

Prepared statements

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Unported
Licentie

Prepared statements

Wat ?

- Server-side object
- Eventueel parameters
- Kan gebruikt worden om performantie te optimaliseren
 - Meerdere keren gelijkaardige query's uitvoeren
 - Complexe query's met veel joins
- Duurt enkel zolang als de huidige DB connectie of wanneer DEALLOCATE gebruikt wordt
 - ⇒ Kan niet door meerdere DB clients/sessions gebruikt worden
 - ⇒ Elke client kan zijn eigen prepared statements hebben, sessie afhankelijk
- Syntax:
 - ⇒ `PREPARE name [(data_type [, ...])] AS statement`

Prepared statements

Flow

1. PREPARE

2. EXECUTE

...

3. DEALLOCATE

Voorbeeld:

```
PREPARE add_customer (numeric(4,0), varchar(20), varchar(20), date)
```

```
AS INSERT INTO klanten VALUES($1, $2, $3, $4);
```

```
EXECUTE add_customer(100, 'Lieve', 'Standaert', now());
```

Prepared statements

PREPARE *statement* -> statement is...

- Parsed
- Analysed
- Rewritten

EXECUTE *statement* ->
statement is...

- Planned
- Executed

DEALLOCATE *statement* ->
statement is...

- Dropped

Prepared statements

Een Prepared Statement kan onderliggend zijn

- SELECT, UPDATE, DELETE, INSERT, ...
- Met parameters gesubstitueerd volgens positie, gebruik \$1, \$2, etc.
- Write **security**: Voordelig omdat geen andere query's uitgevoerd kunnen worden
- Een prepared statement zal uitvoeren wat voorbereid was

Lijst van huidige prepared statements in uw sessie (PostgreSQL):

```
SELECT *  
FROM pg_prepared_statements;
```

PREPARE

Syntax

- `PREPARE name [(parameter [, ...])] AS ..`

Voorbeeld

- `PREPARE add_customer (numeric(4,0), varchar(20),
varchar(20), date)
AS INSERT INTO klanten VALUES($1, $2, $3, $4);`

EXECUTE

Syntax

- EXECUTE name [(parameter [, ...])]

Voorbeeld

- EXECUTE add_customer(100, 'Lieve', 'Standaert', now());

Nota

- Een lijst van parameter data types kunnen ook gespecificeerd worden, bv.
EXECUTE add_customer(100, 'Lieve', 'Standaert', now()::timestamp); -- :: is een dialect voor cast
- Wanneer het datatype van een parameter onbekend is, wordt het datatype afgeleid uit de context waar de parameter gebruikt is (indien mogelijk).

DEALLOCATE

Syntax

- DEALLOCATE { name | ALL }

Voorbeeld

- DEALLOCATE add_customer;
- DEALLOCATE ALL;

Embedded Prepared statements

Embedded in een andere taal , vb. in Java:

- `PreparedStatement add_customer =
c.prepareStatement("INSERT INTO klanten
VALUES(?, ?, ?, ?)");`

ISO?

Context:

- De SQL standaard bevat een PREPARE statement, maar dit is enkel bedoeld voor embedded SQL.
- De directe PostgreSQL PREPARE statement gebruikt een licht andere syntax dan deze standaard, het is een toevoeging.

Meer info:

<https://www.postgresql.org/docs/current/sql-prepare.html>

Prepared statements

Nota's:

- <https://pgbouncer.github.io/faq.html#how-to-use-prepared-statements-with-transaction-pooling>
- <https://jdbc.postgresql.org/documentation/server-prepare/#server-prepared-statements>
- <https://www.psycopg.org/psycopg3/docs/advanced/prepare.html>