FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

20MCA134 ADVANCED DBMS LAB

LABORATORY RECORD

Name: ANNMARIYA LALU

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 2 Batch: A Roll No:23

University Register Number: FIT24MCA-2023

JUNE 2025

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

<u>CERTIFICATE</u>

This is to certify that this is a Bonafide record of the Practical work done and submitted to APJ Abdul Kalam Technological University in the partial fulfilment for the award of the Master Of Computer Applications by **ANNMARIYA LALU (FIT24MCA-2023)** *in the* **20MCA134 ADVANCED DBMS LAB** *of the Federal Institute of Science and Technology during the academic year 2024-2025.*

Signature of Staff in Charge	Signature of H O D
Ms. Anju L.	Dr. Deepa Mary Mathews

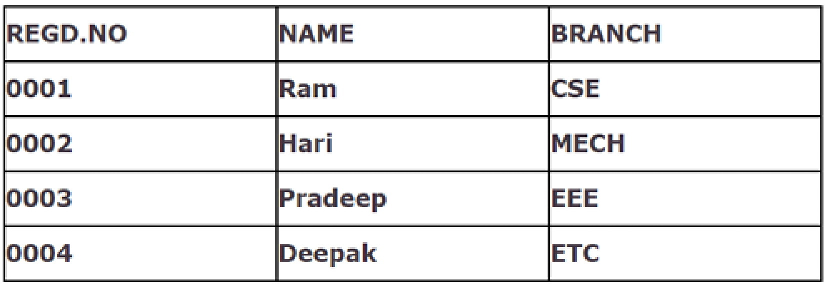
Date of University practical examination ………………………

Signature of Internal Examiner	Signature of External Examiner

CONTENT

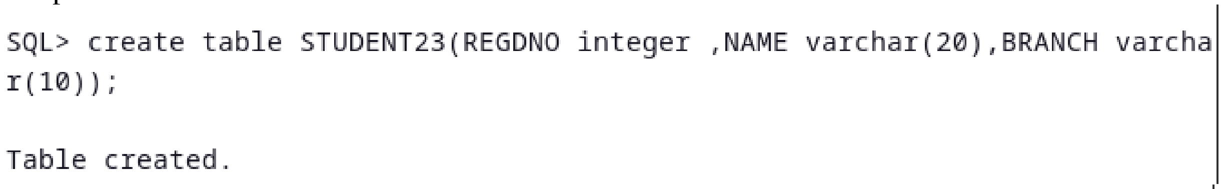
Experiment 1: Creation of a database using DDL commands including integrity constraints

1.Create a table called student with the following values and Write a SQL command which will show the entire STUDENT table.

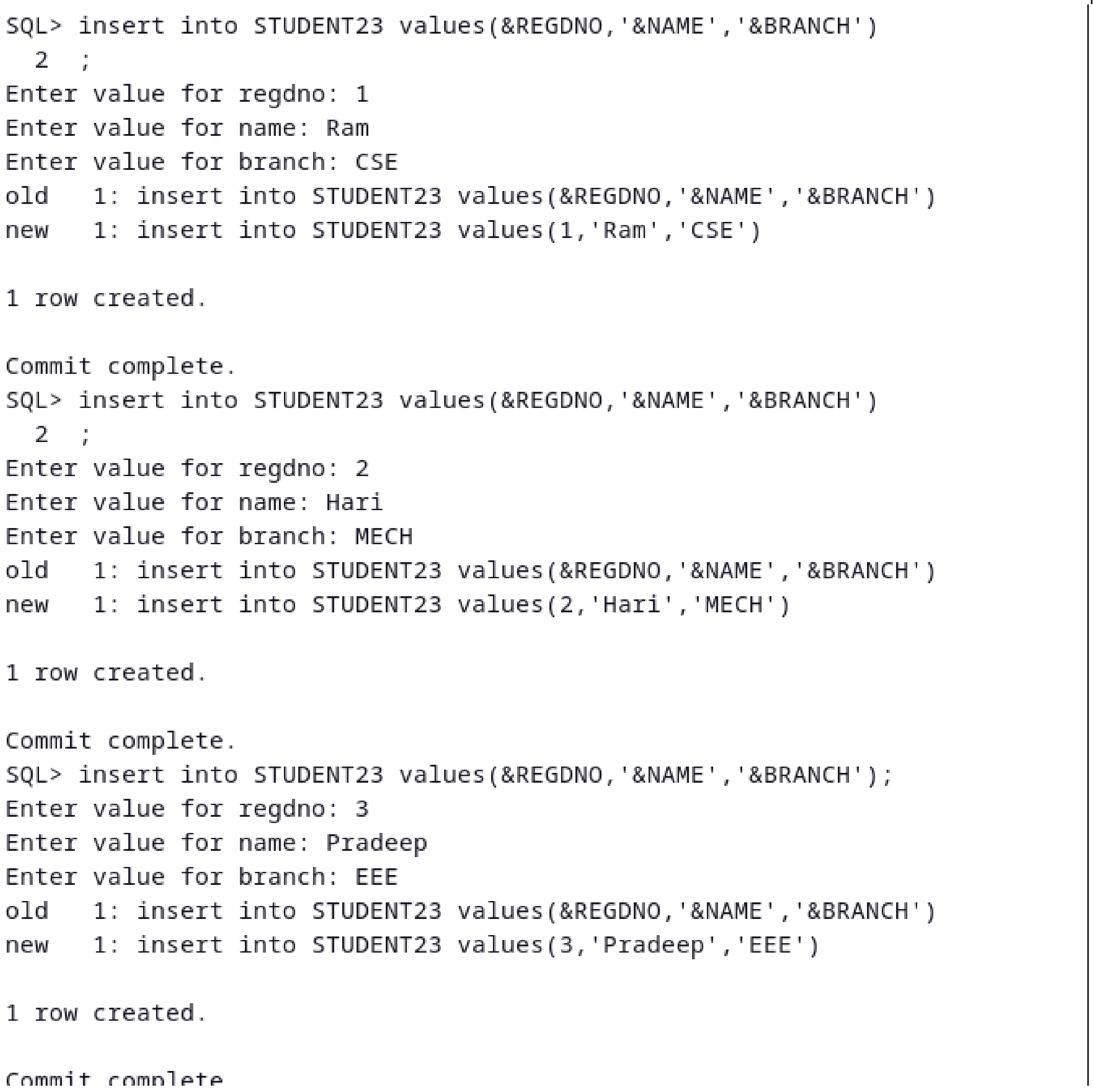


REGD.NO	NAME	BRANCH
0001	Ram	CSE
0002	Hari	MECH
0003	Pradeep	EEE
0004	Deepak	ETC

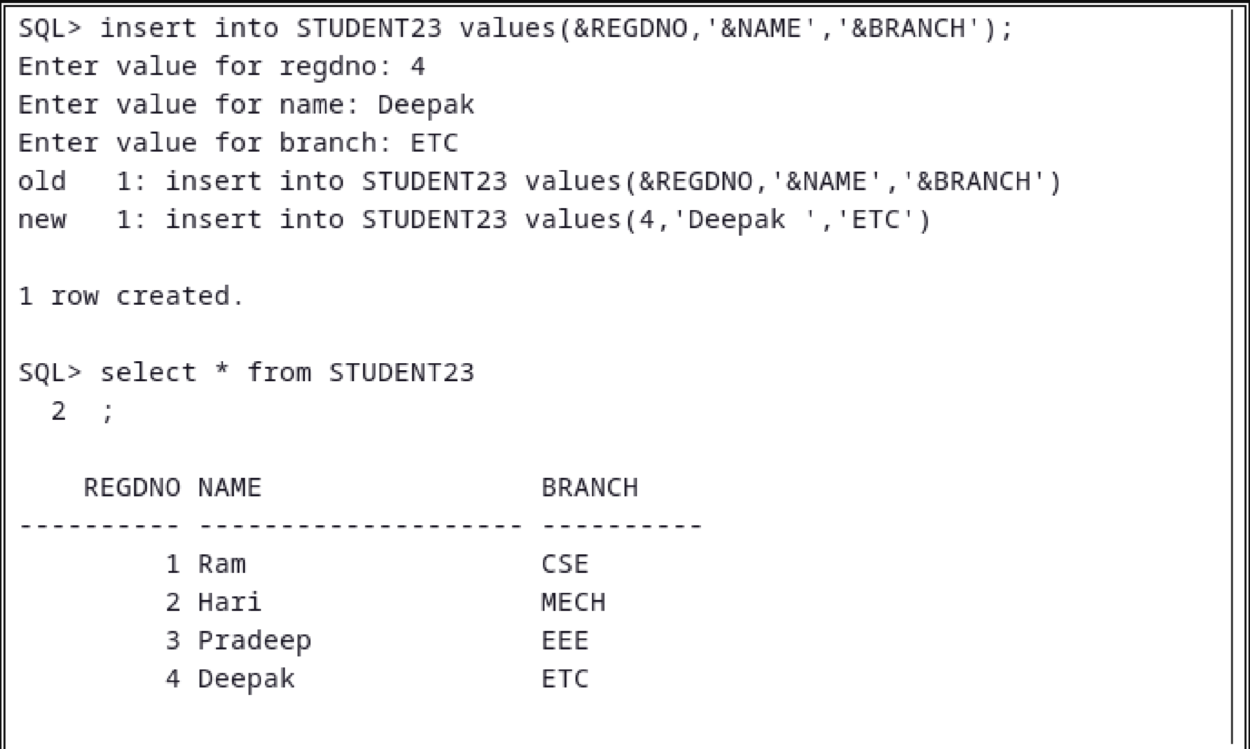
Output:



```
SQL> create table STUDENT23(REGDNO integer ,NAME varchar(20),BRANCH varcha
r(10));

Table created.
```

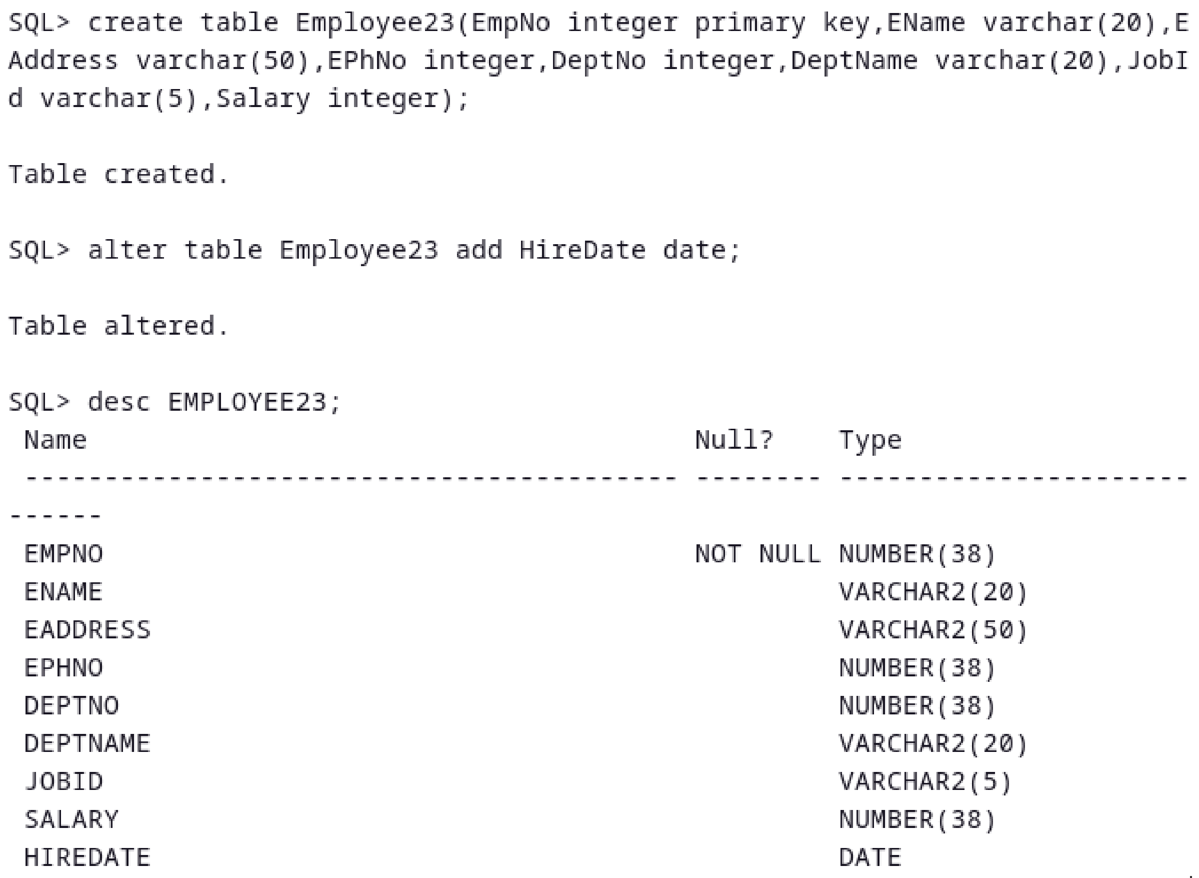


```
SQL> insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH')
  2  ;
Enter value for regdno: 1
Enter value for name: Ram
Enter value for branch: CSE
old   1: insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH')
new   1: insert into STUDENT23 values(1,'Ram','CSE')

1 row created.

Commit complete.
SQL> insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH')
  2  ;
Enter value for regdno: 2
Enter value for name: Hari
Enter value for branch: MECH
old   1: insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH')
new   1: insert into STUDENT23 values(2,'Hari','MECH')

1 row created.

Commit complete.
SQL> insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH');
Enter value for regdno: 3
Enter value for name: Pradeep
Enter value for branch: EEE
old   1: insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH')
new   1: insert into STUDENT23 values(3,'Pradeep','EEE')

1 row created.

Commit complete
```

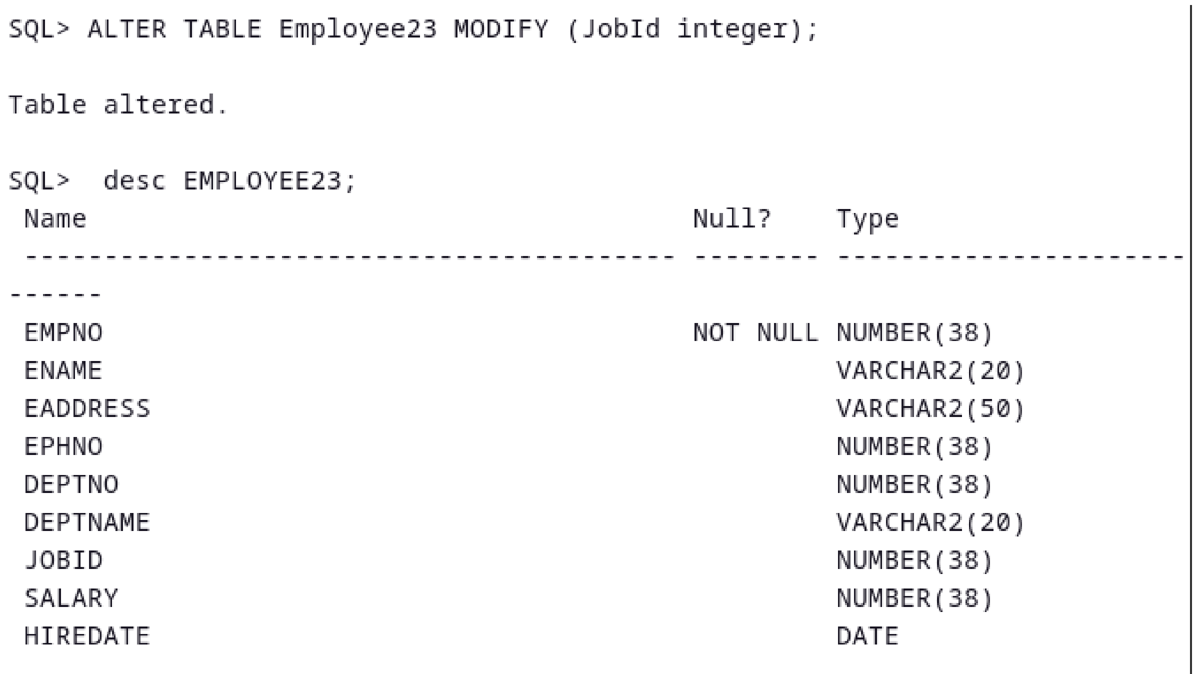


```
SQL> insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH');
Enter value for regdno: 4
Enter value for name: Deepak
Enter value for branch: ETC
old   1: insert into STUDENT23 values(&REGDNO,'&NAME','&BRANCH')
new   1: insert into STUDENT23 values(4,'Deepak ','ETC')

1 row created.

SQL> select * from STUDENT23
  2  ;

    REGDNO NAME                 BRANCH
---------- -------------------- ----------
         1 Ram                  CSE
         2 Hari                 MECH
         3 Pradeep              EEE
         4 Deepak               ETC
```

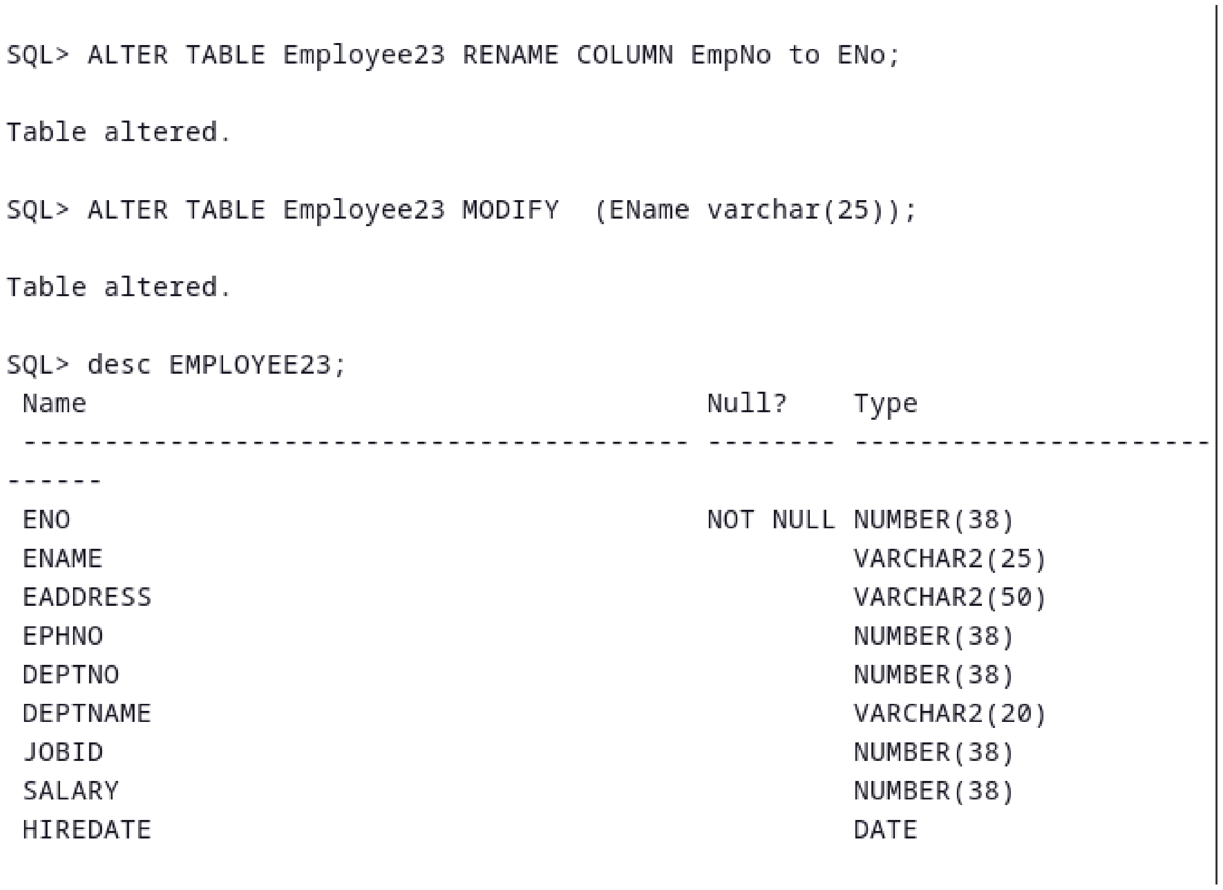
2. Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary) a) Add a new column; HIREDATE to the existing relation. b) Change the datatype of JOB_ID from varchar to integer. c) Change the name of column/field Emp_no to E_no. d). Modify the column width of the Employee name field of emp table.
Output:



```
SQL> create table Employee23(EmpNo integer primary key,EName varchar(20),E
Address varchar(50),EPhNo integer,DeptNo integer,DeptName varchar(20),JobI
d varchar(5),Salary integer);

Table created.

SQL> alter table Employee23 add HireDate date;

Table altered.

SQL> desc EMPLOYEE23;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
------
 EMPNO                                     NOT NULL NUMBER(38)
 ENAME                                              VARCHAR2(20)
 EADDRESS                                           VARCHAR2(50)
 EPHNO                                              NUMBER(38)
 DEPTNO                                             NUMBER(38)
 DEPTNAME                                           VARCHAR2(20)
 JOBID                                              VARCHAR2(5)
 SALARY                                             NUMBER(38)
 HIREDATE                                           DATE
```



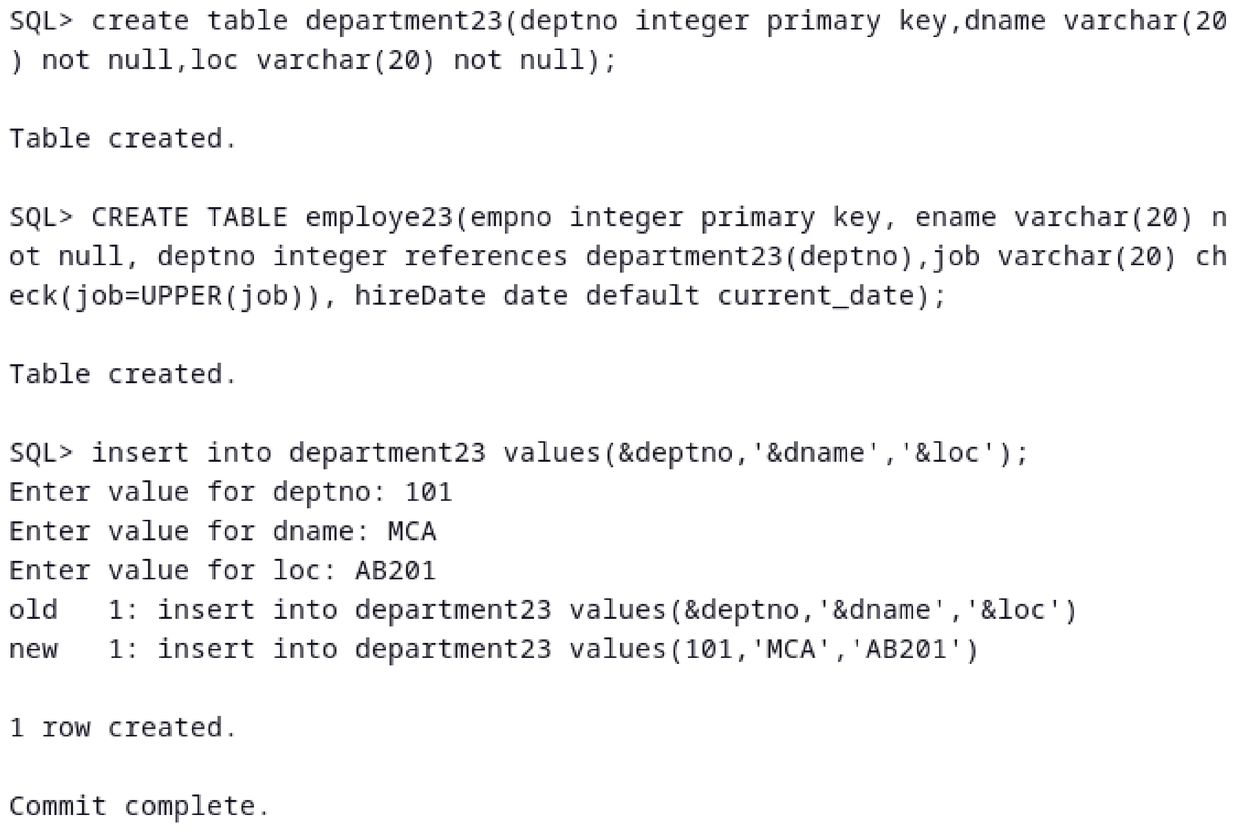
```
SQL> ALTER TABLE Employee23 MODIFY (JobId integer);

Table altered.

SQL>  desc EMPLOYEE23;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
------
 EMPNO                                     NOT NULL NUMBER(38)
 ENAME                                              VARCHAR2(20)
 EADDRESS                                           VARCHAR2(50)
 EPHNO                                              NUMBER(38)
 DEPTNO                                             NUMBER(38)
 DEPTNAME                                           VARCHAR2(20)
 JOBID                                              NUMBER(38)
 SALARY                                             NUMBER(38)
 HIREDATE                                           DATE
```



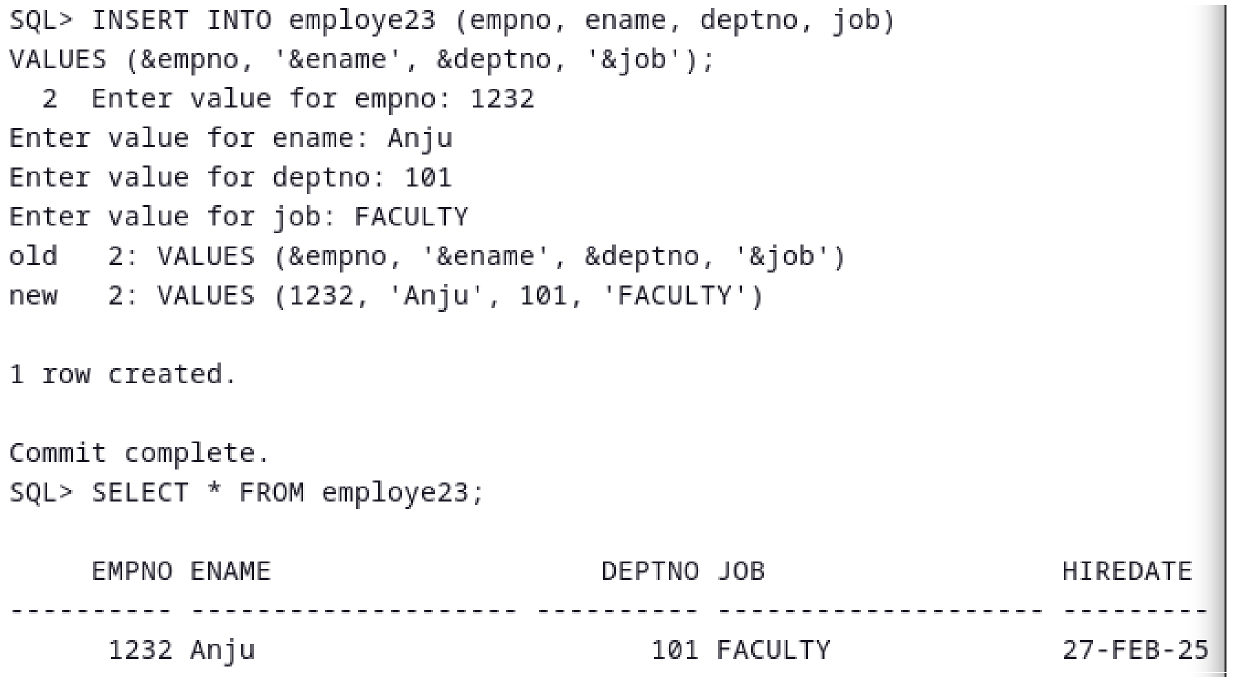
```
SQL> ALTER TABLE Employee23 RENAME COLUMN EmpNo to ENo;

Table altered.

SQL> ALTER TABLE Employee23 MODIFY  (EName varchar(25));

Table altered.

SQL> desc EMPLOYEE23;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
------
 ENO                                       NOT NULL NUMBER(38)
 ENAME                                              VARCHAR2(25)
 EADDRESS                                           VARCHAR2(50)
 EPHNO                                              NUMBER(38)
 DEPTNO                                             NUMBER(38)
 DEPTNAME                                           VARCHAR2(20)
 JOBID                                              NUMBER(38)
 SALARY                                             NUMBER(38)
 HIREDATE                                           DATE
```

3. Write a query in sql to create a table employee and department. Em ployee(empno, ename, deptno, job, hiredate) Department(deptno,dname,loc) Include the following constraints on column of emp table.
a) to make the empno as primary key of the table
b) to ensure that the ename column does not contain NULL values and
c) the job column to have only UPPERCASE entries
d) put the current date as default date in hire date column in case data is not supplied for the column.
Include the following constraints on column of Department table
a) to make deptno as primary key.
b) to ensure dname,loc coloumns does not contain NULL values
c)Also enforce REFERENTIAL INTEGRITY, declare deptno field of dept table as primary key and deptno field of emp table as foreign key.

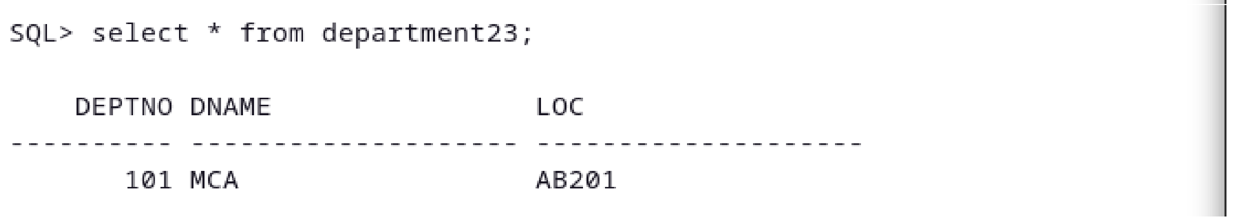
Output:



```
SQL> create table department23(deptno integer primary key,dname varchar(20
) not null,loc varchar(20) not null);

Table created.

SQL> CREATE TABLE employe23(empno integer primary key, ename varchar(20) n
ot null, deptno integer references department23(deptno),job varchar(20) ch
eck(job=UPPER(job)), hireDate date default current_date);

Table created.

SQL> insert into department23 values(&deptno,'&dname','&loc');
Enter value for deptno: 101
Enter value for dname: MCA
Enter value for loc: AB201
old   1: insert into department23 values(&deptno,'&dname','&loc')
new   1: insert into department23 values(101,'MCA','AB201')

1 row created.

Commit complete.
```



```
SQL> INSERT INTO employe23 (empno, ename, deptno, job)
VALUES (&empno, '&ename', &deptno, '&job');
  2  Enter value for empno: 1232
Enter value for ename: Anju
Enter value for deptno: 101
Enter value for job: FACULTY
old   2: VALUES (&empno, '&ename', &deptno, '&job')
new   2: VALUES (1232, 'Anju', 101, 'FACULTY')

1 row created.

Commit complete.
SQL> SELECT * FROM employe23;

     EMPNO ENAME                    DEPTNO JOB                 HIREDATE
---------- -------------------- ---------- ------------------- ---------
      1232 Anju                        101 FACULTY             27-FEB-25
```



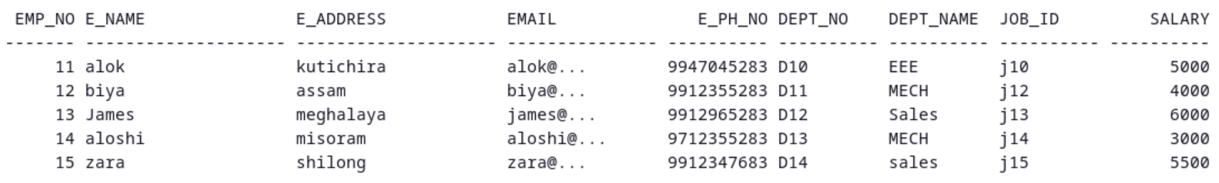
```
SQL> select * from department23;

    DEPTNO DNAME                LOC
---------- -------------------- --------------------
       101 MCA                  AB201
```

Experiment 2: Implementation of DML commands

4. Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary) Write SQL queries for following question:
1. Insert aleast 5 rows in the table.
2. Display all the information of EMP table.
3. Display the record of each employee who works in department D10.
4. Update the city of Emp_no-12 with current city as Nagpur.
5. Display the details of Employee who works in department MECH.
6. Delete the email_id of employee James.
7. Display the complete record of employees working in SALES Department.
8. Find out the employee id, names, salaries of all the employees
9.Find the names of the employees who have a salary greater than or equal to 4800

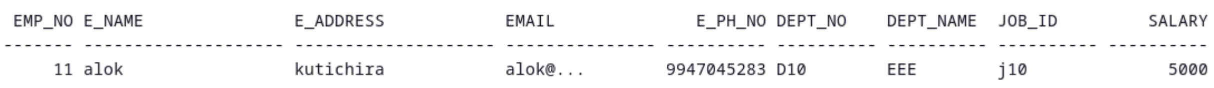
Output:

```
SQL> create table Emp23 (Emp_no integer primary key, E_name varchar(20), E_address varchar(20), Email varchar(15),E_ph_no integer, Dept_no varchar(10), Dept_name varchar(10),Job_id varchar(10), Salary integer);

Table created.

SQL> insert into Emp23 values(11, 'alok', 'kutichira','alok@...', 9947045283, 'D10', 'EEE','j10', 5000);
insert into Emp23 values(12, 'biya', 'assam','biya@...', 9912355283, 'D11', 'MECH','j12', 4000);
insert into Emp23 values(13, 'James', 'meghalaya','james@...', 9912965283, 'D12', 'Sales','j13', 6000);
insert into Emp23 values(14, 'aloshi', 'misoram','aloshi@...', 9712355283, 'D13', 'MECH','j14', 3000);
1 row created.
Commit complete.
SQL>
1 row created.
Commit complete.
SQL>
1 row created.
Commit complete.
SQL>
1 row created.
Commit complete.
SQL>insert into Emp23 values(15, 'zara', 'shilong','zara@...', 9912347683, 'D14', 'sales','j15', 5500);
1 row created.
Commit complete.

SQL> select * from Emp23;
```



```
 EMP_NO E_NAME              E_ADDRESS           EMAIL              E_PH_NO DEPT_NO    DEPT_NAME  JOB_ID         SALARY
------- ------------------- ------------------- -------------- ---------- ---------- ---------- ---------- ----------
     11 alok                kutichira           alok@...        9947045283 D10        EEE        j10              5000
     12 biya                assam               biya@...        9912355283 D11        MECH       j12              4000
     13 James               meghalaya           james@...       9912965283 D12        Sales      j13              6000
     14 aloshi              misoram             aloshi@...      9712355283 D13        MECH       j14              3000
     15 zara                shilong             zara@...        9912347683 D14        sales      j15              5500
```

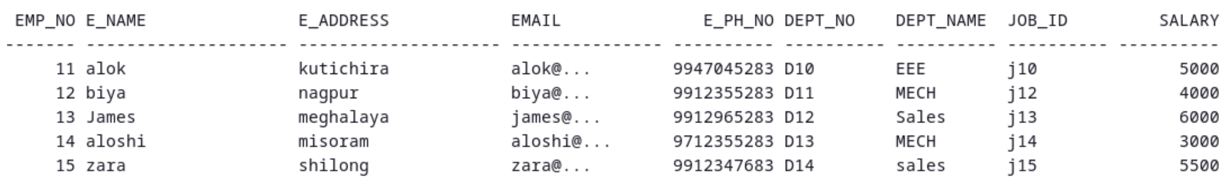
SQL> select * from Emp23 where Dept_no='D10';



```
 EMP_NO E_NAME              E_ADDRESS           EMAIL               E_PH_NO DEPT_NO    DEPT_NAME  JOB_ID         SALARY
------- ------------------- ------------------- -------------- ---------- ---------- ---------- ---------- ----------
     11 alok                kutichira           alok@...         9947045283 D10        EEE        j10              5000
```

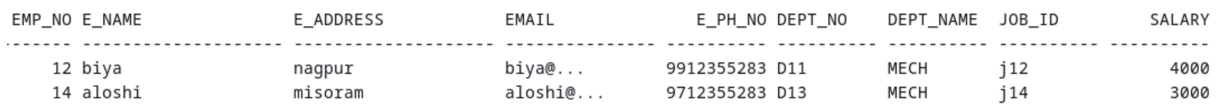
SQL> update Emp23 set E_address='nagpur' where Emp_no=12;

1 row updated.
Commit complete.
SQL> select * from Emp23;



```
 EMP_NO E_NAME              E_ADDRESS           EMAIL               E_PH_NO DEPT_NO    DEPT_NAME  JOB_ID         SALARY
------- ------------------- ------------------- -------------- ---------- ---------- ---------- ---------- ----------
     11 alok                kutichira           alok@...         9947045283 D10        EEE        j10              5000
     12 biya                nagpur              biya@...         9912355283 D11        MECH       j12              4000
     13 James               meghalaya           james@...        9912965283 D12        Sales      j13              6000
     14 aloshi              misoram             aloshi@...       9712355283 D13        MECH       j14              3000
     15 zara                shilong             zara@...         9912347683 D14        sales      j15              5500
```

select * from Emp23 where Dept_name='MECH';

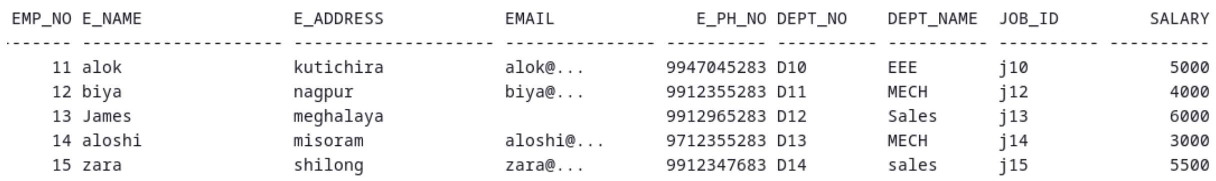


```
EMP_NO E_NAME              E_ADDRESS           EMAIL               E_PH_NO DEPT_NO    DEPT_NAME  JOB_ID         SALARY
------ ------------------- ------------------- -------------- ---------- ---------- ---------- ---------- ----------
    12 biya                nagpur              biya@...         9912355283 D11        MECH       j12              4000
    14 aloshi              misoram             aloshi@...       9712355283 D13        MECH       j14              3000
```

SQL> update Emp23 SET Email = NULL where E_name = 'James';

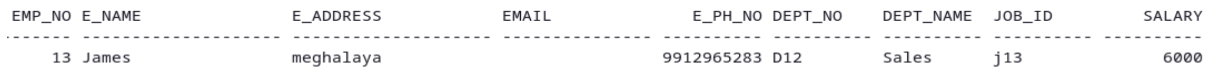
1 row updated.

Commit complete.
SQL> select * from Emp23;



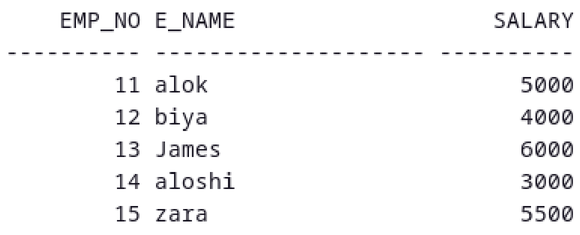
```
EMP_NO E_NAME              E_ADDRESS           EMAIL               E_PH_NO DEPT_NO    DEPT_NAME  JOB_ID         SALARY
------ ------------------- ------------------- -------------- ---------- ---------- ---------- ---------- ----------
    11 alok                kutichira           alok@...         9947045283 D10        EEE        j10              5000
    12 biya                nagpur              biya@...         9912355283 D11        MECH       j12              4000
    13 James               meghalaya                            9912965283 D12        Sales      j13              6000
    14 aloshi              misoram             aloshi@...       9712355283 D13        MECH       j14              3000
    15 zara                shilong             zara@...         9912347683 D14        sales      j15              5500
```

SQL> select * from Emp23 where Dept_name='Sales';



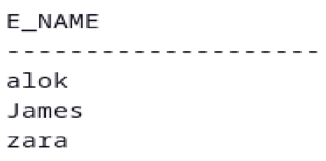
```
EMP_NO E_NAME              E_ADDRESS           EMAIL               E_PH_NO DEPT_NO    DEPT_NAME  JOB_ID         SALARY
------ ------------------- ------------------- -------------- ---------- ---------- ---------- ---------- ----------
    13 James               meghalaya                            9912965283 D12        Sales      j13              6000
```

SQL> select Emp_no,E_name,Salary from Emp23;



```
    EMP_NO E_NAME                  SALARY
---------- ------------------- ----------
        11 alok                      5000
        12 biya                      4000
        13 James                     6000
        14 aloshi                    3000
        15 zara                      5500
```

SQL> select E_name from Emp23 where Salary>=4800;

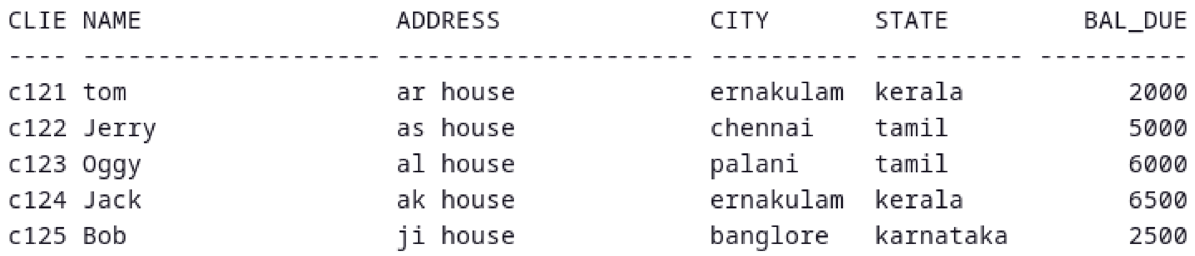


```
E_NAME
--------------------
alok
James
zara
```

5. (Exercise on updating records in table) Create Client_master with the following fields(ClientNO, Name, Address, City, State, bal_due)
a. Insert five records
b. Find the names of clients whose bal_due> 5000 .
c. Change the bal_due of ClientNO " C123" to Rs. 5100
d. Change the name of Client_master to Client12 .
e. Display the bal_due heading as "BALANCE"

Output:

```
SQL> create table client_master23(ClientNO varchar(4),Name varchar(20), Address
varchar(20),City varchar(10),State varchar(10),bal_due integer);

Table created.

insert into client_master23 values('c121','tom','ar house','ernakulam','kerala',2000);
insert into client_master23 values('c122','Jerry','as house','chennai','tamil',5000);
insert into client_master23 values('c123','Oggy','al house','palani','tamil',6000);
insert into client_master23 values('c124','Jack','ak house','ernakulam','kerala',6500);
insert into client_master23 values('c125','Bob','ji house','banglore','karnataka',2500);

1 row created.

Commit complete.
SQL>
1 row created.

Commit complete.
SQL>
1 row created.

Commit complete.
1 row created.

Commit complete.
SQL>
1 row created.

Commit complete.

select * from client_master23;
```



```
CLIE NAME                 ADDRESS              CITY       STATE         BAL_DUE
---- -------------------- -------------------- ---------- ---------- ----------
c121 tom                  ar house             ernakulam  kerala           2000
c122 Jerry                as house             chennai    tamil            5000
c123 Oggy                 al house             palani     tamil            6000
c124 Jack                 ak house             ernakulam  kerala           6500
c125 Bob                  ji house             banglore   karnataka        2500
```

select name from client_master23 where bal_due>5000;

```
NAME
--------------------
Oggy
Jack
```

SQL> update client_master23 set bal_due=5100 where ClientNo='c123';

1 row updated.

Commit complete.

select * from client_master23;

```
CLIE NAME                 ADDRESS              CITY       STATE         BAL_DUE
---- -------------------- -------------------- ---------- ---------- ----------
c121 tom                  ar house             ernakulam  kerala           2000
c122 Jerry                as house             chennai    tamil            5000
c123 Oggy                 al house             palani     tamil            5100
c124 Jack                 ak house             ernakulam  kerala           6500
c125 Bob                  ji house             banglore   karnataka        2500
```

SQL> alter table client_master23 rename to Client1223;

Table altered.

SQL> select bal_due as BALANCE from client1223;

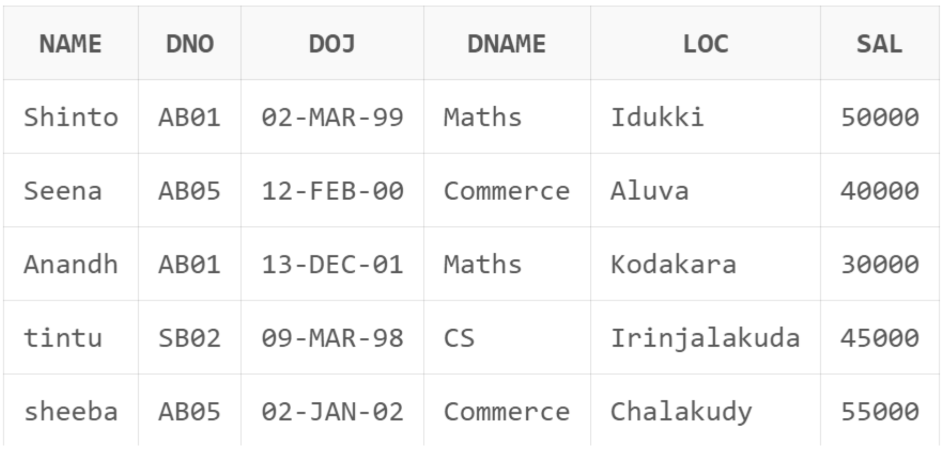
```
   BALANCE
----------
      2000
      5000
      5100
      6500
      2500
```

6. (Rollback and Commit commands) Create Teacher table with the following fields(Name, DeptNo, Date of joining, DeptName, Location, Salary)
a. Insert five records
b. Give Increment of 25% salary for Mathematics Department .
c. Perform Rollback command
d. Give Increment of 15% salary for Commerce Department
e. Perform commit command

Output:

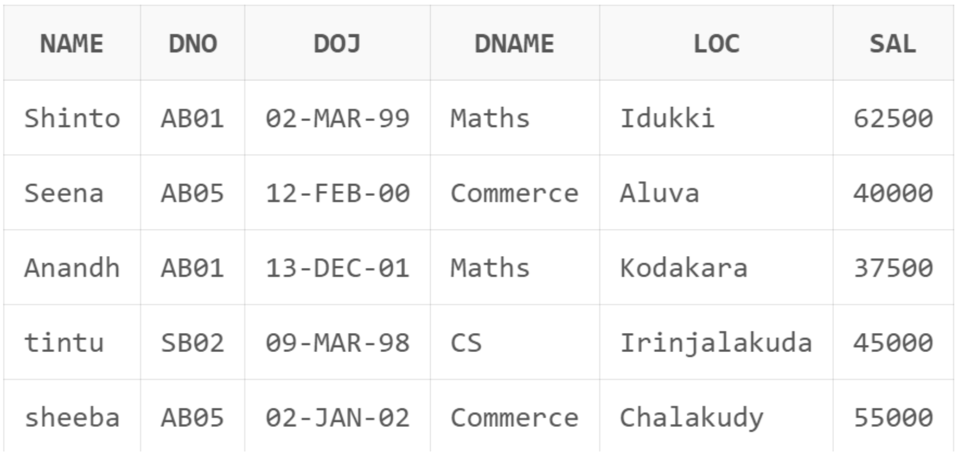
```
Create table teacher23(name varchar(20),dno varchar(10),doj date,dname varchar(10),loc varchar(20),sal integer);
```

```
a)insert into teacher23 values('Shinto','AB01','02-mar-1999','Maths','Idukki',50000);
insert into teacher23 values('Seena','AB05','12-feb-2000','Commerce','Aluva',40000);
insert into teacher23 values('Anandh','AB01','13-dec-2001','Maths','Kodakara',30000);
insert into teacher23 values('tintu','SB02','09-mar-1998','CS','Irinjalakuda',45000);
insert into teacher23 values('sheeba','AB05','02-jan-2002','Commerce','Chalakudy',55000);
select * from teacher23;
```



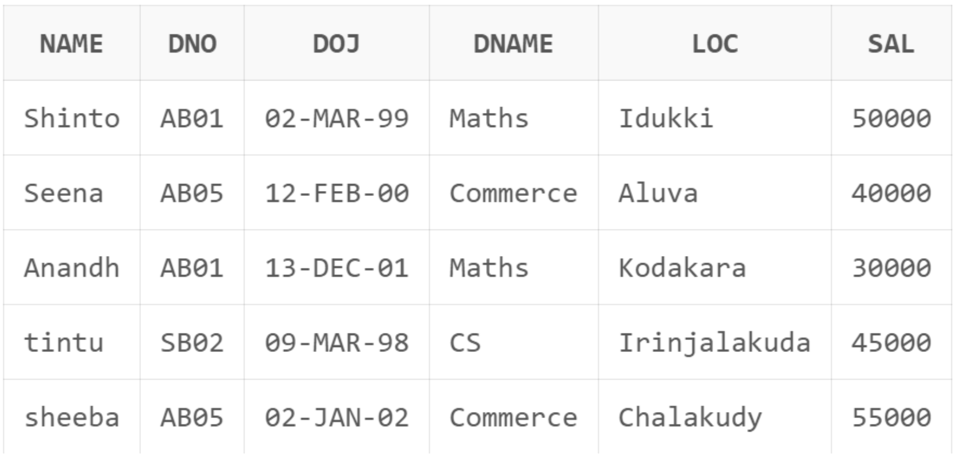
NAME	DNO	DOJ	DNAME	LOC	SAL
Shinto	AB01	02-MAR-99	Maths	Idukki	50000
Seena	AB05	12-FEB-00	Commerce	Aluva	40000
Anandh	AB01	13-DEC-01	Maths	Kodakara	30000
tintu	SB02	09-MAR-98	CS	Irinjalakuda	45000
sheeba	AB05	02-JAN-02	Commerce	Chalakudy	55000

```
b)update teacher23 set sal=sal+(sal*0.25) where dname='Maths';
  select * from teacher23;
```



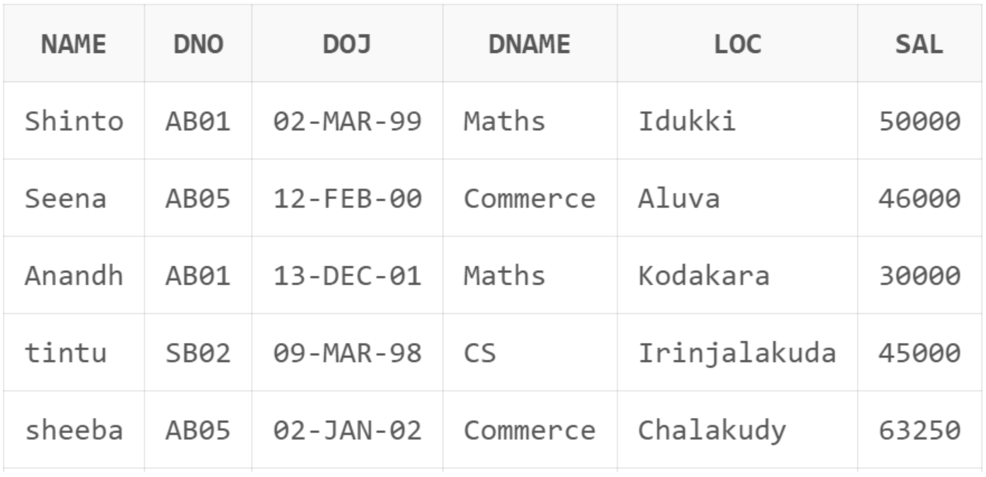
NAME	DNO	DOJ	DNAME	LOC	SAL
Shinto	AB01	02-MAR-99	Maths	Idukki	62500
Seena	AB05	12-FEB-00	Commerce	Aluva	40000
Anandh	AB01	13-DEC-01	Maths	Kodakara	37500
tintu	SB02	09-MAR-98	CS	Irinjalakuda	45000
sheeba	AB05	02-JAN-02	Commerce	Chalakudy	55000

```
c)rollback;
```



NAME	DNO	DOJ	DNAME	LOC	SAL
Shinto	AB01	02-MAR-99	Maths	Idukki	50000
Seena	AB05	12-FEB-00	Commerce	Aluva	40000
Anandh	AB01	13-DEC-01	Maths	Kodakara	30000
tintu	SB02	09-MAR-98	CS	Irinjalakuda	45000
sheeba	AB05	02-JAN-02	Commerce	Chalakudy	55000

d)update teacher23 set sal=sal+(sal*0.15) where dname='Commerce';
select * from teacher23;



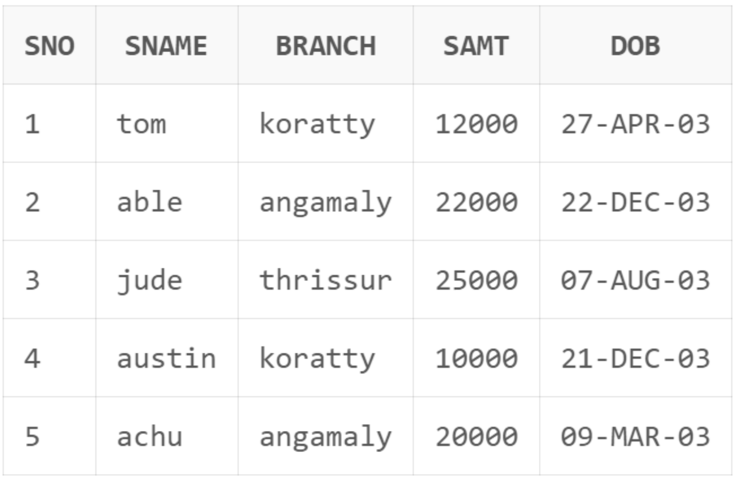
NAME	DNO	DOJ	DNAME	LOC	SAL
Shinto	AB01	02-MAR-99	Maths	Idukki	50000
Seena	AB05	12-FEB-00	Commerce	Aluva	46000
Anandh	AB01	13-DEC-01	Maths	Kodakara	30000
tintu	SB02	09-MAR-98	CS	Irinjalakuda	45000
sheeba	AB05	02-JAN-02	Commerce	Chalakudy	63250

e)commit;

7.(Exercise on order by and group by clauses) Create Sales table with the following fields(Sales No, Salesname, Branch, Salesamount, DOB)
a. Insert five records
b. Calculate total salesamount in each branch
c. Calculate average salesamount in each branch .
d. Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09
e. Display the name and DOB of salesman in alphabetical order of the month.

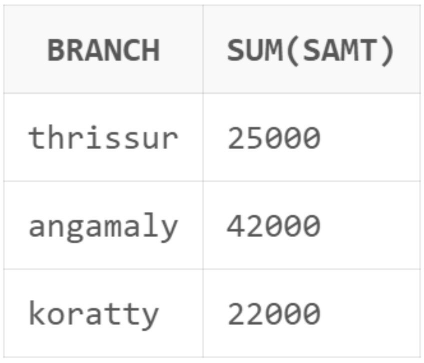
Output:

```
create table Sales23 (SNo integer primary key,Sname varchar(10),Branch varchar(10),Samt
integer, DOB date);

a)insert into Sales23 values(1,'tom', 'koratty',12000,'27-apr-2003');
  insert into Sales23 values(2,'able', 'angamaly',22000,'22-dec-2003');
  insert into Sales23 values(3,'jude', 'thrissur',25000,'07-aug-2003');
  insert into Sales23 values(4,'austin', 'koratty',10000,'21-dec-2003');
  insert into Sales23 values(5,'achu', 'idukki',20000,'09-mar-2003');
  select * from sales23;
```



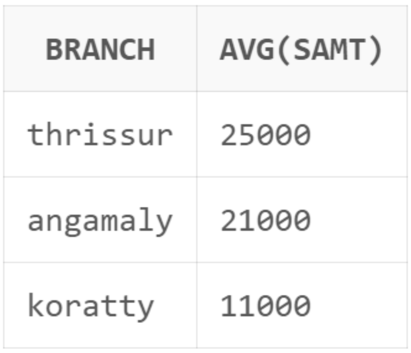
SNO	SNAME	BRANCH	SAMT	DOB
1	tom	koratty	12000	27-APR-03
2	able	angamaly	22000	22-DEC-03
3	jude	thrissur	25000	07-AUG-03
4	austin	koratty	10000	21-DEC-03
5	achu	angamaly	20000	09-MAR-03

```
b)select Branch ,sum(samt) from Sales23 group by Branch;
```



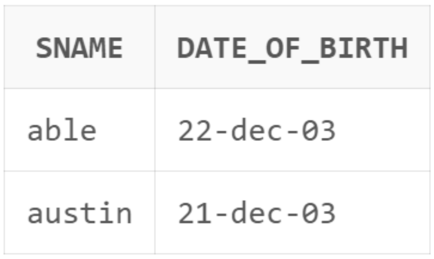
BRANCH	SUM(SAMT)
thrissur	25000
angamaly	42000
koratty	22000

c)select Branch ,avg(samt) from Sales23 group by Branch;



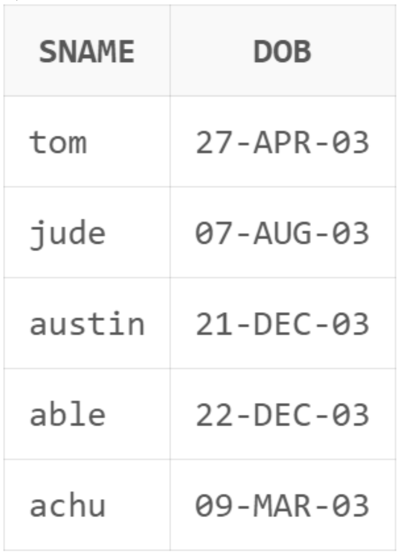
BRANCH	AVG(SAMT)
thrissur	25000
angamaly	21000
koratty	11000

d)Select sname,to_char(dob,'dd-mon-yy') as date_of_birth from sales23 where to_char(dob,'mm')=12;



SNAME	DATE_OF_BIRTH
able	22-dec-03
austin	21-dec-03

e)Select sname,dob from sales23 order by to_char(dob, 'mon');



SNAME	DOB
tom	27-APR-03
jude	07-AUG-03
austin	21-DEC-03
able	22-DEC-03
achu	09-MAR-03

Experiment 3 :Implementation of different types of operators in SQL

8.Create an Emp table with the following fields:(EmpNo, EmpName, Job, Basic, DA, HRA,PF, GrossPay, NetPay) Hint:(PF is calculated as 10% of basic salary) (Calculate DA as 30% of Basic and HRA as 40% of Basic)

a. Insert Five Records and calculate GrossPay and NetPay.

b. Display the employees whose Basic is lowest in each department .

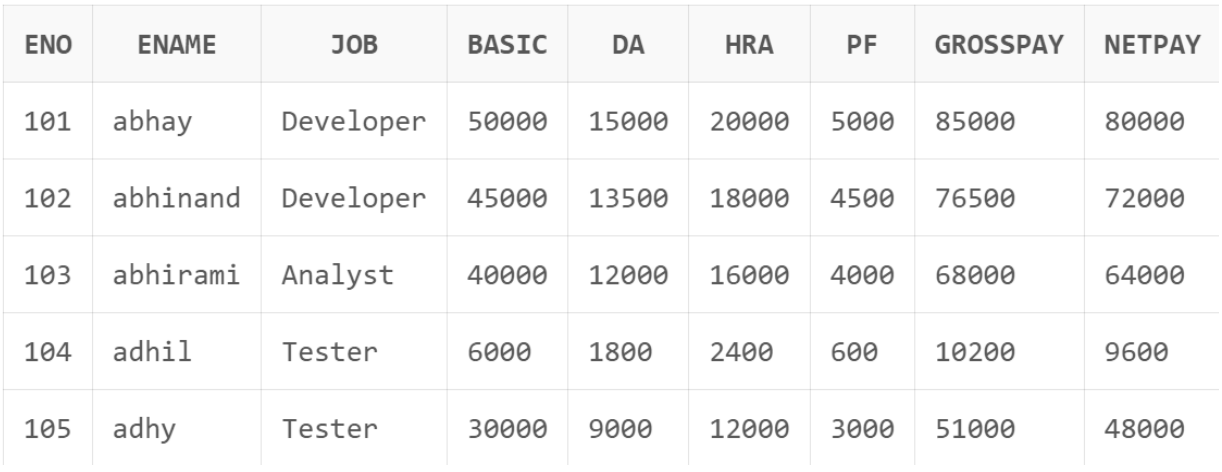
c.If NetPay is less than <Rs. 10,000 add Rs. 1200 as special allowances .

d. Display the employees whose GrossPay lies between 10,000 & 20,000

e. Display all the employees who earn maximum salary .

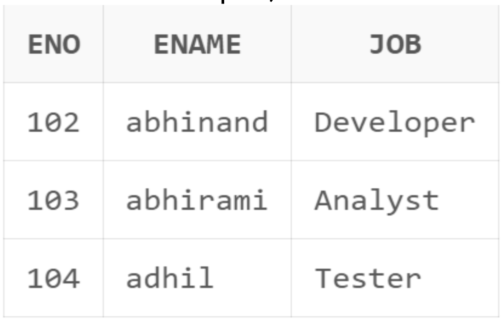
Output:

```
Create table emp23(eno integer,ename varchar(20),job varchar(10),basic decimal(10,2),DA
decimal(10,2),HRA decimal(10,2),PF decimal(10,2),grosspay decimal(10,2),netpay
decimal(10,2));

a)insert into emp23 values(101, 'abhay', 'Developer', 50000, 50000*0.3, 50000*0.4, 50000*0.1,
50000 + 50000*0.3 + 50000*0.4, (50000 + 50000*0.3 + 50000*0.4) - 50000*0.1);
insert into emp23 values(102, 'abhinand', 'Developer', 45000, 45000*0.3, 45000*0.4,
45000*0.1, 45000 + 45000*0.3 + 45000*0.4, (45000 + 45000*0.3 + 45000*0.4) - 45000*0.1);
insert into emp23 values(103, 'abhirami', 'Analyst', 40000, 40000*0.3, 40000*0.4, 40000*0.1,
40000 + 40000*0.3 + 40000*0.4, (40000 + 40000*0.3 + 40000*0.4) - 40000*0.1);
insert into emp23 values(104, 'adhil', 'Tester', 6500, 6500*0.3, 6500*0.4, 6500*0.1, 6500+
6500*0.3 + 6500*0.4, (6500 + 6500*0.3 + 6500*0.4) - 6500*0.1);
insert into emp23 values(105, 'adhy', 'Tester', 30000, 30000*0.3, 30000*0.4, 30000*0.1, 30000
+ 30000*0.3 + 30000*0.4, (30000 + 30000*0.3 + 30000*0.4) - 30000*0.1);
select * from emp23;
```



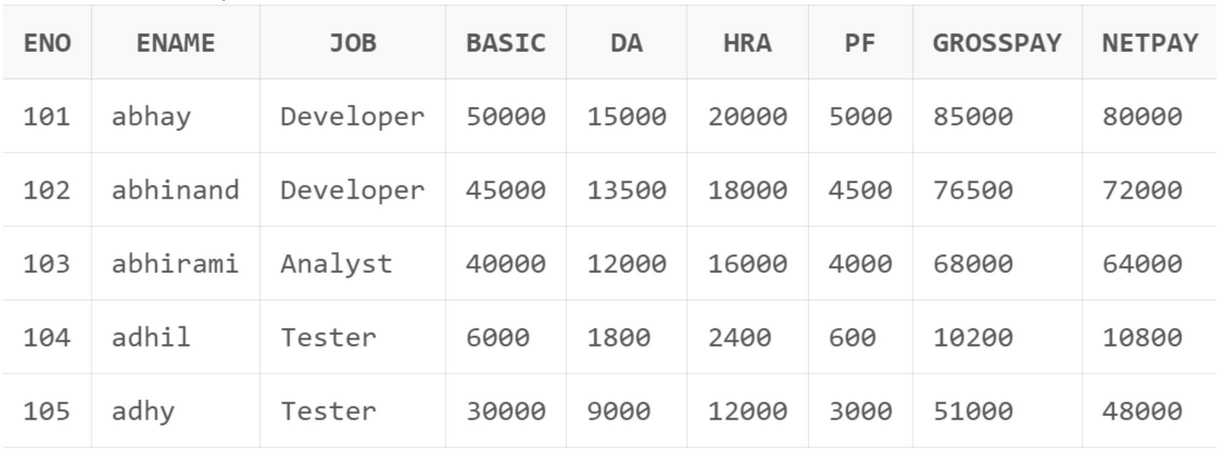
ENO	ENAME	JOB	BASIC	DA	HRA	PF	GROSSPAY	NETPAY
101	abhay	Developer	50000	15000	20000	5000	85000	80000
102	abhinand	Developer	45000	13500	18000	4500	76500	72000
103	abhirami	Analyst	40000	12000	16000	4000	68000	64000
104	adhil	Tester	6000	1800	2400	600	10200	9600
105	adhy	Tester	30000	9000	12000	3000	51000	48000

b)select eno,ename,job from emp23 where (job,basic) in (select job,min(basic) from emp23 group by job);
select * from emp23;



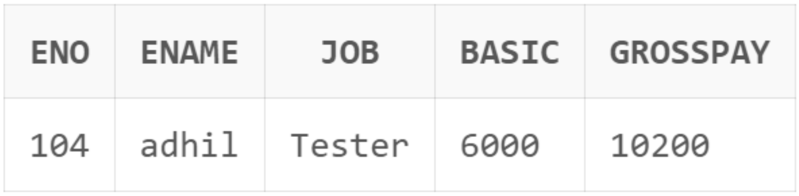
ENO	ENAME	JOB
102	abhinand	Developer
103	abhirami	Analyst
104	adhil	Tester

c)update emp23 set netpay= netpay + 1200 where netpay<10000;
select * from emp23;



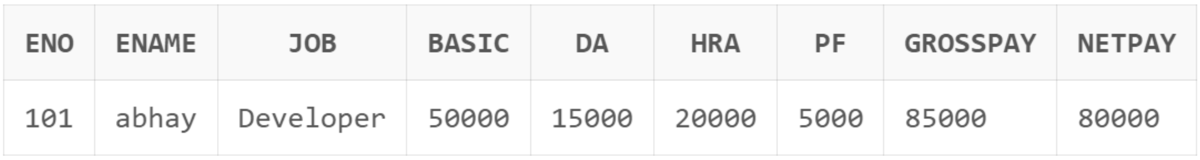
ENO	ENAME	JOB	BASIC	DA	HRA	PF	GROSSPAY	NETPAY
101	abhay	Developer	50000	15000	20000	5000	85000	80000
102	abhinand	Developer	45000	13500	18000	4500	76500	72000
103	abhirami	Analyst	40000	12000	16000	4000	68000	64000
104	adhil	Tester	6000	1800	2400	600	10200	10800
105	adhy	Tester	30000	9000	12000	3000	51000	48000

d)select eno,ename,job,basic,grosspay from emp23 where grosspay between 10000 and 20000;



ENO	ENAME	JOB	BASIC	GROSSPAY
104	adhil	Tester	6000	10200

e)select * from emp23 where netpay = (select max(netpay) from emp23);



ENO	ENAME	JOB	BASIC	DA	HRA	PF	GROSSPAY	NETPAY
101	abhay	Developer	50000	15000	20000	5000	85000	80000

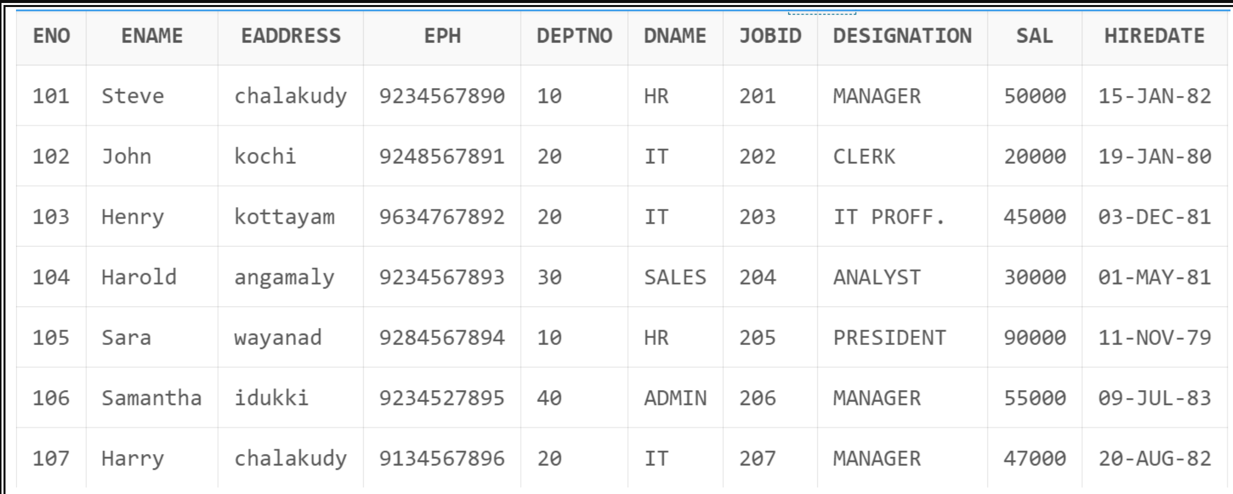
Experiment 4: Implementation of different types of functions with suitable examples

9.Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation , Salary) Write SQL statements for the following query.

1. List the E_no, E_name, Salary of all employees working for MANAGER.
2. Display all the details of the employee whose salary is more than the Sal of any IT PROFF..
3. List the employees in the ascending order of Designations of those joined after 1981.
4. List the employees along with their Experience and Daily Salary.
5. List the employees who are either 'CLERK' or 'ANALYST' .
6. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81,19-JAN-80 .
7. List the employees who are working for the Deptno 10 or20.
8. List the Enames those are starting with 'S' .
9. Dislay the name as well as the first five characters of name(s) starting with 'H'
10. List all the emps except 'PRESIDENT' & 'MANAGR" in asc order of Salaries.

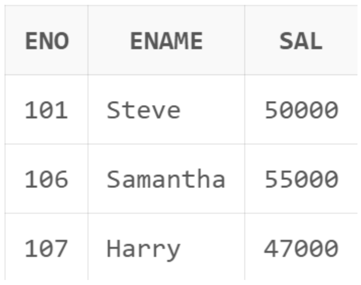
Output:

```
create table emp23 (eno integer,ename varchar(20),eaddress varchar(20),eph varchar(10),deptno integer,dname varchar(10),jobid integer,designation varchar(10),sal integer,hiredate date);

insert into emp23 values (101, 'steve', 'chalakudy', '9234567890', 10, 'hr', 201, 'manager', 50000,to_date('1982-01-15', 'yyyy-mm-dd'));

insert into emp23 values (102, 'john', 'kochi', '9248567891', 20, 'it', 202, 'clerk', 20000,to_date('1980-01-19', 'yyyy-mm-dd'));

insert into emp23 values (103, 'henry', 'kottayam', '9634767892', 20, 'it', 203, 'it proff.', 45000,to_date('1981-12-03', 'yyyy-mm-dd'));

insert into emp23 values (104, 'harold', 'angamaly', '9234567893', 30, 'sales', 204, 'analyst', 30000,to_date('1981-05-01', 'yyyy-mm-dd'));

insert into emp23 values (105, 'sara', 'wayanad', '9284567894', 10, 'hr', 205, 'president', 90000, to_date('1979-11-11', 'yyyy-mm-dd'));

insert into emp23 values (106, 'samantha', 'idukki', '9234527895', 40, 'admin', 206, 'manager', 55000,to_date('1983-07-09', 'yyyy-mm-dd'));

insert into emp23 values (107, 'harry', 'chalakudy', '9134567896', 20, 'it', 207, 'manager', 47000,to_date('1982-08-20', 'yyyy-mm-dd'));

select * from emp23;
```



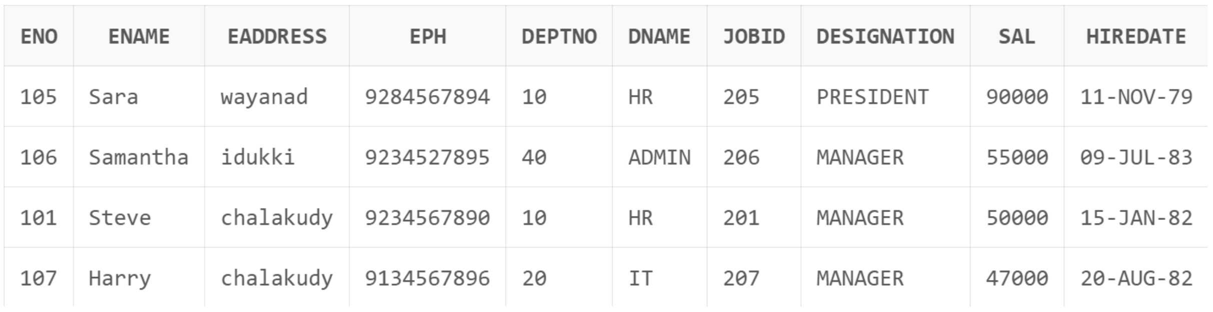
ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
101	Steve	chalakudy	9234567890	10	HR	201	MANAGER	50000	15-JAN-82
102	John	kochi	9248567891	20	IT	202	CLERK	20000	19-JAN-80
103	Henry	kottayam	9634767892	20	IT	203	IT PROFF.	45000	03-DEC-81
104	Harold	angamaly	9234567893	30	SALES	204	ANALYST	30000	01-MAY-81
105	Sara	wayanad	9284567894	10	HR	205	PRESIDENT	90000	11-NOV-79
106	Samantha	idukki	9234527895	40	ADMIN	206	MANAGER	55000	09-JUL-83
107	Harry	chalakudy	9134567896	20	IT	207	MANAGER	47000	20-AUG-82

1)select eno,ename,sal from emp23 where designation='MANAGER';



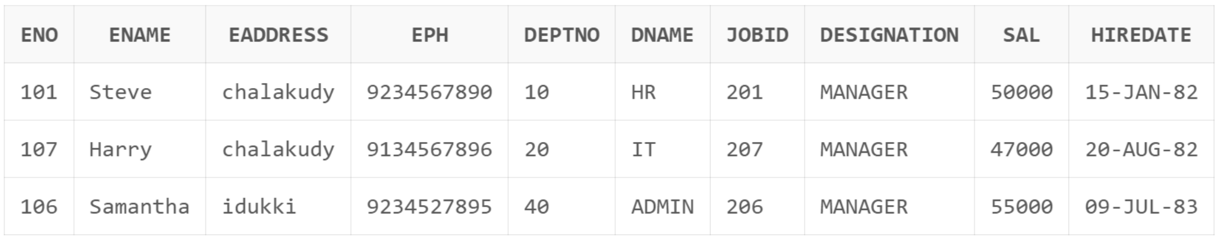
ENO	ENAME	SAL
101	Steve	50000
106	Samantha	55000
107	Harry	47000

2)select * from emp23 where sal>any(select sal from emp23 where designation='IT PROFF.');



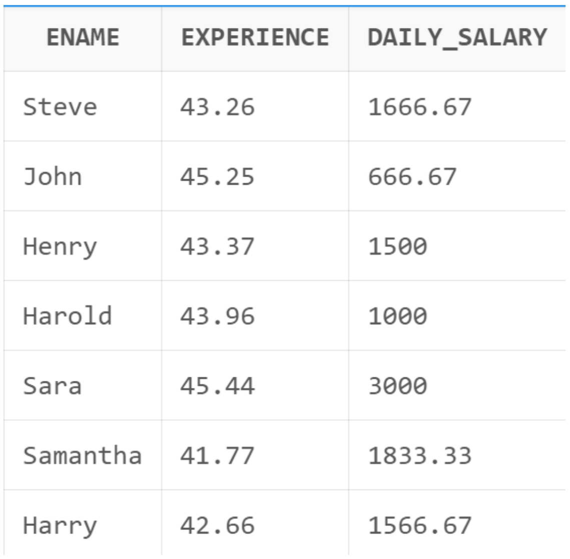
ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
105	Sara	wayanad	9284567894	10	HR	205	PRESIDENT	90000	11-NOV-79
106	Samantha	idukki	9234527895	40	ADMIN	206	MANAGER	55000	09-JUL-83
101	Steve	chalakudy	9234567890	10	HR	201	MANAGER	50000	15-JAN-82
107	Harry	chalakudy	9134567896	20	IT	207	MANAGER	47000	20-AUG-82

3)select * from emp23 where hiredate > to_date('1981-12-31','yyyy-mm-dd') order by designation asc;



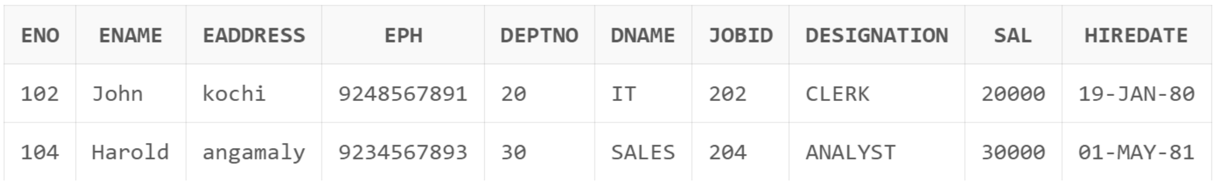
ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
101	Steve	chalakudy	9234567890	10	HR	201	MANAGER	50000	15-JAN-82
107	Harry	chalakudy	9134567896	20	IT	207	MANAGER	47000	20-AUG-82
106	Samantha	idukki	9234527895	40	ADMIN	206	MANAGER	55000	09-JUL-83

4)select ename,round(months_between(sysdate,hiredate)/12,2) as experience , round(sal/30,2) as daily_salary from emp23;



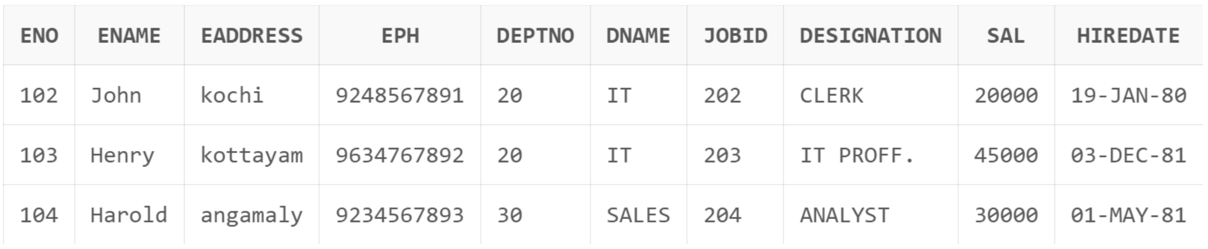
ENAME	EXPERIENCE	DAILY_SALARY
Steve	43.26	1666.67
John	45.25	666.67
Henry	43.37	1500
Harold	43.96	1000
Sara	45.44	3000
Samantha	41.77	1833.33
Harry	42.66	1566.67

5)select * from emp23 where designation in ('CLERK','ANALYST');



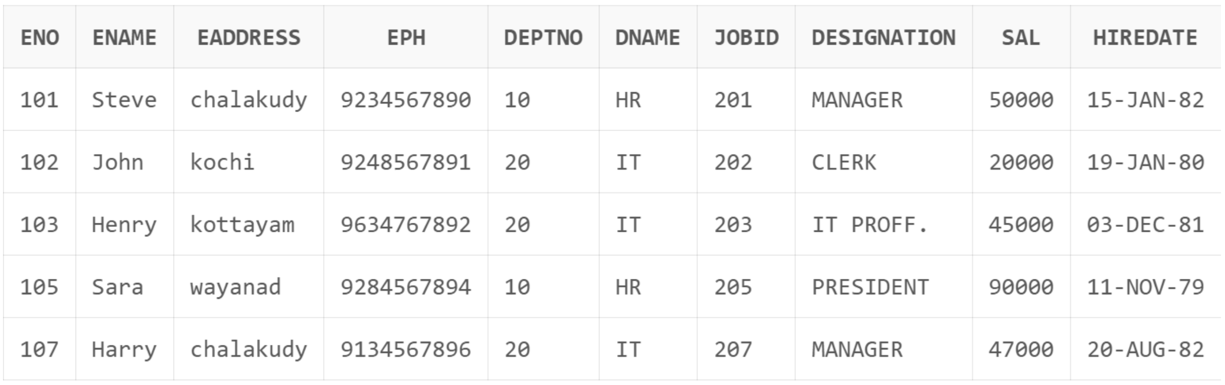
ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
102	John	kochi	9248567891	20	IT	202	CLERK	20000	19-JAN-80
104	Harold	angamaly	9234567893	30	SALES	204	ANALYST	30000	01-MAY-81

6)select * from emp23 where hiredate in ('1-MAY-81', '3-DEC-81', '17-DEC-81','19-JAN-80');



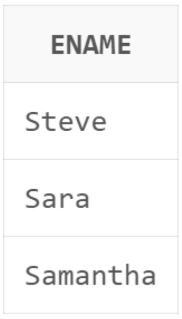
ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
102	John	kochi	9248567891	20	IT	202	CLERK	20000	19-JAN-80
103	Henry	kottayam	9634767892	20	IT	203	IT PROFF.	45000	03-DEC-81
104	Harold	angamaly	9234567893	30	SALES	204	ANALYST	30000	01-MAY-81

7)select * from emp23 where deptno in(10,20);



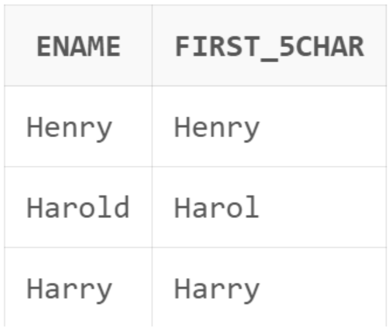
ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
101	Steve	chalakudy	9234567890	10	HR	201	MANAGER	50000	15-JAN-82
102	John	kochi	9248567891	20	IT	202	CLERK	20000	19-JAN-80
103	Henry	kottayam	9634767892	20	IT	203	IT PROFF.	45000	03-DEC-81
105	Sara	wayanad	9284567894	10	HR	205	PRESIDENT	90000	11-NOV-79
107	Harry	chalakudy	9134567896	20	IT	207	MANAGER	47000	20-AUG-82

8)select ename from emp23 where ename like 'S%';



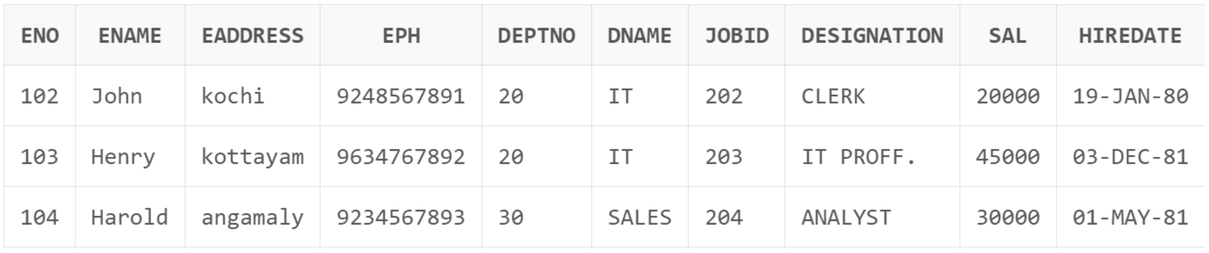
ENAME
Steve
Sara
Samantha

9)select ename,substr(ename,1,5)as first_5char from emp23 where ename like 'H%';



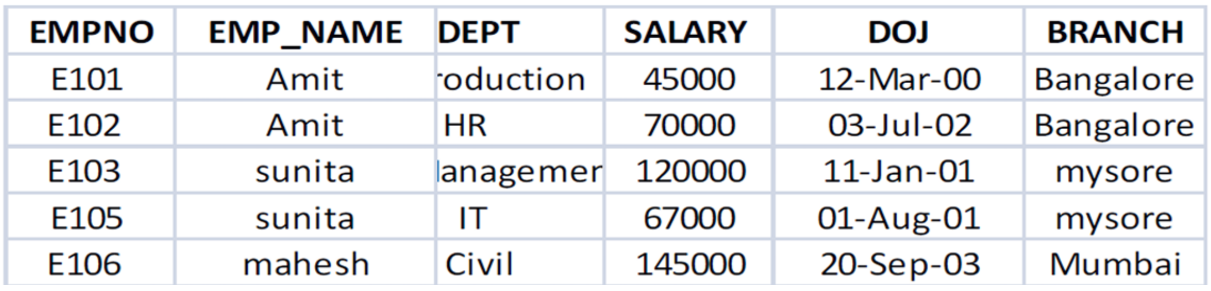
ENAME	FIRST_5CHAR
Henry	Henry
Harold	Harol
Harry	Harry

10)select * from emp23 where designation not in ('PRESIDENT','MANAGER');



ENO	ENAME	EADDRESS	EPH	DEPTNO	DNAME	JOBID	DESIGNATION	SAL	HIREDATE
102	John	kochi	9248567891	20	IT	202	CLERK	20000	19-JAN-80
103	Henry	kottayam	9634767892	20	IT	203	IT PROFF.	45000	03-DEC-81
104	Harold	angamaly	9234567893	30	SALES	204	ANALYST	30000	01-MAY-81

10.Consider Employee table



EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E101	Amit	roduction	45000	12-Mar-00	Bangalore
E102	Amit	HR	70000	03-Jul-02	Bangalore
E103	sunita	lanagemer	120000	11-Jan-01	mysore
E105	sunita	IT	67000	01-Aug-01	mysore
E106	mahesh	Civil	145000	20-Sep-03	Mumbai

Perform the following

1. Display all the fields of employee table

2. Retrieve employee number and their salary

3. Retrieve average salary of all employee

4. Retrieve number of employee

5. Retrieve distinct number of employee

6. Retrieve total salary of employee group by employee name and count similar names

7. Retrieve total salary of employee which is greater than >120000

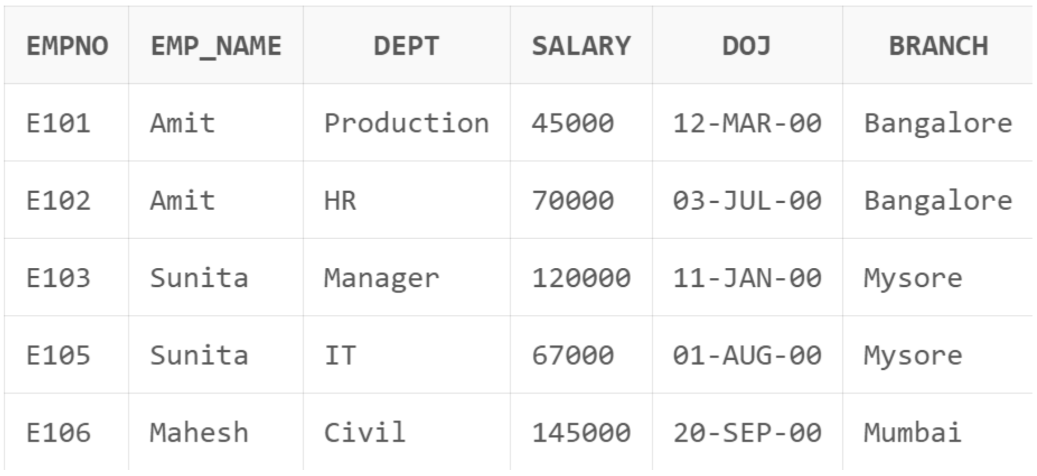
8. Display name of employee in descending order

9. Display details of employee whose name is AMIT and salary greater than 50000;

Output:

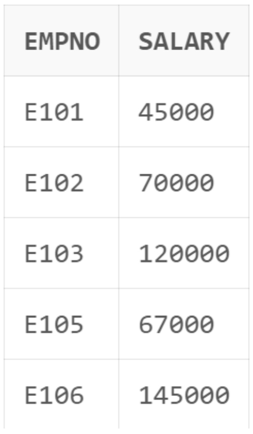
```
create table emp23 (EMPNO varchar(5), EMP_NAME varchar(10),DEPT varchar(20),SALARY integer,DOJ date,BRANCH varchar(20));

insert into emp23 values('E101','Amit','Production',45000,'12-Mar-00','Bangalore');

insert into emp23 values('E102','Amit','HR',70000,'03-Jul-00','Bangalore');

insert into emp23 values('E103','Sunita','Manager',120000,'11-Jan-00','Mysore');

insert into emp23 values('E105','Sunita','IT',67000,'01-Aug-00','Mysore');

insert into emp23 values('E106','Mahesh','Civil',145000,'20-Sep-00','Mumbai');
```

1)select * from emp23;



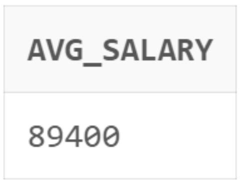
EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E101	Amit	Production	45000	12-MAR-00	Bangalore
E102	Amit	HR	70000	03-JUL-00	Bangalore
E103	Sunita	Manager	120000	11-JAN-00	Mysore
E105	Sunita	IT	67000	01-AUG-00	Mysore
E106	Mahesh	Civil	145000	20-SEP-00	Mumbai

2)select empno,salary from emp23;



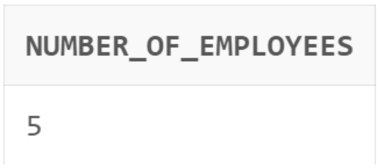
EMPNO	SALARY
E101	45000
E102	70000
E103	120000
E105	67000
E106	145000

3)select avg(salary) as avg_salary from emp23;



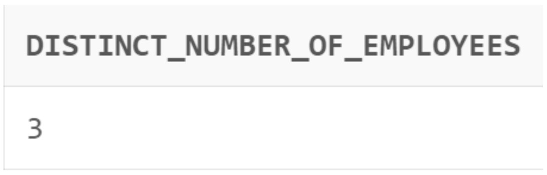
AVG_SALARY
89400

4)select count(*) as number_of_employees from emp23;



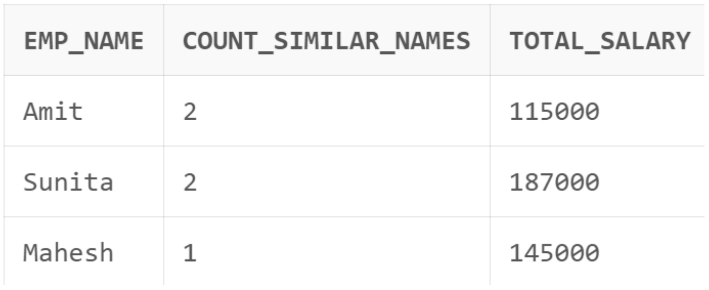
NUMBER_OF_EMPLOYEES
5

5)select count(distinct emp_name) as distinct_number_of_employees from emp23;



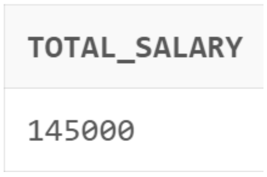
DISTINCT_NUMBER_OF_EMPLOYEES
3

6)select emp_name, count(*) as count_similar_names, sum(salary) as total_salary from emp23 group by emp_name;



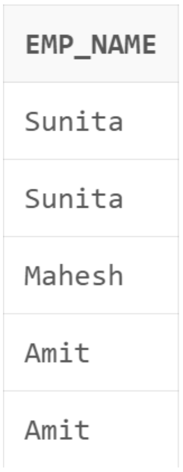
EMP_NAME	COUNT_SIMILAR_NAMES	TOTAL_SALARY
Amit	2	115000
Sunita	2	187000
Mahesh	1	145000

7)select sum(salary) as total_salary from emp23 where salary>120000;



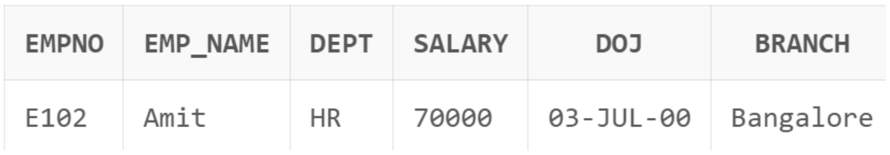
TOTAL_SALARY
145000

8)select emp_name from emp23 order by emp_name desc ;



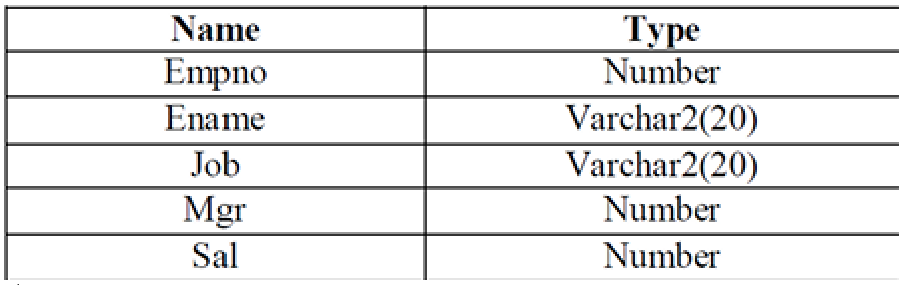
EMP_NAME
Sunita
Sunita
Mahesh
Amit
Amit

9)select * from emp23 where emp_name='Amit' and salary>50000;



EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E102	Amit	HR	70000	03-JUL-00	Bangalore

11.Create a table called Employee with the following structure.

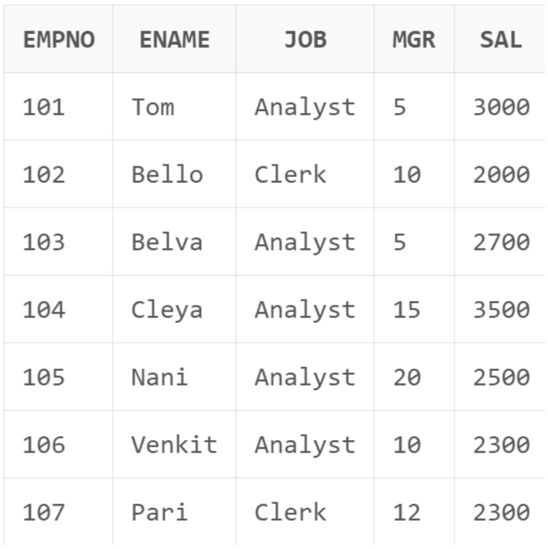


Name	Type
Empno	Number
Ename	Varchar2(20)
Job	Varchar2(20)
Mgr	Number
Sal	Number

a) Display lowest paid employee details under each department.
b) Display number of employees working in each department and their department number.
c) Using built-in functions, display number of employees working in each department and their department name from dept table. Insert deptname to dept table and insert deptname for each row, do the required thing specified above.
 d) List all employees which start with either B or C.
e) Display only these ename of employees where the maximum salary is greater than or equal to 5000.
f) Calculate the average salary for each different job.
g) Show the average salary of each job excluding manager.
h) Show the average salary for all departments employing more than three people. f.) How many days between day of birth to current date.
i) List all employee names, salary and 15% rise in salary.
j) Display lowest paid emp details under each manager
k) Display the average monthly salary bill for each deptno.
l) Show the average salary for all departments employing more than two people.
m) By using the group by clause, display the eid who belongs to deptno 05 along with average salary.
n) Count the number of employees in department 20
o) Find the minimum salary earned by clerk.
p) Find minimum, maximum, average salary of all employees.
q) List the minimum and maximum salaries for each job type.
r) List the employee names in descending order.
s) List the employee id, names in ascending order by empid.

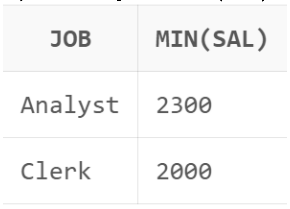
Output:

```
Create table emp23 (empno integer,ename varchar2(20),job varchar2(20),mgr integer,sal integer);
Insert into emp23 values(101,'Tom','Analyst',5,3000);
Insert into emp23 values(102,'Bello','Clerk',10,2000);
Insert into emp23 values(103,'Belva','Analyst',5,2700);
Insert into emp23 values(104,'Cleya','Analyst',15,3500);
Insert into emp23 values(105,'Nani','Analyst',20,2500);
Insert into emp23 values(106,'Venkit','Analyst',10,2300);
Insert into emp23 values(107,'Pari','Clerk',12,2300);
select * from emp23;
```



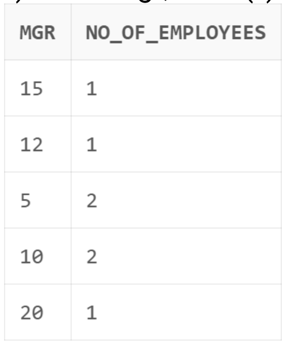
EMPNO	ENAME	JOB	MGR	SAL
101	Tom	Analyst	5	3000
102	Bello	Clerk	10	2000
103	Belva	Analyst	5	2700
104	Cleya	Analyst	15	3500
105	Nani	Analyst	20	2500
106	Venkit	Analyst	10	2300
107	Pari	Clerk	12	2300

a)Select job,min(sal) from emp23 group by job;



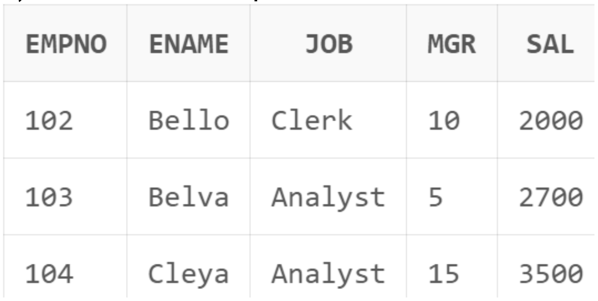
JOB	MIN(SAL)
Analyst	2300
Clerk	2000

b)select mgr,count(*) as no_of_employees from emp23 group by mgr;



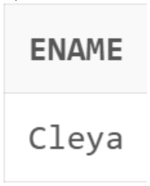
MGR	NO_OF_EMPLOYEES
15	1
12	1
5	2
10	2
20	1

d) select * from emp23 where ename like 'B%' or ename like 'C%';



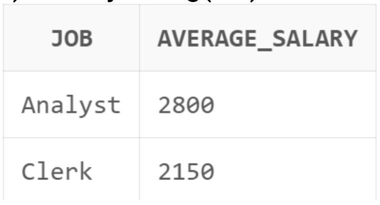
EMPNO	ENAME	JOB	MGR	SAL
102	Bello	Clerk	10	2000
103	Belva	Analyst	5	2700
104	Cleya	Analyst	15	3500

e) select ename from emp23 where sal=(select max(sal) from emp23) and sal>=3000;



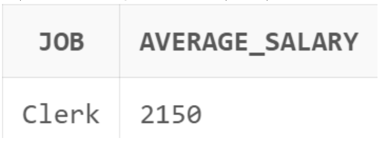
ENAME
Cleya

f)select job,avg(sal) as average_salary from emp23 group by job;



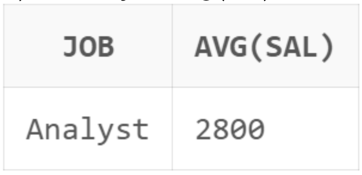
JOB	AVERAGE_SALARY
Analyst	2800
Clerk	2150

g) Select job,avg(sal) as average_salary from emp23 where job != 'Analyst' group by job;



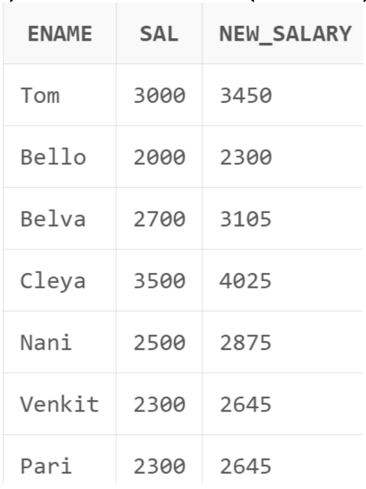
JOB	AVERAGE_SALARY
Clerk	2150

h) select job,avg(sal) from emp23 group by job having count(*)>3;



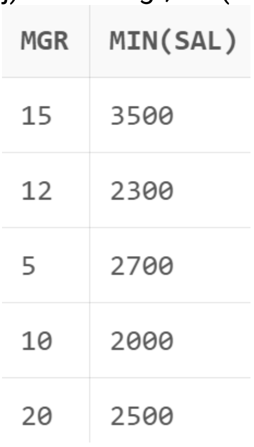
JOB	AVG(SAL)
Analyst	2800

i)select ename,sal,(sal*1.15) as new_salary from emp23;



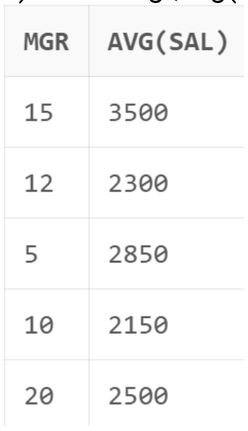
ENAME	SAL	NEW_SALARY
Tom	3000	3450
Bello	2000	2300
Belva	2700	3105
Cleya	3500	4025
Nani	2500	2875
Venkit	2300	2645
Pari	2300	2645

j) select mgr,min(sal) from emp23 group by mgr;



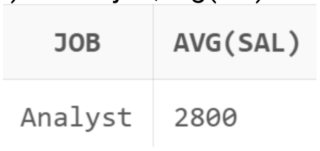
MGR	MIN(SAL)
15	3500
12	2300
5	2700
10	2000
20	2500

k) select mgr,avg(sal) from emp23 group by mgr;



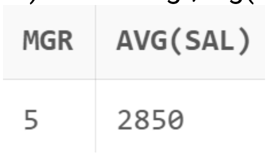
MGR	AVG(SAL)
15	3500
12	2300
5	2850
10	2150
20	2500

l) select job,avg(sal) from emp23 group by job having count(*)>2;



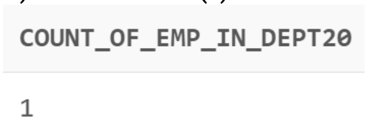
JOB	AVG(SAL)
Analyst	2800

m) select mgr,avg(sal) from emp23 where mgr=05 group by mgr;



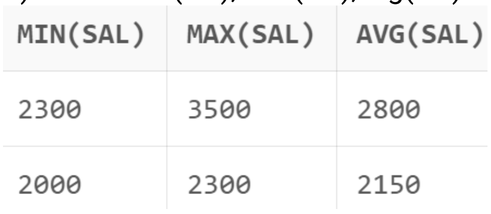
MGR	AVG(SAL)
5	2850

n) select count(*) as count_of_emp_in_dept20 from emp23 where mgr=20;



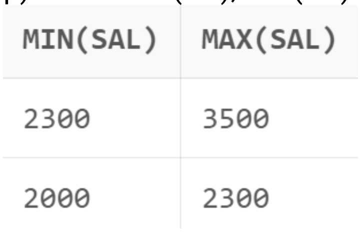
COUNT_OF_EMP_IN_DEPT20
1

o) select min(sal),max(sal),avg(sal) from emp23 group by job;



MIN(SAL)	MAX(SAL)	AVG(SAL)
2300	3500	2800
2000	2300	2150

p) select min(sal),max(sal) from emp23 group by job;



MIN(SAL)	MAX(SAL)
2300	3500
2000	2300

q) select job,min(sal),max(sal) from emp23 group by job;

JOB	MIN(SAL)	MAX(SAL)
Analyst	2300	3500
Clerk	2000	2300

r) select ename from emp23 order by ename desc;

ENAME
Venkit
Tom
Pari
Nani
Cleya
Belva
Bello

s) select empno,ename from emp23 order by empno;

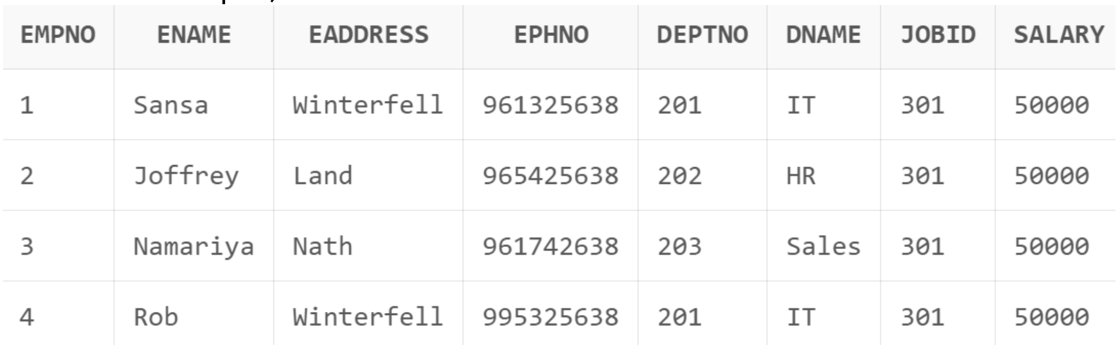
EMPNO	ENAME
101	Tom
102	Bello
103	Belva
104	Cleya
105	Nani
106	Venkit
107	Pari

Experiment 5: Implementation of different types of functions with suitable examples

11.Create a table EMPLOYEE with following schema: (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)
a). Display all the dept numbers available with the dept and emp tables avoiding duplicates.
b). Display all the dept numbers available with the dept and emp tables.
c) Display all the dept numbers available in emp and not in dept tables and vice versa.

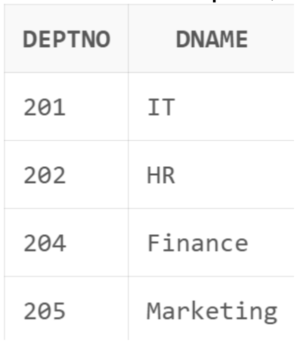
Output :

```
create table emp23 (empno integer,ename varchar(20),eaddress varchar(20),ephno
varchar(10),deptno integer,dname varchar(10),jobid integer,Salary integer);
insert into emp23 values(01,'Sansa','Winterfell','961325638',201,'IT',301,50000);
insert into emp23 values(02,'Joffrey','Land','965425638',202,'HR',301,50000);
insert into emp23 values(03,'Namariya','Nath','961742638',203,'Sales',301,50000);
insert into emp23 values(04,'Rob','Winterfell','995325638',201,'IT',301,50000);
select * from emp23;
```



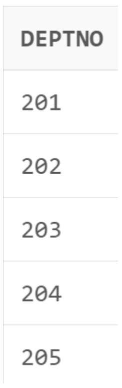
EMPNO	ENAME	EADDRESS	EPHNO	DEPTNO	DNAME	JOBID	SALARY
1	Sansa	Winterfell	961325638	201	IT	301	50000
2	Joffrey	Land	965425638	202	HR	301	50000
3	Namariya	Nath	961742638	203	Sales	301	50000
4	Rob	Winterfell	995325638	201	IT	301	50000

```
a) create table dept23(deptno integer,dname varchar(10));
insert into dept23 values(201,'IT');
insert into dept23 values(202,'HR');
insert into dept23 values(204,'Finance');
insert into dept23 values(205,'Marketing');
select * from dept23;
```



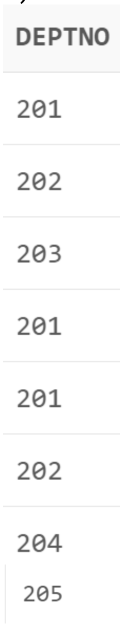
DEPTNO	DNAME
201	IT
202	HR
204	Finance
205	Marketing

select deptno from emp23 UNION select deptno from dept23;



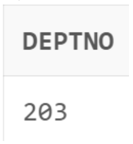
DEPTNO
201
202
203
204
205

b)select deptno from emp23 UNION ALL select deptno from dept23;



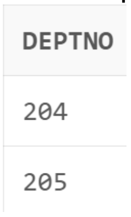
DEPTNO
201
202
203
201
201
202
204
205

c)select deptno from emp23 MINUS select deptno from dept23;



DEPTNO
203

select deptno from dept23 MINUS select deptno from emp23;



DEPTNO
204
205

Experiment 6: Implementation of Join,Views,Set operations

12. Consider the following schema: Sailors (sid, sname, rating, age) Boats (bid, bname, color) Reserves (sid, bid, day(date))

a) Find all the information of sailors who have reserved boat number 101.
b) Find the name of boat reserved by Bob.
c) Find the names of sailors who have reserved a red boat, and list in order of age.
d) Find the names of sailors who have reserved at least one boat.
e) Find the ids and names of sailors who have reserved two different boats on the same day.
f) Find the ids of sailors who have reserved a red boat or a green boat.
g) Find the name and the age of the youngest sailor.
h) Count the number of different sailor names.
i) Find the average age of sailors for each rating level.
j) Find the average age of sailors for each rating level that has at least two sailors.

Output:

```
SQL>create table Sailors (sid number primary key,Sname varchar(20),rating number,age number);
SQL>insert into Sailors values (1, 'Bob', 5, 25);
SQL>insert into Sailors values (2, 'Alice', 3, 22);
SQL>insert into Sailors values (3, 'Charlie', 4, 27);
SQL>insert into Sailors values (4, 'David', 2, 24);
SQL>insert into Sailors values (5, 'Eve', 5, 20);

SQL> create table Boats (bid number primary key,bname varchar(20),color varchar(10));
SQL>insert into Boats values (101, 'Boat1', 'Red');
SQL>insert into Boats values (102, 'Boat2', 'Blue');
SQL>insert into Boats values (103, 'Boat3', 'Green');
SQL>insert into Boats values (104, 'Boat4', 'Red');

SQL> create table Reserves (sid number,bid number day date primary key (sid, bid, day),foreign key (sid) references Sailors(sid),foreign key (bid) references Boats(bid));
SQL> insert into Reserves values (1, 101, TO_DATE('2025-04-17', 'YYYY-MM-DD'));
SQL>insert into Reserves values (2, 102, TO_DATE('2025-04-17', 'YYYY-MM-DD'));
SQL>insert into Reserves values (3, 103, TO_DATE('2025-04-16', 'YYYY-MM-DD'));
SQL>insert into Reserves values (4, 101, TO_DATE('2025-04-17', 'YYYY-MM-DD'));
SQL>insert into Reserves values (5, 104, TO_DATE('2025-04-16', 'YYYY-MM-DD'));
SQL>insert into Reserves values (1, 103, TO_DATE('2025-04-15', 'YYYY-MM-DD'));
SQL>insert into Reserves values (2, 101, TO_DATE('2025-04-16', 'YYYY-MM-DD'));
```

a)SQL> select S.* from Sailors S join Reserves R on S.sid = R.sid where R.bid = 101;

sid	sname	rating	age
1	Bob	5	25
4	David	2	24

b) SQL> select B.bname from Boats B join Reserves R on B.bid = R.bid join Sailors S on R.sid = S.sid where S.sname = 'Bob';

bname
Boat1
Boat3

c) SQL> select S.sname from Sailors S join Reserves R on S.sid = R.sid join Boats B on R.bid = B.bid where B.color = 'Red' order by S.age;

sname
Eve
David
Bob

d) SQL> select DISTINCT S.sname from Sailors S join Reserves R on S.sid = R.sid;

sname
Bob
Alice
Charlie
David
Eve

e)SQL> select R1.sid, S.sname from Reserves R1, Reserves R2 join Sailors S on R1.sid = S.sid where R1.sid = R2.sid and R1.bid != R2.bid and R1.day = R2.day;

sid	sname
1	Bob
2	Alice

f) SQL> select DISTINCT R.sid from Reserves R join Boats B on R.bid = B.bid where B.color in ('Red', 'Green');

sid
1
2
3
4

g)SQL> select S.sname, S.age from Sailors S where S.age = (select MIN(age) from Sailors);

sname	age
Eve	20

h) SQL> select COUNT(DISTINCT sname) as distinct_sailor_names from Sailors;

distinct_sailor_names
5

i) SQL> select rating, AVG(age) as average_age from Sailors group by rating;

rating	average_age
5	22.5
3	22
4	27
2	24

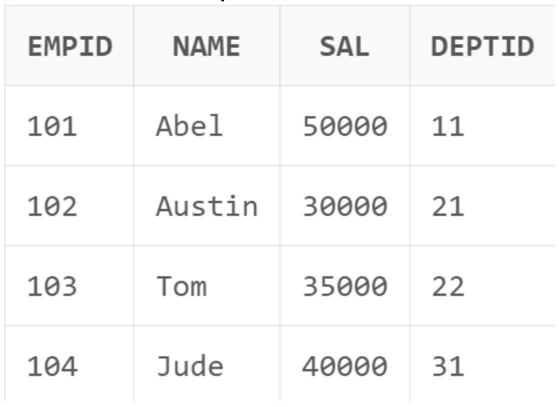
j)SQL> select rating, AVG(age) as average_age from Sailors group by rating having COUNT(sid) >= 2;

rating	average_age
5	22.5

13 Original Table: Employees (employee_id, name, salary, department_id)
Question: Create a view named EmployeeDetails that displays the employee ID, name, and salary from the Employees table.

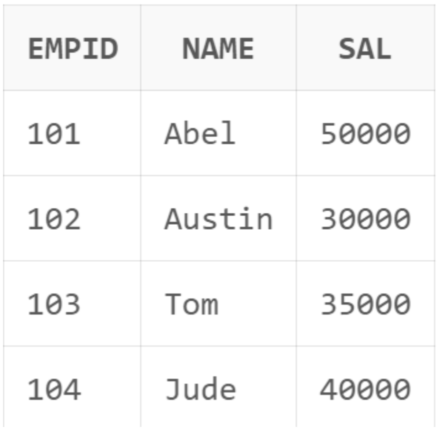
Output:

```
create table emp23(empid integer,name varchar(20),sal integer,deptid integer);
insert into emp23 values(101,'Abel',50000,11);
insert into emp23 values(102,'Austin',30000,21);
insert into emp23 values(103,'Tom',35000,22);
insert into emp23 values(104,'Jude',40000,31);
select * from emp23;
```



EMPID	NAME	SAL	DEPTID
101	Abel	50000	11
102	Austin	30000	21
103	Tom	35000	22
104	Jude	40000	31

```
create view EmployeeDetails as select empid,name,sal from emp23;
select * from EmployeeDetails;
```

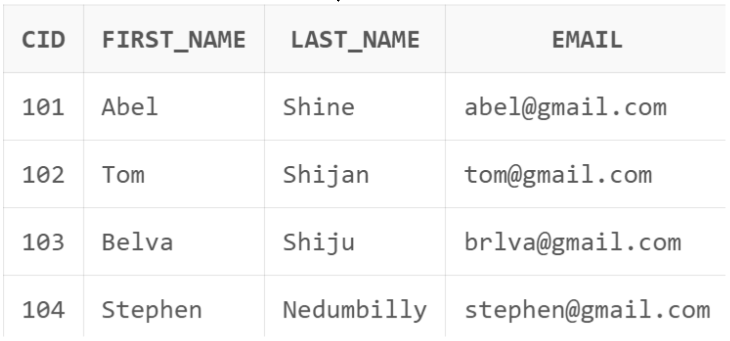


EMPID	NAME	SAL
101	Abel	50000
102	Austin	30000
103	Tom	35000
104	Jude	40000

14. Original Table: Customers (customer_id, first_name, last_name, email)
Question: Write a SQL query to create a view called CustomerContacts that combines the customer's first name, last name, and email address from the Customers table.

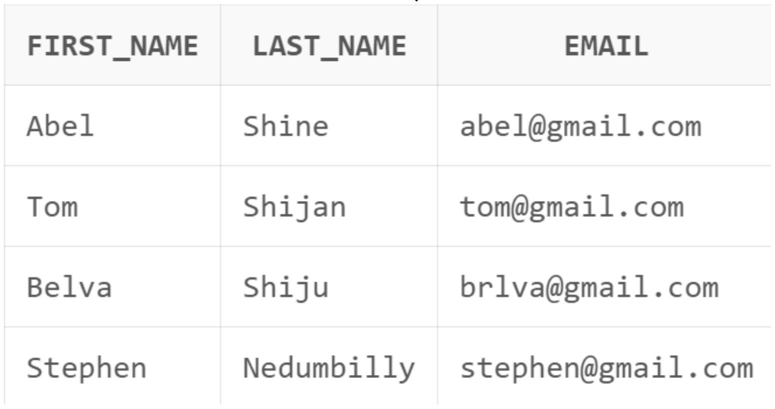
Output:

```
create table customer23(cid integer,first_name varchar(20),last_name varchar(20),email
varchar(20));
insert into customer23 values(101,'Abel','Shine','abel@gmail.com');
insert into customer23 values(102,'Tom','Shijan','tom@gmail.com');
insert into customer23 values(103,'Belva','Shiju','brlva@gmail.com');
insert into customer23 values(104,'Stephen','Nedumbilly','stephen@gmail.com');
select * from customer23;
```



CID	FIRST_NAME	LAST_NAME	EMAIL
101	Abel	Shine	abel@gmail.com
102	Tom	Shijan	tom@gmail.com
103	Belva	Shiju	brlva@gmail.com
104	Stephen	Nedumbilly	stephen@gmail.com

```
create view customercontacts as select first_name,last_name,email from customer23;
select * from customercontacts;
```

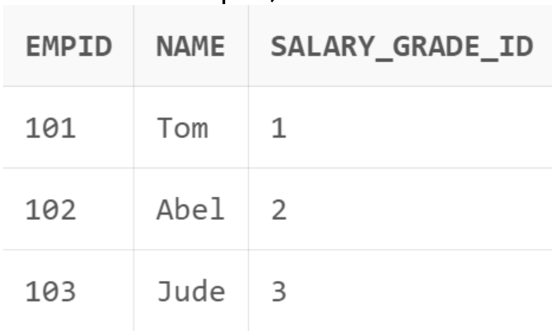


FIRST_NAME	LAST_NAME	EMAIL
Abel	Shine	abel@gmail.com
Tom	Shijan	tom@gmail.com
Belva	Shiju	brlva@gmail.com
Stephen	Nedumbilly	stephen@gmail.com

15.Original Tables: Employees (employee_id, name, salary_grade_id),
SalaryGrades (salary_grade_id, min_salary, max_salary)
Create a view named EmployeeSalaries that shows the employee ID, name, and salary along with the salary grade from the Employees and SalaryGrades tables.

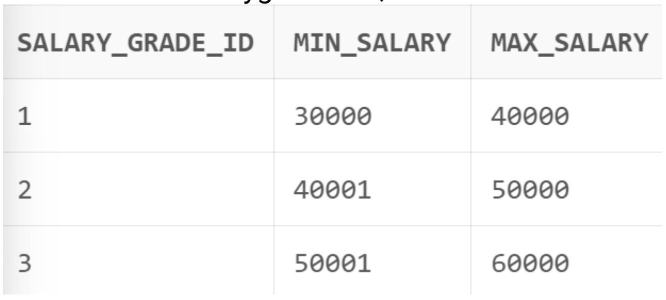
Output:

```
insert into emp23 values (101, 'Tom', 1);
insert into emp23 values (102, 'Abel', 2);
insert into emp23 values (103, 'Jude', 3);
select * from emp23;
```



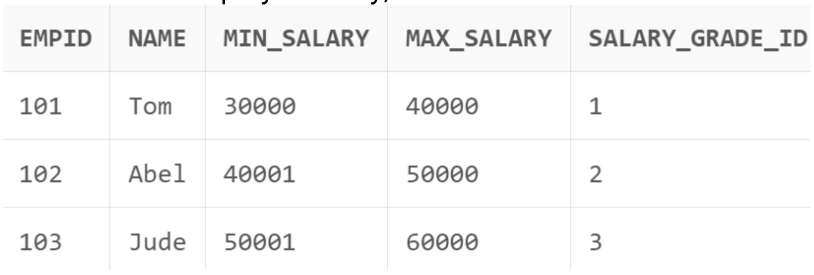
EMPID	NAME	SALARY_GRADE_ID
101	Tom	1
102	Abel	2
103	Jude	3

```
insert into salarygrades23 values (1, 30000, 40000);
insert into salarygrades23 values (2, 40001, 50000);
insert into salarygrades23 values (3, 50001, 60000);
select * from salarygrades23;
```



SALARY_GRADE_ID	MIN_SALARY	MAX_SALARY
1	30000	40000
2	40001	50000
3	50001	60000

```
create view employeesalary as select E.empid , E.name , S.min_salary , S.max_salary ,
S.salary_grade_id from emp23 E join salarygrades23 S on E.salary_grade_id =
S.salary_grade_id;
select * from employeesalary;
```

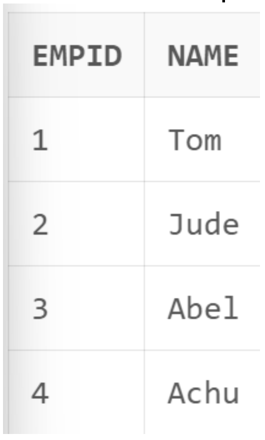


EMPID	NAME	MIN_SALARY	MAX_SALARY	SALARY_GRADE_ID
101	Tom	30000	40000	1
102	Abel	40001	50000	2
103	Jude	50001	60000	3

16. Create tables Employees (employee_id , name) Managers (manager_id, name)
a) Write a SQL query to retrieve the names of all employees and managers, ensuring that duplicate names are removed.
b) Create a query to find the common names between employees and managers.
c) Write a query to find the names of employees who are not managers.
d) Write a query to find the distinct names of all employees and managers, along with their respective roles (employee/manager).

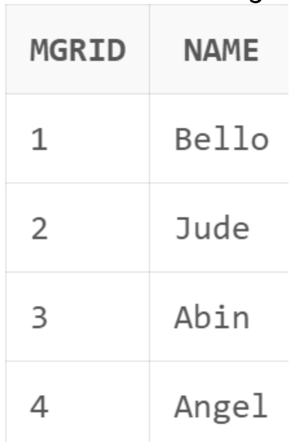
Output:

```
create table emp23 (empid integer,name varchar(20));
create table manager23(mgrid integer,name varchar(20));
insert into emp23 values(1,'Tom');
insert into emp23 values(2,'Jude');
insert into emp23 values(3,'Abel');
insert into emp23 values(4,'Achu');
select * from emp23;
```



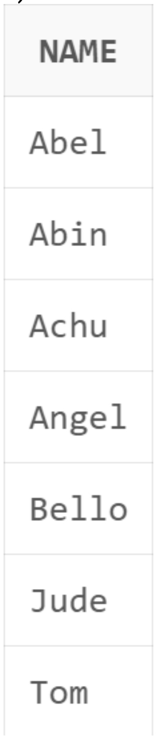
EMPID	NAME
1	Tom
2	Jude
3	Abel
4	Achu

```
insert into manager23 values(1,'Bello');
insert into manager23 values(2,'Jude');
insert into manager23 values(3,'Abin');
insert into manager23 values(4,'Angel');
select * from manager23;
```



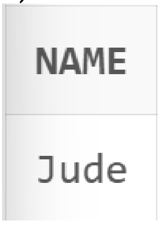
MGRID	NAME
1	Bello
2	Jude
3	Abin
4	Angel

a) select name from emp23 UNION select name from manager23;



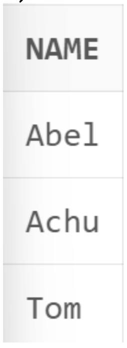
NAME
Abel
Abin
Achu
Angel
Bello
Jude
Tom

b) select name from emp23 INTERSECT select name from manager23;



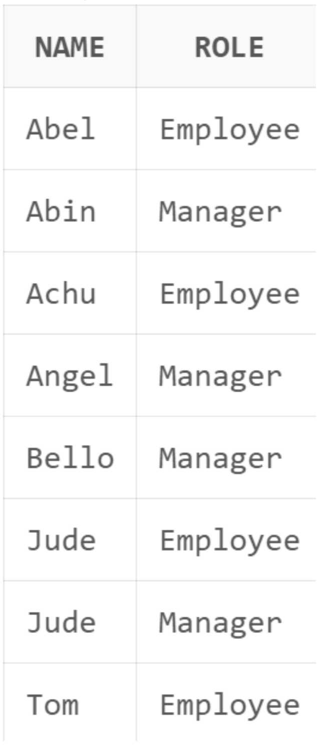
NAME
Jude

c) select name from emp23 MINUS select name from manager23;



NAME
Abel
Achu
Tom

d) select name,'Employee' as role from emp23 UNION select name,'Manager' as role from manager23;



NAME	ROLE
Abel	Employee
Abin	Manager
Achu	Employee
Angel	Manager
Bello	Manager
Jude	Employee
Jude	Manager
Tom	Employee

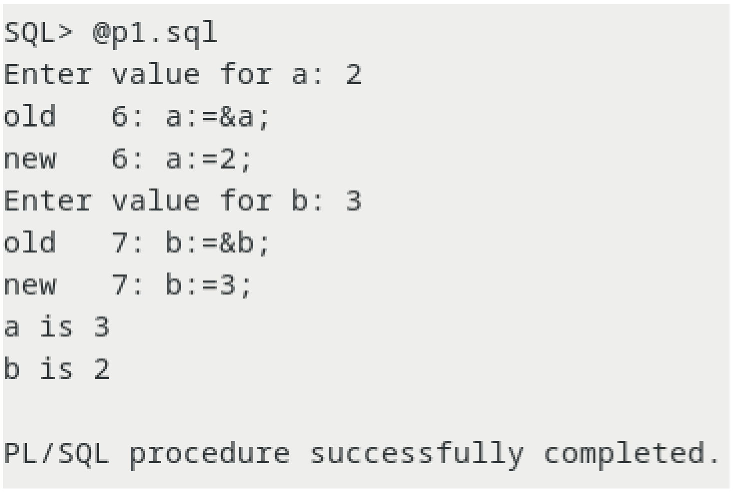
Experiment 7: PLSQL

17. write a PL/SQL program to swap the values of two numbers.

Program:

```
declare
a number;
b number;
temp number;
begin
a:=&a;
b:=&b;

temp:=a;
a:=b;
b:=temp;

dbms_output.put_line('a is'||a);
dbms_output.put_line('b is'||b);
end;
```

Output:



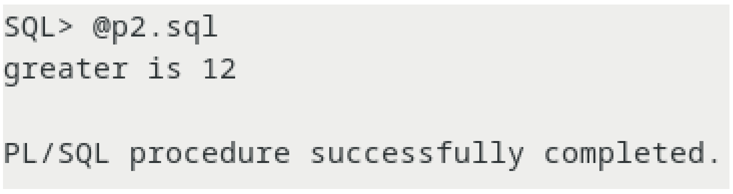
```
SQL> @p1.sql
Enter value for a: 2
old   6: a:=&a;
new   6: a:=2;
Enter value for b: 3
old   7: b:=&b;
new   7: b:=3;
a is 3
b is 2

PL/SQL procedure successfully completed.
```

18.Write a PL/SQL program to determine the largest among three given numbers.

Program:

```
declare
a number;
b number;
c number;
begin
a:=12;
b:=2;
c:=3;
 if a>b and a>c
 then
 dbms_output.put_line('greater is ' ||a);
 elsif b>c
 then
 dbms_output.put_line('greater is ' ||b);
 else
 dbms_output.put_line('greater is ' ||c);
 end if;
 end;
 /
```

Output:



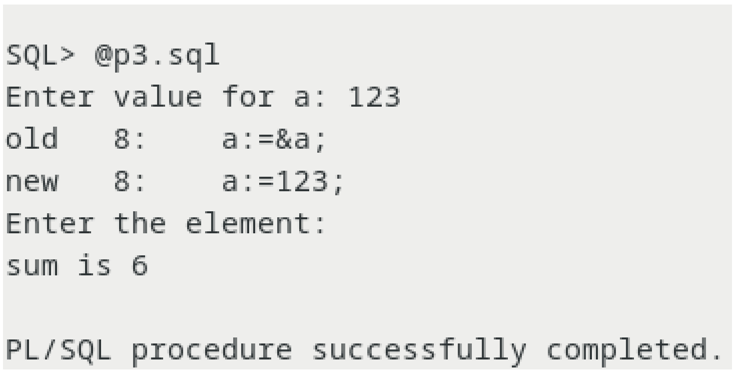
```
SQL> @p2.sql
greater is 12

PL/SQL procedure successfully completed.
```

19.Write a PL/SQL program to compute the sum of digits of a given number.

Program:

```
declare
        a number;
        s int;
        m number;
begin
  s:=0;
  dbms_output.put_line('Enter the element:');
  a:=&a;
  while a>0  loop
  m:= mod(a,10);
  s:= s+m;
  a:= floor(a / 10);
  end loop;
  dbms_output.put_line('sum is ' || s);
end;
/
```

Output:



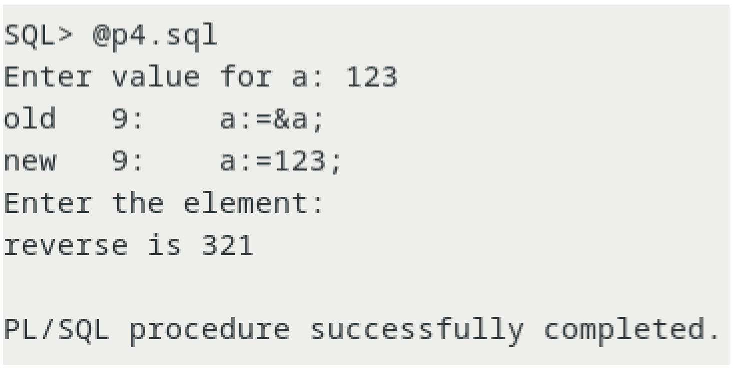
```
SQL> @p3.sql
Enter value for a: 123
old   8:    a:=&a;
new   8:    a:=123;
Enter the element:
sum is 6

PL/SQL procedure successfully completed.
```

20.Write a PL/SQL program to display a given number in reverse order.

Program:

```
declare
        a number;
        s int;
        m number;
begin
   s:= 0;
   dbms_output.put_line('Enter the element:');
   a:=&a;
   while a>0  loop
   m:= mod(a,10);
   s:= (s*10)+m;
   a:= floor(a / 10);
   end loop;
   dbms_output.put_line(' reverse is ' ||s);
end;
/
```

Output;



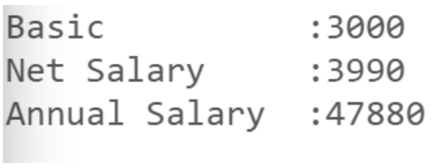
```
SQL> @p4.sql
Enter value for a: 123
old   9:    a:=&a;
new   9:    a:=123;
Enter the element:
reverse is 321

PL/SQL procedure successfully completed.
```

21. Write a PL/SQL program to calculate the net salary and annual salary, considering DA as 30% of basic, HRA as 10% of basic, and PF as:7% if the basic salary is less than 8000 10% if the basic salary is between 8000 and 16000.

Program:

```
declare
  basic number;
  da number;
  hra number;
  pf number;
  net_sal number;
  annual_sal number;
begin
  basic:=3000;
  da:=0.3*basic;
  hra:=0.1*basic;
  if(basic<8000) then
      pf:=0.07*basic;
  elsif(basic<=16000) then
      pf:=0.1*basic;
  else
      dbms_output.put_line('Invalid basic!');
      pf:=0;
  end if;
  net_sal:=basic+da+hra-pf;
  annual_sal:=net_sal*12;
  dbms_output.put_line('Basic          :'||basic);
  dbms_output.put_line('Net Salary     :'||net_sal);
  dbms_output.put_line('Annual Salary  :'||annual_sal);
end;
```

Output:

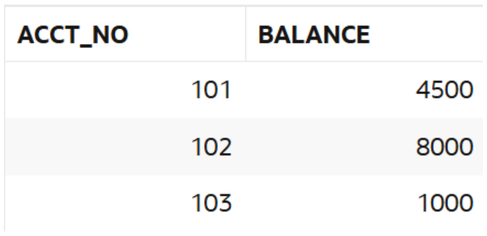

```
Basic          :3000
Net Salary     :3990
Annual Salary  :47880
```

22. Write a PL/SQL program that accepts an account number, checks if the balance is below the minimum required balance, and deducts Rs.100/- from the balance if necessary. The program should be applied to the acct table.

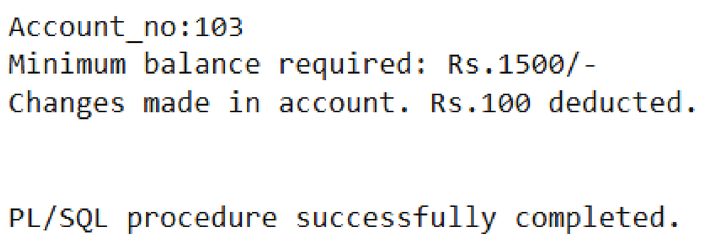
Program:

```
declare
   v_acct_no number ;
   v_balance number;
begin
   v_acct_no:=&v_acct_no;
   dbms_output.put_line('account_no:'||v_acct_no);
   dbms_output.put_line('minimum balance required: rs.1500/-');
   select balance into v_balance from account where acct_no = v_acct_no;

   if v_balance < 1500 then
      update account set balance = balance - 100 where acct_no = v_acct_no;
      dbms_output.put_line('changes made in account. rs.100 deducted.');
   else
      dbms_output.put_line('balance sufficient. no changes needed.');
   end if;
end;
```

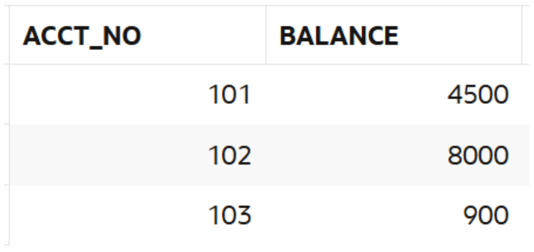
Output:



ACCT_NO	BALANCE
101	4500
102	8000
103	1000



```
Account_no:103
Minimum balance required: Rs.1500/-
Changes made in account. Rs.100 deducted.


PL/SQL procedure successfully completed.
```



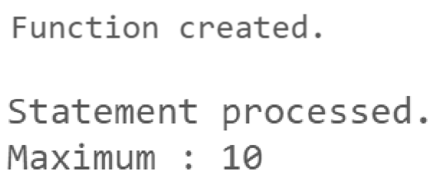
ACCT_NO	BALANCE
101	4500
102	8000
103	900

23.Write a PL/SQL function that computes and returns the maximum of two given values.

Program:

```
create or replace function max(a number,b number)
  return number
as
begin
  return greatest(a,b);
end;

declare
  a number;
  b number;
  c number;
begin
  a:=10;
  b:=9;
  c:=get_max(a,b);
  dbms_output.put_line('Maximum : '||c);
end;
```

Output:



```
Function created.

Statement processed.
Maximum : 10
```

24. Write a PL/SQL function to check whether a given string is a palindrome.

Program:

```
create or replace function is_palindrome(str in varchar)
return varchar2
as
  rev_str varchar(20) := '';
  i integer;
begin
  for i in reverse 1..length(str) loop
    rev_str := rev_str || substr(str, i, 1);
  end loop;

  if lower(str) = lower(rev_str) then
    return 'palindrome';
  else
    return 'not palindrome';
  end if;
end;
/
declare
  result varchar(20);
  string varchar(20);
begin
  string := '&string';
  dbms_output.put_line('given string: ' || string);
  result := is_palindrome(string);
  dbms_output.put_line('result: ' || result);
end;
```

Output:

```
Statement processed.
Given string: malayalam
Result: palindrome
```

25. Write a PL/SQL function that returns the total count of customers in the customers table.

Program:

```
create or replace function customer_count
return number
as
  total number;
begin
  select count(*) into total from customers;
  return total;
end;
 /
declare
  count number;
begin
  count := customer_count;
  dbms_output.put_line('total customers: ' || count);
end;
/
```

Output:

CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL
1	Gokul	krishna	gokul123@gmail.com
2	Sooraj	Santhosh	soorajss11@gmail.com
3	Anandhu	KS	anandhu97@gmail.com

```
Statement processed.
total customers: 3
```

26. Write a PL/SQL procedure to compute and display the sum of two numbers.

Program:

```
create or replace procedure sum_two(a in number, b in number)
as
  total number;
begin
  total := a + b;
  dbms_output.put_line('sum of ' ||a|| ' and ' || b  || ' =' || total);
end;
/

declare
a number;
b number;
begin
a:=&a;
b:=&b;
  sum_two(a, b);
end;
/
```

Output:

```
Statement processed.
Sum of 36 and 24 = 60
```

27. Write a PL/SQL procedure to insert a student's roll number and name into the student table.

Program:

```
create or replace procedure insert_student(p_roll in number, p_name in varchar2)
as
begin
  insert into student(rollno, name) values (p_roll, p_name);
  dbms_output.put_line('student inserted('||p_roll || ',' || p_name || ')');
end;
/

declare
roll number;
name varchar2(20);
begin
roll:=&roll;
name:='&name';
  insert_student(roll, name);
end;
```

Output:

```
Statement processed.
student inserted(23,Ann)
```

28. Write a PL/SQL procedure to retrieve and display the count of instructors in a specified department.

Program:

```
create or replace procedure instructor_count(p_dept in varchar2)
as
  cnt number;
begin
  select count(*) into cnt from instructor where dept = p_dept;
  dbms_output.put_line('instructors in ' || p_dept || ': ' || cnt);
end;
/

declare
begin
  instructor_count('mca');
end;
/
```

Output:

FID	**DEPT**	**FNAME**
1	mca	anu
2	ece	arun
3	mca	deepa
4	eee	varun

```
Statement processed.
Instructors in mca: 2
```

29. Create a Customers table with attributes (CustId (Primary Key), CustName, City). Then, write a PL/SQL program using an explicit cursor to display all details from the Customers table.

Program:

```
declare
   cursor cust_cursor is select * from customers;
   rec customers%rowtype;
begin
   open cust_cursor;
   loop
      fetch cust_cursor into rec;
      exit when cust_cursor%notfound;
      dbms_output.put_line('id: ' || rec.custid ||
                     ', name: ' || rec.custname ||
                     ', city: ' || rec.city);
   end loop;
   close cust_cursor;
end;
```

Output:

```
ID: 1, Name: Alice, City: Ernakulam
ID: 2, Name: Bob, City: Kottayam
ID: 3, Name: Carol, City: Kollam
```

30. Write a PL/SQL program using an explicit cursor to display details of employees working in the MCA department.

Program:

```
declare
   cursor emp_cursor is select * from instructor where dept = 'mca';
   rec instructor%rowtype;
begin
   open emp_cursor;
   loop
      fetch emp_cursor into rec;
      exit when emp_cursor%notfound;
      dbms_output.put_line('id: ' || rec.fid || ', name: ' || rec.fname);
   end loop;
   close emp_cursor;
end;
```

Output:

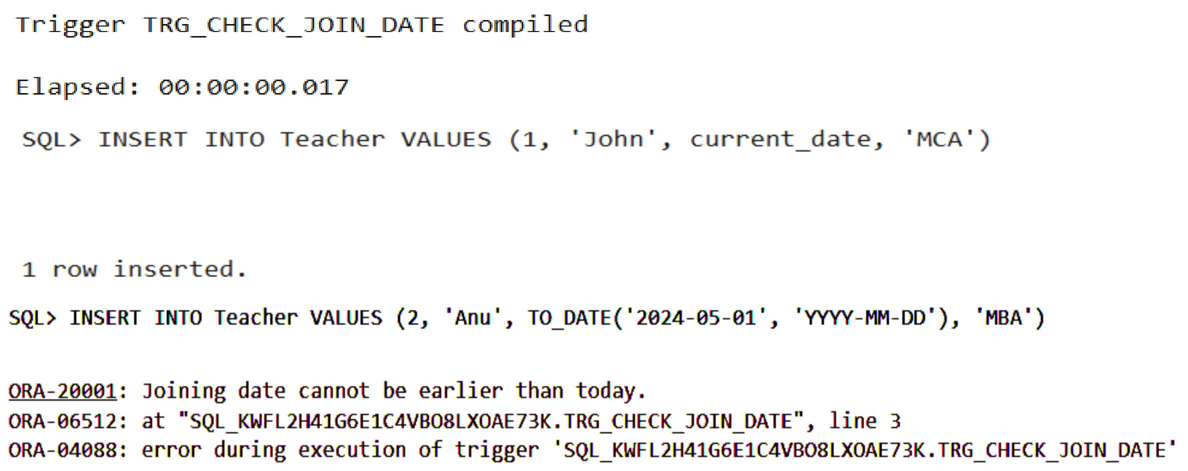
```
ID: 1, Name: Anu
ID: 2, Name: Deepa
```

31. Create a table Teacher with following attributes
Teacher(T_id, T_name, Join_date, Department). Write a trigger that verifies the joining date when a new row is inserted in the 'teacher' table. Joining date should be greater than or equal to current date.

Program:

```
create table teacher ( t_id number primary key,  t_name varchar2(20), join_date
date,department varchar2(20));

create or replace trigger trg_check_join_date
before insert on teacher
for each row
begin
   if :new.join_date < trunc(sysdate) then
      raise_application_error(-20001, 'joining date cannot be earlier than today.');
   end if;
end;
/
insert into teacher values (1, 'john', current_date, 'mca');
insert into teacher values (2, 'anu', to_date('2024-05-01', 'yyyy-mm-dd'), 'mba');
```

Output:



```
Trigger TRG_CHECK_JOIN_DATE compiled

Elapsed: 00:00:00.017

 SQL> INSERT INTO Teacher VALUES (1, 'John', current_date, 'MCA')



 1 row inserted.
SQL> INSERT INTO Teacher VALUES (2, 'Anu', TO_DATE('2024-05-01', 'YYYY-MM-DD'), 'MBA')

ORA-20001: Joining date cannot be earlier than today.
ORA-06512: at "SQL_KWFL2H41G6E1C4VBO8LXOAE73K.TRG_CHECK_JOIN_DATE", line 3
ORA-04088: error during execution of trigger 'SQL_KWFL2H41G6E1C4VBO8LXOAE73K.TRG_CHECK_JOIN_DATE'
```

EXPERIMENT 8: RELATIONAL AND NON-RELATIONAL (NOSQL) DATABASES

32. Comparison between relational and non-relational (NoSQL) databases and the configuration of NoSQL Databases.

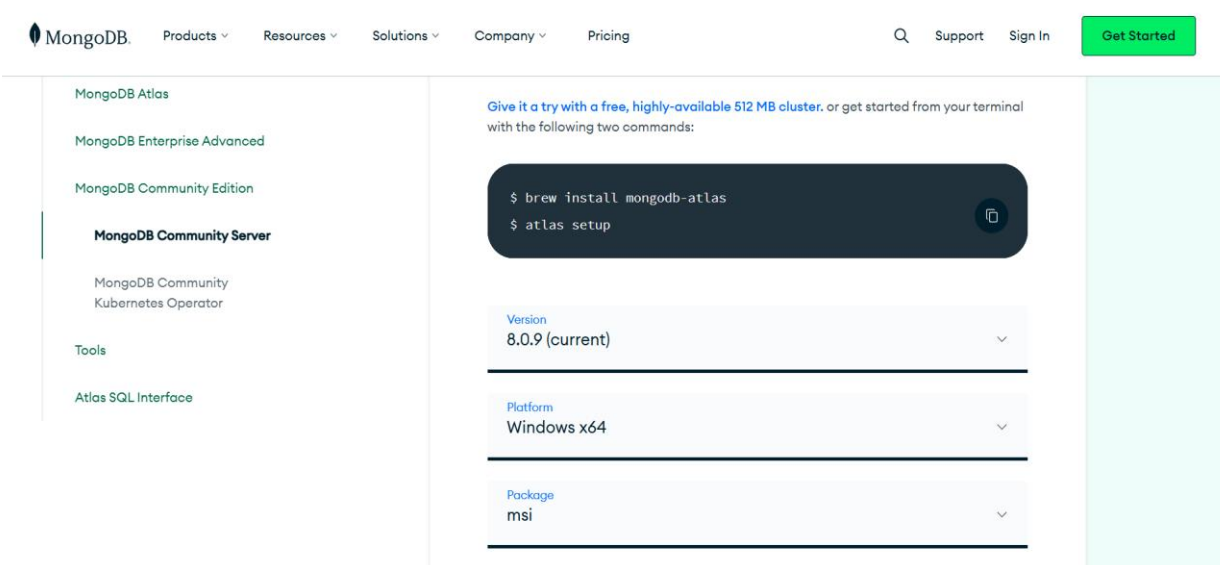
Output:

Feature	**Relational (SQL) Databases**	**Non-Relational (NoSQL) Databases**
Data Model	Structured tables with rows and columns	Flexible models: document, key-value, wide-column, graph
Schema	Fixed schema; predefined structure	Dynamic schema; allows for unstructured or semi-structured data
Scalability	Vertical scaling (adding more power to a single server)	Horizontal scaling (adding more servers to distribute the load)
Query Language	Structured Query Language (SQL)	Varies by database (e.g., MongoDB uses its own query language)
Transactions	Strong ACID compliance (Atomicity, Consistency, Isolation, Durability)	Some support for ACID; others favor BASE (Basically Available, Soft state, Eventual consistency)
Examples	MySQL, PostgreSQL, Oracle, Microsoft SQL Server	MongoDB, Cassandra, Redis, Couchbase, Neo4j
Use Cases	Complex queries, multi-row transactions, structured data	Large volumes of diverse data, real-time analytics, content management, IoT, big data applications
Data Integrity	Enforced through constraints and relationships	Application-level enforcement; less emphasis on strict data integrity
Flexibility	Less flexible; changes require altering the schema	Highly flexible; easy to add new fields or data types without affecting existing data
Performance	Optimized for complex queries and joins	Optimized for high-speed read/write operations and large-scale data handling

EXPERIMENT 9:PROGRAMS USING MONGODB

33.Install the MongoDB and configure it.

Output:



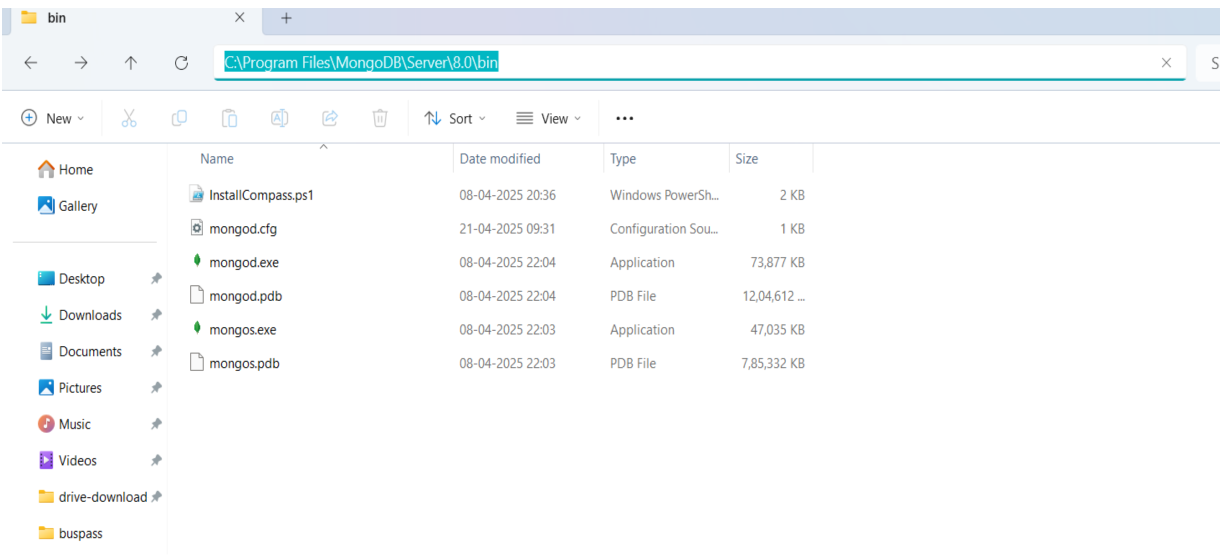
Give it a try with a free, highly-available 512 MB cluster. or get started from your terminal with the following two commands:

```
$ brew install mongodb-atlas
$ atlas setup
```

Version
8.0.9 (current)

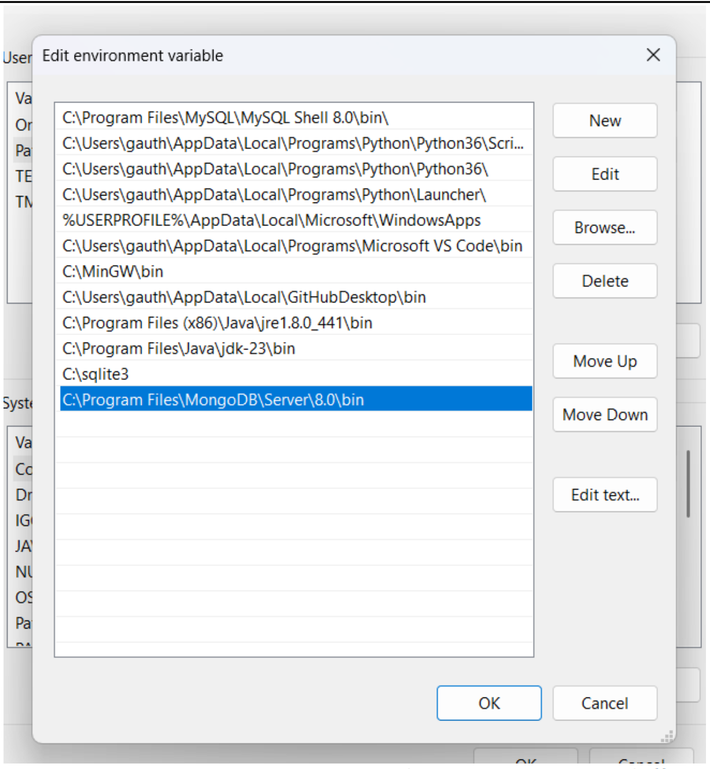
Platform
Windows x64

Package
msi

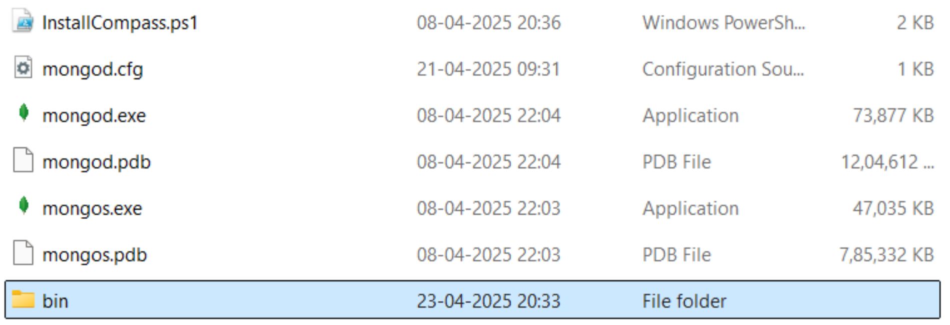


C:\Program Files\MongoDB\Server\8.0\bin

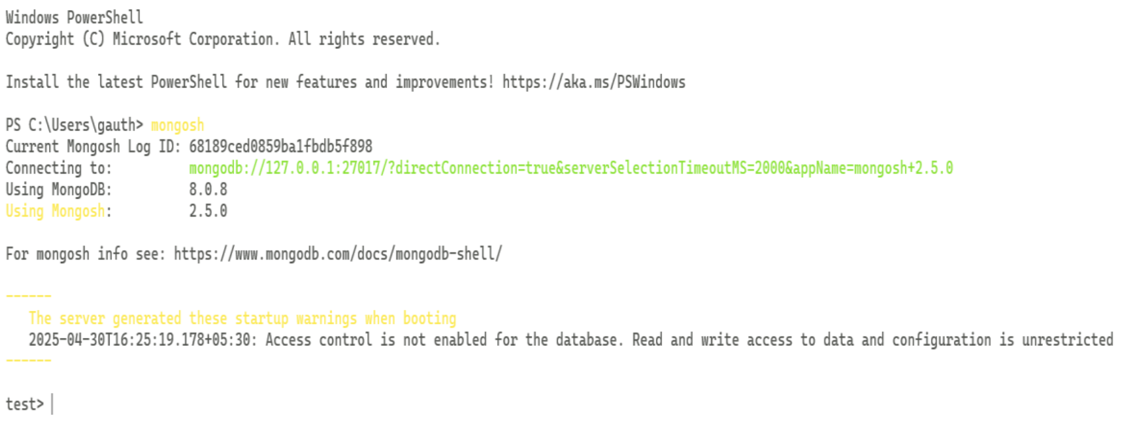
Name	Date modified	Type	Size
InstallCompass.ps1	08-04-2025 20:36	Windows PowerSh...	2 KB
mongod.cfg	21-04-2025 09:31	Configuration Sou...	1 KB
mongod.exe	08-04-2025 22:04	Application	73,877 KB
mongod.pdb	08-04-2025 22:04	PDB File	12,04,612 ...
mongos.exe	08-04-2025 22:03	Application	47,035 KB
mongos.pdb	08-04-2025 22:03	PDB File	7,85,332 KB



```
C:\Program Files\MySQL\MySQL Shell 8.0\bin\
C:\Users\gauth\AppData\Local\Programs\Python\Python36\Scri...
C:\Users\gauth\AppData\Local\Programs\Python\Python36\
C:\Users\gauth\AppData\Local\Programs\Python\Launcher\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\gauth\AppData\Local\Programs\Microsoft VS Code\bin
C:\MinGW\bin
C:\Users\gauth\AppData\Local\GitHubDesktop\bin
C:\Program Files (x86)\Java\jre1.8.0_441\bin
C:\Program Files\Java\jdk-23\bin
C:\sqlite3
C:\Program Files\MongoDB\Server\8.0\bin
```



```
InstallCompass.ps1   08-04-2025 20:36   Windows PowerSh...   2 KB
mongod.cfg           21-04-2025 09:31   Configuration Sou...  1 KB
mongod.exe           08-04-2025 22:04   Application          73,877 KB
mongod.pdb           08-04-2025 22:04   PDB File             12,04,612 ...
mongos.exe           08-04-2025 22:03   Application          47,035 KB
mongos.pdb           08-04-2025 22:03   PDB File             7,85,332 KB
bin                  23-04-2025 20:33   File folder
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\gauth> mongosh
Current Mongosh Log ID: 68189ced0859ba1fbdb5f898
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB:          8.0.8
Using Mongosh:          2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-04-30T16:25:19.178+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> |
```

34.Create a collection student consists of details like rollno, name, phoneno, marks, address, year of course etc.

Program:

```
test> use college
switched to db college
college> db.createCollection("student46")
{ ok: 1 }
college> db.student46.find()
```

35.Insert the details of the multiple students (atleast 5) in the form of documents in the student collection.

Program:

```
college>
db.student46.insertOne({rollno:1,name:"Navya",phoneno:8590451893,marks:94,address:"Kott
ayam",year:2024})
{
  acknowledged: true,
  insertedId: ObjectId('6818b16f0859ba1fbdb5f89e')
}
college>
db.student46.insertOne({rollno:2,name:"Paul",phoneno:8547459618,marks:90,address:"Thriss
ur",year:2024})
{
  acknowledged: true,
  insertedId: ObjectId('6818b1d70859ba1fbdb5f89f')
}
college>
db.student46.insertOne({rollno:3,name:"Sandy",phoneno:9605080027,marks:85,address:"Koll
am",year:2024})
{
  acknowledged: true,
  insertedId: ObjectId('6818b20e0859ba1fbdb5f8a0')
}
college>
db.student46.insertOne({rollno:4,name:"Abhilash",phoneno:9446491760,marks:94,address:"K
annur",year:2022})
{
  acknowledged: true,
  insertedId: ObjectId('6818b2520859ba1fbdb5f8a1')
}
college>
db.student46.insertOne({rollno:5,name:"Achu",phoneno:8547210352,marks:92,address:"Thris
sur",year:2024})
{
  acknowledged: true,
  insertedId: ObjectId('6818b2830859ba1fbdb5f8a3')
}
```

36.Retrieve the fields rollno, name, phoneno, marks, city for all the documents in the collection student.

Program:

```
college> db.student46.find()
[
  {
    _id: ObjectId('6818b16f0859ba1fbdb5f89e'),
    rollno: 1,
    name: 'Vimal',
    phoneno: 8590451893,
    marks: 94,
    address: 'Kottayam',
    year: 2024
  },
  {
    _id: ObjectId('6818b1d70859ba1fbdb5f89f'),
    rollno: 2,
    name: 'Aswathy',
    phoneno: 8547459618,
    marks: 90,
    address: 'Thrissur',
    year: 2024
  },
  {
    _id: ObjectId('6818b20e0859ba1fbdb5f8a0'),
    rollno: 3,
    name: 'Avinash',
    phoneno: 9605012327,
    marks: 85,
    address: 'Kollam',
    year: 2024
  },
  {
    _id: ObjectId('6818b2520859ba1fbdb5f8a1'),
    rollno: 4,
    name: 'Abhilash',
    phoneno: 9446491760,
    marks: 94,
    address: 'Kochi',
    year: 2022
  },
  {
    _id: ObjectId('6818b2830859ba1fbdb5f8a3'),
    rollno: 5,
    name: 'Albert',
    phoneno: 8547210352,
    marks: 92,
    address: 'Thrissur',
    year: 2024
  }
]
```

37.Display the details of students who achieved a score more than 90 and are from ‘Thrissur’.

Program:

```
college> db.student46.find({ marks: { $gt: 90 }, address: "Thrissur" })
[
 {
   _id: ObjectId('6818b2830859ba1fbdb5f8a3'),
   rollno: 5,
   name: 'Albin',
   phoneno: 8547210352,
   marks: 92,
   address: 'Thrissur',
   year: 2024
  }
]
```

38.Update the phone number of Sujith in the student collection. Retrieve the updated Information.

Program:

```
college> db.student46.updateOne({name:"Sujith"},{ $set: {phoneno:9876542130}})
college> db.student46.find({ name: "Sujith" })
[
  {
    _id: ObjectId('6818b20e0859ba1fbdb5f8a0'),
    rollno: 3,
    name: 'Sam',
    phoneno: 9876542130,
    marks: 85,
    address: 'Kollam',
    year: 2024
  }]
```

39.Update the year of course in all the documents in the student collection to 2021. Also retrieve the updated information.

Program:

```
college> db.student46.updateMany({},{$set:{year:2021}})
college> db.student46.find()
[
 {
  _id: ObjectId('6818b16f0859ba1fbdb5f89e'),
  rollno: 1,
  name: 'Ramu',
  phoneno: 8590451893,
  marks: 94,
  address: 'Kottayam',
  year: 2021
 },
 {
  _id: ObjectId('6818b1d70859ba1fbdb5f89f'),
  rollno: 2,
  name: 'Abin',
  phoneno: 8547459618,
  marks: 90,
  address: 'Thrissur',
  year: 2021
 },
 {
  _id: ObjectId('6818b20e0859ba1fbdb5f8a0'),
  rollno: 3,
  name: 'Sujith',
  phoneno: 9876542130,
  marks: 85,
  address: 'Kollam',
  year: 2021
 },
 {
  _id: ObjectId('6818b2520859ba1fbdb5f8a1'),
  rollno: 4,
  name: 'Abhilash',
  phoneno: 9446491760,
  marks: 94,
  address: 'Kannur',
  year: 2021
 },
 {
  _id: ObjectId('6818b2830859ba1fbdb5f8a3'),
  rollno: 5,
  name: 'Albin',
  phoneno: 8547210352,
  marks: 92,
  address: 'Thrissur',
  year: 2021
 }
]
```

40.Display the contact address of ‘Abhilash’.

Program:

```
college> db.student46.find({name:"Abhilash"},{_id:0,phoneno:1})
[ { phoneno: 9446491760 } ]
```

41.Delete the details of the student whose name is ‘Abhilash’ from the student collection.

Program:

```
college> db.student46.deleteOne({ name: "Abhilash" })
```

42.Retrieve the number of students per department from the student collection.

Program:

```
college> db.student46.aggregate([{$group:{_id:"$department", count:{$sum:1}}}])
[ { _id: null, count: 5 } ]
```

43.Arrange the name of the students in ascending order along with all the columns.

Program:

```
college> db.student46.find().sort({ name: 1 })
[
  {
    _id: ObjectId('6818b1d70859ba1fbdb5f89f'),
    rollno: 2,
    name: 'Arnold',
    phoneno: 8547459618,
    marks: 90,
    address: 'Thrissur',
    year: 2021
  },
  {
    _id: ObjectId('6818b2830859ba1fbdb5f8a3'),
    rollno: 5,
    name: 'Ashlin',
    phoneno: 8547210352,
    marks: 92,
    address: 'Thrissur',
    year: 2021
  },
  {
    _id: ObjectId('6818b16f0859ba1fbdb5f89e'),
    rollno: 1,
    name: 'Anu',
    phoneno: 8590451893,
    marks: 94,
    address: 'Kottayam',
    year: 2021
  },
  {
    _id: ObjectId('6818b20e0859ba1fbdb5f8a0'),
    rollno: 3,
    name: 'Subash',
    phoneno: 9876542130,
    marks: 85,
    address: 'Kottayam',
    year: 2021
  }
]
```

44.Rename city as town and add the detail of address consists of apartment no, street name and PIN.

Program:

```
college>db.student46.updateMany({},[{$set:{town:{apartment_no:"101",street:"Gandhinagar",p
in:"680001"}}}])
college> db.student46.updateMany({},{$unset:{address:""}})
college> db.student46.find({rollno:1})
```

OUTPUT

```
[ {
    _id: ObjectId('6818b16f0859ba1fbdb5f89e'),
    rollno: 1,
    name: Anu,
    phoneno: 8590451893,
    marks: 94,
    year: 2021,
    town: { apartment_no: '101', street: 'Gandhinagar', pin: '680001' }
  }]
```