Homework 2.3

Students:

Salvador Orozco Villalever - A07104218

Aranzza Abascal Fararoni - A01329203

Professor:

Dr. Daniel Pérez Rojas

Subject:

Advanced Databases

Due date:

March 9th, 2018

ITESM Campus Puebla

1. Instructions

Implement the scenario proposed by someone else in the 2.2 homework.

Chosen scenario: Orthodontic office
Scenario Author: José Alfredo Jiménez

2. Database Objects

2.1. Events

(Although it couldn't be implemented because PostgreSQL doesn't have built-in event support and because of the lack of the cron installation.)

SELECT cron.schedule('0 9 * * *', $$SELECT COUNT(*)
FROM appointments
WHERE date_time::date = current_date$$);

2.2. Functions

```
CREATE FUNCTION get_sku(product_id INTEGER)

RETURNS table(id INTEGER,
              name VARCHAR(30),
              sku INTEGER)

AS $$
BEGIN
    RETURN query
    SELECT p.id AS id,
           p.name AS name,
           p.sku AS quantity
    FROM products p
    WHERE p.id = product_id;

END;
$$ LANGUAGE plpgsql;
```

Function to get the sku of a given product

2.3. Stored procedures

```
CREATE FUNCTION add_appointment(patient_id INTEGER,
                                date_time TIMESTAMP,
                                appointment_type_id INTEGER,
                                dentist_id INTEGER)

RETURNS void
AS $$
BEGIN
```

```
    LOCK TABLE appointments IN EXCLUSIVE MODE;

    INSERT INTO appointments VALUES(DEFAULT,
                                    patient_id,
                                    FALSE,
                                    date_time,
                                    NOW(),
                                    null,
                                    appointment_type_id,
                                    dentist_id);
END;
$$ LANGUAGE plpgsql;
```

Stored procedure to insert into appointments table

2.4. Transactions

The lock before the insertion in the stored procedure guarantees that this is a transaction; PostgreSQL does not support transactions within stored procedures yet.

2.5. Triggers

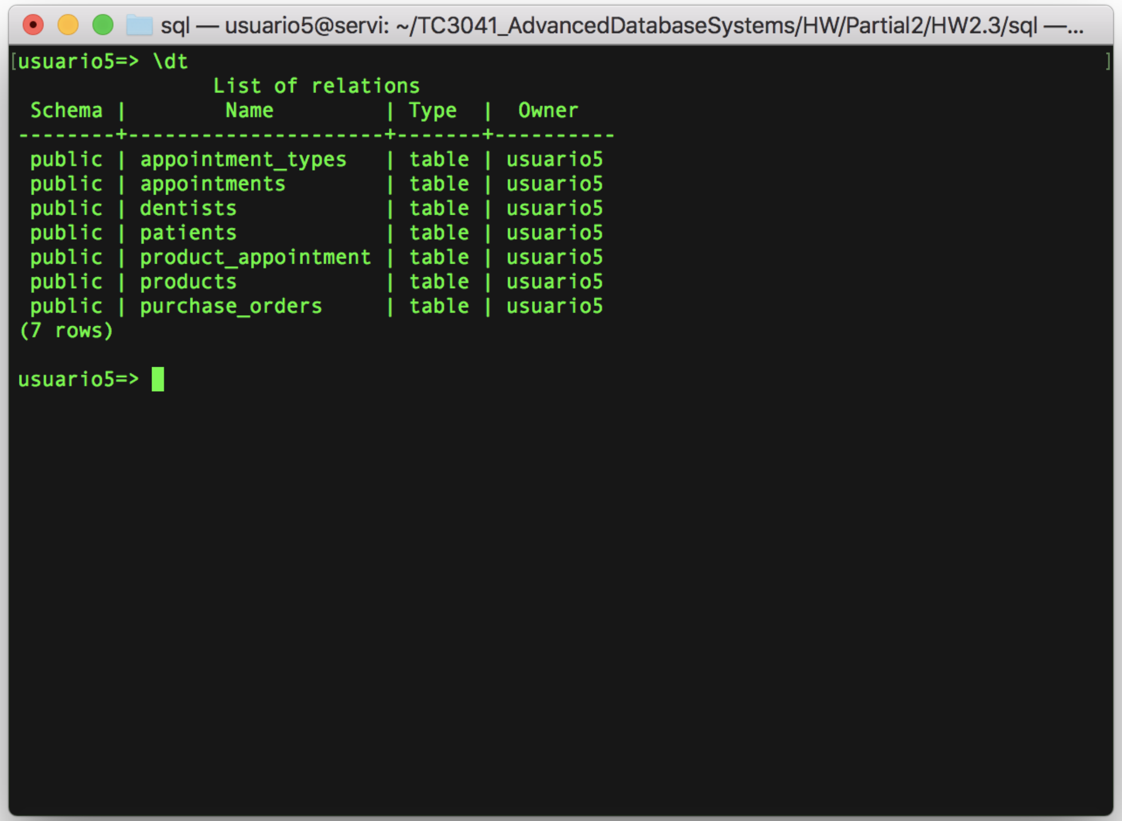
```
CREATE FUNCTION order_product_if_required()

RETURNS void
AS $$
BEGIN
    IF NEW.sku <= NEW.min_req_sku THEN
        INSERT INTO purchase_orders VALUES(DEFAULT,
                                           NEW.id,
                                           NOW(),
                                           NEW.min_req_sku);
    END IF;
END;
```

Function that will be executed in trigger

```
CREATE TRIGGER after_product_sale
  AFTER UPDATE
  ON products
  FOR EACH ROW
  EXECUTE PROCEDURE order_product_if_required();
```

Trigger to insert into purchase_ orders table when sku is lesser or equal to its minimum sku required

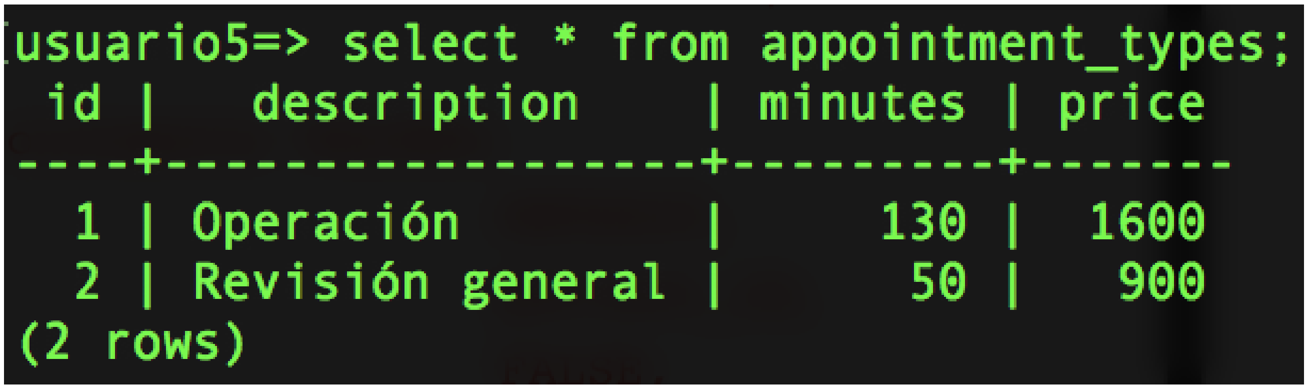
3. Screenshots



sql — usuario5@servi: ~/TC3041_AdvancedDatabaseSystems/HW/Partial2/HW2.3/sql —...

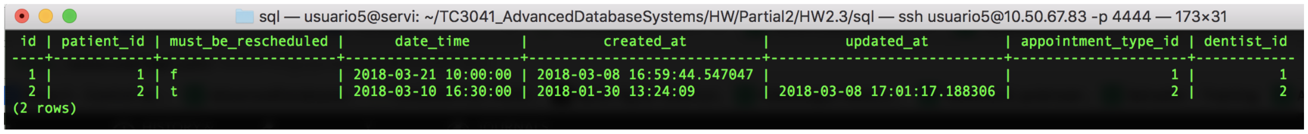
```
usuario5=> \dt
                 List of relations
 Schema |        Name         | Type  |  Owner
--------+---------------------+-------+----------
 public | appointment_types   | table | usuario5
 public | appointments        | table | usuario5
 public | dentists            | table | usuario5
 public | patients            | table | usuario5
 public | product_appointment | table | usuario5
 public | products            | table | usuario5
 public | purchase_orders     | table | usuario5
(7 rows)

usuario5=>
```

Figura 1: *Tables of the "dentists"database*



```
usuario5=> select * from appointment_types;
 id |    description     | minutes | price
----+--------------------+---------+-------
  1 | Operación          |     130 |  1600
  2 | Revisión general   |      50 |   900
(2 rows)
```

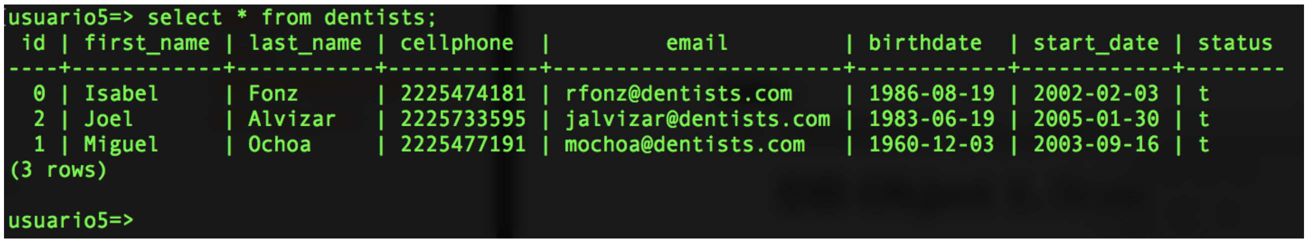
Figura 2: *Contents of the .ªppointment_ types"table*



sql — usuario5@servi: ~/TC3041_AdvancedDatabaseSystems/HW/Partial2/HW2.3/sql — ssh usuario5@10.50.67.83 -p 4444 — 173×31

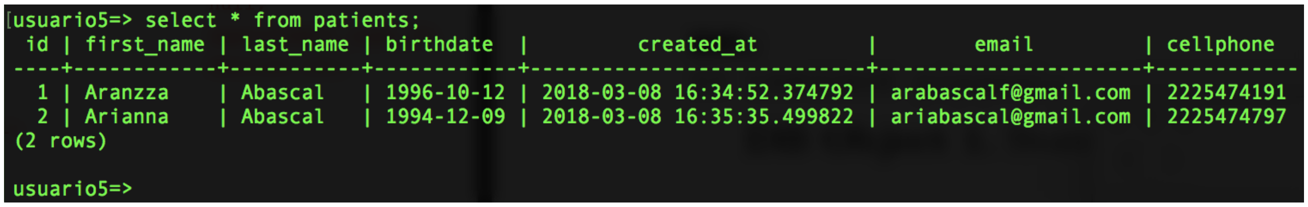
```
 id | patient_id | must_be_rescheduled |      date_time      |         created_at         |         updated_at         | appointment_type_id | dentist_id
----+------------+---------------------+---------------------+----------------------------+----------------------------+---------------------+------------
  1 |          1 | f                   | 2018-03-21 10:00:00 | 2018-03-08 16:59:44.547047 |                            |                   1 |          1
  2 |          2 | t                   | 2018-03-10 16:30:00 | 2018-01-30 13:24:09        | 2018-03-08 17:01:17.188306 |                   2 |          2
(2 rows)
```

Figura 3: *Contents of the appointments table*



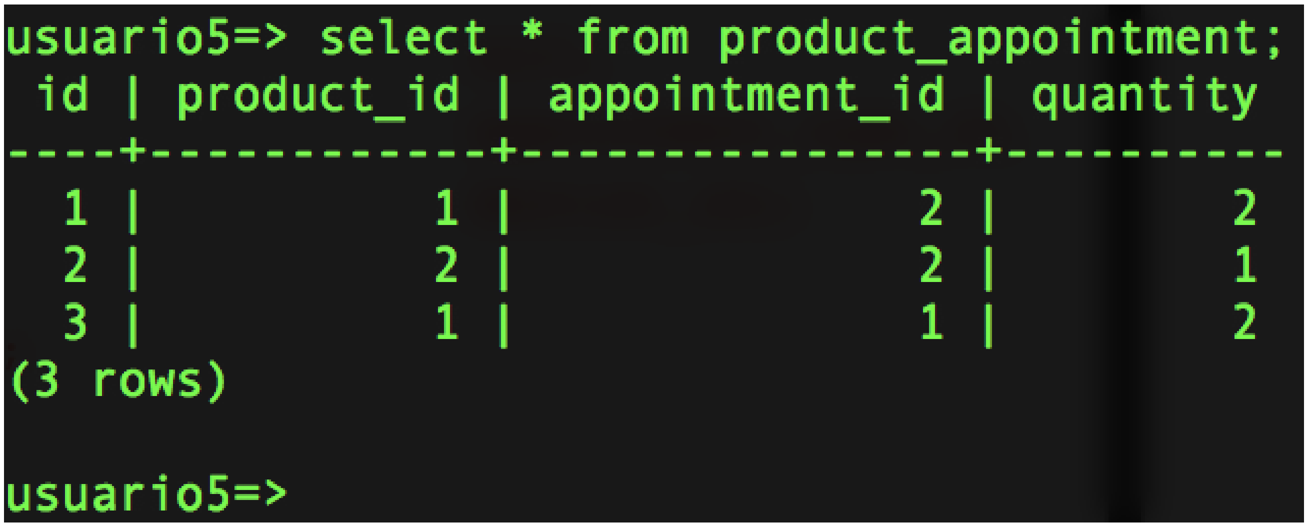
```
usuario5=> select * from dentists;
 id | first_name | last_name | cellphone  |         email          | birthdate  | start_date | status
----+------------+-----------+------------+------------------------+------------+------------+--------
  0 | Isabel     | Fonz      | 2225474181 | rfonz@dentists.com     | 1986-08-19 | 2002-02-03 | t
  2 | Joel       | Alvizar   | 2225733595 | jalvizar@dentists.com  | 1983-06-19 | 2005-01-30 | t
  1 | Miguel     | Ochoa     | 2225477191 | mochoa@dentists.com    | 1960-12-03 | 2003-09-16 | t
(3 rows)

usuario5=>
```

Figura 4: *Contents of the dentists table*



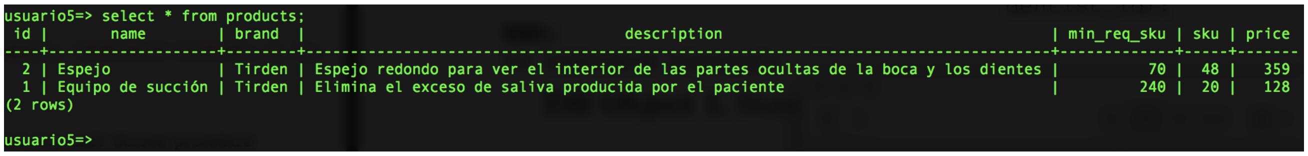
```
usuario5=> select * from patients;
 id | first_name | last_name | birthdate  |         created_at         |         email          | cellphone
----+------------+-----------+------------+----------------------------+------------------------+------------
  1 | Aranzza    | Abascal   | 1996-10-12 | 2018-03-08 16:34:52.374792 | arabascalf@gmail.com   | 2225474191
  2 | Arianna    | Abascal   | 1994-12-09 | 2018-03-08 16:35:35.499822 | ariabascal@gmail.com   | 2225474797
(2 rows)

usuario5=>
```

Figura 5: *Contents of the patients table*



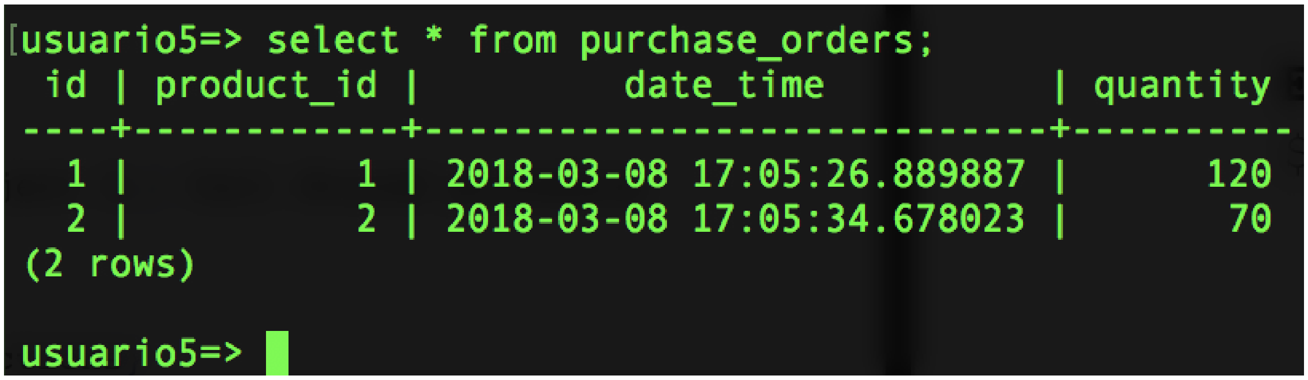
```
usuario5=> select * from product_appointment;
 id | product_id | appointment_id | quantity
----+------------+----------------+----------
  1 |          1 |              2 |        2
  2 |          2 |              2 |        1
  3 |          1 |              1 |        2
(3 rows)

usuario5=>
```

Figura 6: *Contents of the product_ appointment table*



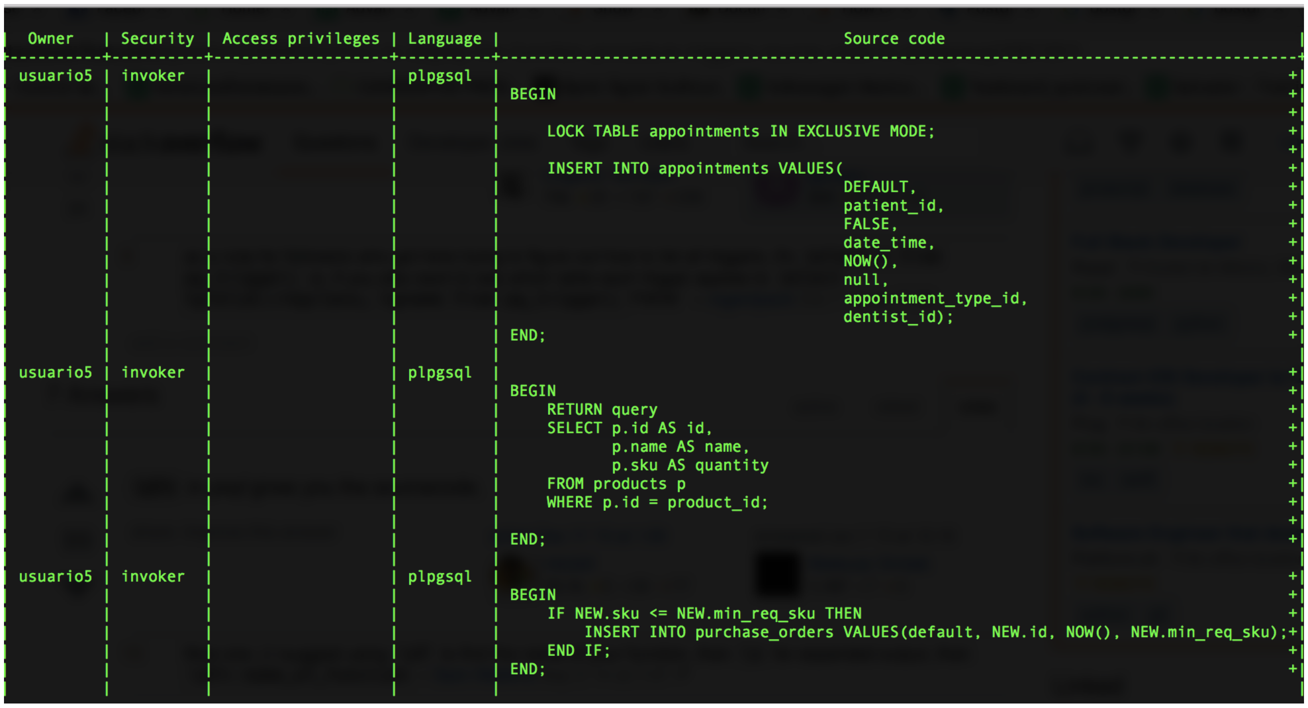
```
usuario5=> select * from products;
 id |       name        | brand  |                                    description                                     | min_req_sku | sku | price
----+-------------------+--------+------------------------------------------------------------------------------------+-------------+-----+-------
  2 | Espejo            | Tirden | Espejo redondo para ver el interior de las partes ocultas de la boca y los dientes |          70 |  48 |   359
  1 | Equipo de succión | Tirden | Elimina el exceso de saliva producida por el paciente                              |         240 |  20 |   128
(2 rows)

usuario5=>
```

Figura 7: *Contents of the products table*



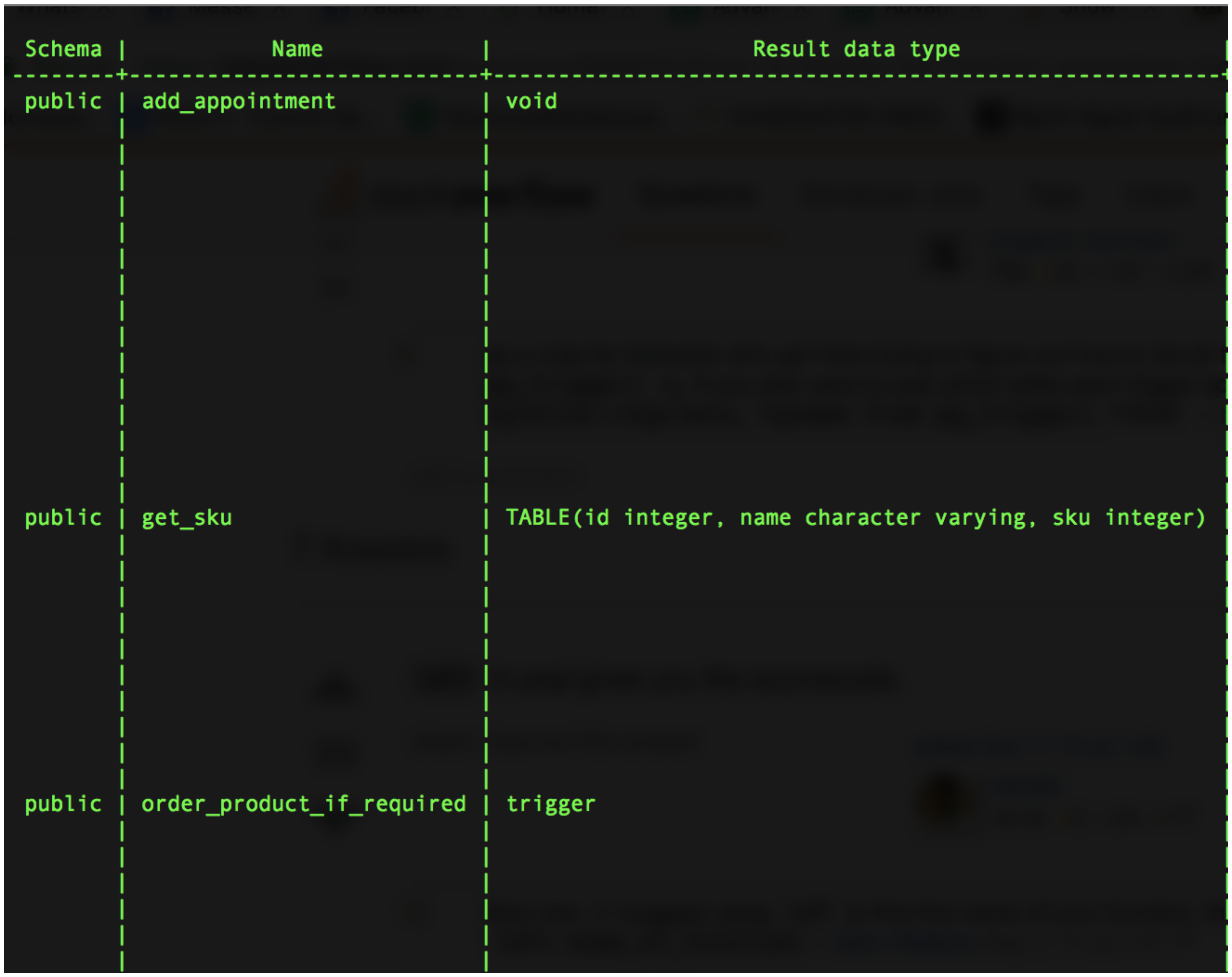
```
[usuario5=> select * from purchase_orders;
 id | product_id |          date_time          | quantity
----+------------+----------------------------+----------
  1 |          1 | 2018-03-08 17:05:26.889887 |      120
  2 |          2 | 2018-03-08 17:05:34.678023 |       70
(2 rows)

usuario5=>
```

Figura 8: *Contents of the purchase_ order table*



```
  Owner   | Security | Access privileges | Language |                              Source code
----------+----------+-------------------+----------+------------------------------------------------------------------------
 usuario5 | invoker  |                   | plpgsql  |                                                                        +
          |          |                   |          | BEGIN                                                                  +
          |          |                   |          |                                                                        +
          |          |                   |          |     LOCK TABLE appointments IN EXCLUSIVE MODE;                         +
          |          |                   |          |                                                                        +
          |          |                   |          |     INSERT INTO appointments VALUES(                                   +
          |          |                   |          |                                 DEFAULT,                               +
          |          |                   |          |                                 patient_id,                            +
          |          |                   |          |                                 FALSE,                                 +
          |          |                   |          |                                 date_time,                             +
          |          |                   |          |                                 NOW(),                                 +
          |          |                   |          |                                 null,                                  +
          |          |                   |          |                                 appointment_type_id,                   +
          |          |                   |          |                                 dentist_id);                           +
          |          |                   |          | END;                                                                   +
          |          |                   |          |
 usuario5 | invoker  |                   | plpgsql  |                                                                        +
          |          |                   |          | BEGIN                                                                  +
          |          |                   |          |     RETURN query                                                       +
          |          |                   |          |     SELECT p.id AS id,                                                 +
          |          |                   |          |            p.name AS name,                                             +
          |          |                   |          |            p.sku AS quantity                                           +
          |          |                   |          |     FROM products p                                                    +
          |          |                   |          |     WHERE p.id = product_id;                                           +
          |          |                   |          |                                                                        +
          |          |                   |          | END;                                                                   +
          |          |                   |          |
 usuario5 | invoker  |                   | plpgsql  |                                                                        +
          |          |                   |          | BEGIN                                                                  +
          |          |                   |          |     IF NEW.sku <= NEW.min_req_sku THEN                                 +
          |          |                   |          |         INSERT INTO purchase_orders VALUES(default, NEW.id, NOW(), NEW.min_req_sku);+
          |          |                   |          |     END IF;                                                            +
          |          |                   |          | END;                                                                   +
          |          |                   |          |
```

Figura 9: *DB objects 1*



```
 Schema |           Name           |                    Result data type
--------+--------------------------+----------------------------------------------------------
 public | add_appointment          | void
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
 public | get_sku                  | TABLE(id integer, name character varying, sku integer)
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
 public | order_product_if_required | trigger
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
        |                          |
```

Figura 10: *DB objects 2*