

Documentación técnica

Relación entre tablas



Para el diseño de la base de datos se siguió el modelo entidad-relación proporcionado en la imagen de referencia, el cual establece las siguientes relaciones entre las entidades Categoría, Producto y Venta.

Relación Categoría - Producto (1 a muchos)

- Una categoría puede tener múltiples productos.
- Esta relación está representada mediante el campo CódigoCategoría en la tabla Producto, que actúa como clave foránea hacia la tabla Categoría.

Relación Producto - Venta (1 a muchos)

- Un producto puede tener muchas ventas.
- La tabla Venta contiene el campo CódigoProducto como clave foránea hacia la tabla Producto, permitiendo asociar cada venta con el producto correspondiente.

Uso de CHAR(36) para claves primarias

Se decidió utilizar el tipo de dato CHAR(36) para los campos clave como CódigoCategoría, CódigoProducto y CódigoVenta, ya que esto permite almacenar identificadores UUID.

Se consideraron las siguientes ventajas:

- Unicidad global: evita colisiones incluso en entornos distribuidos o sistemas que se integran entre sí.
- Seguridad: no expone la cantidad real de registros como lo hacen los ID incrementales.

Tipos de datos utilizados

Tabla Categoría

- CodigoCategoría (CHAR(36)): UUID como clave primaria.
- Nombre (VARCHAR(100)): texto representativo del nombre de la categoría.

Tabla Producto

- CodigoProducto (CHAR(36)): UUID como clave primaria.
- Nombre (VARCHAR(100)): nombre descriptivo del producto.
- Precio (DECIMAL(10,2)): se agregó esta columna para registrar el precio base del producto en el catálogo, ya que este valor puede ser útil para estimaciones, reportes o validaciones antes de realizar una venta.
- CodigoCategoría (CHAR(36)): clave foránea a Categoría, lo cual permite organizar productos por categoría.

Tabla Venta

- CodigoVenta (CHAR(36)): UUID como clave primaria.
- Fecha (DATETIME): registra cuándo se realizó la venta.
- Cantidad (INT): se incluyó para permitir registrar múltiples unidades vendidas en una sola transacción.
- PrecioUnitario (DECIMAL(10,2)): guarda el precio con el que se vendió una unidad específica, ya que puede variar del precio base del producto.
- Total (DECIMAL(12,2)): se agregó para almacenar directamente el monto total de la venta). Esto mejora el rendimiento en reportes al evitar cálculos repetitivos.
- CodigoProducto (CHAR(36)): clave foránea que vincula la venta con el producto vendido.

Convención en nombres

La base de datos (sales) y el usuario (user_sales) fueron definidos en inglés por los siguientes motivos:

- Estándar internacional: el uso del inglés es común en ambientes de desarrollo profesional, facilitando la colaboración en equipos multiculturales.
- Consistencia con otros sistemas: muchas herramientas, ORMs y convenciones de código emplean nombres en inglés por defecto.

Creación de usuario y asignación de permisos

En lugar de utilizar el usuario root del sistema de base de datos, se creó un usuario específico llamado user_sales con contraseña definida.

Razones por las que se creó un usuario distinto al root:

- Buenas prácticas de seguridad: evita usar root para operaciones de desarrollo o producción, minimizando riesgos.
- Principio de privilegios mínimos: se le otorgan permisos solo sobre la base de datos sales, reduciendo la posibilidad de errores o accesos indebidos a otras bases.
- Mejor control y auditoría: en entornos reales, cada aplicación o servicio debe tener su propio usuario para facilitar monitoreo y gestión de accesos.

Consulta SQL

Uno de los requisitos del ejercicio era realizar una consulta que permitiera obtener el nombre de la categoría del producto asociado a la última venta realizada, de acuerdo con la fecha.

Para cumplir con esto, se desarrolló la siguiente consulta SQL, la cual se apoya en las relaciones entre las tablas Venta, Producto y Categoría previamente explicadas:

```
-- consulta que obtiene el nombre de la categoría del producto de la
última venta realizada según la fecha

SELECT c.Nombre -- solo selecciona el nombre de la categoría

FROM Venta v

-- une la tabla Producto, empareja por el código del producto vendido
INNER JOIN Producto p ON v.CodigoProducto = p.CodigoProducto

-- une la tabla Categoría, empareja por el código de categoría del
producto
INNER JOIN Categoría c ON p.CodigoCategoría = c.CodigoCategoría

-- ordena descendente por fecha
ORDER BY v.Fecha DESC

-- limita el resultado a solo 1 fila, osea, la venta mas reciente
LIMIT 1;
```