

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

<b>Roll No.</b>	027
<b>Name</b>	OJHA ABHISHEK DEVMANI
<b>Subject</b>	ADVANCED DATABASE SYSTEMS



## 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 Joining. 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

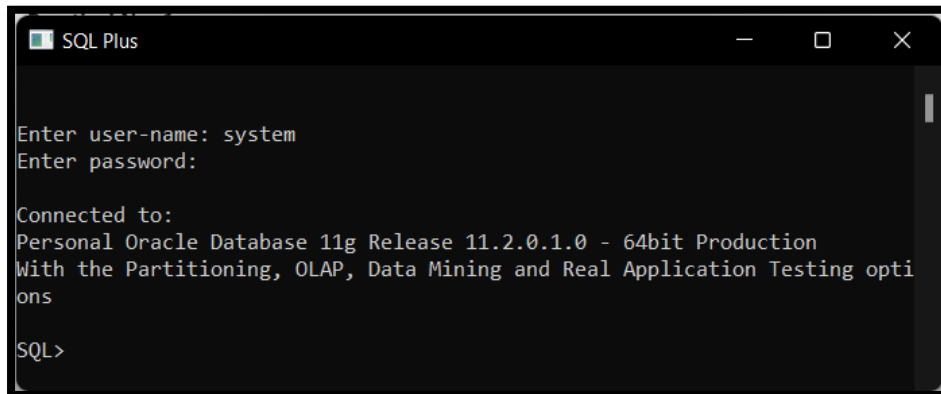
# ADVANCED DATABASE SYSTEMS

## 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:

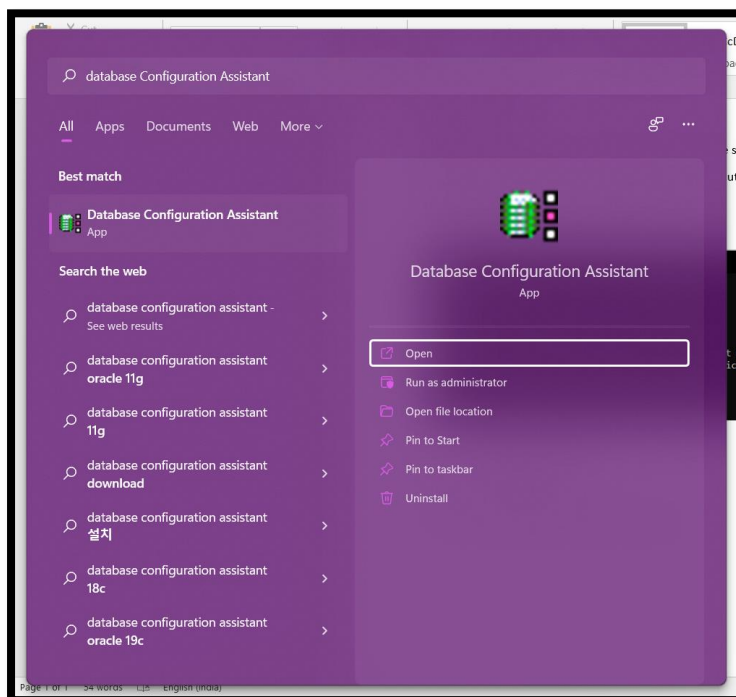
Oracle Database 11g.



## How to Create Two Database

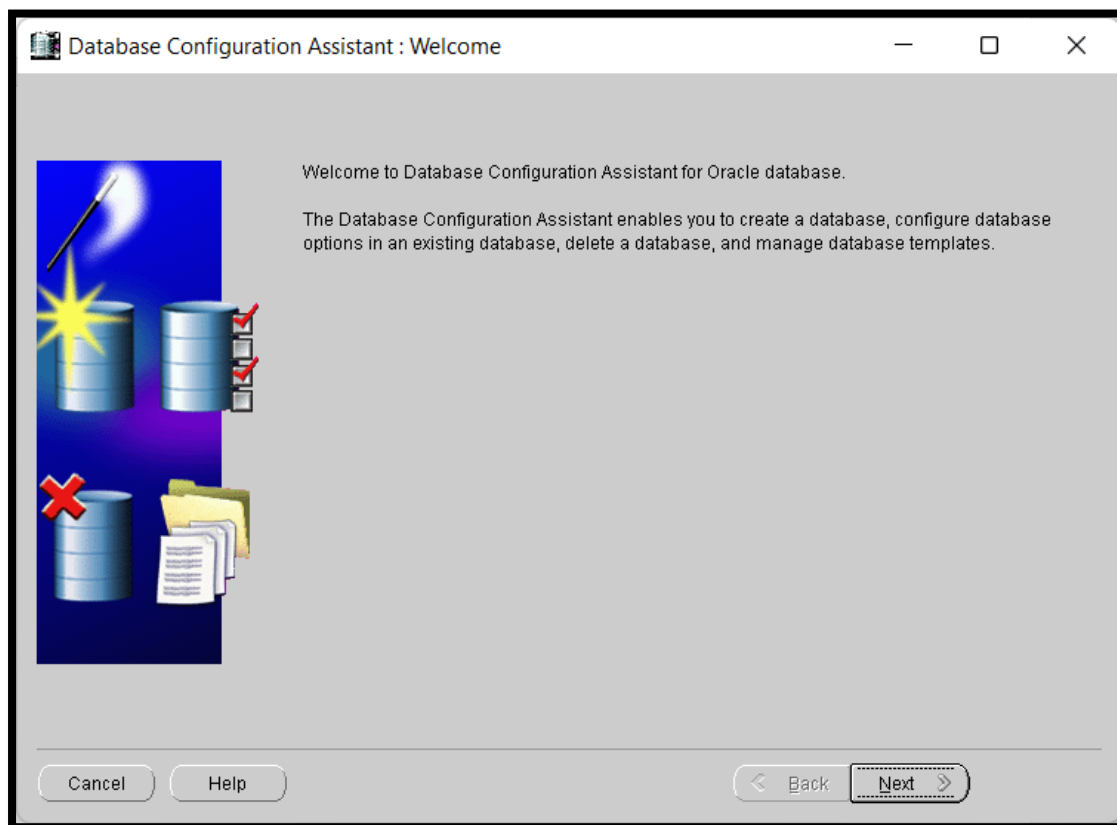
### Steps to Create Database db1 and db2

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

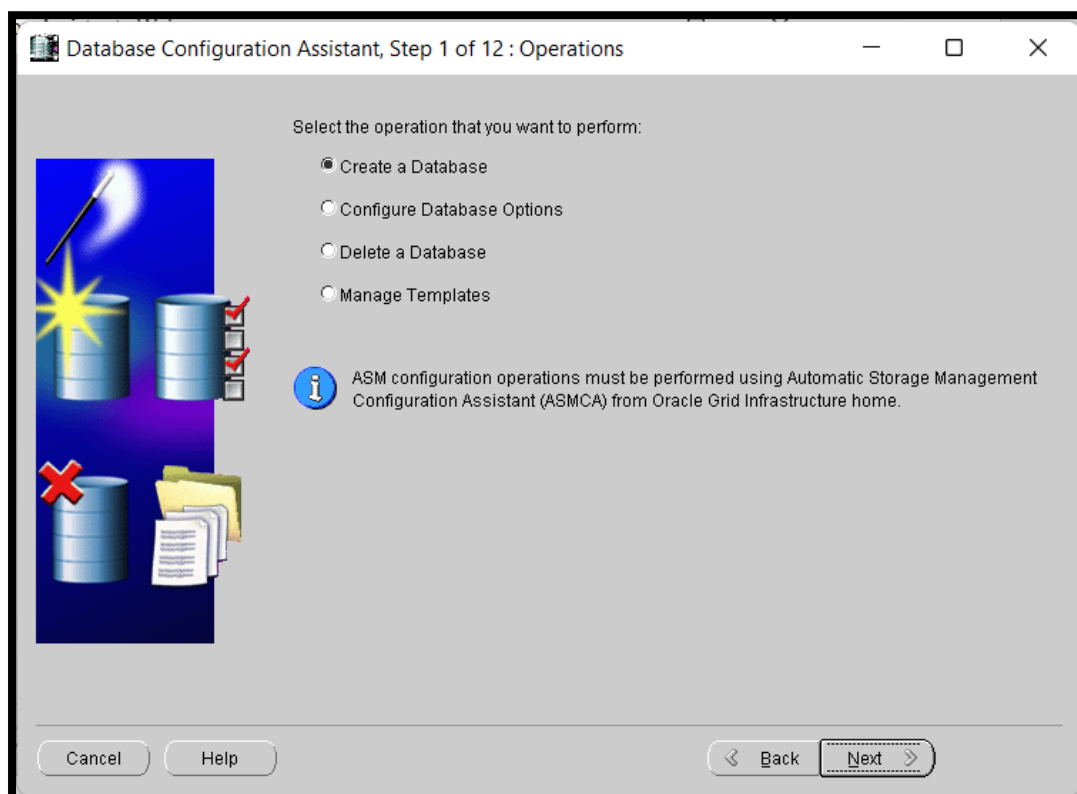


**Step 2:** Click on Next.

# ADVANCED DATABASE SYSTEMS

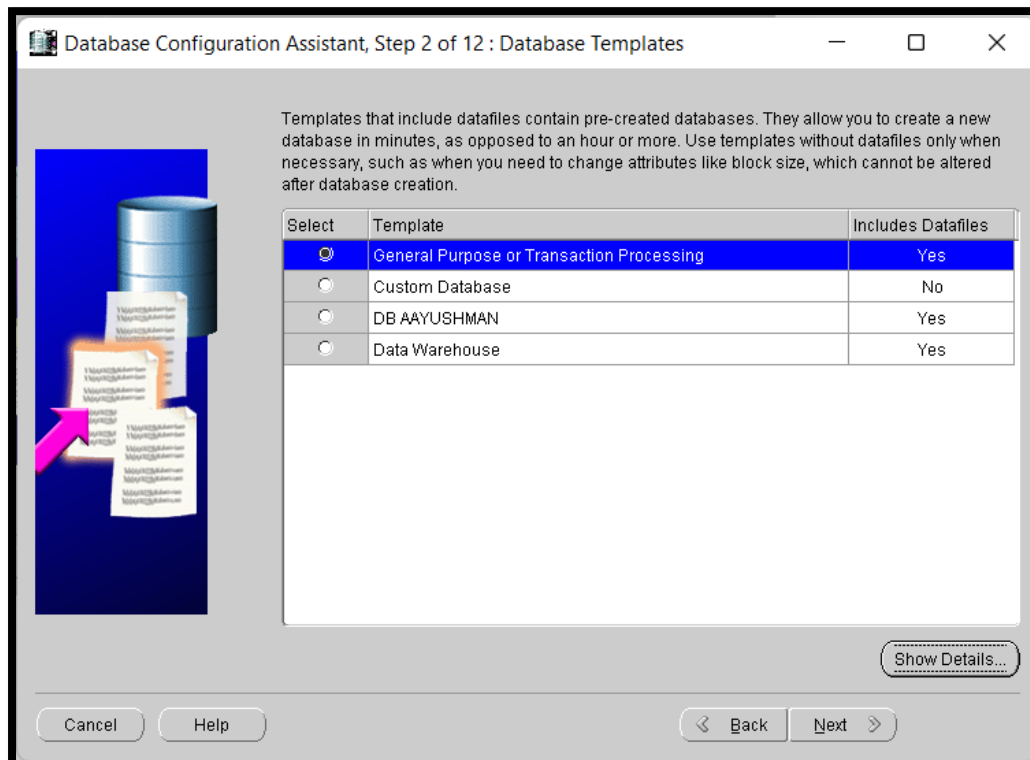


**Step 3:** Select Option Create a Database.

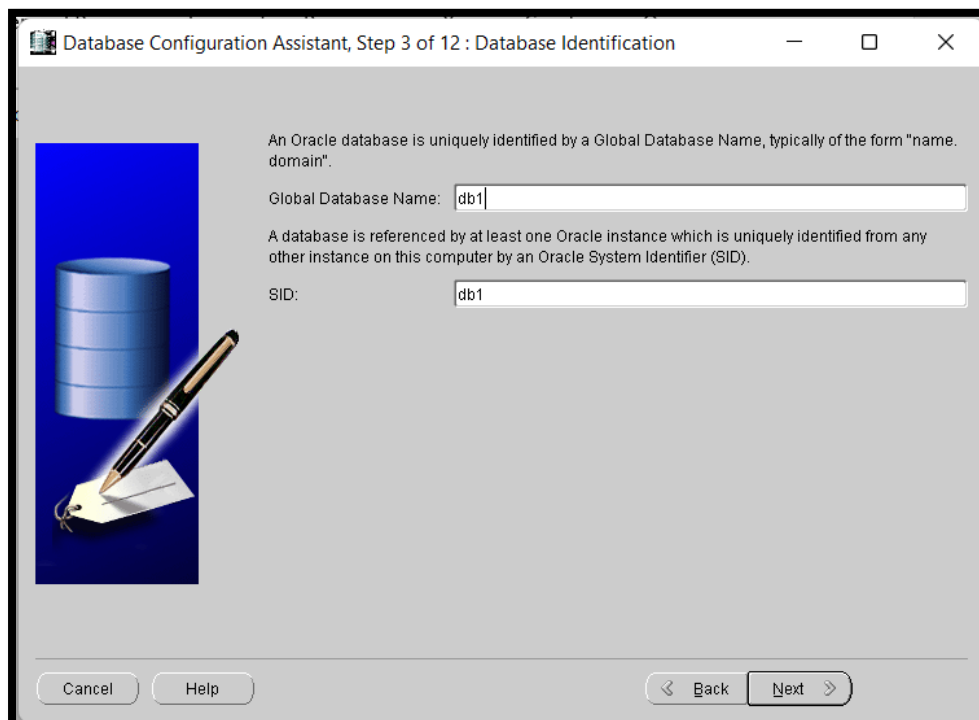


## ADVANCED DATABASE SYSTEMS

**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).



## ADVANCED DATABASE SYSTEMS

Database Configuration Assistant, Step 3 of 12 : Database Identification

An Oracle database is uniquely identified by a Global Database Name, typically of the form "name.domain".

Global Database Name:

A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

SID:

Cancel Help Back Next

**Step 6:** No changes Needed, Click on Next.

Database Configuration Assistant, Step 4 of 12 : Management Options

Enterprise Manager Automatic Maintenance Tasks

☒ Configure Enterprise Manager

☐ Register with Grid Control for centralized management

Management Service:

☒ Configure Database Control for local management

☐ Enable Alert Notifications

Outgoing Mail (SMTP) Server:

Recipient Email Address:

☐ Enable Daily Disk Backup to Recovery Area

Backup Start Time:  :  AM ☐ PM

OS Username:

OS Password:

Cancel Help Back Next

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



## ADVANCED DATABASE SYSTEMS

Database Configuration Assistant, Step 5 of 12 : Database Credentials

For security reasons, you must specify passwords for the following user accounts in the new database.

☒ Use Different Administrative Passwords

User Name	Password	Confirm Password
SYS	*****	*****
SYSTEM	*****	*****
DBSNMP	*****	*****
SYSMAN	*****	*****

☐ Use the Same Administrative Password for All Accounts


Password:

Confirm Password:

Cancel Help < Back Next >

Checks for Password Confirmation, Just Click Yes.

Database Configuration Assistant

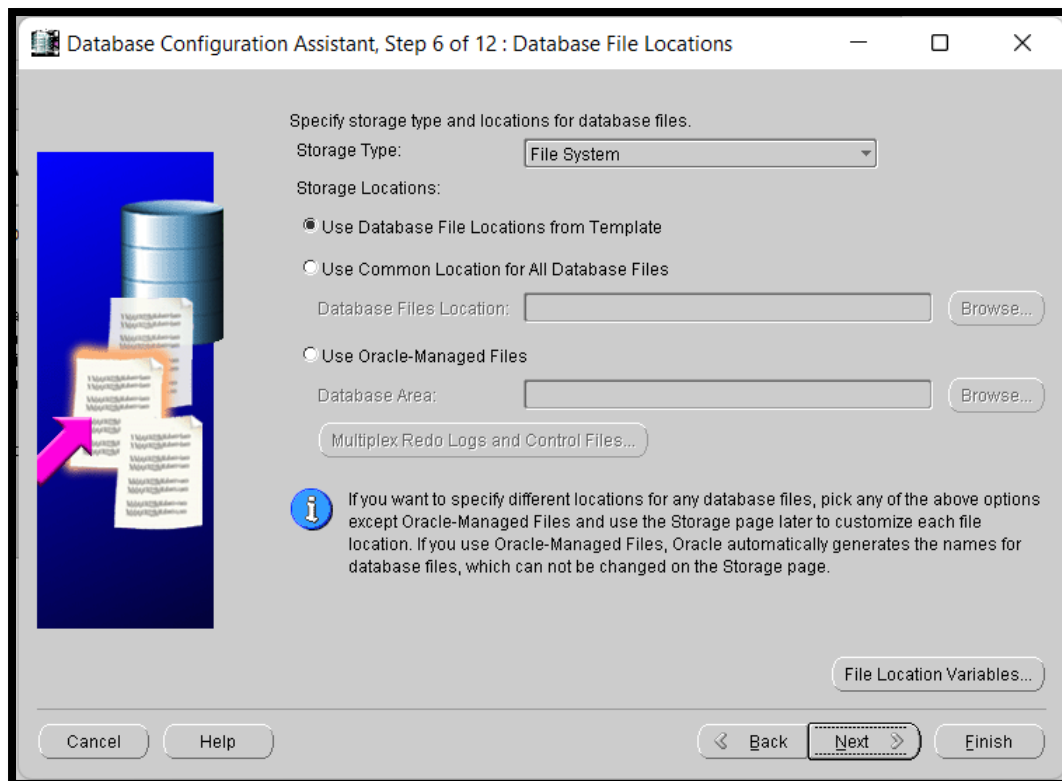
 Password entered does not satisfy Oracle recommended password complexity policy. A password should have minimum of 8 characters in length. In addition, the password must contain at least one upper case character, one lower case character and one digit.

Do you want to continue?

Yes No

## ADVANCED DATABASE SYSTEMS

**Step 8:** No changes Needed, Click on Next.



Database Configuration Assistant, Step 6 of 12 : Database File Locations

Specify storage type and locations for database files.

Storage Type:

Storage Locations:


☒ Use Database File Locations from Template

☐ Use Common Location for All Database Files

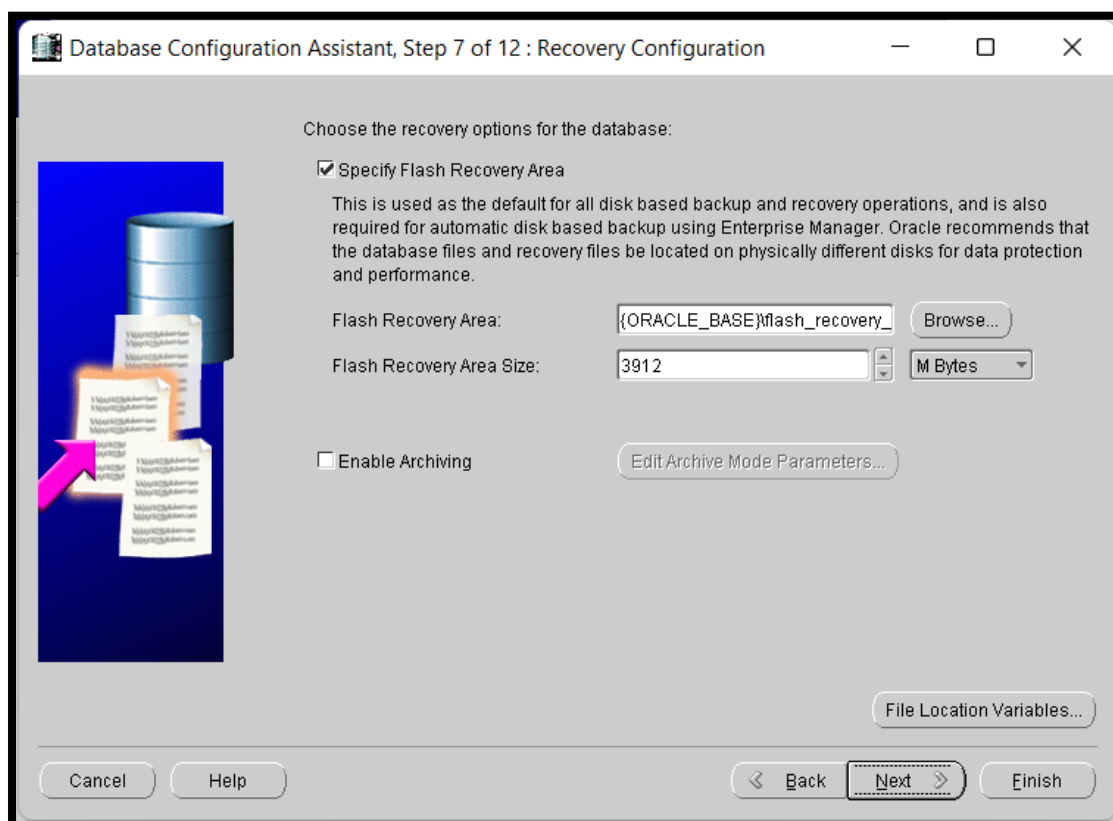
Database Files Location:

☐ Use Oracle-Managed Files

Database Area:

 If you want to specify different locations for any database files, pick any of the above options except Oracle-Managed Files and use the Storage page later to customize each file location. If you use Oracle-Managed Files, Oracle automatically generates the names for database files, which can not be changed on the Storage page.

**Step 9:** No changes Needed, Click on Next.



Database Configuration Assistant, Step 7 of 12 : Recovery Configuration

Choose the recovery options for the database:

☒ Specify Flash Recovery Area

This is used as the default for all disk based backup and recovery operations, and is also required for automatic disk based backup using Enterprise Manager. Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

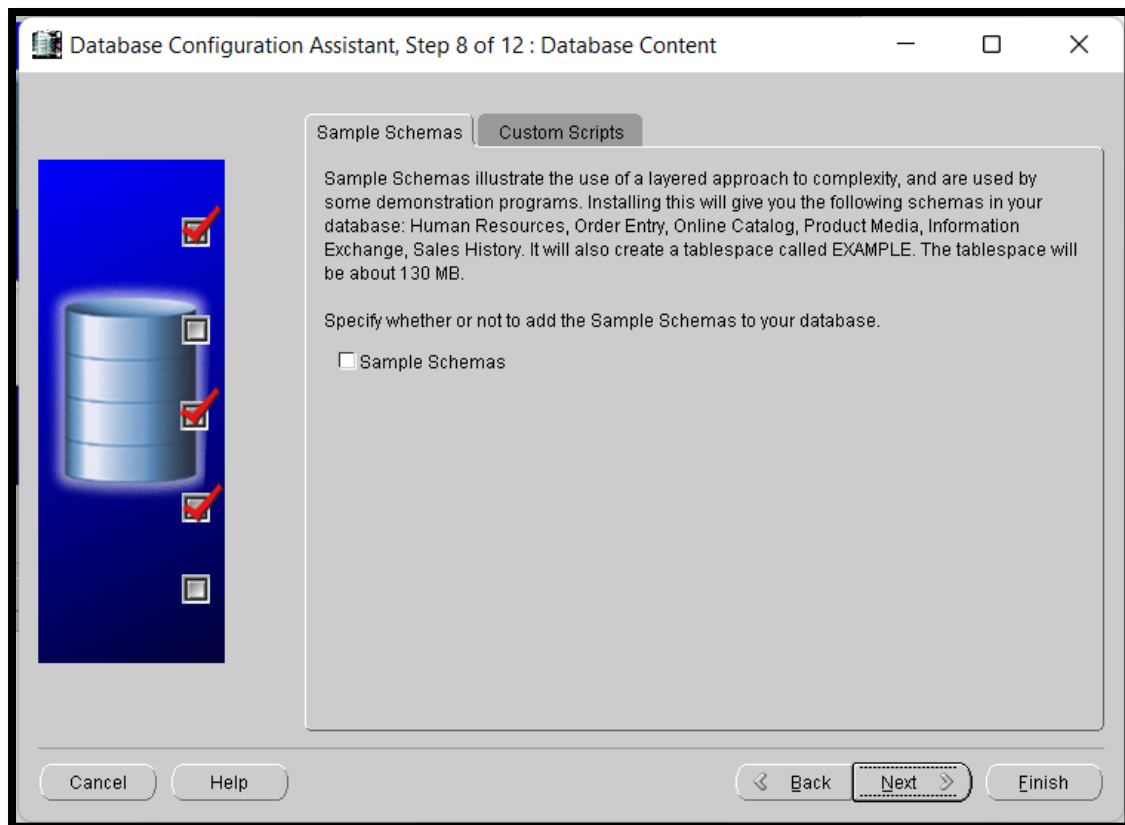
Flash Recovery Area:

Flash Recovery Area Size:

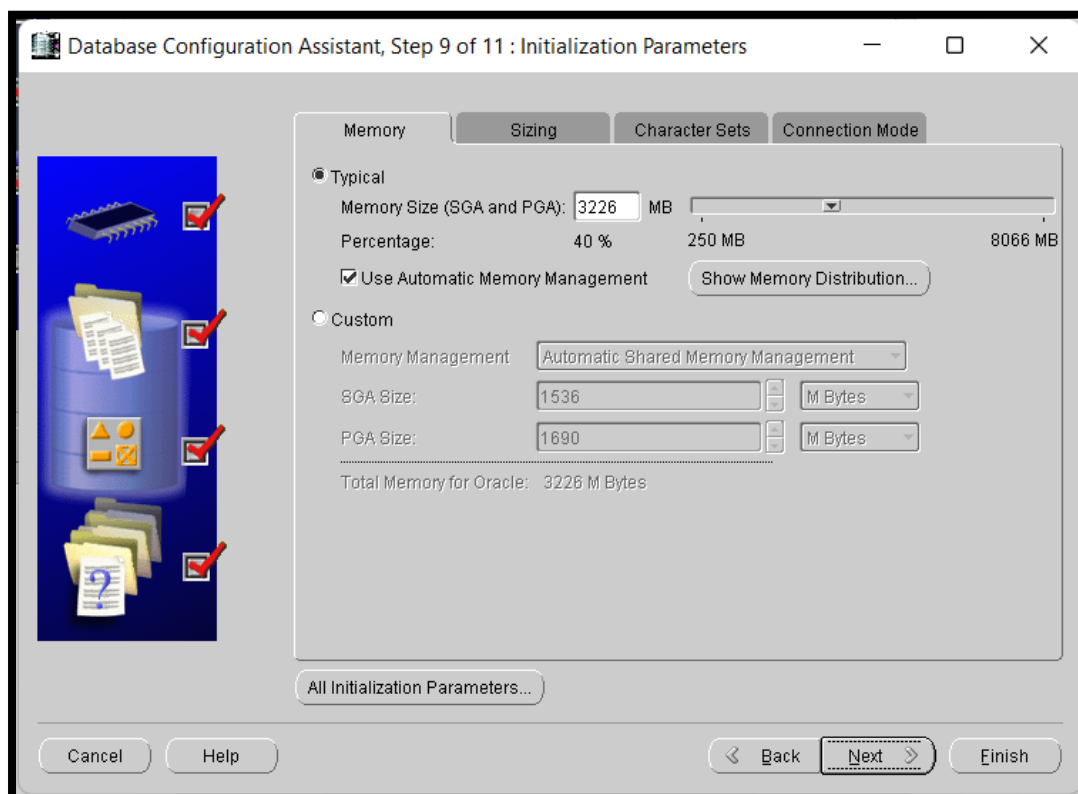
☐ Enable Archiving

## ADVANCED DATABASE SYSTEMS

**Step 10:** No changes Needed, Click on Next.

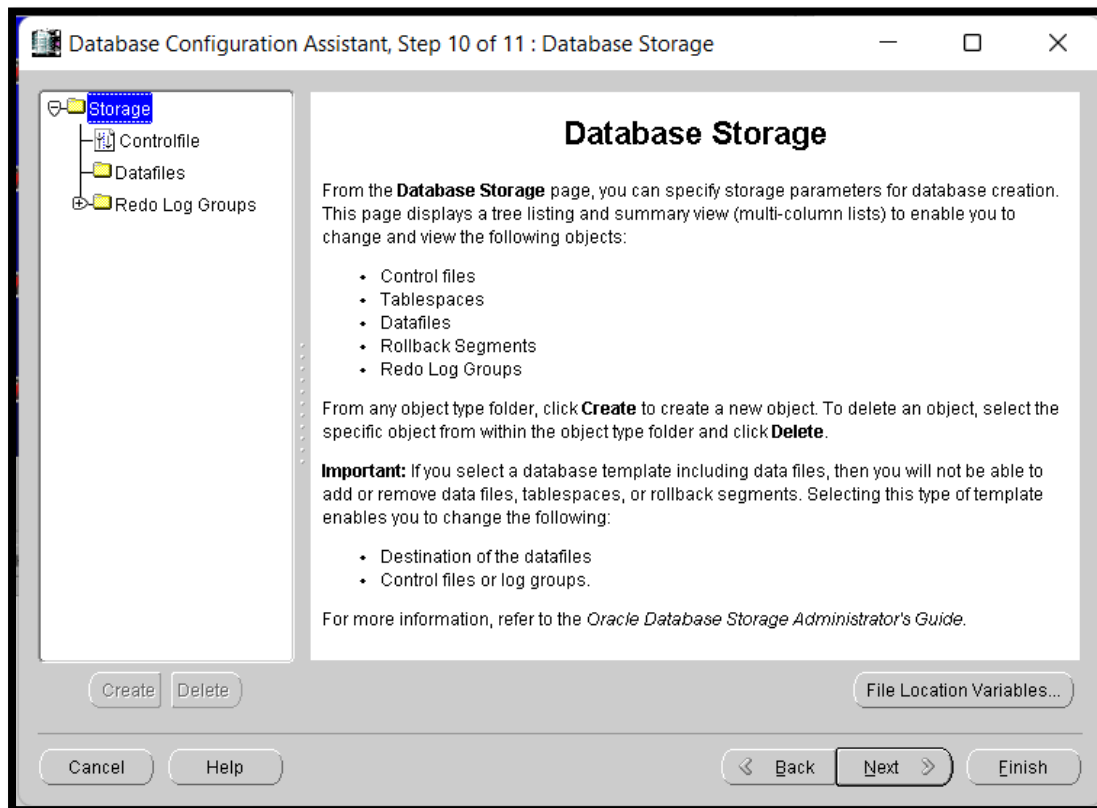


**Step 11:** No changes Needed, Click on Next.



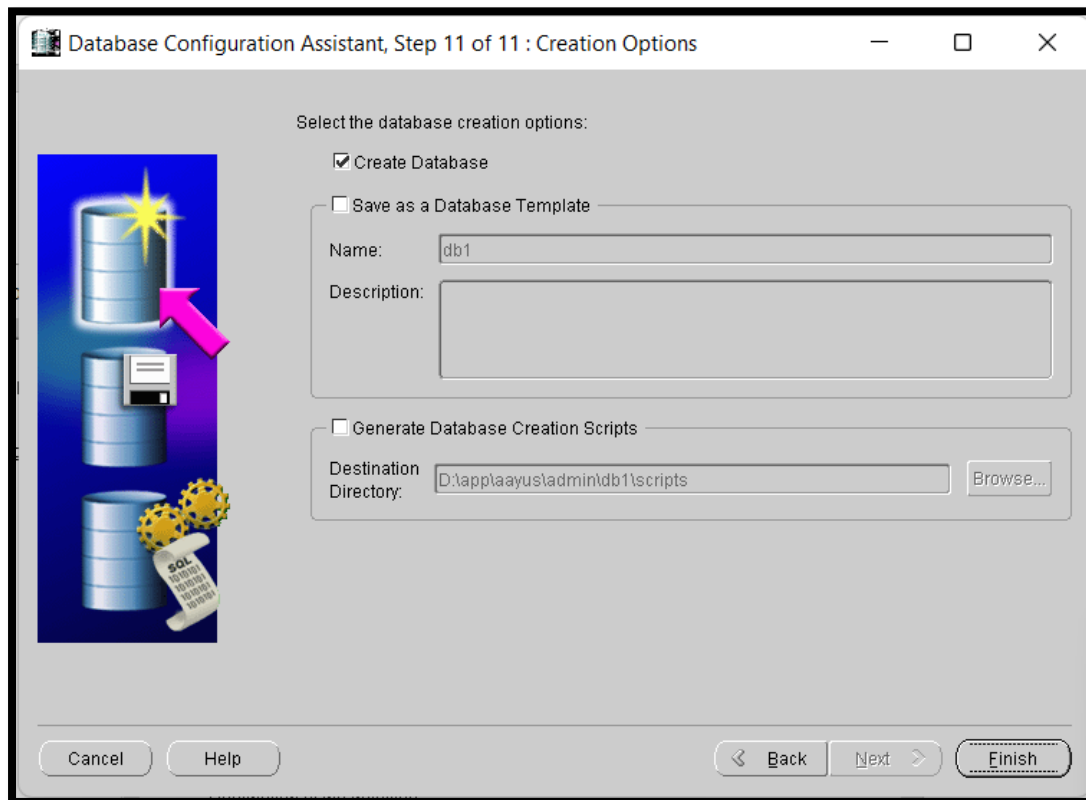
# ADVANCED DATABASE SYSTEMS

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

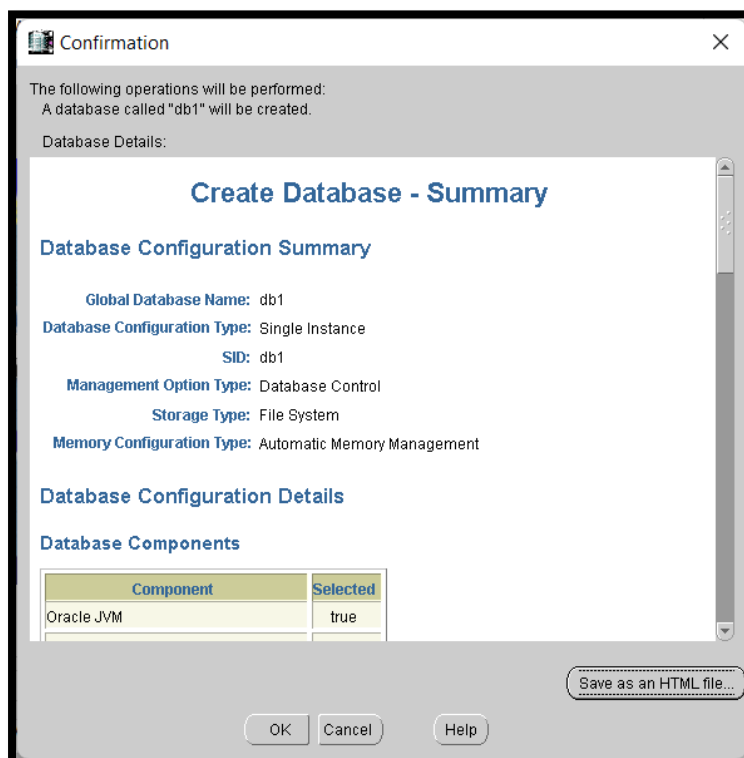


# ADVANCED DATABASE SYSTEMS

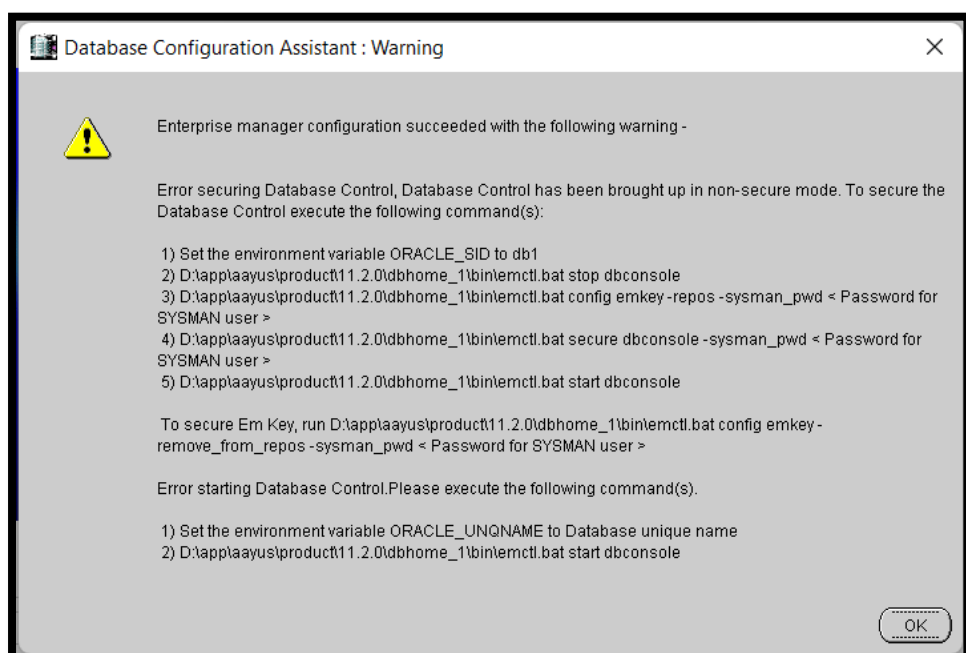
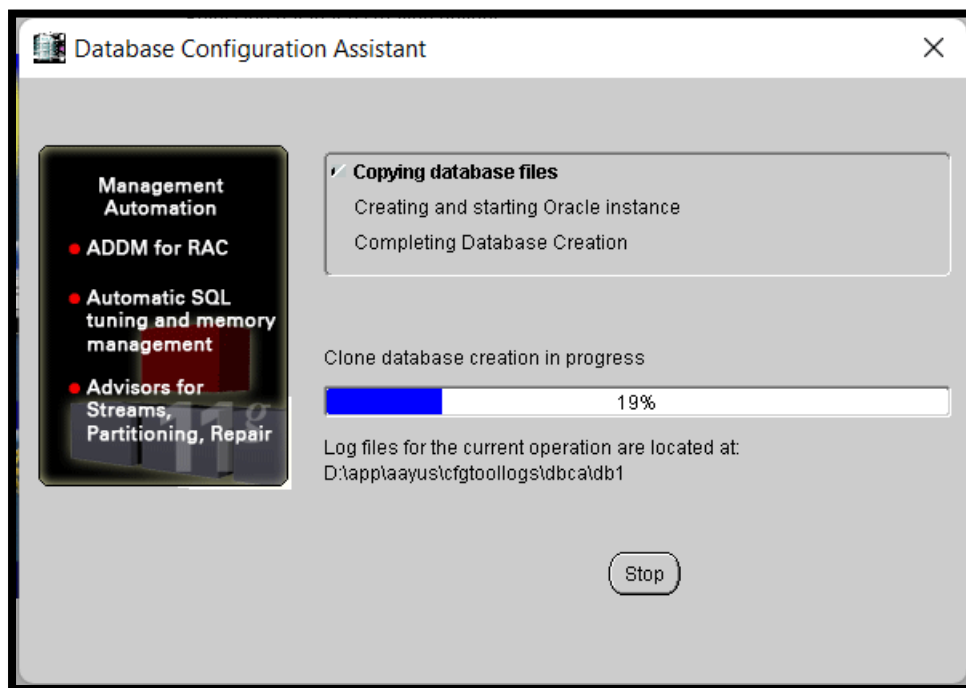
**Step 13:** No changes Needed, Click on Finish.



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.



# ADVANCED DATABASE SYSTEMS



## ADVANCED DATABASE SYSTEMS



Click on Exit and Done.....

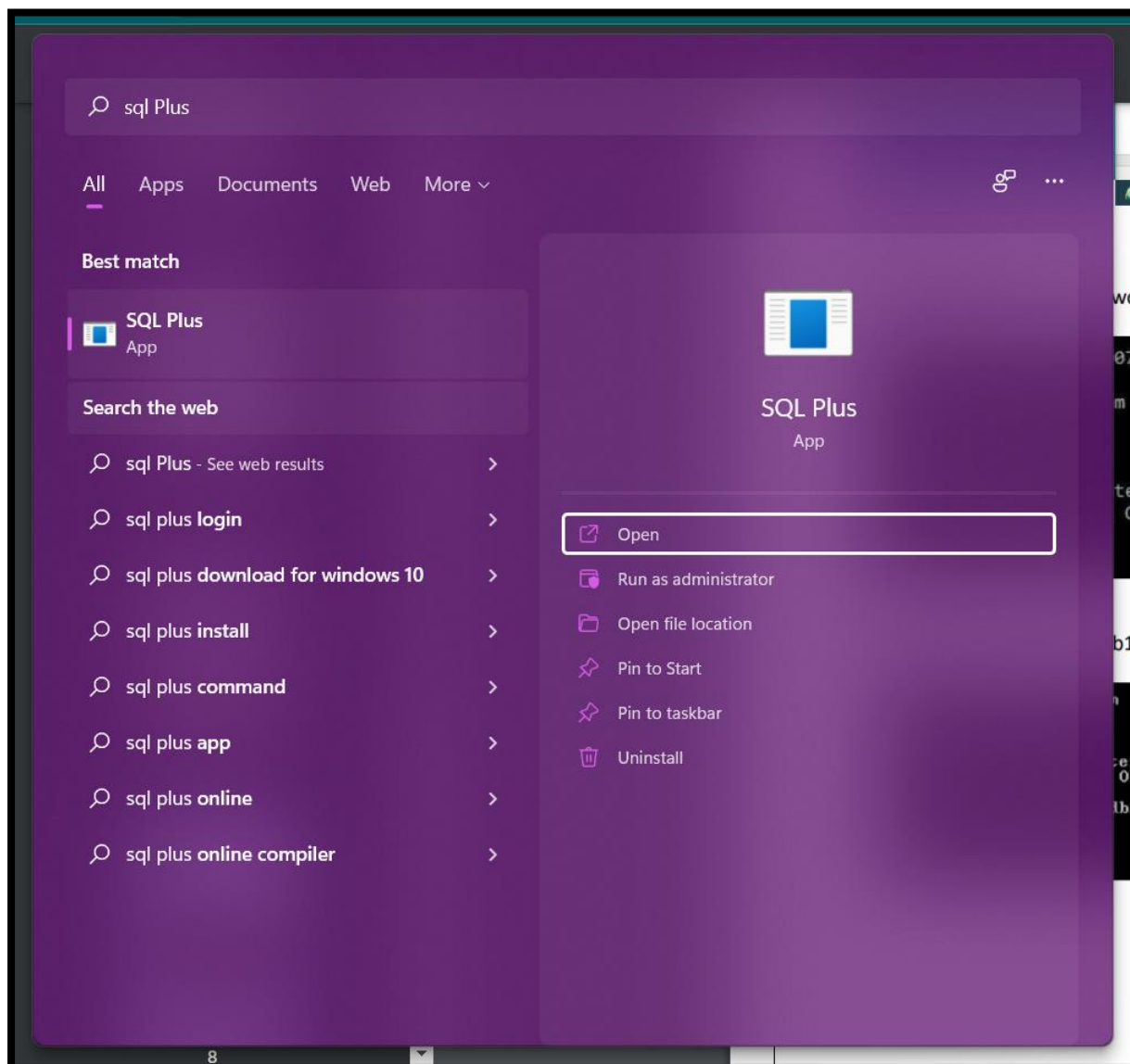
**Follow the Same Steps to create db2,**

Once done with Creating db1 and db2 .

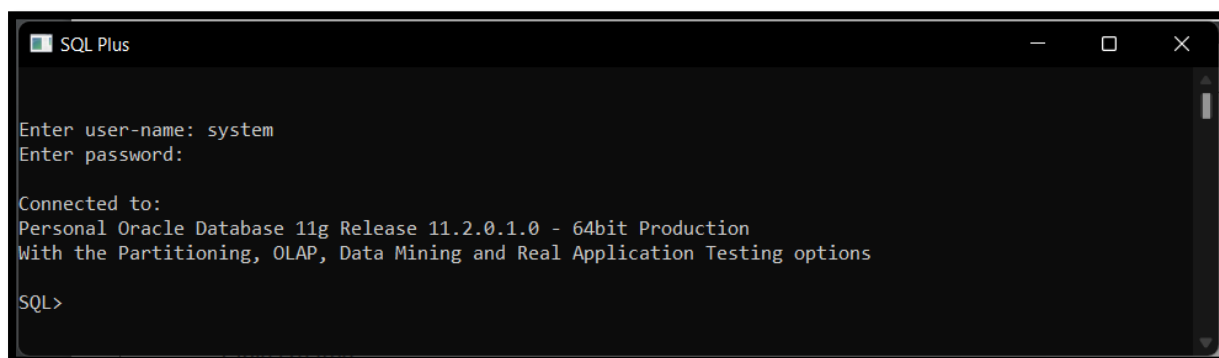
# ADVANCED DATABASE SYSTEMS

## Practical Implementation Steps:

- ✓ **Step 1:-** Open SQLPlus.



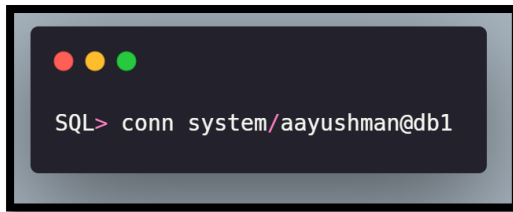
- ✓ **Step 2:** Connect to Your Database .





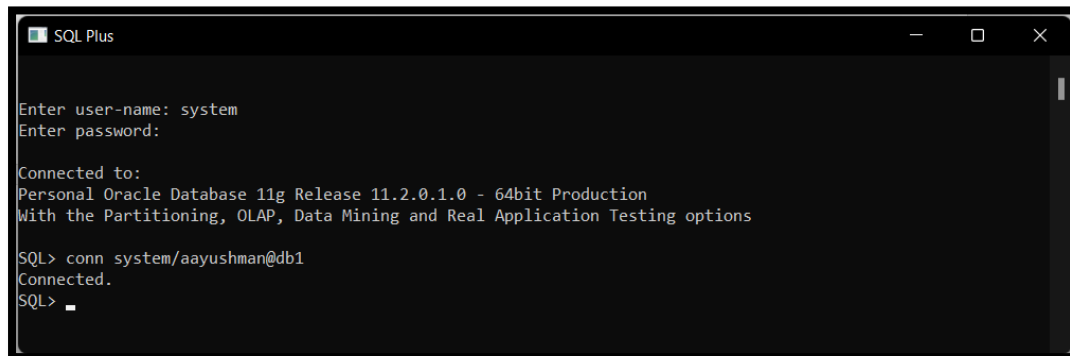
## ADVANCED DATABASE SYSTEMS

- ✓ **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> _
```

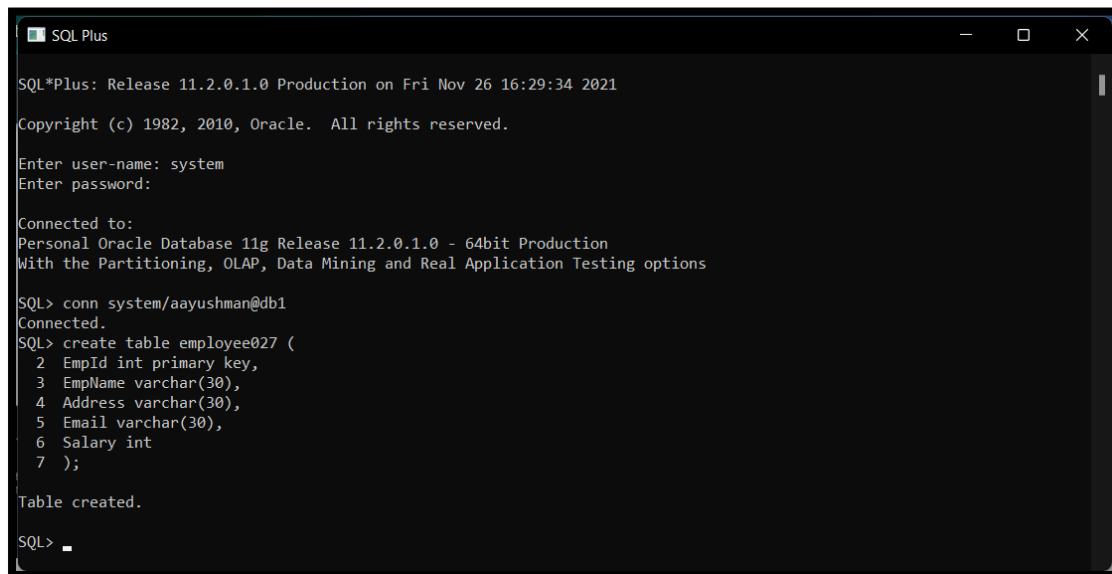
- ✓ **Step 4:** Create one table in database db1.



```
Create one table in database db1.

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

# ADVANCED DATABASE SYSTEMS



```
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:

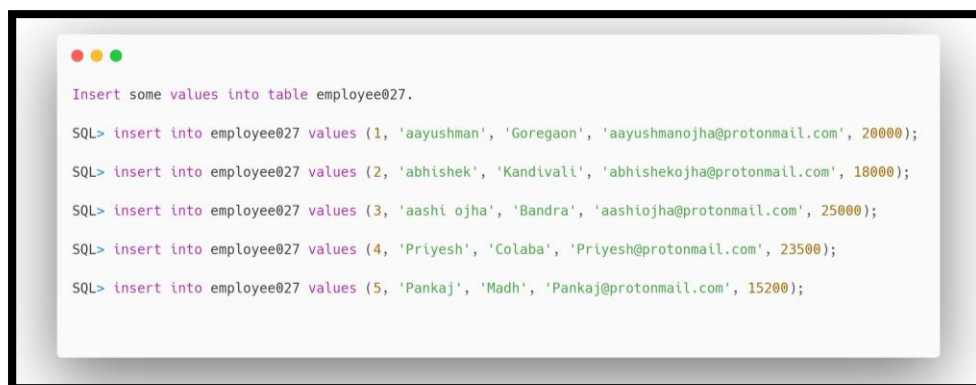
Connected to:
Personal Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

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);
```

# ADVANCED DATABASE SYSTEMS

```
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;
```

## ADVANCED DATABASE SYSTEMS

```
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.

# ADVANCED DATABASE SYSTEMS

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;
```

## ADVANCED DATABASE SYSTEMS

```
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

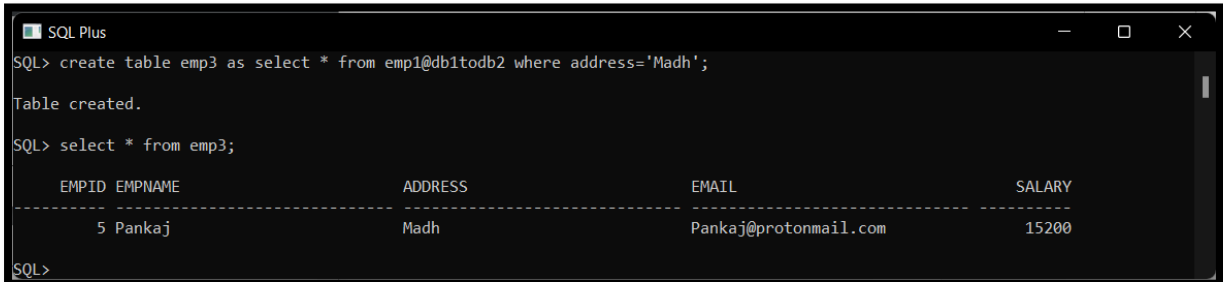
SQL> _
```

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

## ADVANCED DATABASE SYSTEMS

```
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 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.

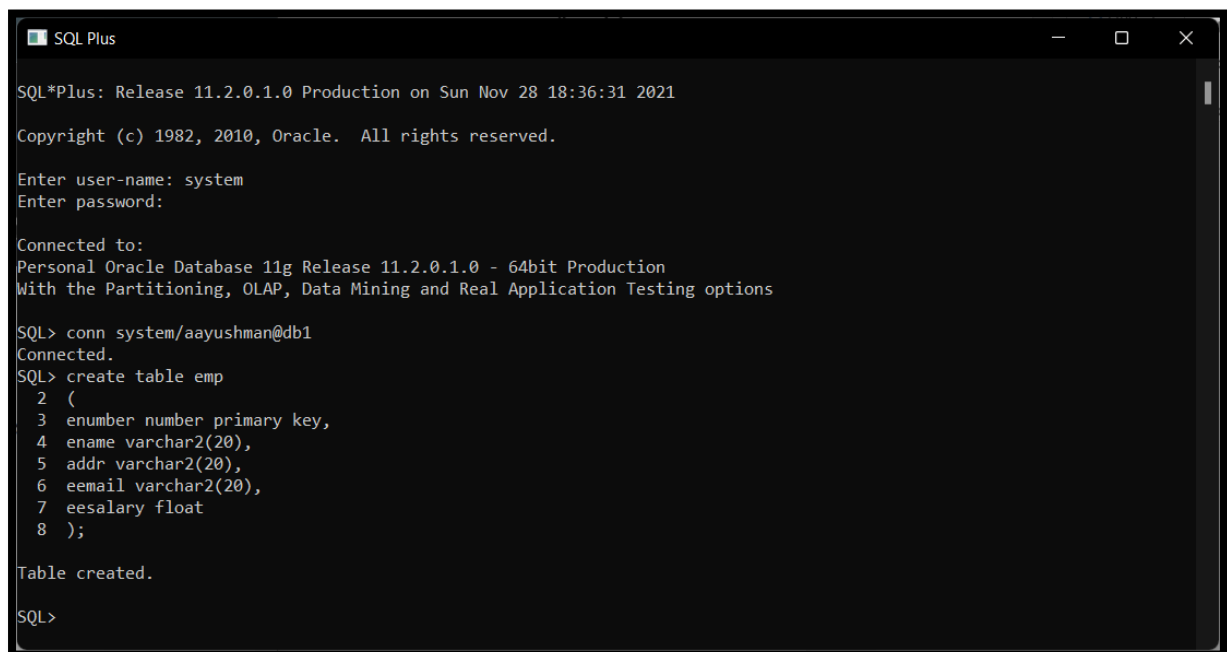
### Software Requirement:

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.



```
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>
```

# ADVANCED DATABASE SYSTEMS

## Step 2: Create Table in db2.

```
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 3: Create Database link.

```
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 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> _
```

# ADVANCED DATABASE SYSTEMS

## 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 email= :new.email,
10 esalary= :new.esalary
11 where enumber= :old.enumber;
12 end;
13 /

Trigger created.
```

## Step 6: Insert Values in Created Table.

```
SQL Plus
SQL> insert into emp values(111,'Aayushman','Bhayandar','aayush@yahoo.com',100000);
1 row created.

SQL> insert into emp values(112,'Sagar','Goregaon','sagar@yahoo.com',10400);
1 row created.

SQL> insert into emp values(113,'Aashi','Borivali','Aashi@yahoo.com',100000);
1 row created.

SQL> insert into emp values(114,'Ashish','Bhayandar','ash@yahoo.com',15000);
1 row created.

SQL> insert into emp values(115,'Tejas','Bhayandar','tej@yahoo.com',5000);
1 row created.

SQL> insert into emp values(116,'Shubham','Malad','shub@yahoo.com',100000);
1 row created.

SQL> insert into emp values(117,'Priyanka','Vasai','priya@yahoo.com',15000);
1 row created.

SQL> insert into emp values(118,'Swapna','Nalasopara','swap@yahoo.com',300000);
1 row created.

SQL> insert into emp values(119,'Samodaya','Goregaon','sam@yahoo.com',15000);
1 row created.

SQL> insert into emp values(120,'Aman','Jogeshwari','aman@yahoo.com',20000);_
```

# ADVANCED DATABASE SYSTEMS

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>
```

# ADVANCED DATABASE SYSTEMS

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;

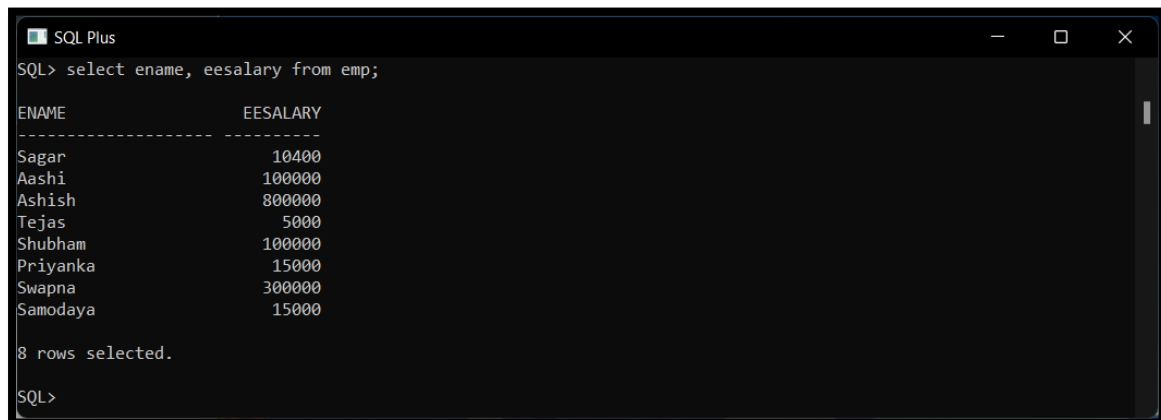
  ENUMBER  ENAME          ADDR          EEMAIL          EESALARY
-----
    112     Sagar      Goregaon    sagar@yahoo.com     10400
    113     Aashi      Borivali    Aashi@yahoo.com    100000
    114    Ashish      Bhayandar    ash@yahoo.com      800000
    115     Tejas      Bhayandar    tej@yahoo.com        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> _
```

## ADVANCED DATABASE SYSTEMS

3. Find the salary of all employees.

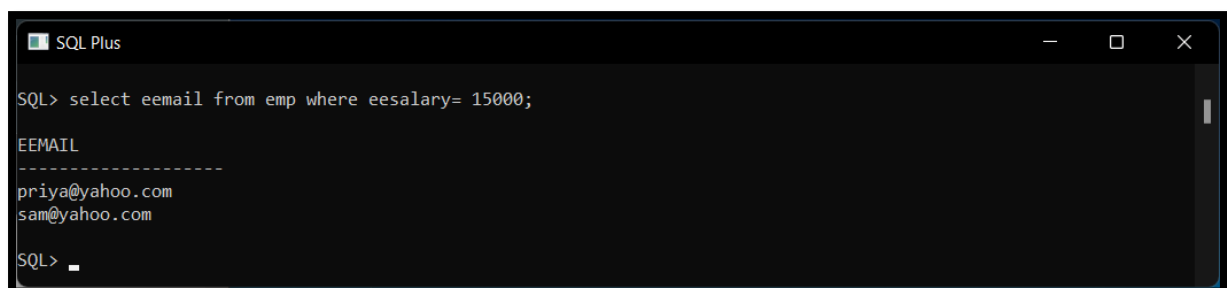


```
SQL> select ename, eesalary from emp;
```

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.

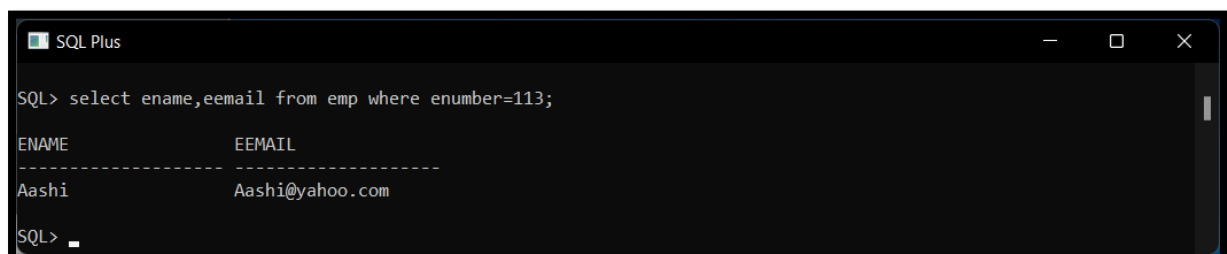


```
SQL> select eemail from emp where eesalary= 15000;
```

EEMAIL
priya@yahoo.com
sam@yahoo.com

```
SQL>
```

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

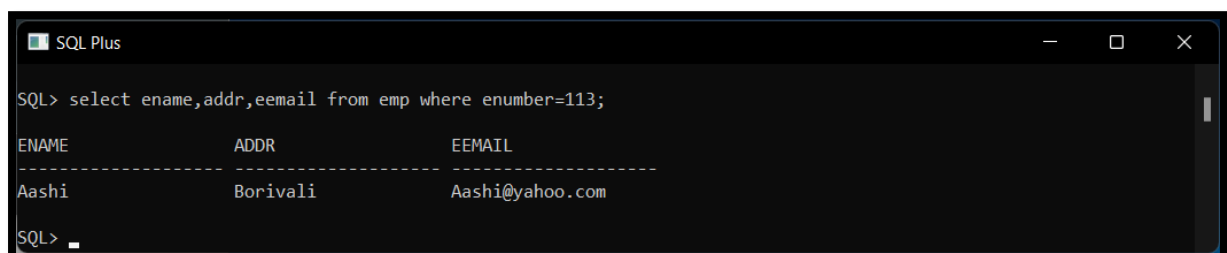


```
SQL> select ename,eemail from emp where enumber=113;
```

ENAME	EEMAIL
Aashi	Aashi@yahoo.com

```
SQL>
```

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



```
SQL> 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 NO 3

CRUD Operation on MongoDB

Abhishek Ojha  
Seat No 027

# ADVANCED DATABASE SYSTEMS

## Practical No: 3

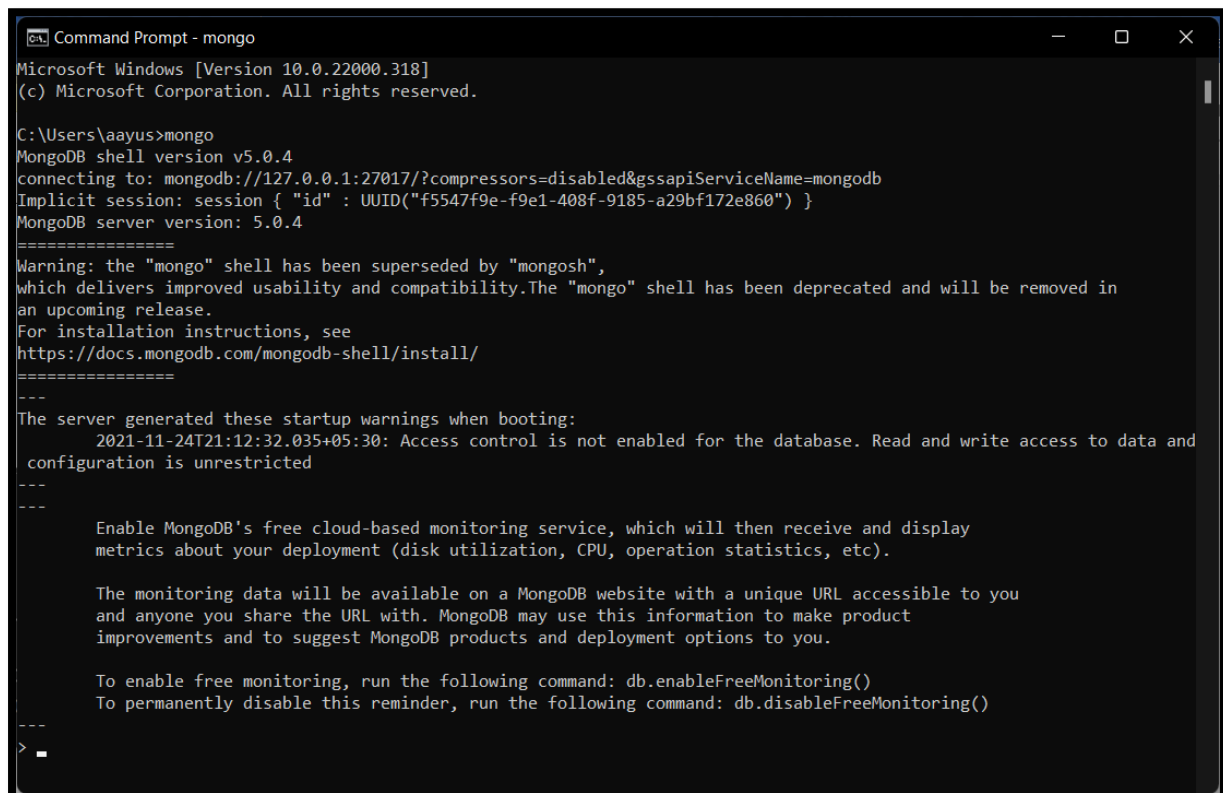
**Aim:** To perform CRUD Operation using MongoDB.

### 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.



```
Command Prompt - mongo
Microsoft Windows [Version 10.0.22000.318]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ayus>mongo
MongoDB shell version v5.0.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("f5547f9e-f9e1-408f-9185-a29bf172e860") }
MongoDB server version: 5.0.4
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-11-24T21:12:32.035+05:30: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

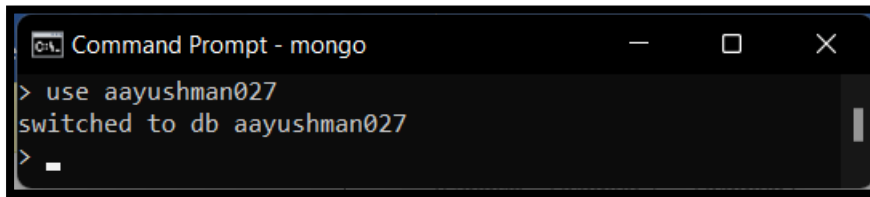
  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

- ✓ **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.



# ADVANCED DATABASE SYSTEMS



```
Command Prompt - mongo
> use aayushman027
switched to db aayushman027
>
```

## ✓ Step 3: Creating Collections and Inserting Values [C - Create]

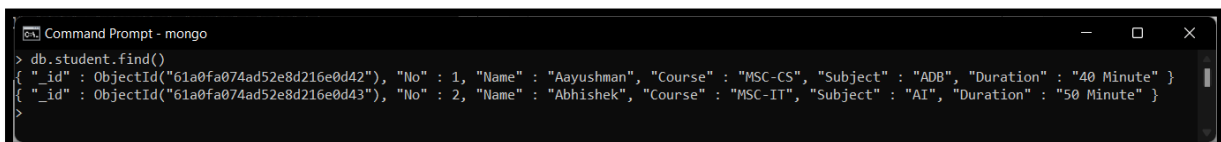
Creating a collection and inserting values can be done together. Here we have orcollection name as 'student '.



```
Command Prompt - mongo
> db.student.insert({name: "aayushman"})
WriteResult({ "nInserted" : 1 })
> db.student.insert(
... { no:2,
... name: "abhishek",
... Course:{CourseName: "MSC ComputerScience", Duration: "2 Years"},
... Address: {City: "Mumbai", State: "Maharashtra", Country: "India"}
... })
WriteResult({ "nInserted" : 1 })
>
```

## ✓ Step 4: Read Data from the Collections [R - Read]

To retrieve the inserted document.



```
Command Prompt - mongo
> db.student.find()
{ "_id" : ObjectId("61a0fa074ad52e8d216e0d42"), "No" : 1, "Name" : "Aayushman", "Course" : "MSC-CS", "Subject" : "ADB", "Duration" : "40 Minute" }
{ "_id" : ObjectId("61a0fa074ad52e8d216e0d43"), "No" : 2, "Name" : "Abhishek", "Course" : "MSC-IT", "Subject" : "AI", "Duration" : "50 Minute" }
>
```

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



```
Command Prompt - mongo
> db.student.update ({No : 2}, {$set: {"Name" : "Aashi"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.student.find()
{ "_id" : ObjectId("61a0fa074ad52e8d216e0d42"), "No" : 1, "Name" : "Aayushman", "Course" : "MSC-CS", "Subject" : "ADB", "Duration" : "40 Minute" }
{ "_id" : ObjectId("61a0fa074ad52e8d216e0d43"), "No" : 2, "Name" : "Aashi", "Course" : "MSC-IT", "Subject" : "AI", "Duration" : "50 Minute" }
>
```

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



```
Command Prompt - mongo
> db.student.remove({No : 2})
WriteResult({ "nRemoved" : 1 })
> db.student.find()
{ "_id" : ObjectId("61a0fa074ad52e8d216e0d42"), "No" : 1, "Name" : "Aayushman", "Course" : "MSC-CS", "Subject" : "ADB", "Duration" : "40 Minute" }
>
```

**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:

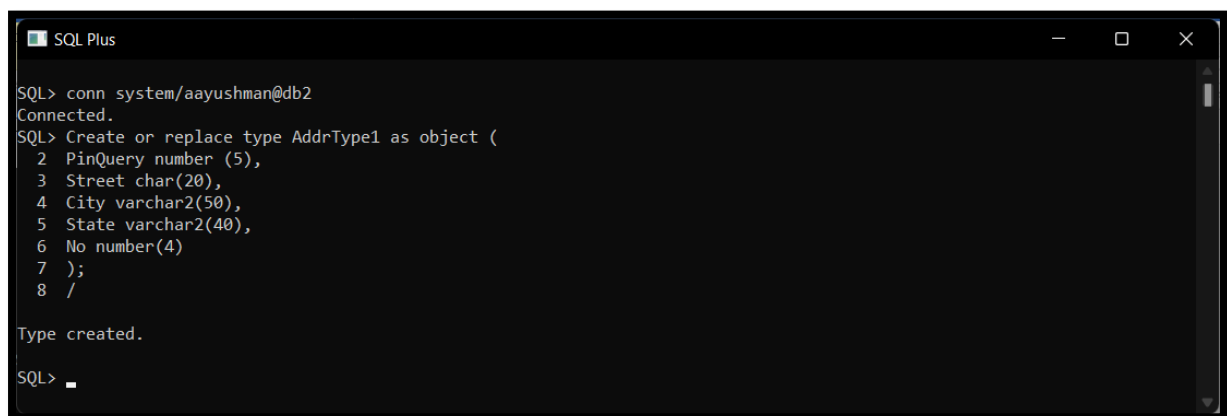
1. AddrType1 (PinQuery: number, Street :char, City : char, state :char) .
2. BranchType (address: AddrType1, phone1: integer,phone2: integer).
3. AuthorType (name:char,,addr AddrType1).
4. PublisherType (name: char, addr: AddrType1, branches: BranchTableType).
5. books(title: varchar, year : date, published\_by 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).



```
SQL Plus
SQL> conn system/aayushman@db2
Connected.
SQL> Create or replace type AddrType1 as object (
2 PinQuery number (5),
3 Street char(20),
4 City varchar2(50),
5 State varchar2(40),
6 No number(4)
7 );
8 /

Type created.
SQL> _
```

# ADVANCED DATABASE SYSTEMS

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

```
SQL Plus
SQL> create or replace type BranchType as object (
  2 Address AddrType1,
  3 Phone1 integer,
  4 Phone2 integer
  5 );
  6 /

Type created.

SQL>
```

```
SQL Plus
SQL> create or replace type BranchType as object (
  2 Address AddrType1,
  3 Phone1 integer,
  4 Phone2 integer
  5 );
  6 /

Type created.

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

Type created.

SQL> _
```

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

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

Type created.

SQL> _
```

```
SQL Plus
SQL> create table Authors of AuthorType;

Table created.

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

Type created.

SQL> _
```

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

# ADVANCED DATABASE SYSTEMS

```
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>
```

## ADVANCED DATABASE SYSTEMS

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

```
SQL Plus
SQL> insert into Publishers
  2 values('McGraw',AddrType1(7007,'LJstreet','mumbai','maharashtra',07), BranchTableType (BranchType ( AddrType1 (70
07,'K street','mumbai','maharashtra',1007), 4543545,8676775)));
1 row created.

SQL> > insert into Publishers values ('Tata',AddrType1(7008,'JW street','mumbai','maharashtra',27), BranchTableType (B
ranchType (AddrType1(1002,'DM street','nasik','maharashtra',1007), 456767,7675757)));
SP2-0734: unknown command beginning "> insert i..." - rest of line ignored.
SQL> insert into Publishers values ('Tata',AddrType1(7008,'JW street','mumbai','maharashtra',27), BranchTableType (Bra
nchType (AddrType1(1002,'DM street','nasik','maharashtra',1007), 456767,7675757)));
1 row created.

SQL> insert into Publishers values ('Nurali', AddrType1(7002,'ST street','pune','maharashtra',1007), BranchTableType (B
ranchType (AddrType1(1002,'SG street','pune','maharashtra',1007), 4543545,8676775)));
1 row created.

SQL> insert into Publishers values('Tata', AddrType1(6002,'Gold street','nasik','maharashtra',1007),BranchTableType (B
ranchType(AddrType1(6002,'South street','nasik','mha',1007), 4543545,8676775)));
1 row created.

SQL> _
```

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

```
SQL Plus
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='Enrique Iglesias';
2 rows created.

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

SQL> insert into books select 'c prog','25-may-1983', ref (pub),AuthorListType(ref(aut)) from Publishers pub,Authors a
ut where pub.name='Vipul' and aut.name='Travis Scott';
0 rows created.

SQL> insert into books select 'c prog','25-may-1983', ref (pub),AuthorListType(ref(aut)) from Publishers pub,Authors a
ut where pub.name='Nurali' and aut.name='Travis Scott';
1 row created.

SQL> _
```

**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;*

## ADVANCED DATABASE SYSTEMS

```
SQL Plus
SQL> select a.name from Authors a, Publishers p where a.Address.pinQuery = p.Address.pinQuery;
NAME
-----
Vikas
Ashish
Richard
SQL> _
```

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

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

```
SQL Plus
SQL> Select title from books b where 1 <= (select count(*) from table(b.authors));
TITLE
-----
IP
IP
IP
IP
ADBMS
c prog
6 rows selected.
```

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

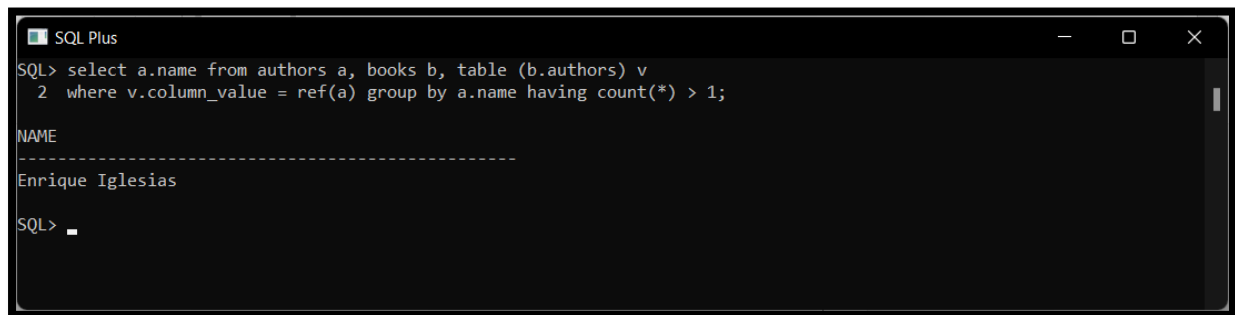
*Select p.name from publishers p, table (p.branches) group by p.name having count(\*) >= all (select count(\*) from publishers p, table(p.branches) group by name);*

```
SQL Plus
6 rows selected.
SQL> Select p.name from publishers p, table (p.branches) group by p.name having count(*) >= all (select count(*) from publishers p, table(p.branches) group by name);
NAME
-----
Tata
SQL>
```

**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;*

## ADVANCED DATABASE SYSTEMS



```
SQL Plus
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;

NAME
-----
Enrique Iglesias

SQL> _
```

**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  
where v.column\_value = ref(a) group by title having count(\*) >  
1;*



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

TITLE
-----
IP

SQL> _
```

**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.

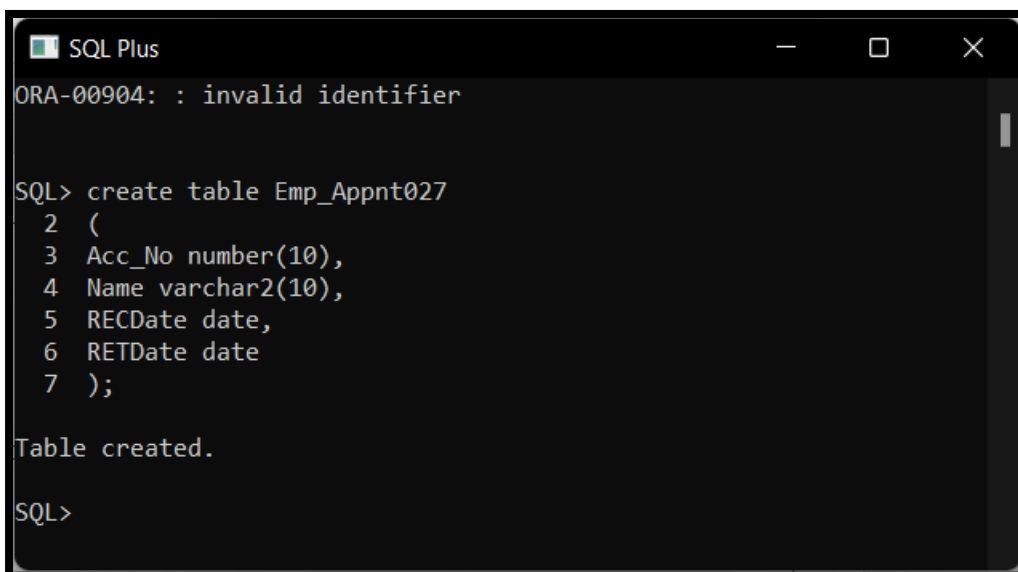
### Software Requirement:

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
ORA-00904: : invalid identifier

SQL> create table Emp_Appnt027
2  (
3  Acc_No number(10),
4  Name varchar2(10),
5  RECDate date,
6  RETDate date
7  );

Table created.

SQL>
```

## ADVANCED DATABASE SYSTEMS

```
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>
```

## ADVANCED DATABASE SYSTEMS

### 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.

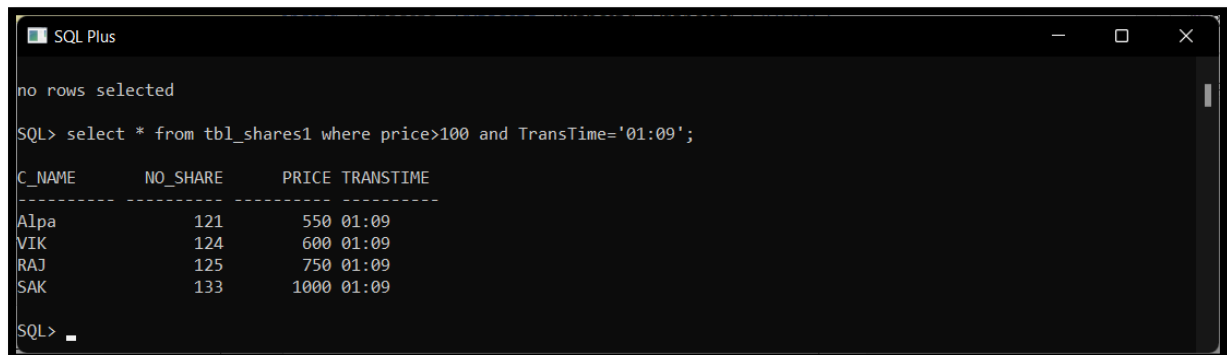
```
SQL Plus
Table created.
SQL> insert into tbl_shares1 values('Aakash', 123,500,Default);
1 row created.
SQL> insert into tbl_shares1 values('Alpa', 121,550,Default)
2 /
1 row created.
SQL> insert into tbl_shares1 values('VIK', 124,600,Default);
1 row created.
SQL> insert into tbl_shares1 values('RAJ', 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>
```

## ADVANCED DATABASE SYSTEMS

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

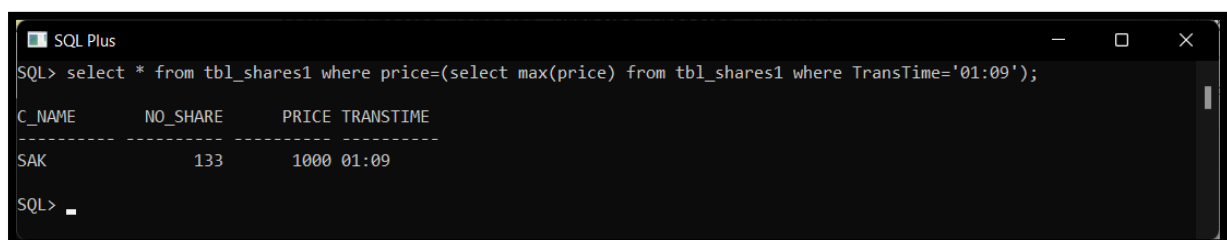


```
SQL Plus
no rows selected
SQL> select * from tbl_shares1 where price>100 and TransTime='01:09';
```

C_NAME	NO_SHARE	PRICE	TRANSTIME
Alpa	121	550	01:09
VIK	124	600	01:09
RAJ	125	750	01:09
SAK	133	1000	01:09

```
SQL> _
```

7. Display the records where price=(select max(price) from tbl\_shares1 where TransTime='02:04');



```
SQL Plus
SQL> select * from tbl_shares1 where price=(select max(price) from tbl_shares1 where TransTime='01:09');
```

C_NAME	NO_SHARE	PRICE	TRANSTIME
SAK	133	1000	01:09

```
SQL> _
```

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

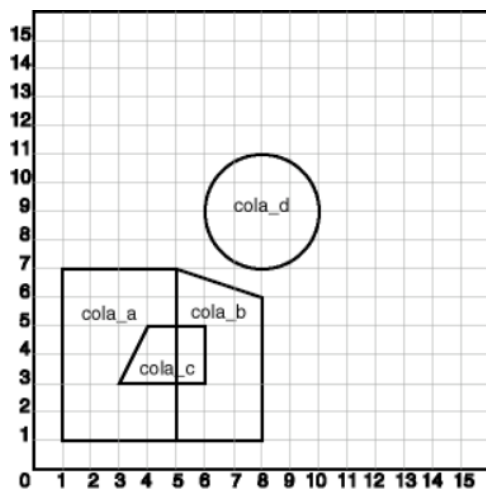
# PRACTICAL NO 6

Spatial Database

# 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:

- Find the topological intersection of two geometries.
- Find whether two geometric figures are equivalent to each other.
- Find the areas of all different locations.
- Find the area of only one location.
- 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 .

```
SQL Plus
SQL> create table cola_mrp
2  (
3  mkt_id number primary key,
4  name varchar(20),
5  shape SDO_Geometry
6  );
Table created.
```

## ADVANCED DATABASE SYSTEMS

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.

```
SQL Plus
SQL> insert into cola_mrp values (1, 'cola_a', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,3),
3 SDO_ORDINATE_ARRAY(1,1,5,7)
4 ));
1 row created.
SQL>
```

```
SQL Plus
1 row created.
SQL> insert into cola_mrp values (2, 'cola_b', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,1),
3 SDO_ORDINATE_ARRAY(5,1,8,1,8,6,5,7,5,1)
4 ));
1 row created.
SQL>
```

```
SQL Plus
SQL> insert into cola_mrp values (3, 'cola_c', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,1),
3 SDO_ORDINATE_ARRAY(3,3,6,3,6,5,4,5,3,3)
4 ));
1 row created.
SQL>
```

```
SQL Plus
SQL> insert into cola_mrp values (4, 'cola_d', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,4),
3 SDO_ORDINATE_ARRAY(8,7,10,9,8,11)
4 ));
1 row created.
SQL>
```



## ADVANCED DATABASE SYSTEMS

```
SQL Plus
SQL> insert into cola_mrp values (1, 'cola_a', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,3),
3 SDO_ORDINATE_ARRAY(1,1,5,7)
4 ));
1 row created.

SQL> insert into cola_mrp values (2, 'cola_b', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,1),
3 SDO_ORDINATE_ARRAY(5,1,8,1,8,6,5,7,5,1)
4 ));
1 row created.

SQL> insert into cola_mrp values (3, 'cola_c', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,1),
3 SDO_ORDINATE_ARRAY(3,3,6,3,6,5,4,5,3,3)
4 ));
1 row created.

SQL> insert into cola_mrp values (4, 'cola_d', SDO_GEOMETRY(2003, NULL, NULL,
2 SDO_ELEM_INFO_ARRAY(1,1003,4),
3 SDO_ORDINATE_ARRAY(8,7,10,9,8,11)
4 ));
1 row created.

SQL> _
```

### 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**).

```
SQL Plus
SQL> insert into user_sdo_geom_metadata
2 (
3 Table_Name,
4 Column_Name,
5 DimInfo,
6 Srid) values ('cola_mrp', 'shape',
7 SDO_DIM_ARRAY(
8 SDO_DIM_ELEMENT('X',0,20,0.0005),
9 SDO_DIM_ELEMENT('Y',0,20,0.0005)), NULL
10 );
1 row created.

SQL>
```

### 4. CREATE THE SPATIAL INDEX.

```
SQL Plus
SQL> create index cola_spatial_idx
2 ON cola_mrp(shape)
3 INDEXTYPE IS MDSYS.SPATIAL_INDEX;

Index created.

SQL> _
```

### 5. PERFORM SOME SPATIAL QUERIES

Return the topological intersection of two geometries.

## ADVANCED DATABASE SYSTEMS

```
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> _
```

## ADVANCED DATABASE SYSTEMS

Return the distance between two geometries.

```
SQL Plus
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)
-----
.846049894

SQL>
```

Is a geometry valid?

```
SQL Plus
SQL> SELECT c.name, SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT(c.shape, 0.005)
2      FROM cola_mrp c WHERE c.name = 'cola_c';

NAME
-----
SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT(C.SHAPE,0.005)
-----
cola_c
TRUE

SQL>
```

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

```
SQL Plus
SQL> CREATE TABLE val_results (sdo_rowid ROWID, result VARCHAR2(2000));

Table created.

SQL> CALL SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('COLA_MRP', 'SHAPE',
2      'VAL_RESULTS', 2);

Call completed.

SQL> SELECT * from val_results;

SDO_ROWID
-----
RESULT
```

**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 Joining. 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 3<sup>rd</sup> record from the table and display the name of an employee.
5. Delete 4<sup>th</sup> record from the table.

### Software Requirements:

Oracle 11g Express Edition, Any browser.

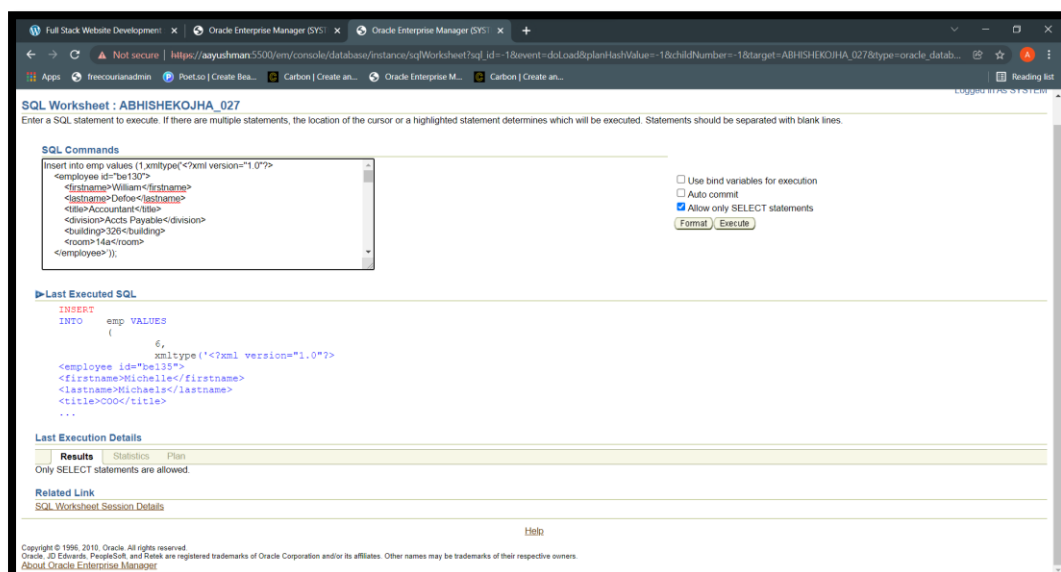
### Practical Implementation:

#### 1. Create Table Employee.



```
SQL Plus
Connected.
SQL> create table emp
2 (
3   emp_id int,
4   emp_spec xmltype
5 );
Table created.
```

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



```
SQL Commands
Insert into emp values (1,xmlltypex('<?xml version="1.0"?>
<employee id="be130">
  <firstname>William</firstname>
  <lastname>Deha</lastname>
  <title>Accountant</title>
  <division>Acts Payable</division>
  <building>326</building>
  <room>14a</room>
</employee> ));

Last Executed SQL
INSERT
INTO emp VALUES
(
6,
xmlltypex('<?xml version="1.0"?>
<employee id="be135">
  <firstname>Michelle</firstname>
  <lastname>Michaels</lastname>
  <title>COO</title>
...

Last Execution Details
Results Statistics Plan
Only SELECT statements are allowed.
Related Link
SQL Worksheet Session Details
Help
```

# ADVANCED DATABASE SYSTEMS

```
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>
```

# ADVANCED DATABASE SYSTEMS

```
SQL Plus

SQL> Insert into emp values (4,xmotype('<?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,xmotype('<?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,xmotype('<?xml version="1.0"?>
2      <employee id="emp06">
3        <firstname>Baila</firstname>
4        <lastname>Connigo</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>
```

## ADVANCED DATABASE SYSTEMS

### 3. Get the first name:

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

X.EMP_SPEC.EXTRACT('///FIRSTNAME/TEXT()').GETSTRINGVAL()
-----
Aayushman
Joye
Enrique
Sandra
Steve
Baila

6 rows selected.

SQL> _
```

### 4. Get the first name and room number.

```
SQL Plus
SQL> 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
-----
Aayushman
11g
Joye
19
Enrique
18

EMP_NAME
-----
ROOM_NO
-----
Sandra
22
Steve
24
Baila
264

6 rows selected.

SQL> _
```



# ADVANCED DATABASE SYSTEMS

## 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 6<sup>th</sup> 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>Connigo</lastname>
5 <title>C00</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> .
```

# ADVANCED DATABASE SYSTEMS

## 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.