

PROYECTO FINAL - GLOBALMART



SHOPPING

INDICE

| | |
|--|---|
| 1. Introducción..... | 2 |
| 2. Análisis del Enunciado..... | 2 |
| 3. Modelo Conceptual..... | 2 |
| 4. Modelo Relacional..... | 4 |
| 5. Implementación en MySQL..... | 5 |
| 6. Consultas Propuestas..... | 7 |
| 7. Ampliación de la Base de Datos..... | 8 |
| 8. Vistas y Triggers..... | 8 |
| 9. Conclusiones..... | 9 |

1. Introducción

Bienvenido a mi proyecto final, vamos a crear una base de datos desde 0. Les mostrare paso a paso cómo lo haremos con el objetivo de ayudar a la empresa GlobalMart tener sus datos de cliente, productos y pedidos más organizados.

2. Análisis del Enunciado

Sistema de Gestión de Compras para GlobalMart Ltda,

"En GlobalMart Ltda., estamos comprometidos con la eficiencia y la transparencia en nuestras operaciones de compras. Para optimizar nuestros procesos y garantizar un mejor control sobre nuestras actividades de adquisición, buscamos implementar un sistema de gestión de compras integral."

Nos piden diseñar un sistema que administre eficientemente información sobre proveedores, productos y pedidos realizados.

3. Modelo Conceptual

En este apartado sirve para tener una visión más clara para saber qué datos se deben almacenar y cómo se relacionan entre sí. Empezaremos por añadir las entidades con sus atributos.

- PROVEEDOR

Aquí almacenaremos la información de GlobalMart donde tendremos en mano su correo electrónico, la dirección, el CIF y el número de contacto.

- PRODUCTO

El producto tiene que estar bien intensificado con su nombre, código de barra y precio, así saber si hay stock para el cliente o si está caducado y hay que destruirlo. También entrará el tipo de envase para saber a la hora de transportarlo si es delicado y requiere un empaquetamiento especial.

- PEDIDO

Cuando el cliente nos hace una compra tendremos un pedido; en el sistema aparecerá la fecha en la que se hizo y el estado del pedido para saber si falta mucho para expedirse o si ya está en reparto.

- CLIENTE

Los clientes se crearán un cuenta online donde tendrán que añadir su correo, nombre, teléfono móvil. Luego también quedará registrado si es particular o una empresa ya que el trato será diferente. También implementamos opcionalmente el método de pago de preferencia para que a la hora de pagar sea eficaz.

- DETALLE DEL PEDIDO

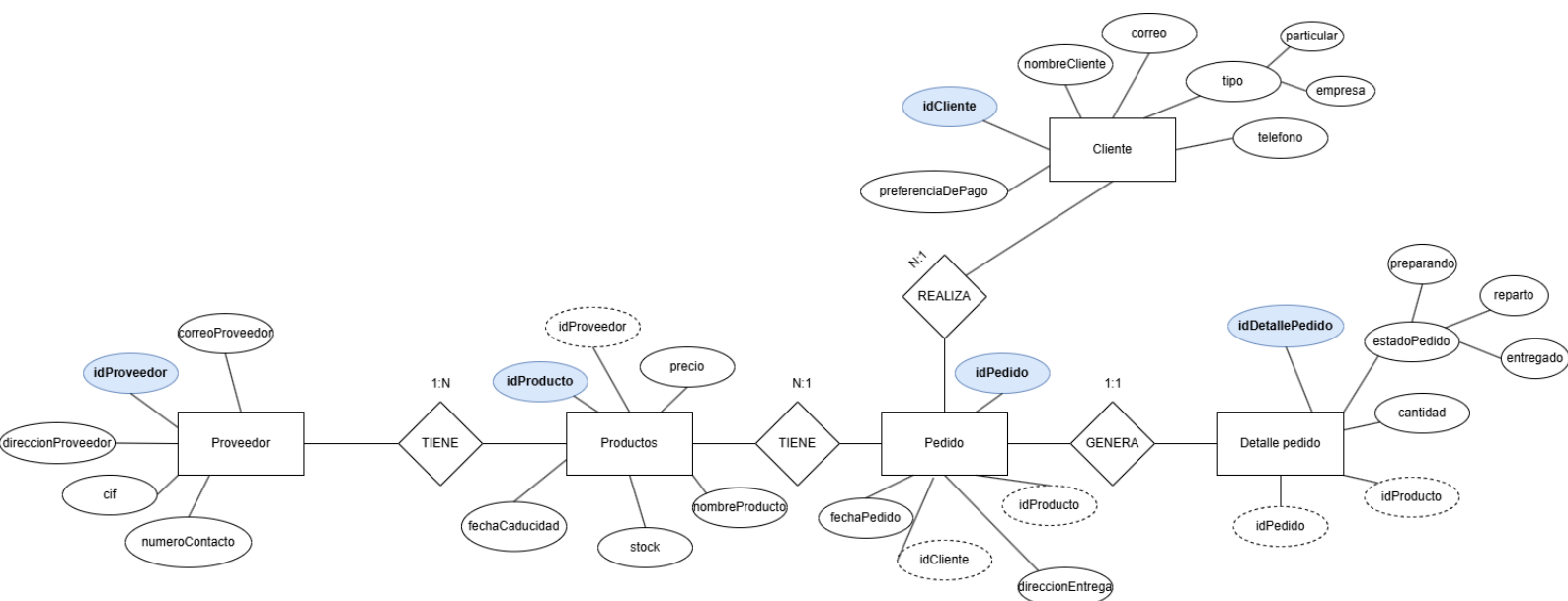
En detalle del pedido podremos ver las compras que han realizado los clientes con las cantidades y la dirección de entrega.

Una vez que tengamos claro las entidades y sus atributos pasaremos con la relación entre entidades. La relación describe cómo dos o más entidades están vinculadas, que se puede representar como 1:1, 1:N, N:N o N:1.

Ejemplo:

Un *Departamento* tiene varios *Empleados* -> *DEPARTAMENTO* 1:N *EMPLEADOS*

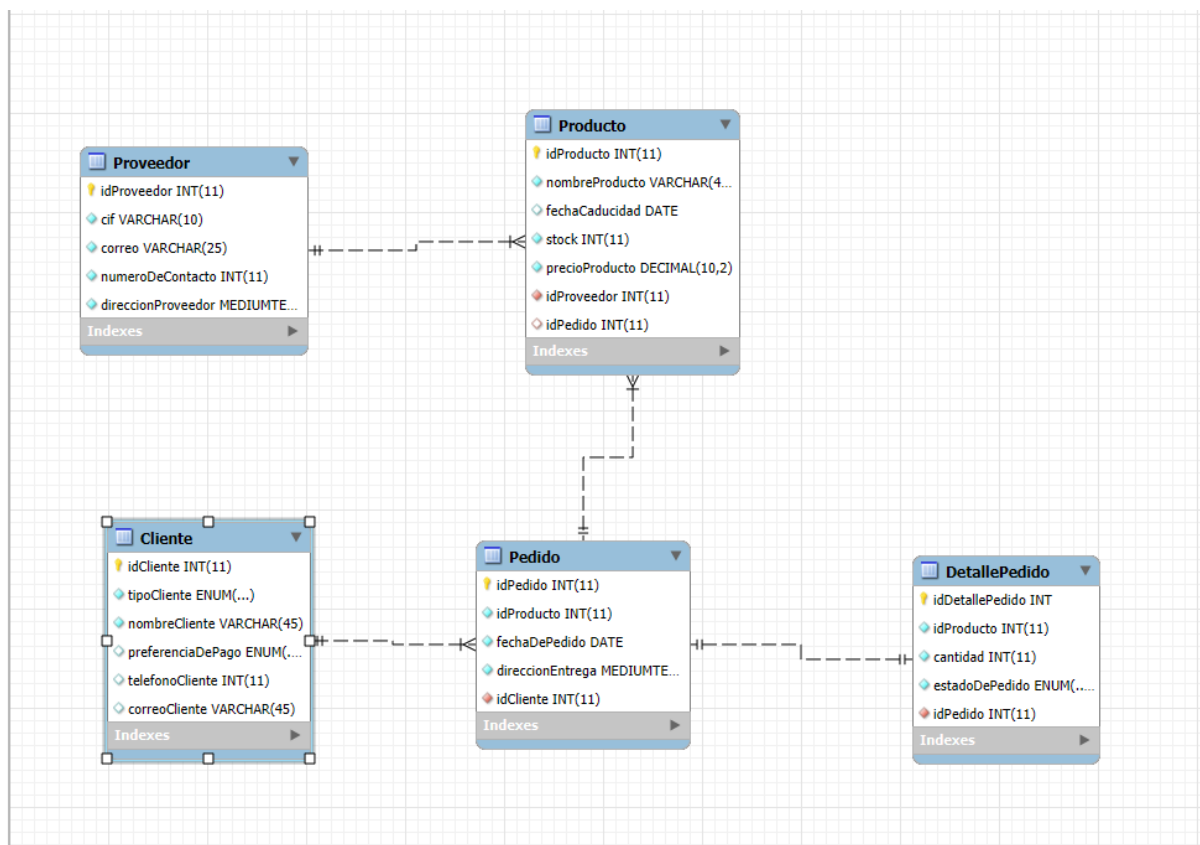
Primero tenemos un **PROVEEDOR** que tiene varios **PRODUCTOS**, luego continuamos con un **PEDIDO** tiene varios **PRODUCTOS**. A continuación el pedido tiene una relación con el cliente y el detalle de pedido aparte, el **CLIENTE** puedes realizar varios **PEDIDOS** y de un **PEDIDO** se genera un **DETALLE DE PEDIDO**.



4. Modelo Relacional

El modelo relacional es una forma de organizar los datos en tablas formadas por filas y columnas.

Cada tabla representa una entidad (como clientes, productos, etc.) y se pueden relacionar entre sí mediante claves(clave primaria y la clave foránea).



5. Implementación en MySQL

Crearemos una base de datos llamado Globlmart, a partir del modelo relacional, se crean las tablas correspondientes con sus atributos, tipos de datos y claves primarias:

Proveedor:

```
CREATE TABLE IF NOT EXISTS `Globlmart`.`Proveedor` (  
  `idProveedor` INT(11) NOT NULL,  
  `cif` VARCHAR(10) NOT NULL,  
  `correo` VARCHAR(25) NOT NULL,  
  `numeroDeContacto` INT(11) NOT NULL,  
  `direccionProveedor` MEDIUMTEXT NOT NULL,  
  PRIMARY KEY (`idProveedor`))
```

Producto:

```
CREATE TABLE IF NOT EXISTS `Globlmart`.`Producto` (  
  `idProducto` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombreProducto` VARCHAR(45) NOT NULL,  
  `fecchaCaducidad` DATE NULL DEFAULT NULL,  
  `stock` INT(11) NOT NULL,  
  `precioProducto` DECIMAL NOT NULL,  
  `idProveedor` INT(11) NOT NULL,  
  `idPedido` INT(11) NOT NULL,  
  PRIMARY KEY (`idProducto`),  
  INDEX `fk_Producto_Proveedor_idx` (`idProveedor` ASC) VISIBLE,  
  INDEX `fk_Producto_Pedido1_idx` (`idPedido` ASC) VISIBLE,  
  CONSTRAINT `fk_Producto_Proveedor`  
    FOREIGN KEY (`idProveedor`)
```

Cliente:

```
CREATE TABLE IF NOT EXISTS `Globlmart`.`Cliente` (  
  `idCliente` INT(11) NOT NULL AUTO_INCREMENT,  
  `tipoCliente` ENUM('particular', 'empresa') NOT NULL,  
  `nombreCliente` VARCHAR(45) NOT NULL,  
  `preferenciaDePago` ENUM('MasterCard', 'Visa') NULL DEFAULT NULL,  
  `telefonoCliente` INT(11) NULL DEFAULT NULL,  
  `correoCliente` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`idCliente`))
```

Pedido:

```
CREATE TABLE IF NOT EXISTS `Globlmart`.`Pedido` (  
  `idPedido` INT(11) NOT NULL AUTO_INCREMENT,  
  `idProducto` INT(11) NOT NULL,  
  `fechaDePedido` DATE NOT NULL,  
  `direccionEntrega` MEDIUMTEXT NOT NULL,  
  `idCliente` INT(11) NOT NULL,  
  PRIMARY KEY (`idPedido`),  
  INDEX `fk_Pedido_Cliente1_idx` (`idCliente` ASC) VISIBLE,  
  CONSTRAINT `fk_Pedido_Cliente1`  
    FOREIGN KEY (`idCliente`)  
    REFERENCES `Globlmart`.`Cliente` (`idCliente`)
```

DetallePedido:

```
CREATE TABLE IF NOT EXISTS `Globlmart`.`DetallePedido` (  
  `idDetallePedido` INT(11) NOT NULL AUTO_INCREMENT,  
  `idProducto` INT(11) NOT NULL,  
  `cantidad` INT(11) NOT NULL,  
  `estadoDePedido` ENUM('preparando', 'reparto', 'entregado') NOT NULL,  
  `idPedido` INT(11) NOT NULL,  
  PRIMARY KEY (`idDetallePedido`),  
  INDEX `fk_DetallePedido_Pedido1_idx` (`idPedido` ASC) VISIBLE,
```

Finalmente, se pueden insertar registros y realizar consultas.

Ahora con el comando INSERT INTO podemos agregar la informacion que queremos a una tabla. Por ejemplo, aqui esta la informacion que hemos agregado a la tabla proveedor:

```
#PROVEEDOR
INSERT INTO globalmart.proveedor (cif, correo, numeroDeContacto, direccionProveedor)
VALUE
('898742356', 'dwall@meandplants.com', 697654291, 'C. Agüere, 17, 38005 Santa Cruz de Tenerife'),
('812569878', 'med@surfsup.com', 679505439, 'Av. de San Sebastián, 152, 38005 Santa Cruz de Tenerife'),
('894756357', 'geeks@general.com', 679452863, 'C. Nava y Grimon, 18, 38201 La Laguna, Santa Cruz de Tenerife'),
('847986542', 'stanley@operations.com', 988745712, 'Polígono Industrial Granadilla, PARCELA, 62, A, 38619, Santa Cruz de Tenerife'),
('87856+635', 'ikea@canariastnf.com', 634687821, 'Parque Comercial, C. Aceviño, 9, 38205 La Laguna, Santa Cruz de Tenerife');
```

Para asegurar que todo esa en orden y no hemos faltado en nada podemos poner SELECT * FROM y luego el nombre de la tabla que queremos ver, por ejemplo “SELECT * FROM globalmart.proveedor;”. Si todo sale bien nos debería aparecer la tabla con los datos que hemos agregado.

| | IdProveedor | cif | correo | numeroDeContacto | direccionProveedor |
|---|-------------|-----------|------------------------|------------------|--|
| | 29 | 898742356 | dwall@globalmart.com | 697654291 | C. Agüere, 17, 38005 Santa Cruz de Tenerife |
| | 30 | 812569878 | med@surfsup.com | 679505439 | Av. de San Sebastián, 152, 38005 Santa Cru... |
| | 31 | 894756357 | geeks@general.com | 679452863 | C. Nava y Grimon, 18, 38201 La Laguna, Sa... |
| | 32 | 847986542 | stanley@operations.... | 988745712 | Poligono Industrial Granadilla, PARCELA, 62, ... |
| | 33 | 87856+635 | ikea@canariastnf.com | 634687821 | Parque Comercial, C. Aceviño, 9, 38205 La L... |
| • | total | total | total | total | total |

6. Consultas

Las consultas sirven para sacar información que consideramos mas importante de la base de datos.

Considero que estas son las más relevantes:

Que nos muestre los productos fuera de stock.

Para poder notificar la falta de stock y así gestionar su reabastecimiento.

Mostrar los tipos de clientes con su método de pago.

Para saber simplificar los pasos a la hora de hacer el pago.

Ordenar los productos más caros a los mas baratos.

Tener una vision mas clara de los productos que estamos vendiendo.

Ver los pedidos que hay por estado.

Así podemos ver cuantos estan entregados, de reparto o hasta si todavía lo esta preparando.

Ver cuando el cliente realizo el pedido y el estado en el que esta.

Para poder ver si hay demoras.

Ver el precio Total de cada pedido.

Así podremos visualizar cuanto se han gastado en un pedido

Mostrar todos los productos que han sido pedidos

Para poder gestionar la compra y la demanda. Si vemos que no se esta vendiendo mucho a algun producto, pues ya sabemos que tendremos que pedir menos cantidades o parar de venderlo y invertir en otra cosa.

7. Ampliación de la Base de Datos

Aquí añadiremos columnas o tablas que creemos que sería relevante.

Hemos creado tres columnas:

1. Una columna de fecha de registro en la tabla de cliente. Para poder ver cuantos clientes nuevos estamos obteniendo.
2. Otra columna de estado de pago en la tabla de pedido. Que nos informará si esta pendiente, pagado o cancelado.
3. Luego es super importante añadir una fecha de estimación de la entrega, que lo añadiremos en la tabla de detallepedido.

Y creamos una tabla de Pago donde pondremos la fecha de pago, el monto y el metodo de pago que se hizo.

8. Vistas y Triggers

Hemos añadido tres triggers y 4 cuatro views.

Triggers

Son procedimientos automáticos que se ejecutan de manera automática cuando ocurre un determinado evento en una tabla o vista.

1. Actualiza el estado el estadoPago en la tabla Pedido y lo pone en 'pagado' cuando se creas un registro en la tabla Pago.
2. Previene que se inserte un detallepedido si el stock disponible del producto es menor que la cantidad solicitada.
3. Actualiza el estado de pedido cuando se entrega y aparece 'completado'.

Vistas/Views

1. Estado de cada pedido agrupado por cliente.

Te muestra todos los pedidos hechos por cada cliente, incluyendo en qué estado está cada pedido (pagado o no, completado o no). Para soporte al cliente ("¿Qué pasó con mi pedido?"), analizar clientes frecuentes o inactivos.

2. Resumen completo de cada pedido con total del precio.

Muestra un resumen detallado de cada pedido, con el total del pedido en dinero, nombre del cliente, fecha, estado de pago y estado general. Sirve para reportes financieros rápidos, ver cuanto ha comprado cada cliente y saber si hay pedido sin pagar o sin completar.

3. Clientes que no han realizado ningún pedido.

Muestra una lista de clientes que están registrados pero no han hecho ningún pedido. Para campañas de marketing, detectar clientes fantasmas o ver si hay fallas en el proceso de venta.

4. Productos que actualmente no tienen stock disponible.

Lista todos los productos cuyo stock está en 0, es decir, no disponibles para venta. Para generar alertas de reposición, reportes de inventario y evitar vender productos que ya no tienen.

9. Conclusion

A lo largo de este proyecto hemos desarrollado de forma estructurada un sistema de gestión de compras para GlobalMart Ltda., con el objetivo de mejorar la organización y eficiencia en la administración de proveedores, productos, clientes y pedidos.

Gracias a la creación de un modelo conceptual y relacional bien definido, fue posible implementar correctamente las tablas necesarias, así como establecer las relaciones adecuadas entre ellas. Las consultas propuestas demuestran la utilidad del sistema al proporcionar información clave para la toma de decisiones, como el control de stock, el seguimiento de pedidos o el análisis del comportamiento de los clientes.

Además, realizamos ampliaciones significativas como la incorporación de una tabla de pagos y campos adicionales, mejorando el control de las operaciones. También integramos vistas y triggers que automatizan procesos y facilitan el monitoreo constante del sistema.

Este proyecto me ayudó a poner en práctica lo aprendido, también me permitió ver cómo una buena base de datos puede marcar la diferencia en la organización y gestión de una empresa. Me voy satisfecho con el resultado, sabiendo que he creado algo funcional, bien estructurado y con potencial para seguir creciendo.