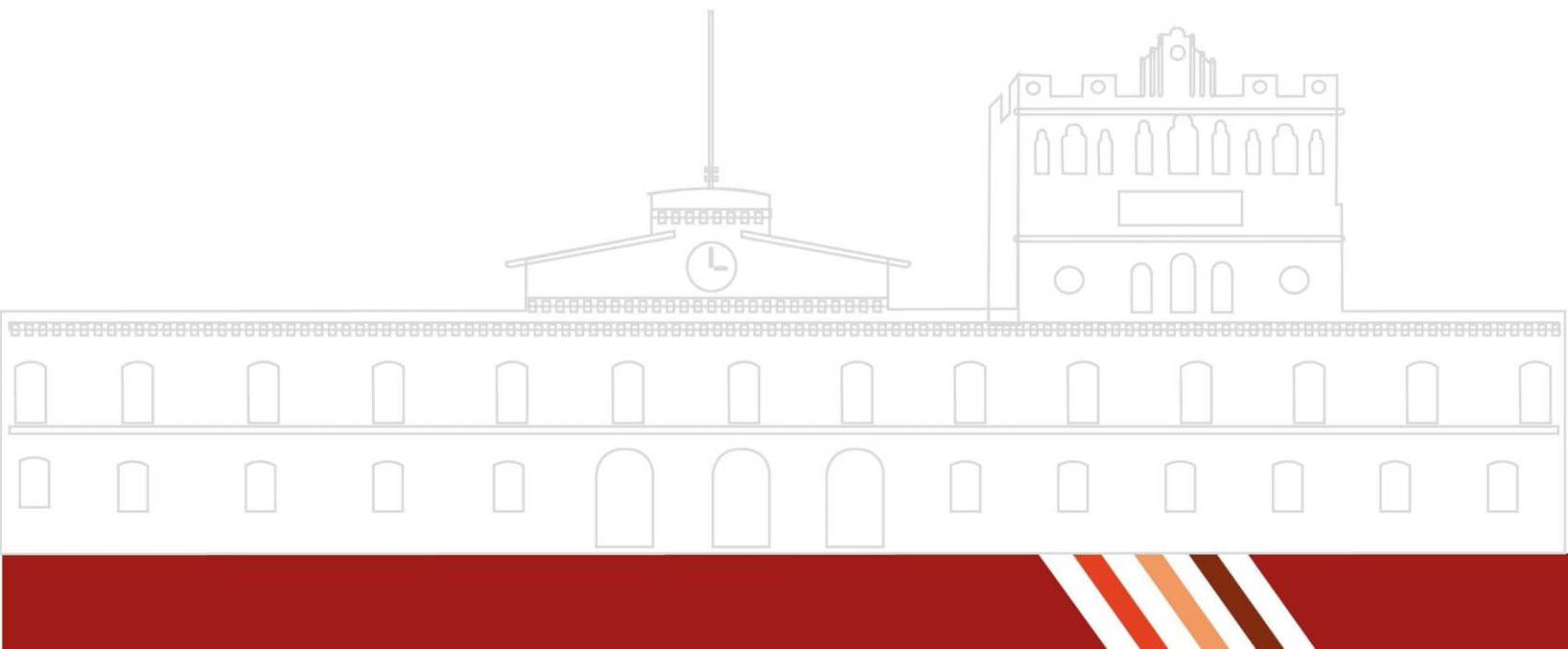


REPORTE DE PRÁCTICA

2.3 Práctica. Inventario herramientas–

ALUMNO: Cristian Cristobal Silverio
Dr. Eduardo Cornejo-Velázquez



1. Introducción

En la actualidad, la gestión eficiente de la distribución y venta de productos es crucial para el éxito de cualquier empresa, especialmente en sectores competitivos como el de herramientas para el hogar. Este documento presenta el diseño de una base de datos para una empresa distribuidora de herramientas, que busca optimizar sus operaciones tanto en la gestión de inventarios como en las ventas directas al consumidor y a minoristas.

2. Marco teórico

Ánalysis de requerimiento

Con el análisis de requerimientos se formaliza la documentación del proyecto de creación de la base de datos distribuida donde se incluyen las expectativas para el sistema final.

El análisis de requerimientos se enfoca en los objetivos de:

1. Rendimiento (performance).
2. Fiabilidad y disponibilidad (reliability and availability).
3. Economía (economics).
4. Capacidad de ampliación (expandability).

Diseño de vistas

en esta etapa se busca realizar actividades que permiten definir las interfaces para los usuarios finales a través de las que se realizará el flujo de datos para cubrir

los requerimientos del sistema. Las vistas suelen usarse para consultar, simplificar y personalizar la percepción de la base de datos para cada usuario, las cuales tienen la misma estructura que una tabla filas y columnas. Las vistas de base de datos se guardan como consultas con nombre y se pueden utilizar para guardar consultas completas que se utilizan con frecuencia, que permiten a los usuarios obtener acceso a los datos por medio de la vista.

Diseño conceptual

el Diseño conceptual (Conceptual design), en el que se analiza el contexto del sistema para determinar las entidades y relaciones que serán incluidas en la base de datos.

El diseño conceptual es una etapa necesaria en la creación de bases de datos, ya que el esquema conceptual generado es la base desde la cual se creará, modificará y extenderá la base de datos. Un buen diseño conceptual permite crear bases de datos más compactas, entendedoras, simples y extensibles.

Análisis de entidades

Se determinan las entidades, sus atributos y las relaciones entre ellas.

Análisis funcional

Se definen las funciones fundamentales para el modelo propuesto.

El análisis funcional es un proceso esencial para el desarrollo de sistemas de software y soluciones tecnológicas en general. Su objetivo principal es descomponer un sistema o proceso en sus componentes más pequeños y comprender cómo interactúan entre sí para lograr un objetivo común. A través del análisis funcional, se busca identificar y documentar las funciones, procesos y características que deben estar presentes en la solución final.

Un análisis funcional efectivo es fundamental para el éxito del proyecto, ya que sienta las bases para el diseño, la implementación y la validación de la solución. Proporciona una visión detallada de lo que se debe lograr y sirve como guía para los equipos de desarrollo y las partes interesadas durante todo el ciclo de vida del proyecto.

Integración de vistas

La integración de vistas es el proceso en el que se diseñan varios esquemas conceptuales individuales (vistas), y posteriormente se funden o combinan en un esquema conceptual global, que representa todos los requisitos de información del sistema. Este enfoque suele utilizarse con bases de datos de gran

tamaño y complejidad. El principal objetivo de la integración de vistas es encontrar todas las partes de los esquemas conceptuales (EC) de entrada que se refieren a la misma porción de la realidad, y unificar su representación.

Esta actividad se llama integración de esquemas y resulta muy complicada, ya que una misma parte de la realidad suele estar modelada (representada) de forma distinta en cada esquema.

Esquema conceptual global

El Esquema Conceptual Global (global conceptual schema, GCS) junto con los patrones de acceso a la información, definidos en el diseño de interfaces de usuario, son la entrada para la etapa de diseño de distribución (distribution design).

Por su parte, el diseño conceptual puede ser dividido en dos grupos de actividades:

- Análisis de entidades (Entity analysis). Se determinan las entidades, sus atributos y las relaciones entre ellas.
- Análisis funcional (Functional analysis). Se definen las funciones fundamentales para el modelo propuesto.

El diseño de distribución tiene el propósito de diseñar el Esquema Conceptual Local (local conceptual schema, LCS) para distribuir las entidades a través de los sitios (nodos) del sistema distribuido.

3. Desarrollo

1. Creamos las tablas:

4 rows in set (0.01 sec)

```
mysql> CREATE DATABASE tool_distribution;
Query OK, 1 row affected (0.05 sec)
```

```
mysql> USE tool_distribution;
Database changed
```

```
mysql> CREATE TABLE supplier (
  ->   idSupplier INT PRIMARY KEY,
  ->   name VARCHAR(100) NOT NULL,
  ->   street VARCHAR(100),
  ->   number VARCHAR(10),
  ->   city VARCHAR(50),
  ->   state VARCHAR(50),
  ->   phone VARCHAR(15),
  ->   email VARCHAR(100),
  ->   contact VARCHAR(100)
  -> );
```

Query OK, 0 rows affected (0.09 sec)

```
mysql> CREATE TABLE tool (
  ->   idTool INT PRIMARY KEY,
  ->   name VARCHAR(100) NOT NULL,
  ->   model VARCHAR(50),
  ->   manufacturer VARCHAR(100),
  ->   color VARCHAR(30),
  ->   size VARCHAR(30),
  ->   description TEXT
  -> );
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> CREATE TABLE warehouse (
  ->   idWarehouse INT PRIMARY KEY,
  ->   street VARCHAR(100),
  ->   number VARCHAR(10),
  ->   city VARCHAR(50),
  ->   state VARCHAR(50),
  ->   area FLOAT,
  ->   phone VARCHAR(15),
  ->   manager VARCHAR(100)
  -> );
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> CREATE TABLE purchase (
  ->   idPurchase INT PRIMARY KEY,
  ->   date DATE NOT NULL,
```

MySQL 9.0 Command Line Cli X + v

```
->   street VARCHAR(100),
->   number VARCHAR(10),
->   city VARCHAR(50),
->   state VARCHAR(50),
->   area FLOAT,
->   phone VARCHAR(15),
->   manager VARCHAR(100)
-> );
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> CREATE TABLE purchase (
  ->   idPurchase INT PRIMARY KEY,
  ->   date DATE NOT NULL,
  ->   time TIME NOT NULL,
  ->   folioNumber VARCHAR(50) NOT NULL,
  ->   paymentMethod VARCHAR(50)
  -> );
```

Query OK, 0 rows affected (0.08 sec)

```
mysql> CREATE TABLE inventory (
  ->   idInventory INT PRIMARY KEY,
  ->   idWarehouse INT,
  ->   zone VARCHAR(30),
  ->   rack VARCHAR(10),
  ->   level VARCHAR(10),
  ->   location VARCHAR(30),
  ->   amount INT NOT NULL,
  ->   purchasePrice DECIMAL(10, 2),
  ->   storeSalePrice DECIMAL(10, 2),
  ->   wholesalePrice DECIMAL(10, 2),
  ->   FOREIGN KEY (idWarehouse) REFERENCES warehouse(idWarehouse)
  -> );
```

Query OK, 0 rows affected (0.06 sec)

```
mysql> CREATE TABLE buyTool (
  ->   idBuyTool INT PRIMARY KEY,
  ->   idSupplier INT,
  ->   idPurchase INT,
  ->   idTool INT,
  ->   amount INT NOT NULL,
  ->   unitPrice DECIMAL(10, 2),
  ->   note TEXT,
  ->   FOREIGN KEY (idSupplier) REFERENCES supplier(idSupplier),
  ->   FOREIGN KEY (idPurchase) REFERENCES purchase(idPurchase),
  ->   FOREIGN KEY (idTool) REFERENCES tool(idTool)
```

Población de la bd y consultas

```
mysql> INSERT INTO supplier (idSupplier, name, street, number, city, state, phone, email, contact) VALUES
-> (1, 'Supplier A', 'Main St', '123', 'City A', 'State A', '123-456-7890', 'contactA@example.com', 'John Doe'),
-> (2, 'Supplier B', 'Second St', '456', 'City B', 'State B', '123-456-7891', 'contactB@example.com', 'Jane Doe'),
-> (3, 'Supplier C', 'Third St', '789', 'City C', 'State C', '123-456-7892', 'contactC@example.com', 'Bob Smith'),
-> (4, 'Supplier D', 'Fourth St', '012', 'City D', 'State D', '123-456-7893', 'contactD@example.com', 'Alice Johnson'),
-> (5, 'Supplier E', 'Fifth St', '345', 'City E', 'State E', '123-456-7894', 'contactE@example.com', 'Charlie Brown');
-> ^Z^C
mysql> (5, 'Supplier E', 'Fifth St', '345', 'City E', 'State E', '123-456-7894', 'contactE@example.com', 'Charlie Brown');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '5, 'Supp
894' 'contac' at line 1
mysql> INSERT INTO tool (idTool, name, model, manufacturer, color, size, description) VALUES
-> (1, 'Hammer', 'M100', 'Brand A', 'Red', 'Medium', 'A sturdy hammer.'),
-> (2, 'Screwdriver', 'S200', 'Brand B', 'Blue', 'Small', 'A reliable screwdriver.'),
-> (3, 'Wrench', 'W300', 'Brand C', 'Green', 'Large', 'A durable wrench.'),
-> (4, 'Pliers', 'P400', 'Brand D', 'Yellow', 'Medium', 'Strong pliers.'),
-> (5, 'Saw', 'S500', 'Brand E', 'Black', 'Large', 'A sharp saw.');
Query OK, 5 rows affected (0.05 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO supplier (idSupplier, name, street, number, city, state, phone, email, contact) VALUES
-> (1, 'Supplier A', 'Main St', '123', 'City A', 'State A', '123-456-7890', 'contactA@example.com', 'John Doe'),
-> (2, 'Supplier B', 'Second St', '456', 'City B', 'State B', '123-456-7891', 'contactB@example.com', 'Jane Doe'),
-> (3, 'Supplier C', 'Third St', '789', 'City C', 'State C', '123-456-7892', 'contactC@example.com', 'Bob Smith'),
-> (4, 'Supplier D', 'Fourth St', '012', 'City D', 'State D', '123-456-7893', 'contactD@example.com', 'Alice Johnson'),
-> (5, 'Supplier E', 'Fifth St', '345', 'City E', 'State E', '123-456-7894', 'contactE@example.com', 'Charlie Brown');
Query OK, 5 rows affected (0.04 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO warehouse (idWarehouse, street, number, city, state, area, phone, manager) VALUES
-> (1, 'Warehouse St', '100', 'City W', 'State W', 1000, '123-456-7895', 'Manager A'),
-> (2, 'Warehouse St', '101', 'City X', 'State X', 1500, '123-456-7896', 'Manager B'),
-> (3, 'Warehouse St', '102', 'City Y', 'State Y', 2000, '123-456-7897', 'Manager C'),
-> (4, 'Warehouse St', '103', 'City Z', 'State Z', 2500, '123-456-7898', 'Manager D'),
-> (5, 'Warehouse St', '104', 'City Q', 'State Q', 3000, '123-456-7899', 'Manager E');
Query OK, 5 rows affected (0.04 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO purchase (idPurchase, date, time, folioNumber, paymentMethod) VALUES
-> (1, '2024-01-05', '10:00:00', 'F12345', 'Credit Card'),
-> (2, '2024-01-15', '11:30:00', 'F12346', 'Cash'),
-> (3, '2024-01-25', '14:00:00', 'F12347', 'Bank Transfer'),
-> (4, '2024-02-05', '15:00:00', 'F12348', 'Credit Card'),
-> (5, '2024-02-15', '16:30:00', 'F12349', 'Cash');
Query OK, 5 rows affected (0.04 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO inventory (idInventory, idWarehouse, zone, rack, level, location, amount, purchasePrice, storeSalePrice, wholesalePrice) VALUES
-> (1, 1, 'Zone A', 'Rack 1', 'Level 1', 'Location 1', 50, 10.00, 12.00, 9.00),
-> (2, 1, 'Zone B', 'Rack 2', 'Level 2', 'Location 2', 30, 15.00, 18.00, 14.00),
-> (3, 2, 'Zone C', 'Rack 3', 'Level 3', 'Location 3', 100, 20.00, 25.00, 19.00),
-> (4, 2, 'Zone D', 'Rack 4', 'Level 4', 'Location 4', 70, 25.00, 30.00, 24.00),
-> (5, 3, 'Zone E', 'Rack 5', 'Level 5', 'Location 5', 90, 30.00, 35.00, 29.00);
Query OK, 5 rows affected (0.05 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

- Lista de compras realizadas en enero:

```
mysql> SELECT
->     p.date,
->     s.name AS supplierName,
->     t.name AS toolName,
->     b.amount,
->     b.unitPrice,
->     (b.amount * b.unitPrice) AS totalPrice
-> FROM
->     buyTool b
-> JOIN
->     supplier s ON b.idSupplier = s.idSupplier
-> JOIN
->     tool t ON b.idTool = t.idTool
-> JOIN
->     purchase p ON b.idPurchase = p.idPurchase
-> WHERE
->     p.date BETWEEN '2024-01-01' AND '2024-01-31'
-> ORDER BY
->     p.date;
+-----+-----+-----+-----+-----+-----+
| date | supplierName | toolName | amount | unitPrice | totalPrice |
+-----+-----+-----+-----+-----+-----+
| 2024-01-05 | Supplier A | Hammer | 10 | 9.00 | 90.00 |
| 2024-01-15 | Supplier B | Screwdriver | 20 | 14.00 | 280.00 |
| 2024-01-25 | Supplier C | Wrench | 30 | 19.00 | 570.00 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

- Listado del inventario de la bodega 1:

```
mysql> SELECT
->     t.name AS toolName,
->     i.amount,
->     (i.amount * i.purchasePrice) AS totalCost
-> FROM
->     inventory i
-> JOIN
->     tool t ON i.idInventory = t.idTool
-> WHERE
->     i.idWarehouse = 1;
+-----+-----+-----+
| toolName | amount | totalCost |
+-----+-----+-----+
| Hammer  |      50 |    500.00 |
| Screwdriver |      30 |    450.00 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

4. Conclusiones

En esta práctica se diseñó e implementó una base de datos para gestionar las operaciones de compra, inventario y almacenamiento de herramientas en una empresa. Se crearon las tablas necesarias para organizar la información de proveedores, herramientas, compras y almacenes, y se realizaron consultas que permiten extraer datos relevantes para la gestión. Esto permitió aprender y aplicar conceptos clave de bases de datos, como la creación de tablas, inserción de datos y consultas, optimizando el manejo de la información empresarial.

Referencias Bibliográficas

References

[1]

http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/15_proceso_de_desarrollo_de_la_base_de_datos.html