

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:

Amuchategui Flores Braulio

1. Introducción

El diseño e implementación de una base de datos es un proceso esencial para el buen funcionamiento y la gestión eficiente de la información en una empresa distribuidora de herramientas. En este tipo de negocio, donde se manejan grandes volúmenes de productos, proveedores, clientes y transacciones, la correcta administración de los datos es fundamental para garantizar la operatividad y la toma de decisiones informadas.

El proceso de diseño de una base de datos comienza con la identificación de los requerimientos de la empresa, tales como el control de inventarios, el seguimiento de ventas, la gestión de proveedores y el control financiero. A partir de este análisis, se procede a la creación de un modelo entidad-relación (ER) que represente las principales entidades involucradas (productos, clientes, proveedores, órdenes de compra, entre otras) y sus relaciones. Esta fase es crucial para garantizar que la estructura de la base de datos refleje adecuadamente las operaciones del negocio.

Una vez que el diseño conceptual está definido, se procede a su implementación en un sistema de gestión de bases de datos (DBMS) MySQL. Durante esta etapa, se lleva a cabo la creación de tablas, la definición de relaciones y la implementación de restricciones, índices y claves foráneas, con el fin de asegurar la integridad de los datos y optimizar su consulta.

Además, la base de datos debe ser diseñada pensando en su escalabilidad y capacidad de adaptación a las necesidades cambiantes del negocio. En el caso de una empresa distribuidora de herramientas, es importante prever el crecimiento del inventario, la ampliación de la cartera de clientes y proveedores, y la evolución de las dinámicas de venta y distribución.

Finalmente, se realizan pruebas exhaustivas para validar el correcto funcionamiento de la base de datos y se asegura que cumple con los requisitos de la empresa. También es clave implementar políticas de respaldo y recuperación de datos para proteger la información crítica del negocio.

Este proceso de diseño e implementación no solo busca optimizar las operaciones actuales de la empresa, sino también prepararla para el futuro, facilitando una gestión de datos eficiente y segura que soporte el crecimiento del negocio.

2. Marco Teórico

El marco teórico proporciona los fundamentos conceptuales y metodológicos necesarios para entender el diseño y la implementación de una base de datos en el contexto de una empresa distribuidora de herramientas. Este marco se apoya en varias disciplinas, incluyendo bases de datos, administración de sistemas de información y gestión empresarial, con el objetivo de establecer los principios que guían la creación de un sistema de gestión de datos eficiente y escalable.

2.1. Modelo Entidad-Relación (E-R)

El modelo E-R es una herramienta de modelado de datos que permite representar las entidades relevantes de un sistema de información, sus interrelaciones y propiedades. Este modelo es crucial para visualizar la estructura de la base de datos antes de su implementación.

2.2. Modelo Relacional

El modelo relacional organiza los datos en tablas (relaciones) con filas (tuplas) y columnas (atributos). Este modelo permite establecer relaciones entre las diferentes tablas mediante el uso de claves primarias y foráneas.

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 desarrollo de la práctica se llevó a cabo en varias etapas:

1. **Análisis de Requisitos:** Definir las necesidades de la empresa (productos, clientes, proveedores, etc.).
2. **Modelado de Datos:** Crear el diagrama ER que describa las entidades y sus relaciones.
3. **Normalización:** Aplicar las formas normales para evitar redundancias y asegurar integridad.
4. **Diseño Físico:** Convertir el modelo ER en tablas y establecer los tipos de datos.
5. **Implementación:** Crear las tablas y relaciones en el DBMS.
6. **Pruebas:** Validar que la base de datos funcione correctamente mediante pruebas SQL.
7. **Optimización:** Mejorar el rendimiento y garantizar la seguridad de la información.

Este proceso garantiza que la base de datos para la empresa distribuidora de herramientas sea funcional, segura y escalable, permitiendo una gestión eficiente de inventarios, ventas, proveedores y clientes.

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

Mejora en la gestión de la información: La implementación de una base de datos relacional para una empresa distribuidora de herramientas permite una organización y control efectivo de la información clave del negocio, como inventarios, proveedores, clientes y ventas. Al centralizar y estructurar adecuadamente los datos, se facilita el acceso rápido a la información crítica para la toma de decisiones.

Reducción de la redundancia y mejora en la integridad de los datos: El proceso de normalización aplicado durante el diseño de la base de datos asegura que los datos estén libres de redundancias y que las actualizaciones se realicen de manera coherente en toda la base. Esto contribuye a evitar inconsistencias y duplicaciones, mejorando la calidad de la información almacenada.

Escalabilidad y flexibilidad del sistema: Una base de datos bien diseñada, normalizada y optimizada permite a la empresa crecer de manera escalable. A medida que el volumen de datos aumenta, o que se requieren nuevas funcionalidades, el sistema puede adaptarse fácilmente sin comprometer su rendimiento o fiabilidad.

Facilitación de consultas y generación de reportes: El uso de consultas SQL sobre una base de datos relacional facilita la extracción de información valiosa para el negocio. Esto incluye informes sobre los productos más vendidos, el control de existencias, el análisis del rendimiento de los proveedores y las tendencias de compra de los clientes, lo que apoya una toma de decisiones informada y basada en datos.

Mayor seguridad y control de acceso a los datos: La implementación de políticas de seguridad y control de acceso asegura que solo los usuarios autorizados pueden realizar modificaciones, consultas o eliminaciones en la base de datos. Esto es esencial para proteger la información sensible del negocio y garantizar que se mantenga la integridad de los datos a lo largo del tiempo.

Optimización de los procesos operativos: Al automatizar tareas como la gestión del inventario, el registro de ventas y la actualización de los datos de los proveedores, la base de datos permite a la empresa reducir tiempos en la ejecución de procesos operativos y minimiza errores humanos, mejorando la eficiencia general del negocio.

Sostenibilidad y mantenimiento a largo plazo: Un sistema de bases de datos bien diseñado no solo cumple con las necesidades actuales de la empresa, sino que también está preparado para ser mantenido y actualizado a lo largo del tiempo. Esto incluye la capacidad de realizar respaldos periódicos, actualizaciones de software, y ajustes en las consultas e índices para mantener un rendimiento óptimo.

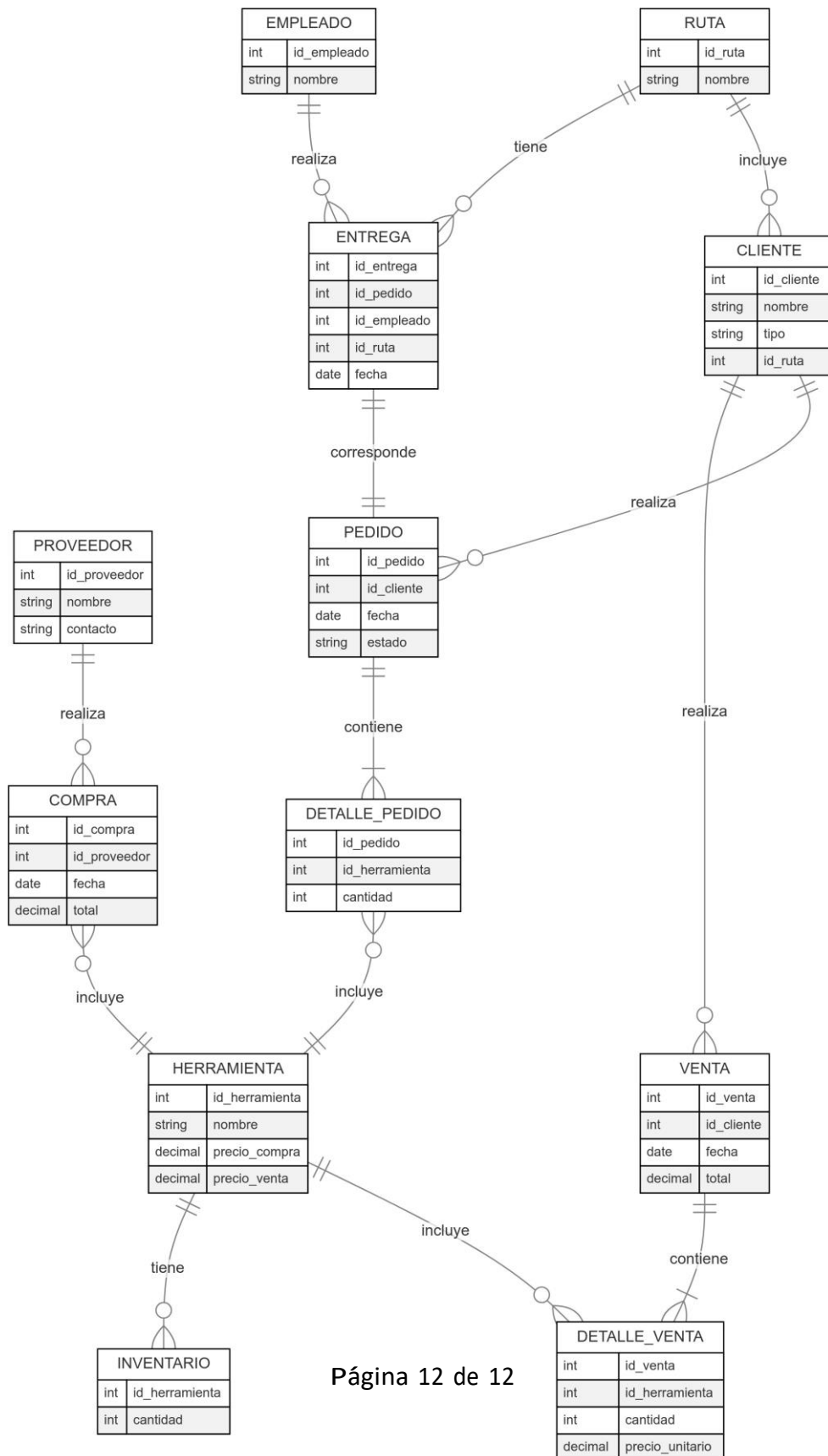


Figura 1: Modelo Entidad-Relación de la Distribuidora de Herramientas