THAKUR COLLEGE OF SCIENCE & COMMERCE

NAAC Accredited with Grade "A" (3rd Cycle)

THAKUR TRUSTS

ISO 9001 : 2015 Certified

Degree College

Computer Journal

CERTIFICATE

SEMESTER 1 **UID No.** _____

Class Msc.cs part - 1 **Roll No.** 4633 **Year** 2023-24

This is to certify that the work entered in this journal is the work of Mst. / Ms. Rahul Prahalad Yadav

who has worked for the year 2023 - 24 in the Computer Laboratory.

Teacher In-Charge

Head of Department

Date : _____

Examiner

INDEX

6.	Practical 6- Create a table that stores the special data and issue queries on it.	52-58	13/11/2023
7.	Practical 7- Create a table employee having dept_id as number datatype and employee_spec as XML data type (XML_Type). The employee_spec is a schema with attributes emp_id, name, email, acc_no, managerEmail, dataOf Joning. Insert 10 tuples into employee table. Fire the following queries on XML database.	59-67	16/11/2023

PRACTICAL 1

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

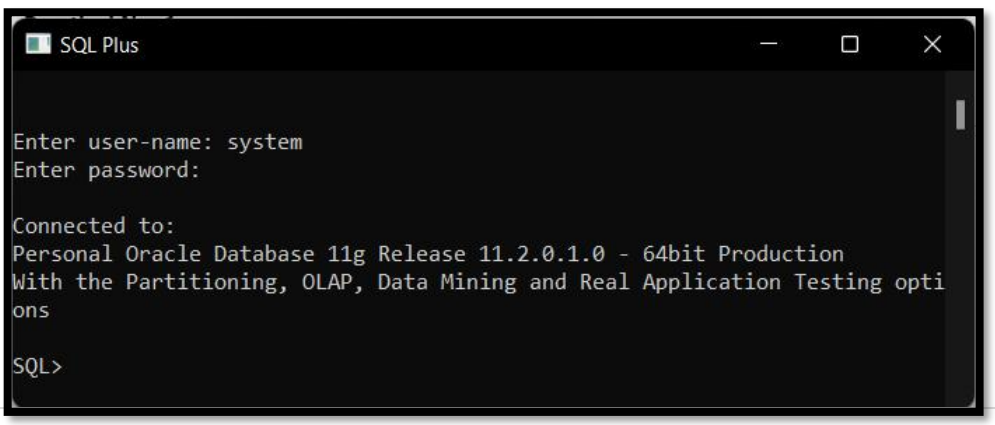
Theory:

A conceptual schema is an abstract representation of the entire database. It provides a high-level view of the data and its organization, focusing on the logical structure and relationships between data entities. It abstracts away the details of how data is physically stored or accessed and focuses on the way data is perceived by users and applications.

The global conceptual schema, in the context of distributed databases, extends the concept of the conceptual schema to encompass the entire distributed database environment. It defines the structure and organization of data across multiple interconnected databases, which may be geographically distributed or managed by different organizations.

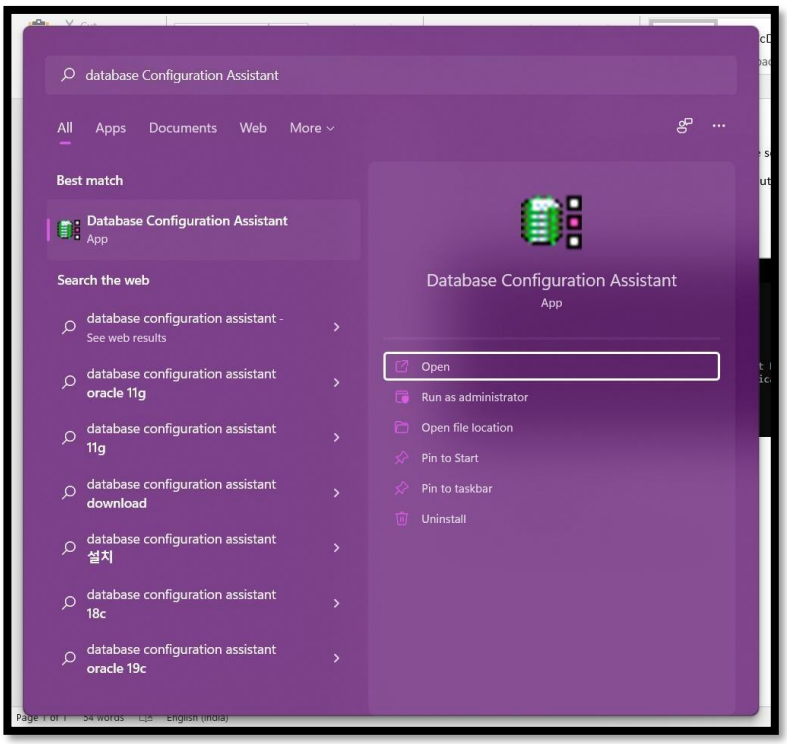
Software Requirement:

Oracle Database 11g.



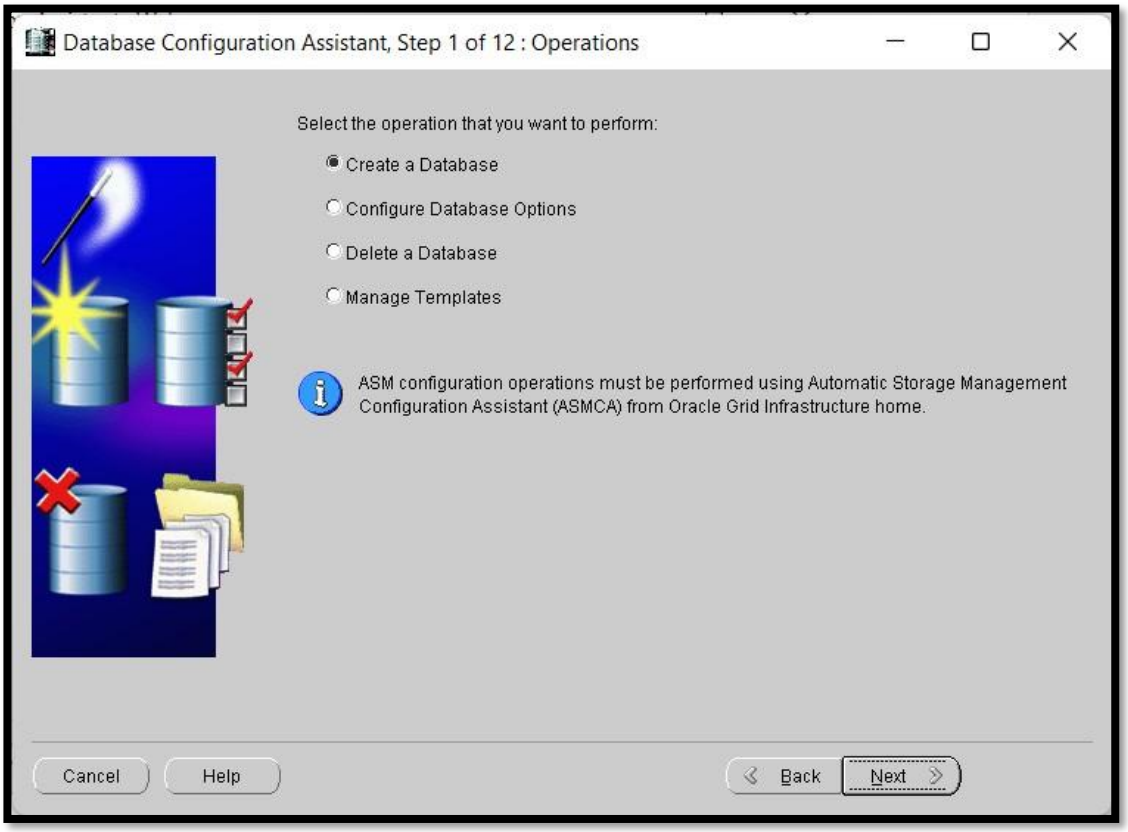
Steps to Create Database db1 and db2:

Step 1:- Open Start Menu on Window Explorer Go to Database Configuration Assistant.

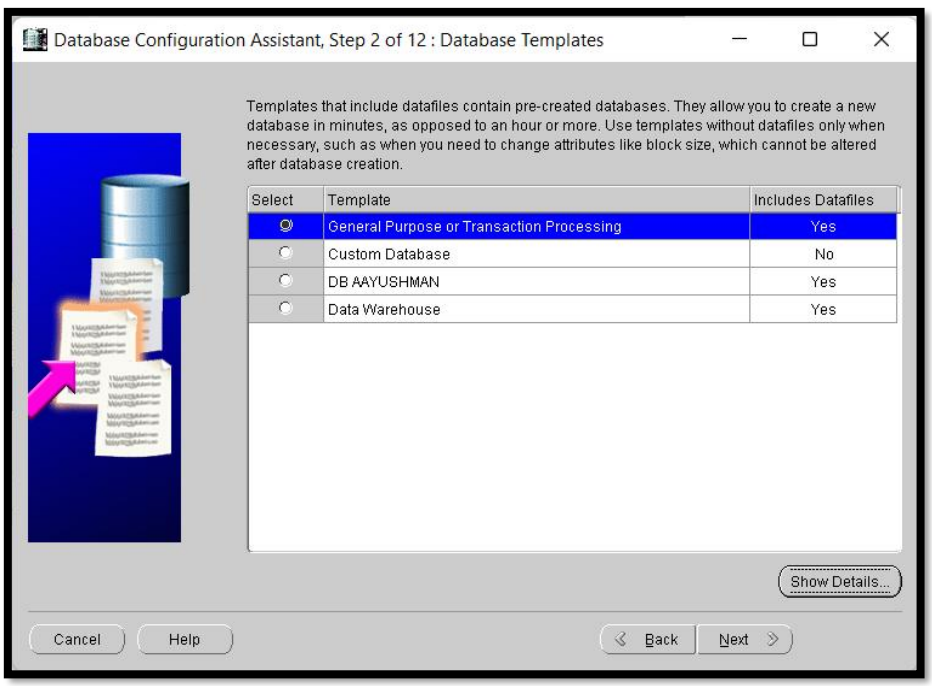


Step 2: Click on Next.

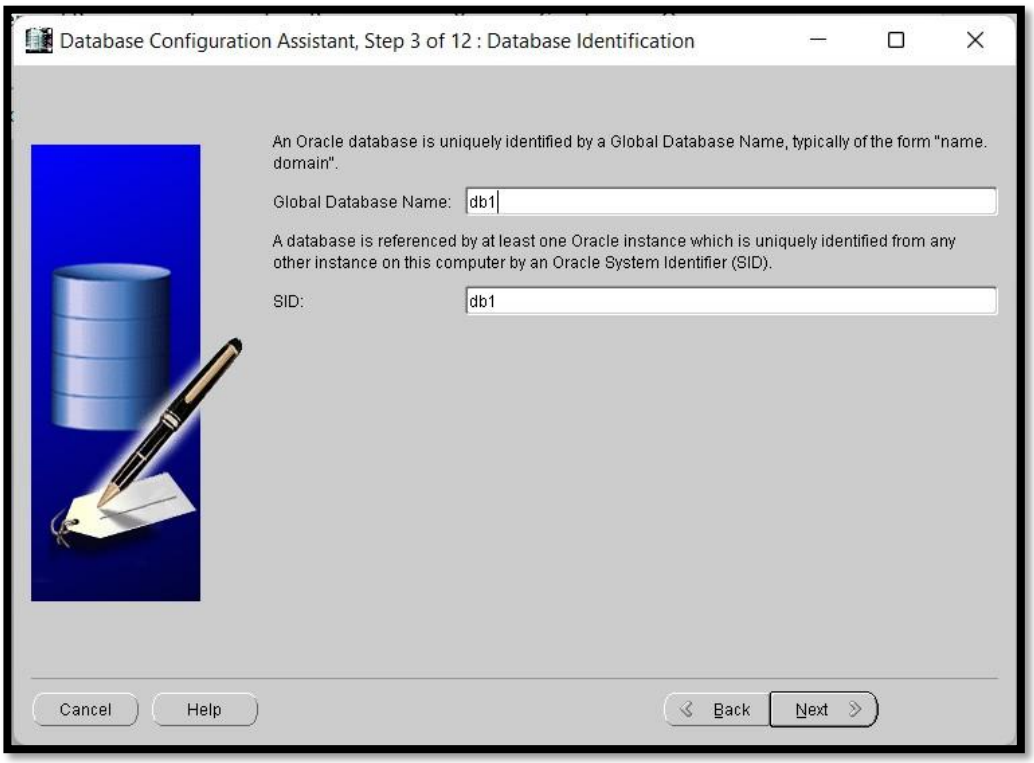
Step 3: Select Option Create a Database.

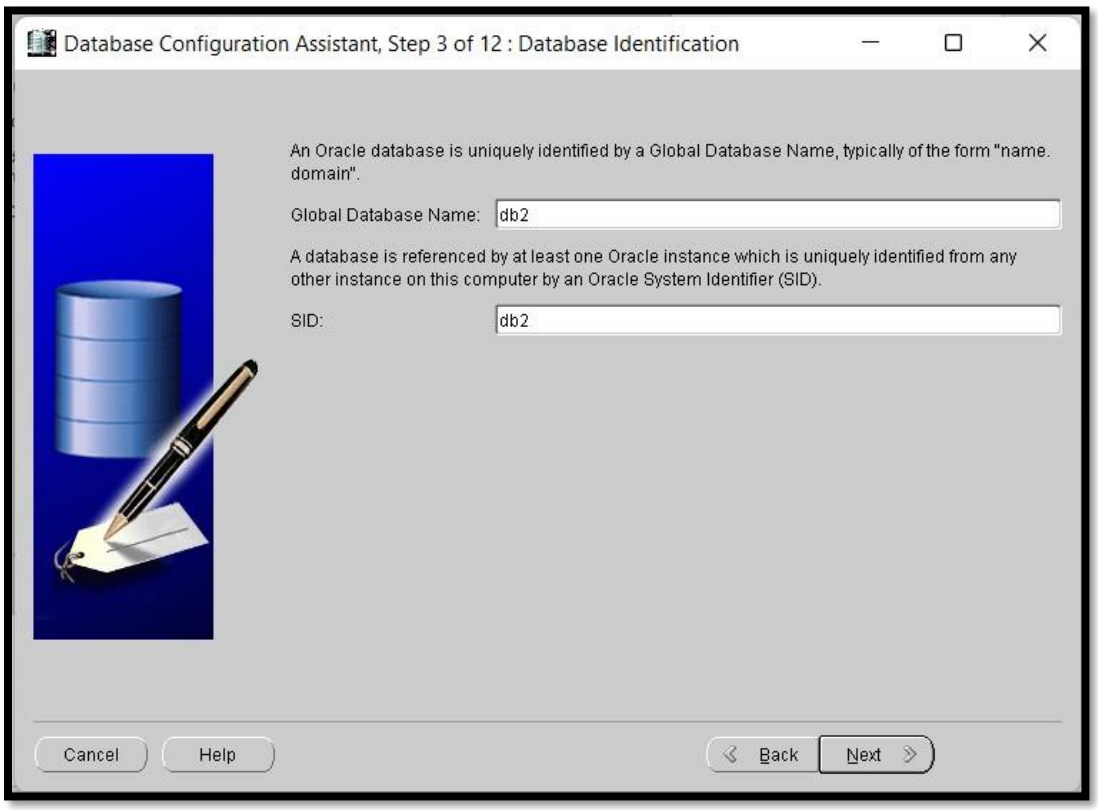


Step 4: Select Option General Purpose or Transaction Processing or You can Create your Own Custom Database.

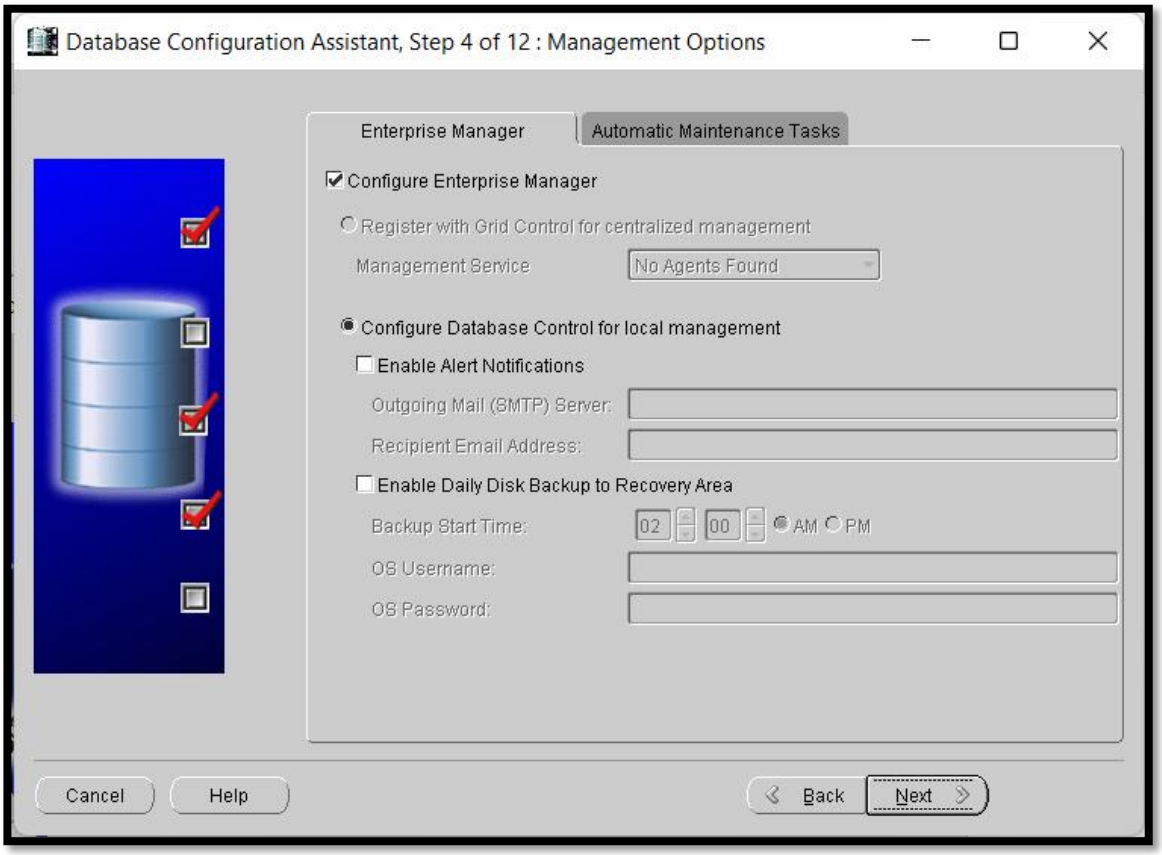


Step 5: Give Database Name as db1 (of your own choice).

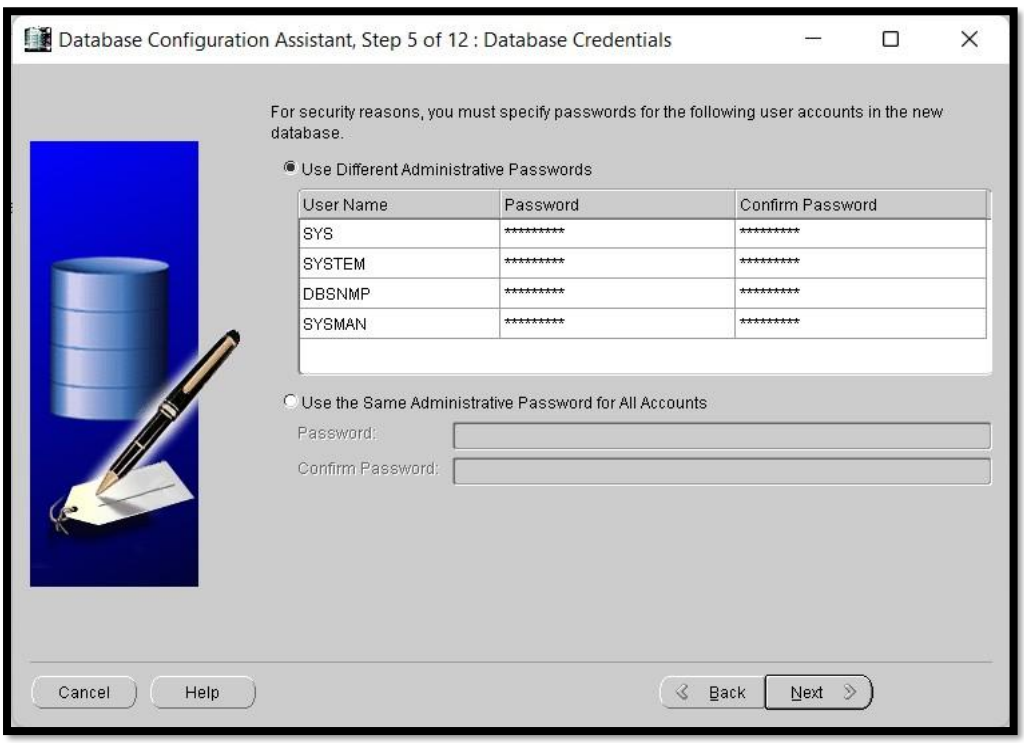




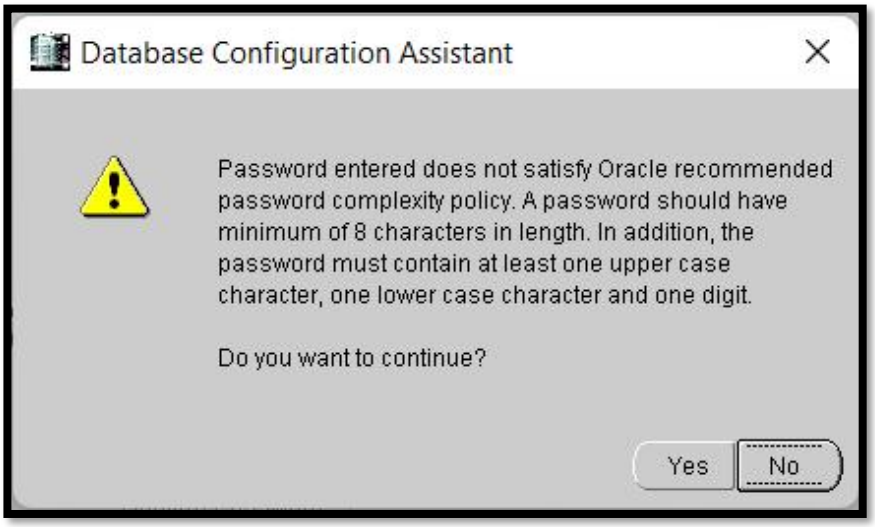
Step 6: No changes Needed, Click on Next.



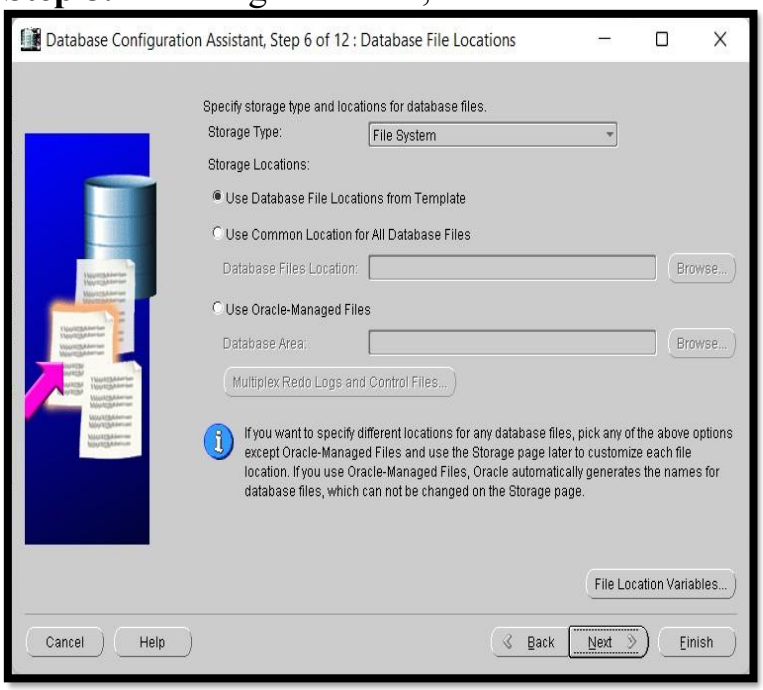
Step 7: Input Password of your choice for Each Fields or Else use your Administrator Credentials for all Profile.



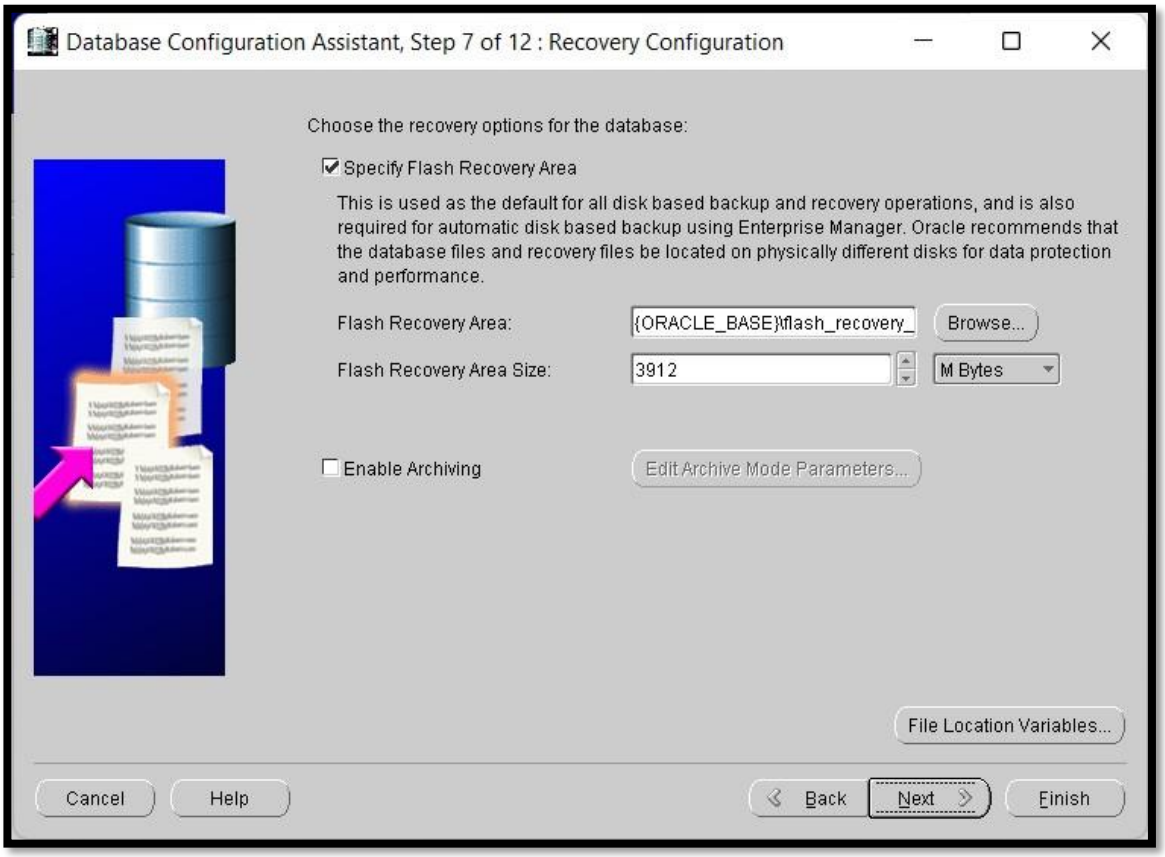
Checks for Password Confirmation, Just Click Yes.



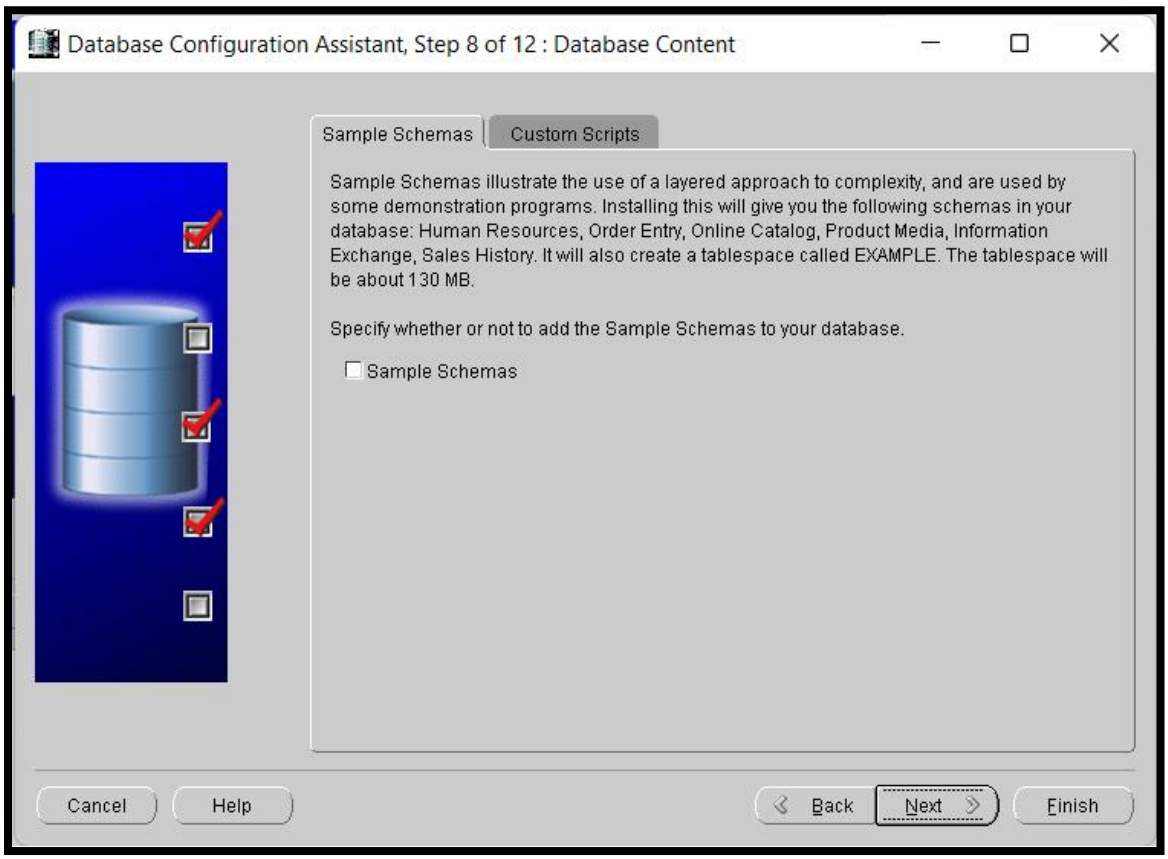
Step 8: No changes Needed, Click on Next.

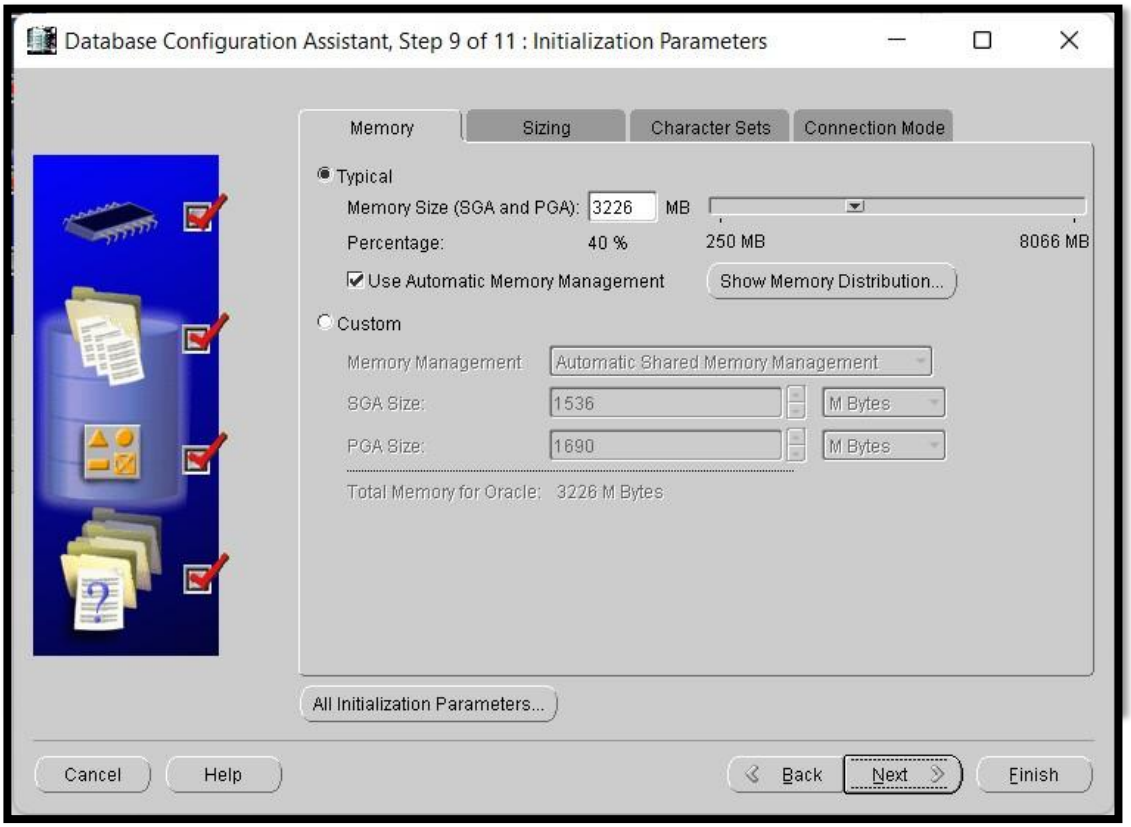


Step 9: No changes Needed, Click on Next.

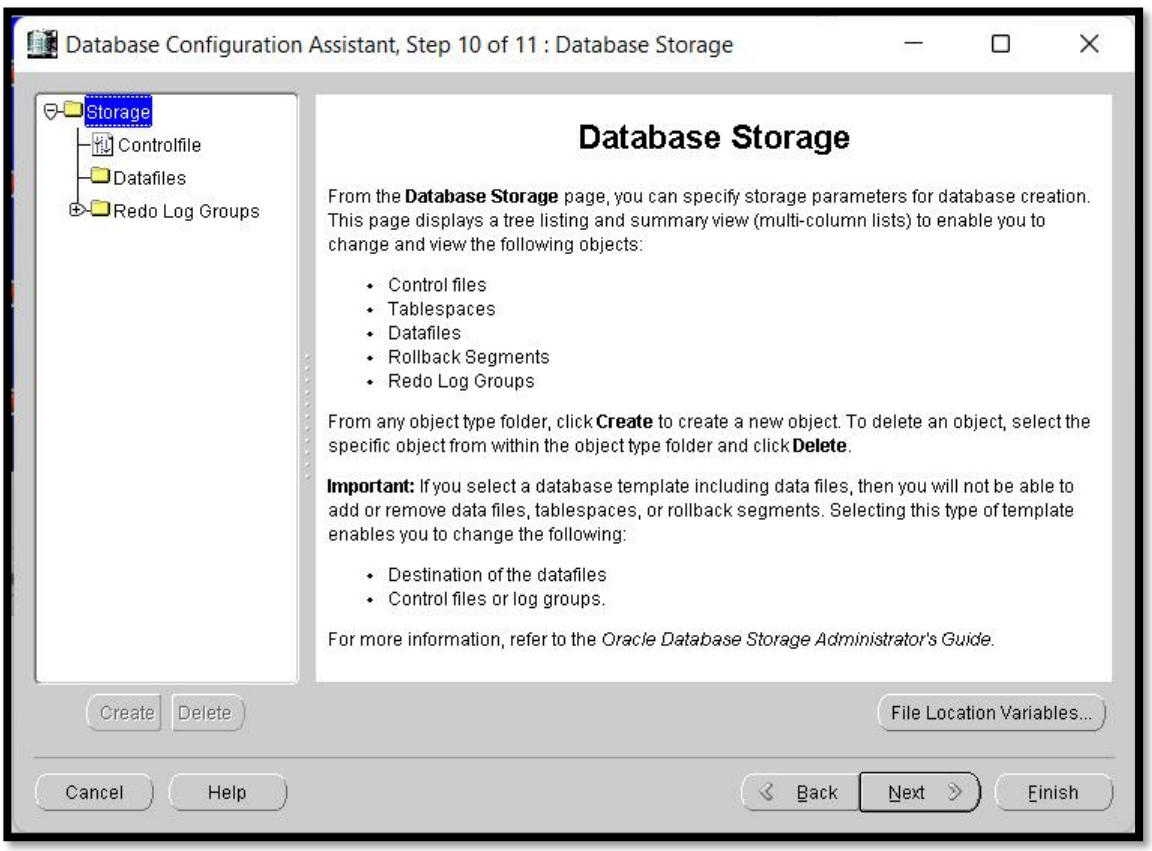


Step 10: No changes Needed, Click on





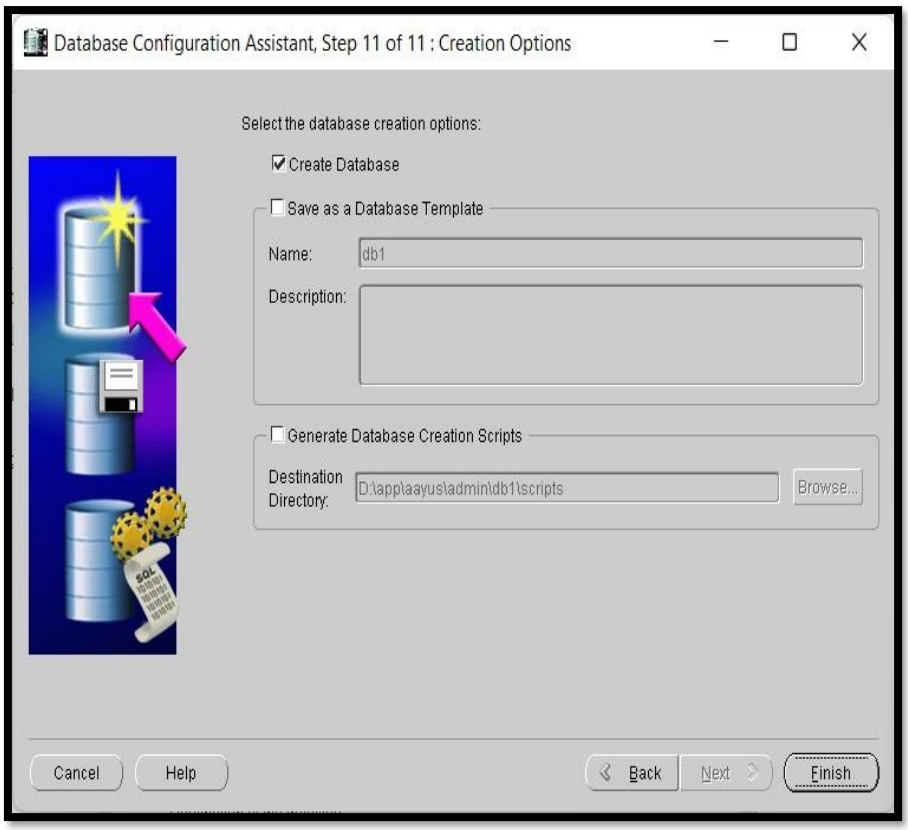
Step 11: No changes Needed, Click on Next.

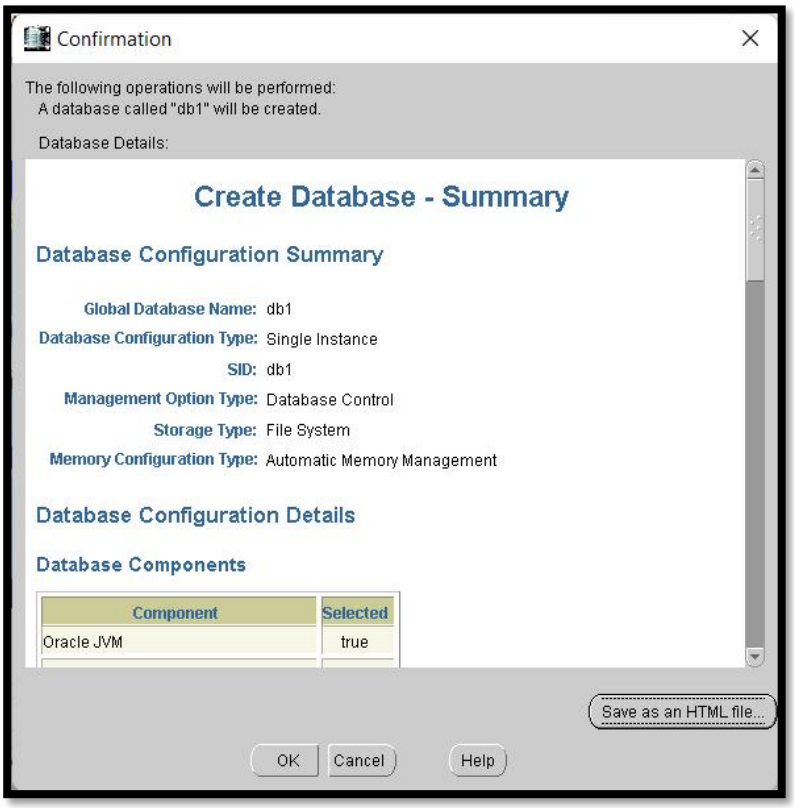


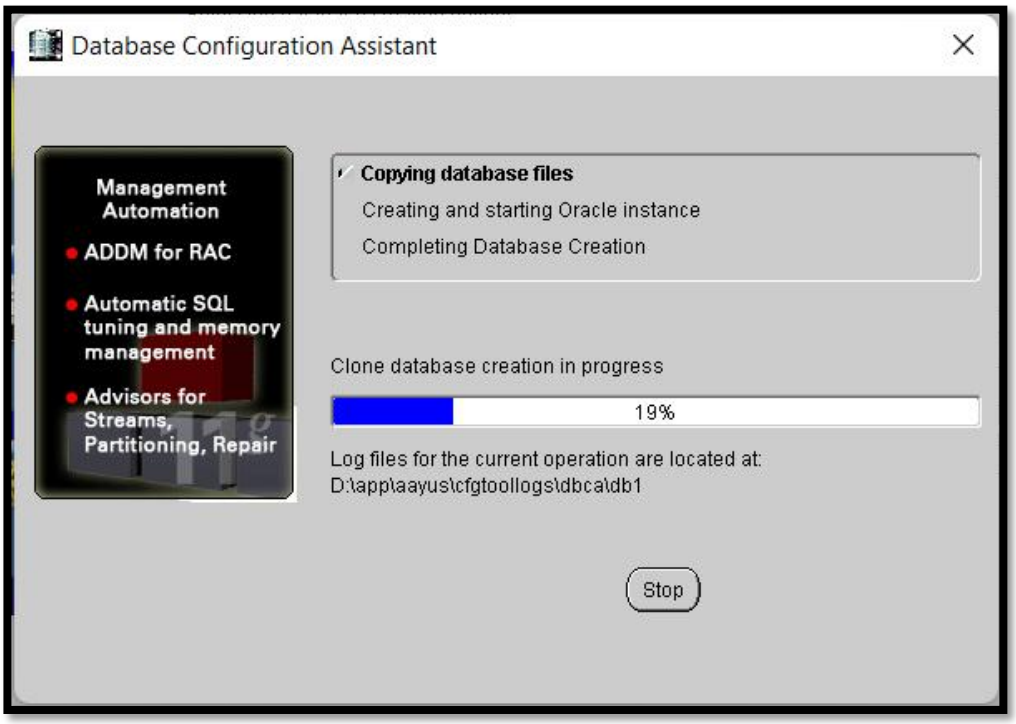
Step 12: No changes Needed, Click on Next.

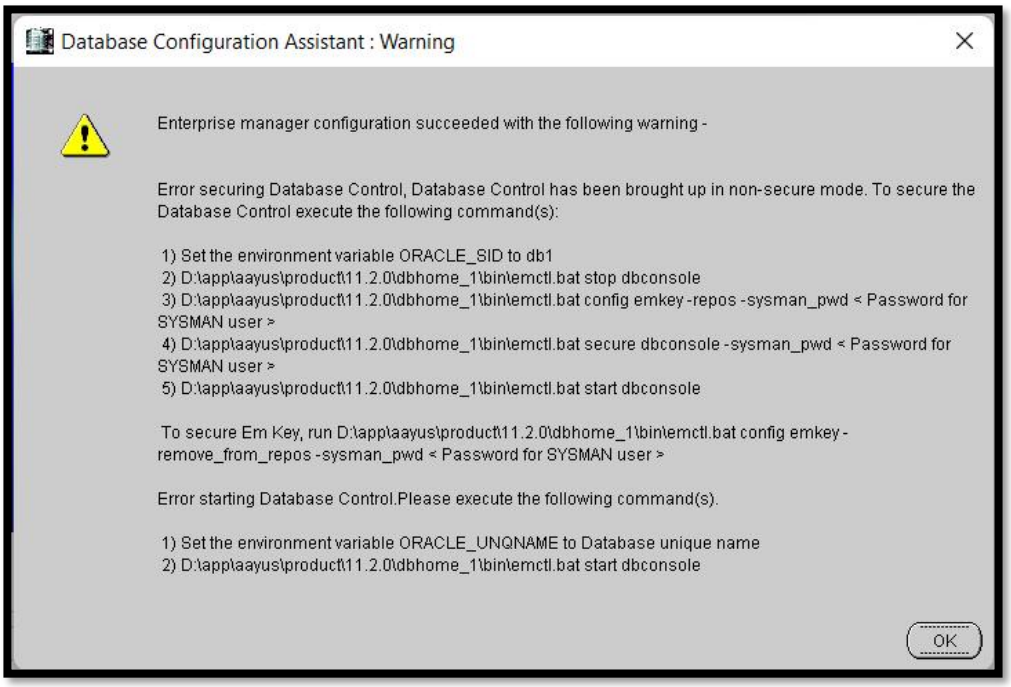
Step 13: No changes Needed, Click on Finish.

Confirmation of Creating Database, You can Save it as well for your database details.Incase you forget credentials for your database, you can take help of this file to get access of your database.




Confirmation
The following operations will be performed:
A database called "db1" will be created.
Database Details:
Create Database - Summary
Database Configuration Summary
Global Database Name: db1
Database Configuration Type: Single Instance
SID: db1
Management Option Type: Database Control
Storage Type: File System
Memory Configuration Type: Automatic Memory Management
Database Configuration Details
Database Components
Component
Selected
Oracle JVM
true
Save as an HTML file...
OK
Cancel
Help


Database Configuration Assistant
Management Automation
ADDM for RAC
Automatic SQL tuning and memory management
Advisors for Streams, Partitioning, Repair
Copying database files
Creating and starting Oracle instance
Completing Database Creation
Clone database creation in progress
19%
Log files for the current operation are located at:
D:\app\aayus\cfgtoollogs\dbca\db1
Stop


Database Configuration Assistant : Warning
Enterprise manager configuration succeeded with the following warning -
Error securing Database Control, Database Control has been brought up in non-secure mode. To secure the Database Control execute the following command(s):
1) Set the environment variable ORACLE_SID to db1
2) D:\app\aayus\product\11.2.0\dbhome_1\bin\emctl.bat stop dbconsole
3) D:\app\aayus\product\11.2.0\dbhome_1\bin\emctl.bat config emkey -repos -sysman_pwd < Password for SYSMAN user >
4) D:\app\aayus\product\11.2.0\dbhome_1\bin\emctl.bat secure dbconsole -sysman_pwd < Password for SYSMAN user >
5) D:\app\aayus\product\11.2.0\dbhome_1\bin\emctl.bat start dbconsole
To secure Em Key, run D:\app\aayus\product\11.2.0\dbhome_1\bin\emctl.bat config emkey -remove_from_repos -sysman_pwd < Password for SYSMAN user >
Error starting Database Control.Please execute the following command(s).
1) Set the environment variable ORACLE_UNQNAME to Database unique name
2) D:\app\aayus\product\11.2.0\dbhome_1\bin\emctl.bat start dbconsole
OK


Database Configuration Assistant
Database creation complete. For details check the logfiles at:
D:\app\aayus\cfgtoollogs\dbca\db1.
Database Information:
Global Database Name: db1
System Identifier(SID): db1
Server Parameter File name: D:\app\aayus\product\11.2.0\dbhome_1\database\spfiledb1.ora
The Database Control URL is http://Aayushman:5501/em
Note: All database accounts except SYS, SYSTEM, DBSNMP, and SYSMAN are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts (except DBSNMP and SYSMAN). From the Password Management window, unlock only the accounts you will use. Oracle Corporation strongly recommends changing the default passwords immediately after unlocking the account.
Password Management...
Exit

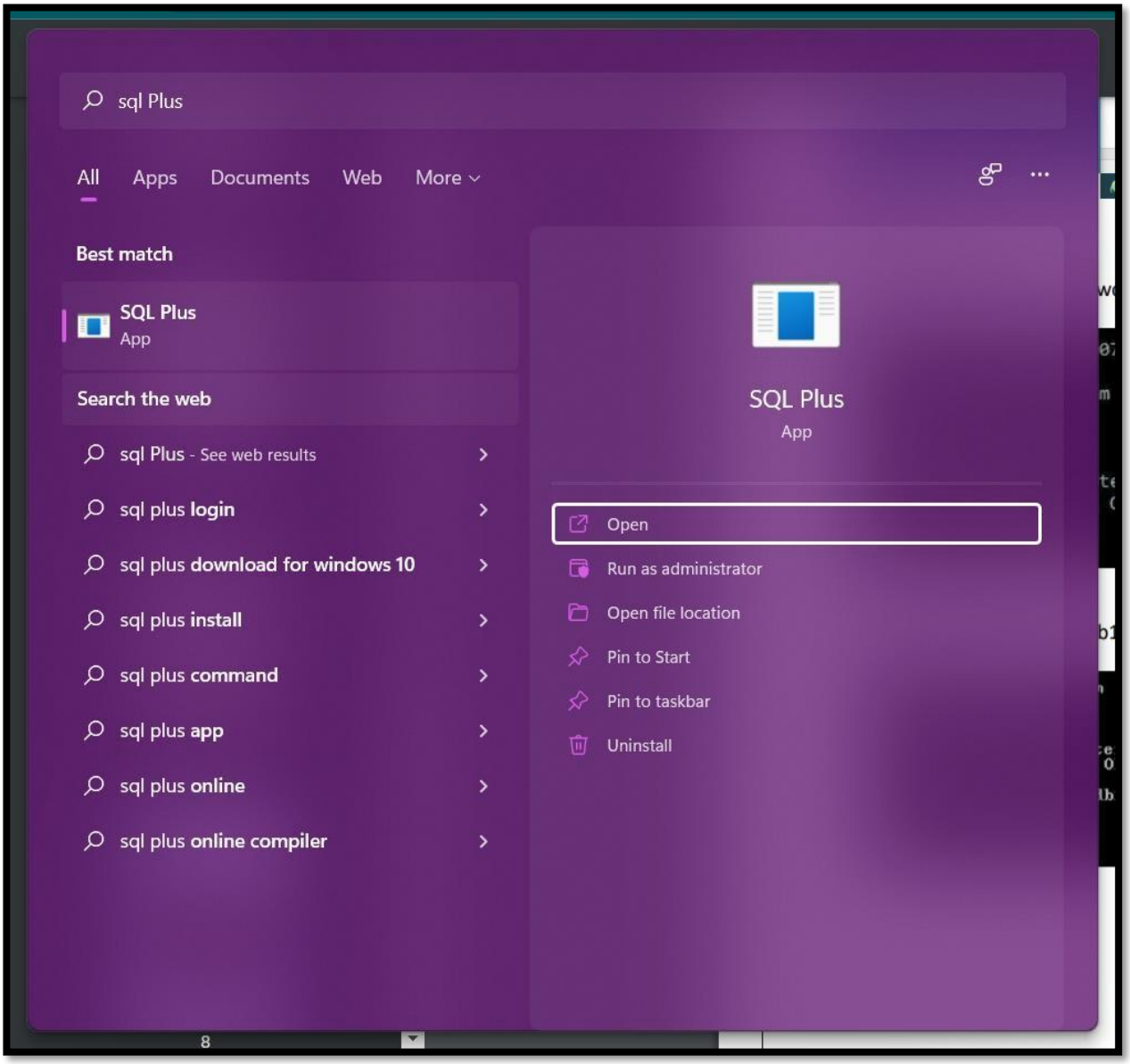
Click on Exit and Done…

Follow the Same Steps to create db2,

Once done with Creating db1 and db2 .

Practical Implementation Steps:

Step 1:- Open SQLPlus.



Step 2: Connect to Your Database .

```
SQL Plus

Enter user-name: system
Enter password:

Connected to:
Personal Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
```

Step 3: Connect your db1 While executing the Command

```
SQL> conn system/aayushman@db1
```

[Where "aayushman" is password of your database, and "db1" is database name].

```
SQL Plus

Enter user-name: system
Enter password:

Connected to:
Personal Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> conn system/aayushman@db1
Connected.
SQL>
```

Step 4: Create one table in database db1.

```
Create one table in database db1.

create table employee027 (
EmpId int primary key,
EmpName varchar(30),
Address varchar(30),
Email varchar(20),
Salary int
);
```

```
SQL Plus

SQL*Plus: Release 11.2.0.1.0 Production on Fri Nov 26 16:29:34 2021

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Personal Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> conn system/aayushman@db1
Connected.
SQL> create table employee027 (
  2  EmpId int primary key,
  3  EmpName varchar(30),
  4  Address varchar(30),
  5  Email varchar(30),
  6  Salary int
  7  );

Table created.

SQL>
```

Step 5: Insert Some values in Created Table.

```
Insert some values into table employee027.

SQL> insert into employee027 values (1, 'aayushman', 'Goregaon', 'aayushmanojha@protonmail.com', 20000);

SQL> insert into employee027 values (2, 'abhishek', 'Kandivali', 'abhishekojha@protonmail.com', 18000);

SQL> insert into employee027 values (3, 'aashi ojha', 'Bandra', 'aashiojha@protonmail.com', 25000);

SQL> insert into employee027 values (4, 'Priyesh', 'Colaba', 'Priyesh@protonmail.com', 23500);

SQL> insert into employee027 values (5, 'Pankaj', 'Madh', 'Pankaj@protonmail.com', 15200);
```

```
SQL Plus

SQL*Plus: Release 11.2.0.1.0 Production on Fri Nov 26 16:29:34 2021

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Personal Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> conn system/aayushman@db1
Connected.
SQL> create table employee027 (
  2  EmpId int primary key,
  3  EmpName varchar(30),
  4  Address varchar(30),
  5  Email varchar(30),
  6  Salary int
  7  );

Table created.

SQL> insert into employee027 values (1, 'aayushman', 'Goregaon', 'aayushmanojha@protonmail.com', 20000);


1 row created.

SQL> insert into employee027 values (2, 'abhishek', 'Kandivali', 'abhishekojha@protonmail.com', 18000);

1 row created.

SQL> insert into employee027 values (3, 'aashi ojha', 'Bandra', 'aashiojha@protonmail.com', 25000);

1 row created.

SQL> insert into employee027 values (4, 'Priyesh', 'Colaba', 'Priyesh@protonmail.com', 23500);

1 row created.

SQL> insert into employee027 values (5, 'Pankaj', 'Madh', 'Pankaj@protonmail.com', 15200);

1 row created.

SQL> _
```

Step 6:

```
Show all tables in employee.

SQL> Select * from employee027;
```

```
SQL Plus
SQL> select * from employee027;

     EMPID EMPNAME                        ADDRESS                        EMAIL                            SALARY
---------- ---------------------------- ---------------------------- ---------------------------- ----------
         6 kyara                        Borivali                     kyara@protonmail.com               15000
         1 aayushman                    Goregaon                     aayushmanojha@protonmail.com       20000
         2 abhishek                     Kandivali                    abhishekojha@protonmail.com        18000
         3 aashi ojha                   Bandra                       aashiojha@protonmail.com           25000
         4 Priyesh                      Colaba                       Priyesh@protonmail.com             23500
         5 Pankaj                       Madh                         Pankaj@protonmail.com              15200

6 rows selected.

SQL>
```

Step 7: Enter following command to create link between two databases.

```
Enter following command to create link between two databases.

SQL> create database link db1todb2 connect system identified by aayushman using 'db2';
```

```
SQL Plus
SQL> create database link db1todb2 connect to system identified by aayushman using 'db2';

Database link created.

SQL>
```

Step 8: Connect to Db2.

```
SQL Plus
SQL> conn system/aayushman@db2
Connected.
SQL>
```

Step 9: Create link to connect db1.

```
Create link to connect db1.

SQL> create database link db2todb1 connect system identified by aayushman using 'db1';
```

```
SQL Plus
SQL> create database link db2todb1 connect to system identified by aayushman using 'db1';

Database link created.

SQL>
```

Step 10: Create emp1 select where salary<18000.

```
Create emp1 select where salary<18000.

SQL> create table emp1 as select * from employee027@db2todb1 where salary<18000;
```

```
SQL Plus

SQL> create table emp1 as select * from employee027@db2todb1 where salary < 18000;

Table created.

SQL> set linesize 1000
SQL> select * from emp1;

     EMPID EMPNAME                        ADDRESS                        EMAIL                              SALARY
---------- ------------------------------ ------------------------------ ------------------------------ ----------
         6 kyara                          Borivali                       kyara@protonmail.com                15000
         5 Pankaj                         Madh                           Pankaj@protonmail.com               15200

SQL>
```

Step 11: Create table emp2 where address='Bandra'.

```
Create table emp2 where address='Bandra'.

SQL> > create table emp2 as select * from employee027@db2todb1 where address='Bandra';
```

```
SQL Plus

SQL> create table emp2 as select * from employee027@db2todb1 where address='Bandra';

Table created.

SQL> select * from emp2;

     EMPID EMPNAME                        ADDRESS                        EMAIL                              SALARY
---------- ------------------------------ ------------------------------ ------------------------------ ----------
         3 aashi ojha                     Bandra                         aashiojha@protonmail.com            25000

SQL>
```

Step 12: Select salary from employee.

```
Select salary from employee

SQL> conn system/aayushman@db2
SQL> select salary from employee027@db2todb1;
```

```
SQL Plus
SQL> conn system/aayushman@db2
Connected.
SQL> select salary from employee027@db2todb1;

    SALARY
----------
     15000
     20000
     18000
     25000
     23500
     15200

6 rows selected.

SQL>
```

Step 13: Select mail whose salary>16000.

```
Select email whose salary>16000.

SQL> select email from employee027@db2todb1 where salary > 16000
```

```
SQL Plus
SQL> select Email from employee027@db2todb1 where salary > 16000;

EMAIL
------------------------------
aayushmanojha@protonmail.com
abhishekojha@protonmail.com
aashiojha@protonmail.com
Priyesh@protonmail.com

SQL>
```

Step 14: Select Employee Name and Email from Employee table where eid=2.

```
Select ename, email from employee where eid=2.

SQL> select EmpName,Email from employee027@db2todb1 where eid=2;
```

```
SQL Plus
SQL> select EmpName, Email from employee027@db2todb1 where EmpId=2;

EMPNAME                        EMAIL
------------------------------ ------------------------------
abhishek                       abhishekojha@protonmail.com

SQL>
```

Step 15: Create table emp3 where address='Madh'.

```
Create table emp3 where address='Madh'.

SQL> create table emp3 as select * from employee027@db2todb1 where address='Madh';
```

```
SQL Plus
SQL> create table emp3 as select * from emp1@db1todb2 where address='Madh';

Table created.

SQL> select * from emp3;

     EMPID EMPNAME                        ADDRESS                        EMAIL                           SALARY
---------- ---------------------------- ---------------------------- ---------------------------- ----------
         5 Pankaj                       Madh                           Pankaj@protonmail.com            15200

SQL>
```

Conclusion:

Successfully executed Schema into horizontal and vertical Fragmentation on different nodes in Distributed Database Environment.

PRACTICAL 2

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

Theory:

The global conceptual schema, in the context of distributed databases, extends the concept of the conceptual schema to encompass the entire distributed database environment. It defines the structure and organization of data across multiple interconnected databases, which may be geographically distributed or managed by different organizations.

The global conceptual schema defines a unified view of data across multiple databases, taking into account data distribution, data replication, and data synchronization among the various components of the distributed system.

It is used to provide a consistent and coherent view of the data for applications and users, even when data is distributed across different physical locations or systems.

Software Requirements:

Oracle 11g.

Query:

1. Update any record in db1 & show in db2
2. Delete any record in db1 & show in db2.
3. Find the salary of all employees.
4. Find the email of all employees where salary = 15000.
5. Find the employee name and email where employee number is known.
6. Find the employee name and address where employee number is known.

Steps:

```
SQL Plus

SQL*Plus: Release 11.2.0.1.0 Production on Sun Nov 28 18:36:31 2021

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Personal Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> conn system/aayushman@db1
Connected.
SQL> create table emp
  2  (
  3  enumber number primary key,
  4  ename varchar2(20),
  5  addr varchar2(20),
  6  eemail varchar2(20),
  7  eesalary float
  8  );

Table created.

SQL>
```

Step 1: Create Table in db1.

```
SQL Plus

SQL> conn system/aayushman@db2
Connected.
SQL> create table emp
  2  (
  3  enumber number primary key,
  4  ename varchar2(20),
  5  addr varchar2(20),
  6  eemail varchar2(20),
  7  eesalary float
  8  );

Table created.

SQL>
```

Step 2: Create Table in db2.

```
SQL Plus

SQL> create database link db1todb3 connect to system identified by aayushman using 'db3';

Database link created.

SQL> create database link db3todb1 connect to system identified by aayushman using 'db1';

Database link created.

SQL>
```

Step 3: Create Database link.

Step 4: Create Trigger to Insert Data.

```
SQL Plus
SQL> conn system/aayushman@db1
Connected.
SQL> create or replace Trigger insert_data
  2  after insert on emp
  3  for each row
  4  begin
  5  insert into emp@db1todb2
  6  values(:new.enumber,:new.ename,:new.addr,:new.eemail,:new.eesalary);
  7  end;
  8  /

Trigger created.

SQL>
```

Step 5: Create Trigger to Update Data in Table.

```
SQL Plus

SQL> create or replace Trigger del_data
  2  before delete on emp
  3  for each row
  4  begin
  5  delete from emp@db1todb2
  6  where enumber=:old.enumber;
  7  end;
  8  /

Trigger created.

SQL>
```

```
SQL Plus

SQL> create or replace Trigger update_data
  2  after update on emp
  3  for each row
  4  begin
  5  update emp@db1todb2
  6  set enumber= :new.enumber,
  7  ename= :new.ename,
  8  addr= :new.addr,
  9  eemail= :new.eemail,
 10  eesalary= :new.eesalary
 11  where enumber= :old.enumber;
 12  end;
 13  /

Trigger created.
```

Step 6: Insert Values in Created Table.

```
SQL> insert into employee values(1,'yash','mumbai','yash@email.com',5000);

1 row created.

SQL> insert into employee values(2,'him','chennai','him@email.com',4000);

1 row created.

SQL> insert into employee values(3,'hiakyu','chennai','hiakyu@email.com',3000);

1 row created.

SQL> insert into employee values(4,'kage','mumbai','kage@email.com',2000);

1 row created.
```

Show Create Tables.

```
SQL Plus

SQL> set linesize 500
SQL> select * from emp;

   ENUMBER ENAME                ADDR                 EEMAIL                EESALARY
---------- -------------------- -------------------- -------------------- ----------
       111 Aayushman            Bhayandar            aayush@yahoo.com         100000
       112 Sagar                Goregaon             sagar@yahoo.com           10400
       113 Aashi                Borivali             Aashi@yahoo.com          100000
       114 Ashish               Bhayandar            ash@yahoo.com             15000
       115 Tejas                Bhayandar            tej@yahoo.com              5000
       116 Shubham              Malad                shub@yahoo.com           100000
       117 Priyanka             Vasai                priya@yahoo.com           15000
       118 Swapna               Nalasopara           swap@yahoo.com           300000
       119 Samodaya             Goregaon             sam@yahoo.com             15000

9 rows selected.

SQL>
```

Query

1. **Update any record in db1 & show in db2.**

```
SQL Plus

SQL> update emp
  2  set eesalary=800000
  3  where enumber=114;

1 row updated.

SQL> select * from emp;

   ENUMBER ENAME                ADDR                 EEMAIL                EESALARY
---------- -------------------- -------------------- -------------------- ----------
       111 Aayushman            Bhayandar            aayush@yahoo.com         100000
       112 Sagar                Goregaon             sagar@yahoo.com           10400
       113 Aashi                Borivali             Aashi@yahoo.com          100000
       114 Ashish               Bhayandar            ash@yahoo.com            800000
       115 Tejas                Bhayandar            tej@yahoo.com              5000
       116 Shubham              Malad                shub@yahoo.com           100000
       117 Priyanka             Vasai                priya@yahoo.com           15000
       118 Swapna               Nalasopara           swap@yahoo.com           300000
       119 Samodaya             Goregaon             sam@yahoo.com             15000

9 rows selected.

SQL>
```

Show Updated Table in db2.

```
Select SQL Plus

SQL> conn system/aayushman@db2
Connected.
SQL> select * from emp;

   ENUMBER ENAME                ADDR                 EEMAIL                EESALARY
---------- -------------------- -------------------- -------------------- ----------
       111 Aayushman            Bhayandar            aayush@yahoo.com         100000
       112 Sagar                Goregaon             sagar@yahoo.com           10400
       113 Aashi                Borivali             Aashi@yahoo.com          100000
       114 Ashish               Bhayandar            ash@yahoo.com            800000
       115 Tejas                Bhayandar            tej@yahoo.com              5000
       116 Shubham              Malad                shub@yahoo.com           100000
       117 Priyanka             Vasai                priya@yahoo.com           15000
       118 Swapna               Nalasopara           swap@yahoo.com           300000
       119 Samodaya             Goregaon             sam@yahoo.com             15000

9 rows selected.

SQL> _
```

2. Delete any record in db1 & show in db2.

```
SQL Plus

SQL> conn system/aayushman@db1
Connected.
SQL> delete from emp where enumber=111;

1 row deleted.

SQL> conn system/aayushman@db2
Connected.
SQL> select * from emp;

   ENUMBER ENAME                ADDR                 EEMAIL                EESALARY
---------- -------------------- -------------------- -------------------- ----------
       112 Sagar                Goregaon             sagar@yahoo.com           10400
       113 Aashi                Borivali             Aashi@yahoo.com          100000
       114 Ashish               Bhayandar            ash@yahoo.com            800000
       115 Tejas                Bhayandar            tej@yahoo.com              5000
       116 Shubham              Malad                shub@yahoo.com           100000
       117 Priyanka             Vasai                priya@yahoo.com           15000
       118 Swapna               Nalasopara           swap@yahoo.com           300000
       119 Samodaya             Goregaon             sam@yahoo.com             15000

8 rows selected.

SQL> _
```

3. Find the salary of all employees.

```
SQL Plus
SQL> select ename, eesalary from emp;

ENAME                 EESALARY
-------------------- ----------
Sagar                    10400
Aashi                   100000
Ashish                  800000
Tejas                     5000
Shubham                 100000
Priyanka                 15000
Swapna                  300000
Samodaya                 15000

8 rows selected.

SQL>
```

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

```
SQL Plus
SQL> select eemail from emp where eesalary= 15000;

EEMAIL
--------------------
priya@yahoo.com
sam@yahoo.com

SQL>
```

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

```
SQL Plus
SQL> select ename,eemail from emp where enumber=113;

ENAME                EEMAIL
-------------------- --------------------
Aashi                Aashi@yahoo.com

SQL>
```

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

```
SQL Plus
SQL> select ename,addr,eemail from emp where enumber=113;

ENAME                ADDR                 EEMAIL
-------------------- -------------------- --------------------
Aashi                Borivali             Aashi@yahoo.com

SQL>
```

Conclusion: Successfully Created Triggers and Perform Different Queries on them.

PRACTICAL 3

Aim:- CRUD operation using MongoDB.

Theory:

MongoDB is a popular, open-source NoSQL database management system that is designed to store and manage large volumes of unstructured or semi-structured data. MongoDB is part of a class of databases known as document-oriented databases, which are designed to be flexible and highly scalable.

CRUD operations are fundamental actions used to manage data in a database:

1) Create (C): It involves adding new data records to the database. This is done using an "insert" operation, where new data is added to the database.

2) Read (R): It involves retrieving or reading data from the database. This is done using a "select" operation to retrieve data from the database.

3) Update (U): It involves modifying or updating existing data in the database. This is done using an "update" operation to change the values of existing records.

4) Delete (D): It involves removing data from the database. This is done using a "delete" operation to remove data records from the database.

Steps:

To run mongoDb from your command prompt:

```
C:\Users\shweta>cd C:\Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongod
2021-03-11T21:26:22.045+0530 I  CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDi
sabledProtocols 'none'
2021-03-11T21:26:22.053+0530 W  ASIO     [main] No TransportLayer configured during NetworkInterface startup
2021-03-11T21:26:22.054+0530 I  CONTROL  [initandlisten] MongoDB starting : pid=8552 port=27017 dbpath=C:\data\db\ 64-bi
t host=DESKTOP-DL3T888
```

By performing the above command you are going to on your mongo DB serverNow to on your mongoDb shell :open new cmd

mongo

```
C:\Program Files\MongoDB\Server\4.2\bin>mongo
MongoDB shell version v4.2.17
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c58bfe33-58d6-42e5-a852-1f217de5dc89") }
MongoDB server version: 4.2.17
Welcome to the MongoDB shell.
For interactive help, type "help".
```

Creating and selecting database;

use msccs

```
> use msccs
switched to db msccs
> db
msccs
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

To check the current connected database : **db**To see the list of all database : **show dbs**

C : CREATING COLLECTION AND INSERTING VALUES :

Creating a collection and inserting values can be done together. Here we have or collection name as ‘student ‘

> **db.student.insert({name:"sds"})**
> **db.student.insert({no:2,name:"yash",course:{coursename:"bsc",duration:"3yrs" },address:{city:"mumbai",state:"maharashtra",country:"india"}})**

```
> db.student.insert({name:"sds"})
WriteResult({ "nInserted" : 1 })
```

```
> db.student.insert({no:2,name:"yash",course:{coursename:"bsc",duration:"3 yrs"},address:{city:"mumbai",state:"maharashtra",country:"india"}})
WriteResult({ "nInserted" : 1 })
>
```

R : READ DATA FROM THE COLLECTION :

To retrieve the inserted document,

> **db.student.find()**

```
>
> db.student.find()
{ "_id" : ObjectId("619f4f3fc5c3f41a15053a00"), "name" : "sds" }
{ "_id" : ObjectId("619f4f7dc5c3f41a15053a01"), "no" : 2, "name" : "yash", "course" : { "coursename" : "bsc", "duration" : "3 yrs"
>
```

U : UPDATING A DOCUMENT IN A COLLECTION :

> **db.student.update({no:2},{$set:{"name":"notyash"}})**

```
> db.student.update({no:2},{$set:{"name":"notyash"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.student.find()
{ "_id" : ObjectId("619f4f3fc5c3f41a15053a00"), "name" : "sds" }
{ "_id" : ObjectId("619f4f7dc5c3f41a15053a01"), "no" : 2, "name" : "notyash", "co
>
```

D : REMOVING AN ENTRY FROM THE COLLECTION (DELETE)

> **db.student.remove({no:2})**

```
> db.student.remove({no:2})
WriteResult({ "nRemoved" : 1 })
> db.student.find()
{ "_id" : ObjectId("619f4f3fc5c3f41a15053a00"), "name" : "sds" }
>
```

Conclusion: We have successfully performed CRUD operations.

PRACTICAL 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) BranchType (address: AddrType1, phone1: integer,phone2: integer)
c) AuthorType (name:char,,addr AddrType1)
d) PublisherType (name: char, addr: AddrType1, branches: BranchTableTypee)
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 anintegrity constraint requiring that the name of an author is unique in a list of authors has not beenspecified).

Theory:

1) Insert: The "insert" operation is used to add new data or records to a database. It creates new entries in the database with specified values.

2) Update: The "update" operation is used to modify existing data or records in the database. It allows you to change the values of specific fields within a record.

3) Delete: The "delete" operation is used to remove data or records from the database. It deletes entries that meet specified criteria, effectively removing them from the database.

Steps:

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

SQL> **create or replace type BranchType as object (address AddrType1, phone1integer,phone2 integer);**

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

```
SQL> Create or replace type AddrType1 as object (PinQuery number (5), Street char(20), City varchar2(50), state varchar2(40), no number(4));
  2  /

Type created.

SQL> create or replace type BranchType as object (address AddrType1, phone1 integer,phone2 integer );
  2  /

Type created.

SQL> create or replace type BranchTableType as table of BranchType;
  2  /

Type created.
```

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

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

```
SQL> create or replace type AuthorType as object (name varchar2 (50), addr AddrType1);
  2  /

Type created.

SQL> create table Authors of AuthorType;

Table created.
```

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

```
SQL> create or replace type AuthorListType as varray(10) of ref AuthorType ;
  2  /

Type created.
```

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

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

```
SQL> create or replace type PublisherType as object(name varchar2(50), addr AddrType1, branches BranchTableType);
  2  /

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 refPublisherType,authorsAuthorListType);**

```
SQL> create table books(title varchar2(50), year date, published_by ref PublisherType,authors AuthorListType);

Table created.
```

> insert into Authors values('Aakash', AddrType1(7000,'AT street','mumbai','maharashtra',1007));
> insert into Authors values(‘abc’, AddrType1(7000,'AT street','mumbai','maharashtra',1007));
> insert into Authors values('Salman',AddrType1(7003,'PL street','mumbai','maharashtra',1003));
> insert into Authors values('Vikas',AddrType1(7008,'AT street', 'mumbai','maharashtra',1007));
> insert into Authors values ('Rajan', AddrType1(7006,'Nehrut','mumbai','maharashtra',1005));
> insert into Authors values ('Tejas', AddrType1(8002,'THstreet', 'pune', 'maharashtra' , 13));
> insert into Authors values('Ashish',AddrType1(7008,'TT street', 'Nasik','maharashtra',1008));
> insert into Authors values('Richard',AddrType1(7002,'FL street','pune','maharashtra',03));

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

1 row created.

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

1 row created.

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

1 row created.

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

1 row created.

SQL> insert into Authors values('Ashish',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.
```

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

insert into Publishers

values('McGraw',AddrType1(7007,'LJstreet','mumbai' ,'maharashtra',07), BranchTableType(BranchType (AddrType1 (7007,'K street','mumbai', 'maharashtra',1007), 4543545,8676775)));

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

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

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

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

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='Abhijith';

insert into books select 'c prog','25-may-1983', ref (pub),AuthorListType(ref(aut)) fromPublishers pub,Authorsaut where pub.name='Vipul' and aut.name='Tejas';

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

0 rows 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='Richard';

2 rows created.

SQL>
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='Abhijith';

0 rows created.

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

0 rows created.
```

Firing Queries on the tables.

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

   select a.name from Authors a, Publishers pwhere a.addr.pinQuery = p.addr.pinQuery;

```
NAME
----------------------------
Vikas
Ashish
Richard
```

2) **List all books that have 2 or more authors**
   Query:

   Select title from books b where 1 <= (select count(*) from table(b.authors));

```
TITLE
----------------
IP
IP
```

3) **List the name of the publisher that has the most branches**
   Query:

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

```
NAME
---------------
Tata
```

4) **List all authors who have published more than one book**<u>**Query:**</u>
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;

```
SQL> select a.name from authors a, books b, table (b.authors) v
  2  where v.column_value = ref(a) group by a.name having count(*) > 1
  3  ;

NAME
--------------------------------------------------
Richard
```

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).**

<u>**Query:**</u>

select title from authors a, books b, table (b.authors) v wherev.column_value = ref(a) group by title having count(*) > 1;

<u>Conclusion:</u>

Successfully implemented different attributes and special data types.

PRACTICAL 5

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

Theory:

A temporal database is a type of database system that is designed to handle and manage data with an explicit notion of time. In temporal databases, data is not just stored with a current timestamp; it also includes information about when data was valid, how it changed over time, and historical records of past states. Temporal databases are especially useful for applications that need to track and analyze changes in data over time such as historical records, versioning systems, and financial systems.

Steps:

Create table:

SQL> create table Emp_Appnt(Acc_Nonumber(10),Name varchar2(10), RECDate date, RETDate date);

```
SQL> create table Emp_Appnt ( Acc_No number(10), Name varchar2(10), RECDate date, RETDate date);

Table created.
```

Inserting rows :

insert into Emp_Appnt values(1235,'Aakash Pal','08-mar-1987','12-oct-2015') ;
insert into Emp_Appnt values(1235,'Alpa','08-oct-1978','19-nov-2020') ;
insert into Emp_Appnt values(1237,'ac','25-jan-1988','20-feb-2021') ;
insert into Emp_Appnt values(1278,'xyz','05-dec-1978','02-mar-2017') ;
insert into emp_appnt values(1789,'mon','06-nov-1999','22-mar-2021');

```
SQL> insert into Emp_Appnt values(1235,'Aakash Pal','08-mar-1987','12-oct- 2015') ;

1 row created.

SQL> insert into Emp_Appnt values(1235,'Alpa','08-oct-1978','19-nov-2020') ;

1 row created.

SQL> insert into Emp_Appnt values(1237,'ac','25-jan-1988','20-feb-2021') ;

1 row created.

SQL> insert into Emp_Appnt values(1278,'xyz','05-dec-1978','02-mar-2017') ;

1 row created.

SQL> insert into emp_appnt values(1789,'mon','06-nov-1999','22-mar-2021');

1 row created.
```

SQL> select * from emp_appnt ;

```
SQL> select * from emp_appnt ;

    ACC_NO NAME       RECDATE   RETDATE
---------- ---------- --------- ---------
      1235 Aakash Pal 08-MAR-87 12-OCT-15
      1235 Aakash Pal 08-MAR-87 12-OCT-15
      1237 ac         25-JAN-88 20-FEB-21
      1278 xyz        05-DEC-78 02-MAR-17
      1789 mon        06-NOV-99 22-MAR-21
      1235 Aakash Pal 08-MAR-87 12-OCT-15
      1235 Alpa       08-OCT-78 19-NOV-20
      1237 ac         25-JAN-88 20-FEB-21
      1278 xyz        05-DEC-78 02-MAR-17
      1789 mon        06-NOV-99 22-MAR-21
```

Queries:

i. Show the employee whose record date is 25th Jan-1988.
select * from emp_appnt where RECDate='25-jan-1988'

```
SQL>    select * from emp_appnt where RECDate='25-jan-1988';

    ACC_NO NAME       RECDATE   RETDATE
---------- ---------- --------- ---------
      1237 ac         25-JAN-88 20-FEB-21
      1237 ac         25-JAN-88 20-FEB-21
```

ii. Show the employee whose record date is 22th mar-2021.

select * from emp_appnt where RETDate='22-mar-2021';

```
SQL> select * from emp_appnt where RETDate='22-mar-2021';

    ACC_NO NAME       RECDATE   RETDATE
---------- ---------- --------- ---------
      1789 mon        06-NOV-99 22-MAR-21
      1789 mon        06-NOV-99 22-MAR-21
```

iii. Create a new table named as tbl_shares.

create table tbl_shares1(
C_Name varchar2(10), No_Share Number(10), Price number(10), TransTime varchar2(10) Default To_char(sysdate,'HH:MI'));

```
SQL> create table tbl_shares1 (
  2  C_Name varchar2(10), No_Share Number(10), Price number(10), TransTime varchar2(10) Default To_char(sysdate,'HH:MI'));
```

Inserting rows :

insert into tbl_shares1 values('Aakash', 123,500,Default);
insert into tbl_shares1 values('Alpa', 121,550,Default);
insert into tbl_shares1 values('VIK', 124,600,Default);
insert into tbl_shares1 values('RAJ', 125,750,Default);
insert into tbl_shares1 values('SAK', 133,1000,Default);

```
SQL> insert into tbl_shares1 values('Aakash', 123,500,Default);

1 row created.

SQL> insert into tbl_shares1 values('Alpa', 121,550,Default);

1 row created.

SQL> insert into tbl_shares1 values('VIK', 124,600,Default);

1 row created.

SQL> insert into tbl_shares1 values('RAJ', 125,750,Default);

1 row created.

SQL> insert into tbl_shares1 values('SAK', 133,1000,Default);

1 row created.
```

iv. Display all the records you have entered in table.

select * from tbl_shares;

```
SQL>  select * from tbl_shares1;

C_NAME       NO_SHARE      PRICE TRANSTIME
---------- ---------- ---------- ----------
Aakash            123        500 03:09
Alpa              121        550 03:09
VIK               124        600 03:09
RAJ               125        750 03:09
SAK               133       1000 03:09
```

v. Display records where price>100 and TransTime='01:24'.

select * from tbl_shares where price>100 and TransTime='01:24';

```
SQL> select * from tbl_shares1 where price>100 and TransTime='03:09';

C_NAME       NO_SHARE      PRICE TRANSTIME
---------- ---------- ---------- ----------
Aakash            123        500 03:09
Alpa              121        550 03:09
VIK               124        600 03:09
RAJ               125        750 03:09
SAK               133       1000 03:09
```

vi. Display the recordswhere price=(select max(price) from tbl_shares where TransTime='02:04');

select * from tbl_shares1 where price=(select max(price) from tbl_shareswhere TransTime='01:25');

```
SQL> select * from tbl_shares1 where price=(select max(price) from tbl_shares1 where TransTime='03:09');

C_NAME       NO_SHARE      PRICE TRANSTIME
---------- ---------- ---------- ----------
SAK               133       1000 03:09
```

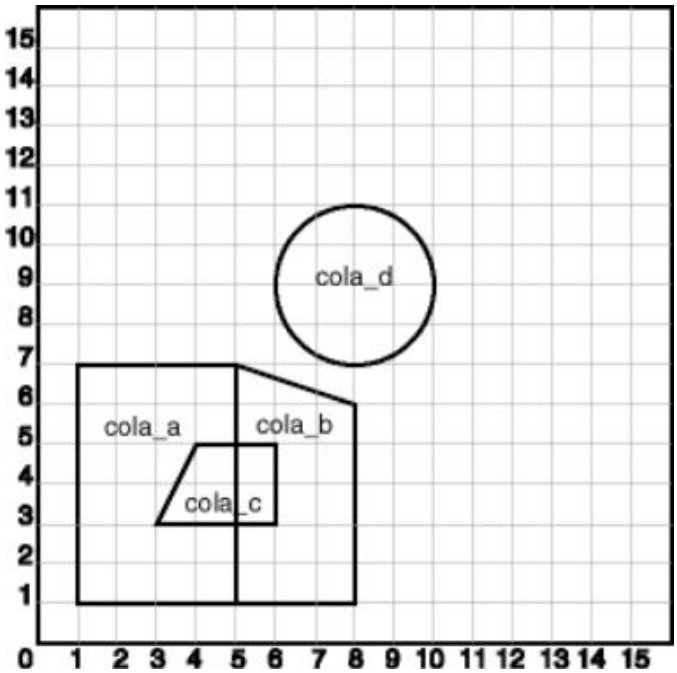
Conclusion: We have successfully created a temporal database.

PRACTICAL 6

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

Theory:

Spatial data, also known as geospatial data, is information that is associated with specific geographic locations on the Earth's surface. It represents data that has a spatial or geographical component, which can be expressed in terms of coordinates, shapes, boundaries, or distances. Spatial data is used to describe and analyze the physical world, including the locations of objects, features, and phenomena in the context of geography and cartography.



Query:

Create a spatial database table that stores the number, name and location, which consists offourdifferent 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.

Steps:

```
Enter user-name: system
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

Query for Creating Table

```
SQL>create table cola_mrp(
        mkt_id number primary key, name varchar(20), shapeMDSYS.SDO_Geometry
        );
```

Queries for inserting rows :

1)

```
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  (1,'cola_a',MDSYS.SDO_GEOMETRY(
  3  2003,NULL,NULL,
  4  MDSYS.SDO_ELEM_INFO_ARRAY
  5  (1,1003,3),
  6  MDSYS.SDO_ORDINATE_ARRAY
  7  (1,1,5,7)))
  8  /

1 row created.
```

2)

```
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
  2  (2,'cola_b',MDSYS.SDO_GEOMETRY(
  3  2003,NULL,NULL,
  4  MDSYS.SDO_ELEM_INFO_ARRAY
  5  (1,1003,1),
  6  MDSYS.SDO_ORDINATE_ARRAY
  7  (5,1,8,1,8,6,5,7,5,1)))
  8  /

1 row created.
```

3)

```
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
  2  (3,'cola_c',MDSYS.SDO_GEOMETRY(
  3  2003,NULL,NULL,
  4  MDSYS.SDO_ELEM_INFO_ARRAY
  5  (1,1003,1),
  6  MDSYS.SDO_ORDINATE_ARRAY
  7  (3,3,6,3,6,5,4,5,3,3)))
  8  /

1 row created.
```

4)

```
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)))
        /
```

```
SQL> insert into cola_mrp values
  2  (4,'cola_d',
  3  MDSYS.SDO_GEOMETRY(
  4  2003,NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY
  6  (1,1003,4),
  7  MDSYS.SDO_ORDINATE_ARRAY
  8  (7,9,10,9,8,11)))
  9  /

1 row created.
```

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);

```
SQL> insert into user_SDO_GEOM_METADATA values
  2  ('cola_mrp','shape',
  3  MDSYS.SDO_DIM_ARRAY(
  4  MDSYS.SDO_DIM_ELEMENT('X',0,20,0.005),
  5  MDSYS.SDO_DIM_ELEMENT('Y',0,20,0.005)
  6  ),NULL);

1 row created.
```

Query for creating index :

SQL>create index cola_spatial_idx oncola_market(location) Indextype Is
mdsys.spatial_index;

```
SQL> create index cola_spatial_idx
  2  on cola_market(location)
  3  Indextype Is mdsys.spatial_index;
on cola_market(location)
   *
ERROR at line 2:
ORA-00942: table or view does not exist
```

Queries :

1) Find the topological intersection of two geometries.

SQL>select SDO_GEOM.SDO_INTERSECTION (c_a.shape,c_c.shape,0.005)
fromcola_mrpc_a,cola_mrpc_c
where c_a.name='cola_a' AND c_c.name='cola_c';

```
SQL> select SDO_GEOM.SDO_INTERSECTION (c_a.shape,c_c.shape,0.005)
  2  from cola_mrp c_a,cola_mrp c_c
  3  where c_a.name='cola_a' AND c_c.name='cola_c';

SDO_GEOM.SDO_INTERSECTION(C_A.SHAPE,C_C.SHAPE,0.005)(SDO_GTYPE, SDO_SRID, SDO_PO

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

SDO_GEOMETRY(2003, NULL, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1), SDO_ORDINATE_ARR

AY(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_mrpc_c, cola_mrpc_a
WHERE c_c.name='cola_c' AND c_a.name = 'cola_a';

```
SQL> SELECT SDO_GEOM.RELATE(c_c.shape, 'EQUAL', c_a.shape,0.005)
  2  FROM cola_mrp c_c, cola_mrp c_a
  3  WHERE c_c.name='cola_c' AND c_a.name = 'cola_a';

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;

```
SQL> select name,SDO_GEOM.SDO_AREA(shape,0.005) from  cola_mrp;

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 cwhere c.name='cola_a';

```
SQL> select c.name,SDO_GEOM.SDO_AREA(c.shape,0.005) from cola_mrp c
  2  where c.name='cola_a';

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)
fromcola_mrpc_b,cola_mrpc_d
where c_b.name= 'cola_b' AND c_d.name ='cola_d';

```
SQL> select SDO_GEOM.SDO_DISTANCE(c_b.shape,c_d.shape,0.005)
  2  from cola_mrp c_b,cola_mrp c_d
  3  where c_b.name= 'cola_b' AND c_d.name ='cola_d';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                       1.8973666
```

Conclusion:

We have successfully created a Spatial database.

PRACTICAL 7

Aim:- Create a table employee having dept_id as number datatype and employee_spec as XML data type (XML_Type). The employee_spec is a schema with attributes emp_id, name, email, acc_no, managerEmail, dataOf Joning. Insert 10 tuples into employee table. Fire the following queries on XML database:

Insert 10 tuples into employee table. Fire the following queries on XML database.

1. Retrieve the names of employee.
2. Retrieve the acc_no of employees.
3. Retrieve the names, acc_no, and email of employees.
4. Update the 3rd record from the table and display the name of an employee.
5. Delete 4th record from the table

Theory:

An XML database is a type of database management system (DBMS) designed specifically for the storage, retrieval, and management of XML (eXtensible Markup Language) data. XML is a text-based data format commonly used for representing structured and semi-structured data, making it suitable for various applications, including web services, data interchange, and document representation.

Steps:

Create a table:

```
create table emp (emp_id int, emp_spec xmltype);
```

Insert records:

```
Insert into emp values (1,xmltype('<?xml version="1.0"?>
```

```
    <employee id="be130">
        <firstname>William</firstname>
        <lastname>Defoe</lastname>
        <title>Accountant</title>
        <division>Accts Payable</division>
        <building>326</building>
        <room>14a</room>
    </employee>'));

Insert into emp values (2,xmltype('<?xml version="1.0"?>
  <employee id="be129">
        <firstname>Jane</firstname>
        <lastname>Doe</lastname>
        <title>Engineer</title>
        <division>Materials</division>
        <building>327</building>
        <room>19</room>
        <supervisor>be131</supervisor>
    </employee>'));


Insert into emp values (3,xmltype('<?xml version="1.0"?>
  <employee id="be129">
        <firstname>Jane</firstname>
        <lastname>Doe</lastname>
        <title>Engineer</title>
        <division>Materials</division>
        <building>327</building>
        <room>19</room>
        <supervisor>be131</supervisor>
    </employee>'));
```

```
Insert into emp values (4,xmltype('<?xml version="1.0"?>
    <employee id="be132">
        <firstname>Sandra</firstname>
        <lastname>Rogers</lastname>
        <title>Engineering</title>
        <division>Materials</division>
        <building>327</building>
        <room>22</room>
    </employee>'));

Insert into emp values (5,xmltype('<?xml version="1.0"?>
    <employee id="be133">
        <firstname>Steve</firstname>
        <lastname>Casey</lastname>
        <title>Engineering</title>
        <division>Materials</division>
        <building>327</building>
        <room>24</room>
    </employee>'));


Insert into emp values (6,xmltype('<?xml version="1.0"?>
    <employee id="be135">
        <firstname>Michelle</firstname>
        <lastname>Michaels</lastname>
        <title>COO</title>
        <division>Management</division>
        <building>216</building>
        <room>264</room>
    </employee>'));
```

```
SQL> Insert into emp values (1,xmltype('<?xml version="1.0"?>
  2      <employee id="be130">
  3          <firstname>William</firstname>
  4          <lastname>Defoe</lastname>
  5          <title>Accountant</title>
  6          <division>Accts Payable</division>
  7          <building>326</building>
  8          <room>14a</room>
  9      </employee>'));

1 row created.

SQL>
SQL> Insert into emp values (2,xmltype('<?xml version="1.0"?>
  2    <employee id="be129">
  3          <firstname>Jane</firstname>
  4          <lastname>Doe</lastname>
  5          <title>Engineer</title>
  6          <division>Materials</division>
  7          <building>327</building>
  8          <room>19</room>
  9          <supervisor>be131</supervisor>
 10      </employee>'));

1 row created.

SQL>
SQL>
SQL> Insert into emp values (3,xmltype('<?xml version="1.0"?>
  2    <employee id="be129">
  3          <firstname>Jane</firstname>
  4          <lastname>Doe</lastname>
  5          <title>Engineer</title>
  6          <division>Materials</division>
  7          <building>327</building>
  8          <room>19</room>
  9          <supervisor>be131</supervisor>
 10      </employee>'));

1 row created.

SQL>
SQL> Insert into emp values (4,xmltype('<?xml version="1.0"?>
  2      <employee id="be132">
  3          <firstname>Sandra</firstname>
  4          <lastname>Rogers</lastname>
  5          <title>Engineering</title>
  6          <division>Materials</division>
  7          <building>327</building>
  8          <room>22</room>
  9      </employee>'));

1 row created.
```

```
SQL> set wrap off
SQL> select * from emp;

    EMP_ID EMP_SPEC
---------- ----------------------------------------
         1 <?xml version="1.0"?>
         2 <?xml version="1.0"?>
         3 <?xml version="1.0"?>
         4 <?xml version="1.0"?>
         5 <?xml version="1.0"?>
         6 <?xml version="1.0"?>
```

Get the first name:

select x.emp_spec.extract('//firstname/text() ').getStringVal() from emp x;

```
SQL> select x.emp_spec.extract('//firstname/text() ').getStringVal() emp_name from emp x;

EMP_NAME
--------------------------------------------------------------------------------
William
Jane
Jane
Sandra
Steve
Michelle

6 rows selected.
```

Get the first name and room number:

select x.emp_spec.extract('//firstname/text() ').getStringVal() emp_name, x.emp_spec.extract('//room/text()').getStringVal() room_No from emp x;

```
EMP_NAME
--------------------
ROOM_NO
--------------------
Sandra
22

Steve
24

Michelle
264
```

Get the first name and room number and title:

```
select x.emp_spec.extract('//firstname/text() ').getStringVal() emp_name,
x.emp_spec.extract('//room/text()').getStringVal() room_No from emp x;
```

```
EMP_NAME
--------------------------------------------------------------------
ROOM_NO
--------------------------------------------------------------------
TITLE
--------------------------------------------------------------------
William
14a
Accountant

Jane
19
Engineer

EMP_NAME
```

Update 3rd record from the table:

```
Update emp set emp_spec=xmltype('<?xml version="1.0"?>
  <employee id="be135">
    <firstname>NotMichelle</firstname>
    <lastname>NotMichaels</lastname>
    <title>COO</title>
    <division>Management</division>
    <building>216</building>
    <room>264</room>
  </employee>') where emp_id=3;
```

```
SQL> Update emp set emp_spec=xmltype('<?xml version="1.0"?>
  2      <employee id="be135">
  3          <firstname>NotMichelle</firstname>
  4          <lastname>NotMichaels</lastname>
  5          <title>COO</title>
  6          <division>Management</division>
  7          <building>216</building>
  8          <room>264</room>
  9      </employee>') where emp_id=3;

1 row updated.
```

```
EMP_SPEC
--------------------------------------------------------------
<?xml version="1.0"?>
    <employee id="be135">
        <firstname>NotMichelle</
```

Delete a record from the table:

Delete from emp where x.emp_spec.extract('//firstname/text() ').getStringVal() =”NotMichelle”;

```
SQL> Delete from emp x where x.emp_spec.extract('//firstname/text() ').getStringVal() ='NotMichelle';

1 row deleted.
```

Conclusion: We have successfully created an XML database.