



“SQL Server Management Studio”

Tarea procesual hito 3

Estudiante: Kevin mamani mamani

Asignatura: BASE DE DATOS I

Carrera: Ingeniería de sistemas

Paralelo: (4)

Docente: Lic. William Barra Paredes

Fecha: 14/9/2023

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or data paths, with lines connecting to small circles.

**QUE ES DDL Y DML, ADICIONALMENTE MUESTRA
UN EJEMPLO EN LA BASE DE DATOS UNIFRANZITOS.**

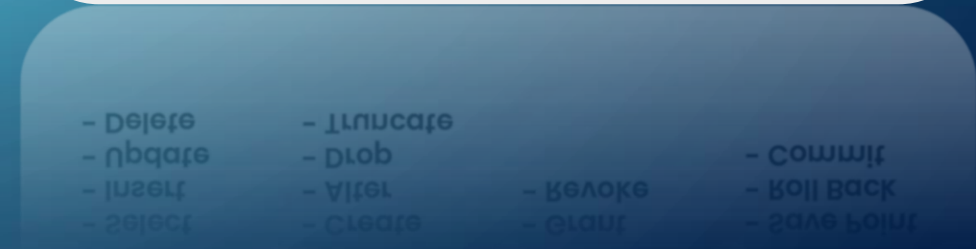
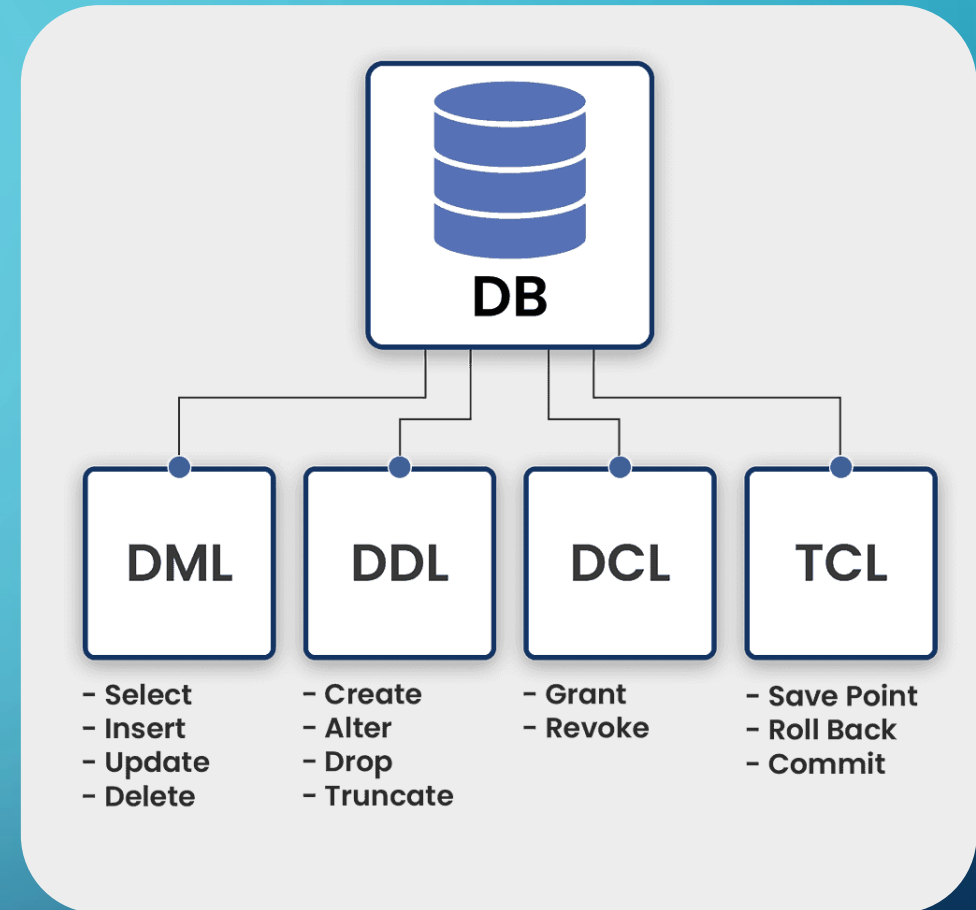
DDL Y DML SON DOS SUBCONJUNTOS DE SQL (STRUCTURED QUERY LANGUAGE) QUE SE UTILIZAN PARA REALIZAR OPERACIONES ESPECÍFICAS EN BASES DE DATOS.

1.DDL (DATA DEFINITION LANGUAGE - LENGUAJE DE DEFINICIÓN DE DATOS):

1. Se utiliza para definir y gestionar la estructura de la base de datos.
2. Incluye comandos como CREATE, ALTER, DROP, etc.
3. Los cambios realizados con DDL afectan la estructura de la base de datos.

2.DML (DATA MANIPULATION LANGUAGE - LENGUAJE DE MANIPULACIÓN DE DATOS):

1. Se utiliza para gestionar los datos almacenados en la base de datos.
2. Incluye comandos como SELECT, INSERT, UPDATE, DELETE, etc.
3. Los cambios realizados con DML afectan los datos almacenados en la base de datos.



Ejemplo DDL

```
CREATE DATABASE UNIFRANZITOS;
```

```
USE UNIFRANZITOS;
```

```
CREATE TABLE campeonato
```

```
(  
    id_campeonato VARCHAR(12) PRIMARY KEY,  
    nombre_campeonato VARCHAR(30) NOT NULL,  
    sede VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE equipo
```

```
(  
    id_equipo VARCHAR(12) PRIMARY KEY,  
    nombre_equipo VARCHAR(30) NOT NULL,  
    categoria VARCHAR(8) NOT NULL ,  
    id_campeonato VARCHAR(12),  
    FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato)  
);
```

Ejemplo DML

Query1.sql - D:\OHGO\Desktop (59)*

```
c1 VARCHAR(50),  
edad INTEGER,  
id_equipo VARCHAR (100)  
FOREIGN KEY (id_equipo) REFERENCES equipo (id_equipo),  
);
```

```
INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)  
VALUES ('camp-111', 'Campeonato Unifranz', 'El Alto'),  
('camp-222', 'Campeonato Unifranz', 'Cochabamba');
```

```
DROP TABLE campeonato; --para borrar tabla
```

```
TRUNCATE TABLE campeonato; -- para borrar datos de la tabla
```

```
INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)  
VALUES('equ-111', 'Google', 'VARONES', 'camp-111'),  
('equ-222', '404 Not found', 'VARONES', 'camp-111'),  
('equ-333', 'girls unifranz', 'MUJERES', 'camp-111');
```

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines connecting to small circles.

QUE SIGNIFICA PRIMARY KEY Y FOREIGN KEY.

PRIMARY KEY (Clave Primaria):

- La PRIMARY KEY es un campo o conjunto de campos en una tabla cuyos valores identifican de manera única cada fila en esa tabla.
- Cada tabla puede tener solo una clave primaria.
- La clave primaria garantiza que no puede haber duplicados en la columna (o conjunto de columnas) y que no puede haber valores nulos en esa columna (o conjunto de columnas)

```
CREATE TABLE campeonato
(
    id_campeonato VARCHAR(12) PRIMARY KEY,
    nombre_campeonato VARCHAR(30) NOT NULL,
    sede VARCHAR(20) NOT NULL
);
```

FOREIGN KEY (Clave Extranjera):

- La FOREIGN KEY es un campo o conjunto de campos en una tabla que hace referencia a la PRIMARY KEY de otra tabla.
- Establece una relación entre dos tablas: la tabla que contiene la clave extranjera y la tabla referenciada por esa clave.
- La clave extranjera ayuda a mantener la integridad referencial, lo que significa que no puedes tener un valor en la clave extranjera que no exista en la clave primaria a la que se hace referencia.

```
CREATE TABLE equipo
(
    id_equipo VARCHAR(12) PRIMARY KEY,
    nombre_equipo VARCHAR(30) NOT NULL,
    categoria VARCHAR(8) NOT NULL,
    id_campeonato VARCHAR(12),
    FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato)
);
```

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines connecting to small circles.

DEFINA QUE ES UNA TABLA Y EL USO DE IDENTITY

- **Tabla:**

- En el contexto de bases de datos relacionales, una tabla es una estructura que organiza los datos en filas y columnas. Cada fila en la tabla representa un registro o una entidad única, y cada columna representa un atributo o campo de esa entidad. Las tablas son la principal forma de almacenar y organizar datos en sistemas de gestión de bases de datos relacionales

- **IDENTITY:**

- IDENTITY es una propiedad que se utiliza para generar automáticamente valores numéricos secuenciales para una columna. Esta característica es especialmente útil cuando se quiere asignar automáticamente identificadores únicos a los registros de una tabla.

The background is a blue gradient. In the corners, there are white line art illustrations of circuit boards or neural networks, with lines connecting to small circles.

PARA QUE SE UTILIZA LA CLÁUSULA WHERE.

- La cláusula WHERE en SQL se utiliza para filtrar los resultados de una consulta basándose en una condición específica. Esta cláusula permite seleccionar solo las filas que cumplen con ciertos criterios, lo que ayuda a refinar y limitar los datos devueltos por una consulta.

La sintaxis básica de la cláusula WHERE es la siguiente:

SELECT: Especifica las columnas que deseas recuperar.

FROM: Indica la tabla de la cual deseas extraer los datos.

WHERE: Define la condición que debe cumplir una fila para ser incluida en los resultados.

```
SELECT nombres, apellidos, edad  
FROM jugador  
WHERE edad > 20;
```

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines connecting to small circles.

PARA QUE SE UTILIZA LA INSTRUCCIÓN INNER
JOIN.

- La instrucción INNER JOIN en SQL se utiliza para combinar filas de dos o más tablas basándose en una condición de unión específica. Este tipo de unión retorna solo las filas que tienen coincidencias en ambas tablas en función de la condición establecida. En otras palabras, solo se incluyen en el resultado aquellas filas que cumplen con la condición de unión en ambas tablas.

```
SELECT columnas  
FROM tabla1  
INNER JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

```
SELECT Empleados.Nombre, Empleados.Cargo, Departamentos.Nombre AS Departamento  
FROM Empleados  
INNER JOIN Departamentos ON Empleados.DepartamentoID = Departamentos.DepartamentoID;
```



APOYÁNDONOS EN EL CONCEPTO DE CONJUNTOS
MUESTRE LOS SIGUIENTE:

2.7.1. EJEMPLO DE INNER JOIN

2.7.2. ADJUNTAR UNA IMAGEN DE CONJUNTOS Y LA
CONSULTA SQL QUE REFLEJE EL INNER JOIN



EJEMPLO

```
SELECT * FROM equipo AS eq INNER JOIN jugador AS jug ON eq.id_equipo = jug.id_equipo WHERE eq.id_equipo = 'equ-222';
```

00 %

Results Messages

	id_equipo	nombre_equipo	categoria	id_campeonato	id_jugador	nombres	apellidos	ci	edad	id_equipo
1	equ-222	404 Not found	VARONES	camp-111	jug-111	Carlos	Villa	8997811LP	19	equ-222
2	equ-222	404 Not found	VARONES	camp-111	jug-222	Pedro	Salas	8997822LP	20	equ-222
3	equ-222	404 Not found	VARONES	camp-111	jug-333	Saul	Araj	8997833LP	21	equ-222