M.Sc C.S - I
SEM I
E-Journal

Roll No.	027
Name	OJHA ABHISHEK DEVMANI
Subject	ADVANCED DATABASE SYSTEMS

Thakur Educational Trust's (Regd.)
Thakur College of Science & Commerce
UGC Recognised • Affiliated to University Of Mumbai
(NAAC Accredited with Grade "A" [3rd Cycle] & ISO 9001:2015 Certified)

CERTIFICATE

This is here to certify that Mr. **OJHA ABHISHEK DEVMANI**, Seat Number **027** of M.Sc. I Computer Science, has satisfactorily completed the required number of experiments prescribed by the UNIVERSITY OF MUMBAI during the academic year 2021 – 2022.

Date:
Place: Mumbai

Teacher In-Charge Head of Department

External Examiner

INDEX

Sr. No.	Practical Name	Date
1	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.	
2	Place the replication of global conceptual schema on different nodes and execute queries that will demonstrate distributed databases environment.	
3	CRUD operation using MongoDB.	
4	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. Fire suitable queries on them.	
5	Create a temporal database and issue queries on it.	
6	Create a table that stores the special data and issue queries on it.	
7	Create a table employee having dept_id as number datatype and employee_spec as XML data type (XM_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.	

PRACTICAL NO 1

Global Schema in DDB

Abhishek Ojha
Seat No 027

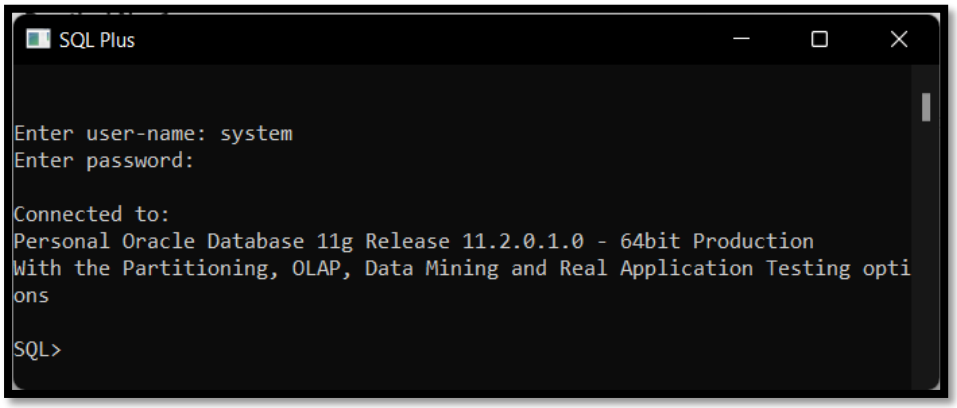
ADVANCED DATABASE SYSTEMS

Practical No: 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.

Software Requirement:

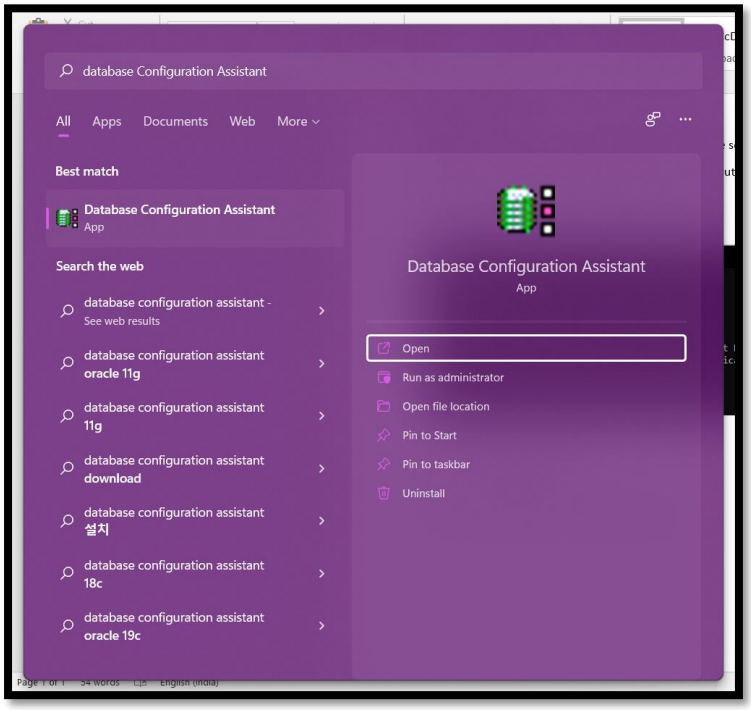
Oracle Database 11g.



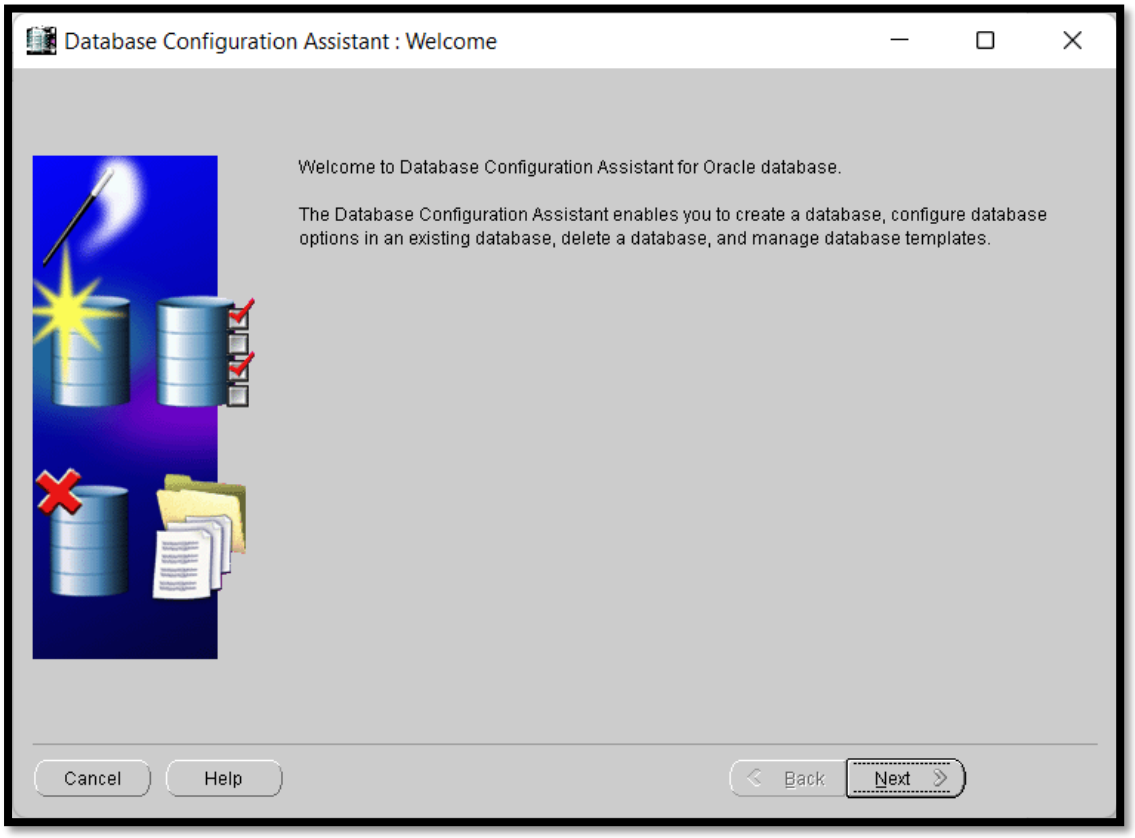
How to Create Two Database

Steps to Create Database db1 and db2

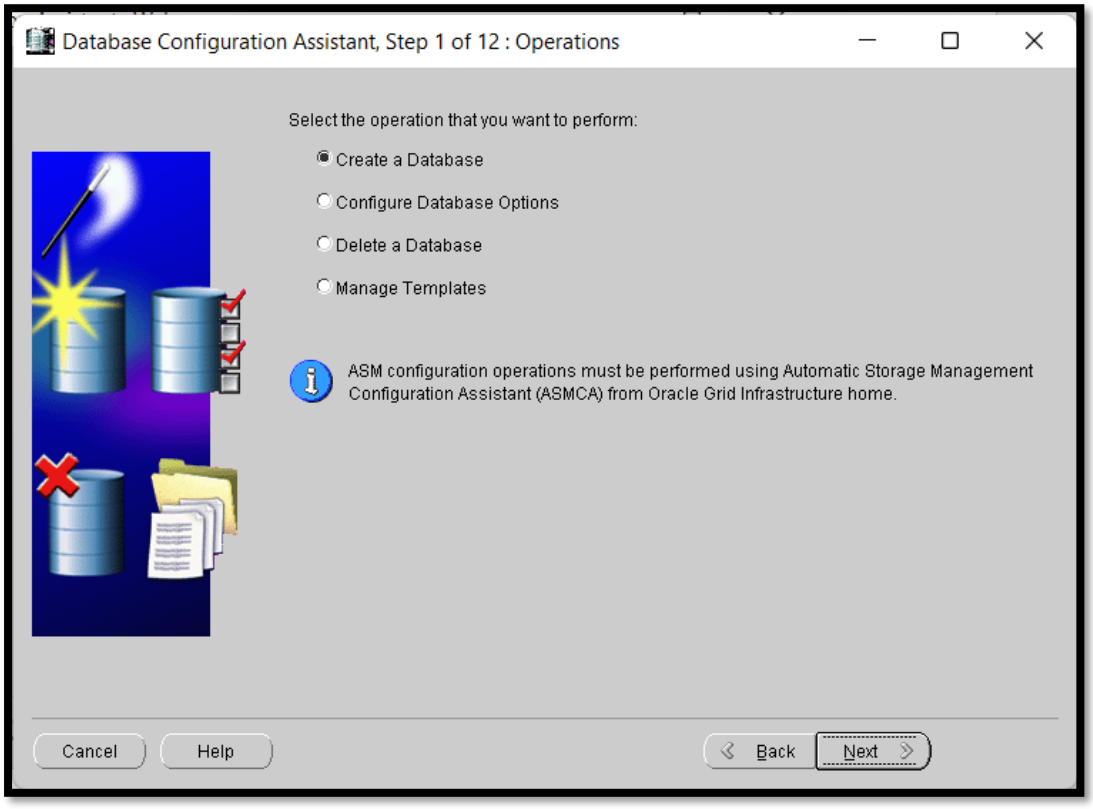
Step 1:- Open Start Menu on Window Explorer Go to Database Configuration Assistant.



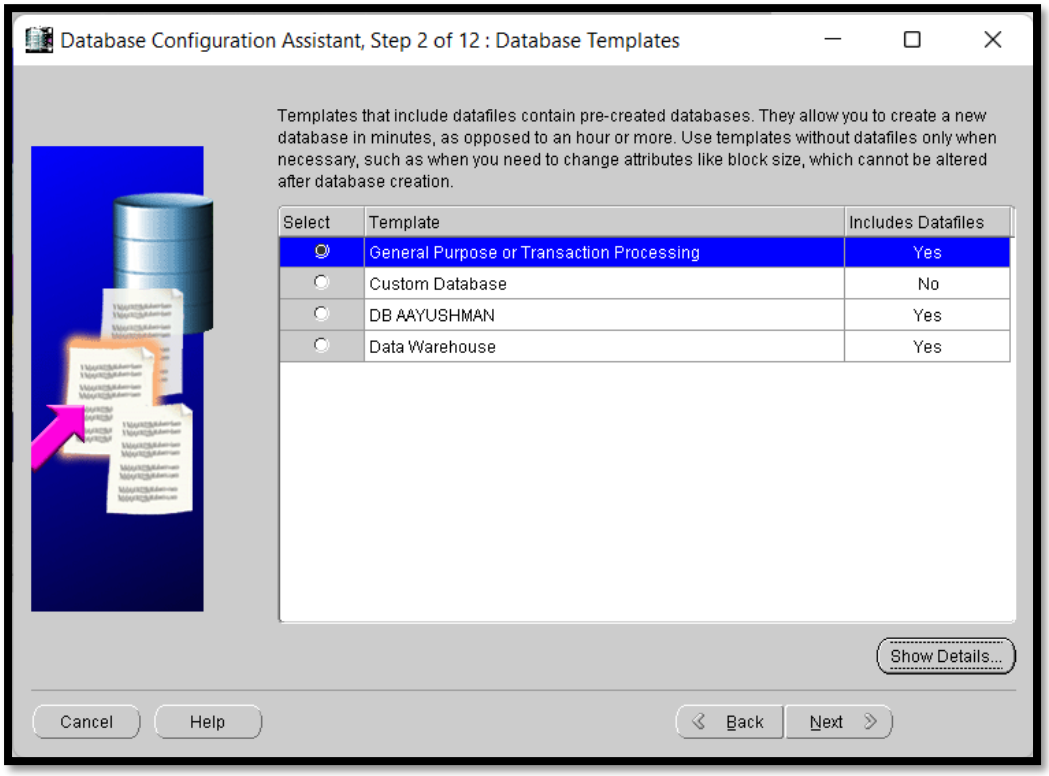
Step 2: Click on Next.



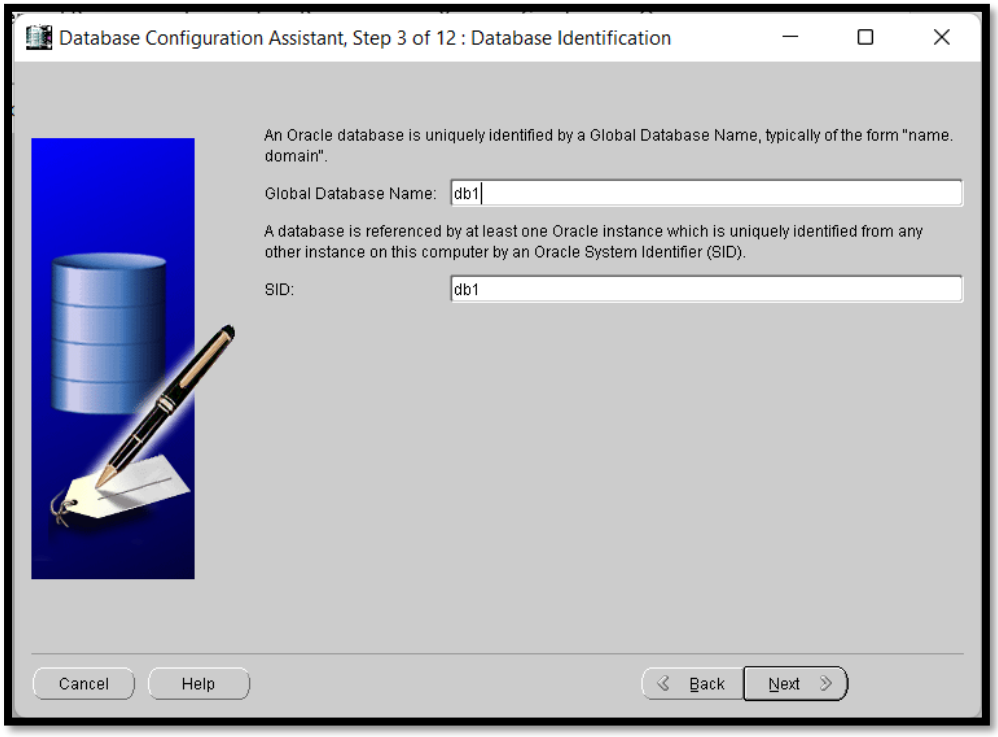
Step 3: Select Option Create a Database.

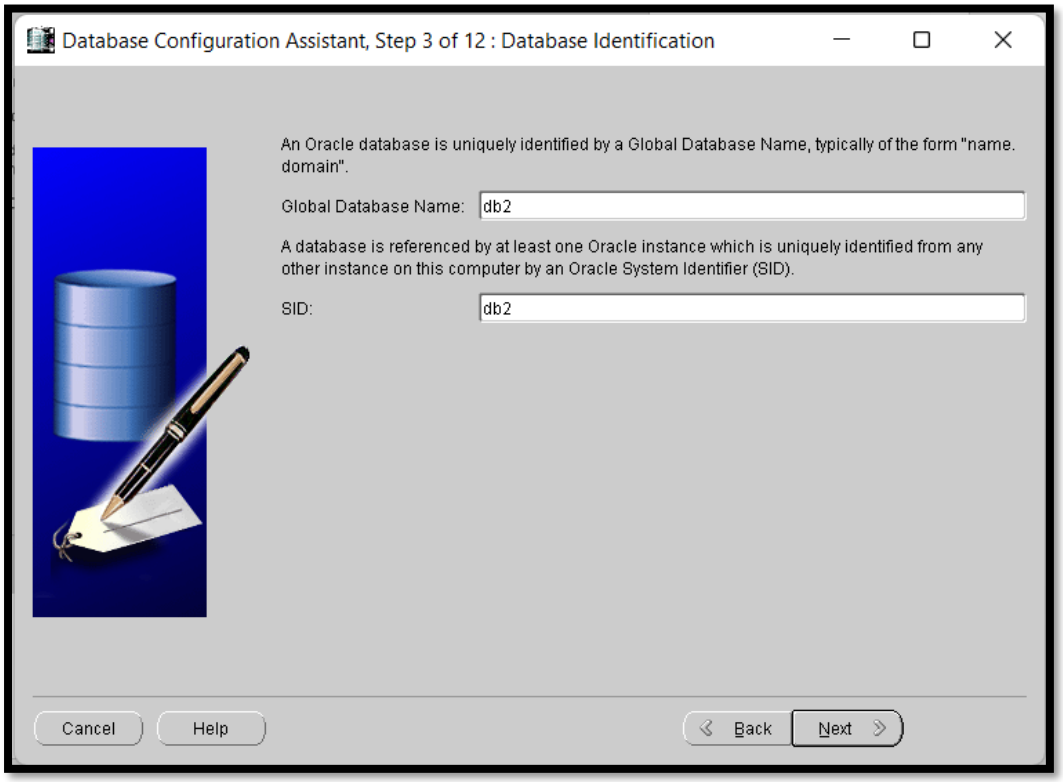


Step 4: Select Option General Purpose or Transaction Processing or You can Create your Own Custom Database.

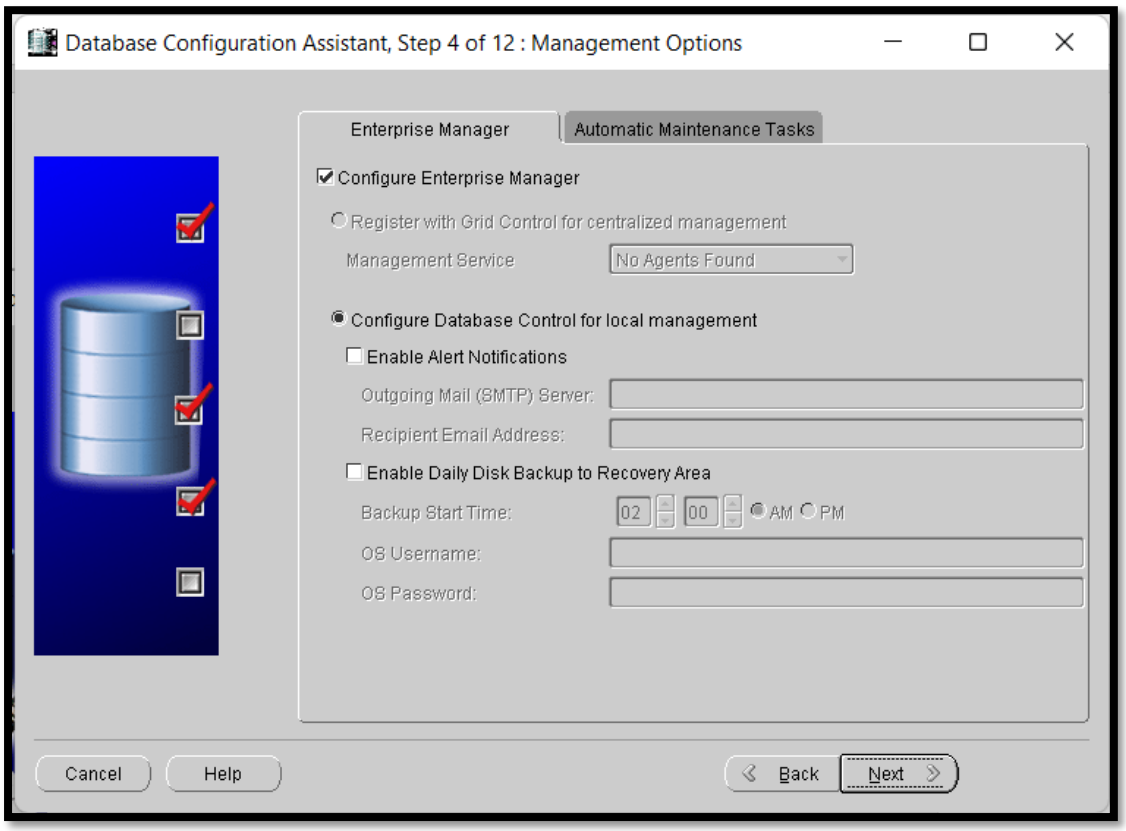


Step 5: Give Database Name as db1 (of your own choice).

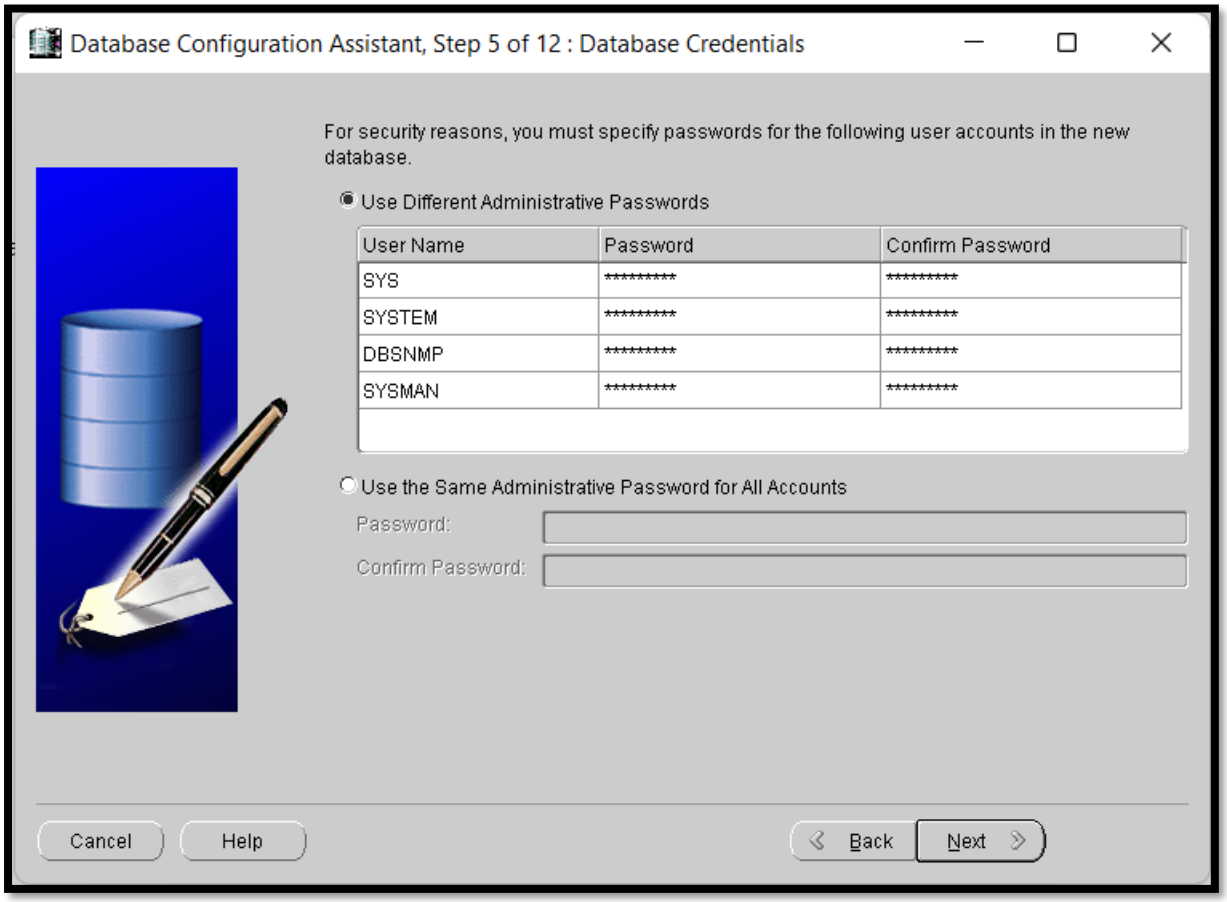




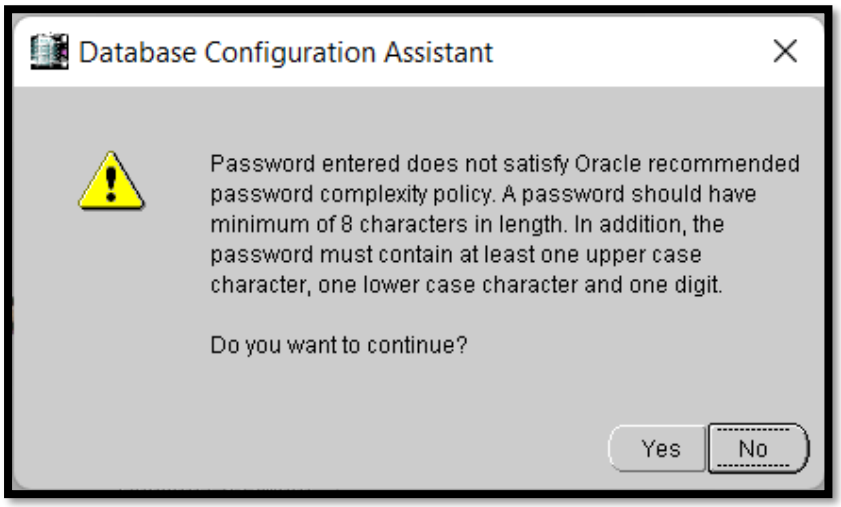
Step 6: No changes Needed, Click on Next.



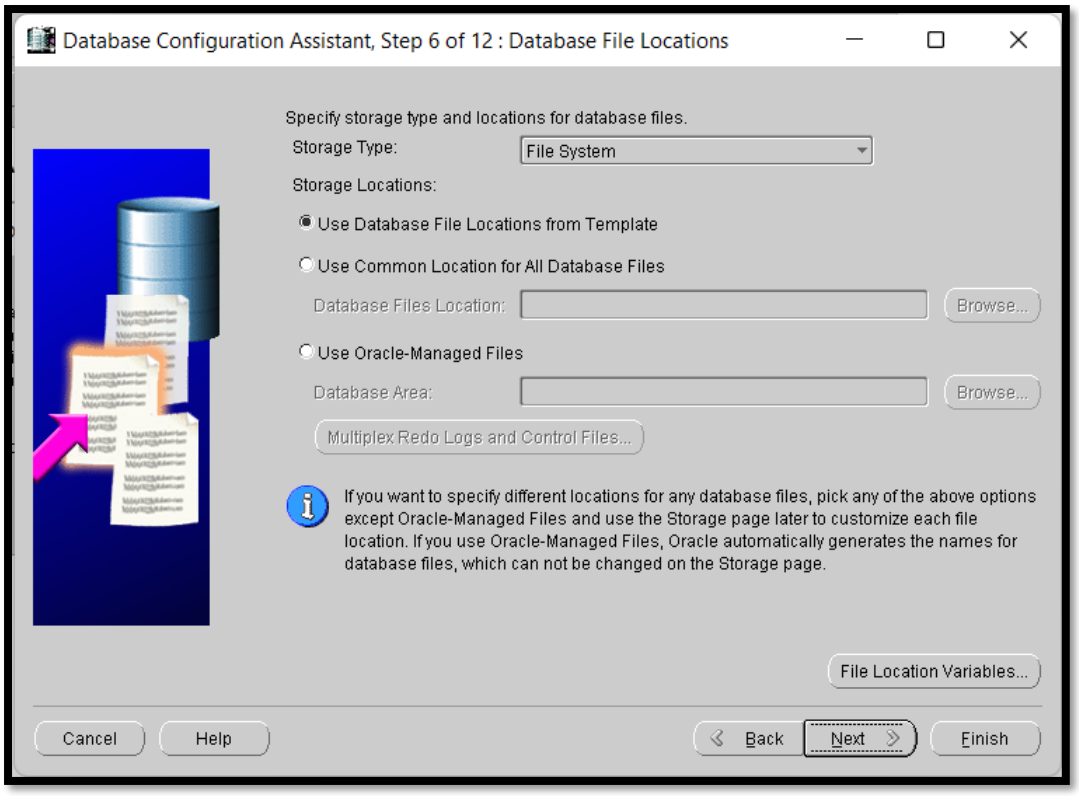
Step 7: Input Password of your choice for Each Fields or Else use your Administrator Credentials for all Profile.



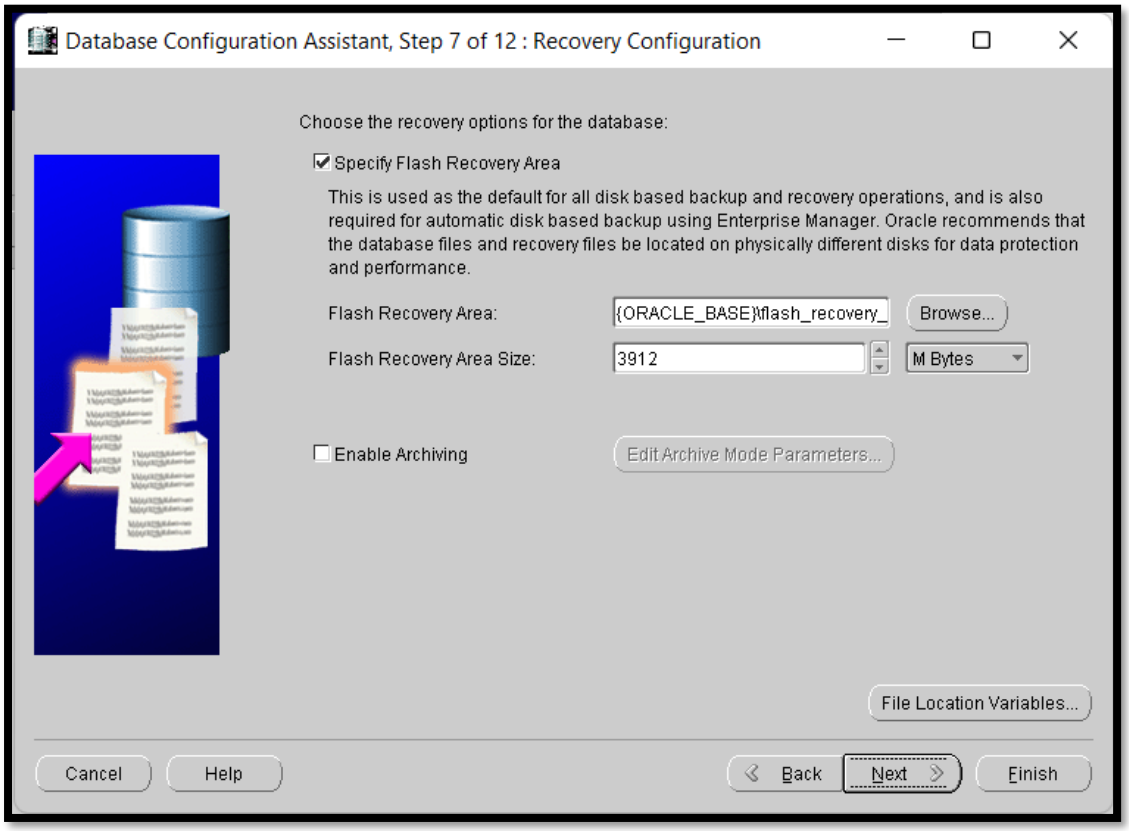
Checks for Password Confirmation, Just Click Yes.



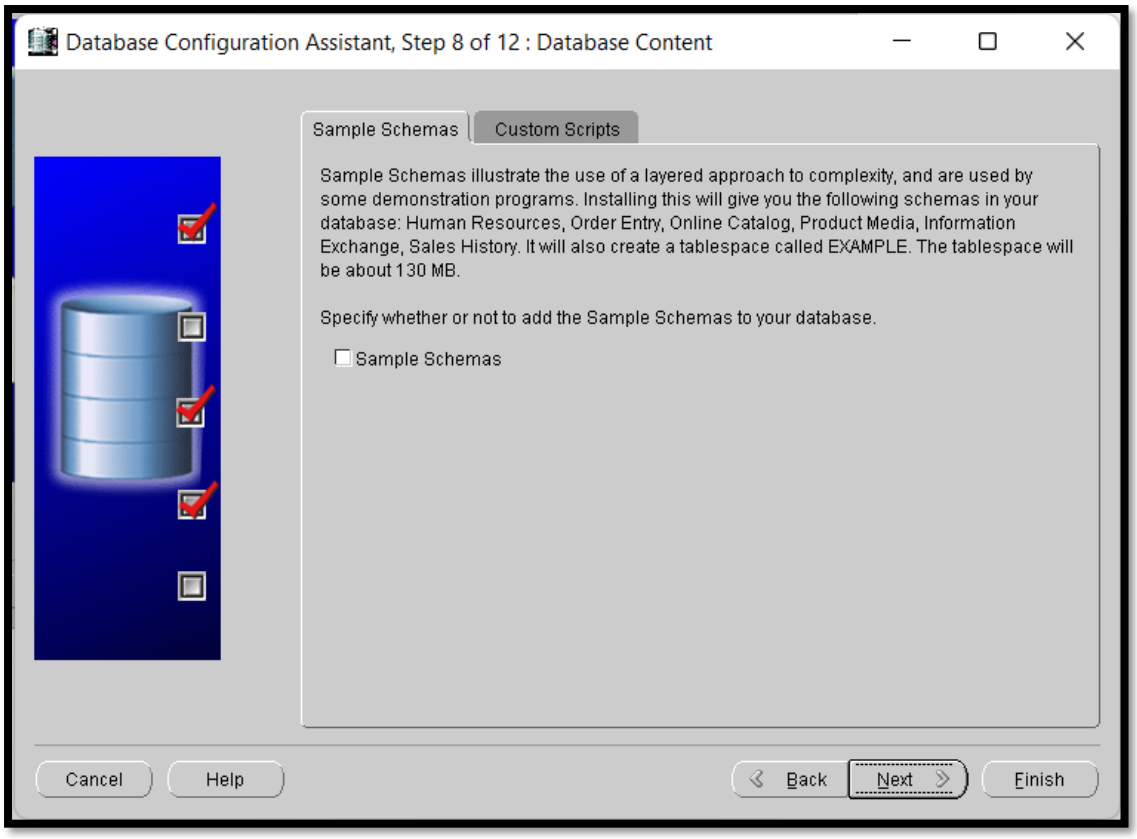
Step 8: No changes Needed, Click on Next.



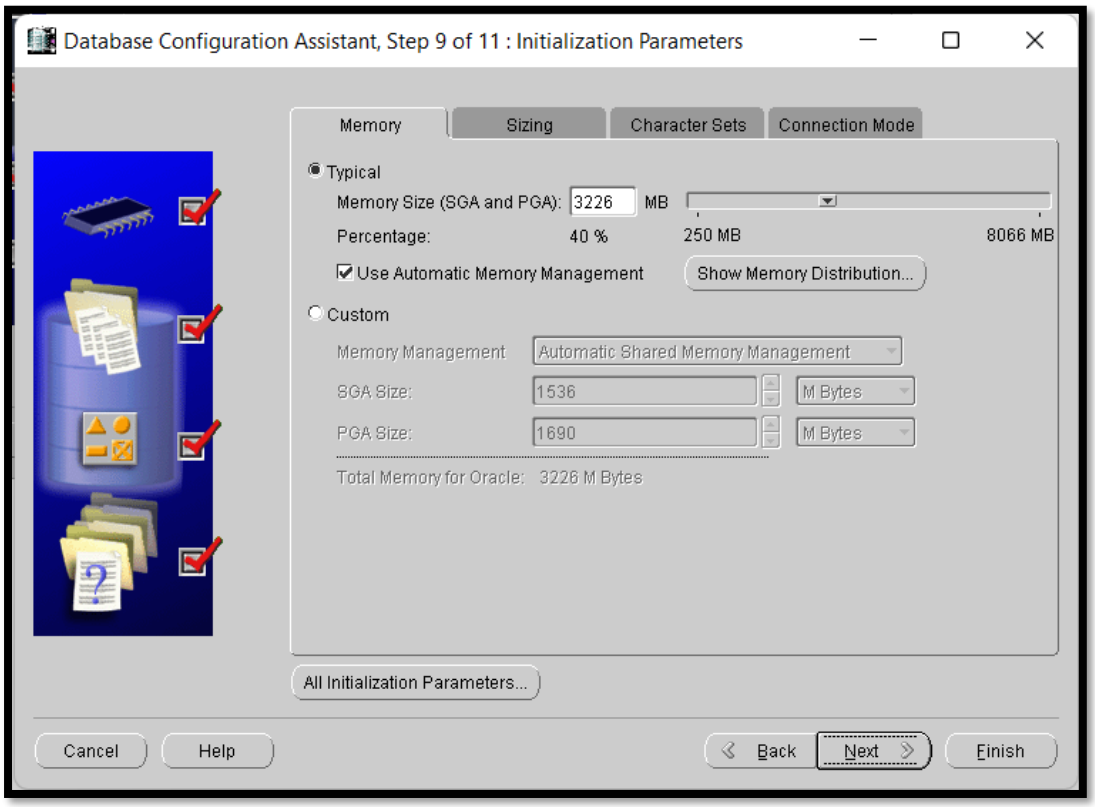
Step 9: No changes Needed, Click on Next.



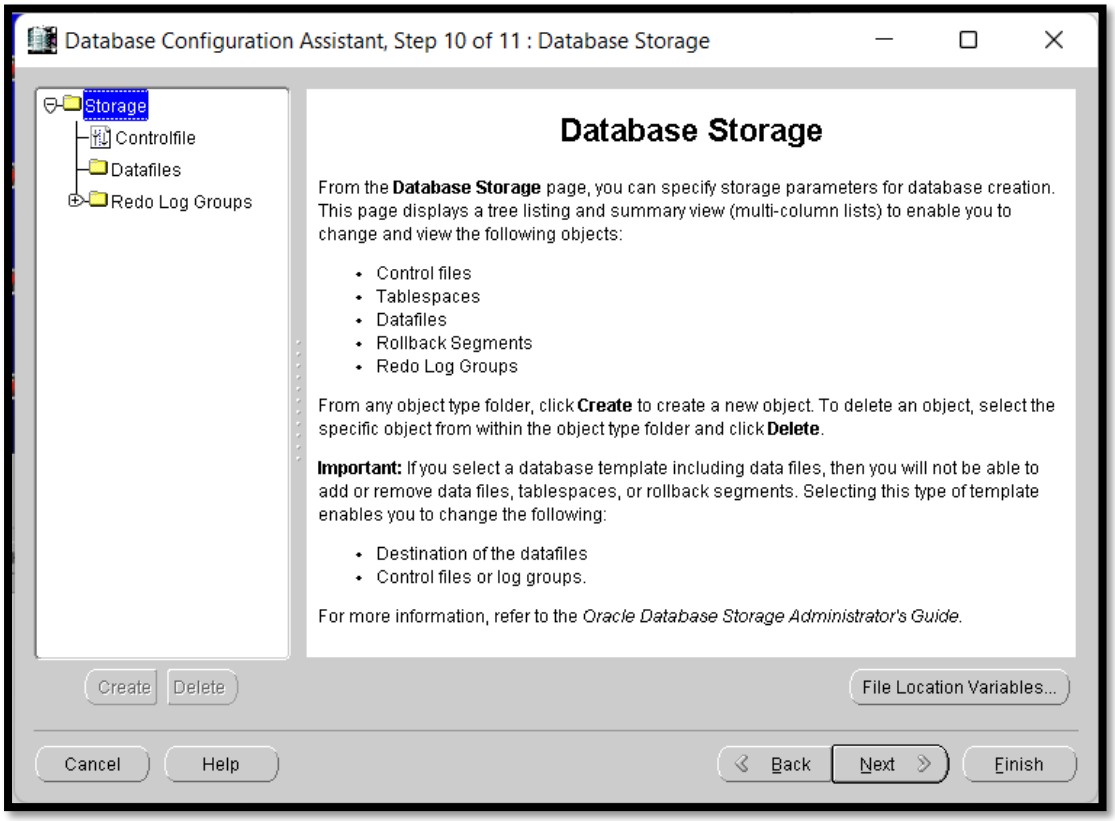
Step 10: No changes Needed, Click on Next.



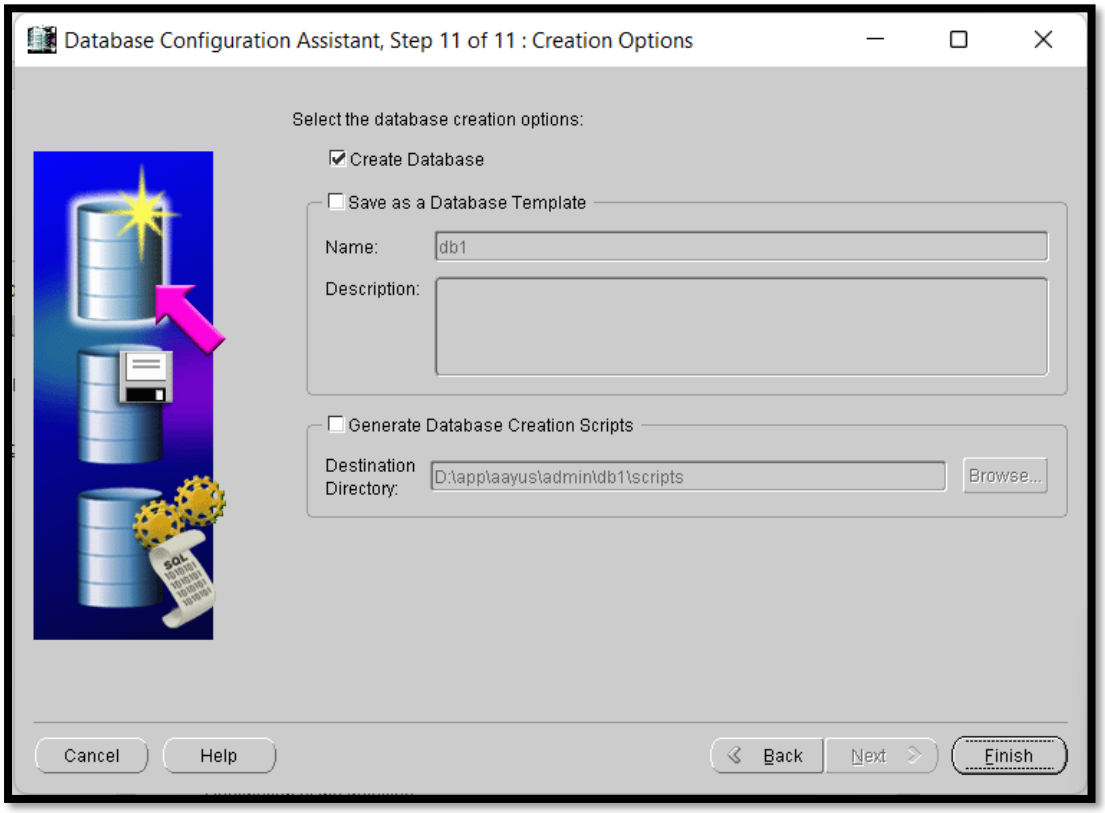
Step 11: No changes Needed, Click on Next.



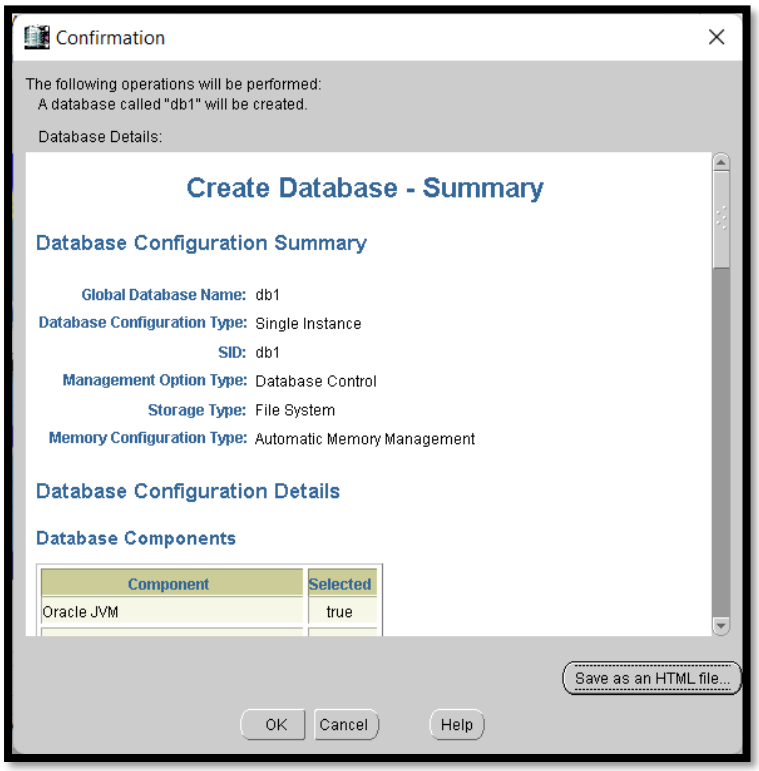
Step 12: No changes Needed, Click on Next..

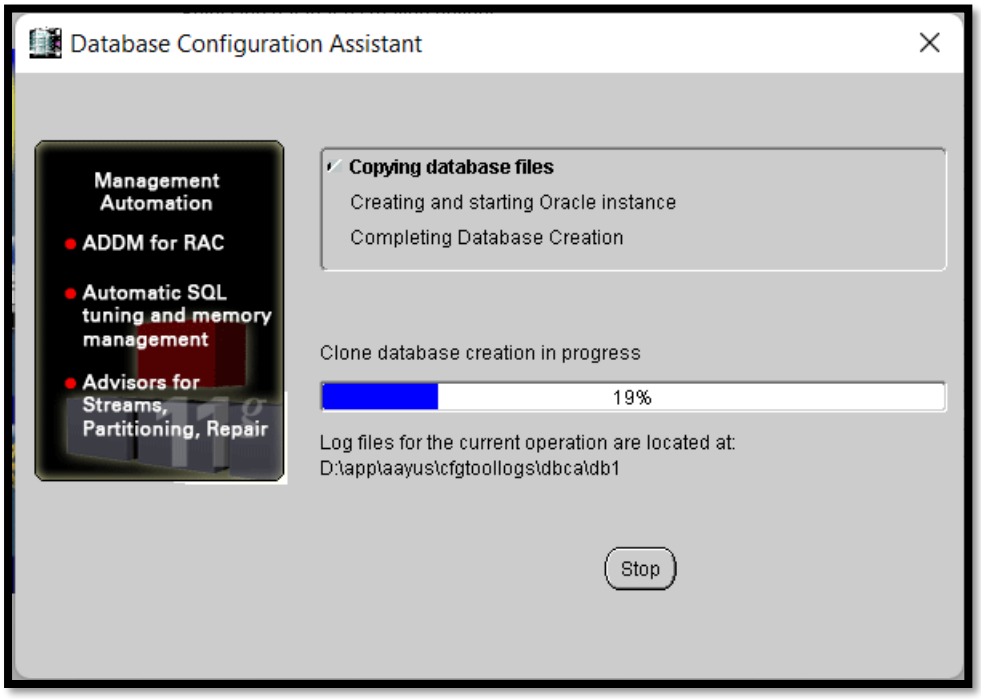


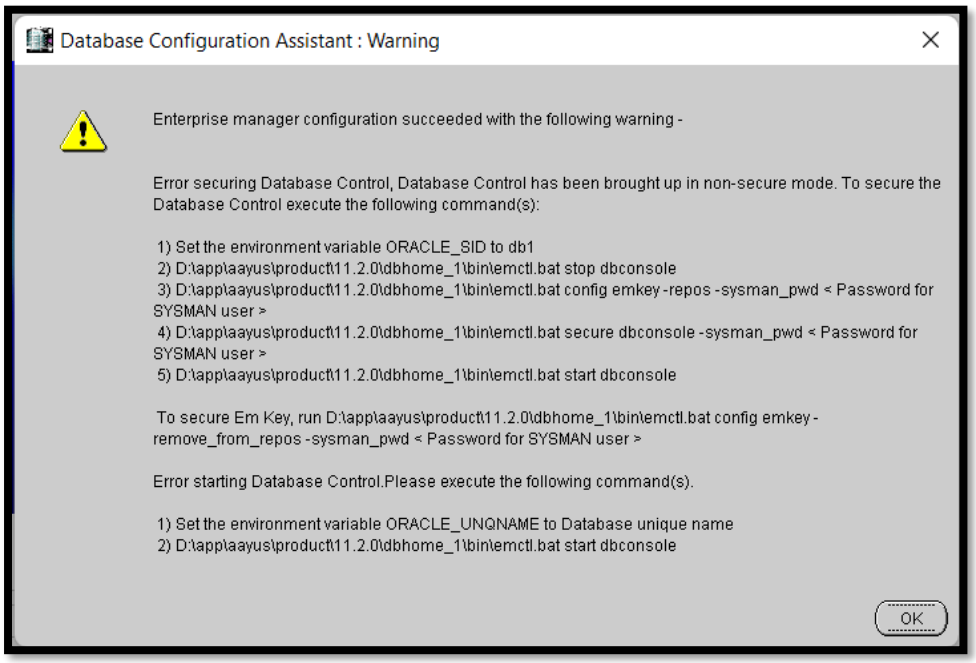
Step 13: No changes Needed, Click on Finish.



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.




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

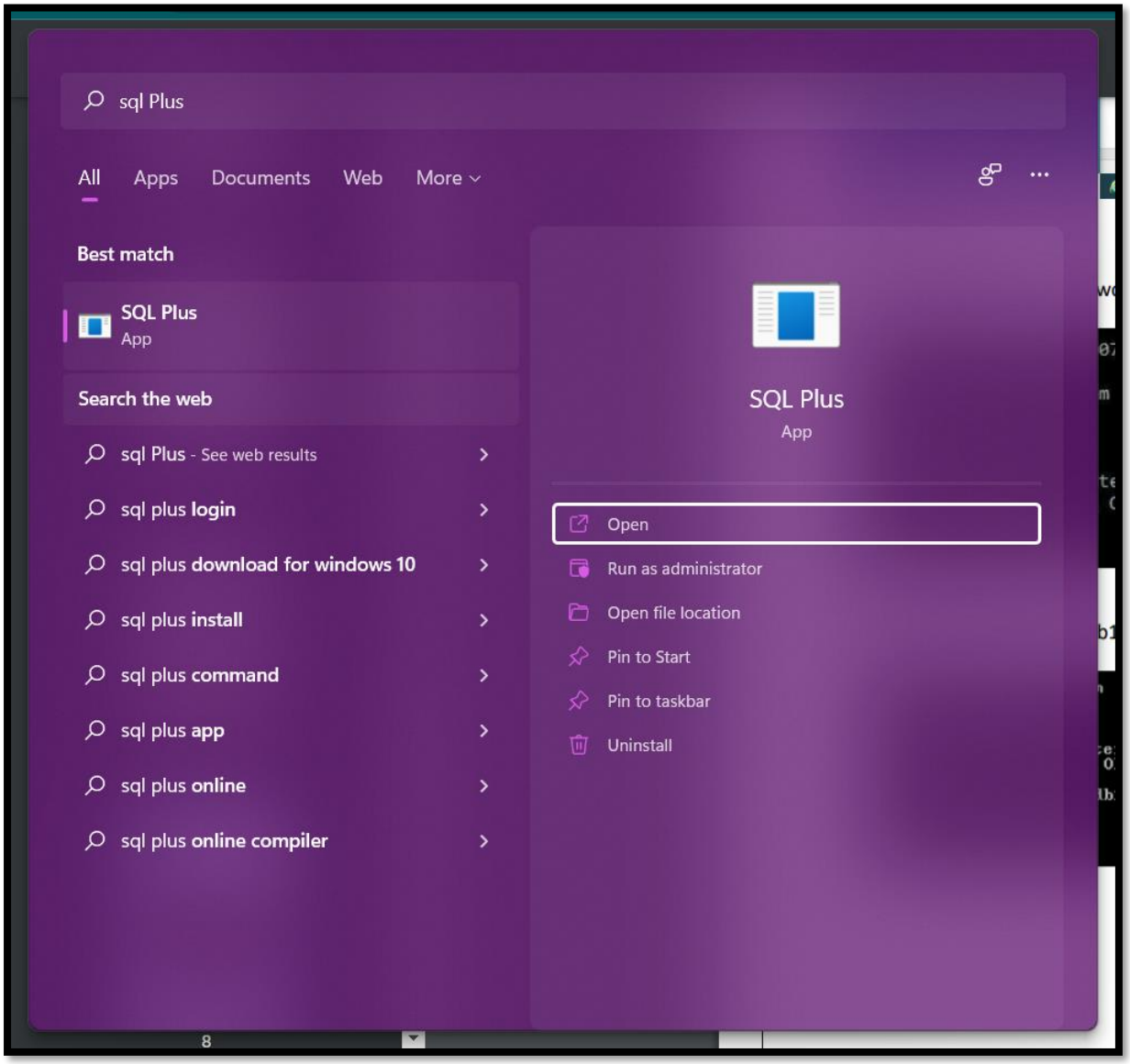
Click on Exit and Done…..

Follow the Same Steps to create db2,

Once done with Creating db1 and db2 .

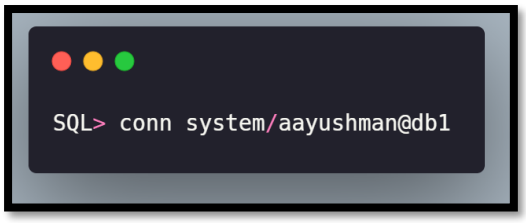
Practical Implementation Steps:

- ✓ **Step 1:-** Open SQLPlus.

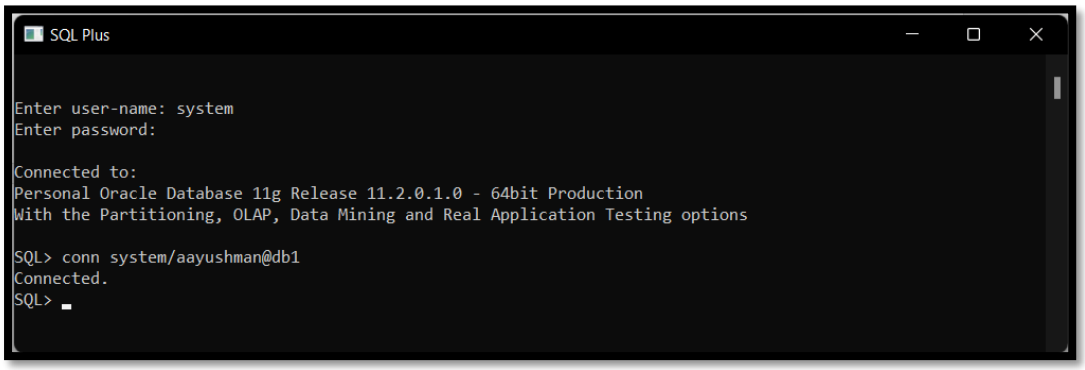


- ✓ **Step 2:** Connect to Your Database .

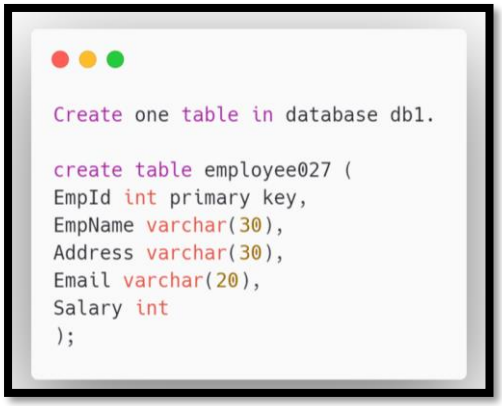
✓ **Step 3:** Connect your db1 While executing the Command



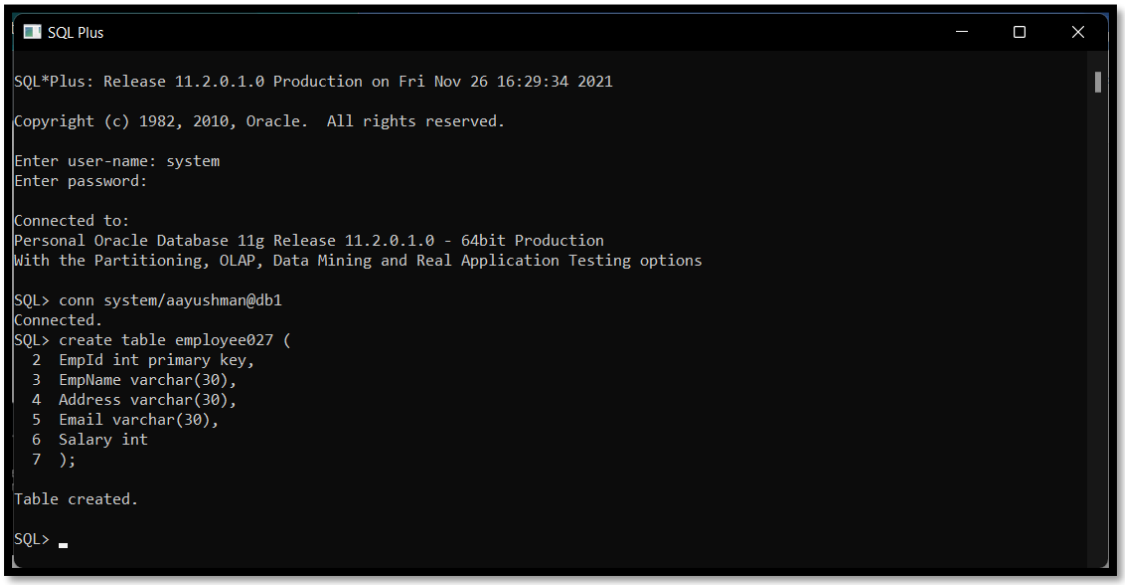
[Where "aayushman" is password of your database, and "db1" is database name].



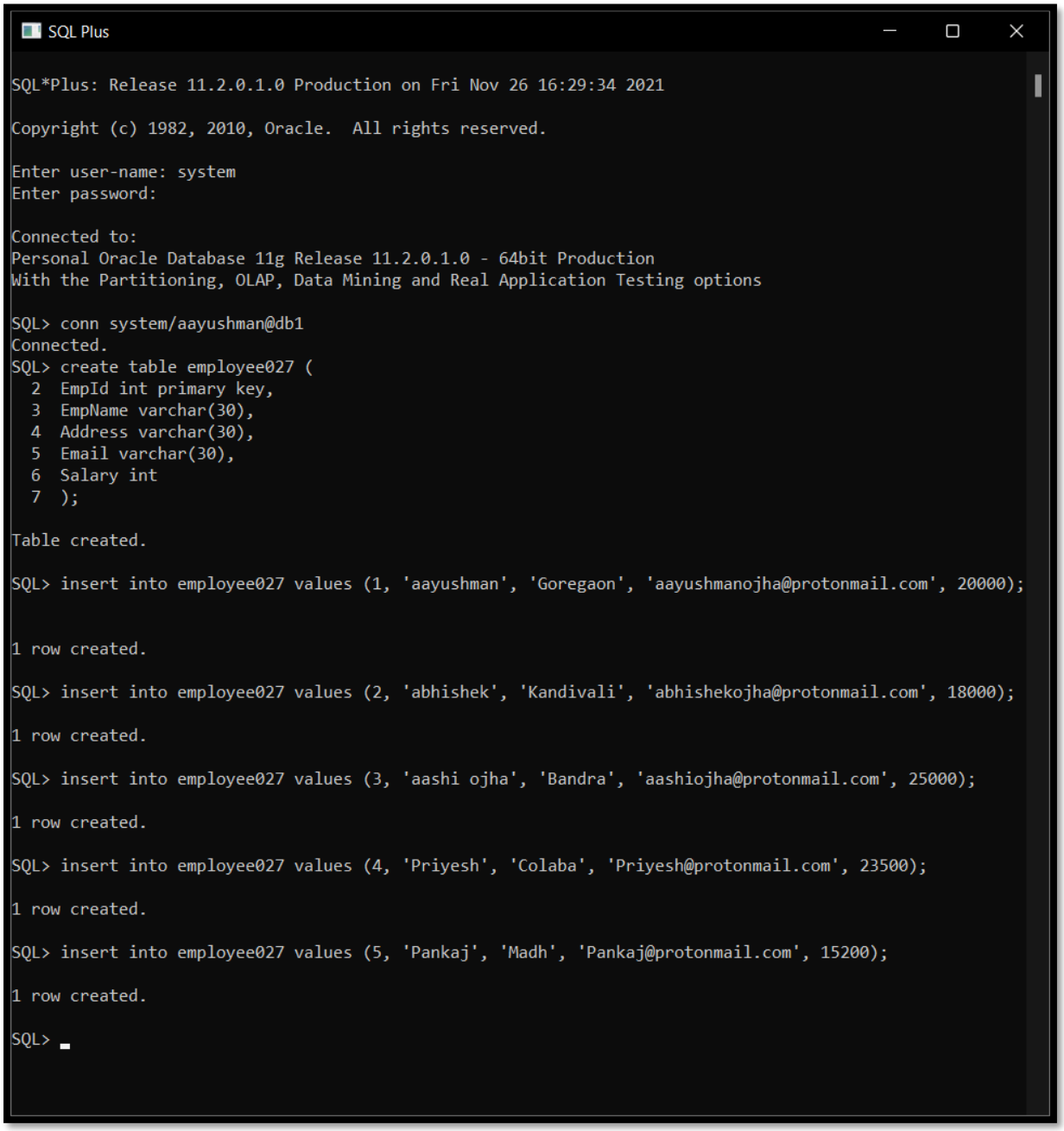
✓ **Step 4:** Create one table in database db1.



```
Create one table in database db1.

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

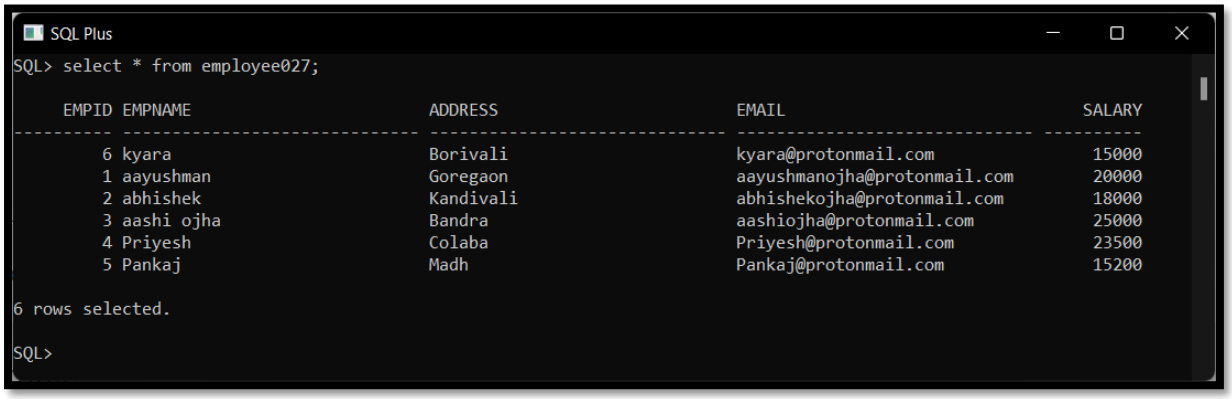


✓ **Step 5:** Insert Some values in Created Table.



✓ **Step 6:**

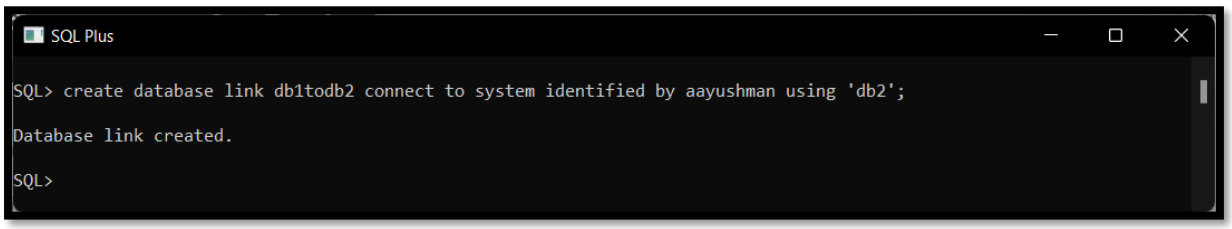
```
Show all tables in employee.

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>
```

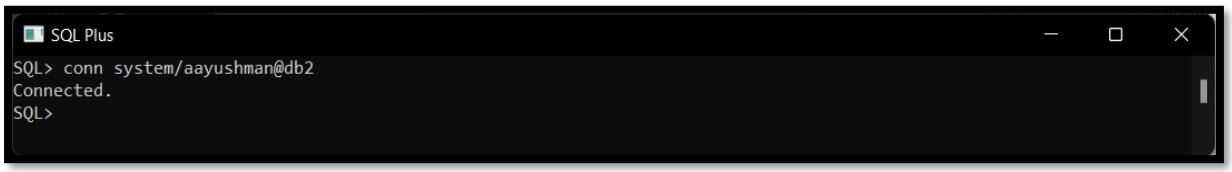
✓ **Step 7:** Enter following command to create link between two databases.

```
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>
```

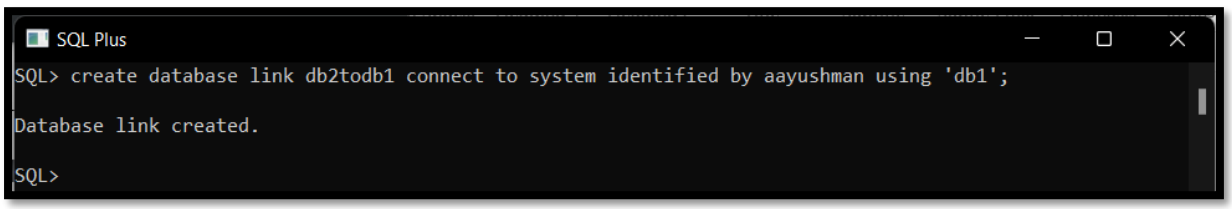
✓ **Step 8:** Connect to Db2.



```
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>
```

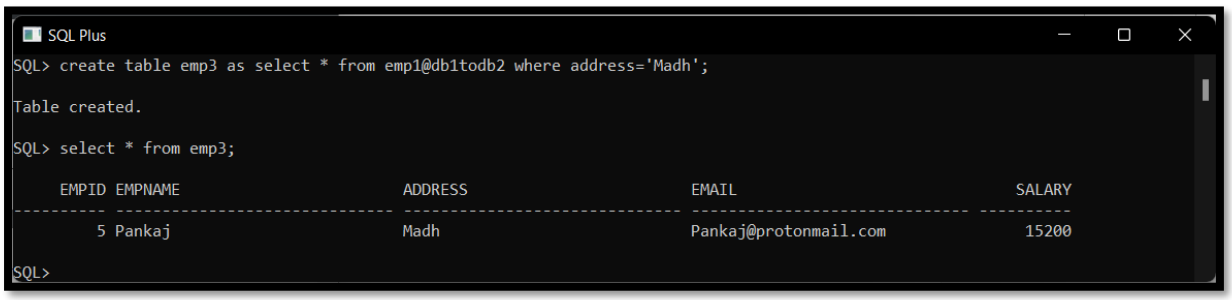
✓ **Step 14:** Select Employee Name and Email from Employee table where eid=2.

```
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';
```



Conclusion: Successfully Execution of Schema into horizontal and vertical Fragmentation on different nodes in Distributed Database Environment.

PRACTICAL NO 2

Triggers

Abhishek Ojha
Seat No 027

ADVANCED DATABASE SYSTEMS

Practical No 2

Aim:

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

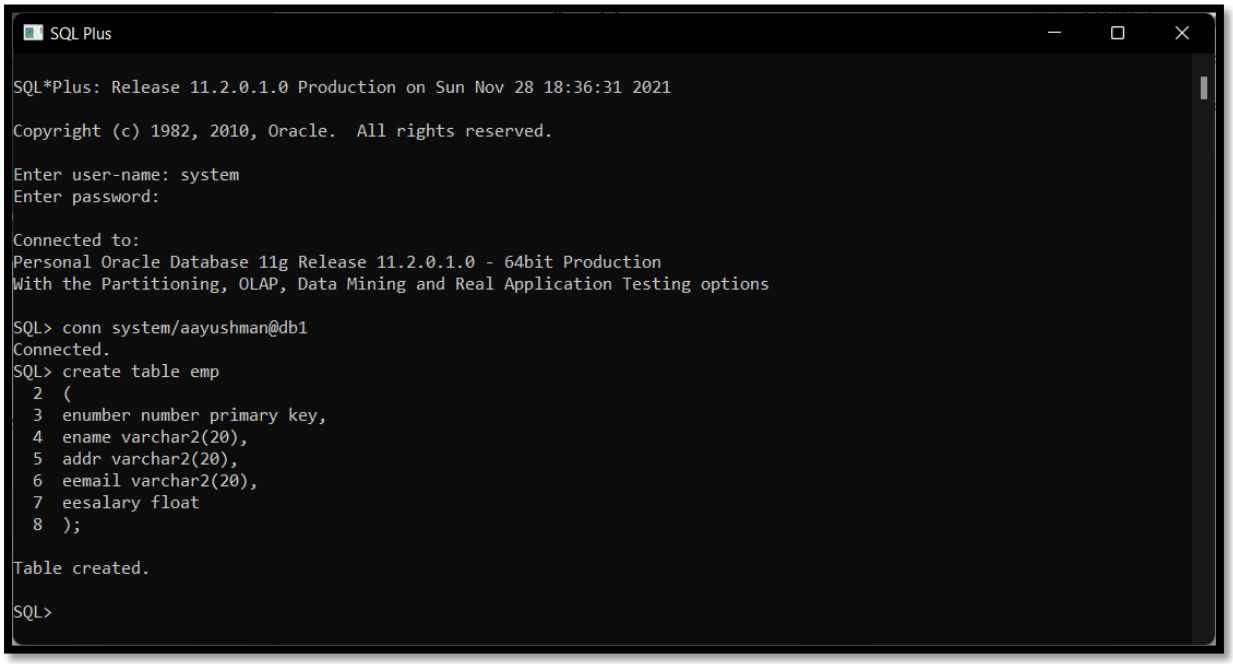
Software Requirement:

Oracle 11g.

Query:

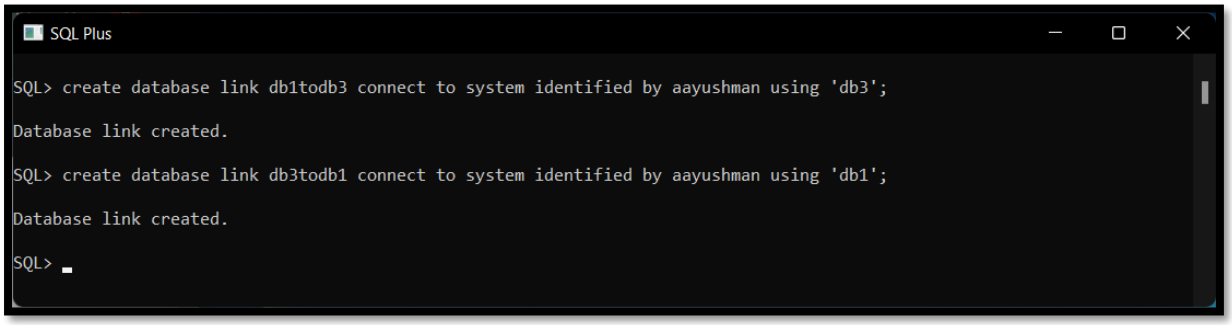
1. Update any record in db1 & show in db2
2. Delete any record in db1 & show in db2.
3. Find the salary of all employees.
4. Find the email of all employees where salary = 15000.
5. Find the employee name and email where employee number is known.
6. Find the employee name and address where employee number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

Step 1: Create Table in db1.

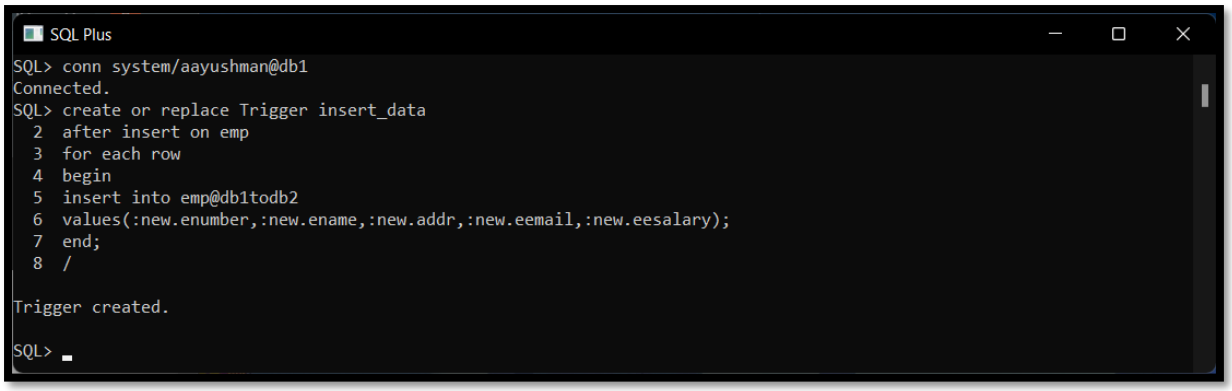


Step 2: Create Table in db2.

Step 3: Create Database link.

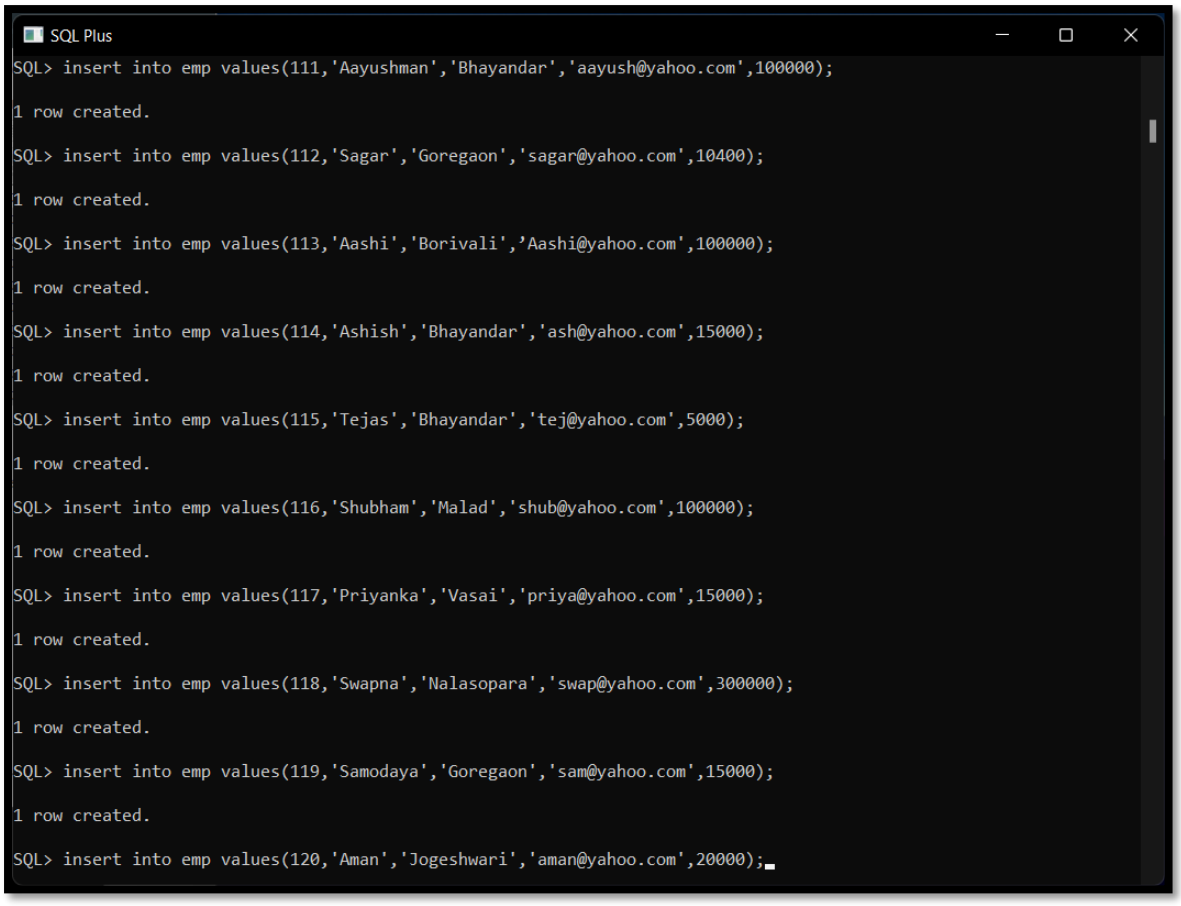


Step 4: Create Trigger to Insert Data.



Step 5: Create Trigger to Update Data in Table.

Step 6: Insert Values in Created Table.



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

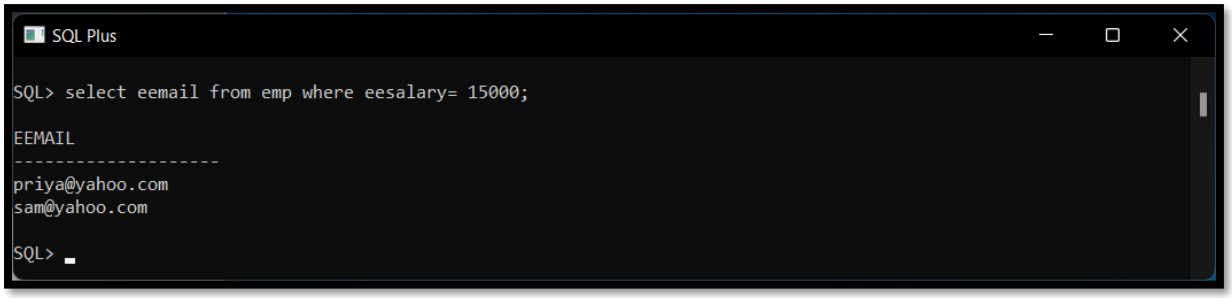
```
SQL> update emp
  2  set eesalary=800000
  3  where enumber=114;

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.

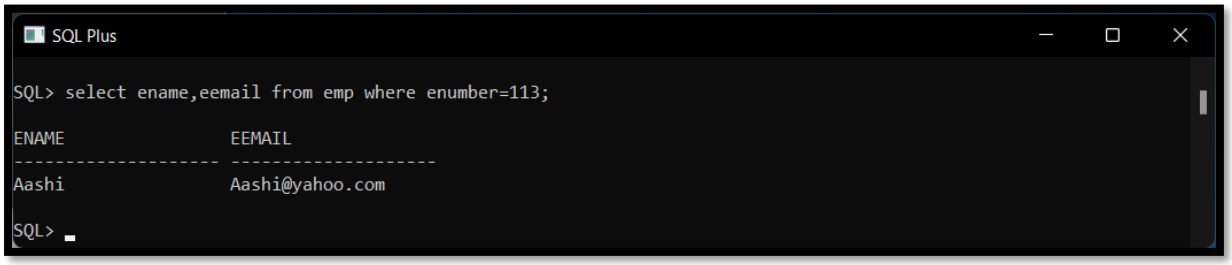
2. Delete any record in db1 & show in db2.

3. Find the salary of all employees.

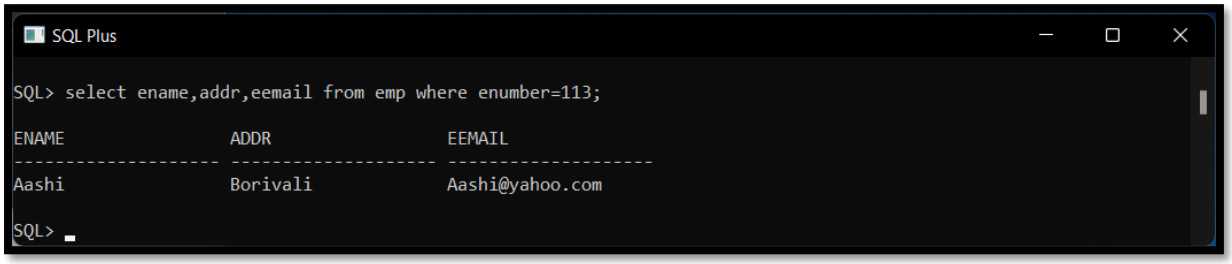
4. Find the email of all employees where salary = 15000.



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



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



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

PRACTICAL NO 3

CRUD Operation on MongoDB

Abhishek Ojha
Seat No 027

ADVANCED DATABASE SYSTEMS

Practical No: 3

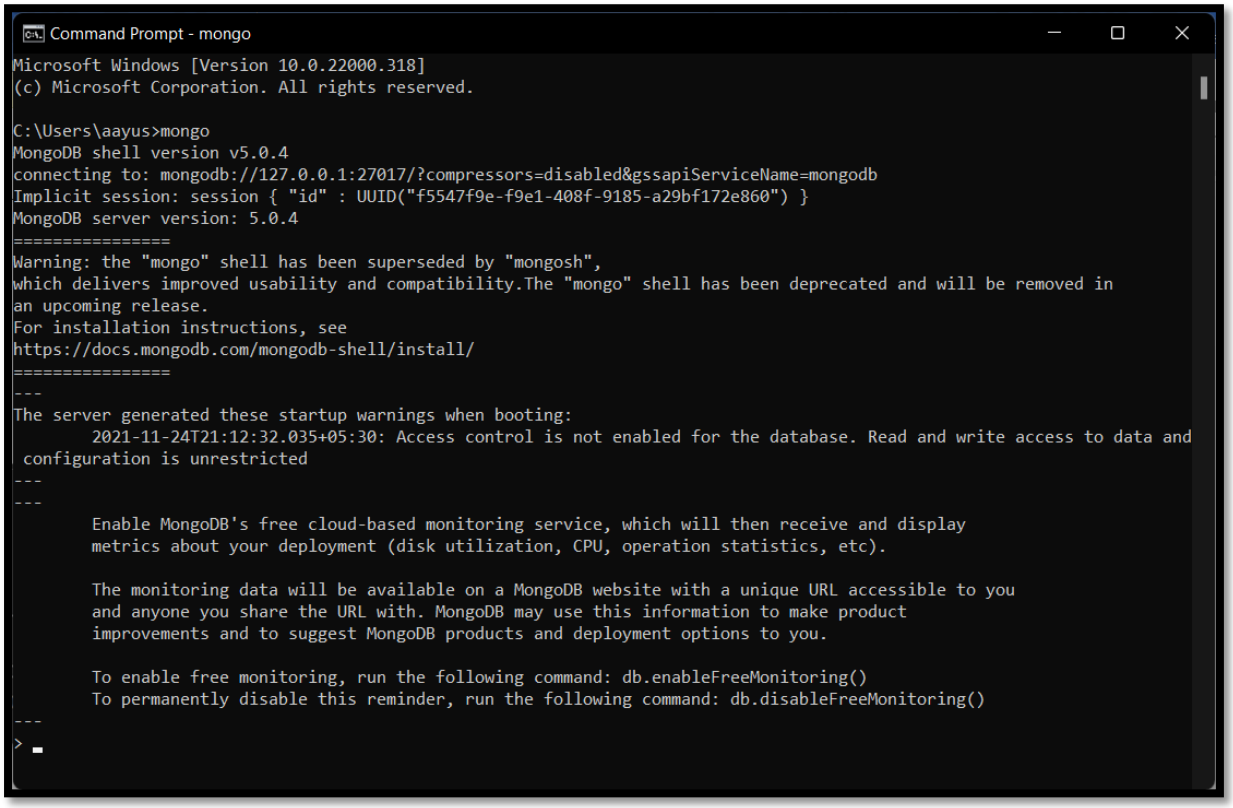
Aim: To perform CRUD Operation using MongoDB.

Software Requirement:

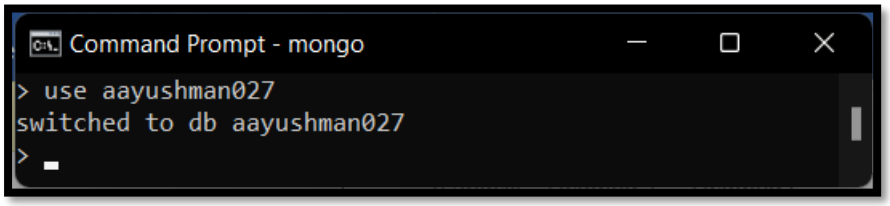
MongoDB.

Practical Implementation Steps :

- ✓ **Step 1:-** Open CMD and hit command "Mongo" [To directly run MongoDB from Command Prompt we need to First Set the Environment Variable for MongoDB].
- ✓ To set Environment Variable Follow the Steps:
  - ❖ Open C drive -> Program Files -> MongoDB -> server -> 5.0 -> bin C:\Program Files\MongoDB\Server\5.0\bin [Copy the Path].
  - ❖ Start -> Search For "Edit the System Environment Variable" -> Open.
  - ❖ Add the Copied Path in System Variable and done.

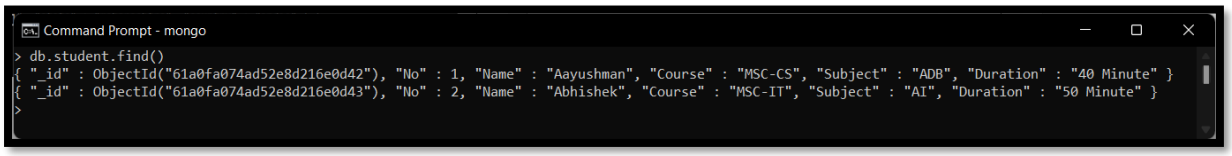


- ✓ **Step 2:** Creating and selecting database
  Command : use aayushman027 [i.e. aayushman027 is Database Name].
  Note: To list all Database use the command : Show dbs.



- ✓ **Step 3:** Creating Collections and Inserting Values **[C - Create]**
  Creating a collection and inserting values can be done together. Here we have orcollection name as 'student '.

- ✓ **Step 4:** Read Data from the Collections **[R - Read]**
  To retrieve the inserted document.



- ✓ **Step 5:** Updating a Document in a Collection [**U - Update**].

- ✓ **Step 6:** Removing an Entry From the Collection[**D- Delete**].

Conclusion: Successfully Performed and Implemented the CRUD Operation Using MongoDB.

PRACTICAL NO 4

CRUD Operations on Types

Abhishek Ojha
Seat No 027

ADVANCED DATABASE SYSTEMS

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.

Software Requirement:

Oracle 11g.

Steps:

1. AddrType1 (PinQuery: number, Street :char, City : char, state :char) .
2. BranchType (address: AddrType1, phone1: integer,phone2: integer).
3. AuthorType (name:char,,addr AddrType1).
4. PublisherType (name: char, addr: AddrType1, branches: BranchTableType.
5. books(title: varchar, year : date, published_by ref PublisherType,authorsAuthorListType).
6. Insert some records into the above tables and fire the following queries.

Query:

1. List all of the authors that have the same pin Query as their publisher.
2. List all books that have 2 or more authors.
3. List the name of the publisher that has the most branches.
4. List all authors who have published more than one Book.
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).

Practical Implementation Steps :

- ✓ **Step 1:-** AddrType1 (PinQuery: number, Street :char, City : char, state :char).

✓ **Step 2:-** BranchType (address: AddrType1, phone1: integer,phone2: integer).

✓ **Step 3:-** AuthorType (name:char,,addr AddrType1).

✓ **Step 4:-** PublisherType (name: char, addr: AddrType1, branches: BranchTableType.

```
SQL Plus
SQL> create or replace type PublisherType as object (
  2  Name varchar2(50),
  3  Address AddrType1,
  4  Branches BranchTableType
  5  );
  6  /

Type created.

SQL>
```

```
SQL Plus
SQL> create table Publishers of PublisherType NESTED TABLE Branches STORE as branchtable;

Table created.

SQL>
```

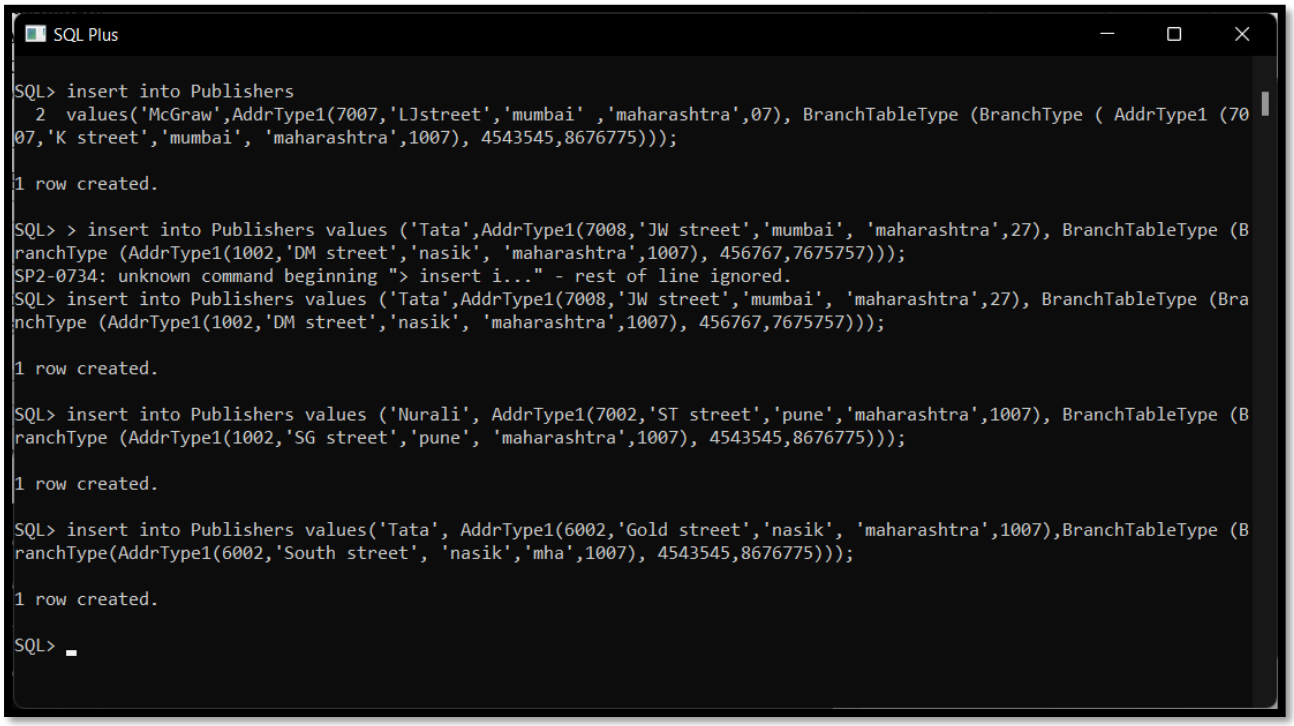
- ✓ **Steps 5**:- books(title: varchar, year : date, published_by ref PublisherType,authorsAuthorListType).

```
SQL Plus

SQL> create table books (
  2  Title varchar2(50),
  3  Year date,
  4  Published_by ref PublisherType,
  5  Authors AuthorListType
  6  );

Table created.

SQL>
```

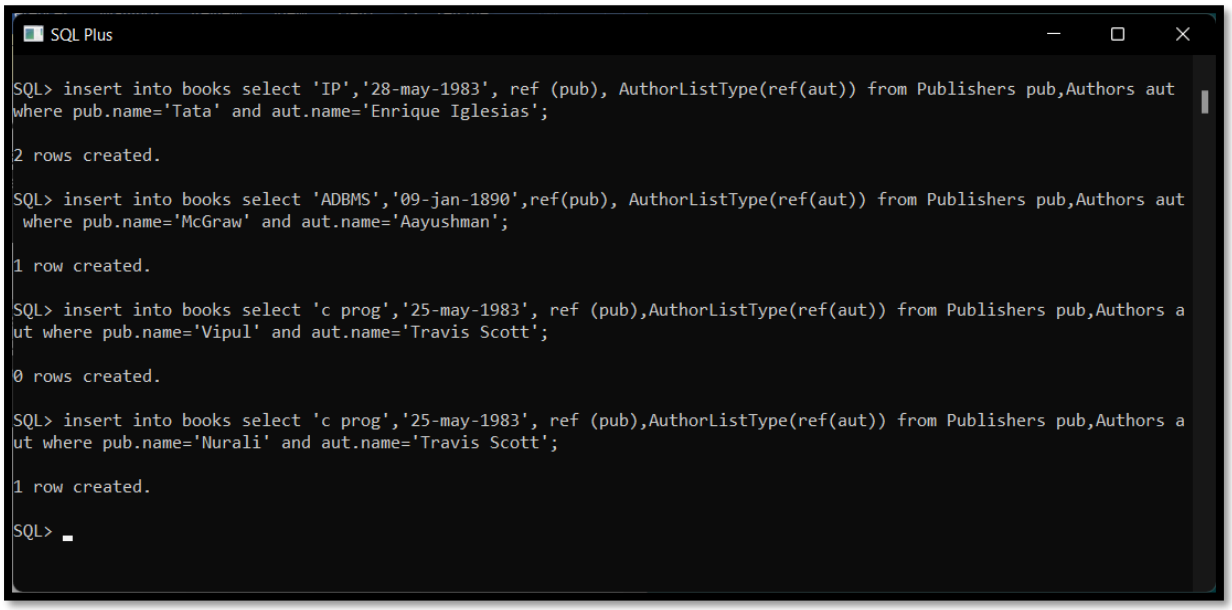
- ✓ **Step 6:-** Insert some records into the above tables and fire the following queries.

```
SQL Plus
SQL> insert into Authors values ('Aayushman', AddrType1(1234, 'Colaba', 'Mumbai', 'Maharashtra', 4000));

1 row created.

SQL> insert into Authors values ('Abhishek', AddrType1(4567, 'Marol', 'Mumbai', 'Maharashtra', 3000));

1 row created.

SQL> insert into Authors values ('Abhishek', AddrType1(8911, 'Borivali', 'Mumbai', 'Maharashtra', 2000));

1 row created.

SQL> insert into Authors values ('Aashi', AddrType1(8726, 'Kandivali', 'Mumbai', 'Maharashtra', 1000));

1 row created.

SQL> insert into Authors values ('Ed Sheeran', AddrType1(5834, 'Paris', 'London', 'United Kingdom', 9000));

1 row created.

SQL> insert into Authors values ('Travis Scott', AddrType1(4568, 'Houston', 'Texas', 'United States', 7000));

1 row created.

SQL> insert into Authors values ('Zack Knight', AddrType1(7825, 'Orlando', 'Florida', 'United States', 1100));

1 row created.

SQL> insert into Authors values ('Enrique Iglesias', AddrType1(2565, 'Miami', 'Madrid', 'Spain', 1120));

1 row created.

SQL>
```

Step 7: - Insert Some records into the above tables and fire the following queries.

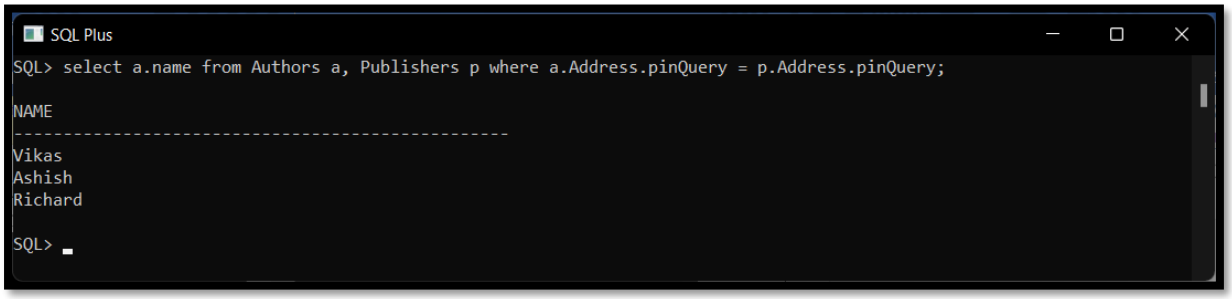


Step 8:- Insert some records into the above tables and fire the following queries



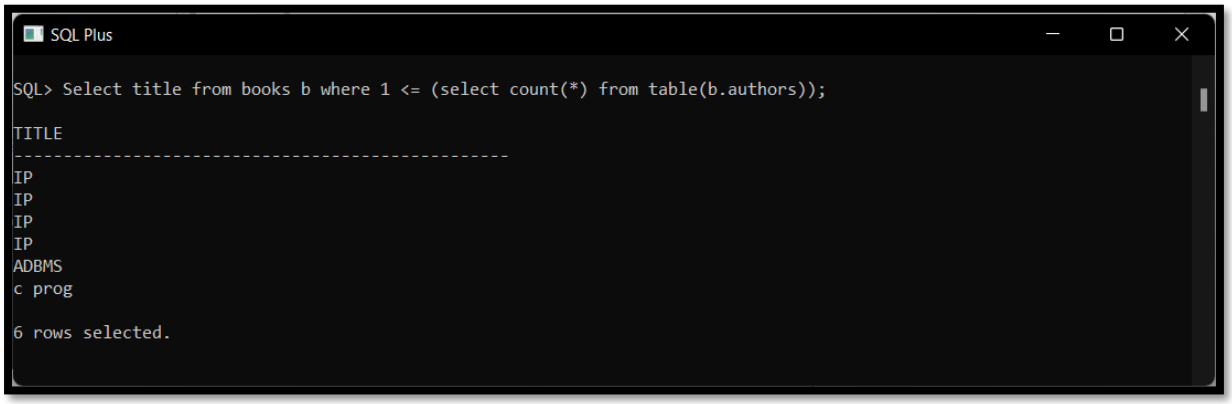
Query : List all of the authors that have the same pin Query as their publisher.

select a.name from Authors a, Publishers p
where a.Address.pinQuery = p.Address.pinQuery;



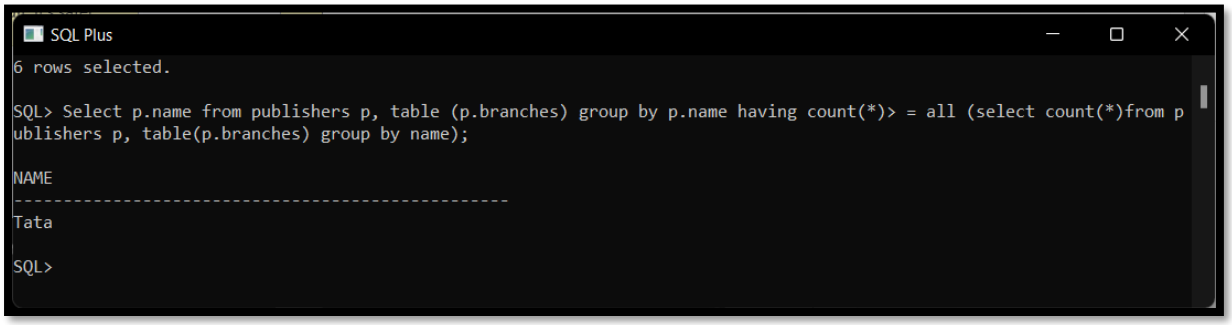
Query: List all books that have 2 or more authors

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



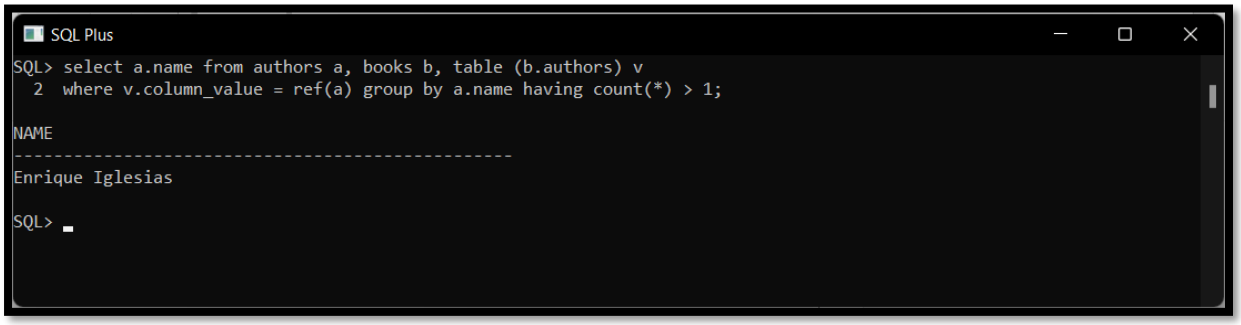
Query: List the name of the publisher that has the most branches

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);



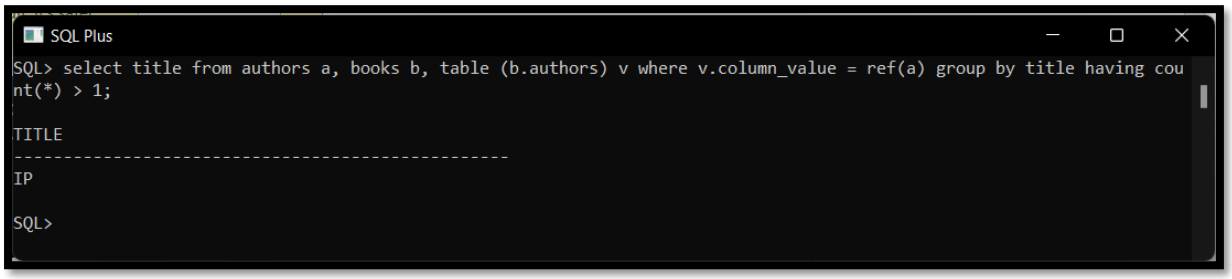
Query: List all authors who have published more than one Book

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;



Query: 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).

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



Conclusion : Successfully Demonstrated insert, update and delete operations on Type

PRACTICAL NO 5

Temporal Database

Abhishek Ojha
Seat No 027

ADVANCED DATABASE SYSTEMS

Practical No: 5

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

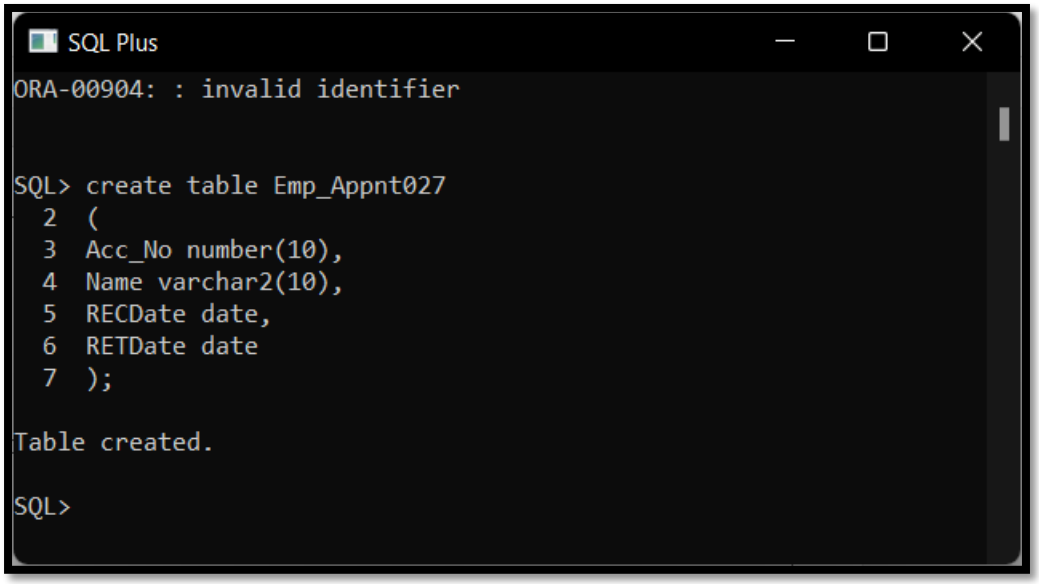
Software Requirement:

MongoDB.

Query:

1. Show the Employee Whose Record Date is 08-Mar-1987.
2. Show the Employee Whose Retired Date is 22-Mar-2021 .
3. Create a new table named as tbl_shares1.
4. Insert Some Row in Table tbl_shares1.
5. Display all the records you have entered in table.
6. Display records where price>100 and TransTime='01:09'.
7. Display the records where price=(select max(price) from tbl_shares1 where TransTime='02:04');.

Practical Implementation:



```
SQL Plus
SQL> insert into Emp_Appnt027 values(1235,'Aakash Pal','08-mar-1987','12-oct- 2015') ;

1 row created.

SQL> insert into Emp_Appnt027 values(1235,'Alpa','08-oct-1978','19-nov-2020') ;

1 row created.

SQL> insert into Emp_Appnt027 values(1237,'ac','25-jan-1988','20-feb-2021') ;

1 row created.

SQL> insert into Emp_Appnt027 values(1278,'xyz','05-dec-1978','02-mar-2017') ;

1 row created.

SQL> insert into emp_appnt027 values(1789,'mon','06-nov-1999','22-mar-2021');

1 row created.

SQL>
```

```
SQL Plus
SQL>  select * from emp_appnt027 ;

    ACC_NO NAME       RECDATE   RETDATE
---------- ---------- --------- ---------
      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

SQL>
```

1. Show the Employee Whose Record Date is 08-Mar-1987.

```
SQL Plus
SQL> select * from emp_appnt027 where RECDate='08-mar-1987';

    ACC_NO NAME       RECDATE   RETDATE
---------- ---------- --------- ---------
      1235 Aakash Pal 08-MAR-87 12-OCT-15

SQL>
```

2. Show the Employee Whose Retired Date is 22-Mar-2021.

```
SQL Plus
SQL> select * from emp_appnt027 where RETDate='22-mar-2021';

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

SQL>
```

3. Create a new table named as tbl_shares1.

```
SQL Plus

SQL> create table tbl_shares1
  2  (
  3  C_Name varchar2(10),
  4  No_Share Number(10),
  5  Price number(10),
  6  TransTime varchar2(10)
  7  Default To_char(sysdate,'HH:MI')
  8  );

Table created.
```

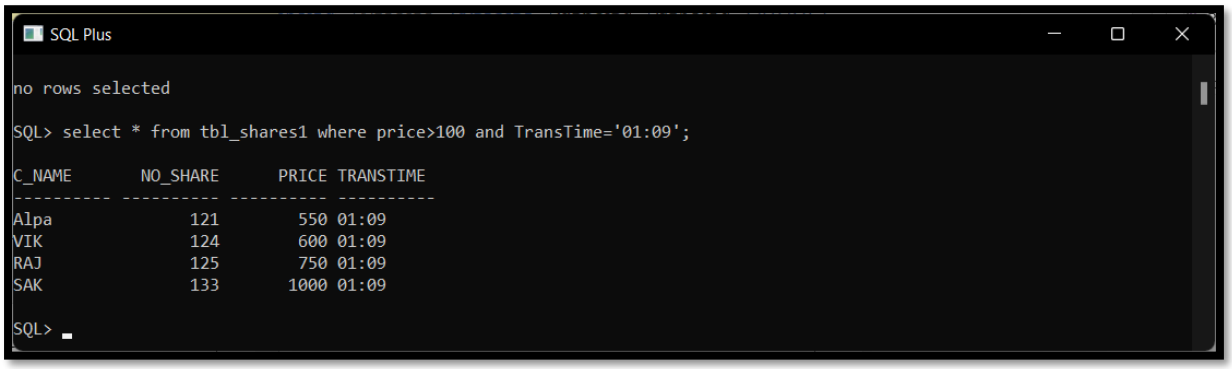
4. Insert Some Row in Table tbl_shares1.

```
SQL Plus
Table created.

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

1 row created.

SQL> insert into tbl_shares1 values('Alpa', 121,550,Default)
  2  /

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.

SQL>
```

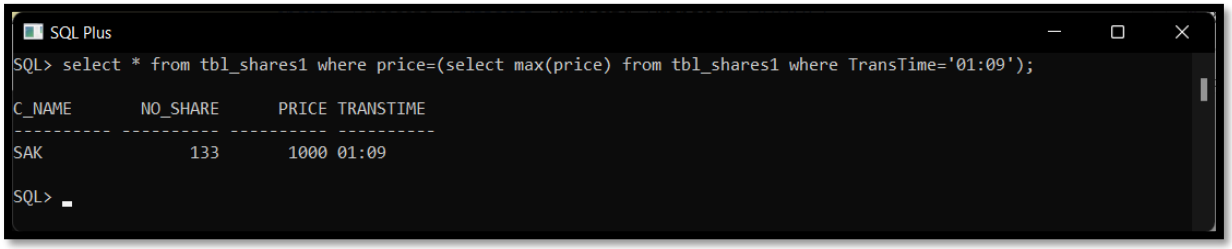
5. Display all the records you have entered in table.

```
SQL Plus

SQL> select * from tbl_shares1;

C_NAME       NO_SHARE      PRICE TRANSTIME
---------- ---------- ---------- ----------
Aakash            123        500 01:08
Alpa              121        550 01:09
VIK               124        600 01:09
RAJ               125        750 01:09
SAK               133       1000 01:09

SQL>
```

6. **Display records where price>100 and TransTime='01:09'.**



7. **Display the records where price=(select max(price) from tbl_shares1 where TransTime='02:04');**



Conclusion : Successfully Performed and Implemented the temporal database and issue queries on Oracle Database.

PRACTICAL NO 6

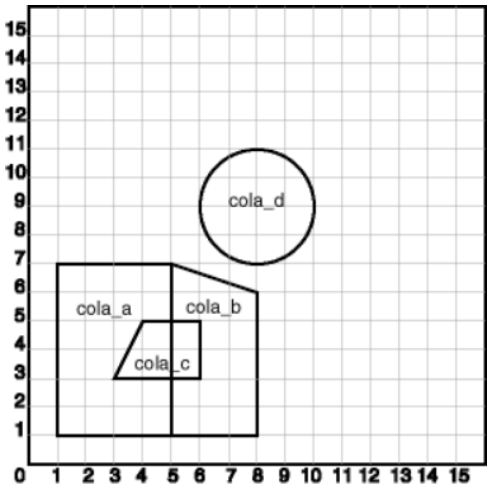
Spatial Database

Abhishek Ojha
Seat No 027

ADVANCED DATABASE SYSTEMS

Practical 6:

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



Software Requirement: Oracle 11g.

Query:

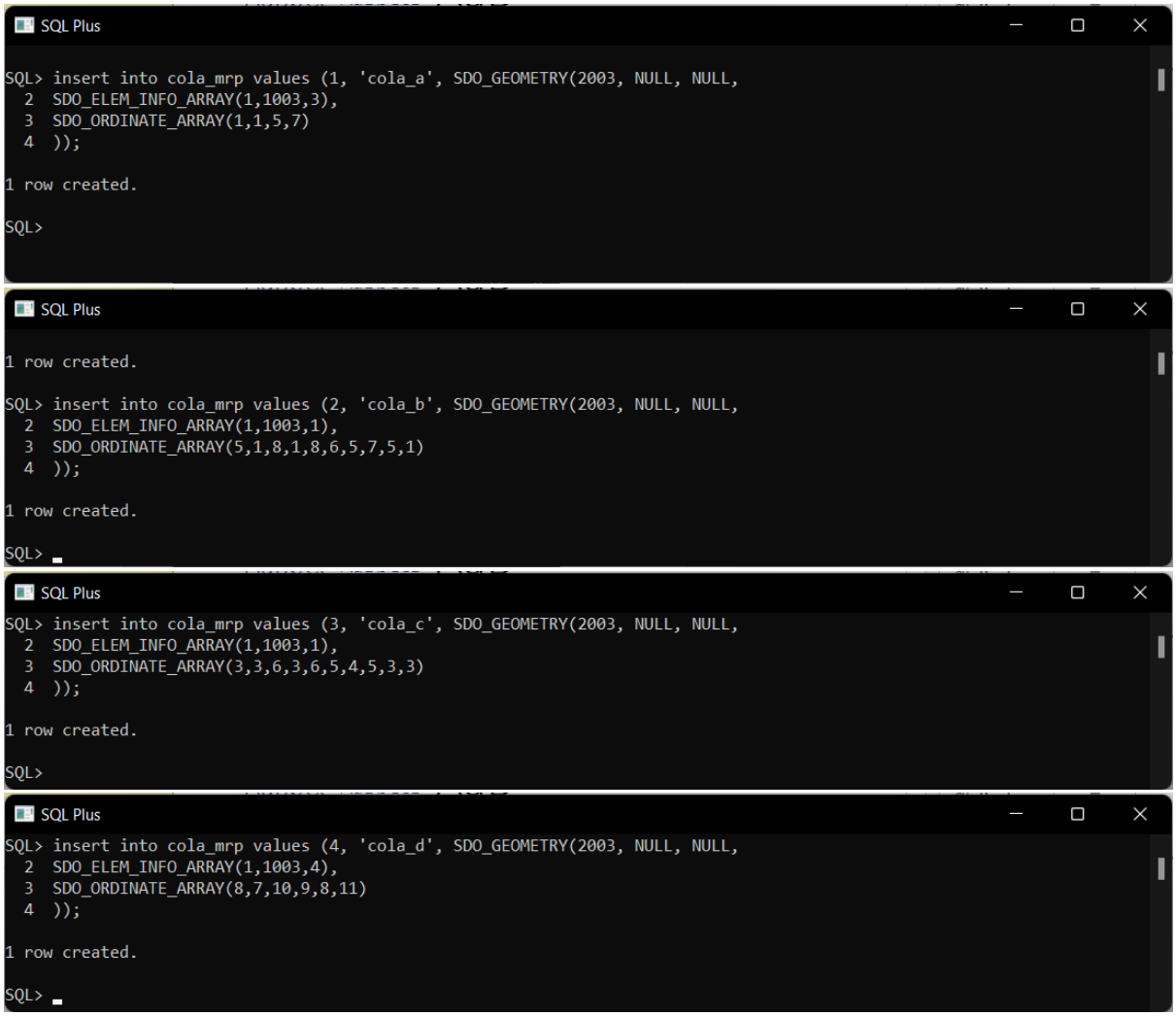
Create a spatial database table that stores the number, name and location, which consists of four different areas say abc, pqr, mno and xyz.
Fire the following queries:

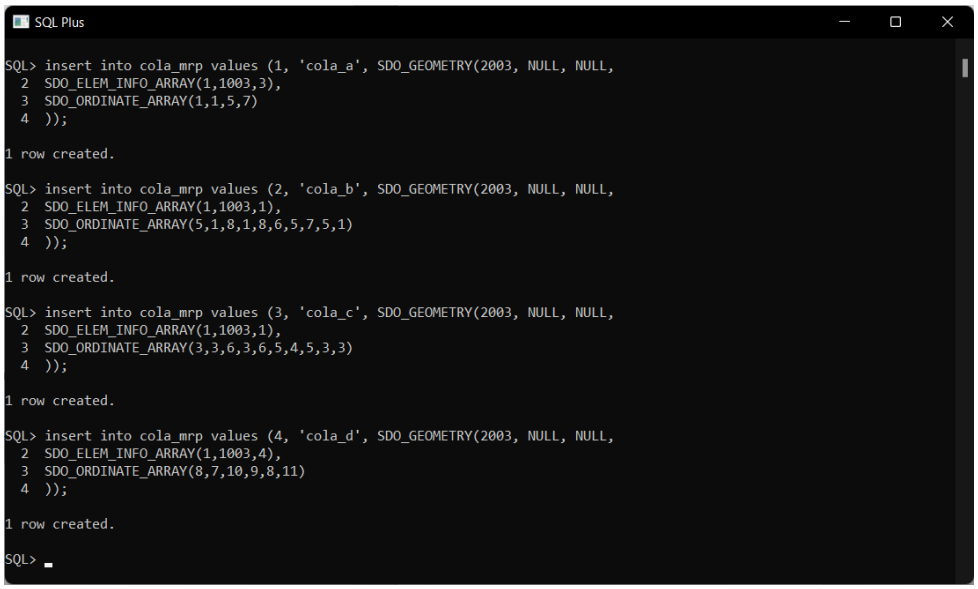
a) Find the topological intersection of two geometries.
b) Find whether two geometric figures are equivalent to each other.
c) Find the areas of all different locations.
d) Find the area of only one location.
e) Find the distance between two geometries.

Practical Implementation:

1. **Create a table for cola (soft drink) markets in a given geography (such as city or state). Each row will be an area of interest for a specific cola (for example, where the cola is most preferred by residents, where the manufacturer believes the cola has growth potential, and so on). (For restrictions on spatial table and column names, see .**

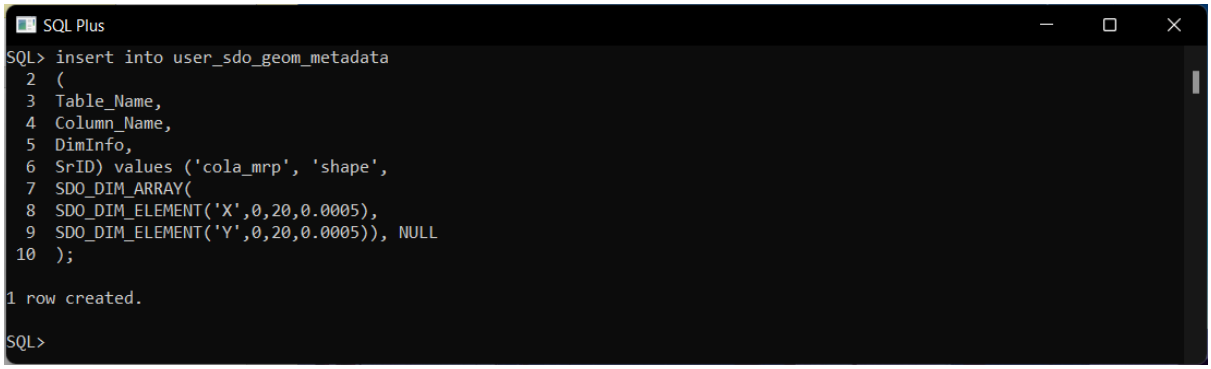
2. The next INSERT statement creates an area of interest for Cola A. This area happens to be a rectangle. The area could represent any user-defined criterion: for example, where Cola A is the preferred drink, where Cola A is under competitive pressure, where Cola A has strong growth potential, and so on.



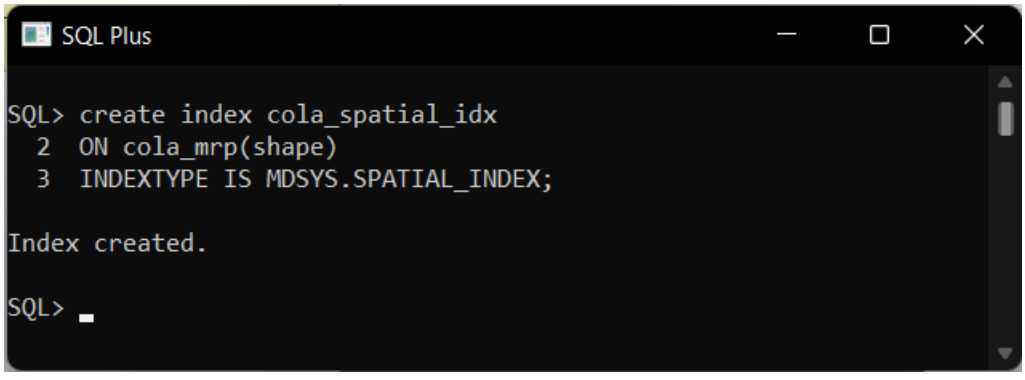


3. UPDATE METADATA VIEW

Update the USER_SDO_GEOM_METADATA view. This is required before the spatial index can be created. Do this only once for each layer (that is, table-column combination; here: COLA_MARKETS and SHAPE).



4. CREATE THE SPATIAL INDEX.



5. PERFORM SOME SPATIAL QUERIES

Return the topological intersection of two geometries.

```
SQL Plus
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))


SQL>
```

Do two geometries have any spatial relationship?.

```
SQL Plus


SQL> SELECT SDO_GEOM.RELATE(c_b.shape, 'anyinteract', 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.RELATE(C_B.SHAPE,'ANYINTERACT',C_D.SHAPE,0.005)
--------------------------------------------------------------------------------
FALSE

SQL>
```

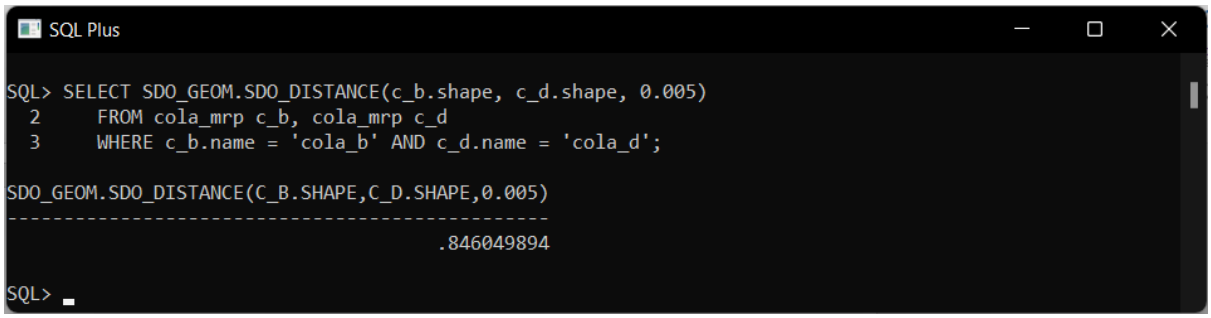
Return the areas of all cola markets.

```
SQL Plus

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                                   12.5663706

SQL>
```

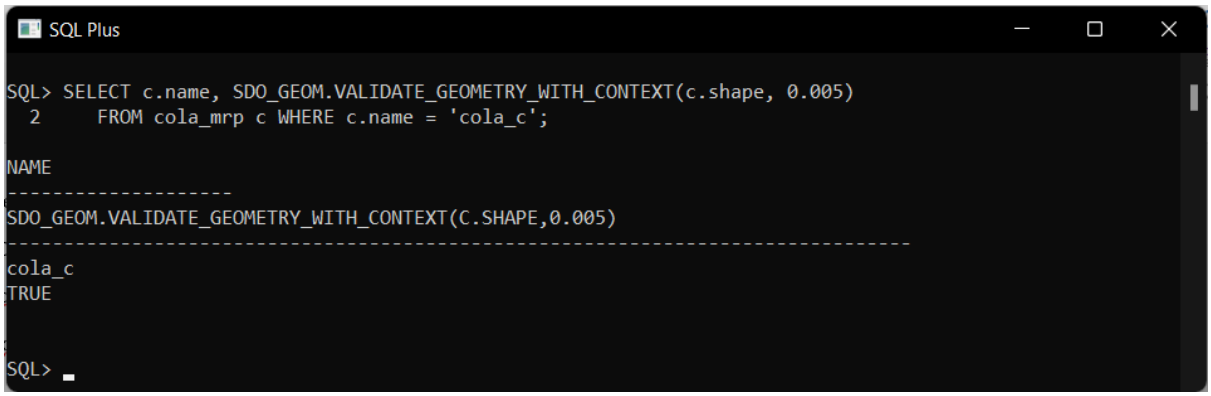
Return the area of just cola_a.

```
SQL Plus
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

SQL>
```

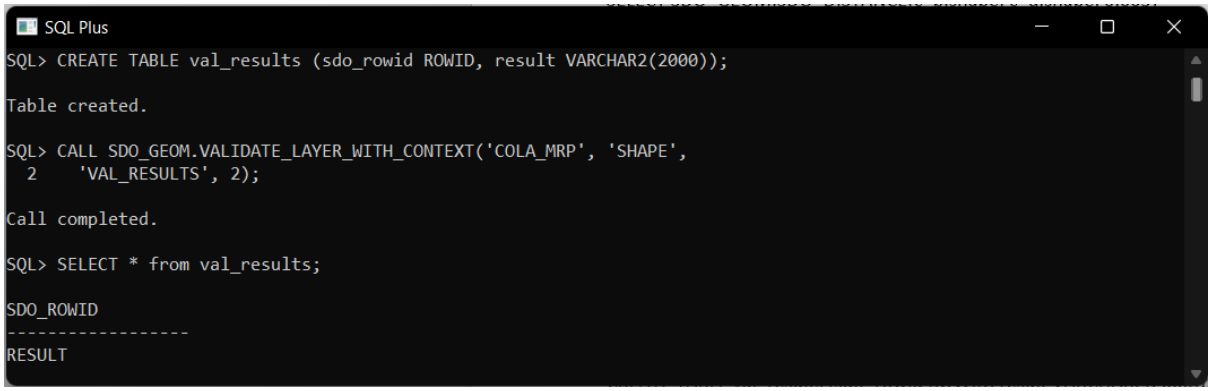
Return the distance between two geometries.



Is a geometry valid?



is a layer valid? (First, create the results table).



Conclusion : Successfully Performed the Spatial Data Queries on Oracle Database.

PRACTICAL NO 7

XML Database

Abhishek Ojha
Seat No 027

ADVANCED DATABASE SYSTEMS

Practical 7: XML Database

Aim:

Create a table employee having dept_id as number datatype and employee_spec as XML data type (XM_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.

Query:

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.

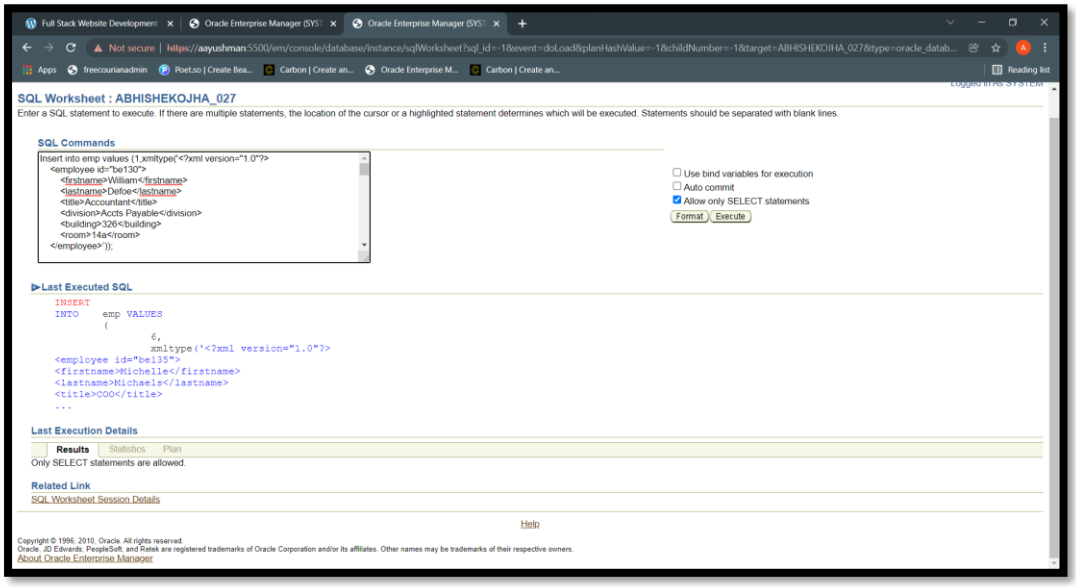
Software Requirements:

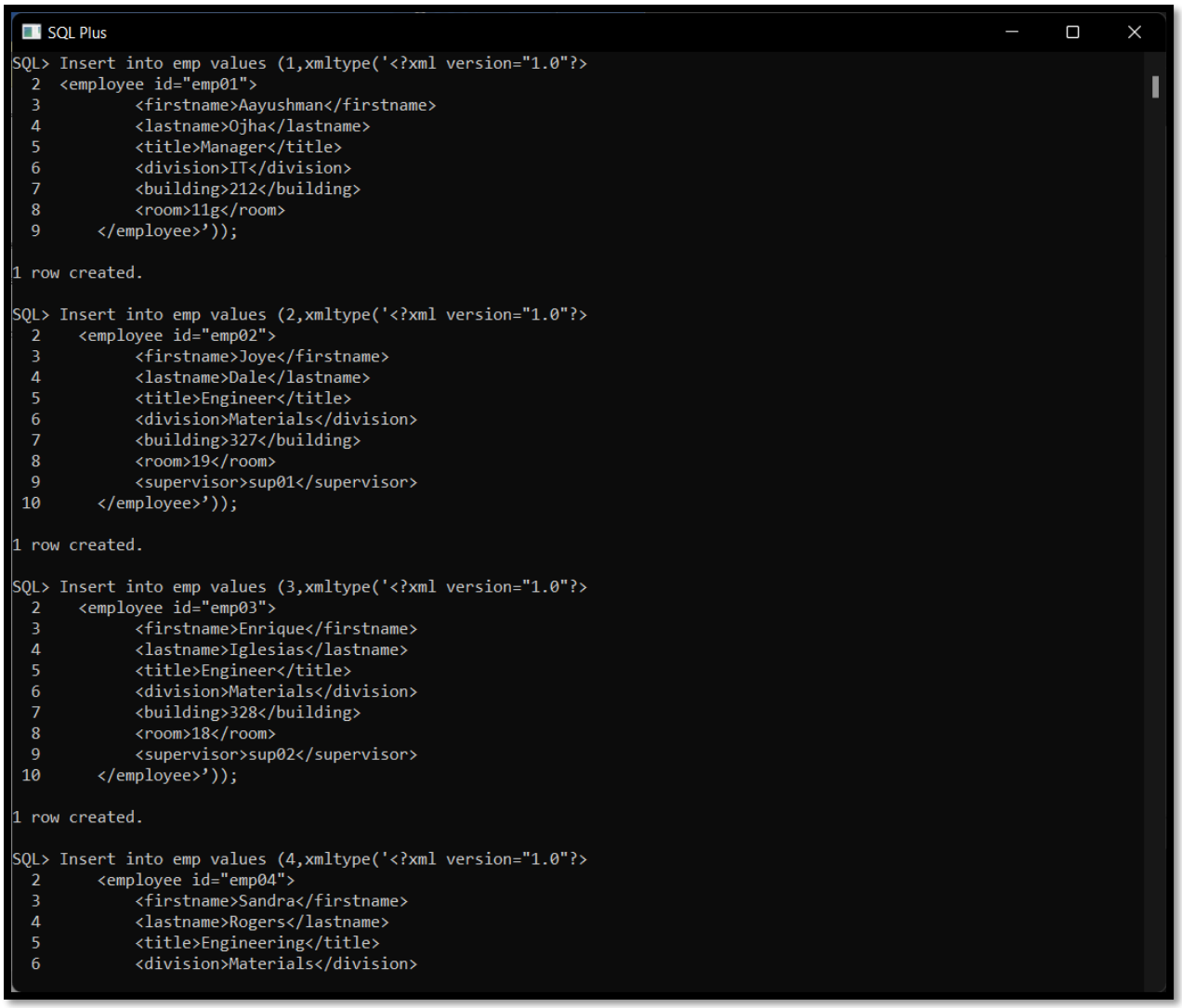
Oracle 11g Express Edition, Any browser.

Practical Implementation:

1. **Create Table Employee.**

2. **Insert Some Records in Created Table.**




SQL Plus
SQL> Insert into emp values (1,xmltype('<?xml version="1.0"?>
2 <employee id="emp01">
3 <firstname>Aayushman</firstname>
4 <lastname>Ojha</lastname>
5 <title>Manager</title>
6 <division>IT</division>
7 <building>212</building>
8 <room>11g</room>
9 </employee>'));
1 row created.
SQL> Insert into emp values (2,xmltype('<?xml version="1.0"?>
2 <employee id="emp02">
3 <firstname>Joye</firstname>
4 <lastname>Dale</lastname>
5 <title>Engineer</title>
6 <division>Materials</division>
7 <building>327</building>
8 <room>19</room>
9 <supervisor>sup01</supervisor>
10 </employee>'));
1 row created.
SQL> Insert into emp values (3,xmltype('<?xml version="1.0"?>
2 <employee id="emp03">
3 <firstname>Enrique</firstname>
4 <lastname>Iglesias</lastname>
5 <title>Engineer</title>
6 <division>Materials</division>
7 <building>328</building>
8 <room>18</room>
9 <supervisor>sup02</supervisor>
10 </employee>'));
1 row created.
SQL> Insert into emp values (4,xmltype('<?xml version="1.0"?>
2 <employee id="emp04">
3 <firstname>Sandra</firstname>
4 <lastname>Rogers</lastname>
5 <title>Engineering</title>
6 <division>Materials</division>

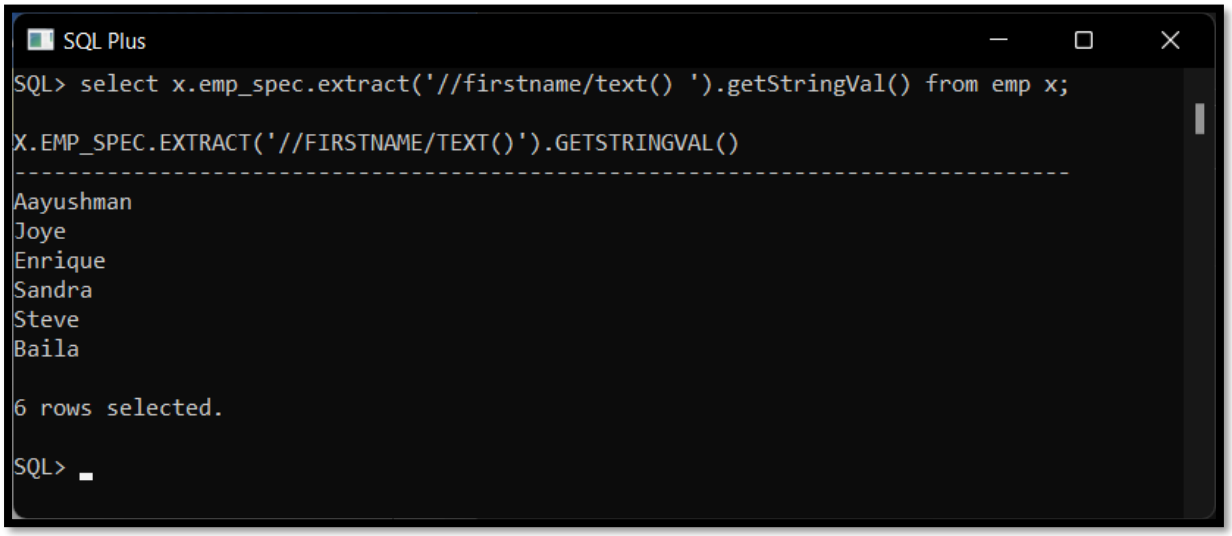
SQL Plus

```
SQL> Insert into emp values (4,xmltype('<?xml version="1.0"?>
  2      <employee id="emp04">
  3          <firstname>Sandra</firstname>
  4          <lastname>Rogers</lastname>
  5          <title>Engineering</title>
  6          <division>Materials</division>
  7          <building>312</building>
  8          <room>22</room>
  9      </employee>'));

1 row created.

SQL> Insert into emp values (5,xmltype('<?xml version="1.0"?>
  2      <employee id="emp05">
  3          <firstname>Steve</firstname>
  4          <lastname>Casey</lastname>
  5          <title>Engineering</title>
  6          <division>Materials</division>
  7          <building>345</building>
  8          <room>24</room>
  9      </employee>'));

1 row created.

SQL> Insert into emp values (6,xmltype('<?xml version="1.0"?>
  2      <employee id="emp06">
  3          <firstname>Baila</firstname>
  4          <lastname>Conmigo</lastname>
  5          <title>COO</title>
  6          <division>Management</division>
  7          <building>216</building>
  8          <room>264</room>
  9      </employee>'));

1 row created.

SQL>
```

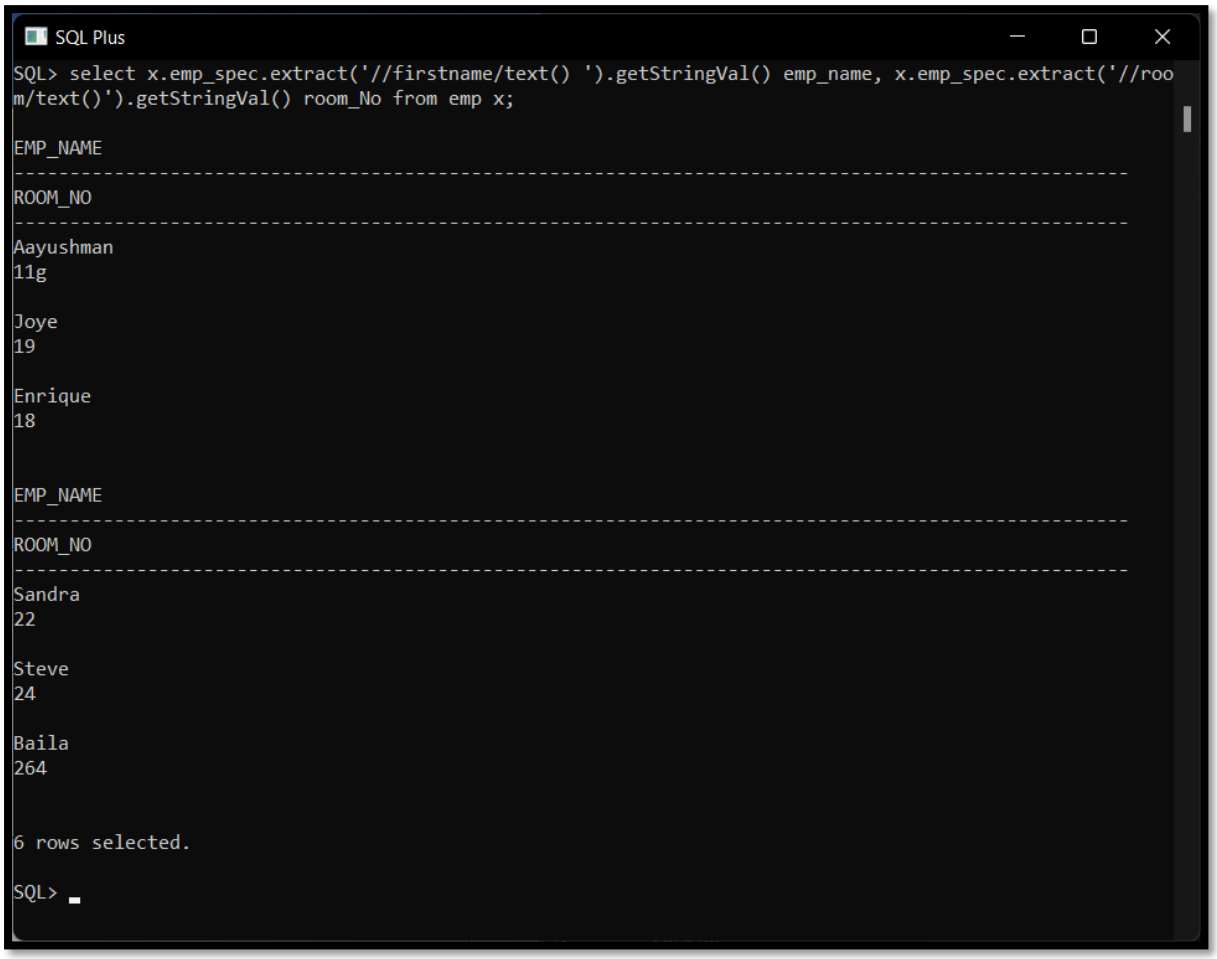
SQL Plus

```
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"?>

6 rows selected.

SQL>
```

3. Get the first name:



4. Get the first name and room number.



5. Get the first name and room number and title.

```
SQL Plus

SQL> select x.emp_spec.extract('//firstname/text() ').getStringVal() emp_name,
  2  x.emp_spec.extract('//room/text() ').getStringVal() room_no,
  3  x.emp_spec.extract('//title/text() ').getStringVal() title
  4  from emp x;

EMP_NAME
--------------------------------------------------------------------------------
ROOM_NO
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------
Aayushman
11g
Manager

Joye
19
Engineer

EMP_NAME
--------------------------------------------------------------------------------
ROOM_NO
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------

Enrique
18
Engineer

Sandra
22

EMP_NAME
--------------------------------------------------------------------------------
ROOM_NO
--------------------------------------------------------------------------------
TITLE
--------------------------------------------------------------------------------
Engineering

Steve
24
```

6. Update 6th record from the table:

```
SQL Plus

SQL> Update emp set emp_spec=xmltype('<?xml version="1.0"?>
  2      <employee id="emp06">
  3    <firstname>Sam</firstname>
  4   <lastname>Conmigo</lastname>
  5   <title>COO</title>
  6   <division>Management</division>
  7   <building>216</building>
  8   <room>264</room>
  9   </employee> ') where emp_id=6;

1 row updated.
```

```
SQL Plus

    EMP_ID
----------
EMP_SPEC
--------------------------------------------------------------------------------
<?xml version="1.0"?>
    <employee id="emp06">
 <firstname>Sam</firstname>
<las


6 rows selected.

SQL>
```

7. **Delete a record from the table:**

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

1 row deleted.
```

Conclusion :- Successfully Performed Operation like Create, Read, Update and Delete on XML Database.