BASE DE DATOS

ESTUDIANTE: LUIS FERNANDO MAMANI CHOQUE

BASE DE DATOS

- Las bases de datos son sistemas de almacenamiento y organización de información que permiten gestionar datos de manera eficiente. Están compuestas por tablas que contienen registros con información específica. Cada tabla tiene campos que representan atributos, y se utiliza una clave primaria para identificar registros de manera única. Las consultas permiten recuperar datos, y las relaciones pueden vincular registros de diferentes tablas. Los sistemas de gestión de bases de datos (DBMS) son software utilizados para crear y gestionar bases de datos. Las bases de datos son esenciales en la gestión de información en organizaciones y desempeñan un papel clave en la toma de decisiones y el análisis de datos. Las bases de datos son sistemas de almacenamiento y organización de información que permiten gestionar datos de manera eficiente. Están compuestas por tablas que contienen registros con información específica. Cada tabla tiene campos que representan atributos, y se utiliza una clave primaria para identificar registros de manera única. Las consultas permiten recuperar datos, y las relaciones pueden vincular registros de diferentes tablas. Los sistemas de gestión de bases de datos (DBMS) son software utilizados para crear y gestionar bases de datos. Las bases de datos son esenciales en la gestión de información en organizaciones y desempeñan un papel clave en la toma de decisiones y el análisis de datos.

¿A QUE SE REFIERE CUANDO SE HABLA DE BASES DE DATOS RELACIONALES?

- Las bases de datos relacionales son sistemas de gestión de bases de datos que se basan en el modelo de datos relacional. En este modelo, los datos se organizan en tablas con filas y columnas, y se establecen relaciones entre las tablas mediante claves primarias y claves foráneas. Esto permite almacenar y gestionar datos de manera estructurada y coherente. Se utiliza el lenguaje SQL para interactuar con estas bases de datos, y ofrecen características como integridad referencial y soporte para transacciones. Ejemplos de sistemas de bases de datos relacionales incluyen MySQL, PostgreSQL, Oracle y SQL Server. Estas bases de datos son ampliamente utilizadas en aplicaciones empresariales y sistemas donde la integridad y la estructura de los datos son esenciales.

¿QUÉ ES EL MODELO ENTIDAD RELACIÓN Y/O DIAGRAMA ENTIDAD RELACIÓN?

- El modelo entidad-relación (ER) es un enfoque de diseño de bases de datos que utiliza diagramas entidad-relación (DER) para representar visualmente la estructura de una base de datos y las relaciones entre sus elementos. En este modelo, las "entidades" representan clases de objetos o conceptos, los "atributos" son características de las entidades, y las "relaciones" indican las asociaciones entre las entidades. Cada entidad tiene una "clave primaria" única, y se utiliza la "cardinalidad" y la "participación" para describir la naturaleza de las relaciones. Los DER son herramientas esenciales para diseñar y planificar bases de datos antes de su implementación, ayudando a garantizar una estructura de datos precisa y eficiente.

¿CUÁLES SON LAS FIGURAS QUE REPRESENTAN A UN DIAGRAMA ENTIDAD RELACIÓN? EXPLIQUE CADA UNA DE ELLAS.

- En un diagrama entidad-relación (DER), se utilizan varias figuras geométricas para representar los elementos clave del modelo entidad-relación. A continuación, se describen las figuras más comunes utilizadas en un DER y su significado:

1. **Rectángulo (Entidad):** Un rectángulo se utiliza para representar una entidad en el modelo entidad-relación. Una entidad es una clase de objeto o concepto que tiene atributos para describirlo. Los atributos se enumeran dentro del rectángulo, y la entidad se nombra en la parte superior del mismo. Por ejemplo, si estamos modelando una base de datos de una biblioteca, podríamos tener un rectángulo etiquetado como "Libro" con atributos como "Título", "Autor" y "Año de Publicación".

2. **Elipse o Círculo (Atributo):** Una elipse o un círculo se utiliza para representar un atributo de una entidad. Los atributos son las propiedades o características de una entidad y se relacionan con esa entidad específica. Por ejemplo, dentro del rectángulo de la entidad "Libro", tendríamos el atributo "Título" representado como una elipse conectada a la entidad.

3. **Línea (Relación):** Las líneas se utilizan para representar las relaciones entre las entidades en el DER. Las líneas conectan las entidades y pueden estar etiquetadas con un verbo que describe la naturaleza de la relación. Por ejemplo, una línea que conecta las entidades "Autor" y "Libro" podría estar etiquetada como "Escribe", indicando que un autor está relacionado con un libro a través de la acción de escribir.

4. **Rombo (Relación de Cardinalidad):** Un rombo se utiliza para indicar la cardinalidad de una relación, es decir, cuántas instancias de una entidad pueden estar relacionadas con las instancias de la otra entidad en una relación. Dentro del rombo se pueden colocar números o símbolos para denotar la cardinalidad, como "1", "N" o "0..1". Por ejemplo, un rombo conectado a una línea entre "Autor" y "Libro" podría indicar "N:N", lo que significa que varios autores pueden escribir varios libros, y viceversa.

5. **Círculo con una Línea (Clave Primaria):** A veces, se utiliza un círculo con una línea para indicar la clave primaria de una entidad. La clave primaria es un atributo único que identifica de manera única cada instancia de esa entidad en la base de datos. Este símbolo se coloca junto al atributo que actúa como clave primaria.

6. **Círculo con un Punto (Clave Parcial):** Un círculo con un punto se utiliza para indicar una clave parcial, que es un atributo que, junto con otros, forma la clave primaria de una entidad. Este símbolo se coloca junto a los atributos que forman parte de la clave parcial.

¿QUÉ ES SQL SERVER Y QUÉ ES SQL SERVER MANAGEMENT STUDIO?

- SQL Server es un sistema de gestión de bases de datos desarrollado por Microsoft, mientras que SQL Server Management Studio (SSMS) es una herramienta de administración y desarrollo que se utiliza para interactuar con bases de datos SQL Server de manera eficiente y conveniente.

¿CÓMO SE CREA UNA BASE DE DATOS?

1. **Selecciona un DBMS:** Elije un sistema de gestión de bases de datos que se adapte a tus necesidades y asegúrate de tenerlo instalado.
2. **Planifica tu base de datos:** Define la estructura de tu base de datos, incluyendo las tablas, atributos y relaciones entre ellas.
3. **Inicia sesión en el DBMS:** Abre la herramienta de administración del DBMS que estás usando.
4. **Crea una nueva base de datos:** Usa comandos SQL o la interfaz gráfica para crear una nueva base de datos y asignarle un nombre.
5. **Define las tablas:** Crea las tablas que compondrán la base de datos, especificando los nombres y atributos de cada tabla.
6. **Establece relaciones (si es necesario):** Define relaciones entre tablas usando claves foráneas.
7. **Inserta datos (opcional):** Inserta registros iniciales si deseas poblar la base de datos con datos iniciales.
8. **Realiza pruebas:** Verifica que la base de datos funcione correctamente mediante consultas y pruebas.
9. **Implementa la base de datos:** Implementa la base de datos en tu aplicación, sitio web u otro sistema donde necesites acceder a los datos almacenados.

- Recuerda que los detalles específicos pueden variar según el sistema de gestión de bases de datos (DBMS) que utilices, por lo que consulta la documentación correspondiente para obtener instrucciones detalladas. La planificación y ejecución cuidadosa son clave para garantizar la integridad y el rendimiento de tus datos.

¿PARA QUÉ SIRVE EL COMANDO USE?

- El comando USE se utiliza en sistemas de gestión de bases de datos relacionales (DBMS) para cambiar el contexto y seleccionar una base de datos específica en la que se ejecutarán consultas y comandos posteriores. Esto simplifica la escritura de consultas y operaciones, ya que permite trabajar directamente con tablas dentro de la base de datos seleccionada sin necesidad de especificar el nombre de la base de datos en cada consulta. Su uso varía según el DBMS específico que estés utilizando, por lo que es importante consultar la documentación correspondiente para obtener instrucciones precisas.

CREAR UNA TABLA CUALQUIERA CON 3 COLUMNAS Y SU PRIMARY KEY.

```
CREATE TABLE Ejemplo (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Nombre VARCHAR(50),
    Edad INT
);
```

INSERTAR 3 REGISTROS A LA TABLA CREADA ANTERIORMENTE.

- -- Insertar el primer registro
- INSERT INTO Ejemplo (Nombre, Edad) VALUES ('Juan', 30);

- -- Insertar el segundo registro
- INSERT INTO Ejemplo (Nombre, Edad) VALUES ('María', 25);

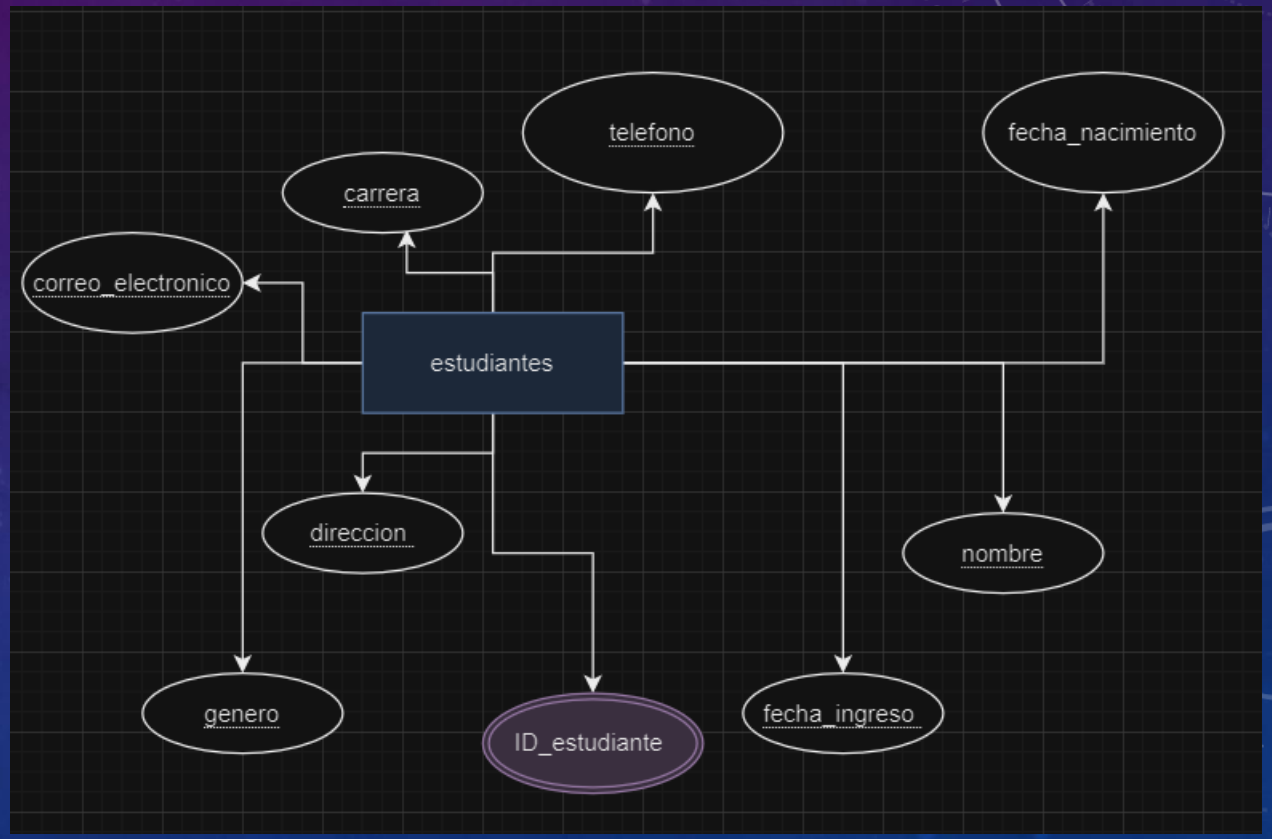
- -- Insertar el tercer registro
- INSERT INTO Ejemplo (Nombre, Edad) VALUES ('Carlos', 35);

¿CÓMO SE ELIMINA UNA TABLA?

- DROP TABLE NombreDeLaTabla;

CREAR EL DISEÑO PARA UN ESTUDIANTE

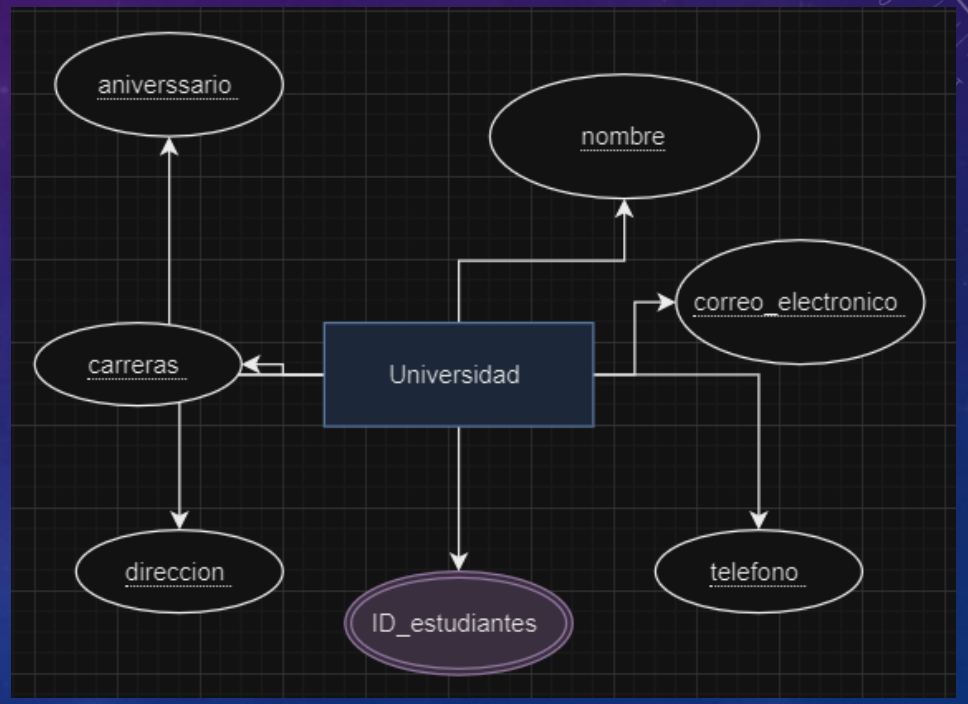
```sql
CREATE DATABASE EVALUACIONH2

USE ESTUDIANTES
-- Crear la tabla Estudiantes
CREATE TABLE Estudiantes
(
    ID_Estudiante INT PRIMARY KEY,
    Nombre VARCHAR(25),
    Fecha_Nacimiento DATE,
    Genero VARCHAR(20),
    Direccion VARCHAR(255),
    Correo_Electronico VARCHAR(10),
    Telefono VARCHAR(20),
    Fecha_Ingreso DATE,
    Carrera VARCHAR(10)
);
```



CREAR EL DISEÑO PARA UNA UNIVERSIDAD

SQLQuery1.sql - M...AMANI\59171 (73))*

```sql
CREATE DATABASE EVALUACIONH2

USE universidad
-- Crear la tabla Estudiantes
CREATE TABLE UNIVERSIDAD
(
    ID_Estudiante INT PRIMARY KEY,
    Nombre VARCHAR(25),
    Aniversario DATE,
    Direccion VARCHAR(255),
    Correo_Electronico VARCHAR(10),
    Telefono VARCHAR(20),
    Carreras VARCHAR(10)
);
```



POLLOS COPA

SQLQuery1.sql - M...AMANI\59171 (73))*

```sql
-- Crear la base de datos POLLOS_COPA
CREATE DATABASE POLLOS_COPA;

-- Cambiar al contexto de la base de datos POLLOS_COPA
USE POLLOS_COPA;

CREATE TABLE cliente
(
    ID_Cliente INT PRIMARY KEY,
    Nombre NVARCHAR(255),
    Direccion NVARCHAR(255),
    Telefono NVARCHAR(20)
);

-- Insertar dos registros en la tabla cliente
INSERT INTO cliente (ID_Cliente, Nombre, Direccion, Telefono)
VALUES
    (1, 'Juan Pérez', 'Calle 123, Ciudad', '123-456-7890'),
    (2, 'María López', 'Avenida 456, Ciudad', '987-654-3210');
    select*
    from cliente

    -- Crear la tabla detalle_pedido
CREATE TABLE detalle_pedido
```

100 %

Results | Messages

	ID_Cliente	Nombre	Direccion	Telefono
1	1	Juan Pérez	Calle 123, Ciudad	123-456-7890
2	2	María López	Avenida 456, Ciudad	987-654-3210

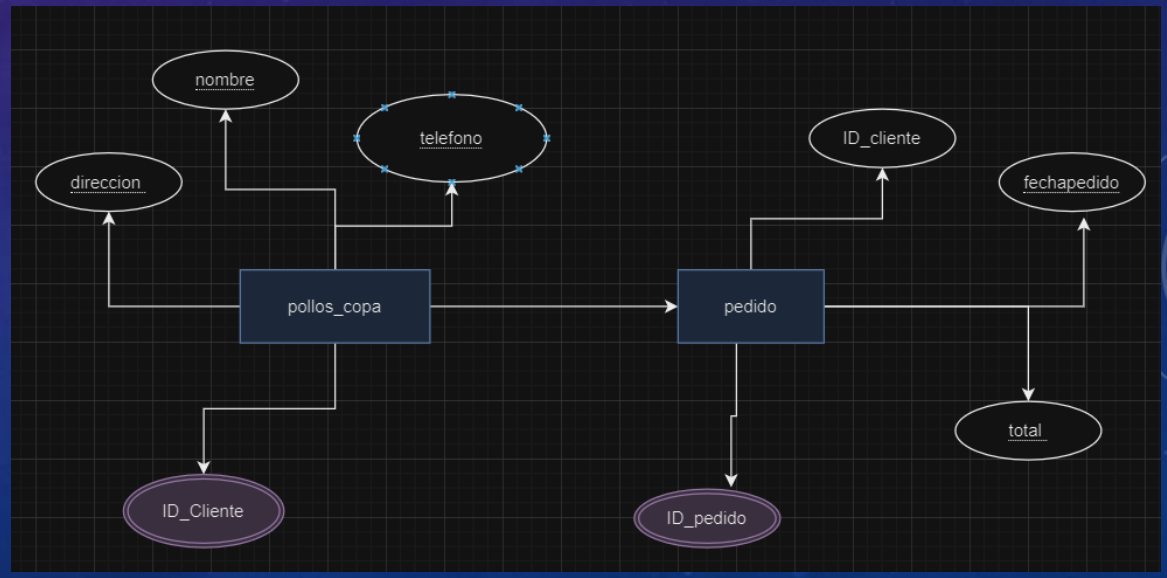
SQLQuery1.sql - M...AMANI\59171 (73))*

```sql
);

-- Insertar dos registros en la tabla detalle_pedido
INSERT INTO detalle_pedido (ID_Detalle, ID_Pedido, ID_Producto, Cantidad, Precio)
VALUES
    (1, 1, 101, 2, 15.99),
    (2, 2, 102, 3, 12.50);

    -- Crear la tabla pedido
CREATE TABLE pedido
(
    ID_Pedido INT PRIMARY KEY,
    ID_Cliente INT,
    FechaPedido DATE,
    Total DECIMAL(10, 2)
);

-- Insertar dos registros en la tabla pedido
INSERT INTO pedido (ID_Pedido, ID_Cliente, FechaPedido, Total)
VALUES
    (1, 1, '2023-09-11', 31.98),
    (2, 2, '2023-09-12', 37.50);
    select*
    from pedido
```

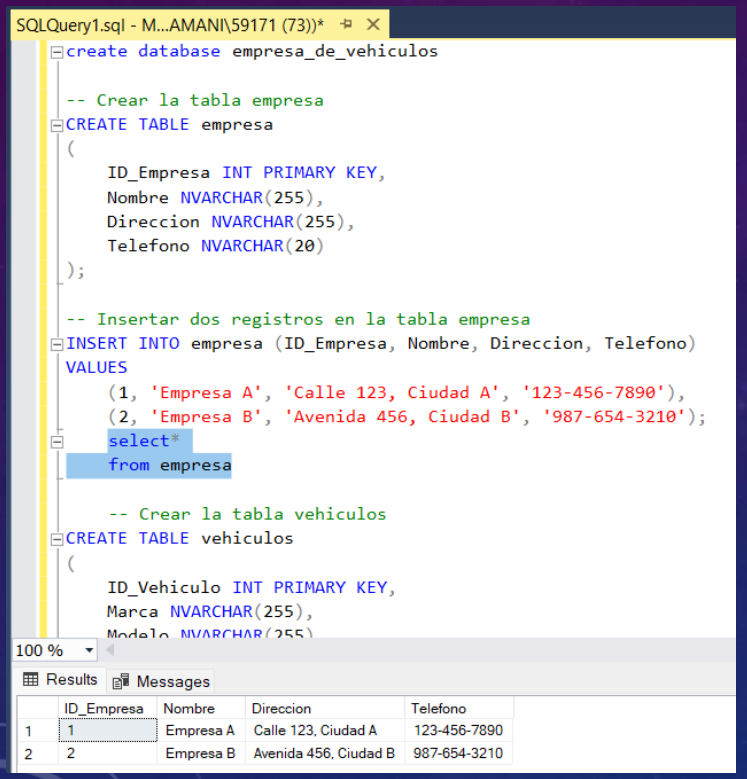
100 %

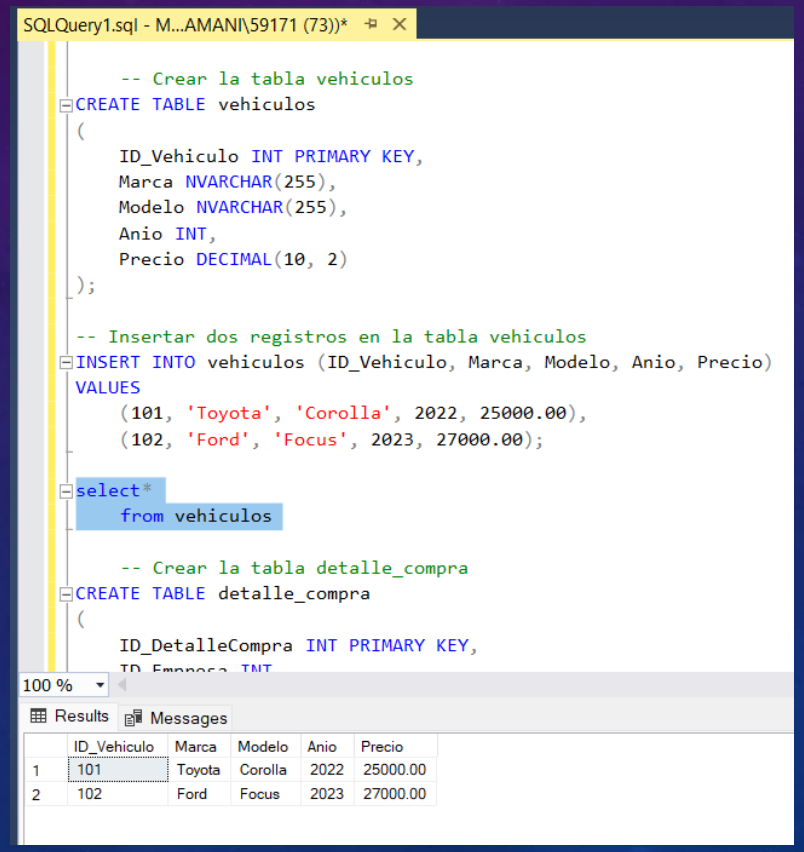
Results | Messages

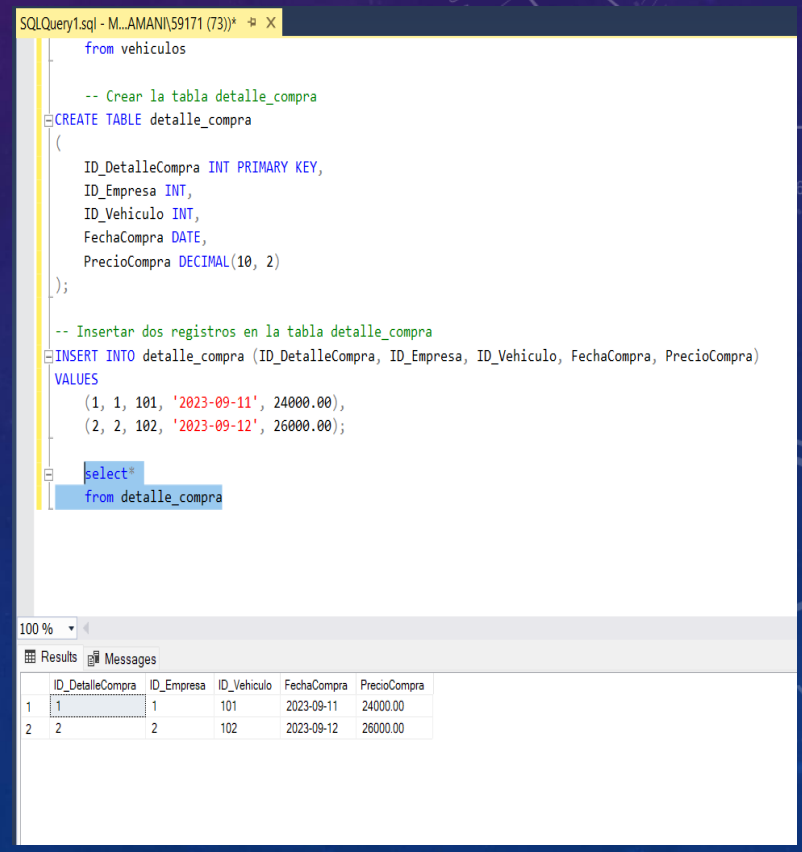
	ID_Pedido	ID_Cliente	FechaPedido	Total
1	1	1	2023-09-11	31.98
2	2	2	2023-09-12	37.50

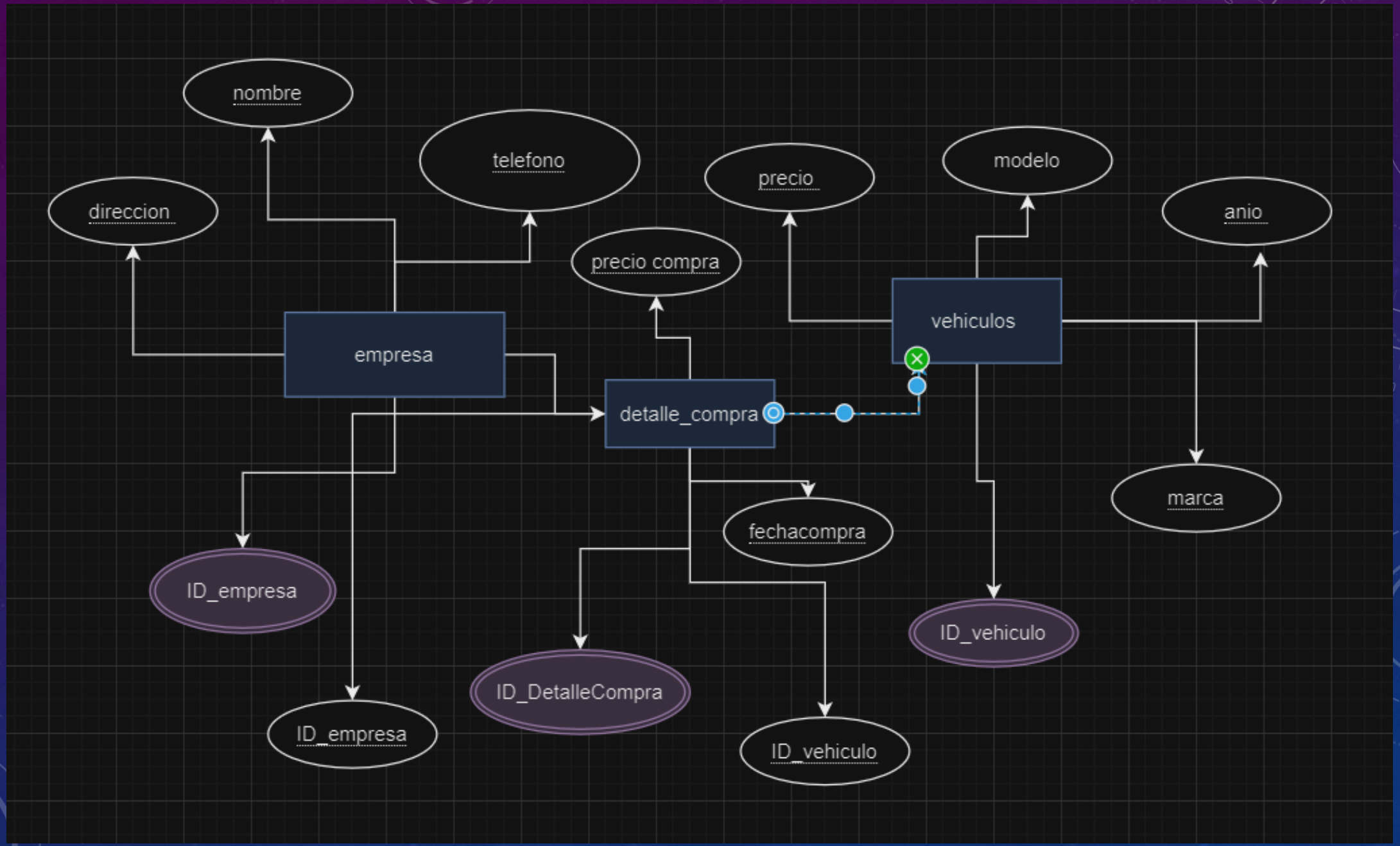


EMPRESA DE VEHICULOS








nombre
telefono
direccion
precio
modelo
anio
precio compra
vehiculos
empresa
detalle_compra
marca
fechacompra
ID_empresa
ID_vehiculo
ID_DetalleCompra
ID_empresa
ID_vehiculo