

Clase 8: Material Complementario

Sitio: [Centro de E-Learning - UTN.BA](#)
Curso: Curso de Backend Developer - Turno
Libro: Noche
Libro: Clase 8: Material Complementario

Imprimido
por:
Día:
Nelson Brian Avila Solano
Tuesday, 9 de December de 2025,
19:46

Tabla de contenidos

1. MySQL-DB Relaciona

1.1. MySQL-DB Relaciona

1. MySQL-DB Relaciona

Temario

- Resumen introductorio de Base de Datos Relacional
- Paso a paso para comenzar con sql:
- Base de datos relacionales
- ACID

1.1. MySQL-DB Relaciona

Resumen introductorio de Base de Datos Relacional

Una base de datos relacional es un sistema que administra, organiza y estructura los datos de manera eficiente.

Permite realizar las operaciones básicas sobre los datos: crear, leer, editar y borrar (CRUD).

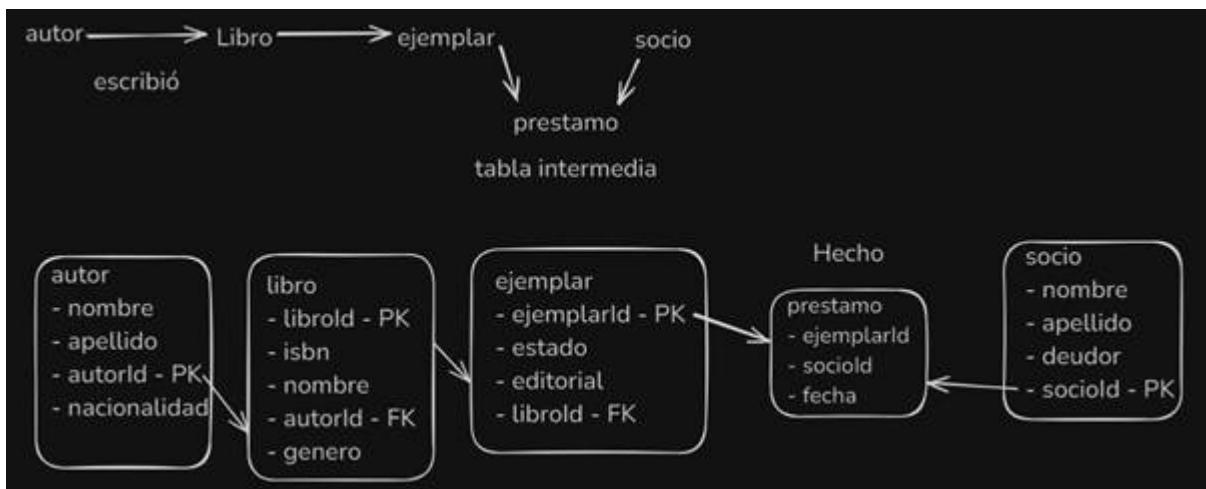
Funciones de una base de datos relacional:

Administración de los datos: Permite almacenar, consultar y modificar la información.

Manejo de estructuras: Gestiona las estructuras que organizan la información, como tablas y esquemas.

Jerarquía de una base de datos relacional:

- **Registro (Filas)**: Es la unidad básica de los datos, cada registro representa una instancia de los datos dentro de una tabla.
- **Tabla (Conjunto de Columnas)**: Una colección de registros organizados en filas y columnas, donde cada columna define un tipo de dato específico.
- **Esquema**: Nivel jerárquico mayor que agrupa varias tablas y define la estructura de la base de datos en su conjunto.



Consultas Posibles en el Sistema de Base de Datos Relacional para nuestro esquema

1. ¿Cuántos libros tiene X autor?

2. ¿Qué géneros escribe X autor?

3. ¿Cuáles son los 10 géneros más comunes por nacionalidad de autor?
4. ¿Qué ejemplares tenemos en mayor cantidad?
5. ¿Qué ejemplares tenemos en menor cantidad?
6. ¿Quién es el socio más antiguo?
7. ¿Quién es el socio que más libros alquila?
8. ¿Cuáles son los géneros favoritos de X socio?
9. ¿Cuáles son los autores favoritos de X socio?
10. ¿En qué fechas del año se prestan más libros?
11. ¿En qué fechas del año se prestan menos libros?
12. ¿Cuáles son los ejemplares más prestados?
13. ¿Cuáles son los autores favoritos de X socio?

Identificadores en Bases de Datos Relacionales

En una base de datos relacional, todo registro debe poseer un identificador único que sea irrepetible.

Este identificador juega un papel crucial en la estructura y relación entre tablas.

Tipos de identificadores:

- **Primary Key** (Clave Primaria): Es el identificador único de un registro dentro de su propia tabla.

Ningún otro registro en esa tabla puede tener el mismo valor.

- **Foreign Key** (Clave Foránea): Es un identificador que aparece en una tabla ajena, y sirve para establecer una relación con la Primary Key de otra tabla.

- **Primary Key compuesta**: En algunos casos, como en una tabla de préstamos, la clave primaria puede estar compuesta por dos claves foráneas (FK).

Esto sucede cuando la combinación de ambos identificadores es lo que garantiza la unicidad del registro.

Herramientas y Tecnologías Utilizadas:

- **Sistema de Administración de Bases de Datos (DBMS)**:

- [DBeaver](#)

- **Servidor**:

- [XAMPP](#)

- Bases de Datos Relacionales:

- MySQL
- SQL Server
- PostgreSQL

Paso a paso para comenzar con sql:

Paso 1: Comprender qué es SQL

- SQL (Structured Query Language) es un lenguaje utilizado para **acceder** y **manipular** bases de datos relacionales.
- Es un lenguaje **declarativo** que permite realizar diversas operaciones en bases de datos.
- Tiene tres grandes grupos de comandos: **DDL** (Data Definition Language), **DML** (Data Manipulation Language) y **DCL** (Data Control Language). Al ser declarativo le digo que quiero obtener, no le digo como debe obtenerlo.

Paso 2: Tipos de Datos en SQL

- SQL admite varios tipos de datos, como **VARCHAR** (para cadenas de texto), **INT** (para valores enteros) y **DATE/TIME** (para fechas y horas), **boolean**.

Paso 3: Consultas en SQL

- Las consultas en SQL se utilizan para recuperar datos de la base de datos.
- Se pueden utilizar **operadores** como = (igual a), > (mayor que), < (menor que), LIKE (para buscar patrones de caracteres) y otros para filtrar resultados.
- También se pueden ordenar los resultados con **ORDER BY** y limitar la cantidad de resultados con **LIMIT**.

Paso 4: Manipulación de Datos en SQL

- Para agregar datos a una tabla, se utiliza el comando **INSERT**.
- Para eliminar datos de una tabla, se utiliza el comando **DELETE**.
- Para modificar datos en una tabla, se utiliza el comando **UPDATE**.

Paso 5: Instalación de MySQL y herramientas

- Para practicar SQL localmente, necesitarás instalar MySQL y un cliente o gestor de bases de datos como phpMyAdmin.

- Puedes descargar WampServer (para Windows) o MAMP (para Mac) para instalar MySQL y otras herramientas.
- Una vez instalado, puedes acceder a phpMyAdmin para ejecutar tus consultas SQL.

Paso 6: Práctica y Referencias

- Practica escribiendo y ejecutando consultas SQL en tu entorno local.
- Utiliza recursos como tutoriales y documentación para ampliar tus conocimientos sobre SQL.

Base de datos relacionales

1- Estructura está basada en tablas

filas (Registro) y columnas (atributos)

2- Relaciones entre tablas

A través de las claves

PK-primary key: Identificador único de cada fila.

FK-foreign key: Se refiere a la clave primaria de otra tabla.

Esto crea una relación entre dos tablas.

3- Integridad referencial

Asegurar que las relaciones entre tablas sean consistentes.

Si en una tabla hay una FK, entonces en otra esta es PK.

Si borro el registro que posee la PK, se debe definir cómo afecta esto al dato que tiene esa clave foránea.
opciones: eliminar o actualizar en cascada.

4- Usa lenguaje SQL (structured query language)

Consultar datos, insertar, actualizar, eliminar, crear tablas y modificar estructura de las tablas.

5- Normalización

Proceso que busca:

evitar la duplicación de datos (redundancia) y mejorar la consistencia.

Definimos la dependencia entre los datos.

6- Operaciones relacionales

- Join (Unión): Combinar datos de dos o más tablas.
- Proyección: Seleccionar columnas específicas de las tablas.
- Seleccionar: Seleccionar filas específicas de una tabla en base a condiciones.

7- Independencia de los datos

Los datos están separados de los programas que los utilizan.

Podrías modificar la estructura de la db sin tener que modificar las aplicaciones que acceden a estos.

ACID

Propiedades que garantizan fiabilidad e integridad de las transacciones en una db.

Cada letra de ACID corresponde a una propiedad que las bd relacionales deben cumplir.

A: Atomicidad-Atomicity:

Todo o nada. Una transacción se completa o no se realiza.

Si una parte de la transacción falla, todo el proceso debe revertirse (rollback).

Rollback: La base vuelve al estado en que estaba antes de que la transacción comenzará.

C: Consistencia-Consistency:

Asegura que una transacción lleve a la db de un estado valido a otro estado valido.

Respeta las reglas, restricciones y triggers definidos.

Después de toda transacción la db va a estar en un estado coherente, donde sus reglas siguen cumpliéndose.

I: Isolation-Aislamiento:

Las transacciones se realizan de forma independiente unas de otras.

De forma secuencial aunque se ejecuten de forma concurrente.

Los cambios de una transacción no están disponibles para otras transacciones hasta que esta no se haya completado. Evita inconsistencias temporales.

D: Durabilidad-Durability:

Una vez que una transacción fue confirmada (commit) los cambios son permanentes incluso si ocurre un fallo.

Fallos: corte de energía o caídas de servidor.

Usando logs o registros podemos almacenar y recuperar las transacciones confirmadas.

Persistencia - Guardado y almacenamiento de los datos.

Ejemplo de creación de base de datos:

Paso 1: Creación de la Base de Datos:

```
CREATE DATABASE DiplomaturaProgramacionWeb;
```

Paso 2: Creación de la Tabla:

```
CREATE TABLE Alumnos (
    ID_alumno INT,
    Nombre VARCHAR(50),
    Apellido VARCHAR(50)
    ADD PRIMARY KEY (ID_alumno),
);

```

Paso 3: Inserción de Datos en la Tabla:

```
INSERT INTO Alumnos (ID_Empleado, Nombre, Apellido)
VALUES
(1, 'Laura', 'Conti'),
(2, 'Juana', 'Diaz'),
(3, 'Carlos', 'García');
```

Paso 4: Consulta de Datos en la Tabla:

```
SELECT * FROM Alumnos;
```

Paso 5: Modificación de la Tabla (Aregar una Columna):

```
ALTER TABLE Alumnos
```

ADD Sueldo INT;

Paso 6: Modificación de Datos:

UPDATE Alumnos

SET Sueldo = 25000

WHERE Nombre = 'Laura';

Paso 7: Consulta de Datos Actualizados:

SELECT * FROM Alumnos;

Paso 8: Modificación de la Tabla (Eliminar una Columna):

ALTER TABLE Alumnos

DROP COLUMN Sueldo;

Paso 9: Eliminación de la Tabla:

DROP TABLE Alumnos;