e-Journal

on

ADVANCED DATABASE SYSTEMS

SUBMITTED BY

KALLIL RAHUL RAVIDNRAN

ROLL NO: 05

Submitted in partial fulfillment of the requirement for

Qualifying

M.Sc. Part I Semester II Examination

2018-19

Department of Information Technology

Ramniranjan Jhunjhunwala College

Station Road, Ghatkopar (w), Mumbai-86

Hindi Vidya Prachar Samiti's

RAMNIRANJAN JHUNJHUNWALA COLLEGE

(AUTONOMOUS)

Opposite Ghatkopar Railway Station, Ghatkopar West, Mumbai-400086

CERTIFICATE

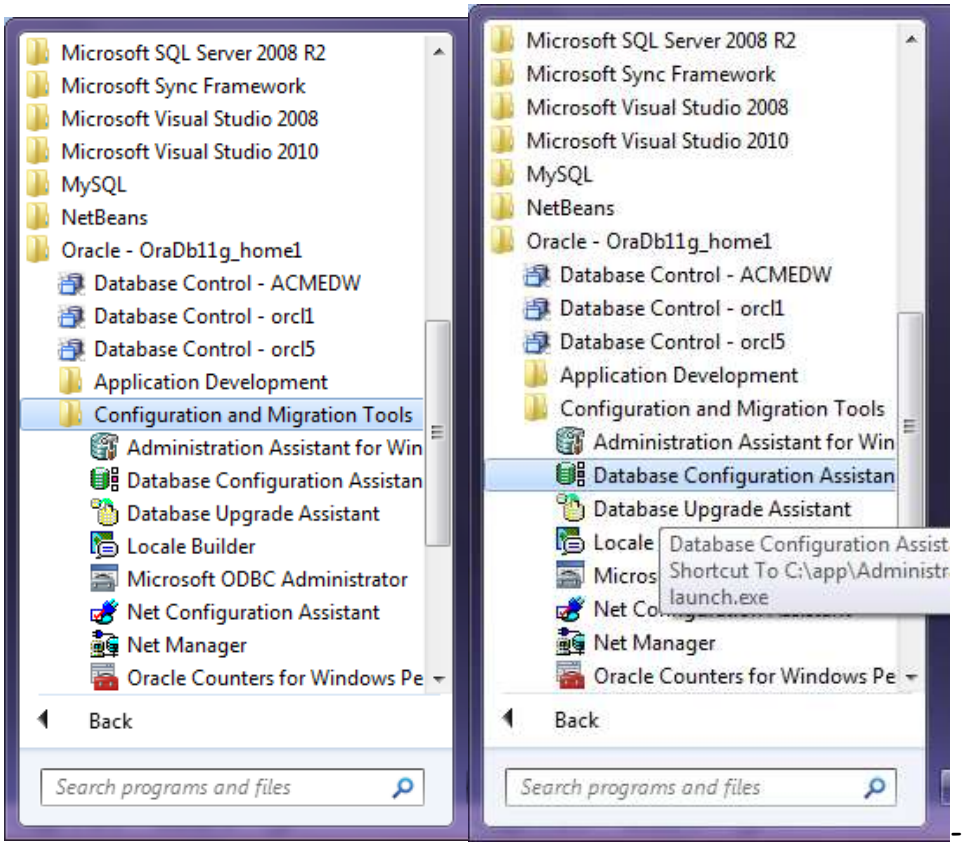
This is to certify that Mr. KALLIL RAHUL RAVINDRAN with Seat No. 05 has successfully completed the necessary course of experiments in the subject of **ADVANCED DATABASE SYSTEMS** during the academic year **2018 – 2019** complying with the requirements of **RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE AND COMMERCE**, for the course of **M.Sc. (IT)** semester -II.

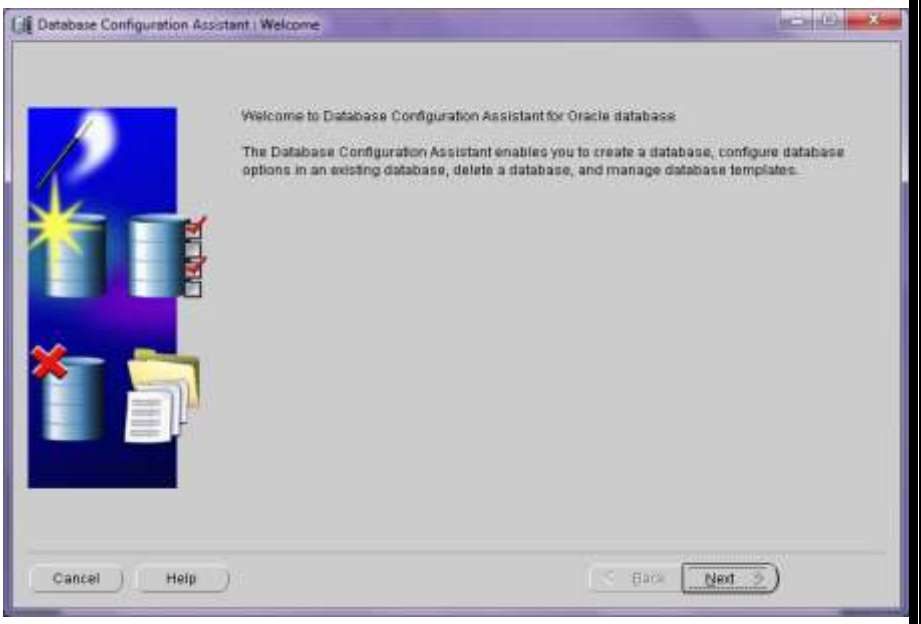
Internal Examiner Date: _____

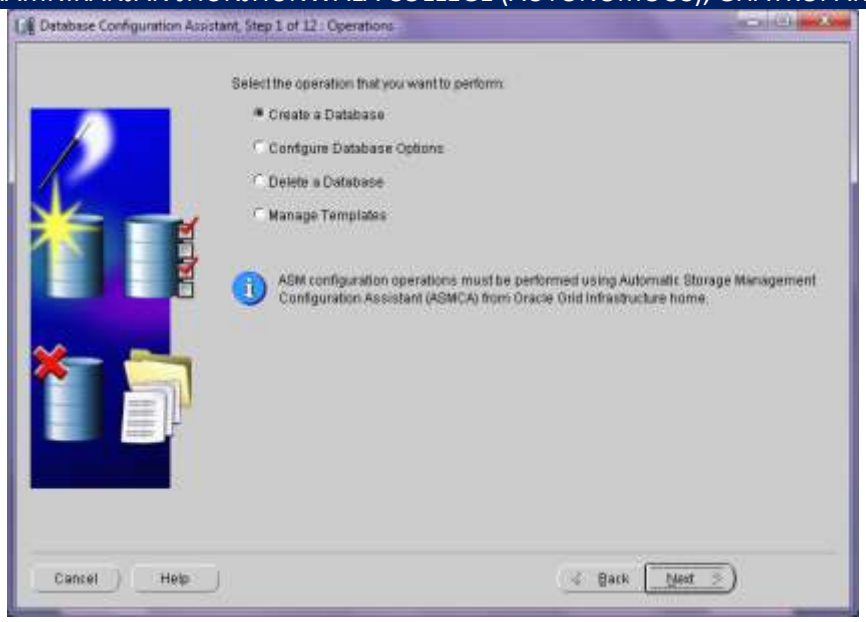
Head of Department College Seal External Examiner

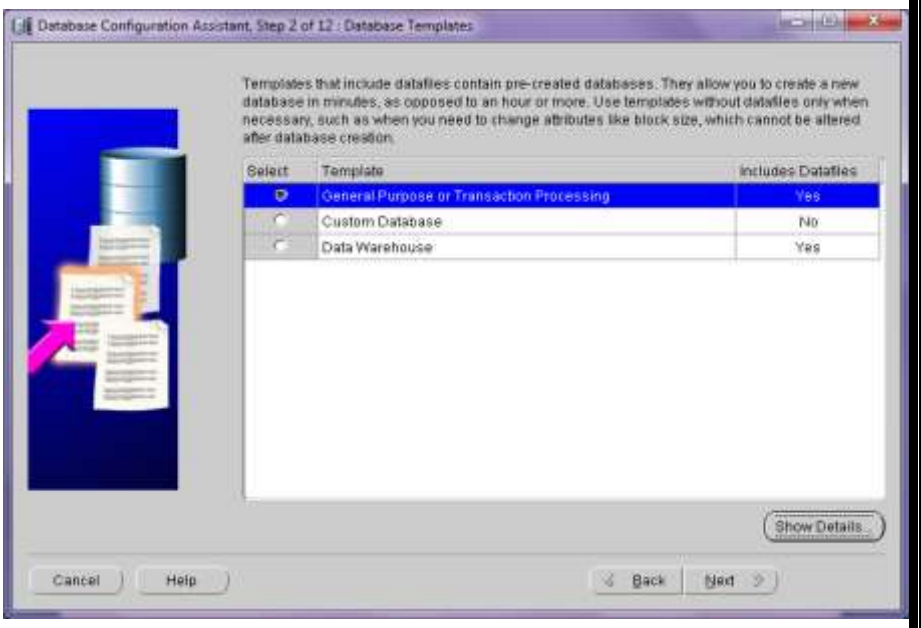
CONTENTS

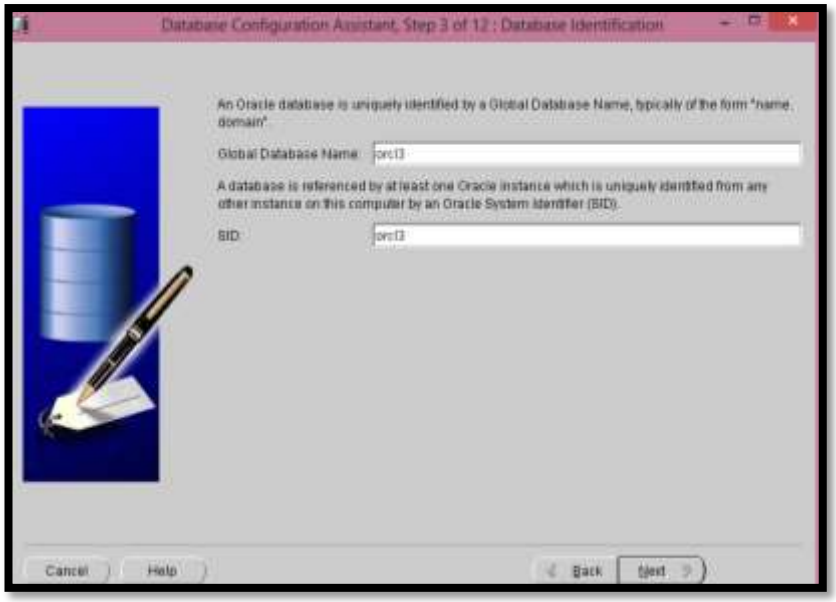
Database Creation Steps:

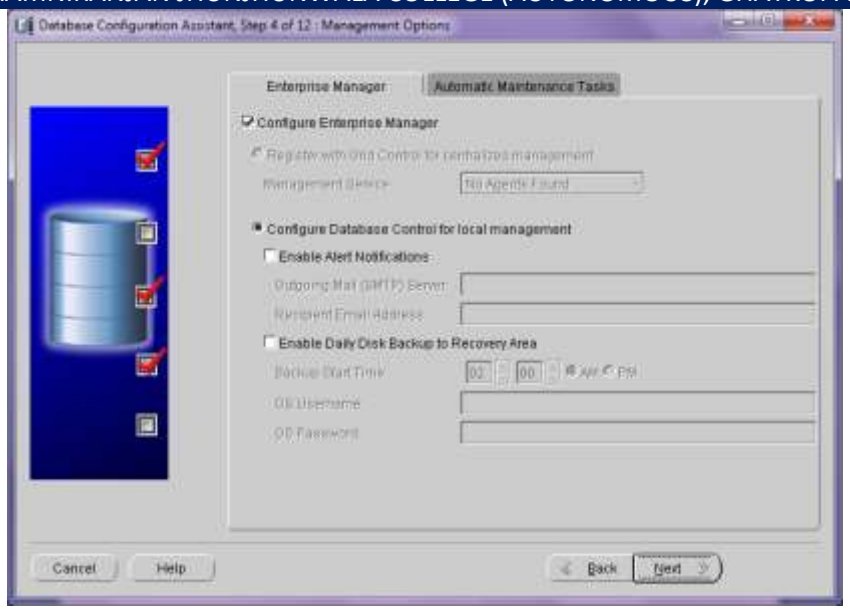


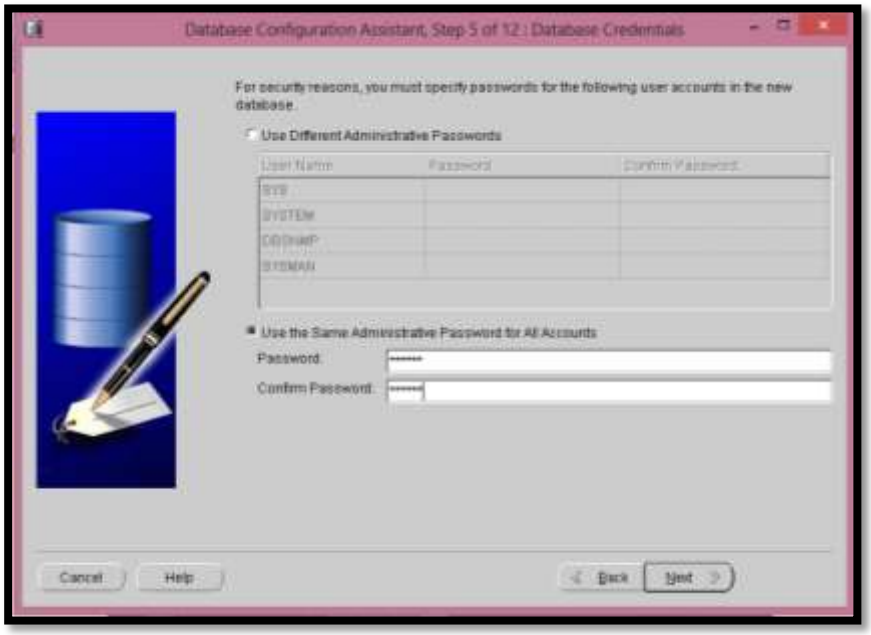



Database Configuration Assistant, Step 1 of 12 : Operations
Select the operation that you want to perform
Create a Database
Configure Database Options
Delete a Database
Manage Templates
ASM configuration operations must be performed using Automatic Storage Management Configuration Assistant (ASMCA) from Oracle Grid Infrastructure home.
Cancel
Help
Back
Next


Database Configuration Assistant, Step 2 of 12 : Database Templates
Templates that include datafiles contain pre-created databases. They allow you to create a new database in minutes, as opposed to an hour or more. Use templates without datafiles only when necessary, such as when you need to change attributes like block size, which cannot be altered after database creation.
Select
Template
Includes Datafiles
General Purpose or Transaction Processing
Yes
Custom Database
No
Data Warehouse
Yes
Show Details
Cancel
Help
Back
Next

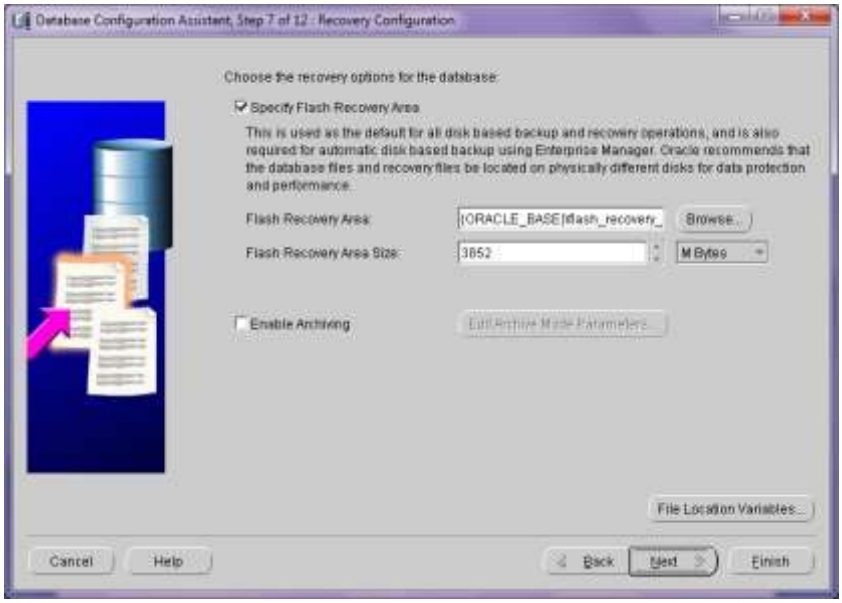

Database Configuration Assistant, Step 3 of 12 : Database Identification
An Oracle database is uniquely identified by a Global Database Name, typically of the form "name.domain".
Global Database Name:
orcl3
A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).
SID:
orcl3
Cancel
Help
Back
Next

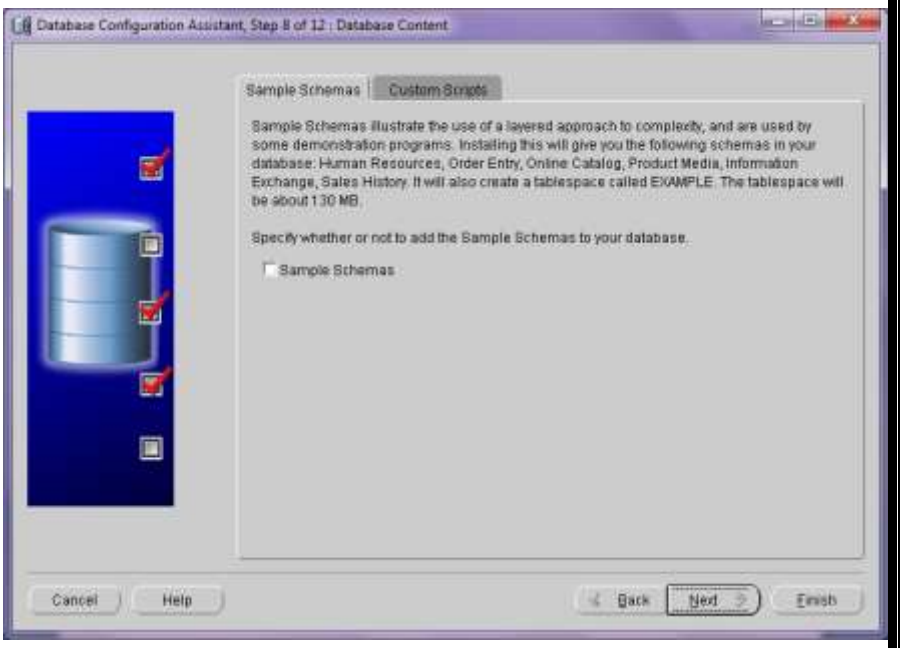

Database Configuration Assistant, Step 4 of 12 : Management Options
Enterprise Manager
Automatic Maintenance Tasks
Configure Enterprise Manager
Configure Database Control for local management
Enable Alert Notifications
Enable Daily Disk Backup to Recovery Area
Cancel
Help
Back
Next

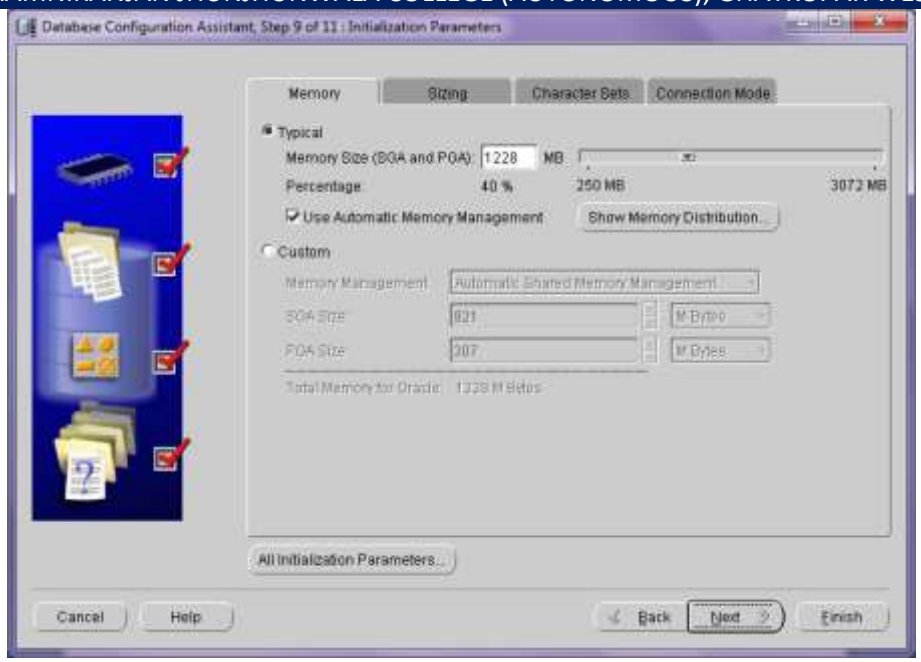

Database Configuration Assistant, Step 5 of 12 : Database Credentials
For security reasons, you must specify passwords for the following user accounts in the new database.
Use Different Administrative Passwords
Use the Same Administrative Password for All Accounts
Password:
Confirm Password:
Cancel
Help
Back
Next

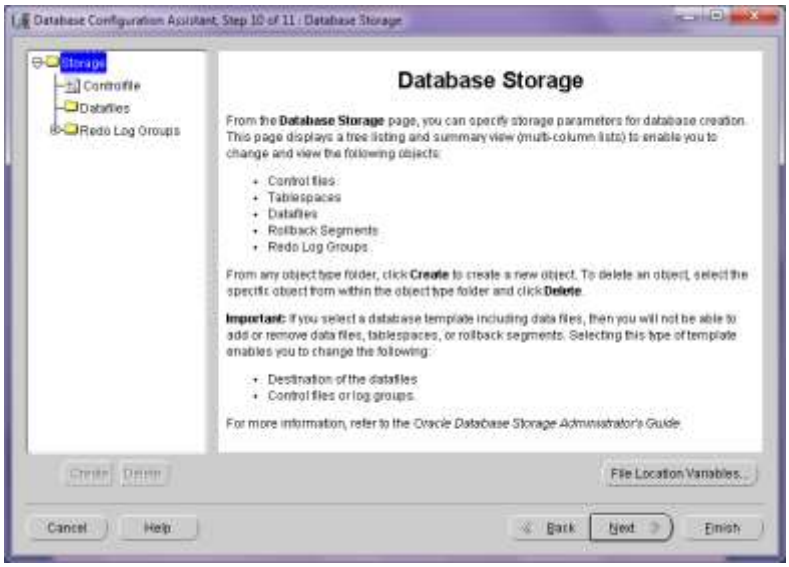
Password:

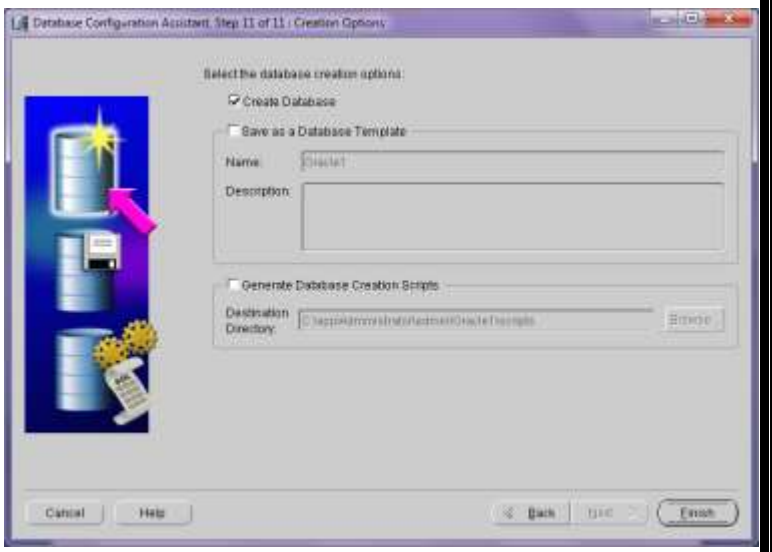

Database Configuration Assistant, Step 6 of 12 : Database File Locations
Specify storage type and locations for database files.
Storage Type:
File System
Storage Locations:
Use Database File Locations from Template
Use Common Location for All Database Files
Use Oracle-Managed Files
If you want to specify 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.
File Location Variables...
Cancel
Help
Back
Next
Finish


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:
{ORACLE_BASE}\flash_recovery_
Browse...
Flash Recovery Area Size:
3852
M Bytes
Enable Archiving
File Location Variables...
Cancel
Help
Back
Next
Finish

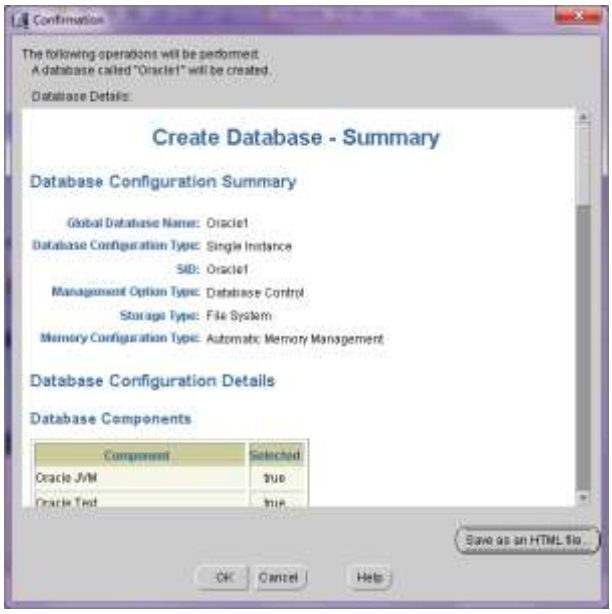

Database Configuration Assistant, Step 8 of 12 : Database Content
Sample Schemas
Custom Scripts
Sample Schemas illustrate the use of a layered approach to complexity, and are used by some demonstration programs. Installing this will give you the following schemas in your database: Human Resources, Order Entry, Online Catalog, Product Media, Information Exchange, Sales History. It will also create a tablespace called EXAMPLE. The tablespace will be about 130 MB.
Specify whether or not to add the Sample Schemas to your database.
Sample Schemas
Cancel
Help
Back
Next
Finish


Database Configuration Assistant, Step 9 of 11 : Initialization Parameters
Memory
Sizing
Character Sets
Connection Mode
Typical
Memory Size (SGA and PGA): 1228 MB
Percentage 40 %
250 MB
3072 MB
Use Automatic Memory Management
Show Memory Distribution...
Custom
Memory Management
Automatic Shared Memory Management
SGA Size
PGA Size
Total Memory for Oracle
All Initialization Parameters...
Cancel
Help
Back
Next
Finish

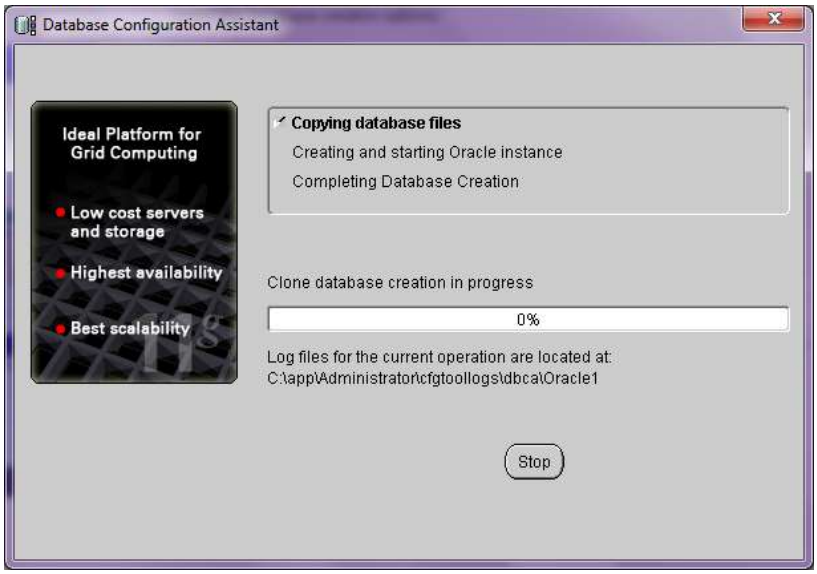

Database Configuration Assistant, Step 10 of 11 : Database Storage
Storage
Controlfile
Datafiles
Redo Log Groups
Database Storage
From the Database Storage page, you can specify storage parameters for database creation. This page displays a tree listing and summary view (multi-column lists) to enable you to change and view the following objects:
Control files
Tablespaces
Datafiles
Rollback Segments
Redo Log Groups
From any object type folder, click Create to create a new object. To delete an object, select the specific object from within the object type folder and click Delete.
Important: If you select a database template including data files, then you will not be able to add or remove data files, tablespaces, or rollback segments. Selecting this type of template enables you to change the following:
Destination of the datafiles
Control files or log groups
For more information, refer to the Oracle Database Storage Administrator's Guide
Create
Delete
File Location Variables...
Cancel
Help
Back
Next
Finish

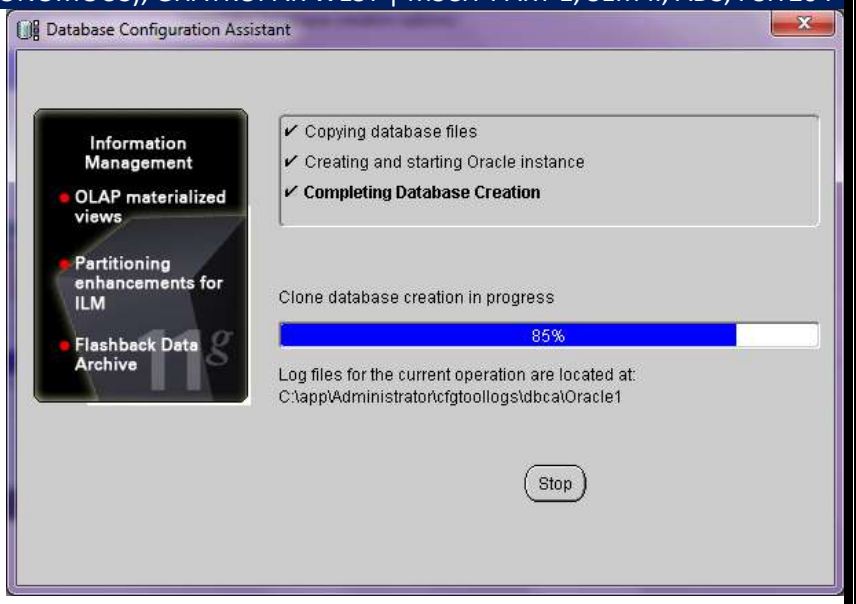

Database Configuration Assistant, Step 11 of 11 : Creation Options
Select the database creation options:
Create Database
Save as a Database Template
Name:
Description:
Generate Database Creation Scripts
Destination Directory:
Cancel
Help
Back
Finish

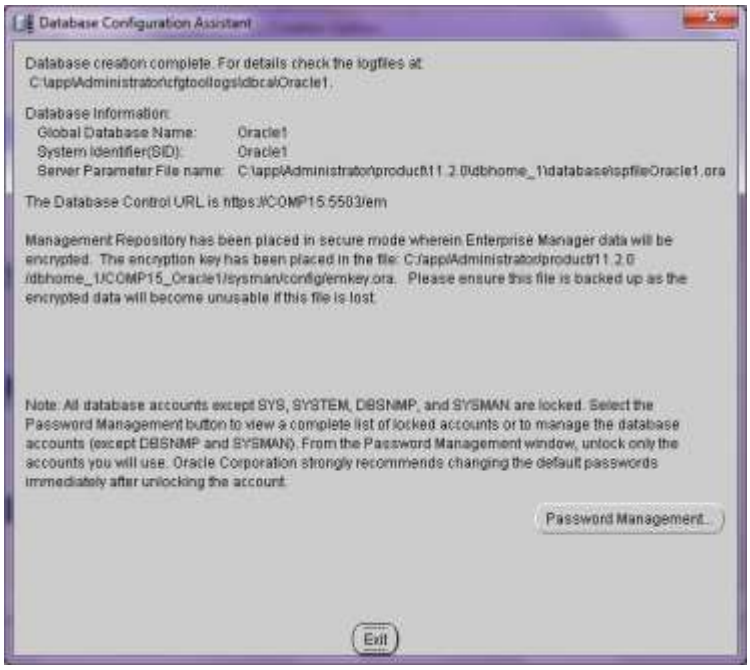
Click Finish



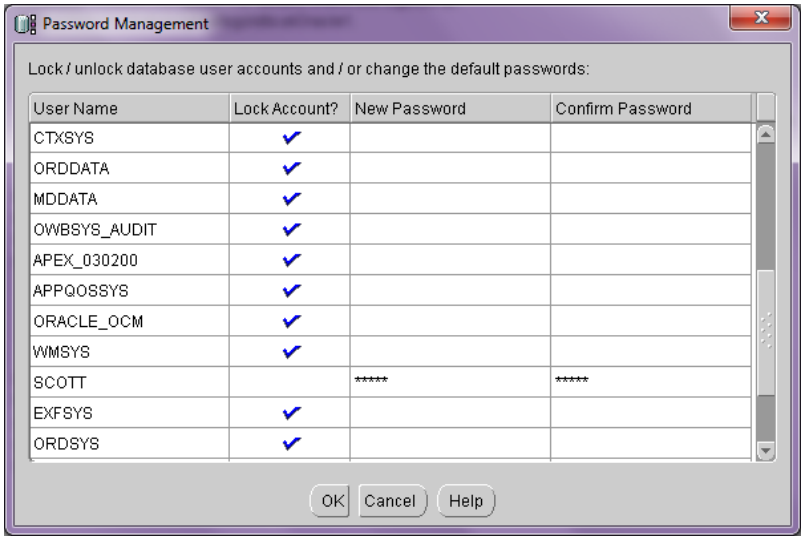
Click OK then you wil get above screen

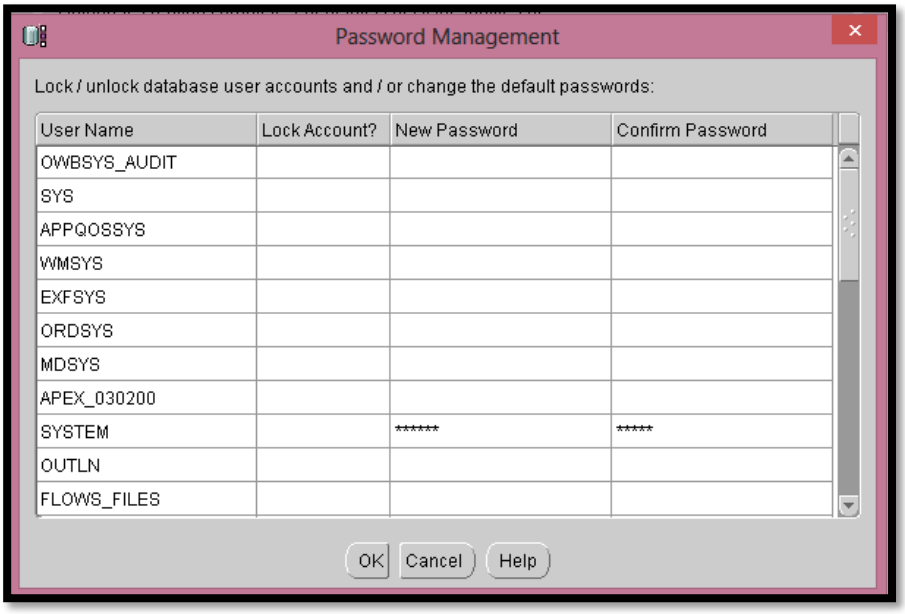






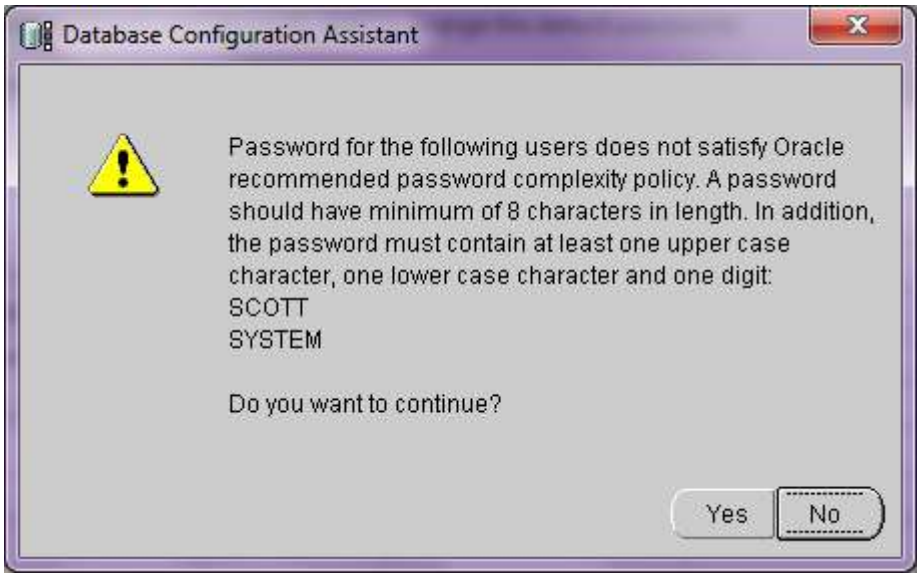
Click on password management



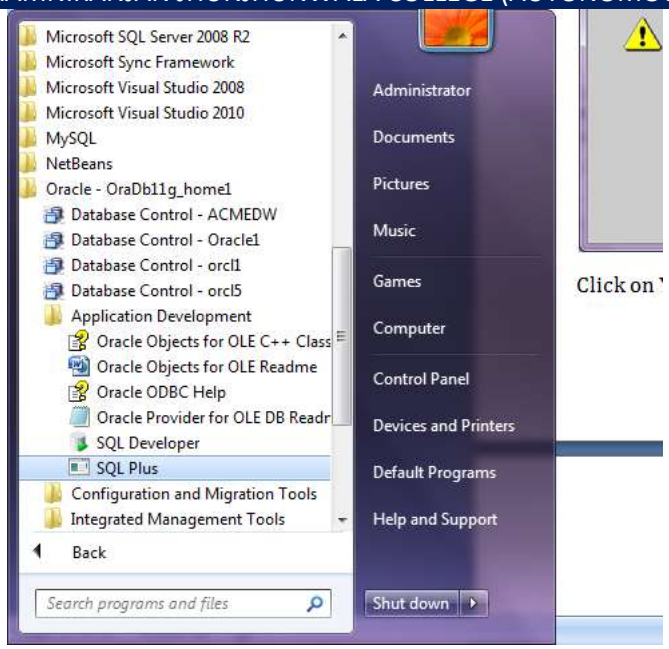


Uncheck everything as shown above write password t=at System

Click Ok



Click on Yes and then Exit



Practical No.1

Horizontal fragmentation of database.

Question: Create a global conceptual schema Emp (Eno, Ename, Address, Email, Salary) and insert 10 records. Divide Emp into horizontal fragments using the condition that Emp contains tuples with salary < 5000 and Emp with 5000 < salary < 20000 on two different nodes. Fire the following queries:

a) Find the salary of all employees.

b) Find the Email of all employees where salary=4000.

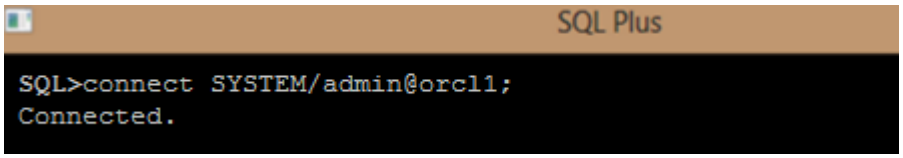
c) Find the employee name and Email where employee number is known.

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

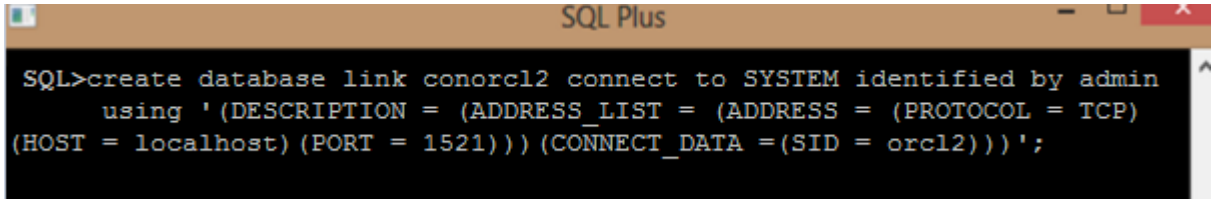
Similarly create database orcl2

Open sqlplus command prompt and login as SYSTEM (USERNAME) and admin (PASSWORD)

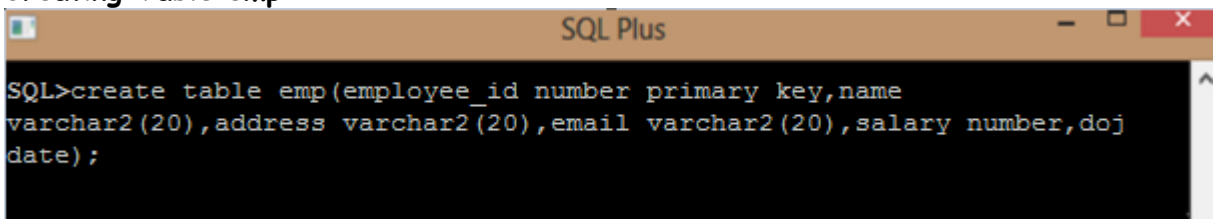
Open ORCL1 Database:



Creating Link In ORCL1:



Creating Table emp:



Inserting Values Into Table emp:

SQL Plus

```
SQL>insert into emp values(1001,'swapnil','Airoli','sj@gmail.com',2000,
'2-jan-2010');
SQL>
SQL>insert into emp values(1002,'vijay','kalwa','vm@gmail.com',2100,'3-
jan-2010');
SQL>
SQL>insert into emp values(1003,'dipesh','mulund','dk@gmail.com',2456,'7-
jan-2010');
SQL>
SQL>insert into emp values(1004,'sunita','mulund','sy@gmail.com',4000,'8-
jan-2010');
SQL>
SQL>insert into emp values(1005,'sibi','dombivli','st@gmail.com',3000,'8-
feb-2010');
```

Creating view hrz_view:

```
SQL>Create view hrz_view as select * from emp UNION select * from
emp@conorcl2;
```

Output Queries:

1) Find the salary of all employees.

```
SQL> select name,salary from emp union all select e1.name,e1.salary from emp@conorcl2 e1;

NAME                   SALARY
-------------------- ----------
siddh                    6000
chandu                   8199
raja                     9456
anup                    10000
sanket                  13000
siddh                    6000
chandu                   8199
raja                     9456
anup                    10000
sanket                  13000

10 rows selected.
```

2) Find the Email of all employees where salary=4000.

```
SQL>select name,salary,email from emp where salary >4000 union select
e1.name,e1.salary,e1.email from emp@conorcl1 e1 where e1.salary>4000;

NAME         SALARY   EMAIL
-------------------- ----------
anup         10000    ak@gmail.com
chandu       8199     cc@gmail.com
raja         9456     rk@gmail.com
sanket      13000     sk@gmail.com
siddh        6000     sm@gmail.com
sunita      10000     sy@gmail.com
6 rows selected.
```

3) Find the employee name and Email where employee number is known

```
SQL>select name,email from emp where employee_id=1005 union select
e1.name ,e1.email from emp@conorcl1 e1 where e1.employee_id=1005

NAME  email
----  --------
sibi  st@gmail.com
```

```
SQL>select name,address from emp where employee_id=1005 union select
e1.name ,e1.address from emp@conorcl1 e1 where e1.employee_id=1005

NAME  ADDRESS
----  --------
sibi  Dombivli
```

Practical No.2

Aim: Vertical fragmentation of database.

Question: Create a global conceptual schema Emp (Employee_id, first_name,last_name,email,address1,address2,address3,doj,Salary,designation) and insert 10 records. Divide Emp into vertical fragments Emp(Employee_id,first_name,last_name,email) and emp(Employee_id ,address1,address2,address3),Emp(Employee_id,doj,Salary ,designation) on two different nodes.

Fire the following queries:

a) Find the salary of an Employee where employee number is known.

b) Find the Email where the employee name is known.

c) Find the employee name and Email where employee number is known.

d) Find the employee name whose salary is > 200.

Solution : create databases orcl,orcl1,orcl2 as shown above

Open ORCL Database:

```
SQL> connect SYSTEM/admin@orcl;
```

Creating link in ORCL:

```
SQL>create database link conorcl connect to SYSTEM identified by admin
using '(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST =
localhost)(PORT = 1521)))(CONNECT_DATA =(SID = orcl )))';
```

Creating Table emp:

```
SQL> create table emp(employee_id number primary key,first_name
varchar2(20),last_name varchar2(20),email varchar2(20));
```

Inserting Values Into Table emp:

```
SQL>insert into emp values(1001,'swapnil','jadhav','sj78@gmail.com');
SQL>insert into emp values(1002,'vijay','mhatre','112.vijay@gmail.com');
SQL>insert into emp values(1003,'dipesh','kamble','daku36@gmail.com');
SQL>insert into emp values(1004,'sunita','yadav','sy@gmail.com');
SQL>insert into emp values(1005,'hunusukh','wangadu','hw@gmail.com');
```

Open ORCL1 Database:

```
SQL>connect SYSTEM/admin@orcl;
Connected
```

Creating Link in ORCL1:

```
SQL>create database link conorcl1 connect to SYSTEM identified by admin
using '(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST =
localhost)(PORT = 1521)))(CONNECT_DATA =(SID = orcl1)))';
```

Creating Table emp:

```
SQL> create table emp(employee_id number primary key,address1 varchar2(20),
address2 varchar2(20), address3 varchar2(20));
```

Inserting Values Into Table emp:

```
SQL>insert into emp values(1001,'near','khadi','airoli gaon');
SQL>insert into emp values(1002,'mhatre','nivas','kalwa country');
SQL>insert into emp values(1003,'6f','nilsagar daku colony','mulund');
SQL>insert into emp values(1004,'33f','samshan','mulund');
SQL>insert into emp values(1005,'4a','ladakh','china');
```

Open ORCL2 Database:

Creating Link In ORCL2:

```
SQL>create database link conorcl2 connect to SYSTEM identified by admin
using '(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST =
localhost)(PORT = 1521)))(CONNECT_DATA =(SID = orcl2)))';
```

Creating Table emp:

```
SQL> create table emp(employee_id number primary key,doj date,salary
number,designation varchar2(20));

Table created
```

Inserting Values Into Table emp:

```
SQL>insert into emp values(1001,'17-dec-2009',200,'Developer');

SQL>insert into emp values(1002,'10-jan-2010',250,'SR developer');

SQL>insert into emp values(1003,'10-jan-2010',250,'Entertainer');

SQL>insert into emp values(1004,'1-jan-2010',251,'manager');

SQL>insert into emp values(1005,'7-jan-2010',500,'sr manager');
```

Creating view vrt_view:

```
SQL>create view vrt_view as select
e1.employee_id,e1.first_name,e1.last_name,e1.email,e2.address1,e2.address2
e2.address3,e3.doj,e3.salary,e3.designation from emp e1,emp@conorcl1
e2,emp@conorcl2 e3 where e1.employee_id=e2.employee_id and

e2.employee_id=e3.employee_id;
```

Output Queries:

1) Find the salary of an Employee where employee number is known.

```
SQL>select e1.employee_id,e1.first_name,e2.salary from emp e1,emp@conorcl2
e2 where e1.employee_id=e2.employee_id and e1.employee_id=1001;

EMPLOYEE_ID FIRST_NAME        SALARY

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

1001         swapnil          200
```

2) Find the Email where the employee name is known.

```
SQL>select e1.employee_id,e1.first_name,e1.email from emp e1 where
e1.first_name='vijay'

EMPLOYEE_ID FIRST_NAME     EMAIL

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

1002         vijay        vijay@gmail.com
```

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

```
SQL>select e1.employee_id,e1.first_name,e1.email from emp e1 where
e1.employee_id=1001;

EMPLOYEE_ID FIRST_NAME     EMAIL

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

1001         swapnil       sj78@gmail.com
```

4) Find the employee name whose salary is > 200.

```
SQL>select e1.employee_id,e1.first_name,e2.salary from emp e1,emp@conorcl2
e2 where e1.employee_id=e2.employee_id

and e2.salary>=200;

EMPLOYEE_ID FIRST_NAME SALARY

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

1001        swapnil     200

1002        vijay       250

1003        dipesh      250

1004        sunita      251

1005        hunusuk     500
```

Practical No.3

Creating Replica of database.

Question: Create global conceptual schema Emp (Eno ,Ename, Address, Email, Salary) and insert 10 records. Store the replication of Emp into two different nodes and

Fire The Following Queries.

a) Find the salary of all employees.

b) Find the email of all employees where salary = 15000.

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

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

Open ORCL Database:

```
SQL> connect SYSTEM/admin@orcl;
```

Creating link in ORCL:

```
SQL>create database link conorcl connect to SYSTEM identified by admin
using '(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST =
localhost)(PORT = 1521)))(CONNECT_DATA =(SID = orcl )))';
```

Creating Table emp:

create table emp (eno number, ename varchar2(20), address varchar(20), email varchar2(20), salary number);

Open ORCL2 Database:

Open ORCL2 Database:

Creating Link In ORCL2:

```
SQL>create database link conorcl2 connect to SYSTEM identified by admin
using '(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST =
localhost)(PORT = 1521)))(CONNECT_DATA =(SID = orcl2)))';
```

Creating Table emp:

```
SQL>create table emp (eno number, ename varchar2(20), address varchar(20),
email varchar2(20), salary number);

Table created.
```

Creating Trigger emptrigger:

```
SQL>create or replace trigger emptrigger
  2 after insert or update of ename,address,email,salary or delete on emp
  3 for each row
  4 begin
  5 if inserting then
  6 INSERT INTO emp@conorcl1
  7 VALUES(:new.eno,:new.ename,:new.address,:new.email,:new.salary);
  8 INSERT INTO emp@conorcl2
  9 VALUES(:new.eno,:new.ename,:new.address,:new.email,:new.salary);
 10 end if;
 11 if updating then
 12 update emp@conorcl1 set ename=:new.ename where eno=:new.eno;
 13 update emp@conorcl2 set ename=:new.ename where eno=:new.eno;
 14 update emp@conorcl1 set address=:new.address where eno=:new.eno;
 15 update emp@conorcl2 set address=:new.address where eno=:new.eno;
 16 update emp@conorcl1 set email=:new.email where eno=:new.eno;
 17 update emp@conorcl2 set email=:new.email where eno=:new.eno;
 18 update emp@conorcl1 set salary=:new.salary where eno=:new.eno;
 19 update emp@conorcl2 set salary=:new.salary where eno=:new.eno;
 20 end if;
 21 if deleting then
 22 delete from emp@conorcl1 where eno=:new.eno;
 23 delete from emp@conorcl2 where eno=:new.eno;
 24 end if;
 25 end;
 26 /
```

Output Queries:

A) Inserting values into employee table:

```
SQL>insert into emp
values(1002,'vijay','mhatre','112.vijay@gmail.com',3000);

1 row created.

SQL> select * from emp

E_ID       ENAME      ADDRESS     EMAIL          SALARY

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

101         Vijay       Kalwa vijay@gmail.com 10000
```

Connect to orcl1 database:

```
SQL> connect SYSTEM/admin@orcl;
```

```
SQL> select * from emp

E_ID       ENAME      ADDRESS     EMAIL          SALARY
---------- --------  ----------  ----------    ----------
101         Vijay       Kalwa vijay@gmail.com 10000
```

Connect to orcl2 database:

```
SQL Plus
SQL> connect SYSTEM/admin@orcl2;
```

```
SQL> select * from emp

E_ID       ENAME      ADDRESS     EMAIL          SALARY
---------- --------  ----------  ----------    ----------
101         Vijay       Kalwa vijay@gmail.com 10000
```

B) Updating the values into employee table:

```
SQL>update emp set salary = 20000 where e_id=101 ;

1 row updated

SQL> select * from emp;

E_ID   ENAME ADDRESS    EMAIL         SALARY
-----  ----- -------- ----------      ---------
101    Vijay Kalwa     vijay@gmail.com 20000
```

Connect to orcl1 database:

```
SQL> connect SYSTEM/admin@orcl;
```

```
SQL> select * from emp

E_ID       ENAME      ADDRESS     EMAIL          SALARY

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

101         Vijay       Kalwa vijay@gmail.com 10000
```

Connect to orcl2 database:


```
SQL> select * from emp

E_ID       ENAME      ADDRESS     EMAIL          SALARY

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

101         Vijay       Kalwa vijay@gmail.com 10000
```

C) Deleting the values into employee table:

```
SQL>delete from emp where eno=101;

Output:

1 row deleted.

SQL> select * from emp ;

no rows selected
```

Connect to orcl1 database:

```
SQL> connect SYSTEM/admin@orcl;
```

Fire the query:

```
SQL> select * from emp ;

no rows selected
```

Connect to orcl2 database:

Fire the query:

```
SQL> select * from emp ;

no rows selected
```

Practical No. 4

Implement ORDBMS Application.

Create or replace type AddrType as object(pin integer, street varchar2(50), city varchar2(50), state varchar2(50), rno integer);

```
SQL> Create or replace type AddrType as object(pin integer, street
varchar2(50),city varchar2(50), state varchar2(50),  rno integer);

Type created.
```

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

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

Type created.
```

create or replace type BranchTableType as table of BranchType;

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

Type created.
```

create or replace type AuthorType as object(name varchar2 (50), addr AddrType1);
/
create table Authors of AuthorType;

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

```
SQL> create or replace type AuthorType as object(name varchar2 (50), addr AddrTy
pe1);
  2  /

Type created.

SQL> create table Authors of AuthorType;

Table created.

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

Type created.
```

```
create or replace type PublisherType as object(name varchar2(50), addr AddrType1,branches BranchTableType);
/
create table Publishers of PublisherType NESTED TABLE branches STORE as branchtable;

create table books(title varchar2(50), year date, published_by ref PublisherType,authorsAuthorListType);
```

```
SQL> create or replace type PublisherType as object(name varchar2(50), addr Addr
Type1,branches BranchTableType);
  2  /

Type created.

SQL> create table Publishers of PublisherType NESTED TABLE branches STORE as bra
nchtable;

Table created.

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

Table created.
```

```
SQL>insert into Publishers values('Raj', AddrType1(4002,'Park street',
'mumbai','maharashtra',03), BranchTableType(BranchType(AddrType1(5002,
'Pali street','mumbai','maharashtra',03),23406,69896)));
1 row created.

SQL>insert into Publishers values('Rohit',AddrType1(7007,'Lovely
street','mumbai','maharashtra',07),BranchTableType(BranchType(AddrType1
(7007,'K street','mumbai','maharashtra',1007),4543545,8676775)));
1 row created.

SQL>insert into Publishers values('Tata',AddrType1(7008,'Jewel
street','mumbai','maharashtra',27),BranchTableType(BranchType(AddrType1
(1002,'Diamondstreet','nasik', 'maharashtra',1007),456767,7675757)));
1 row created.

SQL>insert into Publishers values('Mcgrew', AddrType1(7002,'South
street','pune','maharashtra',1007),BranchTableType(BranchType(AddrType1
(1002,'Southstreet','pune', 'maharashtra',1007),4543545,8676775)));
1 row created.

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

```
SQL> insert into Authors values('Sangoi', AddrType1(7000,'Dalal street', 'mumbai
','maharashtra',1007));

1 row created.

SQL> insert into Authors values('Schiller',AddrType1(7008,'Pali street', 'nasik'
,'maharashtra',1008));

1 row created.

SQL> insert into Authors values('Jerry',AddrType1(7003,'Tagore street', 'mumbai'
,'maharashtra',1003));

1 row created.

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

1 row created.

SQL> insert into Authors values ('A.K.Mehta', AddrType1 (7006,'Nehru street','mu
mbai','maharashtra',1005));

1 row created.

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

1 row created.

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

1 row created.
```

select * from Authors;

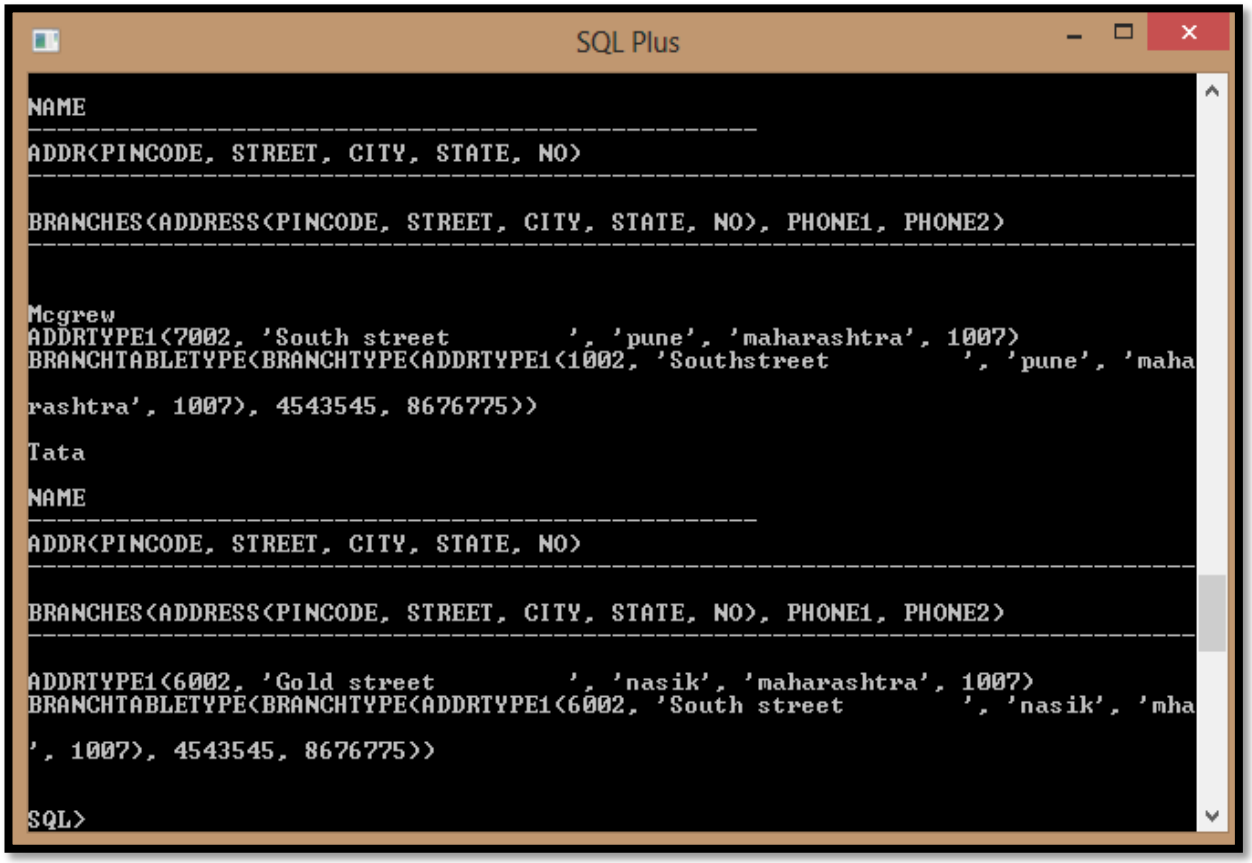
```
SQL> select * from Authors;

NAME
--------------------------------------------------
ADDR(PINCODE, STREET, CITY, STATE, NO)
--------------------------------------------------------------------------------

Sangoi
ADDRTYPE1(7000, 'Dalal street        ', 'mumbai', 'maharashtra', 1007)

Schiller
ADDRTYPE1(7008, 'Pali street         ', 'nasik', 'maharashtra', 1008)

Jerry
ADDRTYPE1(7003, 'Tagore street       ', 'mumbai', 'maharashtra', 1003)


NAME
--------------------------------------------------
ADDR(PINCODE, STREET, CITY, STATE, NO)
--------------------------------------------------------------------------------

Sangoi
ADDRTYPE1(7008, 'Dalal street        ', 'mumbai', 'maharashtra', 1007)

A.K.Mehta
ADDRTYPE1(7006, 'Nehru street        ', 'mumbai', 'maharashtra', 1005)

Ramakrishnan
ADDRTYPE1(8002, 'Thakur street       ', 'pune', 'maharashtra', 13)


NAME
--------------------------------------------------
ADDR(PINCODE, STREET, CITY, STATE, NO)
--------------------------------------------------------------------------------

Richard
ADDRTYPE1(7002, 'Flower street       ', 'pune', 'maharashtra', 3)


7 rows selected.
```

```
SQL> insert into Publishers values('Raj', AddrType1(4002,'Park street', 'mumbai'
,'maharashtra',03), BranchTableType(BranchType(AddrType1(5002,'Pali street','mum
bai','maharashtra',03),23406,69896)));

1 row created.

SQL> insert into Publishers values('Rohit',AddrType1(7007,'Lovely street','mumba
i','maharashtra',07), BranchTableType(BranchType(AddrType1(7007,'K street','numb
ai','maharashtra',1007),4543545,8676775)));

1 row created.

SQL> insert into Publishers values('Tata',AddrType1(7008,'Jewel street','mumbai'
,'maharashtra',27),BranchTableType(BranchType(AddrType1(1002,'Diamondstreet','na
sik', 'maharashtra',1007),456767,7675757)));

1 row created.

SQL> insert into Publishers values('Mcgrew', AddrType1(7002,'South street','pune
','maharashtra',1007), BranchTableType(BranchType(AddrType1(1002,'Southstreet','
pune', 'maharashtra',1007),4543545,8676775)));

1 row created.
```

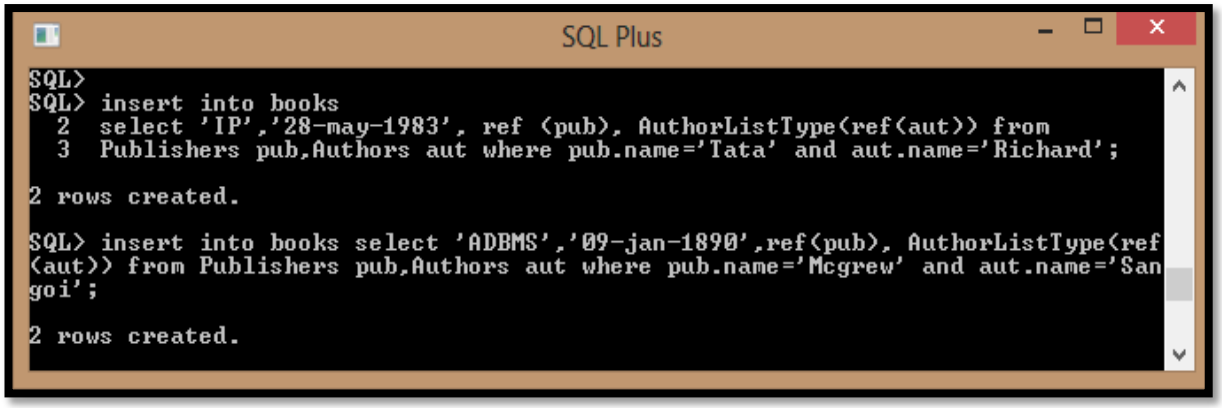
select * from Publishers;

SQL Plus

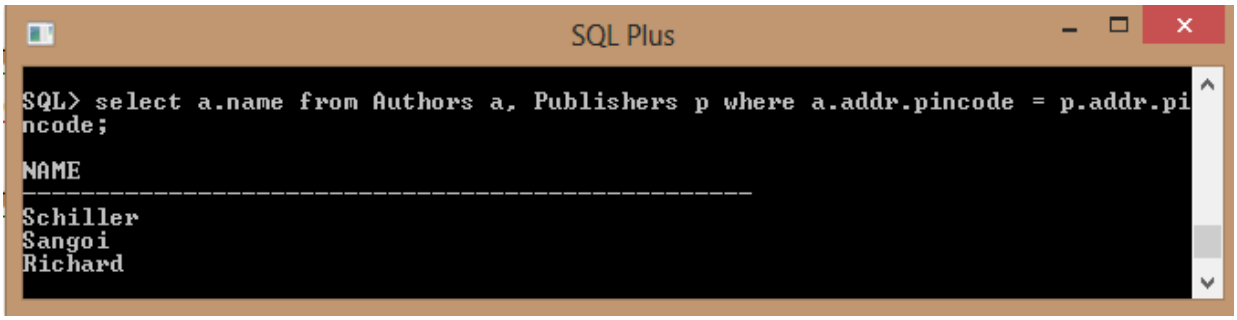
```
SQL> select * from Publishers;

NAME
--------------------------------------------------
ADDR(PINCODE, STREET, CITY, STATE, NO)
--------------------------------------------------------------------------------

BRANCHES(ADDRESS(PINCODE, STREET, CITY, STATE, NO), PHONE1, PHONE2)
--------------------------------------------------------------------------------

Raj
ADDRTYPE1(4002, 'Park street          ', 'mumbai', 'maharashtra', 3)
BRANCHTABLETYPE(BRANCHTYPE(ADDRTYPE1(5002, 'Pali street          ', 'mumbai', 'ma

harashtra', 3), 23406, 69896))

Rohit
ADDRTYPE1(7007, 'Lovely street        ', 'mumbai', 'maharashtra', 7)

NAME
--------------------------------------------------
ADDR(PINCODE, STREET, CITY, STATE, NO)
--------------------------------------------------------------------------------

BRANCHES(ADDRESS(PINCODE, STREET, CITY, STATE, NO), PHONE1, PHONE2)
--------------------------------------------------------------------------------

BRANCHTABLETYPE(BRANCHTYPE(ADDRTYPE1(7007, 'K street             ', 'mumbai', 'ma

harashtra', 1007), 4543545, 8676775))

Tata
ADDRTYPE1(7008, 'Jewel street         ', 'mumbai', 'maharashtra', 27)
BRANCHTABLETYPE(BRANCHTYPE(ADDRTYPE1(1002, 'Diamondstreet        ', 'nasik', 'mah

arashtra', 1007), 456767, 7675757))
```




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

insert into books select 'ADBMS','09-jan-1890',ref(pub), AuthorListType(ref(aut)) from Publishers
pub,Authorsaut where pub.name='Mcgrew' and aut.name='Sangoi';
```

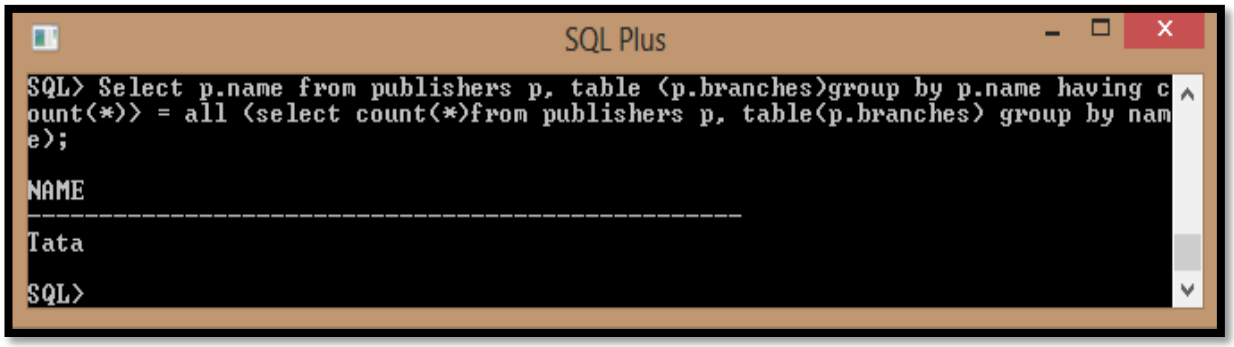



```
insert into books
select 'c prog','25-may-1983', ref (pub),AuthorListType(ref(aut)) from
Publishers pub,Authorsaut where pub.name='Raj' and aut.name='Ramkrishnan.';

select a.name from Authors a, Publishers p where a.addr.pincode = p.addr.pincode;
```



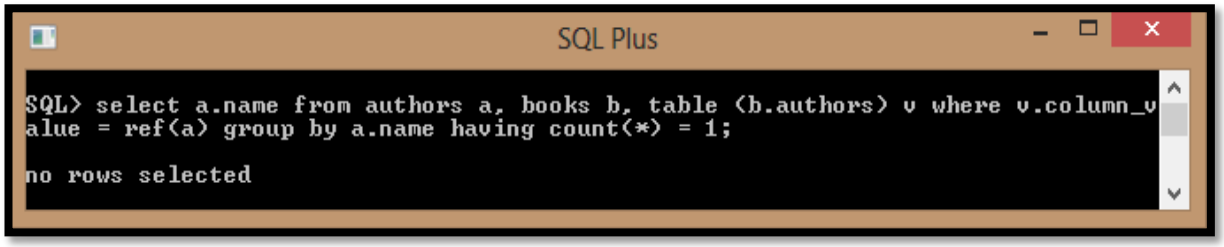
List the name of the publisher that has the most branches:

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);



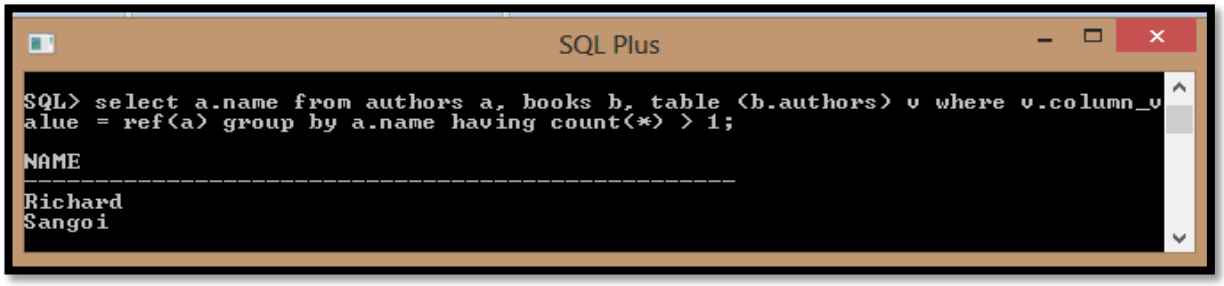
List all the authors who have published more than one book & Name of authors who have published books with atleast two different publishers

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;



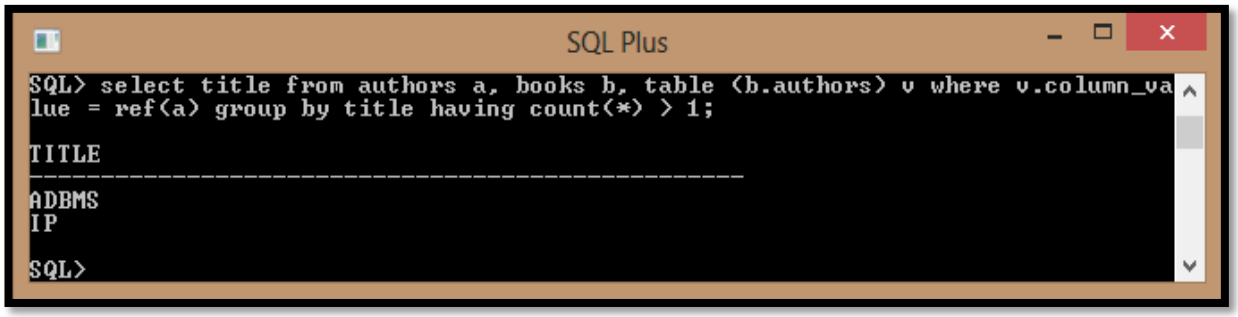
List all the authors who have published more than one book & Name of authors who have published books with atleast two different publishers:

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;



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 isunique in a list of authors has not been specified):

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



Practical No. 5

Aim: Implement XML Application

A]Create an XML Application using XML as database and ASP.

B] Create a table employee having dept_id as number datatype and employee_spec as XML datatype(XM_Type). The employee_spec is a schema with attributes emp_id, name, email, acc_no, managerEmail, dataOfJoning. Insert 10 tuples into employee table.

Fire the following queries on XML database.

a) Retrieve the names of employee.
b) Retrieve the acc_no of employees.
c) Retrieve the names, acc_no, email of employees.
d) Update the 3rd record from the table and display the name of an employee.
e) Delete 4th record from the table.

A] Create an XML Application using XML as database and ASP.

VerifyPerson.asp:

```
<%@ Language=JScript%>
<%Server.ScriptTimeout=21478836%>
<%Response.Buffer=false%>

// 1) To check whether the user has already pressed the submit button

<%
var submit = Request.Form("submit").Count;
if( submit > 0 ){


// 2) Create the needed to variables to store data from the form
var name = Request.Form("Name");
var age =  Request.Form("Age");
var gender = Request.Form("Gender");
varpcode = Request.Form("PostalCode");
var city = Request.Form("City");

// 3) First check whether the user has entered anything or not !

var error = "";
if ( name == "" )
error = "Name ";
if ( age == "" )
```

```
error += "Age ";




if ( pcode == "")
error += "PostalCode ";
if ( city == "")
error += "City ";

// 4) We have found that the user didnt entered anything

if(error!=""){
Response.Write("<center>");
Response.Write("<font fac=verdana size=2>");
Response.Write("Please enter the following data:<br>");
Response.Write("<b>");
Response.Write(error);
Response.Write("</b>");
Response.Write("</font>");
Response.Write("</center>");
}
```

```
//Response.Write(name+" has been added to your Friend list.");
// 5) Load the xmlDoc and create the required elements/nodes
```

```
varxmlDoc=Server.CreateObject("MICROSOFT.FreeThreadedXMLDOM");
xmlDoc.async="false";
xmlDoc.load(Server.MapPath("Person.xml"));
varnodeList = xmlDoc.getElementsByTagName("PersonList");
if(nodeList.length> 0){
varparentNode = nodeList(0) ;
varpersonNode = xmlDoc.createElement("Person");
varnameNode = xmlDoc.createElement("Name");
varageNode = xmlDoc.createElement("Age");
vargenderNode = xmlDoc.createElement("Gender");
varpcodeNode = xmlDoc.createElement("PostalCode");
varcityNode = xmlDoc.createElement("City");
nameNode.text = name;
ageNode.text = age;
genderNode.text= gender;
pcodeNode.text = pcode;
cityNode.text = city;
```

```
// 6) Attach the nodes to the parent node (Person)
```

```
parentNode.appendChild(personNode);
personNode.appendChild(nameNode);
```

```
personNode.appendChild(ageNode);
personNode.appendChild(genderNode);
personNode.appendChild(pcodeNode);
personNode.appendChild(cityNode);
// 7) Now save the nodes to the file
xmlDoc.save(Server.MapPath("Person.xml"));
}
}
}
%>
<html><body>
```

```
<TD width="69"><FONT face=Verdana size=2><STRONG style="BACKGROUND-COLOR:
silver">Age</STRONG></FONT></TD>
<TD width="112"><FONT face=Verdana size=2><STRONG style="BACKGROUND-COLOR:
silver">Gender</STRONG></FONT></TD>
<TD width="115"><FONT face=Verdana size=2><STRONG style="BACKGROUND-COLOR: silver">Postal
Code</STRONG></FONT></TD>
<TD width="115"><FONT face=Verdana size=2><STRONG style="BACKGROUND-COLOR:
silver">City</STRONG></FONT></TD>
</TR>
<%
// This part is used to display the data in a table via XSL
varobjXMLDoc = Server.CreateObject("MICROSOFT.FreeThreadedXMLDOM");
objXMLDoc.async = false;
objXMLDoc.load(Server.MapPath("person.xml"));
varxsl=Server.CreateObject("MICROSOFT.FreeThreadedXMLDOM");
xsl.async = false;
xsl.load(Server.MapPath("person.xsl"));
varxmlQuery="//Person";
vardocHeadlines=objXMLDoc.documentElement.selectNodes(xmlQuery);
varnumNodes;
numNodes=docHeadlines.length;
varnn;
for(vari=0;i<numNodes;i++){
nn = docHeadlines.nextNode();
Response.Write(nn.transformNode(xsl));
}
%>
</table>
</body>
</html>
```

Person.xml :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<PersonList>
<Person><Name>Sonu</Name><Age>25</Age><Gender>Male</Gender><PostalCode>99999</PostalCode><City>Thane</City></Person>
<Person><Name>vijay</Name><Age>23</Age><Gender>Male</Gender><PostalCode>121</PostalCode><City>mumbai</City></Person></PersonList>
```

Person.xsl :

```
<xsl:stylesheetxmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="Person">
<tr>
<td><font face="verdana" size="2"><xsl:value-of select="Name"/></font></td>
<td><font face="verdana" size="2"><xsl:value-of select="Age"/></font></td>
<td><font face="verdana" size="2"><xsl:value-of select="Gender"/></font></td>
<td><font face="verdana" size="2"><xsl:value-of select="PostalCode"/></font></td>
<td><font face="verdana" size="2"><xsl:value-of select="City"/></font></td>
</tr>
</xsl:template>
</xsl:stylesheet>
```

B] XML Database

Creating Table emp_xml15

```
SQL> create table emp_xml15(
dept_id number(4),
employee_specXMLtype);
Table created.
Inserting Values Into Table emp_xml15:
SQL> insert into emp_xml15 values(1,XMLtype(
'<emp id="1">
<name>sharmila</name>
<email>dave@yahoo.com</email>
<acc_no>23456</acc_no>
<mgr_email>rekha.shah@hotmail.com</mgr_email>
<doj>12/12/2003</doj>
</emp>'));
1 row created.
SQL> insert into emp_xml15 values(1,XMLtype(
'<emp id="2">
<name>anita</name>
<email>ani@yahoo.com</email>
<acc_no>234346</acc_no>
<mgr_email>rekha.shah@hotmail.com</mgr_email>
<doj>2/6/2003</doj>
</emp>'));
1 row created.
SQL> insert into emp_xml15 values(1,XMLtype(
'<emp id="3">
<name>ekta</name>
<email>ektabhatt@yahoo.com</email>
<acc_no>2343456</acc_no>
<mgr_email>ekta.bhatt@hotmail.com</mgr_email>
<doj>24/5/2001</doj>
</emp>'));
1 row created.
```

SQL Plus

```
SQL> insert into emp_xml15 values(1,XMLtype(
'<emp id="4">
<name>nancy</name>
<email>nancyshah@yahoo.com</email>
<acc_no>2343678</acc_no>
<mgr_email>ekta.shah@hotmail.com</mgr_email>
<doj>21/5/2002</doj>
</emp>'));
1 row created.
SQL> insert into emp_xml15 values(1,XMLtype(
'<emp id="5">
<name>falguni</name>
<email>falgunishah@yahoo.com</email>
<acc_no>2343345</acc_no>
<mgr_email>falguni.shah@hotmail.com</mgr_email>
<doj>1/8/2002</doj>
</emp>'));
1 row created.
SQL> insert into emp_xml15 values(1,XMLtype(
'<emp id="6">
<name>sweta</name>
<email>swetamehta@yahoo.com</email>
<acc_no>2343890</acc_no>
<mgr_email>sweta.mehta@hotmail.com</mgr_email>
<doj>2/1/2001</doj>
</emp>'));
1 row created.
SQL> insert into emp_xml15 values(2,XMLtype(
'<emp id="7">
<name>aarti</name>
<email>aartigupta@yahoo.com</email>
```

```
<acc_no>23433898</acc_no>
<mgr_email>falguni.shah@hotmail.com</mgr_email>
<doj>4/9/2002</doj>
</emp>'));
1 row created.
```

```
SQL> insert into emp_xml15 values(2,XMLtype(
'<emp id="8">
<name> sandy </name>
<email>sagupta@yahoo.com</email>
<acc_no>23567898</acc_no>
<mgr_email>sweta.shah@hotmail.com</mgr_email>
<doj>4/4/2004</doj>
</emp>'))
1 row created.
```

Queries:

a) Retrieve the names of employee:

```
SQL> select e.employee_spec.extract('//name/text()').getStringVal()
"EMP_NAME" from emp_xml15 e;


EMP_NAME
--------------------------------------------------------------------------------
-----
sharmila
anita
ekta
nancy
falguni
sweta
aarti
sandy
```

b) Retrieve the acc_no of employees:

```
SQL>select e.employee_spec.extract('//acc_no/text()').getStringVal()
"Acc_No" from emp_xml15 e;

Acc_No
-----------
23456
234346
2343456
2343678
2343345
```

c) Retrieve the names, acc_no, email of employees:

```
SQL> select e.employee_spec.extract('//name/text()').getStringVal()
"NAME",e.employee_spec.extract('//acc_no/text()').getStringVal()
"ACC_NO",e.employee_spec.extract('//email/text()').getStringVal()
"EMAIL" from emp_xml15 e;

NAME     ACC_NO  EMAIL
-------  --------  ----------
sharmila 23456 dave@yahoo.com
anita    234346 ani@yahoo.com
ekta    2343456 ektabhatt@yahoo.com
nancy   2343678 nancyshah@yahoo.com
falguni 2343345 falgunishah@yahoo.com
sweta   2343890 swetamehta@yahoo.com
aarti   23433898 aartigupta@yahoo.com
sandy   23567898 sagupta@yahoo.com
```

d) Update the 3rd record from the table and display the name of an employee:

```
SQL> update emp_xml15 e set employee_spec=XMLtype('<emp id="3">
<name>ekta</name>
<email>ektabhatt@yahoo.com</email>
<acc_no>2343456</acc_no>
<mgr_email>ekta.bhatt@hotmail.com</mgr_email>
<doj>24/5/2001</doj>
<update>This is the updated record</update>
</emp>')
where e.employee_spec.extract('//name/text()').getStringVal()  ='ekta';
```

```
SQL>
SQL> select e.employee_spec.extract('//name/text()').getStringVal()"NAME",
e.employee_spec.getClobVal() "EMP_SPECIFICATION" from emp_xml15 e where
e.employee_spec.extract('//name/text()').getStringVal()='ekta '
SQL>
```

Output:

NAME

-

EMP_SPECIFICATION

-

ekta

<emp id="3">
<name>ekta</name>
<email>ektabhatt@yahoo.com</email>
<acc_no>2343456</acc_no>
<mgr_email>ekta.bhatt@hotmail.com</mgr_email>
<doj>24/5/2001</doj>
<update>This is the updated record</update>
</emp>

e) Delete the 4th record from the table:

```
SQL> delete from emp_xml15 e
wheree.employee_spec.extract('//name/text()').getStringVal()
='nancy '
SQL> select e.employee_spec.extract('//name/text()').getStringVal() "NAME
from  emp_xml151 e;

NAME
--------
sharmila
anita
ekta
falguni
sweta
aarti
```

Practical No.6

Implement Active database using Triggers.

Question: Create table emptab (eno, ename, hrs, pno, super_no) and project (pname, pno, thrs, head_no) where thrs is the total hours and is the derived attribute. Its value is the sum of all employees working on that project. eno and pno are primary keys, head_no is foreign key to emp relation.

Insert 10 tuples and write triggers to do the following.

a) Creating a trigger to insert new employee tuple and display the new total hours from project table.

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

c) Cr eating a trigger to change the project of an employee and display the new total hours from project table.

d) Creating a trigger to delete the project of an employee.

```
Create table emptabl(eno number primary key, enamevarchar(12), hrs number, pno number, super_no number)
```

```
SQL> CONNECT SYSTEM/SYSTEM@ORCL2;
Connected.
SQL> grant create database link to scott;

Grant succeeded.
```

```
create table project(pname varchar2(20), pno number primary key, thrs number, head_no number references emptabl(eno));
```

```
SQL> Create table emptabl(eno number primary key, ename varchar(12), hrs number, pno number, super_no number)
  2  ;

Table created.

SQL> create table project(pname varchar2(20), pno number primary key, thrs number, head_no number references emptabl(eno));

Table created.

SQL> insert into emptabl values(1001,'Sunil',55,10,null);
```

```
insert into emptabl values(1001,'Sunil',55,10,null);

insert into emptabl values(1002,'vijaya',155,20,1001);

insert into emptabl values(1003,'dipti',175,40,1001);

insert into emptabl values(1004,'sonu',455,30,null);

insert into emptabl values(1005,'anupam',155,10,1003);
```

```
insert into emptabl values(1006,'sunil',110,20,1002);

insert into emptabl values(1007,'chandni',55,10,1004);

insert into emptabl values(1008,'shoobi',255,30,1002);

insert into emptabl values(1009,'sid',155,10,1005);

insert into emptabl values(1010,'raj',235,40,1006);
```

```
SQL> insert into emptabl values(1001,'Sunil',55,10,null);

1 row created.

SQL> insert into emptabl values(1002,'vijaya',155,20,1001);

1 row created.

SQL> insert into emptabl values(1003,'dipti',175,40,1001);

1 row created.

SQL> insert into emptabl values(1004,'sonu',455,30,null);

1 row created.

SQL> insert into emptabl values(1005,'anupam',155,10,1003);

1 row created.

SQL> insert into emptabl values(1006,'sunil',110,20,1002);

1 row created.

SQL> insert into emptabl values(1007,'chandni',55,10,1004);

1 row created.

SQL> insert into emptabl values(1008,'shoobi',255,30,1002);

1 row created.

SQL> insert into emptabl values(1009,'sid',155,10,1005);

1 row created.

SQL> insert into emptabl values(1010,'raj',235,40,1006);

1 row created.
```

```
insert into project values('Bank Management',10,22,1001);

insert into project values('Hospital Management',20,220,1002);

insert into project values('Speech Recognation',30,220,1004);

insert into project values('Cyber Cafe',40,220,1003);
```

```
SQL> insert into project values('Bank Management',10,22,1001);

1 row created.

SQL> insert into project values('Hospital Management',20,220,1002);

1 row created.

SQL> insert into project values('Speech Recognation',30,220,1004);

1 row created.

SQL> insert into project values('Cyber Cafe',40,220,1003);

1 row created.
```

```
update project set thrs=thrs+(select sum(hrs) from emptabl where pno=10) wherepno=10;

update project set thrs=thrs+(select sum(hrs) from emptabl where pno=20) wherepno=20;

update project set thrs=thrs+(select sum(hrs) from emptabl where pno=30)wherepno=30;

update project set thrs=thrs+(select sum(hrs) from emptabl where pno=40) wherepno=40;
```

```
SQL> update project set thrs=thrs+(select sum(hrs) from emptabl where pno=10)
  2  where pno=10;

1 row updated.

SQL>
SQL> update project set thrs=thrs+(select sum(hrs) from emptabl where pno=20)
  2  where pno=20;

1 row updated.

SQL>
SQL> update project set thrs=thrs+(select sum(hrs) from emptabl where pno=30)
  2  where pno=30;

1 row updated.

SQL>
SQL> update project set thrs=thrs+(select sum(hrs) from emptabl where pno=40)
  2  where pno=40;
```

```
create or replace trigger emptrigg
after insert on emptabl
for each row
when(New.pno!=0)
begin
update project
setthrs=thrs+:new.hrs
wherepno=:new.pno;
end;
/
```

```
insert into emptabl values(1011,'sameer',21,10,1001);
```

```
SQL> create or replace trigger thrs_4
  2  after update of pno on emptabl
  3  for each row
  4  when(old.pno!=0)
  5  begin
  6  update project
  7  set thrs=thrs-:old.hrs
  8  where pno=:new.pno;
  9  end;
 10  /

Trigger created.

SQL> update emptabl
  2  set pno=10
  3  where eno=1008;

1 row updated.

SQL>
SQL> SELECT * FROM PROJECT;

PNAME                      PNO       THRS    HEAD_NO
-------------------- ---------- ---------- ----------
Bank Management            10        508       1001
Hospital Management        20        485       1002
Speech Recognation         30        675       1004
Cyber Cafe                 40        630       1003
```

```
create or replace trigger emptrigg1
after update on emptabl
for each row
when(New.pno!=0)
begin
update project
setthrs=thrs-:old.hrs+:new.hrs
wherepno=:new.pno;
end;
/
```

```
SQL> create or replace trigger emptrigg
  2  after insert on emptabl
  3  for each row
  4  when(New.pno!=0)
  5  begin
  6  update project
  7  set thrs=thrs+:new.hrs
  8  where pno=:new.pno;
  9  end;
 10  /

Trigger created.
```

```
UPDATE EMPTABl
 SET HRS=100
 WHERE ENO=1001;
```

```
SQL>
SQL> SELECT * FROM PROJECT;

PNAME                      PNO       THRS    HEAD_NO
-------------------- ---------- ---------- ----------
Bank Management            10        508       1001
Hospital Management        20        485       1002
Speech Recognation         30        675       1004
Cyber Cafe                 40        630       1003
```

```
create or replace trigger emptrigg2
after update of pno on emptabl
for each row
when(New.pno!=0)
begin
update project
setthrs=thrs+:new.hrs
wherepno=:new.pno;
update project
setthrs=thrs-:old.hrs
wherepno=:old.pno;
end;
/
```

```
SQL> create or replace trigger thrs_4
  2  after update of pno on emptabl
  3  for each row
  4  when(old.pno!=0)
  5  begin
  6  update project
  7  set thrs=thrs-:old.hrs
  8  where pno=:new.pno;
  9  end;
 10  /
```

```
updateemptabl

setpno=10

whereeno=1001;
```

```
SQL> UPDATE EMPTABl
  2  SET HRS=100
  3  WHERE ENO=1001;

1 row updated.
```

```
SQL> create or replace trigger thrs_4
  2  after update of pno on emptabl
  3  for each row
  4  when(old.pno!=0)
  5  begin
  6  update project
  7  set thrs=thrs-:old.hrs
  8  where pno=:new.pno;
  9  end;
 10  /
```

```
create or replace trigger thrs_4
after update of pno on emptabl
for each row
when(old.pno!=0)
begin
update project
setthrs=thrs-:old.hrs
wherepno=:new.pno;
end;
/
```

```
updateemptabl
setpno=10
whereeno=1008;
```

```
SQL> SELECT * FROM PROJECT;

PNAME                     PNO       THRS    HEAD_NO
------------------- ---------- ---------- ----------
Bank Management            10        508       1001
Hospital Management        20        485       1002
Speech Recognation         30        675       1004
Cyber Cafe                 40        630       1003
```

Practical No.7

Aim: Create Temporal Database.

A] Create a table tbl_shares, which stores the, name of company, number of shares, and price per share at transaction time. Insert 10 records and fire the following queries.

1) Find all the names of a company whose share price is more than Rs.100 at 11:45 A.M.

2)Find the name of company which has highest shares price at 5.00 P.M.

B] Create a table employees, which stores the, employee_id, name, department, salary at transaction time using Time DB .insert 3 records and fire the following queries.

1) Find all the details where employee_id=10;

A)

Open ORCL1 Database:

```
SQL> grant create database link to scott;

Grant succeeded.

SQL> grant create view to scott;

Grant succeeded.

SQL> conn scott/tiger@orcl1;
Connected.
```

Create table tbl_shares15

(

cname varchar2(20),

nofshares number(5),

pricepshare number(5),

transtime timestamp(6)

)

insert into tbl_shares15 values('Cap Gemini',250,25,'17-dec-94 11.55.00.000000 am');

insert into tbl_shares15 values('Tata',205,20,'05-jun-04 11.45.00.000000 am');

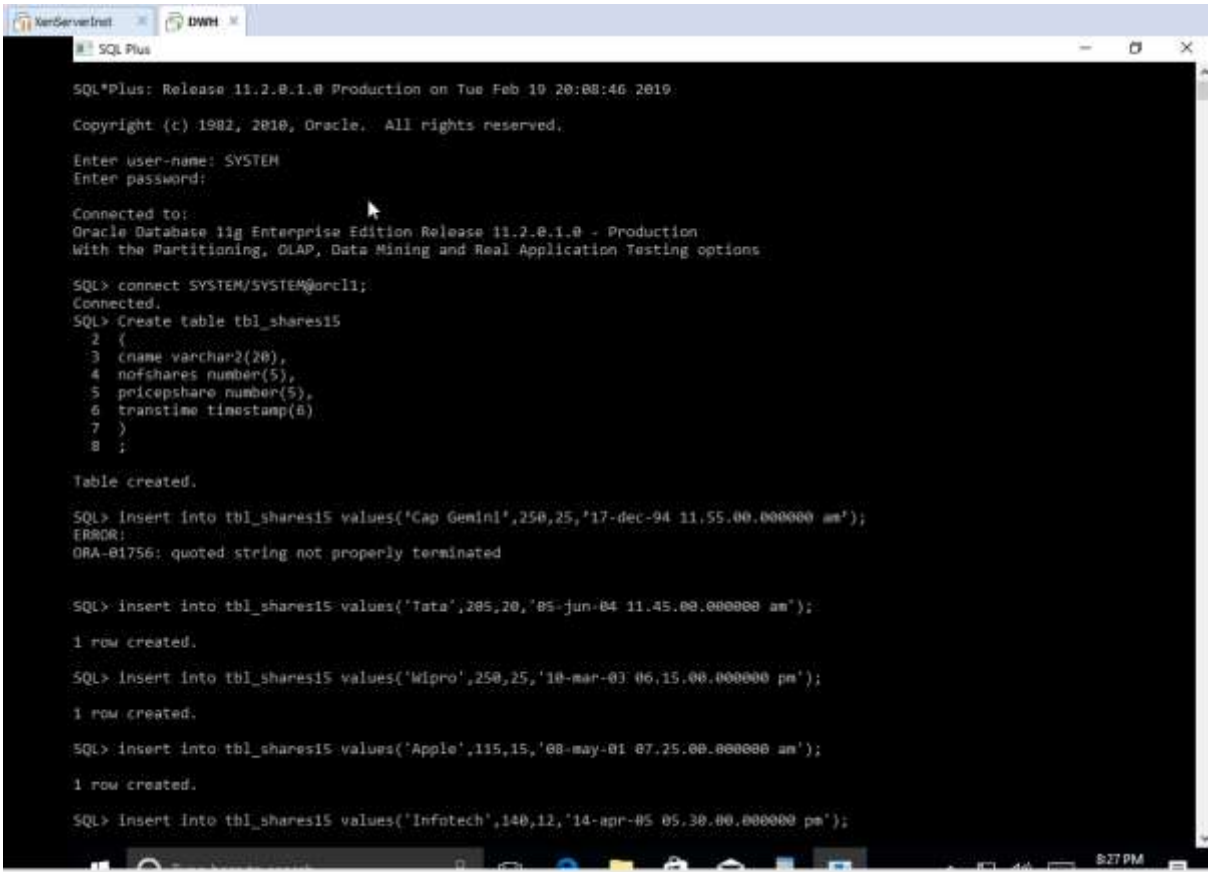
insert into tbl_shares15 values('Wipro',250,25,'10-mar-03 06.15.00.000000 pm');

insert into tbl_shares15 values('Apple',115,15,'08-may-01 07.25.00.000000 am');

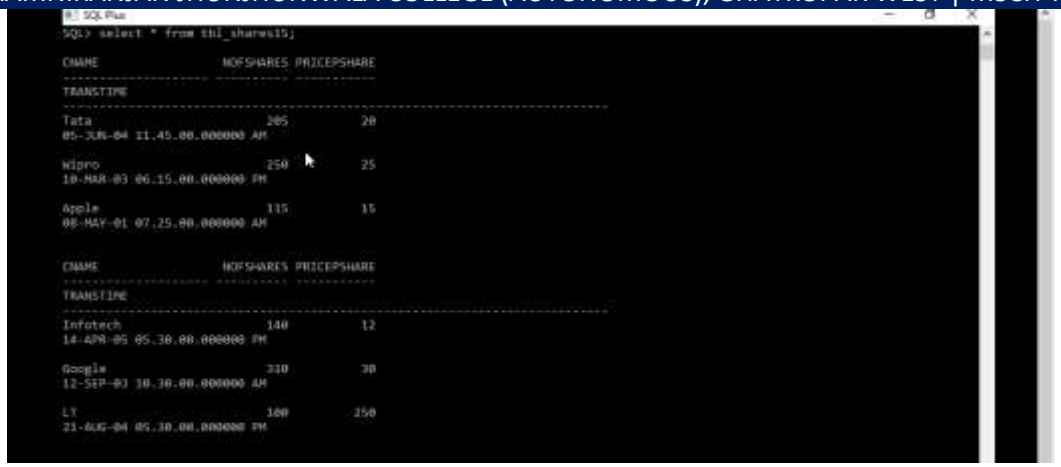
insert into tbl_shares15 values('Infotech',140,12,'14-apr-05 05.30.00.000000 pm');

insert into tbl_shares15 values('Google',310,30,'12-sep-03 10.30.00.000000 am');

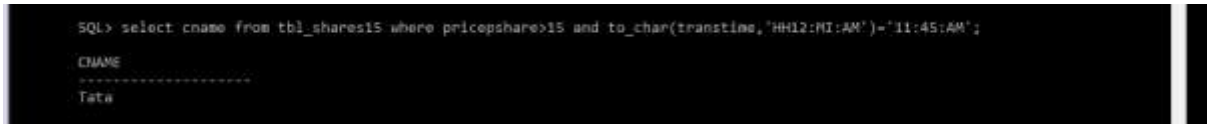
insert into tbl_shares15 values('LT',100,250,'21-aug-04 05.30.00.000000 pm')



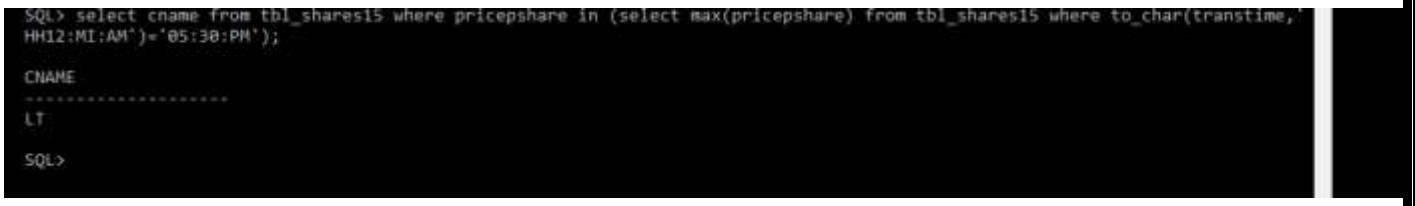
select * from tbl_shares15;



selectcname from tbl_shares15 where pricepshare>15 and to_char(transtime,'HH12:MI:AM')='11:45:AM';



selectcname from tbl_shares15 where pricepshare in (select max(pricepshare) from tbl_shares15 where to_char(transtime,'HH12:MI:AM')='05:30:PM');



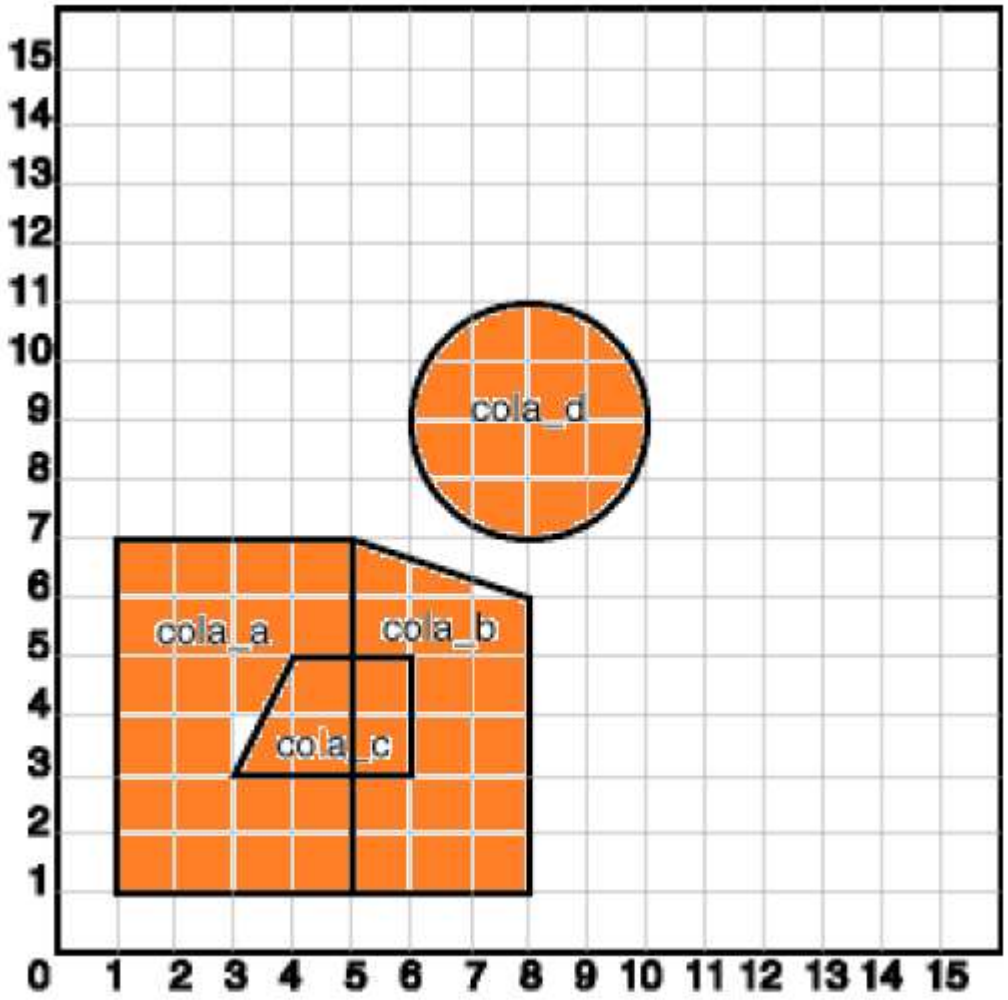
Practical No.8

Implement and retrieve records from a Spatial Database.

A] Spatial Database

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

1. Find the topological intersection of two geometries.
2. Find whether two geometric figures are equivalent to each other.
3. Find the areas of all direction locations.
4. Find the area of only one location.
5. Find the distance between two geometries.

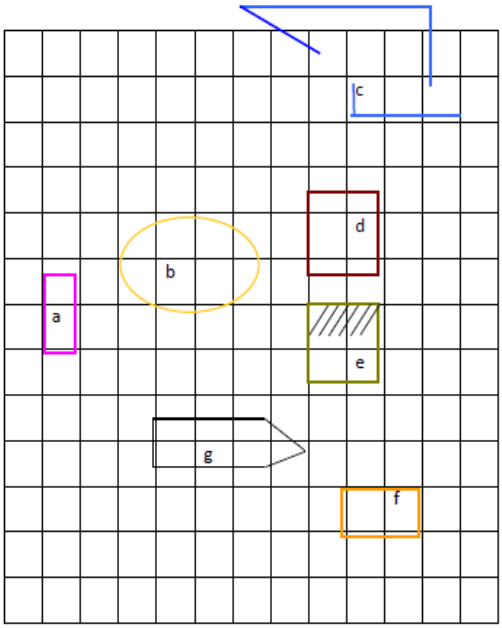


B] Spatial Database

Create a spatial database that stores the number, name and location, which consists of different areas within the university campus including the main gate, the playground, the arts and science college buildings, the lad and print facility building. Fire the following queries.

1. Display area for each object.
2. Find out the distance of the main gate from all other objects
3. Find the intersection area of lab and print facility
4. Distance between arts and science building
5. Find the spatial relationship between canteen and print facility

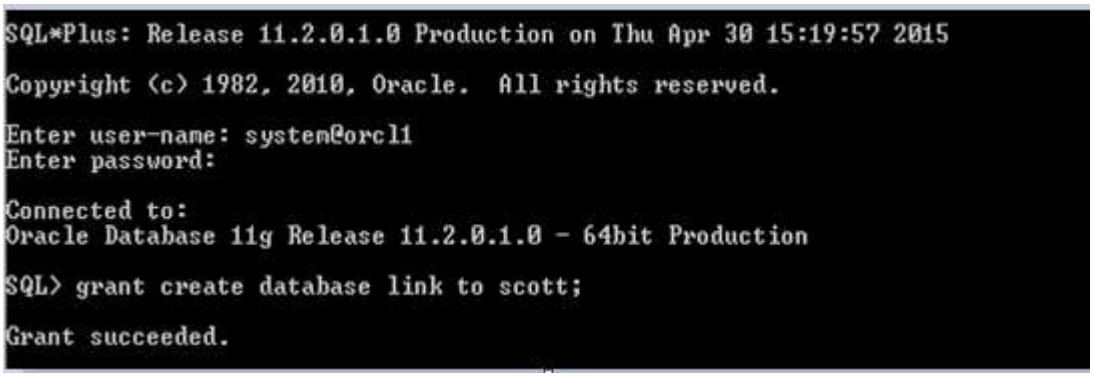
6. Find the distance between college buildings and canteen.



Symbol	Area
A	Main gate
B	Playground
C	Science building
D	Lab
E	Print facility
F	Canteen
G	Arts building

A)

```
CREATE TABLE cola_markets1 (
mkt_id NUMBER PRIMARY KEY,
name VARCHAR2(32),
shape MDSYS.SDO_GEOMETRY);

INSERT INTO cola_markets1
VALUES(1,'abc',MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1
003,3),MDSYS.SDO_ORDINATE_ARRAY(1,1, 5,7) ));

INSERT INTO cola_markets1
VALUES(2,'pqr',MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1
003,1),MDSYS.SDO_ORDINATE_ARRAY(5,1, 8,1, 8,6, 5,7, 5,1)));

INSERT INTO cola_markets1 VALUES(3,'mno',MDSYS.SDO_GEOMETRY(2003,
NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(3,3,
6,3, 6,5, 4,5, 3,3)));

INSERT INTO cola_markets1 VALUES(4,'xyz',MDSYS.SDO_GEOMETRY(2003,
NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,4),MDSYS.SDO_ORDINATE_ARRAY(8,7,
10,9, 8,11)));
```




```
INSERT INTO USER_SDO_GEOM_METADATA
VALUES ('cola_markets1','shape',
MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X', 0, 20, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y', 0, 20, 0.005)
),NULL
);
```

```
SQL> INSERT INTO USER_SDO_GEOM_METADATA
  2  VALUES ('cola_markets1','shape',
  3  MDSYS.SDO_DIM_ARRAY(
  4  MDSYS.SDO_DIM_ELEMENT('X', 0, 20, 0.005),
  5  MDSYS.SDO_DIM_ELEMENT('Y', 0, 20, 0.005)
  6  ),NULL
  7  );

1 row created.
```

CREATE INDEX cola_spatial_idx
ON cola_markets1(shape)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```
SQL> CREATE INDEX cola_spatial_idx
  2  ON cola_markets1(shape)
  3  INDEXTYPE IS MDSYS.SPATIAL_INDEX;

Index created.
```

SELECT SDO_GEOM.SDO_INTERSECTION(c_a.shape, c_c.shape, 0.005)
FROM cola_markets1 c_a, cola_markets1 c_c
WHERE c_a.name = 'abc' AND c_c.name = 'mno';

```
SQL> SELECT SDO_GEOM.SDO_INTERSECTION(c_a.shape, c_c.shape, 0.005)
  2  FROM cola_markets1 c_a, cola_markets1 c_c
  3  WHERE c_a.name = 'abc' AND c_c.name = 'mno';

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))
```

SELECT SDO_GEOM.RELATE(c_b.shape, 'equal', c_d.shape, 0.005)
FROM cola_markets1 c_b, cola_markets1 c_d
WHERE c_b.name = 'abc' AND c_d.name = 'mno';

```
SQL> SELECT SDO_GEOM.RELATE(c_b.shape, 'equal', c_d.shape, 0.005)
  2  FROM cola_markets1 c_b, cola_markets1 c_d
  3  WHERE c_b.name = 'abc' AND c_d.name = 'mno';

SDO_GEOM.RELATE(C_B.SHAPE,'EQUAL',C_D.SHAPE,0.005)
--------------------------------------------------------------------------------
FALSE
```

SELECT name, SDO_GEOM.SDO_AREA(shape, 0.005) FROM cola_markets1;

SELECT c.name, SDO_GEOM.SDO_AREA(c.shape, 0.005) FROM cola_markets1 c WHERE c.name = 'xyz';

```
SQL> SELECT name, SDO_GEOM.SDO_AREA(shape, 0.005) FROM cola_markets1;

NAME                           SDO_GEOM.SDO_AREA(SHAPE,0.005)
------------------------------ ------------------------------
abc                                                        24
pqr                                                      16.5
mno                                                         5
xyz                                                12.5663706

SQL> SELECT c.name, SDO_GEOM.SDO_AREA(c.shape, 0.005) FROM cola_markets1 c WHERE
 c.name = 'xyz';

NAME                           SDO_GEOM.SDO_AREA(C.SHAPE,0.005)
------------------------------ --------------------------------
xyz                                                12.5663706
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM cola_markets1 c_b, cola_markets1 c_d
WHERE c_b.name = 'abc' AND c_d.name = 'xyz';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM cola_markets1 c_b, cola_markets1 c_d
  3  WHERE c_b.name = 'abc' AND c_d.name = 'xyz';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                      1.60555128
```

B)

CREATE TABLE university_camp (
mkt_id NUMBER PRIMARY KEY,
name VARCHAR2(32),
shape MDSYS.SDO_GEOMETRY);

INSERT INTO university_campVALUES(
1,'a',
MDSYS.SDO_GEOMETRY(
2003,NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
MDSYS.SDO_ORDINATE_ARRAY(1,5, 2,8)
)
);

INSERT INTO university_campVALUES(
4,'d',
MDSYS.SDO_GEOMETRY(
2003,NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),

```
MDSYS.SDO_ORDINATE_ARRAY(8,6,10,9)
)
);

INSERT INTO university_campVALUES(
5,'e',
MDSYS.SDO_GEOMETRY(
2003,NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
MDSYS.SDO_ORDINATE_ARRAY(8,4,10,7)
)
);

INSERT INTO university_campVALUES(
6,'f',
MDSYS.SDO_GEOMETRY(
2003,NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
MDSYS.SDO_ORDINATE_ARRAY(9,1,11,3)
)
);


INSERT INTO university_campVALUES(
3,'c',
MDSYS.SDO_GEOMETRY(
2003,NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),
MDSYS.SDO_ORDINATE_ARRAY(8,10,11,10,11,13,6,13,8,11,8,10)
)
);


INSERT INTO university_campVALUES(
7,'g',
MDSYS.SDO_GEOMETRY(
2003, NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),
MDSYS.SDO_ORDINATE_ARRAY(4,2,7,2,8,3,7,4,4,4,4,2)
)
);
```

```
INSERT INTO university_campVALUES(
2,'b',
MDSYS.SDO_GEOMETRY(
2003, NULL,NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,4),
MDSYS.SDO_ORDINATE_ARRAY(3,7, 5,5,5,9)
)
);

INSERT INTO USER_SDO_GEOM_METADATA
VALUES ('university_camp','shape',
MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X', 0, 20, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y', 0, 20, 0.005)
),NULL
);
```

```
SQL> CREATE TABLE university_camp (
  2  mkt_id NUMBER PRIMARY KEY,
  3  name VARCHAR2(32),
  4  shape MDSYS.SDO_GEOMETRY);

Table created.

SQL> INSERT INTO university_camp VALUES(
  2  1,'a',
  3  MDSYS.SDO_GEOMETRY(
  4  2003,NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
  6  MDSYS.SDO_ORDINATE_ARRAY(1,5, 2,8)
  7  )
  8  );

1 row created.

SQL>
SQL> INSERT INTO university_camp VALUES(
  2  4,'d',
  3  MDSYS.SDO_GEOMETRY(
  4  2003,NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
  6  MDSYS.SDO_ORDINATE_ARRAY(8,6,10,9)
  7  )
  8  );

1 row created.

SQL>
SQL> INSERT INTO university_camp VALUES(
  2  5,'e',
  3  MDSYS.SDO_GEOMETRY(
  4  2003,NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
  6  MDSYS.SDO_ORDINATE_ARRAY(8,4,10,7)
  7  )
  8  );

1 row created.

SQL>
SQL> INSERT INTO university_camp VALUES(
  2  6,'f',
  3  MDSYS.SDO_GEOMETRY(
  4  2003,NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
  6  MDSYS.SDO_ORDINATE_ARRAY(9,1,11,3)
  7  )
  8  );

1 row created.
```

```
SQL> INSERT INTO university_camp VALUES(
  2  3,'c',
  3  MDSYS.SDO_GEOMETRY(
  4  2003,NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),
  6  MDSYS.SDO_ORDINATE_ARRAY(8,10,11,10,11,13,6,13,8,11,8,10)
  7  )
  8  );

1 row created.

SQL>
SQL> INSERT INTO university_camp VALUES(
  2  7,'g',
  3  MDSYS.SDO_GEOMETRY(
  4  2003, NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),
  6  MDSYS.SDO_ORDINATE_ARRAY(4,2,7,2,8,3,7,4,4,4,4,2)
  7  )
  8  );

1 row created.

SQL>
SQL> INSERT INTO university_camp VALUES(
  2  2,'b',
  3  MDSYS.SDO_GEOMETRY(
  4  2003, NULL,NULL,
  5  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,4),
  6  MDSYS.SDO_ORDINATE_ARRAY(3,7, 5,5,5,9)
  7  )
  8  );

1 row created.

SQL>
SQL> INSERT INTO USER_SDO_GEOM_METADATA
  2  VALUES ('university_camp','shape',
  3  MDSYS.SDO_DIM_ARRAY(
  4  MDSYS.SDO_DIM_ELEMENT('X', 0, 20, 0.005),
  5  MDSYS.SDO_DIM_ELEMENT('Y', 0, 20, 0.005)
  6  ),NULL
  7  );

1 row created.
```

CREATE INDEX cola_spatial_idx1
ON university_camp(shape)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```
SQL> CREATE INDEX cola_spatial_idx1
  2  ON university_camp(shape)
  3  INDEXTYPE IS MDSYS.SPATIAL_INDEX;

Index created.
```

SELECT name, SDO_GEOM.SDO_AREA(shape, 0.005) FROM university_camp;

```
SQL> SELECT name, SDO_GEOM.SDO_AREA(shape, 0.005) FROM university_camp;

NAME                           SDO_GEOM.SDO_AREA(SHAPE,0.005)
------------------------------ ------------------------------
a                                                           3
d                                                           6
e                                                           6
f                                                           4
c                                                          11
g                                                           7
b                                                    12.5663706

7 rows selected.
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'a' AND c_d.name = 'b';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'a' AND c_d.name = 'b';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                               1
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'a' AND c_d.name = 'c';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'a' AND c_d.name = 'c';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                      6.32455532
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'a' AND c_d.name = 'd';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'a' AND c_d.name = 'd';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                               6
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d

WHERE c_b.name = 'a' AND c_d.name = 'e';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'a' AND c_d.name = 'e';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                               6
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'a' AND c_d.name = 'f';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'a' AND c_d.name = 'f';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                      7.28010989
```

SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'a' AND c_d.name = 'g';

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'a' AND c_d.name = 'g';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                      2.23606798
```

SELECT SDO_GEOM.SDO_INTERSECTION(c_a.shape, c_c.shape, 0.005)
FROM university_campc_a, university_campc_c
WHERE c_a.name = 'd' AND c_c.name = 'e';

```
SQL> SELECT SDO_GEOM.SDO_INTERSECTION(c_a.shape, c_c.shape, 0.005)
  2  FROM university_camp c_a, university_camp c_c
  3  WHERE c_a.name = 'd' AND c_c.name = 'e';

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(8, 7, 8, 6, 10, 6, 10, 7, 8, 7))
```

```
SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'c' AND c_d.name = 'g';
```

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'c' AND c_d.name = 'g';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                      6.08276253
```

```
SELECT SDO_GEOM.RELATE(c_b.shape, 'anyinteract', c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'e' AND c_d.name = 'f';
```

```
SQL> SELECT SDO_GEOM.RELATE(c_b.shape, 'anyinteract', c_d.shape, 0.005) FROM uni
versity_camp c_b, university_camp c_d
  2  WHERE c_b.name = 'e' AND c_d.name = 'f';

SDO_GEOM.RELATE(C_B.SHAPE,'ANYINTERACT',C_D.SHAPE,0.005)
--------------------------------------------------------------------------------

FALSE
```

```
SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'c' AND c_d.name = 'f';
```

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'c' AND c_d.name = 'f';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                               7
```

```
SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
FROM university_campc_b, university_campc_d
WHERE c_b.name = 'g' AND c_d.name = 'f';
```

```
SQL> SELECT SDO_GEOM.SDO_DISTANCE(c_b.shape, c_d.shape, 0.005)
  2  FROM university_camp c_b, university_camp c_d
  3  WHERE c_b.name = 'g' AND c_d.name = 'f';

SDO_GEOM.SDO_DISTANCE(C_B.SHAPE,C_D.SHAPE,0.005)
------------------------------------------------
                                               1
```

Practical No.9

Implement Prolog Programming.

a) Map colorings.

b) Two factorial definitions.

c) Towers of Hanoi puzzle.

d) Tree data and relations.

e) Animal identification game.

A] Map colorings

Source Code:

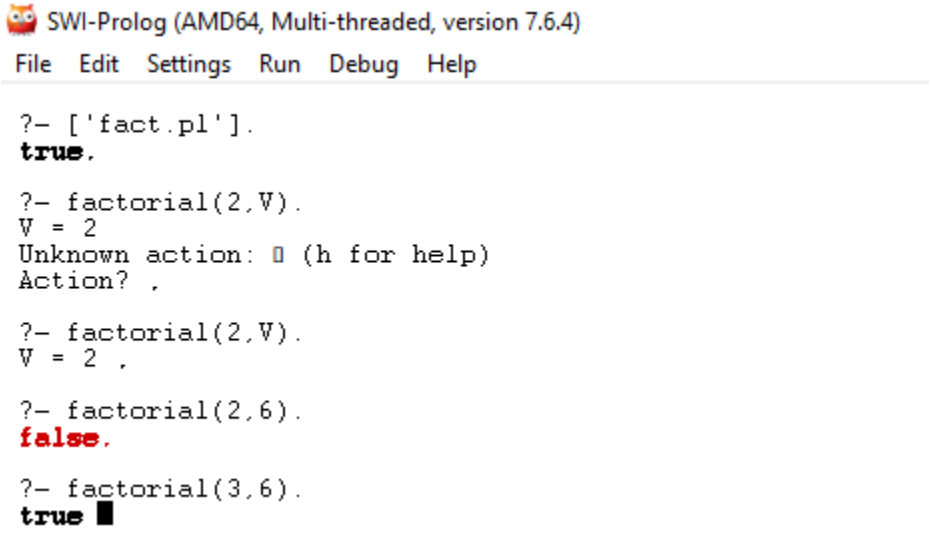
```
adjacent(1,2).          adjacent(2,1).
adjacent(1,3).          adjacent(3,1).
adjacent(1,4).          adjacent(4,1).
adjacent(1,5).          adjacent(5,1).
adjacent(2,3).          adjacent(3,2).
adjacent(2,4).          adjacent(4,2).
adjacent(3,4).          adjacent(4,3).
adjacent(4,5).          adjacent(5,4).

color(1,red,a).         color(1,red,b).
color(2,blue,a).        color(2,blue,b).
color(3,green,a).       color(3,green,b).
color(4,yellow,a).      color(4,blue,b).
color(5,blue,a).        color(5,green,b).

conflict(Coloring) :-
        adjacent(X,Y),
        color(X,Color,Coloring),
        color(Y,Color,Coloring).
conflict(R1,R2,Coloring) :-
        adjacent(R1,R2),
        color(R1,Color,Coloring),
color(R2,Color,Coloring).
```

```
File  Edit  Settings  Run  Debug  Help
?- conflict(R1,R2,b)
R1=2 R2=4
?- conflict(R1,R2,b),color(R1,C,b).
R1=2 R2=4 C=blue
?-conflict(which)
 false
?-adjacent(2,3)
true
```

B] Two factorial definitions :

Source Code:

```
factorial(0,1).
factorial(N,F) :-
N>0,
N1 is N-1,
factorial(N1,F1),
F is N * F1.
factorial(0,F,F).
factorial(N,A,F) :-
N > 0,
A1 is N*A,
N1 is N -1,
factorial(N1,A1,F).
```



C] Towers of Hanoi puzzle

Source Code:

```
move(1,X,Y,_) :-
        write('Move top disk from '),
        write(X),
        write(' to '),
        write(Y),
        nl.
move(N,X,Y,Z) :-
        N>1,
        M is N-1,
        move(M,X,Z,Y),
        move(1,X,Y,_),
move(M,Z,Y,X).
```

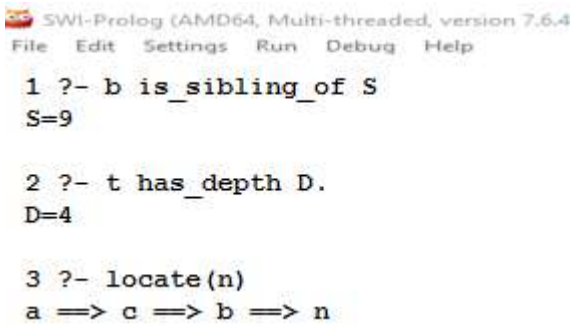
```
?- ['hanoi.pl'].
true.

?- move(3,left,right,center).
Move top disk from left to right
Move top disk from left to center
Move top disk from right to center
Move top disk from left to right
Move top disk from center to left
Move top disk from center to right
Move top disk from left to right
true
```

D] Tree data and relations

Source Code:

```
        :- op(500,xfx,'is_parent').
        ais_parent b. c is_parent g. f is_parent l. j is_parent q.
        ais_parent c. c is_parent h. f is_parent m. j is_parent r.
        ais_parent d. c is_parenti. h is_parent n. j is_parent s.
        bis_parent e. d is_parent j. iis_parent o. m is_parent t.
        bis_parent f. e is_parent k. iis_parent p.
        /* X and Y are siblings */
        :- op(500,xfx,'is_sibling_of').
        X is_sibling_ofY :- Z is_parent X, Z is_parent Y, X \== Y.
        /* X and Y are on the same level in the tree. */
        :-op(500,xfx,'is_same_level_as').
        X is_same_level_asX .
        X is_same_level_asY :- W is_parent , Z is_parent Y, W is_same_level_as Z.
        /* Depth of node in the tree. */
        :- op(500,xfx,'has_depth').
        ahas_depth 0 :- !.
        Node has_depthD :- Mother is_parent Node, Mother has_depth D1,
        D is D1 + 1.
        /* Locate node by finding a path from root down to the node. */
```

```
locate(Node) :- path(Node), write(Node), nl.
path(a). /* Can start at a. */
path(Node) :- Mother is_parent Node, /* Choose parent, */
path(Mother), /* find path and then */
write(Mother),
write(' --> ').
/* Calculate the height of a node, length of longest path to
a leaf under the node. */
height(N,H) :- setof(Z,ht(N,Z),Set), /* See section 2.8 for 'setof'. */
max(Set,0,H).
ht(Node,0) :- leaf(Node), !.
ht(Node,H) :- Node is_parent Child,
ht(Child,H1),
H is H1 +1.
leaf(Node) :- not(is_parent(Node,Child)). /* Node grounded */
max([],M,M).
max([X|R],M,A) :- (X > M -> max(R,X,A) ; max(R,M,A)).
```



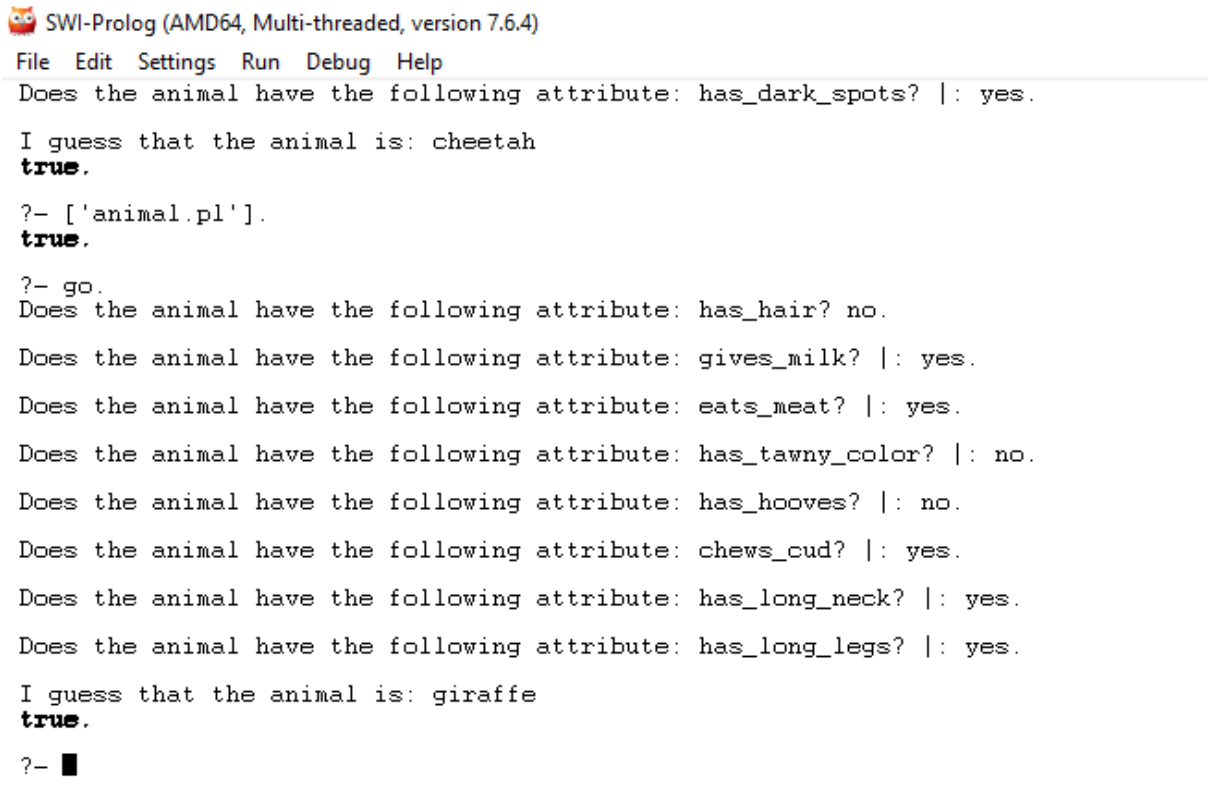
E] Animal identification game

Source Code:

```
/* start with ?- go. */
go :- hypothesize(Animal), write('I guess that the animal is: '), write(Animal), nl, undo.
/* hypotheses to be tested */
hypothesize(cheetah) :- cheetah, !.
hypothesize(tiger) :- tiger, !.
hypothesize(giraffe) :- giraffe, !.
hypothesize(zebra) :- zebra, !.
hypothesize(ostrich) :- ostrich, !.
hypothesize(penguin) :- penguin, !.
hypothesize(albatross) :- albatross, !.
hypothesize(unknown). /* no diagnosis */
/* animal identification rules */
cheetah :- mammal,
carnivore,
verify(has_tawny_color),
```

```
verify(has_dark_spots).
tiger :- mammal,
carnivore,

verify(has_tawny_color),
verify(has_black_stripes).
giraffe :- ungulate,
verify(has_long_neck),
verify(has_long_legs).
zebra :- ungulate,
verify(has_black_stripes).
ostrich :- bird,
verify(does_not_fly),
verify(has_long_neck).
penguin :- bird,
verify(does_not_fly),
verify(swims),
verify(is_black_and_white).
albatross :- bird,
verify(appears_in_story_Ancient_Mariner),
verify(flys_well).
/* classification rules */
mammal :- verify(has_hair), !.
mammal :- verify(gives_milk).
bird :- verify(has_feathers), !.
bird :- verify(flys),
verify(lays_eggs).
carnivore :- verify(eats_meat), !.
carnivore :- verify(has_pointed_teeth),
verify(has_claws),
verify(has_forward_eyes).
ungulate :- mammal,
verify(has_hooves), !.
ungulate :- mammal,
verify(chews_cud).
/* how to ask questions */
ask(Question) :-
write('Does the animal have the following attribute: '),
write(Question),
write('? '),
read(Response),
nl,
( (Response == yes ; Response == y)
->
assert(yes(Question)) ;
assert(no(Question)), fail).
:- dynamic yes/1,no/1.
/* How to verify something */
```

```
verify(S) :-
(yes(S)
->
true ;
(no(S)
->
fail ;
ask(S))).
/* undo all yes/no assertions */
undo :- retract(yes(_)),fail.
undo :- retract(no(_)),fail.
undo.
```



Practical No.10

Aim:-

1. Create XML Parser

2. Using XML DOM Traverse XML Document

Source Code:-

XML file for parsing in Java

Here is xml file Stocks.xml which contains some stocks and there price, quantity we will use this in our xml parsing example in Java.

```
<?xml version="1.0" encoding="UTF-8"?>
<stocks>
    <stock>
        <symbol>Citibank</symbol>
        <price>100</price>
        <quantity>1000</quantity>
    </stock>
    <stock>
        <symbol>Axis bank</symbol>
        <price>90</price>
        <quantity>2000</quantity>
    </stock>
</stocks>
```

Code Example of Parsing XML File in Java using DOM Parser

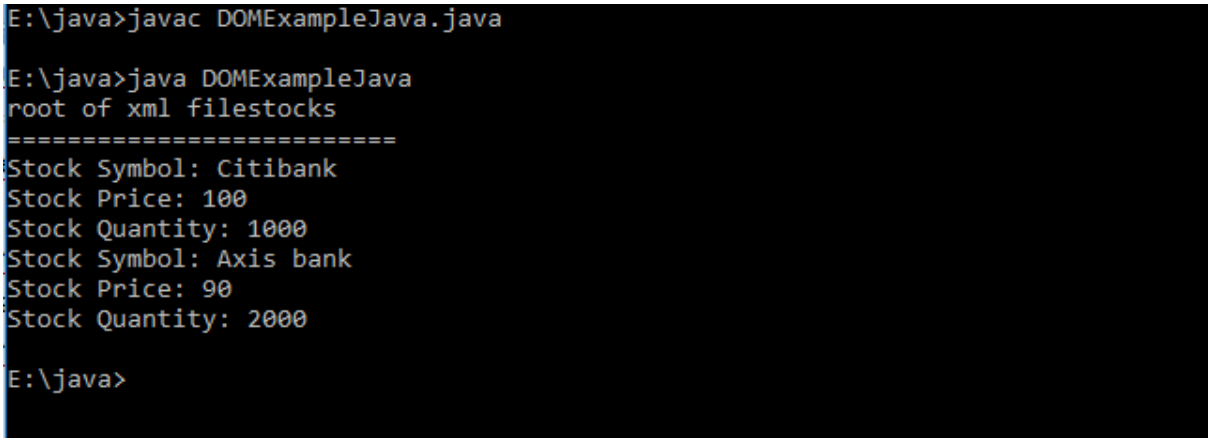
Here is a code example of parsing above xml file in Java using DOM parser:

```
import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
```

```
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMExampleJava {

public static void main(String args[]) {
try {

File stocks = new File("Stocks.xml");
DocumentBuilderFactorydbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilderdBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(stocks);
doc.getDocumentElement().normalize();

System.out.println("root of xml file" + doc.getDocumentElement().getNodeName());
NodeList nodes = doc.getElementsByTagName("stock");
System.out.println("==========================");

for (int i = 0; i<nodes.getLength(); i++) {
Node node = nodes.item(i);

if (node.getNodeType() == Node.ELEMENT_NODE) {
Element element = (Element) node;
System.out.println("Stock Symbol: " + getValue("symbol", element));
System.out.println("Stock Price: " + getValue("price", element));
System.out.println("Stock Quantity: " + getValue("quantity", element));
}
}
} catch (Exception ex) {
ex.printStackTrace();
}
}

private static String getValue(String tag, Element element) {
NodeList nodes = element.getElementsByTagName(tag).item(0).getChildNodes();
Node node = (Node) nodes.item(0);
return node.getNodeValue();
}
}
```

Output:

root of xml file stocks
==========
Stock Symbol: Citibank
Stock Price: 100
Stock Quantity: 1000
Stock Symbol: Axis bank
Stock Price: 90
Stock Quantity: 2000



Practical No.11

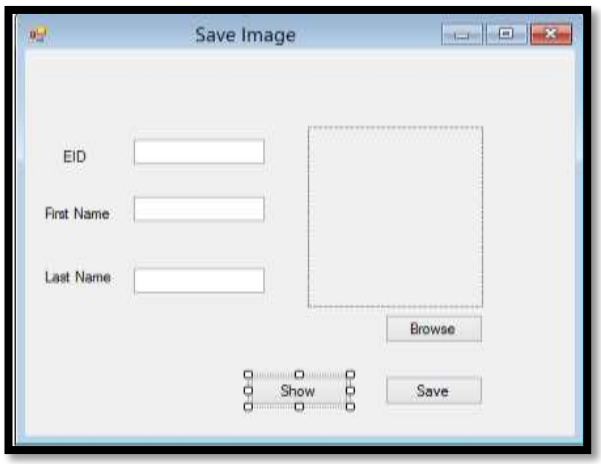
Inserting and Retrieving Multimedia Objects in Database (Image/audio/video).

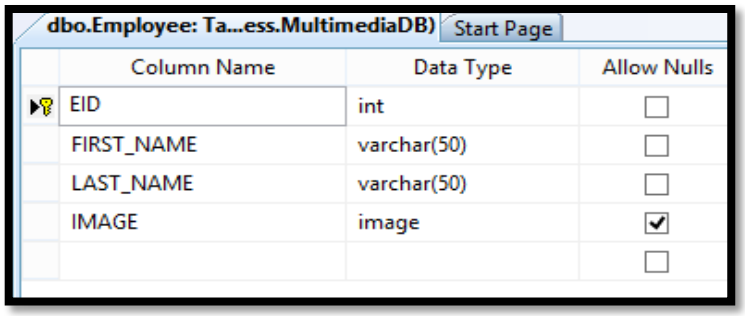
Coding:

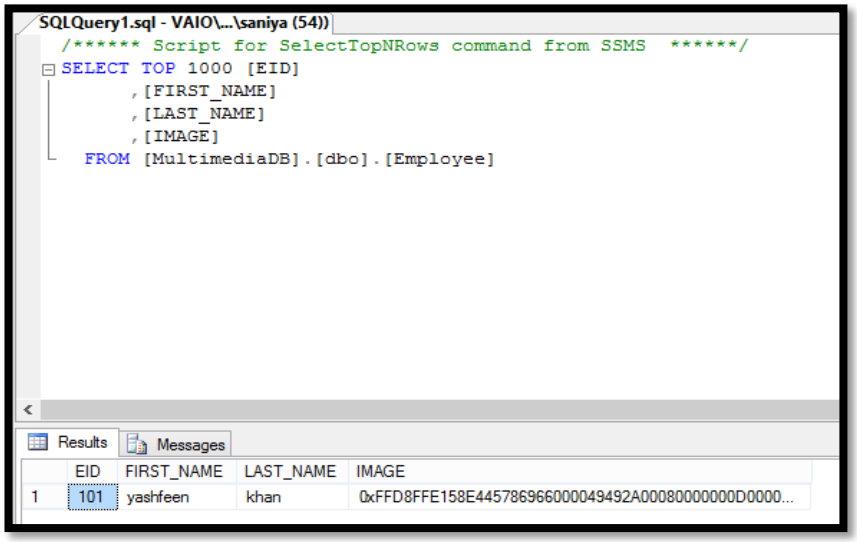
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;
namespace ImageSaveToSQLServer
{
publicpartialclassForm1 : Form
   {
SqlConnection conn = newSqlConnection(@"Data Source=VAIO\SQLEXPRESS;Initial
Catalog=MultimediaDB;Integrated Security=True");
SqlCommand command;
string imgLoc = "";
public Form1()
      {
         InitializeComponent();
      }

privatevoid Form1_Load(object sender, EventArgs e)
      {

      }

privatevoid buttonBrowse_Click(object sender, EventArgs e)
      {
try
         {
OpenFileDialog dlg = newOpenFileDialog();
            dlg.Filter = "JPG Files(*.jpg)|*.jpg|GIF Files(*.gif)|*.gif|All Files(*.*)|*.*";
            dlg.Title = "Select employee picture";
if (dlg.ShowDialog() == DialogResult.OK)
            {
               imgLoc = dlg.FileName.ToString();
               picEmp.ImageLocation = imgLoc;
            }
```

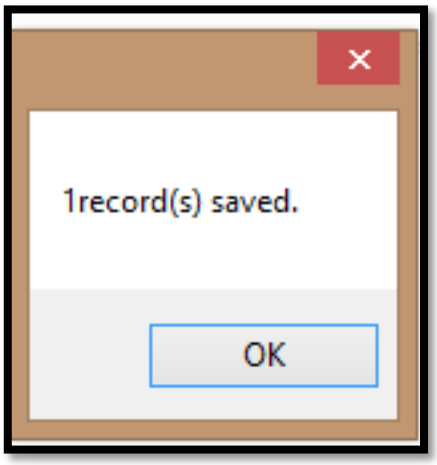
```
        }
catch (Exception ex)
        {
MessageBox.Show(ex.Message);
        }
    }

privatevoid buttonSave_Click(object sender, EventArgs e)
    {
try
        {
byte[] img = null;
FileStream fs = newFileStream(imgLoc, FileMode.Open, FileAccess.Read);
BinaryReader br = newBinaryReader(fs);
            img = br.ReadBytes((int)fs.Length);
string sql = "insert into Employee(EID,FIRST_NAME,LAST_NAME,IMAGE)values(" +
textBoxEID.Text + ",'" + textBoxFName.Text + "','" + textBoxLName.Text + "',@img)";
if (conn.State != ConnectionState.Open)
                conn.Open();
            command = newSqlCommand(sql, conn);
            command.Parameters.Add(newSqlParameter("@img", img));
int x = command.ExecuteNonQuery();
            conn.Close();
MessageBox.Show(x.ToString() + "record(s) saved.");

            textBoxEID.Text = "";
            textBoxFName.Text = "";
            textBoxLName.Text = "";
            picEmp.Image = null;
        }
catch (Exception ex)
        {
            conn.Close();
MessageBox.Show(ex.Message);
        }
    }

privatevoid buttonShow_Click(object sender, EventArgs e)
    {
try
        {
string sql="SELECT FIRST_NAME,LAST_NAME,IMAGE FROM Employee WHERE
EID="+textBoxEID.Text+"";
if (conn.State != ConnectionState.Open)
                conn.Open();
            command = newSqlCommand(sql, conn);
```

```
SqlDataReader reader = command.ExecuteReader();
            reader.Read();
if (reader.HasRows)
            {
                textBoxFName.Text = reader[0].ToString();
                textBoxLName.Text = reader[1].ToString();
byte[] img = (byte[])(reader[2]);
if (img == null)
                    picEmp.Image = null;
else
                {
MemoryStream ms = newMemoryStream(img);
                    picEmp.Image = Image.FromStream(ms);

                }
            }
else
            {
MessageBox.Show("This does not Exist.");
            }
            conn.Close();
        }
catch (Exception ex)
        {
            conn.Close();
MessageBox.Show(ex.Message);
        }
    }
  }
}
```






SQLQuery1.sql - VAIO\...\saniya (54))
/****** Script for SelectTopNRows command from SSMS ******/
SELECT TOP 1000 [EID]
,[FIRST_NAME]
,[LAST_NAME]
,[IMAGE]
FROM [MultimediaDB].[dbo].[Employee]
Results
Messages
EID
FIRST_NAME
LAST_NAME
IMAGE
1
101
yashfeen
khan
0xFFD8FFE158E44578696600004949 2A00080000000D0000...


1record(s) saved.
OK