



INFORME DE BASES DE DATOS

Entrega de avance 1º corte

Información

NOMBRE DEL ESPACIO ACADÉMICO: DESARROLLO DE BASE DE DATOS

PROGRAMA: INGENIERÍA SISTEMAS

DOCENTE: BRAYAN ARCOS

YADIR ADRIAN MUÑOZ

yamunozr22@itp.edu.co

Contenido

Resumen Ejecutivo	2
Introducción.....	3
Metodología	4
Herramientas Utilizadas	4
Procedimientos	4
Desarrollo del Informe	5
Descripción de la Base de Datos.....	5
Esquema de la Base de Datos: Tablas, relaciones y claves principales	1
Consultas Realizadas	5
Consideraciones de Diseño	9
Nombre de la Base de Datos y Descripción	9
Entidades y Atributos	9
Claves Primarias y Relaciones entre Tablas.....	17

Resumen Ejecutivo

Este informe se está realizando con el propósito de facilitar el aprendizaje de los conceptos fundamentales en el diseño y desarrollo de bases de datos. En este caso para un sistema de comercio electrónico, utilizando **MySQL Workbench** como herramienta principal, escogí este tema de comercio electrónico ya que me gustaría desarrollar un sistema de ventas, con el conocimiento que valla adquiriendo en el transcurso del curso quiero crear una base de datos bien estructurada.

A lo largo de esta primera parte del informe, se documentan paso a paso las metodologías necesarias para crear una base de datos eficiente y escalable, con el fin de aplicar estos conocimientos de manera práctica en futuros proyectos.

El objetivo es que, como estudiante, puedas entender cómo estructurar una base de datos a través de un ejemplo real: un sistema de **e-commerce**. Este ejemplo práctico incluye tablas para gestionar usuarios, productos, categorías y órdenes, lo que permite aprender a diseñar un esquema relacional y aplicar comandos **SQL** para manipular los datos.

También, se explican las consultas **SQL** utilizadas para insertar, consultar, actualizar y eliminar datos, así como para modificar la estructura de las tablas. Este modelo práctico permite comprender conceptos como la normalización, el uso de claves primarias y foráneas, y la importancia de mantener un historial de modificaciones mediante los campos **createdAt** y **updatedAt**. Estos atributos son recomendados por nuestro docente, ya que estos registros muy importantes en proyectos reales.

El informe concluye con recomendaciones para optimizar y mejorar el rendimiento de la base de datos, lo que no solo consolida el aprendizaje

teórico, sino que también ofrece una aplicación tangible de los conocimientos adquiridos.

Introducción

El informe lo estoy desarrollando con el objetivo de que me sirva como una herramienta de aprendizaje práctico para el diseño y manejo de bases de datos, utilizando **MySQL Workbench**. Me permite adquirir y reforzar habilidades esenciales en SQL, incluyendo el diseño de esquemas, la manipulación de datos y la optimización de consultas.

El informe abarca tanto aspectos teóricos como prácticos, cubriendo temas clave como el diseño de tablas, la normalización, y la creación de relaciones entre entidades. A través de la implementación de un ejemplo concreto de e-commerce, se busca que el estudiante pueda aplicar los conceptos aprendidos en clase a situaciones reales, favoreciendo una comprensión más profunda de las bases de datos relacionales.

Además, este documento incluye instrucciones detalladas sobre el uso de comandos **SQL** para insertar, consultar, actualizar y eliminar datos, así como la modificación de la estructura de las tablas. Este enfoque práctico proporciona un contexto realista para el aprendizaje, facilitando la asimilación de los conceptos mediante la práctica activa.

El objetivo principal es que, como estudiante, pueda desarrollar un entendimiento integral del proceso de diseño y gestión de bases de datos, adquiriendo habilidades que son fundamentales para proyectos de desarrollo de software reales, donde implica mucha responsabilidad.

Metodología

Herramientas Utilizadas

- ✓ **MySQL Workbench:** Herramienta gráfica para la creación, gestión y diseño de bases de datos.
- ✓ **MySQL Server:** Motor de base de datos donde se almacena toda la información.
- ✓ **SQL:** Lenguaje utilizado para interactuar con la base de datos.

Procedimientos

1. **Creación de la base de datos** y definición del esquema.
2. **Diseño de tablas** con relaciones entre ellas, asegurando la integridad referencial.
3. **Implementación de comandos SQL** para gestionar datos (inserciones, actualizaciones, consultas, eliminaciones).
4. **Documentación** de las consultas SQL y resultados obtenidos.

Desarrollo del Informe

Descripción de la Base de Datos

De acuerdo con lo aprendido en clases, para este informe he decidido adentrarme en la parte del E-commerce, para ello me he estado documentando sobre las tablas necesarias que debe tener una base de datos de comercio electrónico.

Mencionare las tablas que debe tener una base de datos de comercio electrónico:

Usuarios

- ❖ id: INT (PK)
- ❖ nombre: VARCHAR
- ❖ email: VARCHAR
- ❖ contraseña: VARCHAR
- ❖ telefono: VARCHAR
- ❖ direccion: TEXT
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Productos

- ❖ id: INT (PK)
- ❖ nombre: VARCHAR
- ❖ descripcion: TEXT
- ❖ precio: DECIMAL
- ❖ stock: INT
- ❖ categoria_id: INT (FK a Categorías)
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Categorías

- id: INT (PK)
- nombre: VARCHAR
- descripcion: TEXT
- createdAt: DATETIME
- updatedAt: DATETIME

Proveedores

- ❖ id: INT (PK)
- ❖ nombre: VARCHAR
- ❖ direccion: TEXT
- ❖ telefono: VARCHAR
- ❖ email: VARCHAR
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Órdenes

- ❖ id: INT (PK)
- ❖ usuario_id: INT (FK a Usuarios)
- ❖ estado: ENUM('Pendiente', 'Enviado', 'Entregado', 'Cancelado')
- ❖ total: DECIMAL
- ❖ direccion_envio: TEXT
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Detalles de Órdenes

- ❖ id: INT (PK)
- ❖ orden_id: INT (FK a Órdenes)
- ❖ producto_id: INT (FK a Productos)
- ❖ cantidad: INT
- ❖ precio_unitario: DECIMAL
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Métodos de Pago

- ❖ id: INT (PK)
- ❖ nombre: VARCHAR
- ❖ descripcion: TEXT
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Pagos

- ❖ id: INT (PK)
- ❖ orden_id: INT (FK a Órdenes)
- ❖ metodo_pago_id: INT (FK a Métodos de Pago)
- ❖ monto: DECIMAL
- ❖ estado_pago: ENUM('Pendiente', 'Completado', 'Fallido')
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Envíos

- ❖ id: INT (PK)
- ❖ orden_id: INT (FK a Órdenes)

- ❖ direccion_envio: TEXT
- ❖ estado_envio: ENUM('Pendiente', 'Enviado', 'Entregado')
- ❖ fecha_envio: DATETIME
- ❖ fecha_entrega: DATETIME
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Carritos de Compra

- ❖ id: INT (PK)
- ❖ usuario_id: INT (FK a Usuarios)
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Detalles del Carrito

- ❖ id: INT (PK)
- ❖ carrito_id: INT (FK a Carritos)
- ❖ producto_id: INT (FK a Productos)
- ❖ cantidad: INT
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Descuentos

- ❖ id: INT (PK)
- ❖ codigo: VARCHAR
- ❖ porcentaje_descuento: DECIMAL
- ❖ fecha_inicio: DATETIME
- ❖ fecha_fin: DATETIME
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Reseñas de Productos

- ❖ id: INT (PK)
- ❖ usuario_id: INT (FK a Usuarios)
- ❖ producto_id: INT (FK a Productos)

- ❖ comentario: TEXT
- ❖ calificacion: INT (1-5)
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Direcciones de Envío

- ❖ id: INT (PK)
- ❖ usuario_id: INT (FK a Usuarios)
- ❖ direccion: TEXT
- ❖ ciudad: VARCHAR
- ❖ codigo_postal: VARCHAR
- ❖ pais: VARCHAR
- ❖ createdAt: DATETIME
- ❖ updatedAt: DATETIME

Esquema de la Base de Datos: Tablas, relaciones y claves principales

Las relaciones entre las tablas en son de diferentes tipos y representan conexiones lógicas entre las entidades.

las relaciones y su tipo:

1. Relación entre Productos y Categorías

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Un producto pertenece a una categoría. Una categoría puede tener muchos productos.
- **Clave Foránea:** categoria_id en la tabla Productos hace referencia a id en Categorías.

2. Relación entre Productos y Proveedores

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Un producto puede ser proporcionado por un proveedor. Un proveedor puede proporcionar muchos productos.
- **Clave Foránea:** proveedor_id en la tabla Productos hace referencia a id en Proveedores.

3. Relación entre Ordenes y Usuarios

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Una orden es realizada por un usuario. Un usuario puede realizar varias órdenes.
- **Clave Foránea:** usuario_id en la tabla Ordenes hace referencia a id en Usuarios.

4. Relación entre Detalles_Ordenes, Ordenes y Productos

- **Tipo de Relación:**
 - Uno a Muchos entre Detalles_Ordenes y Ordenes.
 - Uno a Muchos entre Detalles_Ordenes y Productos.

- **Descripción:**
 - Cada detalle de una orden pertenece a una orden específica.
 - Cada detalle de una orden contiene un producto específico.
- **Claves Foráneas:**
 - orden_id en la tabla Detalles_Ordenes hace referencia a id en Ordenes.
 - producto_id en la tabla Detalles_Ordenes hace referencia a id en Productos.

5. Relación entre Pagos y Ordenes

- **Tipo de Relación:** Uno a Uno.
- **Descripción:** Cada pago está relacionado con una orden específica.
- **Clave Foránea:** orden_id en la tabla Pagos hace referencia a id en Ordenes.

6. Relación entre Pagos y Metodos_Pago

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Un pago está hecho con un método de pago específico. Un método de pago puede ser utilizado en muchos pagos.
- **Clave Foránea:** metodo_pago_id en la tabla Pagos hace referencia a id en Metodos_Pago.

7. Relación entre Envios y Ordenes

- **Tipo de Relación:** Uno a Uno.
- **Descripción:** Cada envío está relacionado con una orden específica.
- **Clave Foránea:** orden_id en la tabla Envios hace referencia a id en Ordenes.

8. Relación entre Carritos_Compra y Usuarios (Uno a Uno)

- **Tipo de Relación:** Uno a Uno.

- **Descripción:** Cada usuario tiene un único carrito de compra.
- **Clave Foránea:** usuario_id en la tabla Carritos_Compra hace referencia a id en Usuarios.

9. Relación entre Detalles_Carrito y Carritos_Compra

- **Tipo de Relación:** Uno a Mucho).
- **Descripción:** Un carrito de compra puede tener varios detalles (productos).
- **Clave Foránea:** carrito_id en la tabla Detalles_Carrito hace referencia a id en Carritos_Compra.

10. Relación entre Detalles_Carrito y Productos

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Cada detalle del carrito se refiere a un producto específico.
- **Clave Foránea:** producto_id en la tabla Detalles_Carrito hace referencia a id en Productos.

11. Relación entre Reseñas_Productos y Usuarios

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Un usuario puede realizar varias reseñas.
- **Clave Foránea:** usuario_id en la tabla Reseñas_Productos hace referencia a id en Usuarios.

12. Relación entre Reseñas_Productos y Productos

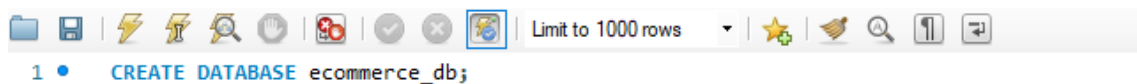
- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Un producto puede tener varias reseñas.
- **Clave Foránea:** producto_id en la tabla Reseñas_Productos hace referencia a id en Productos.

13. Relación entre Direcciones_Envio y Usuarios

- **Tipo de Relación:** Uno a Muchos.
- **Descripción:** Un usuario puede tener varias direcciones de envío.
- **Clave Foránea:** usuario_id en la tabla Direcciones_Envio hace referencia a id en Usuarios.

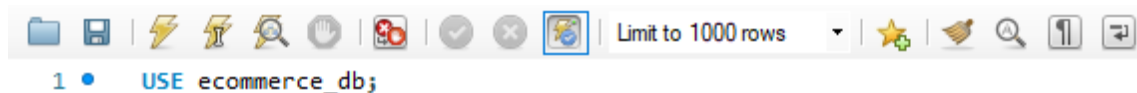
Mostrare de forma breve como cree la base de datos:

Primero: se usa el comando **CREATE DATABASE** seguidamente el nombre de la base de datos, en este caso **ecommerce_db**. Así:



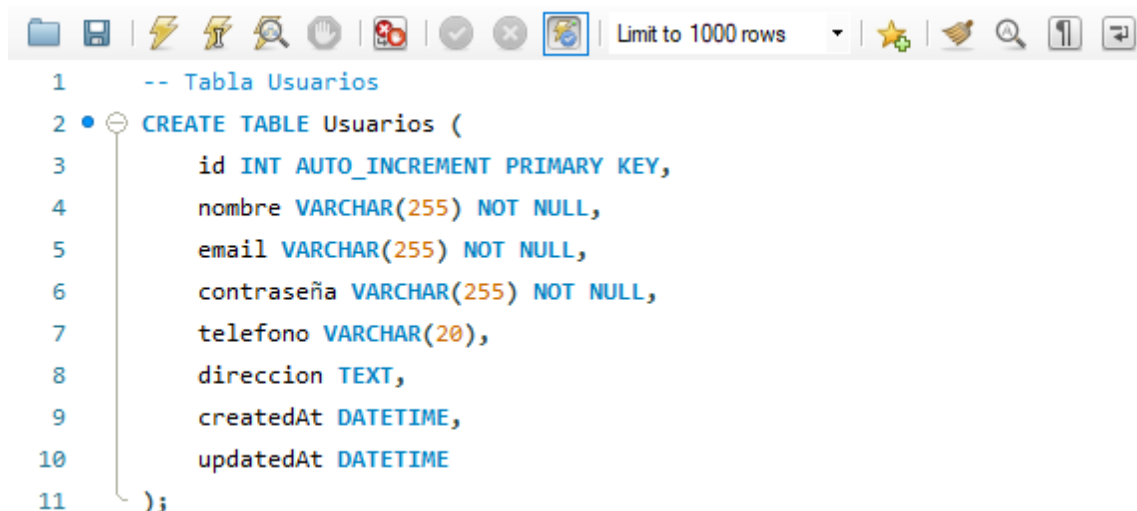
A screenshot of a SQL editor interface. The toolbar at the top includes icons for file operations, execution, and search. Below the toolbar, a single line of SQL code is displayed: `1 • CREATE DATABASE ecommerce_db;`. The text is in a blue monospace font.

Segundo: usando el comando **USE** seguidamente el nombre de la base de datos previamente definido. Así:



A screenshot of a SQL editor interface. The toolbar at the top includes icons for file operations, execution, and search. Below the toolbar, a single line of SQL code is displayed: `1 • USE ecommerce_db;`. The text is in a blue monospace font.

Tercero: Creamos las tablas y sus relaciones de la siguiente manera:



A screenshot of a SQL editor interface. The toolbar at the top includes icons for file operations, execution, and search. Below the toolbar, a block of SQL code is displayed, creating a table named 'Usuarios'. The code is as follows:

```

1      -- Tabla Usuarios
2 • CREATE TABLE Usuarios (
3         id INT AUTO_INCREMENT PRIMARY KEY,
4         nombre VARCHAR(255) NOT NULL,
5         email VARCHAR(255) NOT NULL,
6         contraseña VARCHAR(255) NOT NULL,
7         telefono VARCHAR(20),
8         direccion TEXT,
9         createdAt DATETIME,
10        updatedAt DATETIME
11    );

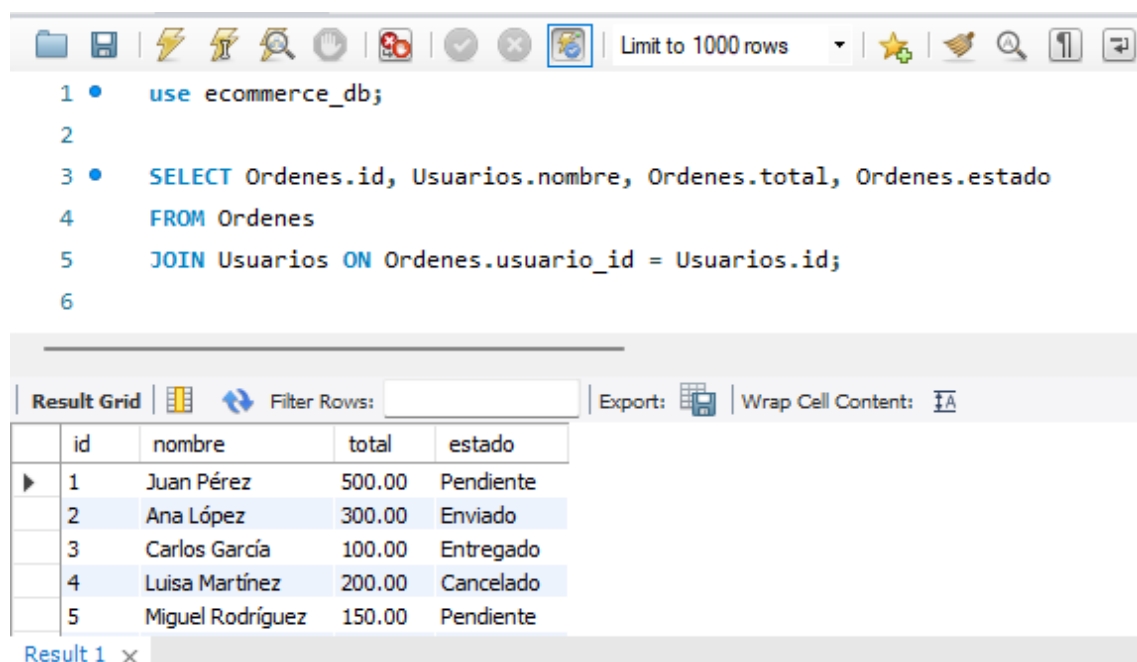
```

Consultas Realizadas

Mostrare las consultas que podemos realizar a la base de datos, claro que debemos tener datos o un registro en la base de datos que creamos, este paso lo omití porque es muy fácil de insertar registros a las tablas, con el comando INSERT INTO nombre de la tabla, mención de atributos, ya podemos insertar nuestros registros.

Procedo a mostrar los registros usando consultas vistas en clases:

Consulta 1: Mostrar información de todas las órdenes junto con el nombre del usuario que las realizó, para ello se hace de la siguiente manera:



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and search. The SQL query is as follows:

```
1 • use ecommerce_db;
2
3 • SELECT Ordenes.id, Usuarios.nombre, Ordenes.total, Ordenes.estado
4 FROM Ordenes
5 JOIN Usuarios ON Ordenes.usuario_id = Usuarios.id;
6
```

Below the query, the 'Result Grid' tab is active, displaying the results of the query. The grid has columns for 'id', 'nombre', 'total', and 'estado'. The results are as follows:

	id	nombre	total	estado
▶	1	Juan Pérez	500.00	Pendiente
	2	Ana López	300.00	Enviado
	3	Carlos García	100.00	Entregado
	4	Luisa Martínez	200.00	Cancelado
	5	Miguel Rodríguez	150.00	Pendiente

At the bottom, there is a tab labeled 'Result 1' with a close button (x).

Consulta 2: Listar todos los métodos de pago utilizados en las órdenes, para ello se hace de la siguiente manera:

1	•	use ecommerce_db;
2		
3	•	SELECT Ordenes.id, Metodos_Pago.nombre, Pagos.monto, Pagos.estado_pago
4		FROM Pagos
5		JOIN Ordenes ON Pagos.orden_id = Ordenes.id
6		JOIN Metodos_Pago ON Pagos.metodo_pago_id = Metodos_Pago.id;

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
id	nombre	monto	estado_pago
1	Tarjeta de Crédito	500.00	Pendiente
2	PayPal	300.00	Completado
3	Transferencia Bancaria	100.00	Completado
4	Crédito en Tienda	200.00	Fallido
5	Bitcoin	150.00	Pendiente
6	Tarjeta de Débito	75.00	Completado
7	Pago contra entrega	400.00	Completado
8	Google Pay	350.00	Pendiente
9	Apple Pay	600.00	Completado
10	PSE	250.00	Pendiente

Consulta 3: Consultar detalles de los productos y proveedores para una orden específica, para ello se hace de la siguiente manera:

<

Consulta 4: Ver las órdenes pendientes de ser enviadas

```
1 use ecommerce_db;
2
3 SELECT Ordenes.id, Usuarios.nombre, Ordenes.total, Envios.estado_envio
4 FROM Ordenes
5 JOIN Usuarios ON Ordenes.usuario_id = Usuarios.id
6 JOIN Envios ON Ordenes.id = Envios.orden_id
7 WHERE Envios.estado_envio = 'Pendiente';
```

Result Grid

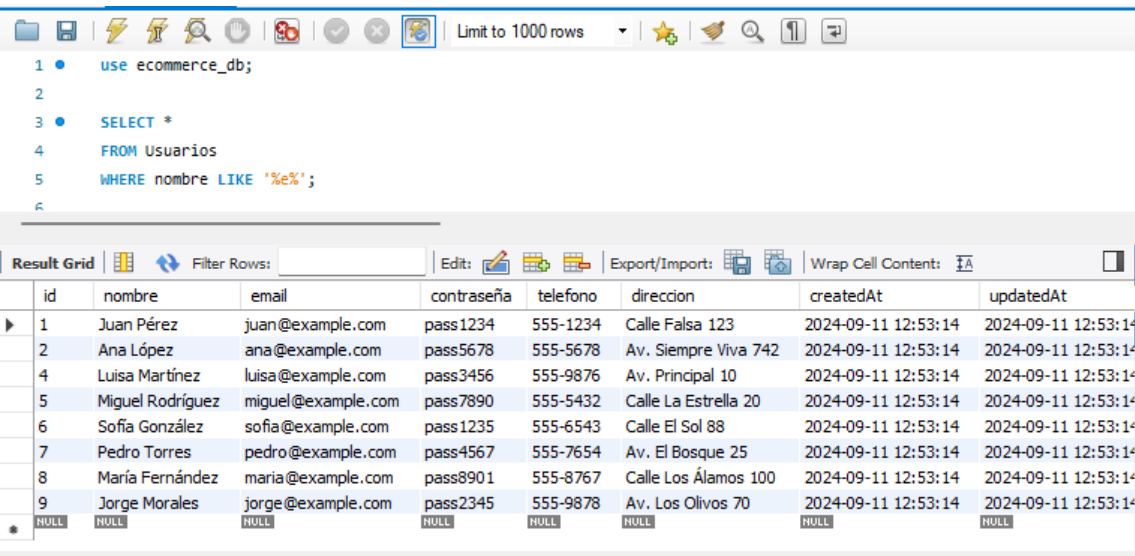
	id	nombre	total	estado_envio
▶	1	Juan Pérez	500.00	Pendiente
	1	Juan Pérez	500.00	Pendiente
	1	Juan Pérez	500.00	Pendiente
	4	Luisa Martínez	200.00	Pendiente
	4	Luisa Martínez	200.00	Pendiente
	5	Miguel Rodríguez	150.00	Pendiente
	9	Jorge Morales	600.00	Pendiente

Consulta 5: Buscar registros donde el nombre del usuario comienza con la letra 'A'.

Explicación:

- **LIKE 'A%':** Busca todos los registros en la tabla Usuarios donde el nombre comienza con la letra 'A'.

Consulta 7: Buscar registros donde el nombre del usuario contiene la letra 'e'



The screenshot shows a database query tool interface. The SQL query is as follows:

```
1 use ecommerce_db;
2
3 SELECT *
4 FROM Usuarios
5 WHERE nombre LIKE '%e%';
6
```

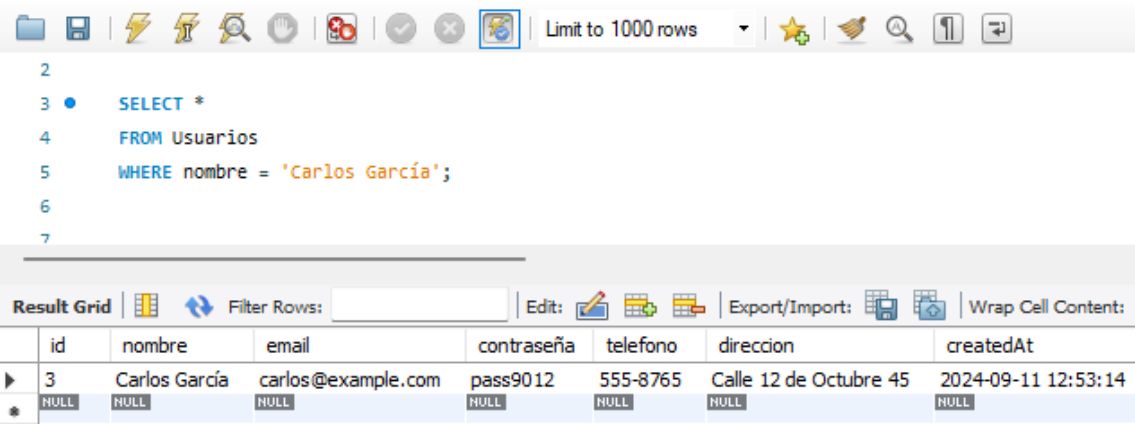
The results are displayed in a table with the following columns: id, nombre, email, contraseña, telefono, direccion, createdAt, and updatedAt. The table contains 9 rows of data, all of which include the letter 'e' in the 'nombre' column.

id	nombre	email	contraseña	telefono	direccion	createdAt	updatedAt
1	Juan Pérez	juan@example.com	pass1234	555-1234	Calle Falsa 123	2024-09-11 12:53:14	2024-09-11 12:53:14
2	Ana López	ana@example.com	pass5678	555-5678	Av. Siempre Viva 742	2024-09-11 12:53:14	2024-09-11 12:53:14
4	Luisa Martínez	luisa@example.com	pass3456	555-9876	Av. Principal 10	2024-09-11 12:53:14	2024-09-11 12:53:14
5	Miguel Rodríguez	miguel@example.com	pass7890	555-5432	Calle La Estrella 20	2024-09-11 12:53:14	2024-09-11 12:53:14
6	Sofía González	sofia@example.com	pass1235	555-6543	Calle El Sol 88	2024-09-11 12:53:14	2024-09-11 12:53:14
7	Pedro Torres	pedro@example.com	pass4567	555-7654	Av. El Bosque 25	2024-09-11 12:53:14	2024-09-11 12:53:14
8	María Fernández	maria@example.com	pass8901	555-8767	Calle Los Álamos 100	2024-09-11 12:53:14	2024-09-11 12:53:14
9	Jorge Morales	jorge@example.com	pass2345	555-9878	Av. Los Olivos 70	2024-09-11 12:53:14	2024-09-11 12:53:14
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Explicación:

- **LIKE '%e%'**: Busca todos los registros en la tabla Usuarios donde el nombre contiene la letra 'e' en cualquier posición.

Consulta 8: Buscar un usuario con nombre exacto "Carlos García"



The screenshot shows a database query tool interface. The SQL query is as follows:

```
2
3 SELECT *
4 FROM Usuarios
5 WHERE nombre = 'Carlos García';
6
7
```

The results are displayed in a table with the following columns: id, nombre, email, contraseña, telefono, direccion, and createdAt. The table contains 1 row of data, which is the user 'Carlos García'.

id	nombre	email	contraseña	telefono	direccion	createdAt
3	Carlos García	carlos@example.com	pass9012	555-8765	Calle 12 de Octubre 45	2024-09-11 12:53:14
*	NULL	NULL	NULL	NULL	NULL	NULL

Explicación:

- **WHERE nombre = 'Carlos García'**: Encuentra el registro exacto en la tabla Usuarios donde el nombre es "Carlos García".

Consulta 9: Buscar registros donde el nombre del usuario esté entre las letras 'A' y 'M'

The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is entered in a text area:

```

2
3 • SELECT *
4 FROM Usuarios
5 WHERE nombre BETWEEN 'A' AND 'M';
6
7

```

Below the query editor, there's a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Edit' button, and an 'Export/Import' button. The main area displays a table of results with the following columns: id, nombre, email, contraseña, telefono, direccion, and createdAt. The table contains 10 rows of data, with the first 6 rows highlighted in blue. The last row is marked with an asterisk and 'NULL' values.

	id	nombre	email	contraseña	telefono	direccion	createdAt
▶	1	Juan Pérez	juan@example.com	pass1234	555-1234	Calle Falsa 123	2024-09-11 12:53:14
	2	Ana López	ana@example.com	pass5678	555-5678	Av. Siempre Viva 742	2024-09-11 12:53:14
	3	Carlos García	carlos@example.com	pass9012	555-8765	Calle 12 de Octubre 45	2024-09-11 12:53:14
	4	Luisa Martínez	luisa@example.com	pass3456	555-9876	Av. Principal 10	2024-09-11 12:53:14
	9	Jorge Morales	jorge@example.com	pass2345	555-9878	Av. Los Olivos 70	2024-09-11 12:53:14
	10	Laura Castillo	laura@example.com	pass6789	555-5433	Calle El Prado 15	2024-09-11 12:53:14
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Explicación:

- **BETWEEN 'A' AND 'M'**: Devuelve los registros donde el nombre del usuario esté alfabéticamente entre las letras 'A' y 'M'.

Consideraciones de Diseño

Nombre de la Base de Datos y Descripción

Nombre de la Base de Datos: ecommerce

Descripción: La base de datos ecommerce está diseñada para gestionar todos los aspectos esenciales de un sistema de comercio electrónico. Incluye entidades para usuarios, productos, proveedores, órdenes, pagos, y más, facilitando así el manejo eficiente de información relacionada con compras, ventas, y administración de productos.

Entidades y Atributos

Entidades:

1. Usuarios

- **Atributos:**

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del usuario.
- nombre (VARCHAR(255)): Nombre completo del usuario.
- email (VARCHAR(255)): Dirección de correo electrónico del usuario.
- contraseña (VARCHAR(255)): Contraseña del usuario (almacenada de manera segura).
- telefono (VARCHAR(20)): Número de teléfono del usuario.
- direccion (TEXT): Dirección de residencia del usuario.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

2. Categorías

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único de la categoría.
- nombre (VARCHAR(255)): Nombre de la categoría.
- descripcion (TEXT): Descripción de la categoría.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

3. Proveedores

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del proveedor.
- nombre (VARCHAR(255)): Nombre del proveedor.
- direccion (TEXT): Dirección del proveedor.
- telefono (VARCHAR(20)): Número de teléfono del proveedor.
- email (VARCHAR(255)): Dirección de correo electrónico del proveedor.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

4. Productos

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del producto.
- nombre (VARCHAR(255)): Nombre del producto.
- descripcion (TEXT): Descripción del producto.
- precio (DECIMAL(10, 2)): Precio del producto.
- stock (INT): Cantidad disponible en inventario.
- categoria_id (INT, FOREIGN KEY): Clave foránea referenciando la categoría.
- proveedor_id (INT, FOREIGN KEY): Clave foránea referenciando el proveedor.
- createdAt (DATETIME): Fecha y hora de creación del registro.

- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

5. Ordenes

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único de la orden.
- usuario_id (INT, FOREIGN KEY): Clave foránea referenciando al usuario.
- estado (ENUM): Estado de la orden (Pendiente, Enviado, Entregado, Cancelado).
- total (DECIMAL(10, 2)): Total de la orden.
- direccion_envio (TEXT): Dirección de envío de la orden.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

6. Detalles_Ordenes

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del detalle de la orden.
- orden_id (INT, FOREIGN KEY): Clave foránea referenciando la orden.
- producto_id (INT, FOREIGN KEY): Clave foránea referenciando el producto.
- cantidad (INT): Cantidad del producto en la orden.
- precio_unitario (DECIMAL(10, 2)): Precio unitario del producto.

- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

7. Metodos_Pago

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del método de pago.
- nombre (VARCHAR(255)): Nombre del método de pago.
- descripcion (TEXT): Descripción del método de pago.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

8. Pagos

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del pago.
- orden_id (INT, FOREIGN KEY): Clave foránea referenciando la orden.
- metodo_pago_id (INT, FOREIGN KEY): Clave foránea referenciando el método de pago.
- monto (DECIMAL(10, 2)): Monto del pago.
- estado_pago (ENUM): Estado del pago (Pendiente, Completado, Fallido).
- createdAt (DATETIME): Fecha y hora de creación del registro.

- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

9. Envios

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del envío.
- orden_id (INT, FOREIGN KEY): Clave foránea referenciando la orden.
- direccion_envio (TEXT): Dirección de envío.
- estado_envio (ENUM): Estado del envío (Pendiente, Enviado, Entregado).
- fecha_envio (DATETIME): Fecha del envío.
- fecha_entrega (DATETIME): Fecha de entrega.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

10. Carritos_Compra

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del carrito.
- usuario_id (INT, FOREIGN KEY): Clave foránea referenciando al usuario.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

11. Detalles_Carrito

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del detalle del carrito.
- carrito_id (INT, FOREIGN KEY): Clave foránea referenciando el carrito.
- producto_id (INT, FOREIGN KEY): Clave foránea referenciando el producto.
- cantidad (INT): Cantidad del producto en el carrito.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

12. Descuentos

○ Atributos:

- id (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del descuento.
- codigo (VARCHAR(50)): Código del descuento.
- porcentaje_descuento (DECIMAL(5, 2)): Porcentaje del descuento.
- fecha_inicio (DATETIME): Fecha de inicio del descuento.
- fecha_fin (DATETIME): Fecha de fin del descuento.
- createdAt (DATETIME): Fecha y hora de creación del registro.
- updatedAt (DATETIME): Fecha y hora de la última actualización del registro.

13. Reseñas_Productos

- **Atributos:**

- **id** (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único de la reseña.
- **usuario_id** (INT, FOREIGN KEY): Clave foránea referenciando al usuario.
- **producto_id** (INT, FOREIGN KEY): Clave foránea referenciando el producto.
- **comentario** (TEXT): Comentario de la reseña.
- **calificacion** (INT): Calificación del producto (de 1 a 5).
- **createdAt** (DATETIME): Fecha y hora de creación del registro.
- **updatedAt** (DATETIME): Fecha y hora de la última actualización del registro.

14. **Direcciones_Envio**

- **Atributos:**

- **id** (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único de la dirección de envío.
- **usuario_id** (INT, FOREIGN KEY): Clave foránea referenciando al usuario.
- **direccion** (TEXT): Dirección de envío.
- **es_principal** (BOOLEAN): Indica si es la dirección principal del usuario.
- **createdAt** (DATETIME): Fecha y hora de creación del registro.
- **updatedAt** (DATETIME): Fecha y hora de la última actualización del registro.

Claves Primarias y Relaciones entre Tablas

Claves Primarias:

- Cada tabla tiene una columna id que actúa como clave primaria, garantizando la unicidad de cada registro.

Relaciones entre Tablas:

1. Usuarios y Ordenes

- **Relación:** Uno a Muchos
- **Descripción:** Un usuario puede tener muchas órdenes. La tabla Ordenes tiene una clave foránea usuario_id que hace referencia a Usuarios(id).

2. Ordenes y Detalles_Ordenes

- **Relación:** Uno a Muchos
- **Descripción:** Una orden puede tener muchos detalles. La tabla Detalles_Ordenes tiene una clave foránea orden_id que hace referencia a Ordenes(id).

3. Productos y Detalles_Ordenes

- **Relación:** Uno a Muchos
- **Descripción:** Un producto puede aparecer en muchos detalles de órdenes. La tabla Detalles_Ordenes tiene una clave foránea producto_id que hace referencia a Productos(id).

4. Productos y Categorías

- **Relación:** Muchos a Uno
- **Descripción:** Un producto pertenece a una categoría. La tabla Productos tiene una clave foránea categoria_id que hace referencia a Categorías(id).

5. Productos y Proveedores

- **Relación:** Muchos a Uno

- **Descripción:** Un producto es suministrado por un proveedor. La tabla Productos tiene una clave foránea proveedor_id que hace referencia a Proveedores(id).

6. Ordenes y Pagos

- **Relación:** Uno a Uno
- **Descripción:** Cada orden tiene un pago asociado. La tabla Pagos tiene una clave foránea orden_id que hace referencia a Ordenes(id).

7. Ordenes y Envios

- **Relación:** Uno a Uno
- **Descripción:** Cada orden tiene un envío asociado. La tabla Envios tiene una clave foránea orden_id que hace referencia a Ordenes(id).

8. Usuarios y Carritos_Compra

- **Relación:** Uno a Uno
- **Descripción:** Cada usuario tiene un carrito de compras. La tabla Carritos_Compra tiene una clave foránea usuario_id que hace referencia a Usuarios(id).

9. Carritos_Compra y Detalles_Carrito

- **Relación:** Uno a Muchos
- **Descripción:** Un carrito puede tener muchos detalles. La tabla Detalles_Carrito tiene una clave foránea carrito_id que hace referencia a Carritos_Compra(id).

10. Productos y Reseñas_Productos

- **Relación:** Uno a Muchos
- **Descripción:** Un producto puede tener muchas reseñas. La tabla Reseñas_Productos tiene una clave foránea producto_id que hace referencia a Productos(id).

11. Usuarios y Reseñas_Productos

- **Relación:** Uno a Muchos
- **Descripción:** Un usuario puede escribir muchas reseñas. La tabla Reseñas_Productos tiene una clave foránea usuario_id que hace referencia a Usuarios(id).

12. Usuarios y Direcciones_Envio

- **Relación:** Uno a Muchos
- **Descripción:** Un usuario puede tener muchas direcciones de envío. La tabla Direcciones_Envio tiene una clave foránea usuario_id que hace referencia a Usuarios(id).

De momento, es todo, este seria el primer avance, queda pendiente el diagrama.

Anexo el repositorio de GitHub:

https://github.com/1007748140/BD_REPOSITORIO.git