REPORTE DE PRÁCTICA

DISEÑO DE BASE DE DATOS PARA GESTIÓN DE HERRAMIENTAS

ALUMNO: Lozada Garcia Ana Laura

PROFESOR: Cornejo Velazquez Eduardo

1. Introducción

El diseño de una base de datos relacional para la gestión de herramientas en un entorno de almacén y compras es clave para garantizar una administración eficiente de los recursos, tanto en la adquisición como en la distribución de herramientas. Esta base de datos permite mantener un control detallado del inventario, las órdenes de compra y los proveedores, asegurando que la información esté actualizada y sea accesible para los responsables de la gestión. A través de un sistema estructurado, la empresa puede mejorar su capacidad de respuesta ante las demandas operativas, optimizando los tiempos de procesamiento y reduciendo los errores en la gestión.

2. Marco teórico

Una base de datos relacional organiza la información en tablas relacionadas entre sí mediante claves primarias y foráneas. Este modelo es eficaz para empresas que manejan grandes volúmenes de datos, ya que permite almacenar, consultar y manipular la información de manera eficiente. En el contexto de la gestión de herramientas, la base de datos relacional es ideal para mantener un control preciso de las compras, inventarios y distribución de herramientas.

El control de inventarios en un almacén implica el registro detallado de las herramientas disponibles, su cantidad, estado y ubicación. Este proceso es vital para asegurar que siempre haya suficiente stock disponible para satisfacer las necesidades operativas y evitar interrupciones en el trabajo debido a la falta de herramientas esenciales. La gestión de inventarios en una base de datos permite automatizar este control y generar reportes de manera rápida y eficiente.

La normalización es un proceso de optimización del diseño de bases de datos que se lleva a cabo para eliminar la redundancia de datos y evitar inconsistencias. A través de la normalización, los datos se organizan en tablas independientes que contienen información relacionada, lo que facilita la gestión y mejora el rendimiento del sistema. Las tres primeras formas normales (1NF, 2NF, 3NF) son fundamentales para garantizar que los datos se almacenen de manera óptima y sin duplicidades.

El Diagrama Entidad-Relación es una representación gráfica de las entidades y sus relaciones en un sistema de información. En el caso de la gestión de herramientas, las entidades clave podrían ser: "Herramientas", "Proveedores", "Órdenes de Compra", "Entradas de Almacén" y "Salidas de Almacén". Este diagrama facilita la comprensión de cómo interactúan las distintas partes del sistema y asegura que el diseño conceptual esté bien estructurado antes de proceder con la implementación física.

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

Análisis de requisitos

Basado en la información proporcionada, los requisitos principales del sistema son:

- Gestionar información de proveedores
- Registrar y rastrear herramientas
- Manejar compras de herramientas
- Controlar el inventario en diferentes almacenes
- Permitir consultas sobre compras e inventario

Diagrama Entidad-Relación para Gestión de Herramientas

A continuación se presenta el Diagrama Entidad-Relación para el sistema de gestión de herramientas:

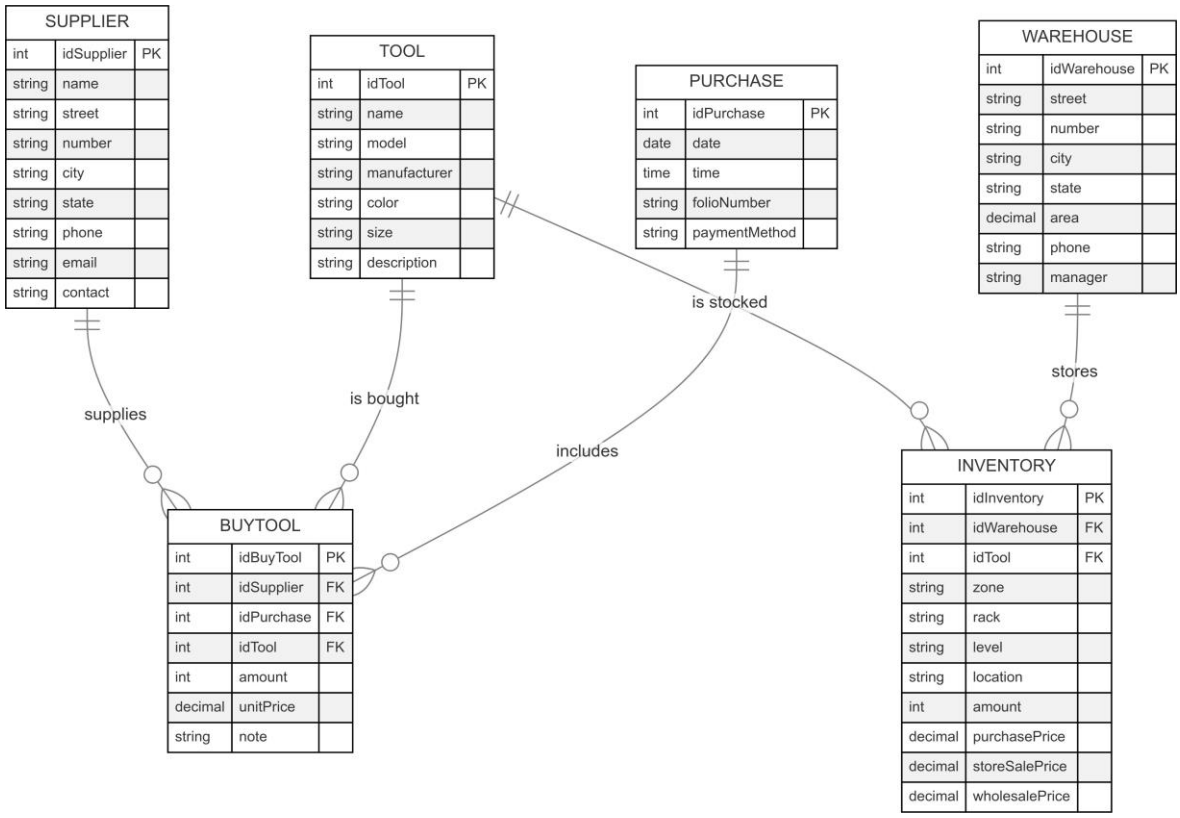

Figure 1: Diagrama Entidad - Relación propuesto.

Tabla de mapeo de atributos

Modelo relacional

Basado en el Modelo Entidad-Relación y la tabla de mapeo, el modelo relacional seria:

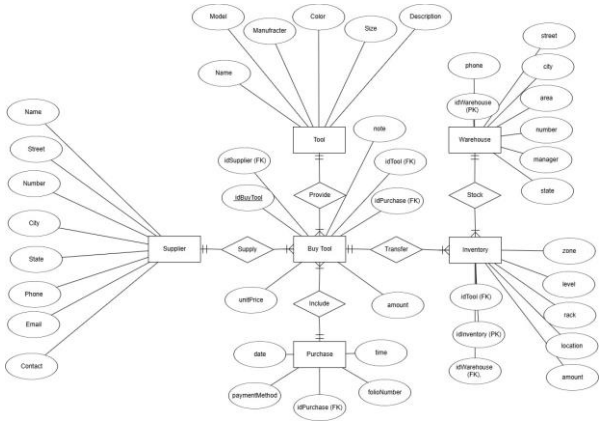

Figure 2: Modelo Entidad relacion

Table 1: Mapeo de atributos

Entidad	Atributos
supplier	idSupplier, name, street, number, city, state, phone, email, contact
tool	idTool, name, model, manufacturer, color, size, description
buyTool	idBuyTool, idSupplier, idPurchase, idTool, amount, unitPrice, note
purchase	idPurchase, date, time, folioNumber, paymentMethod
inventory	idInventory, idWarehouse, zone, rack, level, location, amount, purchasePrice, storeSalePrice, wholesalePrice
warehouse	idWarehouse, street, number, city, state, area, phone, manager

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

Donde los atributos subrayados son claves primarias y los marcados con * son claves foráneas.

Sentencias SQL

A continuación se presentan las sentencias SQL para crear las tablas:

Listing 1: Crear tablas en la base de datos.

```
CREATE TABLE supplier (
    idSupplier INT PRIMARY KEY,
    name VARCHAR(100),
    street VARCHAR(100),
    number VARCHAR(20),
    city VARCHAR(50),
    state VARCHAR(50),
    phone VARCHAR(20),
```

```
    email VARCHAR(100),
    contact VARCHAR(100)
);

CREATE TABLE tool (
    id Tool INT PRIMARY KEY,
    name VARCHAR(100),
    model VARCHAR(50),
    manufacturer VARCHAR(100),
    color VARCHAR(30),
    size VARCHAR(30),
    description TEXT
);

CREATE TABLE purchase (
    id Purchase INT PRIMARY KEY,
    date DATE,
    time TIME,
    folio Number VARCHAR(50),
    paymentMethod VARCHAR(50)
);

CREATE TABLE buyTool (
    idBuyTool INT PRIMARY KEY,
    idSupplier INT,
    idPurchase INT,
    id Tool INT,
    amount INT,
    unitPrice DECIMAL(10,2),
    note TEXT,
    FOREIGN KEY (idSupplier) REFERENCES supplier(idSupplier),
    FOREIGN KEY (id Purchase) REFERENCES purchase(id Purchase),
    FOREIGN KEY (idTool) REFERENCES tool(idTool)
);

CREATE TABLE warehouse (
    idWarehouse INT PRIMARY KEY,
    street VARCHAR(100),
    number VARCHAR(20),
    city VARCHAR(50),
    state VARCHAR(50),
    area DECIMAL(10,2),
    phone VARCHAR(20),
    manager VARCHAR(100)
);

CREATE TABLE inventory (
    idInventory INT PRIMARY KEY,
    idWarehouse INT,
    idTool INT,
    zone VARCHAR(50),
    rack VARCHAR(50),
    level VARCHAR(50),
    location VARCHAR(50),
```

```
    amount INT,
    purchase Price DECIMAL(10, 2),
    storeSalePrice DECIMAL(10, 2),
    wholesalePrice DECIMAL(10, 2),
    FOREIGN KEY (idWarehouse) REFERENCES warehouse(idWarehouse),
    FOREIGN KEY (idTool) REFERENCES tool(idTool)
);
```

Ahora, las sentencias SQL para insertar registros (se muestran 5 para cada tabla principal y 10 para buyTool):

Listing 2: Insertar registros en las tablas.

```
-- Insertar en supplier
INSERT INTO supplier VALUES
(1, 'ToolMaster-Inc.', 'Main-St', '123', 'Springfield', 'IL', '555-0101', 'contact@toolmas
(2, 'Hardware-Plus', 'Oak-Ave', '456', 'Shelbyville', 'IL', '555-0202', 'info@hardwareplus
(3,'Construction-Supplies-Co.','Elm-St','789','Capital-City','IL','555-0303','sales
(4, 'Build Right-Tools', 'Pine-Rd', '321', 'Ogdenville', 'IL', '555-0404', 'service@buildrig
(5,'Pro-Equipment-Ltd.','Maple-Dr','654','North-Haverbrook','IL','555-0505','info@p

-- Insertar en tool
INSERT INTO tool VALUES
(1, 'Hammer', 'H-2000', 'Hammertime', 'Red', '16-inch', 'Heavy-duty-claw-hammer'),
(2, 'Screwdriver', 'S-100', 'ScrewMaster', 'Blue', '8-inch', 'Phillips-head-screwdriver'),
(3,'Wrench','W-500','WrenchKing','Silver','12-inch','Adjustable-wrench'),
(4, 'Drill', 'D-3000', 'Power Drill', 'Yellow', '10-inch', 'Cordless-drill-with-variable-sp
(5,'Saw','S-750','CutPro','Green','18-inch','Handheld-circular-saw');

-- Insertar en purchase
INSERT INTO purchase VALUES
(1, '2023-01-15', '10:30:00', 'PO-001', 'Credit-Card'),
(2, '2023-02-20', '14:45:00', 'PO-002', 'Bank-Transfer'),
(3, '2023-03-25', '09:15:00', 'PO-003', 'Cash'),
(4, '2023-04-30', '16:00:00', 'PO-004', 'Credit-Card'),
(5, '2023-05-05', '11:30:00', 'PO-005', 'Bank-Transfer');

-- Insertar en buyTool
INSERT INTO buyTool VALUES
(1, 1, 1, 1, 10, 25.99, 'Bulk-purchase-of-hammers'),
(2, 1, 1, 2, 20, 12.50, 'Screwdrivers-for-new-project'),
(3, 2, 2, 3, 15, 18.75, 'Wrenches-for-maintenance-team'),
(4, 2, 2, 4, 5, 89.99, 'New-drills-for-carpenter-team'),
(5, 3, 3, 5, 8, 129.99, 'Saws-for-woodworking-class'),
(6, 3, 3, 1, 12, 24.99, 'Additional-hammers'),
(7, 4, 4, 2, 25, 11.99, 'Screwdrivers-for-retail'),
(8, 4, 4, 3, 18, 17.50, 'Wrenches-for-auto-shop'),
(9, 5, 5, 4, 7, 84.99, 'Drills-for-electrical-team'),
(10, 5, 5, 5, 10, 124.99, 'Saws-for-construction-project');

-- Insertar en warehouse
INSERT INTO warehouse VALUES
(1, 'Storage-Blvd', '100', 'Springfield', 'IL', 5000.00, '555-1010', 'Mike-Manager'),
(2, 'Inventory-St', '200', 'Shelbyville', 'IL', 7500.00, '555-2020', 'Sarah-Supervisor'),
(3, 'Stockpile-Ave', '300', 'Capital-City', 'IL', 10000.00, '555-3030', 'Tom-Tracker'),
(4, 'Depot-Dr', '400', 'Ogdenville', 'IL', 6000.00, '555-4040', 'Diana-Director'),
(5, 'Supply-Ln', '500', 'North-Haverbrook', 'IL', 8000.00, '555-5050', 'Peter-Planner');
```

```sql
-- Insertar en inventory
INSERT INTO inventory VALUES
(1, 1, 1, 'A', '1', '1', 'A1-1', 50, 20.00, 35.99, 30.99),
(2, 1, 2, 'A', '1', '2', 'A1-2', 100, 10.00, 18.99, 15.99),
(3, 2, 3, 'B', '2', '1', 'B2-1', 75, 15.00, 25.99, 22.99),
(4, 2, 4, 'B', '2', '2', 'B2-2', 30, 75.00, 119.99, 99.99),
(5, 3, 5, 'C', '3', '1', 'C3-1', 40, 100.00, 159.99, 139.99);
```

Consultas SQL

A continuación se presentan las sentencias SQL para obtener los datos de las consultas solicitadas:

Listing 3: Consulta de compras en enero.

```sql
SELECT s.name AS supplier_name, t.name AS tool_name, bt.amount, bt.unitPrice,
       (bt.amount * bt.unitPrice) AS total_price
FROM buyTool bt
JOIN supplier s ON bt.idSupplier = s.idSupplier
JOIN tool t ON bt.idTool = t.idTool
JOIN purchase p ON bt.idPurchase = p.idPurchase
WHERE MONTH(p.date) = 1
ORDER BY p.date;
```

Listing 4: Consulta de inventario en bodega 1.

```sql
SELECT t.name AS tool_name, i.amount,
       (i.amount * i.purchasePrice) AS total_cost
FROM inventory i
JOIN tool t ON i.idTool = t.idTool
WHERE i.idWarehouse = 1;
```

5. Conclusiones

El diseño e implementación de una base de datos relacional para la gestión de herramientas en un entorno de almacén y compras ofrece una solución eficiente para la organización y control de inventarios, compras y distribución de herramientas. La normalización asegura una estructura de datos óptima y libre de redundancias, mientras que la implementación en un DBMS proporciona un sistema escalable y fácil de mantener.

La automatización del control de inventarios y las órdenes de compra mejora la capacidad del almacén para responder a las demandas operativas de manera rápida y precisa, minimizando errores humanos y optimizando la eficiencia general del sistema.

Referencias Bibliográficas

References

[1] Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7th ed.). Pearson.

[2] Coronel, C., & Morris, S. (2016). Database Systems: Design, Implementation, & Management (12th ed.). Cengage Learning.

[3] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database System Concepts (7th ed.). McGraw-Hill Education.

[4] Ramakrishnan, R., & Gehrke, J. (2020). Database Management Systems (3rd ed.). McGraw-Hill Education.