**INTRODUCTION**

**NAME:** KARTIKEYA NARIAN KHANNA

**BRANCH:** DEVOPS

**ROLL.NO:** R171219049

**SAP ID:** 500078285

**SEMESTER:** 3

**SUBJECT:**ADVANCED DATABASE MANAGEMENT SYSTEMS

**SUBMITTED TO:** Mr. Ankit Khare

**DATE OF SUBMISSION:** 3rd DEC,2020

**EXPERIMENT-1**

**Title:** To understand DDL and DML commands

**Objective:** To understand the concept of designing issue related to the database with creating, populating the tables. Also familiarize students with different ways of manipulation in database.

**1. Create the tables:**

**Table name:** CLIENT1

**Code:**

use lab\_work;

CREATE TABLE CLIENT1 (

Colunmn\_name varchar(10),

Name1 varchar(50),

ADDRESS1 varchar(30),

ADDRESS2 varchar(30),

CITY varchar(15),

PINCODE int,

STATE varchar(15),

BALDUE decimal(10,2)

);

**Table Name:** PRODUCT\_MASTER

**Code:**

use lab\_work;

CREATE TABLE PRODUCT\_Master (

PRODUCTNO varchar(10),

DESCRIPTION1 varchar(50),

PROFITPERCENT decimal(4,2),

UNIT\_MEASURE varchar(30),

QTYONHAND int,

REORDERL\_VL int,

SELLPRICE decimaL (8,2),

COSTPRICE decimal(10,2)

);

**Table Name:** SALESMAN\_MASTER

**Code:**

use lab\_work;

CREATE TABLE SALESMAN\_Master (

SALESMANNO varchar(10),

SALESMANName varchar(50),

ADDRESS1 varchar(30),

ADDRESS2 varchar(30),

CITY varchar(15),

PINCODE int,

STATE varchar(15)

);

**2. Insert the data into their respective tables:**

**CLIENT1**

**Code:**

use lab\_work;

INSERT INTO CLIENT1

VALUES('C00001', 'Ivan bayross','ANDHERI ','EAST', 'Mumbai', 400054, 'Maharashtra', 15000),

('C00002', 'Mamta muzumdar', 'Shanthi Colony Main Road','AI Block','Madras', 780001, 'Tamil nadu', 0),

('C00003', 'Chhaya bankar','ANDHERI','WEST', 'Mumbai', 400057 ,'Maharashtra' ,5000),

('C00004', 'Ashwini joshi', 'Sarvepalli Radhakrishnan Rd','Soladevanahalli', 'Bangalore', 560001, 'Karnataka', 0),

('C00005', 'Hansel colaco', 'Goregaon', 'WEST', 'Mumbai', 400060, 'Maharashtra', 2000),

('C00006', 'Deepak sharma','Sturrock Rd','Attavar', 'Mangalore', 560050 ,'Karnataka', 0);

**PRODUCT\_MASTER**

**Code:**

use lab\_work;

INSERT INTO PRODUCT\_Master

VALUES('P00001', 'T-Shirt', 5 ,'Piece', 200,50, 350, 250),

('P0345', 'Shirts', 6, 'Piece', 150 ,50 ,500, 350),

('P06734' ,'Cotton jeans' ,5 ,'Piece' ,100 ,20 ,600,450),

('P07865' ,'Jeans', 5 ,'Piece', 100, 20, 750, 500),

('P07868' ,'Trousers' ,2 ,'Piece' ,150 ,50 ,850 ,550),

('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450),

('P07965', 'Denim jeans' ,4 ,'Piece' ,100 ,40 ,350 ,250),

('P07975', 'Lycra tops', 5, 'Piece', 70, 30 ,300, 175),

('P08865' ,'Skirts ',5 ,'Piece' ,75 ,30 ,450 ,300);

**SALESMAN\_MASTER**

**Code:**

use lab\_work;

INSERT INTO SALESMAN\_Master

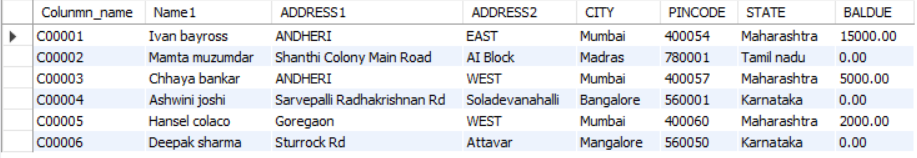
VALUES( 'S00001' ,'Aman' ,'A/14','Worli','Mumbai',400002,'Maharashtra'),

('S00002', 'Omkar','65','Nariman', 'Mumbai', 400001, 'Maharashtra'),

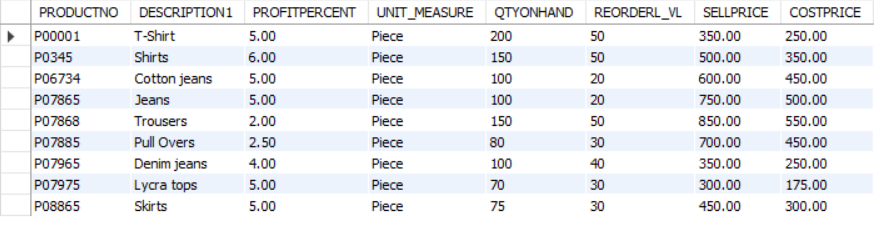
('S00003' ,'Raj' ,'P-7', 'Bandra','Mumbai',400032 ,'Maharashtra'),

('S00004' ,'Ashish' ,'A/5','Juhu','Mumbai',400044 ,'Maharashtra');

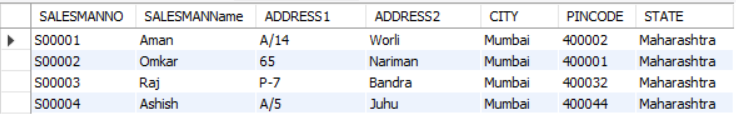
**Output**



**CLIENT1**



**PRODUCT\_Master**



**SALESMAN\_Master**

**3. Exercise on retrieving records from a table**

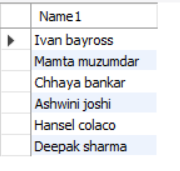
a. Find out the names of all the clients.

**Code:**

use lab\_work;

select Name1 from CLIENT1;

**Output:**



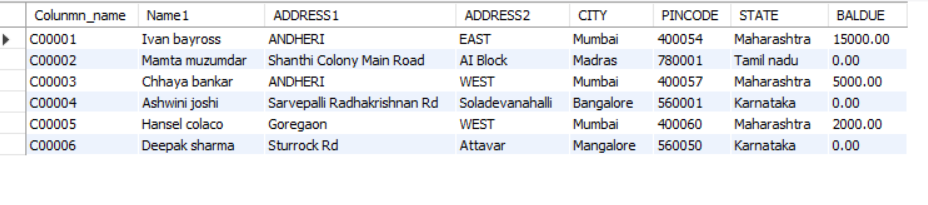
b. Retrieve the entire contents of the Client\_Master table

**Code:**

use lab\_work;

select \* from CLIENT1 ;

**Output:**



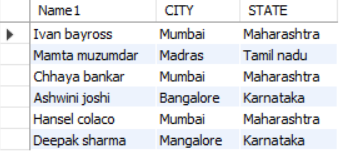
c. Retrieve the list of names, city and the state of all the clients.

**Code:**

use lab\_work;

select Name1,CITY,STATE from CLIENT1;

**Output:**



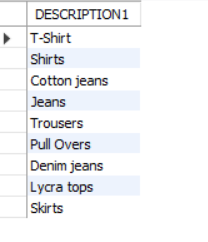
d. List the various products available from the Product\_Master table.

**Code:**

use lab\_work;

select DESCRIPTION1 from PRODUCT\_Master;

**Output:**



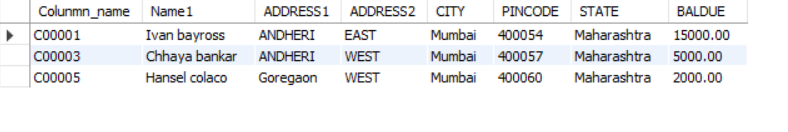
e. List all the clients who are located in Mumbai.

**Code:**

use lab\_work;

select \* from CLIENT1 where CITY = 'Mumbai';

**Output:**



**4. Exercise on updating records in a table**

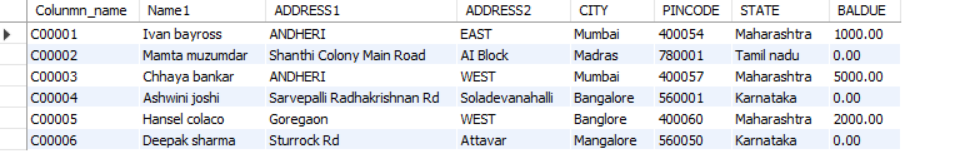
a. Change the city of ClientNo ‘C00005’ to ‘Bangalore’

**Code:**

use lab\_work;

update CLIENT1 set CITY = 'Banglore' where Colunmn\_name = 'C00005';

**Output:**



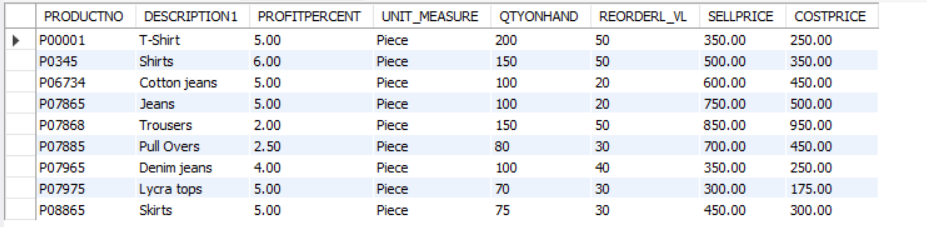
b. Change the BalDue of ClientNo ‘C00001’ to Rs.1000.

**Code:**

use lab\_work;

update CLIENT1 set BALDUE = 1000 where Colunmn\_name = 'C00001';

**Output:**



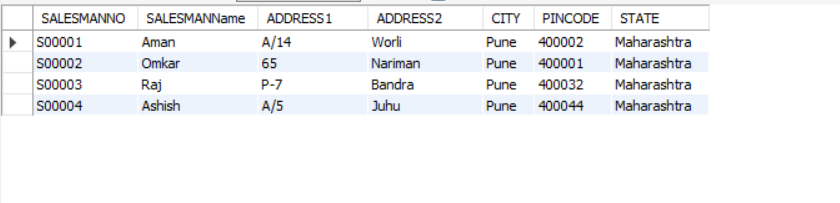
c. Change the city of the salesman to Pune.

**Code:**

use lab\_work;

update SALESMAN\_Master set CITY = 'Pune' ;

**Output:**



**5. Exercise on deleting records in a table**

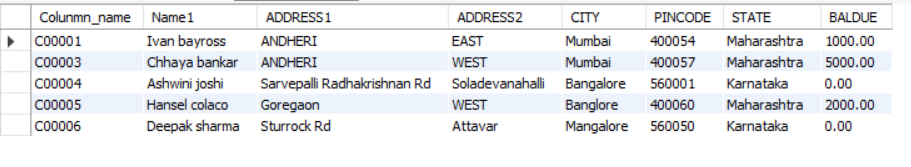
a. Delete all products from Product\_Master where the quantity on hand is equal to 100

**Code:**

use lab\_work;

DELETE FROM PRODUCT\_Master where QTYONHAND =100;

**Output:**



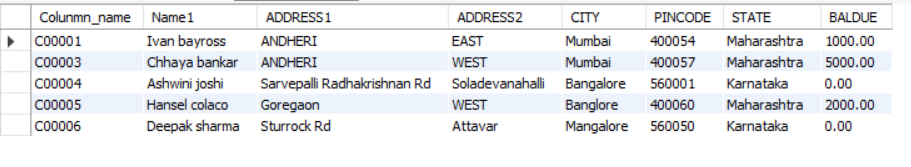
b. Delete from Client1 where the column state holds the value ‘Tamil Nadu’

**Code:**

use lab\_work;

DELETE FROM CLIENT1 where STATE ='Tamil nadu';

**Output:**



**6. Exercise on altering the table structure**

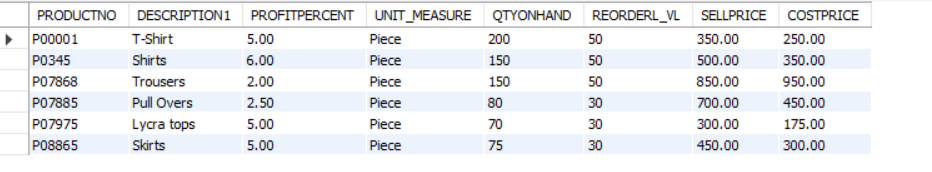
a. Add a column called ‘Telephone’ of data type integer to the Client\_Master table

**Code:**

use lab\_work;

alter Table CLIENT1 Add telephone int(10);

**Output:**



b. Change the size off SellPrice column in Product \_Master to 10, 2.

**Code:**

use lab\_work;

alter Table PRODUCT\_Master Modify SELLPRICE decimaL (10,2);

**7. Exercise on deleting the table structure along with the data**

a. Destroy the table Client1 along with its data.

**Code:**

use lab\_work;

DROP TABLE CLIENT1;

**8. Exercise on renaming the table**

a. Change the name of the Salesman\_Master to sman\_mast

**Code:**

use lab\_work;

Alter Table SALESMAN\_Master Rename to sman\_mast;

**EXPERIMENT-2**

**Title:** To understand and apply the concept of Constraints.

**Objective:** To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

**1. Create the following table.**

**Table name:** CLIENT\_MASTER\_1

**Code:**

use lab\_work;

create table CLIENT\_MASTER\_1 (

CLIENTNO Varchar (6) primary key check(CLIENTNO like '%c'),

NAME1 Varchar (20) not null,

ADDRESS1 Varchar (30),

ADDRESS2 Varchar (30),

CITY Varchar (15),

PINCODE Integer (8),

STATE Varchar (15),

BALDUE Decimal (10,2)

);

**Table Name:** PRODUCT\_MASTER\_1

**Code:**

use lab\_work;

CREATE TABLE PRODUCT\_MASTER\_1(

PRODUCTNO Varchar (6) Primary Key check( PRODUCTNO like '%p'),

DESCRIPTIONS Varchar (15) Not Null,

PROFITPERCENT Decimal (4,2) Not Null,

UNIT\_MEASURE Varchar (10) Not Null,

QTYONHAND Integer (8) Not Null,

REORDERL\_VL Integer (8) Not Null,

SELLPRICE Decimal (8,2) Not Null,

COSTPRICE Decimal (8,2) Not Null

);

**Table Name:** SALESMAN\_MASTER \_1

**Code:**

use lab\_work;

CREATE TABLE SALESMAN\_MASTER\_1(

SALESMANNO Varchar (6) Primary Key check(SALESMANNO like '%s') ,

SALESMANNAME Varchar (20) Not Null,

ADDRESS1 Varchar (30) Not Null,

ADDRESS2 Varchar (30),

CITY Varchar (20),

PINCODE Integer (8),

STATE Varchar (20),

SALAMT Real (8,2) Not Null check(SALAMT!=0),

TGTTOGET Decimal (6,2) Not Null check(TGTTOGET!=0),

YTDSALES Double (6,2) Not Null,

REMARKS Varchar (60)

);

**2. Reinsert the data in these tables.**

**Table name:** CLIENT\_MASTER\_1

**Code:**

use lab\_work;

INSERT INTO CLIENT\_MASTER\_1

VALUES('c00001', 'Ivan bayross','ANDHERI ','EAST', 'Mumbai', 400054, 'Maharashtra', 15000),

('c00002', 'Mamta muzumdar', 'Shanthi Colony Main Road','AI Block','Madras', 780001, 'Tamil nadu', 0),

('c00003', 'Chhaya bankar','ANDHERI','WEST', 'Mumbai', 400057 ,'Maharashtra' ,5000),

('c00004', 'Ashwini joshi', 'Sarvepalli Radhakrishnan Rd','Soladevanahalli', 'Bangalore', 560001, 'Karnataka', 0);

**Table Name:** PRODUCT\_MASTER\_1

**Code:**

use lab\_work;

INSERT INTO PRODUCT\_MASTER\_1

VALUES('P00001', 'T-Shirt', 5 , 200,50, 350, 250,56),

('P0345', 'Shirts', 6, 150 ,50 ,500, 350,53),

('P06734' ,'Cotton jeans' ,5 ,100 ,20 ,600,450,54),

('P08865' ,'Skirts ',5 ,75 ,30 ,450 ,300,64);

**Table Name:** SALESMAN\_MASTER \_1

**Code:**

use lab\_work;

INSERT INTO SALESMAN\_MASTER\_1

VALUES( 's00001' ,'Aman' ,'A/14','Worli','Mumbai',400002,'Maharashtra',56.0,85.5,556.0,'Na'),

('s00002', 'Omkar','65','Nariman', 'Mumbai', 400001, 'Maharashtra',78.0,44.8,564.0,'NA'),

('s00003' ,'Raj' ,'P-7', 'Bandra','Mumbai',400032 ,'Maharashtra',75.0,98.0,435.5,'Na'),

('s00004' ,'Ashish' ,'A/5','Juhu','Mumbai',400044 ,'Maharashtra',74.5,55.2,554.3,'Na');

**3. Display the contents of each table**

**Code:**

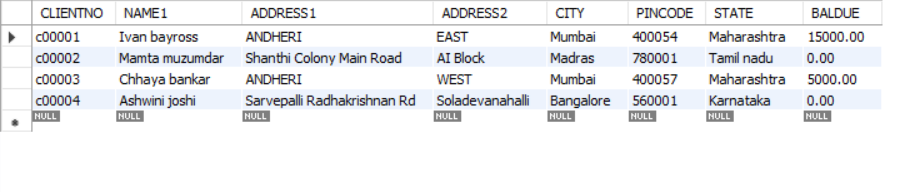
use lab\_work;

select \* from CLIENT\_MASTER\_1 ;

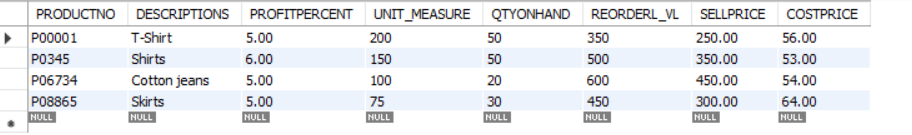
select \* from PRODUCT\_MASTER\_1;

select \* from SALESMAN\_MASTER\_1;

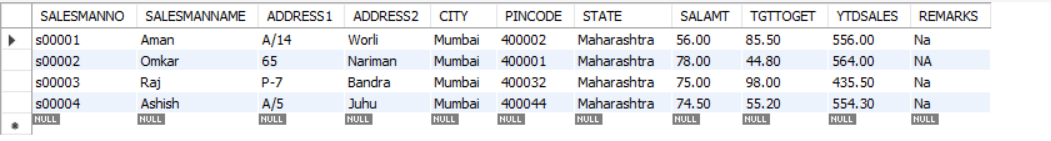
**Output:**



**CLIENT\_MASTER\_1**



**PRODUCT\_MASTER\_1**



**SALESMAN\_MASTER\_1**

**4. Create table AUTHOR = {Author\_ID , Lastname, Firstname, Email, City, Country}**

**Code:**

use lab\_work;

CREATE TABLE AUTHOR

(

Author\_ID varchar(5) primary key,

Lastname varchar(15),

Firstname varchar(15),

Email varchar(40),

City varchar(15),

Country varchar(15)

);

**5. Create Table BOOK={ Book\_ID, Book\_Title, Copies}**

**Code:**

use lab\_work;

CREATE TABLE BOOK

(

Book\_ID varchar(5) primary key check(Book\_ID like 'B%'),

Book\_Title varchar(30) not null,

Copies int check(copies >2)

);

**6. Create table AUTHOR\_LIST = {Author\_ID , Book\_ID , Role}**

**Code:**

use lab\_work;

CREATE TABLE AUTHOR\_LIST(

Author\_ID varchar(5) references AUTHOR(Author\_ID),

Book\_ID varchar(5) references BOOK(Book\_ID),

Roles varchar(15),

primary key(Author\_ID,Book\_ID)

);

**7. Add four records in each tables AUTHOR, BOOK, Author\_LIST.**

Table Name: AUTHOR

**Code:**

use lab\_work;

INSERT INTO AUTHOR

VALUES ('A0001','Khan','Haseen','haseen@gmail.com','Dehradun','India'),

('A0002','kumar','Kishore', 'kishore@gmail.com','Delhi','India'),

('A0003','Pant', 'Gaurav', 'gaurav@gmail.com', 'Mumbai','India'),

('A0004', 'Katariya', 'Pal', 'Pal@gmail.com', 'Goa', 'India');

Table Name: BOOK

**Code:**

use lab\_work;

INSERT INTO BOOK

VALUES ( 'B0001', 'xyz', 5),

('B0002', 'abc', 8),

('B0003', 'def',9),

('B0004','jkl',6);

Table Name: AUTHOR\_LIST

**Code:**

use lab\_work;

INSERT INTO AUTHOR\_LIST

VALUES ('A0001','B0001', 'xyz'),

('A0002','B0002', 'abc'),

('A0003','B0003', 'def'),

('A0004','B0004', 'jkl');

**8. Alter structure of table AUTHOR\_LIST add the field Publisher data type of 30 Character.**

**Code:**

use lab\_work;

alter Table AUTHOR\_LIST Add Publisher varchar(30);

**EXPERIMENT-3**

**Title:** To understand and use SQL Sub-Query

**Objective:** To understand the use of sql subquery.

**1. Create the following table.**

(a) Supplier-(scode,sname,scity,turnover)

**Code:**

use lab\_work;

create table Supplier (

Scode varchar(10) primary key,

Sname varchar(50),

Scity varchar(20),

Turnover int

);

(b) Part-(pcode,weigh,color,cost,sellingprice)

**Code:**

use lab\_work;

create table Part(

Pcode varchar(10) primary key,

Weigh varchar(20),

Color varchar(30),

Cost decimal(8,2),

Sellingprice decimal (8,2)

);

(c) Supplier\_Part-(scode,pcode,qty)

**Code:**

use lab\_work;

create table Supplier\_Part (

Scode varchar(10) references Supplier(Scode),

Pcode varchar(10) references Part(Pcode),

Qty int,

primary key(Scode,Pcode)

);

**2. Populate the table**

(a) Supplier

**Code:**

use lab\_work;

INSERT INTO Supplier

VALUES ('A0001','Haseen', 'Dehradun',50000),

('A0002','Kishore','Delhi',1500000),

('A0003','Gaurav','Mumbai', 200000),

('A0004','Rachit', 'Mumbai',5400000),

('A0005', 'Ritvik', 'Chennai',800000),

('A0006','Akshay', 'Mumbai', 7900000),

('A0007','Ajay', 'Delhi' , 90000),

('A0008','Pal', 'Mumbai',5500000);

(b) Part

**Code:**

use lab\_work;

INSERT INTO Part

VALUES ('P0001','26','Black','100','120'),

('P0002','27','White','80','150'),

('P0003','37','Red','75','130'),

('P0004','40','Blue','70','100'),

('P0005', '28','Green', '20', '120'),

('P0006', '32', 'Yellow','30','85'),

('P0007', '34', 'Orange', '40','78'),

('P0008', '29', 'Grey', '45', '200');

(c) Supplier\_part

**Code:**

use lab\_work;

INSERT INTO Supplier\_Part

VALUES ('A0001','P0001',500),

('A0002','P0002',1000),

('A0003','P0003',200),

('A0004','P0004',250),

('A0005','P0005',400),

('A0006','P0006',350),

('A0007','P0007',850),

('A0008','P0008',650);

**Output**

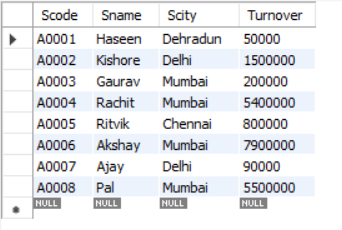
**Code:**

use lab\_work;

select \* from Supplier;

select \* from Part;

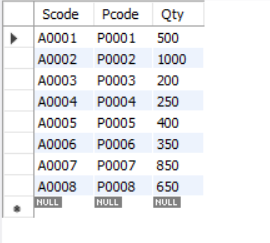
select \* from Supplier\_Part;



**Supplier**



**Part**



**Supplier\_part**

**3. Write appropriate SQL Statement for the following:**

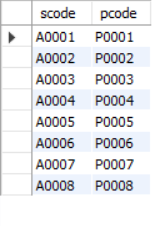
1. Get the supplier number and part number in ascending order of supplier number

**Code:**

use lab\_work;

select Scode, Pcode from supplier\_part order by Scode;

**Output:**



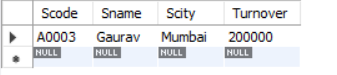
2. Get the details of supplier who operate from Bombay with turnover 50.

**Code:**

use lab\_work;

select \* from supplier where Scity = 'Mumbai' and Turnover='200000';

**Output:**



3. Get the total number of supplier.

**Code:**

use lab\_work;

select count(sname) from supplier;

**Output:**



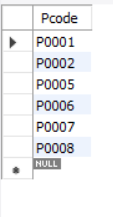
4. Get the part number weighing between 25 and 35.

**Code:**

use lab\_work;

select Pcode from part where Weigh between 25 and 35;

**Output:**



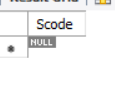
5. Get the supplier number whose turnover is null.

**Code:**

use lab\_work;

select Scode from supplier where Turnover='null';

**Output:**



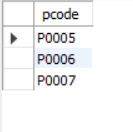
6. Get the part number that cost 20, 30 or 40 rupees.

**Code:**

use lab\_work;

select pcode from part where cost in (20,30,40);

**Output:**



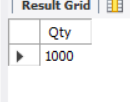
7. Get the total quantity of part 2 that is supplied.

**Code:**

use lab\_work;

select Qty from supplier\_part where Pcode='P0002';

**Output:**



8. Get the name of supplier who supply part 2

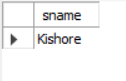
**Code:**

use lab\_work;

select sname from supplier where scode =

(select scode from supplier\_part where pcode='P0002');

**Output:**



9. Get the part number whose cost is greater than the average cost.

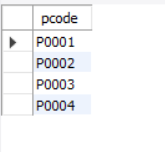
**Code:**

use lab\_work;

select pcode from part where

cost > (select avg(cost) from part);

**output:**



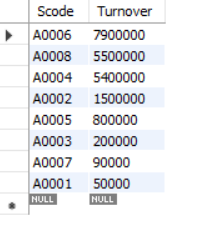
10. Get the supplier number and turnover in descending order of turnover.

**Code:**

use lab\_work;

select Scode, Turnover from supplier order by Turnover desc;

**Output:**



**EXPERIMENT-4**

**Title:** Use of Inbuilt functions and relational algebra operation.

**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

**1. Create the following two tables**

**Table name:** EMP

**Code:**

use lab\_work;

CREATE TABLE EMP (

EMPNO int,

ENAME varchar(30),

JOB varchar(30),

MGR varchar(30) ,

HIREDATE date ,

SAL int,

COMM varchar(30),

DEPTNO int REFERENCES DEPT ( DEPTNO)

);

**Table name:** DEPT

**Code:**

use lab\_work;

CREATE TABLE DEPT (

DEPTNO int primary key,

DNAME varchar(30),

LOC varchar(30)

);

**2. Reinsert the data in these tables.**

**Table name:** EMP

**Code:**

use lab\_work;

INSERT INTO EMP

VALUES ('7369', 'SMITH' ,'CLERK', '7902', '1980-12-17','500', '800','20'),

('7499' ,'ALLEN' ,'SALESMAN' ,'7698' ,'1981-02-20' ,'1600' ,'300','30'),

('7521', 'WARD', 'SALESMAN', '7698', '1981-02-22', '1250', '500', '30'),

('7566' ,'JONES' ,'MANAGER' ,'7839' ,'1981-04-02' ,'2975' ,'','20'),

('7654', 'MARTIN', 'SALESMAN', '7698', '1981-09-28', '1250', '1400', '30'),

('7698' ,'BLAKE' ,'MANAGER' ,'7839' ,'1981-05-01' ,'2850' ,'','30'),

('7782', 'CLARK', 'MANAGER' ,'7839', '1981-06-09', '2450','' ,'10'),

('7788', 'SCOTT', 'ANALYST', '7566', '1982-12-09', '3000', '','20'),

( '7839', 'KING', 'PRESIDENT', '','1981-11-17', '5000','','10'),

('7844' ,'TURNER' ,'SALESMAN' ,'7698' ,'1981-09-08' ,'1500' ,'0','30'),

('7876', 'ADAMS', 'CLERK' ,'7788', '1981-01-12', '1100','', '20'),

('7900' ,'JAMES' ,'CLERK' ,'7698' ,'1981-12-03' ,'950' ,'','30'),

('7902', 'FORD', 'ANALYST', '7566', '1981-12-03', '3000','', '20'),

('7934' ,'MILLER' ,'CLERK' ,'7782' ,'1982-01-23' ,'1300','' ,'10');

**Table name:** DEPT

**Code:**

use lab\_work;

INSERT INTO DEPT

VALUES ( '10', 'ACCOUNTING', 'NEW YORK'),

('20' ,'RESEARCH' ,'DALLAS'),

('30', 'SALES', 'CHICAGO'),

('40' ,'OPERATIONS' ,'BOSTON');

**3. Display the contents of each table**

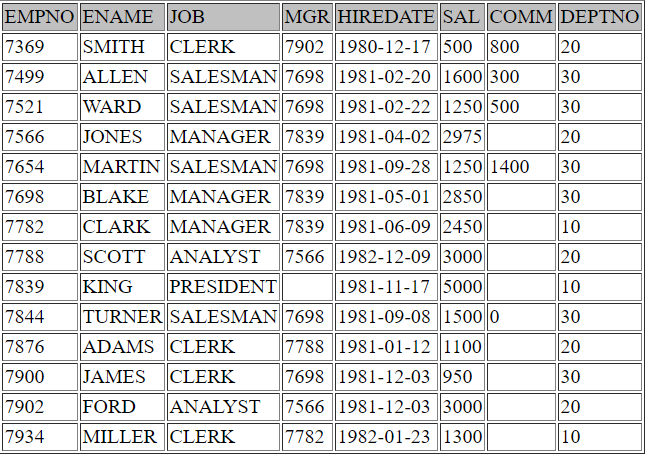
**Code:**

use lab\_work;

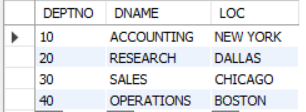
select \* from EMP ;

select \* from DEPT ;

**Output:**

****

**EMP**

****

**DEPT**

**4.Write the Nested Queries for the following queries.**

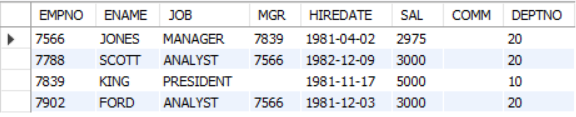
1. List the details of the emps whose Salaries more than the employee BLAKE.

**Code:**

use lab\_work;

select \* from EMP where SAL>( select SAL from EMP where ENAME='BLAKE');

**Output:**

****

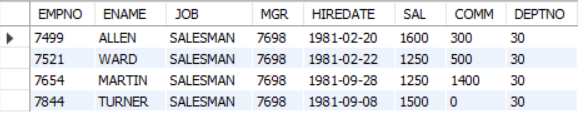
2. List the emps whose Jobs are same as ALLEN.

**Code:**

use lab\_work;

select \* from EMP where JOB = (select JOB from EMP where ENAME ='ALLEN');

**Output:**

****

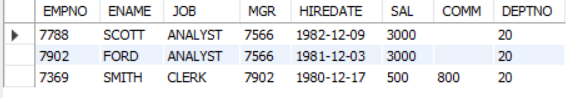
3. List the Emps whose Sal is same as FORD or SMITH in desc order of Names.

**Code:**

use lab\_work;

Select \* from EMP where SAL in (select SAL from EMP where ( ENAME = 'SMITH' or ENAME = 'FORD' )) order by SAL desc;

**Output:**

****

4. List the emps Whose Jobs are same as MILLER or Sal is more than ALLEN.

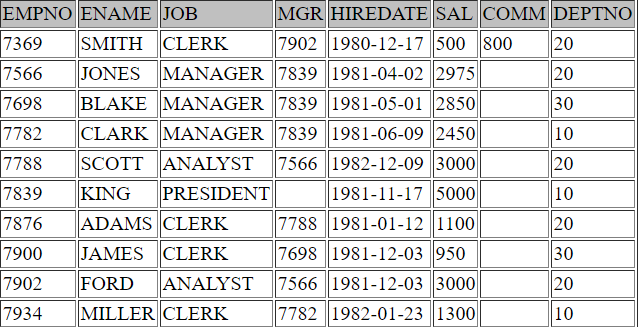
**Code:**

use lab\_work;

select \* from EMP where JOB = (select JOB from EMP where ENAME ='MILLER' ) or

SAL>(select SAL from EMP where ENAME = 'ALLEN');

**Output:**

****

5. Find the highest paid employee of sales department.

**Code:**

use lab\_work;

select \* from EMP where sal in (select max(SAL) from EMP where DEPTNO=(select DEPTNO from DEPT where DNAME= 'SALES'));

**Output:**

****

6. List the employees who are senior to most recently hired employee working under king.

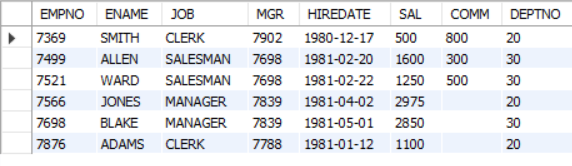
**Code:**

use lab\_work;

select \* from EMP where HIREDATE < (select max(HIREDATE) from EMP where MGR

in (select EMPNO from EMP where ENAME= 'KING')) ;

**Output:**

****

7. List the names of the emps who are getting the highest sal dept wise.

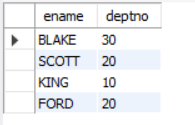
**Code:**

use lab\_work;

select ENAME, DEPTNO from EMP where SAL in

(select max(SAL) from EMP group by DEPTNO) ;

**Output:**

****

8. List the emps whose sal is equal to the average of max and minimum .

**Code:**

use lab\_work;

select \* from EMP where SAL =(select (max(SAL)+min(SAL))/2 from EMP);

**Output:**

****

**No Data Found**

9. List the emps who joined in the company on the same date.

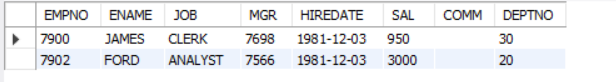
**Code:**

use lab\_work;

select \* from EMP E where HIREDATE in

(select HIREDATE from EMP where E.EMPNO <> EMPNO);

**Output:**

****

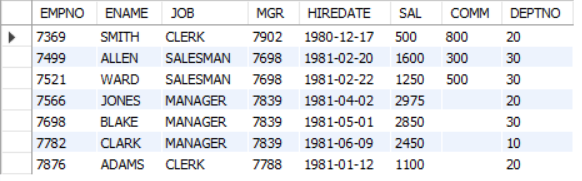
10. Find out the emps who joined in the company before their Managers.

**Code:**

use lab\_work;

select \* FROM EMP E WHERE HIREDATE < (select HIREDATE FROM EMP WHERE EMPNO = E.MGR);

**Output:**

****

**EXPERIMENT-5**

**Title:** Use of different SQL clauses and join.

**Objective:** To understand the use of group by and having clause and execute the SQL commands using JOIN.

**1. Write the SQL Queries for the following queries**

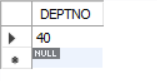
1. List the Deptno where there are no emps.

**Code:**

use lab\_work;

select DEPTNO from DEPT where DEPTNO not in (select DEPTNO from EMP);

**Output:**

****

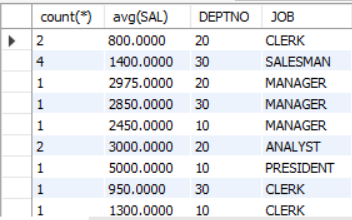
2. List the No.of emp’s and Avg salary within each department for each job.

**Code:**

use lab\_work;

select count(\*), avg(SAL),DEPTNO,JOB from EMP group by DEPTNO,JOB;

**Output:**

****

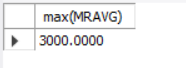
3. Find the maximum average salary drawn for each job except for ‘President’.

**Code:**

use lab\_work;

select max(MRAVG) from(select avg(SAL) MRAVG from EMP where JOB!='PRESIDENT' group by JOB)a

**Output:**

****

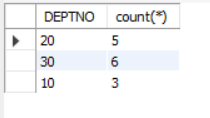
4. List the department details where at least two emps are working.

**Code:**

use lab\_work;

select DEPTNO, count(\*) from EMP group by DEPTNO having count(\*)>=2;

**Output:**

****

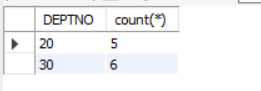
5. List the no. of emps in each department where the no. is more than 3.

**Code:**

use lab\_work;

select DEPTNO, count(\*) from EMP group by DEPTNO having count(\*)>3;

**Output:**

****

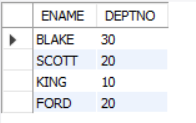
6. List the names of the emps who are getting the highest sal dept wise.

**Code:**

use lab\_work;

select ENAME ,DEPTNO from EMP where SAL in ( select max(SAL) from EMP group by DEPTNO);

**Output:**

****

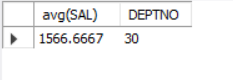
7. List the Deptno and their average salaries for dept with the average salary less than the averages for all departments.

**Code:**

use lab\_work;

select avg(SAL), DEPTNO from EMP group by DEPTNO having avg(SAL)<(select avg(SAL) from EMP);

**Output:**

****

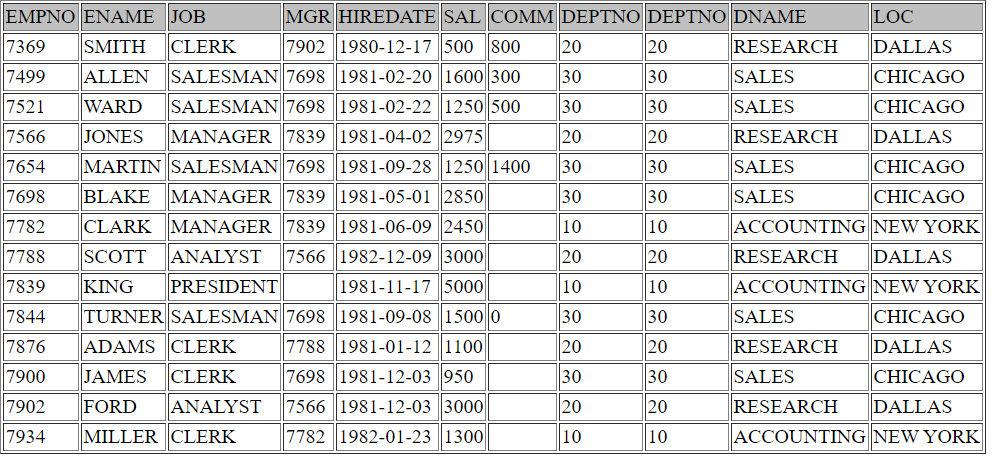
**2. Execute the experiment 4 using sql join.**

**Code:**

use lab\_work;

select \* from EMP join DEPT on EMP.DEPTNO=DEPT.DEPTNO;

**Output:**

****

**EXPERIMENT-6**

**Title:**To understand the concepts of Views.

**Objective:** Students will be able to implement the concept of views.

1. Create table of table name: EMPLOYEES

use lab\_work;

CREATE TABLE EMPLOYEES (

Employee\_id Char (10) primary key,

First\_Name Char (30) not null,

Last\_Name Char (30) not null,

DOB Date,

Salary int not null ,

Department\_id varchar (10)

);

1) Create View of name emp\_view and the column would be Employee\_id, Last\_Name, salary and department\_id only.

use lab\_work;

CREATE VIEW emp\_view AS

SELECT Employee\_id,Last\_Name,Salary, Department\_id

FROM EMPLOYEES ;

2) Insert values into view(remove the NOT NULL constraint and then insert values):

use lab\_work;

Alter Table EMPLOYEES modify First\_Name varchar(30) null;

Alter Table EMPLOYEES modify Last\_Name varchar(30) null;

Alter Table EMPLOYEES modify Salary int null;

use lab\_work;

insert into emp\_view

values('E01','Khan','100000','D01'),

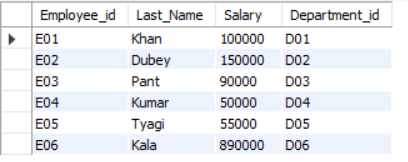
('E02','Dubey','150000','D02'),

('E03','Pant','90000','D03'),

('E04','Kumar','50000','D04'),

('E05','Tyagi','55000','D05'),

('E06','Kala','890000','D06');



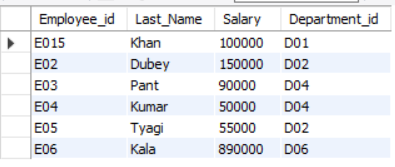
3) Modify, delete and drop operations are performed on view

use lab\_work;

Update emp\_view set Department\_id='D04' where Employee\_id='E03';

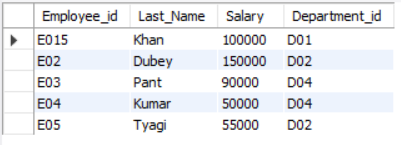
Update emp\_view set Employee\_id='E015' where Department\_id='D01';

Update emp\_view set Department\_id='D02' where Employee\_id='E05';



use lab\_work;

Delete from emp\_view where Salary='890000';



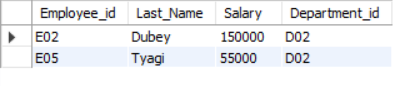
use lab\_work;

Drop view emp\_view;

4) Creates a view named salary\_view. The view shows the employees in department 20 and their annual salary.

use lab\_work;

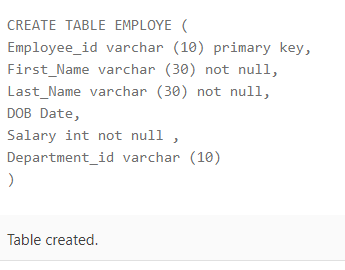
create view Sal\_view as select Employee\_id,Last\_Name ,Salary,Department\_id from EMPLOYEES where Department\_id='D02';



**EXPERIMENT-7**

**Title:** To understand the concepts of Index.

**Objective:** Students will be able to implement the concept of index.



insert into EMPLOYE

values('E01','Haseen','Khan',to\_date('29-12-1996','dd-mm-yyyy'),'100000','D01')

insert into EMPLOYE

values('E02' ,'Juhi','Dubey',to\_date('17-02-1999','dd-mm-yyyy'),'150000','D02')

insert into EMPLOYE

values('E03','Gaurav','Pant',to\_date('30-03-1994','dd-mm-yyyy'),'90000','D03')

insert into EMPLOYE

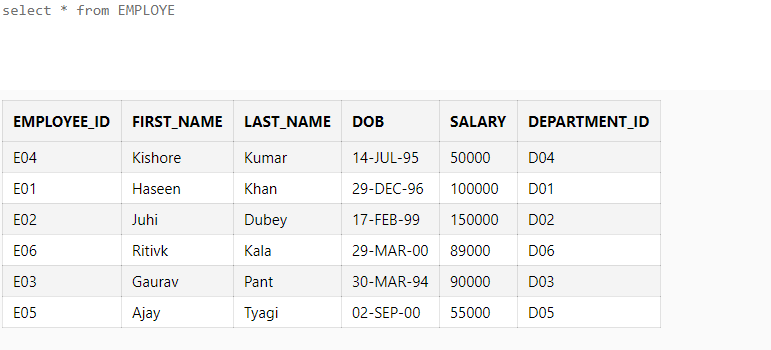
values('E04' ,'Kishore','Kumar',to\_date('14-07-1995','dd-mm-yyyy'),'50000','D04')

insert into EMPLOYE

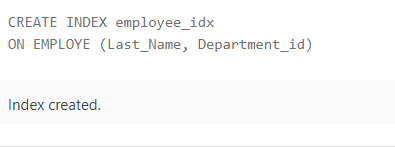
values('E05' ,'Ajay','Tyagi',to\_date('02-09-2000','dd-mm-yyyy'),'55000','D05')

insert into EMPLOYE

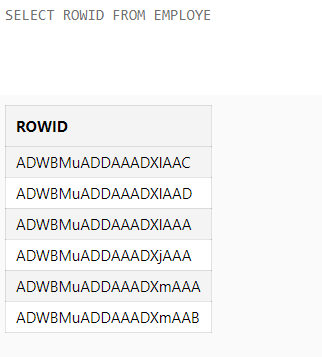
values('E06' ,'Ritivk','Kala',to\_date('29-03-2000','dd-mm-yyyy'),'89000','D06')

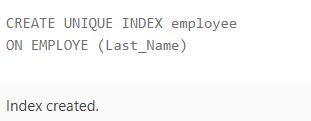


1) Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id

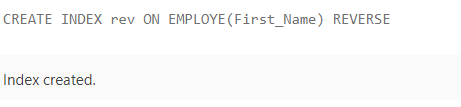


2) Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.

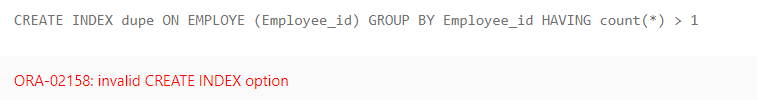




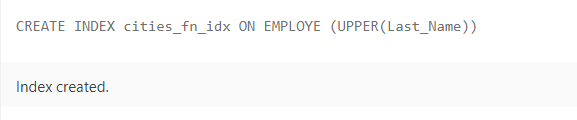
3) Create a reverse index on employee\_id column of the EMPLOYEES.



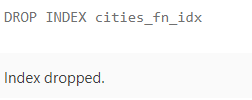
4) Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.



5) Create Function-based indexes defined on the SQL functions UPPER(column\_name) or LOWER(column\_name) to facilitate case-insensitive searches(on column Last\_Name).



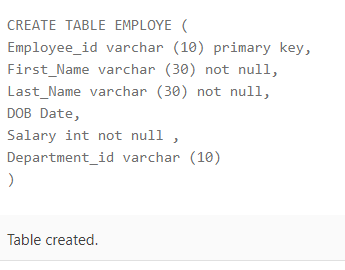
6) Drop the function based index on column Last\_Name



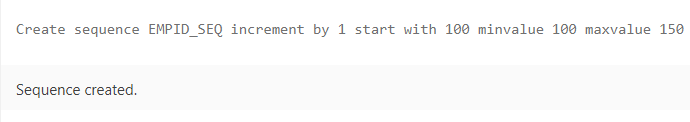
**EXPERIMENT-8**

**Title:** To understand the concepts of Sequence.

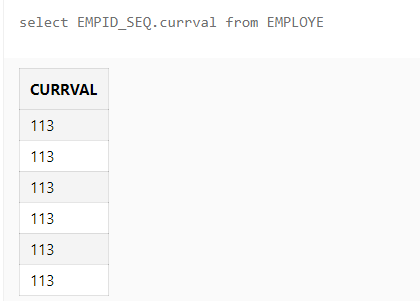
**Objective:** Students will be able to implement the concept of sequence.

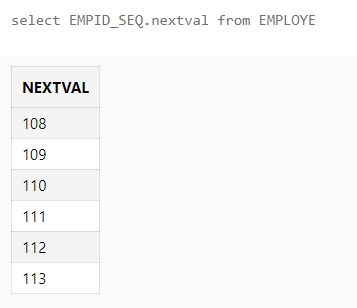
****

1) Create a sequence by name EMPID\_SEQ starting with value 100 with an interval of 1.

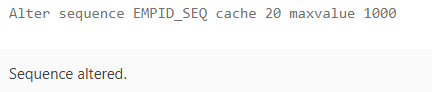


2) Write a SQL command for finding the current and the next status of EMPID\_SEQ.





3) Change the Cache value of the sequence EMPID\_SEQ to 20 and maxvalue to 1000.



4) Insert values in employees table using sequences for employee\_id column.

insert into EMPLOYE

values(EMPID\_SEQ.nextval ,'Haseen','Khan',to\_date('29-12-1996','dd-mm-yyyy'),'100000','D01')

insert into EMPLOYE

values(EMPID\_SEQ.nextval ,'Juhi','Dubey',to\_date('17-02-1999','dd-mm-yyyy'),'150000','D02')

insert into EMPLOYE

values(EMPID\_SEQ.nextval ,'Gaurav','Pant',to\_date('30-03-1994','dd-mm-yyyy'),'90000','D03')

insert into EMPLOYE

values(EMPID\_SEQ.nextval ,'Kishore','Kumar',to\_date('14-07-1995','dd-mm-yyyy'),'50000','D04')

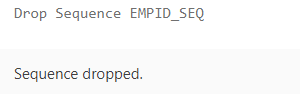
insert into EMPLOYE

values(EMPID\_SEQ.nextval ,'Ajay','Tyagi',to\_date('02-09-2000','dd-mm-yyyy'),'55000','D05')

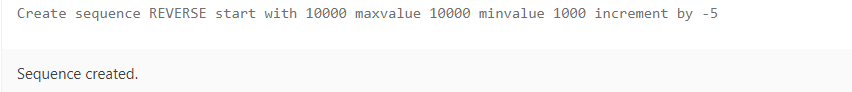
insert into EMPLOYE

values(EMPID\_SEQ.nextval ,'Ritivk','Kala',to\_date('29-03-2000','dd-mm-yyyy'),'89000','D06')

5) Drop sequence EMPID\_SEQ.



6) Create a sequence called REVERSE to generate numbers in the descending order from 10000 to 1000 with a decrement of 5.

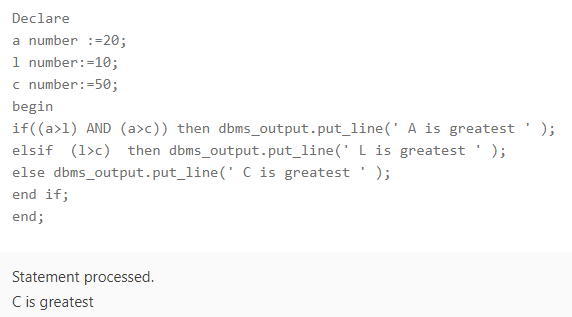


**EXPERIMENT-9**

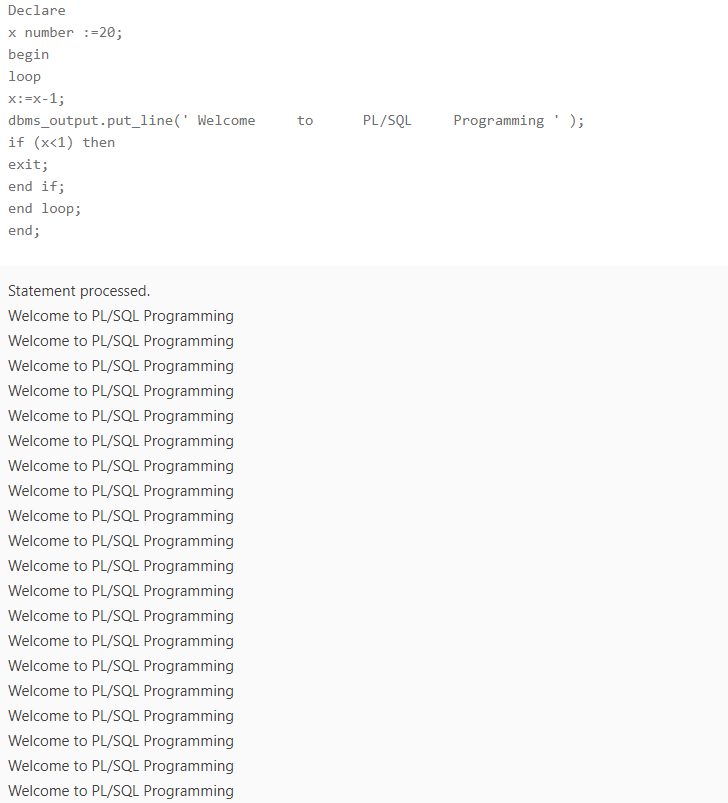
**Title:**To understand the concepts of PL/SQL programming.

**Objective:** Students will be able to implement the basic concepts of Pl/SQL.

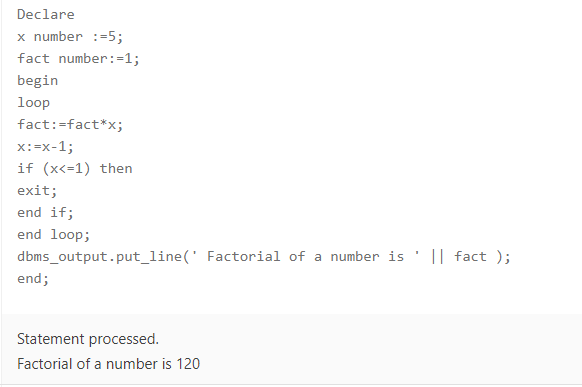
1) Write a PL/SQL code to accept the value of A, B & C display which is greater.



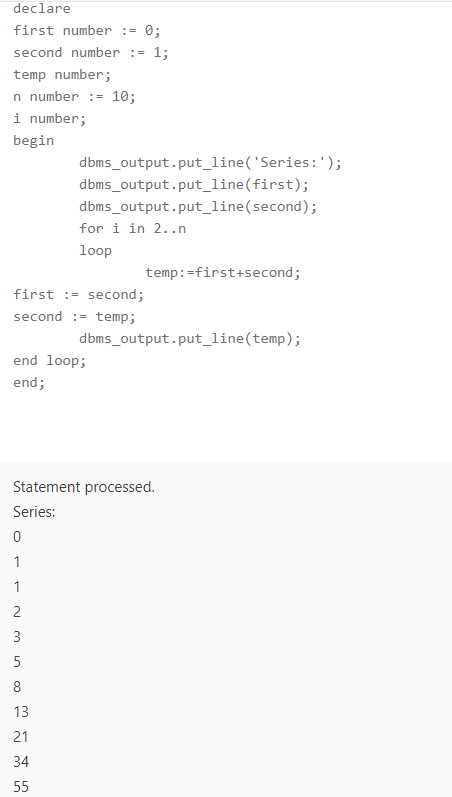
2) Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.



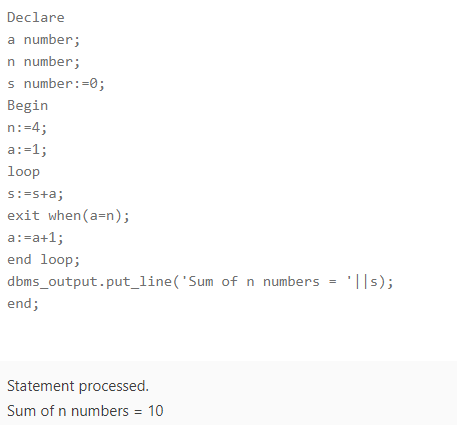
3) Write a PL/SQL code block to find the factorial of a number.



4) Write a PL/SQL program to generate Fibonacci series.



5) Write a PL/SQL code to fund the sum of first N numbers



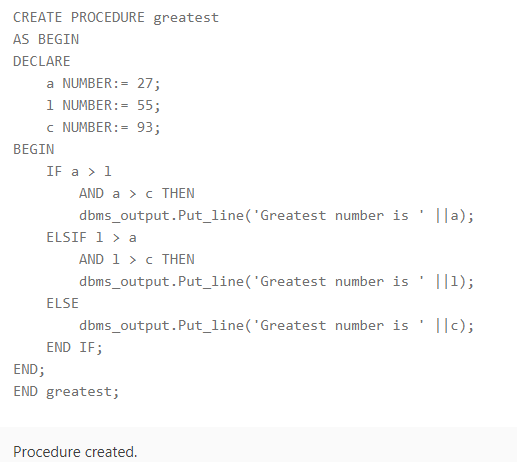
. **EXPERIMENT-10**

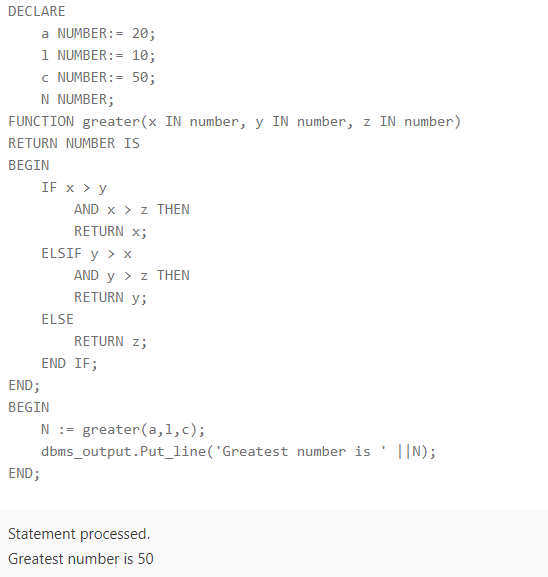
**Title:**To understand the concepts of function and procedure in PL/SQL.

**Objective:** Students will be able to implement the Pl/SQL programs using function and procedure.

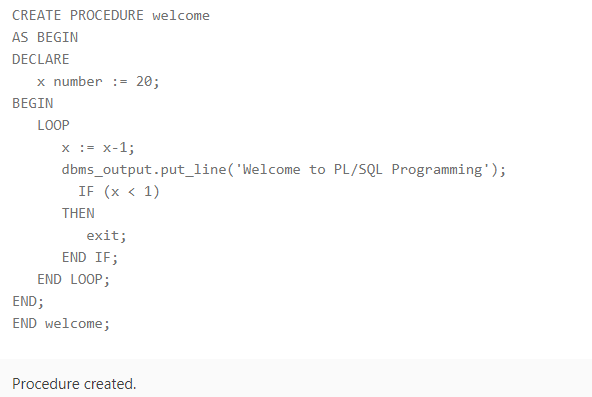
Implement the experiments of lab 9 using functions and procedures

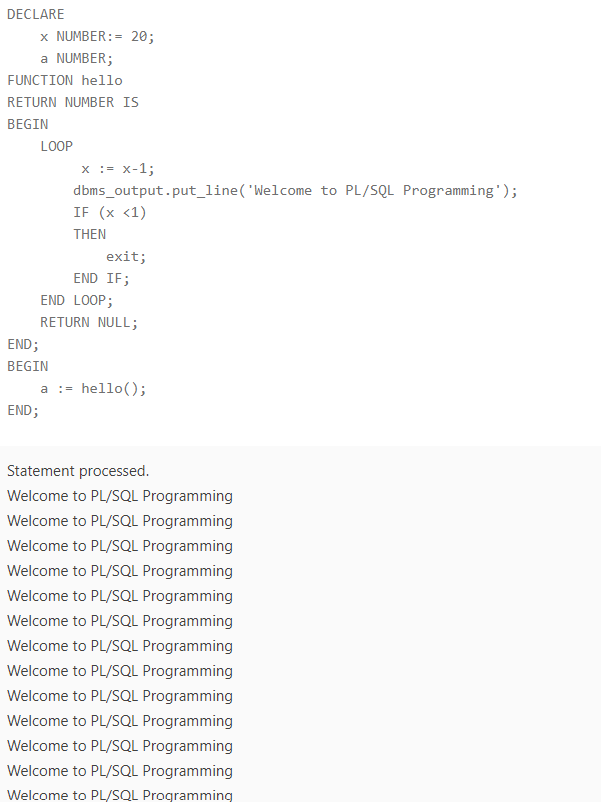
1) Write a PL/SQL code to accept the value of A, B & C display which is greater.



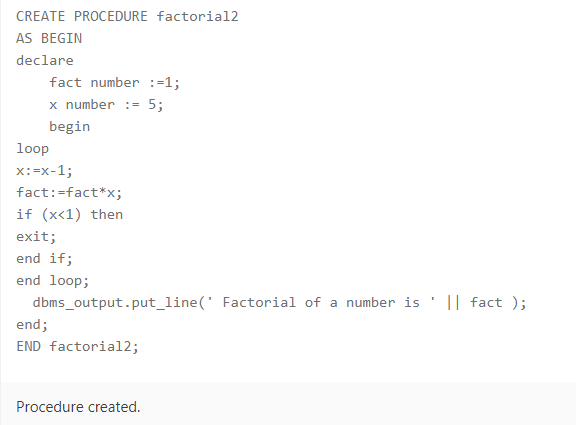


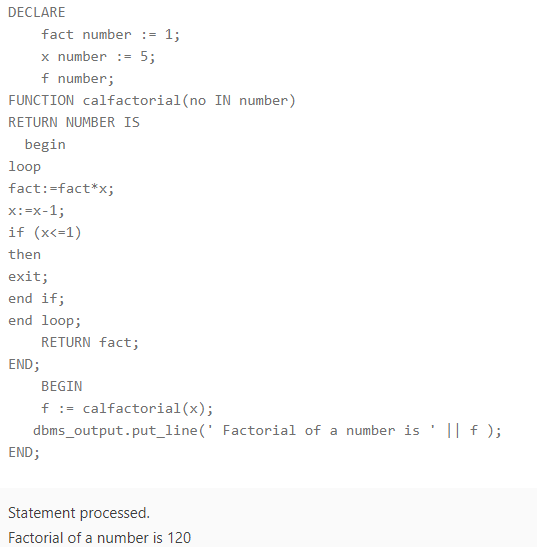
2) Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.



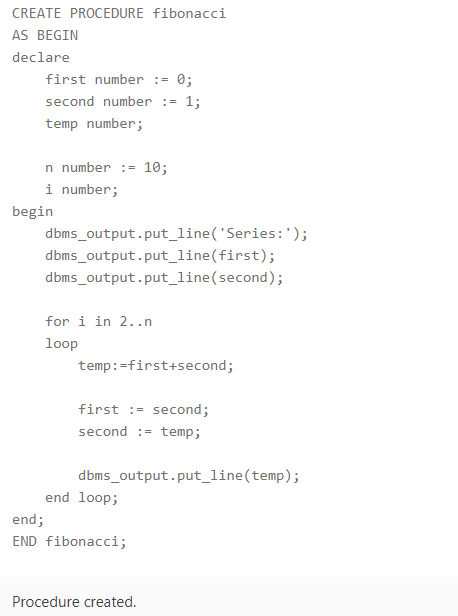


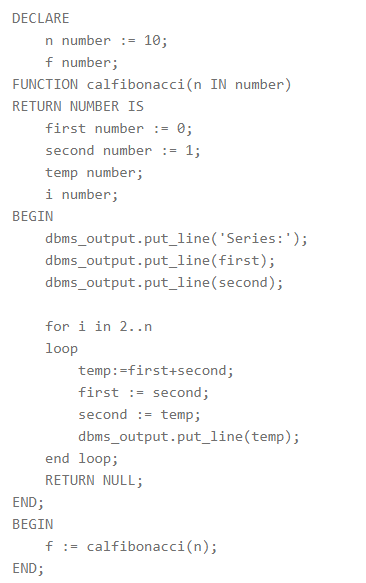
3) Write a PL/SQL code block to find the factorial of a number.

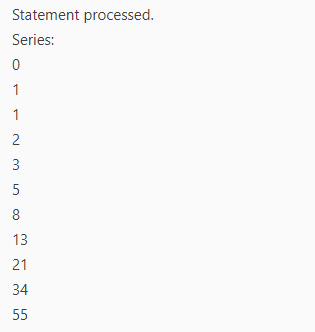




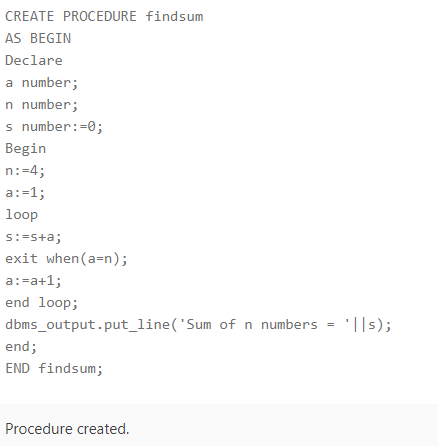
4) Write a PL/SQL program to generate Fibonacci series.

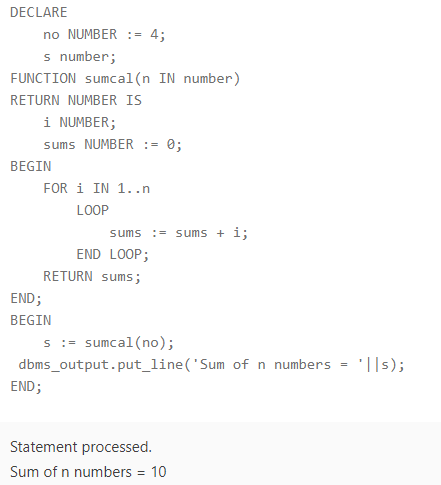






5) Write a PL/SQL code to fund the sum of first N numbers

****

****