

**NOMBRE:** ROGER FRANCO CERVANTES

**ASIGNATURA:** METODOS DE GESTION, TRATAMIENTO Y ANALISIS DE DATOS

**ACTIVIDAD 1 “MYSQL Y POWER BI” jardinería**

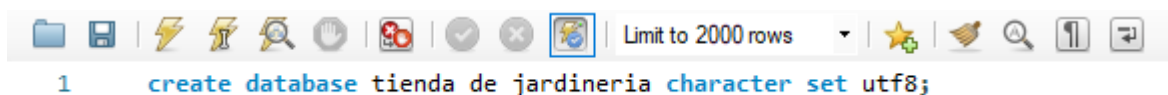
- **Responder a la pregunta: ¿Qué representa la tabla detalle\_pedido?**

La tabla detalle\_pedido representa el detalle de los diferentes pedidos realizados, por lo tanto, contiene toda la información clave de la tienda de jardinería, mediante esa tabla podemos obtener la información de que ha comprado cada cliente, cuales productos y la cantidad que ha comprado, lo que permite realizar un análisis de cual está siendo el Core del negocio, y a su vez diseñar estrategias que vayan acorde a los datos que nos suministra.

Adicional a esto es la tabla que contiene columnas donde se encuentran los datos que unen a las diferentes tablas, por lo tanto, es de vital importancia para todas las consultas que vayamos a realizar.

- **Código SQL correspondiente a la creación de la base de datos y a las diferentes tablas**

Se crea la base de datos llamada “tienda\_de\_jardineria”



Luego se crean las diferentes bases de datos

- **cliente**

```
1 • create table cliente (  
2     id int auto_increment primary key,  
3     nombre_cliente varchar(50) not null,  
4     nombre_contacto varchar(50) not null,  
5     apellido_contacto varchar(50) not null,  
6     telefono varchar(50),  
7     fax varchar(50) not null,  
8     linea_direccion1 varchar(50) not null,  
9     linea_direccion2 varchar(50),  
10    ciudad varchar(50),  
11    region varchar(50),  
12    pais varchar(50) not null,  
13    codigo_postal varchar(50) not null,  
14    codigo_empleado_rep_ventas varchar(50) not null,  
15    limite_credito varchar(50) not null  
16 );
```

- **detalle\_pedido**

```
1 • create table detalle_pedido (  
2     id int auto_increment primary key,  
3     codigo_pedido varchar(50) not null,  
4     codigo_producto varchar(50) not null,  
5     cantidad varchar(50) not null,  
6     precio_unidad varchar(50) not null,  
7     numero_linea varchar(50) not null,  
8 ✖ );
```

- **empleado**

```

1 • ○ create table empleado (
2     id int auto_increment primary key,
3     nombre varchar(50) not null,
4     apellido1 varchar(50) not null,
5     apellido2 varchar(50),
6     extension varchar(50) not null,
7     email varchar(50) not null,
8     codigo_oficina varchar(50) not null,
9     puesto varchar (50)
10 );

```

- **gama\_producto**

```

1 • ○ create table gama_producto (
2     id int auto_increment primary key,
3     gama varchar(50) not null,
4     descripcion_texto varchar(50) not null
5 );

```

- **oficina**

```

1 • ○ create table oficina (
2     id int auto_increment primary key,
3     codigo_oficina varchar(50) not null,
4     ciudad varchar(50) not null,
5     pais varchar(50) not null,
6     region varchar(50) not null,
7     codigo_postal varchar(50) not null,
8     telefono varchar(50) not null,
9     linea_direccion1 varchar(50) not null,
10    linea_direccion2 varchar(50)
11 );

```

- **pago**

```
1 • ○ create table pago (  
2     id int auto_increment primary key,  
3     codigo_cliente varchar(50) not null,  
4     forma_pago varchar(50) not null,  
5     id_transaccion varchar(50) not null,  
6     fecha_pago varchar(50) not null,  
7     total varchar(50) not null  
8 );
```

- **pedido**

```
1 • ○ create table pedido (  
2     id int auto_increment primary key,  
3     codigo_pedido varchar(50) not null,  
4     fecha_pedido varchar(50) not null,  
5     fecha_esperada varchar(50) not null,  
6     fecha_entrega varchar(50),  
7     estado varchar(50) not null,  
8     comentarios text,  
9     codigo_cliente varchar(50) not null  
10 );
```

- **producto**

```
1 • ○ create table producto (  
2   id int auto_increment primary key,  
3   codigo_producto varchar(50) not null,  
4   nombre varchar(50) not null,  
5   gama varchar(50) not null,  
6   dimensiones varchar(50),  
7   proveedor varchar(50) not null,  
8   descripcion varchar(50),  
9   cantidad_en_stock varchar(50) not null,  
10  precio_venta varchar(50) not null,  
11  precio_proveedor varchar(50) not null  
12  );
```

- Código SQL y resultado de las siguientes consultas (se puede entregar como pantallazo, siempre que se vea el código SQL y el resultado):

1. Número de empleados.

```

1 • use tienda_de_jardineria;
2
3 • select count(*) val_count from empleado
4

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	val_count			
▶	31			

## 2. Listado con la ciudad y el teléfono de las oficinas de España.

```

1 • use tienda_de_jardineria;
2
3 • select ciudad, telefono
4     from oficina
5     where pais="España"
6
7
8

```

Result Grid			Filter Rows:	Export
	ciudad	telefono		
▶	Barcelona	+34 93 3561182		
	Madrid	+34 91 7514487		
	Talavera de la Reina	+34 925 867231		

## 3. Listado de todos los clientes franceses.

```

1 • use tienda_de_jardineria;
2
3 • select nombre_cliente, pais
4   from cliente
5  where pais = 'France'

```

Result Grid			Filter Rows:
	nombre_cliente	pais	
▶	france telecom	France	
	Musée du Louvre	France	

**4. ¿Cuántos pedidos hay en cada estado? Ordena el resultado de forma descendente por el número de pedidos.**

```

1   use tienda_de_jardineria;
2
3 • select estado, count(1)
4   from pedido
5  group by estado

```

Result Grid			Filter Rows:
	estado	count(1)	
▶	Entregado	61	
	Pendiente	30	
	Rechazado	24	

5. Calcula la suma de la cantidad total de todos los productos que aparecen en cada uno de los pedidos.

```
1 • use tienda_de_jardineria;
2
3 • select codigo_pedido, count(*) as codigo_producto
4   from detalle_pedido
5  group by codigo_pedido
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	codigo_pedido	codigo_producto		
▶	1	5		
	2	7		
	3	6		
	4	8		
	8	3		

6. Número de clientes por país.

```
1 • use tienda_de_jardineria;
2
3 • select pais, count(*) as cliente
4   from cliente
5  group by pais
```

Result Grid		Filter Rows:	Export:
	pais	cliente	
	USA	4	
	Spain	27	
	France	2	
▶	Australia	2	
	United Kingdom	1	



## 7. Producto que tiene más unidades de stock

```
1 • use tienda_de_jardineria;  
2  
3 • select nombre, cantidad_en_stock  
4   from producto  
5   where cantidad_en_stock > 390
```

Result Grid	Filter Rows:	Export:	Wrap
nombre	cantidad_en_stock		
Rosal copa	400		
Membrillero Gigante de Wranja	400		
Nogal Común	400		
Parra Uva de Mesa	400		
▶ Nectarina	400		

## 8. Listado con el nombre de los representantes de ventas y el número de clientes que atiende cada uno.

```
1 • select a.nombre, count(b.id) as numero_clientes  
2   from cliente b  
3  join empleado a on b.codigo_empleado_rep_ventas = a.codigo_empleado  
4  group by a.nombre
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre	numero_clientes		
▶ Walter Santiago	2		
Lorena	2		
Mariano	4		
Emmanuel	5		
Julian	5		

## 9. Listado de todos los pedidos realizados en el 2009.

```
1 • select *
2   from pedido
3  where year (fecha_pedido) = 2009
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id	codigo_pedido	fecha_pedido	fecha_esperada	fecha_entrega	estado	comentarios	codigo_cliente
▶	4	4	2009-01-20	2009-01-26		Pendiente		5
	7	10	2009-01-15	2009-01-20		Pendiente	El cliente llama para confirmar la fecha - Espera...	3
	8	11	2009-01-20	2009-01-27		Pendiente	El cliente requiere que el pedido se le entregue ...	1
	9	12	2009-01-22	2009-01-27		Pendiente	El cliente requiere que el pedido se le entregue ...	1
	10	13	2009-01-12	2009-01-14	2009-01-15	Entregado		7

## 10. Listado sin duplicados de todas las formas de pago diferentes.

```
1 • select distinct forma_pago
2   from pago;
```

Result Grid    Filter Rows: <input type="text"/>	
	forma_pago
▶	PayPal
	Transferencia
	Cheque

## 11. Listado de la cantidad que se ha vendido de cada producto.

```
1 • select codigo_producto, sum(cantidad) as total_vendido
2   from detalle_pedido
3  group by codigo_producto
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	codigo_producto	total_vendido		
▶	FR-67	285		
	OR-127	40		
	OR-141	45		
	OR-241	25		
	OR-99	38		

## 12. Devuelve un listado con todos los clientes que sean de la ciudad de Madrid y cuyo representante de ventas tenga el código de empleado

```
1 • SELECT *
2   FROM cliente
3  WHERE ciudad = 'Madrid'
4     AND codigo_empleado_rep_ventas IN (11, 30);
```

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content: [IA](#)

	id	nombre_cliente	nombre_contacto	apellido_contacto	telefono	fax	linea_direccion1	linea_direccion2	ciudad	region
▶	6	Beragua	Jose	Bermejo	654987321	916549872	C/pintor segundo	Getafe	Madrid	Madrid
	7	Club Golf Puerta del hierro	Paco	Lopez	62456810	919535678	C/sinesio delgado	Madrid	Madrid	Madrid
	8	Naturagua	Guillermo	Rengifo	689234750	916428956	C/majadahonda	Boadilla	Madrid	Madrid
	9	DaraDistribuciones	David	Serrano	675598001	916421756	C/azores	Fuenlabrada	Madrid	Madrid

### 13. Listado de los productos que nunca han aparecido en un pedido.

```
1 • SELECT codigo_producto
2 FROM producto
3 WHERE codigo_producto NOT IN (SELECT DISTINCT codigo_producto FROM detalle_pedido);
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	codigo_producto			
▶	FR-19			
	FR-20			
	FR-49			
	FR-5			
	OR-001			

### 14. Listado de las direcciones de las oficinas que tengan clientes en Fuenlabrada.

```
1 • SELECT b.linea_direccion1, b.linea_direccion2
2 FROM oficina b
3 JOIN cliente a ON b.codigo_postal = a.codigo_postal
4 WHERE a.ciudad = 'Fuenlabrada';
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	linea_direccion1	linea_direccion2		

**\* No hay ninguna oficina que este ubicada en Fuenlabrada**

**15. Listado de las diferentes gamas de producto que ha comprado cada cliente.**

```
1 • SELECT dp.codigo_producto, gp.gama
2 FROM detalle_pedido dp
3 JOIN producto p ON dp.codigo_producto = p.codigo_producto
4 JOIN gama_producto gp ON p.gama = gp.gama;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	codigo_producto	gama			
▶	OR-141	Ornamentales			
	OR-241	Ornamentales			
	OR-141	Ornamentales			
	OR-159	Ornamentales			
	OR-227	Ornamentales			