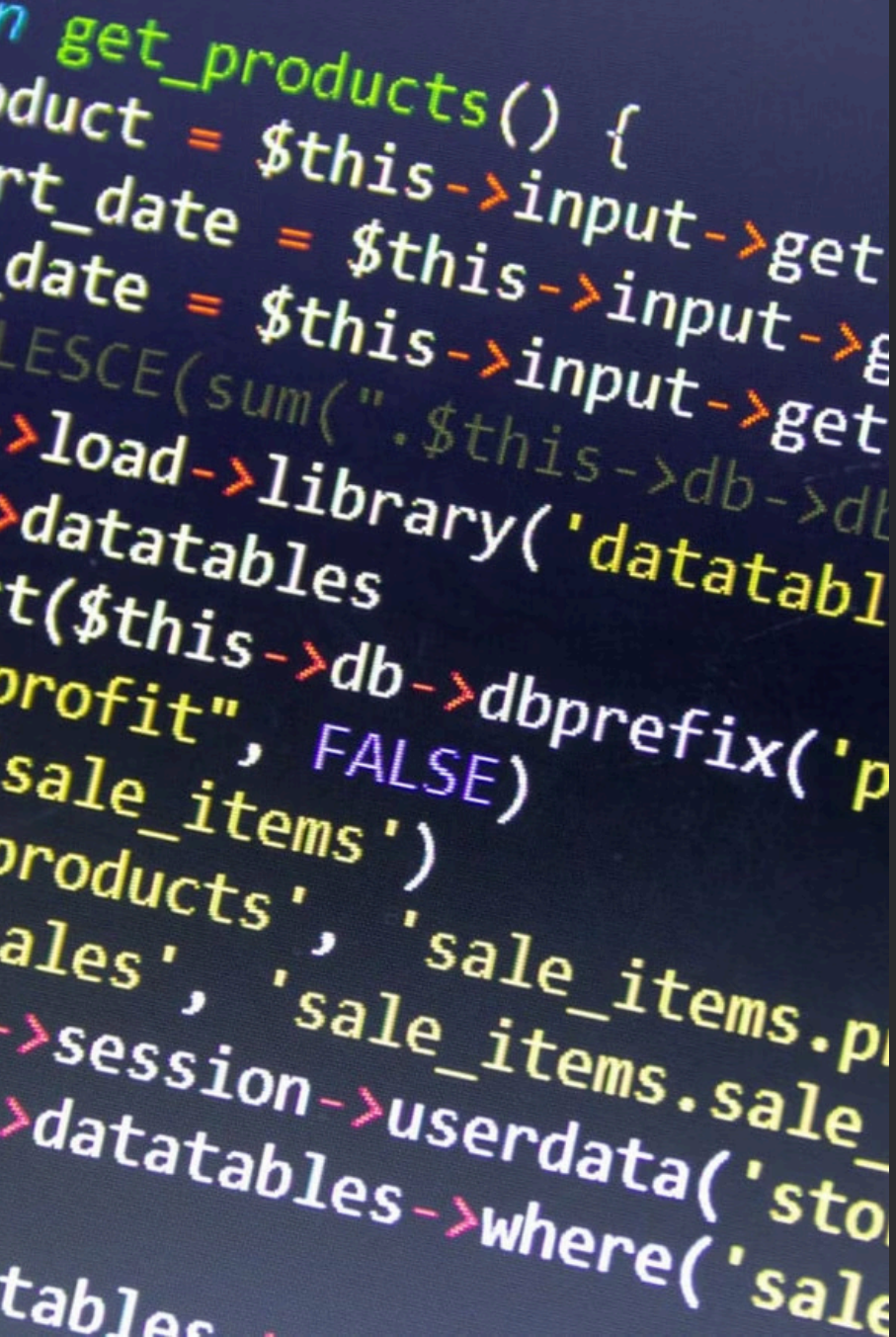

Operaciones de Manipulación de Bases de Datos en PHP con MySQL

Esta presentación explorará las operaciones esenciales para la manipulación de datos en bases de datos MySQL utilizando PHP. Cubriremos desde la selección de registros hasta la inserción, modificación y eliminación de datos, incluyendo aspectos cruciales de seguridad y manejo de transacciones. Prepárense para un recorrido práctico y detallado.

Preparación del Entorno: MySQL y PHP

Instalación de MySQL

Antes de comenzar, asegúrese de tener MySQL instalado y configurado. Puede descargar MySQL Server desde su sitio web oficial y seguir las instrucciones de instalación para su sistema operativo.

Instalación de PHP

PHP también debe estar instalado y configurado en su servidor. Puede descargar la versión más reciente de PHP desde su sitio web oficial y seguir las instrucciones de instalación. Asegúrese de habilitar la extensión MySQLi en su archivo php.ini.

Conexión PHP a MySQL

Una vez instalados, configure la conexión entre PHP y MySQL. Use las funciones mysqli_connect() para establecer la conexión, proporcionando el host, usuario, contraseña y nombre de la base de datos.

Selección de Registros (SELECT): Sintaxis y Ejemplos

Sintaxis Básica

La instrucción **SELECT** se utiliza para recuperar datos de una o más tablas en una base de datos. La sintaxis básica es: *SELECT columna1, columna2 FROM nombre_tabla WHERE condición;*

Ejemplo Simple

Para seleccionar todos los registros de una tabla llamada "usuarios", se utiliza la siguiente consulta: *SELECT * FROM usuarios;* Esto devolverá todas las columnas y filas de la tabla.

Condiciones (WHERE)

La cláusula **WHERE** permite filtrar los registros según una condición específica. Por ejemplo, para seleccionar solo los usuarios cuyo nombre sea "Juan": SELECT * FROM usuarios WHERE nombre = 'Juan';

Eliminación de Registros Específicos (DELETE con WHERE)

1

Sintaxis DELETE

La instrucción **DELETE** se utiliza para eliminar registros de una tabla. Es crucial usar la cláusula **WHERE** para especificar qué registros eliminar. *DELETE FROM nombre_tabla WHERE condición;*

2

Ejemplo con WHERE

Para eliminar un usuario específico con ID = 5, se utiliza la siguiente consulta: *DELETE FROM usuarios WHERE id = 5;* Asegúrese de verificar la condición antes de ejecutar la consulta.

3

Precauciones

La omisión de la cláusula **WHERE** en una instrucción **DELETE** resultará en la eliminación de todos los registros de la tabla. Por lo tanto, siempre revise su consulta antes de ejecutarla.

Eliminación de Todos los Registros (TRUNCATE vs DELETE)

DELETE

DELETE elimina registros uno por uno, registrando cada operación. Permite usar **WHERE** para condiciones específicas, y es más lento que **TRUNCATE**.

TRUNCATE

TRUNCATE elimina todos los registros rápidamente, reiniciando el contador *AUTO_INCREMENT*. No registra operaciones individuales, siendo más eficiente pero sin opción de **WHERE**.

Consideraciones

Use **TRUNCATE** cuando necesite vaciar la tabla rápidamente y reiniciar *AUTO_INCREMENT*. Use **DELETE** para eliminación selectiva o cuando necesite registrar cada operación.

Modificación de Registros (UPDATE): Ejemplos Prácticos

Sintaxis UPDATE

La instrucción **UPDATE** se utiliza para modificar registros existentes en una tabla. La sintaxis básica es: *UPDATE nombre_tabla SET columna1 = valor1, columna2 = valor2 WHERE condición;*

Ejemplo

Para actualizar el correo electrónico de un usuario con ID = 3: *UPDATE usuarios SET email = 'nuevo_email@ejemplo.com' WHERE id = 3;*

Múltiples Columnas

Puede modificar múltiples columnas en una sola consulta: UPDATE usuarios SET nombre = 'Nuevo Nombre', email = 'nuevo_email@ejemplo.com' WHERE id = 3;

Inserción de Nuevos Registros (INSERT): Varias formas

1 Sintaxis Básica

La instrucción **INSERT** se utiliza para agregar nuevos registros a una tabla. La sintaxis básica es: *INSERT INTO nombre_tabla (columna1, columna2) VALUES (valor1, valor2);*

2 Insertar Todos los Campos

Si va a insertar valores en todas las columnas, puede omitir la lista de columnas: *INSERT INTO usuarios VALUES (null, 'Nombre', 'email@ejemplo.com');* (null para *AUTO_INCREMENT*).

3 Múltiples Registros

Puede insertar múltiples registros en una sola consulta: INSERT INTO usuarios (nombre, email) VALUES ('Nombre1', 'email1@ejemplo.com'), ('Nombre2', 'email2@ejemplo.com');

Seguridad: Prevención de Inyecciones SQL

1

Entender el Riesgo

Las inyecciones SQL ocurren cuando datos maliciosos son insertados en consultas SQL, permitiendo a atacantes ejecutar código no autorizado.

2

Sanitizar Entradas

Utilice funciones como **mysqli_real_escape_string()** para escapar caracteres especiales en las entradas del usuario antes de incluirlas en consultas SQL.

3

Consultas Preparadas

Las consultas preparadas (prepared statements) separan la lógica de la consulta de los datos, previniendo inyecciones SQL de manera efectiva.

Transacciones: Asegurando la Integridad de los Datos

ACID

ACID

Las transacciones deben ser **Atómicas**, **Consistentes**, **Aisladas** y **Duraderas** (ACID) para asegurar la integridad de los datos.

START

START

Inicie una transacción con **mysqli_begin_transaction($conexion)**. Esto marca el comienzo de un bloque de operaciones que deben ser tratadas como una unidad.

COMMIT

COMMIT

Si todas las operaciones son exitosas, confirme los cambios con **mysqli_commit($conexion)**. Esto guarda permanentemente los cambios en la base de datos.

ROLLBACK

ROLLBACK

Si alguna operación falla, revierta todos los cambios con **mysqli_rollback($conexion)**. Esto asegura que la base de datos permanezca en un estado consistente.

Fuentes:

1. Instalación de MySQL y PHP

- **Instalación de MySQL** (Documentación oficial de MySQL)
- **Instalación de PHP** (Documentación oficial de PHP)
- **Guía de instalación de MySQL y PHP en W3Schools**

2. Conexión entre PHP y MySQL

- **Uso de** `mysqli_connect()`
- **Ejemplo de conexión en W3Schools**

3. Selección de Registros (`SELECT`)

- **Documentación oficial de MySQL sobre** `SELECT`
- **Ejemplos de** `SELECT` **en PHP**

4. Eliminación de Registros (`DELETE` y `TRUNCATE`)

- **Diferencias entre** `DELETE` **y** `TRUNCATE` **en MySQL**
- **Ejemplos de** `DELETE` **en PHP**

5. Modificación de Registros (`UPDATE`)

- **Guía de** `UPDATE` **en MySQL**
- **Ejemplo de** `UPDATE` **en PHP**

6. Inserción de Registros (`INSERT`)

- **Documentación sobre** `INSERT` **en MySQL**
- **Ejemplo de** `INSERT` **en PHP**

7. Seguridad: Prevención de Inyecciones SQL

- **Buenas prácticas para evitar inyecciones SQL**
- **Uso de consultas preparadas en PHP**

8. Transacciones en MySQL y PHP

- **Concepto ACID en transacciones**
- **Uso de** `mysqli_begin_transaction()`, `COMMIT` **y** `ROLLBACK`