Pruebas de Base de Datos

Creación tablas:

CATEGORIES

```sql
CREATE TABLE categories (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100)
);
```

PRODUCTS

```sql
CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    price DECIMAL(10, 2),
    category_id INT REFERENCES categories(id)
);
```

INVENTORY

```sql
CREATE TABLE inventory (
    id SERIAL PRIMARY KEY,
    product_id INT REFERENCES products(id),
    quantity INT
);
```

Inserción de datos:

CATEGORIES

```sql
INSERT INTO categories (id, name) VALUES
(1, 'Electrónicos'),
(2, 'Ropa'),
(3, 'Hogar');
```

PRODUCTS

```sql
INSERT INTO products (id, name, price, category_id) VALUES
(1, 'Laptop Acer', 799.99, 1),
(2, 'Smartphone Samsung', 499.99, 1),
(3, 'Tablet Lenovo', 299.99, 1);
```

INVENTORY

```sql
INSERT INTO inventory (product_id, quantity) VALUES
(1, 10),
(2, 25),
(3, 15);
```

Consulta Simple – SELECT

Descripción: Verificar que se puedan consultar todos los productos de la tienda.

```
select * from categories;
```

Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)
1	1	Electrónicos
2	2	Ropa
3	3	Hogar

```
select * from products;
```

Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	15	Logitech	79.99	1
5	14	Mouse Logitech	899.99	2

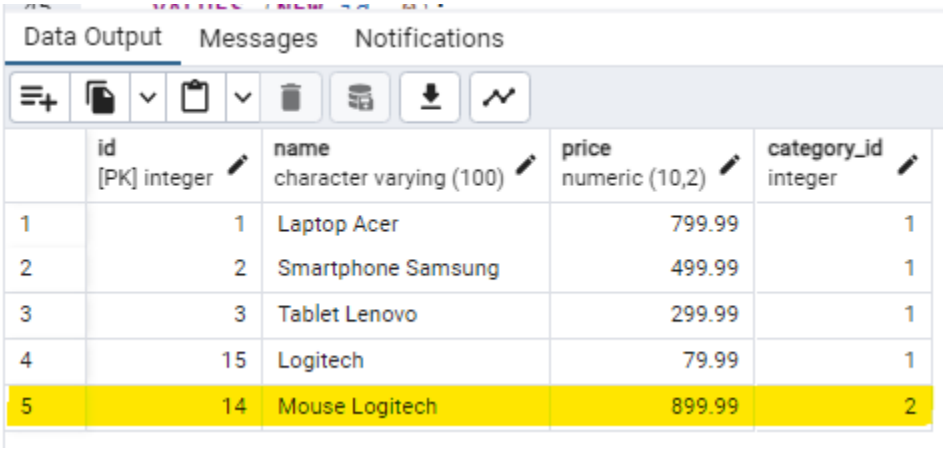
```
select * from inventory;
```

Data Output | Messages | Notifications

	id [PK] integer	product_id integer	quantity integer
1	1	1	10
2	2	2	25
3	3	3	15
4	5	15	76
5	4	14	98

Consulta Simple - UPDATE:

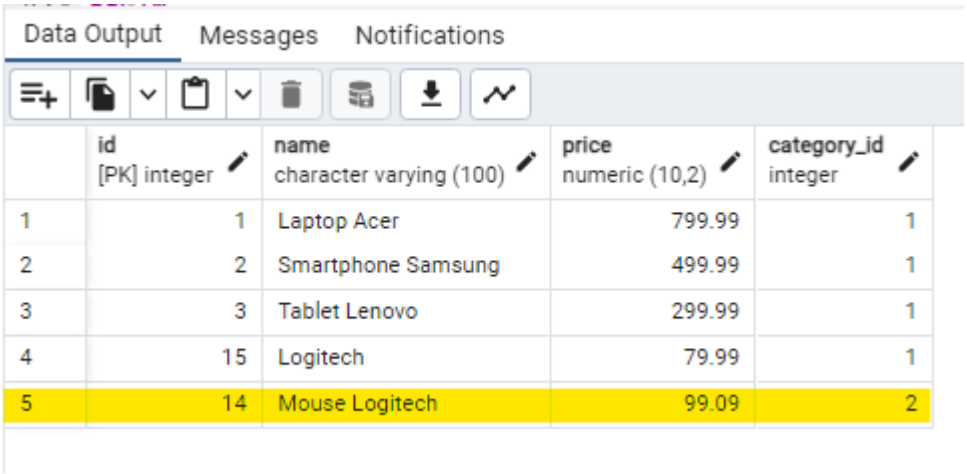
Descripción: Verificar que se pueda actualizar la información de un producto.



Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	15	Logitech	79.99	1
5	14	Mouse Logitech	899.99	2

```
update products set price = 99.09 where id = 14
```

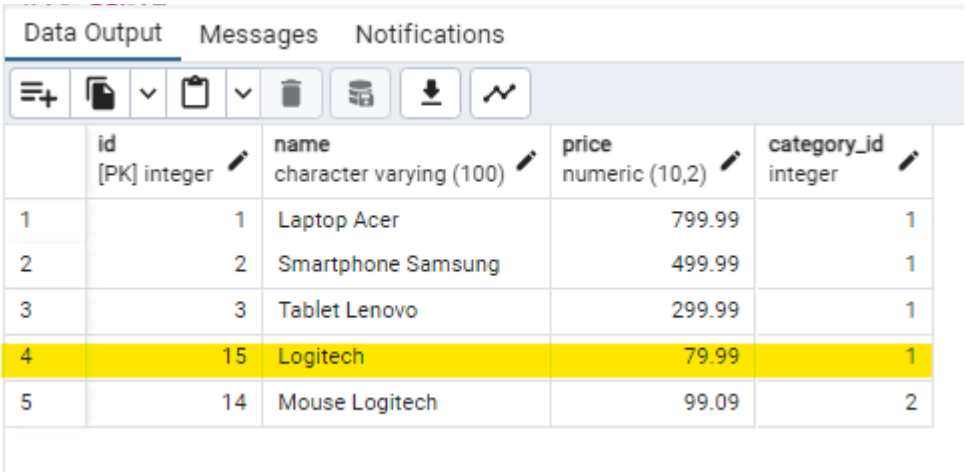


Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	15	Logitech	79.99	1
5	14	Mouse Logitech	99.09	2

Consulta Simple - DELETE:

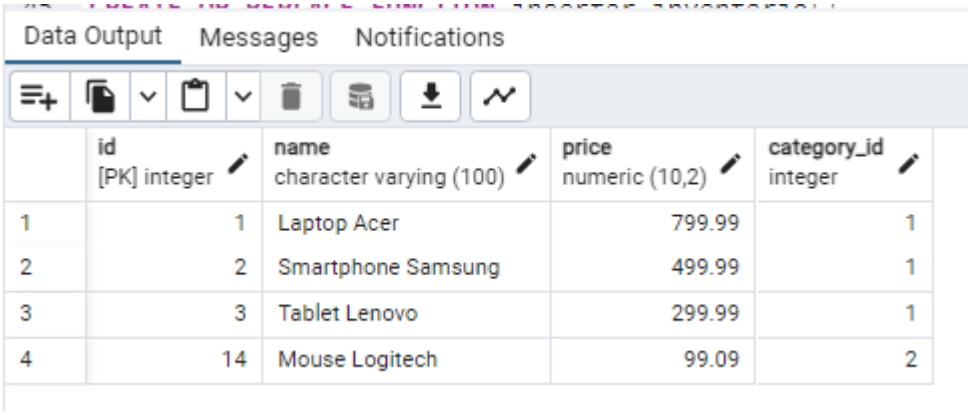
Descripción: Verificar que se pueda eliminar un producto.



Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	15	Logitech	79.99	1
5	14	Mouse Logitech	99.09	2

```
delete from products where id = 15
```



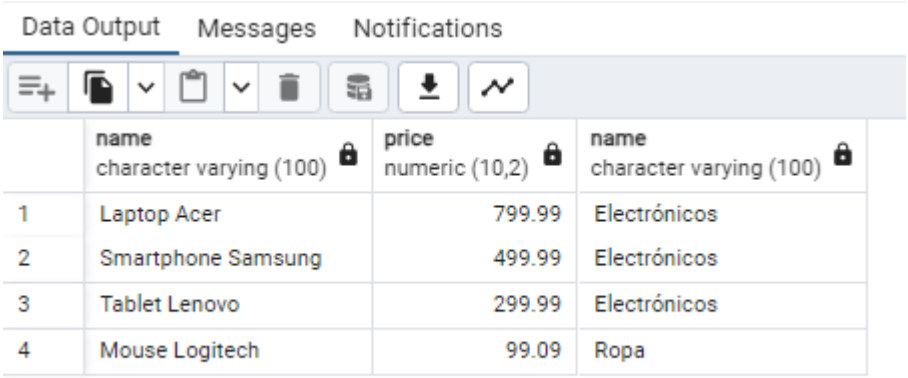
Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	14	Mouse Logitech	99.09	2

Consulta JOIN:

Descripción: Verificar que se puedan obtener los productos junto con su categoría.

```
select pr.name, pr.price, ct.name from products pr
inner join categories ct
on pr.category_id = ct.id
```



Data Output | Messages | Notifications

	name character varying (100)	price numeric (10,2)	name character varying (100)
1	Laptop Acer	799.99	Electrónicos
2	Smartphone Samsung	499.99	Electrónicos
3	Tablet Lenovo	299.99	Electrónicos
4	Mouse Logitech	99.09	Ropa

Creación del Trigger:

Descripción: Verificar que se active un trigger al insertar un nuevo producto en la tabla 'products' para mantener actualizado el inventario en la tabla 'inventory'.

```
CREATE OR REPLACE FUNCTION insertar_inventario()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO inventory (product_id, quantity)
    VALUES (NEW.id, 0);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;


CREATE TRIGGER trg_insert_inventario
AFTER INSERT ON products
FOR EACH ROW
EXECUTE FUNCTION insertar_inventario();
```

Ejecución:

Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	14	Mouse Logitech	99.09	2

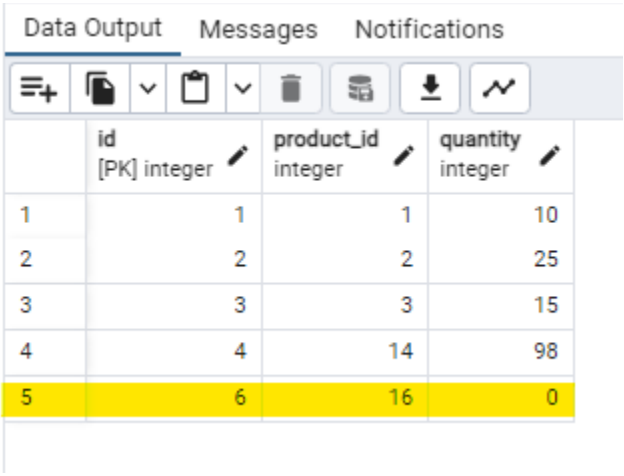
```
INSERT INTO products (name, price, category_id) VALUES ('Buzo lana', 109.99, 2);
```

Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	14	Mouse Logitech	99.09	2
5	16	Buzo lana	109.99	2

Ejecución del trigger

```
select * from inventory;
```



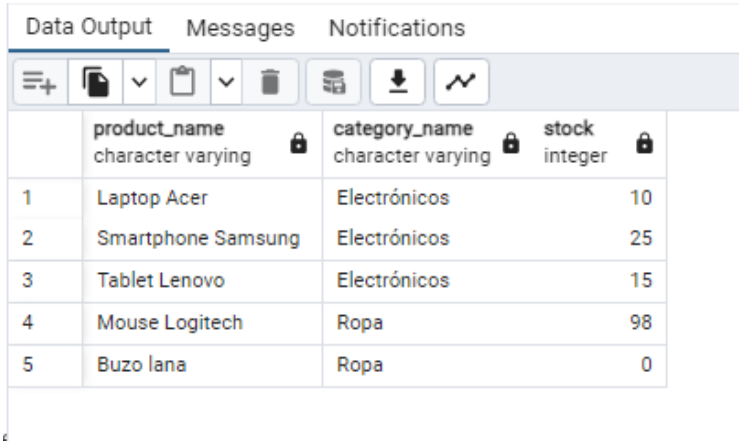
Data Output | Messages | Notifications

	id [PK] integer	product_id integer	quantity integer
1	1	1	10
2	2	2	25
3	3	3	15
4	4	14	98
5	6	16	0

Stored Procedure:

Descripción: Crear un Stored Procedure que devuelva el nombre y la cantidad disponible de un producto en base a su ID.

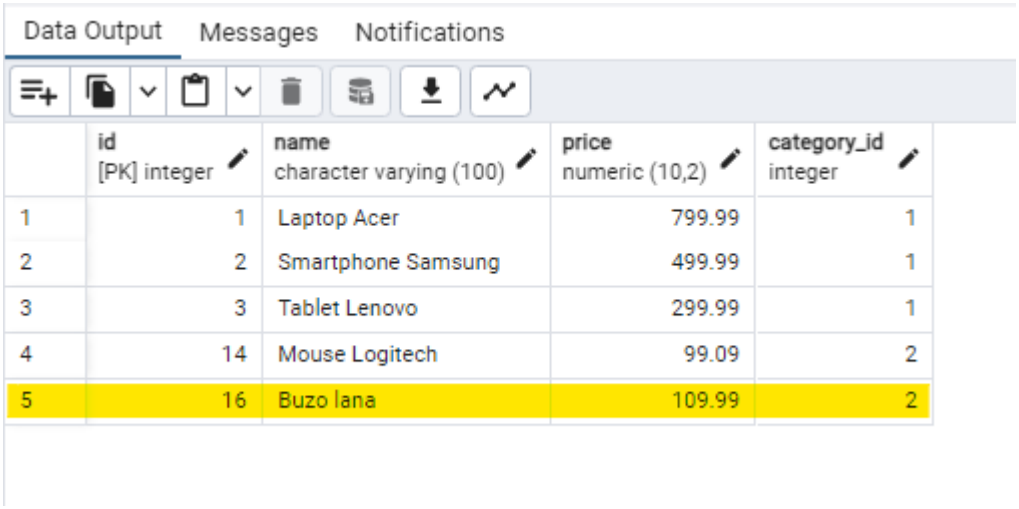
```
CREATE OR REPLACE FUNCTION obtener_reporte_productos()
RETURNS TABLE (
    product_name VARCHAR,
    category_name VARCHAR,
    stock INT
)
AS $$
BEGIN
    RETURN QUERY
    SELECT p.name, c.name, i.quantity
    FROM products p
    JOIN categories c ON p.category_id = c.id
    JOIN inventory i ON p.id = i.product_id;
END;
$$ LANGUAGE plpgsql;

SELECT * FROM obtener_reporte_productos();
```



Data Output | Messages | Notifications

	product_name character varying	category_name character varying	stock integer
1	Laptop Acer	Electrónicos	10
2	Smartphone Samsung	Electrónicos	25
3	Tablet Lenovo	Electrónicos	15
4	Mouse Logitech	Ropa	98
5	Buzo lana	Ropa	0

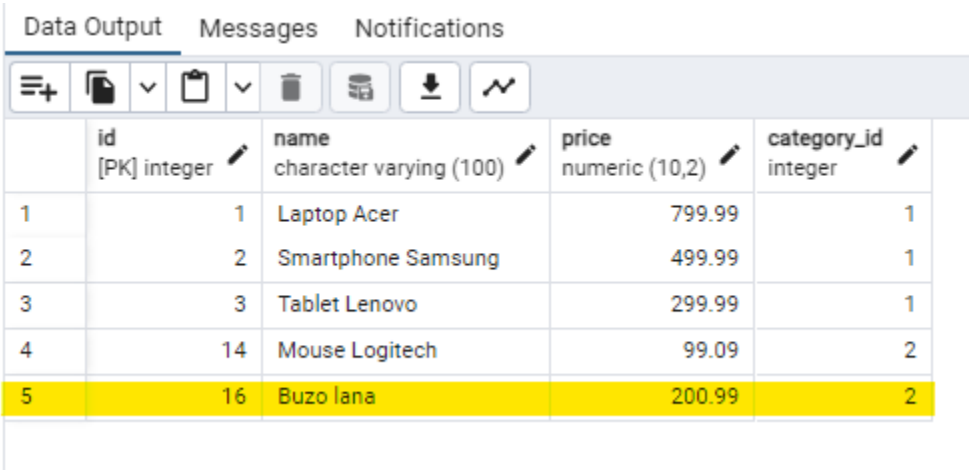
Stored Procedure



Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	14	Mouse Logitech	99.09	2
5	16	Buzo lana	109.99	2

```sql
CREATE OR REPLACE PROCEDURE actualizar_precio_producto(p_id INT, nuevo_precio DECIMAL)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE products SET price = nuevo_precio WHERE id = p_id;
    COMMIT;
END;
$$;

CALL actualizar_precio_producto(16, 200.99);
```



Data Output | Messages | Notifications

	id [PK] integer	name character varying (100)	price numeric (10,2)	category_id integer
1	1	Laptop Acer	799.99	1
2	2	Smartphone Samsung	499.99	1
3	3	Tablet Lenovo	299.99	1
4	14	Mouse Logitech	99.09	2
5	16	Buzo lana	200.99	2