

Replace

REPLACE – Zeichen und Strings effizient ersetzen

Einführung

Die SQL-Funktion **REPLACE** wird verwendet, um Teile eines Textes oder Strings durch andere Werte zu ersetzen. Sie ist eine leistungsstarke Funktion, um Datenbereinigung und Transformationen durchzuführen, insbesondere wenn unerwünschte Zeichen oder Muster in Strings vorkommen.

In diesem Thema lernst du:

1. Die Grundlagen und Syntax von **REPLACE**.
2. Praktische Anwendungsbeispiele für die Datenmanipulation.
3. Wie du **REPLACE** mit anderen SQL-Funktionen kombinierst.

Syntax von REPLACE

Die allgemeine Syntax lautet:

```
1 REPLACE(string, old_text, new_text)
2
```

SQL



Parameter:

- **string** : Die Eingabezeichenkette, in der ein Teil ersetzt werden soll.
- **old_text** : Der Teilstring, der ersetzt werden soll.
- **new_text** : Der neue Teilstring, der anstelle des alten eingefügt wird.

Ergebnis:

Die Funktion gibt eine neue Zeichenkette zurück, bei der **alle Vorkommen** von **old_text** in der Eingabezeichenkette durch **new_text** ersetzt wurden.



Beispiele

Beispiel 1: Einfache Ersetzung

Angenommen, du möchtest in einer Zeichenkette alle Leerzeichen durch Bindestriche ersetzen:

SQL



```
1 SELECT REPLACE('Hello World', ' ', '-') AS  
   replaced_string;  
2
```

Ergebnis:

Hello-World

Beispiel 2: Entfernen von Zeichen

Wenn du ein bestimmtes Zeichen komplett entfernen möchtest, kannst du es durch einen leeren String (' ') ersetzen:

SQL



```
1 SELECT REPLACE('123-456-789', '-', '') AS  
   cleaned_string;  
2
```

Ergebnis:

123456789

Beispiel 3: Ersetzen von Wörtern

Nehmen wir an, du möchtest in einer Produktbeschreibung das Wort "alt" durch "neu" ersetzen:

SQL



```
1 SELECT REPLACE('Dieses altmodische Produkt ist alt',  
   'alt', 'neu') AS updated_text;  
2
```

Ergebnis:

Dieses neumodische Produkt ist neu

Praktische Anwendungsfälle

1. Bereinigung von Sonderzeichen

Wenn deine Datenbank unerwünschte Sonderzeichen wie # oder & enthält, kannst du diese entfernen oder ersetzen.

SQL



```
1 SELECT REPLACE('Data & Analytics #1', '&', 'and') AS  
   cleaned_text;
```

2

Ergebnis:

Data and Analytics #1

2. Ersetzen von Präfixen

Wenn bestimmte Werte ein unerwünschtes Präfix enthalten, kannst du es durch einen leeren String ersetzen:

```
1 SELECT REPLACE('ID-12345', 'ID-', '') AS product_id;
2
```

Ergebnis:

12345

3. Standardisierung von Werten

Du kannst unterschiedliche Schreibweisen eines Wertes vereinheitlichen, z. B. "OK", "ok", "Okay" zu "ok":

```
1 SELECT REPLACE(LOWER('Okay'), 'okay', 'ok') AS
  standardized_value;
2
```

Ergebnis:

ok

4. Datenbereinigung in Kombination mit anderen Funktionen

Kombiniere REPLACE mit anderen Funktionen wie TRIM oder LOWER, um umfassendere Transformationen durchzuführen:

```
1 SELECT TRIM(REPLACE(LOWER('RETAIL & CONSUMER GOODS
  '), '&', 'and')) AS cleaned_industry;
2
```

Ergebnis:

retail and consumer goods

Einschränkungen von REPLACE

1. Fallunterscheidung:

`REPLACE` in PostgreSQL unterscheidet zwischen Groß- und Kleinschreibung. Wenn du unabhängig von der Schreibweise ersetzen möchtest, nutze `LOWER()` oder `UPPER()`.

```
1 SELECT REPLACE(LOWER('HELLO World'), 'hello', 'hi') AS
  replaced_text;
2
```

Ergebnis:

```
hi world
```

2. Keine Unterstützung für reguläre Ausdrücke:

Wenn du komplexere Muster ersetzen möchtest (z. B. basierend auf regulären Ausdrücken), musst du stattdessen `REGEXP_REPLACE` verwenden.

Übungen

1. Entferne alle Bindestriche (-) aus einer Telefonnummer (z. B. "123-456-789" → "123456789").
2. Ersetze alle Vorkommen des Zeichens # durch das Wort "Nummer" in einer Spalte comments .
3. Standardisiere Schreibweisen des Werts "rcg" auf "Retail & Consumer Goods" .

Best Practices

1. **Kombiniere mit TRIM:**

Bereinige Werte am Anfang und Ende zusätzlich zu Ersetzungen.

2. Fallunterscheidung beachten:

Verwende `LOWER()` oder `UPPER()` , wenn die Groß-/Kleinschreibung keine Rolle spielen soll.

3. Optimierung durch REGEXP_REPLACE :

Für komplexere Ersetzungen mit Mustern ist `REGEXP_REPLACE` sinnvoller als `REPLACE`.

Mit der Funktion `REPLACE` kannst du effizient Zeichenketten transformieren, Daten bereinigen und Text standardisieren. Falls du noch Fragen hast, lass es mich wissen! 😊

