

EXAMEN BASE DE DATOS FEBRERO

ENDIKA PEÑA ALONSO

Ejercicio 1:

He creado el usuario con una contraseña débil y luego se la he cambiado para que sea algo más fuerte.

Consulta:

-- USER SQL

```
CREATE USER "C##TIENDA" IDENTIFIED BY "12345678";
```

```
ALTER USER "C##TIENDA" IDENTIFIED BY "1a2b3c4D";
```

-- QUOTAS

```
ALTER USER "C##TIENDA" QUOTA UNLIMITED ON "USERS";
```

-- ROLES

```
GRANT "CONNECT" TO "C##TIENDA" ;
```

-- SYSTEM PRIVILEGES

```
GRANT CREATE ANY TABLE TO "C##TIENDA" ;
```


Ejercicio 2:

Creación de la estructura de la base de datos.

Hoja de Trabajo

Generador de Consultas

CREATE TABLE fabricante (
id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
nombre VARCHAR2(100) NOT NULL
);

CREATE TABLE producto (
id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
nombre VARCHAR2(100) NOT NULL,
precio NUMBER(10,2) NOT NULL,
id_fabricante NUMBER NOT NULL,
CONSTRAINT fk_producto_fabricante FOREIGN KEY (id_fabricante) REFERENCES fabricante(id)
);

INSERT INTO fabricante (nombre) VALUES ('Asus');
INSERT INTO fabricante (nombre) VALUES ('Lenovo');
INSERT INTO fabricante (nombre) VALUES ('Hewlett-Packard');
INSERT INTO fabricante (nombre) VALUES ('Samsung');
INSERT INTO fabricante (nombre) VALUES ('Seagate');
INSERT INTO fabricante (nombre) VALUES ('Crucial');
INSERT INTO fabricante (nombre) VALUES ('Gigabyte');
INSERT INTO fabricante (nombre) VALUES ('Huawei');
INSERT INTO fabricante (nombre) VALUES ('Xiaomi');

INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Disco duro SATA3 1TB', 86.99, 5);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Memoria RAM DDR4 8GB', 120, 6);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Disco SSD 1 TB', 150.99, 4);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('GeForce GTX 1050Ti', 185, 7);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('GeForce GTX 1080 Xtreme', 755, 6);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Monitor 24 LED Full HD', 202, 1);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Monitor 27 LED Full HD', 245.99, 1);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Portátil Yoga 520', 559, 2);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Portátil Ideapad 320', 444, 2);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Impresora HP Deskjet 3720', 59.99, 3);
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Impresora HP Laserjet Pro M26nw', 180, 3);

Salida de Script x

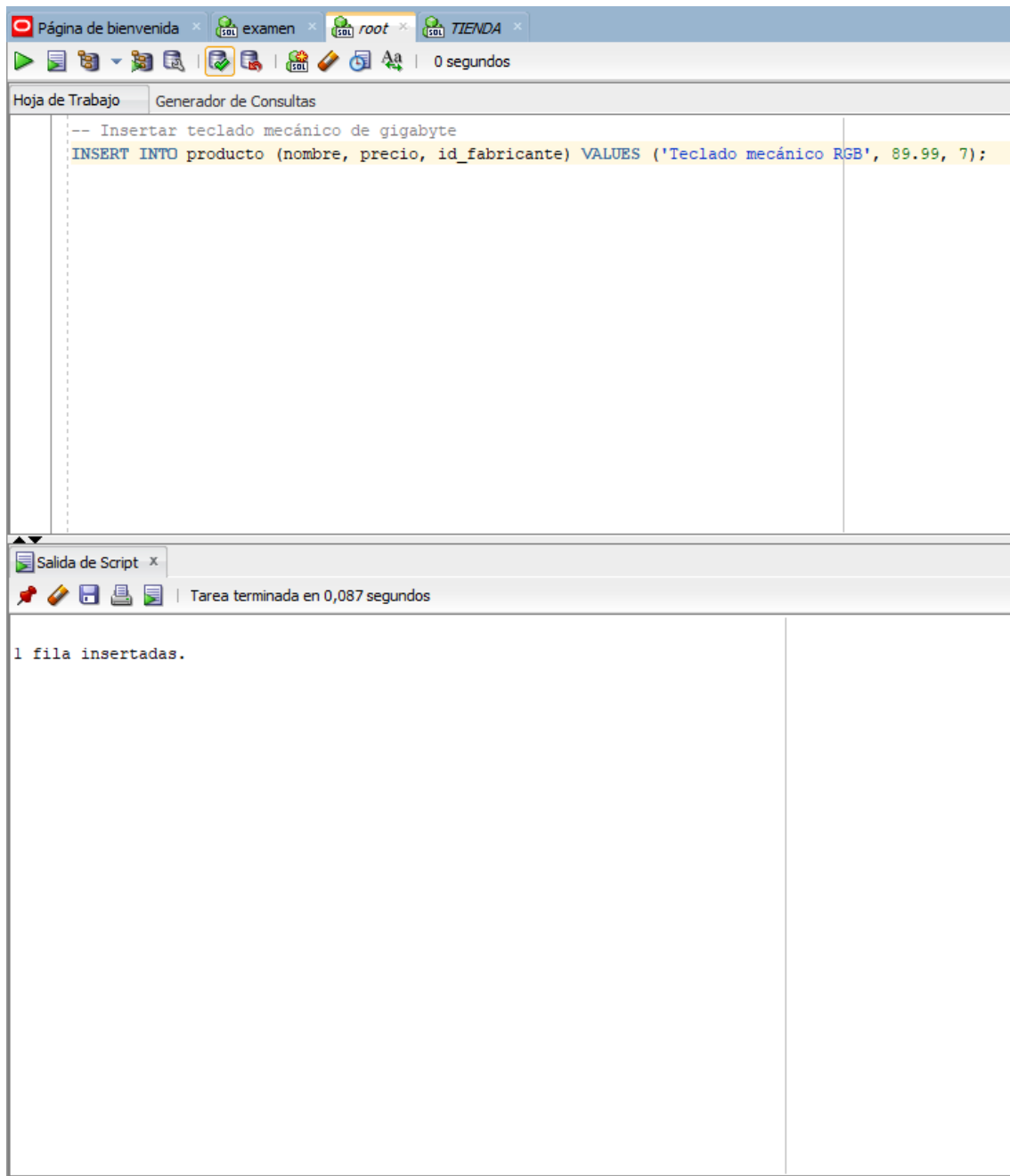
Tarea terminada en 0,869 segundos

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

A)



B)

Precio original según lo introducido por el script

Consulta:

```
SELECT p.*, f.nombre FROM PRODUCTO p JOIN FABRICANTE f ON  
p.ID_FABRICANTE = f.ID WHERE f.NOMBRE = 'Samsung';
```

150.99

Actualización de los precios

The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains the following SQL code:

```
-- Bajar precios de samsung
UPDATE PRODUCTO SET precio = (precio - (precio * 0.10)) WHERE id_fabricante = (SELECT f.ID FROM PRODUCTO p JOIN FABRICANTE f ON p.ID_FABRICANTE = f.ID WHERE f.NOMBRE = 'Samsung');
COMMIT;
```

The results pane shows the output of the query:

```
1 fila actualizadas.
Confirmación terminada.
1 fila actualizadas.
1 fila actualizadas.
1 fila actualizadas.
Confirmación terminada.
```

Comprobación de que ha sido efectivo el cambio.

The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains the following SQL code:

```
SELECT p.*, f.nombre FROM PRODUCTO p JOIN FABRICANTE f ON p.ID_FABRICANTE = f.ID WHERE f.NOMBRE = 'Samsung';
```

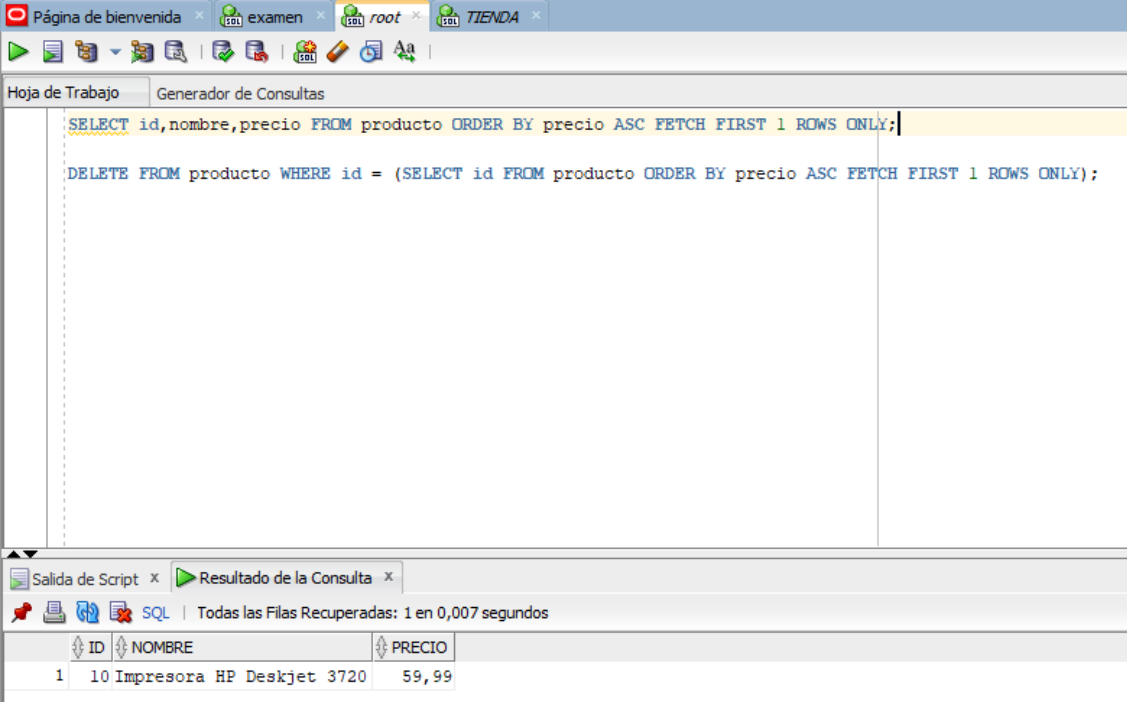
The results pane shows the output of the query:

ID	NOMBRE	PRECIO	ID_FABRICANTE	NOMBRE_1
1	3 Disco SSD 1 TB	135,89	4	Samsung

C) Eliminar producto más barato

Buscando primero el producto más barato

```
SELECT id,nombre,precio FROM producto ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY;
```



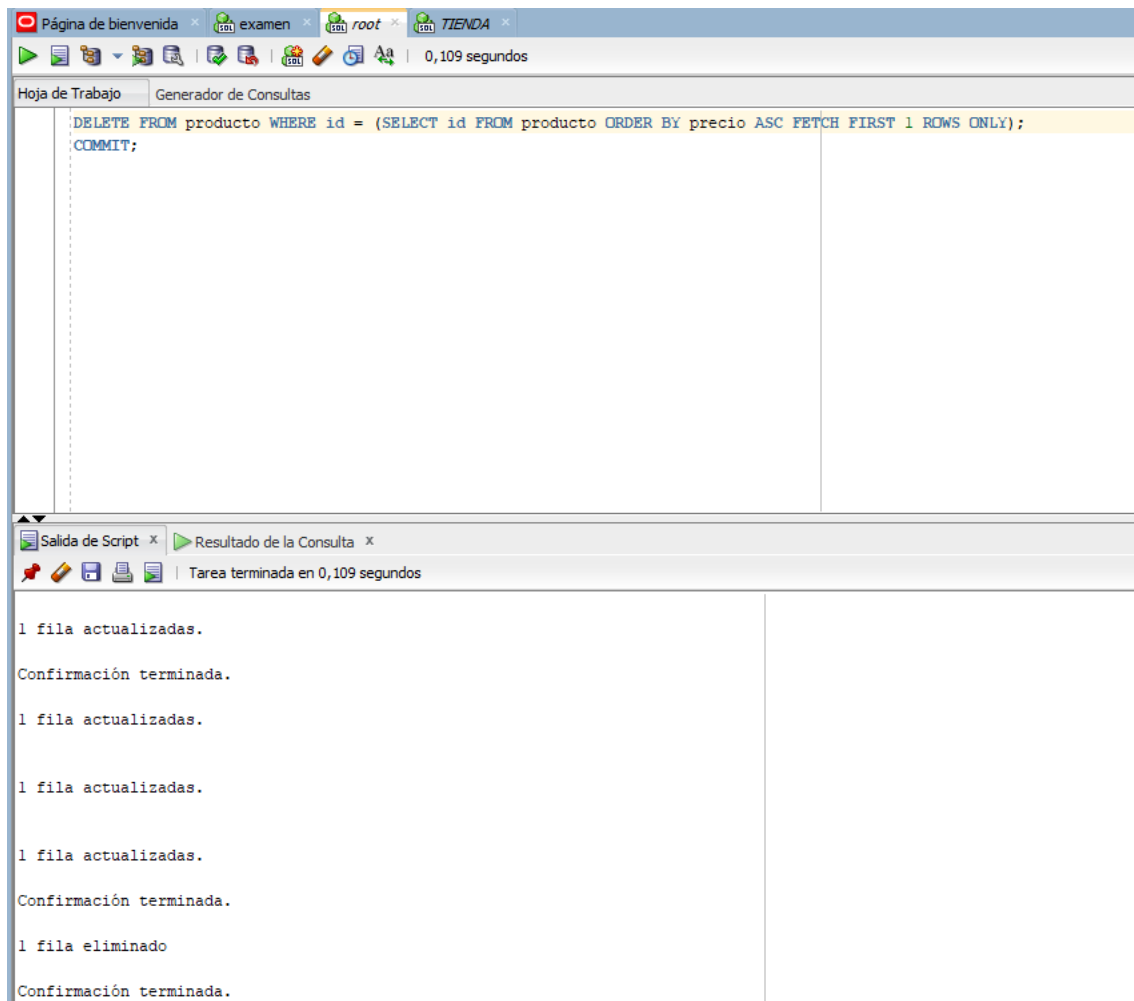
The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains two SQL statements: a SELECT statement to find the cheapest product and a DELETE statement to remove it. The results pane shows the output of the first query, which is a single row representing the cheapest product.

```
SELECT id,nombre,precio FROM producto ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY;
DELETE FROM producto WHERE id = (SELECT id FROM producto ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY);
```

ID	NOMBRE	PRECIO
1	10 Impresora HP Deskjet 3720	59,99

Eliminar el producto más barato

```
DELETE FROM producto WHERE id = (SELECT id FROM producto ORDER BY precio
ASC FETCH FIRST 1 ROWS ONLY);
```



Verificamos que el producto ya más barato ya no es el mismo.

SELECT id,nombre,precio FROM producto ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY;

Página de bienvenida × examen × root × TIENDA ×

Hoja de Trabajo Generador de Consultas

```

SELECT id,nombre,precio FROM producto ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY;
  
```

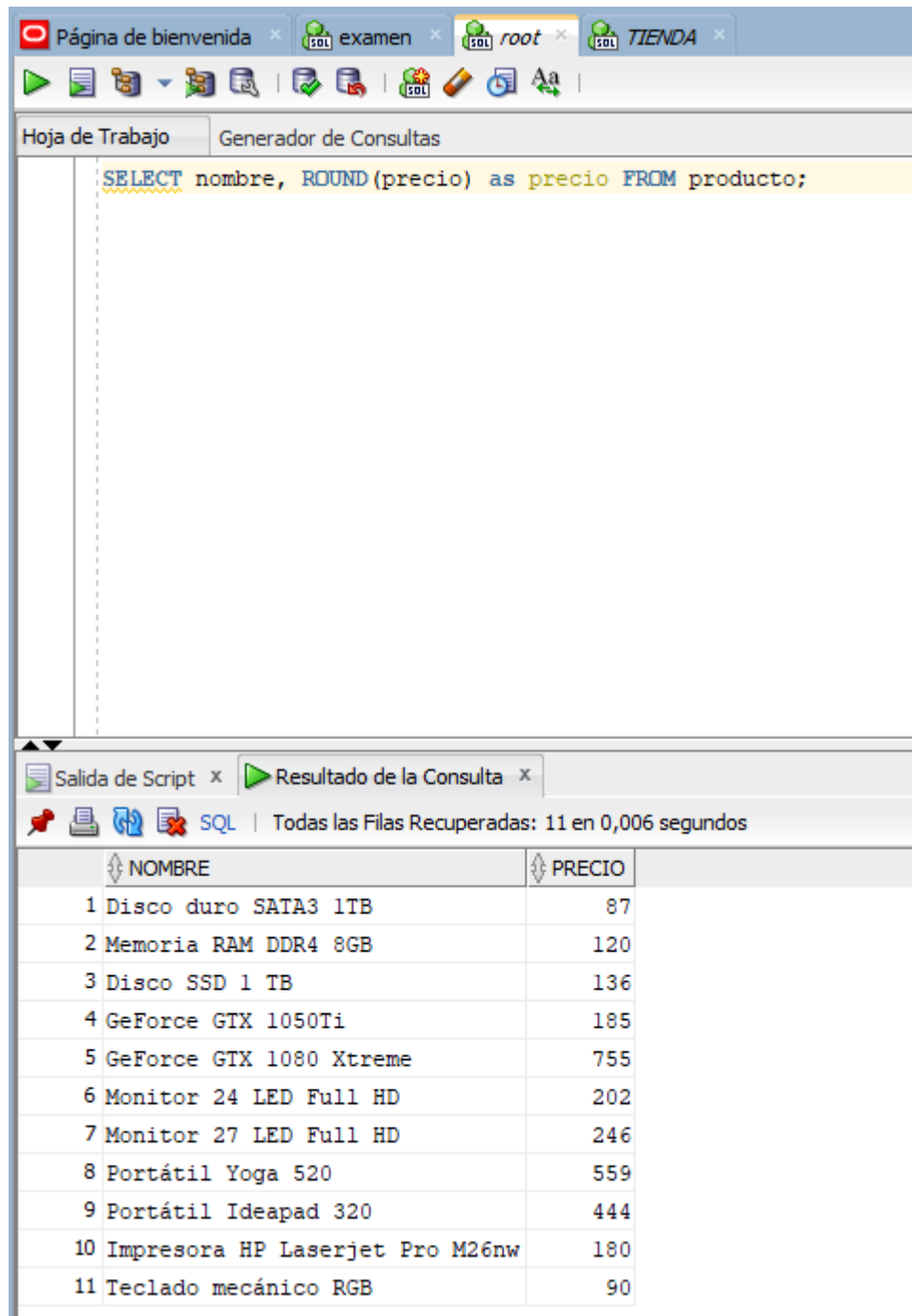
Salida de Script × Resultado de la Consulta ×

SQL | Todas las Filas Recuperadas: 1 en 0,001 segundos

ID	NOMBRE	PRECIO
1	1 Disco duro SATA3 1TB	86,99

Ejercicio 3:

1.



The screenshot shows a SQL IDE interface with multiple tabs at the top: 'Página de bienvenida', 'examen', 'root', and 'TIENDA'. The 'root' tab is active, displaying a 'Hoja de Trabajo' (Worksheet) with a SQL query: `SELECT nombre, ROUND(precio) as precio FROM producto;`. Below the query editor, the 'Resultado de la Consulta' (Query Result) tab is active, showing the results of the query. The results are displayed in a table with two columns: 'NOMBRE' and 'PRECIO'. The table contains 11 rows of data, listing various products and their prices, rounded to the nearest integer.

	NOMBRE	PRECIO
1	Disco duro SATA3 1TB	87
2	Memoria RAM DDR4 8GB	120
3	Disco SSD 1 TB	136
4	GeForce GTX 1050Ti	185
5	GeForce GTX 1080 Xtreme	755
6	Monitor 24 LED Full HD	202
7	Monitor 27 LED Full HD	246
8	Portátil Yoga 520	559
9	Portátil Ideapad 320	444
10	Impresora HP Laserjet Pro M26nw	180
11	Teclado mecánico RGB	90

2.

The screenshot shows a SQL query editor interface. The top toolbar includes icons for running queries, saving, and other database functions. The main editor area contains the following SQL query:

```
SELECT nombre FROM producto ORDER BY precio ASC FETCH FIRST 1 ROWS ONLY;
```

Below the editor, the 'Resultado de la Consulta' (Query Result) window is open, displaying the results of the query. The results are shown in a table with one column, 'NOMBRE', and one row containing the text '1 Disco duro SATA3 1TB'.

NOMBRE
1 Disco duro SATA3 1TB

3.

The screenshot shows a SQL IDE interface with the following components:

- Top Bar:** Contains tabs for 'Página de bienvenida', 'examen', 'root', and 'TIENDA'. Below the tabs is a toolbar with various icons for file operations and execution.
- Hoja de Trabajo (Worksheet):** The main area for writing SQL queries. It contains the following query:

```
SELECT COUNT(DISTINCT(nombre)) as numero_fabricantes FROM fabricante;
```
- Bottom Panel:** Divided into two sections:
 - Salida de Script:** A tab for viewing the output of the script.
 - Resultado de la Consulta:** A tab showing the result of the query. It displays a table with one column, 'NUMERO_FABRICANTES', and one row with the value '9'. The status bar indicates 'Todas las Filas Recuperadas: 1 en 0,009 segundos'.

	NUMERO_FABRICANTES
1	9

4.

The screenshot shows a SQL IDE interface with the following components:

- Top Bar:** Contains tabs for 'Página de bienvenida', 'examen', 'root', and 'TIENDA'. Below the tabs is a toolbar with various icons for file operations and execution.
- Editor Area:** The 'Generador de Consultas' tab is active, displaying the SQL query: `SELECT nombre FROM producto WHERE nombre LIKE '%Portátil%';`
- Bottom Panel:** Contains two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, showing the query results in a table format.

The query results table is as follows:

	NOMBRE
1	Portátil Yoga 520
2	Portátil Ideapad 320

5.

The screenshot shows a SQL IDE with the following components:

- Windows:**
 - Página de bienvenida
 - examen
 - root
 - TIENDA
- Hoja de Trabajo (Worksheet):**
 - Generador de Consultas
 - Query: `SELECT DISTINCT(f.nombre) FROM fabricante f RIGHT JOIN producto p ON f.id = p.id fabricante;`
- Salida de Script (Script Output):**
 - Resultados: Todas las Filas Recuperadas: 7 en 0,009 segundos
- Resultado de la Consulta (Query Results):**

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

6.

The screenshot shows a SQL IDE with the following components:

- Windows:**
 - Página de bienvenida
 - examen
 - root
 - TIENDA
- Hoja de Trabajo (Worksheet):**
 - Generador de Consultas
 - Query: `SELECT MIN(precio) as precio_minimo, MAX(precio) as precio_maximo, AVG(precio) as promedio, COUNT(p.id) as total_productos FROM producto p WHERE id_fabricante = (SELECT id FROM fabricante WHERE nombre = 'Crucial');`
- Salida de Script (Script Output):**
 - Resultados: Todas las Filas Recuperadas: 1 en 0,012 segundos
- Resultado de la Consulta (Query Results):**

	PRECIO_MINIMO	PRECIO_MAXIMO	PROMEDIO	TOTAL_PRODUCTOS
1	120	755	437,5	2

7. No me acuerdo de esta

8.

The screenshot shows a web-based SQL interface. The top navigation bar includes tabs for 'Página de bienvenida', 'examen', 'root', and 'TIENDA'. Below this is a toolbar with various icons. The main workspace is divided into two panes: 'Hoja de Trabajo' (left) and 'Generador de Consultas' (right). The 'Generador de Consultas' pane contains the following SQL query:

```
SELECT DISTINCT( f.nombre) FROM fabricante f LEFT JOIN producto p ON f.id = p.id_fabricante(+);
```

Below the query editor, there is a status bar showing 'Salida de Script' and 'Resultado de la Consulta'. Below this, a message indicates 'Todas las Filas Recuperadas: 9 en 0,009 segundos'. The results are displayed in a table with the following data:

	NOMBRE
1	Seagate
2	Crucial
3	Samsung
4	Gigabyte
5	Asus
6	Lenovo
7	Hewlett-Packard
8	Huawei
9	Xiaomi

9.

The screenshot shows a database management interface with a query editor and a results pane. The query editor contains the following SQL statement:

```
SELECT nombre FROM fabricante WHERE id IN (SELECT id_fabricante FROM producto GROUP BY id_fabricante HAVING count(*) >= 2);
```

The results pane displays the output of the query, showing a table with one column named 'NOMBRE' and four rows of data:

NOMBRE
1 Asus
2 Lenovo
3 Crucial
4 Gigabyte

The interface also shows a status bar indicating that all rows were recovered in 0.019 seconds.

10.

The screenshot shows a web-based SQL query tool interface. The top navigation bar includes tabs for 'Página de bienvenida', 'examen', 'root', and 'TIENDA'. Below the navigation bar is a toolbar with various icons. The main workspace is divided into two panes: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Generator). The 'Generador de Consultas' pane contains the following SQL query:

```
SELECT DISTINCT(f.nombre) FROM fabricante f JOIN producto p ON f.id = p.id_fabricante WHERE p.nombre LIKE '%$Monitor$';
```

Below the query editor, there are two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, displaying the results of the query. The results are shown in a table with one column, 'NOMBRE', and one row, '1 Asus'.

NOMBRE
1 Asus

Ejercicio 4:

Sentencias SQL para la creación de la base de datos

```
CREATE TABLE clientes (
```

```
    id_cliente NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
```

```
    nombre VARCHAR2(100) NOT NULL,
```

```
    correo VARCHAR2(150) UNIQUE NOT NULL,
```

```
    fecha_registro DATE DEFAULT SYSDATE
```

```
);
```

```
CREATE TABLE productos (
```

```
    id_producto NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
```

```
    nombre VARCHAR2(100) NOT NULL,
```

```
    precio NUMBER(10,2) NOT NULL ,
```

```
    stock NUMBER DEFAULT 0 NOT NULL,
```

```
    CONSTRAINT check_precio CHECK (precio > 0),
```

```
    CONSTRAINT check_stock CHECK (stock >= 0)
```

```
);
```

```
CREATE TABLE pedidos (
```

```
    id_pedido NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
```

```
    id_cliente NUMBER(5),
```

```
    fecha_pedido DATE DEFAULT SYSDATE NOT NULL,
```

```
    CONSTRAINT fk_pedido_cliente FOREIGN KEY (id_cliente) REFERENCES  
    clientes(id_cliente) ON DELETE CASCADE
```

```
);
```

```
CREATE TABLE detalle_pedido (
```

```
    id_detalle NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
```

```

id_pedido NUMBER(5),

id_producto NUMBER(5),

cantidad NUMBER(5) NOT NULL,

subtotal NUMBER(7,2) NOT NULL,

CONSTRAINT check_cantidad CHECK (cantidad >= 0),

CONSTRAINT check_subtotal CHECK (subtotal > 0),

CONSTRAINT fk_detalle_pedido_pedido FOREIGN KEY (id_pedido) REFERENCES
pedidos(id_pedido) ON DELETE CASCADE,

CONSTRAINT fk_detalle_pedido_producto FOREIGN KEY (id_producto)
REFERENCES productos(id_producto)

);

```

Hoja de Trabajo	Generador de Consultas
<pre> CREATE TABLE clientes (id_cliente NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, nombre VARCHAR2(100) NOT NULL, correo VARCHAR2(150) UNIQUE NOT NULL, fecha_registro DATE DEFAULT SYSDATE); CREATE TABLE productos (id_producto NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, nombre VARCHAR2(100) NOT NULL, precio NUMBER(10,2) NOT NULL, stock NUMBER DEFAULT 0 NOT NULL, CONSTRAINT check_precio CHECK (precio > 0), CONSTRAINT check_stock CHECK (stock >= 0)); CREATE TABLE pedidos (id_pedido NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, id_cliente NUMBER(5), fecha_pedido DATE DEFAULT SYSDATE NOT NULL, CONSTRAINT fk_pedido_cliente FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente) ON DELETE CASCADE); CREATE TABLE detalle_pedido (id_detalle NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, id_pedido NUMBER(5), id_producto NUMBER(5), cantidad NUMBER(5) NOT NULL, subtotal NUMBER(7,2) NOT NULL, CONSTRAINT check_cantidad CHECK (cantidad >= 0), CONSTRAINT check_subtotal CHECK (subtotal > 0), CONSTRAINT fk_detalle_pedido_pedido FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido) ON DELETE CASCADE, CONSTRAINT fk_detalle_pedido_producto FOREIGN KEY (id_producto) REFERENCES productos(id_producto)); </pre>	

Table CLIENTES creado.

Table PRODUCTOS creado.

Table PEDIDOS creado.

Table DETALLE_PEDIDO creado.