

College of Engineering, Trivandrum

Department of Computer Science and Engineering



CS333 APPLICATION SOFTWARE DEVELOPMENT LAB

LABORATORY REPORT 7

String and pattern matching

Student Name

1. Justine Biju(S5)

Student ID

170445(Roll No:37)

Submission Date : 09/08/2019

1 Introduction

There are many SQL queries that help support string and pattern matching. Few of what we are going to discuss are listed below:

1. SUBSTR

- (a) The SUBSTR() function extracts a substring from a string (starting at any position).
- (b) Syntax: SUBSTR(string, start, length)

2. LENGTH

- (a) The LENGTH() function returns the length of a string (in bytes).
- (b) Syntax: LENGTH(string)

3. LOWER

- (a) The LOWER() function converts a string to lower-case.
- (b) Syntax: LOWER(text)

4. UPPER

- (a) The UPPER() function converts a string to upper-case.
- (b) Syntax: UPPER(text)

5. CONCAT

- (a) The CONCAT() function adds two or more expressions together.
- (b) Syntax: CONCAT(expression1, expression2, expression3,...)

6. LPAD

- (a) The LPAD() function left-pads a string with another string, to a certain length.
- (b) Syntax: LPAD(string, length, lpad_string)

7. RTRIM

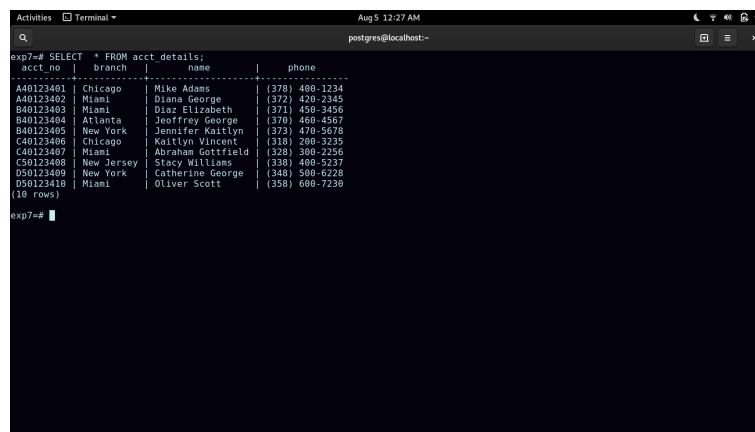
- (a) The RTRIM() function removes trailing spaces from a string.
- (b) Syntax: RTRIM(string)

8. INSTR

- (a) The INSTR() function returns the position of the first occurrence of a string in another string.
- (b) Syntax: INSTR(string1, string2)

All these SQL queries are supported by PostgreSQL and can be implemented in a similar fashion.

2 Questions



A terminal window titled 'Terminal' with a search bar and window controls. The prompt is 'postgres@localhost:~'. The command entered is 'exp7=# SELECT * FROM acct_details;'. The output is a table with 10 rows and 4 columns: acct_no, branch, name, and phone. The data is as follows:

acct_no	branch	name	phone
A40123401	Chicago	Mike Adams	(378) 480-1234
A40123402	Miami	Diana George	(372) 420-2345
B40123403	Miami	Diaz Elizabeth	(371) 450-3456
B40123404	Atlanta	Jeffrey George	(370) 460-4567
B40123405	New York	Jennifer Kaitlyn	(373) 470-5678
C40123406	Chicago	Kaitlyn Vincent	(318) 200-3235
C40123407	Miami	Abraham Gottfield	(320) 300-2256
C50123408	New Jersey	Stacy Williams	(330) 400-5237
D50123409	New York	Catherine George	(348) 500-6228
D50123410	Miami	Oliver Scott	(358) 600-7230

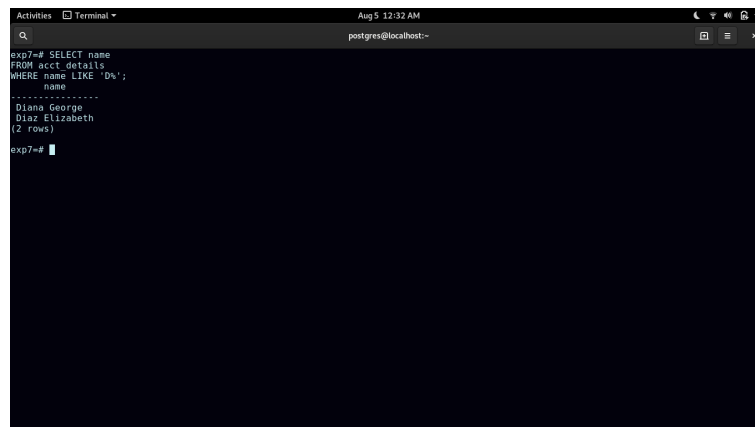
(10 rows)

exp7=#

Figure 1: Entries in 'acct_details'

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

```
SELECT name
FROM acct_details
WHERE name LIKE 'D%';
```



A terminal window titled 'Terminal' with a search bar and window controls. The prompt is 'postgres@localhost:~'. The command entered is 'exp7=# SELECT name FROM acct_details WHERE name LIKE 'D%';'. The output is a table with 2 rows and 1 column: name. The data is as follows:

name
Diana George
Diaz Elizabeth

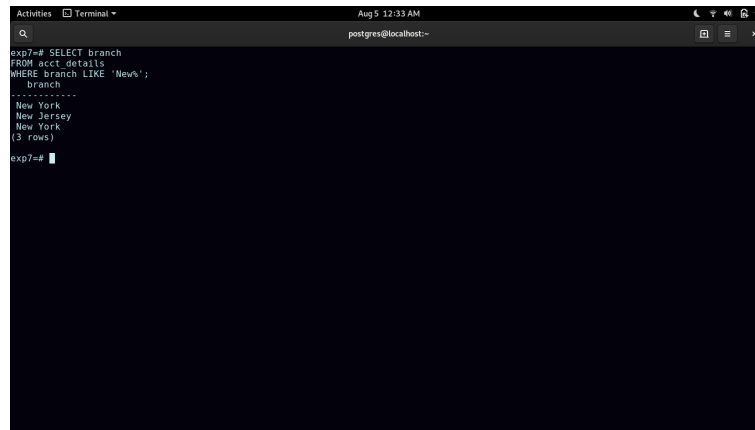
(2 rows)

exp7=#

Figure 2: Question 1

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

```
SELECT branch
FROM acct_details
WHERE branch LIKE 'New%';
```

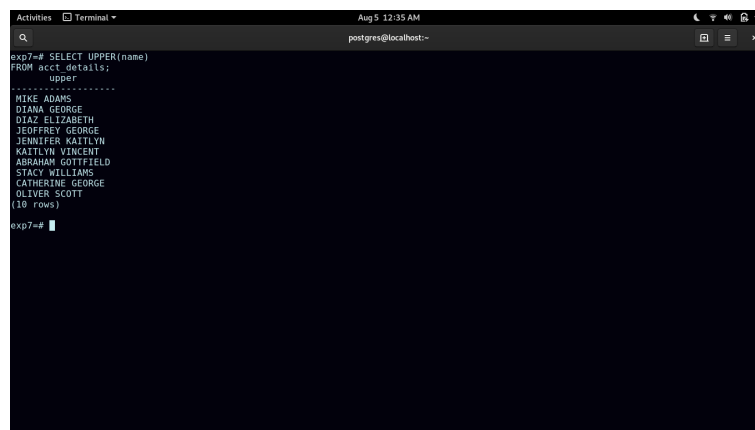


```
exp7=# SELECT branch
FROM acct_details
WHERE branch LIKE 'New%';
 branch
-----
New York
New Jersey
New York
(3 rows)
exp7=#
```

Figure 3: Question 2

3. List all the names in Upper Case Format

```
SELECT UPPER(name)
FROM acct_details;
```

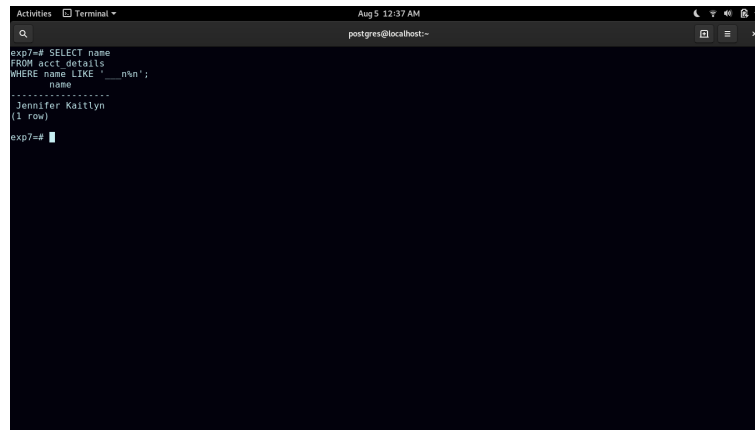


```
exp7=# SELECT UPPER(name)
FROM acct_details;
      upper
-----
MIKE ADAMS
DIANA GEORGE
DIAZ ELIZABETH
JEFFREY GEORGE
JENNIFER KAITLYN
KAITLYN VINCENT
ABRAHAM GOTTFIELD
STACY WILLIAMS
CATHERINE GEORGE
OLIVER SCOTT
(10 rows)
exp7=#
```

Figure 4: Question 3

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

```
SELECT name
FROM acct_details
WHERE name LIKE '___n%n';
```

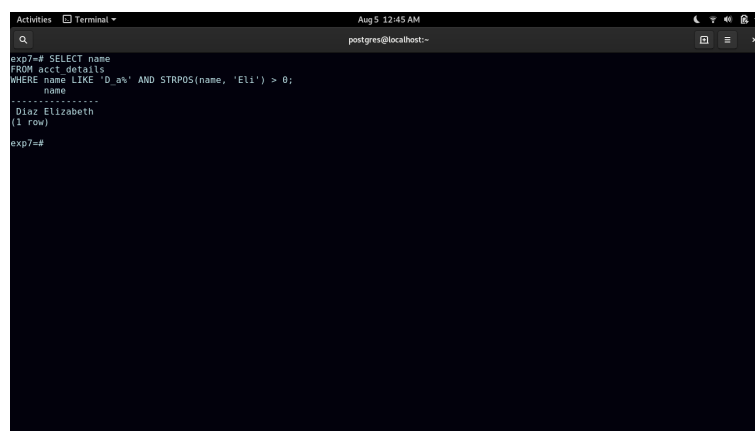


```
exp7=# SELECT name
FROM acct_details
WHERE name LIKE '____n%';
 name
-----
Jennifer Kaitlyn
(1 row)
exp7=#
```

Figure 5: Question 4

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

```
SELECT name
FROM acct_details
WHERE name LIKE 'D_a%' AND
STRPOS(name, 'Eli') > 0;
```



```
exp7=# SELECT name
FROM acct_details
WHERE name LIKE 'D_a%' AND STRPOS(name, 'Eli') > 0;
 name
-----
Diaz Elizabeth
(1 row)
exp7=#
```

Figure 6: Question 5

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

```
SELECT name
FROM acct_details
WHERE acct_no LIKE '%6';
```

```
Activities Terminal Aug 8 12:46 AM postgres@localhost:~
exp7=# SELECT name
FROM acct_details
WHERE acct_no LIKE '%6';
-----
Kaitlyn Vincent
(1 row)
exp7=#
```

Figure 7: Question 6

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

```
UPDATE acct_details
SET name = UPPER(name);

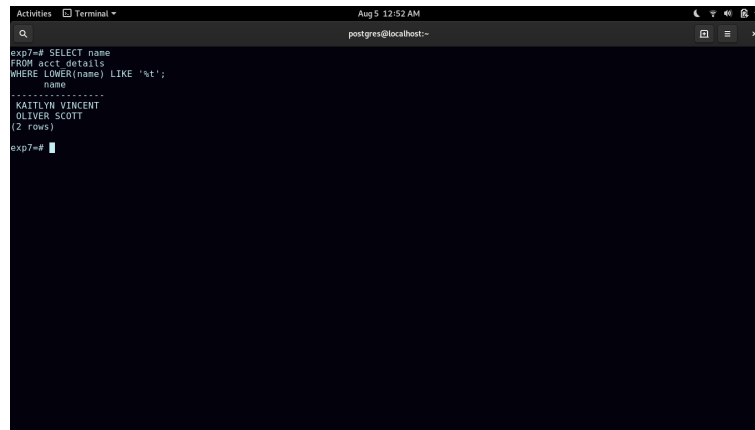
SELECT * FROM acct_details;
```

```
Activities Terminal Aug 8 12:49 AM postgres@localhost:~
exp7=# UPDATE acct_details
exp7=# SET name = UPPER(name);
UPDATE 10
exp7=# SELECT * FROM acct_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 | JOFFREY 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)
exp7=#
```

Figure 8: Question 7

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

```
SELECT name
FROM acct_details
WHERE LOWER(name) LIKE '%t';
```

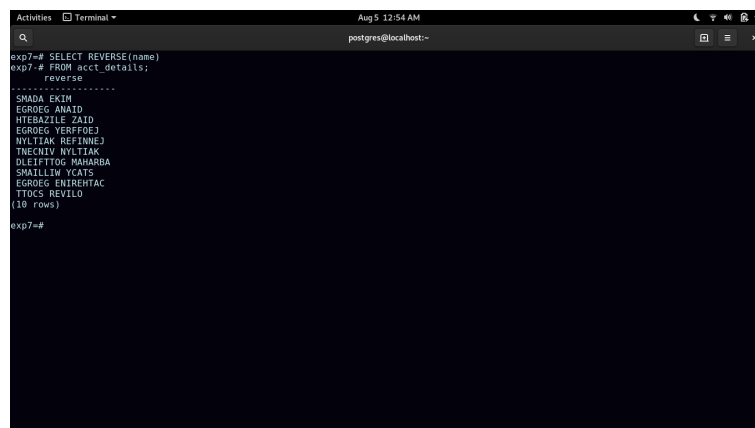


```
exp7=# SELECT name
FROM acct_details
WHERE LOWER(name) LIKE 't';
-----
KAITLYN VINCENT
OLIVER SCOTT
(2 rows)
exp7=#
```

Figure 9: Question 8

9. List all the names in reverse

```
SELECT REVERSE(name)
FROM acct_details;
```



```
exp7=# SELECT REVERSE(name)
exp7=# FROM acct_details;
-----
SMADA EKIM
EGROEG ANAID
HTEBAZLE ZAID
EGROEG YENFFREJ
NYLTIAK REFINNEJ
TNECNIV NYLTIAK
OLEFFITOG MAHARBA
SMAILLIW YCATS
EGROEG ENIREHTAC
TTOKS REVILLO
(10 rows)
exp7=#
```

Figure 10: Question 9

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

```
SELECT LPAD(phone, 16, '+1')
FROM acct_details;
```

```

exp7=# SELECT LPAD(phone, 16, '1')
FROM acct_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) 600-7239
(10 rows)
exp7=#

```

Figure 11: Question 10

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]') ,
       branch , name , phone
FROM acct_details;

```

```

exp7=# SELECT LTRIM(acct_no, '[ABCD]')
FROM acct_details;
ltrim
-----
40123401
40123402
40123403
40123404
40123405
40123406
40123407
50123408
50123409
50123410
(10 rows)
exp7=#

```

Figure 12: Question 11

12. Display the details of all people whose account number starts in '4' and name contains the substring 'Williams'.

```

ELECT *
FROM acct_details
WHERE TRIM(acct_no , '[ABCD]')
LIKE '5%'
AND STRPOS(name, 'WILLIAMS') > 0;

```



```
Activities Terminal Aug 5 1:17 AM postgres@localhost:~
exp7=# SELECT *
FROM acct_details
WHERE TRIM(acct_no, '[ABCD]') LIKE '5%' AND STRPOS(name, 'WILLIAMS') > 0;
acct_no | branch | name | phone
-----
C50123408 | New Jersey | STACY WILLIAMS | (338) 400-5237
(1 row)
exp7=#
```

Figure 13: Question 12

Following questions done on system table DUAL:

13. Find the reverse of the string ' nmutuAotedOehT'

```
SELECT
REVERSE( 'NMUTUAOTEDOEH' );
```

```
Activities Terminal Aug 5 1:22 AM postgres@localhost:~
exp7=# SELECT REVERSE('NMUTUAOTEDOEH');
reverse
-----
THEOOTOAOTURN
(1 row)
exp7=#
```

Figure 14: Question 13

14. Find the reverse of the string ' nmutuAotedOehT'

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

```
Activities Terminal Aug 5 1:24 AM postgres@localhost:~
exp7=# SELECT LTRIM('123231XYZTECH','123XYZ');
-----
TECH
(1 row)
exp7=#
```

Figure 15: Question 14

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

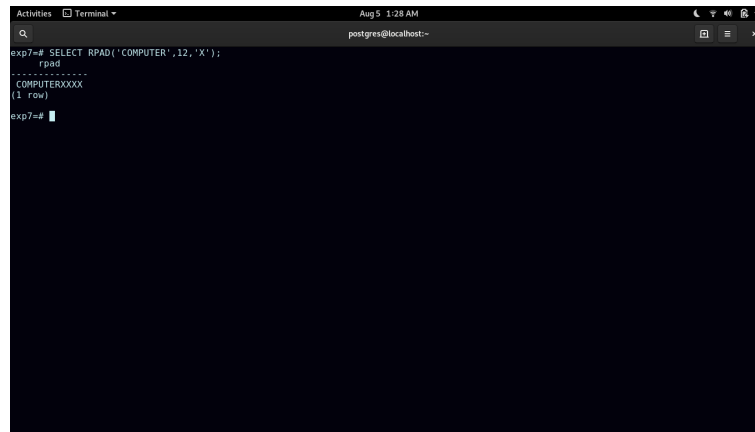
```
SELECT
RTRIM('COMPUTER ');
```

```
Activities Terminal Aug 5 1:26 AM postgres@localhost:~
exp7=# SELECT RTRIM('COMPUTER ');
-----
COMPUTER
(1 row)
exp7=#
```

Figure 16: Question 15

16. Perform RPAD on 'computer' to obtain the output as 'computerXXXXX'

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

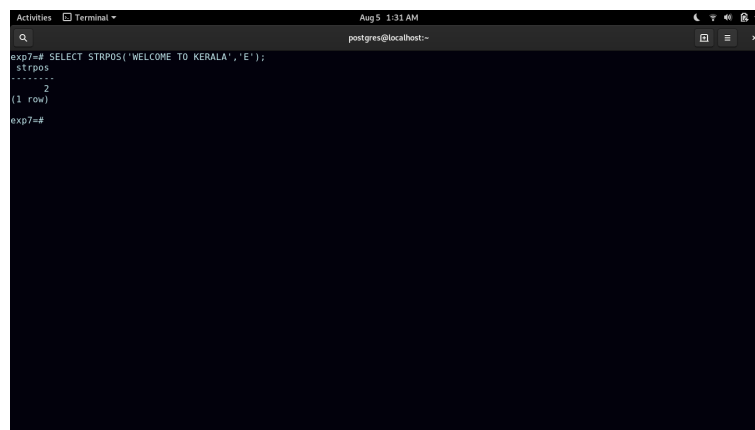
A terminal window titled 'Terminal' with a search bar and window controls. The prompt is 'postgres@localhost:~'. The user enters 'exp7=# SELECT RPAD('COMPUTER',12,'X');'. The output shows 'rpad' followed by a table with one row containing 'COMPUTERXXXXX'.

```
exp7=# SELECT RPAD('COMPUTER',12,'X');
rpad
-----
COMPUTERXXXXX
(1 row)
exp7=#
```

Figure 17: Question 16

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

```
SELECT
STRPOS( 'WELCOME TO KERALA' , 'E' );
```

A terminal window titled 'Terminal' with a search bar and window controls. The prompt is 'postgres@localhost:~'. The user enters 'exp7=# SELECT STRPOS('WELCOME TO KERALA','E');'. The output shows 'strpos' followed by a table with one row containing the value '2'.

```
exp7=# SELECT STRPOS('WELCOME TO KERALA','E');
strpos
-----
2
(1 row)
exp7=#
```

Figure 18: Question 17

18. Perform INITCAP function on 'mARKcALAwAY'

```
SELECT
INITCAP( 'MARKCALAWAY' );
```

```
Activities Terminal Aug 5 1:34 AM postgres@localhost:~
exp7=# SELECT INITCAP('MARK CALAWAY');
initcap
-----
Mark Calaway
(1 row)
exp7=#
```

Figure 19: Question 18

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

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

```
Activities Terminal Aug 5 1:36 AM postgres@localhost:~
exp7=# SELECT LENGTH('DATABASE MANAGEMENT SYSTEMS');
length
-----
27
(1 row)
exp7=#
```

Figure 20: Question 19

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

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

```
Activities Terminal Aug 5 1:37 AM postgres@localhost:~
exp7=# SELECT CONCAT('JULIUS', 'CAESAR');
concat
-----
JULIUSCAESAR
(1 row)
exp7=#
```

Figure 21: Question 20

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

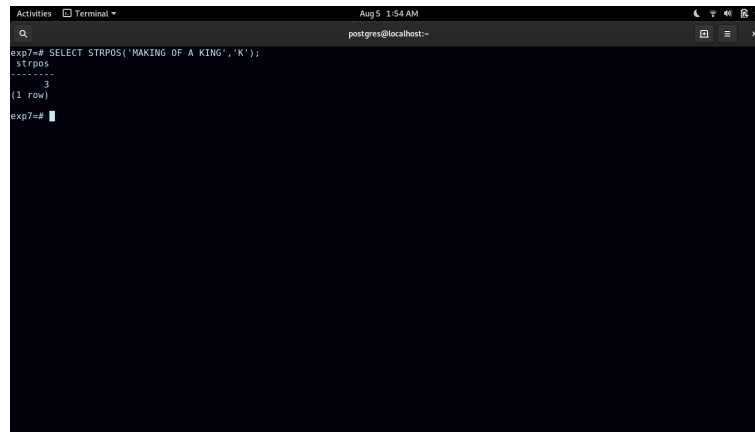
```
SELECT
SUBSTR('INDIA IS MY COUNTRY', 7, 2);
```

```
Activities Terminal Aug 5 1:39 AM postgres@localhost:~
exp7=# SELECT SUBSTR('INDIA IS MY COUNTRY', 7, 2);
substr
-----
IS
(1 row)
exp7=#
```

Figure 22: Question 21

22. 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);
```

A terminal window titled 'Terminal' with a search bar and window controls. The prompt is 'postgres@localhost:~'. The user enters the SQL query 'exp7=# SELECT STRPOS('MAKING OF A KING','K');'. The output shows 'strpos' as the column name, followed by a single row with the value '3'. The prompt returns to 'exp7=#'.

```
exp7=# SELECT STRPOS('MAKING OF A KING','K');
strpos
-----
      3
(1 row)
exp7=#
```

Figure 23: Question 22

3 Result

- Successfully implemented pattern matching and string operation queries on PostgreSQL.