

Reformatting string and character data

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

SQL

Brian Piccolo

Sr. Director, Digital Strategy

Topics

- Reformatting string and character data.
- Parsing string and character data.
- Determine string length and character position.
- Truncating and padding string data.

The string concatenation operator

```
SELECT
  first_name,
  last_name,
  first_name || ' ' || last_name AS full_name
FROM customer
```

```
+-----+-----+-----+
| first_name | last_name | full_name      |
|-----|-----|-----|
| MARY       | SMITH     | MARY SMITH     |
| LINDA      | WILLIAMS  | LINDA WILLIAMS |
+-----+-----+-----+
```

String concatenation with functions

```
SELECT
  CONCAT(first_name, ' ', last_name) AS full_name
FROM customer;
```

```
+-----+
| first_name | last_name | full_name      |
+-----+
| MARY       | SMITH     | MARY SMITH     |
| LINDA      | WILLIAMS  | LINDA WILLIAMS |
+-----+
```

String concatenation with a non-string input

```
SELECT
  customer_id || ': '
  || first_name || ' '
  || last_name AS full_name
FROM customer;
```

```
+-----+
| full_name          |
+-----+
| 1: MARY SMITH      |
| 2: LINDA WILLIAMS  |
+-----+
```

Changing the case of string

```
SELECT
  UPPER(email)
FROM customer;
```

```
+-----+
| UPPER(email) |
+-----+
| MARY.SMITH@SAKILACUSTOMER.ORG |
| PATRICIA.JOHNSON@SAKILACUSTOMER.ORG |
| LINDA.WILLIAMS@SAKILACUSTOMER.ORG |
+-----+
```

Changing the case of string

```
SELECT
  LOWER(title)
FROM film;
```

```
+-----+
| LOWER(title) |
+-----+
| academy dinosaur |
| ace goldfinger |
| adaptation holes |
+-----+
```

Changing the case of string

```
SELECT  
  INITCAP(title)  
FROM film;
```

```
+-----+  
| INITCAP(title) |  
+-----+  
| Academy Dinosaur |  
| Ace Goldfinger   |  
| Adaptation Holes |  
+-----+
```


Replacing characters in a string

```
SELECT description FROM film;
```

```
+-----+
| description |
+-----+
| A Epic Drama of a Feminist And a Mad Scientist... |
| A Astounding Epistle of a Database Administrator... |
| A Astounding Reflection of a Lumberjack And a Car... |
| A Fanciful Documentary of a Frisbee And a Lumberjack... |
| A Fast-Paced Documentary of a Pastry Chef And a... |
+-----+
```

Replacing characters in a string

```
SELECT
    REPLACE(description, 'A Astounding',
              'An Astounding') as description
FROM film;
```

```
+-----+
| description                                     |
|-----|
| A Epic Drama of a Feminist And a Mad Scientist... |
| An Astounding Epistle of a Database Administrator... |
| An Astounding Reflection of a Lumberjack And a Car... |
+-----+
```

Manipulating string data with REVERSE

```
SELECT
  title,
  REVERSE(title)
FROM
  film AS f;
```

```
+-----+
| title          | reverse(title) |
+-----+
| ACADEMY DINOSAUR | RUASONID YMEDACA |
| ACE GOLDFINGER   | REGNIFDLOG ECA   |
+-----+
```

Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

Parsing string and character data

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

SQL

Brian Piccolo

Sr. Director, Digital Strategy

Determining the length of a string

```
SELECT
    title,
    CHAR_LENGTH(title)
FROM film;
```

```
+-----+-----+
| title          | CHAR_LENGTH(title) |
+-----+-----+
| ACADEMY DINOSAUR | 16                  |
| ACE GOLDFINGER   | 14                  |
| ADAPTATION HOLES | 16                  |
+-----+-----+
```

Determining the length of a string

```
SELECT
    title,
    LENGTH(title)
FROM film;
```

```
+-----+-----+
| title          | LENGTH(title) |
+-----+-----+
| ACADEMY DINOSAUR | 16            |
| ACE GOLDFINGER  | 14            |
| ADAPTATION HOLES | 16            |
+-----+-----+
```

Finding the position of a character in a string

```
SELECT
    email,
    POSITION('@' IN email)
FROM customer;
```

```
+-----+-----+
| email                                | POSITION('@' IN email) |
+-----+-----+
| MARY.SMITH@sakilacustomer.org       | 11                     |
| PATRICIA.JOHNSON@sakilacustomer.org | 17                     |
| LINDA.WILLIAMS@sakilacustomer.org   | 15                     |
+-----+-----+
```


Finding the position of a character in a string

```
SELECT
    email,
    STRPOS(email, '@')
FROM customer;
```

```
+-----+-----+
| email                                | STRPOS(email, '@') |
+-----+-----+
| MARY.SMITH@sakilacustomer.org        | 11                  |
| PATRICIA.JOHNSON@sakilacustomer.org  | 17                  |
| LINDA.WILLIAMS@sakilacustomer.org    | 15                  |
+-----+-----+
```

Parsing string data

```
SELECT
    LEFT(description, 50)
FROM film;
```

```
+-----+
| description |
+-----+
| A Epic Drama of a Feminist And a Mad Scientist who |
| A Astounding Epistle of a Database Administrator A |
| A Astounding Reflection of a Lumberjack And a Car |
+-----+
```

Parsing string data

```
SELECT
    RIGHT(description, 50)
FROM film;
```

```
+-----+
| description |
+-----+
| who must Battle a Teacher in The Canadian Rockies |
| nd a Explorer who must Find a Car in Ancient China |
| Car who must Sink a Lumberjack in A Baloon Factory |
+-----+
```

Extracting substrings of character data

```
SELECT
    SUBSTRING(description, 10, 50)
FROM
    film AS f;
```

```
+-----+
| description |
+-----+
| ama of a Feminist And a Mad Scientist who must Bat |
| ing Epistle of a Database Administrator And a Expl |
| ing Reflection of a Lumberjack And a Car who must |
+-----+
```

Extracting substrings of character data

```
SELECT
    SUBSTRING(email FROM 0 FOR POSITION('@' IN email))
FROM
    customer;
```

```
+-----+
| SUBSTRING(email FROM 0 FOR POSITION('@' IN email)) |
|-----|
| MARY.SMITH                                         |
| PATRICIA.JOHNSON                                  |
| LINDA.WILLIAMS                                     |
+-----+
```

Extracting substrings of character data

```
SELECT
    SUBSTRING(email FROM POSITION('@' IN email)+1 FOR CHAR_LENGTH(email))
FROM
    customer;
```

```
+-----+
| SUBSTRING(email FROM POSITION('@' IN email)+1 FOR CHAR_LENGTH(email)) |
|-----|
| sakilacustomer.org |
| sakilacustomer.org |
| sakilacustomer.org |
+-----+
```

Extracting substrings of character data

```
SELECT
```

```
    SUBSTR(description, 10, 50)
```

```
FROM
```

```
  film AS f;
```

```
+-----+
| description |
+-----+
| ama of a Feminist And a Mad Scientist who must Bat |
| ing Epistle of a Database Administrator And a Expl |
| ing Reflection of a Lumberjack And a Car who must |
+-----+
```

Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

Truncating and padding string data

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

SQL

Brian Piccolo

Sr. Director, Digital Strategy

Removing whitespace from strings

```
TRIM([leading | trailing | both] [characters] from string)
```

- **First parameter:** [leading | trailing | both]
- **Second parameter:** [characters]
- **Third parameter:** from string

Removing whitespace from strings

```
SELECT TRIM(' padded');
```

```
+-----+  
| TRIM   |  
|-----|  
| padded |  
+-----+
```

Removing whitespace from strings

```
SELECT LTRIM(' padded');
```

```
+-----+  
| LTRIM |  
+-----+  
| padded |  
+-----+
```

Removing whitespace from strings

```
SELECT RTRIM(' padded');
```

```
+-----+  
| RTRIM |  
+-----+  
| padded |  
+-----+
```

Padding strings with character data

```
SELECT LPAD('padded', 10, '#');
```

```
+-----+  
| LPAD   |  
|-----|  
| ####padded |  
+-----+
```

Padding strings with whitespace

```
SELECT LPAD('padded', 10);
```

```
+-----+  
| LPAD   |  
|-----|  
| padded |  
+-----+
```

```
SELECT LPAD('padded', 5);
```

```
+-----+  
| LPAD   |  
|-----|  
|  padde |  
+-----+
```

Padding strings with whitespace

```
SELECT RPAD('padded', 10, '#');
```

```
+-----+  
| RPAD   |  
|-----|  
| padded#### |  
+-----+
```


Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL