

Ejercicio Grupal – AE1_ABPRO: Diseño Base de Datos Relacional para "Todo Ventas en Línea"

1. Rol de las Bases de Datos Relacionales

Un Sistema de Gestión de Bases de Datos Relacionales (RDBMS) permite centralizar la información de clientes, productos y pedidos en una única plataforma, evitando la dispersión y redundancia de datos típica de archivos y planillas. Facilita consultas complejas en segundos, reportes útiles para la toma de decisiones, y asegura la integridad y consistencia de la información mediante reglas y restricciones. Además, es escalable, soporta múltiples usuarios y grandes volúmenes de datos, y permite acceso seguro y controlado a la información.

Características principales de un RDBMS:

- - Integridad referencial: mantiene relaciones correctas entre tablas.
- - Normalización: evita duplicidades y dependencias innecesarias.
- - Lenguaje SQL: consulta y manipulación eficiente.
- - Transaccionalidad: operaciones completas y seguras.
- - Seguridad: control de acceso y privilegios.

2. Selección del RDBMS y Justificación

RDBMS seleccionado: MySQL

Justificación: MySQL es amigable para usuarios nuevos, tiene una comunidad extensa, es gratuito y escalable. Permite gestionar desde proyectos pequeños hasta grandes volúmenes de datos y usuarios. Tiene abundante documentación, herramientas visuales (MySQL Workbench), es multiplataforma y ampliamente utilizado en la industria.

3. Instalación y Configuración

Cada miembro debe instalar MySQL Community Edition y MySQL Workbench. Durante la instalación, definir el usuario 'root' y contraseña segura. Comprobar la conexión creando una base de datos de prueba y accediendo a ella desde Workbench o consola.

4. Diseño de la Base de Datos

Tablas requeridas:

a) Clientes

id_cliente (PK)	nombre	correo	dirección	teléfono
INT, AUTO	VARCHAR	VARCHAR UNIQUE	VARCHAR	VARCHAR

b) Productos

id_producto (PK)	nombre_producto	categoría	precio	stock
INT, AUTO	VARCHAR	VARCHAR	DECIMAL	INT

c) Pedidos

id_pedido (PK)	id_cliente (FK)	fecha	estado
INT, AUTO	INT	DATE	VARCHAR

d) Detalle_Pedidos (tabla de unión para productos pedidos)

id_detalle (PK)	id_pedido (FK)	id_producto (FK)	cantidad
INT, AUTO	INT	INT	INT

Nota: La tabla intermedia 'Detalle_Pedidos' permite que un pedido incluya múltiples productos y cantidades distintas.

Modelo Entidad-Relación (descripción):

- Un cliente puede realizar muchos pedidos (relación 1:N).
- Un pedido puede tener muchos productos y un producto puede estar en muchos pedidos (relación N:M, mediante Detalle_Pedidos).

Diagrama lógico (simplificado):

Clientes (1) ---< (N) Pedidos (1) ---< (N) Detalle_Pedidos (N) >--- (1) Productos

Script de creación de tablas (MySQL):

```
CREATE TABLE Clientes (  
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    correo VARCHAR(100) UNIQUE NOT NULL,  
    direccion VARCHAR(150),  
    telefono VARCHAR(30)  
);
```

```
CREATE TABLE Productos (  
    id_producto INT AUTO_INCREMENT PRIMARY KEY,  
    nombre_producto VARCHAR(100) NOT NULL,  
    categoria VARCHAR(50),  
    precio DECIMAL(10,2) NOT NULL,  
    stock INT DEFAULT 0  
);
```

```
CREATE TABLE Pedidos (  
    id_pedido INT AUTO_INCREMENT PRIMARY KEY,  
    id_cliente INT NOT NULL,  
    fecha DATE,  
    estado VARCHAR(30),  
    FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)  
);
```

```
CREATE TABLE Detalle_Pedidos (  
    id_detalle INT AUTO_INCREMENT PRIMARY KEY,  
    id_pedido INT NOT NULL,  
    id_producto INT NOT NULL,  
    cantidad INT NOT NULL,  
    FOREIGN KEY (id_pedido) REFERENCES Pedidos(id_pedido),  
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)  
);
```

5. Consideraciones adicionales

- Las claves primarias son autoincrementales para simplificar el ingreso de registros.
- La tabla de detalles permite flexibilidad para pedidos con varios productos y cantidades distintas.
- La integridad referencial asegura que no existan pedidos sin cliente, ni detalles sin pedido/producto.
- Se puede expandir fácilmente agregando más tablas (usuarios, métodos de pago, etc.).

6. Conclusión

Implementar un RDBMS como MySQL mejora la gestión de la información en 'Todo Ventas en Línea', asegura integridad y escalabilidad, y facilita el análisis y control de pedidos y clientes a medida que la empresa crece.

7. Integrantes

Rodrigo Chandía C.

Repositorio: [portafolioTD/v0 trabajos bootcamp/m3 bd relacional/AE1 ABPRO at main · rodrigoNXCL/portafolioTD · GitHub](https://github.com/portafolioTD/v0_trabajos_bootcamp/m3_bd_relacional/AE1_ABPRO_at_main)