



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ Sentencias para la manipulación de datos y transaccionalidad

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Realizar consultas con funciones de agrupación diferenciando los distintos métodos
- ✓ Analizar los distintos tipos de JOIN que existen y sus funciones
- ✓ Realizar consultas SQL utilizando JOIN

LEARNING PATHWAY

3.

Start! 🏁

Fundamentos de bases de datos relacionales

El objetivo de este tema es brindar una descripción básica de los conceptos asociados a las bases de datos relacionales que te permitirán desde tu ordenador instalar las herramientas necesarias para establecer una conexión a BD y comenzar a manipular dichos datos.

Data Manipulation Language (DML)

Organizando una Tabla

Agregar información a una tabla

Agregar nuevos usuarios utilizando INSERT

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Conocer los componentes principales de Data Manipulation Language (DML)



Insertar información en una tabla utilizando DML





Rompehielo 🧊

Contexto: 🙌

Imaginemos que queremos insertar información en una base de datos de una plataforma de películas online. Analicemos las siguientes preguntas y respondamos juntos.



Consigna: 📝

- ¿Cómo se estructura el comando INSERT en SQL?
- ¿Qué datos necesitas brindarle a la query para que se genere el registro esperado?
- Reflexiona sobre cómo podrías evitar insertar correos electrónicos duplicados en la base de datos



➤ Data Manipulation Language (DML)



Data Manipulation Language (DML)

¿Qué es DML y para qué se utiliza?:

Es una parte esencial del lenguaje SQL (Structured Query Language) que se utiliza para interactuar con los datos almacenados en una base de datos. Su **función principal es permitir la manipulación, inserción, actualización y eliminación de datos** dentro de las tablas de una base de datos.

Permite agregar nuevos datos, actualizar información existente y eliminar registros innecesarios. Esto asegura que la información almacenada en la base de datos esté siempre actualizada y sea coherente con la realidad que representa.





Data Manipulation Language (DML)



El DML se compone de varios comandos que permiten realizar acciones específicas sobre los datos:

INSERT: Se utiliza para agregar nuevos registros (filas) a una tabla. Puedes especificar los valores que deseas insertar en cada columna o utilizar una subconsulta para insertar datos desde otra fuente.

UPDATE: Permite modificar los valores de una o varias columnas en una o varias filas existentes de una tabla. Puedes utilizar una cláusula WHERE para especificar qué filas deben ser actualizadas.



Data Manipulation Language (DML)

El DML se compone de varios comandos que permiten realizar acciones específicas sobre los datos:

DELETE: Se utiliza para eliminar filas de una tabla. Al igual que con UPDATE, puedes utilizar una cláusula WHERE para definir qué filas se eliminarán.

SELECT: Aunque a menudo se asocia con la recuperación de datos, SELECT también es parte del DML. Se utiliza para recuperar información de una o varias tablas de la base de datos.

Aunque SELECT no modifica los datos en sí, es fundamental para la manipulación de datos al proporcionar los resultados que se pueden modificar utilizando INSERT, UPDATE y DELETE.

➤ Agregar información en una tabla

Agregar información en una tabla

¿Cómo luce una sentencia INSERT en SQL?:

```
INSERT INTO nombre_de_tabla (columna1, columna2, columna3, ...)
VALUES (valor1, valor2, valor3, ...);
```

- **nombre_de_tabla** es el nombre de la tabla a la que deseas insertar los datos.
- **columna1, etc.** son los nombres de las columnas a las que deseas insertar datos.
- **valor1, etc.** son los valores que deseas insertar en las respectivas columnas.



Agregar información en una tabla

Aquí vemos un ejemplo más concreto

Supongamos que tienes una tabla llamada "usuarios" con las columnas "id", "nombre" y "correo". Quieres agregar un nuevo usuario con ID 1, nombre "Juan" y correo "juan@example.com".

La sentencia INSERT sería:

```
INSERT INTO usuarios (nombre, correo)
VALUES ('Juan', 'juan@example.com');
```

Si la columna "id" es autoincremental puedes omitirla en la sentencia INSERT.



Agregar información en una tabla

✕ ¿Cómo indicar que un ID debe ser autogenerado e incremental?

Para indicar que un ID debe ser autogenerado al realizar una operación de inserción (INSERT) en SQL, generalmente se utiliza una característica llamada **"autoincrement"** o **"identity"** dependiendo del sistema de gestión de bases de datos que estés utilizando.

Esta característica permite que la base de datos genere automáticamente un valor único para el ID cada vez que se inserta un nuevo registro.

✕ **No necesitas proporcionar un valor específico para la columna "id"** en la sentencia INSERT, ya que la base de datos manejará la generación automática.



Agregar información en una tabla

Aquí tienes ejemplos de cómo hacerlo MySQL

En MySQL, puedes usar el atributo AUTO_INCREMENT en la definición de la columna para indicar que el ID debe ser autogenerado. La columna debe ser una clave primaria.

```
CREATE TABLE MiTabla (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(50),  
    ...  
);  
  
-- Al insertar, no es necesario proporcionar un valor para "id"  
INSERT INTO MiTabla (nombre, ...)  
VALUES ('Ejemplo', ...);
```


Agregar información en una tabla

Aquí tienes ejemplos de cómo hacerlo SQL Server:

En SQL Server, puedes usar la propiedad IDENTITY en la definición de la columna para lograr el mismo efecto.

```
CREATE TABLE MiTabla (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    nombre VARCHAR(50),  
    ...  
);
```

```
-- Al insertar, no es necesario proporcionar un valor para "id"  
INSERT INTO MiTabla (nombre, ...)  
VALUES ('Ejemplo', ...);
```

Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



LIVE CODING

Ejemplo en vivo

Agregar nuevos usuarios utilizando INSERT:

*Agregar nuevos usuarios en una una tabla llamada "Usuarios".
Antes de comenzar, debes crear una conexión a una base de datos llamada "Alke Wallet".*

1. *Crear 3 tablas llamadas: Usuario, Transacción y Moneda.*

2. *Crear la Tabla "Usuarios":*

- *user_id (clave primaria)*
- *nombre*
- *correo electrónico*
- *contraseña*
- *saldo.*
- *fecha de creación*

3. *Crear la Tabla "Transaccion":*

- *transaction_id (Primary Key)*
- *sender_user_id (Foreign Key referenciando a User)*
- *receiver_user_id (Foreign Key referenciando a User)*
- *valor*
- *transaction_date.*

LIVE CODING

Ejemplo en vivo

Agregar nuevos usuarios utilizando INSERT:

*Agregar nuevos usuarios en una una tabla llamada "Usuarios".
Antes de comenzar, debes crear una conexión a una base de datos llamada "Alke Wallet".*

1. Crear la Tabla "Mondea":

currency_id (Primary Key)

currency_name

currency_symbol

2. Insertar al menos 3

usuarios y 3 transacciones.

3. Consultar usuarios creados

○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Diferenciar los componentes principales de Data Manipulation Language (DML)**
- ✓ **Insertar datos en una tabla**
- ✓ **Crear ID autogenerados**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 📌 📌 📌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. Material 1 (*Lectura de la Lección 3: Sentencias para la manipulación de datos y transaccionalidad, páginas 1-3*)
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

Time-out!

🕒 5 min.

