

# REPORTE DE PRÁCTICA NO. 4

## DISEÑO DE BASE DE DATOS PARA GESTIÓN DE HER- RAMIENTAS

ALUMNO: JuanDiego Alvarado Moreno

CATEDRATICO: Dr. Eduardo Cornejo-Velázquez



## 1. Introducción

Esta práctica se enfoca en el diseño de una base de datos relacional para la gestión de herramientas en un entorno de almacén y compras. El proceso sistemático y metodológico para realizar el modelado de bases de datos relacionales es crucial para garantizar la integridad, eficiencia y escalabilidad de los sistemas de información.

## 2. Marco teórico

### Análisis de requerimientos

El análisis de requerimientos es la fase inicial en el diseño de bases de datos donde se identifican y documentan las necesidades del usuario y del sistema [1]. Este proceso implica la recopilación de información sobre los datos que se manejarán, las operaciones que se realizarán y las restricciones que se aplicarán.

### Diseño de vistas

El diseño de vistas implica la creación de representaciones parciales de la base de datos que se ajustan a las necesidades específicas de diferentes grupos de usuarios o aplicaciones [2]. Esto permite una mayor flexibilidad y seguridad en el acceso a los datos.

### Diseño conceptual

#### 2.3.1 Análisis de entidades

El análisis de entidades implica identificar y definir los objetos o conceptos principales sobre los que se almacenará información en la base de datos [3]. En nuestro caso, las entidades principales incluyen proveedores, herramientas, compras, inventario y almacenes.

#### 2.3.2 Análisis funcional

El análisis funcional se centra en identificar las operaciones y procesos que se realizarán con los datos almacenados [4]. Esto incluye operaciones como registrar compras, actualizar inventario y generar informes.

### Integración de vistas

La integración de vistas es el proceso de combinar diferentes perspectivas o vistas parciales del sistema en un diseño coherente y unificado [1]. Esto asegura que todas las necesidades de los usuarios estén representadas en el esquema final.

### Esquema conceptual global

El esquema conceptual global es una representación completa y abstracta de toda la base de datos, que integra todas las vistas y requisitos identificados [3]. Este esquema sirve como base para el diseño lógico y físico de la base de datos.

### 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:

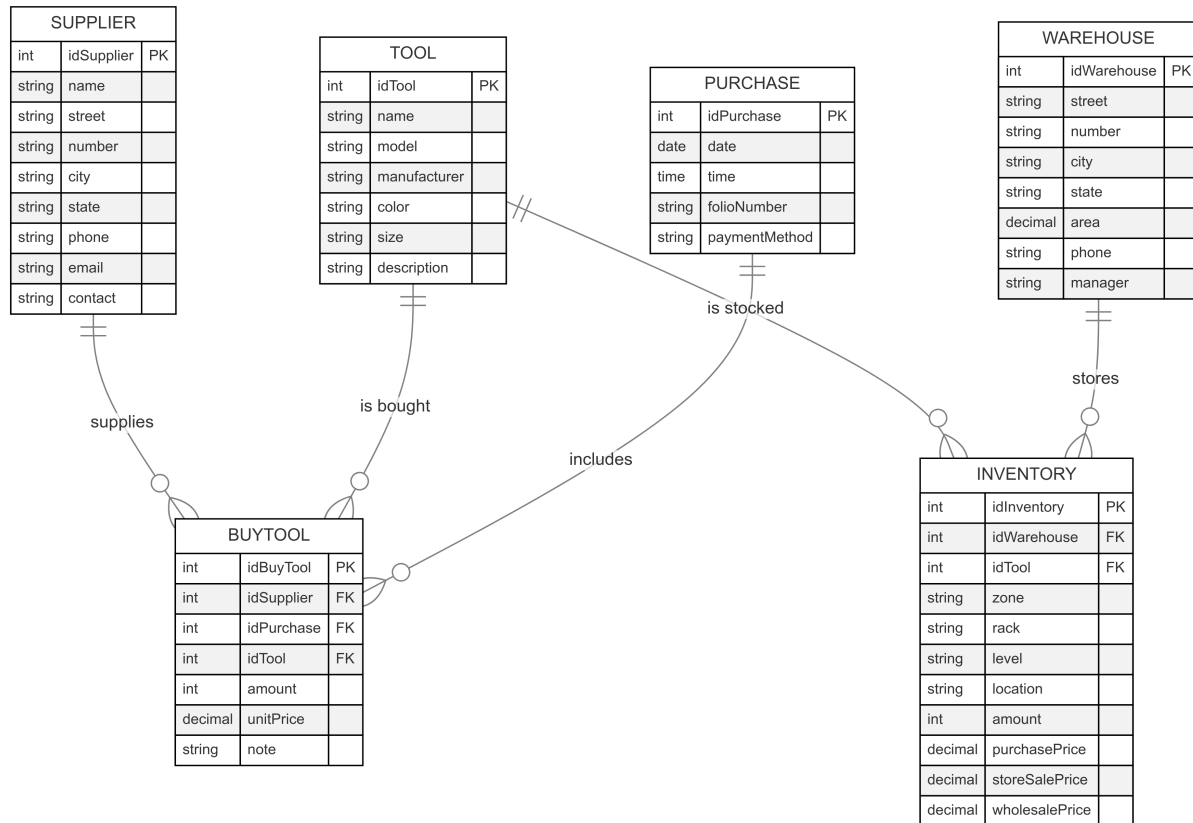


Figure 1: Diagrama Entidad - Relación propuesto.

### Tabla de mapeo de atributos

### Modelo relacional

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

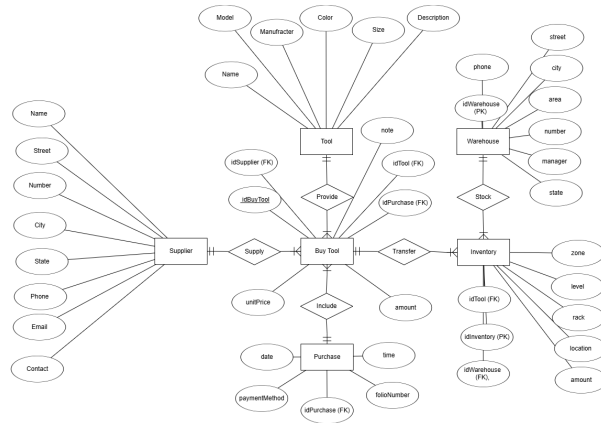


Figure 2: Modelo Entidad relacion

Table 1: Mapeo de atributos

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

- 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 (  
    idTool INT PRIMARY KEY,  
    name VARCHAR(100),  
    model VARCHAR(50),  
    manufacturer VARCHAR(100),  
    color VARCHAR(30),  
    size VARCHAR(30),  
    description TEXT  
);
```

```
CREATE TABLE purchase (  
    idPurchase INT PRIMARY KEY,  
    date DATE,  
    time TIME,  
    folioNumber 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 (idSupplier) REFERENCES supplier(idSupplier),  
    FOREIGN KEY (idPurchase) REFERENCES purchase(idPurchase),  
    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,
    purchasePrice 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, 'BuildRight-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', 'PowerDrill', 'Yellow', '10-inch', 'Cordless-drill-with-variable-spe
(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

A través de esta práctica, he fortalecido mis habilidades en el diseño de bases de datos relacionales, específicamente:

- Análisis de requisitos y traducción de necesidades de negocio a estructuras de datos.
- Creación de modelos Entidad-Relación y su posterior conversión a modelos relacionales.
- Implementación de esquemas de base de datos utilizando SQL, incluyendo la definición de tablas con claves primarias y foráneas.
- Desarrollo de consultas SQL para extraer información específica de múltiples tablas relacionadas.

Un aspecto que me sorprendió gratamente (WOW) fue la capacidad de SQL para unir múltiples tablas y realizar cálculos en una sola consulta, como se ve en la consulta de compras en enero.

Por otro lado, un desafío (WTF) fue asegurar la integridad referencial en la tabla buyTool, que requiere manejar tres claves foráneas simultáneamente. Esto resalta la importancia de un diseño cuidadoso y la consideración de las relaciones entre entidades.

Esta práctica ha reforzado mi comprensión de cómo un buen diseño de base de datos puede facilitar la gestión eficiente de información compleja en escenarios del mundo real, como la gestión de inventario y compras de herramientas.

## 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.