College Of Engineering Trivandrum

Application Software Development Lab

Abhishek Manoharan
S5 CSE Roll No:2

TVE17CS002

Department of Computer Science

August 4, 2019

Contents

List of Figures

Cycle 1

Exp No 6

STRING FUNCTIONS AND PATTERN MATCHING

1 Aim

Study String Functions & Pattern Matching

- SUBSTR
- LTRIM
- LOWER
- LENGTH
- RPAD
- UPPER
- INITCAP
- REVERSE
- LPAD
- RTRIM
- CONCAT
- POSITIION

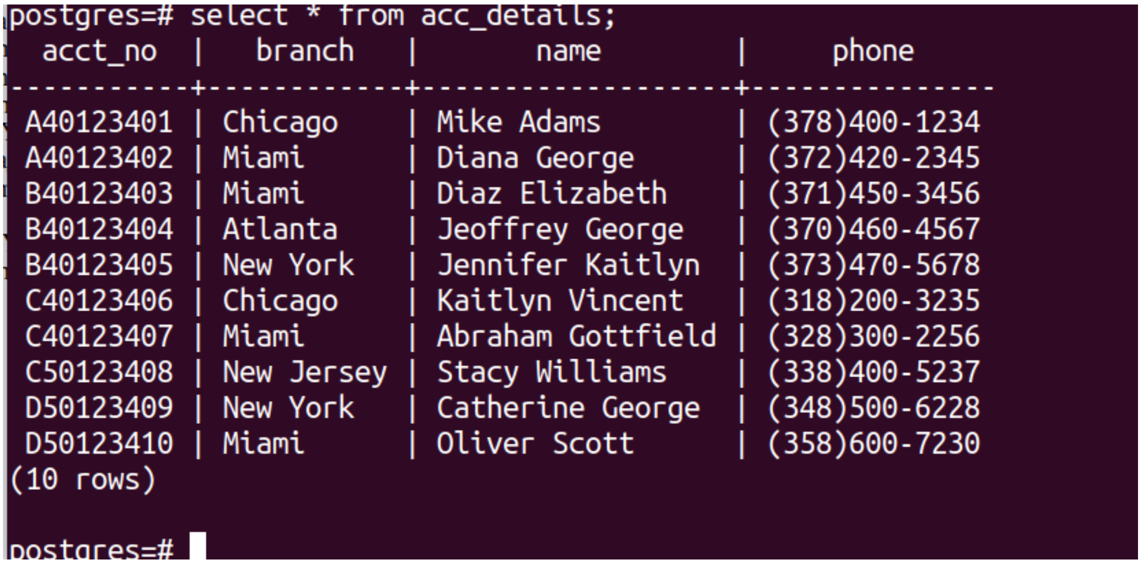
2 Questions

A. Create a table named acct_details and populate the table .

```
create table acc_details(Acct_no char(9) primary key,Branch varchar(10),
name varchar(20),phone int);


insert into acc_details values(’A40123401’,’Chicago’ ,’Mike Adams’ ,’(378)400-1234’);
insert into acc_details values(’A40123402’,’Miami’,’Diana George’,’(372)420-2345’);
insert into acc_details values(’B40123403’,’Miami’,’Diaz Elizabeth’,’(371)450-3456’);
insert into acc_details values(’B40123404’,’Atlanta’,’Jeoffrey George’,’(370)460-4567’);
insert into acc_details values(’B40123405’,’New York’,’Jennifer Kaitlyn’,’(373)470-5678’
insert into acc_details values(’C40123406’,’Chicago’,’Kaitlyn Vincent’,’(318)200-3235’);
insert into acc_details values(’C40123407’,’Miami’,’Abraham Gottfield’,’(328)300-2256’);
insert into acc_details values(’C50123408’,’New Jersey’,’Stacy Williams’,’(338)400-5237’
insert into acc_details values(’D50123409’,’New York’,’Catherine George’,’(348)500-6228’
insert into acc_details values(’D50123410’,’Miami’,’Oliver Scott’,’(358)600-7230’);
```

0. Display the Table

```
select * from acc_details;
```

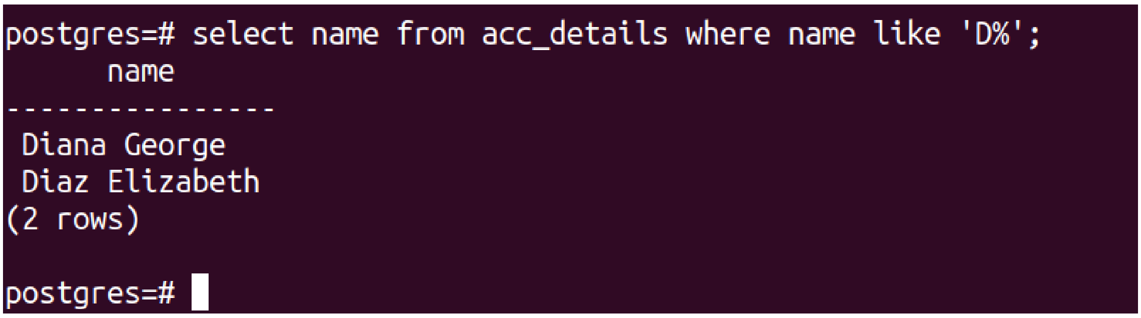


```
postgres=# select * from acc_details;
  acct_no  |   branch   |       name        |     phone
-----------+------------+-------------------+---------------
 A40123401 | Chicago    | Mike Adams        | (378)400-1234
 A40123402 | Miami      | Diana George      | (372)420-2345
 B40123403 | Miami      | Diaz Elizabeth    | (371)450-3456
 B40123404 | Atlanta    | Jeoffrey George   | (370)460-4567
 B40123405 | New York   | Jennifer Kaitlyn  | (373)470-5678
 C40123406 | Chicago    | Kaitlyn Vincent   | (318)200-3235
 C40123407 | Miami      | Abraham Gottfield | (328)300-2256
 C50123408 | New Jersey | Stacy Williams    | (338)400-5237
 D50123409 | New York   | Catherine George  | (348)500-6228
 D50123410 | Miami      | Oliver Scott      | (358)600-7230
(10 rows)

postgres=#
```

Figure 1: acc_details Table

1. Find the names of all people starting on the alphabet ‘D’.

```
select name from acc_details where name like ’D%’;
```

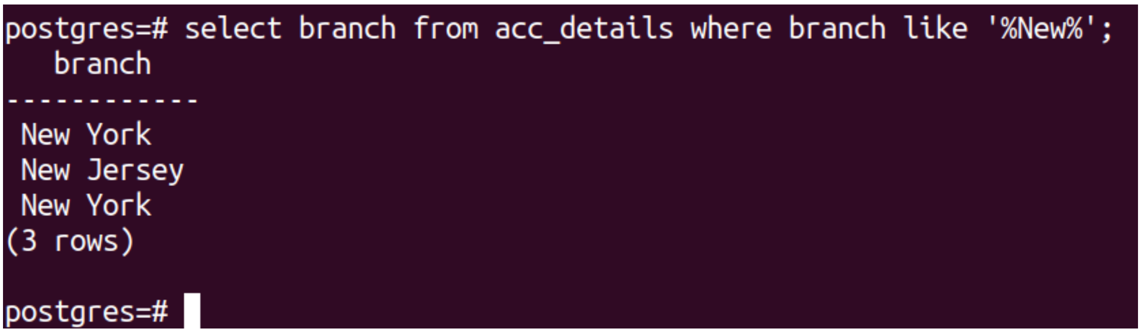


```
postgres=# select name from acc_details where name like 'D%';
      name
----------------
 Diana George
 Diaz Elizabeth
(2 rows)

postgres=#
```

Figure 2: People with name starting in D

2. List the names of all branches containing the substring ‘New’

```
select branch from acc_details where branch like ’%New%’;
```

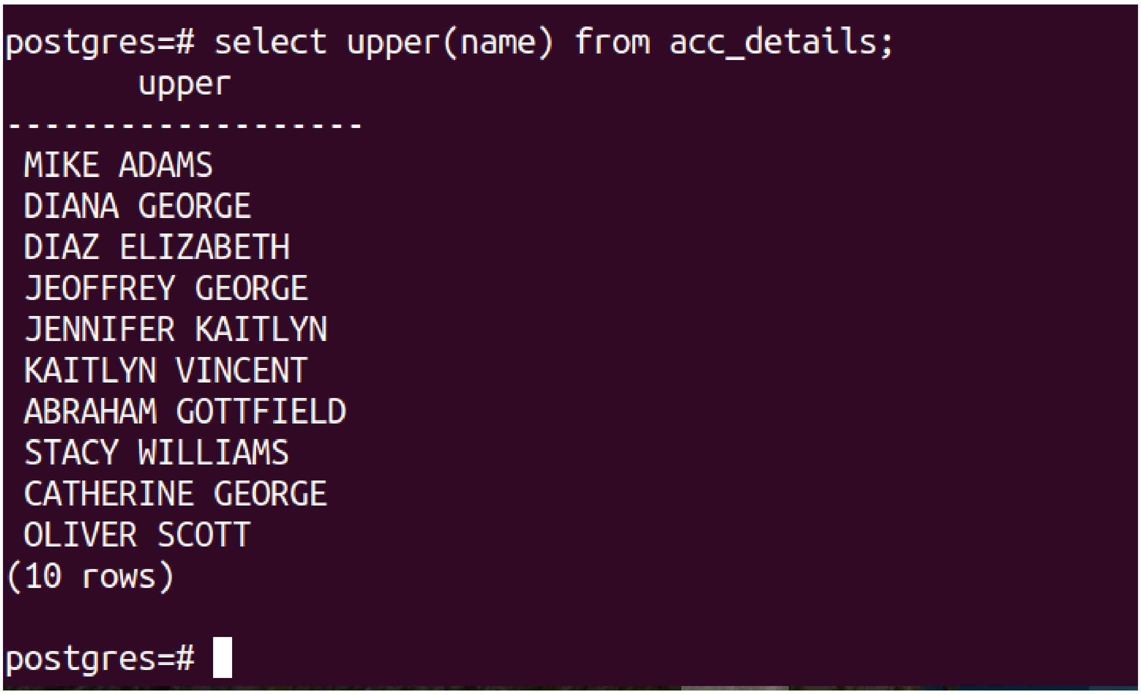


```
postgres=# select branch from acc_details where branch like '%New%';
   branch
------------
 New York
 New Jersey
 New York
(3 rows)

postgres=#
```

Figure 3: Branch name containing new

3. List all the names in Upper Case Format

```
select upper(name) from acc_details;
```

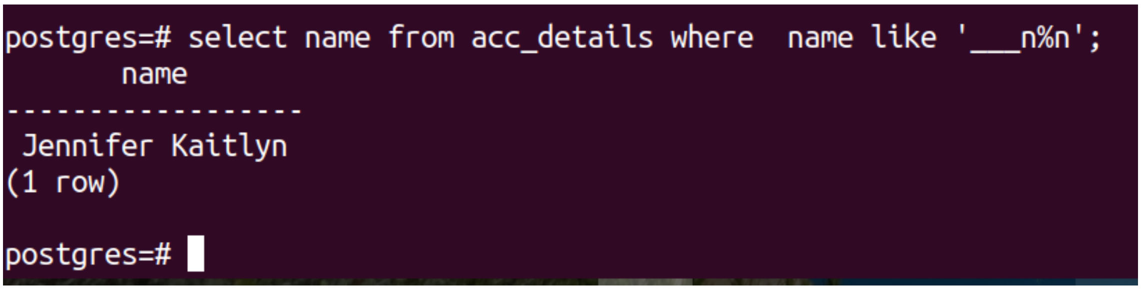


```
postgres=# select upper(name) from acc_details;
       upper
-------------------
 MIKE ADAMS
 DIANA GEORGE
 DIAZ ELIZABETH
 JEOFFREY GEORGE
 JENNIFER KAITLYN
 KAITLYN VINCENT
 ABRAHAM GOTTFIELD
 STACY WILLIAMS
 CATHERINE GEORGE
 OLIVER SCOTT
(10 rows)

postgres=# █
```

Figure 4: Names in Upper case

4. List the names where the 4th letter is ‘n’ and last letter is ‘n’

```
select name from acc_details where name like ’___n%n’;
```

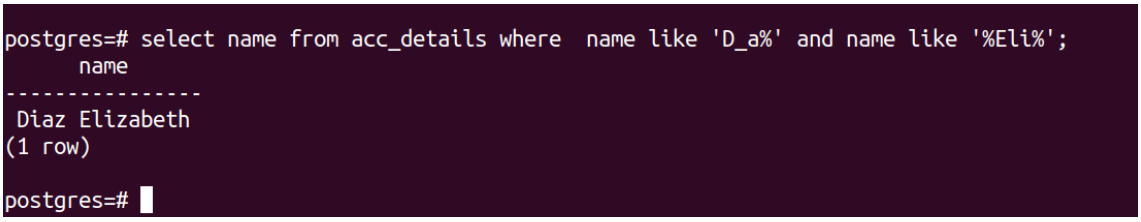


```
postgres=# select name from acc_details where  name like '___n%n';
       name
------------------
 Jennifer Kaitlyn
(1 row)

postgres=# █
```

Figure 5: Fourth and last ’n’

5. List the names starting on ‘D’ , 3 rd letter is ‘a’ and contains the substring ‘Eli’

```
select name from acc_details where name like ’D_a%’ and name like ’%Eli%’;
```

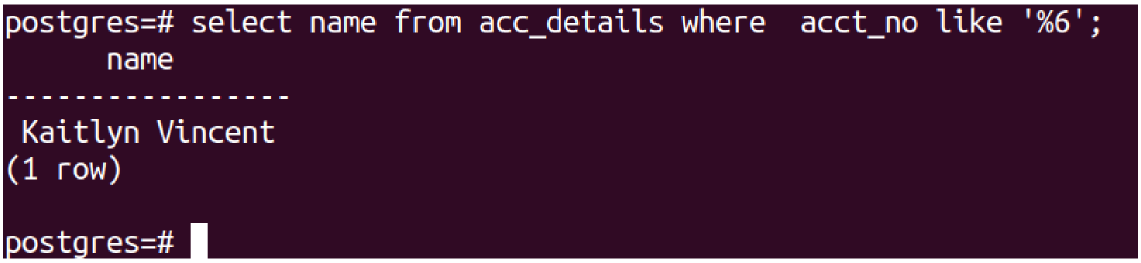


```
postgres=# select name from acc_details where  name like 'D_a%' and name like '%Eli%';
      name
---------------
 Diaz Elizabeth
(1 row)

postgres=# █
```

Figure 6: D__a and substring eli

6. List the names of people whose account number ends in ‘6’.

```
select name from acc_details where acct_no like ’%6’;
```

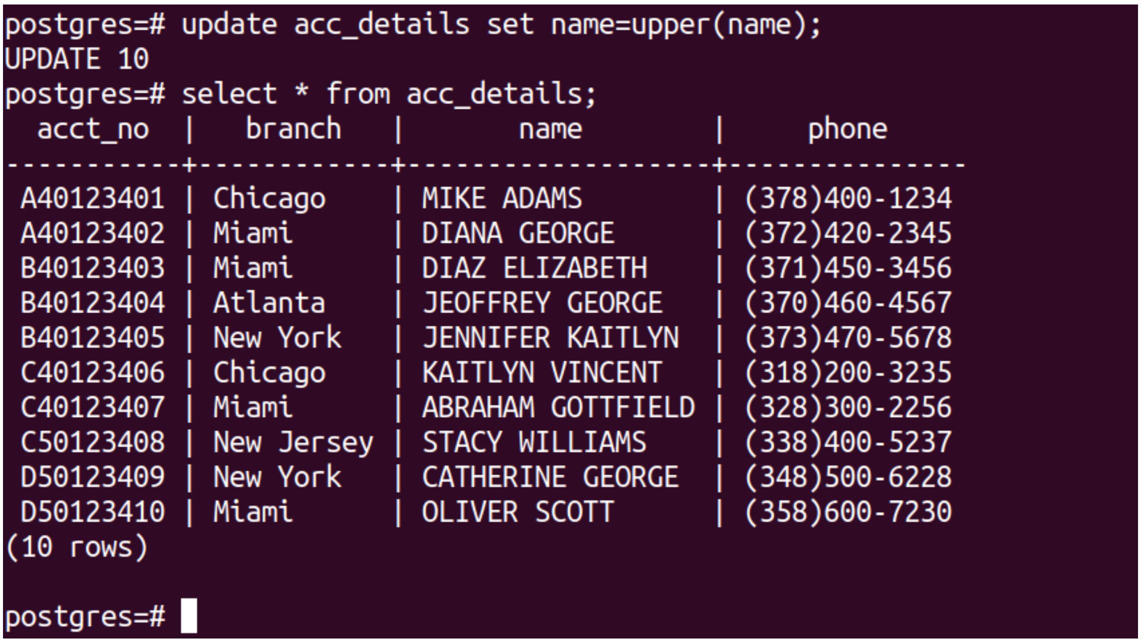


```
postgres=# select name from acc_details where  acct_no like '%6';
      name
-----------------
 Kaitlyn Vincent
(1 row)

postgres=# 
```

Figure 7: Account number ends in 6

7. Update the table so that all the names are in Upper Case Format

```
update acc_details set name=upper(name);
```

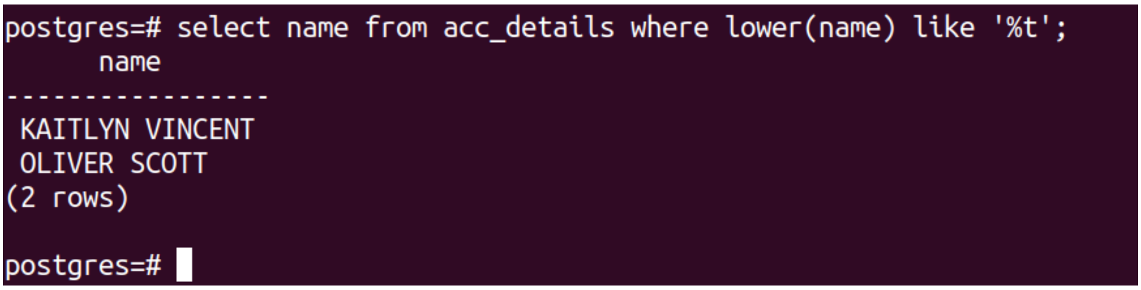


```
postgres=# update acc_details set name=upper(name);
UPDATE 10
postgres=# select * from acc_details;
  acct_no   |   branch   |       name        |     phone
------------+------------+-------------------+---------------
 A40123401 | Chicago    | MIKE ADAMS        | (378)400-1234
 A40123402 | Miami      | DIANA GEORGE      | (372)420-2345
 B40123403 | Miami      | DIAZ ELIZABETH    | (371)450-3456
 B40123404 | Atlanta    | JEOFFREY GEORGE   | (370)460-4567
 B40123405 | New York   | JENNIFER KAITLYN  | (373)470-5678
 C40123406 | Chicago    | KAITLYN VINCENT   | (318)200-3235
 C40123407 | Miami      | ABRAHAM GOTTFIELD | (328)300-2256
 C50123408 | New Jersey | STACY WILLIAMS    | (338)400-5237
 D50123409 | New York   | CATHERINE GEORGE  | (348)500-6228
 D50123410 | Miami      | OLIVER SCOTT      | (358)600-7230
(10 rows)

postgres=# 
```

Figure 8: Updating name into upper case

8. List the names of all people ending on the alphabet ‘t’;

```
select name from acc_details where lower(name) like ’%t’
```

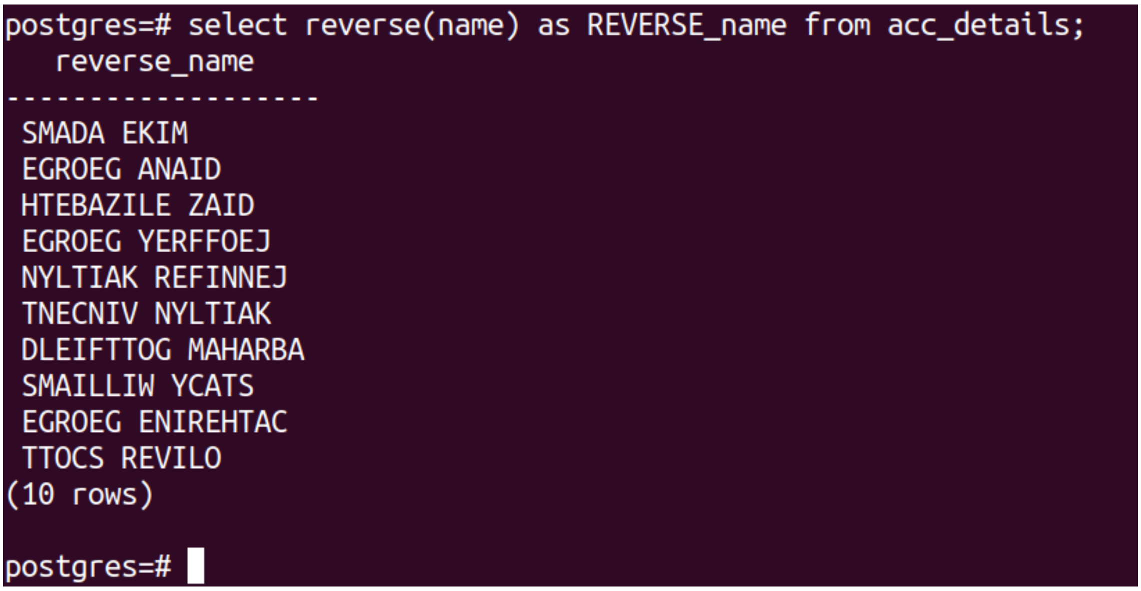


```
postgres=# select name from acc_details where lower(name) like '%t';
      name
-----------------
 KAITLYN VINCENT
 OLIVER SCOTT
(2 rows)

postgres=# 
```

Figure 9: Name ends with t

9. List all the names in reverse

```
select reverse(name) as reverse_name from acc_details;
```

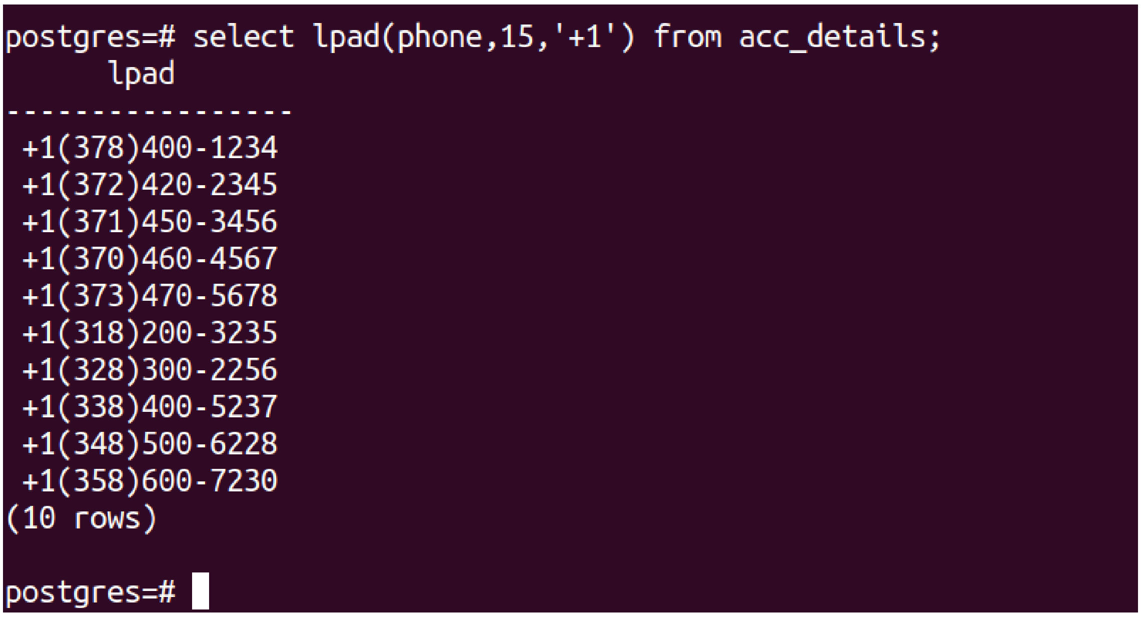


```
postgres=# select reverse(name) as REVERSE_name from acc_details;
   reverse_name
-------------------
 SMADA EKIM
 EGROEG ANAID
 HTEBAZILE ZAID
 EGROEG YERFFOEJ
 NYLTIAK REFINNEJ
 TNECNIV NYLTIAK
 DLEIFTTOG MAHARBA
 SMAILLIW YCATS
 EGROEG ENIREHTAC
 TTOCS REVILO
(10 rows)

postgres=# 
```

Figure 10: Reverse the name

10. Display all the phone numbers including US Country code (+1). For eg: (378)400-1234should be displayed as +1(378)400-1234. Use LPAD function

```
select lpad(phone,15,'+1') from acc_details;
```

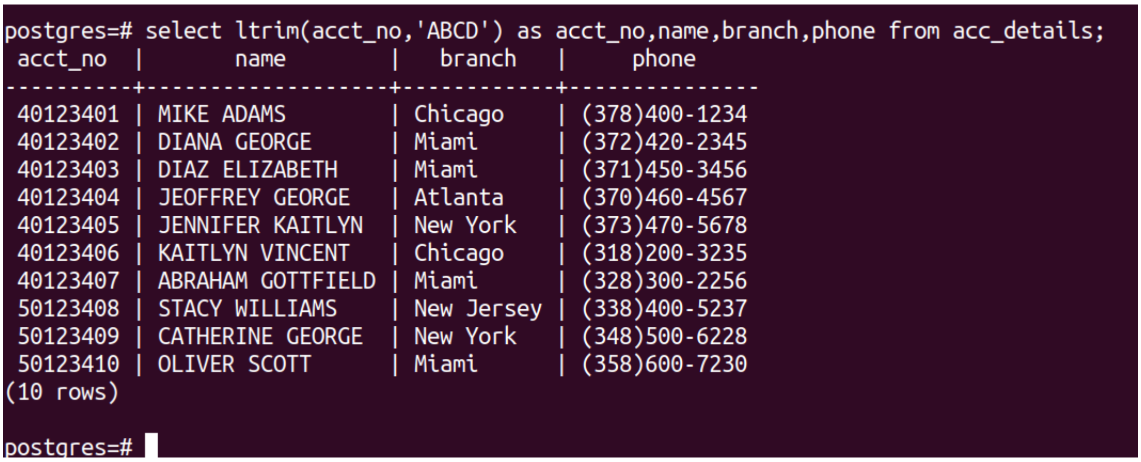


```
postgres=# select lpad(phone,15,'+1') from acc_details;
      lpad
-----------------
 +1(378)400-1234
 +1(372)420-2345
 +1(371)450-3456
 +1(370)460-4567
 +1(373)470-5678
 +1(318)200-3235
 +1(328)300-2256
 +1(338)400-5237
 +1(348)500-6228
 +1(358)600-7230
(10 rows)

postgres=# 
```

Figure 11: Phone number including +1

11. Display all the account numbers. The starting alphabet associated with the Account_No should be removed. Use LTRIM function.

```
select ltrim(acct_no,’ABCD’) as acct_no,name,branch,phone from acc_details;
```

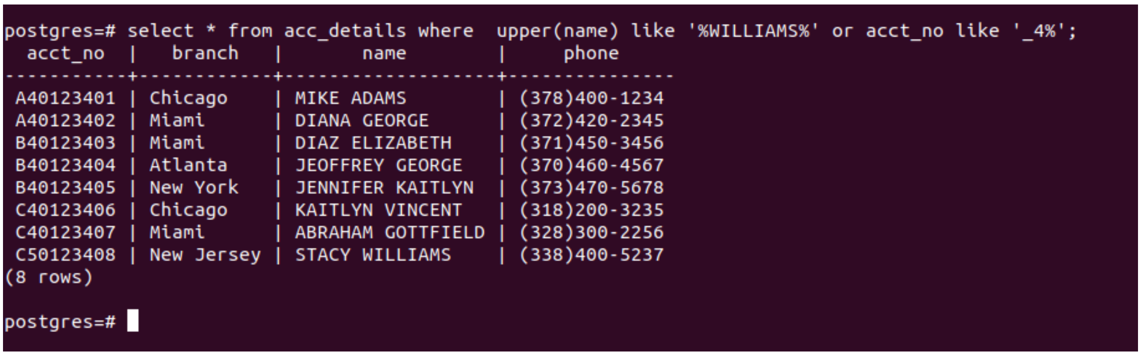


```
postgres=# select ltrim(acct_no,'ABCD') as acct_no,name,branch,phone from acc_details;
 acct_no  |       name        |   branch   |     phone
----------+-------------------+------------+---------------
 40123401 | MIKE ADAMS        | Chicago    | (378)400-1234
 40123402 | DIANA GEORGE      | Miami      | (372)420-2345
 40123403 | DIAZ ELIZABETH    | Miami      | (371)450-3456
 40123404 | JEOFFREY GEORGE   | Atlanta    | (370)460-4567
 40123405 | JENNIFER KAITLYN  | New York   | (373)470-5678
 40123406 | KAITLYN VINCENT   | Chicago    | (318)200-3235
 40123407 | ABRAHAM GOTTFIELD | Miami      | (328)300-2256
 50123408 | STACY WILLIAMS    | New Jersey | (338)400-5237
 50123409 | CATHERINE GEORGE  | New York   | (348)500-6228
 50123410 | OLIVER SCOTT      | Miami      | (358)600-7230
(10 rows)

postgres=#
```

Figure 12: Removing first alphabets from acc no

12. Display the details of all people whose account number starts in ‘4’ and name contains the sub string ‘Williams’.

```
select * from acc_details where upper(name) like ’%WILLIAMS%’ or acct_no like ’_4%’;
```



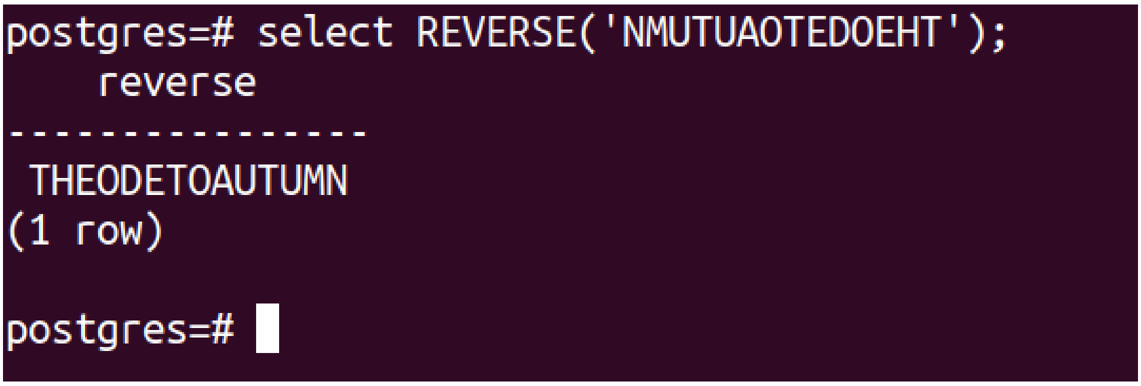
```
postgres=# select * from acc_details where  upper(name) like '%WILLIAMS%' or acct_no like '_4%';
  acct_no  |   branch   |       name        |     phone
-----------+------------+-------------------+---------------
 A40123401 | Chicago    | MIKE ADAMS        | (378)400-1234
 A40123402 | Miami      | DIANA GEORGE      | (372)420-2345
 B40123403 | Miami      | DIAZ ELIZABETH    | (371)450-3456
 B40123404 | Atlanta    | JEOFFREY GEORGE   | (370)460-4567
 B40123405 | New York   | JENNIFER KAITLYN  | (373)470-5678
 C40123406 | Chicago    | KAITLYN VINCENT   | (318)200-3235
 C40123407 | Miami      | ABRAHAM GOTTFIELD | (328)300-2256
 C50123408 | New Jersey | STACY WILLIAMS    | (338)400-5237
(8 rows)

postgres=#
```

Figure 13: name contain williams or acc no starts in 4

B. Use the system table DUAL for the following questions:

1. Find the reverse of the string ‘ nmutuAotedOehT’.

```
SELECT REVERSE('NMUTUAOTEDOEHT');
```

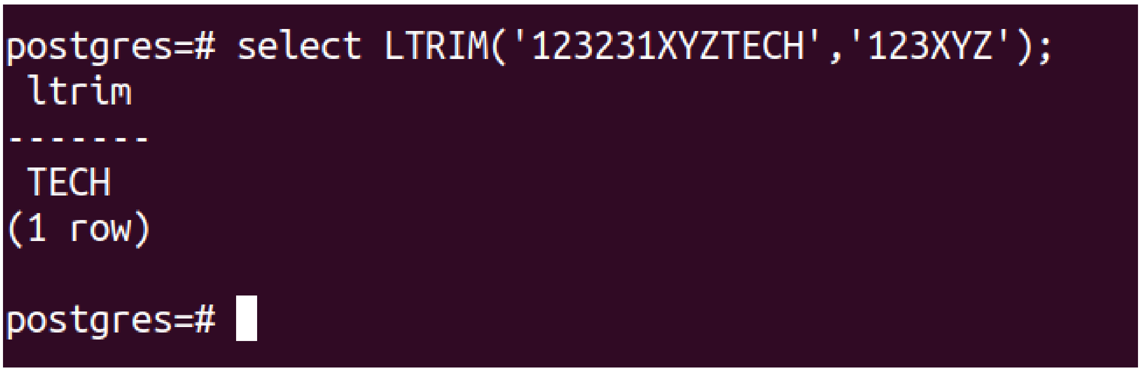


```
postgres=# select REVERSE('NMUTUAOTEDOEHT');
    reverse
----------------
 THEODETOAUTUMN
(1 row)

postgres=#
```

Figure 14: Function Reverse

2. Use LTRIM function on ‘123231xyzTech’ so as to obtain the output ‘Tech’

```
SELECT LTRIM('123231XYZTECH','123XYZ');
```

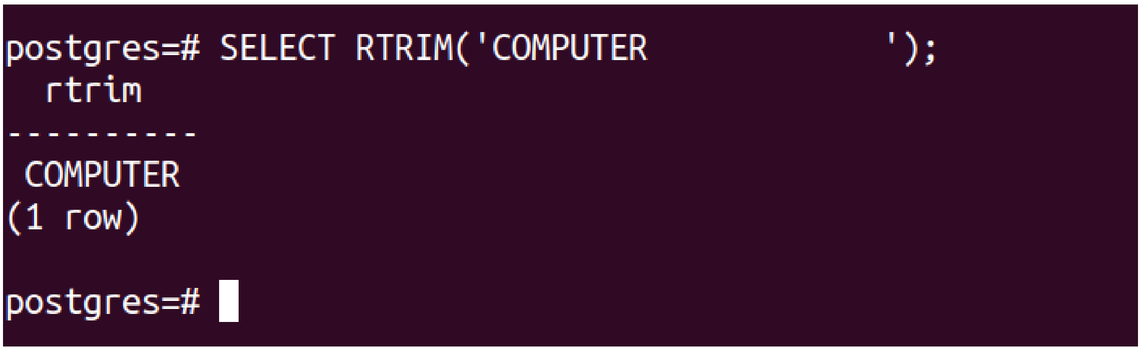


```
postgres=# select LTRIM('123231XYZTECH','123XYZ');
 ltrim
-------
 TECH
(1 row)

postgres=#
```

Figure 15: Function ltrim

.3. Use RTRIM function on ‘Computer ‘ to remove the trailing spaces.

```
SELECT RTRIM('COMPUTER');
```

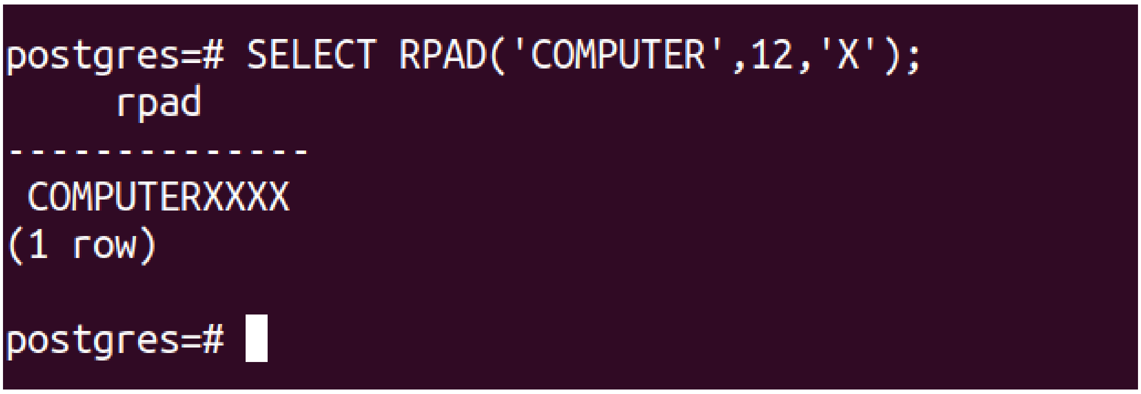


```
postgres=# SELECT RTRIM('COMPUTER             ');
  rtrim
----------
 COMPUTER
(1 row)

postgres=#
```

Figure 16: Function rtrim

4. Perform RPAD on ‘computer’ to obtain the output as ‘computerXXXX’

```
SELECT RPAD('COMPUTER',12,'X');
```

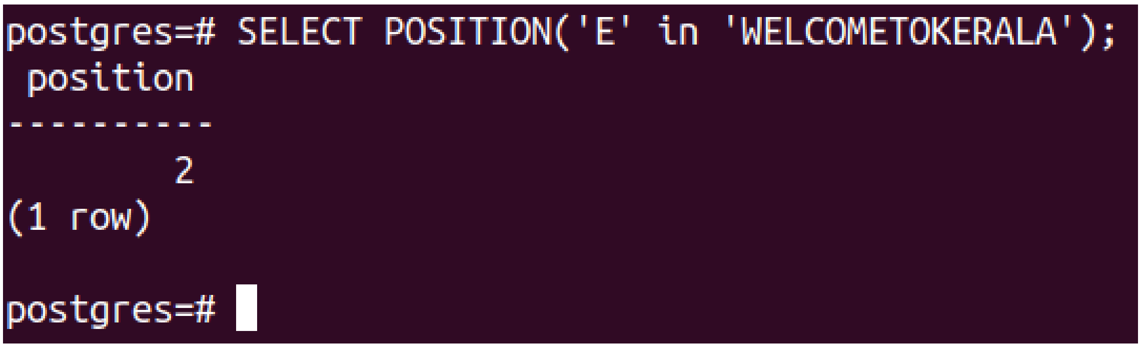


```
postgres=# SELECT RPAD('COMPUTER',12,'X');
     rpad
--------------
 COMPUTERXXXX
(1 row)

postgres=# 
```

Figure 17: Function rpad

5. Use POSITION function to find the first occurrence of ‘e’ in the string ‘Welcome to Kerala’.

```
SELECT POSITION('E' in 'WELCOMETOKERALA');
```

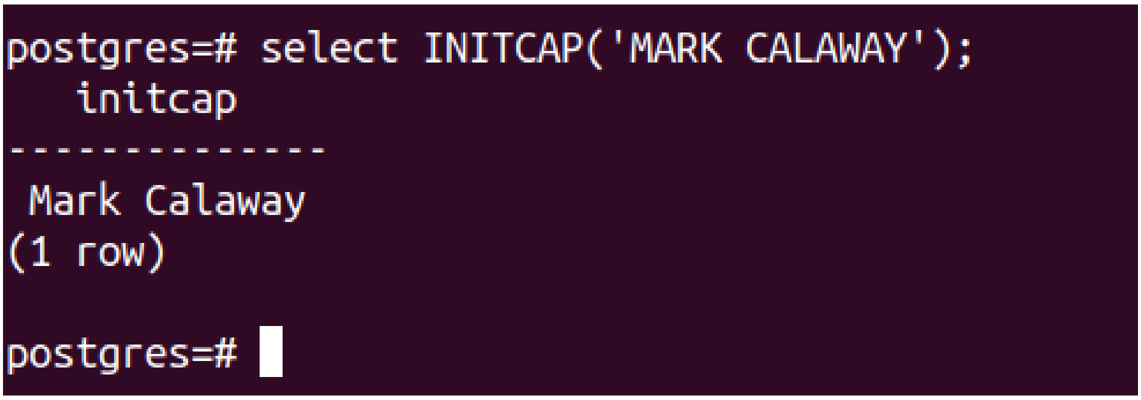


```
postgres=# SELECT POSITION('E' in 'WELCOMETOKERALA');
 position
----------
        2
(1 row)

postgres=# 
```

Figure 18: Function position

6. Perform INITCAP function on ‘mARKcALAwaY’.

```
SELECT INITCAP('MARK CALAWAY');
```

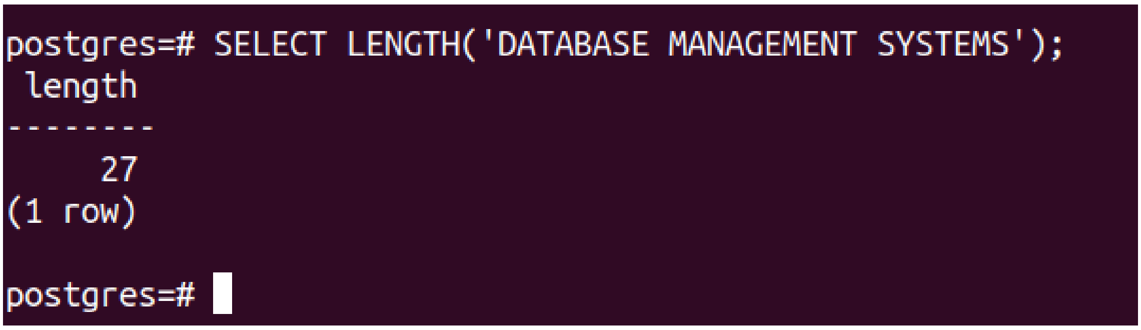


```
postgres=# select INITCAP('MARK CALAWAY');
   initcap
--------------
 Mark Calaway
(1 row)

postgres=# 
```

Figure 19: Function initcap

7. Find the length of the string ‘Database Management Systems’..

```
SELECT LENGTH(’DATABASE MANAGEMENT SYSTEMS’);
```

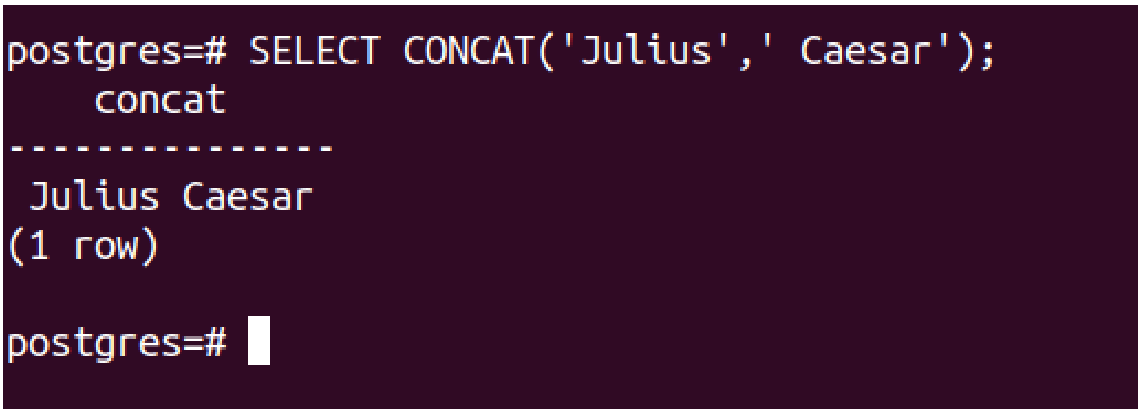


```
postgres=# SELECT LENGTH('DATABASE MANAGEMENT SYSTEMS');
 length
--------
     27
(1 row)

postgres=#
```

Figure 20: Function Length

8. Concatenate the strings ‘Julius’ and ‘Caesar’.

```
SELECT CONCAT(’JULIUS’,’CAESAR’);
```

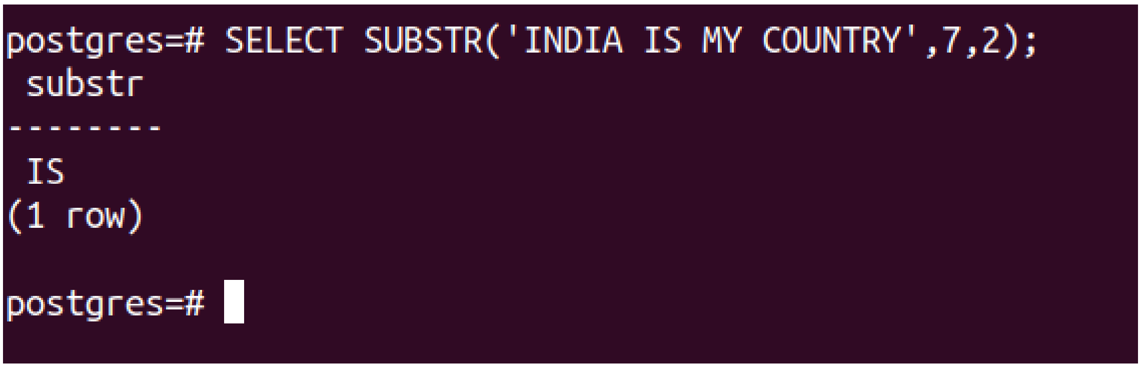


```
postgres=# SELECT CONCAT('Julius',' Caesar');
    concat
---------------
 Julius Caesar
(1 row)

postgres=#
```

Figure 21: Function concat

9. Use SUBSTR function to retrieve the substring ‘is’ from the string ‘India is my country’.

```
SELECT SUBSTR(’INDIA IS MY COUNTRY’,7,2);
```



```
postgres=# SELECT SUBSTR('INDIA IS MY COUNTRY',7,2);
 substr
--------
 IS
(1 row)

postgres=#
```

Figure 22: Function substr

10. Use INSTR function to find the second occurrence of ‘k’ from the last. The string is ‘Making of a King’.

```
SELECT INSTR('MAKING OF A KING','K', -1,2);
```

3 Result

The query was executed and the output was obtained.