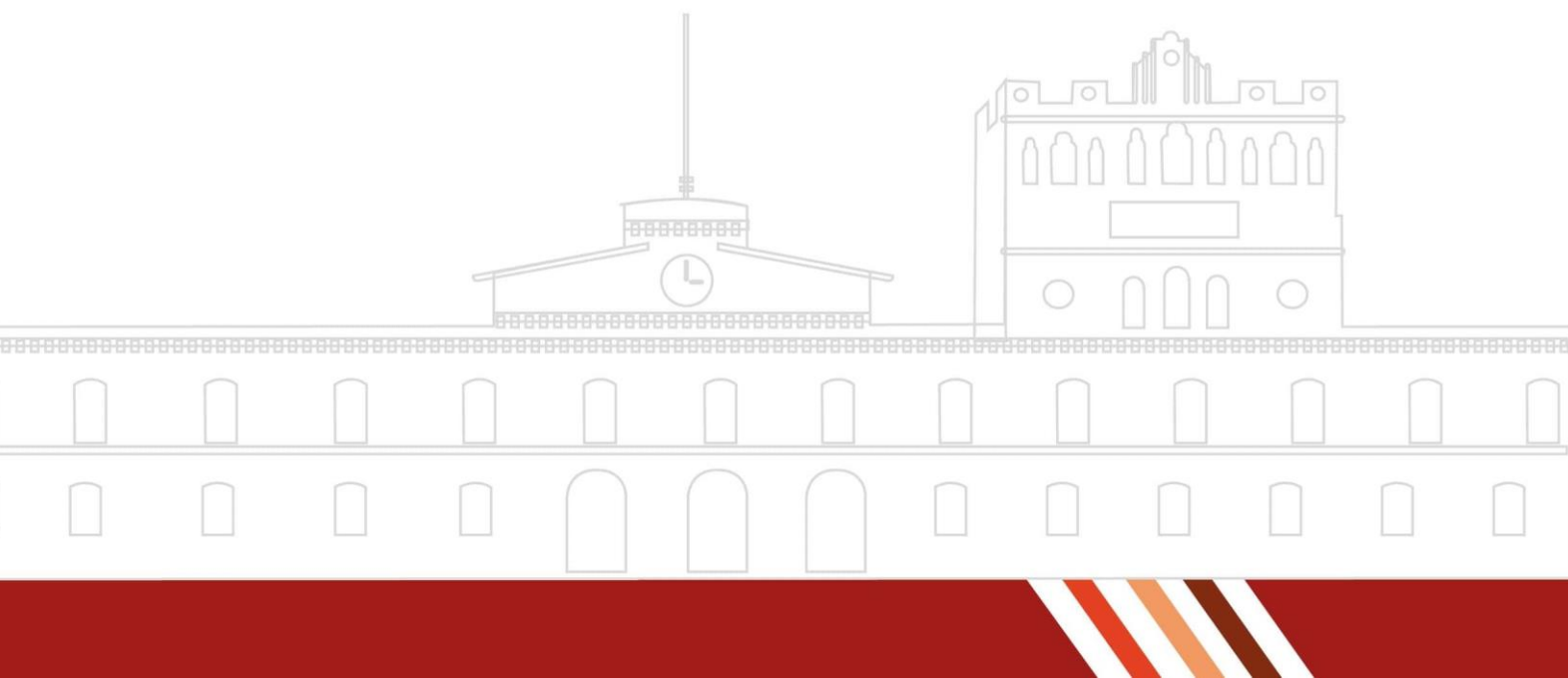


REPORTE DE PRÁCTICA

DISEÑO DE BASE DE DATOS PARA GESTIÓN DE HERRAMIENTAS

ALUMNO: Amuchategui Flores Braulio

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 crucial para optimizar el control de inventarios, el seguimiento de las compras y la distribución de productos. Esta base de datos tiene como objetivo mejorar la organización y administración de las herramientas, desde su adquisición hasta su almacenamiento y eventual distribución. Mediante el uso de un Sistema de Gestión de Bases de Datos (DBMS), se puede asegurar que la información esté bien estructurada, sea accesible y se mantenga actualizada, lo que facilita la toma de decisiones y mejora la eficiencia operativa en el entorno del almacén.

2. Marco teórico

Una base de datos relacional organiza los datos en tablas que están interconectadas mediante claves primarias y foráneas. Este modelo permite mantener la integridad de los datos y facilita la ejecución de consultas complejas mediante SQL. En el contexto de un almacén de herramientas, las bases de datos relacionales son ideales para gestionar grandes volúmenes de información sobre las herramientas, los proveedores, las compras y las existencias.

La gestión de almacenes involucra la administración del inventario de herramientas, que incluye tareas como el registro de entradas y salidas, el control de stock y la asignación de productos a distintas áreas. Por otro lado, la gestión de compras se refiere a la adquisición de herramientas a través de proveedores y la planificación de las necesidades futuras del almacén. La integración de ambas áreas en una base de datos permite un control eficiente del flujo de herramientas.

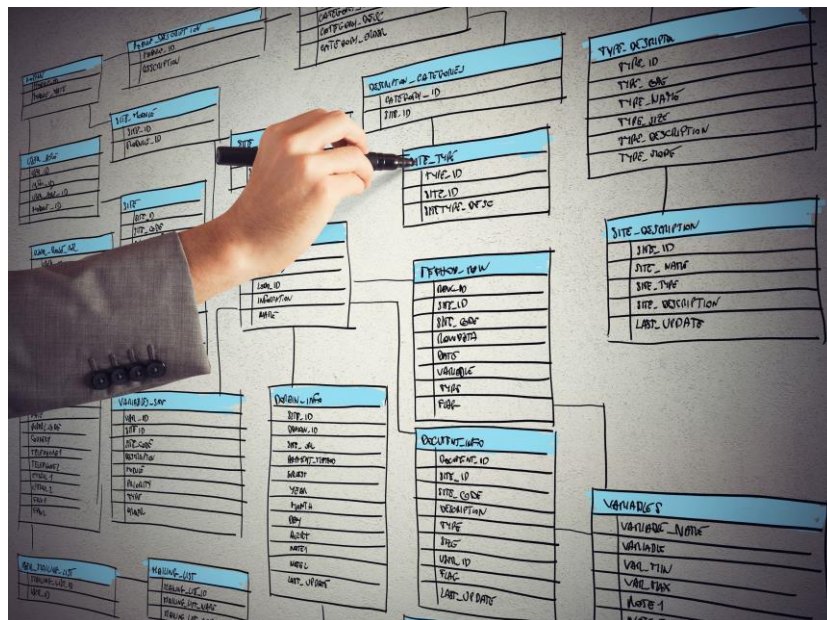
El modelo Entidad-Relación es una técnica de modelado utilizada para representar las entidades del sistema y las relaciones entre ellas. Para un almacén y compras, las entidades clave pueden incluir "Herramientas", "Proveedores", "Órdenes de Compra", "Entradas de Almacén", y "Salidas de Almacén". Este modelo conceptual ayuda a visualizar la estructura de la base de datos antes de su implementación física en el DBMS.

La normalización es un proceso que se utiliza para optimizar la estructura de la base de datos eliminando la redundancia de datos y mejorando la consistencia. A través de las formas normales (1NF, 2NF y 3NF), se garantiza que las tablas de la base de datos estén correctamente estructuradas, lo que permite evitar la duplicación innecesaria de datos y mejorar el rendimiento del sistema.

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:

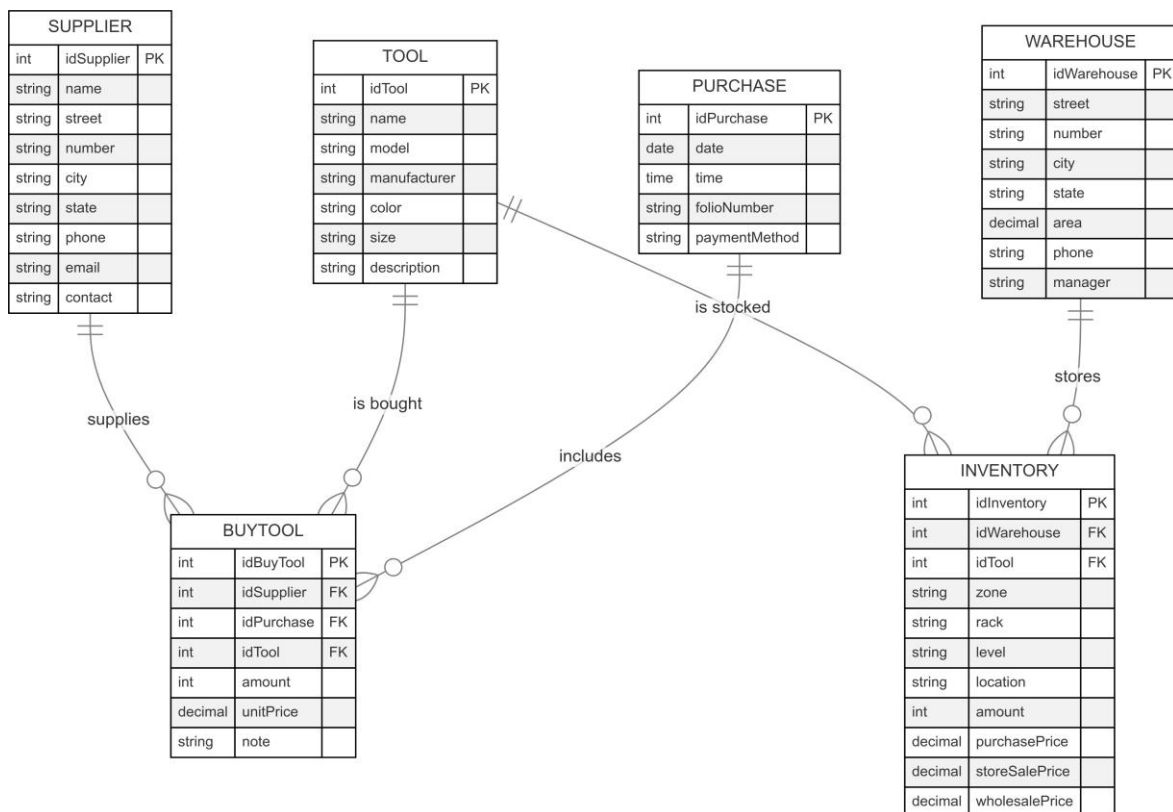


Figure 1: Diagrama Entidad - Relación propuesto.

Tabla de mapeo de atributos

Modelo relacional

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

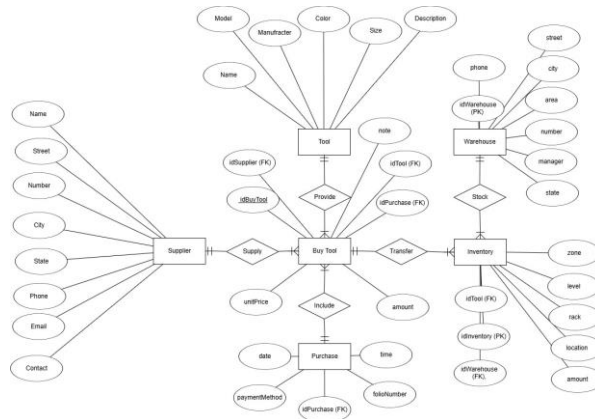


Figure 2: Modelo Entidad relacion

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,  
    idTool INT,  
    amount INT,  
    unitPrice DECIMAL( 10 , 2 ),  
    note TEXT,  
    FOREIGN KEY ( id Supplier ) REFERENCES supplier ( id Supplier ),  
    FOREIGN KEY ( id Purchase ) REFERENCES purchase ( id Purchase ),  
    FOREIGN KEY ( id Tool ) REFERENCES tool ( id Tool )  
);
```

```
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');

```


-- 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.

```
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.

```
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 proporciona una estructura eficiente para organizar y controlar las operaciones del almacén. La base de datos permite mantener un control preciso del inventario, gestionar las compras y facilitar la entrada y salida de herramientas de manera sistemática.

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.