



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

MATERIA:
BASE DE DATOS DISTRIBUIDAS

PROFESOR:
CORNEJO VELAZQUEZ EDUARDO

“DISTRIBUIDORA DATABASE”

ALUMNO:
AMUCHATEGUI FLORES BRAULIO

1. INTRODUCCIÓN

En el ámbito de la distribución de herramientas, la organización y el control de la información son elementos fundamentales para lograr una gestión eficiente y competitiva. Las empresas distribuidoras se enfrentan a retos importantes, como el manejo de inventarios amplios y variados, la administración de relaciones con múltiples proveedores y clientes, así como la optimización de los procesos de compra y venta. Para hacer frente a estos desafíos, una base de datos relacional en MySQL representa una solución efectiva para almacenar y procesar grandes volúmenes de datos de manera estructurada y segura.

El presente proyecto tiene como propósito el diseño e implementación de una base de datos en MySQL que permita la gestión integral de una distribuidora de herramientas, abarcando desde el control de existencias y precios hasta la coordinación de pedidos y facturación. La base de datos se ha diseñado para proporcionar acceso rápido y seguro a la información, permitiendo a la empresa realizar un seguimiento en tiempo real de los productos en inventario, identificar patrones de venta y mantener actualizada la información de proveedores y clientes. Además, la estructura de esta base de datos está orientada a minimizar errores y mejorar la eficiencia operativa mediante la normalización de los datos, evitando redundancias y facilitando su mantenimiento a largo plazo.

El sistema de base de datos se compone de diversas tablas y relaciones que reflejan los diferentes componentes de la operación de la distribuidora, incluyendo información detallada sobre los productos, como nombre, código, categoría y precio, así como datos de proveedores, clientes y transacciones. También se han implementado restricciones y reglas de integridad para garantizar la precisión y confiabilidad de los datos, permitiendo a la empresa tomar decisiones informadas y estratégicas basadas en un análisis completo y en tiempo real de su operación.

2. MARCO TEÓRICO

La administración de bases de datos es esencial en el desarrollo de sistemas de información modernos, especialmente en sectores donde la precisión y el control de datos resultan cruciales para las operaciones diarias, como en la distribución de herramientas. Una base de datos estructurada permite organizar la información de manera que facilite el acceso y la toma de decisiones estratégicas. Las bases de datos relacionales, como MySQL, utilizan el modelo entidad-relación (E-R), que representa datos en tablas donde cada una describe entidades específicas como proveedores, productos o almacenes, conectadas entre sí para permitir la organización y recuperación eficiente de los datos.

En el caso de una distribuidora de herramientas, una base de datos en MySQL debe contemplar diversas entidades para capturar los componentes esenciales del negocio. Por ejemplo, se necesita una tabla de proveedores o "supplier" para almacenar información de los socios comerciales de la empresa, quienes suministran las herramientas; una tabla de compras o "purchase" que registre las transacciones realizadas con cada proveedor, vinculando esta información mediante claves foráneas. La tabla de herramientas o "tool" permite almacenar datos detallados de cada producto como modelo, tamaño o fabricante, lo cual facilita su identificación y control en el inventario. Por su parte, la tabla de almacenes o "warehouse" centraliza la información de las ubicaciones físicas donde se almacenan las herramientas, mientras que la tabla de inventarios o "inventory" detalla la ubicación específica de cada producto en el almacén y la cantidad disponible, además de los precios asociados para el cálculo de márgenes y costos. Finalmente, una tabla de detalle de compra o "buyTool" relaciona compras con inventarios y especifica las cantidades y precios unitarios, facilitando así el control financiero y la trazabilidad de cada producto.

La normalización de esta base de datos asegura la eliminación de redundancias y el cumplimiento de la integridad referencial mediante claves primarias y foráneas, reduciendo así los errores y mejorando la eficiencia. La administración del inventario se simplifica al permitir una visualización precisa de las entradas y salidas de cada herramienta, lo que ayuda a optimizar la rotación y reducir costos. MySQL, como sistema de gestión de bases de datos relacional, resulta una opción adecuada por su rapidez, flexibilidad y capacidad de manejo de grandes volúmenes de información, además de ser una herramienta accesible y escalable ideal para empresas medianas.

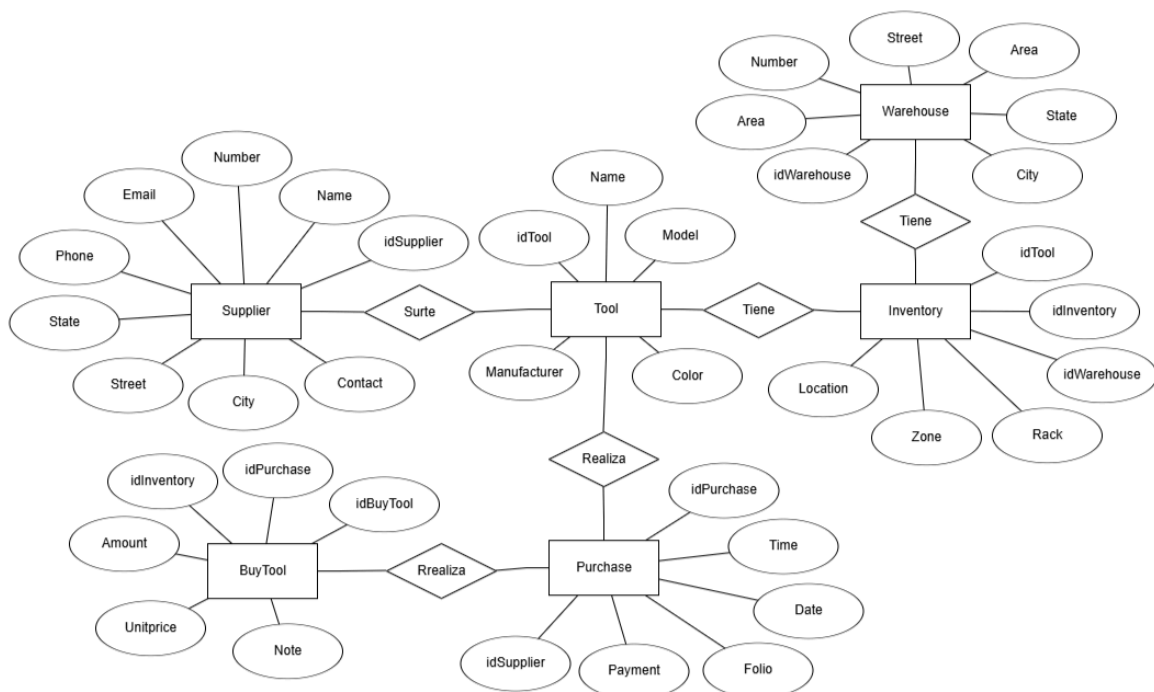
3. HERRAMIENTAS EMPLEADAS

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

1. ERD Plus: Una herramienta en línea para crear diagramas de Entidad-Relación. Se utiliza para el diseño conceptual de la base de datos.
2. MySQL Server: Un sistema de gestión de bases de datos relacional. Se utiliza para implementar y gestionar la base de datos diseñada.

4. DESARROLLO

- supplier(*idSupplier*, name, street, number, city, state, phone, email, contact)
- purchase(*idPurchase*, idSupplier, date, time, folio, payment)
- tool(*idTool*, name, model, manufacturer, color, size, description)
- warehouse(*idWarehouse*, street, number, city, state, area, phone, manager)
- inventory(*idInventory*, idWarehouse, idTool, zone, rack, level, location, amount, purchasePrice, storeSalePrice, wholeSalePrice)
- buyTool(*idBuyTool*, idPurchase, idInventory, amount, unitPrice, note)



```
CREATE DATABASE DISTRIBUIDORAA;  
USE DISTRIBUIDORAA;
```

```
CREATE TABLE supplier (  
    idSupplier INT PRIMARY KEY,  
    namee VARCHAR(50),  
    street VARCHAR(100),  
    numbeer VARCHAR(15),  
    city VARCHAR(50),  
    state VARCHAR(50),  
    phone VARCHAR(15),  
    email VARCHAR(25),  
    contact VARCHAR(100)  
);
```

```
CREATE TABLE purchase (  
    idPurchase INT PRIMARY KEY,  
    idSupplier INT,  
    datee DATE,  
    timee TIME,  
    folio VARCHAR(20),  
    payment VARCHAR(50),  
    FOREIGN KEY (idSupplier) REFERENCES supplier(idSupplier)  
);
```

```
CREATE TABLE tool (  
    idTool INT PRIMARY KEY,  
    namee VARCHAR(50),  
    model VARCHAR(50),  
    manufacturer VARCHAR(100),  
    color VARCHAR(30),  
    size VARCHAR(30),  
    description TEXT  
);
```

```
CREATE TABLE warehouse (  
    idWarehouse INT PRIMARY KEY,  
    street VARCHAR(100),  
    numbeer VARCHAR(15),  
    city VARCHAR(50),  
    state VARCHAR(50),  
    area VARCHAR (30),  
    phone VARCHAR(15),  
    manager VARCHAR(100)  
);
```

```

CREATE TABLE inventory (
  idInventory INT PRIMARY KEY,
  idWarehouse INT,
  idTool INT,
  zone VARCHAR(50),
  rack VARCHAR(50),
  level INT,
  location VARCHAR(100),
  amount INT,
  purchasePrice DECIMAL(10, 2),
  storeSalePrice DECIMAL(10, 2),
  wholeSalePrice DECIMAL(10, 2),
  FOREIGN KEY (idWarehouse) REFERENCES warehouse(idWarehouse),
  FOREIGN KEY (idTool) REFERENCES tool(idTool)
);

```

```

CREATE TABLE buyTool (
  idBuyTool INT AUTO_INCREMENT PRIMARY KEY,
  idPurchase INT,
  idInventory INT,
  amount INT,
  unitPrice DECIMAL(10, 2),
  note VARCHAR (10),
  FOREIGN KEY (idPurchase) REFERENCES purchase(idPurchase),
  FOREIGN KEY (idInventory) REFERENCES inventory(idInventory)
);

```

```

INSERT INTO supplier (namee, street, numbeer, city, state, phone, email, contact) VALUES
('Proveedor 1', 'Calle Flores', '126', 'Zempoala', 'Hidalgo', '7711284554',
'holaamigos@gmail.com', 'Braulio Amuchategui'),
('Proveedor 2', 'Calle Viaducto', '222', 'Actopan', 'Hidalgo', '7715642223',
'maria22@hotmail.com', 'Maria Flores'),
('Proveedor 3', 'Calle Colosio', '310', 'Atotonilco', 'Hidalgo', '7717891234',
'carloslopez@gmail.com', 'Carlos Lopez'),
('Proveedor 4', 'Calle Azteca', '811', 'Iztapalapa', 'Ciudad de Mexico', '7712148005',
'garcia18@hotmail.com', 'Ana Garcia'),
('Proveedor 5', 'Calle Mar', '762', 'Xochimilco', 'Ciudad de Mexico', '7778889999',
'goatmessi@gmail.com', 'Lionel Messi');

```

```

INSERT INTO tool VALUES (1, 'Martillo', 'Modelo A', 'Marca X', 'Rojo', 'Mediano', 'Martillo de
carpintero'),(2, 'Taladro', 'Modelo B', 'Marca Y', 'Negro', 'Grande', 'Taladro de uso
industrial'),(3, 'Sierra', 'Modelo C', 'Marca Z', 'Azul', 'Pequeño', 'Sierra manual'),(4,
'Destornillador', 'Modelo D', 'Marca W', 'Amarillo', 'Pequeño', 'Destornillador de precisión'),(5,
'Llave Inglesa', 'Modeol E', 'Marca V', 'Verde', 'Grande', 'Llave ajustable para uso mecánico');

```

INSERT INTO warehouse VALUES (1, 'Calle Almacen 1', '101', 'Ciudad Zempoala', 'Estado Hidalgo', 100.50, '6789012345', 'Pedro Ponce'),(2, 'Calle Almacen 2', '102', 'Ciudad Orizaba', 'Estado Veracruz', 200.75, '7890123456', 'Carmen Diaz'),(3, 'Calle Almacen 3', '103', 'Ciudad León', 'Estado Guanajuato', 150.60, '8901234567', 'Roberto Sosa'),(4, 'Calle Almacen 4', '104', 'Ciudad Monterrey', 'Estado Nuevo Leon', 80.00, '9012345678', 'Lorena Mora'),(5, 'Calle Almacen 5', '105', 'Ciudad Acapulco', 'Estado Guerrero', 120.45, '0123456789', 'Esteban Cuevas');

INSERT INTO purchase VALUES (1, 1, '2024-11-01', '08:30:00', 'FOL12345', 'Contado'),(2, 2, '2024-11-02', '09:00:00', 'FOL12346', 'Credito'),(3, 3, '2024-11-03', '10:15:00', 'FOL12347', 'Contado'),(4, 4, '2024-11-04', '11:45:00', 'FOL12348', 'Credito'),(5, 5, '2024-11-05', '12:00:00', 'FOL12349', 'Contado'),(6, 1, '2024-11-06', '13:00:00', 'FOL1

5. CONCEPTOS

Grado de la Relación

El grado de la relación se refiere al número de entidades que participan en una relación. En una base de datos relacional, una relación binaria conecta dos entidades, como `supplier` y `purchase`.

Clave Candidata, Clave Primaria y Superclave

- **Clave candidata:** Es un atributo o conjunto de atributos que pueden identificar de manera única cada tupla (registro) en una tabla. En la tabla `supplier`, por ejemplo, tanto `idSupplier` como el correo electrónico `email` pueden ser claves candidatas.
- **Clave primaria:** Es una clave candidata elegida para identificar de forma única cada registro en la tabla. En `supplier`, hemos elegido `idSupplier` como la clave primaria.
- **Superclave:** Es cualquier conjunto de atributos que puede identificar de manera única una tupla en una relación. Toda clave primaria es también una superclave, pero una superclave puede incluir atributos adicionales.

Clave Foránea

Una clave foránea es un atributo en una tabla que establece una relación con la clave primaria de otra tabla. En `purchase`, el atributo `idSupplier` es una clave foránea que referencia `idSupplier` en la tabla `supplier`, permitiendo asociar cada compra con un proveedor.

Cardinalidad

La cardinalidad describe el número de elementos en una relación entre entidades. En tu caso, la relación entre `supplier` y `purchase` es de uno a muchos, ya que un proveedor puede realizar muchas compras, pero cada compra solo tiene un proveedor.

La **fragmentación** es una técnica utilizada en bases de datos distribuidas para mejorar el rendimiento:

- **Fragmentación horizontal:** Se divide una tabla en subconjuntos basados en las filas. Por ejemplo, podrías fragmentar la tabla `purchase` para almacenar las compras recientes y las antiguas en distintos fragmentos.
- **Fragmentación vertical:** Divide la tabla en subconjuntos basados en las columnas. Por ejemplo, podrías separar la tabla `supplier` en dos fragmentos: uno con los datos de contacto y otro con la dirección y la ciudad.

Índice (Index)

Un **índice** mejora la velocidad de las consultas en la base de datos. Crear un índice en la columna name de la tabla tool permitiría buscar herramientas por nombre de manera más rápida.

Vista (View)

Una **vista** es una tabla virtual que contiene el resultado de una consulta. Crear una vista puede ser útil para simplificar consultas complejas o para dar acceso a datos específicos sin exponer la tabla completa.

6. CONCLUSIONES

El desarrollo de esta base de datos para una distribuidora de herramientas ha permitido implementar y analizar diversos conceptos clave de bases de datos relacionales, como el uso de claves primarias, claves candidatas y claves foráneas para asegurar la integridad de las relaciones entre tablas, así como la cardinalidad y el grado de relación. La estructura de la base de datos facilita el almacenamiento, consulta y administración de datos relacionados con proveedores, compras, inventarios, herramientas y almacenes, logrando una organización eficiente y una mayor coherencia en los datos.

A lo largo de este proyecto, se han aplicado técnicas de fragmentación horizontal y vertical, simulando la distribución de los datos en bases de datos más grandes, lo que resulta en una gestión óptima y una posible mejora en el rendimiento para el caso de bases de datos distribuidas. Además, el uso de índices en las tablas ha permitido mejorar la eficiencia de las consultas, y la creación de vistas simplifica el acceso a información compleja, facilitando a los usuarios finales la visualización y el análisis de los datos.

Asimismo, las consultas estructuradas han sido fundamentales para representar de manera formal las operaciones de consulta y manipulación de datos. Esto ha permitido construir expresiones para obtener información específica, como la obtención de inventarios por almacén o la relación entre proveedores y compras, logrando un diseño que favorece la toma de decisiones en tiempo real y el análisis de patrones en las compras y el inventario de herramientas.