

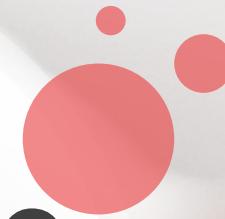
BASE DE DATOS I

INGENIERIA
DE SISTEMAS

Tarea HITO2

Base de Datos I
Unifranz
Sede el Alto
Hito2

Jamil Alex Quispe Valencia



Manejo de conceptos



1. ¿QUÉ SON LAS BASES DE DATOS?

Las bases de datos son conjuntos organizados de información estructurada que se almacenan en formato electrónico y se pueden acceder, gestionar y actualizar de manera eficiente.



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

Cuando se habla de bases de datos relacionales, se refiere a un tipo de base de datos en la que los datos se almacenan en tablas con filas y columnas relacionadas entre sí mediante claves o identificadores comunes. Estas relaciones permiten consultas y operaciones avanzadas.



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

El modelo entidad-relación (ER) es una representación gráfica de la estructura de una base de datos que muestra las entidades (objetos o conceptos) y las relaciones entre ellas. Un diagrama entidad-relación es una visualización de ese modelo.



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

En un diagrama entidad-relación, las figuras principales son:

- **Entidad:** Representa un objeto o concepto, generalmente se muestra como un rectángulo.
- **Relación:** Representa una conexión o vínculo entre entidades, generalmente se muestra como un rombo.
- **Atributo:** Representa una propiedad o característica de una entidad y se muestra como un óvalo unido a la entidad por una línea.

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

SQL Server es un sistema de gestión de bases de datos (DBMS) desarrollado por Microsoft. SQL Server Management Studio (SSMS) es una herramienta de interfaz gráfica que se utiliza para administrar y trabajar con bases de datos SQL Server.



6. ¿ CÓMO SE CREA UNA BASE DE DATOS?

Para crear una base de datos en SQL Server, se utiliza la sentencia SQL CREATE DATA BASE seguida del nombre de la base de datos que se desea crear.



7. ¿PARA QUÉ SIRVE EL COMANDO USE?

El comando **USE** se utiliza para cambiar el contexto de la base de datos actual en SQL Server. Permite seleccionar la base de datos con la que se desea trabajar, lo que significa que todas las consultas y operaciones posteriores se aplicarán a esa base de datos.



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

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



9. INSERTAR 3 REGISTROS A LA TABLA CREADA ANTERIORMENTE

```
INSERT INTO MiTabla (ID, Nombre, Edad)  
VALUES (1, 'Juan', 30);  
  
INSERT INTO MiTabla (ID, Nombre, Edad)  
VALUES (2, 'María', 25);  
  
INSERT INTO MiTabla (ID, Nombre, Edad)  
VALUES (3, 'Pedro', 35);
```

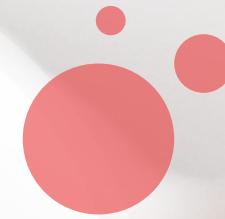


10.¿CÓMO SE ELIMINA UNA TABLA?

Para eliminar una tabla en SQL Server, se utiliza la sentencia SQL DROP TABLE seguida del nombre de la tabla que se desea eliminar. Por ejemplo:

```
DROP TABLE MiTabla;
```





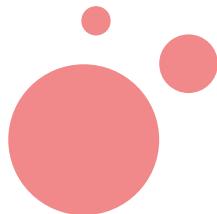
Parte Práctica





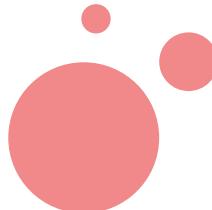
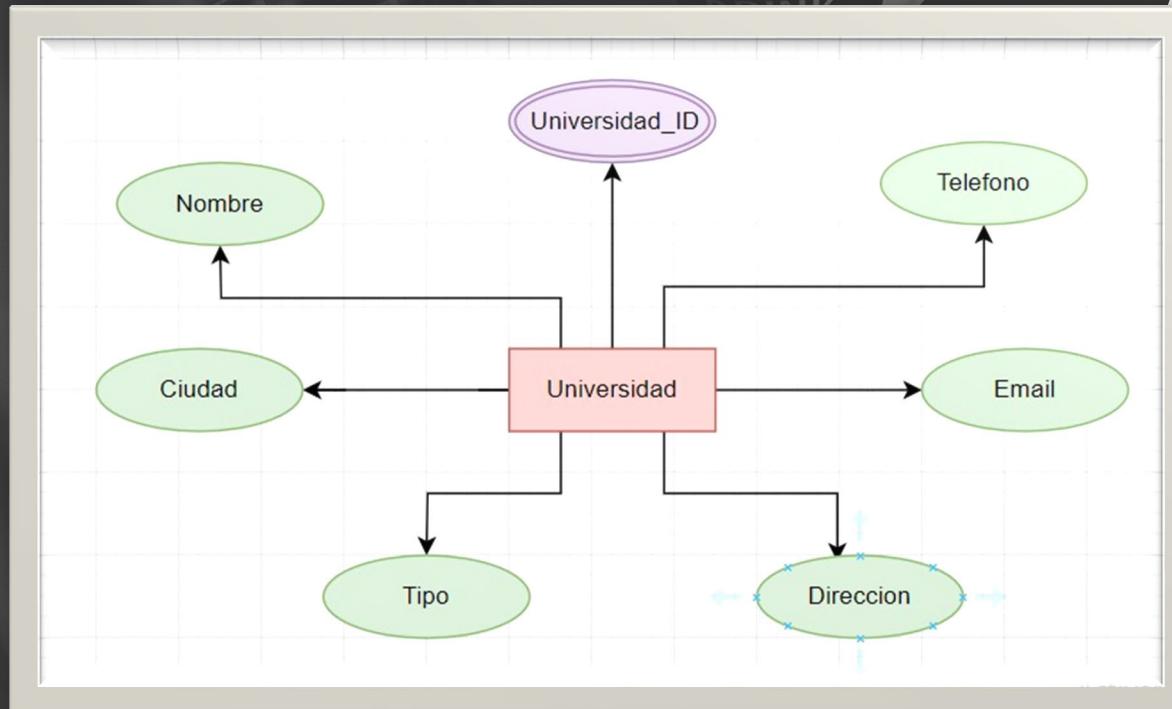
11.Crear el diseño para una UNIVERSIDAD

Universidad_ID (Integer, Llave Primaria y unica)
Nombre (varchar = string)
Ciudad (varchar = string)
Tipo (varchar = string)
Direccion (varchar = string)
Email (varchar = string)
Telefono (integer)





12. Crear el diagrama Entidad Relación E-R para el ejercicio anterior.



13.Crear la tabla universidad en base al diseño anterior

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer pane displays the server structure under 'DESKTOP-5J6J26P\SQLEXPRESS (SQL Server)'. The 'Databases' node is selected. On the right, the 'SQLQuery1.sql - DE...-5J6J26P\Alex (61)*' tab is active, showing the following SQL code:

```
--CREAMOS LA BASE DE DATOS DE HITO2
CREATE DATABASE Hito2Tarea;

--Creamos la tabla Universidad en la base de datos Hito2Tarea
USE Hito2Tarea;
CREATE TABLE Universidad
(
    Universidad_ID INT PRIMARY KEY,
    Nombre VARCHAR(100),
    Ciudad VARCHAR(20),
    Tipo VARCHAR(10),
    Direccion VARCHAR(50),
    Email VARCHAR(30),
    Telefono INTEGER NOT NULL
);
--DROP TABLE nos permitira borrar la tabla que creamos
```



14. Agregar registros a la tabla creada anteriormente.

```
--Agregamos registros a la tabla Universidad
INSERT INTO Universidad (Universidad_ID, Nombre, Ciudad, Tipo, Direccion, Email, Telefono)
VALUES
(010101, 'UNIFRANZ', 'El Alto', 'Privada', 'Avenida del Aeropuerto Internacional', 'unifranz@unifranz.edu.bo', '712502211'),
(022200, 'EMI', 'La Paz', 'Privada', 'Avenida Rafael Pavon', 'islaedicion@adm.emi.edu.bo', '2793144'),
(300303, 'UMSA', 'La Paz', 'Publica', 'Avenida Arce', 'diresi@umsa.bo', '22441963'),
(040440, 'UNIVALLE', 'La Paz', 'Privada', 'Avenida Argentina', 'univalle@univalle.edu.', '22001800');

SELECT * FROM Universidad
```

Tabla

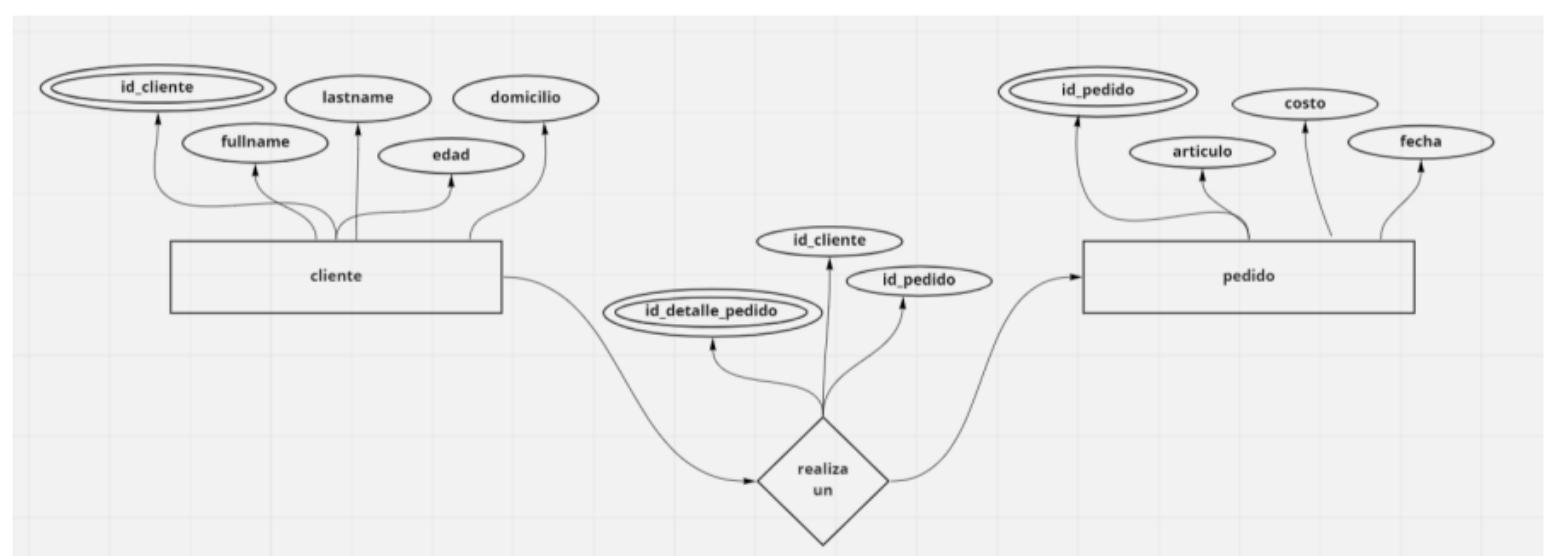


A screenshot of a Microsoft SQL Server Management Studio (SSMS) window showing a table named 'Results'. The table contains four rows of data about universities. The columns are: Universidad_ID, Nombre, Ciudad, Tipo, Direccion, Email, and Telefono. The data is as follows:

	Universidad_ID	Nombre	Ciudad	Tipo	Direccion	Email	Telefono
1	10101	UNIFRANZ	El Alto	Privada	Avenida del Aeropuerto Internacional	unifranz@unifranz.edu.bo	712502211
2	22200	EMI	La Paz	Privada	Avenida Rafael Pavon	islaedicion@adm.emi.edu.bo	2793144
3	40440	UNIVALLE	La Paz	Privada	Avenida Argentina	univalle@univalle.edu.	22001800
4	300303	UMSA	La Paz	Publica	Avenida Arce	diresi@umsa.bo	22441963

15.Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER

○ Diseno.





Object Explorer

Connect ▾

- DESKTOP-5J6J26P\SQLEXPRESS (SQL Server)
- Databases
- System Databases
- Database Snapshots
- Hito?Tarea

SQLQuery7.sql - DE...-5J6J26P\Alex (52)* SQLQuery1.sql - not connected*

```
-- Creamos la base de datos POLLOS_COPA
CREATE DATABASE POLLOS_COPA;

-- Usamos la base de datos POLLOS_COPA
USE POLLOS_COPA;
```

```
-- Creamos la tabla cliente
CREATE TABLE cliente
(
    id_cliente INT PRIMARY KEY,
    lastname VARCHAR(50) NOT NULL,
    fullname VARCHAR(50) NOT NULL,
    domicilio VARCHAR(50),
    edad INT
);
--para borrar la tabla cliente
DROP TABLE cliente;
```

```
-- Creamos la tabla pedido
CREATE TABLE pedido
(
    id_pedido INT PRIMARY KEY,
    costo DECIMAL(10,2) NOT NULL,
    articulo VARCHAR(50) NOT NULL,
    fecha DATE NOT NULL
);
DROP TABLE pedido;
```

```
CREATE TABLE detalle_pedido
(
    id_detalle_pedido INT PRIMARY KEY,
    realiza_un INT NOT NULL,
    FOREIGN KEY (realiza_un) REFERENCES cliente(id_cliente),
    contiene_un INT NOT NULL,
    FOREIGN KEY (contiene_un) REFERENCES pedido(id_pedido)
);
DROP TABLE detalle_pedido;
```

```
-- Insertar dos registros en la tabla pedido
INSERT INTO pedido (id_pedido, costo, articulo, fecha)
VALUES (1, 15.00, 'Pollo a la brasa', '2023-09-13'),
       (2, 20.00, 'Pollo al horno', '2023-09-14');

-- Insertar dos registros en la tabla detalle_pedido
INSERT INTO detalle_pedido (id_detalle_pedido, realiza_un, contiene_un)
VALUES (1, 1, 1),
       (2, 2, 2);

SELECT * FROM cliente;
SELECT * FROM pedido;
SELECT * FROM detalle_pedido
```

Tabla

The screenshot shows a database interface with three tables displayed in a grid:

	id_cliente	lastname	fullname	domicilio	edad
1	1	Perez	Juan	Av. Bolivia 123	25
2	2	Garcia	Maria	Calle Sucre 456	30

	id_pedido	costo	articulo	fecha
1	1	15.00	Pollo a la brasa	2023-09-13
2	2	20.00	Pollo al homo	2023-09-14

	id_detalle_pedido	realiza_un	contiene_un
1	1	1	1
2	2	2	2



Este seria el código

—Creamos la base de datos POLLOS_COPA

```
CREATE DATABASE POLLOS_COPA;
```

■ Usamos la base de datos POLLOS_COPAUSE POLLOS_COPA;

■ -- Creamos la tabla clienteCREATE TABLE cliente (id_cliente
INT PRIMARY KEY, lastname VARCHAR(50) NOT NULL,
fullname VARCHAR(50) NOT NULL, domicilio VARCHAR(50),
edad INT);--para borrar la tabla clienteDROP TABLE cliente;--

Creamos la tabla pedidoCREATE TABLE pedido (id_pedido
INT PRIMARY KEY, costo DECIMAL(10,2) NOT NULL,
articulo VARCHAR(50) NOT NULL, fecha DATE NOT
NULL);DROP TABLE pedido;-- Creamos la tabla

detalle_pedidoCREATE TABLE detalle_pedido
(id_detalle_pedido INT PRIMARY KEY, realiza_un INT NOT
NULL, FOREIGN KEY (realiza_un) REFERENCES
cliente(id_cliente), contiene_un INT NOT NULL, FOREIGN
KEY (contiene_un) REFERENCES pedido(id_pedido));DROP

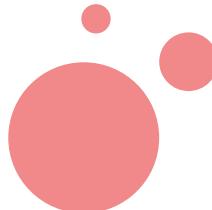
TABLE detalle_pedido;-- Insertar dos registros en la tabla

clienteINSERT INTO cliente (id_cliente, lastname, fullname,
domicilio, edad)VALUES (1, 'Perez', 'Juan', 'Av. Bolivia 123', 25),
(2, 'Garcia', 'Maria', 'Calle Sucre 456', 30);-- Insertar dos

registros en la tabla pedidoINSERT INTO pedido (id_pedido,
costo, articulo, fecha)VALUES (1, 15.00, 'Pollo a la brasa', '2023-
09-13'), (2, 20.00, 'Pollo al horno', '2023-09-14');-- Insertar dos

registros en la tabla detalle_pedidoINSERT INTO detalle_pedido
(id_detalle_pedido, realiza_un, contiene_un)VALUES (1, 1, 1),
(2, 2, 2);SELECT * FROM cliente;SELECT * FROM

pedido;SELECT * FROM detalle_pedido



```
-- Creamos la base de datos POLLOS_COPA
CREATE DATABASE POLLOS_COPA;

-- Usamos la base de datos POLLOS_COPA
USE POLLOS_COPA;

-- Creamos la tabla cliente
CREATE TABLE clientes
(
    id_cliente INT PRIMARY KEY,
    lastname VARCHAR(50) NOT NULL,
    fullname VARCHAR(50) NOT NULL,
    domicilio VARCHAR(50),
    edad INT
);
--para borrar la tabla cliente
DROP TABLE clientes;

-- Creamos la tabla pedido
CREATE TABLE pedidos
(
    id_pedido INT PRIMARY KEY,
    costo DECIMAL(10,2) NOT NULL,
    articulo VARCHAR(50) NOT NULL,
    fecha DATE NOT NULL
);
DROP TABLE pedidos;
-- Creamos la tabla detalle_pedido

CREATE TABLE detalle_pedido
(
    id_detalle_pedido INT PRIMARY KEY,
    realiza_un INT NOT NULL,
    FOREIGN KEY (realiza_un) REFERENCES clientes(id_cliente),
    contiene_un INT NOT NULL,
    FOREIGN KEY (contiene_un) REFERENCES pedidos(id_pedido)
);
DROP TABLE detalle_pedido;

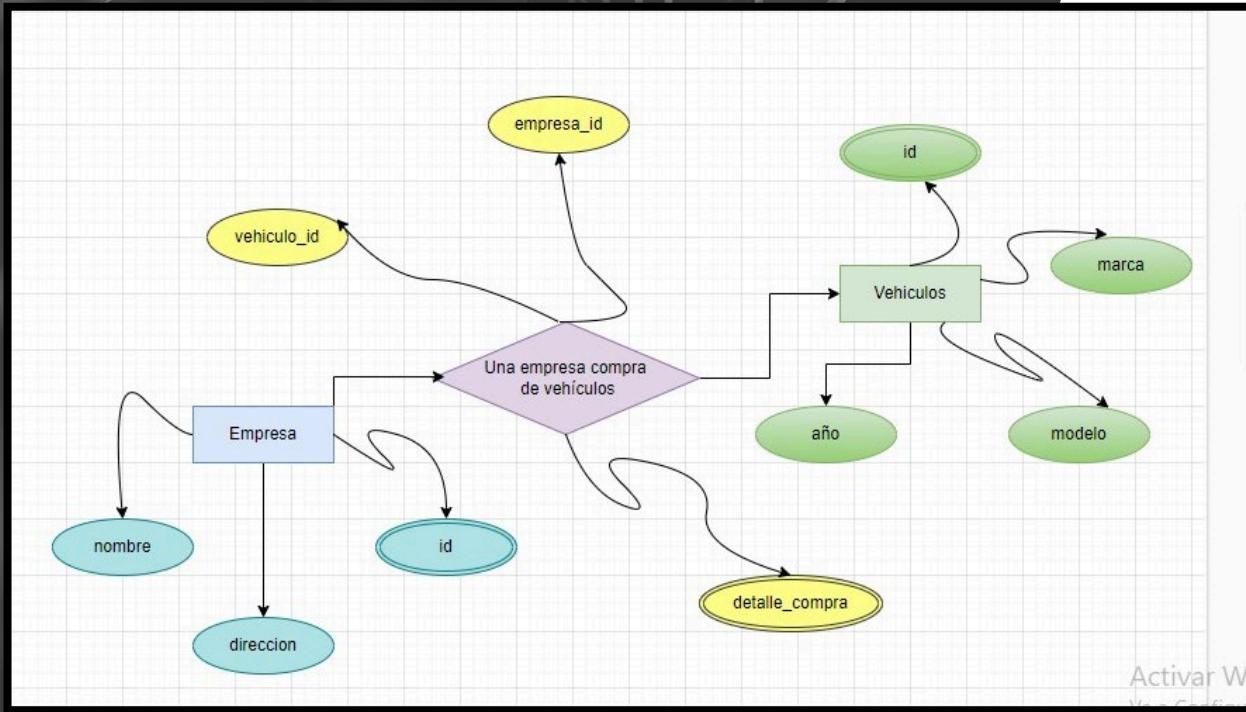
-- Insertar dos registros en la tabla clientes
INSERT INTO clientes (id_cliente, lastname, fullname, domicilio, edad)
VALUES (1, 'Perer', 'Juan', 'Av. Bolivia 123', 25),
       (2, 'Garcia', 'Maria', 'Calle Sucre 456', 30);

-- Insertar dos registros en la tabla pedido
INSERT INTO pedidos (id_pedido, costo, articulo, fecha)
VALUES (1, 15.00, 'Pollo a la brasa', '2023-09-17'),
       (2, 20.00, 'Pollo al horno', '2023-09-14');

-- Insertar dos registros en la tabla detalle_pedido
INSERT INTO detalle_pedido (id_detalle_pedido, realiza_un, contiene_un)
VALUES (1, 1, 1),
       (2, 2, 2);

SELECT * FROM clientes;
SELECT * FROM pedidos;
SELECT * FROM detalle_pedido;
```

16. Crear el modelo entidad relación ER y su código SQL.



Activar W
y Clicke

```
-- CREAMOS LA BASE DE DATOS una_empresa_compra_vehiculos
CREATE DATABASE una_empresa_compra_vehiculos;

-- Creamos las tablas en la base de datos una_empresa_compra_vehiculos
USE una_empresa_compra_vehiculos;

-- Creamos la tabla empresa
CREATE TABLE empresa
(
    id INT PRIMARY KEY,
    nombre VARCHAR(25) NOT NULL,
    direccion VARCHAR(25)
);

drop table empresa
-- Creamos la tabla vehiculos

CREATE TABLE vehiculos
(
    id INT PRIMARY KEY,
    marca VARCHAR(25) NOT NULL,
    modelo VARCHAR(25),
    año INT
);
```

```
-- Creamos la tabla detalle_compra
CREATE TABLE detalle_compra
(
    id INT PRIMARY KEY,
    empresa_id INT,
    vehiculo_id INT,
    fecha_compra DATE,
    precio DECIMAL(10, 2),
    FOREIGN KEY (empresa_id) REFERENCES empresa(id),
    FOREIGN KEY (vehiculo_id) REFERENCES vehiculos(id)
);
```

```
);

-- Insertar datos en la tabla empresa
INSERT INTO empresa (id, nombre, direccion)
VALUES
    (1, 'Empresa A', 'Calle Principal 123'),
    (2, 'Empresa B', 'Avenida Secundaria 456'),
    (3, 'Empresa C', 'Calle del Centro 789');

-- Insertar datos a tabla vehiculos
INSERT INTO vehiculos (id, marca, modelo, año)
VALUES
    (1, 'Toyota', 'Corolla', 2022),
    (2, 'Ford', 'F-150', 2021),
    (3, 'Honda', 'Civic', 2023);

-- Insertar datos tabla detalle_compra

INSERT INTO detalle_compra (id, empresa_id, vehiculo_id, fecha_compra, precio)
VALUES
    (1, 1, 1, '2023-01-15', 25000.00),
    (2, 2, 2, '2023-02-20', 35000.00),
    (3, 3, 3, '2023-03-25', 28000.00);

SELECT * FROM empresa;
SELECT * FROM vehiculos;
SELECT * FROM detalle_compra;
```

Código



CREAMOS LA BASE DE DATOS una_empresa_compra_vehículos

CREATE DATABASE una_empresa_compra_vehiculos;

Creamos las tablas en la base de datos una_empresa_compra_vehículos

USE una_empresa_compra_vehiculos;

Creamos la tabla empresaCREATE TABLE empresa (id INT PRIMARY KEY, nombre VARCHAR(25) NOT NULL, direccion VARCHAR(25));drop table empresa

-- Creamos la tabla vehiculosCREATE TABLE vehiculos(id INT PRIMARY KEY, marca VARCHAR(25) NOT NULL, modelo VARCHAR(25), año INT);-- Creamos la tabla detalle_compra CREATE TABLE detalle_compra (id INT PRIMARY KEY, empresa_id INT, vehiculo_id INT , fecha_compra DATE, precio DECIMAL(10, 2), FOREIGN KEY (empresa_id) REFERENCES Empresa(id), FOREIGN KEY (vehiculo_id) REFERENCES vehiculos(id));-- Insertar datos en la tabla empresasINSERT INTO empresa (id, nombre, direccion)VALUES (1, 'Empresa A', 'Calle Principal 123'), (2, 'Empresa B', 'Avenida Secundaria 456'), (3, 'Empresa C', 'Calle del Centro 789');

-- Insertar datos a tabla vehiculosINSERT INTO vehiculos (id, marca, modelo, año)VALUES (1, 'Toyota', 'Corolla', 2022), (2, 'Ford', 'F-150', 2021), (3, 'Honda', 'Civic', 2023);

-- Insertar datos tabla detalle_compraINSERT INTO detalle_compra (id, empresa_id, vehiculo_id, fecha_compra, precio)VALUES (1, 1, 1, '2023-01-15', 25000.00), (2, 2, 2, '2023-02-20', 35000.00), (3, 3, 3, '2023-03-25', 28000.00);

LECT * FROM empresa; SELECT * FROM vehiculos; SELECT * FROM detalle_compra

Tabla

The screenshot shows a database interface with three tables displayed in tabs:

- Results** tab: Contains the data from the three tables.
- Messages** tab: Empty.

Table 1: empresas

	id	nombre	direccion
1	1	Empresa A	Calle Principal 123
2	2	Empresa B	Avenida Secundaria 456
3	3	Empresa C	Calle del Centro 789

Table 2: vehiculos

	id	marca	modelo	año
1	1	Toyota	Corolla	2022
2	2	Ford	F-150	2021
3	3	Honda	Civic	2023

Table 3: compras

	id	empresa_id	vehiculo_id	fecha_compra	precio
1	1	1	1	2023-01-15	25000.00
2	2	2	2	2023-02-20	35000.00
3	3	3	3	2023-03-25	28000.00



00 2.0)
(DEA)

A close-up photograph of a person's hands holding a smartphone. The screen of the phone displays a complex network diagram consisting of several hexagonal nodes connected by white lines, resembling a molecular or network structure. Below the diagram, a terminal window shows the command "# nmap KA -T4" followed by the output of an Nmap scan. The output includes the host status ("HR5t is up"), its IP address ("rDNS rZcord for 74.207.GN1.2"), and open ports ("PORTS CXZTA SERVXCE 22/tcp open ssh").

nmap KA -T4

Nmap scan report for scanme.nmap.org (74.207.69.2)

HR5t is up (0.022s latency).

rDNS rZcord for 74.207.69.2

PORTS CXZTA SERVXCE

22/tcp open ssh