Creación y Gestión de Base de Datos para una Tienda Virtual en MySQL

Presentado por:

Carlos Andrés Montoya Hoyos

Presentado A:

Brayan Arcos Burbano

Instituto Tecnológico del Putumayo

Desarrollo de Base de Datos

2024-10-25

Índice

[1. Resumen Ejecutivo 3](#_Toc180795541)

[2. Introducción 3](#_Toc180795542)

[Contexto y Motivación 3](#_Toc180795543)

[Alcance del Informe 3](#_Toc180795544)

[3. Metodología 3](#_Toc180795545)

[Herramientas Utilizadas 3](#_Toc180795546)

[Procedimientos 3](#_Toc180795547)

[4. Desarrollo del Informe 4](#_Toc180795548)

[Descripción de la Base de Datos 4](#_Toc180795549)

[Tablas de la Base de Datos 4](#_Toc180795550)

[Normalización 4](#_Toc180795551)

[Cardinalidad 5](#_Toc180795552)

[Diagrama 6](#_Toc180795553)

[5. Análisis y Discusión 7](#_Toc180795554)

[6. Conclusiones 7](#_Toc180795555)

[7. Recomendaciones 8](#_Toc180795556)

[Capacitación 8](#_Toc180795557)

[Mantenimiento Regular 8](#_Toc180795558)

[Evaluación de Necesidades Futuras 8](#_Toc180795559)

[8. Referencias 8](#_Toc180795560)

# 1. Resumen Ejecutivo

El presente informe documenta el diseño, desarrollo y análisis de una base de datos creada para la gestión de una tienda virtual, denominada prueba. La base de datos centraliza la información de ciudades, personas, usuarios, roles, productos, proveedores, almacenes, facturas, órdenes, métodos de pago y su historial. Se presenta un análisis detallado de la estructura de datos, junto con recomendaciones para su mantenimiento y escalabilidad futura. Este informe tiene como objetivo facilitar la administración de la tienda y mejorar la eficiencia en la organización de datos.

# 2. Introducción

### Contexto y Motivación

Con el crecimiento de las tiendas en línea, es esencial contar con una base de datos estructurada que gestione adecuadamente los datos de productos, clientes, órdenes y pagos. La base de datos prueba ha sido desarrollada para abordar estos requerimientos, buscando ofrecer una solución centralizada y escalable.

### Alcance del Informe

Este informe abarca desde la construcción de la base de datos, su estructura, hasta el análisis de normalización y cardinalidad. También se incluye un diagrama de relaciones y recomendaciones para su optimización futura.

# 3. Metodología

### Herramientas Utilizadas

* **MySQL**: Se utilizó para la creación, modificación y consulta de la base de datos.
* **Workbench**: Para el diseño del diagrama relacional y la ejecución de pruebas.
* **Lenguaje SQL**: Para la creación de tablas e inserción de datos.

### Procedimientos

El desarrollo de la base de datos se llevó a cabo en los siguientes pasos:

1. Diseño y creación de tablas.
2. Definición de relaciones y restricciones.
3. Inserción de datos para pruebas y validación.
4. Revisión de integridad referencial y normalización.
5. Análisis y documentación del diseño final.

# 4. Desarrollo del Informe

### Descripción de la Base de Datos

La base de datos prueba está compuesta por varias tablas para almacenar la información de la tienda virtual, como clientes, productos, inventario, órdenes y pagos.

### Tablas de la Base de Datos

1. **City**: Almacena información sobre las ciudades.
2. **Document Type**: Define los tipos de documento.
3. **People**: Almacena datos personales de clientes y empleados.
4. **People Phone**: Detalla los números de contacto de las personas.
5. **Users**: Contiene información de los usuarios de la tienda.
6. **Roles**: Define los roles disponibles.
7. **User Roles**: Relaciona usuarios con roles.
8. **Product Categories**: Clasificación de productos.
9. **Products**: Información sobre productos en inventario.
10. **Product Details**: Detalles adicionales de productos.
11. **Supplier**: Información de proveedores.
12. **Supplier Product**: Relación entre proveedores y productos.
13. **Store**: Datos sobre los almacenes.
14. **Store Products**: Relación entre almacenes y productos.
15. **Invoices**: Registro de facturas emitidas.
16. **Invoice Details**: Detalles de cada factura.
17. **Orders**: Registro de órdenes de compra.
18. **Order Details**: Detalles de cada orden.
19. **Payment Method**: Métodos de pago permitidos.
20. **Payments**: Información sobre pagos realizados.
21. **Payment Status History**: Histórico de estados de los pagos.

### Normalización

La normalización es un proceso que se utiliza en el diseño de bases de datos para reducir la redundancia de datos y mejorar la integridad. Generalmente, se divide en varias formas normales (1NF, 2NF, 3NF).

#### Primera Forma Normal (1NF)

Para cumplir con la 1NF, una tabla debe:

* Contener solo valores atómicos: Cada celda debe contener un solo valor.
* Tener filas únicas: No puede haber duplicados en las filas.
* Evaluación de tu base de datos:

Cumple: Todas las tablas tienen columnas con valores atómicos. Por ejemplo, la tabla people tiene campos como firstName, middleName, lastName, y cada uno contiene un solo valor.

#### Segunda Forma Normal (2NF)

Para cumplir con la 2NF, una tabla debe:

* Cumplir con la 1NF.
* Tener todas las columnas no clave dependientes de la clave primaria: Esto significa que si una tabla tiene una clave compuesta, todas las columnas deben depender completamente de esta clave.

Cumple: No hay tablas con claves compuestas en la estructura dada. Cada tabla tiene una clave primaria simple (como id), y las columnas dependen completamente de esa clave.

#### Tercera Forma Normal (3NF)

Para cumplir con la 3NF, una tabla debe:

* Cumplir con la 2NF.
* No tener dependencias transitivas: Esto significa que no puede haber columnas que dependan de otra columna no clave.

Cumple: Todas las columnas de las tablas dependen únicamente de la clave primaria. Por ejemplo, en la tabla people, los campos como firstName, middleName, y lastName dependen de la clave id, y no hay ninguna dependencia transitiva (como una columna que depende de otra columna que no es clave).

### Cardinalidad

La cardinalidad describe la relación entre las tablas y especifica cuántas filas de una tabla pueden estar relacionadas con filas de otra tabla. Las cardinalidades comunes son:

* **1:1** (uno a uno): Un registro en una tabla se relaciona con un único registro en otra tabla.
* **1**

(uno a muchos): Un registro en una tabla se relaciona con múltiples registros en otra tabla.

* **N**

(muchos a muchos): Múltiples registros en una tabla se relacionan con múltiples registros en otra tabla.

* **Users (Usuarios) y Invoices (Facturas):**

Cardinalidad: 1

Justificación: Un usuario puede tener múltiples facturas, pero cada factura está asociada a un solo usuario.

* **Products (Productos) y Invoice Details (Detalles de Factura):**

Cardinalidad: 1

Justificación: Un producto puede aparecer en múltiples detalles de factura, pero cada detalle de factura se refiere a un único producto.

* **Products (Productos) y Order Details (Detalles de Orden):**

Cardinalidad: 1

Justificación: Un producto puede estar en muchas órdenes, pero cada detalle de orden se relaciona con un único producto.

* **Orders (Órdenes) y Users (Usuarios):**

Cardinalidad: 1

Justificación: Un usuario puede realizar múltiples órdenes, pero cada orden pertenece a un solo usuario.

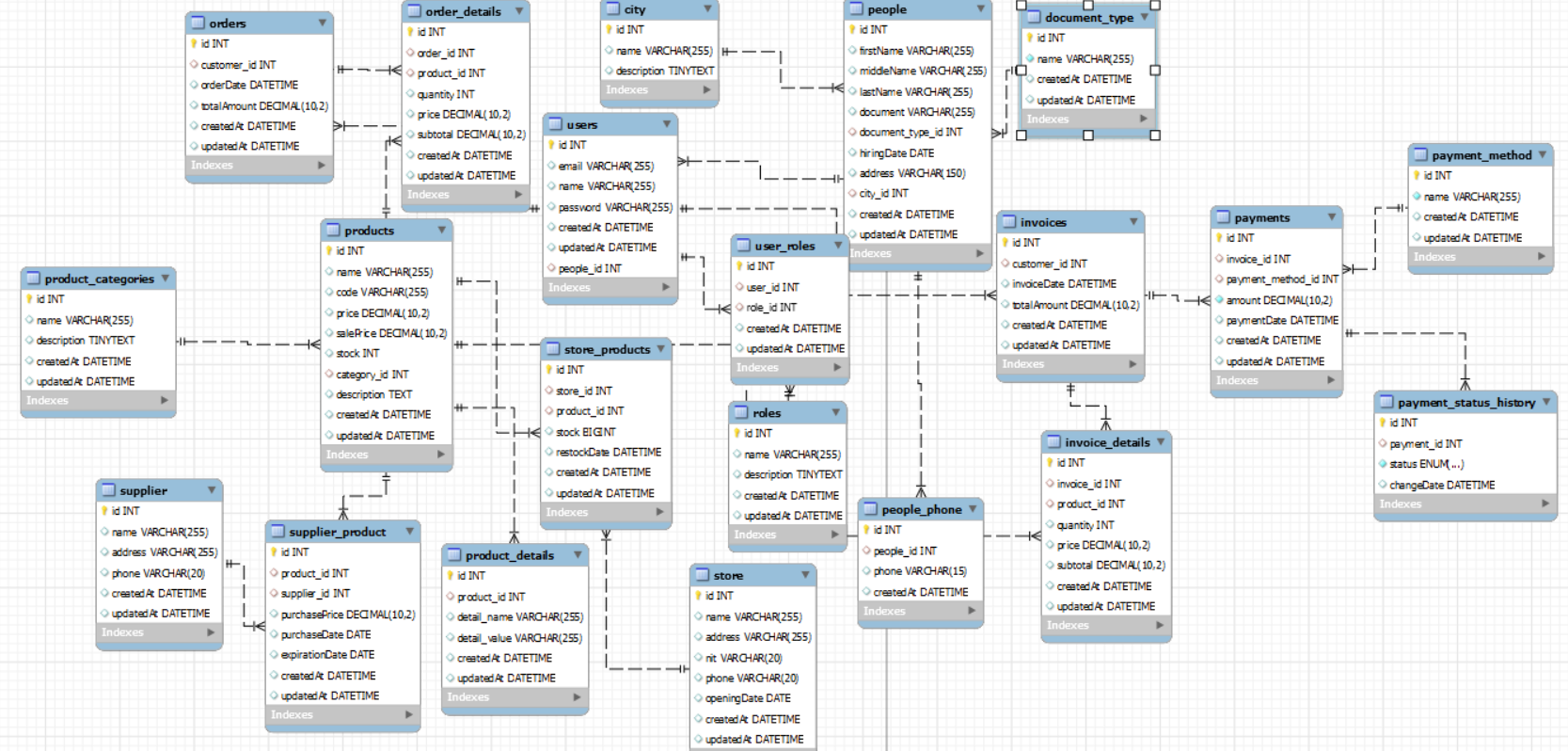
* **Users (Usuarios) y Roles (Roles):**

Cardinalidad: N

Justificación: Un usuario puede tener múltiples roles y un rol puede ser asignado a múltiples usuarios. Esto se maneja mediante la tabla user\_roles.

### Diagrama

A continuación, se presenta el diagrama de la base de datos



# 5. Análisis y Discusión

La base de datos prueba fue diseñada teniendo en cuenta la eficiencia y la facilidad de administración de datos de la tienda. Las relaciones y el modelo de datos permiten realizar consultas rápidas para obtener información sobre clientes, productos, órdenes y pagos. La estructura normalizada asegura que los datos se almacenen sin redundancia y se mantenga la integridad.

# 6. Conclusiones

La base de datos prueba cumple con los requerimientos de una tienda virtual moderna, gestionando eficazmente los datos relacionados con clientes, inventario, proveedores y órdenes. La normalización hasta la 3NF y las relaciones bien definidas entre tablas garantizan la consistencia de los datos y su rápida recuperación.

# 7. Recomendaciones

### Capacitación

Capacitar al personal en la gestión de consultas SQL avanzadas y en las estructuras de la base de datos para mejorar el uso de la información disponible.

### Mantenimiento Regular

Implementar un sistema de mantenimiento preventivo para actualizar la base de datos y hacer copias de seguridad.

### Evaluación de Necesidades Futuras

Monitorear el crecimiento de datos y considerar la escalabilidad de la base de datos, evaluando la posible incorporación de optimizaciones o migración a otros sistemas.

# 8. Referencias

* Documentación de MySQL: <https://dev.mysql.com/doc/>
* Guía de normalización en bases de datos
* Manual de usuario de MySQL Workbench