



## ÍNDICE

<b>ÍNDICE</b>	<b>1</b>
<b>PROYECTO TECNOFULL</b>	<b>2</b>
<b>OBJETIVO DE ESTA BASE DE DATOS</b>	<b>2</b>
<b>LISTADO Y DESCRIPCIÓN DE TABLAS</b>	<b>3</b>
<b>DESCRIPCIÓN MODELO ENTIDAD-RELACIÓN</b>	<b>8</b>
ENTIDADES	8
RELACIONES	8
<b>CREACIÓN DE NUEVAS TABLAS</b>	<b>14</b>
<b>CREACIÓN DE VISTAS</b>	<b>15</b>
DESCRIPCIÓN DE VISTAS	15
<b>CREACIÓN DE FUNCIONES</b>	<b>17</b>
<b>CREACIÓN DE STORED PROCEDURES</b>	<b>18</b>
<b>CREACIÓN DE TRIGGERS</b>	<b>18</b>
<b>DML (DATA MANIPULATION LANGUAGE)</b>	<b>19</b>
<b>DCL (DATA CONTROL LANGUAGE)</b>	<b>19</b>
<b>TABLA PARA GRÁFICOS</b>	<b>20</b>

# PROYECTO TECNOFULL

(Pyme de soluciones tecnológicas)

El modelo entidad-relación (ER) de TECNOFULL representa la estructura de datos de una empresa de venta de insumos de computadoras y celulares, así como de servicios de reparación y asesoramiento remoto. Este modelo consta de 15 tablas que capturan las entidades principales de la empresa y sus relaciones.

Las tablas permiten realizar consultas y análisis sobre las relaciones entre los distintos elementos de la empresa, así como realizar pronósticos sobre el comportamiento de la empresa a medida que las entidades interactúan. El modelo entidad-relación también permite identificar potenciales problemas y desafíos de la empresa.

## OBJETIVO DE ESTA BASE DE DATOS

1. Organización de datos: Esta base de datos me permite organizar y almacenar de manera eficiente toda la información relacionada con la empresa TecnoFull, como clientes, empleados, productos, servicios, proveedores, entre otros. También puede ayudar a los empleados a trabajar más eficientemente y acelerar el proceso de negocio.
2. Acceso rápido a la información: Con esta base de datos, es fácil y rápido acceder a la información necesaria en cualquier momento. Esto permite al personal de TecnoFull realizar consultas, generar informes y tomar decisiones fundamentadas de manera más eficiente. Esto también ayuda a optimizar el tiempo de respuesta de los clientes, lo que les brinda una mejor experiencia. Además permite aplicar mejores métodos de marketing para aumentar el éxito de sus productos.
3. Integridad de los datos: Al utilizar un sistema como MySQL, se pueden aplicar restricciones de integridad referencial (que es garantizar la consistencia y la validez de los datos almacenados), y otros mecanismos para garantizar la coherencia y la precisión de los datos almacenados. Esto puede ser útil para proteger contra la introducción de datos no autorizados o incorrectos, así como para proteger contra el robo de datos. También permite una mayor seguridad de los datos almacenados.
4. Análisis y reportes: Cuando tenga los datos almacenados en mi base de datos, se pueden realizar análisis detallados y generar informes que ayuden a

comprender mejor el rendimiento del negocio, identificar tendencias, oportunidades y áreas de mejora. Los informes que resultan se deberían analizar de manera rápida y eficiente para tomar decisiones de negocio porque podemos detectar problemas y realizar mejoras en el sistema.

5. **Facilita la toma de decisiones:** Si tenemos acceso a datos precisos y actualizados, las personas a cargo y propietarios de TecnoFull pueden tomar decisiones informadas y estratégicas que impulsen el crecimiento y la rentabilidad de la empresa. Esto permite responder rápidamente al mercado, haciendo una mayor competitividad y ahorrar costos. Además le da a los empleados una mayor confianza en sus decisiones y una mayor motivación para seguir trabajando en la empresa.

## LISTADO Y DESCRIPCIÓN DE TABLAS

1. **Cliente:** Registra información sobre los clientes que realizan compras y solicitan servicios.

**Descripción:** Almacena la información de los clientes.

**Campos:**

- id\_cliente (PK, AUTOINCREMENT): Identificador único del cliente.
- nombre: Nombre del cliente.
- dirección: Dirección del cliente.
- teléfono: Teléfono del cliente.

2. **Empleado:** Almacena detalles de los empleados que trabajan en la empresa.

**Descripción:** Contiene información sobre los empleados de la empresa.

**Campos:**

- id\_empleado (PK, AUTOINCREMENT): Identificador único del empleado.
- nombre: Nombre del empleado.
- cargo: Cargo del empleado.
- sueldo: Sueldo del empleado.
- id\_sucursal (FK): Identificador de la sucursal donde trabaja.

3. **Proveedor:** Contiene información sobre los proveedores que suministran productos a la empresa.

**Descripción:** Almacena los datos de los proveedores de los productos.

**Campos:**

- id\_proveedor (PK, AUTOINCREMENT): Identificador único del proveedor.
- nombre: Nombre del proveedor.
- contacto: Información de contacto del proveedor.
- teléfono: Teléfono del proveedor.

4. **Transporte:** Registra los detalles de los servicios de transporte utilizados para envíos a domicilio.

**Descripción:** Almacena información sobre los medios de transporte utilizados para los envíos a domicilio.

**Campos:**

- id\_transporte (PK, AUTOINCREMENT): Identificador único del transporte.
- tipo: Tipo de transporte.

5. **Producto:** Almacena información detallada sobre los productos vendidos por la empresa.

**Descripción:** Contiene información sobre los productos disponibles para la venta.

**Campos:**

- id\_producto (PK, AUTOINCREMENT): Identificador único del producto.
- nombre: Nombre del producto.
- descripción: Descripción del producto.
- precio: Precio del producto.
- id\_categoria (FK): Identificador de la categoría del producto.
- id\_marca (FK): Identificador de la marca del producto.

6. **Servicio:** Registra los diferentes servicios ofrecidos por la empresa, incluyendo reparación en el local y asesoramiento remoto.

**Descripción:** Registra los servicios ofrecidos por la empresa.

**Campos:**

- id\_servicio (PK, AUTOINCREMENT): Identificador único del servicio.
- tipo: Tipo de servicio (reparación local o asesoramiento remoto).

7. **Sucursal:** Contiene información sobre las sucursales de la empresa.

**Descripción:** Contiene información sobre las diferentes sucursales de la empresa.

**Campos:**

- id\_sucursal (PK, AUTOINCREMENT): Identificador único de la sucursal.

- dirección: Dirección de la sucursal.
- teléfono: Teléfono de la sucursal.
- id\_encargado (FK): Identificador del encargado de la sucursal.

8. **Dueño:** Registra los datos del dueño de la empresa.

**Descripción:** Información sobre el dueño de la empresa.

**Campos:**

- id\_dueño (PK, AUTOINCREMENT): Identificador único del dueño.
- nombre: Nombre del dueño.
- contacto: Información de contacto del dueño.

**Aclaración:** Aunque haya un único dueño en cada franquicia, el campo autoincrementable permite gestionar a los dueños de múltiples franquicias de forma ordenada y evita conflictos al agregar registros.

9. **Encargado:** Almacena información sobre los encargados asignados a cada sucursal.

**Descripción:** Información sobre los encargados de las sucursales.

**Campos:**

- id\_encargado (PK, AUTOINCREMENT): Identificador único del encargado.
- nombre: Nombre del encargado.
- id\_sucursal (FK): Identificador de la sucursal donde es encargado.

10. **Categoría de Producto:** Contiene las categorías en las que se clasifican los productos.

**Descripción:** Categorías a las que pertenecen los productos.

**Campos:**

- id\_categoria (PK, AUTOINCREMENT): Identificador único de la categoría.
- nombre\_categoria: Nombre de la categoría.

**Descripción:** Categorías a las que pertenecen los productos.

**Campos:**

- id\_categoria (PK, AUTOINCREMENT): Identificador único de la categoría.
- nombre\_categoria: Nombre de la categoría.

11. **Marca:** Registra las marcas de los productos.

**Descripción:** Marcas de los productos.

**Campos:**

- id\_marca (PK, AUTOINCREMENT): Identificador único de la marca.
- nombre\_marca: Nombre de la marca.

12. **Modelo:** Almacena los modelos específicos de los productos.

**Descripción:** Modelos específicos de los productos.

**Campos:**

- id\_modelo (PK, AUTOINCREMENT): Identificador único del modelo.
- nombre\_modelo: Nombre del modelo.
- id\_marca (FK): Identificador de la marca a la que pertenece el modelo.

13. **Detalle de Compra:** Contiene los detalles de cada compra realizada por un cliente.

**Descripción:** Registra los detalles de las compras realizadas por los clientes, incluyendo productos y cantidades.

**Campos:**

- id\_detalle (PK, AUTOINCREMENT): Identificador único del detalle.
- id\_compra (FK): Identificador de la compra relacionada.
- id\_producto (FK): Identificador del producto comprado.
- cantidad: Cantidad del producto comprado.
- precio\_unitario: Precio unitario del producto.

14. **Detalle de Servicio:** Registra los detalles de cada servicio solicitado por un cliente.

**Descripción:** Relación entre clientes y servicios utilizados.

**Campos:**

- id\_detalle\_servicio (PK, AUTOINCREMENT): Identificador único de la relación.
- id\_cliente (FK): Identificador del cliente.
- id\_servicio (FK): Identificador del servicio utilizado.
- fecha\_servicio: Fecha del servicio.

15. **Detalle de Envío:** Contiene los detalles de cada envío realizado por la empresa.

**Descripción:** Información sobre los envíos realizados a los clientes.

**Campos:**

- id\_detalle\_envio (PK, AUTOINCREMENT): Identificador único del envío.
- id\_compra (FK): Identificador de la compra relacionada.
- id\_transporte (FK): Identificador del transporte utilizado.
- fecha\_envio: Fecha del envío.
- direccion\_envio: Dirección de envío.

# DESCRIPCIÓN MODELO ENTIDAD-RELACIÓN

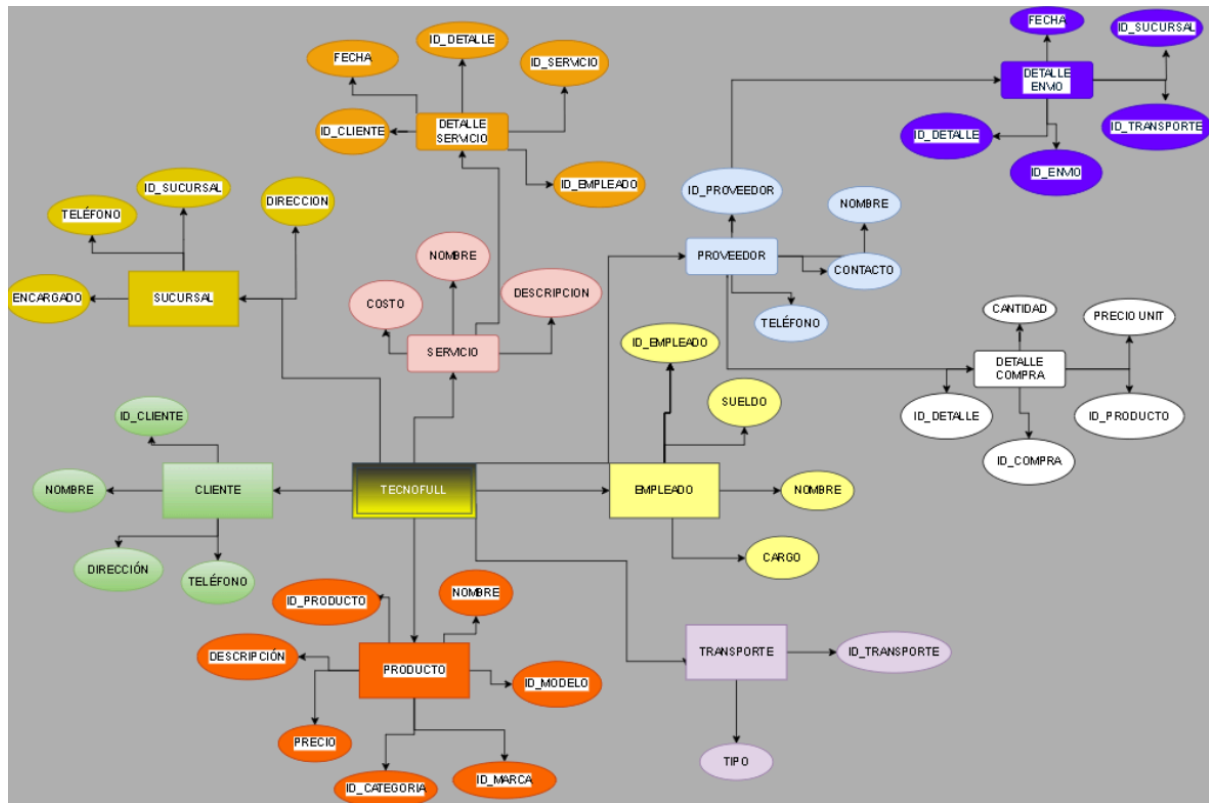
## ENTIDADES

1. **Cliente** (id\_cliente PK, nombre, dirección, teléfono, ...)
2. **Empleado** (id\_empleado PK, nombre, cargo, sueldo, ...)
3. **Proveedor** (id\_proveedor PK, nombre, contacto, teléfono, ...)
4. **Transporte** (id\_transporte PK, tipo, ...)
5. **Producto** (id\_producto PK, nombre, descripción, precio, ...)
6. **Servicio** (id\_servicio PK, tipo, ...)
7. **Sucursal** (id\_sucursal PK, dirección, teléfono, encargado, ...)

## RELACIONES

1. Cliente realiza Compra (Cliente.id\_cliente -> Compra.id\_cliente)
2. Empleado atiende Compra (Empleado.id\_empleado -> Compra.id\_empleado)
3. Proveedor suministra Producto (Proveedor.id\_proveedor -> Producto.id\_proveedor)
4. Producto se asocia con Servicio (Producto.id\_producto -> Servicio.id\_producto)
5. Sucursal tiene Empleado (Sucursal.id\_sucursal -> Empleado.id\_sucursal)
6. Sucursal ofrece Servicio (Sucursal.id\_sucursal -> Servicio.id\_sucursal)
7. Cliente solicita Servicio (Cliente.id\_cliente -> Servicio.id\_cliente)
8. Empleado realiza Servicio (Empleado.id\_empleado -> Servicio.id\_empleado)
9. Transporte realiza Envío (Transporte.id\_transporte -> Envío.id\_transporte)
10. Sucursal envía Producto (Sucursal.id\_sucursal -> Producto.id\_sucursal)

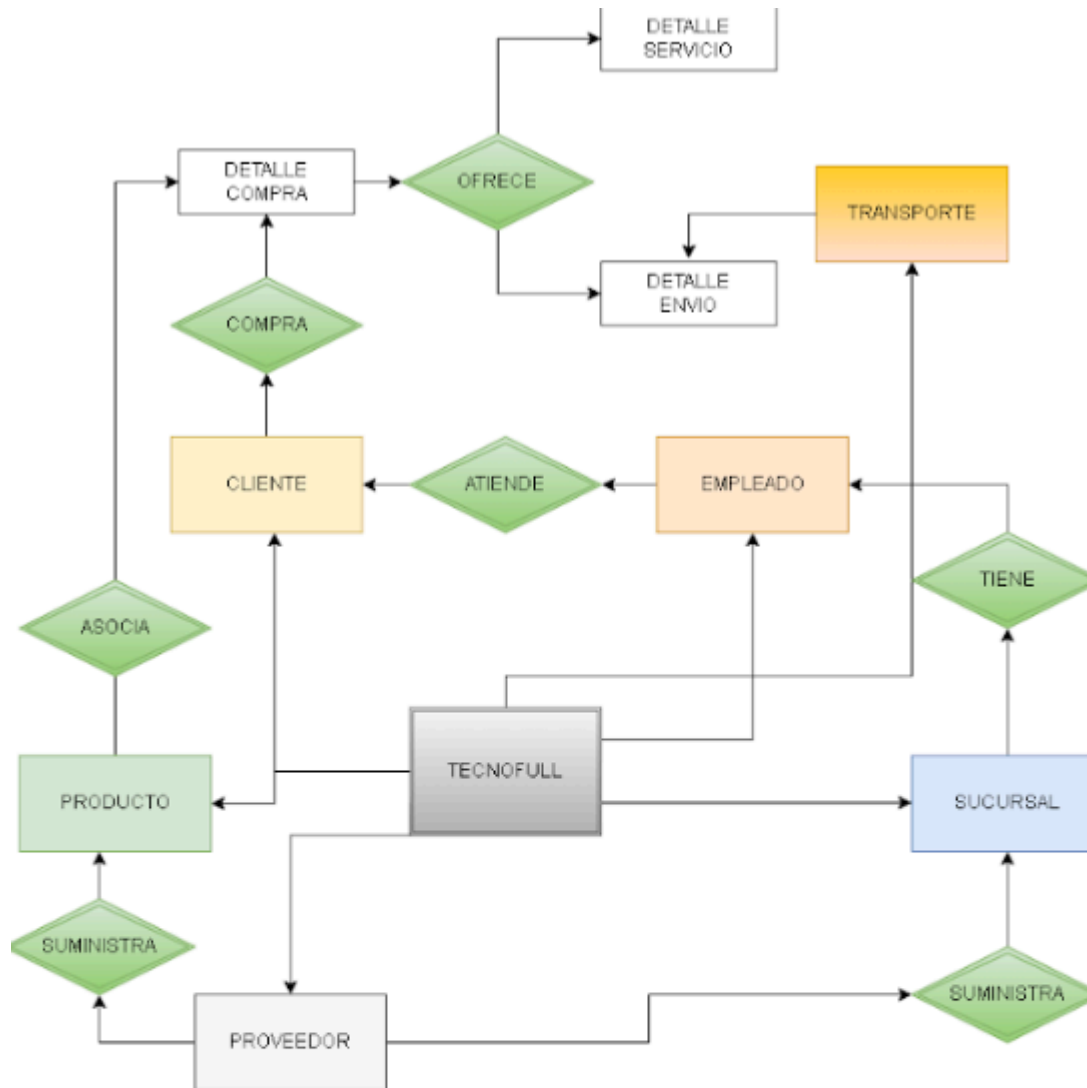




Link para visualizar el diagrama:

[https://drive.google.com/file/d/1h-9\\_OgEZ\\_tNYfbS2Ibu7n\\_YXHzyQQXMp/view?usp=sharing](https://drive.google.com/file/d/1h-9_OgEZ_tNYfbS2Ibu7n_YXHzyQQXMp/view?usp=sharing)





Link para visualizar este diagrama:

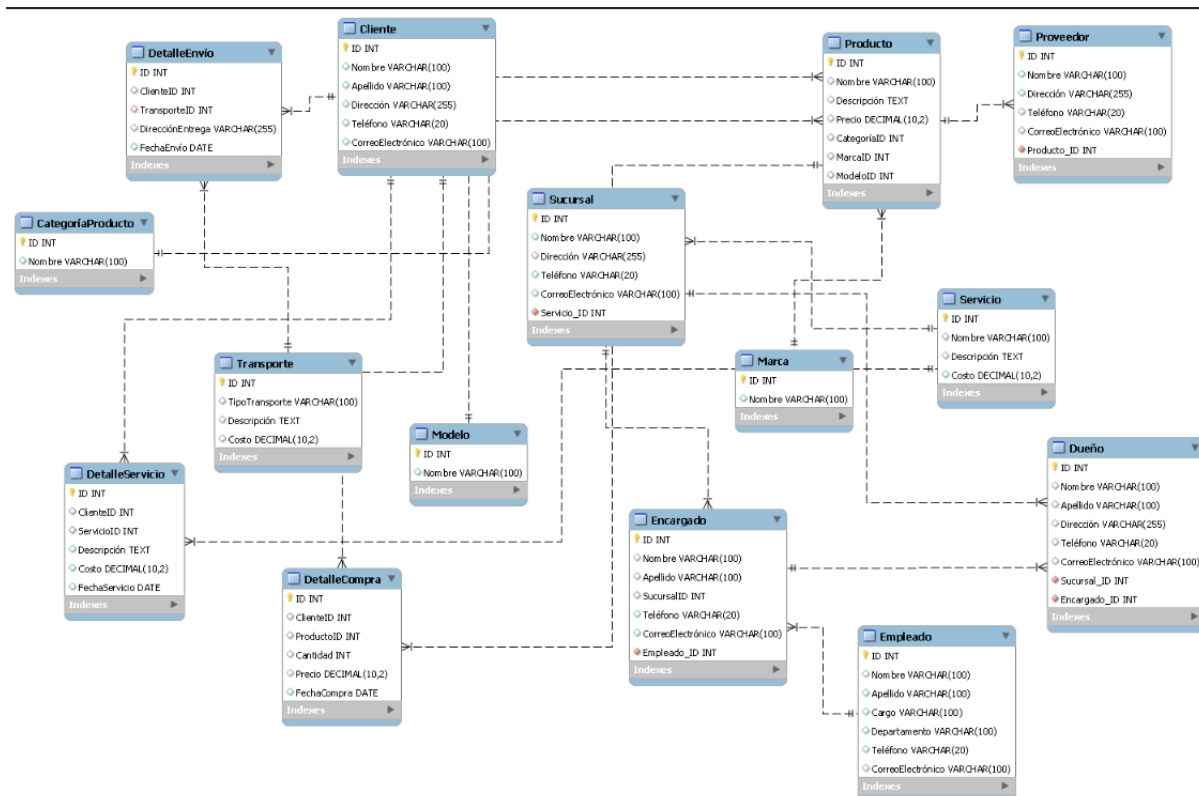
[https://drive.google.com/file/d/11G8Y4FFm\\_FW0tTLYscqLBh7WylvEwbGd/view?usp=sharing](https://drive.google.com/file/d/11G8Y4FFm_FW0tTLYscqLBh7WylvEwbGd/view?usp=sharing)

TABLA	NOMBRE CAMPO	CAMPO ABREVIADO	CLAVE PRIMARIA	CLAVE FORÁNEA	TIPO DE DATO
Cliente	ID_cliente	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
	Apellido	Apellido			Varchar(100)
	Dirección	Dirección			Varchar(255)
	Teléfono	Teléfono			Varchar(20)
	Correo electrónico	Correo electrónico			Varchar(100)
Empleado	ID_Empleado	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
	Apellido	Apellido			Varchar(100)
	Cargo	Cargo			Varchar(100)
	Departamento	Departamento			Varchar(20)
	Teléfono	Teléfono			Varchar(100)
	Correo Electrónico	Correo Electrónico			Varchar(100)
Proveedor	ID_Proveedor	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
	Dirección	Dirección			Varchar(255)
	Teléfono	Teléfono			Varchar(250)
	Correo electrónico	Correo electrónico			Varchar(20)
CategoriaProducto	ID_Categoria	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
Marca	ID_Marca	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
	ID_Modelo	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
Producto	ID_Producto	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			Varchar(100)
	Descripción	Descripción			Text
	Precio	Precio			Decimal (10,2)
	CategorialD	CategorialD		FK	int

	MarcaID	MarcaID		FK	int
	ModeloID	ModeloID		FK	int
Servicio	ID_Servicio	ID	PK (AUTOINCREMENT)		Int
	Nombre	Nombre			Varchar (100)
	Descripción	Descripción			Text
	Costo	Costo			Decimal (10,2)
Sucursal	ID_Sucursal	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			varchar(100)
	Dirección	Dirección			varchar(255)
	Teléfono	Teléfono			varchar(20)
	CorreoElectrónico	CorreoElectrónico			varchar(100)
Dueño	ID_Dueño	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			varchar(100)
	Apellido	Apellido			varchar(100)
	Dirección	Dirección			varchar(255)
	Teléfono	Teléfono			varchar(20)
	CorreoElectrónico	CorreoElectrónico			varchar(100)
Encargado	ID_Encargado	ID	PK (AUTOINCREMENT)		int
	Nombre	Nombre			varchar(100)
	Apellido	Apellido			varchar(100)
	ID_Sucursal	ID_Sucursal		FK	int
	Teléfono	Teléfono			varchar(20)
	CorreoElectrónico	CorreoElectrónico			varchar(100)
DetalleCompra	ID_DetalleCompra	ID	PK (AUTOINCREMENT)		int
	ID_Cliente	ID_Cliente		FK	int
	ID_Producto	ID_Producto		FK	int
	Cantidad	Cantidad			int
	Precio	Precio			Decimal(10.2)
	FechaCompra	FechaCompra			Date
DetalleServicio	ID_DetalleServicio	ID	PK (AUTOINCREMENT)		int
	ID_Cliente	ID_Cliente		FK	int
	ID_Servicio	ID_Servicio		FK	int
	Descripción	Descripción			Text

	Costo	Costo			Decimal(10,2)
	FechaServicio	FechaServicio			Date
Detalle_Envio	ID_DetalleEnvio	ID	PK (AUTOINCREMENT)		int
	ID_Cliente	ID_Cliente		FK	int
	ID_Transporte	ID_Transporte		FK	int
	DirecciónEntrega	DirecciónEntrega			Varchar(255)
	FechaEnvio	FechaEnvio			Date
Venta	ID_Venta	ID	PK (AUTOINCREMENT)	pk	int
	Fecha	FechaVenta			Date
	Monto	MontoVenta			Decimal (10,2)
	ID_Empleado	IDEmpleado		FK	

Mi grafico Entidad Relación se modificó un poco luego de realizar las asociaciones, realizar la carga en MySql y cargar las tablas en el mismo, quedando así:



Link del PDF:

[https://drive.google.com/file/d/1cvoGBY3EVqBUqCsnGRDP1\\_gJLsezWT8D/view?usp=sharing](https://drive.google.com/file/d/1cvoGBY3EVqBUqCsnGRDP1_gJLsezWT8D/view?usp=sharing)

## CREACIÓN DE NUEVAS TABLAS

Analizando el trabajo para realizar la segunda parte, me di cuenta que hacía falta la creación de una nueva tabla llamada VENTAS, siendo así la tabla 16 de mi Base de datos Tecnofull:

**16. Ventas:** La tabla Venta es esencial para llevar un registro detallado de todas las transacciones de ventas que se realizan en la empresa.

**Descripción:** Historial de todas las ventas de cada empleado con sus respectivas fechas y montos.

**Campos:**

- ID: Identificador único de la venta.
- Fecha: Fecha de la venta.
- Monto: Monto de la venta.
- IDEmpleado: Identificador del empleado que realizó la venta (clave foránea que se refiere a la tabla Empleado).

## CREACIÓN DE VISTAS

Repasando un poco, las vistas en bases de datos son consultas almacenadas que actúan como tablas virtuales. Ofrecen varias ventajas y funcionalidades importantes:

1. **Simplificación de Consultas Complejas:** Permiten encapsular consultas complejas y reutilizarlas sin necesidad de escribir el SQL nuevamente.
2. **Seguridad y Control de Acceso:** Permiten restringir el acceso a ciertos datos sensibles mostrando solo lo que es necesario a los usuarios, sin otorgar acceso directo a las tablas subyacentes.
3. **Facilitar el Mantenimiento y la Reutilización:** Ayudan a mantener la consistencia en las consultas usadas frecuentemente y facilitan la reutilización de consultas comunes.
4. **Agregación y Resumen de Datos:** Se utilizan para crear resúmenes de datos, como informes y análisis agregados, sin modificar los datos originales.
5. **Abstracción de la Estructura Física de la Base de Datos:** Proveen una capa de abstracción, lo cual permite cambios en la estructura de las tablas subyacentes sin afectar a las aplicaciones que usan las vistas.

## DESCRIPCIÓN DE VISTAS

- **Creación de vista ArticulosProveedores:**

**Funcionalidad:** Lista productos junto con sus proveedores y precios.

**Tablas involucradas:** Producto, Proveedor.

**Objetivo:** Facilitar la revisión de productos y precios para realizar nuevos pedidos.

Al querer crear esta vista, me topaba con errores, hasta que me di cuenta que necesitaba ajustar el esquema de la vista para que utilice los nombres correctos de las columnas en mis tablas. Es decir, en la tabla Producto, la columna que se relacionaba con Proveedor no existía directamente. Resolvi esto con la siguiente Query:

Agregue la columna ProveedorID en la tabla Producto:

```
ALTER TABLE Producto ADD ProveedorID INT;
```

- **Creacion de Vista VentasEmpleados:**

**Funcionalidad:** Muestra el total de ventas por empleado.

**Tablas involucradas:** Empleado, Venta.

**Objetivo:** Evaluar el desempeño de los empleados y gestionar comisiones.

-

- **Creación de VistaClientesFrecuentes:**

**Funcionalidad:** Esta vista muestra una lista de clientes que han realizado más compras.

**Tablas que la Componen:** Cliente, Venta

**Objetivo:** Identificar clientes leales y de alto valor para diseñar estrategias de fidelización y marketing.

Con esta vista me paso similar a la vista ArticulosProductos, ya que tuve que ajustar el esquema, agregando en la tabla ventas la columna ClienteID.

ALTER TABLE venta ADD ClienteID INT;

- **Creacion de Vista ProductosMasVendidos:**

**Funcionalidad:** Esta vista muestra los productos que han sido vendidos con mayor frecuencia.

**Tablas que la Componen:** Producto, Venta

**Objetivo:** Ayuda a identificar tendencias de ventas y ajustar inventarios y promociones según la demanda.

En este caso cree la columna ProductoID.

ALTER TABLE producto ADD ProductoID INT;

- **Creacion de Vista DetalleEnvios:**

**Funcionalidad:**Esta vista proporciona detalles de los envíos realizados, incluyendo información del cliente, dirección y estado del envío.

**Objetivo:** Facilita la gestión logística y el seguimiento de envíos.

**Tablas que la Componen:** Envio, Cliente, Transporte

En este caso cree la columna ClienteID.

ALTER TABLE transporte ADD ClienteID INT;



# CREACIÓN DE FUNCIONES

Las funciones son como pequeños programas que realizan una tarea específica y devuelve un resultado.

Esta me costó un poco porque no encontraba errores respecto a la sintaxis que me salía.

Leyendo e investigando un poco me di con que la sintaxis para crear una función en MySQL debe seguir ciertas reglas, como el uso de delimitadores específicos para separar las declaraciones SQL dentro de la función, en este caso requiere cambiar el delimitador predeterminado ; para evitar confusiones con las declaraciones internas de la función. Normalmente, se usa // o \$\$ como delimitador temporal.

## Función 1: total\_VENDIDO\_PORPRODUCTO

**Descripción:** Consulta el total de unidades vendidas por un producto.

**Tabla involucrada:** Detalle\_Venta.

**Objetivo:** Facilitar la lectura de la cantidad vendida de un artículo.

## Función 2: OPERACION\_POR\_EMPLEADO

**Descripción:** Cuenta las operaciones realizadas por un empleado.

**Tabla involucrada:** Venta.

**Objetivo:** Permitir saber el total de operaciones de un empleado.

## CREACIÓN DE STORED PROCEDURES

Herramientas que permiten reutilizar código, mejorar la eficiencia, aumentar la seguridad y facilitar el mantenimiento de la base de datos.

### Procedimiento 1: ordenarProducto

**Descripción:** Ordena la tabla de productos por una columna especificada.

**Tabla involucrada:** Producto.

**Objetivo:** Facilitar el ordenamiento de la tabla de productos.

### Procedimiento 2: agregar\_eliminar\_producto

**Descripción:** Agrega o elimina un producto.

**Tabla involucrada:** Producto.

**Objetivo:** Facilitar la carga y eliminación de productos.

## CREACIÓN DE TRIGGERS

Los Triggers son funciones especiales que se activan automáticamente respondiendo a eventos específicos en la base de datos.

### Trigger para Productos (movimientos\_productos)

**Descripción:** Registra actualizaciones y eliminaciones en la tabla de productos.

**Tabla de log:** movimientos\_productos.

**Objetivo:** Registrar cambios para control de mercadería.

### Trigger para Ventas (movimientos\_ventas)

**Descripción:** Registra inserciones y eliminaciones en la tabla de ventas.

**Tabla de log:** movimientos\_ventas.

**Objetivo:** Controlar movimientos de operaciones que involucran dinero.

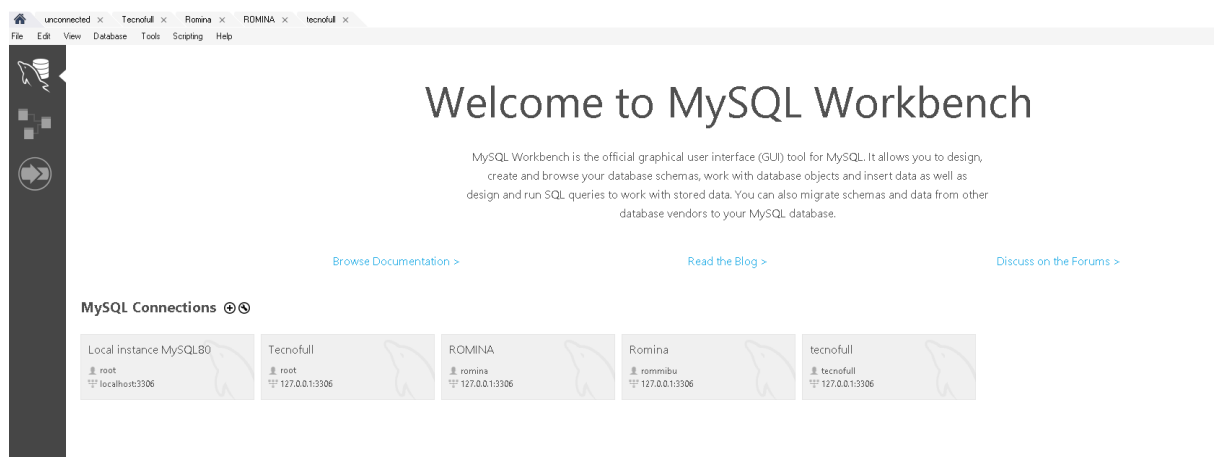
## DML (DATA MANIPULATION LANGUAGE)

- INSERT: Inserté datos en las tablas.
- UPDATE: Actualice la dirección de un cliente.
- DELETE: Elimine un cliente.

## DCL (DATA CONTROL LANGUAGE)

En esta parte cree, modifique, borre, autorice todo y algunos permisos de los usuarios ROMINA Y Romina, esto principalmente a través de las funciones:

- GRANT: Conceder permisos de lectura a un usuario.
- REVOKE: Revocar permisos de lectura a un usuario.



## TABLA PARA GRÁFICOS

Creo una nueva tabla para realizar gráficos descriptivos de mi Base Tecnofull. En mi caso son muchas tablas que puedo y debo hacer (anteriormente cree varias vistas con las cuales también puedo hacer gráficos), pero en este caso para la entrega y a modo de ejemplo voy a crear una tabla que te permita graficar el resultado de las ventas por mes, pero para ello primero debo extraer la información de las ventas de la tabla **"venta"** y verlo por mes, de esta manera entonces creo una tabla específica para almacenar estos datos agregados.