

A glowing lightbulb with a circuit board overlay. The lightbulb is on the right side of the image, with its filament glowing. A circuit board overlay is visible on the right side of the image, with lines connecting to the lightbulb. The background is a solid blue color.

BASE DE DATOS DEFENSA DEL HITO 2

POR:MARCO ANTONIO CALLE VAQUIATA

CONCEPTO

Una **base de datos** es un conjunto de información organizada de manera que pueda ser utilizada eficientemente. Un directorio telefónico, un diccionario, un calendario o un libro de recetas son ejemplos **de bases de datos**. La información en una **base** está organizada en forma de registros.

PARTE TEORICA

1. ¿QUÉ SON LAS BASES DE DATOS?

UNA BASE DE DATOS ES UNA RECOPIACIÓN ORGANIZADA DE INFORMACIÓN O DATOS ESTRUCTURADOS, QUE NORMALMENTE SE ALMACENA DE FORMA ELECTRÓNICA EN UN SISTEMA INFORMÁTICO. NORMALMENTE, UNA BASE DE DATOS ESTÁ CONTROLADA POR UN SISTEMA DE GESTIÓN DE BASES DE DATOS (DBMS).

2. ¿A QUE SE REFIERE CUANDO SE HABLA DE BASES DE DATOS RELACIONALES?2. ¿

Una **base de datos relacional** es, en esencia, un conjunto de tablas (o relaciones) formadas por filas (registros) y columnas (campos); así, cada registro (cada fila) tiene una ID única, denominada clave y las columnas de la tabla contienen los atributos de los **datos**.

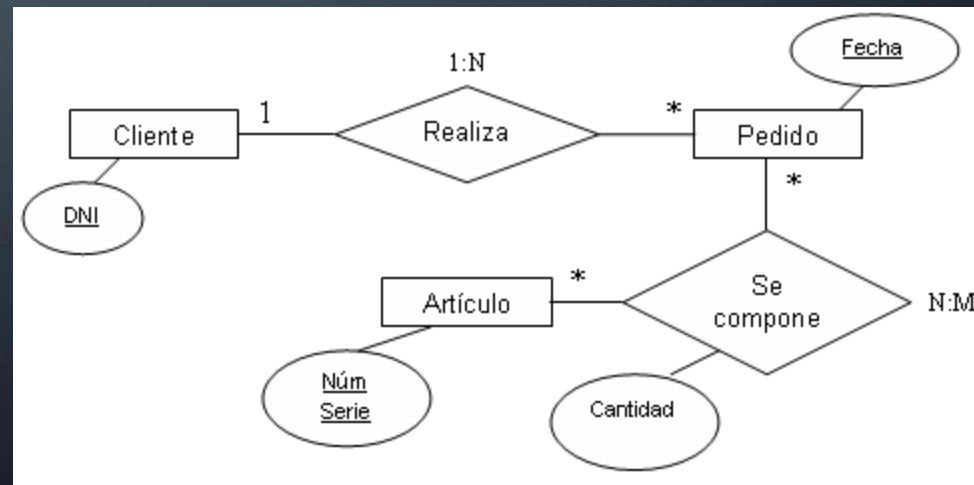
3. ¿QUÉ ES EL MODELO ENTIDAD RELACIÓN Y/O DIAGRAMA ENTIDAD RELACIÓN?

UN DIAGRAMA ENTIDAD-RELACIÓN, TAMBIÉN CONOCIDO COMO MODELO ENTIDAD RELACIÓN O ERD, ES UN TIPO DE DIAGRAMA DE FLUJO QUE ILUSTRÁ CÓMO LAS "ENTIDADES", COMO PERSONAS, OBJETOS O CONCEPTOS, SE RELACIONAN ENTRE SÍ DENTRO DE UN SISTEMA.

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

ENTIDAD:

Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual. Hay dos tipos de entidades: fuertes y débiles



CONJUNTO DE ENTIDADES:

ES UNA COLECCIÓN DE ENTIDADES QUE COMPARTEN LOS MISMOS ATRIBUTOS O CARACTERÍSTICAS. EJEMPLOS: • TODOS LOS ATLETAS QUE PARTICIPAN EN LOS JUEGOS OLÍMPICOS, COMPARTEN SUS ATRIBUTOS: NOMBRE, NÚMERO DE IDENTIFICACIÓN, EDAD, PESO, CATEGORÍA... • TODOS LOS PAÍSES DEL MUNDO, COMPARTEN LAS CARACTERÍSTICAS: NOMBRE, CONTINENTE, ÁREA, LENGUA PRINCIPAL, LENGUA SECUNDARIA, MONEDA.

RELACIÓN (INTERRELACIÓN) :

ES UNA CORRESPONDENCIA O ASOCIACIÓN ENTRE DOS O MÁS ENTIDADES. CADA RELACIÓN TIENE UN NOMBRE QUE DESCRIBE SU FUNCIÓN. LAS RELACIONES SE REPRESENTAN GRÁFICAMENTE MEDIANTE ROMBOS Y SU NOMBRE APARECE EN EL INTERIOR.

ATRIBUTO

ES UNA CARACTERÍSTICA DE INTERÉS O UN HECHO SOBRE UNA ENTIDAD O SOBRE UNA RELACIÓN. LOS ATRIBUTOS REPRESENTAN LAS PROPIEDADES BÁSICAS DE LAS BASES DE DATOS MIS 308 4 ENTIDADES Y DE LAS RELACIONES. TODA LA INFORMACIÓN EXTENSIVA ES PORTADA POR LOS ATRIBUTOS. GRÁFICAMENTE, SE REPRESENTAN MEDIANTE BOLITAS QUE CUELGAN DE LAS ENTIDADES O RELACIONES A LAS QUE PERTENECEN.

IDENTIFICADOR

UN IDENTIFICADOR DE UNA ENTIDAD ES UN ATRIBUTO O CONJUNTO DE ATRIBUTOS QUE DETERMINA DE MODO ÚNICO CADA OCURRENCIA DE ESA ENTIDAD. UN IDENTIFICADOR DE UNA ENTIDAD DEBE CUMPLIR DOS CONDICIONES:

1. NO PUEDEN EXISTIR DOS OCURRENCIAS
2. SI SE OMITE CUALQUIER ATRIBUTO DEL IDENTIFICADOR, LA CONDICIÓN ANTERIOR DEJA DE CUMPLIRSE

DE LA ENTIDAD CON EL MISMO VALOR DEL.

JERARQUÍA DE GENERALIZACIÓN

UNA ENTIDAD E ES UNA GENERALIZACIÓN DE UN GRUPO DE ENTIDADES E_1, E_2, \dots, E_n , SI CADA OCURRENCIA DE CADA UNA DE ESAS ENTIDADES ES TAMBIÉN UNA OCURRENCIA DE E . TODAS LAS PROPIEDADES DE LA ENTIDAD GENÉRICA E SON HEREDADAS POR LAS SUBENTIDADES. CADA JERARQUÍA ES TOTAL O PARCIAL, Y EXCLUSIVA O SUPERPUESTA. UNA JERARQUÍA ES TOTAL SI CADA OCURRENCIA DE LA ENTIDAD GENÉRICA CORRESPONDE AL MENOS CON UNA OCURRENCIA DE ALGUNA SUBENTIDAD.

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

SQL SERVER MANAGEMENT STUDIO (SSMS PARA ABREVIAR) ES UN ENTORNO DE DESARROLLO INTEGRADO PARA ADMINISTRAR CUALQUIER INFRAESTRUCTURA SQL. SE UTILIZA PARA ACCEDER, ADMINISTRAR, CONFIGURAR Y DESARROLLAR TODOS LOS COMPONENTES DE SQL SERVER Y SQL DATABASE

6. ¿CÓMO SE CREA UNA BASE DE DATOS?

Crear una base de datos en blanco

Haga clic en la pestaña Archivo, en Nuevo y luego, en **Base de datos** en blanco. (junto al cuadro Nombre de archivo), vaya a la nueva ubicación y haga clic en Aceptar. Haga clic en Crear.

7. ¿PARA QUÉ SIRVE EL COMANDO USE?

EN BASE DATA LOS EL COMANDO USE SE USA EN EL MANAGER ESTUDIO Y SIRVE PARA BÁSICAMENTE DESIGNAR EN QUE CARPETA TRABAJAREMOS O CREAREMOS LOS DEMÁS COMANDOS

```
create database defensa  
use defensa
```

8. CREAR UNA TABLA CUALQUIERA CON 3 COLUMNAS Y SU PRIMARYKEY

PARA CREAR UNA TABLA CON 3 COLUMNAS Y SU PRIMARY KEY, TENEMOS QUE SELECCIONAR EN LA CARPETA QUE TRABAJAREMOS EN MI CASO ME CREE UNA CARPETA LLAMADA **DEFENSA** Y LUEGO DE ESO CREAMOS LA TABLA CON EL NOMBRE QUE USTEDES QUIERAN, DEBAJO DE ESO COLOCAMOS ID_ (EL NOMBRE QUE PUSIERON EN LA TABLA) EH IGUAL SEÑALAMOS QUE SI ES VARCHAR O INT. DESPUES COLOCAMOS EL LIMITE DE CADA UNO DE ELLOS.

PROCEDEMOS A COLOCAR A CADA UNO DE LAS COLUMNAS SI SON VARCHAR O INT EL VARCHAR ES PARA SEÑALAR QUE PONDREMOS NUMEROS Y LETRAS EL INT ES PARA SOLO COLOCAR NUMEROS


```
SQLQuery1.sql - DE...4\Marco Calle (52))*
use defensa
create table colegio
(
  id_colegio varchar (40) primary key,
  nombres varchar (30),
  apellido varchar (30),
  curso varchar (30),
  edad int,
);
insert into colegio(id_colegio,nombres,apellido,curso,edad)
values('123123','Marco','Calle','curso-23',19);
```

100 %

Messages

(1 row affected)

Completion time: 2022-04-03T14:31:23.8629299-03:00

9. INSERTAR 3 REGISTROS A LA TABLA CREADA ANTERIORMENTE.

PARA PODER INSERTAR DATOS A NUESTRA BASE DE DATOS SE USA EL COMANDO INSERT INTO(LOS NOMBRES DE LAS COLUMNAS Y PARA SEPARARLOS SE COLOCA UNA COMA ENTRE ELLAS)

VALUES SIRVE PARA YA COLOCAR LOS DATOS PERO LOS DATOS TIENE QUE SER INTRODUCIDO ENTRE “ PARA QUE ENTRE LOS DATOS CORRESPONDIENTES

10.¿CÓMO SE ELIMINA UNA TABLA?

PARA ELIMANAR UNA TABLE O DATABSE SE USA EL COMANDO DROP Y ESTE COMANDO LO BORRA PERO SE USA DE ESTA FORMA.

DROP TABLE (EL NOMBRE DE LA TABLA)
DROP DATABASE (EL NOMBRE DE LA DATABSE)

```
DROP DATABASE defensa
```

```
DROP TABLE colegio
```

PARTE PRACTICA

11. Crear el diseño para una UNIVERSIDAD.

UNIVERSIDAD

ID_UNIVERSIDAD VARCHAR (30) PRIMARY KEY

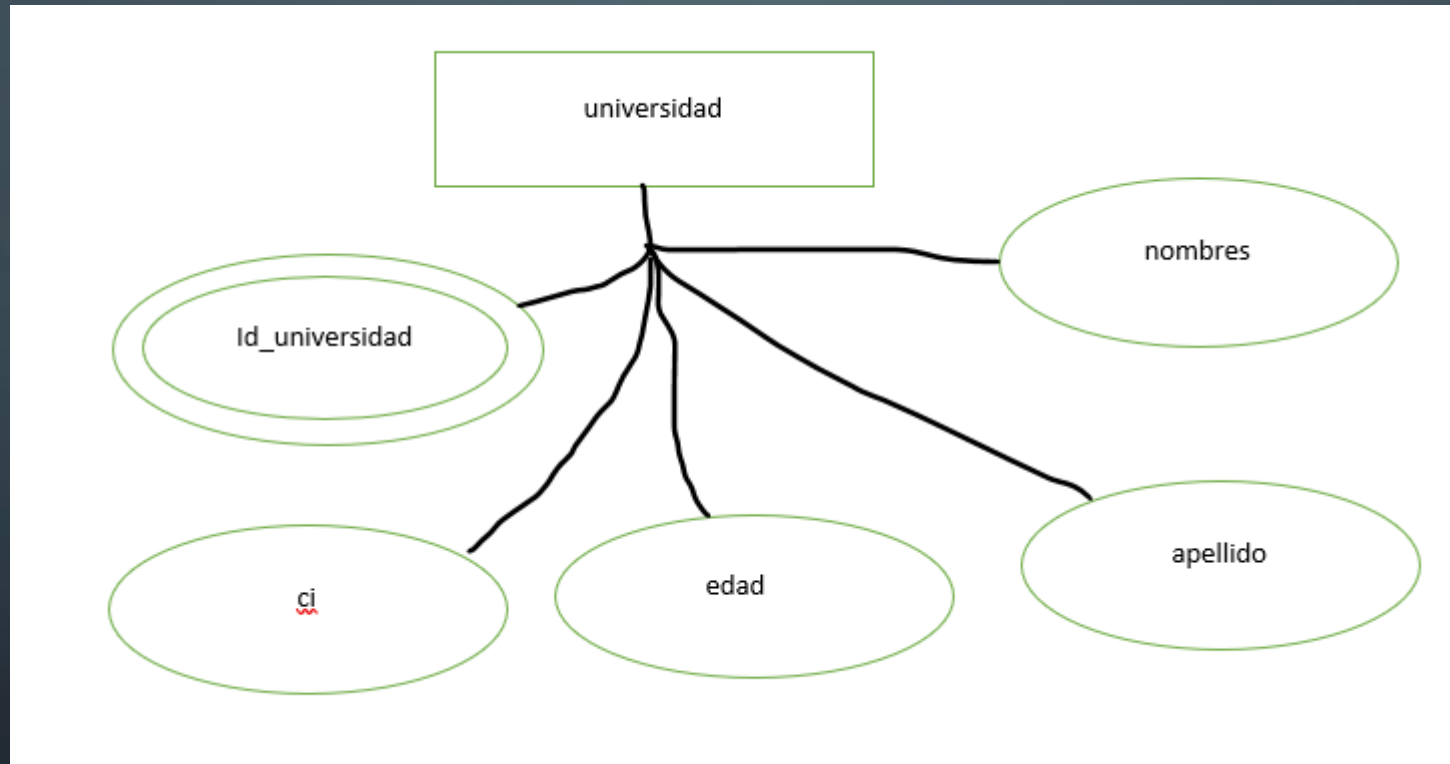
NOMBRE= VARCHAR (30)

APELLIDO =VARCHAR (30)

EDAD =INT

CI =VARCHAR (30)

12. CREAR EL DIAGRAMA ENTIDAD RELACIÓN E-R PARA EL EJERCICIO ANTERIOR



13.CREAR LA TABLA UNIVERSIDAD EN BASE AL DISEÑO ANTERIOR

SQLQuery1.sql - DE...4\Marco Calle (52))*

```
create database Hito2Tarea
USE Hito2Tarea
CREATE TABLE UNIVERSIDAD
(
  ID_UNIVERSIDAD VARCHAR (30) PRIMARY KEY,
  NOMBRES VARCHAR (30),
  APELLIDOS VARCHAR (30),
  EDAD INT,
  CI VARCHAR (30),
);
```

14. AGREGAR REGISTROS A LA TABLA CREADA ANTERIORMENTE.

SQLQuery1.sql - DE...4\Marco Calle (52))*

```
(  
  ID_UNIVERSIDAD VARCHAR (30) PRIMARY KEY,  
  NOMBRES VARCHAR (30),  
  APELLIDOS VARCHAR (30),  
  EDAD INT,  
  CI VARCHAR (30),  
);  
INSERT INTO UNIVERSIDAD(ID_UNIVERSIDAD,NOMBRES,APELLIDOS,EDAD,CI)  
values('UNI-1234', 'MARCO', 'CALLE',22, 'LP123123123');
```


15.CREAR LAS TABLAS Y 2 REGISTROS PARA CADA TABLA PARA EL SIGUIENTE MODELO ER

SQLQuery1.sql - DE...4\Marco Calle (52))*

```
(  
  ID_UNIVERSIDAD VARCHAR (30) PRIMARY KEY,  
  NOMBRES VARCHAR (30),  
  APELLIDOS VARCHAR (30),  
  EDAD INT,  
  CI VARCHAR (30),  
);  
INSERT INTO UNIVERSIDAD(ID_UNIVERSIDAD,NOMBRES,APELLIDOS,EDAD,CI)  
values('UNI-1234', 'MARCO', 'CALLE',22, 'LP123123123');
```

14. AGREGAR REGISTROS A LA TABLA CREADA ANTERIORMENTE.

SQLQuery1.sql - DE...4\Marco Calle (52))*

```
(  
  ID_UNIVERSIDAD VARCHAR (30) PRIMARY KEY,  
  NOMBRES VARCHAR (30),  
  APELLIDOS VARCHAR (30),  
  EDAD INT,  
  CI VARCHAR (30),  
);  
INSERT INTO UNIVERSIDAD(ID_UNIVERSIDAD,NOMBRES,APELLIDOS,EDAD,CI)  
values('UNI-1234', 'MARCO', 'CALLE',22, 'LP123123123');
```

15. CREAR LAS TABLAS Y 2 REGISTROS PARA CADA TABLA PARA EL SIGUIENTE MODELO ER..

```
SQLQuery1.sql - DEMO (Marco Calle (33))
CREATE DATABASE POLLO_COPA
USE POLLO_COPA
CREATE TABLE CLIENTE
(
  ID_CLIENTE VARCHAR (30) PRIMARY KEY,
  FULLNAME VARCHAR (30),
  LASTNAME VARCHAR (30),
  EDAD INT,
  DOMICILIO VARCHAR (30),
);
INSERT INTO CLIENTE (ID_CLIENTE, FULLNAME, LASTNAME, EDAD, DOMICILIO)
VALUES ('CLI-231', 'MARCO', 'CALLE', 22, 'AROMA1231');

INSERT INTO CLIENTE (ID_CLIENTE, FULLNAME, LASTNAME, EDAD, DOMICILIO)
VALUES ('CLI-456', 'MARCO', 'CALLE', 22, 'PLAZA9881');

USE POLLO_COPA
CREATE TABLE PEDIDO
(
  ID_PEDIDO VARCHAR (40) PRIMARY KEY,
  ARTICULO VARCHAR (40),
  COSTO INT,
```

SQLQuery2.sql - DE...4\Marco Calle (55))*

```
COSTO INT,  
FECHA VARCHAR (40),  
);  
= INSERT INTO PEDIDO (ID_PEDIDO,ARTICULO,COSTO,FECHA)  
VALUES ('PEDI-1', 'ZAPATOS',220, '04-ABRIL-2022');  
= INSERT INTO PEDIDO (ID_PEDIDO,ARTICULO,COSTO,FECHA)  
VALUES ('PEDI-2', 'CHANCLETAS',170, '04-ABRIL-2022');  
  
USE POLLO_COPA  
= CREATE TABLE REALIZANDO  
(  
ID_REALIZANDO VARCHAR (30) PRIMARY KEY,  
ID_CLIENTE VARCHAR (30),  
ID_PEDIDO VARCHAR (40),  
  
FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTE(ID_CLIENTE),  
FOREIGN KEY (ID_PEDIDO) REFERENCES PEDIDO (ID_PEDIDO)  
);  
= INSERT INTO REALIZANDO(ID_REALIZANDO,ID_CLIENTE,ID_PEDIDO)  
VALUES('REA-A', 'CLI-231', 'PEDI-1');|  
= INSERT INTO REALIZANDO(ID_REALIZANDO,ID_CLIENTE,ID_PEDIDO)  
VALUES('REA-B', 'CLI-456', 'PEDI-2');
```

16. CREAR EL MODELO ENTIDAD RELACIÓN ER Y SU CÓDIGO SQL

COMPRA

NOMBRE=VARCHAR

APELLIDO=VARCHAR

COSTO=INT

EDAD=INT

VEHICULOS

EMPRESA=VARCHAR

FECHA=VARCHAR

MODELO=VARCHAR

Modelo entidad relación ER

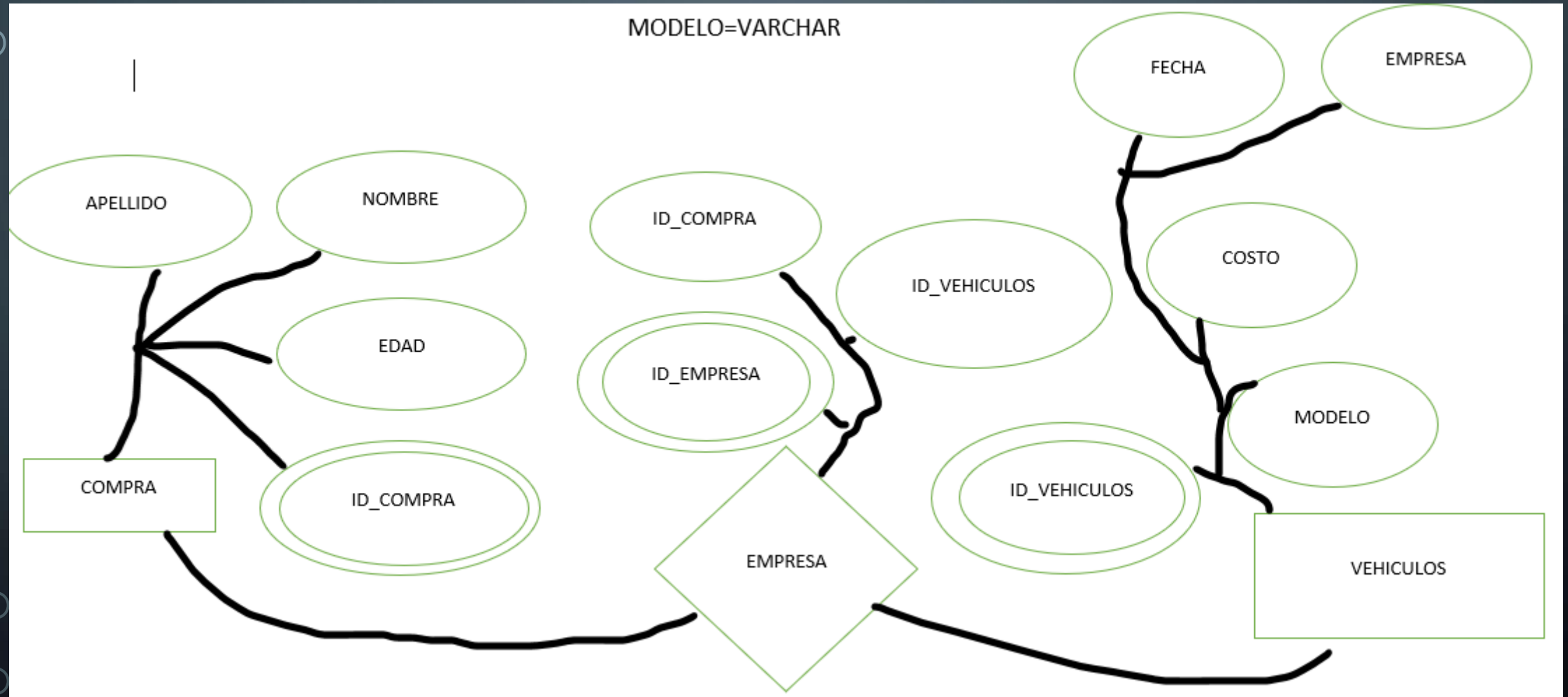
The diagram illustrates an Entity-Relationship (ER) model for a car purchase system. It features three main entities: COMPRA, EMPRESA, and VEHICULOS.

- COMPRA Entity:** Attributes include APELLIDO, NOMBRE, EDAD, and ID_COMPRA (underlined as a primary key).
- EMPRESA Entity:** Attribute is ID_EMPRESA (underlined as a primary key).
- VEHICULOS Entity:** Attributes include FECHA, EMPRESA, COSTO, MODELO, and ID_VEHICULOS (underlined as a primary key).

Relationships are indicated by lines connecting the entities:

- A line connects the COMPRA entity to the EMPRESA entity.
- A line connects the EMPRESA entity to the VEHICULOS entity.

Handwritten notes include "MODELO=VARCHAR" near the top and a vertical line on the left side.



Código SQL.

SQLQuery3.sql - DE...4\Marco Calle (54))*

```
create database hito2
use hito2
create table compra
(
    id_compra varchar (25) primary key,
    nombre varchar (25),
    apellidos varchar (25),
    edad int
);
insert into compra(id_compra,nombre,apellidos,edad)
values('COM-1', 'MARCO', 'CALLE',22)
insert into compra(id_compra,nombre,apellidos,edad)
values('COM-2', 'MARCO', 'CALLE',22)

use hito2
create table vehiculos
(
    id_vehiculos varchar(30) primary key,
    empresa varchar(30),
    fecha varchar(30),
    costo int,
    modelo varchar(30)
);
insert into vehiculos(id_vehiculos,empresa,fecha,costo,modelo)
values('v-1','MARC','03-abril-2022',230.000,'yampa-123')
insert into vehiculos(id_vehiculos,empresa,fecha,costo,modelo)
```

100 %

Código SQL.

SQLQuery3.sql - DE...4\Marco Calle (54))*

```
    modelo varchar(30)
);
insert into vehiculos(id_vehiculos,empresa,fecha,costo,modelo)
values('v-1','MARC','03-abril-2022',230.000,'yampa-123')
insert into vehiculos(id_vehiculos,empresa,fecha,costo,modelo)
values('v-2','MARC','04-abril-2022',280.000,'pepenix-21')

use hito2
create table empresa
(
    id_empresa varchar (30) primary key,
    id_compra varchar (25),
    id_vehiculos varchar(30),

    foreign key (id_compra) references compra(id_compra),
    foreign key (id_vehiculos) references vehiculos (id_vehiculos)
);
insert into empresa(id_empresa,id_compra,id_vehiculos)
values('NOXO-21','COM-1','v-1')
insert into empresa(id_empresa,id_compra,id_vehiculos)
values('NOXO-321','COM-2','v-2')
--terminamos
```