

UNIVERSIDAD AUTONOMA DEL ESTADO DE HIDALGO

Diseño E Implementación de Base de Datos para Distribuidora de Herramientas

Asignatura:

Base de Datos Distribuidas

Profesor:

Cornejo Velázquez Eduardo

Alumno:

Lozada García Ana Laura

1. Introducción

El diseño e implementación de una base de datos para una empresa distribuidora de herramientas es una tarea fundamental para mejorar la eficiencia y la gestión de los datos. Una base de datos bien estructurada permite organizar información relacionada con productos, proveedores, clientes y transacciones comerciales, garantizando la accesibilidad, integridad y seguridad de los datos. Además, una base de datos relacional facilita la automatización de procesos operativos clave, como el control de inventarios y la emisión de reportes. Este proyecto tiene como objetivo crear una base de datos optimizada que permita a la empresa gestionar sus operaciones diarias de forma ágil, escalable y confiable.

2. Marco Teórico

Una base de datos relacional es un sistema que organiza los datos en tablas interrelacionadas a través de claves primarias y foráneas. Este enfoque permite mantener la integridad de los datos y realizar consultas complejas mediante SQL. La base de datos relacional es particularmente útil para empresas que manejan grandes volúmenes de información y requieren garantizar la coherencia y exactitud de los datos.

La normalización es un proceso de diseño de bases de datos que tiene como objetivo eliminar la redundancia de datos y garantizar que las dependencias entre los atributos estén correctamente definidas. A través de varias formas normales (1NF, 2NF, 3NF), se asegura que la estructura de la base de datos minimice inconsistencias, redundancias y pérdidas de información.

El modelo Entidad-Relación es una herramienta de modelado que se utiliza para representar gráficamente las entidades involucradas en un sistema y las relaciones entre ellas. En el caso de una distribuidora de herramientas, las entidades más comunes incluyen productos, proveedores, clientes y pedidos. El diagrama ER facilita el diseño conceptual de la base de datos al mostrar cómo se relacionan estas entidades y qué atributos las componen.

3. Herramientas Empleadas

Para el desarrollo de esta práctica se utilizaron las siguientes herramientas:

- **MySQL:** Sistema de gestión de bases de datos relacional utilizado para la implementación de la base de datos.
- **MySQL Workbench:** Herramienta visual de diseño de bases de datos que facilita la creación del modelo E-R y la generación de código SQL.

4. Desarrollo de la Práctica

El primer paso en la implementación de la base de datos es entender las necesidades específicas de la empresa. En este caso, se requiere una base de datos que gestione:

- **Productos:** Nombre, código, precio, categoría, cantidad en stock.
- **Proveedores:** Nombre, contacto, productos que suministra.
- **Clientes:** Nombre, dirección, historial de compras.
- **Pedidos:** Detalles de las transacciones, productos vendidos, cantidad, fecha y cliente.

Se construye un diagrama Entidad-Relación que refleja las entidades y sus atributos. Las principales entidades identificadas son:

- **Productos** (con atributos como código de producto, nombre, precio, stock).
- **Proveedores** (nombre, contacto, dirección).
- **Clientes** (nombre, dirección, teléfono).
- **Pedidos** (ID de pedido, fecha, cliente, producto, cantidad).

Las relaciones entre estas entidades son:

- Un proveedor suministra muchos productos (relación uno a muchos).
- Un cliente realiza muchos pedidos (relación uno a muchos).
- Un pedido puede contener varios productos (relación muchos a muchos).

5. Implementación en MySQL

Se crearon las tablas en MySQL utilizando el siguiente código SQL:

```
1 CREATE DATABASE distribuidor_herramientas;
2 USE distribuidor_herramientas;
3
4 CREATE TABLE Proveedor (
5     id_proveedor INT PRIMARY KEY AUTO_INCREMENT,
6     nombre VARCHAR(100) NOT NULL,
7     contacto VARCHAR(100)
8 );
9
10 CREATE TABLE Compra (
11     id_compra INT PRIMARY KEY AUTO_INCREMENT,
12     id_proveedor INT,
13     fecha DATE NOT NULL,
14     total DECIMAL(10, 2) NOT NULL,
15     FOREIGN KEY (id_proveedor) REFERENCES Proveedor(
16         id_proveedor)
17 );
18
19 CREATE TABLE Herramienta (
20     id_herramienta INT PRIMARY KEY AUTO_INCREMENT,
21     nombre VARCHAR(100) NOT NULL,
22     precio_compra DECIMAL(10, 2) NOT NULL,
23     precio_venta DECIMAL(10, 2) NOT NULL
24 );
25
26 CREATE TABLE Inventario (
27     id_herramienta INT,
28     cantidad INT NOT NULL,
29     FOREIGN KEY (id_herramienta) REFERENCES Herramienta(
30         id_herramienta)
31 );
32
33 CREATE TABLE Ruta (
34     id_ruta INT PRIMARY KEY AUTO_INCREMENT,
35     nombre VARCHAR(100) NOT NULL
36 );
37
38 CREATE TABLE Cliente (
39     id_cliente INT PRIMARY KEY AUTO_INCREMENT,
40     nombre VARCHAR(100) NOT NULL,
41     tipo ENUM('consumidor', 'minorista') NOT NULL,
42     id_ruta INT,
43     FOREIGN KEY (id_ruta) REFERENCES Ruta(id_ruta)
44 );
```

```
44 CREATE TABLE Venta (
45     id_venta INT PRIMARY KEY AUTO_INCREMENT,
46     id_cliente INT,
47     fecha DATE NOT NULL,
48     total DECIMAL(10, 2) NOT NULL,
49     FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
50 );
51
52 CREATE TABLE Detalle_Venta (
53     id_venta INT,
54     id_herramienta INT,
55     cantidad INT NOT NULL,
56     precio_unitario DECIMAL(10, 2) NOT NULL,
57     FOREIGN KEY (id_venta) REFERENCES Venta(id_venta),
58     FOREIGN KEY (id_herramienta) REFERENCES Herramienta(
59         id_herramienta)
60 );
61
62 CREATE TABLE Pedido (
63     id_pedido INT PRIMARY KEY AUTO_INCREMENT,
64     id_cliente INT,
65     fecha DATE NOT NULL,
66     estado ENUM('pendiente', 'pagado', 'entregado') NOT NULL,
67     FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
68 );
69
70 CREATE TABLE Detalle_Pedido (
71     id_pedido INT,
72     id_herramienta INT,
73     cantidad INT NOT NULL,
74     FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),
75     FOREIGN KEY (id_herramienta) REFERENCES Herramienta(
76         id_herramienta)
77 );
78
79 CREATE TABLE Empleado (
80     id_empleado INT PRIMARY KEY AUTO_INCREMENT,
81     nombre VARCHAR(100) NOT NULL
82 );
83
84 CREATE TABLE Entrega (
85     id_entrega INT PRIMARY KEY AUTO_INCREMENT,
86     id_pedido INT,
87     id_empleado INT,
88     id_ruta INT,
89     fecha DATE NOT NULL,
90     FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),
91     FOREIGN KEY (id_empleado) REFERENCES Empleado(id_empleado)
92 ),
```

```
90 FOREIGN KEY (id_ruta) REFERENCES Ruta(id_ruta)
91 );
```

Listing 1: Creación de tablas en MySQL

6. Población de la Base de Datos

Se insertaron datos de ejemplo en las tablas para probar su funcionamiento. A continuación, se muestran algunas sentencias SQL de ejemplo para la población de datos y consultas:

6.1. Inserción de Datos

```
1 INSERT INTO Proveedor (nombre, contacto) VALUES
2 ('Herramientas SA', 'contacto@herramientassa.com'),
3 ('Ferretería Industrial', 'ventas@ferreteriaindustrial.com')
4 ;
5 INSERT INTO Herramienta (nombre, precio_compra, precio_venta)
6 VALUES
7 ('Martillo', 10.00, 15.00),
8 ('Destornillador', 5.00, 8.00),
9 ('Sierra', 20.00, 30.00);
10 INSERT INTO Cliente (nombre, tipo, id_ruta) VALUES
11 ('Juan Perez', 'consumidor', NULL),
12 ('Ferretería Local', 'minorista', 1);
```

Listing 2: Inserción de datos de ejemplo

6.2. Consultas de Ejemplo

6.2.1. Inventario Actual

Esta consulta muestra el inventario actual con el número de piezas y el costo total basado en el último precio de compra:

```
1 SELECT h.nombre, i.cantidad,
2        h.precio_compra,
3        (i.cantidad * h.precio_compra) AS costo_total
4 FROM Herramienta h
5 JOIN Inventario i ON h.id_herramienta = i.id_herramienta;
```

Listing 3: Consulta de inventario actual

6.2.2. Ventas Mensuales en Tienda

Esta consulta genera un reporte de ventas mensuales en la tienda del distribuidor:


```
1 SELECT YEAR(fecha) AS a o , MONTH(fecha) AS mes ,  
2       SUM(total) AS ventas_totales  
3 FROM Venta  
4 GROUP BY YEAR(fecha), MONTH(fecha)  
5 ORDER BY a o , mes;
```

Listing 4: Consulta de ventas mensuales

6.2.3. Ranking de Clientes Minoristas

Esta consulta muestra un ranking de clientes minoristas por número de herramientas vendidas y monto total de compras:

```
1 SELECT c.nombre ,  
2       SUM(dv.cantidad) AS total_herramientas ,  
3       SUM(v.total) AS monto_total  
4 FROM Cliente c  
5 JOIN Venta v ON c.id_cliente = v.id_cliente  
6 JOIN Detalle_Venta dv ON v.id_venta = dv.id_venta  
7 WHERE c.tipo = 'minorista'  
8 GROUP BY c.id_cliente  
9 ORDER BY monto_total DESC;
```

Listing 5: Ranking de clientes minoristas

Estas consultas demuestran cómo la base de datos puede proporcionar información valiosa para la gestión del negocio, desde el control de inventario hasta el análisis de ventas y clientes.

7. Conclusiones

El diseño e implementación de una base de datos para una empresa distribuidora de herramientas ha resultado en un sistema eficaz que permite gestionar de manera eficiente el inventario, las relaciones con proveedores y clientes, así como los pedidos y ventas. La normalización del diseño asegura la integridad de los datos y reduce la redundancia, mientras que la implementación en un DBMS proporciona un sistema escalable y fácil de mantener.

La automatización de tareas clave, como el control de inventarios y la generación de reportes de ventas, permite optimizar los procesos internos de la empresa, mejorando la toma de decisiones y la eficiencia operativa. Con este sistema, la empresa puede adaptarse mejor a las demandas del mercado y continuar escalando sus operaciones sin comprometer la calidad y consistencia de los datos.

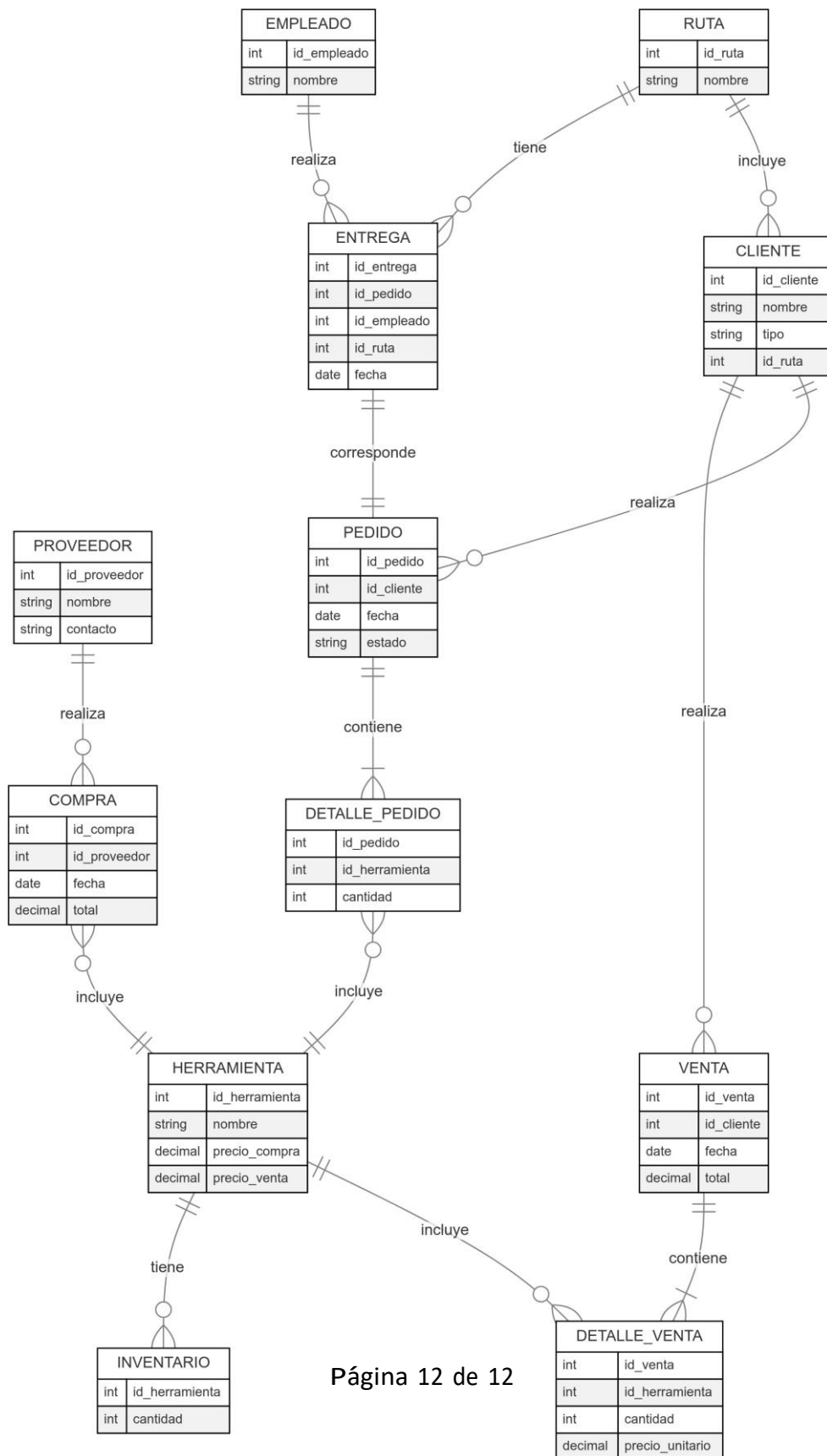


Figura 1: Modelo Entidad-Relación de la Distribuidora de Herramientas