Functions, Procedures, Triggers

1

1

Function, Stored Procedure

- Definition

  a set of SQL and procedural commands such as declarations, assignments, loops, flow-of-control etc. stored on the database server and can be involved using the SQL interface

- Purpose

  Grouping operations often evaluated according to the same scenario

- Creation

  Specified input arguments, and data returned

- Syntax

  https://www.postgresql.org/docs/13/sql-createfunction.html

2

2

Example

```
CREATE OR REPLACE FUNCTION new_customer (IN firstname_in character varying, IN
lastname_in character varying, … IN username_in character varying, …, OUT customerid_out
integer)
RETURNS integer AS $BODY$
DECLARE rows_returned INT;
  BEGIN
    SELECT COUNT(*) INTO rows_returned
    FROM CUSTOMERS WHERE USERNAME = username_in;
    IF rows_returned = 0 THEN
            INSERT INTO CUSTOMERS (FIRSTNAME, LASTNAME, ….       )
            VALUES (firstname_in, lastname_in, …. ) ;
            select currval(pg_get_serial_sequence('customers', 'customerid')) into customerid_out;
     ELSE
            customerid_out := 0;
     END IF;
END
$BODY$  LANGUAGE plpgsql VOLATILE  COST 100;
```

3

3

Example

```
CREATE OR REPLACE FUNCTION select_products(IN cat integer)
RETURNS TABLE(prod_id integer, category integer, title character
varying, actor character varying, price numeric, special smallint, common
integer)
AS $$
BEGIN
    RETURN QUERY  SELECT * from products where category = $1;
END
$$
LANGUAGE 'plpgsql';
```

Call function: select select_products(6)

4

4

Prepare

- PREPARE ~ prepared statement
- a server-side object used to optimize performance
- Syntax: https://www.postgresql.org/docs/current/sql-prepare.html

5

5

Example

```
Prepare select_products(int) AS
    SELECT * from products where category = $1;

Execute select_products(6);
```

6

6

Triggers

- associated with the specified table and will perform certain operations on that table.
- Associated with event INSERT, UPDATE, DELETE
- Syntax https://www.postgresql.org/docs/13/sql-createtrigger.html

7

7

Example

```
CREATE OR REPLACE FUNCTION check_totalamount_update()
RETURNS trigger AS $BODY$
BEGIN
  IF ((SELECT * FROM ORDERS
      WHERE orderid = new.orderid and totalamount <> netamount + tax) is not null)
 THEN
            UPDATE ORDERS
            SET totalamount = netamount + tax
            WHERE orderid = new.orderid;
 END IF;
Return new;
END     $BODY$  LANGUAGE plpgsql VOLATILE  COST 100;

CREATE TRIGGER order_insert
    AFTER INSERT ON orders
    FOR EACH ROW
    EXECUTE PROCEDURE check_totalamount_update();
```

8

8