

## STRING FUNCTIONS AND PATTERN MATCHING

### AIM:

String Functions & Pattern Matching

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

### QUESTION:

Create a table named acct\_details and populate the table as shown below.

```
hishamalip@savage: ~
asdlab=# CREATE TABLE acct_details(Acct_No TEXT, Branch TEXT, Name TEXT, Phone BIGINT);
CREATE TABLE
asdlab=# INSERT INTO acct_details VALUES('A40123401', 'Chicago', 'Mike Adams', 3784001234),
                                           ('A40123402', 'Miami', 'Diana George', 3724202345),
                                           ('B40123403', 'Miami', 'Diaz Elizabeth', 3714503456),
                                           ('B40123404', 'Atlanta', 'Jeoffrey George', 3704604567),
                                           ('B40123405', 'New York', 'Jennifer Kaitlyn', 3734705678);
INSERT 0 5
asdlab=# INSERT INTO acct_details VALUES('C40123406', 'Chicago', 'Kaitlyn Vincent', 3182003235),
                                           ('C40123407', 'Miami', 'Abraham Gottfield', 3283002256),
                                           ('C50123408', 'New Jersey', 'Stacy Williams', 3384005237),
                                           ('D50123409', 'New York', 'Catherine George', 3485006228),
                                           ('D50123410', 'Miami', 'Oliver Scott', 3586007230);
asdlab=#
asdlab=#
asdlab=#
asdlab=#
asdlab=#
asdlab=#
asdlab=#
asdlab=#
asdlab=#
asdlab=#
```

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

```
hishamalip@savage: ~
asdlab=# SELECT name FROM acct_details WHERE Name LIKE 'D%';
      name
-----
Diana George
Diaz Elizabeth
(2 rows)

asdlab=#
```

2. List the names of all branches containing the substring 'New'

```
hishamalip@savage: ~
asdlab=# SELECT branch FROM acct_details WHERE branch LIKE '%New%';
      branch
-----
New York
New Jersey
New York
(3 rows)

asdlab=#
```

3. List all the names in Upper Case Format

```
hishamalip@savage: ~
asdlab=# SELECT UPPER(name) FROM acct_details;
      upper
-----
MIKE ADAMS
DIANA GEORGE
DIAZ ELIZABETH
JEFFREY GEORGE
JENNIFER KAITLYN
KAITLYN VINCENT
ABRAHAM GOTTFIELD
STACY WILLIAMS
CATHERINE GEORGE
OLIVER SCOTT
(10 rows)

asdlab=#
```

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

```
hishamalip@savage: ~
asdlab=# SELECT name FROM acct_details WHERE Name LIKE '___n%n';
      name
-----
Jennifer Kaitlyn
(1 row)

asdlab=#
```

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

```
hishamalip@savage: ~
asdlab=# SELECT name FROM acct_details WHERE name LIKE 'D_a%' AND name LIKE '%Eli%';
          name
-----
Diaz Elizabeth
(1 row)

asdlab=#
```

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

```
hishamalip@savage: ~
asdlab=# SELECT name FROM acct_details WHERE acct_no LIKE '%6';
          name
-----
Kaitlyn Vincent
(1 row)

asdlab=#
```

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

```
hishamalip@savage: ~
asdlab=# UPDATE acct_details SET name = UPPER(name);
UPDATE 10
asdlab=# SELECT * FROM acct_details ;
  acct_no | branch |      name      | phone
-----+-----+-----+-----
A40123401 | Chicago | MIKE ADAMS     | 3784001234
A40123402 | Miami  | DIANA GEORGE   | 3724202345
B40123403 | Miami  | DIAZ ELIZABETH | 3714503456
B40123404 | Atlanta | JEOFFREY GEORGE | 3704604567
B40123405 | New York | JENNIFER KAITLYN | 3734705678
C40123406 | Chicago | KAITLYN VINCENT | 3182003235
C40123407 | Miami  | ABRAHAM GOTTFIELD | 3283002256
C50123408 | New Jersey | STACY WILLIAMS | 3384005237
D50123409 | New York | CATHERINE GEORGE | 3485006228
D50123410 | Miami  | OLIVER SCOTT    | 3586007230
(10 rows)

asdlab=#
```

8. List the names of all people ending on the alphabet 't'

```
hishamalip@savage: ~  
asdlab=# SELECT name FROM acct_details WHERE LOWER(name) LIKE '%t';  
          name  
-----  
    KAITLYN VINCENT  
    OLIVER SCOTT  
(2 rows)  
asdlab=#
```

9. List all the names in reverse

```
hishamalip@savage: ~  
asdlab=# SELECT REVERSE(name) FROM acct_details ;  
          reverse  
-----  
    SMADA EKIM  
    EGROEG ANAID  
    HTEBAZILE ZAID  
    EGROEG YERFF0EJ  
    NYLTIK REFINNEJ  
    TNECNIV NYLTIK  
    DLEIFTTOG MAHARBA  
    SMAILLIW YCATS  
    EGROEG ENIREHTAC  
    TTOCS REVILO  
(10 rows)  
asdlab=#
```

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

```
hishamalip@savage: ~  
asdlab=# SELECT LPAD(phone, 12, '+1') FROM acct_details;  
lpad  
-----  
+13784001234  
+13724202345  
+13714503456  
+13704604567  
+13734705678  
+13182003235  
+13283002256  
+13384005237  
+13485006228  
+13586007230  
(10 rows)  
asdlab=#
```

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

```
hishamalip@savage: ~  
asdlab=# SELECT LTRIM(acct_no, 'ABCD') AS Acct_No FROM acct_details ;  
acct_no  
-----  
40123401  
40123402  
40123403  
40123404  
40123405  
40123406  
40123407  
50123408  
50123409  
50123410  
(10 rows)  
asdlab=#
```

12. Display the details of all people whose account number starts in '5' and name contains the substring 'Williams'.

```
hishamalip@savage: ~  
asdlab=#  
asdlab=# SELECT * FROM acct_details WHERE LTRIM(acct_no, 'ABCD') LIKE '5%'  
asdlab=# AND name LIKE '%WILLIAMS%';  
  acct_no | branch | name | phone  
-----+-----+-----+-----  
C50123408 | New Jersey | STACY WILLIAMS | 3384005237  
(1 row)  
asdlab=#
```

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

1. Find the reverse of the string 'nmutuAotedOehT'

```
hishamalip@savage: ~  
asdlab=# SELECT REVERSE('nmutuAotedOehT');  
      reverse  
-----  
The0detoAutumn  
(1 row)  
asdlab=#
```

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

```
hishamalip@savage: ~  
asdlab=# SELECT LTRIM('123231xyzTech', '123xyz');  
      ltrim  
-----  
Tech  
(1 row)  
asdlab=#
```

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

```
hishamalip@savage: ~  
postgres=# SELECT RTRIM('Computer ');  
rtrim  
-----  
Computer  
(1 row)  
postgres=#
```

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

```
hishamalip@savage: ~  
postgres=# SELECT RPAD('Computer', 12, 'x');  
rpad  
-----  
Computerxxxx  
(1 row)  
postgres=#
```

5. Use INSTR function to find the first occurrence of 'e' in the string 'Welcome to Kerala'

```
hishamalip@savage: ~  
postgres=# SELECT POSITION('e' in 'Welcome to Kerala');  
position  
-----  
2  
(1 row)  
postgres=#
```

6. Perform INITCAP function on 'mARKcALAWaY'

```
hishamalip@savage: ~  
postgres=# SELECT INITCAP('mARK cALAWaY');  
initcap  
-----  
Mark Calaway  
(1 row)  
postgres=#
```



7. Find the length of the string 'Database Management Systems'.

```
hishamalip@savage: ~  
postgres=# SELECT LENGTH('Database Management Systems');  
length  
-----  
27  
(1 row)  
  
postgres=#
```

8. Concatenate the strings 'Julius' and 'Caesar'

```
hishamalip@savage: ~  
postgres=# SELECT CONCAT('Julius','Caesar');  
concat  
-----  
JuliusCaesar  
(1 row)  
  
postgres=#
```

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

```
hishamalip@savage: ~  
postgres=# SELECT SUBSTR('India is my country', 7, 2);  
substr  
-----  
is  
(1 row)  
  
postgres=#
```

## RESULT:

The query was executed and the output was obtained.