

# Procedimientos almacenados y disparadores, ¿para qué son necesarios?

NOMBRE Y APELLIDOS: GLORIA MARÍA MANRESA SANTAMARÍA

# **EJERCICIO 1 (35%)**

#### APARTADO A

Se modifica la tabla para insertar como valor por defecto la fecha y hora de la inserción de la fila. De esta manera cada vez que se inserte una nueva fila de datos se rellenará automáticamente este campo.

```
1 -- EJERCICIO 1A
 2
     BEGIN WORK;
 3
     ALTER TABLE erp.tb_pump
 4
 5
           ADD COLUMN updated_dt_tm timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP;
 6
 7
     COMMIT WORK;
 8
 9
     SELECT *
     FROM erp.tb_pump
Data Output
               Messages
                            Notifications
=+
      pm_id
                     pm_parent_id
                                      gas_id
                                                      pm_descr
                                                                             updated_dt_tm
                                      character (5)
                                                      character varying (20)
                                                                             timestamp without time zone
      [PK] integer
1
                  1
                                      GS01
                                                      GS01
                                                                              2022-12-10 11:44:45.883894
                  2
2
                                      GS02
                                                      GS02
                                                                              2022-12-10 11:44:45.883894
3
                  3
                                      GS03
                                                      GS03
                                                                              2022-12-10 11:44:45.883894
                                      GS04
                                                      GS04
                                                                              2022-12-10 11:44:45.883894
4
                  4
5
                  5
                                      GS05
                                                      GS05
                                                                              2022-12-10 11:44:45.883894
                                      GS01
                                                      Surtidor Nº1
                                                                              2022-12-10 11:44:45.883894
6
                  6
                                  1
7
                  7
                                  1
                                      GS01
                                                      Surtidor Nº2
                                                                              2022-12-10 11:44:45.883894
8
                  8
                                  1
                                      GS01
                                                      Surtidor Nº3
                                                                              2022-12-10 11:44:45 883894
9
                  9
                                  1
                                      GS01
                                                      Surtidor Nº4
                                                                              2022-12-10 11:44:45.883894
10
                 10
                                  2
                                      GS02
                                                      Surtidor Nº1
                                                                              2022-12-10 11:44:45.883894
                                      GS02
                                                      Surtidor Nº2
                                                                              2022-12-10 11:44:45.883894
11
                 11
                                  2
```







#### **APARTADO B**

```
-- EJERCICIO 1B
1
2
3 -- Añadir nuevo campo
4 ALTER TABLE erp.tb_lines_invoice
   ADD COLUMN line_updated_dt_tm varchar(20);
5
6
7
8 -- Crear función
9 CREATE OR REPLACE FUNCTION fn_line_inserted()
10 RETURNS TRIGGER AS $$
11 ▼ BEGIN
12 NEW.line_updated_dt_tm = TO_CHAR(NOW(), 'yyyy-mm-dd HH24:MI');
13
   RETURN NEW;
14 END;
   $$ LANGUAGE plpgsql;
15
16
   -- Crear disparador
17
18 CREATE TRIGGER tg_line_inserted
19 BEFORE UPDATE OR INSERT
20 ON erp.tb_lines_invoice
```

Para rellenar la fila de datos se han borrado los datos de la tabla y vuelto a insertar una vez creado el trigger.

Se ha comprobado que el trigger también funciona al modificar los datos.







#### **APARTADO C**

```
-- EJERCICIO 1C
1
2
   -- CREAR TRES NUEVOS ATRIBUTOS SEGÚN EL ENUNCIADO
3
1
5
   ALTER TABLE erp.tb_invoice
6
        ADD COLUMN inv_updated_dt DATE,
7
        ADD COLUMN inv_update_counter int DEFAULT 0,
8
        ADD COLUMN inv_insert_counter int DEFAULT 0;
9
11
   -- CREAR FUNCIÓN
   CREATE OR REPLACE FUNCTION fn_invoice_updated()
12
13 RETURNS TRIGGER AS $$
14 ▼ BEGIN
15
16 ▼ if TG_OP='INSERT' then
   UPDATE erp.tb_invoice
17
18
   SET inv_updated_dt = current_date,
19
        inv_insert_counter = (SELECT count(inv_id)
20
                                FROM erp.tb_lines_invoice
21
                                WHERE inv_id = new.inv_id
                              GROUP BY inv id
22
23
    WHERE inv_id = new.inv_id;
24
25
    end if;
26
27 ▼ if TG_OP='UPDATE' then
   UPDATE erp.tb_invoice
28
29 SET
           inv_updated_dt = current_date,
            inv_update_counter = inv_update_counter+1
30
   WHERE inv_id = old.inv_id;
31
32
   end if;
   RETURN NEW;
33
34
   END;
   $$ LANGUAGE plpgsql;
35
38
   -- Crear disparador
   CREATE TRIGGER tg_invoice_updated
39
   AFTER INSERT OR UPDATE ON erp.tb_lines_invoice
40
   FOR EACH ROW EXECUTE PROCEDURE fn_invoice_updated();
41
42
```



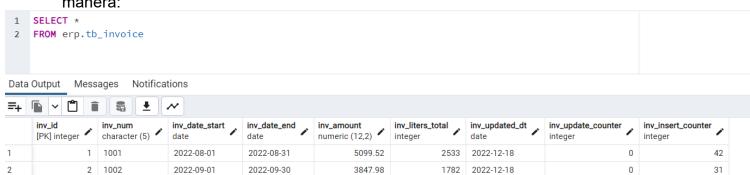




0

31

# La tabla tb\_invoice después de crear el procedimiento anterior queda de la siguiente manera:



5498.16

2721 2022-12-18

3

3 1003

2022-10-01

2022-10-31





# **EJERCICIO 2 (55%)**

#### APARTADO A

```
1 -- EJERCICIO 2A
2 -- CREAR NUEVA COLUMNA
   ALTER TABLE erp.tb_refueling
4 ADD COLUMN rf_cost NUMERIC;
6 -- CREAR FUNCIÓN
7 CREATE OR REPLACE FUNCTION fn_calc_cost()
8 RETURNS TRIGGER AS $$
9 ▼ BEGIN
10
11 UPDATE erp.tb_refueling
12 SET rf_cost = (SELECT (rf_liters*fp_import)
13
                   FROM (erp.tb_refueling NATURAL JOIN erp.tb_cars)
14
                          LEFT JOIN erp.tb_fuel_price ON (rf_date=fp_date AND cars_fuel=fp_fuel)
15
                   WHERE cars_registration = new.cars_registration AND
                   rf_liters = new.rf_liters AND
16
                   rf_date = new.rf_date
17
18
19 WHERE cars_registration = new.cars_registration AND
20
       rf_liters = new.rf_liters AND
21
          rf_date = new.rf_date;
22
23 RETURN NEW;
   END;
24
25
   $$ LANGUAGE plpgsql;
28 -- Crear disparador
29 CREATE TRIGGER tg_refueling_cost
30 AFTER INSERT ON erp.tb_refueling
31 FOR EACH ROW EXECUTE PROCEDURE fn_calc_cost();
```

Para que se actualicen los datos de la columna rf\_cost, se han borrado los datos de la tabla y se han introducido de nuevo.

La tabla queda de la siguiente manera:







1 SELECT \*

2 FROM erp.tb\_refueling

3

### Data Output Messages Notifications

=+							
	gas_id character (5)	cars_registration character (7)	pm_id integer	rf_liters integer	rf_date date	rf_km integer	rf_cost numeric
1	GS01	3685HDP	29	43	2022-09-19	303075	85.828
2	GS01	3685HDP	27	33	2022-09-30	[null]	65.637
3	GS02	3685HDP	33	33	2022-08-15	304275	62.667
4	GS02	3685HDP	35	30	2022-08-30	304825	60.660
5	GS02	3685HDP	35	20	2022-09-25	305392	38.360
6	GS03	3685HDP	37	42	2022-09-04	306108	82.194
7	GS04	3685HDP	43	33	2022-09-09	306442	63.558
8	GS05	3685HDP	55	34	2022-09-14	306992	69.054



#### APARTADO B

```
1 -- EJERCICIO 2B
 2 -- CREAR FUNCIÓN
 3
   CREATE OR REPLACE FUNCTION fn_get_cost_by_car_type(initial_date DATE,
 4
                                                       final_date DATE,
                                                       car_function CHARACTER(20))
 5
        RETURNS TABLE (
 6
           car_function_selected CHARACTER(20),
7
8
           range_date DATE,
           rf_liters_selected INT,
9
           rf_cost_selected NUMERIC
10
11
        )
12
        AS $$
        BEGIN
13
            RETURN QUERY
14
15
                    SELECT cars_function, rf_date, rf_liters,rf_cost
16
                    FROM erp.tb_cars NATURAL JOIN erp.tb_refueling
                    WHERE cars_function=car_function AND rf_date BETWEEN initial_date AND final_date;
17
18
19
20 ▼ IF LENGTH(car_function)>20 THEN
21 RAISE EXCEPTION 'Longitud excesiva del valor introducido para el tipo de vehículo.';
22 END IF;
23
24 END;
25 $$LANGUAGE plpgsql;
```

#### Comprobamos el funcionamiento de la función:

```
1    SELECT *
2    FROM fn_get_cost_by_car_type('2022-08-18','2022-08-28','comercial')
```

Data Output Messages Notifications

=+		<b>.</b> ~			
	car_function_selected character	range_date date	rf_liters_selected integer	rf_cost_selected numeric	
1	comercial	2022-08-19	16	32.944	
2	comercial	2022-08-24	22	46.200	
3	comercial	2022-08-27	21	43.281	
4	comercial	2022-08-23	31	65.503	
5	comercial	2022-08-23	28	61.964	
6	comercial	2022-08-26	23	50.002	
7	comercial	2022-08-20	30	65.970	







#### APARTADO C

```
1 -- EJERCICIO 2C
2
    CREATE TYPE tb_invoice_cost_summary AS(
3
        cars_registration CHARACTER(10),
4
 5
        cars_fuel
                           CHARACTER(10),
                           INT,
        invoice_year
 6
 7
        invoice_quarter
                            INT,
8
        invoice_month
                            INT,
9
        total_liters
                            INT,
10
        total_cost
                           NUMERIC(10,2),
        number_of_lines
11
                            INT,
12
        date_time
                            CHARACTER(20)
13
14
   );
15
16
    CREATE OR REPLACE FUNCTION invoice_cost_summary()
17
18 RETURNS SETOF tb_invoice_cost_summary AS $$
19 DECLARE
20
   car tb_invoice_cost_summary;
21 ▼ BEGIN
    FOR car IN
22
                SELECT cars_registration,
23
24
                        cars_fuel,
25
                        EXTRACT(YEAR FROM inv_date_start),
                        EXTRACT(QUARTER FROM inv_date_start),
26
                        EXTRACT(MONTH FROM inv_date_start),
27
                        SUM(linv_liters),
28
29
                        SUM(linv_amount),
30
                        COUNT(cars_registration),
                        line_updated_dt_tm
31
32
                FROM
                        erp.tb_lines_invoice
33
                        NATURAL JOIN erp.tb_cars
34
                        NATURAL JOIN erp.tb_invoice
                GROUP BY
                            cars_registration,
35
36
                            cars_fuel,
37
                            EXTRACT(YEAR FROM inv_date_start),
                            EXTRACT(QUARTER FROM inv_date_start),
38
39
                            EXTRACT(MONTH FROM inv_date_start),
40
                            line_updated_dt_tm
41
41
42 ₹
                  L<sub>00</sub>P
43
    RETURN NEXT car;
44
    END LOOP;
45
    RETURN;
46
   END;
47
    $$LANGUAGE plpgsql;
```







# Comprobamos el funcionamiento de la función:

- 1 SELECT \*
- 2 FROM invoice\_cost\_summary()

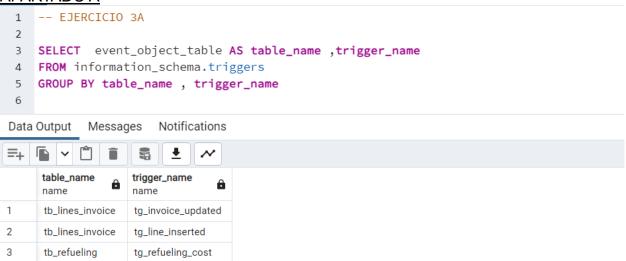
Data	Data Output Messages Notifications								
=+									
	cars_registration character (10)	cars_fuel character (10)	invoice_year integer	invoice_quarter integer	invoice_month integer	total_liters integer	total_cost numeric (10,2)	number_of_lines integer	date_time character (20)
1	0019GVM	gasoil	2022	3	8	301	601.43	5	2022-12-18 18:13
2	0815GYR	gasoil	2022	3	9	308	615.42	5	2022-12-18 18:13
3	6392KPT	gasoil	2022	3	9	166	331.83	4	2022-12-18 18:13
4	4273GFK	gasoil	2022	4	10	478	964.19	5	2022-12-18 18:13
5	3685HDP	gasoil	2022	3	9	188	373.93	2	2022-12-18 18:13
6	8806KZN	gasolina	2022	3	8	108	235.58	3	2022-12-18 18:13
7	3685HDP	gasoil	2022	3	8	188	373.93	2	2022-12-18 18:13
8	2093GSW	gasoil	2022	3	9	394	787.19	5	2022-12-18 18:13
9	0815GYR	gasoil	2022	3	8	308	615.42	5	2022-12-18 18:13
10	2093GSW	gasoil	2022	4	10	394	787.19	5	2022-12-18 18:13





# **EJERCICIO 3 (10%)**

#### APARTADO A



En la captura anterior podemos observar que tenemos 3 triggers en la base de datos. También podemos ver a que tabla hace referencia cada uno de los triggers.

#### APARTADO B

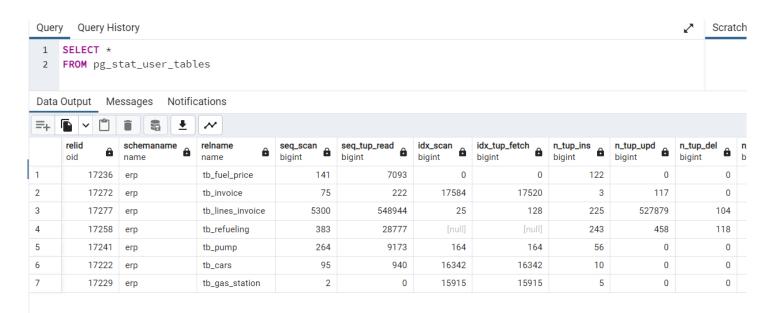
Statistics collector es un subsistema de PostgreSQL que recopila información sobre la actividad del servidor y nos aporta información sobre como está trabajando PostgreSQL.

Podemos, por ejemplo, ver una tabla con todos los usuarios y roles de la base de datos (pg\_roles), obtener información sobre los clientes conectados a la base de datos (pg\_stat\_activity) o obtener información de todas las tablas de usuario, entre otras posibilidades.

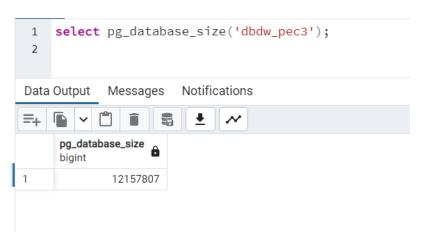








También se puede obtener información sobre el tamaño de la base de datos. En la siguiente imagen podemos observar la consulta para obtener dicha información, en bytes.







## Criterios de valoración

En el enunciado se indica el peso/valoración de cada ejercicio.

Para conseguir la puntuación máxima en los ejercicios, es necesario explicar con claridad la solución que se propone.

# Formato y fecha de entrega

Tenéis que enviar la PEC al buzón de Entrega y registro de EC disponible en el aula (apartado Evaluación). El formato del archivo que contiene vuestra solución puede ser .pdf, .doc y .docx. Para otras opciones, por favor, contactar previamente con vuestro consultor. El nombre del fichero debe contener el código de la asignatura, vuestro apellido y vuestro nombre, así como el número de actividad (PEC3).

La fecha límite para entregar la PEC3 es el 19/12/2022.

#### Nota: Propiedad intelectual

Al presentar una práctica o PEC que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL etc.). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica o PEC. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Será necesario, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente, si así corresponde.



