****

**Name –PRATHMESH PATIL**

**AIIT BCA SEM-III**

**A71004819033**

**Database Management Systems LAB**

**DBMS LAB INDEX-2020**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No** | **Practical Name** | **Date** | **Page no.** | **Remarks** |
| 1 | DDL Commands(Create,Alter, Truncate,Drop, Rename) | 4/8/20 | 4 |  |
| 2 | Data Manipulation Language (DML) Commands(insert, update, delete, select) |  | 8 |  |
| 3 | Data Control Languages (DCL-grant, revoke) and Transaction Controls Languages (TCL-rollback ,commit) | 11/08 | 11 |  |
| 4 | Constraints(Primary key, foreign key, not null, check and default) |  | 14 |  |
| 5 | SQL Functions:   1. Aggregate Functions 2. Numeric Functions 3. String Functions | 18/08 | 20 |  |
| 6 | SQL Operators:   1. Arithmetic operator 2. Logical operators(in,between and, Any, any 3. Like operator |  | 35 |  |
| 7 | Clause:   1. Group by 2. Having count 3. Order by 4. Distinct | 01/09 | 40 |  |
| 8 | SQL subqueries | 15/09 | 63 |  |
| 9 | SQL joins | 22/09 | 66 |  |
| 10 | Views, Index | 29/09 | 72 |  |
| 11 | PL/SQL   1. Write a PL/SQL block to accept the value from the user and display it. 2. Write a pl/sql block to perform addition of 2 nos 3. Write the pl/sql block to find the sum of first 100 natural number 4. Write a pl/sql block to display the information of given students using the following table. Stud(sno,snmae,address,city) 5. Write a pl/sql block to raise the salary by 20% of a given employee on the following table Emp\_salary(eno,ename,city,salary) 6. Write a PL-SQL code to define parameterized cursor that accept department name from the user & display all employees of that dept. 7. Write an Implicit Cursor to accept the employee number from the user. You have to delete this record and display the appropriate message on the following table. Emp (eno, ename, address, city) 8. Create trigger on insert,update or delete to display the salary difference 9. Write a trigger on insert to convert the name into capital letters. | 20/10  27/10  03/11  10/11  24/11 | 81 |  |

**PRACTICAL NO .1**

**Aim:-** To perform DDL commands.

**1) Data Definition Language (DDL) Commands: -**

**Description:**

1.Create: -

To create a new table or database With attribute name and data type.

2.Alter: -

Alter table is used to alter add, delete or modify columns in an existing table.

3.Truncate: -

Truncate command is used to delete complete data from an existing table.

4.Drop: -

Delete entire structure of the table.

5.Rename: -

Rename command is used to give a new name to the existing table.

**2) Data Manipulation Language (DML) Commands: -**

**Description:**

1.Insert: -

Insert command is used to insert the values into the table.

2.Update: -

Update query is used to modify the existing records in a table. You can use var clause in update query to update the selected rows, otherwise all roles will be updated.

e.g:update customers set address=’Pune’ where id=6

If you want to modify address and salary column

e.g:update customers address=’Pune’, salary=50000

3.Delete: -

Delete query is used to delete existing records from a table. You can use var clause with the delete query to delete selected rows, otherwise all records will be deleted.

e.g delete from customers where id=6

4.Select: -

Select statement is used to fetch the data from a database

e.g:select sid,address from student

ASSIGNMENT 1 ON DDL and DML

1. Create table student with sid, lname, fname and mobile with proper data types.
2. Insert 5 values to the table student.
3. Display the content of the table student.
4. Add column DOB as date to the existing table student using alter command.
5. Update table student, add DOB where sid=1,2 & 3.
6. Drop the column DOB.
7. Delete data from student where sid=5 & display it.
8. Truncate table student that is to delete entire data from the table student.
9. Rename student to student1.
10. Alter table student1 rename column lname to last name.
11. Alter the table student and modify the datatype of fname as varchar(30).

1.Create table student with sid, lname, fname and mobile with proper data types.

Ans- create table student(sid int, lname varchar(15), fname varchar(15), mobile number(10));

Response:

Table created.

2.Insert 5 values to the table student.

Ans- insert into student values(1,'Patil','Prathmesh',7208055391);

insert into student values(2,'Patil','Monu',8652184656);

insert into student values(3,'Patil','Kaustubh',9594448714);

insert into student values(4,'Patil','Sohal',8108040556);

insert into student values(5,'Patil','Anurag',3948273922);

Response:

1. row(s) inserted. 1 row(s) inserted. 1 row(s) inserted. 1 row(s) inserted. 1 row(s) inserted.

3.Display the content of the table student.

Ans-

Response-

|  |  |  |  |
| --- | --- | --- | --- |
| **SID** | **LNAME** | **FNAME** | **MOBILE** |
| 1 | Patil | Prathmesh | 7208155391 |
| 2 | Patil | Monu | 8652184656 |
| 3 | Patil | Kaustubh | 9594448714 |
| 4 | Patil | Sohal | 8108040556 |
| 5 | Patil | Anurag | 3948273922 |
|  |  |  |  |

4.Add column DOB as date to the existing table student using alter command.

Ans-alter table student add DOB date;

Response-

Table altered.

5.Update table student, add DOB where sid=1,2 & 3.

Ans- update student set DOB = '01-June-2001' where sid = 1;

update student set DOB = '06-April-2000' where sid = 2;

update student set DOB = '03-Dec-2000' where sid = 3;

Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **LNAME** | **FNAME** | **MOBILE** | **DOB** |
| **1** | Patil | **Prathmesh** | **7208155391** | **01-June-2001** |
| **2** | Patil | **Monu** | **8652184656** | **06-April-2000** |
| **3** | Patil | **Kaustubh** | **9594448714** | **03-Dec-2000** |
| **4** | Patil | **Sohal** | **8108040556** | **-** |
| **5** | Patil | **Anurag** | 3948273922 | **-** |
|  |  |  |  |  |

6.Drop the column DOB.

Ans- alter table student drop column DOB;

Response-

Table altered.

7.Delete data from student where sid=5 & display it.

Ans- delete from student where sid =5;

Response-

1. row(s) deleted.

8.Truncate table student that is to delete entire data from the table student.

Ans- truncate table student;

Response- Table truncated.

9.Rename student to student1.

Ans- alter table student rename to student1;

Response- Table altered.

10.Alter table student1 rename column lname to last name.

Ans- alter table student1 rename column lname to last\_name;

Response- Table altered.

11.Alter the table student and modify the datatype of fname as varchar(30).

Ans- alter table student1 modify fname varchar(30);

Response-  
Table altered.

FINAL OUTPUT-

|  |  |  |  |
| --- | --- | --- | --- |
| **SID** | **LAST\_NAME** | **FNAME** | **MOBILE** |
| 1 | Patil | Prathmesh | 7208155391 |
| 2 | Patil | Monu | 8652184656 |
| 3 | Patil | Kaustubh | 9594448714 |
| 4 | Patil | Sohal | 8108040556 |

**PRACTICAL NO .2**

**Aim:-** To perform DML commands.

**2) Data Manipulation Language (DML) Commands: -**

**Description:**

1.Insert: -

Insert command is used to insert the values into the table.

2.Update: -

Update query is used to modify the existing records in a table. You can use var clause in update query to update the selected rows, otherwise all roles will be updated.

e.g:update customers set address=’Pune’ where id=6

If you want to modify address and salary column

e.g:update customers address=’Pune’, salary=50000

3.Delete: -

Delete query is used to delete existing records from a table. You can use var clause with the delete query to delete selected rows, otherwise all records will be deleted.

e.g delete from customers where id=6

4.Select: -

Select statement is used to fetch the data from a database

e.g:select sid,address from student

1.Create table student with sid, lname, fname and mobile with proper data types.

Ans- create table student(sid int, lname varchar(15), fname varchar(15), mobile number(10));

Response:

Table created.

2.Insert 5 values to the table student.

Ans- insert into student values(1,'Patil','Prathmesh',7208055391);

insert into student values(2,'Patil','Monu',8652184656);

insert into student values(3,'Patil','Kaustubh',9594448714);

insert into student values(4,'Patil','Sohal',8108040556);

insert into student values(5,'Patil','Anurag',3948273922);

Response:

1. row(s) inserted. 1 row(s) inserted. 1 row(s) inserted. 1 row(s) inserted. 1 row(s) inserted.

3.Display the content of the table student.

Ans-

Response-

|  |  |  |  |
| --- | --- | --- | --- |
| **SID** | **LNAME** | **FNAME** | **MOBILE** |
| 1 | Patil | Prathmesh | 7208155391 |
| 2 | Patil | Monu | 8652184656 |
| 3 | Patil | Kaustubh | 9594448714 |
| 4 | Patil | Sohal | 8108040556 |
| 5 | Patil | Anurag | 3948273922 |
|  |  |  |  |

4.Add column DOB as date to the existing table student using alter command.

Ans-alter table student add DOB date;

Response-

Table altered.

5.Update table student, add DOB where sid=1,2 & 3.

Ans- update student set DOB = '01-June-2001' where sid = 1;

update student set DOB = '06-April-2000' where sid = 2;

update student set DOB = '03-Dec-2000' where sid = 3;

Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **LNAME** | **FNAME** | **MOBILE** | **DOB** |
| **1** | Patil | **Prathmesh** | **7208155391** | **01-June-2001** |
| **2** | Patil | **Monu** | **8652184656** | **06-April-2000** |
| **3** | Patil | **Kaustubh** | **9594448714** | **03-Dec-2000** |
| **4** | Patil | **Sohal** | **8108040556** | **-** |
| **5** | Patil | **Anurag** | 3948273922 | **-** |
|  |  |  |  |  |

6.Drop the column DOB.

Ans- alter table student drop column DOB;

Response-

Table altered.

7.Delete data from student where sid=5 & display it.

Ans- delete from student where sid =5;

Response-

1. row(s) deleted.

8.Truncate table student that is to delete entire data from the table student.

Ans- truncate table student;

Response- Table truncated.

9.Rename student to student1.

Ans- alter table student rename to student1;

Response- Table altered.

10.Alter table student1 rename column lname to last name.

Ans- alter table student1 rename column lname to last\_name;

Response- Table altered.

11.Alter the table student and modify the datatype of fname as varchar(30).

Ans- alter table student1 modify fname varchar(30);

Response-  
Table altered.

FINAL OUTPUT-

|  |  |  |  |
| --- | --- | --- | --- |
| **SID** | **LAST\_NAME** | **FNAME** | **MOBILE** |
| 1 | Patil | Prathmesh | 7208155391 |
| 2 | Patil | Monu | 8652184656 |
| 3 | Patil | Kaustubh | 9594448714 |
| 4 | Patil | Sohal | 8108040556 |

**Practical no 3**

**Aim:** Data Control Languages (DCL) and Transaction Controls Languages (TCL)

**Description:**

**1.Data control languages**

Dcl commands are used to enforce database security in a multiple user environment

Commands used in dcl are

**A.Grant**

grant command is used to assign permission to the user such as insert, delete, create and select.

e.g,grant create table to username

grant drop any table to username

grant create any table to username

e.g. first we need to create the user with particular password then we have to assign the permission or privileges to the user created.

To create a new user the command is create user username

Create user smita

Identified by s1;

We need to grant create session to smita

Command:grant create session, grant any privileges to smita

e.g.grant select on student to smita;

this command grants a select permission on student table to smita

grant create any table to smita

**2.TCL Commands**

TCL commands are used to manage transactions in the database.

Command are

**1.Commit command:**

Commit command is used to permanently save any transaction in the database

e.g. desc student;

select \* from student;

commit;

**2.Rollback command**

This command restores the database to last commited state.

**Code :-**

**1)** Querry> alter session set "\_ORACLE\_SCRIPT"=true

Ans:-

SQL> alter session set "\_ORACLE\_SCRIPT"=true

2 ;

Session altered.

**2)** Querry> create user Amity identified by Prathmesh;

Ans:-

SQL> create user Amity identified by Prathmesh;

User created.

**3)** Querry> grant create session to Amity;

Ans:-

SQL> grant create session to Amity;

Grant succeeded.

**4)** Querry> create table true(id int,lname varchar(15));

Ans:-

SQL> create table true(id int,lname varchar(15));

Table created.

**5)** Querry> revoke create table from Amity;

Ans:-

SQL> revoke create table from Amity

2 ;

Revoke succeeded.

**6**) Querry> drop user Amity cascade;

Ans:-

SQL> drop user Amity ccascade;

User dropped.

**7)** Querry> create table true(id int,lname varchar(15));

Ans:-

SQL> create table true(id int,lname varchar(15));

Table created.

**8)** Querry> insert into tues values(1,'Prathmesh');

Querry> select \* from true;

Ans:-

QL> create table true(id int,lname varchar(15));

Table created.

SQL> insert into true values(1,'Prathmesh');

1 row created.

SQL> select \* from true;

ID LNAME

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

1 Prathmesh

**9)** Querry> rollback;

Querry> select \* from true;

Ans:-

SQL> rollback;

Rollback complete.

SQL> select \* from true;

no rows selected

**10)** Querry> insert into true values(1,'Prathmesh');

Querry> commit;

Querry> rollback;

Querry> select \* from true;

Ans:-

SQL> insert into true values(1,'Prathmesh');

1 row created.

SQL> commit;

Commit complete.

SQL> rollback;

Rollback complete.

SQL> select \* from true;

ID LNAME

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

1 Prathmesh

**Practical no 4**

**Aim:-** To create table student with reg\_no as a primary key and fname , lname ,dob, address, major as the other attributes and perform queries.

**Theory**:-

SQL constraints are used to specify rules for the table. IF there is any violation between the constraints and the data action, the action is aborted by the constraint.

Constraints can be specified when the table is created i.e. is inside the create table statement or after the table is created i.e. inside the alter table statement.

Following are the constraints:

1.**Not NULL**- Indicates that a column cannot store NULL values.

2.**Unique** – Ensure that each rows for a column must have a unique value. E.g id-1,2,3

3.**Primary Key**- A combination of a Not NULL and Unique constraint is called Primary Key.

4.**Foreign Key**-Ensure the **referential integrity** of the data in one table to match values in other table.

5.**Check**-Ensure that the value in a column meets a specific condition for e.g sid > 5.

6.**Default**-Specifies a default value when specified nothing for this column.

**Code:-**

Querry:- create table student1(reg\_no primary key,fname,lname ,address,dob ,major );

Ans:-

SQL> create table student1(reg\_no int primary key,fname varchar2(5),lname varchar2(10),address varchar2(50),dob date,major varchar2(10));

Table created.

Querry:- alter table student drop primary key;

Ans:-

SQL> alter table student drop primary key;

Table altered.

Querry:- desc student

Ans:-

SQL> desc student

Name Null? Type

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

REG\_NO NUMBER(38)

FNAME VARCHAR2(5)

LNAME VARCHAR2(10)

ADDRESS VARCHAR2(50)

DOB DATE

MAJOR VARCHAR2(10)

Querry:- alter table student add primary key(reg\_no);

Ans:-

SQL> alter table student add primary key(reg\_no);

Table altered.

SQL> desc student

Name Null? Type

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

REG\_NO NOT NULL NUMBER(38)

FNAME VARCHAR2(5)

LNAME VARCHAR2(10)

ADDRESS VARCHAR2(50)

DOB DATE

MAJOR VARCHAR2(10)

Querry:- alter table student modify reg\_no int check(reg\_no>5);

Ans:-

SQL> alter table student modify reg\_no int check(reg\_no>5);

Table altered.

SQL> desc student

Name Null? Type

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

REG\_NO NOT NULL NUMBER(38)

FNAME VARCHAR2(5)

LNAME VARCHAR2(10)

ADDRESS VARCHAR2(50)

DOB DATE

MAJOR VARCHAR2(10)

SQL> alter table student drop primary key;

Table altered.

SQL> desc student

Name Null? Type

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

REG\_NO NUMBER(38)

FNAME VARCHAR2(5)

LNAME VARCHAR2(10)

ADDRESS VARCHAR2(50)

DOB DATE

MAJOR VARCHAR2(10)

SQL> insert into student2 values(1,'Monu','Patil','CBD sec-19','01 June 2001','BCA');

1 row created.

SQL> insert into student2 values(2,'Sohal','Patil','Belapur sec-20','19 June 2001','BTECH');

1 row created.

SQL> insert into student2 values(1,'Binod','Yt','Vashi sec-10','23 April 2000','BMS');

1 row created.

SQL> insert into student2 values(1,'Rashi','Behn','Mumbai sec-1','03 Aug 2020','CHANE');

1 row created.

SQL> insert into student2 values(1,'Gopi','Behn','Kurla sec-15','03 Aug 2020','COOKER');

1 row created.

SQL> select \* from student2;

REG\_NO FNAME LNAME ADDRESS DOB MAJOR

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

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

1 Monu Patil CBD sec-19 01-JUN-01 BCA

2 Sohal Patil Belapur sec-20 19-JUN-01 BTECH

1 Binod Yt Vashi sec-10 23-APR-00 BMS

REG\_NO FNAME LNAME ADDRESS DOB MAJOR

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

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

1 Rashi Behn Mumbai sec-1 03-AUG-20 CHANE

1 Gopi Behn Kurla sec-15 03-AUG-20 COOKER

1.Create table enroll with reg\_no as a foreign key references to student,course\_no,semester,marks,enroll\_date.

2.Apply reg\_no, course\_no and semester as composite unique key.

3.Add course\_name in enroll table and give the default value as MCA to course\_name.

Code :-

SQL> Create table student(reg\_no int primary key,fname varchar(10),lname varchar(10),address varchar(10));

Table created.

SQL> Create table course(id varchar(10) primary key,lname varchar(10),

2 fees int,duration int,reg\_no int,foreign key(reg\_no) references student(reg\_no));

Table created.

SQL> insert into studentt values(1,'Prathmesh','Patil','Belapur');

1 row created.

SQL> insert into studentt values(2,'Kuldeep','Singhvi','Panvel');

1 row created.

SQL> insert into studentt values(3,'Kaustubh',’Patil',’mumbai');

1 row created.

SQL> insert into studentt values(4,'Anshul','Sinha','mumbai');

1 row created.

SQL> insert into studentt values(5,'Boom','Obsi','mumbai');

1 row created.

SQL> insert into course values(101,'BCA',11890,1,2);

1 row created.

SQL> insert into course values(103,'BCOM',11891,3,4);

1 row created.

SQL> select \* from course;

ID NAME FEES DURATION REG\_NO

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

101 BCA 11890 1 2

103 BCOM 11891 3 4

SQL> select \* from course;

ID NAME FEES DURATION REG\_NO

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

101 BCA 11890 1 2

103 BCOM 11891 3 4

SQL> insert into course values(105,'BCOM',11892,4,5);

1 row created.

SQL> insert into studentt values(6,'Patil','Prathmesh','Thane');

1 row created.

SQL> insert into course values(107,'MCA',11893,5,6);

1 row created.

SQL> select \* from course;

ID NAME FEES DURATION REG\_NO

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

101 BCA 11890 1 2

103 BCOM 11891 3 4

105 BCOM 11892 4 5

107 MCA 11893 5 6

SQL> create table enroll(reg\_no int,course\_no int,semester varchar(10),

2 marks int,enroll\_date date,foreign key(reg\_no)

3 references studentt(reg\_no));

Table created.

SQL> enroll desc

SP2-0734: unknown command beginning "enroll des..." - rest of line ignored.

SQL> desc enroll

Name Null? Type

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

REG\_NO NUMBER(38)

COURSE\_NO NUMBER(38)

SEMESTER VARCHAR2(10)

MARKS NUMBER(38)

ENROLL\_DATE DATE

SQL> delete enroll;

0 rows deleted.

SQL> create table enroll1(reg\_no int,course\_no int,semester varchar(10),

2 marks int default '0',enroll\_date date,foreign key(reg\_no)

3 references studentt(reg\_no));

Table created.

SQL> insert into enroll1 values(1,101,1,70 ,'03-AUG-2018');

1 row created.

SQL> insert into enroll1 values(2,102,1,80 ,'03-AUG-2018');

1 row created.

SQL> insert into enroll1 values(3,103,1,90 ,'03-AUG-2018');

1 row created.

SQL> insert into enroll1 values(4,104,'1',100,'03-AUG-2018');

1 row created.

SQL> alter table enroll1 add course\_name varchar(15);

Table altered.

**Practical no 5**

**Aim:-** to use inbuilt functions , SQL Functions:

1. Aggregate Functions
2. Numeric Functions
3. String Functions

Theory:-

1. Aggregate Functions:-

An SQL aggregate function calculates on a set of values and returns a single value. For example, the average function ( AVG) takes a list of values and returns the average.

Because an aggregate function operates on a set of values, it is often used with the [GROUP BY](https://www.sqltutorial.org/sql-group-by/) clause of the [SELECT](https://www.sqltutorial.org/sql-select/) statement. The GROUP BY clause divides the result set into groups of values and the aggregate function returns a single value for each group.

The following illustrates how the aggregate function is used with the [GROUP BY](https://www.sqltutorial.org/sql-group-by/) clause:

**SELECT** c1, aggregate\_function(c2)

**FROM** **table**

**GROUP** **BY** c1;

The following are the commonly used SQL aggregate functions:

* [AVG()](https://www.sqltutorial.org/sql-aggregate-functions/sql-avg/) – returns the average of a set.
* [COUNT()](https://www.sqltutorial.org/sql-aggregate-functions/sql-count/) – returns the number of items in a set.
* [MAX()](https://www.sqltutorial.org/sql-max/) – returns the maximum value in a set.
* [MIN()](https://www.sqltutorial.org/sql-aggregate-functions/sql-min/) – returns the minimum value in a set
* [SUM()](https://www.sqltutorial.org/sql-aggregate-functions/sql-sum/) – returns the sum of all or distinct values in a set

Except for the COUNT() function, SQL aggregate functions ignore null.

You can use aggregate functions as expressions only in the following:

* The select list of a [SELECT](https://www.sqltutorial.org/sql-select/) statement, either a subquery or an outer query.
* A [HAVING](https://www.sqltutorial.org/sql-having/) clause

**Code:-**

create table customer1(id int,name varchar(20),address varchar(20),age number,salary number)

Table created.

insert into customer1 values(1,'Prathmesh','Mumbai','20','40000')

1 row(s) inserted.

insert into customer1 values(2,'Ninad','Mumbai','20','45000')

1 row(s) inserted.

insert into customer1 values(3,'Anshul','Dehli','21','50000')

1 row(s) inserted.

insert into customer1 values(4,'Kuldeep','Dehli','20','55000')

1 row(s) inserted.

insert into customer1 values(5,'Akanksha','Nashik','19','60000')

1 row(s) inserted.

insert into customer1 values(6,'Prathmesh','Nashik','22','65000')

1 row(s) inserted.

select \* from customer1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NAME** | **ADDRESS** | **AGE** | **SALARY** |
| 1 | Prathmesh | Mumbai | 20 | 40000 |
| 2 | Ninad | Mumbai | 20 | 45000 |
| 3 | Anshul | Dehli | 21 | 50000 |
| 4 | Kuldeep | Dehli | 20 | 55000 |
| 5 | Akanksha | Nashik | 19 | 60000 |
| 6 | Prathmesh | Nashik | 22 | 65000 |

6 rows selected.

select count(id) from customer1 group by age

|  |
| --- |
| **COUNT(ID)** |
| 1 |
| 3 |
| 1 |
| 1 |

4 rows selected.

select count(salary) from customer1 group by name

|  |
| --- |
| **COUNT(SALARY)** |
| 1 |
| 2 |
| 1 |
| 1 |
| 1 |

5 rows selected.

select count(age) from customer1 group by salary

|  |
| --- |
| **COUNT(AGE)** |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |

6 rows selected.

select count(name) from customer1 group by address

|  |
| --- |
| **COUNT(NAME)** |
| 2 |
| 2 |
| 2 |

3 rows selected.

select sum(salary) from customer1 group by age

|  |
| --- |
| **SUM(SALARY)** |
| 50000 |
| 140000 |
| 65000 |
| 60000 |

4 rows selected.

select name,sum(salary) from customer1 group by name

|  |  |
| --- | --- |
| **NAME** | **SUM(SALARY)** |
| Ninad | 45000 |
| Prathmesh | 105000 |
| Kuldeep | 55000 |
| Akanksha | 60000 |
| Anshul | 50000 |

5 rows selected.

select avg(salary) from customer1 group by salary

|  |
| --- |
| **AVG(SALARY)** |
| 55000 |
| 65000 |
| 45000 |
| 60000 |
| 40000 |
| 50000 |

6 rows selected.

select avg(salary) from customer1 group by address

|  |
| --- |
| **AVG(SALARY)** |
| 52500 |
| 62500 |
| 42500 |

3 rows selected.

select sum(salary)/count(id),address from customer1 group by address

|  |  |
| --- | --- |
| **SUM(SALARY)/COUNT(ID)** | **ADDRESS** |
| 52500 | Dehli |
| 62500 | Nashik |
| 42500 | Mumbai |

3 rows selected.

select sum(salary)/count(id),name from customer1 group by name

|  |  |
| --- | --- |
| **SUM(SALARY)/COUNT(ID)** | **NAME** |
| 45000 | Ninad |
| 52500 | Prathmesh |
| 55000 | Kuldeep |
| 60000 | Akanksha |
| 50000 | Anshul |

5 rows selected.

select sum(salary) from customer1 group by age having count(age) >= 2

|  |
| --- |
| **SUM(SALARY)** |
| 140000 |

select min(salary) from customer1

|  |
| --- |
| **MIN(SALARY)** |
| 40000 |

select min(age) from customer1

|  |
| --- |
| **MIN(AGE)** |
| 19 |

select min(id) from customer1

|  |
| --- |
| **MIN(ID)** |
| 1 |

select min(address) from customer1

|  |
| --- |
| **MIN(ADDRESS)** |
| Dehli |

select max(salary) from customer1

|  |
| --- |
| **MAX(SALARY)** |
| 65000 |

select max(age) from customer1

|  |
| --- |
| **MAX(AGE)** |
| 22 |

select max(id) from customer1

|  |
| --- |
| **MAX(ID)** |
| 6 |

select max(address) from customer1

|  |
| --- |
| **MAX(ADDRESS)** |
| Nashik |

select avg(salary) from customer1

|  |
| --- |
| **AVG(SALARY)** |
| 52500 |

select avg(age) from customer1

|  |
| --- |
| **AVG(AGE)** |
| 20.33333333333333333333333333333333333333 |

select count(id) from customer1

|  |
| --- |
| **COUNT(ID)** |
| 6 |

select count(salary) from customer1

|  |
| --- |
| **COUNT(SALARY)** |
| 6 |

select count(age) from customer1

|  |
| --- |
| **COUNT(AGE)** |
| 6 |

select sum(salary) from customer1

|  |
| --- |
| **SUM(SALARY)** |
| 315000 |

select sum(age) from customer1

|  |
| --- |
| **SUM(AGE)** |
| 122 |

select sum(id) from customer1

|  |
| --- |
| **SUM(ID)** |
| 21 |

select abs(-23.3)absolute\_value from dual

|  |
| --- |
| **ABSOLUTE\_VALUE** |
| 23.3 |

select sqrt(36)from dual

|  |
| --- |
| **SQRT(36)** |
| 6 |

select round(20.045,2)from dual

|  |
| --- |
| **ROUND(20.045,2)** |
| 20.05 |

select exp(4)from dual

|  |
| --- |
| **EXP(4)** |
| 54.59815003314423907811026120286087840308 |

select power(2,2)from dual

|  |
| --- |
| **POWER(2,2)** |
| 4 |

select mod(5,2)from dual

|  |
| --- |
| **MOD(5,2)** |
| 1 |

1. **Numeric Functions**

It performs mathematical operations on numerical data type or returns numeric information.

Dual is as system table which return only 1 value i.e. 1 row and 1 column

Functions are,

**1.ABS()**

It returns absolute value of a specific number.

**2.Power()**

Returns the value of a numeric expression to the power of integer expression.

**3.Round()**

Returns the numeric expression rounded to the decimal point.

**4.SQRT()**

Returns the square root of a float expression

**5.exp()**

Returns the exponential value of float expression.

**6.Mod()**

Returns the reminder of integer.

Greatest(m1,m2,m3)

Least(m1,m2,m3

**1.**

SQL> select abs(-20.4)absolute\_value from dual;

ABSOLUTE\_VALUE

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

20

**2.**

SQL> select sqrt(4)from dual;

SQRT(4)

----------

2

**3.**

SQL> select round(20.045,2) from dual;

ROUND(20.045,2)

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

20.05

**4.**

SQL> select exp(4) from dual;

EXP(4)

----------

54.59815

**5.**

SQL> select power(2,2) from dual;

POWER(2,2)

----------

4

**6.**

SQL> select mod(5,2) from dual;

MOD(5,2)

----------

1

**7.**

SQL> select greatest(m1,m2,m3) from student1;

GREATEST(M1,M2,M3)

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

77

76

85

88

89

**8.**

SQL> select least(m1,m2,m3)marks from student1;

MARKS

----------

69

65

50

49

69

**9.**

SQL> select trunc(123.815,1) from dual;

TRUNC(123.815,1)

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



1. **String Function:-**

**Code:-**

SQL> create table customer1(id int primary key,name varchar(20),address varchar(20),age number,salary number);

Table created.

SQL> insert into customer1 values(1,'Prathmesh','Mumbai','20','40000');

1 row created.

SQL> insert into customer1 values(2,'Ninad','Mumbai','20','45000');

1 row created.

SQL> insert into customer1 values(3,'Anshul','Dehli','21','50000');

1 row created.

SQL> insert into customer1 values(4,'Kuldeep','Dehli','20','55000');

1 row created.

SQL> insert into customer1 values(5,'Akanksha','Nashik','19','60000');

1 row created.

SQL> insert into customer1 values(6,'Prathmesh','Nashik','22','65000');

1 row created.

SQL> select \* from customer1;

ID NAME ADDRESS AGE SALARY

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

1 Prathmesh Mumbai 20 40000

2 Ninad Mumbai 20 45000

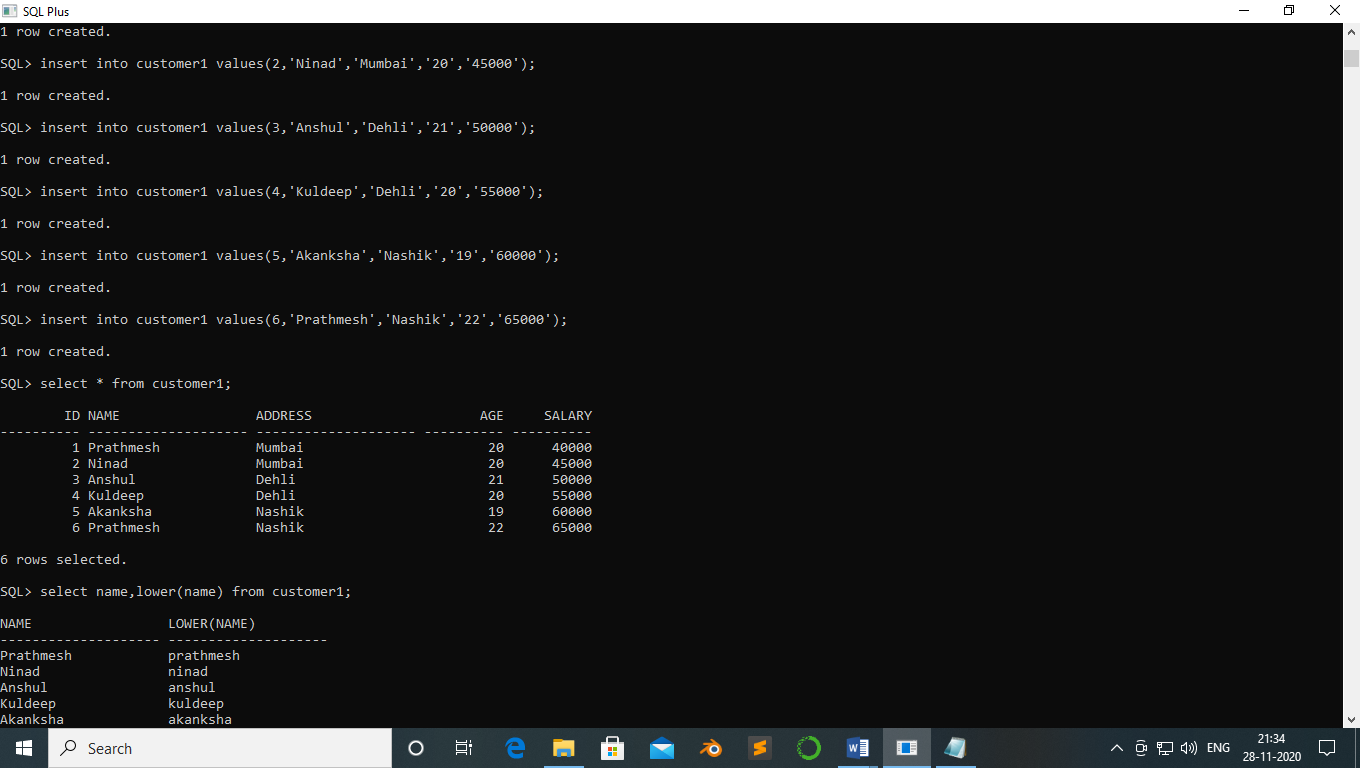
3 Anshul Dehli 21 50000

4 Kuldeep Dehli 20 55000

5 Akanksha Nashik 19 60000

6 Prathmesh Nashik 22 65000

6 rows selected.



SQL> select name,lower(name) from customer1;

NAME LOWER(NAME)

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

Prathmesh prathmesh

Ninad ninad

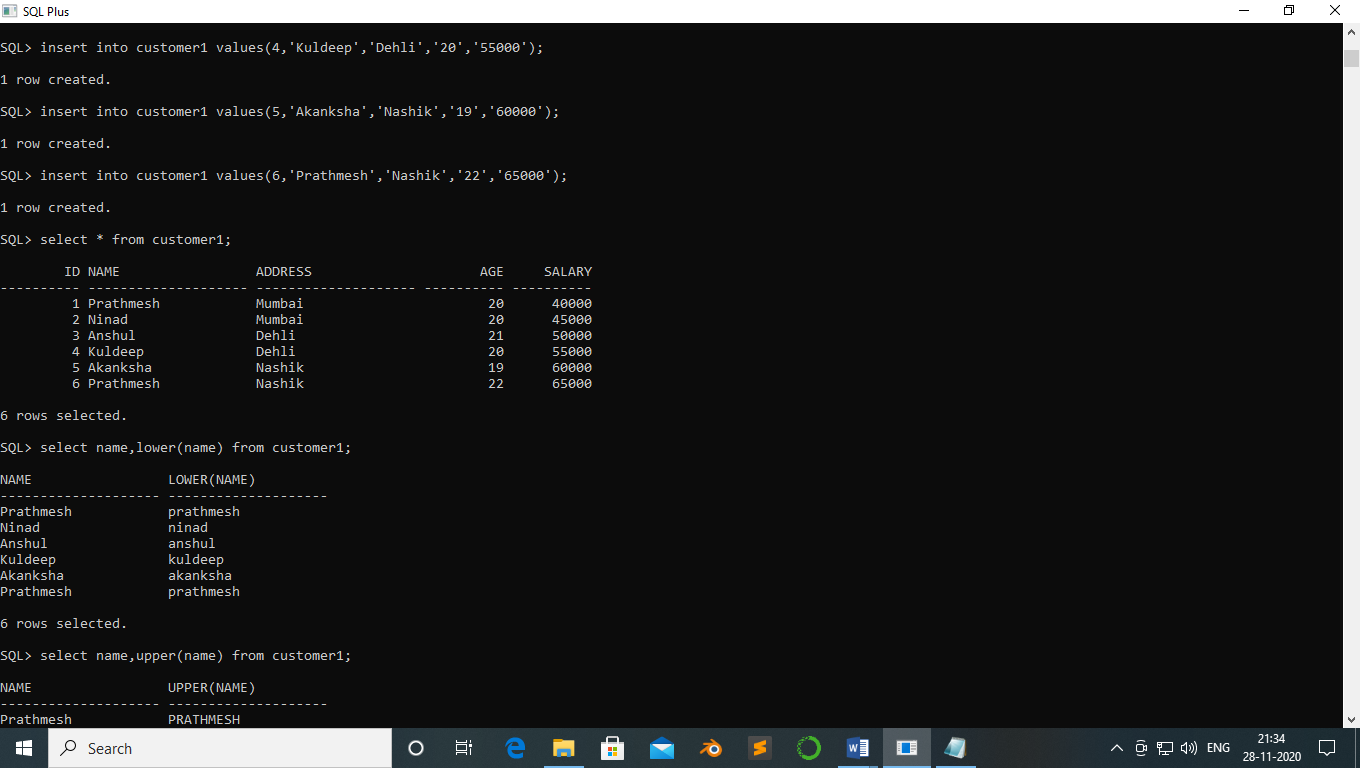
Anshul anshul

Kuldeep kuldeep

Akanksha akanksha

Prathmesh prathmesh

6 rows selected.



SQL> select name,upper(name) from customer1;

NAME UPPER(NAME)

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

Prathmesh PRATHMESH

Ninad NINAD

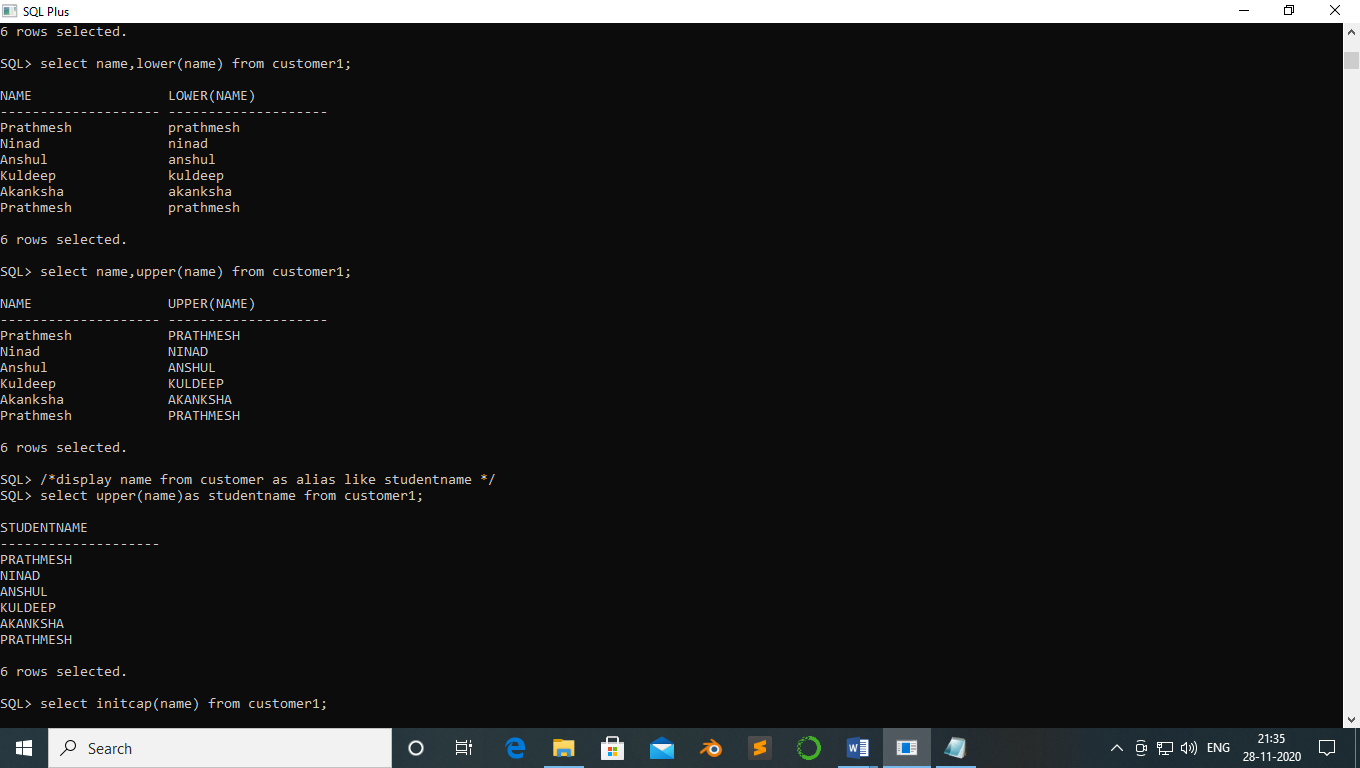
Anshul ANSHUL

Kuldeep KULDEEP

Akanksha AKANKSHA

Prathmesh PRATHMESH

6 rows selected.



SQL> /\*display name from customer as alias like studentname \*/

SQL> select upper(name)as studentname from customer1;

STUDENTNAME

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

PRATHMESH

NINAD

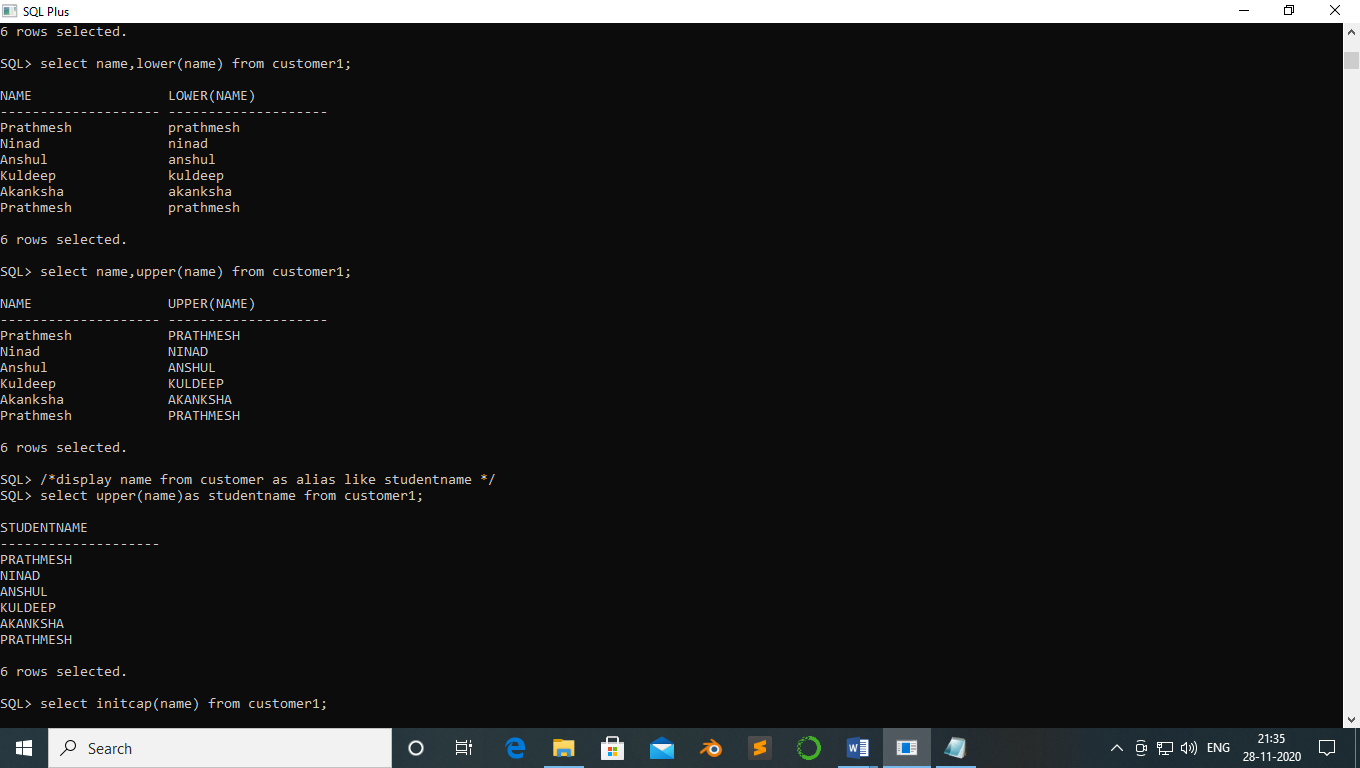
ANSHUL

KULDEEP

AKANKSHA

PRATHMESH

6 rows selected.



SQL> select initcap(name) from customer1;

INITCAP(NAME)

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

Prathmesh

Ninad

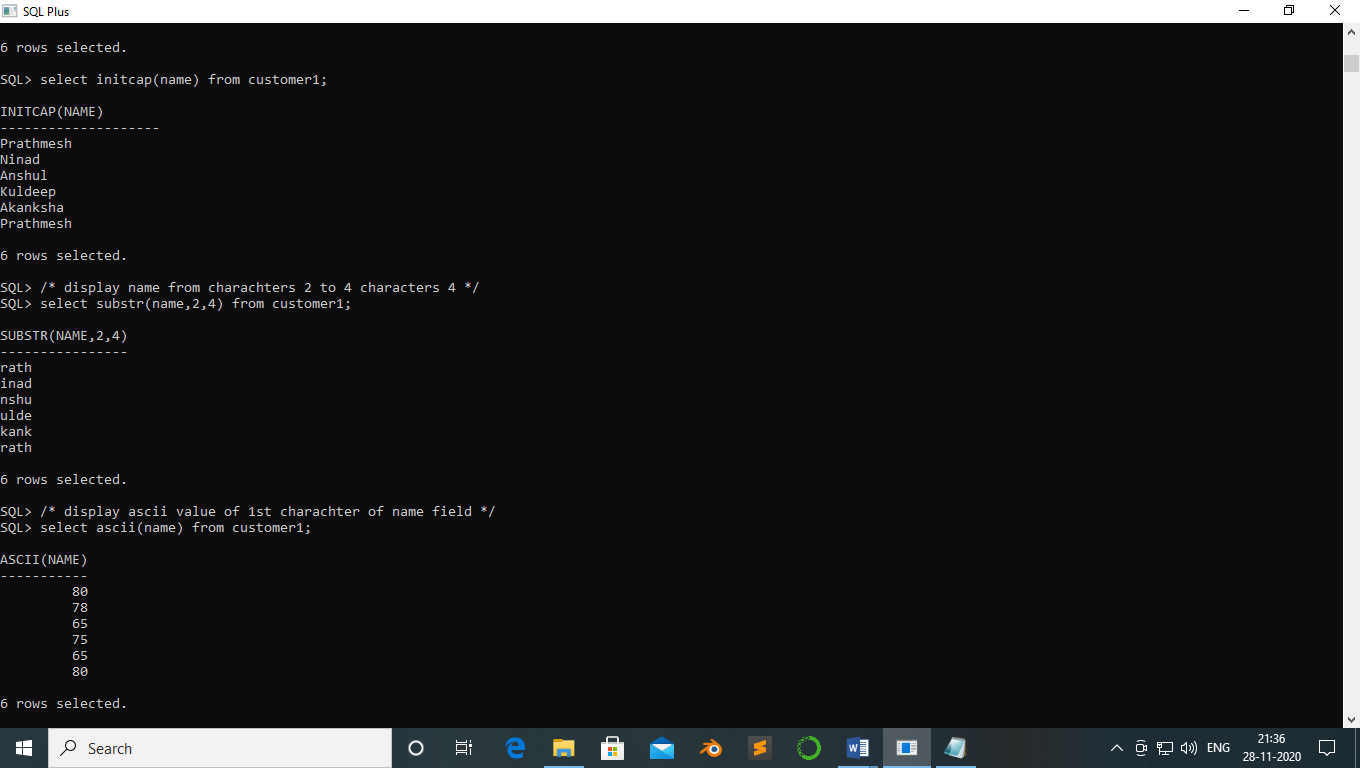
Anshul

Kuldeep

Akanksha

Prathmesh

6 rows selected.



SQL> /\* display name from charachters 2 to 4 characters 4 \*/

SQL> select substr(name,2,4) from customer1;

SUBSTR(NAME,2,4)

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

rath

inad

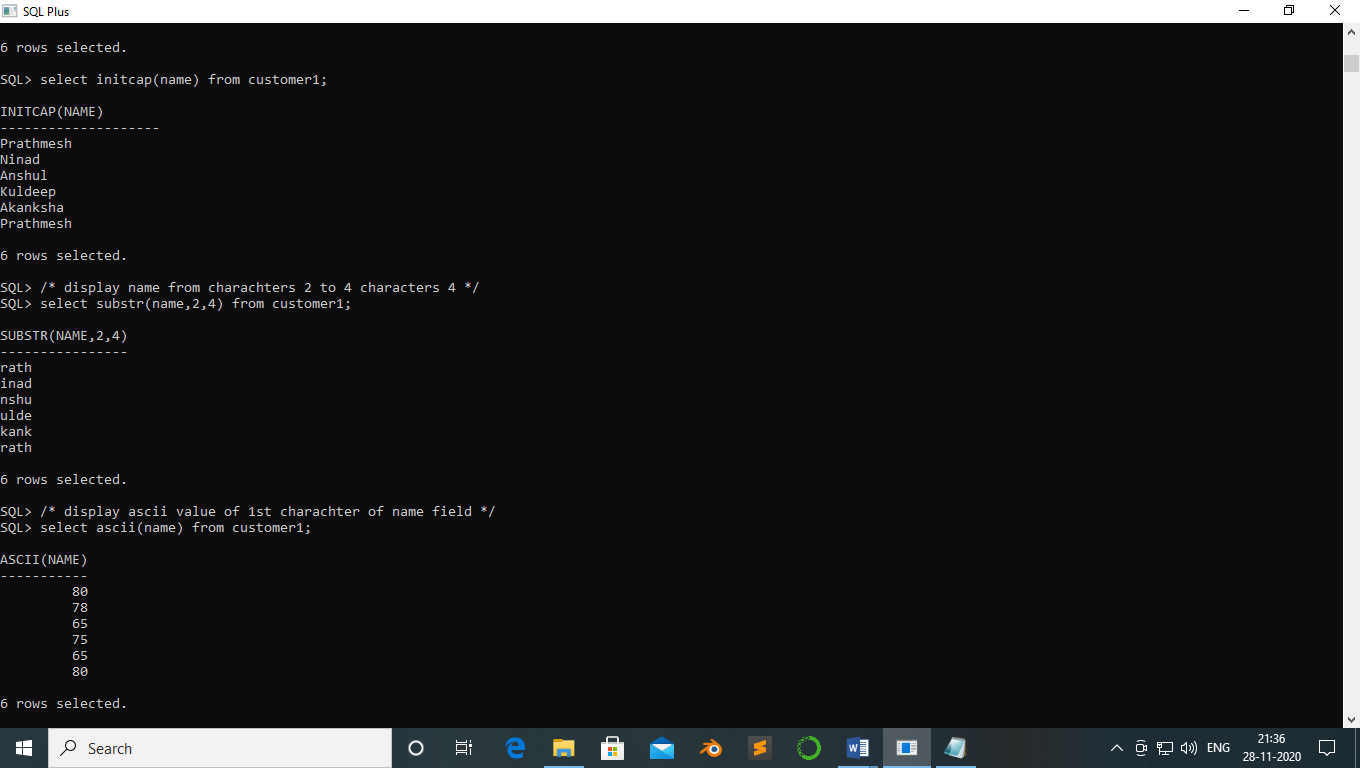
nshu

ulde

kank

rath

6 rows selected.



SQL> /\* display ascii value of 1st charachter of name field \*/

SQL> select ascii(name) from customer1;

ASCII(NAME)

-----------

80

78

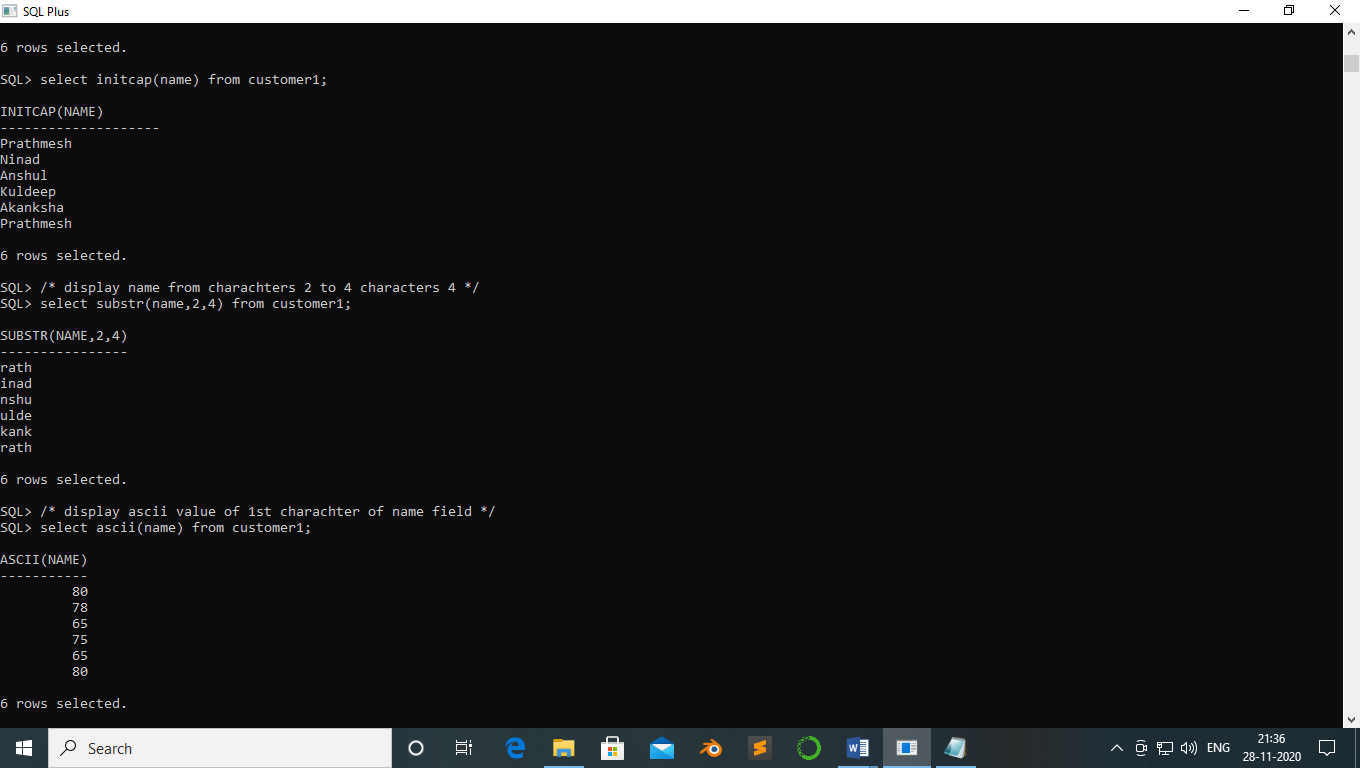
65

75

65

80

6 rows selected.



SQL> insert into customer1 values(7, 'Kaustubh','Belapur','20','65000');

1 row created.

SQL> select \* from customer1;

ID NAME ADDRESS AGE SALARY

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

1 Prathmesh Mumbai 20 40000

2 Ninad Mumbai 20 45000

3 Anshul Dehli 21 50000

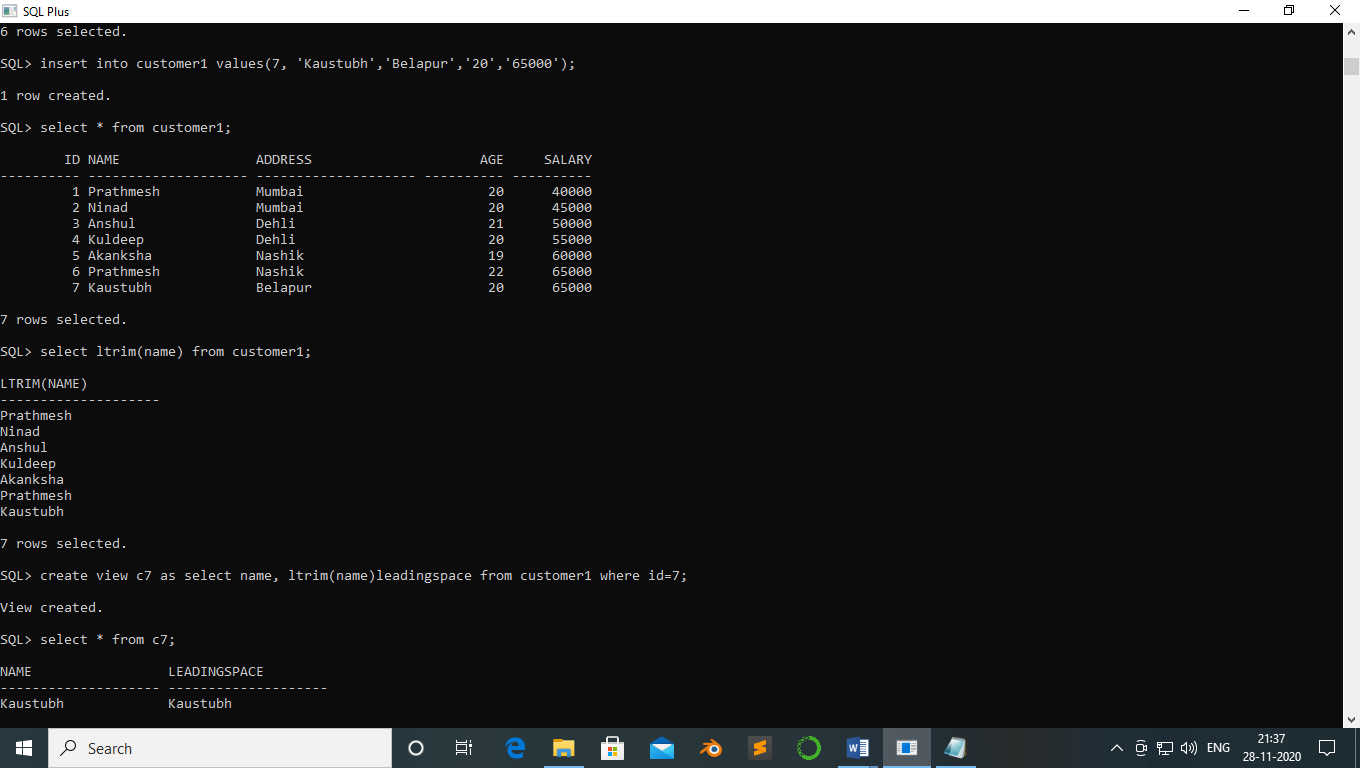
4 Kuldeep Dehli 20 55000

5 Akanksha Nashik 19 60000

6 Prathmesh Nashik 22 65000

7 Kaustubh Belapur 20 65000

7 rows selected.



SQL> select ltrim(name) from customer1;

LTRIM(NAME)

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

Prathmesh

Ninad

Anshul

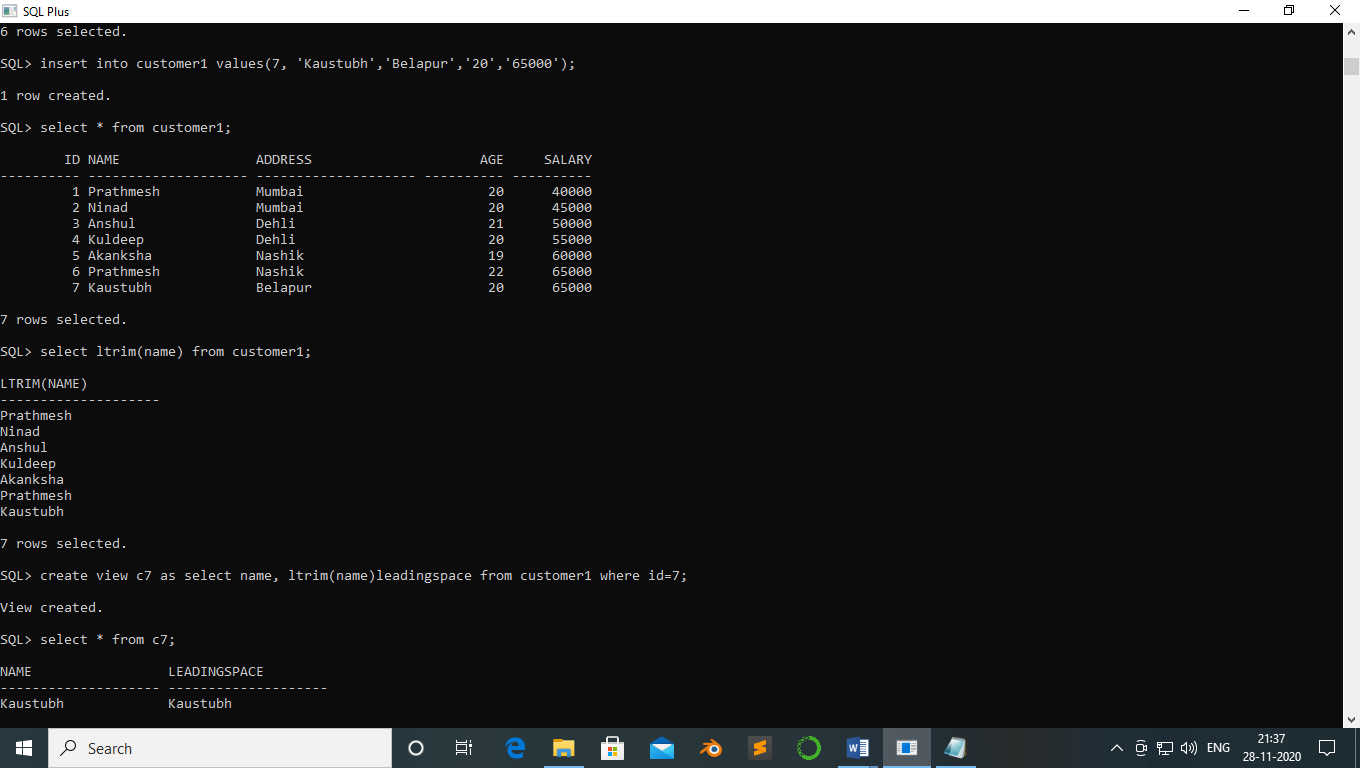
Kuldeep

Akanksha

Prathmesh

Kaustubh

7 rows selected.



SQL> select lpad(name,9,'\*') from customer1;

LPAD(NAME,9,'\*')

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

Prathmesh

\*\*\*\*Ninad

\*\*\*Anshul

\*\*Kuldeep

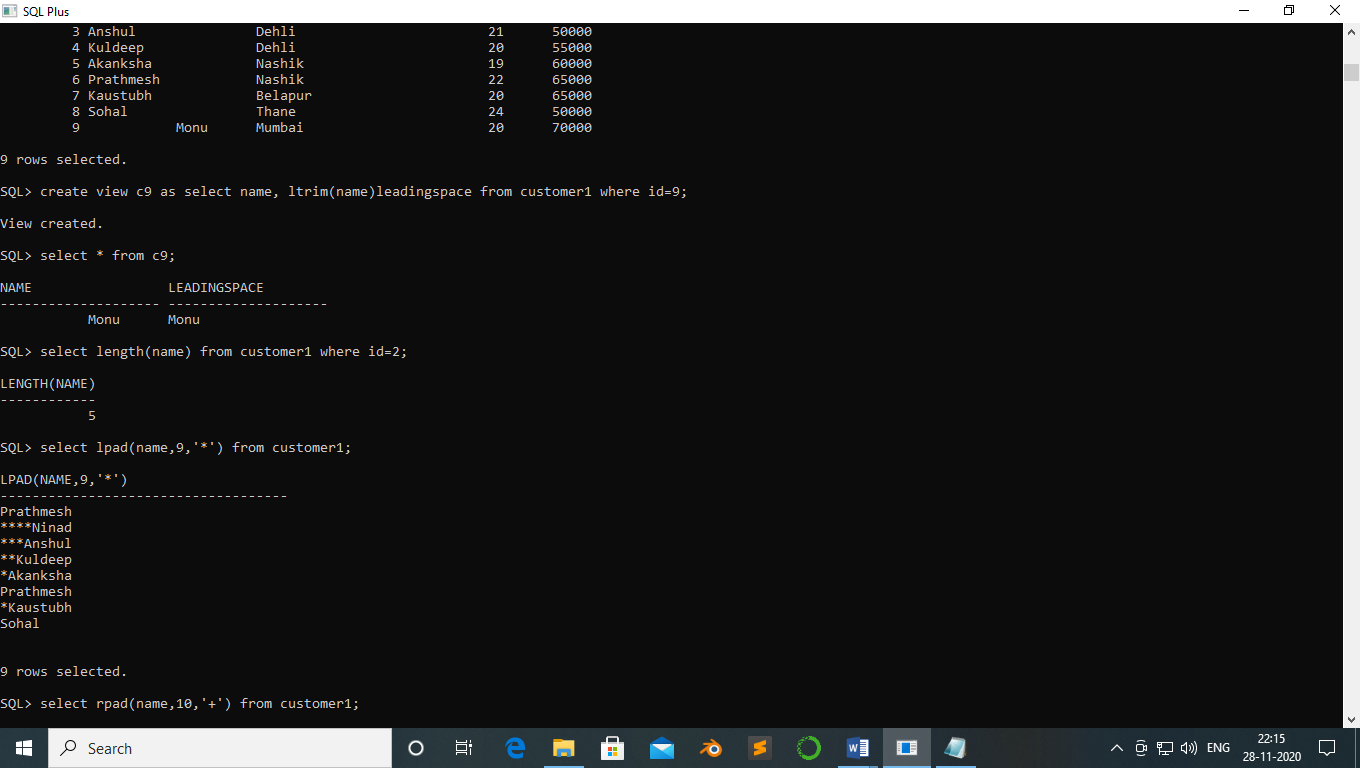
\*Akanksha

Prathmesh

\*Kaustubh

Sohal

9 rows selected.

‘

SQL> select rpad(name,10,'+') from customer1;

RPAD(NAME,10,'+')

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

Prathmesh+

Ninad+++++

Anshul++++

Kuldeep+++

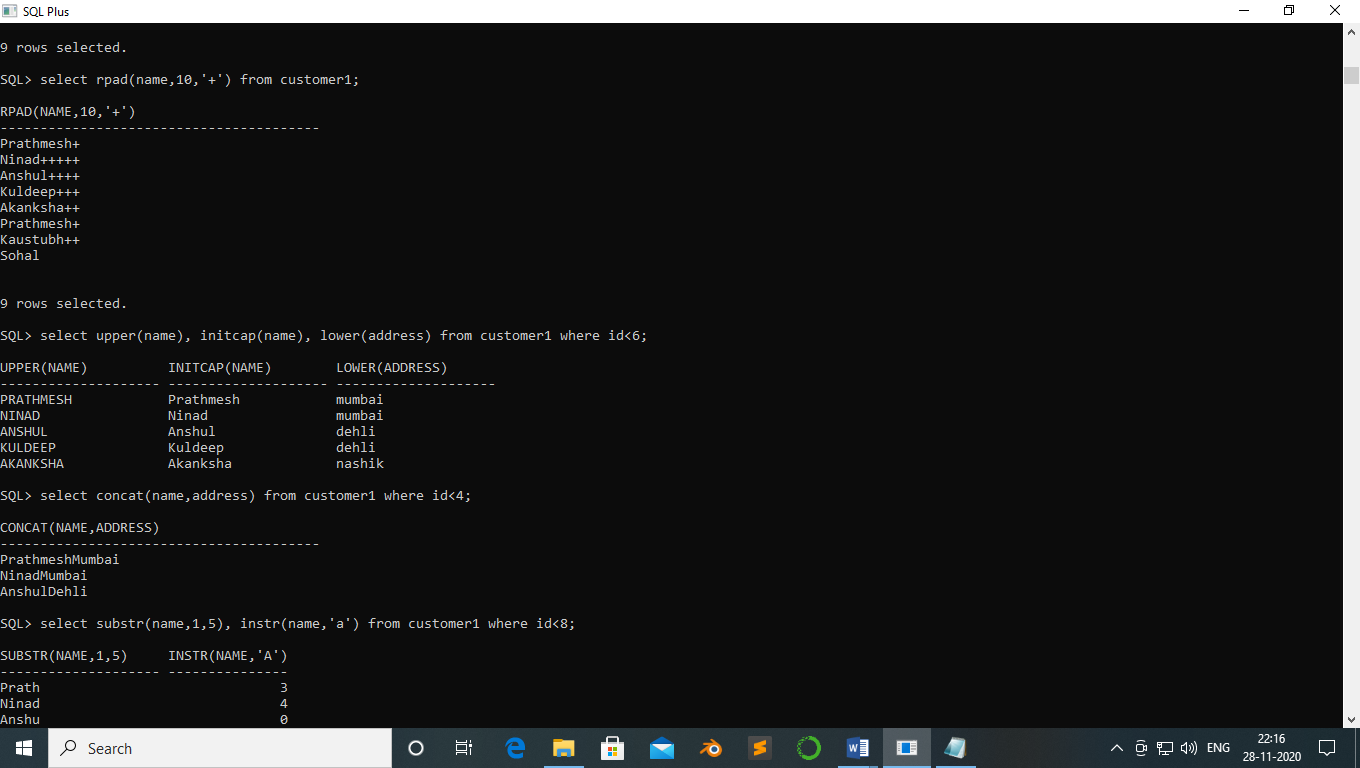
Akanksha++

Prathmesh+

Kaustubh++

Sohal

9 rows selected.



SQL> select upper(name), initcap(name), lower(address) from customer1 where id<6;

UPPER(NAME) INITCAP(NAME) LOWER(ADDRESS)

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

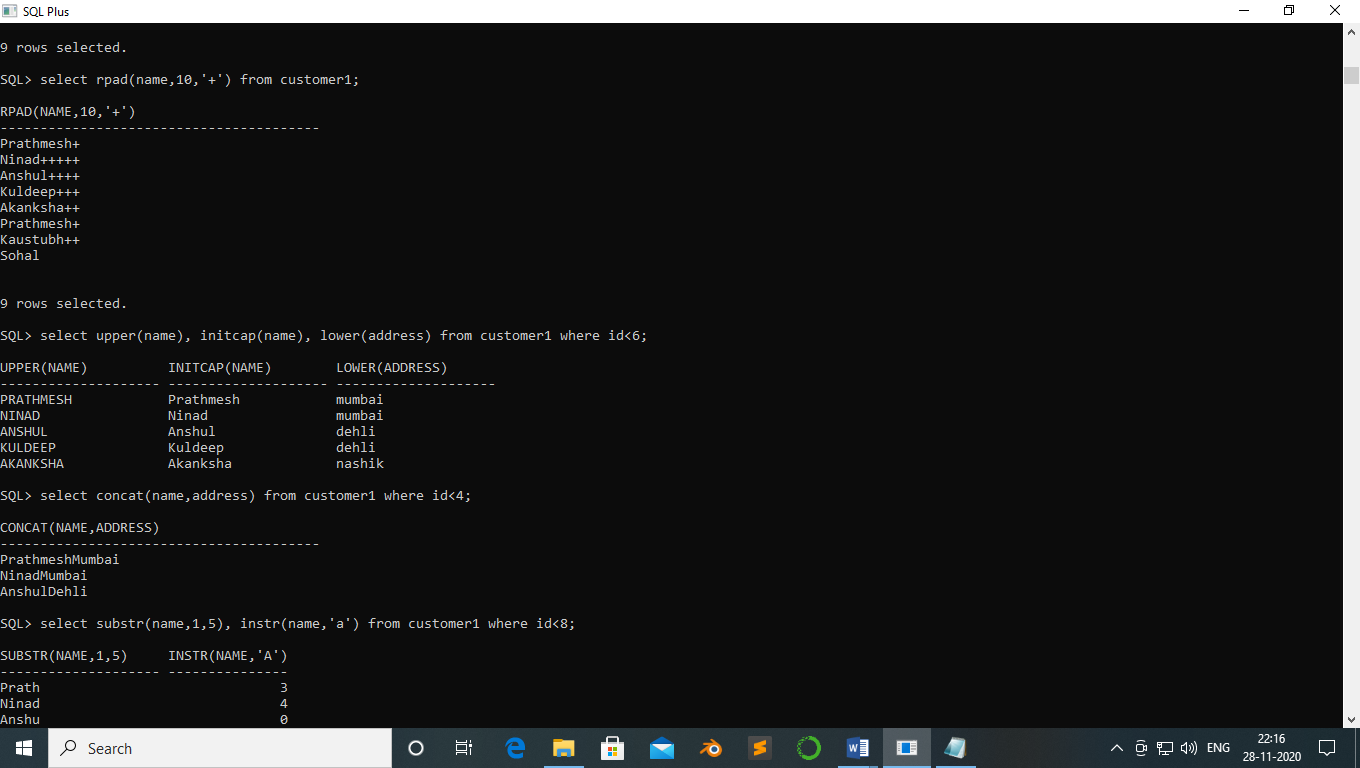
PRATHMESH Prathmesh mumbai

NINAD Ninad mumbai

ANSHUL Anshul dehli

KULDEEP Kuldeep dehli

AKANKSHA Akanksha nashik



SQL> select concat(name,address) from customer1 where id<4;

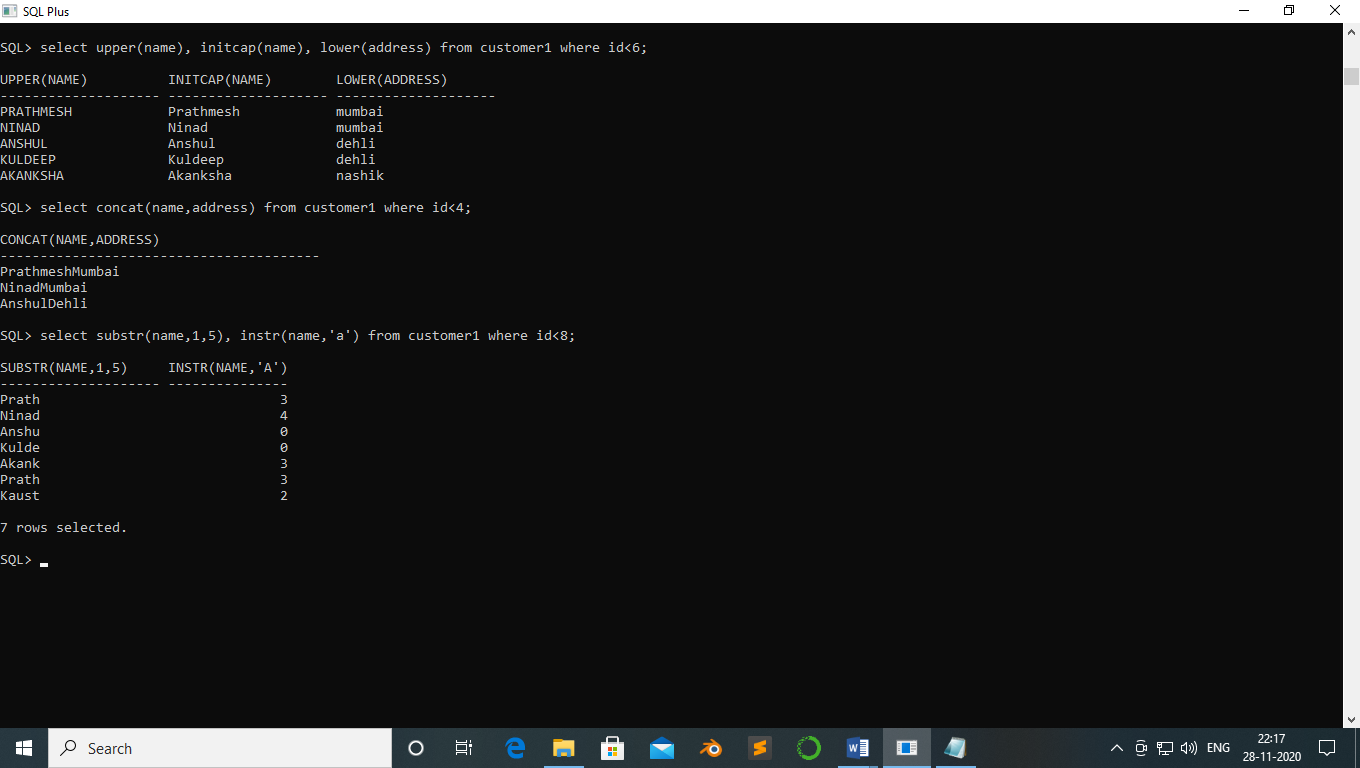
CONCAT(NAME,ADDRESS)

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

PrathmeshMumbai

NinadMumbai

AnshulDehli



SQL> select substr(name,1,5), instr(name,'a') from customer1 where id<8;

SUBSTR(NAME,1,5) INSTR(NAME,'A')

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

Prath 3

Ninad 4

Anshu 0

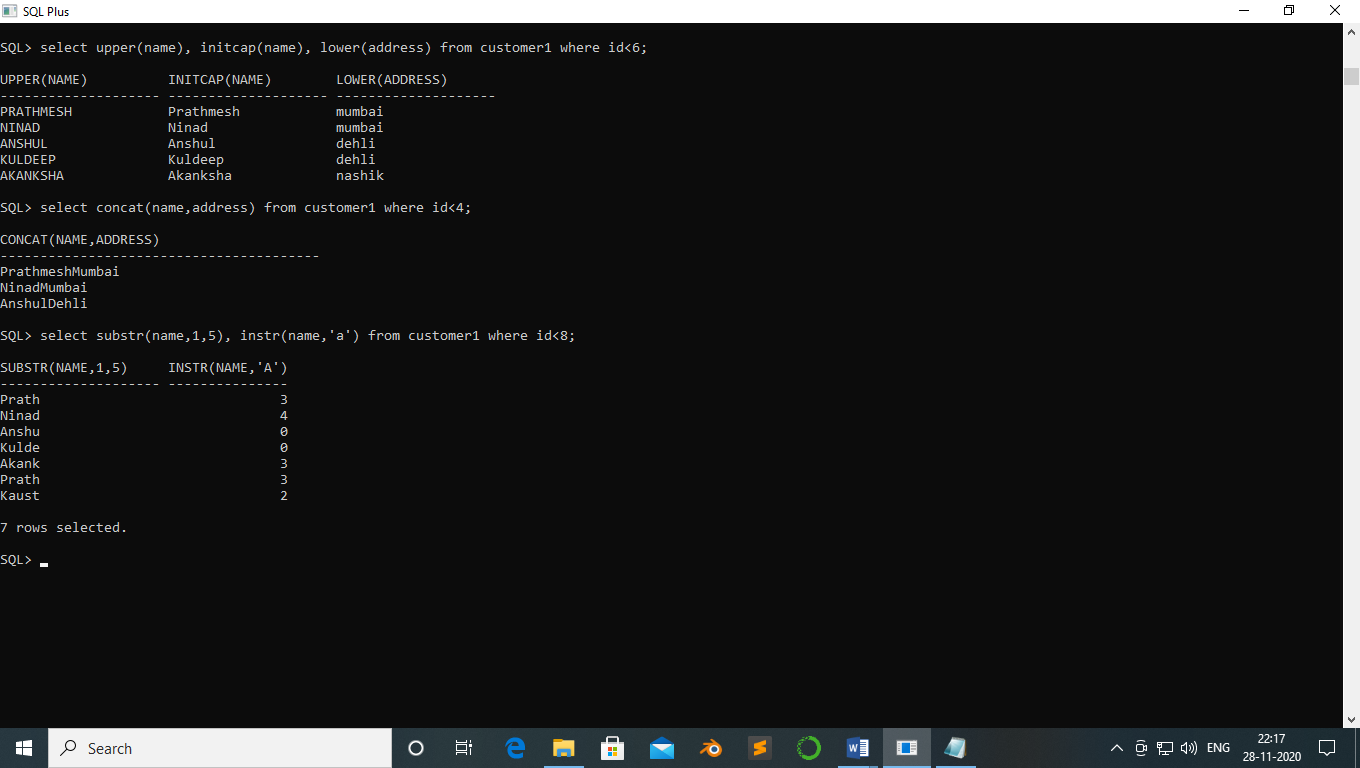
Kulde 0

Akank 3

Prath 3

Kaust 2

7 rows selected.



**Practical no 6**

**Aim:-** SQL Operators:

1. Arithmetic operator
2. Logical operators(in,between and, Any, anyLike operator

**Theory:-**

**Concatenation Operator:**

The concatenation operator manipulates character strings and clob data.

Select 'first name is' || FNAME from student;

**Code:-**

***Concatenation Operator:***

SQL> select \* from studentt;

SID LNAME FNAME MOBILE

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

1 Patil Prathmesh 7208055391

2 Patil Monu 8652184656

3 Patil Kaustubh 9594448714

4 Patil Sohal 8108040556

5 Patil Anurag 3948273922

SQL> select ' first name is'||fname from studentt;

Output:-

'FIRSTNAMEIS'||FNAME

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

first name isPrathmesh

first name isMonu

first name isKaustubh

first name isSohal

first name isAnurag

SQL> select ' first name is '||fname from studentt;

'FIRSTNAMEIS'||FNAME

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

first name is Prathmesh

first name is Monu

first name is Kaustubh

first name is Sohal

first name is Anurag

**AND operator :-**

SQL> select \* from studentt where sid between 3 and 5;

Output:-

SID LNAME FNAME MOBILE

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

3 Patil Kaustubh 9594448714

4 Patil Sohal 8108040556

5 Patil Anurag 3948273922

SQL> select \* from studentt where sid between 4 and 5;

SID LNAME FNAME MOBILE

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

4 Patil Sohal 8108040556

5 Patil Anurag 3948273922

SQL> create table customer(customerId int primary key,customerName varchar(15) not null,contactno number unique, addressCity varchar(20),

2 postalcode int,country varchar(8),salary number);

Table created.

insert into customer values(1,'Prathmesh','7208155391','Belapur','400614','India','50000');

1 row created.

SQL> insert into customer values(2,'Kuldeep','90543633678','Panvel','400619','India','55000');

1 row created.

SQL> insert into customer values(3,'Ninad','598236359547','Sanpada','400645','Nepal','65000');

1 row created.

SQL> insert into customer values(4,'Anshul','984547594564','Nerul','400640','England','75000');

1 row created.

SQL> insert into customer values(5,'Akanksha','92826549451','Vashi','400670','India','85000');

1 row created.

SQL> insert into customer values(6,'Aditya','4378590473','Kharghar','400618','India','75000');

1 row created.

SQL> insert into customer values(7,'Mihir','295454329873','Mulund','400645','India','65000');

1 row created.

SQL> select \* from customer;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

4 Anshul 9.8455E+11 Nerul 400640 England

75000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

6 Aditya 4378590473 Kharghar 400618 India

75000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

7 Mihir 2.9545E+11 Mulund 400645 India

65000

7 rows selected.

SQL> select \* from Customer where Salary between 50000 and 70000;

Output:-

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

7 Mihir 2.9545E+11 Mulund 400645 India

65000

SQL> update Customer set salary=80000 where CUSTOMERID=1;

1 row updated.

SQL> update Customer set salary=100000 where CUSTOMERID=5;

1 row updated.

SQL> select \* from customer;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

4 Anshul 9.8455E+11 Nerul 400640 England

75000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

6 Aditya 4378590473 Kharghar 400618 India

75000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

7 Mihir 2.9545E+11 Mulund 400645 India

65000

7 rows selected.

SQL> select \* from customer where CUSTOMERID between 1 and 3;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

**OR operator :**

SQL> select \* from Customer where COUNTRY='India' OR ADDRESSCITY='Belapur';

Output:-

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

6 Aditya 4378590473 Kharghar 400618 India

75000

7 Mihir 2.9545E+11 Mulund 400645 India

65000

**Practical no 7**

**Aim:-** Clause:

1. Group by
2. Having count
3. Order by

Distinct

Question :-

1. Create table Dept\_master and Employee\_master with specified contraints.

Table:Dept\_master

Used to store Department information

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size | Constraints |
| Dept\_no | varchar | 5 | Primary key |
| Dept\_name | Varchar | 15 | Not null |

Table:Emp\_master

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size | Constraints |
| Emp\_no | Number | 3 | Primary key |
| Emp\_name | Varchar2 | 20 | Not null |
| Emp\_add | Varchar2 | 15 | Default ‘virar’ |
| Joining\_date | Date | - | - |
| Dept\_no | varchar2 | 4 | Foreign key |
| Salary | Number |  | Check(Salary >0) |

1. Add column designation in table Employee\_master.
2. Insert appropriate records in both the tables.
3. Display information of Dept\_master and Emp\_master table.
4. Display Employee information whose name starts with ‘S’
5. Display Employee name whose salary is between 20000 and 30000.
6. Add column designation in table Employee\_master.
7. Add appropriate designations information in table.
8. Display Employee name and salary in descending order.
9. Display Employee name in ascending order.
10. Display details of department which contains ‘a’ as second character.
11. Display Employee name whose designation is not decided.
12. Change address of Tanuj Bhandary with ‘Nerul’.
13. Display Employee name whose salary is more than 30,000 and designation is manager.
14. Display Employee name for those employees who is either Manager, Officer or Clerk.
15. Display Employee name which contains at least 2 occurrences of ‘a’ in their name.
16. Display Department name that does not contains ‘r’ in their name.
17. Increase salary of all employee by 7000 whose designation is ‘Manager’.
18. Display Employee name and working Department name.
19. Display employee name and Department name for those employees whose earning salary more than 50000.
20. Display name of employee who earning highest salary.
21. Display department name of Tanuj Bhandary.
22. Display employee name and employee address whose designation is manager or CEO and salary > 30000.
23. Display employee name, employee address and salary whose designation is manager or CEO and salary > 30000.

**Code:-**

SQL> select \* from Customer ORDER BY CUSTOMERID DESC;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

7 Mihir 2.9545E+11 Mulund 400645 India

65000

6 Aditya 4378590473 Kharghar 400618 India

75000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

4 Anshul 9.8455E+11 Nerul 400640 England

75000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

7 rows selected.

SQL> select \* from Customer ORDER BY CUSTOMERID ;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

4 Anshul 9.8455E+11 Nerul 400640 England

75000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

6 Aditya 4378590473 Kharghar 400618 India

75000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

7 Mihir 2.9545E+11 Mulund 400645 India

65000

7 rows selected.

SQL> select \* from Customer ORDER BY COUNTRY DESC ;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

1 Prathmesh 7208155391 Belapur 400614 India

80000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

7 Mihir 2.9545E+11 Mulund 400645 India

65000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

6 Aditya 4378590473 Kharghar 400618 India

75000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

4 Anshul 9.8455E+11 Nerul 400640 England

75000

7 rows selected.

SQL> select \* from Customer ORDER BY COUNTRY ;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

4 Anshul 9.8455E+11 Nerul 400640 England

75000

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

7 Mihir 2.9545E+11 Mulund 400645 India

65000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

6 Aditya 4378590473 Kharghar 400618 India

75000

1 Prathmesh 7208155391 Belapur 400614 India

80000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

7 rows selected.

SQL> select \* from Customer ORDER BY CUSTOMERNAME DESC;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

7 Mihir 2.9545E+11 Mulund 400645 India

65000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

4 Anshul 9.8455E+11 Nerul 400640 England

75000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

6 Aditya 4378590473 Kharghar 400618 India

75000

7 rows selected.

SQL> select \* from Customer ORDER BY CUSTOMERNAME;

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

6 Aditya 4378590473 Kharghar 400618 India

75000

5 Akanksha 9.2827E+10 Vashi 400670 India

100000

4 Anshul 9.8455E+11 Nerul 400640 England

75000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

2 Kuldeep 9.0544E+10 Panvel 400619 India

55000

7 Mihir 2.9545E+11 Mulund 400645 India

65000

3 Ninad 5.9824E+11 Sanpada 400645 Nepal

65000

CUSTOMERID CUSTOMERNAME CONTACTNO ADDRESSCITY POSTALCODE COUNTRY

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

SALARY

----------

1 Prathmesh 7208155391 Belapur 400614 India

80000

7 rows selected.

select \* from customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 2 | Kuldeep | 90543633678 | Panvel | 400619 | India | 55000 |
| 3 | Ninad | 598236359547 | Sanpada | 400645 | Nepal | 65000 |
| 4 | Anshul | 984547594564 | London | 400640 | England | 75000 |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |
| 7 | Mihir | 295454329873 | New York | 400645 | USA | 65000 |

7 rows selected.

select \* from customer where COUNTRY in('India')

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 2 | Kuldeep | 90543633678 | Panvel | 400619 | India | 55000 |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |

4 rows selected.

select \* from customer where SALARY in(55000)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 2 | Kuldeep | 90543633678 | Panvel | 400619 | India | 55000 |

select \* from customer where SALARY in(50000,75000)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 4 | Anshul | 984547594564 | London | 400640 | England | 75000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |

3 rows selected.

select \* from customer where COUNTRY in('India','USA')

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 2 | Kuldeep | 90543633678 | Panvel | 400619 | India | 55000 |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |
| 7 | Mihir | 295454329873 | New York | 400645 | USA | 65000 |

5 rows selected.

select \* from customer where CUSTOMERNAME LIKE 'P%'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |

select \* from customer where CUSTOMERNAME LIKE 'P%' or CUSTOMERNAME LIKE 'O%'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |

select \* from customer where CUSTOMERNAME LIKE 'P%' or CUSTOMERNAME LIKE 'N%'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 3 | Ninad | 598236359547 | Sanpada | 400645 | Nepal | 65000 |

2 rows selected.

select \* from customer where CUSTOMERNAME LIKE 'P%' or CUSTOMERNAME LIKE 'M%'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 7 | Mihir | 295454329873 | New York | 400645 | USA | 65000 |

2 rows selected.

select \* from customer where CUSTOMERNAME LIKE 'P\_\_t\_mesh'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |

select \* from customer where CUSTOMERNAME LIKE 'Aka\_\_sha'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |

select \* from customer where CUSTOMERNAME LIKE 'P%h'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |

select \* from customer where CUSTOMERNAME LIKE '%h%'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 1 | Prathmesh | 7208155391 | Belapur | 400614 | India | 50000 |
| 4 | Anshul | 984547594564 | London | 400640 | England | 75000 |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 7 | Mihir | 295454329873 | New York | 400645 | USA | 65000 |

4 rows selected.

select \* from customer where CUSTOMERNAME LIKE 'A%a'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |

2 rows selected.

select \* from customer where CUSTOMERNAME NOT LIKE 'P%'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 2 | Kuldeep | 90543633678 | Panvel | 400619 | India | 55000 |
| 3 | Ninad | 598236359547 | Sanpada | 400645 | Nepal | 65000 |
| 4 | Anshul | 984547594564 | London | 400640 | England | 75000 |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |
| 7 | Mihir | 295454329873 | New York | 400645 | USA | 65000 |

6 rows selected.

select \* from customer where CUSTOMERNAME LIKE 'A%' and SALARY>40000 order by SALARY desc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **CONTACTNO** | **ADDRESSCITY** | **POSTALCODE** | **COUNTRY** | **SALARY** |
| 5 | Akanksha | 92826549451 | Vashi | 400670 | India | 85000 |
| 6 | Aditya | 4378590473 | Kharghar | 400618 | India | 75000 |
| 4 | Anshul | 984547594564 | London | 400640 | England | 75000 |

3 rows selected.

select distinct SALARY from customer Order by SALARY

|  |
| --- |
| **SALARY** |
| 50000 |
| 55000 |
| 65000 |
| 75000 |
| 85000 |

5 rows selected.

select distinct SALARY from customer Order by SALARY

|  |
| --- |
| **SALARY** |
| 50000 |
| 55000 |
| 65000 |
| 75000 |
| 85000 |

5 rows selected.

select distinct SALARY from customer Order by SALARY desc

|  |
| --- |
| **SALARY** |
| 85000 |
| 75000 |
| 65000 |
| 55000 |
| 50000 |

5 rows selected.

select distinct SALARY from customer where SALARY>60000 Order by SALARY desc

|  |
| --- |
| **SALARY** |
| 85000 |
| 75000 |
| 65000 |

3 rows selected.

SQL> create table Dept\_master(Dept\_no varchar(4) primary key,Dept\_name varchar(10) not null);

Table created.

SQL> create table Emp\_master(Emp\_no number(3) primary key,Emp\_name varchar(20) not null,

2 Emp\_add varchar(20) default'Virar',joining\_date date,Dept\_no varchar(4),

3 Salary number check(Salary>0),foreign key(Dept\_no) references Dept\_master(Dept\_no));

Table created.

SQL> insert into Dept\_master values(101,'IT');

1 row created.

SQL> insert into Dept\_master values(102,'Comms');

1 row created.

SQL> insert into Dept\_master values(103,'Security');

1 row created.

SQL> insert into Dept\_master values(104,'Medical');

1 row created.

SQL> insert into Dept\_master values(105,'Bank');

1 row created.

SQL> insert into Dept\_master values(106,'Accounts');

1 row created.

SQL> insert into Dept\_master values(107,'Records');

1 row created.

SQL> select \* from Dept\_master;

DEPT DEPT\_NAME

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

101 IT

102 Comms

103 Security

104 Medical

105 Bank

106 Accounts

107 Records

7 rows selected.

SQL> alter table Emp\_master add designation varchar(15);

Table altered.

SQL> insert into Emp\_master values(1,'Prathmesh Patil','Navi Mumbai','18-June-2012',101,40000,'Manager');

1 row created.

SQL> insert into Emp\_master values(2,'Kuldeep Singhvi','New Panvel','20-July-2013',102,40000,'CEO');

1 row created.

SQL> insert into Emp\_master values(3,'Akanksha Bhagat','Vashi','21-March-2014',103,40000,'Manager');

1 row created.

SQL> insert into Emp\_master values(4,'Ninad Obsi','Sanpada','13-May-2015',104,50000,'Manager');

1 row created.

SQL> insert into Emp\_master values(5,'Anshul Sinha','Nerul','24-Aug-2016',105,60000,'Head');

1 row created.

SQL> insert into Emp\_master values(6,'Sohal Patil','Ulwe','02-Sep-2017',106,60000,'CEO');

1 row created.

SQL> insert into Emp\_master values(7,'Oswald','Belapur','26-Oct-2018',107,60000,'Clerk');

1 row created.

SQL> select \* from Emp\_master;

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

1 Prathmesh Patil Navi Mumbai 18-JUN-12 101 40000

Manager

2 Kuldeep Singhvi New Panvel 20-JUL-13 102 40000

CEO

3 Akanksha Bhagat Vashi 21-MAR-14 103 40000

Manager

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

4 Ninad Obsi Sanpada 13-MAY-15 104 50000

Manager

5 Anshul Sinha Nerul 24-AUG-16 105 60000

Head

6 Sohal Patil Ulwe 02-SEP-17 106 60000

CEO

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

7 Oswald Belapur 26-OCT-18 107 60000

Clerk

7 rows selected.

SQL> select \* from Emp\_master where EMP\_NAME like 'S%';

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

6 Sohal Patil Ulwe 02-SEP-17 106 60000

CEO

SQL> select EMP\_NAME from Emp\_master where SALARY between 20000 and 40000;

EMP\_NAME

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

Prathmesh Patil

Kuldeep Singhvi

Akanksha Bhagat

SQL> select EMP\_NAME , SALARY from Emp\_master;

EMP\_NAME SALARY

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

Prathmesh Patil 40000

Kuldeep Singhvi 40000

Akanksha Bhagat 40000

Ninad Obsi 50000

Anshul Sinha 60000

Sohal Patil 60000

Oswald 60000

7 rows selected.

SQL> select EMP\_NAME , SALARY from Emp\_master order by salary;

EMP\_NAME SALARY

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

Prathmesh Patil 40000

Kuldeep Singhvi 40000

Akanksha Bhagat 40000

Ninad Obsi 50000

Anshul Sinha 60000

Sohal Patil 60000

Oswald 60000

7 rows selected.

SQL> select EMP\_NAME , SALARY from Emp\_master order by salary desc;

EMP\_NAME SALARY

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

Sohal Patil 60000

Anshul Sinha 60000

Oswald 60000

Ninad Obsi 50000

Kuldeep Singhvi 40000

Prathmesh Patil 40000

Akanksha Bhagat 40000

7 rows selected.

SQL> select \* from Emp\_master order by EMP\_NAME,salary desc;

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

3 Akanksha Bhagat Vashi 21-MAR-14 103 40000

Manager

5 Anshul Sinha Nerul 24-AUG-16 105 60000

Head

2 Kuldeep Singhvi New Panvel 20-JUL-13 102 40000

CEO

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

4 Ninad Obsi Sanpada 13-MAY-15 104 50000

Manager

7 Oswald Belapur 26-OCT-18 107 60000

Clerk

1 Prathmesh Patil Navi Mumbai 18-JUN-12 101 40000

Manager

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

6 Sohal Patil Ulwe 02-SEP-17 106 60000

CEO

7 rows selected.

SQL> select \* from Emp\_master order by EMP\_NAME;

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

3 Akanksha Bhagat Vashi 21-MAR-14 103 40000

Manager

5 Anshul Sinha Nerul 24-AUG-16 105 60000

Head

2 Kuldeep Singhvi New Panvel 20-JUL-13 102 40000

CEO

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

4 Ninad Obsi Sanpada 13-MAY-15 104 50000

Manager

7 Oswald Belapur 26-OCT-18 107 60000

Clerk

1 Prathmesh Patil Navi Mumbai 18-JUN-12 101 40000

Manager

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

6 Sohal Patil Ulwe 02-SEP-17 106 60000

CEO

7 rows selected.

SQL> select \* from Dept\_master where Dept\_name like '\_o%';

DEPT DEPT\_NAME

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

102 Comms

SQL> select \* from EMP\_master where DESIGNATION like 'CEO';

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

2 Kuldeep Singhvi New Panvel 20-JUL-13 102 40000

CEO

6 Sohal Patil Ulwe 02-SEP-17 106 60000

CEO

SQL> update Emp\_master set EMP\_ADD = 'CBD' where EMP\_NAME='Prathmesh Patil';

1 row updated.

SQL> select \* from Emp\_master;

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

1 Prathmesh Patil CBD 18-JUN-12 101 40000

Manager

2 Kuldeep Singhvi New Panvel 20-JUL-13 102 40000

CEO

3 Akanksha Bhagat Vashi 21-MAR-14 103 40000

Manager

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

4 Ninad Obsi Sanpada 13-MAY-15 104 50000

Manager

5 Anshul Sinha Nerul 24-AUG-16 105 60000

Head

6 Sohal Patil Ulwe 02-SEP-17 106 60000

CEO

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

7 Oswald Belapur 26-OCT-18 107 60000

Clerk

7 rows selected.

SQL> select EMP\_NAME from Emp\_master Where SALARY>30000 and DESIGNATION like 'Manager';

EMP\_NAME

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

Prathmesh Patil

Akanksha Bhagat

Ninad Obsi

SQL> select EMP\_NAME from Emp\_master where DESIGNATION = 'Head' or DESIGNATION = 'Clerk' or DESIGNATION = 'CEO';

EMP\_NAME

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

Kuldeep Singhvi

Anshul Sinha

Sohal Patil

Oswald

SQL> select \* from Emp\_master where EMP\_NAME like('%A%a%');

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

3 Akanksha Bhagat Vashi 21-MAR-14 103 40000

Manager

5 Anshul Sinha Nerul 24-AUG-16 105 60000

Head

SQL> select \* from Dept\_master where DEPT\_NAME not like ('O%');

DEPT DEPT\_NAME

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

101 IT

102 Comms

103 Security

104 Medical

105 Bank

106 Accounts

107 Records

7 rows selected.

SQL> update Emp\_master set SALARY=SALARY+7000;

7 rows updated.

SQL> select \* from Emp\_master ;

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

1 Prathmesh Patil CBD 18-JUN-12 101 47000

Manager

2 Kuldeep Singhvi New Panvel 20-JUL-13 102 47000

CEO

3 Akanksha Bhagat Vashi 21-MAR-14 103 47000

Manager

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

4 Ninad Obsi Sanpada 13-MAY-15 104 57000

Manager

5 Anshul Sinha Nerul 24-AUG-16 105 67000

Head

6 Sohal Patil Ulwe 02-SEP-17 106 67000

CEO

EMP\_NO EMP\_NAME EMP\_ADD JOINING\_D DEPT SALARY

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

DESIGNATION

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

7 Oswald Belapur 26-OCT-18 107 67000

Clerk

7 rows selected.

SQL> select EMP\_NAME,DEPT\_NAME from Emp\_master,Dept\_master where SALARY<50000;

EMP\_NAME DEPT\_NAME

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

Prathmesh Patil IT

Prathmesh Patil Comms

Prathmesh Patil Security

Prathmesh Patil Medical

Prathmesh Patil Bank

Prathmesh Patil Accounts

Prathmesh Patil Records

Kuldeep Singhvi IT

Kuldeep Singhvi Comms

Kuldeep Singhvi Security

Kuldeep Singhvi Medical

EMP\_NAME DEPT\_NAME

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

Kuldeep Singhvi Bank

Kuldeep Singhvi Accounts

Kuldeep Singhvi Records

Akanksha Bhagat IT

Akanksha Bhagat Comms

Akanksha Bhagat Security

Akanksha Bhagat Medical

Akanksha Bhagat Bank

Akanksha Bhagat Accounts

Akanksha Bhagat Records

21 rows selected.

SQL> insert into Dept\_master values(108,'Software');

1 row created.

SQL> select \* from Dept\_master;

DEPT DEPT\_NAME

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

101 IT

102 Comms

103 Security

104 Medical

105 Bank

106 Accounts

107 Records

108 Software

8 rows selected.

SQL> insert into Emp\_master values(8,'Anthony','Thane','20-dec-2019',108,'20000','NOT DECIDE');

1 row created.

SQL> select EMP\_NAME ,DEPT\_NAME from Emp\_master,Dept\_master where SALARY > 50000;

EMP\_NAME DEPT\_NAME

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

Ninad Obsi IT

Ninad Obsi Comms

Ninad Obsi Security

Ninad Obsi Medical

Ninad Obsi Bank

Ninad Obsi Accounts

Ninad Obsi Records

Ninad Obsi Software

Anshul Sinha IT

Anshul Sinha Comms

Anshul Sinha Security

EMP\_NAME DEPT\_NAME

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

Anshul Sinha Medical

Anshul Sinha Bank

Anshul Sinha Accounts

Anshul Sinha Records

Anshul Sinha Software

Sohal Patil IT

Sohal Patil Comms

Sohal Patil Security

Sohal Patil Medical

Sohal Patil Bank

Sohal Patil Accounts

EMP\_NAME DEPT\_NAME

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

Sohal Patil Records

Sohal Patil Software

Oswald IT

Oswald Comms

Oswald Security

Oswald Medical

Oswald Bank

Oswald Accounts

Oswald Records

Oswald Software

32 rows selected.

SQL> select EMP\_NAME , EMP\_ADD from Emp\_master where DESIGNATION='CEO' or DESIGNATION='Manager' and SALARY > 30000;

EMP\_NAME EMP\_ADD

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

Prathmesh Patil CBD

Kuldeep Singhvi New Panvel

Akanksha Bhagat Vashi

Ninad Obsi Sanpada

Sohal Patil Ulwe

SQL> select EMP\_NAME , EMP\_ADD, SALARY from Emp\_master where DESIGNATION='CEO' or DESIGNATION='Manager' and SALARY > 30000;

EMP\_NAME EMP\_ADD SALARY

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

Prathmesh Patil CBD 47000

Kuldeep Singhvi New Panvel 47000

Akanksha Bhagat Vashi 47000

Ninad Obsi Sanpada 57000

Sohal Patil Ulwe 67000

**Practical no 8**

**Aim:-**SQL subquerries

**Theory:-**

Subquery:

A subquery is a query that is nested inside a SELECT,INSERT,UPDATE OR DELETE statement, or inside another subquery. A subquery can be used anywhere an expression in another query.

Syntax of subquery:

From table\_name

Where coloumn\_name expression operator

(SELECT COLOUMN\_NAME from TABLE\_NAME WHERE ....);

**Code:-**

SQL> create table database (id int,name varchar(20),roll\_no number, location varchar(15), phone\_no number);

Table created.

SQL> insert into database values(1,'Prathmesh','101','Navi Mumbai','7208155391');

1 row created.

SQL> insert into database values(2,'Ninad','102','Mumbai','396687542');

1 row created.

SQL> insert into database values(3,'Anshul','103','Nerul','234976437');

1 row created.

SQL> insert into database values(4,'Kuldeep','104','Panvel','482536334');

1 row created.

SQL> insert into database values(5,'Akanksha','105','Vashi','346081576');

1 row created.

SQL> select \* from database;

ID NAME ROLL\_NO LOCATION PHONE\_NO

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

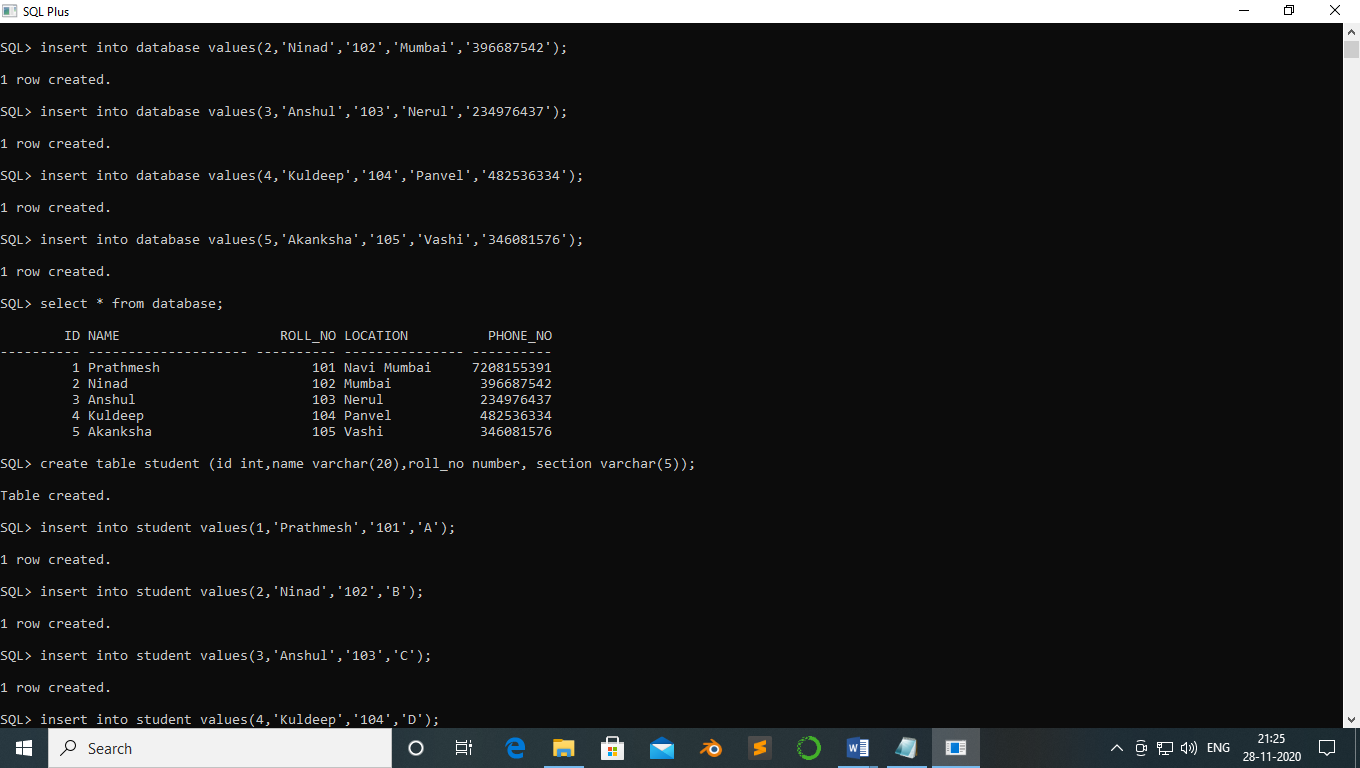
1 Prathmesh 101 Navi Mumbai 7208155391

2 Ninad 102 Mumbai 396687542

3 Anshul 103 Nerul 234976437

4 Kuldeep 104 Panvel 482536334

5 Akanksha 105 Vashi 346081576



SQL> create table student (id int,name varchar(20),roll\_no number, section varchar(5));

Table created.

SQL> insert into student values(1,'Prathmesh','101','A');

1 row created.

SQL> insert into student values(2,'Ninad','102','B');

1 row created.

SQL> insert into student values(3,'Anshul','103','C');

1 row created.

SQL> insert into student values(4,'Kuldeep','104','D');

1 row created.

SQL> insert into student values(5,'Akanksha','105','A');

1 row created.

SQL> select \* from student;

ID NAME ROLL\_NO SECTI

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

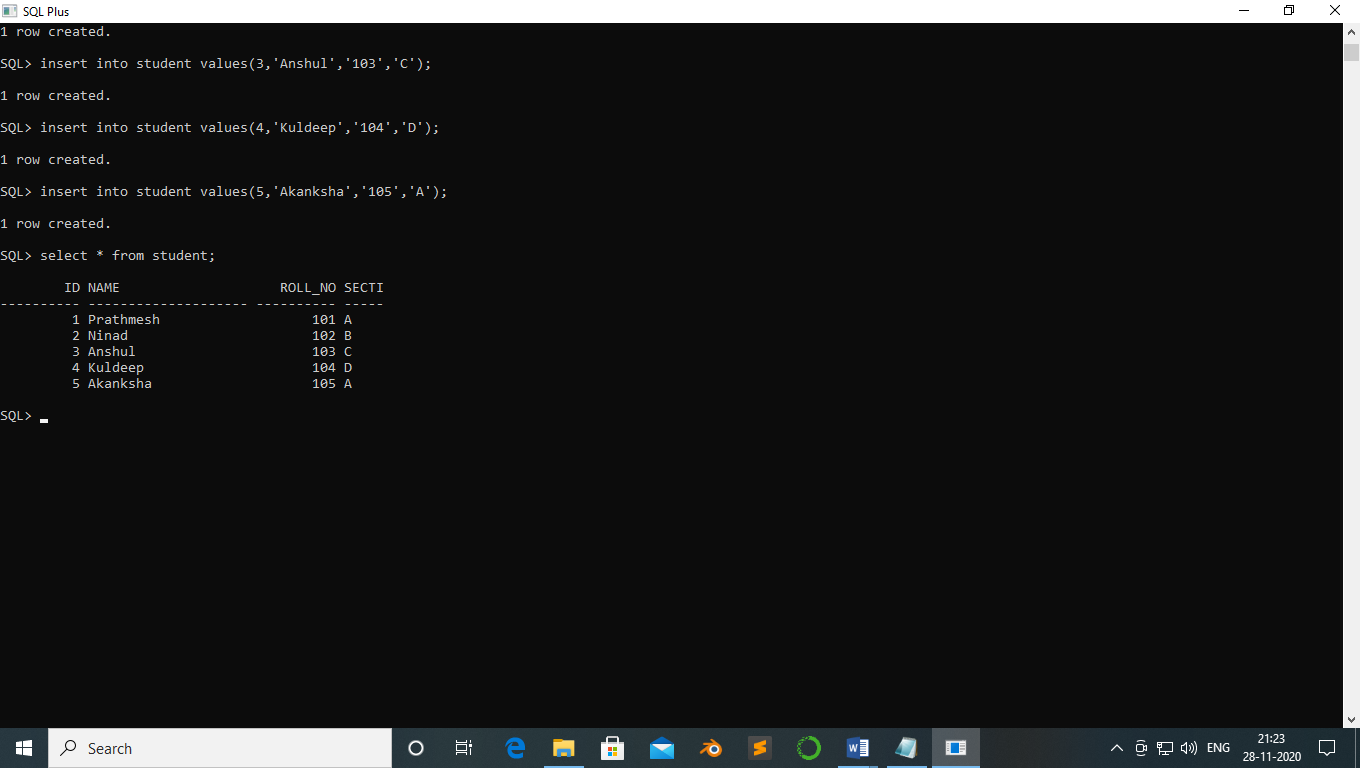
1 Prathmesh 101 A

2 Ninad 102 B

3 Anshul 103 C

4 Kuldeep 104 D

5 Akanksha 105 A



SQL> select roll\_no from student where section='B';

ROLL\_NO

----------

102

SQL> select roll\_no from student where section='A' and roll\_no>102;

ROLL\_NO

----------

105

SQL> alter table database add salary number;

Table altered.

SQL> update database set salary='20000' where id=1;

1 row updated.

SQL> update database set salary='21000' where id=2;

1 row updated.

SQL> update database set salary='22000' where id=3;

1 row updated.

SQL> update database set salary='23000' where id=4;

1 row updated.

SQL> update database set salary='24000' where id=5;

1 row updated.

SQL> select \* from database;

ID NAME ROLL\_NO LOCATION PHONE\_NO SALARY

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

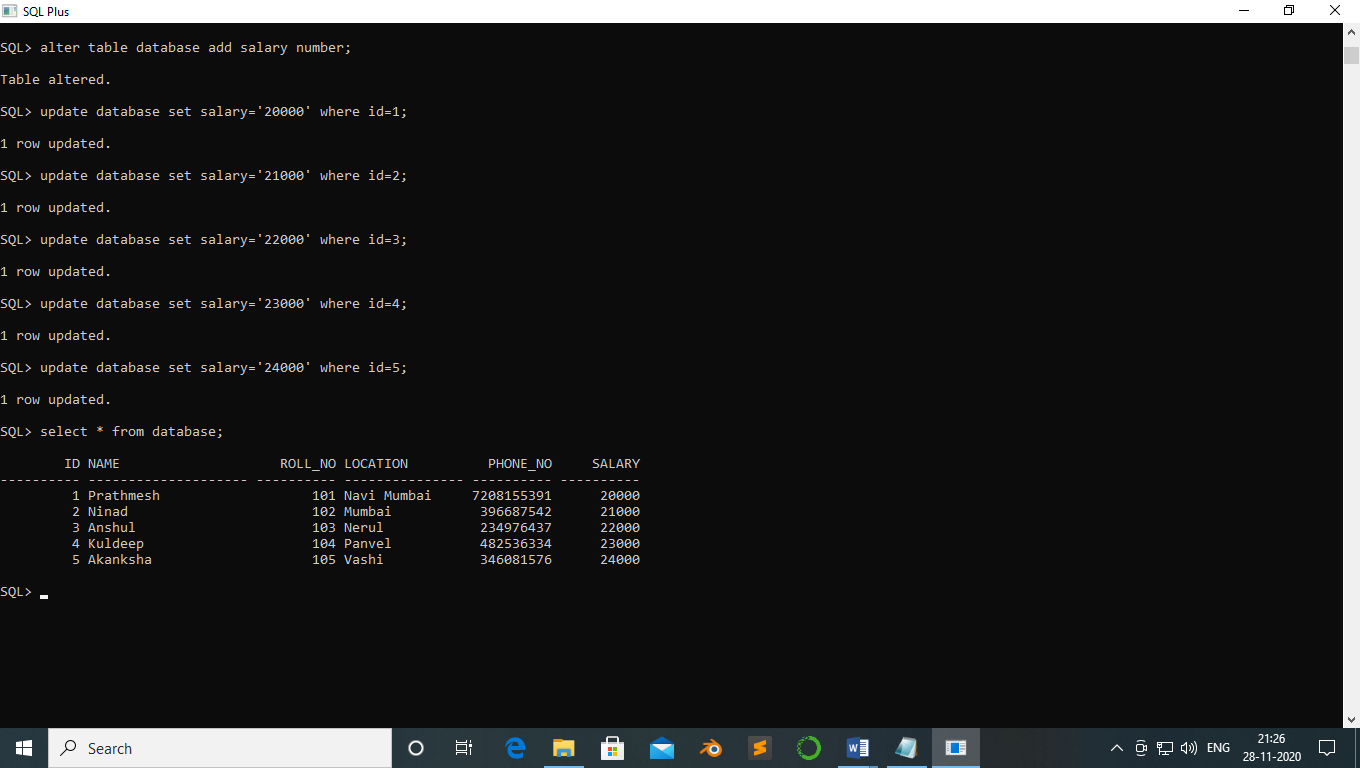
1 Prathmesh 101 Navi Mumbai 7208155391 20000

2 Ninad 102 Mumbai 396687542 21000

3 Anshul 103 Nerul 234976437 22000

4 Kuldeep 104 Panvel 482536334 23000

5 Akanksha 105 Vashi 346081576 24000



**PRACTICAL NO:-9**

**Aim:-** To perform query related to joins.

**Theory:-**

Inner join :-

The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns. If there are records in the "Orders" table that do not have matches in "Customers", these orders will not be shown!

Left join :-

The LEFT JOIN keyword returns all records from the left table (Customers), even if there are no matches in the right table (Orders).

Right join :-

The RIGHT JOIN keyword returns all records from the right table (Employees), even if there are no matches in the left table (Orders).

Full outer join :-

The FULL OUTER JOIN keyword returns all matching records from both tables whether the other table matches or not. So, if there are rows in "Customers" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customers", those rows will be listed as well.

**Code:-**

create table customer1(id int primary key,name varchar(20),address varchar(20),age number,salary number)

Table created.

insert into customer1 values(1,'Prathmesh','Mumbai','20','40000')

1 row(s) inserted.

insert into customer1 values(2,'Ninad','Mumbai','20','45000')

1 row(s) inserted.

insert into customer1 values(3,'Anshul','Dehli','21','50000')

1 row(s) inserted.

insert into customer1 values(4,'Kuldeep','Dehli','20','55000')

1 row(s) inserted.

insert into customer1 values(5,'Akanksha','Nashik','19','60000')

1 row(s) inserted.

insert into customer1 values(6,'Prathmesh','Nashik','22','65000')

1 row(s) inserted.

**Output:-**

select \* from customer1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NAME** | **ADDRESS** | **AGE** | **SALARY** |
| 1 | Prathmesh | Mumbai | 20 | 40000 |
| 2 | Ninad | Mumbai | 20 | 45000 |
| 3 | Anshul | Dehli | 21 | 50000 |
| 4 | Kuldeep | Dehli | 20 | 55000 |
| 5 | Akanksha | Nashik | 19 | 60000 |
| 6 | Prathmesh | Nashik | 22 | 65000 |

6 rows selected.

create table orders(oid int, id int,ordername varchar(20),foreign key(id)references customer1(id))

Table created.

insert into orders values(101,1,'table')

1 row(s) inserted.

insert into orders values(102,2,'Chairs')

1 row(s) inserted.

insert into orders values(103,3,'Stools')

1 row(s) inserted.

insert into orders values(104,4,'Wires')

1 row(s) inserted.

insert into orders values(105,5,'table')

1 row(s) inserted.

insert into orders values(106,6,'Paints')

1 row(s) inserted.

**Output:-**

select \* from orders

|  |  |  |
| --- | --- | --- |
| **OID** | **ID** | **ORDERNAME** |
| 101 | 1 | table |
| 102 | 2 | Chairs |
| 103 | 3 | Stools |
| 104 | 4 | Wires |
| 105 | 5 | table |
| 106 | 6 | Paints |

6 rows selected.

Q1. Display common data from both the table ?

select customer1.id, customer1.name, orders.ordername from customer1, orders where customer1.id=orders.id

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 2 | Ninad | Chairs |
| 3 | Anshul | Stools |
| 4 | Kuldeep | Wires |
| 5 | Akanksha | table |
| 6 | Prathmesh | Paints |

6 rows selected.

select customer1.id, customer1.name, orders.ordername from customer1, orders where customer1.id=orders.id and customer1.name like 'P%'

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 6 | Prathmesh | Paints |

2 rows selected.

Q2. Display common data from both the table?

select customer1.id, customer1.name, orders.ordername from customer1 inner join orders on customer1.id=orders.id

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 2 | Ninad | Chairs |
| 3 | Anshul | Stools |
| 4 | Kuldeep | Wires |
| 5 | Akanksha | table |
| 6 | Prathmesh | Paints |

6 rows selected.

Q3. Display data from right table having the same values as left table?

select customer1.id, customer1.name, orders.ordername from customer1 left join orders on customer1.id=orders.id

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 2 | Ninad | Chairs |
| 3 | Anshul | Stools |
| 4 | Kuldeep | Wires |
| 5 | Akanksha | table |
| 6 | Prathmesh | Paints |

6 rows selected.

Q4. display data from left table having the same values as right table?

select customer1.id, customer1.name, orders.ordername from customer1 right join orders on customer1.id=orders.id

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 2 | Ninad | Chairs |
| 3 | Anshul | Stools |
| 4 | Kuldeep | Wires |
| 5 | Akanksha | table |
| 6 | Prathmesh | Paints |

6 rows selected.

Q5. display data from data using full outer joint?

select customer1.id, customer1.name, orders.ordername from customer1 full outer join orders on customer1.id=orders.id

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 2 | Ninad | Chairs |
| 3 | Anshul | Stools |
| 4 | Kuldeep | Wires |
| 5 | Akanksha | table |
| 6 | Prathmesh | Paints |

6 rows selected.

Q6. display data from both data using inner joint whose name start from R?

select customer1.id, customer1.name, orders.ordername from customer1 inner join orders on customer1.id=orders.id and customer1.name like 'A%'

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 3 | Anshul | Stools |
| 5 | Akanksha | table |

2 rows selected.

Q7. Use inner join and display name in ascending order?

select customer1.id, customer1.name, orders.ordername from customer1 inner join orders on customer1.id=orders.id order by name desc;

**Output:-**

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **ORDERNAME** |
| 1 | Prathmesh | table |
| 6 | Prathmesh | Paints |
| 2 | Ninad | Chairs |
| 4 | Kuldeep | Wires |
| 3 | Anshul | Stools |
| 5 | Akanksha | table |

6 rows selected.

**Practical no 10**

**Aim:-**Views and index

**Code:-**

SQL> create table customer1(id int primary key,name varchar(20),address varchar(20),age number,salary number);

Table created.

SQL> insert into customer1 values(1,'Prathmesh','Mumbai','20','40000');

1 row created.

SQL> insert into customer1 values(2,'Ninad','Mumbai','20','45000');

1 row created.

SQL> insert into customer1 values(3,'Anshul','Dehli','21','50000');

1 row created.

SQL> insert into customer1 values(4,'Kuldeep','Dehli','20','55000');

1 row created.

SQL> insert into customer1 values(5,'Akanksha','Nashik','19','60000');

1 row created.

SQL> insert into customer1 values(6,'Prathmesh','Nashik','22','65000');

1 row created.

SQL> select \* from customer1;

ID NAME ADDRESS AGE SALARY

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

1 Prathmesh Mumbai 20 40000

2 Ninad Mumbai 20 45000

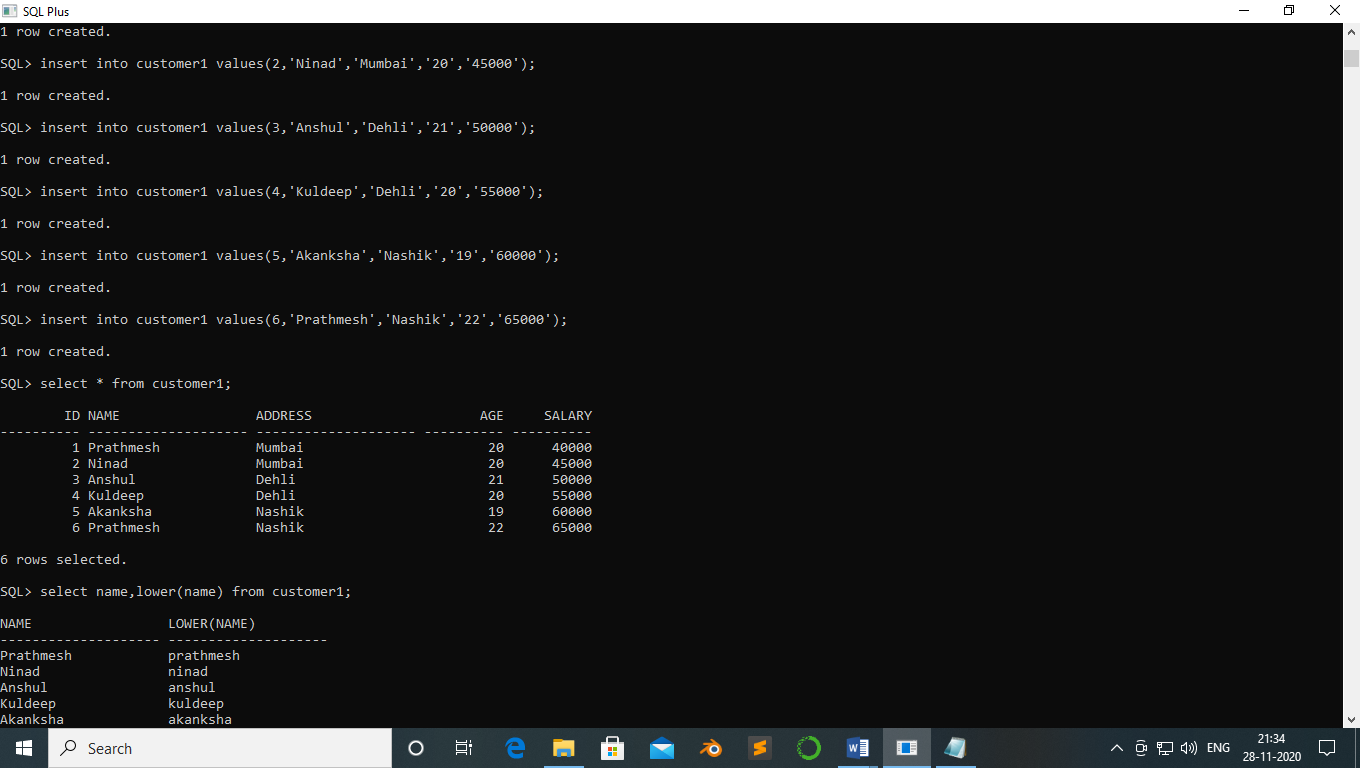
3 Anshul Dehli 21 50000

4 Kuldeep Dehli 20 55000

5 Akanksha Nashik 19 60000

6 Prathmesh Nashik 22 65000

6 rows selected.



SQL> select name,lower(name) from customer1;

NAME LOWER(NAME)

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

Prathmesh prathmesh

Ninad ninad

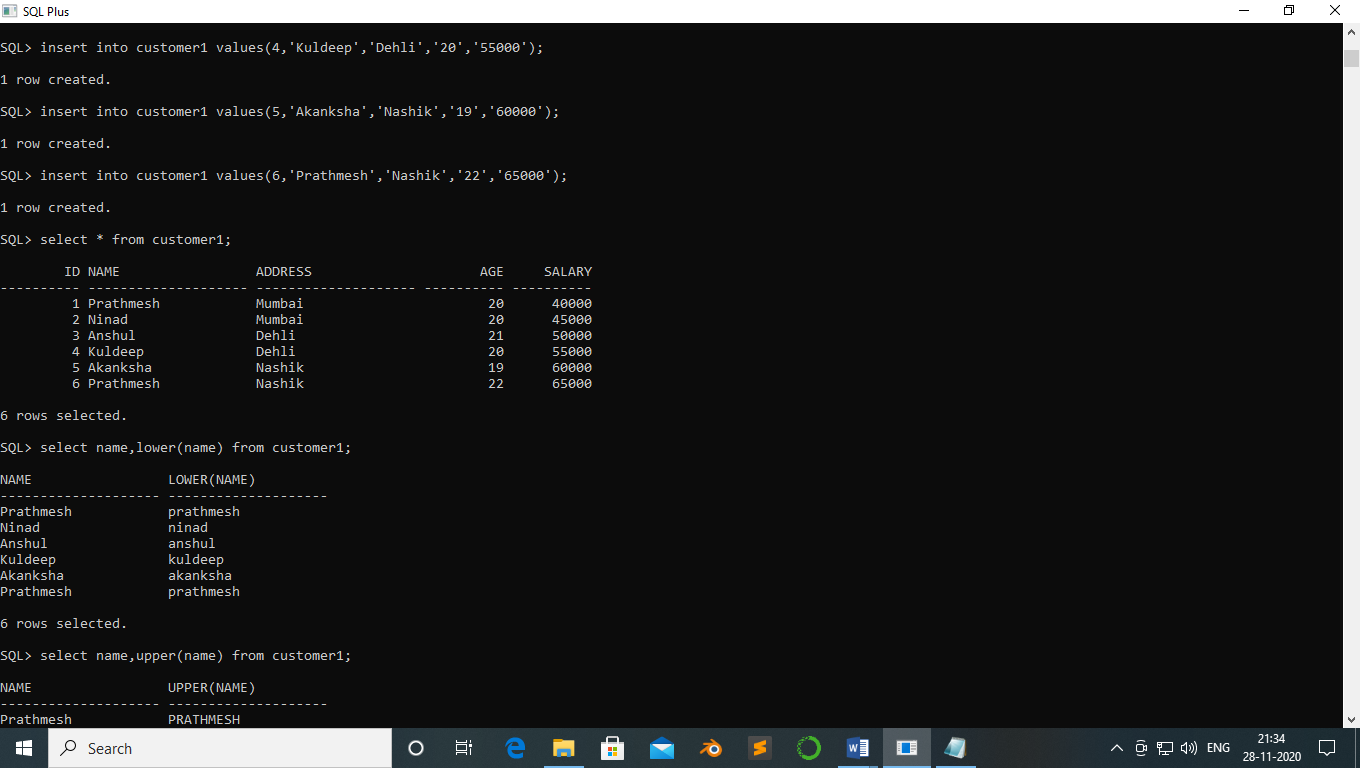
Anshul anshul

Kuldeep kuldeep

Akanksha akanksha

Prathmesh prathmesh

6 rows selected.



SQL> select name,upper(name) from customer1;

NAME UPPER(NAME)

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

Prathmesh PRATHMESH

Ninad NINAD

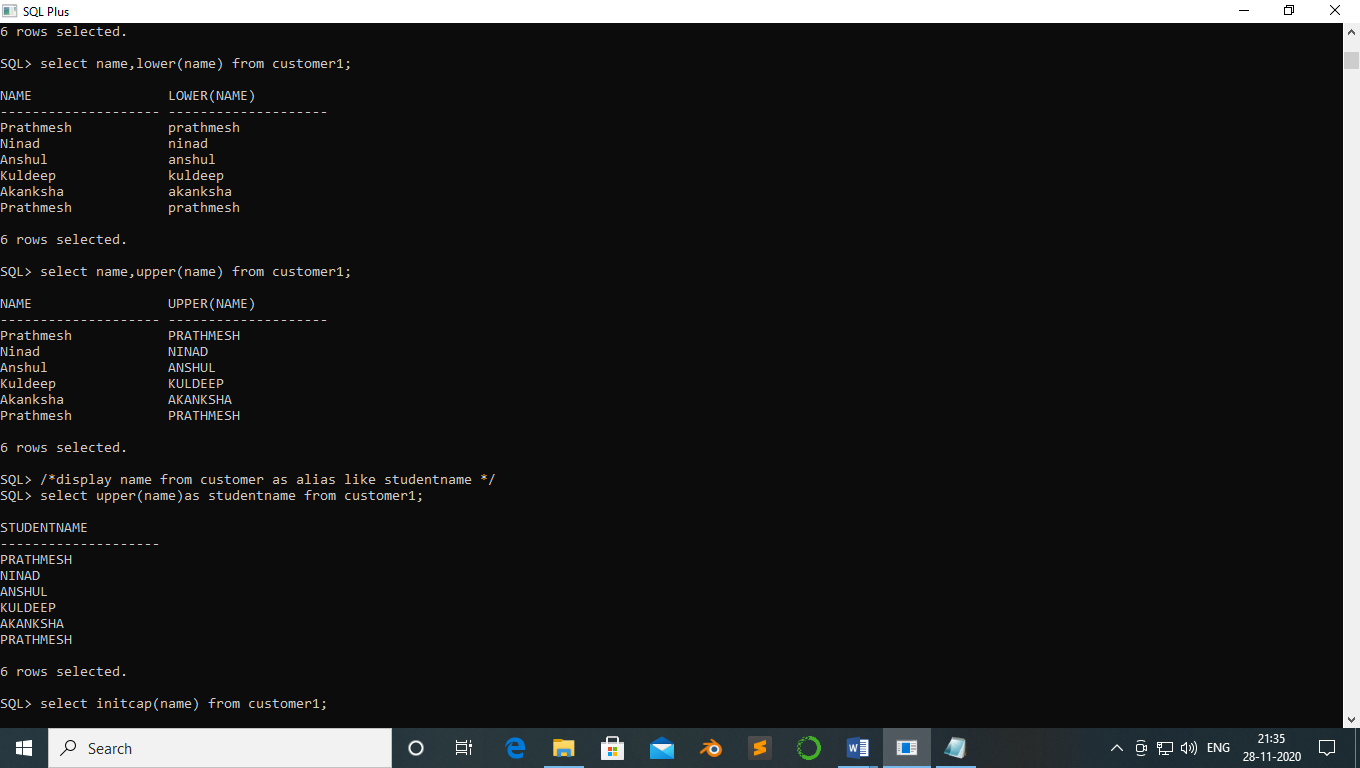
Anshul ANSHUL

Kuldeep KULDEEP

Akanksha AKANKSHA

Prathmesh PRATHMESH

6 rows selected.



SQL> /\*display name from customer as alias like studentname \*/

SQL> select upper(name)as studentname from customer1;

STUDENTNAME

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

PRATHMESH

NINAD

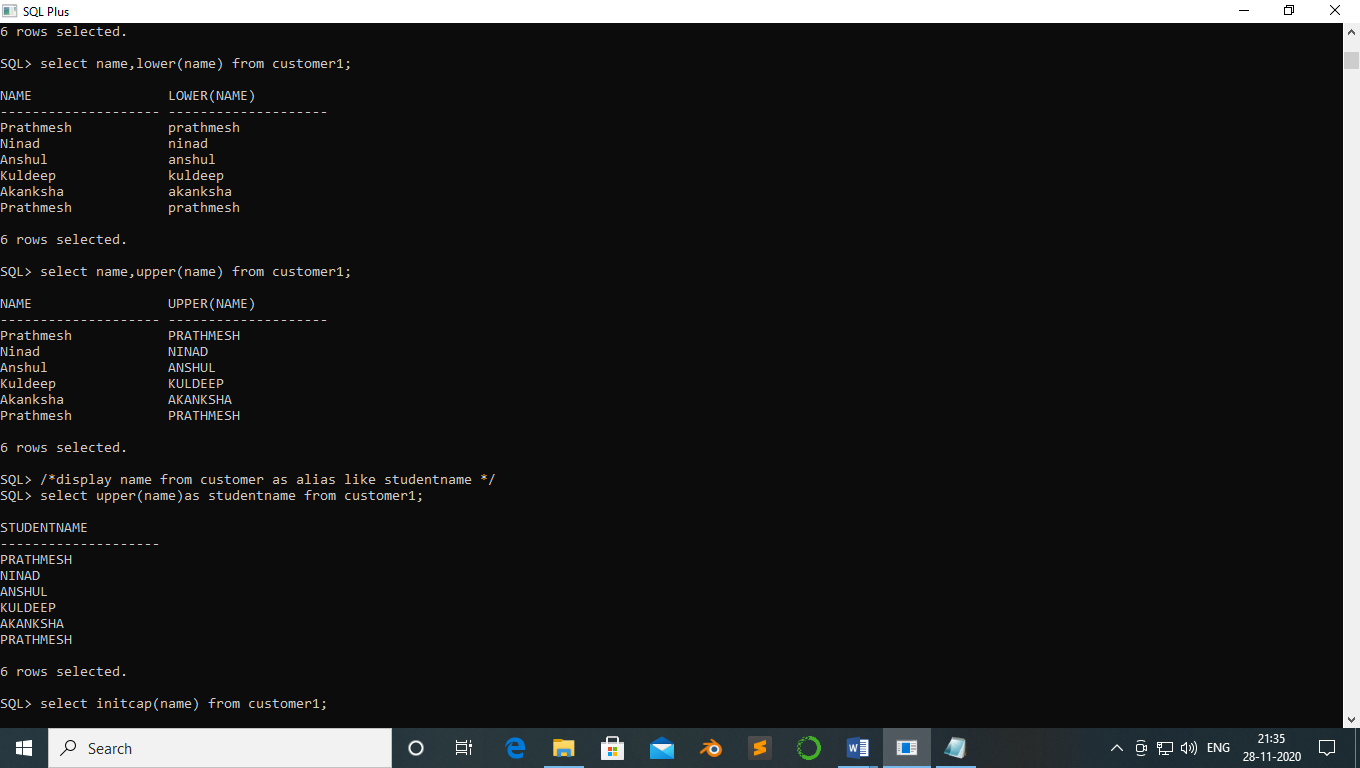
ANSHUL

KULDEEP

AKANKSHA

PRATHMESH

6 rows selected.



SQL> select initcap(name) from customer1;

INITCAP(NAME)

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

Prathmesh

Ninad

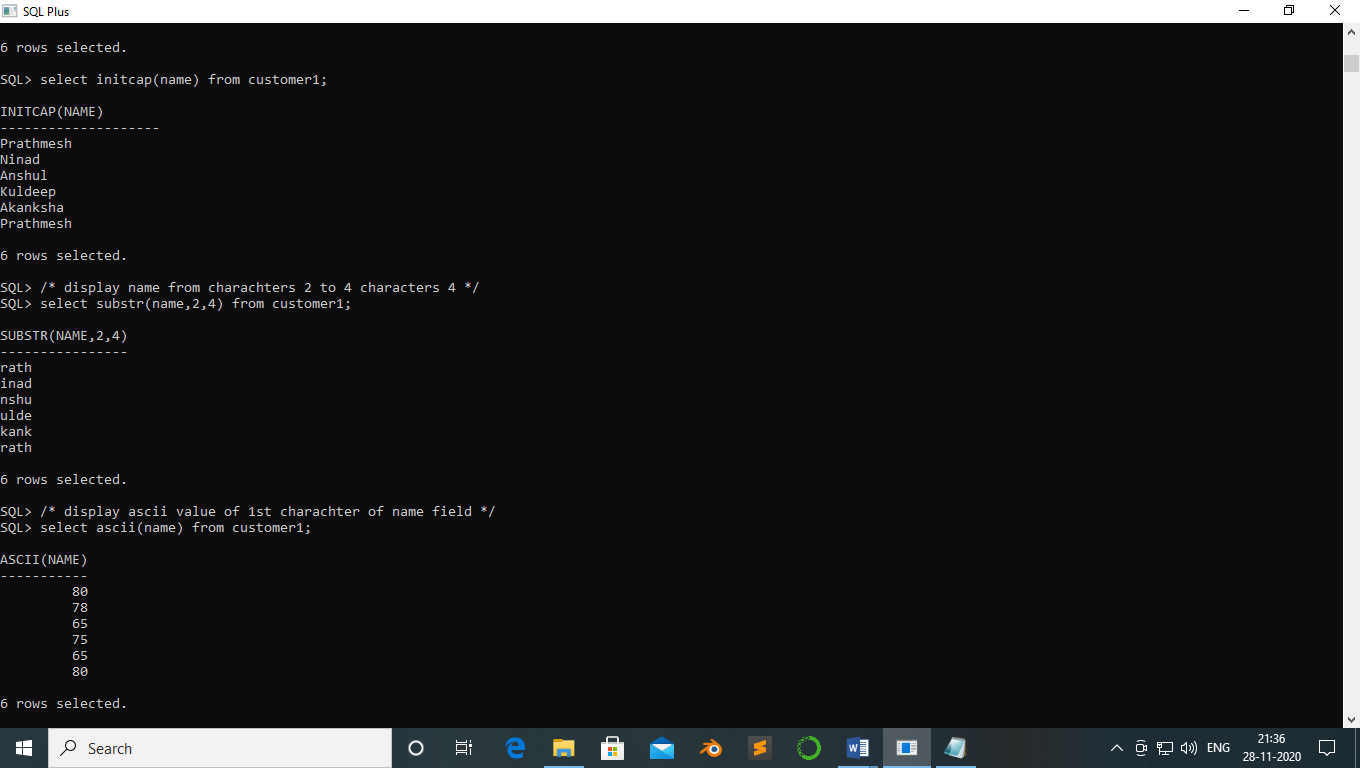
Anshul

Kuldeep

Akanksha

Prathmesh

6 rows selected.



SQL> /\* display name from charachters 2 to 4 characters 4 \*/

SQL> select substr(name,2,4) from customer1;

SUBSTR(NAME,2,4)

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

rath

inad

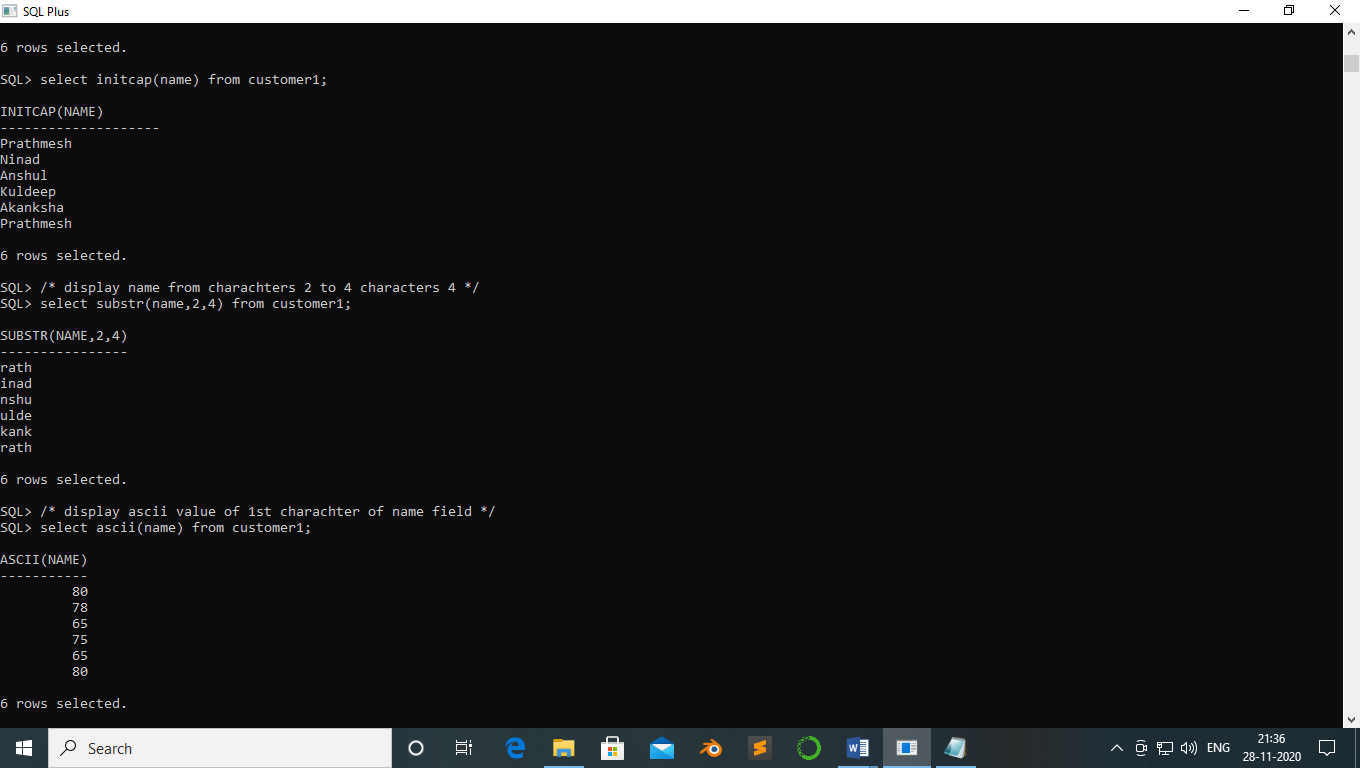
nshu

ulde

kank

rath

6 rows selected.



SQL> /\* display ascii value of 1st charachter of name field \*/

SQL> select ascii(name) from customer1;

ASCII(NAME)

-----------

80

78

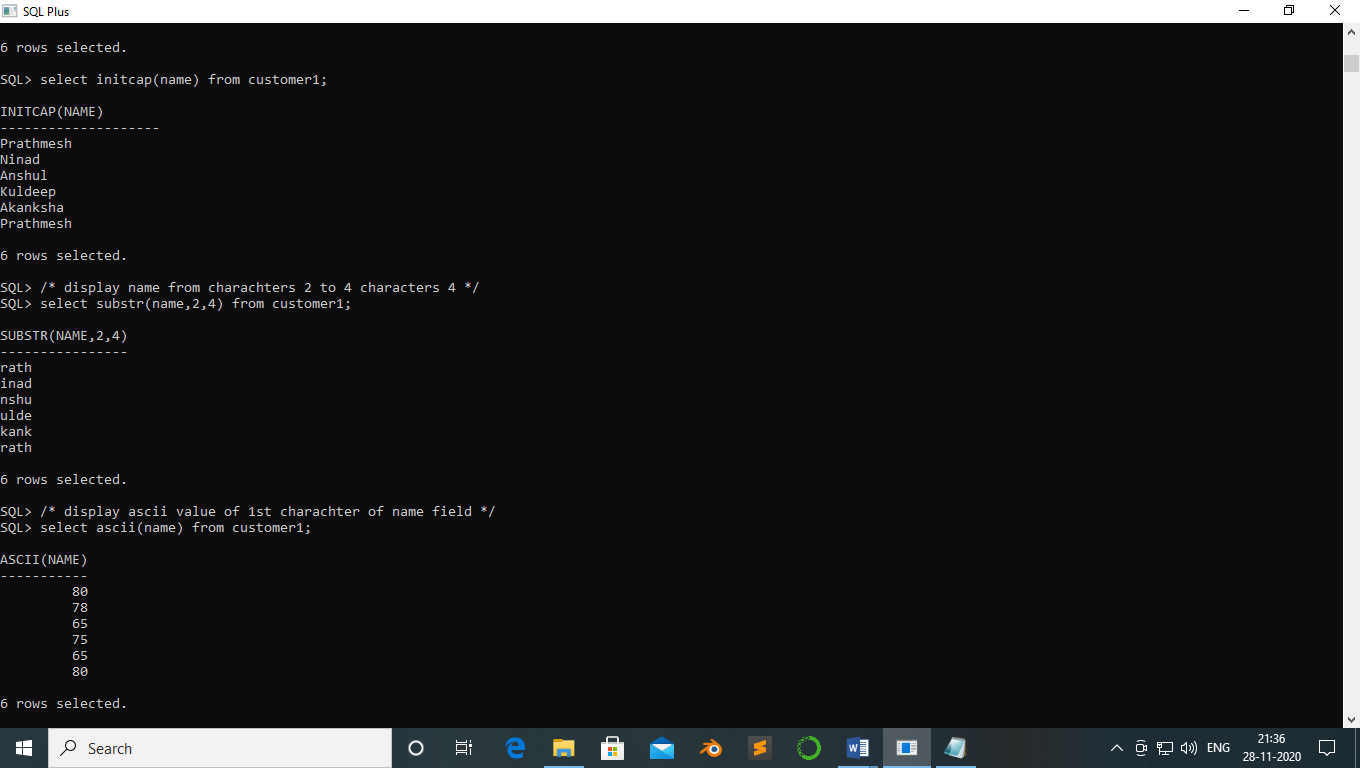
65

75

65

80

6 rows selected.



SQL> insert into customer1 values(7, 'Kaustubh','Belapur','20','65000');

1 row created.

SQL> select \* from customer1;

ID NAME ADDRESS AGE SALARY

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

1 Prathmesh Mumbai 20 40000

2 Ninad Mumbai 20 45000

3 Anshul Dehli 21 50000

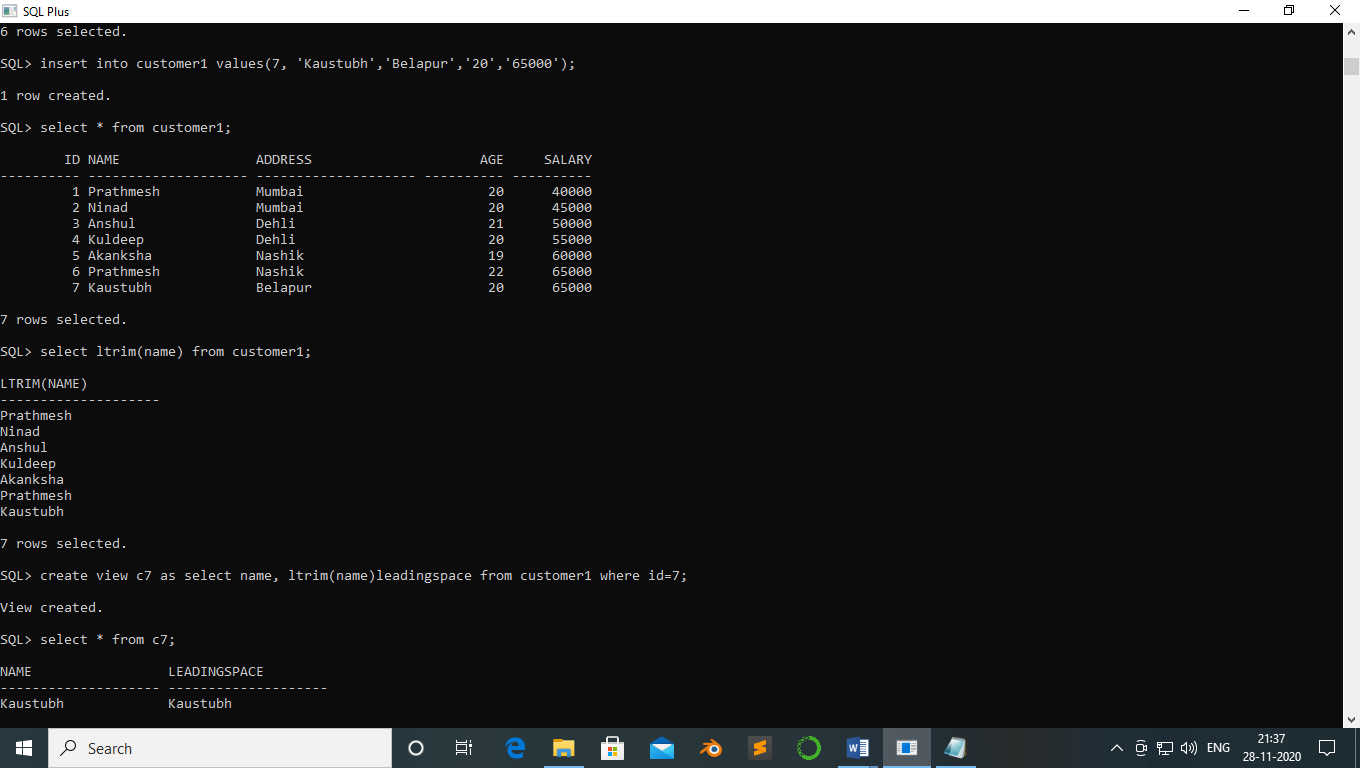
4 Kuldeep Dehli 20 55000

5 Akanksha Nashik 19 60000

6 Prathmesh Nashik 22 65000

7 Kaustubh Belapur 20 65000

7 rows selected.



SQL> select ltrim(name) from customer1;

LTRIM(NAME)

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

Prathmesh

Ninad

Anshul

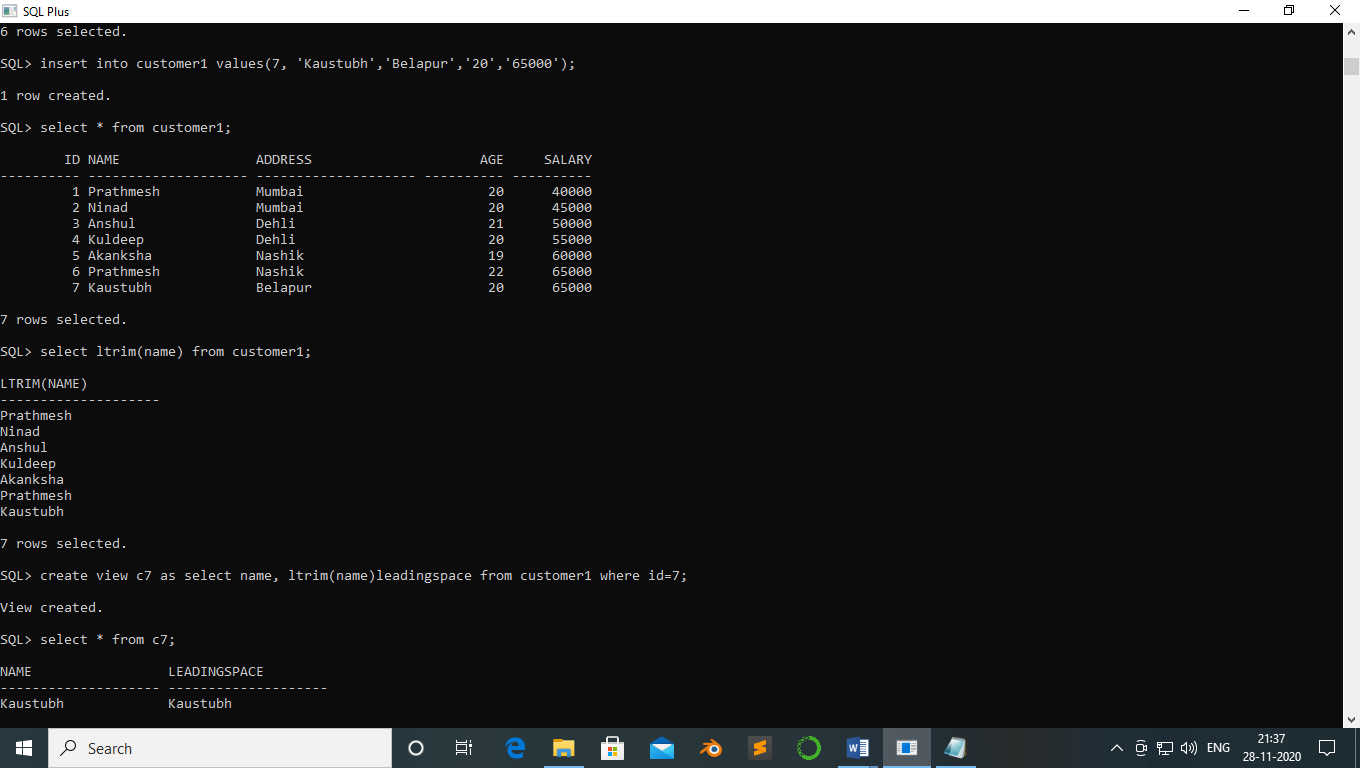
Kuldeep

Akanksha

Prathmesh

Kaustubh

7 rows selected.



SQL> create view c7 as select name, ltrim(name)leadingspace from customer1 where id=7;

View created.

SQL> select \* from c7;

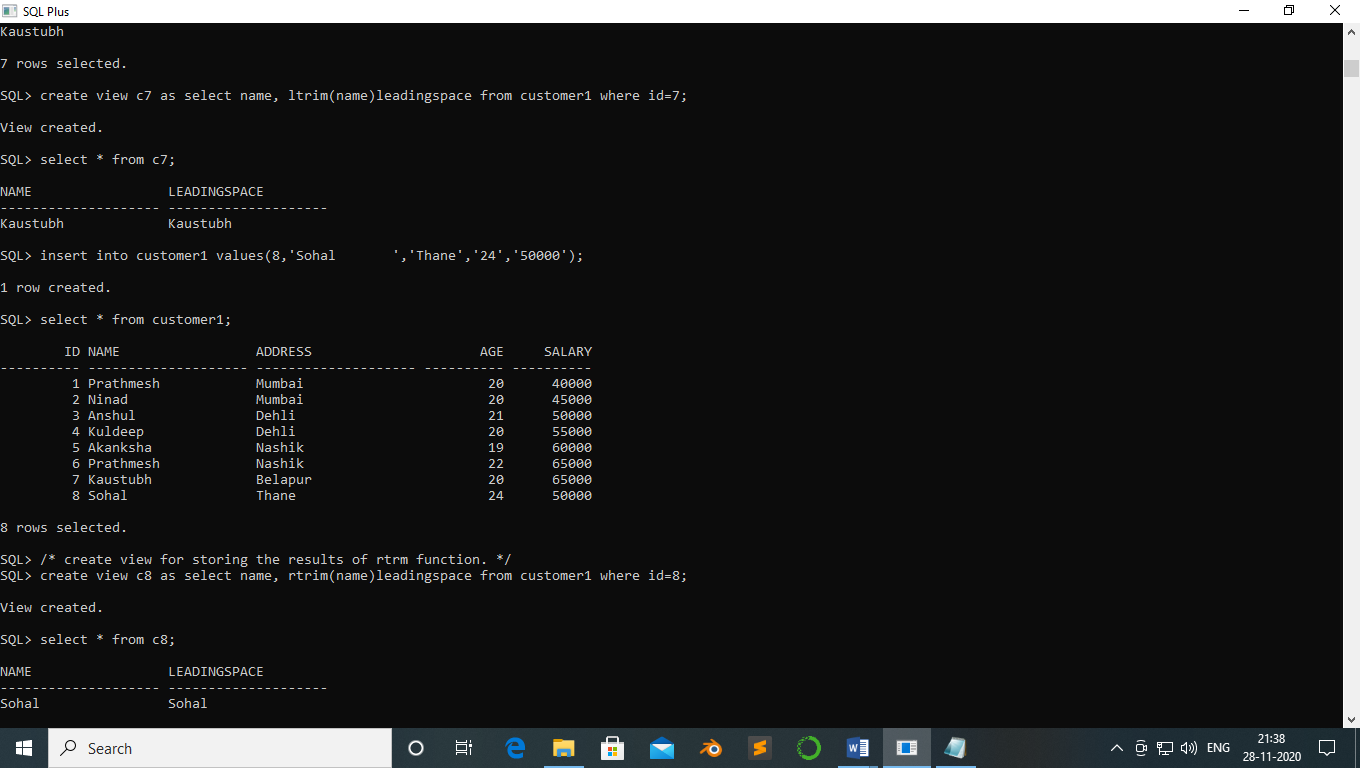
NAME LEADINGSPACE

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

Kaustubh Kaustubh

SQL> insert into customer1 values(8,'Sohal ','Thane','24','50000');

1 row created.



SQL> select \* from customer1;

ID NAME ADDRESS AGE SALARY

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

1 Prathmesh Mumbai 20 40000

2 Ninad Mumbai 20 45000

3 Anshul Dehli 21 50000

4 Kuldeep Dehli 20 55000

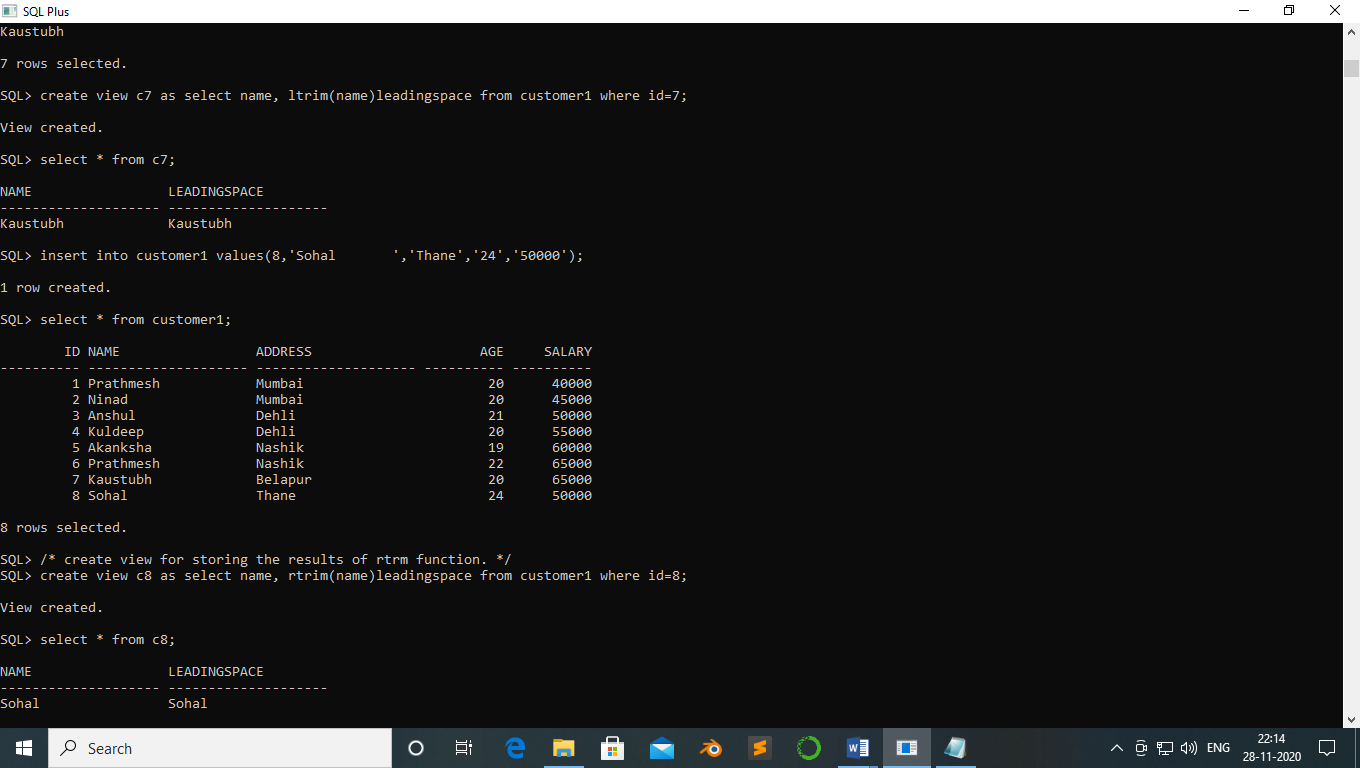
5 Akanksha Nashik 19 60000

6 Prathmesh Nashik 22 65000

7 Kaustubh Belapur 20 65000

8 Sohal Thane 24 50000

8 rows selected.



SQL> /\* create view for storing the results of rtrm function. \*/

SQL> create view c8 as select name, rtrim(name)leadingspace from customer1 where id=8;

View created.

SQL> select \* from c8;

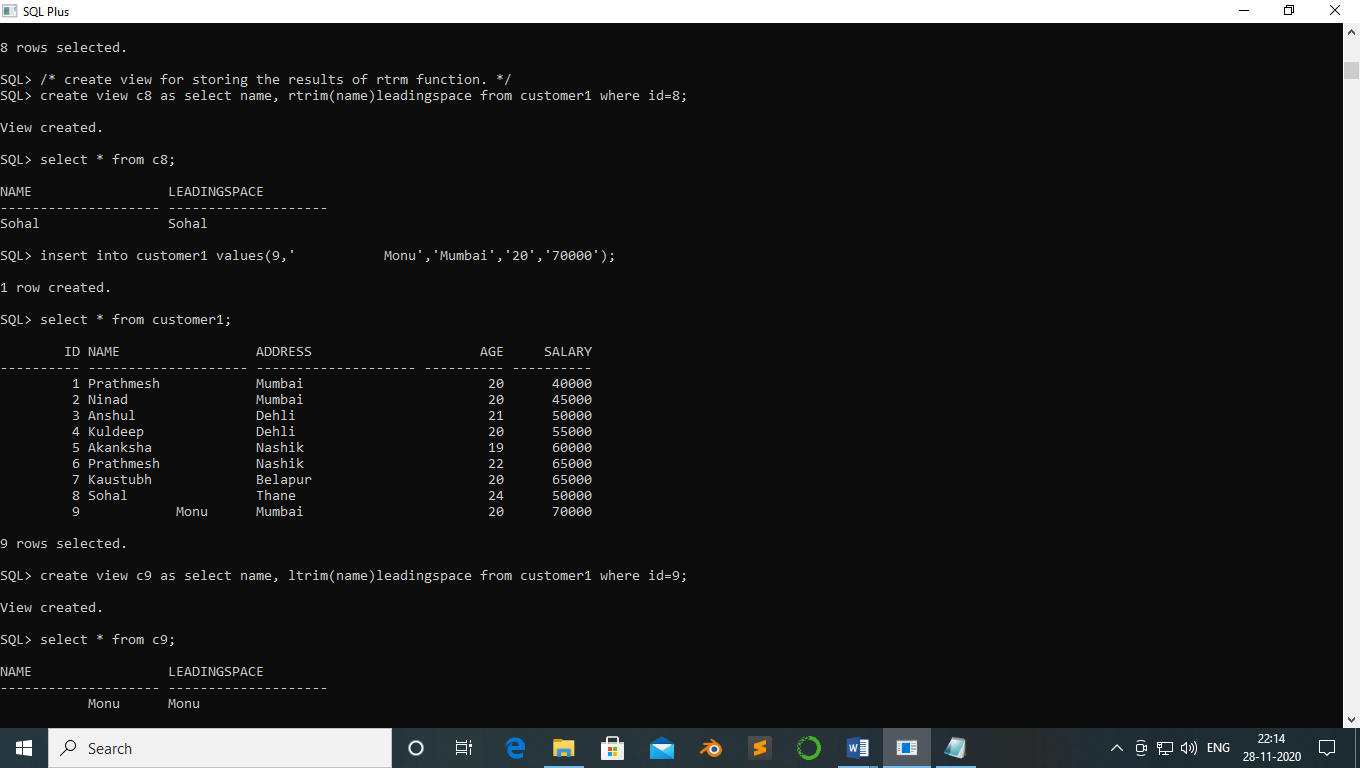
NAME LEADINGSPACE

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

Sohal Sohal

SQL> insert into customer1 values(9,' Monu','Mumbai','20','70000');

1 row created.



SQL> select \* from customer1;

ID NAME ADDRESS AGE SALARY

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

1 Prathmesh Mumbai 20 40000

2 Ninad Mumbai 20 45000

3 Anshul Dehli 21 50000

4 Kuldeep Dehli 20 55000

5 Akanksha Nashik 19 60000

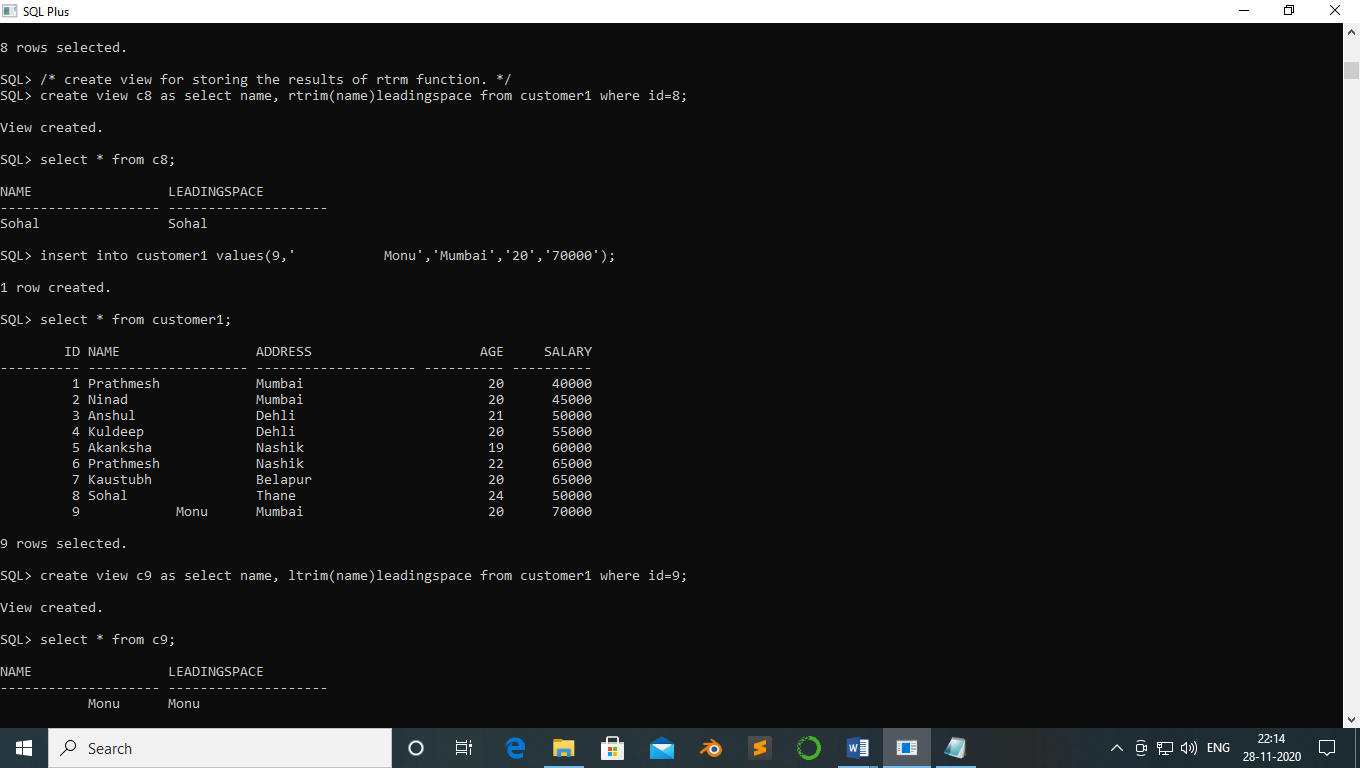
6 Prathmesh Nashik 22 65000

7 Kaustubh Belapur 20 65000

8 Sohal Thane 24 50000

9 Monu Mumbai 20 70000

9 rows selected.



SQL> create view c9 as select name, ltrim(name)leadingspace from customer1 where id=9;

View created.

SQL> select \* from c9;

NAME LEADINGSPACE

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

Monu Monu

SQL> select length(name) from customer1 where id=2;

LENGTH(NAME)

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

5

SQL> select lpad(name,9,'\*') from customer1;

LPAD(NAME,9,'\*')

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

Prathmesh

\*\*\*\*Ninad

\*\*\*Anshul

\*\*Kuldeep

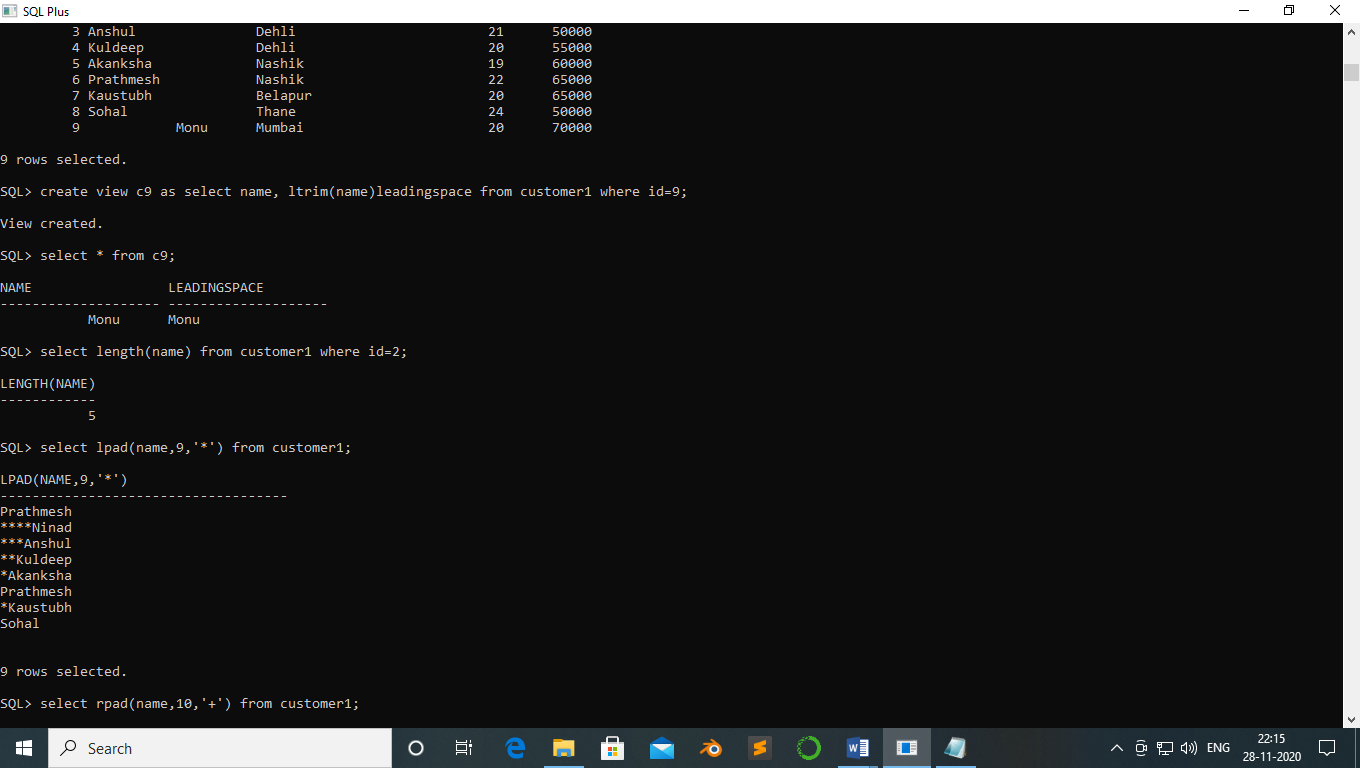
\*Akanksha

Prathmesh

\*Kaustubh

Sohal

9 rows selected.

‘

SQL> select rpad(name,10,'+') from customer1;

RPAD(NAME,10,'+')

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

Prathmesh+

Ninad+++++

Anshul++++

Kuldeep+++

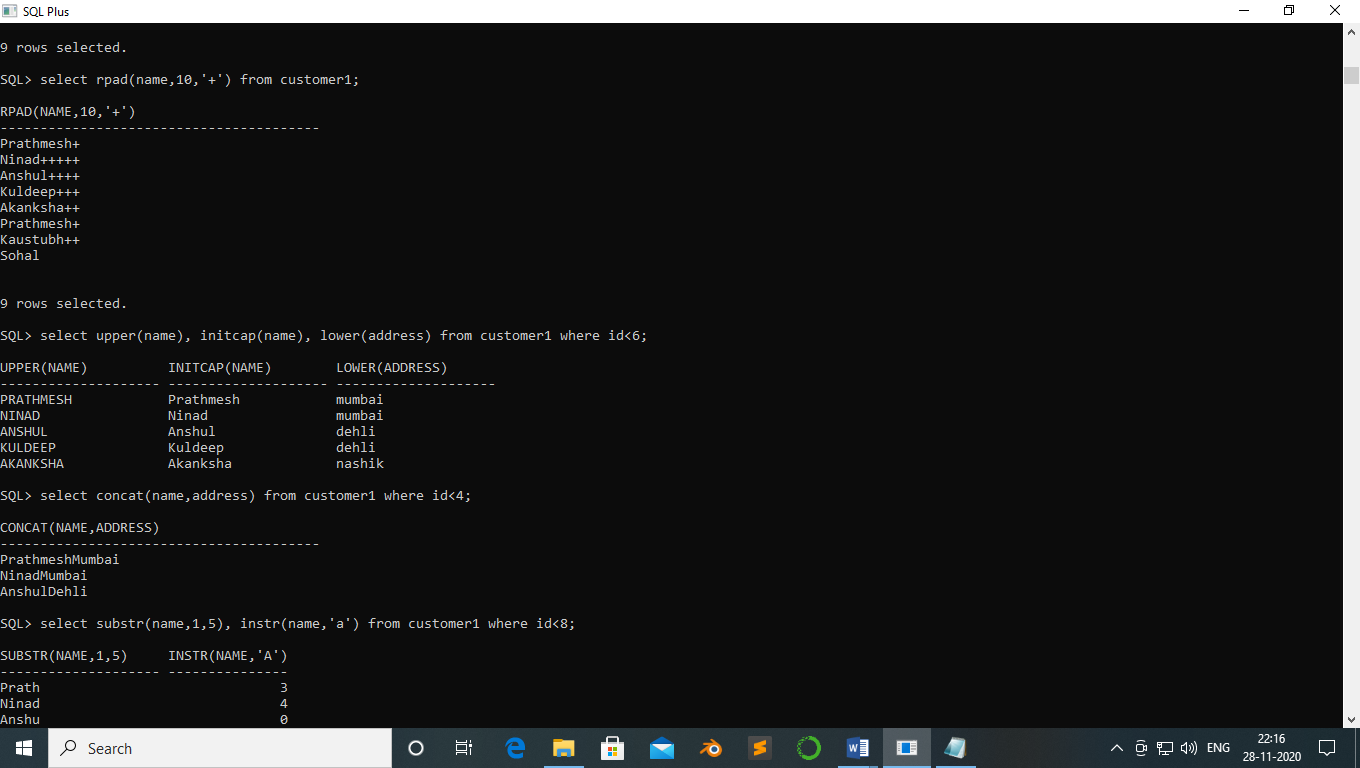
Akanksha++

Prathmesh+

Kaustubh++

Sohal

9 rows selected.



SQL> select upper(name), initcap(name), lower(address) from customer1 where id<6;

UPPER(NAME) INITCAP(NAME) LOWER(ADDRESS)

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

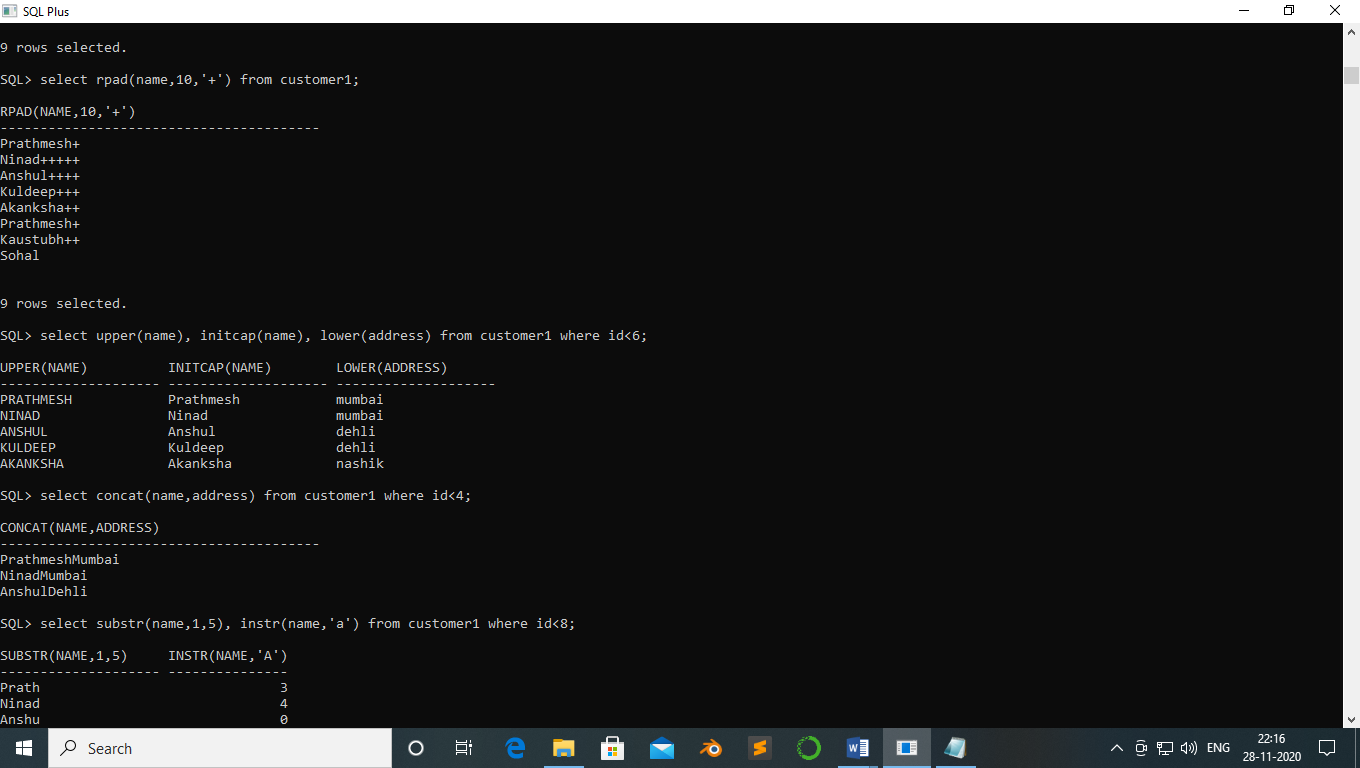
PRATHMESH Prathmesh mumbai

NINAD Ninad mumbai

ANSHUL Anshul dehli

KULDEEP Kuldeep dehli

AKANKSHA Akanksha nashik



SQL> select concat(name,address) from customer1 where id<4;

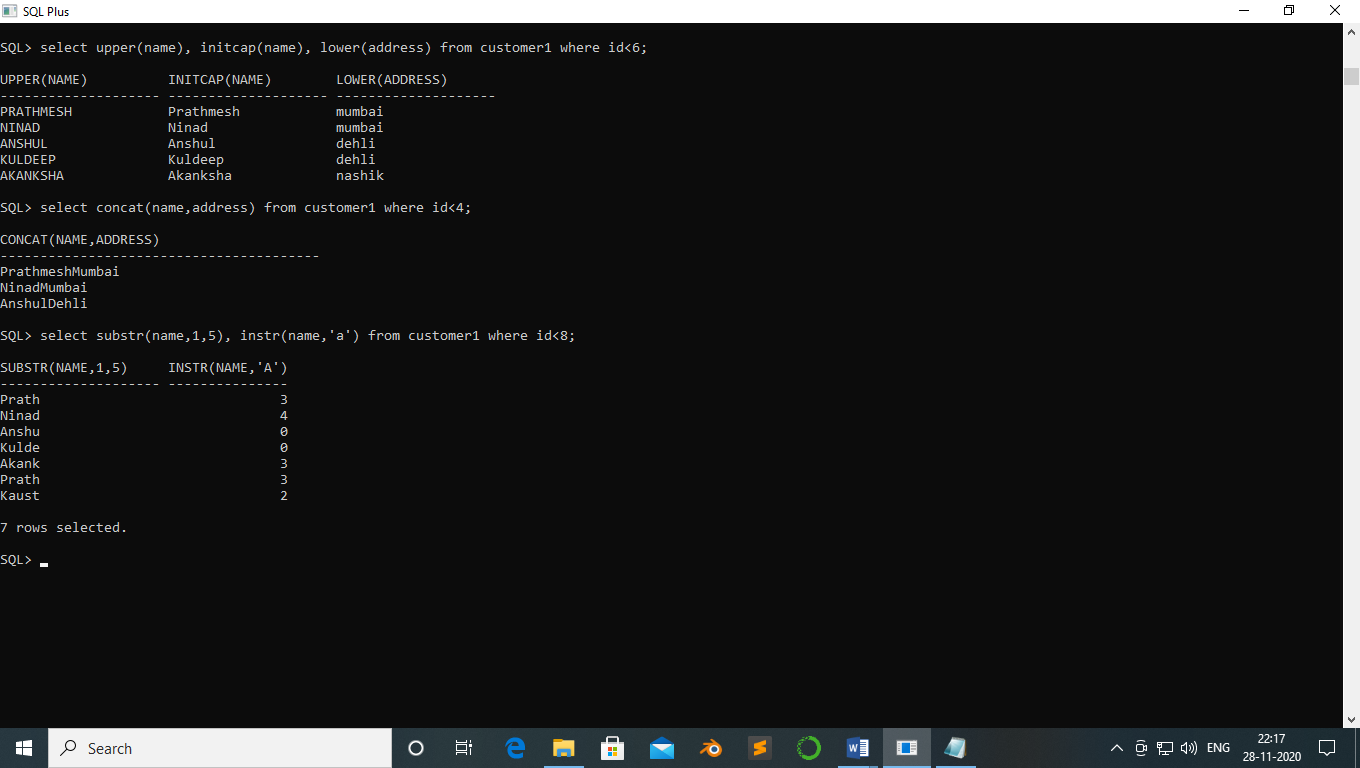
CONCAT(NAME,ADDRESS)

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

PrathmeshMumbai

NinadMumbai

AnshulDehli



SQL> select substr(name,1,5), instr(name,'a') from customer1 where id<8;

SUBSTR(NAME,1,5) INSTR(NAME,'A')

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

Prath 3

Ninad 4

Anshu 0

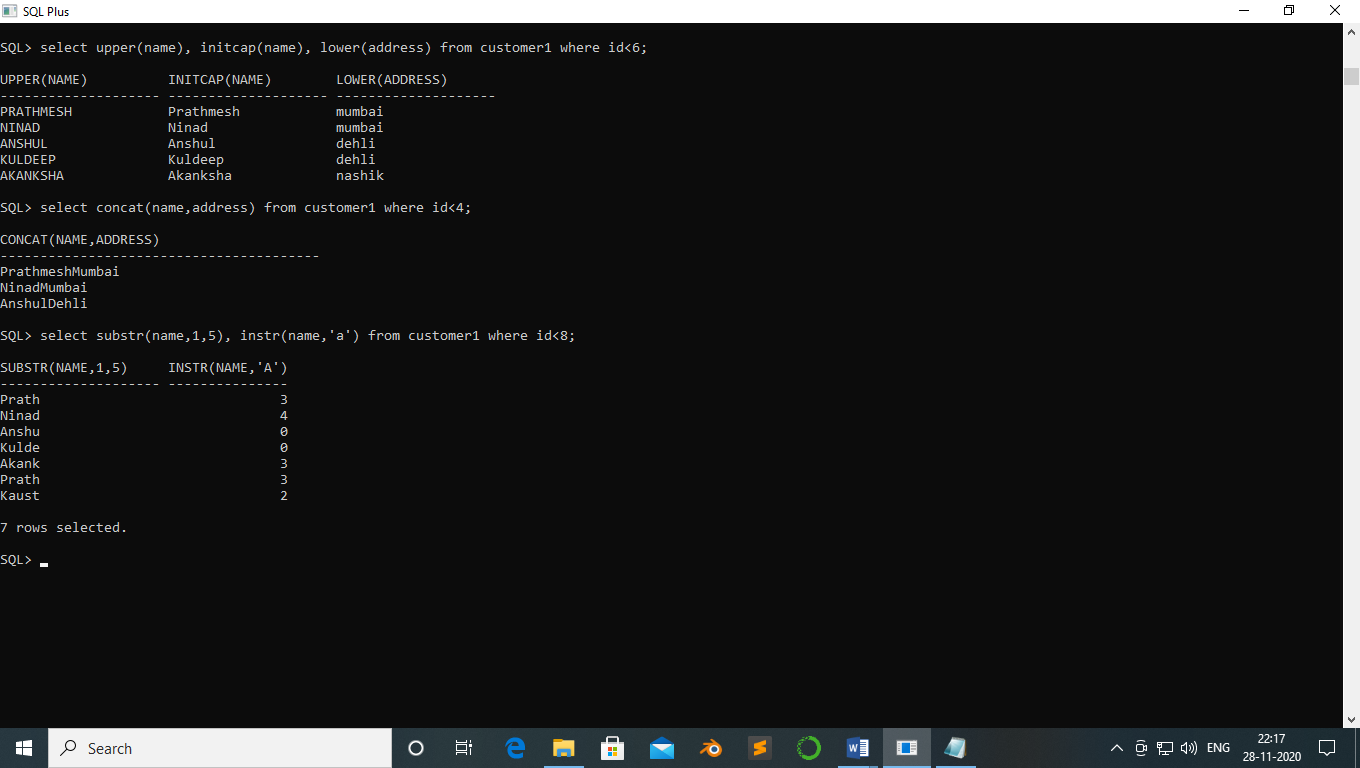
Kulde 0

Akank 3

Prath 3

Kaust 2

7 rows selected.



**Practical no 11**

Aim:- pl/sql

Theory

PL/SQL-procedural sql is a combination of SQL along with the procedural features of

programming languages. It was developed by Oracle Corporation in the early 90&#39;s to enhance the

capabilities of SQL.

Advantages of PL/SQL

PL/SQL has the following advantages −

 SQL is the standard database language and PL/SQL is strongly integrated with SQL.

PL/SQL supports both static and dynamic SQL. Static SQL supports DML operations

and transaction control from PL/SQL block. In Dynamic SQL, SQL allows embedding

DDL statements in PL/SQL blocks.

 PL/SQL allows sending an entire block of statements to the database at one time. This

reduces network traffic and provides high performance for the applications.

 PL/SQL gives high productivity to programmers as it can query, transform, and update

data in a database.

 PL/SQL saves time on design and debugging by strong features, such as exception

handling, encapsulation, data hiding, and object-oriented data types.

 Applications written in PL/SQL are fully portable.

 PL/SQL provides high security level.

 PL/SQL provides access to predefined SQL packages.

 PL/SQL provides support for Object-Oriented Programming.

 PL/SQL provides support for developing Web Applications and Server Pages.

PL/SQL which is a block-structured language; this means that the PL/SQL

programs are divided and written in logical blocks of code. Each block consists

of three sub-parts –

The ‘Hello World’ Example

DECLARE

message varchar2(20):= ‘Hello, World’;

BEGIN

dbms\_output.put\_line(message);

END;

/

DECLARE

message varchar2(20):= &message;

BEGIN

dbms\_output.put\_line(message);

END;

/

Perform addition of two numbers;

1. Assign values

2. Accept values from user

If statements:

1. IF-THEN- END IF

2. IF-THEN-ELSE-END IF

3. IF-THEN-ELSE-END IF

1. IF-THEN- END IF: It is also known as simple if statement. The simple if statement

performs action statements if the result of the condition is true.

Eg:

SQL> set serveroutput on;

SQL>

declare

a int;

begin

a:=10; //accept the value from the user &a

if a = 10

then

a:=a+1;

end if;

dbms\_output.put\_line(a);

end;

PL/SQL procedure successfully completed.

2. IF-THEN-ELSE-END IF: This statement performs action statements for the TRUE

outcome as well as the FALSE outcome.

Syntax:

IF condition Then

Action statement 1

Else

Action statement 2

End If;

For Eg:

3. IF-THEN-ELSE-END IF:

Syntax:

IF condition1 Then

Action statement 1

ELSIF condition2 then

Action statement 2

……….

ELSIF condition then

Action statement N

[Else action statements]

END If

Ex:

1 declare

2 v\_score int;

3 v\_g varchar(10);

4 begin

5 v\_score:=70;//v\_score:=&amp;v\_score;//accepting value from user

6 if v\_score<50 and v\_score>60 then

7 v\_g:=&#39;second&#39;;

8 elsif v\_score<60 and v\_score>75 then

9 v\_g:=’first’;

10 else v\_g:=’fail’;

11 end if;

12 dbms\_output.put\_line(v\_g);

13\* end;

Ex: accept the marks of various subject from user .Find percentage and print grade.

Loops:

1. Basic Loop

2. While Loop

3. For Loop

1. Basic Loop: a basic loop is a loop that is performed repeatedly. Once loop is entered all

statements are performed.

Syntax:

Loop

Looping statement1;

…..

Exit[when condition];

End loop;

Ex:

2. While Loop: The while loop is an alternative to the basic loop. The while loop is

performed as long as the condition is true. It terminates when the condition is false. If the

condition is false at the beginning of the loop, the loop is not performed at all. The while

loop does not needed an Exit statement.

Syntax:

While condition Loop

Looping statement1;

Looping statement2;

…

Looping statement;

End Loop;

Program:

For Loop:

Couter does not needed to write separately. The counter in the loop is

implicitly declared as an integer. And destroyed in the loop’s termination.

Syntax:

FOR counter IN[Reverse]lower…upper LOOP

Looping statement1;

Looping statement2;

….

Looping statementn;

END LOOP;

Q1. Write a pl/sql block to find the maximum of 3 number

Code :-

SQL> set serveroutput on;

SQL> declare

2 a number;

3 b number;

4 c number;

5 begin

6 a:=&a;

7 b:=&b;

8 c:=&c;

9 if(a>b and a>c) then

10 dbms\_output.put\_line('a is maximum: '||a);

11 elsif(b>a and b>c) then

12 dbms\_output.put\_line('b is maximum: '||b);

13 else

14 dbms\_output.put\_line('c is maximum: '||c);

15 end if;

16 end;

17 /

Enter value for a: 30

old 6: a:=&a;

new 6: a:=30;

Enter value for b: 20

old 7: b:=&b;

new 7: b:=20;

Enter value for c: 40

old 8: c:=&c;

new 8: c:=40;

c is maximum: 40

PL/SQL procedure successfully completed.

Q2 write a pl/sql block to find the sum of first 100 natural numbers

Code:-

SQL> declare

2 a number:=0;

3 begin

4 for i in 1..100

5 loop

6 a:=a+i;

7 end loop;

8 dbms\_output.put\_line('The sum of 1st 100 natural numbers are: '||a);

9 end;

10 /

The sum of 1st 100 natural numbers are: 5050

PL/SQL procedure successfully completed.

Q3 write a ppl/sql block to take score and display the grade

Code :-

SQL> declare

2 v\_score int;

3 v\_g varchar(10);

4 begin

5 v\_score:=&v\_score;

6 if v\_score>50 and v\_score<60 then

7 v\_g:='average';

8 elsif v\_score>=60 and v\_score<75 then

9 v\_g:='extension';

10 elsif v\_score>=75 and v\_score<100 then

11 v\_g:='exellent';

12 elsif v\_score>=35 and v\_score<50 then

13 v\_g:='pass';

14 else v\_g:='fail';

15 end if;

16 dbms\_output.put\_line(v\_g);

17 end;

18 /

Enter value for v\_score: 80

old 5: v\_score:=&v\_score;

new 5: v\_score:=80;

exellent

PL/SQL procedure successfully completed.

Q4 write a pl/sql block to display the table detail of a employee whose eno given by user

Code :-

SQL> declare

2 no number;

3 n number;

4 name varchar(20);

5 add varchar(20);

6 ct varchar(20);

7 begin

8 n:=&n;

9 select sno,sname,address,city into no,name,add,ct from stude

10 where sno=n;

11 dbms\_output.put\_line('The sno is: '||no);

12 dbms\_output.put\_line('The address is: '||add);

13 dbms\_output.put\_line('The sname is: '||name);

14 dbms\_output.put\_line('The city is: '||ct);

15 end;

16 /

Enter value for n: 1

old 8: n:=&n;

new 8: n:=1;

The sno is: 1

The address is: sec-19/20

The sname is: Prathmesh

The city is: Belapur

PL/SQL procedure successfully completed.

Q5 write a pl/sql block to increase the salary of employee by 20 percent whose eno given by user

Code:-

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 96000

2 Ninad sec-20 Sanpada 50000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

SQL> declare

2 n number;

3 s int;

4 begin

5 n:=&n;

6 select salary into s from emp where eno=n;

7 update emp set salary=salary+(salary\*0.20) where eno=n;

8 end;

9 /

Enter value for n: 2

old 5: n:=&n;

new 5: n:=2;

PL/SQL procedure successfully completed.

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 96000

2 Ninad sec-20 Sanpada 60000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

Aim :- PL/SQL

Theory:-

**The %ROWTYPE attribute provides a record type that represents a row in a database table. The record can store an entire row of data selected from the table or fetched from a cursor or cursor variable.**Variables declared using **%ROWTYPE** are treated like those declared using a datatype name. You can use the **%ROWTYPE** attribute in variable declarations as a datatype specifier.

Code :-

1. Write a block that accept eno from user and display the salary of that emp from user

Using type.

Ans:-

SQL> set serveroutput on;

SQL> declare

2 sal\_var emp.salary%type;

3 empno emp.eno%type;

4 begin

5 select salary into sal\_var from emp where eno=&empno;

6 dbms\_output.put\_line('salary is: '||sal\_var);

7 END;

8 /

Enter value for empno: 1

old 5: select salary into sal\_var from emp where eno=&empno;

new 5: select salary into sal\_var from emp where eno=1;

salary is: 40000

PL/SQL procedure successfully completed.

1. write a block which accept the eno and update salary of that employee to new slaary entered by

the user.

Ans:-

SQL> declare

2 sal\_var emp.salary%type;

3 empno emp.eno%type;

4 begin

5 update emp set salary=salary+1000 where eno=&empno;

6 END;

7 /

Enter value for empno: 1

old 5: update emp set salary=salary+1000 where eno=&empno;

new 5: update emp set salary=salary+1000 where eno=1;

PL/SQL procedure successfully completed.

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 41000

2 Ninad sec-20 Sanpada 50000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

1. Write a PL/SQL block which accept the eno and update salary

Ans:-

SQL> declare

2 old\_sal emp.salary%type;

3 new\_sal emp.salary%type;

4 empno emp.eno%type;

5 begin

6 empno:=&eno;

7 new\_sal:=&new\_salary;

8 select salary into old\_sal from emp where eno=empno;

9 update emp set salary=new\_sal where eno=empno;

10 dbms\_output.put\_line('salary of eno: '||empno||' has been changed from '||old\_sal||' to '||new\_sal);

11 commit;

12 END;

13 /

Enter value for eno: 1

old 6: empno:=&eno;

new 6: empno:=1;

Enter value for new\_salary: 80000

old 7: new\_sal:=&new\_salary;

new 7: new\_sal:=80000;

salary of eno: 1 has been changed from 41000 to 80000

PL/SQL procedure successfully completed.

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 80000

2 Ninad sec-20 Sanpada 50000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

1. Write a PL/SQL block using %ROWTYPE to display all coloumn values .eno entered by the user.

Ans:-

SQL> declare

2 e1 emp%ROWTYPE;

3 begin

4 select eno,ename,salary into e1.eno,e1.ename,e1.salary from emp where eno=&e;

5 dbms\_output.put\_line('empno is: '||e1.eno);

6 dbms\_output.put\_line('ename is: '||e1.ename);

7 dbms\_output.put\_line('salary is: '||e1.salary);

8 END;

9 /

Enter value for e: 1

old 4: select eno,ename,salary into e1.eno,e1.ename,e1.salary from emp where eno=&e;

new 4: select eno,ename,salary into e1.eno,e1.ename,e1.salary from emp where eno=1;

empno is: 1

ename is: Prathmesh

salary is: 80000

PL/SQL procedure successfully completed.

1. Get the age from user and display message whether user is eligible for voting or not

Ans:-

SQL> declare

2 age number(3);

3 begin

4 age:=&age;

5 if age>18 then

6 dbms\_output.put\_line('Eligible for voting');

7 else

8 dbms\_output.put\_line('Not eligible for voting');

9 end if;

10 end;

11 /

Enter value for age: 19

old 4: age:=&age;

new 4: age:=19;

Eligible for voting

PL/SQL procedure successfully completed.

1. Write a block which accepts one no from user and display the message whether that no is prime or not

Ans:-

SQL> declare

2 n number;

3 i number;

4 flag number;

5 begin

6 i:=2;

7 flag:=1;

8 n:=&n;

9 for i in 2..n/2

10 loop

11 if mod(n,i)=0

12 then

13 flag=0;

14 exit;

15 end if;

16 end loop;

17 if flag=1

18 then

19 dbms\_output.put\_line('Entered no is Prime');

20 else

21 dbms\_output.put\_line('Entered ni is not Prime');

22 end if;

23 end;

24 /

Enter value for n: 45

old 8: n:=&n;

new 8: n:=45;

Entered no is not Prime

PL/SQL procedure successfully completed.

Aim: To implement cursor and trigger in pl/sql block.

Theory:

Trigger

In this chapter, we will discuss Triggers in PL/SQL. Triggers are stored programs,

which are automatically executed or fired when some events occur. Triggers are, in

fact, written to be executed in response to any of the following events −

A database manipulation (DML) statement (DELETE, INSERT, or

UPDATE)

A database definition (DDL) statement (CREATE, ALTER, or DROP).

A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP,

or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the

event is associated.

Benefits of Triggers

Triggers can be written for the following purposes −

Generating some derived column values automatically

Enforcing referential integrity

Event logging and storing information on table access

Auditing

Synchronous replication of tables

Imposing security authorizations

Preventing invalid transactions

Questions and Solutions:

Q1) Write a pl/sql code to define paremeterized cursor that acceptsdept name from

the user and display all employees of that dept.

Code:-

SQL> set serveroutput on;

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 96000

2 Ninad sec-20 Sanpada 60000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

Q2) Write an Implicit Cursor to accept the employee number from the user. You

have to delete this record and display the appropriate message on the following

table.Emp (eno, ename, address, city)

Ans:-

SQL> declare

2 cursor e\_emp is

3 select eno,ename,address from emp where address='&address';

4 peid emp.eno%type;

5 pename emp.ename%type;

6 paddress emp.address%type;

7 begin

8 open e\_emp;

9 if e\_emp%isopen then loop

10 fetch e\_emp into peid,pename,paddress;

11 exit when e\_emp%notfound;

12 if e\_emp%rowcount<=10 then

13 dbms\_output.put\_line(peid||''||pename||''||paddress);

14 end if;

15 end loop;

16 close e\_emp;

17 else

18 dbms\_output.put\_line('recor not found..');

19 end if;

20 end;

21 /

Enter value for address: sec-20

old 3: select eno,ename,address from emp where address='&address';

new 3: select eno,ename,address from emp where address='sec-20';

2Ninadsec-20

PL/SQL procedure successfully completed.

Q3) Create a pl/sql block to delete a record if found.

Ans:- SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 96000

2 Ninad sec-20 Sanpada 60000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

SQL> declare

2 n number;

3 begin

4 n:=&n;

5 delete from emp where eno=n;

6 if sql%found then

7 dbms\_output.put\_line('The record'||n||'success fully deleted');

8 else

9 dbms\_output.put\_line('The record'||n||'not found');

10 end if;

11 end;

12 /

Enter value for n: 2

old 4: n:=&n;

new 4: n:=2;

The record2success fully deleted

PL/SQL procedure successfully completed.

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 Prathmesh sec-19/20 Belapur 96000

3 Anshul sec-10 Nerul 60000

4 Kuldeep sec-5 Panvel 70000

5 Akanksha sec-15 Vashi 80000

Q4) Write a pl/sql block implementing triggers and updating a salary from the

table.

Ans:-

SQL> create or replace trigger display\_salary\_changes

2 before delete or insert or update on customer

3 for each row

4 when(new.id>0)

5 declare

6 sal\_diff number;

7 begin

8 sal\_diff := :new.salary - :old.salary;

9 dbms\_output.put\_line(‘Old Salary:’||:old.salary);

10 dbms\_output.put\_line(‘New Salary:’||:new.salary);

11 dbms\_output.put\_line(‘Salary difference:’|| sal\_diff);

12 end;

13 /

Trigger created.

SQL> update customer

2 set salary=salary+500

3 where id=2;

Old Salary:60000

New Salary:60500

Salary difference:500

Q5) Write a trigger on insert to convert the name into capital letters.

Ans:-

SQL> create or replace trigger t1

2 before insert or update on emp1

3 for each row

4 declare

5 oper varchar2(10);

6 begin

7 oper:=:new.ename;

8 :new.ename:=upper(oper);

9 end;

10 /

Trigger created.

SQL> select \* from emp;

ENO ENAME ADDRESS CITY SALARY

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

1 prathmesh sec-19/20 Belapur 96000

2 ninad sec-20 Sanpada 60000

3 anshul sec-10 Nerul 60000

4 kuldeep sec-5 Panvel 70000

5 akanksha sec-15 Vashi 80000

5 rows selected.