

(P) Diseño y programación de una base de datos

Producto 2. Consultas simples y modificación simple de datos

COMPONENTES DEL GRUPO C2047 LAC

M^aCelia García Molina

Liliana Díaz Ibáñez

Agnès García Mateo

CONSULTORA

Rita de la Torre Chirivella

Descripción	2
Objetivos	2
4. Insertar los registros de PRODUCTOS y COMPRAS que el consultor facilitará vía tablón. Para ello, deberéis repartir la información de la tabla de productos, en las diferentes tablas de la Base de Datos según corresponda y, posteriormente, insertar los registros de compras de la tabla .csv.	2
5. Realizar las siguientes consultas y anotar la sentencia SQL y su salida en un documento:	5
a. Indicar el número de ingredientes registrados en la Base de Datos.	5
b. Listar todos los productos ordenados por la categoría a la cual pertenecen.	5
c. Mostrar en el listado la categoría, código, nombre y alerta_stock.	7
d. Hacer una relación de los proveedores de Verduras y Hortalizas.	8
e. Listar las compras realizadas durante el mes en curso.	9
f. Indicar el número de platos de ensalada registrados.	10
g. Mostrar aquellos proveedores cuyo nombre comience por la letra M.	10
h. Mostrar los productos que contengan alérgenos.	11
i. Reportar el total de compras en € realizadas a cada proveedor. La consulta deberá tener los siguientes campos: Empresa, Número de productos y Total Euros.	12
j. Indicar aquellos productos que han cambiado de precio.	12
k. Indicar la cantidad de platos ofertados por tipo de plato.	13
K. Indicar la cantidad de platos ofertados por tipo de plato.	13
L. Indicar cuántos platos registrados NO se están ofertando en Menú.	14
m. Contar el total de donaciones por producto.	14
n. Formular una consulta que tenga una agrupación con Group By.	15
o. Formular una consulta que tenga una agrupación con Group By y un filtraje con Having.	15
6. Realizar las siguientes modificaciones en los datos insertados y anotar la sentencia SQL y su salida en un documento:	16
a. Actualizar el campo Porcentaje_Merma con la división de la Cantidad_Neta entre la Cantidad_Bruta en la tabla INGREDIENTES_PLATO.	16
Código:	17
UPDATE INGREDIENTE_PLATO SET Porcentaje_Merma = (cantNetalnPI/cantBrutalnPI)*100;	17
b. Cambiar la fecha de caducidad de los productos, sumándole 7 días.	17
c. Actualizar el PVP de los platos, aumentando en un 2%.	18
d. Eliminar aquellas CATEGORÍAS que no tengan PROVEEDORES.	18
e. Borrar los productos elaborados que nunca se hayan vendido. Para ello deberás revisar la tabla COMANDA_ELABORADOS.	18

Descripción

En este producto vamos a realizar consultas y modificaciones en nuestra base de datos, utilizando el lenguaje SQL (DML), empezando con consultas y modificaciones sencillas para asentar los conocimientos.

Objetivos

El objetivo principal del producto es:

- Obtener consultas sencillas de una base de datos, con filtraje, agrupación, ordenación y funciones del SQL.

4. Insertar los registros de PRODUCTOS y COMPRAS que el consultor facilitará vía tablón. Para ello, deberéis repartir la información de la tabla de productos, en las diferentes tablas de la Base de Datos según corresponda y, posteriormente, insertar los registros de compras de la tabla .csv.

PRODUCTOS

The screenshot displays a database management interface. On the left, a 'SCHEMAS' panel shows a tree view of the database structure, including tables like ALERGENO, ALERGENOS_PRODUCTO, CATEGORIA, COMANDA, COMANDA_ELABORADOS, COMANDA_PLATO, COMPRAS_PRODUCTO, DETALLE_DONACION, DONACIONES, and ELABORADOS. The 'ELABORADOS' table is selected, showing its columns: idProdElab (PK), pvp, and iva. The main area shows a list of SQL insert statements for the PRODUCTO and INGREDIENTE tables. The statements are numbered 1 through 22. The PRODUCTO statements use the columns (idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) and the INGREDIENTE statements use (idProdIngr, conservIngr). The values are provided for each statement.

```
1 • insert into PRODUCTO (idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
2 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('2001', 'Guardar limpio en el re...)
```

```
3 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
4 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1101', 'Guardar en un lugar seci...)
```

```
5 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
6 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1102', 'Guardar en un lugar fre:...
```

```
7 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
8 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1103', 'Guardar en un lugar fre:...
```

```
9 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
10 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1104', 'Guardar en un lugar fre:...
```

```
11 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
12 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1105', 'Guardar en un lugar fre:...
```

```
13 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
14 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1106', 'Guardar en un lugar fre:...
```

```
15 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
16 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1107', 'Guardar en lugar fresco...
```

```
17 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
18 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1002', 'Guardar en un lugar osci...)
```

```
19 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
20 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('6005', 'Guardar en una bolsa se:...
```

```
21 • insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values (...)
```

```
22 • insert into INGREDIENTE (idProdIngr, conservIngr) values ('1607', 'Guardar en un lugar fre:...
```

Código:

```
insert into PRODUCTO (idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('2001', 'Pechuga marinada', 'Kilogramos', 30, 2, 'INGREDIENTE') ;

insert into INGREDIENTE (idProdIngr, conservIngr) values ('2001', 'Guardar limpio en el refrigerador en bolsas de congelación');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1101', 'Sal de Himalaya', 'Kilogramos', 2, 11, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1101', 'Guardar en un lugar seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1102', 'Pimienta dulce', 'Kilogramos', 5, 11, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1102', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1103', 'Hierbas provenzales', 'Kilogramos', 5, 11, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1103', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1104', 'Orégano', 'Kilogramos', 5, '11', 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1104', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1105', 'Romero', 'Kilogramos', 2, '11', 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1105', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1106', 'Laurel en hojas', 'Kilogramos', 5, 11, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1106', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1107', 'Ajo', 'Kilogramos', 10, 11, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1107', 'Guardar en lugar fresco, seco y alejado de la luz');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1002', 'Aceite de maíz', 'Litro', 20, 1, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1002', 'Guardar en un lugar oscuro de la despensa');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('6005', 'Malla de Limón (500gr)', 'Unidades', 36, 6, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('6005', 'Guardar en una bolsa sellada en el cajón de frigorífico');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1607', 'Harina de trigo (1 kg)', 'Unidades', 240, 16, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1607', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1701', 'Salsa Worcester', 'Unidades', 360, 17, 'INGREDIENTE');

insert into INGREDIENTE (idProdIngr, conservIngr) values ('1701', 'Guardar en un lugar fresco y seco');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1501', 'Copa de nata con café', 'Unidades', 70, 15, 'ELABORADO');

insert into ELABORADOS (idProdElab) values ('1501');

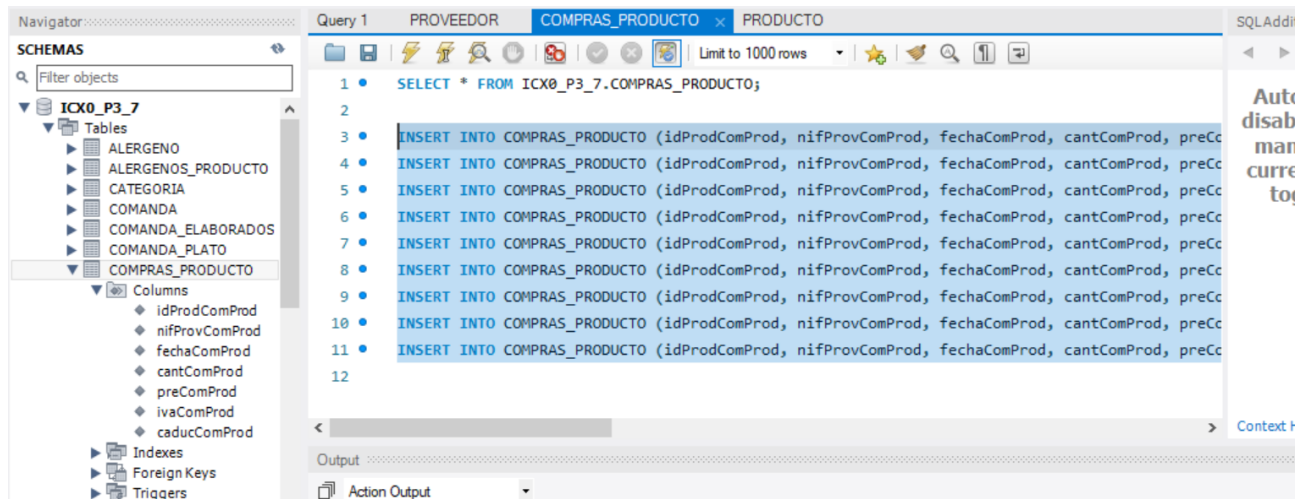
insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('1502', 'Flan de huevo (100g)', 'Unidades', 70, 15, 'ELABORADO');

insert into ELABORADOS (idProdElab) values ('1502');

insert into PRODUCTO(idProd, nomProd, uniProd, alertaStockProd, idCatProd, Tipo) values ('4001', 'Cápsulas de café cappuccino (16 cápsulas)', 'Unidades', 20, 4, 'ELABORADO');

insert into ELABORADOS (idProdElab) values ('4001');
```

COMPRAS



Código:

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd, caducComProd) VALUES ('2001','B23373624','2021/08/01',60,3.24,10.00,'2021/08/16');
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd, caducComProd) VALUES ('2001','A50090349','2021/08/10',25,2.99,10.00,'2021/08/20');
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd, caducComProd) VALUES ('1106','A28647451','2021/09/02',5,8.75,10.00,'2023/09/02');
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd, caducComProd) VALUES ('1105','B90307034','2021/09/17',2,10.91,10.00,'2020/04/14');
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd, caducComProd) VALUES ('1107','B87867834','2021/09/18',5,6.50,0.00,'2019/10/30');
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd) VALUES ('4001','A28647451','2021/09/07',480,4.25,10.00);
```

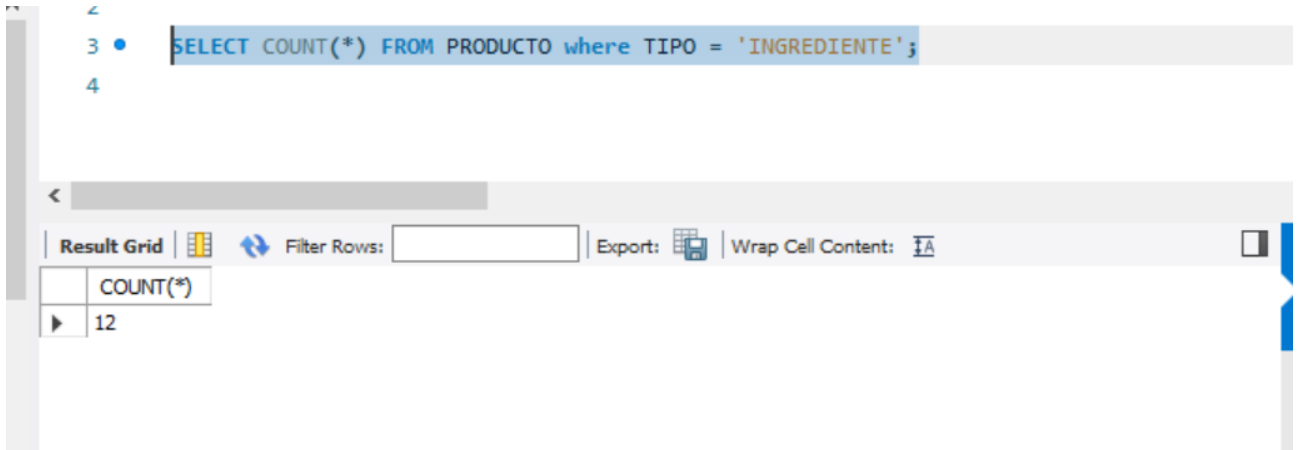
```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd) VALUES ('1607','A28647451','2021/09/07',700,0.52,10.00);
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd) VALUES ('1501','A28647451','2021/09/07',100,0.32,10.00);
```

```
INSERT INTO COMPRAS_PRODUCTO (idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd) VALUES ('1502','A58058868','2021/09/05',100,1.60,10.00);
```

5. Realizar las siguientes consultas y anotar la sentencia SQL y su salida en un documento:

a. Indicar el número de ingredientes registrados en la Base de Datos.



Código:

```
SELECT COUNT(*) FROM PRODUCTO where TIPO = 'INGREDIENTE';
```

b. Listar todos los productos ordenados por la categoría a la cual pertenecen.

Usamos **ORDER BY** para ordenar los productos pero para ordenar por los nombres de categorías debemos hacer un JOIN entre producto y categoría

Limit to 1000 rows

```

1 • SELECT PRODUCTO.nomProd, CATEGORIA.nomCat
2 FROM PRODUCTO
3 INNER JOIN CATEGORIA ON CATEGORIA.idCat = PRODUCTO.idCatProd
4 Order by CATEGORIA.nomCat
5

```

Result Grid

nomProd	nomCat
Aceite de maíz	Aceites y vinagres
Cápsulas de café cappuccino (16 cápsulas)	Cafés
Pechuga marinada	Carnes y Aves
Sal de Himalaya	Especies
Pimienta dulce	Especies
Hierbas provenzales	Especies
Orégano	Especies
Romero	Especies
Laurel en hojas	Especies
Ajo	Especies
Malla de Limón (500gr)	Fruta y Verdura
Cebolla	Fruta y Verdura
Tomates	Fruta y Verdura
Ajo	Fruta y Verdura
Harina de trigo (1 kg)	Harinas
Pan Integral	Pan y Bollería
Gambas	Pescados Y Maris...
Flan de huevo (100q)	Postres

Result 4 x

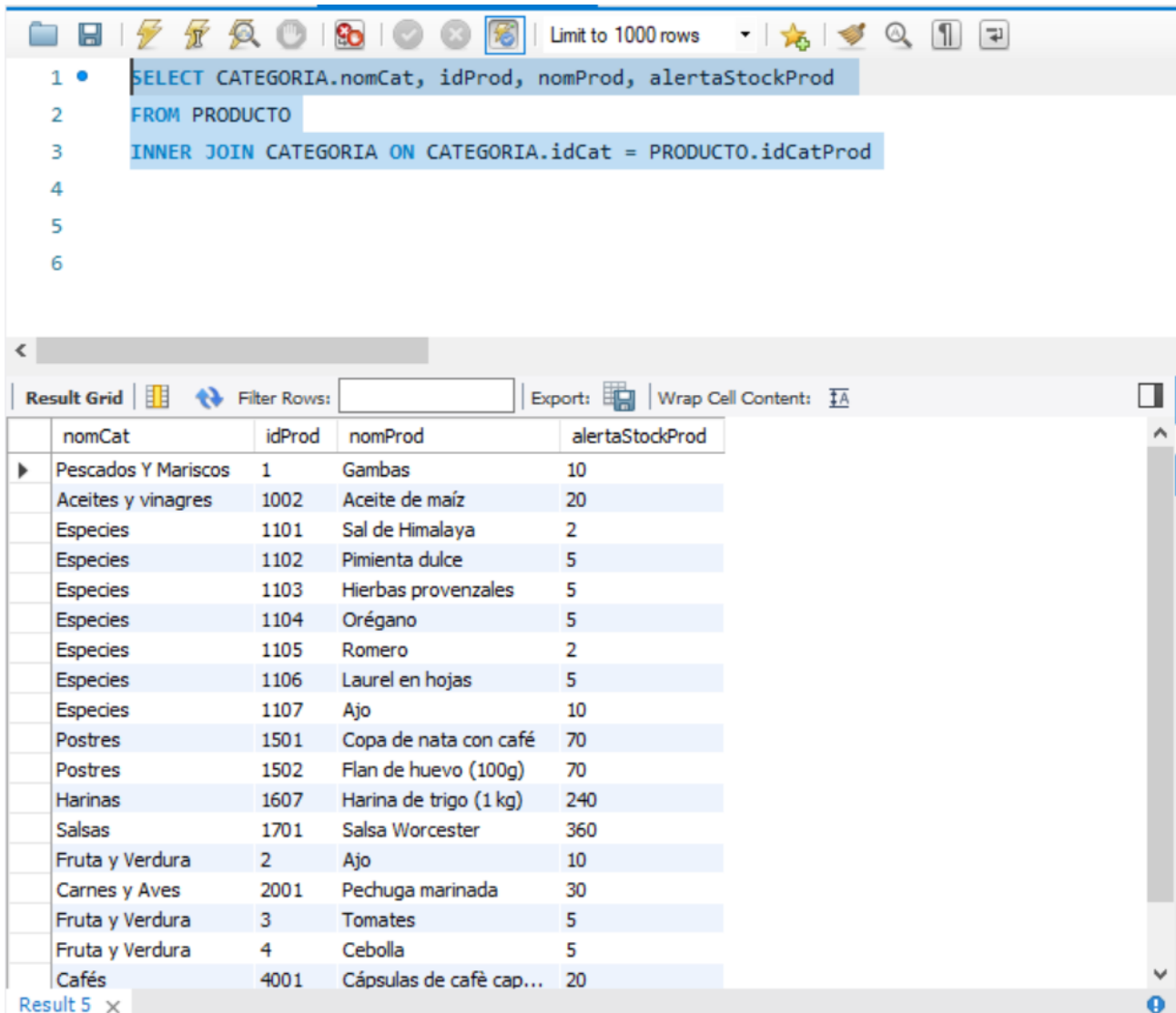
Código:

```

SELECT PRODUCTO.nomProd, CATEGORIA.nomCat
FROM PRODUCTO
INNER JOIN CATEGORIA ON CATEGORIA.idCat = PRODUCTO.idCatProd
Order by CATEGORIA.nomCat

```

c. Mostrar en el listado la categoría, código, nombre y alerta_stock.



The screenshot shows a database query editor with a toolbar at the top. The SQL query is as follows:

```
1 SELECT CATEGORIA.nomCat, idProd, nomProd, alertaStockProd
2 FROM PRODUCTO
3 INNER JOIN CATEGORIA ON CATEGORIA.idCat = PRODUCTO.idCatProd
4
5
6
```

Below the query editor, the results are displayed in a grid. The grid has columns: nomCat, idProd, nomProd, and alertaStockProd. The results are as follows:

nomCat	idProd	nomProd	alertaStockProd
Pescados Y Mariscos	1	Gambas	10
Aceites y vinagres	1002	Aceite de maíz	20
Especies	1101	Sal de Himalaya	2
Especies	1102	Pimienta dulce	5
Especies	1103	Hierbas provenzales	5
Especies	1104	Orégano	5
Especies	1105	Romero	2
Especies	1106	Laurel en hojas	5
Especies	1107	Ajo	10
Postres	1501	Copa de nata con café	70
Postres	1502	Flan de huevo (100g)	70
Harinas	1607	Harina de trigo (1 kg)	240
Salsas	1701	Salsa Worcester	360
Fruta y Verdura	2	Ajo	10
Carnes y Aves	2001	Pechuga marinada	30
Fruta y Verdura	3	Tomates	5
Fruta y Verdura	4	Cebolla	5
Cafés	4001	Cápsulas de café cap...	20

Código:

```
SELECT CATEGORIA.nomCat, idProd, nomProd, alertaStockProd
FROM PRODUCTO
INNER JOIN CATEGORIA ON CATEGORIA.idCat = PRODUCTO.idCatProd
```


d. Hacer una relación de los proveedores de Verduras y Hortalizas.

The screenshot shows a database query editor with the following SQL query:

```
1 select PROVEEDOR.*
2 FROM PROVEEDOR
3 INNER JOIN PROVEEDOR_CATEGORIA on PROVEEDOR_CATEGORIA.nifProv = PROVEEDOR.nifProv
4 INNER JOIN CATEGORIA on CATEGORIA.idCat = PROVEEDOR_CATEGORIA.idCat
5 WHERE CATEGORIA.idCat = 6
6
7
```

Below the query editor, the 'Result Grid' displays the results of the query. The grid has columns: nifProv, nomProv, nomContProv, telsProv, dirProv, and ma. The results are as follows:

nifProv	nomProv	nomContProv	telsProv	dirProv	ma
A01189364	ALAVESA DE PATATAS	J. Suarez Tascon	945400429	Poligono industrial Lurgorri s/n Alegria-Dulantzi 0...	alav
B90307034	FRUTAS CUEVAS	J. Cuevas	954417158	Poligono Pagusa Calle Labrador 47, 41007,Sevilla	info

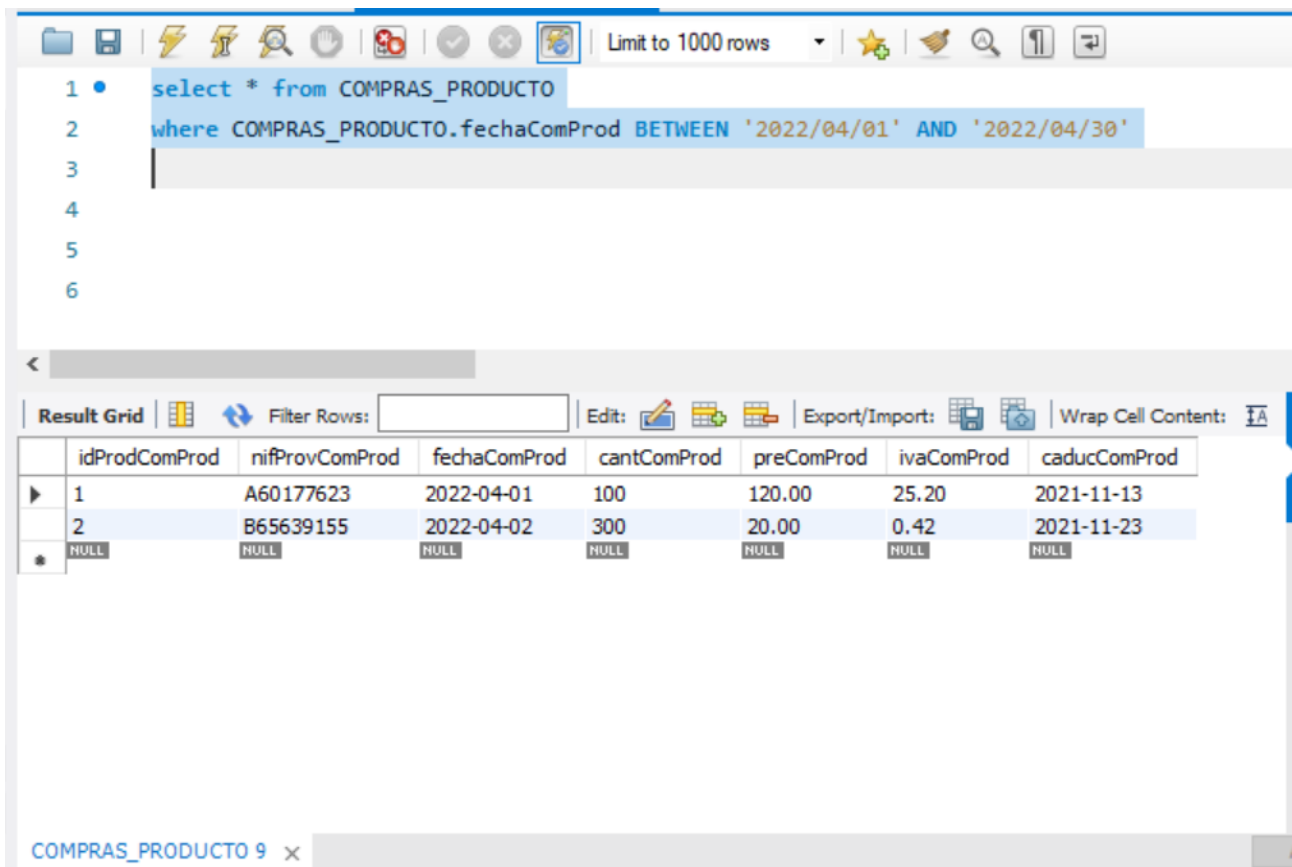
Código:

SELECT PROVEEDOR.* **FROM** PROVEEDOR **INNER JOIN** PROVEEDOR_CATEGORIA on PROVEEDOR_CATEGORIA.nifProv = PROVEEDOR.nifProv **INNER JOIN** CATEGORIA on CATEGORIA.idCat = PROVEEDOR_CATEGORIA.idCat **WHERE** CATEGORIA.idCat = 6

e. Listar las compras realizadas durante el mes en curso.

Usaremos **Where**, **Between** y **AND**.

Usando estos comandos seleccionamos la tabla donde (WHERE) habrá que establecer una fecha de inicio y de fin (Between y AND).



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1 • select * from COMPRAS_PRODUCTO
2 where COMPRAS_PRODUCTO.fechaComProd BETWEEN '2022/04/01' AND '2022/04/30'
3
4
5
6
```

Below the editor is the 'Result Grid' section, which displays the query results in a table. The table has the following columns: idProdComProd, nifProvComProd, fechaComProd, cantComProd, preComProd, ivaComProd, and caducComProd. The results show two rows of data for purchases made in April 2022.

	idProdComProd	nifProvComProd	fechaComProd	cantComProd	preComProd	ivaComProd	caducComProd
▶	1	A60177623	2022-04-01	100	120.00	25.20	2021-11-13
	2	B65639155	2022-04-02	300	20.00	0.42	2021-11-23
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The bottom of the interface shows a tab labeled 'COMPRAS_PRODUCTO 9'.

Código:

```
select * from COMPRAS_PRODUCTO where COMPRAS_PRODUCTO.fechaComProd BETWEEN  
'2022/04/01' AND '2022/04/30'
```

f. Indicar el número de platos de ensalada registrados.

The screenshot shows a SQL query editor with the following query:

```
1 select PLATO.nomPlato, TIPO.descTipoDePlato
2 FROM PLATO
3 INNER JOIN TIPO ON TIPO.idTipoPlato = PLATO.idTipoPlato
4 WHERE TIPO.idTipoPlato = 4
5
```

Below the query, the result grid is displayed with the following data:

nomPlato	descTipoDePlato
Ensalada de tomate	Ensalada

At the bottom, a tab labeled "Result 11" is visible.

Código

```
select PLATO.nomPlato, TIPO.descTipoDePlato FROM PLATO INNER JOIN TIPO ON  
TIPO.idTipoPlato = PLATO.idTipoPlato WHERE TIPO.idTipoPlato = 4
```

g. Mostrar aquellos proveedores cuyo nombre comience por la letra M.

Usamos el operador lógico **LIKE** se usa con una sentencia **WHERE** para poder determinar si una cadena de caracteres específica coincide con un patrón específico.

Junto con el Like usamos:

- **%** : Cualquier cadena de cero o más caracteres.
- **_** : Cualquier carácter individual.

The screenshot shows a SQL query editor with the following query:

```
1 SELECT * FROM PROVEEDOR WHERE nomContProv LIKE 'M%';
```

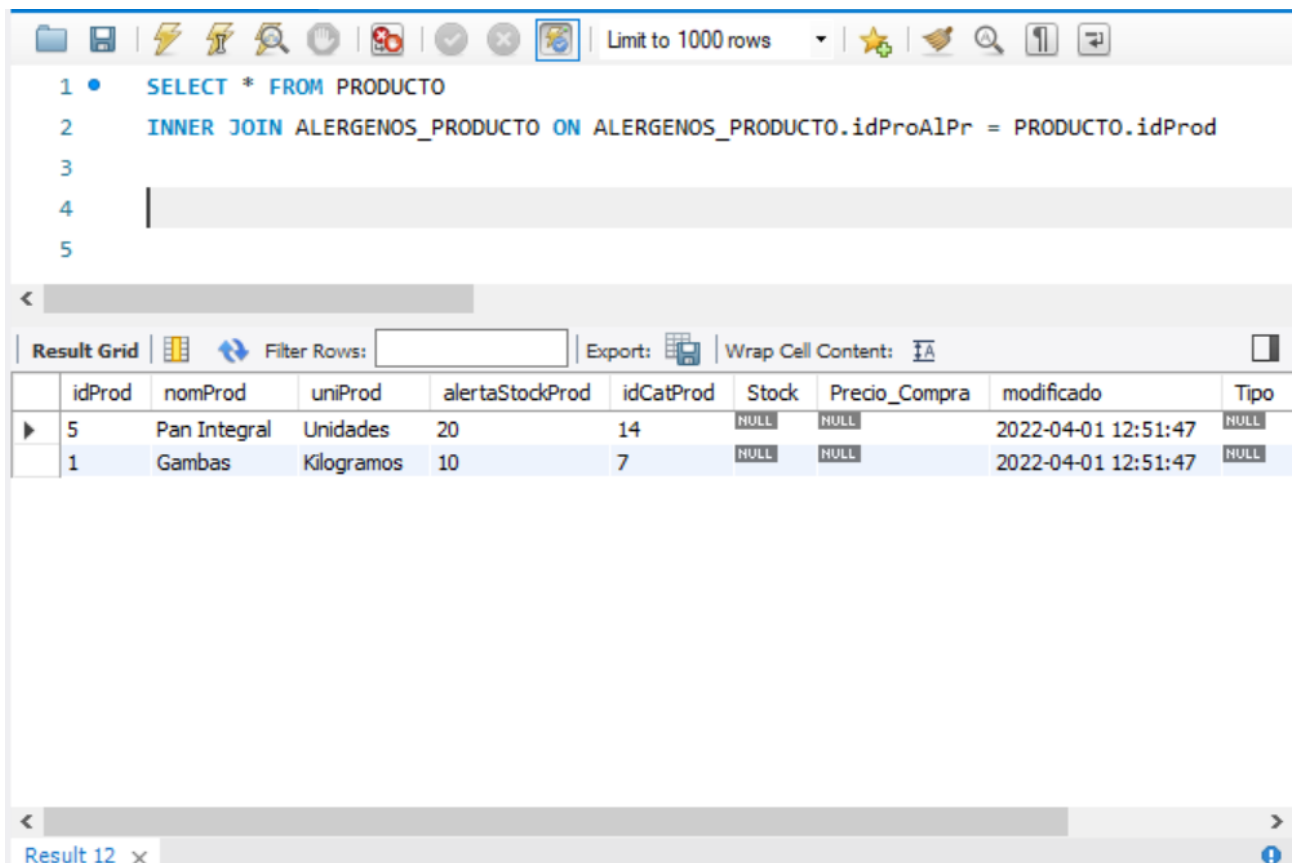
Below the query, the result grid is displayed with the following data:

nifProv	nomProv	nomContProv	telsProv	dirProv	mailProv	webProv	cod
B73148793	AGRORIZAO	M. Agrorizao	968425470	Carretera Nacional 340 (KM 614), - 30850 Tota...	ventas@agrorizao.com	www.agrorizao.com	6

Código:

```
SELECT * FROM PROVEEDOR WHERE nomContProv LIKE 'M%';
```

h. Mostrar los productos que contengan alérgenos.



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT * FROM PRODUCTO
2 INNER JOIN ALERGENOS_PRODUCTO ON ALERGENOS_PRODUCTO.idProAlPr = PRODUCTO.idProd
3
4
5
```

Below the query editor is a "Result Grid" showing two rows of data:

	idProd	nomProd	uniProd	alertaStockProd	idCatProd	Stock	Precio_Compra	modificado	Tipo
▶	5	Pan Integral	Unidades	20	14	NULL	NULL	2022-04-01 12:51:47	NULL
	1	Gambas	Kilogramos	10	7	NULL	NULL	2022-04-01 12:51:47	NULL

Código:

```
SELECT * FROM PRODUCTO INNER JOIN ALERGENOS_PRODUCTO ON
ALERGENOS_PRODUCTO.idProAlPr = PRODUCTO.idProd
```

i. Reportar el total de compras en € realizadas a cada proveedor. La consulta deberá tener los siguientes campos: Empresa, Número de productos y Total Euros.

The screenshot shows a SQL query editor with the following query:

```
1 SELECT PROVEEDOR.nomProv,
2    sum(cantComProd) as "Total Productos", sum(cantComProd*preComProd) as "Total Euros"
3    from COMPRAS_PRODUCTO INNER JOIN PROVEEDOR ON COMPRAS_PRODUCTO.nifProvComProd=nifProv group
4
5
```

Below the query, a "Result Grid" is displayed with the following data:

nomProv	Total Productos	Total Euros
BOU Cafe	100	12000.00
FRUTAS CUEVAS	2	21.82
MAKRO	1285	2479.75
CHEF FRUIT	5	32.50
HUEVERIAS BONET, S.A.	100	160.00
BIRRA 365	500	8000.00
ALBERTO POLO DISTRIBUCIONES	25	74.75
INDUSTRIA AVICOLA SURENA	60	194.40

The interface includes a toolbar at the top with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" button on the right. The bottom status bar shows "Result 15" and "Read Only".

Código:

```
SELECT PROVEEDOR.nomProv,
sum(cantComProd) as "Total Productos", sum(cantComProd*preComProd) as "Total Euros"
from COMPRAS_PRODUCTO INNER JOIN PROVEEDOR ON
COMPRAS_PRODUCTO.nifProvComProd=nifProv group by nomProv;
```

j. Indicar aquellos productos que han cambiado de precio.

Hacemos una actualización de los precios en los Productos para que se note el cambio de precio:

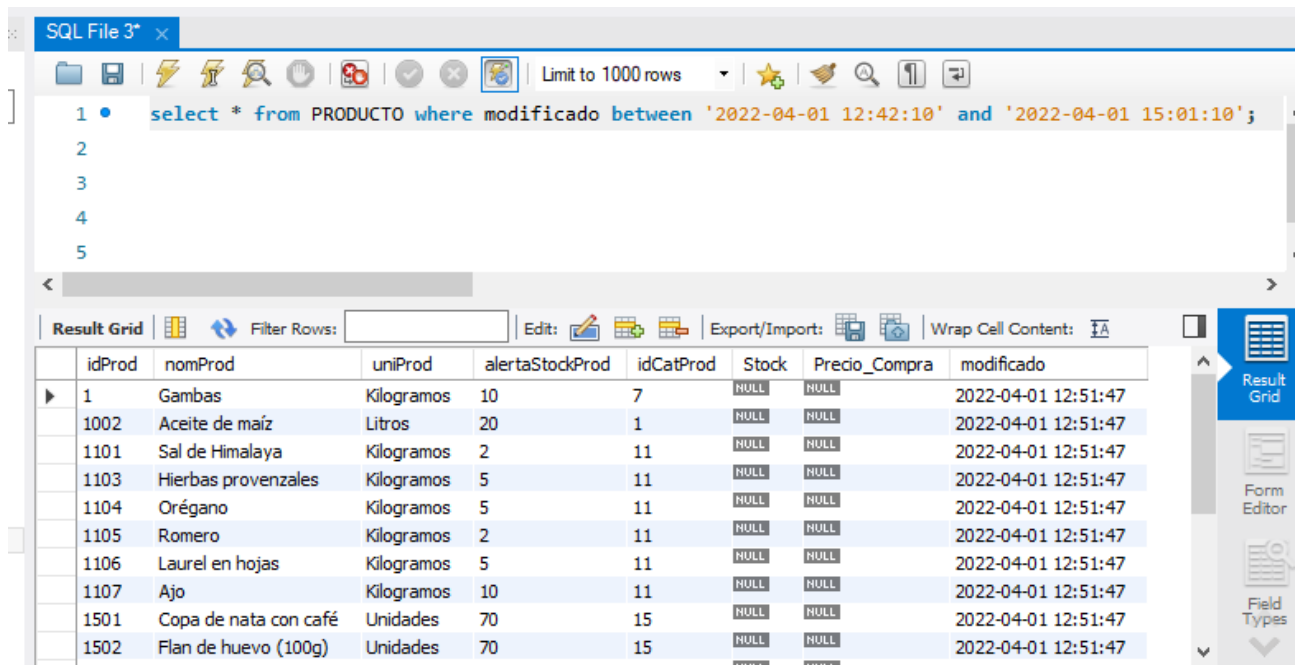
```
update PRODUCTO set precio_compra = precio_compra * 1.2 where idProd = 2;
```

Creamos una tabla para compilar los cambios realizados de precio. De esta manera, lo comparamos de manera más fácil.

```
alter table PRODUCTO add modificado timestamp default current_timestamp on update
current_timestamp after Precio_Compra;
```

Y para ver los productos que hemos modificado el precio en un rango de fecha:

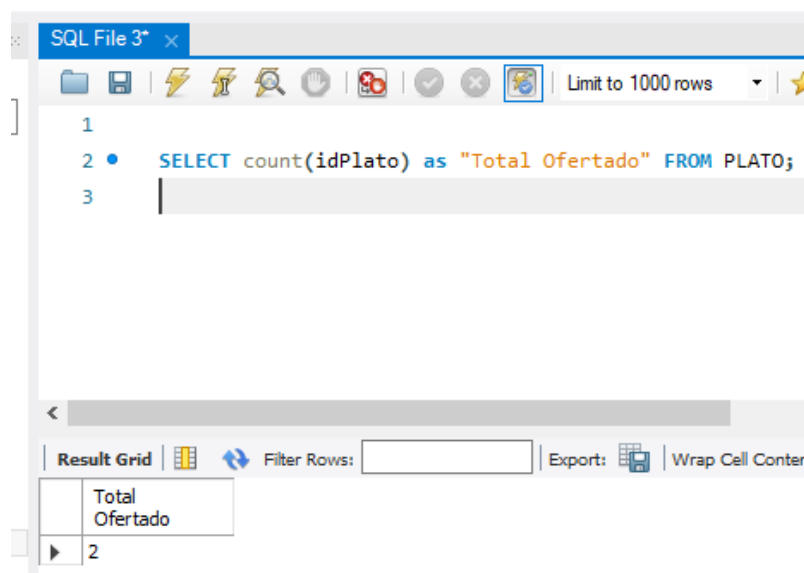
select * from PRODUCTO where modificado between '2022-04-01 12:42:10' and '2022-04-01 15:01:10';



The screenshot shows a SQL IDE window titled "SQL File 3*" with a toolbar and a query editor. The query is: `select * from PRODUCTO where modificado between '2022-04-01 12:42:10' and '2022-04-01 15:01:10';`. Below the editor is a "Result Grid" showing 12 rows of data. The columns are: idProd, nomProd, uniProd, alertaStockProd, idCatProd, Stock, Precio_Compra, and modificado. The data includes products like Gambas, Aceite de maíz, Sal de Himalaya, Hierbas provenzales, Orégano, Romero, Laurel en hojas, Ajo, Copa de nata con café, and Flan de huevo (100g).

idProd	nomProd	uniProd	alertaStockProd	idCatProd	Stock	Precio_Compra	modificado
1	Gambas	Kilogramos	10	7	NULL	NULL	2022-04-01 12:51:47
1002	Aceite de maíz	Litros	20	1	NULL	NULL	2022-04-01 12:51:47
1101	Sal de Himalaya	Kilogramos	2	11	NULL	NULL	2022-04-01 12:51:47
1103	Hierbas provenzales	Kilogramos	5	11	NULL	NULL	2022-04-01 12:51:47
1104	Orégano	Kilogramos	5	11	NULL	NULL	2022-04-01 12:51:47
1105	Romero	Kilogramos	2	11	NULL	NULL	2022-04-01 12:51:47
1106	Laurel en hojas	Kilogramos	5	11	NULL	NULL	2022-04-01 12:51:47
1107	Ajo	Kilogramos	10	11	NULL	NULL	2022-04-01 12:51:47
1501	Copa de nata con café	Unidades	70	15	NULL	NULL	2022-04-01 12:51:47
1502	Flan de huevo (100g)	Unidades	70	15	NULL	NULL	2022-04-01 12:51:47

K. Indicar la cantidad de platos ofertados por tipo de plato.



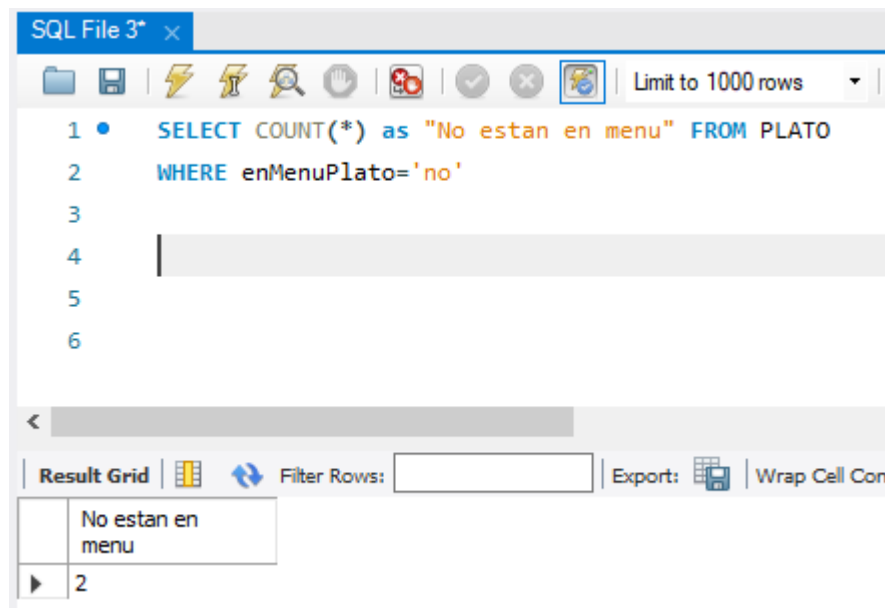
The screenshot shows a SQL IDE window titled "SQL File 3*" with a toolbar and a query editor. The query is: `SELECT count(idPlato) as "Total Ofertado" FROM PLATO;`. Below the editor is a "Result Grid" showing a single row with the value 2 under the column "Total Ofertado".

Total Ofertado
2

Código:

SELECT count(idPlato) as "Total Ofertado" **FROM** PLATO;

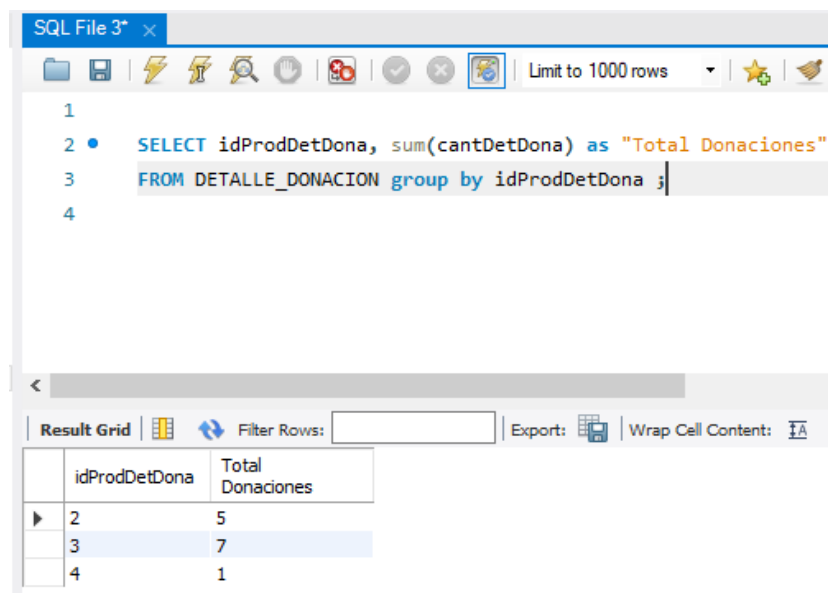
L. Indicar cuántos platos registrados NO se están ofertando en Menú.



Código:

```
SELECT COUNT(*) as "No están en menu" FROM PLATO WHERE enMenuPlato="no"
```

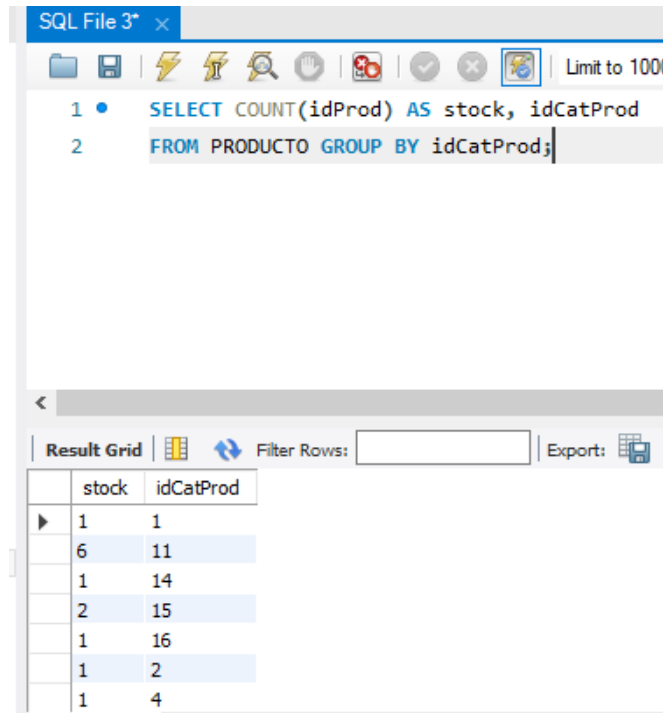
m. Contar el total de donaciones por producto.



Código:

```
SELECT idProdDetDona, sum(cantDetDona) AS "Total Donaciones" FROM DETALLE_DONACION
group by idProdDetDona ;
```

n. Formular una consulta que tenga una agrupación con Group By.



The screenshot shows a SQL editor window titled "SQL File 3*" with a toolbar and a "Limit to 100" option. The query is as follows:

```
1 • SELECT COUNT(idProd) AS stock, idCatProd
2 FROM PRODUCTO GROUP BY idCatProd;
```

Below the query, the "Result Grid" is displayed with a "Filter Rows" input and an "Export" button. The results are shown in a table with two columns: "stock" and "idCatProd".

	stock	idCatProd
▶	1	1
	6	11
	1	14
	2	15
	1	16
	1	2
	1	4

Código:

SELECT COUNT(idProd) AS stock, idCatProd FROM PRODUCTO GROUP BY idCatProd;

o. Formular una consulta que tenga una agrupación con Group By y un filtraje con Having.

Listar los mayores proveedores de productos, que serán aquellos que nos venden más de 1000 euros.

SQL Query:

```

1 SELECT PROVEEDOR.nomProv,
2 sum(cantComProd) as "Total Productos", sum(cantComProd*preComProd) as "Total Euros"
3 from COMPRAS_PRODUCTO INNER JOIN PROVEEDOR ON COMPRAS_PRODUCTO.nifProvComProd=nifProv
4 group by nomProv having sum(cantComProd*preComProd) > 1000;
5

```

nomProv	Total Productos	Total Euros
BOU Cafe	100	12000.00
MAKRO	1285	2479.75
BIRRA 365	500	8000.00

Código:

```

SELECT PROVEEDOR.nomProv, sum(cantComProd) as "Total Productos",
sum(cantComProd*preComProd) as "Total Euros" from COMPRAS_PRODUCTO INNER JOIN
PROVEEDOR ON COMPRAS_PRODUCTO.nifProvComProd=nifProv group by nomProv having
sum(cantComProd*preComProd) > 1000;

```

6. Realizar las siguientes modificaciones en los datos insertados y anotar la sentencia SQL y su salida en un documento:

a. Actualizar el campo **Porcentaje_Merma** con la división de la **Cantidad_Neta** entre la **Cantidad_Bruta** en la tabla **INGREDIENTES_PLATO**.

Al estar en un modo seguro, tenemos que deshabilitar estas opciones. *Preferencias > SQL Editor > Safe Updates....* y finalmente reiniciamos mySQL Workbench.

SQL Query:

```

1 UPDATE INGREDIENTES_PLATO
2 SET Porcentaje_Merma = (cantNetaInPl/cantBrutaInPl)*100;
3

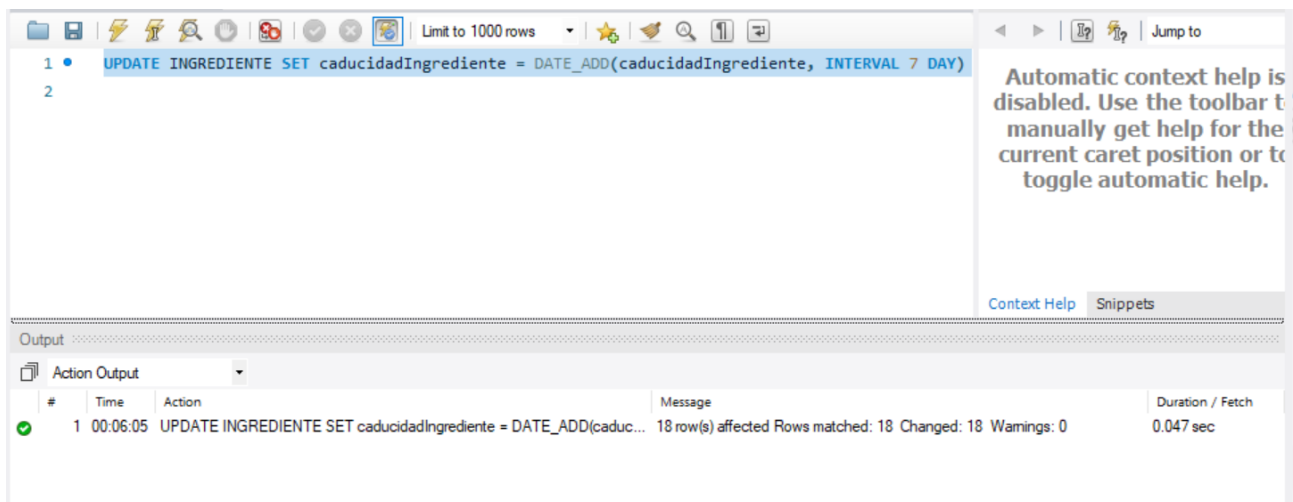
```

Output: Action Output

Código:

UPDATE INGREDIENTE_PLATO **SET** Porcentaje_Merma = (cantNetalnP/cantBrutalnP)*100;

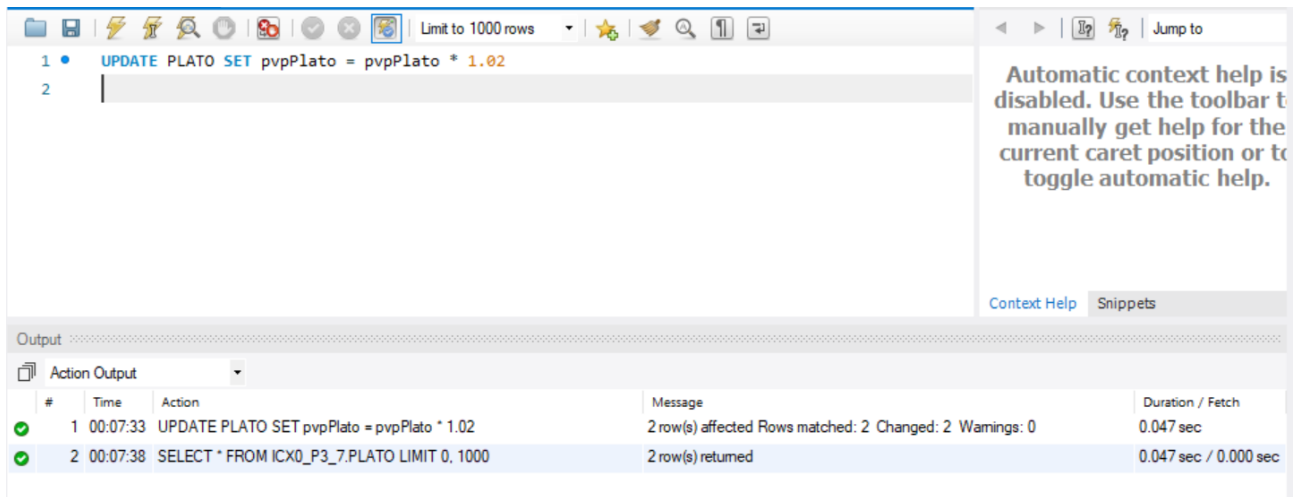
b. Cambiar la fecha de caducidad de los productos, sumándole 7 días.



Código:

UPDATE INGREDIENTE **SET** caducidadIngrediente = DATE_ADD(caducidadIngrediente, **INTERVAL 7 DAY**)

c. Actualizar el PVP de los platos, aumentando en un 2%.



Código:

```
UPDATE PLATO SET pvpPlato = pvpPlato * 1.02
```

d. Eliminar aquellas CATEGORÍAS que no tengan PROVEEDORES.

No podemos ejecutar porque tendríamos que borrar los productos asociados a estas categorías también, la instrucción sería la siguiente:

```
DELETE
FROM CATEGORIA
where idCat not in (
select idCat from PROVEEDOR_CATEGORIA
)
```

e. Borrar los productos elaborados que nunca se hayan vendido. Para ello deberás revisar la tabla COMANDA_ELABORADOS.

Como en el caso anterior no podemos ejecutar este borrado porque hay referencias en otras tablas y tendríamos que borrar esas también, la instrucción es la siguiente:

```
DELETE
FROM PRODUCTO
where idProd not in (
select idProdComElab from COMANDA_ELABORADOS
)
```