Clase 8: MySQL

Sitio: Centro de E-Learning - UTN.BA
Curso: Curso de Backend Developer - Turno Noche
Libro: Clase 8: MySQL
Imprimido por: Nilo Crespi
Día: Friday, 23 de January de 2026, 10:19

Descripción

Objetivos

- Comprender el concepto y la importancia de las llaves foráneas en las bases de datos relacionales.
- Aprender a crear y gestionar llaves foráneas en phpMyAdmin.
- Aplicar llaves foráneas en consultas JOIN y mantener la integridad referencial.
- Realizar consultas avanzadas utilizando JOIN, subconsultas y UNION.

Tabla de contenidos

1. Introducción

En esta clase exploraremos el uso de llaves foráneas y consultas avanzadas en MySQL, conceptos fundamentales para gestionar bases de datos relacionales de manera eficiente. Aprenderemos cómo las llaves foráneas permiten establecer vínculos entre tablas, garantizando la integridad referencial y facilitando la recuperación de datos mediante consultas más estructuradas. También veremos cómo definir y gestionar estas llaves en phpMyAdmin, y cómo configurarlas con opciones como ON DELETE y ON UPDATE para controlar el comportamiento de las relaciones entre tablas.

Además, nos adentraremos en la ejecución de consultas avanzadas con SQL, utilizando JOIN para combinar información de múltiples tablas, subconsultas para realizar búsquedas más específicas y UNION para unificar resultados de diferentes consultas. A través de ejemplos prácticos, aplicaremos estos conocimientos en la gestión de una base de datos para una empresa de transportes, permitiéndonos diseñar estructuras de datos más funcionales y optimizadas para la toma de decisiones.

2. Llaves foráneas y consultas avanzadas.

SQL

El DML (Data Manipulation Language) o Lenguaje de Manipulación de Datos es la parte de SQL dedicada a la manipulación de los datos. Las sentencias DML son las siguientes:

- SELECT: se utiliza para realizar consultas y extraer información de la base de datos.
- INSERT: se utiliza para insertar registros en las tablas de la base de datos.
- UPDATE: se utiliza para actualizar los registros de una tabla.
- DELETE: se utiliza para eliminar registros de una tabla.

¿Qué es una llave foránea?

Una columna en una tabla que establece una relación con la columna de una tabla diferente.

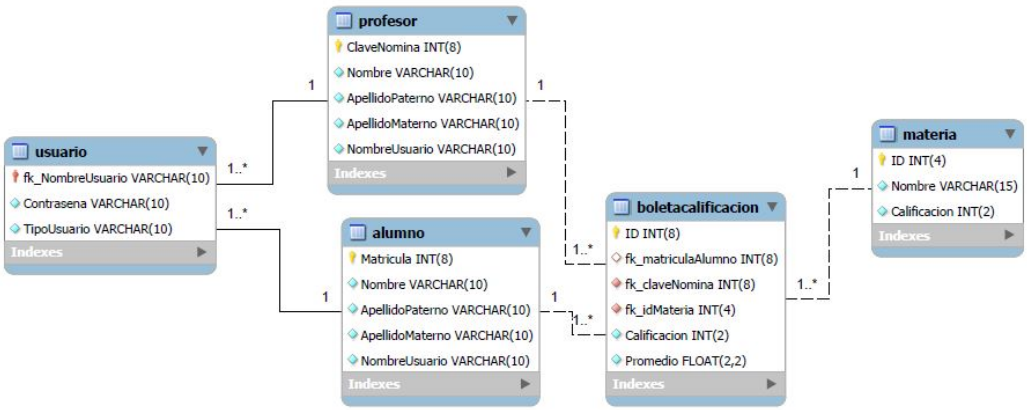
Mantiene la integridad referencial entre tablas.

Relación entre llaves

Llave primaria: Identifica de manera única cada registro en una tabla.

Llave foránea: Enlaza registros de una tabla con registros de otra tabla.

Relación con clave foranea




```
ALTER TABLE `alumnos`
  ADD CONSTRAINT `alumnos_casas` FOREIGN KEY
(`fk_id_casa`) REFERENCES `casas` (`id_casa`);
```

Configuración de las llaves foráneas

ON DELETE y ON UPDATE:

- **ON DELETE: Define la acción que se debe tomar cuando se elimina un registro en la tabla referenciada.**
  - **CASCADE: Si se elimina un registro en la tabla principal, se eliminarán automáticamente los registros en la tabla secundaria que hacen referencia al registro eliminado.**
  - **SET NULL: Si se elimina un registro en la tabla principal, se establecerán en NULL los valores de las columnas en la tabla secundaria que hacen referencia al registro eliminado.**
  - **RESTRICT: No permite la eliminación de un registro en la tabla principal si existen registros en la tabla secundaria que hacen referencia al registro principal.**
  - **NO ACTION: Similar a RESTRICT, pero la verificación se realiza al final de la transacción, no en el momento de la eliminación.**

- **ON UPDATE: Define la acción que se debe tomar cuando se actualiza una clave primaria en la tabla referenciada.**
  - **CASCADE: Si se actualiza una clave primaria en la tabla principal, se actualizarán automáticamente las claves foráneas correspondientes en la tabla secundaria.**
  - **SET NULL: Si se actualiza una clave primaria en la tabla principal, se establecerán en NULL los valores de las columnas en la tabla secundaria que hacen referencia a la clave primaria actualizada.**
  - **RESTRICT: No permite la actualización de una clave primaria en la tabla principal si existen registros en la tabla secundaria que hacen referencia a la clave primaria.**
  - **NO ACTION: Similar a RESTRICT, pero la verificación se realiza al final de la transacción.**

¡Para practicar!

- **Creá una segunda tabla y definí una llave foránea que relacione ambas tablas.**

3. Sentencias

Insert

La INSERT INTO declaración se utiliza para insertar nuevos registros en una tabla.

Es posible escribir la INSERT INTO declaración de dos maneras:

1. Especificando tanto los nombres de las columnas como los valores que se insertarán:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

2. Si está agregando valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, asegúrese de que el orden de los valores sea el mismo que el de las columnas de la tabla.

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

UPDATE

UPDATE se utiliza para modificar los registros existentes en una tabla.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;


UPDATE Customers
SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'
WHERE CustomerID = 1;
```

UPDATE múltiples registros

Es la WHERE cláusula que determina cuántos registros se actualizarán.

```
UPDATE Customers
SET PostalCode = 00000
WHERE Country = 'Mexico';
```

DELETE

La DELETE instrucción se utiliza para eliminar registros existentes en una tabla.

```
DELETE FROM table_name WHERE condition;
```

4. Consultas

JOIN

La sentencia JOIN nos permite unir dos o más tablas en una consulta, utilizando una condición específica. Por ejemplo, si tenemos una tabla "clientes" y una tabla "pedidos" y queremos obtener los nombres de los clientes y el número de pedidos que han realizado, podemos utilizar la siguiente sentencia:

```
SELECT clientes.nombre, COUNT(pedidos.id) as total_pedidos
FROM clientes
JOIN pedidos ON clientes.id = pedidos.id_cliente
GROUP BY clientes.id;
```

SUBQUERY

La sentencia SUBQUERY nos permite utilizar una consulta como si fuera una tabla en otra consulta. Por ejemplo, si queremos obtener los nombres de los clientes que han realizado más de 3 pedidos, podemos utilizar la siguiente sentencia:

```
SELECT nombre
FROM clientes
WHERE id IN (
  SELECT id_cliente
  FROM pedidos
  GROUP BY id_cliente
  HAVING COUNT(*) > 3
);
```

UNIÓN

La sentencia UNION nos permite unir los resultados de dos o más consultas en una sola tabla. Por ejemplo, si tenemos una tabla "ventas_2019" y una tabla "ventas_2020" y queremos obtener el total de ventas de ambos años, podemos utilizar la siguiente sentencia:

```
SELECT '2019' as año, SUM(total) as total_ventas
FROM ventas_2019
UNION
SELECT '2020' as año, SUM(total) as total_ventas
FROM ventas_2020;
```

Base de datos de ejemplo

Base de datos. Caso Instituto (ejemplo clase)

5. Espacio de práctica

Espacio de práctica

Objetivo: Diseñar y gestionar una base de datos para una empresa de transportes que incluye camiones, camioneros, paquetes, provincias y clientes. Implementar llaves foráneas, realizar consultas avanzadas y configurar restricciones.

1. **Diseñar la base de datos para gestionar la información de una empresa de transportes que incluye:**

- **Camioneros que distribuyen paquetes.**
- **Paquetes que llegan a provincias.**
- **Camiones que son conducidos por camioneros.**
- **Productos vendidos a clientes.**

2. Implementación en MySQL:

- **Crear las tablas correspondientes con llaves primarias y foráneas.**

Configurar las opciones de cascada para las llaves foráneas.

Tabla a crear

- Camioneros
- Camiones
- paquetes
- Provincias
- Clientes

Realizar las siguientes consultas SQL usando la base de datos creada:

- Consulta 1: Obtén todos los paquetes junto con el nombre de la provincia a la que llegan.
- Consulta 2: Muestra todos los paquetes distribuidos por el camionero con DNI '12345678A'.
- Consulta 3: Listar todos los camiones conducidos por el camionero con DNI '12345678A' en un rango de fechas.
- Consulta 4: Obtén la lista de clientes junto con los productos que han comprado.
- Consulta 5: Muestra el total de paquetes distribuidos por cada camionero.