



Universidad Tecnológica Nacional

Unidad 3: TP - Fundamentos del Modelo Relacional

Alumno: Calcatelli Renzo - rcalcatelli@gmail.com

Comisión: M2025-1

Materia: Bases de Datos I

Profesor: Gustavo Sturtz

Trabajo Práctico: Diseño e Implementación de una Base de Datos Relacional para una Tienda de Calzados

Este trabajo práctico se centrará en dos fases cruciales del diseño de bases de datos: el **Diseño Conceptual** (Diagrama Entidad-Relación) y la **Traducción al Modelo Lógico/Físico** (Creación de Tablas en SQL).

Objetivo: Diseñar un modelo conceptual (Diagrama Entidad-Relación) para una tienda de calzados y traducirlo a un modelo lógico/físico mediante la creación de tablas en SQL, aplicando los principios del modelo relacional.

Caso de Estudio: Una tienda de calzados necesita una base de datos para gestionar sus operaciones principales, incluyendo vendedores, productos (calzados) y clientes, registrando las ventas. Cada venta puede incluir uno o varios tipos de calzados, y es fundamental registrar quién realizó cada venta y qué productos específicos se vendieron en cada transacción.

Fase 1: Diseño Conceptual (Diagrama Entidad-Relación - E-R)

El Diagrama E-R es una representación visual y de alto nivel de las entidades (objetos importantes) del sistema y cómo se relacionan entre sí, independientemente de cómo se almacenarán físicamente los datos.

1. Identificación de Entidades

Basándonos en el caso de estudio, las principales entidades involucradas son:

- **Vendedor:** Representa a los empleados que realizan las ventas.
- **Cliente:** Representa a las personas que compran calzados en la tienda.
- **Calzado:** Representa los diferentes tipos de calzado disponibles para la venta (productos).
- **Venta:** Representa una transacción de compra completa.
- **Detalle_Venta:** Representa cada línea de un producto específico dentro de una venta (qué calzado se vendió y en qué cantidad dentro de una venta particular). Esta entidad es necesaria para manejar la relación "muchos a muchos" entre **Venta** y **Calzado**.

2. Definición de Atributos para Cada Entidad

Para cada entidad, identificamos los atributos relevantes, incluyendo los identificadores únicos que serán nuestras Claves Primarias (PK).

- **Vendedor**
 - **ID_Vendedor** (PK): Identificador único del vendedor. (Ejemplo: V001, V002)
 - **Nombre**: Nombre del vendedor.
 - **Apellido**: Apellido del vendedor.
 - **DNI**: Documento Nacional de Identidad del vendedor (puede ser una Clave Candidata o Alternativa).
 - **Fecha_Contratacion**: Fecha en que el vendedor fue contratado.
 - **Telefono**: Número de teléfono del vendedor.
 - **Email**: Correo electrónico del vendedor.
- **Cliente**
 - **ID_Cliente** (PK): Identificador único del cliente. (Ejemplo: C1001, C1002)
 - **Nombre**: Nombre del cliente.
 - **Apellido**: Apellido del cliente.
 - **Direccion**: Dirección de residencia del cliente.
 - **Telefono**: Número de teléfono del cliente.
 - **Email**: Correo electrónico del cliente.
- **Calzado**
 - **ID_Calzado** (PK): Identificador único del tipo de calzado. (Ejemplo: CALZ001, BOTA005)
 - **Nombre**: Nombre o modelo del calzado (ej. "Zapatilla Deportiva X", "Botín de Cuero").
 - **Marca**: Marca del calzado.
 - **Talla**: Talla del calzado (ej. 38, 42).
 - **Color**: Color del calzado.
 - **Precio_Unitario**: Precio de venta por unidad de este calzado.
 - **Stock**: Cantidad de unidades disponibles en el inventario.
- **Venta**
 - **ID_Venta** (PK): Identificador único de la venta. (Ejemplo: VTA2024001, VTA2024002)
 - **Fecha_Venta**: Fecha y hora en que se realizó la venta.
 - **Total_Venta**: Monto total de la venta.
 - **ID_Vendedor** (FK): Clave Foránea que referencia al **Vendedor** que realizó la venta.
 - **ID_Cliente** (FK): Clave Foránea que referencia al **Cliente** que realizó la compra (puede ser nulo si la venta es a un cliente no registrado).
- **Detalle_Venta**
 - **ID_Detalle** (PK - Sustituta): Identificador único de cada línea de detalle de venta. (Ejemplo: DV001, DV002)
 - **ID_Venta** (FK): Clave Foránea que referencia a la **Venta** a la que pertenece este detalle. Es parte de una Clave Primaria Compuesta junto con **ID_Calzado** para la unicidad de la relación.
 - **ID_Calzado** (FK): Clave Foránea que hace referencia al **Calzado** específico que se vendió. Es parte de una Clave Primaria Compuesta.
 - **Cantidad**: Cantidad de unidades de ese **Calzado** vendidas en esa **Venta**.
 - **Precio_Linea**: Precio total de esa línea de detalle (Cantidad * Precio_Unitario_al_momento_de_venta).

3. Establecimiento de Relaciones y Cardinalidad

Las relaciones describen cómo las entidades interactúan. La cardinalidad especifica cuántas instancias de una entidad se relacionan con cuántas instancias de otra.

- **Vendedor - Venta:**

- **Tipo de Relación:** "realiza"
- **Cardinalidad:** 1:N (Uno a Muchos)
 - Un **Vendedor** puede realizar muchas **Ventas**.
 - Una **Venta** es realizada por un único **Vendedor**.
- **Implementación:** **ID_Vendedor** como FK en la tabla **Venta**.

- **Cliente - Venta:**

- **Tipo de Relación:** "realiza" o "es_comprado_por"
- **Cardinalidad:** 1:N (Uno a Muchos)
 - Un **Cliente** puede realizar muchas **Ventas**.
 - Una **Venta** es realizada por un único **Cliente** (o ninguno si es una venta anónima, permitiendo **NULL** en la FK).
- **Implementación:** **ID_Cliente** como FK en la tabla **Venta**.

- **Venta - Detalle_Venta:**

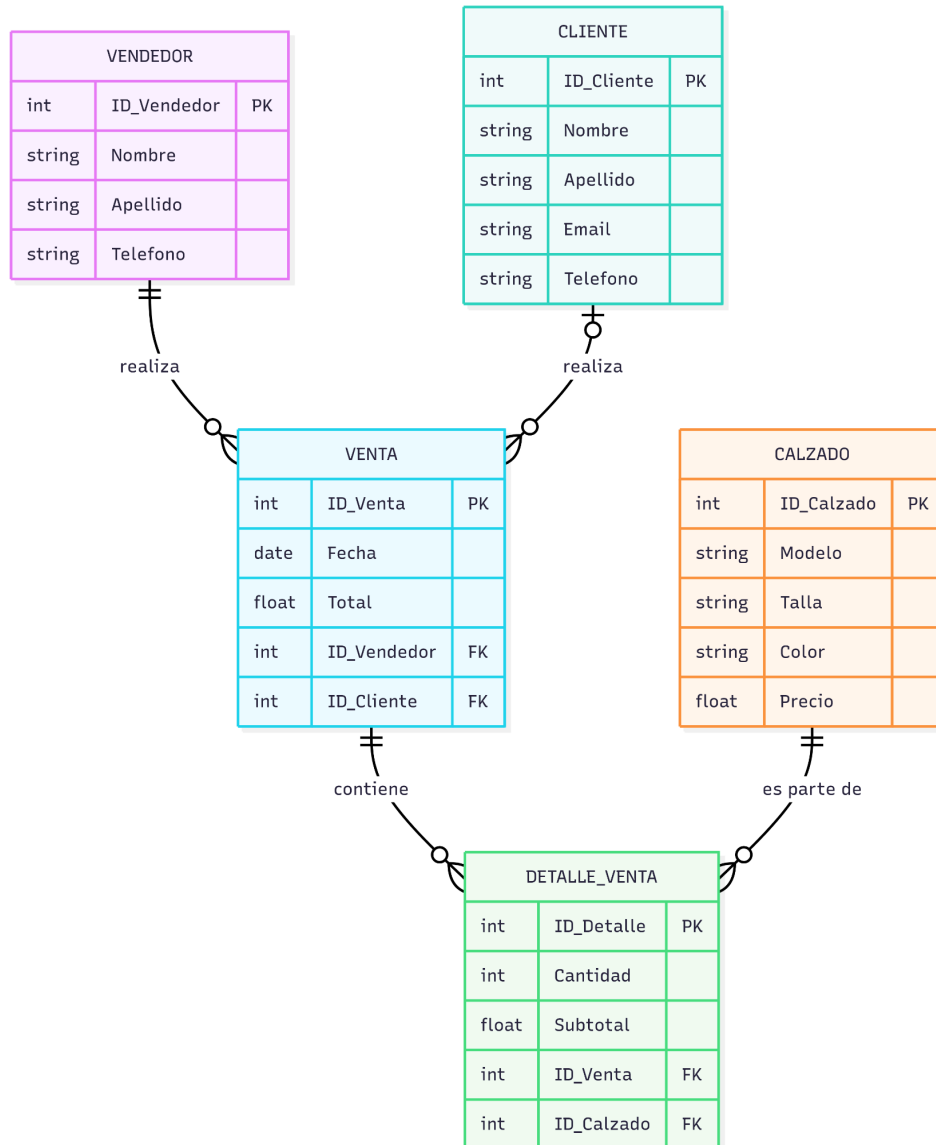
- **Tipo de Relación:** "contiene"
- **Cardinalidad:** 1:N (Uno a Muchos)
 - Una **Venta** puede contener muchos **Detalle_Venta** (líneas de productos).
 - Un **Detalle_Venta** pertenece a una única **Venta**.
- **Implementación:** **ID_Venta** como FK en la tabla **Detalle_Venta**.

- **Calzado - Detalle_Venta:**

- **Tipo de Relación:** "es_parte_de" o "incluye"
- **Cardinalidad:** 1:N (Uno a Muchos)
 - Un **Calzado** puede aparecer en muchos **Detalle_Venta** (es decir, venderse en muchas transacciones).
 - Un **Detalle_Venta** se refiere a un único **Calzado**.
- **Implementación:** **ID_Calzado** como FK en la tabla **Detalle_Venta**.
- **Nota sobre M:N:** La relación original "Una Venta puede incluir uno o varios tipos de Calzados" y "Un Calzado puede estar en varias Ventas" es intrínsecamente una relación M:N (Muchos a Muchos) entre **Venta** y **Calzado**. Para resolverla en un modelo relacional, se introduce la tabla intermedia **Detalle_Venta**.

4. Dibujo del Diagrama E-R (Conceptual)

El diagrama Entidad-Relación (E-R) de la zapatería ha sido elaborado utilizando la **notación de pata de gallo**, la cual permite representar de manera visual y precisa las **cardinalidades** de cada relación entre entidades. En este modelo, un **vendedor** puede realizar muchas ventas, mientras que cada venta es atribuida a un único vendedor, reflejando una relación de **uno a muchos (1:N)**. De manera similar, un **cliente** puede efectuar cero o muchas ventas, siendo cada venta asociada a un solo cliente, lo que representa una relación de **cero o uno a muchos (0..1:N)**. Cada **venta** puede descomponerse en múltiples **detalles de venta**, cada uno correspondiente a un producto específico, configurando una relación **1:N** entre venta y detalle. Por último, un **calzado** puede encontrarse en muchos detalles de venta, y cada detalle se refiere a un único calzado, constituyendo igualmente una relación **1:N**. El uso de esta notación facilita la comprensión rápida de la realidad operativa de la zapatería y sirve como base para el diseño del modelo lógico de la base de datos, determinando la estructura de claves primarias, foráneas y las restricciones necesarias en SQL.



Fase 2: Traducción al Modelo Lógico/Físico (Esquema Relacional en SQL)

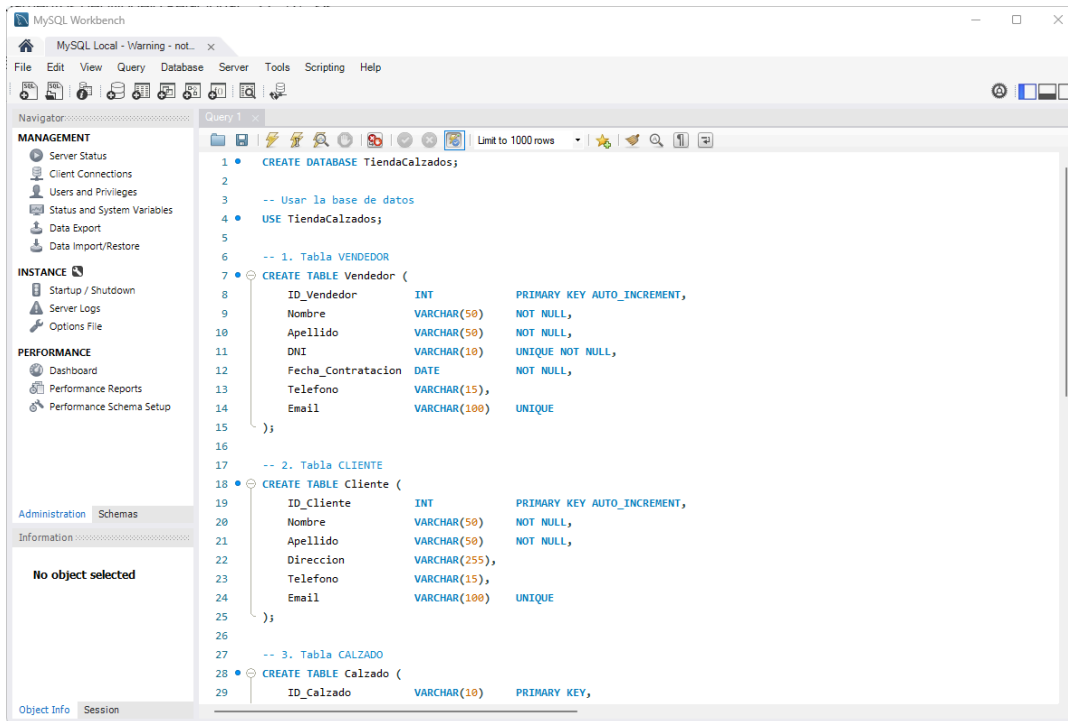
En esta fase, convertimos nuestro diseño conceptual en un esquema de base de datos concreto, definiendo las tablas, sus columnas (con tipos de datos), claves primarias y claves foráneas, y otras restricciones en lenguaje SQL.

Consideraciones Clave:

- **Claves Primarias (PK):** Serán los identificadores únicos para cada tabla.
- **Claves Foráneas (FK):** Establecerán las relaciones entre tablas. Utilizaremos las acciones referenciales **ON DELETE RESTRICT** y **ON UPDATE CASCADE** como ejemplo (puedes ajustarlas según necesidad).
 - **RESTRICT:** Impide la eliminación de una fila padre si tiene filas hijo referenciadas.
 - **CASCADE:** Si se actualiza el valor de una clave primaria en la tabla padre, ese cambio se propaga automáticamente a las claves foráneas correspondientes en las tablas hijo.
- **Tipos de Datos:** Seleccionaremos tipos de datos apropiados para cada atributo (ej. **VARCHAR** para texto, **INT** para números enteros, **DECIMAL** para moneda, **DATE** o **DATETIME** para fechas y horas).
- **Restricciones de Integridad:**
 - **NOT NULL:** Asegura que un campo no pueda estar vacío.
 - **UNIQUE:** Garantiza que todos los valores en una columna sean diferentes (ej. para DNI o Email si son Claves Candidatas).
 - **CHECK:** Permite definir una condición que los valores de una columna deben cumplir (ej. Stock ≥ 0).

Código SQL para la Creación de Tablas:

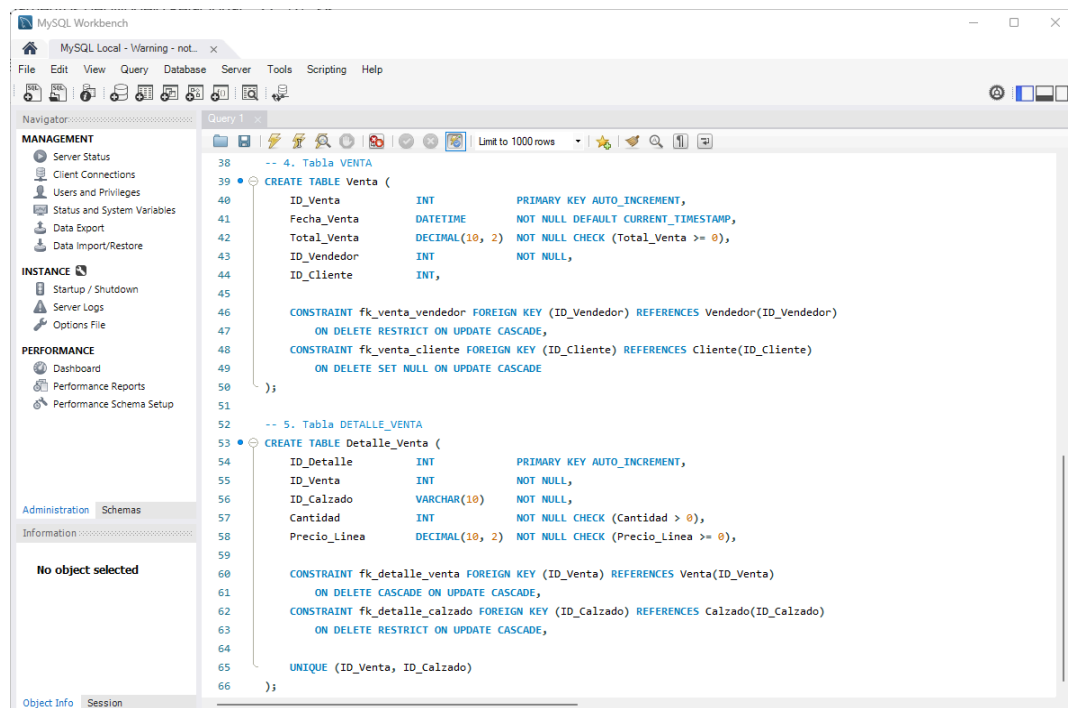
Paso 1: Preparación del Entorno



```

1  CREATE DATABASE TiendaCalzados;
2
3  -- Usar la base de datos
4  USE TiendaCalzados;
5
6  -- 1. Tabla VENDEDOR
7  CREATE TABLE Vendedor (
8      ID_Vendedor      INT          PRIMARY KEY AUTO_INCREMENT,
9      Nombre           VARCHAR(50)  NOT NULL,
10     Apellido          VARCHAR(50)  NOT NULL,
11     DNI               VARCHAR(10)  UNIQUE NOT NULL,
12     Fecha_Contratacion DATE        NOT NULL,
13     Telefono          VARCHAR(15),
14     Email             VARCHAR(100) UNIQUE
15 );
16
17 -- 2. Tabla CLIENTE
18 CREATE TABLE Cliente (
19     ID_Cliente        INT          PRIMARY KEY AUTO_INCREMENT,
20     Nombre            VARCHAR(50)  NOT NULL,
21     Apellido          VARCHAR(50)  NOT NULL,
22     Direccion         VARCHAR(255),
23     Telefono          VARCHAR(15),
24     Email             VARCHAR(100) UNIQUE
25 );
26
27 -- 3. Tabla CALZADO
28 CREATE TABLE Calzado (
29     ID_Calzado        VARCHAR(10)  PRIMARY KEY,

```



```

38 -- 4. Tabla VENTA
39 CREATE TABLE Venta (
40     ID_Venta          INT          PRIMARY KEY AUTO_INCREMENT,
41     Fecha_Venta       DATETIME    NOT NULL DEFAULT CURRENT_TIMESTAMP,
42     Total_Venta       DECIMAL(10, 2) NOT NULL CHECK (Total_Venta >= 0),
43     ID_Vendedor       INT          NOT NULL,
44     ID_Cliente        INT,
45
46     CONSTRAINT fk_venta_vendedor FOREIGN KEY (ID_Vendedor) REFERENCES Vendedor(ID_Vendedor)
47         ON DELETE RESTRICT ON UPDATE CASCADE,
48     CONSTRAINT fk_venta_cliente FOREIGN KEY (ID_Cliente) REFERENCES Cliente(ID_Cliente)
49         ON DELETE SET NULL ON UPDATE CASCADE
50 );
51
52 -- 5. Tabla DETALLE_VENTA
53 CREATE TABLE Detalle_Venta (
54     ID_Detalle        INT          PRIMARY KEY AUTO_INCREMENT,
55     ID_Venta          INT          NOT NULL,
56     ID_Calzado        VARCHAR(10)  NOT NULL,
57     Cantidad          INT          NOT NULL CHECK (Cantidad > 0),
58     Precio_Linea      DECIMAL(10, 2) NOT NULL CHECK (Precio_Linea >= 0),
59
60     CONSTRAINT fk_detalle_venta FOREIGN KEY (ID_Venta) REFERENCES Venta(ID_Venta)
61         ON DELETE CASCADE ON UPDATE CASCADE,
62     CONSTRAINT fk_detalle_calzado FOREIGN KEY (ID_Calzado) REFERENCES Calzado(ID_Calzado)
63         ON DELETE RESTRICT ON UPDATE CASCADE,
64
65     UNIQUE (ID_Venta, ID_Calzado)
66 );

```

```
CREATE DATABASE TiendaCalzados;

-- Usar la base de datos
USE TiendaCalzados;

-- 1. Tabla VENDEDOR
CREATE TABLE Vendedor (
    ID_Vendedor          INT          PRIMARY KEY AUTO_INCREMENT,
    Nombre                VARCHAR(50)  NOT NULL,
    Apellido              VARCHAR(50)  NOT NULL,
    DNI                   VARCHAR(10)  UNIQUE NOT NULL,
    Fecha_Contratacion    DATE          NOT NULL,
    Telefono              VARCHAR(15),
    Email                 VARCHAR(100)  UNIQUE
);

-- 2. Tabla CLIENTE
CREATE TABLE Cliente (
    ID_Cliente           INT          PRIMARY KEY AUTO_INCREMENT,
    Nombre                VARCHAR(50)  NOT NULL,
    Apellido              VARCHAR(50)  NOT NULL,
    Direccion             VARCHAR(255),
    Telefono              VARCHAR(15),
    Email                 VARCHAR(100)  UNIQUE
);

-- 3. Tabla CALZADO
CREATE TABLE Calzado (
    ID_Calzado           VARCHAR(10)  PRIMARY KEY,
    Nombre                VARCHAR(100) NOT NULL,
    Marca                 VARCHAR(50)  NOT NULL,
    Talla                 DECIMAL(4,1) NOT NULL CHECK (Talla > 0),
    Color                 VARCHAR(30)  NOT NULL,
    Precio_Unitario       DECIMAL(10, 2) NOT NULL CHECK (Precio_Unitario >=
0),
    Stock                 INT          NOT NULL CHECK (Stock >= 0)
);

-- 4. Tabla VENTA
CREATE TABLE Venta (
    ID_Venta              INT          PRIMARY KEY AUTO_INCREMENT,
```



```
Fecha_Venta      DATETIME      NOT NULL DEFAULT CURRENT_TIMESTAMP,
Total_Venta      DECIMAL(10, 2) NOT NULL CHECK (Total_Venta >= 0),
ID_Vendedor      INT           NOT NULL,
ID_Cliente       INT,

        CONSTRAINT fk_venta_vendedor FOREIGN KEY (ID_Vendedor) REFERENCES
Vendedor(ID_Vendedor)
        ON DELETE RESTRICT ON UPDATE CASCADE,
        CONSTRAINT fk_venta_cliente FOREIGN KEY (ID_Cliente) REFERENCES
Cliente(ID_Cliente)
        ON DELETE SET NULL ON UPDATE CASCADE
);

-- 5. Tabla DETALLE_VENTA
CREATE TABLE Detalle_Venta (
    ID_Detalle      INT           PRIMARY KEY AUTO_INCREMENT,
    ID_Venta        INT           NOT NULL,
    ID_Calzado      VARCHAR(10)   NOT NULL,
    Cantidad        INT           NOT NULL CHECK (Cantidad > 0),
    Precio_Linea    DECIMAL(10, 2) NOT NULL CHECK (Precio_Linea >= 0),

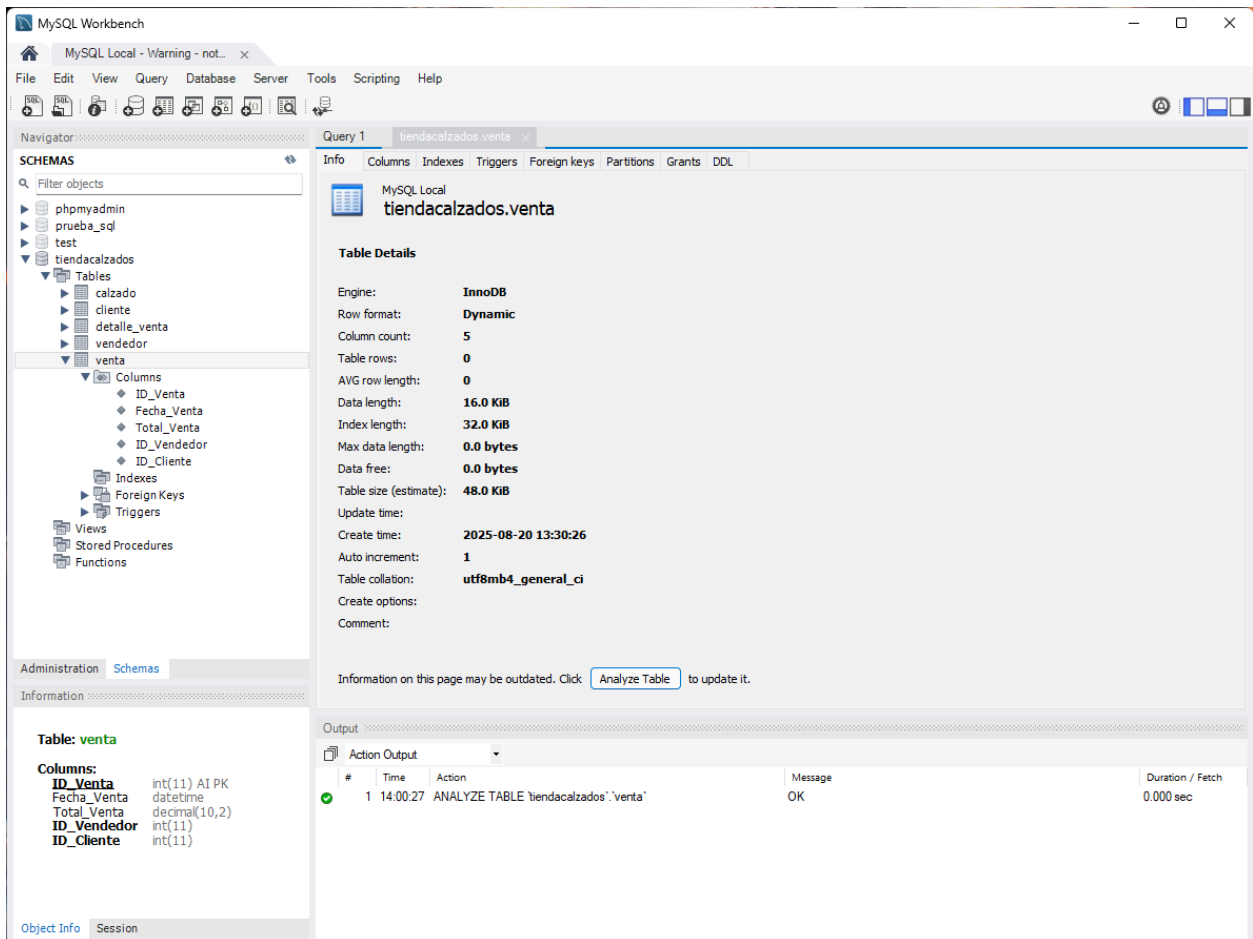
        CONSTRAINT fk_detalle_venta FOREIGN KEY (ID_Venta) REFERENCES
Venta(ID_Venta)
        ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT fk_detalle_calzado FOREIGN KEY (ID_Calzado) REFERENCES
Calzado(ID_Calzado)
        ON DELETE RESTRICT ON UPDATE CASCADE,

    UNIQUE (ID_Venta, ID_Calzado)
);
```

Paso 2: Ejecución del Código SQL

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓ 1	13:30:26	CREATE DATABASE TiendaCalzados	1 row(s) affected	0.000 sec
✓ 2	13:30:26	USE TiendaCalzados	0 row(s) affected	0.000 sec
✓ 3	13:30:26	CREATE TABLE Vendedor (ID_Vendedor INT PRIMARY KEY AUTO_INCREMENT, N...	0 row(s) affected	0.016 sec
✓ 4	13:30:26	CREATE TABLE Cliente (ID_Cliente INT PRIMARY KEY AUTO_INCREMENT, Nombr...	0 row(s) affected	0.015 sec
✓ 5	13:30:26	CREATE TABLE Calzado (ID_Calzado VARCHAR(10) PRIMARY KEY, Nombre VA...	0 row(s) affected	0.000 sec
✓ 6	13:30:26	CREATE TABLE Venta (ID_Venta INT PRIMARY KEY AUTO_INCREMENT, Fecha_...	0 row(s) affected	0.016 sec
✓ 7	13:30:26	CREATE TABLE Detalle_Venta (ID_Detalle INT PRIMARY KEY AUTO_INCREMENT, ...	0 row(s) affected	0.078 sec

Paso 3: Visualización de Tablas y Estructura



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'tiendacalzados' database selected, with its tables listed: calzado, cliente, detalle_venta, vendedor, and venta. The 'venta' table is selected, and its columns are listed: ID_Venta, Fecha_Venta, Total_Venta, ID_Vendedor, and ID_Cliente. The 'Table Details' pane on the right shows the table's properties: Engine: InnoDB, Row format: Dynamic, Column count: 5, Table rows: 0, AVG row length: 0, Data length: 16.0 KiB, Index length: 32.0 KiB, Max data length: 0.0 bytes, Data free: 0.0 bytes, Table size (estimate): 48.0 KiB, Create time: 2025-08-20 13:30:26, Auto increment: 1, Table collation: utf8mb4_general_ci, and Create options. The 'Output' pane at the bottom shows the result of the 'ANALYZE TABLE' command, indicating that the table structure has been updated.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' navigator shows the database structure, with the 'venta' table selected under the 'tiendacalzados' database. The main window shows the 'Columns' tab for the 'tiendacalzados.venta' table, listing five columns: ID_Venta (int(11)), Fecha_Venta (datetime), Total_Venta (decimal(10,2)), ID_Vendedor (int(11)), and ID_Cliente (int(11)). Below the table structure, the 'Output' tab shows the results of an 'ANALYZE TABLE' operation, indicating success with a message 'OK' and a duration of 0.000 seconds.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
ID_Venta	int(11)		NO			select,insert,update,references	auto_i
Fecha_Venta	datetime	current_timestamp()	NO			select,insert,update,references	
Total_Venta	decimal(10,2)		NO			select,insert,update,references	
ID_Vendedor	int(11)		NO			select,insert,update,references	
ID_Cliente	int(11)	NULL	YES			select,insert,update,references	

Count: 5

Table: venta

Columns:

- ID_Venta int(11) AI PK
- Fecha_Venta datetime
- Total_Venta decimal(10,2)
- ID_Vendedor int(11)
- ID_Cliente int(11)

Output

#	Time	Action	Message	Duration / Fetch
1	14:00:27	ANALYZE TABLE 'tiendacalzados'.venta	OK	0.000 sec

The screenshot shows the MySQL Workbench interface with the 'tiendacalzados.venta' table selected. The 'Indexes in Table' tab is active, displaying the following data:

Key	Type	Uni...	Columns
PRIMARY	BTREE	YES	ID_Venta
fk_venta_vendedor	BTREE	NO	ID_Vendedor
fk_venta_cliente	BTREE	NO	ID_Cliente

The 'Columns in table' tab is also visible, showing the following data:

Column	Type	Nullable	Indexes
ID_Venta	int(11)	NO	PRIMARY
Fecha_Venta	datetime	NO	
Total_Venta	decimal(10,2)	NO	
ID_Vendedor	int(11)	NO	fk_venta_vendedor
ID_Cliente	int(11)	YES	fk_venta_cliente

The 'Output' tab at the bottom shows the results of an 'ANALYZE TABLE' command:

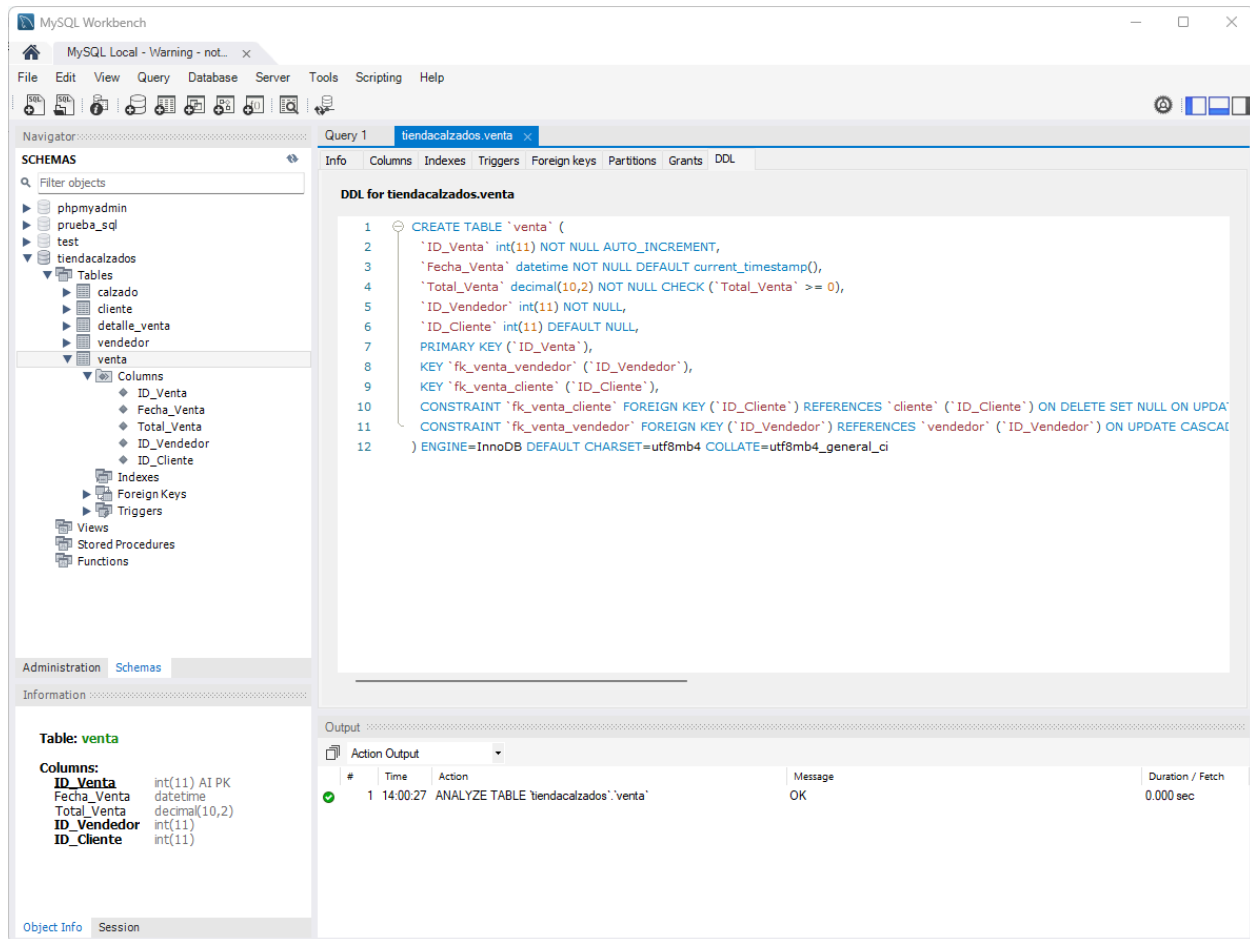
#	Time	Action	Message	Duration / Fetch
1	14:00:27	ANALYZE TABLE 'tiendacalzados'.venta	OK	0.000 sec

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'tiendacalzados' database expanded, showing tables like 'calzado', 'cliente', 'detalle_venta', 'vendedor', and 'venta'. The 'venta' table is selected, and its columns are listed: ID_Venta (int(11) AI PK), Fecha_Venta (datetime), Total_Venta (decimal(10,2)), ID_Vendedor (int(11)), and ID_Cliente (int(11)).

The main window shows the 'Foreign keys' tab for the 'tiendacalzados.venta' table. It displays three foreign key relationships:

Name	Schema	Table	Column	Referenced Sch...	Referenced Table	Referenced Col...
fk_detalle_venta	tiendacalzados	detalle_venta	ID_Venta	tiendacalzados	venta	ID_Venta
fk_venta_cliente	tiendacalzados	venta	ID_Cliente	tiendacalzados	cliente	ID_Cliente
fk_venta_vendedor	tiendacalzados	venta	ID_Vendedor	tiendacalzados	vendedor	ID_Vendedor

The 'Output' tab at the bottom shows the result of an 'ANALYZE TABLE' command on the 'tiendacalzados.venta' table, indicating it was successful.



5. Conclusión

El desarrollo de este trabajo práctico ha permitido aplicar y consolidar los fundamentos del Modelo Relacional en un escenario práctico de gestión para una tienda de calzados. Se ha logrado diseñar un sistema de base de datos que aborda las necesidades iniciales de la tienda, permitiendo la gestión eficiente de vendedores, clientes, calzados y, fundamentalmente, las transacciones de venta y sus detalles.

La fase de **diseño conceptual** mediante el Diagrama Entidad-Relación (E-R) fue crucial para visualizar las entidades clave, sus atributos y las complejas relaciones existentes, como la cardinalidad muchos a muchos entre Venta y Calzado, que se resolvió exitosamente a través de la entidad Detalle_Venta. Este enfoque conceptual, independiente de la implementación física, aseguró una comprensión clara del dominio del problema.

Posteriormente, la **traducción al modelo lógico/físico** mediante sentencias SQL demostró la robustez y

flexibilidad del Modelo Relacional. Se han definido tablas, seleccionado tipos de datos apropiados y, lo más importante, se han implementado claves primarias y foráneas junto con restricciones de integridad. Esto garantiza la unicidad de los registros, la coherencia de las relaciones entre tablas (integridad referencial) y la validez de los datos almacenados, previniendo inconsistencias y anomalías.

En síntesis, este ejercicio práctico no solo ha resultado en un diseño de base de datos funcional para el caso de estudio, sino que también ha reforzado la comprensión de los principios esenciales del Modelo Relacional, su aplicabilidad en escenarios reales y la importancia de un diseño estructurado para garantizar la integridad y eficiencia de la información.