Sahil',123,500) ;
1 row created.
SQL>insert into tbl_shares(C_Name,No_Share,Price) values(‘Rahul’,121,810,) ;
1 row created.
SQL>insert into tbl_shares(C_Name,No_Share,Price) values(‘Ankit’,233,600) ;
1 row created.
SQL>insert into tbl_shares(C_Name,No_Share,Price) values(‘Nilesh’,203,650) ;
1 row created.
SQL> insert into tbl_shares(C_Name,No_Share,Price) values(‘Rupesh’,212,880);
1 row created.

SQL>select * from tbl_shares;

C_NAME	NO_SHARE	PRICE	TRANSTIME
Sahil	123	500	02:03
Rahul	121	810	02:04
Ankit	233	600	02:05
Nilesh	203	650	02:06
Rupesh	212	880	02:06

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

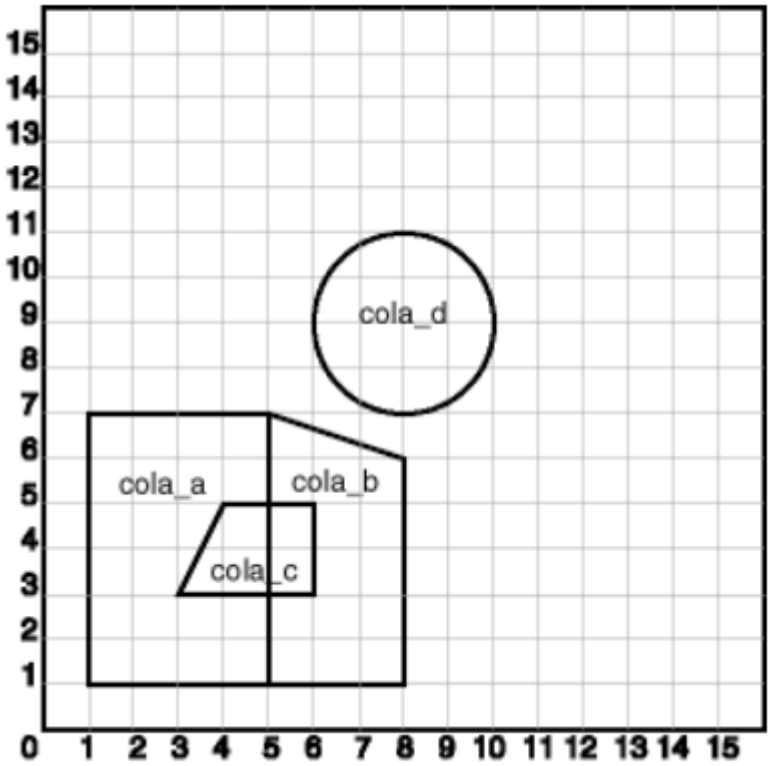
C_NAME	NO_SHARE	PRICE	TRANSTIME
Ankit	233	600	02:05

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

C_NAME	NO_SHARE	PRICE	TRANSTIME
Rahul	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_mrp values(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_mrp values(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_mrp values(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:

```
SQL>insert into user_SDO_GEOM_METADATA values('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 :

```
SQL>create index cola_spatial_idx on cola_market(location) Indextype Is mdsys.spatial_index;
```

<u>Queries:</u>

1) Find the topological intersection of two geometries.

```
SQL>select SDO_GEOM.SDO_INTERSECTION (c_a.shape,c_c.shape,0.005)
from cola_mrp c_a,cola_mrp c_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.

SQL>SELECT SDO_GEOM.RELATE(c_c.shape, 'EQUAL', c_a.shape,0.005)
FROM cola_mrp c_c, cola_mrp c_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

SQL>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.

SQL>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.

SQL>select SDO_GEOM.SDO_DISTANCE(c_b.shape,c_d.shape,0.005)
from cola_mrp c_b,cola_mrp c_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), pno number(5) primary key, thrs number(5),
head_no number(5));
Table created.

SQL>create table Employee1 (eno number(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 :

a) **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.

b) Inserting into Employee1:-

SQL> insert into Employee1 values(0001,'Ankit',5,2,001);
1 row created.
SQL> insert into Employee1 values(0002,'Sahil',4,3,002);
1 row created.
SQL>insert into Employee1 values(0003,'Rahul',6,4,004);
1 row created.
SQL>insert into Employee1 values(0004,'Prathmesh',6,2,002) ;
1 row created.
SQL>insert into Employee1 values(0005,'Nilesh',5,3,005);
1 row created.

2) 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 Ankit              5          2          1
         2 Sahil              4          3          2
         3 Rahul              6          4          4
         4 Prathmesh          6          2          2
         5 Nilesh             5          3          5
```

5 rows selected.

SQL> select * from Project1;

```
PNAME           PNO       THRS    HEAD_NO
---------- ---------- ---------- ----------
prj1             1        10         1
prj2             2        20         2
prj3             3        10         3
prj4             4        14         4
prj5             5        10         5
```

5 rows selected.

<u>After trigger</u> :-

```
SQL>update Employee1 set hrs=2 where eno=2;
1 row updated.

SQL> select * from Employee1;

ENO   ENAME   HRS   SUPER_NO      PNO
---------- ---------- ---------- ---------- ----------
         1 Ankit            5          2          1
         2 Sahil            2          3          2
         3 Rahul            6          4          4
         4 Prathmesh        6          2          2
         5 Nilesh           5          3          5
5 rows selected.

SQL> select * from Project1;

PNAME           PNO       THRS    HEAD_NO
---------- ---------- ---------- ----------
prj1             1        10         1
prj2             2        18         2
prj3             3        10         3
prj4             4        14         4
prj5             5        10         5
5 rows selected.
```

3) 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 Ankit            5          2          1
         2 Sahil            2          3          2
         3 Rahul            6          4          4
         4 Prathmesh        6          2          2
         5 Nilesh           5          3          5
```

5 rows selected.

SQL>select * from Project1;

```
PNAME           PNO       THRS    HEAD_NO
---------- ---------- ---------- ----------
prj1             1        10         1
prj2             2        18         2
prj3             3        10         3
prj4             4        14         4
prj5             5        10         5
```

5 rows selected.

After Trigger:-

SQL>Update Employee1 Set pno=2 where eno=3;

1 row updated.

SQL>select * from Employee1;

```
ENO   ENAME   HRS   SUPER_NO     PNO
---------- ---------- ---------- ---------- ----------
         1 Ankit            5          2          1
         2 Sahil            2          3          2
         3 Rahul            6          4          2
         4 Prathmesh        6          2          2
         5 Nilesh           5          3          5
```

5 rows selected.

SQL>select * from Project1;

```
PNAME           PNO       THRS    HEAD_NO
---------- ---------- ---------- ----------
prj1             1        10         1
prj2             2        24         2
prj3             3        10         3
prj4             4         8         4
prj5             5        10         5
```

5 rows selected.

4) Creating a trigger to deleting the project of an employee.

```
SQL>create or replace trigger delemp
        after delete 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 Ankit          5         2         1
        2 Sahil          2         3         2
        3 Rahul          6         4         4
        4 Prathmesh      6         2         2
        5 Nilesh         5         3         5
5 rows selected.

SQL> select * from Project1;

PNAME          PNO      THRS    HEAD_NO
---------- ---------- ---------- ----------
prj1            1        10        1
prj2            2        24        2
prj3            3        10        3
prj4            4         8        4
prj5            5        10        5
5 rows selected.
```

After Trigger :-

SQL>delete from Employee1 where eno=1;
1 row deleted.

SQL>select * from Employee1;

ENO	ENAME	HRS	SUPER_NO	PNO
2	Sahil	2	3	2
3	Rahul	6	4	2
4	Prathmesh	6	2	2
5	Nilesh	5	3	5

4 rows selected.

SQL>select * from Project1;

PNAME	PNO	THRS	HEAD_NO
prj1	1	5	1
prj2	2	24	2
prj3	3	10	3
prj4	4	8	4
prj5	5	10	5

5 rows selected.

PARCTICAL NO: 8

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

Query:

1) Creating Employee table:

SQL>CREATE TABLE employee (Dept_idnumber(5),emp_specification XMLTYPE);
Table created.

2) Inserting data for XML:

```
SQL>insert into employee values
(1,XMLTYPE('<emp>
<e_id>1</e_id>
<ename>Sahil</ename>
<email>sahil@yahoo.com</email>
<acc_no>101</acc_no>
<mngr_email>aditya@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>Rahul</ename>
<email>rahul@yahoo.com</email>
<acc_no>102</acc_no>
<mngr_email>aditya@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>Ankit</ename>
<email>ankit@yahoo.com</email>
<acc_no>103</acc_no>
<mngr_email>aditya@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>Nilesh</ename>
<email>nilesh@yahoo.com</email>
<acc_no>104</acc_no>
<mngr_email>aditya@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>Abhay</ename>
<email>abhay@yahoo.com</email>
<acc_no>105</acc_no>
<mngr_email>aditya@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>Rupesh</ename>
<email>rupesh@yahoo.com</email>
<acc_no>106</acc_no>
<mngr_email>aditya@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>Manish</ename>
<email>manish@yahoo.com</email>
<acc_no>107</acc_no>
<mngr_email>aditya@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>Suraj</ename>
<email>suraj@yahoo.com</email>
<acc_no>108</acc_no>
<mngr_email>aditya@yahoo.com</mngr_email>
<doj>22 aug 2011</doj>
</emp>'));
```

1 row created.

```
SQL>insert into employee values
(9,XMLTYPE('<emp>
<e_id>9</e_id>
<ename>Rana</ename>
<email>rana@yahoo.com</email>
<acc_no>109</acc_no>
<mngr_email>aditya@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>Yogesh</ename>
<email>yogesh@yahoo.com</email>
<acc_no>110</acc_no>
<mngr_email>aditya@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;

OUTPUT:

```
E.EMP_SPECIFICATION.EXTRACT('/EMP/ENAME/TEXT()').GETSTRINGVAL()
--------------------------------------------------------------------------------
Sahil
Rahul
Ankit
Nilesh
Abhay
Rupesh
Manish
Suraj
Rana
Yogesh
```

2) Retrieve the acc_no of employees:

SQL>Select e.emp_specification.EXTRACT('/emp/acc_no/text()').getStringVal() from employee e;

OUTPUT:

```
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;

OUTPUT:

Name	Account_no	Email
Sahil	101	sahil@yahoo.com
Rahul	102	rahul@yahoo.com
Ankit	103	ankit@yahoo.com
Nilesh	104	nilesh@yahoo.com
Abhay	105	abhay@yahoo.com
Rupesh	106	rupesh@yahoo.com
Manish	107	manish@yahoo.com
Suraj	108	suraj@yahoo.com
Rana	109	rana@yahoo.com
Yogesh	110	yogesh@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>Aniket</ename>
<email>nilesh@yahoo.com</email>
<acc_no>104</acc_no>
<mngr_email>aditya@yahoo.com</mngr_email>
<doj>22 april 2011</doj>
</emp>')
where e.emp_specification.EXTRACT('/emp/ename/text()').getStringVal()='nilesh';

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()='nilesh';

OUTPUT:

Name
---
nilesh

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()='Aniket';

OUTPUT:
Name

Aniket

5) Delete 10th record from the table:

SQL>delete from employee e
where e.emp_specification.EXTRACT('/emp/ename/text()').getStringVal()='Yogesh';

1 row deleted.

SQL> select * from employee;

DEPT_ID

EMP_SPECIFICATION

1
<emp>
<e_id>1</e_id>
<ename>Sahil</ename>
<email>sahil@yahoo.com</email>
2
<emp>
<e_id>2</e_id>
<ename>Rahul</ename>

DEPT_ID

EMP_SPECIFICATION

<email>rahul@yahoo.com</email>
3
<emp>

DEPT_ID

EMP_SPECIFICATION

<e_id>3</e_id>
<ename>Ankit</ename>
<email>ankit@yahoo.com</emai
4

<emp>
<e_id>4</e_id>
<ename>Aniket</ename>
<email>nilesh@yahoo.com</email>

DEPT_ID

EMP_SPECIFICATION

5
<emp>
<e_id>5</e_id>
<ename>Abhay</ename>
<email>abhay@yahoo.com</email>
6
DEPT_ID

EMP_SPECIFICATION

<emp>
<e_id>6</e_id>
<ename>Rupesh</ename>
<email>rupesh@yahoo.com</
7
<emp>
<e_id>7</e_id>
<ename>Manish</ename>
DEPT_ID

EMP_SPECIFICATION

<email>manish@yahoo.com</email>
8
<emp>
<e_id>8</e_id>
<ename>Suraj</ename>
<email>suraj@yahoo.com</email>
9
DEPT_ID

EMP_SPECIFICATION

<emp>
<e_id>9</e_id>
<ename>Rana</ename>
<email>rana@yahoo.com</email>

9 rows selected.