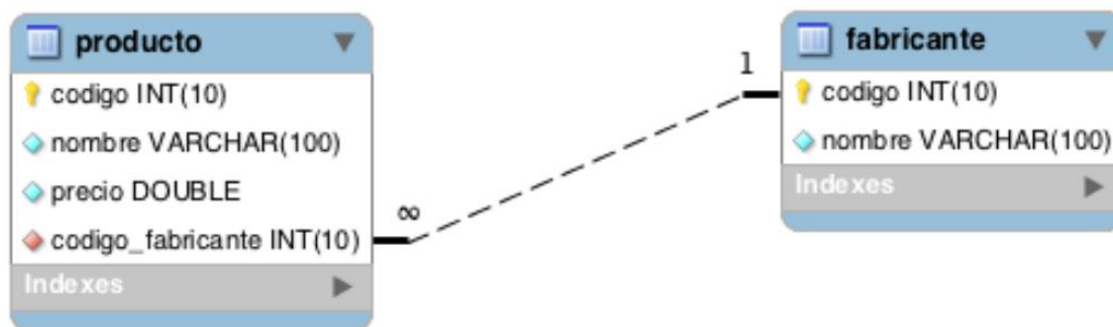
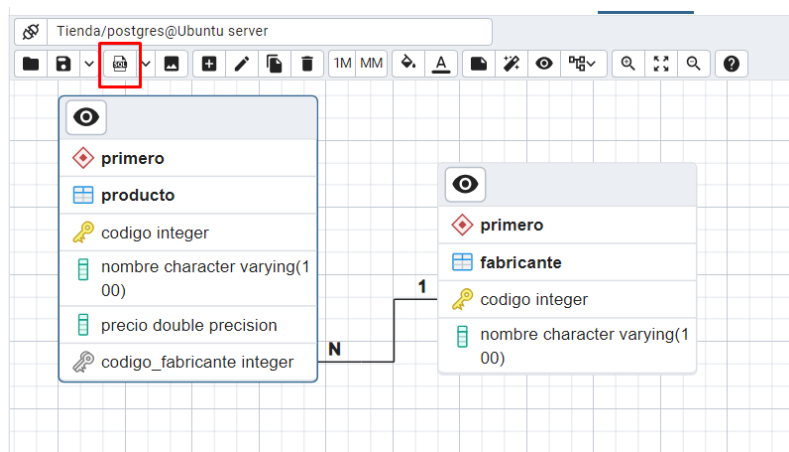


1. Desde PgAdmin crea las siguientes tablas y relaciones en una nueva BBDD llamada Tienda.



Mediante el uso de ERD tool que esta disponible en la pestaña Tools en la parte de arriba de la ventana podremos acceder a este “entorno gráfico” y crear las tablas como queramos y conectarlas haciendo uso de las diferentes opciones que nos aparecen. Una vez esté todo a nuestro gusto dándole al botón remarcado podremos convertir este diseño en una instrucción.



```

1  -- This script was generated by the ERD tool in pgAdmin 4.
2  -- Please log an issue at https://redmine.postgresql.org/projects/pgadmin4/issues/new if you find any bugs
3  BEGIN;
4
5
6  CREATE TABLE IF NOT EXISTS primero.producto
7  (
8      codigo integer NOT NULL,
9      nombre character varying(100),
10     precio double precision,
11     codigo_fabricante integer,
12     PRIMARY KEY (codigo)
13 );
14
15  CREATE TABLE IF NOT EXISTS primero.fabricante
16  (
17     codigo integer NOT NULL,
18     nombre character varying(100),
19     PRIMARY KEY (codigo)
20 );
21
22  ALTER TABLE IF EXISTS primero.producto
23  ADD FOREIGN KEY (codigo_fabricante)
24  REFERENCES primero.fabricante (codigo) MATCH SIMPLE
25  ON UPDATE NO ACTION
26  ON DELETE NO ACTION
27  NOT VALID;
28
29  END;
    
```

2. Introduce los siguientes fabricantes:

1. Seagate
2. Crucial
3. Samsung
4. Gigabyte
5. Asus
6. Lenovo
7. Hewlett-Packard

Manualmente usando el propio gestor de pgAdmin.

Query

Query History

1

SELECT * FROM primero.fabricante

2

ORDER BY codigo ASC

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

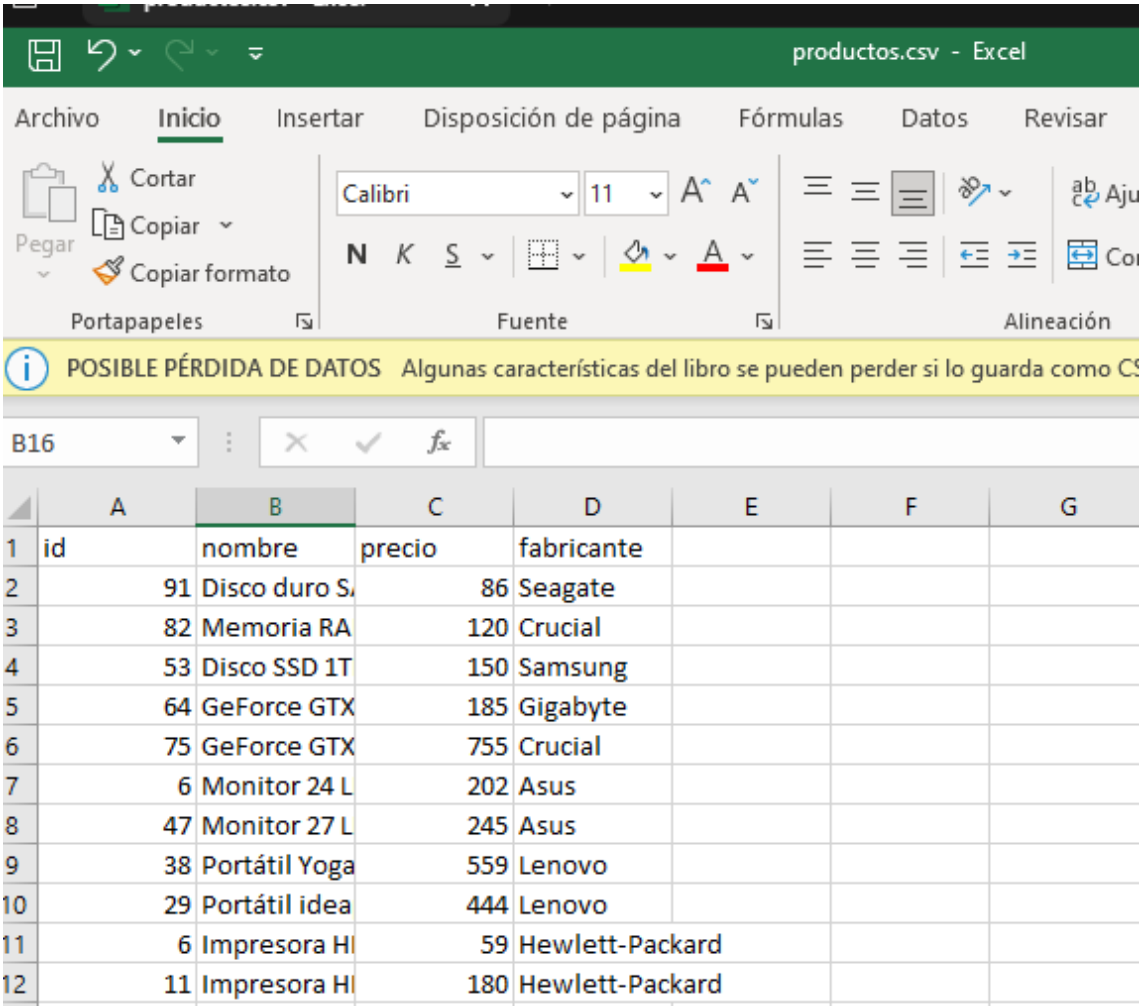
⬇️

📈

	codigo [PK] integer	nombre character varying (100)
1	1	Seagate
2	2	Crucial
3	3	Samsung
4	4	Gigabyte
5	5	Asus
6	6	Lenovo
7	7	Hewlett-Packard

3. Introduce los siguientes productos, utilizando un fichero CSV. Asigna un código cualquiera como identificador primario:

nombre	precio	nombre
Disco duro SATA3 1TB	86	Seagate
Memoria RAM DDR4 8GB	120	Crucial
Disco SSD 1 TB	150	Samsung
GeForce GTX 1050Ti	185	Gigabyte
GeForce GTX 1080 Xtreme	755	Crucial
Monitor 24 LED Full HD	202	Asus
Monitor 27 LED Full HD	245	Asus
Portátil Yoga 520	559	Lenovo
Portátil Ideapd 320	444	Lenovo
Impresora HP Deskjet 3720	59	Hewlett-Packard
Impresora HP Laserjet Pro M26nw	180	Hewlett-Packard



Especifiqué al importar tanto la codificación del archivo como el tipo de separador y que contenía encabezado para evitar fallos.

Import/Export data - table 'producto'

General Options Columns

Import/Export ☒ Import ☐ Export

Filename

Format

Encoding

Import/Export data - table 'producto'

General Options Columns

OID ☐

Header ☒

Delimiter
Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Quote
Specifies the quoting character to be used when a data value is quoted. The default is double-quote. This must be a single one-byte character. This option is allowed only when using CSV format.

Escape
Specifies the character that should appear before a data character that matches the QUOTE value. The default is the same as the QUOTE value (so that the quoting character is doubled if it appears in the data). This must

No se si hay alguna manera de hacerlo poniendo directamente el nombre de los fabricantes pero como esta especificado en las tablas que el fabricante es un código para que coincida con la clave foránea tuve que volver a hacer el csv cambiando los nombres por el código numérico que había en la tabla de fabricantes para evitar fallos.

Process Watcher - Import - Copying table data

Copying table data 'primero.producto' on database 'Tienda' and server 'Ubuntu server (192.168.188.128:5432)'
Running command:

```
--command " \copy primero.producto (codigo, nombre, precio, codigo_fabricante) FROM 'D:/Clase/2º DAM/Sistemas de gestión empresarial/Ejercicios/productos.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' QUOTE '\"' ESCAPE '\"';"
```

Start time: Tue Oct 03 2023 18:25:38 GMT+0200 (hora de verano de Europa central)

Stop Process

ERROR: invalid input syntax for type integer: "Seagate"
CONTEXT: COPY producto, line 2, column codigo_fabricante: "Seagate"

Failed (exit code: 1).

Execution time: 0.13 seconds

productos.csv - Excel

productos.csv - Excel						
Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar						
Cortar Copiar Copiar formato Portapapeles Fuente Alineación						
POSIBLE PÉRDIDA DE DATOS Algunas características del libro se pueden perder si lo guarda como CSV						
	A	B	C	D	E	F
1	id	nombre	precio	fabricante		
2	91	Disco duro S.	86	1		
3	82	Memoria RA	120	2		
4	53	Disco SSD 1T	150	3		
5	64	GeForce GTX	185	4		
6	75	GeForce GTX	755	2		
7	6	Monitor 24 L	202	5		
8	47	Monitor 27 L	245	5		
9	38	Portátil Yoga	559	6		
10	29	Portátil idea	444	6		
11	6	Impresora HI	59	7		
12	11	Impresora HI	180	7		

Con los nuevos códigos ya no da error importarlo

<input type="checkbox"/>	<input type="checkbox"/>	PID	Type	Server	Object	Start Time	Status	Time Taken (sec)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	9796	Import Data	Ubuntu server (192.168.188.128:5...	Tienda/primero.producto	3/10/2023, 18:32:24	Finished	0.08
<input type="checkbox"/>	<input checked="" type="checkbox"/>	29640	Import Data	Ubuntu server (192.168.188.128:5...	Tienda/primero.producto	3/10/2023, 18:25:38	Failed	0.13

Finalmente podemos comprobar que se han añadido correctamente

primero.producto/Tienda/postgres@Ubuntu server

No limit

```

1 SELECT * FROM primero.producto
2 ORDER BY codigo ASC
    
```

Data Output Messages Notifications

	codigo [PK] integer	nombre character varying (100)	precio double precision	codigo_fabricante integer
1	6	Monitor 24 LED Full HD	202	5
2	8	Impresora HP deskjet 3720	59	7
3	11	Impresora HP Laserjet Pro M26nw	180	7
4	29	Portátil ideapad 320	444	6
5	38	Portátil Yoga 520	559	6
6	47	Monitor 27 LED Full HD	245	5
7	53	Disco SSD 1TB	150	3
8	64	GeForce GTX 1050Ti	185	4
9	75	GeForce GTX 1080 Xtreme	755	2
10	82	Memoria RAM DDR4 8GB	120	2
11	91	Disco duro SATA3 1TB	86	1

4. Realiza las siguientes consultas SQL sobre la base de datos

En los ejercicios o) y p) tuve que hacer más búsquedas y utilizar cosas con las que no recuerdo haber trabajado el curso anterior como la subquery y el DISTINCT ya que con lo que sabía hacer no tenía manera de resolverlos o almenos a mi no se me ocurría.

Se me olvidó hacer capturas para mostrar que se había añadido un producto en el ejercicio q), la modificación en el r) y los precios antes de aplicar el descuento en el s) pero como los precios antes eran enteros y en la captura final tienen decimales y también se puede comprobar el producto que añadí y el que modifiqué en esa lista no creo que haya problemas.

a) Selecciona los nombres de todos los productos de la tienda.

Tienda/postgres@Ubuntu server

No limit

Query Query History

1 SELECT nombre FROM primero.producto

Data Output Messages Notifications

	nombre character varying (100)
1	Disco duro SATA3 1TB
2	Memoria RAM DDR4 8GB
3	Disco SSD 1TB
4	GeForce GTX 1050Ti
5	GeForce GTX 1080 Xtreme
6	Monitor 24 LED Full HD
7	Monitor 27 LED Full HD
8	Portátil Yoga 520
9	Portátil Ideapad 320
10	Impresora HP deskjet 3720
11	Impresora HP Laserjet Pro M26nw

b) Selecciona los nombres y los precios de todos los productos de la tienda.

Tienda/postgres@Ubuntu server

Query Query History

```
1 SELECT nombre, precio FROM primero.producto
```

Data Output Messages Notifications

	nombre character varying (100)	precio double precision
1	Disco duro SATA3 1TB	86
2	Memoria RAM DDR4 8GB	120
3	Disco SSD 1TB	150
4	GeForce GTX 1050Ti	185
5	GeForce GTX 1080 Xtreme	755
6	Monitor 24 LED Full HD	202
7	Monitor 27 LED Full HD	245
8	Portátil Yoga 520	559
9	Portátil ideapad 320	444
10	Impresora HP deskjet 3720	59
11	Impresora HP Laserjet Pro M26nw	180

c) Seleccione el nombre de los productos con un precio menor o igual a 200.

Tienda/postgres@Ubuntu server

No limit

Query

Query History

```
1 SELECT nombre, precio FROM primero.producto WHERE precio <= 200
```

Data Output

Messages

Notifications

	nombre character varying (100)	precio double precision
1	Disco duro SATA3 1TB	86
2	Memoria RAM DDR4 8GB	120
3	Disco SSD 1TB	150
4	GeForce GTX 1050Ti	185
5	Impresora HP deskjet 3720	59
6	Impresora HP Laserjet Pro M26nw	180

d) Selecciona todos los productos con un precio entre 60 y 120.

Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
primero.fabrica

Tienda/postgres@Ubuntu server

No limit

Query
Query History

1
SELECT nombre, precio FROM primero.producto WHERE precio BETWEEN 60 AND 120

Data Output
Messages
Notifications

	nombre character varying (100)	precio double precision
1	Disco duro SATA3 1TB	86
2	Memoria RAM DDR4 8GB	120

e) Calcula el precio promedio de todos los productos.

QueryQuery History

1SELECT AVG(precio) AS promedio FROM primero.producto;

Data OutputMessagesNotifications

+File▼Clipboard▼TrashDatabaseDownloadChart

	promedio double precision
1	271.3636363636364

f) Calcula el precio promedio de todos los productos con código de fabricante igual a 2.

Tienda/postgres@Ubuntu server▼Database

FolderSave▼Edit▼Filter▼No limit▼StopPlay▼ExportImport▼RefreshRefresh▼MenuHelp

QueryQuery History

1SELECT AVG(precio) AS promedio FROM primero.producto WHERE codigo_fabricante = 2;

Data OutputMessagesNotifications

+File▼Clipboard▼TrashDatabaseDownloadChart

	promedio double precision
1	437.5

g) Calcula el número de productos con un precio mayor o igual a 180.

QueryQuery History

1SELECT COUNT(*) AS numero_productos FROM primero.producto WHERE precio >= 180;

Data OutputMessagesNotifications

+

	numero_productos
	bigint
1	7

h) Selecciona el nombre y el precio de todos los productos con un precio mayor o igual a 180 y ordena primero por precio (en orden descendente) y luego por nombre (en orden ascendente).

Tienda/postgres@Ubuntu server

No limit

QueryQuery History

1SELECT nombre, precio FROM primero.producto
2WHERE precio >= 180
3ORDER BY precio DESC, nombre ASC;

Data OutputMessagesNotifications

+

	nombre	precio
	character varying (100)	double precision
1	GeForce GTX 1080 Xtreme	755
2	Portátil Yoga 520	559
3	Portátil ideapad 320	444
4	Monitor 27 LED Full HD	245
5	Monitor 24 LED Full HD	202
6	GeForce GTX 1050Ti	185
7	Impresora HP Laserjet Pro M26nw	180

i) Selecciona todos los datos de los productos, incluidos todos los datos del fabricante de cada producto.

Dashboard Properties SQL Statistics Dependencies Dependents Processes primero.fabric... primero.produc... primero.produc...

Tienda/postgres@Ubuntu server

Query Query History Scratch Pad

```

1 SELECT p.*, f.*
2 FROM primero.producto p
3 JOIN primero.fabricante f ON p.codigo_fabricante = f.codigo;

```

Data Output Messages Notifications

	<small>codigo integer</small>	<small>nombre character varying (100)</small>	<small>precio double precision</small>	<small>codigo_fabricante integer</small>	<small>codigo integer</small>	<small>nombre character varying (100)</small>
1	91	Disco duro SATA3 1TB	86	1	1	Seagate
2	82	Memoria RAM DDR4 8GB	120	2	2	Crucial
3	53	Disco SSD 1TB	150	3	3	Samsung
4	64	GeForce GTX 1050Ti	185	4	4	Gigabyte
5	75	GeForce GTX 1080 Xtreme	755	2	2	Crucial
6	6	Monitor 24 LED Full HD	202	5	5	Asus
7	47	Monitor 27 LED Full HD	245	5	5	Asus
8	38	Portátil Yoga 520	559	6	6	Lenovo
9	29	Portátil Ideapad 320	444	6	6	Lenovo
10	8	Impresora HP deskjet 3720	59	7	7	Hewlett-Packard
11	11	Impresora HP LaserJet Pro M26nw	180	7	7	Hewlett-Packard

j) Selecciona el nombre del producto, el precio y el nombre del fabricante de todos los productos.

Query

Query History

```
1 SELECT p.nombre AS nombre_producto, p.precio, f.nombre AS nombre_fabricante
2 FROM primero.producto p
3 JOIN primero.fabricante f ON p.codigo_fabricante = f.codigo;
```

Data Output

Messages

Notifications

	nombre_producto character varying (100)	precio double precision	nombre_fabricante character varying (100)
1	Disco duro SATA3 1TB	86	Seagate
2	Memoria RAM DDR4 8GB	120	Crucial
3	Disco SSD 1TB	150	Samsung
4	GeForce GTX 1050Ti	185	Gigabyte
5	GeForce GTX 1080 Xtreme	755	Crucial
6	Monitor 24 LED Full HD	202	Asus
7	Monitor 27 LED Full HD	245	Asus
8	Portátil Yoga 520	559	Lenovo
9	Portátil ideapad 320	444	Lenovo
10	Impresora HP deskjet 3720	59	Hewlett-Packard
11	Impresora HP Laserjet Pro M26nw	180	Hewlett-Packard

k) Selecciona el precio promedio de los productos de cada fabricante, mostrando solo el código del fabricante.

Dashboard Properties SQL Statistics Dependencies Dependents Processes primero.fabric... p

Tienda/postgres@Ubuntu server

No limit

Query Query History

```

1 SELECT p.codigo_fabricante AS codigo_fabricante, AVG(p.precio) AS precio_promedio
2 FROM primero.producto p
3 GROUP BY p.codigo_fabricante;

```

Data Output Messages Notifications

	codigo_fabricante integer	precio_promedio double precision
1	3	150
2	5	223.5
3	4	185
4	6	501.5
5	2	437.5
6	7	119.5
7	1	86

1) Selecciona el precio promedio de los productos de cada fabricante, mostrando el nombre del fabricante.

Tienda/postgres@Ubuntu server

No limit

Query Query History

```

1 SELECT f.nombre AS nombre_fabricante, AVG(p.precio) AS precio_promedio
2 FROM primero.producto p
3 JOIN primero.fabricante f ON p.codigo_fabricante = f.codigo
4 GROUP BY f.nombre;

```

Data Output Messages Notifications

	nombre_fabricante character varying (100)	precio_promedio double precision
1	Asus	223.5
2	Lenovo	501.5
3	Samsung	150
4	Hewlett-Packard	119.5
5	Gigabyte	185
6	Crucial	437.5
7	Seagate	86

m) Selecciona los nombres de los fabricantes cuyos productos tienen un precio promedio mayor o igual a 150.

Tienda/postgres@Ubuntu server

No limit

Query

Query History

1

2

3

4

5

```
SELECT f.nombre AS nombre_fabricante
FROM primero.fabricante f
JOIN primero.producto p ON f.codigo = p.codigo_fabricante
GROUP BY f.nombre
HAVING AVG(p.precio) >= 150;
```

Data Output

Messages

Notifications

	nombre_fabricante character varying (100)
1	Asus
2	Lenovo
3	Samsung
4	Gigabyte
5	Crucial

n) Selecciona el nombre y precio del producto más económico.

Tienda/postgres@Ubuntu server

No limit

Query

Query History

1

2

3

```
SELECT nombre, precio
FROM primero.producto
WHERE precio = (SELECT MIN(precio) FROM primero.producto);
```

Data Output

Messages

Notifications

	nombre character varying (100)	precio double precision
1	Impresora HP deskjet 3720	59

Tienda/postgres@Ubuntu server

No limit

Query

Query History

```

1 SELECT f.nombre AS nombre_fabricante, p.nombre AS nombre_producto, p.precio AS precio_mas_caro
2 FROM primero.fabricante f
3 JOIN (
4     SELECT codigo_fabricante, MAX(precio) AS precio_maximo
5     FROM primero.producto
6     GROUP BY codigo_fabricante
7 ) subquery
8 ON f.codigo = subquery.codigo_fabricante
9 JOIN primero.producto p
10 ON p.codigo_fabricante = f.codigo
11 AND p.precio = subquery.precio_maximo;

```

Data Output

Messages

Notifications

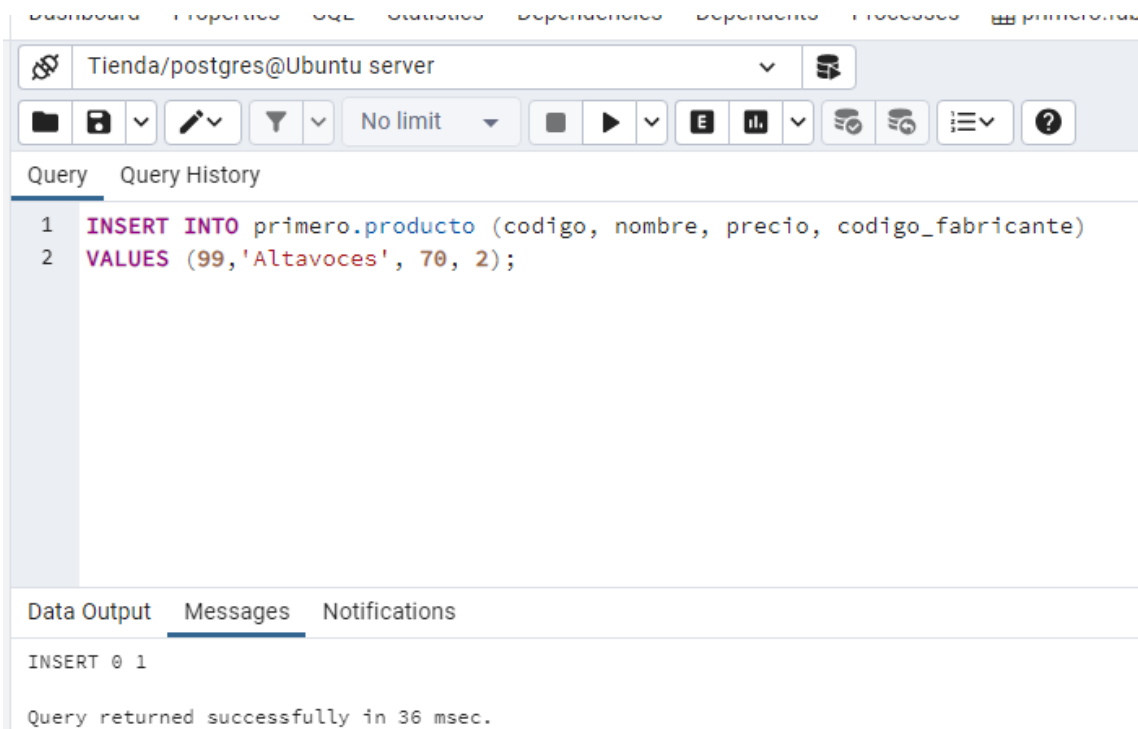
	nombre_fabricante character varying (100)	nombre_producto character varying (100)	precio_mas_caro double precision
1	Seagate	Disco duro SATA3 1TB	86
2	Samsung	Disco SSD 1TB	150
3	Gigabyte	GeForce GTX 1050Ti	185
4	Crucial	GeForce GTX 1080 Xtreme	755
5	Asus	Monitor 27 LED Full HD	245
6	Lenovo	Portátil Yoga 520	559
7	Hewlett-Packard	Impresora HP Laserjet Pro M26nw	180

```
Query Query History
1 SELECT f.nombre AS nombre_fabricante
2 FROM primero.fabricante f
3 JOIN primero.producto p ON f.codigo = p.codigo_fabricante
4 GROUP BY f.nombre
5 HAVING AVG(p.precio) > 145
6      AND COUNT(DISTINCT p.nombre) >= 2;
```

Data Output Messages Notifications

	nombre_fabricante character varying (100) 🔒
1	Asus
2	Crucial
3	Lenovo

q) Agrega un nuevo producto: Altavoces, 70, fabricante 2.



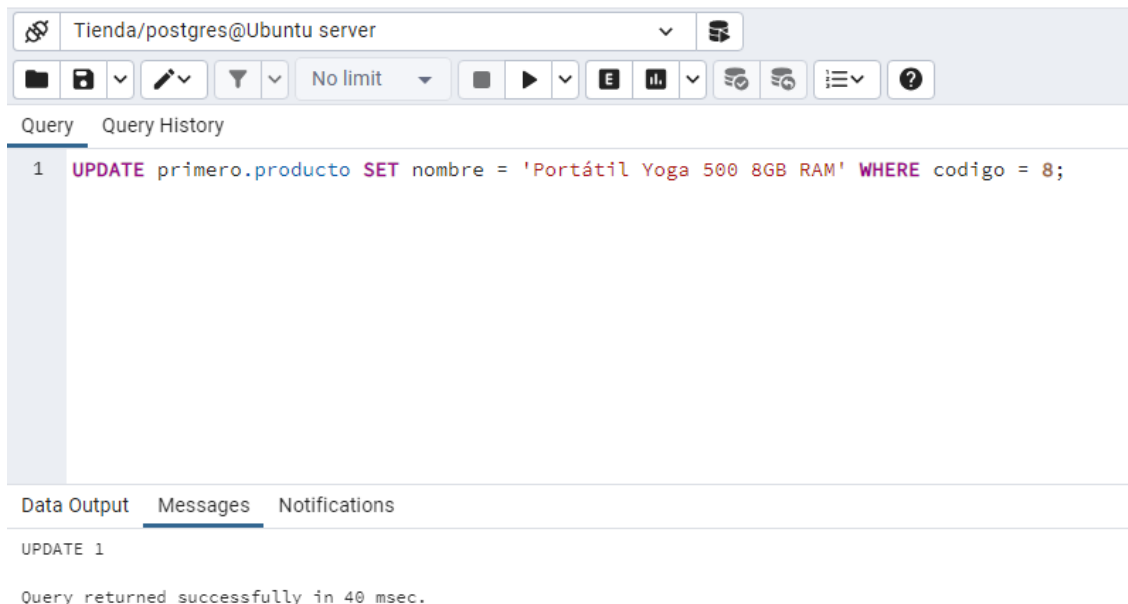
The screenshot shows a PostgreSQL query editor interface. At the top, there's a connection bar with 'Tienda/postgres@Ubuntu server'. Below it is a toolbar with various icons. The main area is divided into 'Query' and 'Query History' tabs. The 'Query' tab contains the following SQL statement:

```
1 INSERT INTO primero.producto (codigo, nombre, precio, codigo_fabricante)
2 VALUES (99, 'Altavoces', 70, 2);
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the result of the query:

```
INSERT 0 1
Query returned successfully in 36 msec.
```

r) Actualiza el nombre del producto 8 a "Portátil Yoga 500 8GB RAM".



The screenshot shows the same PostgreSQL query editor interface. The 'Query' tab contains the following SQL statement:

```
1 UPDATE primero.producto SET nombre = 'Portátil Yoga 500 8GB RAM' WHERE codigo = 8;
```

Below the query editor, the 'Messages' tab is active, showing the result of the query:

```
UPDATE 1
Query returned successfully in 40 msec.
```


s) Aplicar un 10% de descuento en todos los productos.

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesprimero.fabric...

Tienda/postgres@Ubuntu server

No limit

QueryQuery History

1 UPDATE primero.producto SET precio = precio * 0.9;

Data OutputMessagesNotifications

UPDATE 12

Query returned successfully in 39 msec.

	codigo [PK] integer	nombre character varying (100)	precio double precision	codigo_fabricante integer
1	6	Monitor 24 LED Full HD	181.8	5
2	8	Portátil Yoga 500 8GB RAM	53.1	7
3	11	Impresora HP Laserjet Pro M26nw	162	7
4	29	Portátil ideapad 320	399.6	6
5	38	Portátil Yoga 520	503.1	6
6	47	Monitor 27 LED Full HD	220.5	5
7	53	Disco SSD 1TB	135	3
8	64	GeForce GTX 1050Ti	166.5	4
9	75	GeForce GTX 1080 Xtreme	679.5	2
10	82	Memoria RAM DDR4 8GB	108	2
11	91	Disco duro SATA3 1TB	77.4	1
12	99	Altavoces	63	2

t) Aplicar un 10% de descuento a todos los productos con un precio mayor o igual a 120.

QueryQuery History

1 UPDATE primero.producto

2 SET precio = precio * 0.9

3 WHERE precio >= 120;

Data OutputMessagesNotifications

UPDATE 8

Query returned successfully in 41 msec.

QueryQuery History

1 SELECT * FROM primero.producto

2 ORDER BY codigo ASC

Data OutputMessagesNotifications

	codigo [PK] integer	nombre character varying (100)	precio double precision	codigo_fabricante integer
1	6	Monitor 24 LED Full HD	163.62	5
2	8	Portátil Yoga 500 8GB RAM	53.1	7
3	11	Impresora HP Laserjet Pro M26nw	145.8	7
4	29	Portátil ideapad 320	359.64000000000004	6
5	38	Portátil Yoga 520	452.79	6
6	47	Monitor 27 LED Full HD	198.45000000000002	5
7	53	Disco SSD 1TB	121.5	3
8	64	GeForce GTX 1050Ti	149.85	4
9	75	GeForce GTX 1080 Xtreme	611.55000000000001	2
10	82	Memoria RAM DDR4 8GB	108	2
11	91	Disco duro SATA3 1TB	77.4	1
12	99	Altavoces	63	2