

SERVICIO NACIONAL DE APRENDIZAJE SENA

Programa
ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha:
3069140

Título de La evidencia:
Joins, Subconsultas y Vistas

Instructor Responsable:
Javier Yara

Aprendiz:
Brenda Castiblanco
Camilo Hurtado
Juan Pablo Valderrama
Juan Camilo Gil
Robinson Almonacid
Mauren Gonzalez

SERVICIO NACIONAL DE APRENDIZAJE SENA

2025.

CONTENIDO

INNER JOIN.....	4
1.1 Clientes y Tipo de Documento.....	4
1.2 Productos y Categorías.....	4
1.3 Facturas y Clientes.....	5
1.4 Productos Facturados y Detalle de Facturas.....	5
1.5 Usuarios y Roles.....	6
LEFT JOIN.....	6
2.1 Clientes y Usuario.....	6
2.3 Usuarios sin Roles.....	7
2.4 Facturas sin Cliente.....	8
2.5 Ventas con Productos Inexistentes.....	8
3.1 Usuarios sin Clientes.....	9
3.2 Categorías con o sin Productos.....	9
3.3 Roles sin Usuarios.....	10
3.4 Clientes sin Facturas.....	10
3.5 Productos sin Ventas.....	11
SUBCONSULTAS.....	11
4.1 Cliente con Más Facturas.....	12
4.2 Producto Más Vendido.....	12
4.3 Clientes con Alto Gasto.....	13
4.4 Productos Nunca Vendidos.....	13
4.5 Usuarios Sin Relación con Clientes.....	14
VISTAS.....	14
5.1 Vista de Clientes y Documentos.....	14
5.4 Vista de Usuarios y Roles.....	16
5.5 Vista de Detalle de Ventas.....	16
TRIGGERS.....	17
6.1 Reducir stock cuando se facture un producto.....	17
6.2 Validar stock antes de facturar.....	18
6.3 Registrar bitácora de cambios de usuarios.....	18
6.4 Calcular automáticamente precio de venta en productos_facturados.....	19
6.5 Verificación de longitud mínima de contraseñas en usuarios.....	20
PROCEDIMIENTOS.....	21
7.1 Obtener facturas de un cliente Devuelve todas las facturas de un cliente específico.....	21
7.2 Procedimiento para actualizar el stock de un producto.....	22
7.3 Crear un nuevo cliente, Inserta un registro en la tabla clientes de forma controlada.....	23
7.4 Lista todos los productos activos/inactivos de una categoría específica.....	24

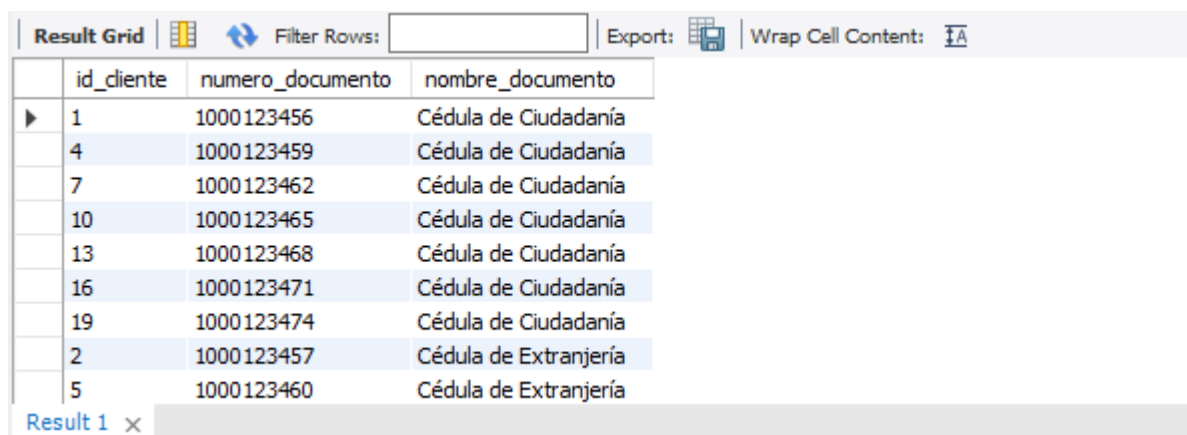
7.5 Generar detalle de factura -- Devuelve información completa de una factura: -- datos del cliente, productos facturados, precios y subtotales.....	24
FUNCIONES.....	25
8.1 Obtener nombre completo del cliente.....	25
8.2 Calcular total facturado de un cliente.....	26
8.3 Calcular stock valorizado de un producto.....	27
8.4 Obtener rol de un usuario.....	28
8.5 Contar productos por categoría.....	29
ENCRIPCIÓN.....	31
9.1 encriptar contraseñas función.....	31
9.2 Procedimiento para registrar usuario con contraseña encriptada.....	31

INNER JOIN

1.1 Clientes y Tipo de Documento.

```
SELECT c.id_cliente, c.numero_documento, td.nombre_documento
FROM clientes c
INNER JOIN tipo_documento td ON c.id_tipo_documento = td.id_tipo_documento;
```

-- **Explicación:** Devuelve solo los clientes que tiene un tipo de documento válido registrado.



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of the SQL query. The columns are 'id_cliente', 'numero_documento', and 'nombre_documento'. The results are as follows:

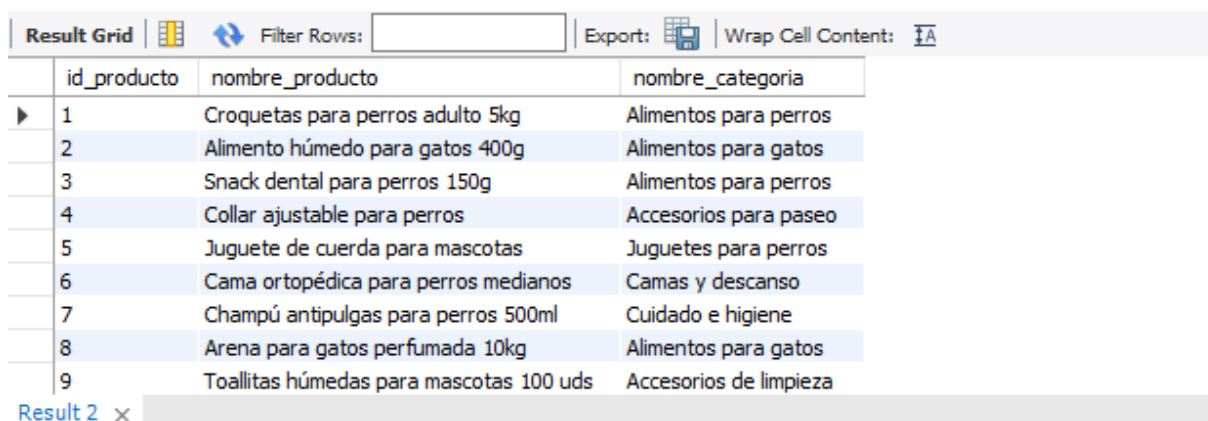
	id_cliente	numero_documento	nombre_documento
▶	1	1000123456	Cédula de Ciudadanía
	4	1000123459	Cédula de Ciudadanía
	7	1000123462	Cédula de Ciudadanía
	10	1000123465	Cédula de Ciudadanía
	13	1000123468	Cédula de Ciudadanía
	16	1000123471	Cédula de Ciudadanía
	19	1000123474	Cédula de Ciudadanía
	2	1000123457	Cédula de Extranjería
	5	1000123460	Cédula de Extranjería

Result 1 x

1.2 Productos y Categorías

```
SELECT p.id_producto, p.nombre_producto, c.nombre_categoria
FROM productos p
INNER JOIN categorias c ON p.id_categoria = c.id_categoria;
```

-- **Explicación:** Lista los productos con el nombre de su categoría.



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of the SQL query. The columns are 'id_producto', 'nombre_producto', and 'nombre_categoria'. The results are as follows:

	id_producto	nombre_producto	nombre_categoria
▶	1	Croquetas para perros adulto 5kg	Alimentos para perros
	2	Alimento húmedo para gatos 400g	Alimentos para gatos
	3	Snack dental para perros 150g	Alimentos para perros
	4	Collar ajustable para perros	Accesorios para paseo
	5	Juguete de cuerda para mascotas	Juguetes para perros
	6	Cama ortopédica para perros medianos	Camas y descanso
	7	Champú antipulgas para perros 500ml	Cuidado e higiene
	8	Arena para gatos perfumada 10kg	Alimentos para gatos
	9	Toallitas húmedas para mascotas 100 uds	Accesorios de limpieza

Result 2 x

1.3 Facturas y Clientes

```
SELECT f.id_factura, f.numero_factura, cl.primer_nombre, cl.primer_apellido
FROM facturas f
INNER JOIN clientes cl ON f.id_cliente = cl.id_cliente;
```

-- **Explicación:** Muestra las facturas junto con el nombre del cliente que las generó.

id_factura	numero_factura	primer_nombre	primer_apellido
1	100001	Juan	Pérez
2	100002	María	López
3	100003	Carlos	Gómez
4	100004	Laura	Ramírez
5	100005	Andrés	Mendoza
6	100006	Juliana	Castillo
7	100007	Sebastián	Moreno
8	100008	Camila	Sánchez
9	100009	David	Rojas

1.4 Productos Facturados y Detalle de Facturas

```
SELECT pf.id_producto_facturado, f.numero_factura, p.nombre_producto, pf.cantidad,
pf.precio_venta
FROM productos_facturados pf
INNER JOIN facturas f ON pf.id_factura = f.id_factura
INNER JOIN productos p ON pf.id_producto = p.id_producto;
```

-- **Explicación:** Muestra qué productos fueron vendidos en cada factura.

id_producto_facturado	numero_factura	nombre_producto	cantidad	precio_venta
1	100001	Croquetas para perros adulto 5kg	2	15000.00
2	100002	Alimento húmedo para gatos 400g	1	23999.99
3	100003	Snack dental para perros 150g	5	10240.00
4	100004	Collar ajustable para perros	3	26000.25
5	100005	Juguete de cuerda para mascotas	4	31250.00
6	100006	Cama ortopédica para perros medianos	2	22500.45
7	100007	Champú antipulgas para perros 500ml	1	67000.00
8	100008	Arena para gatos perfumada 10kg	3	9933.66
9	100009	Toallitas húmedas para mascotas 100 uds	2	45500.00

1.5 Usuarios y Roles

```
SELECT u.id_usuario, u.correo_electronico, r.nombre_rol
FROM usuarios u
INNER JOIN usuario_rol ur ON u.id_usuario = ur.id_usuario
INNER JOIN roles r ON ur.id_rol = r.id_rol;
```

-- **Explicación:** Une usuarios con sus roles asignados.

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
id_usuario	correo_electronico	nombre_rol	
1	admin@empresa.com	Administrador	
3	user2@empresa.com	Administrador	
5	user4@empresa.com	Administrador	
7	user6@empresa.com	Administrador	
9	user8@empresa.com	Administrador	
11	user10@empresa.com	Administrador	
13	user12@empresa.com	Administrador	
15	user14@empresa.com	Administrador	
17	user16@empresa.com	Administrador	

Result 5 x

LEFT JOIN

2.1 Clientes y Usuario

```
SELECT c.id_cliente, c.primer_nombre, u.correo_electronico
FROM clientes c
LEFT JOIN usuarios u ON c.id_usuario = u.id_usuario;
```

-- **Explicación:** Devuelve todos los clientes, incluso los que no tienen usuario en el sistema.

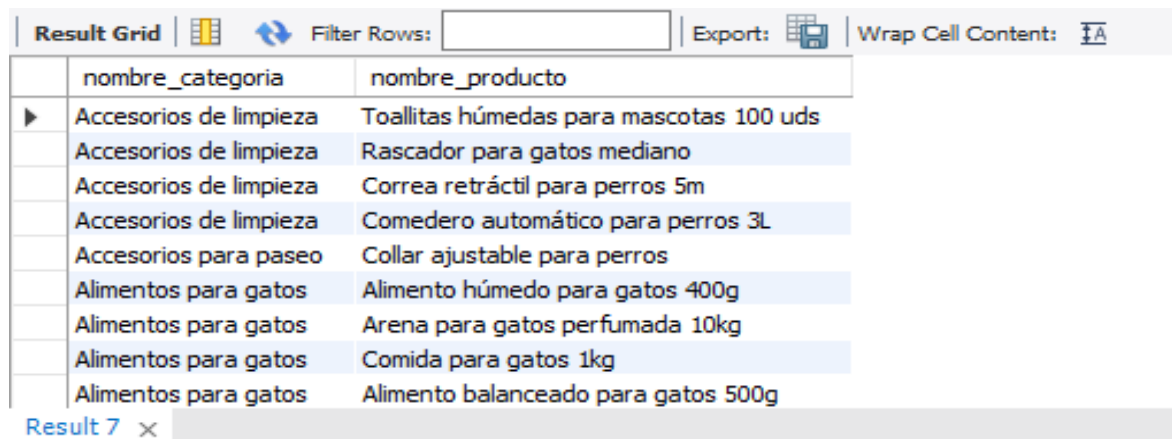
Result Grid	Filter Rows:	Export:	Wrap Cell Content:
id_cliente	primer_nombre	correo_electronico	
1	Juan	admin@empresa.com	
2	María	user1@empresa.com	
3	Carlos	NULL	
4	Laura	NULL	
5	Andrés	user2@empresa.com	
6	Juliana	NULL	
7	Sebastián	NULL	
8	Camila	user3@empresa.com	
9	David	NULL	

Result 6 x

2.2 Categorías sin Productos

```
SELECT c.nombre_categoria, p.nombre_producto
FROM categorias c
LEFT JOIN productos p ON c.id_categoria = p.id_categoria;
```

-- **Explicación:** Devuelve todas las categorías, aunque no tengan productos.



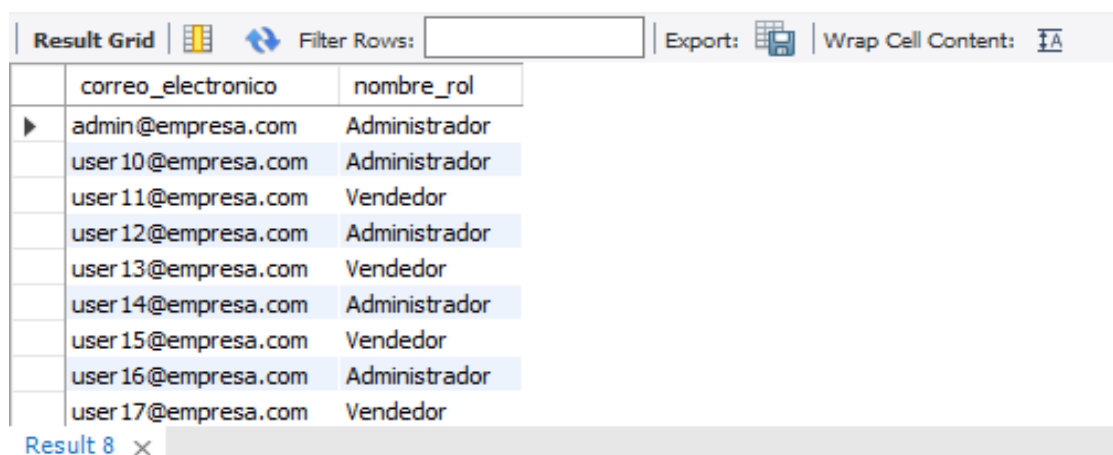
The screenshot shows a database query result grid with two columns: 'nombre_categoria' and 'nombre_producto'. The grid lists various categories and their corresponding products. The categories include 'Accesorios de limpieza', 'Accesorios para paseo', and 'Alimentos para gatos'. The products listed are 'Toallitas húmedas para mascotas 100 uds', 'Rascador para gatos mediano', 'Correa retráctil para perros 5m', 'Comedero automático para perros 3L', 'Collar ajustable para perros', 'Alimento húmedo para gatos 400g', 'Arena para gatos perfumada 10kg', 'Comida para gatos 1kg', and 'Alimento balanceado para gatos 500g'.

nombre_categoria	nombre_producto
Accesorios de limpieza	Toallitas húmedas para mascotas 100 uds
Accesorios de limpieza	Rascador para gatos mediano
Accesorios de limpieza	Correa retráctil para perros 5m
Accesorios de limpieza	Comedero automático para perros 3L
Accesorios para paseo	Collar ajustable para perros
Alimentos para gatos	Alimento húmedo para gatos 400g
Alimentos para gatos	Arena para gatos perfumada 10kg
Alimentos para gatos	Comida para gatos 1kg
Alimentos para gatos	Alimento balanceado para gatos 500g

2.3 Usuarios sin Roles

```
SELECT u.correo_electronico, r.nombre_rol
FROM usuarios u
LEFT JOIN usuario_rol ur ON u.id_usuario = ur.id_usuario
LEFT JOIN roles r ON ur.id_rol = r.id_rol;
```

-- **Explicación:** Muestra todos los usuarios, aunque no tengan roles asignados.



The screenshot shows a database query result grid with two columns: 'correo_electronico' and 'nombre_rol'. The grid lists various users and their assigned roles. The users include 'admin@empresa.com', 'user10@empresa.com', 'user11@empresa.com', 'user12@empresa.com', 'user13@empresa.com', 'user14@empresa.com', 'user15@empresa.com', 'user16@empresa.com', and 'user17@empresa.com'. The roles listed are 'Administrador' and 'Vendedor'.

correo_electronico	nombre_rol
admin@empresa.com	Administrador
user10@empresa.com	Administrador
user11@empresa.com	Vendedor
user12@empresa.com	Administrador
user13@empresa.com	Vendedor
user14@empresa.com	Administrador
user15@empresa.com	Vendedor
user16@empresa.com	Administrador
user17@empresa.com	Vendedor

2.4 Facturas sin Cliente

```
SELECT f.numero_factura, cl.primer_nombre, cl.primer_apellido
FROM facturas f
LEFT JOIN clientes cl ON f.id_cliente = cl.id_cliente;
```

-- **Explicación:** Lista todas las facturas, incluso si el cliente ya no existe.

	numero_factura	primer_nombre	primer_apellido
▶	100001	Juan	Pérez
	100002	María	López
	100003	Carlos	Gómez
	100004	Laura	Ramírez
	100005	Andrés	Mendoza
	100006	Juliana	Castillo
	100007	Sebastián	Moreno
	100008	Camila	Sánchez
	100009	David	Rojas

Result 9 ×

2.5 Ventas con Productos Inexistentes

```
SELECT pf.id_producto_facturado, pf.cantidad, p.nombre_producto
FROM productos_facturados pf
LEFT JOIN productos p ON pf.id_producto = p.id_producto;
```

-- **Explicación:** Devuelve todas las ventas registradas, aunque el producto ya no exista en inventario.

	id_producto_facturado	cantidad	nombre_producto
▶	1	2	Croquetas para perros adulto 5kg
	2	1	Alimento húmedo para gatos 400g
	3	5	Snack dental para perros 150g
	4	3	Collar ajustable para perros
	5	4	Juguete de cuerda para mascotas
	6	2	Cama ortopédica para perros medianos
	7	1	Champú antipulgas para perros 500ml
	8	3	Arena para gatos perfumada 10kg
	9	2	Toallitas húmedas para mascotas 100 uds

Result 10 ×

RIGHT JOIN

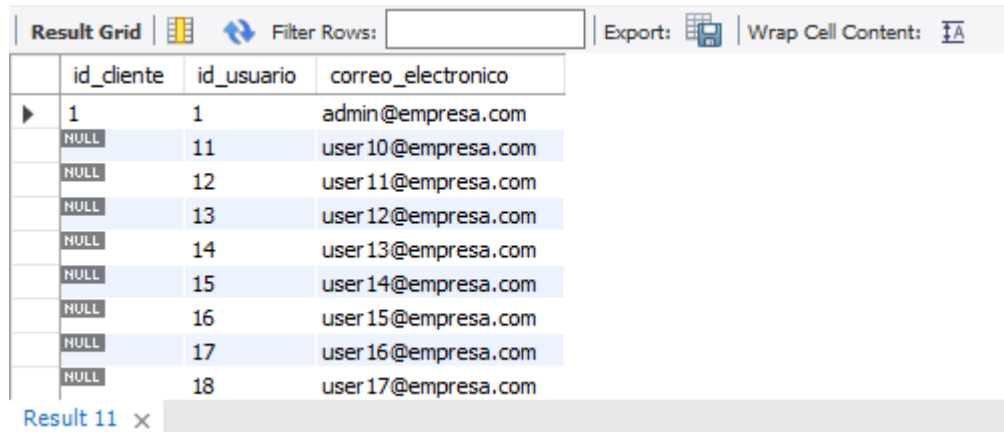
3.1 Usuarios sin Clientes

```
SELECT c.id_cliente, u.id_usuario, u.correo_electronico
```

```
FROM clientes c
```

```
RIGHT JOIN usuarios u ON c.id_usuario = u.id_usuario;
```

-- **Explicación:** Muestra todos los usuarios, incluso si no están asociados a un cliente.



	id_cliente	id_usuario	correo_electronico
▶	1	1	admin@empresa.com
	NULL	11	user10@empresa.com
	NULL	12	user11@empresa.com
	NULL	13	user12@empresa.com
	NULL	14	user13@empresa.com
	NULL	15	user14@empresa.com
	NULL	16	user15@empresa.com
	NULL	17	user16@empresa.com
	NULL	18	user17@empresa.com

Result 11 ×

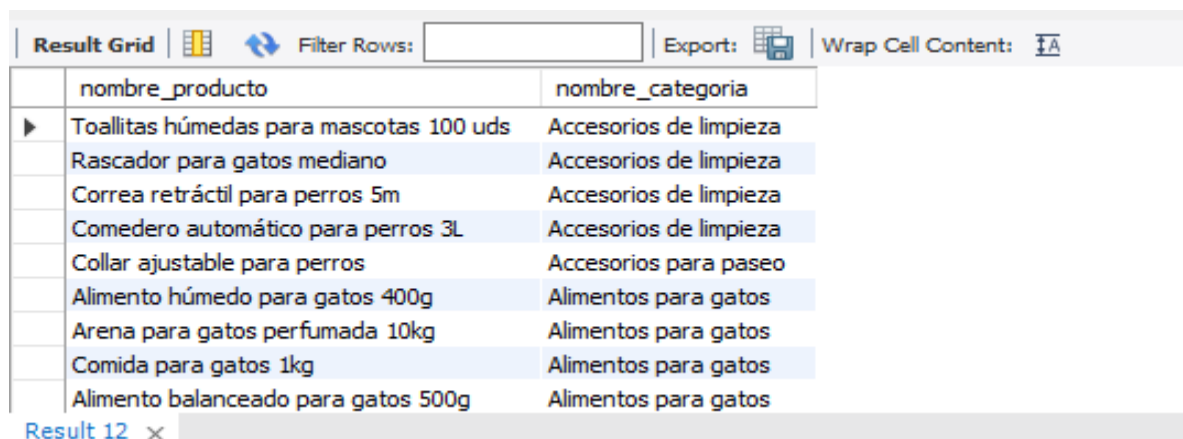
3.2 Categorías con o sin Productos

```
SELECT p.nombre_producto, c.nombre_categoria
```

```
FROM productos p
```

```
RIGHT JOIN categorias c ON p.id_categoria = c.id_categoria;
```

-- **Explicación:** Muestra todas las categorías, destacando incluso las que no tienen productos.



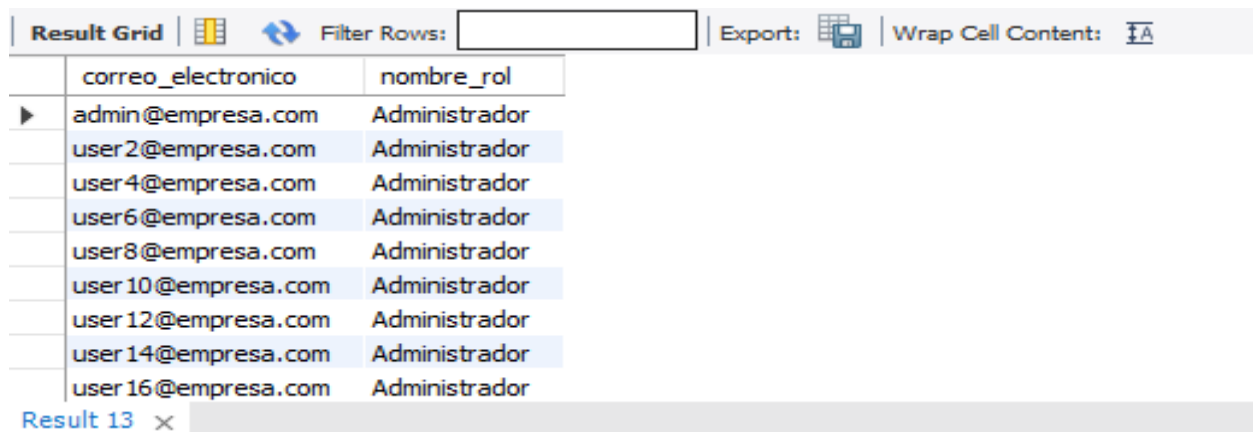
	nombre_producto	nombre_categoria
▶	Toallitas húmedas para mascotas 100 uds	Accesorios de limpieza
	Rascador para gatos mediano	Accesorios de limpieza
	Correa retráctil para perros 5m	Accesorios de limpieza
	Comedero automático para perros 3L	Accesorios de limpieza
	Collar ajustable para perros	Accesorios para paseo
	Alimento húmedo para gatos 400g	Alimentos para gatos
	Arena para gatos perfumada 10kg	Alimentos para gatos
	Comida para gatos 1kg	Alimentos para gatos
	Alimento balanceado para gatos 500g	Alimentos para gatos

Result 12 ×

3.3 Roles sin Usuarios

```
SELECT u.correo_electronico, r.nombre_rol
FROM usuarios u
RIGHT JOIN usuario_rol ur ON u.id_usuario = ur.id_usuario
RIGHT JOIN roles r ON ur.id_rol = r.id_rol;
```

-- **Explicación:** Lista todos los roles, incluso si no tienen usuarios.



The screenshot shows a database query result grid with the following data:

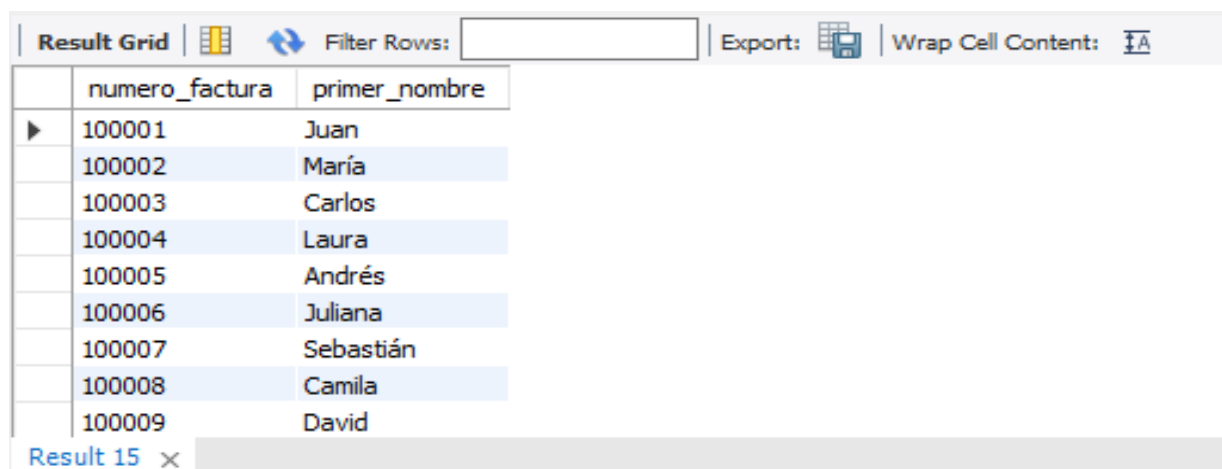
	correo_electronico	nombre_rol
▶	admin@empresa.com	Administrador
	user2@empresa.com	Administrador
	user4@empresa.com	Administrador
	user6@empresa.com	Administrador
	user8@empresa.com	Administrador
	user10@empresa.com	Administrador
	user12@empresa.com	Administrador
	user14@empresa.com	Administrador
	user16@empresa.com	Administrador

Result 13 x

3.4 Clientes sin Facturas

```
SELECT f.numero_factura, cl.primer_nombre
FROM facturas f
RIGHT JOIN clientes cl ON f.id_cliente = cl.id_cliente;
```

-- **Explicación:** Lista todos los clientes, incluso si nunca han hecho una factura.



The screenshot shows a database query result grid with the following data:

	numero_factura	primer_nombre
▶	100001	Juan
	100002	María
	100003	Carlos
	100004	Laura
	100005	Andrés
	100006	Juliana
	100007	Sebastián
	100008	Camila
	100009	David

Result 15 x

3.5 Productos sin Ventas

```
SELECT pf.id_producto_facturado, p.nombre_producto
```

```
FROM productos_facturados pf
```

```
RIGHT JOIN productos p ON pf.id_producto = p.id_producto;
```

-- **Explicación:** Devuelve todos los productos, incluso si no se han vendido nunca.

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	id_producto_facturado	nombre_producto			
▶	1	Croquetas para perros adulto 5kg			
	11	Croquetas para perros adulto 5kg			
	2	Alimento húmedo para gatos 400g			
	12	Alimento húmedo para gatos 400g			
	3	Snack dental para perros 150g			
	13	Snack dental para perros 150g			
	4	Collar ajustable para perros			
	14	Collar ajustable para perros			
	5	Juguete de cuerda para mascotas			

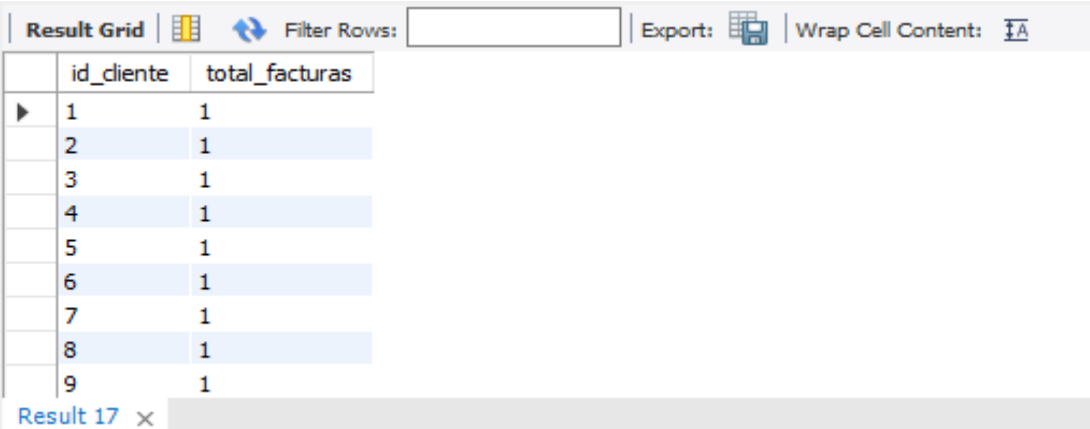
Result 16 x

SUBCONSULTAS

4.1 Cliente con Más Facturas

```
SELECT id_cliente, COUNT(*) AS total_facturas
FROM facturas
GROUP BY id_cliente
HAVING COUNT(*) = (
    SELECT MAX(cantidad)
    FROM (SELECT COUNT(*) AS cantidad FROM facturas GROUP BY id_cliente) AS t
);
```

-- **Explicación:** Encuentra al cliente que más facturas tiene.



The screenshot shows a database query result grid. The grid has two columns: 'id_cliente' and 'total_facturas'. There are 9 rows of data, all with a value of 1 in the 'total_facturas' column. The grid is titled 'Result 17' and includes a 'Filter Rows' button and an 'Export' button.

	id_cliente	total_facturas
▶	1	1
	2	1
	3	1
	4	1
	5	1
	6	1
	7	1
	8	1
	9	1

4.2 Producto Más Vendido

```
SELECT id_producto, SUM(cantidad) AS total_vendido
FROM productos_facturados
GROUP BY id_producto
HAVING SUM(cantidad) = (
    SELECT MAX(total)
    FROM (SELECT SUM(cantidad) AS total FROM productos_facturados GROUP BY id_producto)
t
);
```

-- **Explicación:** Devuelve el producto más vendido en la historia.



The screenshot shows a database query result grid. The grid has two columns: 'id_producto' and 'total_vendido'. There is 1 row of data with a value of 8 in the 'total_vendido' column. The grid is titled 'Result 17' and includes a 'Filter Rows' button and an 'Export' button.

	id_producto	total_vendido
▶	3	8

4.3 Clientes con Alto Gasto

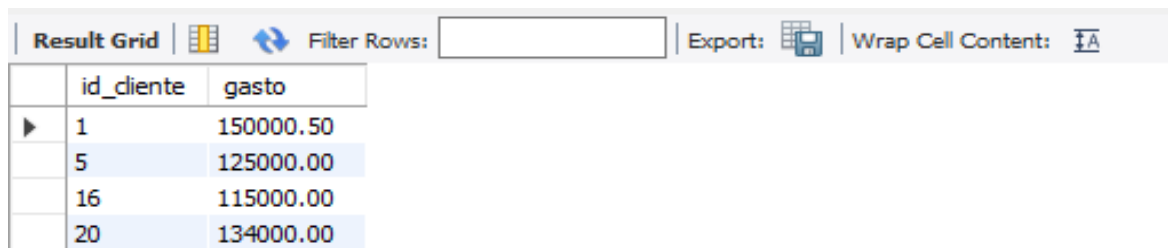
```
SELECT id_cliente, SUM(total) AS gasto
```

```
FROM facturas
```

```
GROUP BY id_cliente
```

```
HAVING SUM(total) > 100000;
```

-- **Explicación:** Subconsulta implícita dentro de agregación para detectar clientes grandes compradores.



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query. The columns are 'id_cliente' and 'gasto'. The data rows are as follows:

id_cliente	gasto
1	150000.50
5	125000.00
16	115000.00
20	134000.00

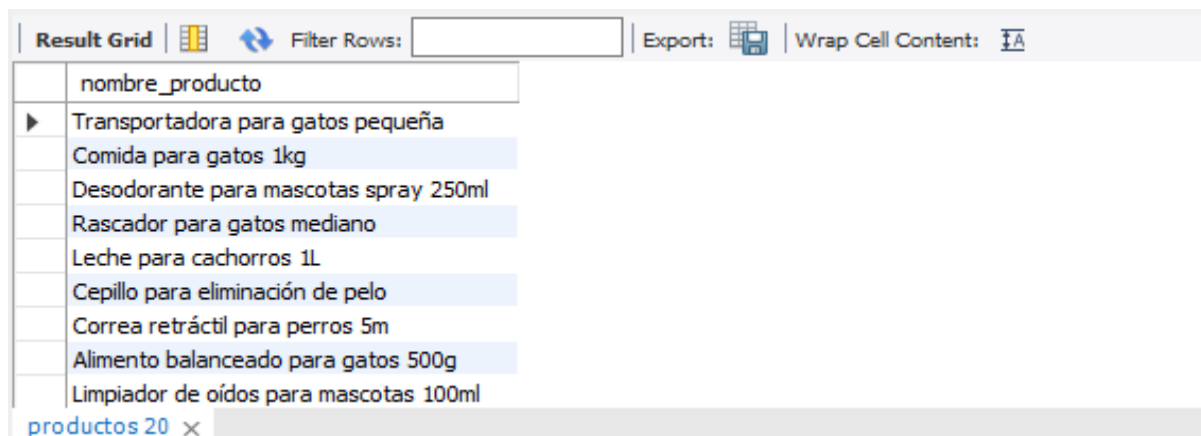
4.4 Productos Nunca Vendidos

```
SELECT nombre_producto
```

```
FROM productos
```

```
WHERE id_producto NOT IN (SELECT DISTINCT id_producto FROM productos_facturados);
```

-- **Explicación:** Lista los productos que nunca se han vendido.



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query. The column is 'nombre_producto'. The data rows are as follows:

nombre_producto
Transportadora para gatos pequeña
Comida para gatos 1kg
Desodorante para mascotas spray 250ml
Rascador para gatos mediano
Leche para cachorros 1L
Cepillo para eliminación de pelo
Correa retráctil para perros 5m
Alimento balanceado para gatos 500g
Limpiador de oídos para mascotas 100ml

productos 20 x

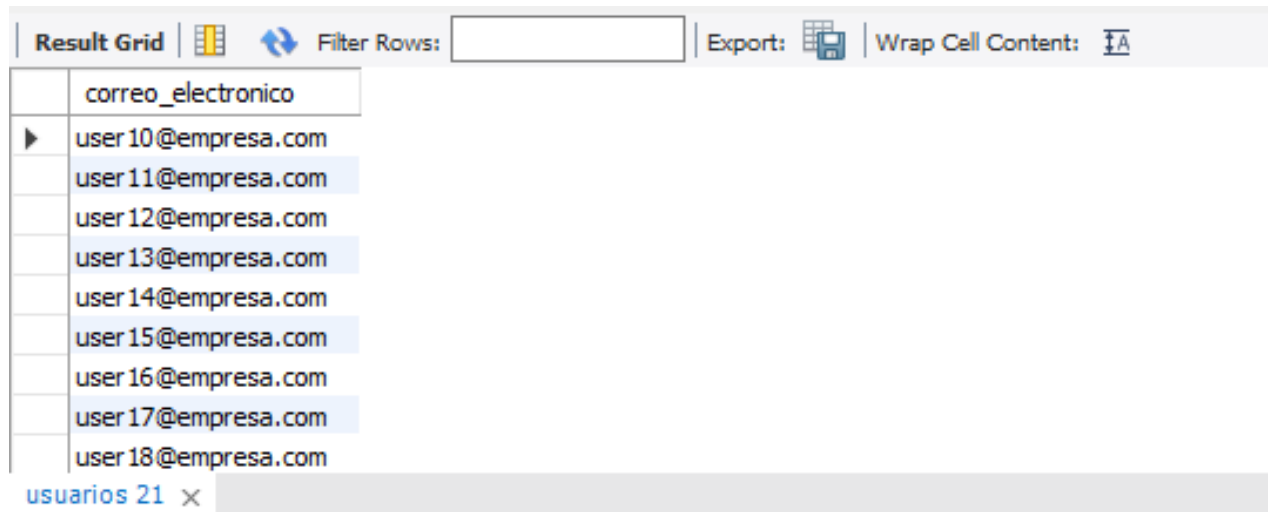
4.5 Usuarios Sin Relación con Clientes

```
SELECT correo_electronico
```

```
FROM usuarios
```

```
WHERE id_usuario NOT IN (SELECT id_usuario FROM clientes WHERE id_usuario IS NOT NULL);
```

-- **Explicación:** Encuentra usuarios que no tienen relación con clientes.



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays a single column titled 'correo_electronico'. Below the header, there are 18 rows of email addresses, all starting with 'user' followed by a number (10 to 18) and '@empresa.com'. The interface includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' option. At the bottom, a status bar indicates 'usuarios 21'.

correo_electronico
user10@empresa.com
user11@empresa.com
user12@empresa.com
user13@empresa.com
user14@empresa.com
user15@empresa.com
user16@empresa.com
user17@empresa.com
user18@empresa.com

VISTAS

5.1 Vista de Clientes y Documentos

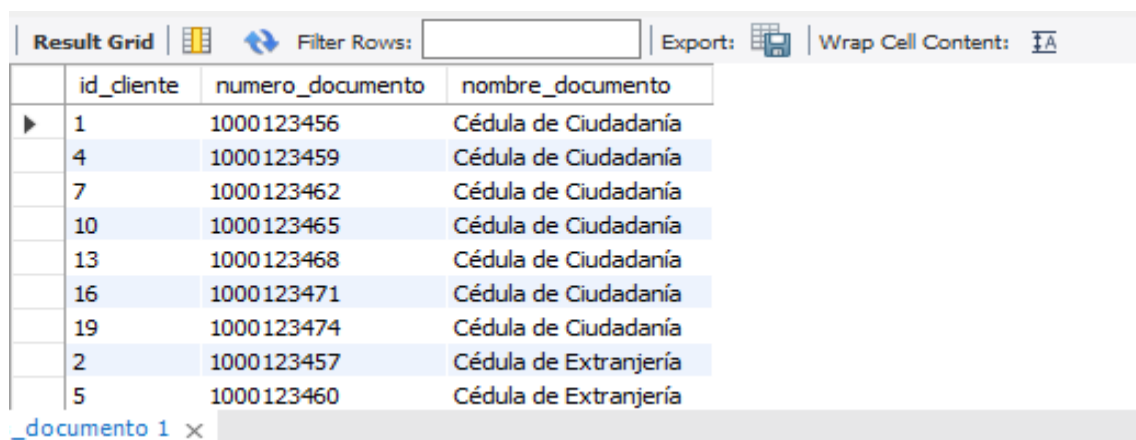
```
CREATE VIEW vw_clientes_documento AS
```

```
SELECT c.id_cliente, c.numero_documento, td.nombre_documento
```

```
FROM clientes c
```

```
INNER JOIN tipo_documento td ON c.id_tipo_documento = td.id_tipo_documento;
```

-- **Explicación:** Vista rápida para obtener clientes con su tipo de documento.



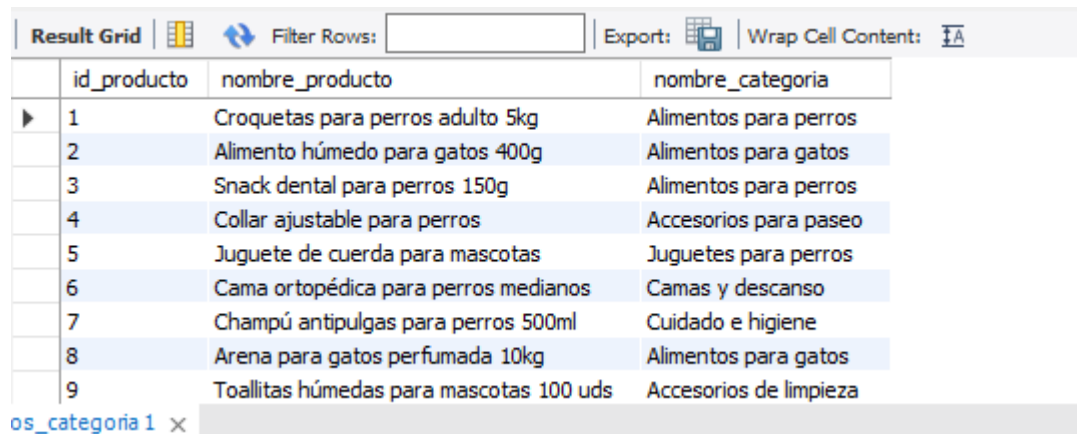
The screenshot shows a database interface with a 'Result Grid' tab. The grid displays three columns: 'id_cliente', 'numero_documento', and 'nombre_documento'. There are 10 rows of data. The first 9 rows show 'Cédula de Ciudadanía' for various client IDs and document numbers. The last two rows show 'Cédula de Extranjería' for client IDs 2 and 5. The interface includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' option. At the bottom, a status bar indicates '_documento 1'.

id_cliente	numero_documento	nombre_documento
1	1000123456	Cédula de Ciudadanía
4	1000123459	Cédula de Ciudadanía
7	1000123462	Cédula de Ciudadanía
10	1000123465	Cédula de Ciudadanía
13	1000123468	Cédula de Ciudadanía
16	1000123471	Cédula de Ciudadanía
19	1000123474	Cédula de Ciudadanía
2	1000123457	Cédula de Extranjería
5	1000123460	Cédula de Extranjería

5.2 Vista de Productos y Categorías

```
CREATE VIEW vw_productos_categoria AS  
SELECT p.id_producto, p.nombre_producto, c.nombre_categoria  
FROM productos p  
INNER JOIN categorias c ON p.id_categoria = c.id_categoria;
```

-- **Explicación:** Muestra productos junto con su categoría.



	id_producto	nombre_producto	nombre_categoria
▶	1	Croquetas para perros adulto 5kg	Alimentos para perros
	2	Alimento húmedo para gatos 400g	Alimentos para gatos
	3	Snack dental para perros 150g	Alimentos para perros
	4	Collar ajustable para perros	Accesorios para paseo
	5	Juguete de cuerda para mascotas	Juguetes para perros
	6	Cama ortopédica para perros medianos	Camas y descanso
	7	Champú antipulgas para perros 500ml	Cuidado e higiene
	8	Arena para gatos perfumada 10kg	Alimentos para gatos
	9	Toallitas húmedas para mascotas 100 uds	Accesorios de limpieza

os_categoria 1 x

5.3 Vista de Facturas y Clientes

```
CREATE VIEW vw_facturas_clientes AS  
SELECT f.numero_factura, f.total, cl.primer_nombre, cl.primer_apellido  
FROM facturas f  
INNER JOIN clientes cl ON f.id_cliente = cl.id_cliente;
```

-- **Explicación:** Une facturas con sus clientes.

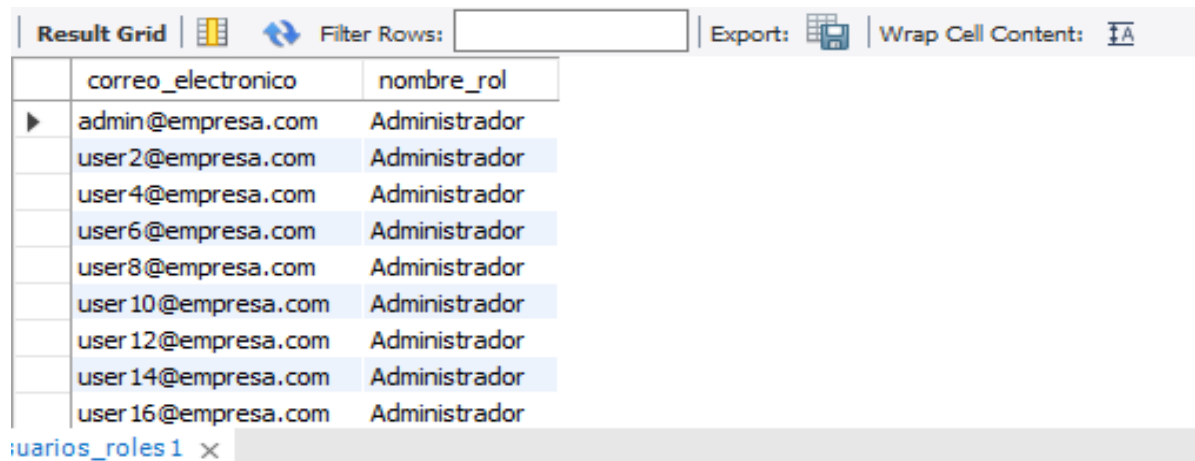


	numero_factura	total	primer_nombre	primer_apellido
▶	100001	150000.50	Juan	Pérez
	100002	23999.99	María	López
	100003	51200.00	Carlos	Gómez
	100004	78000.75	Laura	Ramírez
	100005	125000.00	Andrés	Mendoza
	100006	45000.90	Juliana	Castillo
	100007	67000.00	Sebastián	Moreno
	100008	29800.99	Camila	Sánchez
	100009	91000.00	David	Rojas

ras_clientes 1 x

5.4 Vista de Usuarios y Roles

```
CREATE VIEW vw_usuarios_rol AS  
SELECT u.correo_electronico, r.nombre_rol  
FROM usuarios u  
INNER JOIN usuario_rol ur ON u.id_usuario = ur.id_usuario  
INNER JOIN roles r ON ur.id_rol = r.id_rol;  
-- Explicación: Relación clara de usuarios y sus roles.
```

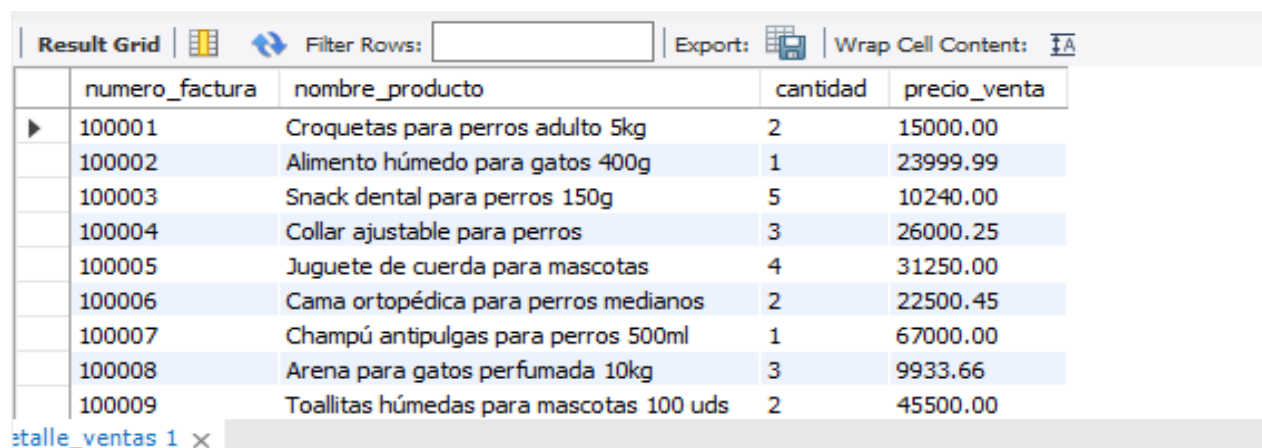


	correo_electronico	nombre_rol
▶	admin@empresa.com	Administrador
	user2@empresa.com	Administrador
	user4@empresa.com	Administrador
	user6@empresa.com	Administrador
	user8@empresa.com	Administrador
	user10@empresa.com	Administrador
	user12@empresa.com	Administrador
	user14@empresa.com	Administrador
	user16@empresa.com	Administrador

usuarios_rols 1 x

5.5 Vista de Detalle de Ventas

```
CREATE VIEW vw_detalle_ventas AS  
SELECT f.numero_factura, p.nombre_producto, pf.cantidad, pf.precio_venta  
FROM productos_facturados pf  
INNER JOIN facturas f ON pf.id_factura = f.id_factura  
INNER JOIN productos p ON pf.id_producto = p.id_producto;  
-- Explicación: Vista que muestra el detalle de cada venta.
```



	numero_factura	nombre_producto	cantidad	precio_venta
▶	100001	Croquetas para perros adulto 5kg	2	15000.00
	100002	Alimento húmedo para gatos 400g	1	23999.99
	100003	Snack dental para perros 150g	5	10240.00
	100004	Collar ajustable para perros	3	26000.25
	100005	Juguete de cuerda para mascotas	4	31250.00
	100006	Cama ortopédica para perros medianos	2	22500.45
	100007	Champú antipulgas para perros 500ml	1	67000.00
	100008	Arena para gatos perfumada 10kg	3	9933.66
	100009	Toallitas húmedas para mascotas 100 uds	2	45500.00

detalle_ventas 1 x

TRIGGERS

6.1 Reducir stock cuando se facture un producto

```
DELIMITER //
```

```
CREATE TRIGGER tr_reducir_stock  
AFTER INSERT ON productos_facturados  
FOR EACH ROW  
BEGIN  
    UPDATE productos  
    SET stock = stock - NEW.cantidad  
    WHERE id_producto = NEW.id_producto;  
END //
```

```
DELIMITER ;
```

```
264 • SELECT stock FROM productos WHERE id_producto = 1;  
265  
266 -- Insertar producto facturado (debería reducir stock)  
267 • INSERT INTO productos_facturados (id_factura, id_producto, cantidad, precio_venta)  
268 VALUES (1, 1, 2, 1000);  
269  
270 -- Verifica stock después  
271 • SELECT stock FROM productos WHERE id_producto = 1;
```



stock
70

6.2 Validar stock antes de facturar

```
DELIMITER //
```

```
CREATE TRIGGER tr_validar_stock
```

```
BEFORE INSERT ON productos_facturados
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE cantidad_disponible INT;
```

```
    SELECT stock INTO cantidad_disponible
```

```
    FROM productos
```

```
    WHERE id_producto = NEW.id_producto;
```

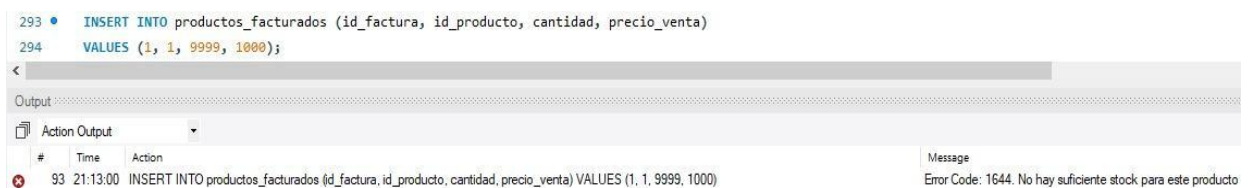
```
    IF cantidad_disponible < NEW.cantidad THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'No hay suficiente stock para este producto';
```

```
    END IF;
```

```
END //
```



6.3 Registrar bitácora de cambios de usuarios

```
CREATE TABLE bitacora_usuarios (
```

```
    id_log INT AUTO_INCREMENT PRIMARY KEY,
```

```
    id_usuario INT,    accion VARCHAR(50),
```

```
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
DELIMITER //
```

```
CREATE TRIGGER tr_log_update_usuario
AFTER UPDATE ON usuarios
FOR EACH ROW
BEGIN
    INSERT INTO bitacora_usuarios (id_usuario, accion)
    VALUES (NEW.id_usuario, 'Usuario actualizado');
END //
DELIMITER ;
```

```
311 • SELECT * FROM bitacora_usuarios;
312
313 • UPDATE usuarios
314     SET idioma = 'Español'
315     WHERE id_usuario = 1;
```



The screenshot shows a database interface with a SQL query editor and a result grid. The query is: `SELECT * FROM bitacora_usuarios;`. The result grid displays the following data:

	id_log	id_usuario	accion	fecha
▶	1	1	Usuario actualizado	2025-09-12 18:24:19
*	NULL	NULL	NULL	NULL

6.4 Calcular automáticamente precio de venta en productos_facturados

```
DELIMITER //
```

```
CREATE TRIGGER tr_set_precio_venta
BEFORE INSERT ON productos_facturados
```

FOR EACH ROW

BEGIN show

triggers;

IF NEW.precio_venta IS NULL OR NEW.precio_venta = 0 THEN

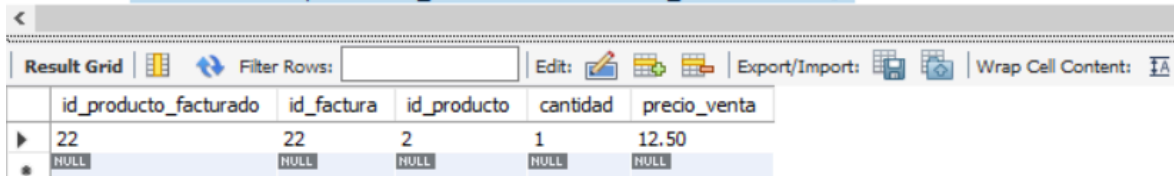
SET NEW.precio_venta = (SELECT precio_unitario FROM productos WHERE id_producto = NEW.id_producto);

END IF;

END//

DELIMITER ;

```
669 -- Insertar producto sin precio_venta
670 • INSERT INTO productos_facturados (id_factura, id_producto, cantidad, precio_venta)
671 VALUES (22, 2, 1, 0);
672
673 -- Verificar que el precio se haya puesto automáticamente
674 • SELECT * FROM productos_facturados WHERE id_factura = 22;
```



	id_producto_facturado	id_factura	id_producto	cantidad	precio_venta
▶	22	22	2	1	12.50
•	NULL	NULL	NULL	NULL	NULL

6.5 Verificación de longitud mínima de contraseñas en usuarios

DELIMITER //

CREATE TRIGGER tr_validar_password_length

BEFORE INSERT ON usuarios

FOR EACH ROW

BEGIN

IF CHAR_LENGTH(NEW.contrasena) < 8 THEN

SIGNAL SQLSTATE '45000'

```

        SET MESSAGE_TEXT = 'La contraseña debe tener al menos 8 caracteres';
    END IF;

    END//

```

```

DELIMITER ;

```



PROCEDIMIENTOS

7.1 Obtener facturas de un cliente Devuelve todas las facturas de un cliente específico

```

DELIMITER //

CREATE PROCEDURE sp_facturas_cliente(IN p_id_cliente INT)

BEGIN

    SELECT f.id_factura, f.numero_factura, f.fecha_factura, f.total, f.tipo_pago

    FROM facturas f

    WHERE f.id_cliente = p_id_cliente

    ORDER BY f.fecha_factura DESC;

END //

DELIMITER ;

```

```

426 DELIMITER //
427 • CREATE PROCEDURE sp_facturas_cliente(IN p_id_cliente INT)
428 BEGIN
429     SELECT f.id_factura, f.numero_factura, f.fecha_factura, f.total, f.tipo_pago
430     FROM facturas f
431     WHERE f.id_cliente = p_id_cliente
432     ORDER BY f.fecha_factura DESC;
433 END //
434 DELIMITER ;
435 • CALL sp_facturas_cliente(1);
436

```

Result Grid

	id_factura	numero_factura	fecha_factura	total	tipo_pago
▶	1	100001	2025-09-11 19:42:35	150000.50	efectivo

7.2 Procedimiento para actualizar el stock de un producto

DELIMITER //

CREATE PROCEDURE sp_actualizar_stock(IN p_id_producto INT, IN p_cantidad INT)

BEGIN

UPDATE productos

SET stock = stock + p_cantidad

WHERE id_producto = p_id_producto;

END //

DELIMITER ;

```

442 DELIMITER //
443 • CREATE PROCEDURE sp_actualizar_stock(IN p_id_producto INT, IN p_cantidad INT)
444 BEGIN
445     UPDATE productos
446     SET stock = stock + p_cantidad
447     WHERE id_producto = p_id_producto;
448 END //
449 DELIMITER ;
450 • CALL sp_actualizar_stock(1, 10);

```

Result Grid

	id_producto	stock
▶	1	70
•	NULL	NULL

[illegible]

7.4 Lista todos los productos activos/inactivos de una categoría específica

```
DELIMITER //
```

```
CREATE PROCEDURE sp_productos_por_categoria(IN p_id_categoria INT)
```

```
BEGIN
```

```
    SELECT p.id_producto, p.nombre_producto, p.precio_unitario, p.stock, p.estado
```

```
    FROM productos p
```

```
    WHERE p.id_categoria = p_id_categoria;
```

```
END //
```

```
DELIMITER ;
```

```
497 DELIMITER //
498 • CREATE PROCEDURE sp_productos_por_categoria(IN p_id_categoria INT)
499   BEGIN
500     SELECT p.id_producto, p.nombre_producto, p.precio_unitario, p.stock, p.estado
501     FROM productos p
502     WHERE p.id_categoria = p_id_categoria;
503   END //
504 DELIMITER ;
505 • CALL sp_productos_por_categoria(1);
506
```

Result Grid					
Filter Rows:		Export:		Wrap Cell Content:	
	id_producto	nombre_producto	precio_unitario	stock	estado
▶	1	Croquetas para perros adulto 5kg	35.99	70	activo
	3	Snack dental para perros 150g	8.99	75	activo
	10	Comida para peces perros 200g	5.99	60	activo
	15	Leche para cachorros 1L	20.00	30	activo

7.5 Generar detalle de factura -- Devuelve información completa de una factura: -- datos del cliente, productos facturados, precios y subtotales

```
DELIMITER //
```

```
CREATE PROCEDURE sp_factura_detalle(IN p_id_factura INT)
```

```
BEGIN
```

```
    SELECT f.numero_factura, f.fecha_factura,
```



```

        CONCAT(c.primer_nombre, ' ', c.primer_apellido) AS cliente,
        p.nombre_producto, pf.cantidad, pf.precio_venta,
        (pf.cantidad * pf.precio_venta) AS subtotal
FROM facturas f
INNER JOIN clientes c ON f.id_cliente = c.id_cliente
INNER JOIN productos_facturados pf ON f.id_factura = pf.id_factura
INNER JOIN productos p ON pf.id_producto = p.id_producto
WHERE f.id_factura = p_id_factura;
END //
DELIMITER ;

```

```

513 DELIMITER //
514 • CREATE PROCEDURE sp_factura_detalle(IN p_id_factura INT)
515 BEGIN
516     SELECT f.numero_factura, f.fecha_factura,
517           CONCAT(c.primer_nombre, ' ', c.primer_apellido) AS cliente,
518           p.nombre_producto, pf.cantidad, pf.precio_venta,
519           (pf.cantidad * pf.precio_venta) AS subtotal
520     FROM facturas f
521     INNER JOIN clientes c ON f.id_cliente = c.id_cliente
522     INNER JOIN productos_facturados pf ON f.id_factura = pf.id_factura
523     INNER JOIN productos p ON pf.id_producto = p.id_producto
524     WHERE f.id_factura = p_id_factura;
525 END //
526 DELIMITER ;
527 • CALL sp_factura_detalle(1);

```

numero_factura	fecha_factura	cliente	nombre_producto	cantidad	precio_venta	subtotal
100001	2025-09-11 19:42:35	Juan Pérez	Croquetas para perros adulto 5kg	2	15000.00	30000.00

FUNCIONES

8.1 Obtener nombre completo del cliente

-- Devuelve primer_nombre + primer_apellido

```
DELIMITER //
```

```
CREATE FUNCTION fn_nombre_cliente(p_id_cliente INT)
```

```

RETURNS VARCHAR(255)

DETERMINISTIC

BEGIN

    DECLARE nombre_completo VARCHAR(255);

    SELECT CONCAT(primer_nombre, ' ', primer_apellido)

    INTO nombre_completo

    FROM clientes

    WHERE id_cliente = p_id_cliente;

    RETURN nombre_completo;

END //

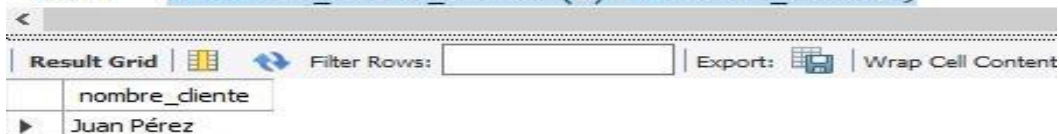
DELIMITER ;

```

```

532 DELIMITER //
533 • CREATE FUNCTION fn_nombre_cliente(p_id_cliente INT)
534 RETURNS VARCHAR(255)
535 DETERMINISTIC
536 BEGIN
537     DECLARE nombre_completo VARCHAR(255);
538
539     SELECT CONCAT(primer_nombre, ' ', primer_apellido)
540     INTO nombre_completo
541     FROM clientes
542     WHERE id_cliente = p_id_cliente;
543
544     RETURN nombre_completo;
545 END //
546 DELIMITER ;
547
548 • SELECT fn_nombre_cliente(1) AS nombre_cliente;

```



The screenshot shows a SQL IDE interface. The top part displays the SQL code for creating a function and calling it. The bottom part shows the 'Result Grid' with the following data:

nombre_cliente
Juan Pérez

```

-- Devuelve la suma de todas sus facturas

DELIMITER //

CREATE FUNCTION fn_total_facturado_cliente(p_id_cliente INT)

RETURNS DECIMAL(12,2)

```

DETERMINISTIC

BEGIN

DECLARE total_cliente DECIMAL(12,2);

SELECT IFNULL(SUM(total),0)

INTO total_cliente

FROM facturas

WHERE id_cliente = p_id_cliente;

RETURN total_cliente;

END //

DELIMITER ;

```
554 DELIMITER //
555 • CREATE FUNCTION fn_total_facturado_cliente(p_id_cliente INT)
556 RETURNS DECIMAL(12,2)
557 DETERMINISTIC
558 BEGIN
559     DECLARE total_cliente DECIMAL(12,2);
560     SELECT IFNULL(SUM(total),0)
561     INTO total_cliente
562     FROM facturas
563     WHERE id_cliente = p_id_cliente;
564     RETURN total_cliente;
565 END //
566 DELIMITER ;
567 • SELECT fn_total_facturado_cliente(1) AS total_facturado;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
total_facturado				
▶		150000.50		

8.3 Calcular stock valorizado de un producto

-- Devuelve stock * precio_unitario

DELIMITER //

CREATE FUNCTION fn_stock_valorizado(p_id_producto INT)

```

RETURNS DECIMAL(12,2)

DETERMINISTIC

BEGIN

    DECLARE valor DECIMAL(12,2);

    SELECT stock * precio_unitario

    INTO valor

    FROM productos

    WHERE id_producto = p_id_producto;

    RETURN valor;

END //

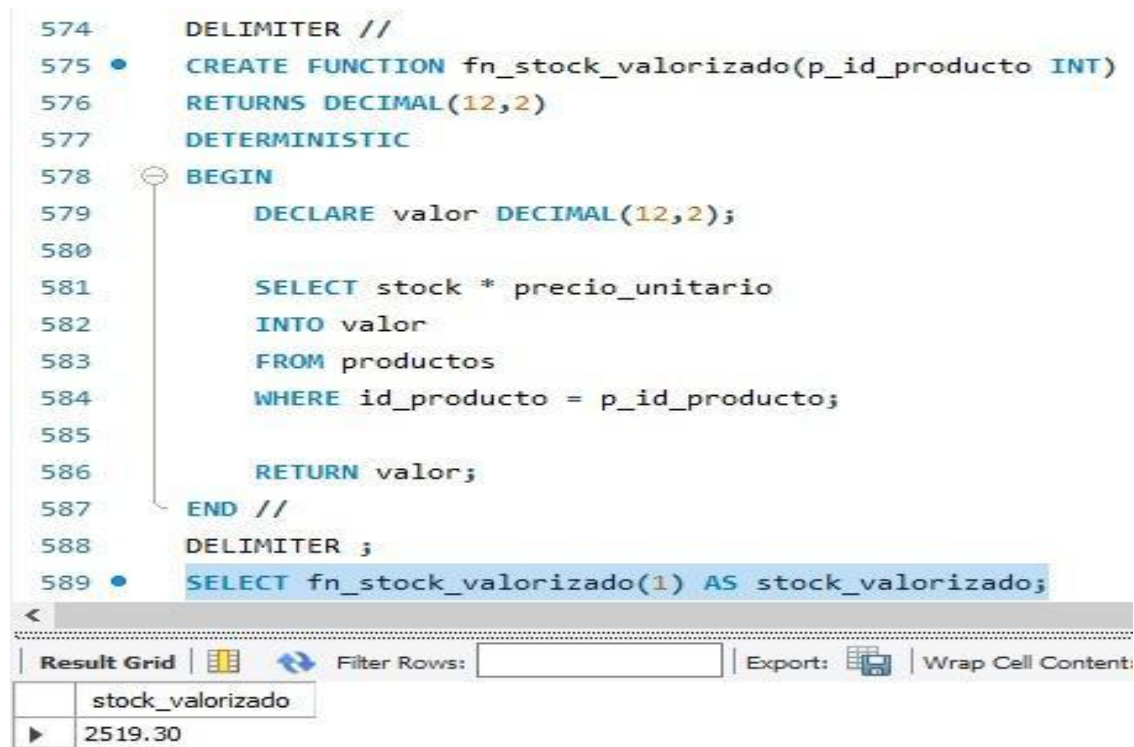
DELIMITER ;

```

```

574 DELIMITER //
575 • CREATE FUNCTION fn_stock_valorizado(p_id_producto INT)
576 RETURNS DECIMAL(12,2)
577 DETERMINISTIC
578 BEGIN
579     DECLARE valor DECIMAL(12,2);
580
581     SELECT stock * precio_unitario
582     INTO valor
583     FROM productos
584     WHERE id_producto = p_id_producto;
585
586     RETURN valor;
587 END //
588 DELIMITER ;
589 • SELECT fn_stock_valorizado(1) AS stock_valorizado;

```



stock_valorizado
2519.30

8.4 Obtener rol de un usuario

-- Devuelve el rol principal asociado a un usuario

```

DELIMITER //

CREATE FUNCTION fn_rol_usuario(p_id_usuario INT)

RETURNS VARCHAR(100)

DETERMINISTIC

BEGIN

    DECLARE rol_nombre VARCHAR(100);

    SELECT r.nombre_rol

    INTO rol_nombre

    FROM roles r

    INNER JOIN usuario_rol ur ON r.id_rol = ur.id_rol

    WHERE ur.id_usuario = p_id_usuario

    LIMIT 1;

    RETURN rol_nombre;

END //

DELIMITER ;

```

```

596     DELIMITER //
597 •   CREATE FUNCTION fn_rol_usuario(p_id_usuario INT)
598     RETURNS VARCHAR(100)
599     DETERMINISTIC
600     BEGIN
601         DECLARE rol_nombre VARCHAR(100);
602
603         SELECT r.nombre_rol
604         INTO rol_nombre
605         FROM roles r
606         INNER JOIN usuario_rol ur ON r.id_rol = ur.id_rol
607         WHERE ur.id_usuario = p_id_usuario
608         LIMIT 1;
609
610         RETURN rol_nombre;
611     END //
612     DELIMITER ;
613 •   SELECT fn_rol_usuario(1) AS rol_usuario;

```

rol_usuario
Administrador

8.5 Contar productos por categoría

-- Devuelve cuántos productos hay en una categoría

```

DELIMITER //

CREATE FUNCTION fn_total_productos_categoria(p_id_categoria INT)

RETURNS INT

DETERMINISTIC

BEGIN

    DECLARE total INT;

    SELECT COUNT(*)

    INTO total

    FROM productos

    WHERE id_categoria = p_id_categoria;

    RETURN total;

END //

use petshop;

DELIMITER ;

```

```

619 DELIMITER //
620 • CREATE FUNCTION fn_total_productos_categoria(p_id_categoria INT)
621 RETURNS INT
622 DETERMINISTIC
623 BEGIN
624     DECLARE total INT;
625     SELECT COUNT(*)
626     INTO total
627     FROM productos
628     WHERE id_categoria = p_id_categoria;
629     RETURN total;
630 END //
631
632 use petshop;
633 DELIMITER ;
634 • SELECT fn_total_productos_categoria(1) AS total_productos;

```

Result Grid

total_productos
4

Filter Rows: Export: Wrap Cell Content:

ENCRIPTACIÓN.

9.1 encriptar contraseñas función

```
DELIMITER //
```

```
CREATE FUNCTION fn_encriptar(p_pass VARCHAR(255))
```

```
RETURNS VARCHAR(255)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    RETURN SHA2(p_pass,256);
```

```
END //
```

```
DELIMITER ;
```

```
INSERT INTO usuarios (contrasena, correo_electronico)
```

```
VALUES (fn_encriptar('MiPassword123'), 'user@correo.com');
```

```
ALTER TABLE usuarios
```

```
MODIFY contrasena VARCHAR(256) NOT NULL;
```

```
#Cuando un usuario se cree, la contraseña se convertirá automáticamente en hash:
```

```
DELIMITER //
```

9.2 Procedimiento para registrar usuario con contraseña encriptada

```
CREATE PROCEDURE RegistrarUsuario(
```

```
    IN p_username VARCHAR(100),
```

```
    IN p_password VARCHAR(255),
```

```
    IN p_email VARCHAR(150),
```

```
    IN p_nombre VARCHAR(100) )
```

```
BEGIN
```

```
    DECLARE v_salt VARCHAR(32);
```

```
    DECLARE v_encrypted VARCHAR(64);
```

```
    DECLARE v_user_exists INT DEFAULT 0;
```

```
    DECLARE i INT DEFAULT 0;
```

```

-- Verificar si el usuario ya existe

SELECT COUNT(*) INTO v_user_exists

FROM usuarios

WHERE username = p_username OR email = p_email;

IF v_user_exists > 0 THEN

    SELECT "ERROR: USUARIO O EMAIL YA EXISTE" AS mensaje, FALSE AS exito;

ELSE

    SET v_salt = SUBSTRING(MD5(CONCAT(RAND(), NOW())), 1, 32);

    -- Hash inicial con salt

    SET v_encrypted = SHA2(CONCAT(v_salt, p_password), 256);

    -- Simulación de múltiples rondas de hash (ajustable)

    SET i = 0;

    WHILE i < 10 DO

        SET v_encrypted = SHA2(CONCAT(v_salt, v_encrypted), 256);

        SET i = i + 1;

    END WHILE;

    -- Insertar usuario

    INSERT INTO usuarios (username, password_hash, salt, email, nombre)

    VALUES (p_username, v_encrypted, v_salt, p_email, p_nombre);

    SELECT "USUARIO REGISTRADO EXITOSAMENTE" AS mensaje, TRUE AS exito;

END IF;

END //

DELIMITER ; select *

from usuarios;

```