CASO PRÁCTICO 1

CREACIÓN DE SENTENCIAS SQL Y EJECUCIÓN EN UN SISTEMA GESTOR DE BASES DE DATOS

Contexto

Estoy trabajando en una compañía de venta de electrodomésticos, y mi tarea es gestionar los datos de la empresa utilizando un sistema gestor de bases de datos. A continuación, presento los pasos realizados para cumplir con los requisitos del caso práctico, implementando las mejores prácticas en cada etapa.

Paso 1: Creación de la base de datos "Prueba"

El primer paso es crear una base de datos llamada Prueba. Para evitar errores si ya existe una base de datos con este nombre, utilizo el comando **CREATE DATABASE** con la cláusula **IF NOT EXISTS.**

```
mysql> CREATE DATABASE IF NOT EXISTS Prueba;
Query OK, 1 row affected (0.01 sec)
```

Verificación de la creación de la base de datos

Para confirmar que la base de datos se creó correctamente, uso el comando:

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| prueba             |
| sys                |
| tiendamuebles      |
+--------------------+
6 rows in set (0.00 sec)
```

Esto lista todas las bases de datos del sistema. Como comprobamos prueba aparece en la lista, significa que se ha creado correctamente.

Posteriormente, selecciono la base de datos para trabajar con ella:

```
mysql> USE Prueba;
Database changed
```

Comprobación de la estructura inicial

Verifico que la base de datos está vacía, es decir, que no tiene tablas creadas, con el siguiente comando:

```
mysql> SHOW TABLES;
Empty set (0.02 sec)
```

Al ejecutar este comando, no aparece ninguna tabla, ya que aún no se ha creado ninguna.

Paso 2: Creación de la tabla "Artículos"

El siguiente paso es diseñar una tabla llamada Artículos para gestionar los productos del inventario. El diseño considera los siguientes campos:

1. **Codarticulo**:
    - Tipo: INT.
    - Propiedades: Identificador único, autoincremental (AUTO_INCREMENT), clave primaria (PRIMARY KEY).
2. **Descripción**:
    - Tipo: VARCHAR(100).
    - Propiedades: Almacena hasta 100 caracteres, obligatorio (NOT NULL), no se permiten valores duplicados (UNIQUE).
3. **Precio**:
    - Tipo: DECIMAL(10,2).
    - Propiedades: Representa valores monetarios con dos decimales, obligatorio (NOT NULL), restricción para que el precio no sea negativo (CHECK).
4. **Inventario**:
    - Tipo: INT.
    - Propiedades: Almacena la cantidad disponible, obligatorio (NOT NULL), no se permiten valores negativos (CHECK).

Comando para crear la tabla

Con esta estructura, el comando para crear la tabla es el siguiente:

```
mysql> CREATE TABLE IF NOT EXISTS Artículos (
    ->     Codarticulo INT AUTO_INCREMENT PRIMARY KEY,
    ->     Descripción VARCHAR(100) NOT NULL UNIQUE,
    ->     Precio DECIMAL(10,2) NOT NULL CHECK (Precio >= 0),
    ->     Inventario INT NOT NULL CHECK (Inventario >= 0)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

Verificación de la creación de la tabla
Para comprobar que la tabla Artículos se creó correctamente, utilizo:

```
mysql> SHOW TABLES;
+----------------+
| Tables_in_prueba |
+----------------+
| artículos      |
+----------------+
1 row in set (0.00 sec)
```

Esto lista las tablas existentes en la base de datos Prueba. como artículos aparece en la lista, se confirma que la tabla fue creada.

También verifico la estructura de la tabla con:

```
mysql> DESCRIBE Artículos;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| Codarticulo | int           | NO   | PRI | NULL    | auto_increment |
| Descripción | varchar(100)  | NO   | UNI | NULL    |                |
| Precio      | decimal(10,2) | NO   |     | NULL    |                |
| Inventario  | int           | NO   |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

Esto muestra los campos, los tipos de datos, las restricciones y las propiedades de la tabla.

Paso 3: Inserción de registros en la tabla

Con la tabla artículos creada, el siguiente paso es insertar registros representativos de un inventario típico. Los registros incluyen descripciones, precios y cantidades en inventario:

```
mysql> INSERT INTO Artículos (Descripción, Precio, Inventario) VALUES
    -> ('Ratón inalámbrico', 15.99, 25),
    -> ('Teclado mecánico', 49.99, 10),
    -> ('Monitor LED 24 pulgadas', 189.50, 5),
    -> ('Disco duro externo 1TB', 79.99, 12),
    -> ('Cargador USB-C', 9.99, 50),
    -> ('Laptop Gamer', 1299.99, 3),
    -> ('Pendrive 64GB', 14.99, 100),
    -> ('Impresora láser', 239.50, 7);
Query OK, 8 rows affected (0.02 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

Explicación de los datos

- Los precios están representados en formato decimal con dos dígitos después del punto, como es común en valores monetarios.
- Las cantidades (Inventario) son valores enteros, representando unidades físicas de cada artículo.
- Los nombres son descriptivos y representativos de un inventario típico en una empresa de venta de electrodomésticos.

Paso 4: Verificación de los registros

Una vez insertados los registros, verifico que se hayan añadido correctamente con el siguiente comando:

```
mysql> SELECT * FROM Artículos;
+-------------+-------------------------+---------+------------+
| Codarticulo | Descripción             | Precio  | Inventario |
+-------------+-------------------------+---------+------------+
|           1 | Ratón inalámbrico       |   15.99 |         25 |
|           2 | Teclado mecánico        |   49.99 |         10 |
|           3 | Monitor LED 24 pulgadas |  189.50 |          5 |
|           4 | Disco duro externo 1TB  |   79.99 |         12 |
|           5 | Cargador USB-C          |    9.99 |         50 |
|           6 | Laptop Gamer            | 1299.99 |          3 |
|           7 | Pendrive 64GB           |   14.99 |        100 |
|           8 | Impresora láser         |  239.50 |          7 |
+-------------+-------------------------+---------+------------+
8 rows in set (0.00 sec)
```

Este comando muestra todos los campos y valores de la tabla Artículos, confirmando que los registros se insertaron de forma adecuada.

Consultas adicionales y validaciones

Realizo consultas adicionales para validar los datos y demostrar el uso práctico de la tabla:

1. Contar el número total de artículos en el inventario:

```
mysql> SELECT COUNT(*) AS TotalArtículos FROM Artículos;
+----------------+
| TotalArtículos |
+----------------+
|              8 |
+----------------+
1 row in set (0.00 sec)
```

2. Consultar los artículos con menos de 10 unidades en el inventario:

```
mysql> SELECT * FROM Artículos WHERE Inventario < 10;
+-------------+-------------------------+---------+------------+
| Codarticulo | Descripción             | Precio  | Inventario |
+-------------+-------------------------+---------+------------+
|           3 | Monitor LED 24 pulgadas |  189.50 |          5 |
|           6 | Laptop Gamer            | 1299.99 |          3 |
|           8 | Impresora láser         |  239.50 |          7 |
+-------------+-------------------------+---------+------------+
3 rows in set (0.00 sec)
```

3. Ordenar los artículos por precio de mayor a menor:

```
mysql> SELECT * FROM Artículos ORDER BY Precio DESC;
+-------------+-------------------------+---------+------------+
| Codarticulo | Descripción             | Precio  | Inventario |
+-------------+-------------------------+---------+------------+
|           6 | Laptop Gamer            | 1299.99 |          3 |
|           8 | Impresora láser         |  239.50 |          7 |
|           3 | Monitor LED 24 pulgadas |  189.50 |          5 |
|           4 | Disco duro externo 1TB  |   79.99 |         12 |
|           2 | Teclado mecánico        |   49.99 |         10 |
|           1 | Ratón inalámbrico       |   15.99 |         25 |
|           7 | Pendrive 64GB           |   14.99 |        100 |
|           5 | Cargador USB-C          |    9.99 |         50 |
+-------------+-------------------------+---------+------------+
8 rows in set (0.00 sec)
```

4. Calcular el valor total del inventario (precio x inventario):

```
mysql> SELECT Descripción, Precio, Inventario, (Precio * Inventario) AS ValorTotal FROM Artículos;
+-------------------------+---------+------------+------------+
| Descripción             | Precio  | Inventario | ValorTotal |
+-------------------------+---------+------------+------------+
| Ratón inalámbrico       |   15.99 |         25 |     399.75 |
| Teclado mecánico        |   49.99 |         10 |     499.90 |
| Monitor LED 24 pulgadas |  189.50 |          5 |     947.50 |
| Disco duro externo 1TB  |   79.99 |         12 |     959.88 |
| Cargador USB-C          |    9.99 |         50 |     499.50 |
| Laptop Gamer            | 1299.99 |          3 |    3899.97 |
| Pendrive 64GB           |   14.99 |        100 |    1499.00 |
| Impresora láser         |  239.50 |          7 |    1676.50 |
+-------------------------+---------+------------+------------+
8 rows in set (0.00 sec)
```

Reflexión y creatividad

Este caso práctico me ha permitido diseñar, implementar y gestionar una base de datos para una empresa de venta de electrodomésticos. Considero que el trabajo tiene varios puntos clave que destacan:

- **Diseño optimizado:** Me aseguré de incluir restricciones como NOT NULL, UNIQUE y CHECK, que garantizan la integridad y la calidad de los datos almacenados en la tabla. Esto ayuda a evitar inconsistencias y errores.
- **Escalabilidad:** He diseñado la base de datos pensando en el futuro, de modo que pueda ampliarse fácilmente. Por ejemplo, sería posible añadir tablas relacionadas, como una tabla de categorías de artículos o una para los empleados responsables de gestionar el inventario.
- **Consultas útiles:** Las consultas adicionales que he incluido son herramientas prácticas que permiten extraer información valiosa. Estas consultas, como identificar artículos con bajo inventario o calcular el valor total del inventario, son fundamentales para la toma de decisiones dentro de una empresa.
- **Creatividad:** Incorporé restricciones y validaciones que aseguran que los datos en la base de datos sean confiables. Además, las consultas avanzadas que diseñé ofrecen posibilidades de análisis en profundidad, lo que aporta valor al uso de la base de datos en escenarios reales.

En resumen, este proyecto me ha permitido aplicar conocimientos técnicos mientras diseño una solución práctica, escalable y profesional que responde a las necesidades de la empresa.

Tabla de Resumen de la Estructura de la Tabla "Artículos"

Campo	Tipo de dato	Descripción	Restricciones
Codarticulo	INT	Identificador único para cada artículo.	AUTO_INCREMENT, PRIMARY KEY
Descripción	VARCHAR(100)	Breve descripción del artículo.	NOT NULL, UNIQUE
Precio	DECIMAL(10,2)	Precio del artículo con dos decimales.	NOT NULL, CHECK (Precio >= 0)
Inventario	INT	Cantidad disponible en el inventario.	NOT NULL, CHECK (Inventario >= 0)

Sección de Bibliografía o Fuentes

Bibliografía y fuentes consultadas:

1. **Documentación oficial de MySQL**
   - URL: https://dev.mysql.com/doc/
   - Consulté esta fuente para verificar las sintaxis y ejemplos relacionados con la creación de tablas y restricciones en MySQL.
2. **W3Schools - SQL Tutorial**
   - URL: https://www.w3schools.com/sql/
   - Este tutorial me ayudó a repasar comandos SQL como CREATE TABLE, INSERT INTO y consultas avanzadas con SELECT.