

The background is a gradient of dark blue and purple, transitioning from a lighter purple at the top to a darker blue at the bottom. It is decorated with several faint, white, circular patterns. On the left side, there are concentric circles with radial lines, resembling a compass or a clock face, with numbers like 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, and 260. Other circles with arrows and dashed lines are scattered across the image, creating a sense of motion and data flow. The overall aesthetic is technical and futuristic.

# PRESENTACIÓN DE BASE DE DATOS

POR KENNETH OMAR MAMANI ZE.

# Manejo de conceptos

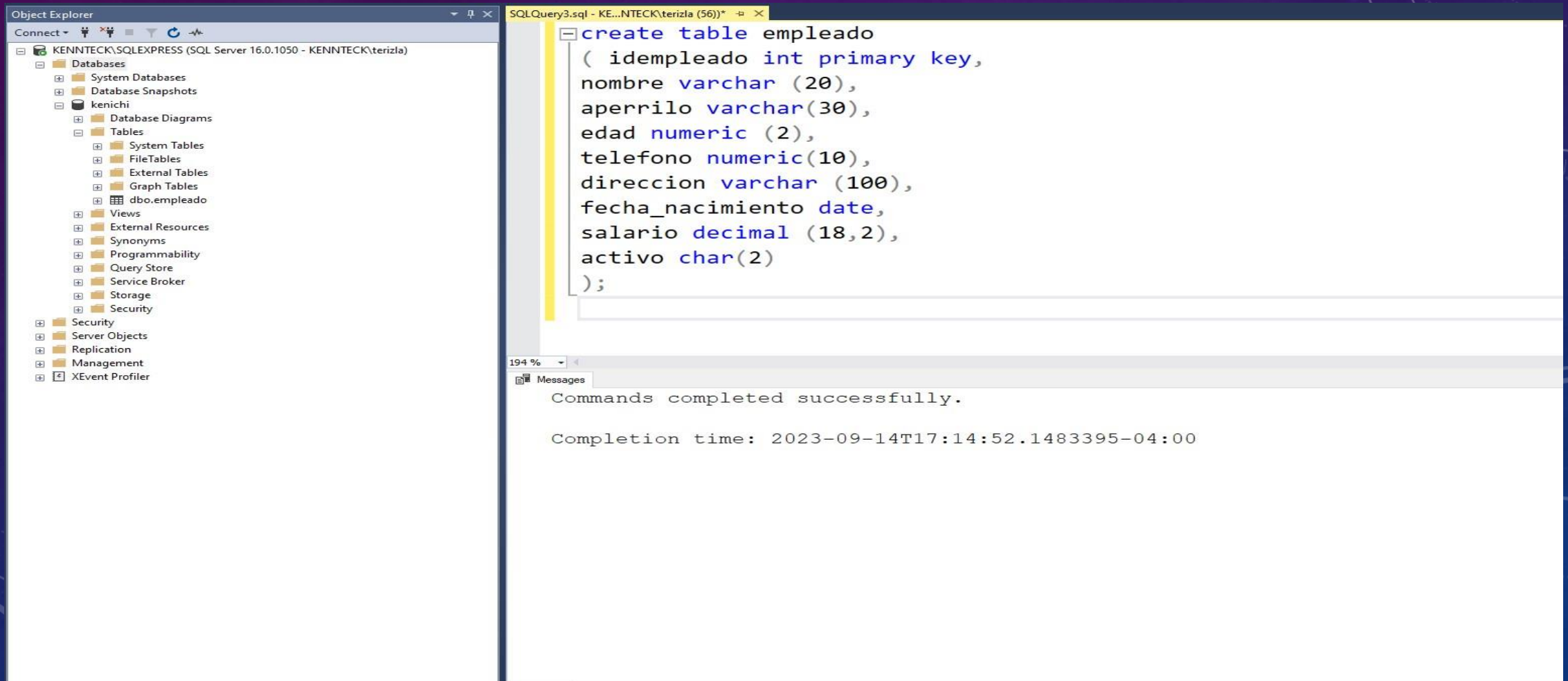
- ¿Qué son las bases de datos?
- Las bases de datos son conjuntos organizados de datos que se almacenan y gestionan electrónicamente. Proporcionan una forma eficiente de almacenar, recuperar, actualizar y gestionar información.
- ¿A qué se refiere cuando se habla de bases de datos relacionales?
- Las bases de datos relacionales son un tipo de base de datos que organiza los datos en tablas relacionadas. Cada tabla contiene registros con información y se pueden establecer relaciones entre las tablas a través de claves primarias y foráneas.
- ¿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 se utiliza para representar la estructura de la base de datos de manera gráfica. Un diagrama entidad-relación muestra entidades (tablas), atributos (columnas) y relaciones entre las entidades.

- ¿Cuáles son las figuras que representan a un diagrama entidad-relación? Explique cada una de ellas.
- Las figuras en un diagrama ER incluyen:
- Rectángulos: Representan las entidades (tablas).
- Elipses o círculos: Representan los atributos (columnas).
- Líneas: Representan las relaciones entre las entidades.
- Rombos: Representan atributos multivaluados.
- Diamantes: Representan relaciones de muchos a muchos.
- ¿Qué es SQL Server y qué es SQL Server Management Studio?
- SQL Server es un sistema de gestión de bases de datos relacionales desarrollado por Microsoft. SQL Server Management Studio (SSMS) es una herramienta de administración y desarrollo que se utiliza para interactuar con bases de datos SQL Server. Permite crear, gestionar y consultar bases de datos.



- ¿Cómo se crea una base de datos?
- En SQL Server, puedes crear una base de datos utilizando el comando SQL `CREATE DATABASE nombre_de_la_base_de_datos;`
- ¿Para qué sirve el comando USE?
- El comando USE se utiliza para seleccionar una base de datos específica con la que deseas trabajar. Por ejemplo, `USE_nombre_de_la_base_de_datos;` te permite cambiar el contexto para que los comandos posteriores se ejecuten en esa base de datos.

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



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows the database structure for 'KENNTECK\SQLSERVER (SQL Server 16.0.1050 - KENNTECK\terizla)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', and 'kenichi'. Under 'kenichi', the 'Tables' folder is expanded, showing 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', and 'dbo.empleado'. The 'dbo.empleado' table is selected. On the right, the 'SQLQuery3.sql' window shows the following SQL code:

```
create table empleado
( idempleado int primary key,
  nombre varchar (20),
  aperrilo varchar(30),
  edad numeric (2),
  telefono numeric(10),
  direccion varchar (100),
  fecha_nacimiento date,
  salario decimal (18,2),
  activo char(2)
);
```

Below the code editor, the 'Messages' pane shows the following output:

```
Commands completed successfully.

Completion time: 2023-09-14T17:14:52.1483395-04:00
```

# INSERTE 3 REGISTROS A LA TABLE CREADA ANTERIORMENTE

KENNTTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTTECK\terizla)

- Databases
  - System Databases
  - Database Snapshots
  - kenichi
    - Database Diagrams
    - Tables
      - System Tables
      - FileTables
      - External Tables
      - Graph Tables
      - dbo.empleado
        - Columns
          - idempleado (PK, int, not null)
          - nombre (varchar(20), null)
          - aperrilo (varchar(30), null)
          - edad (numeric(2,0), null)
          - telefono (numeric(10,0), null)
          - direccion (varchar(100), null)
          - fecha\_nacimiento (date, null)
          - salario (decimal(18,2), null)
          - activo (char(2), null)
        - Keys
        - Constraints
        - Triggers
        - Indexes
        - Statistics
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Query Store
      - Service Broker
      - Storage
      - Security
    - Security
    - Server Objects
    - Replication
    - Management
    - XEvent Profiler

```
insert into empleado
values (1, 'jose', 'feliciano', 45, 70625784, 'calle nro 1',
'1980-07-02', 300000.12, 'si');

insert into empleado
values (2, 'Maria', 'Lopez', 35, 84221784, 'calle nro 4',
'1990-05-04', 300000.12, 'si');

insert into empleado
values (3, 'Kenneth', 'Zegarra', 28, 74272784, 'calle nro 23',
'1997-09-04', 300000.12, 'si');
```

194 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-09-14T17:40:37.9181809-04:00



PARA CORROBORAR QUE SE REGISTRO  
CORRECTAMENTE PONEMOS

The screenshot displays the SQL Server Enterprise Manager interface on the left and the SQL Query Editor on the right. The Enterprise Manager shows the database structure for 'KENNTECK\SQLSERVER (SQL Server 16.0.1050 - KENNTECK\terizla)', including 'Databases', 'kenichi', 'Tables', and 'dbo.empleado'. The Query Editor shows the SQL query 'select \* from empleado;' and the results table.

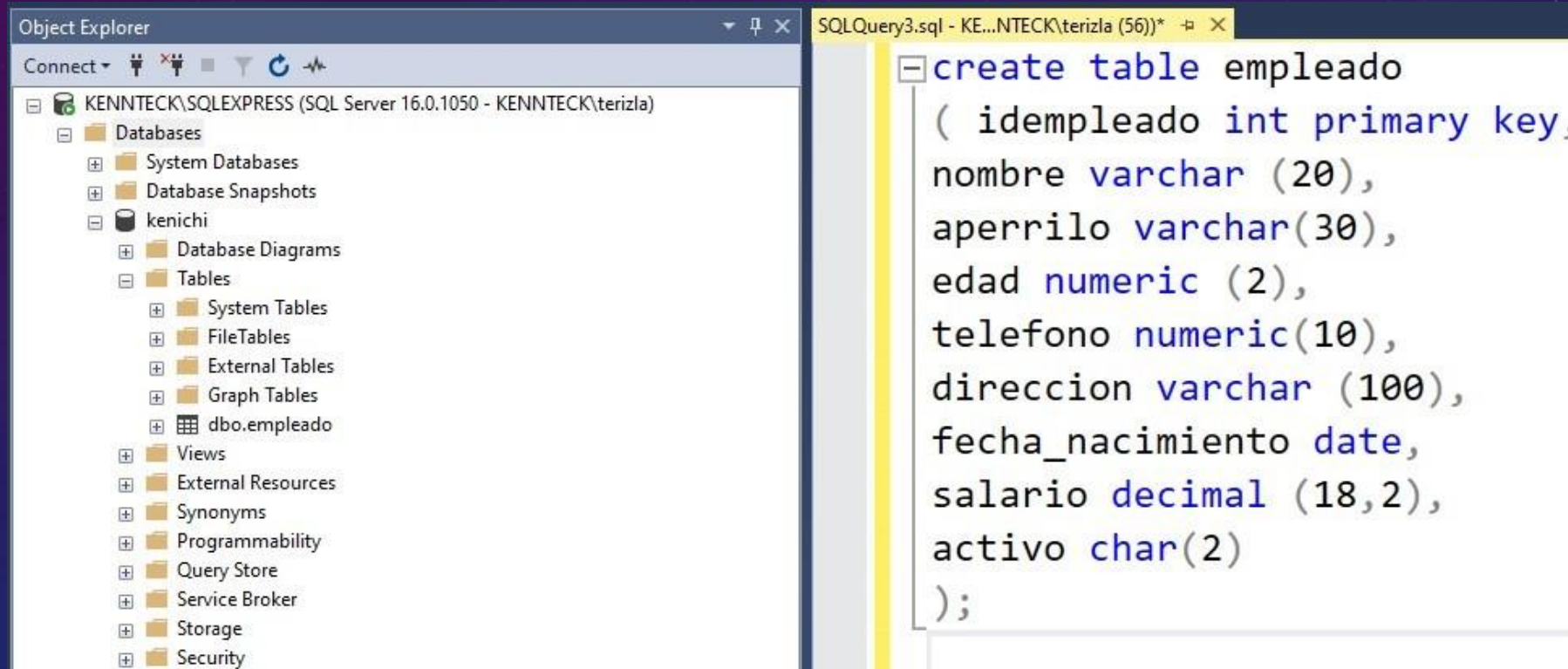
	idempleado	nombre	apellido	edad	telefono	direccion	fecha_nacimiento	salario	activo
1	1	jose	feliciano	45	70625784	calle nro 1	1980-07-02	300000.12	si
2	2	Maria	Lopez	35	84221784	calle nro 4	1990-05-04	300000.12	si
3	3	Kenneth	Zegarra	28	74272784	calle nro 23	1997-09-04	300000.12	si

# COMO SE ELIMINA UNA TABLA ?

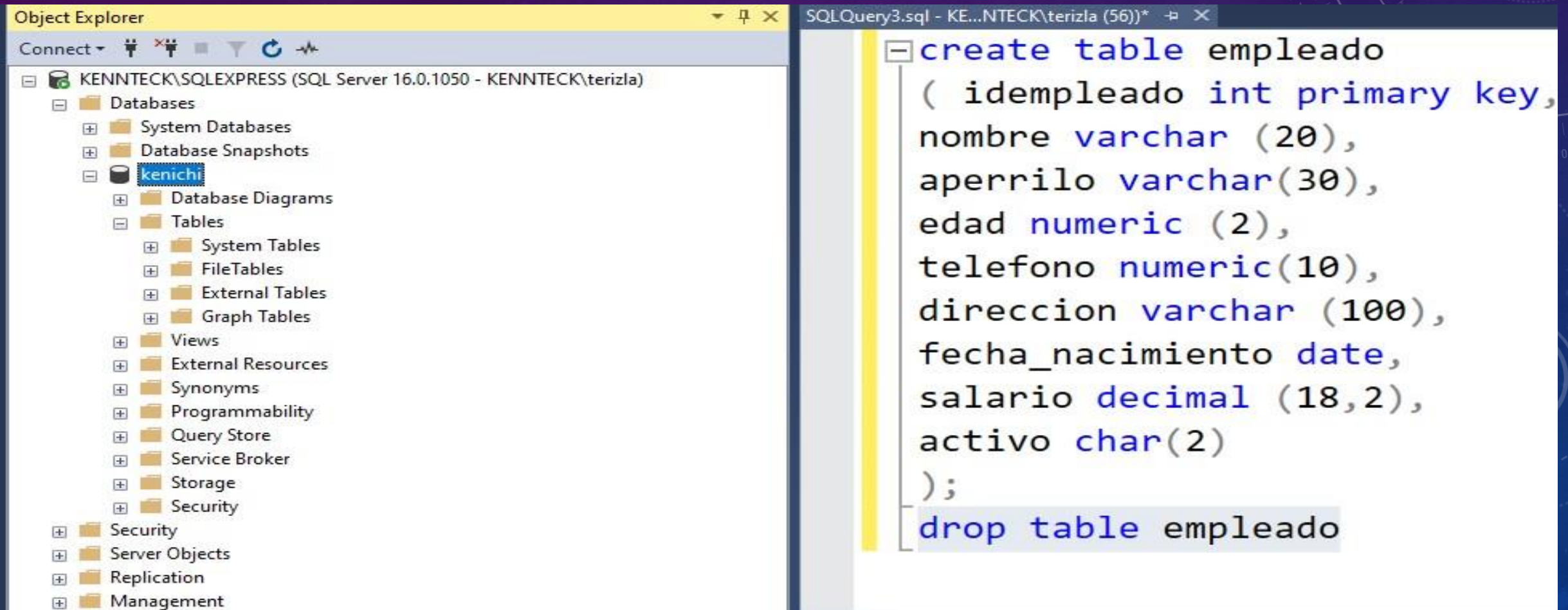
- Puedes eliminar una tabla utilizando el comando DROP TABLE (nombre de la tabla)



# TENEMOS CREADA LA TABLA EMPLEADO



# AQUÍ VEREMOS YA COMO SE ELIMINA CON DROP TABLE EL\_NOMBRE\_DE\_LA\_TABLA



The image shows a screenshot of a SQL Server environment. On the left, the 'Object Explorer' window displays the database structure for 'KENNTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTECK\terizla)'. The 'Databases' folder is expanded, showing 'kenichi' as the selected database. Under 'kenichi', the 'Tables' folder is expanded, listing various system and user tables. On the right, the 'SQLQuery3.sql' window shows the following SQL code:

```
create table empleado  
( idempleado int primary key,  
  nombre varchar (20),  
  aperrilo varchar(30),  
  edad numeric (2),  
  telefono numeric(10),  
  direccion varchar (100),  
  fecha_nacimiento date,  
  salario decimal (18,2),  
  activo char(2)  
);  
drop table empleado
```

# CREAR EL DISEÑO PARA UNIVERSIDAD

## Universidad

ID\_universidad INT **PRIMARY KEY**

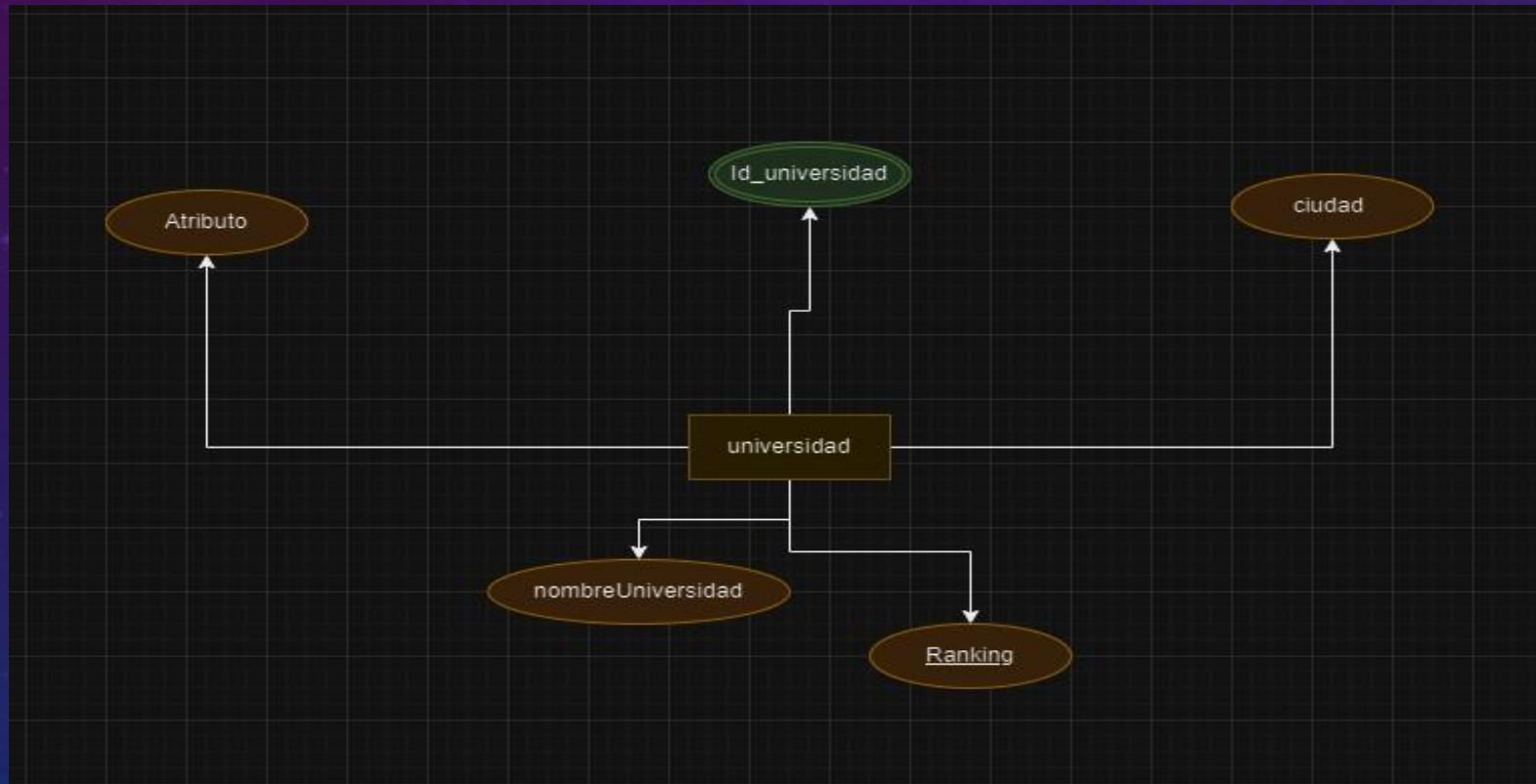
NombreUniversidad **VARCHAR(100)**

Ciudad **VARCHAR(50)**

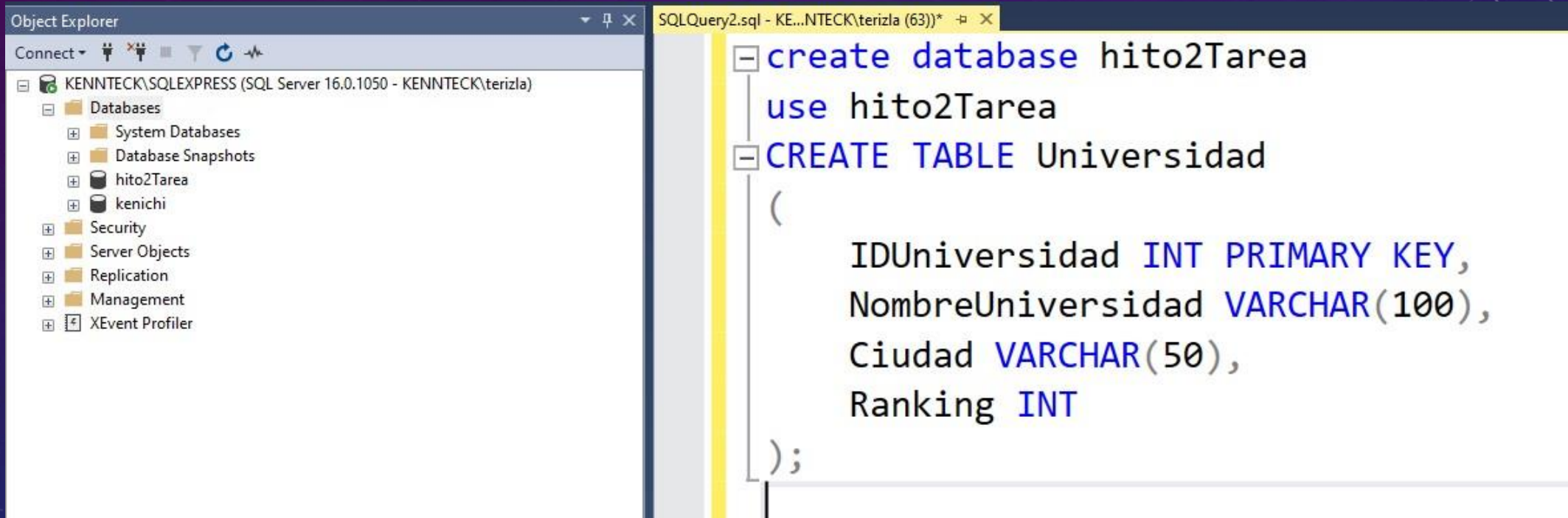
Ranking **INT**



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



# CREAR LA TABLA UNIVERSIDAD EN BASE AL DISEÑO ANTERIOR.



AGREGAR REGISTROS A LA TABLA CREADA ANTERIORMENTE.  
DATO EL RANKIN SE SACARON DE LA PAGINA:  
[HTTPS://WWW.UNIVERSITYGURU.COM/ES/UNIVERSIDADES-  
LA-PAZ](https://www.universityguru.com/es/universidades-la-paz)

The screenshot displays the SQL Server Enterprise Manager interface on the left and a SQL Query window on the right. The Enterprise Manager shows the database structure for 'KENNTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTECK\terizla)'. The 'dbo.Universidad' table is expanded, showing columns: IDUniversidad (PK, int, not null), NombreUniversidad (varchar(100), null), Ciudad (varchar(50), null), and Ranking (int, null). The SQL Query window contains the following SQL script:

```
insert into Universidad
values (456487481,'University of San Andrés','la paz ',1);
insert into Universidad
values (984478018,'
Bolivian Catholic University','la paz',2);
insert into Universidad
values (874848169,'
Autonomous University of Baja California Sur','la paz',3);
select*from Universidad
```

Below the query, the 'Results' tab shows a table with 4 rows and 5 columns: IDUniversidad, NombreUniversidad, Ciudad, and Ranking. The data is as follows:

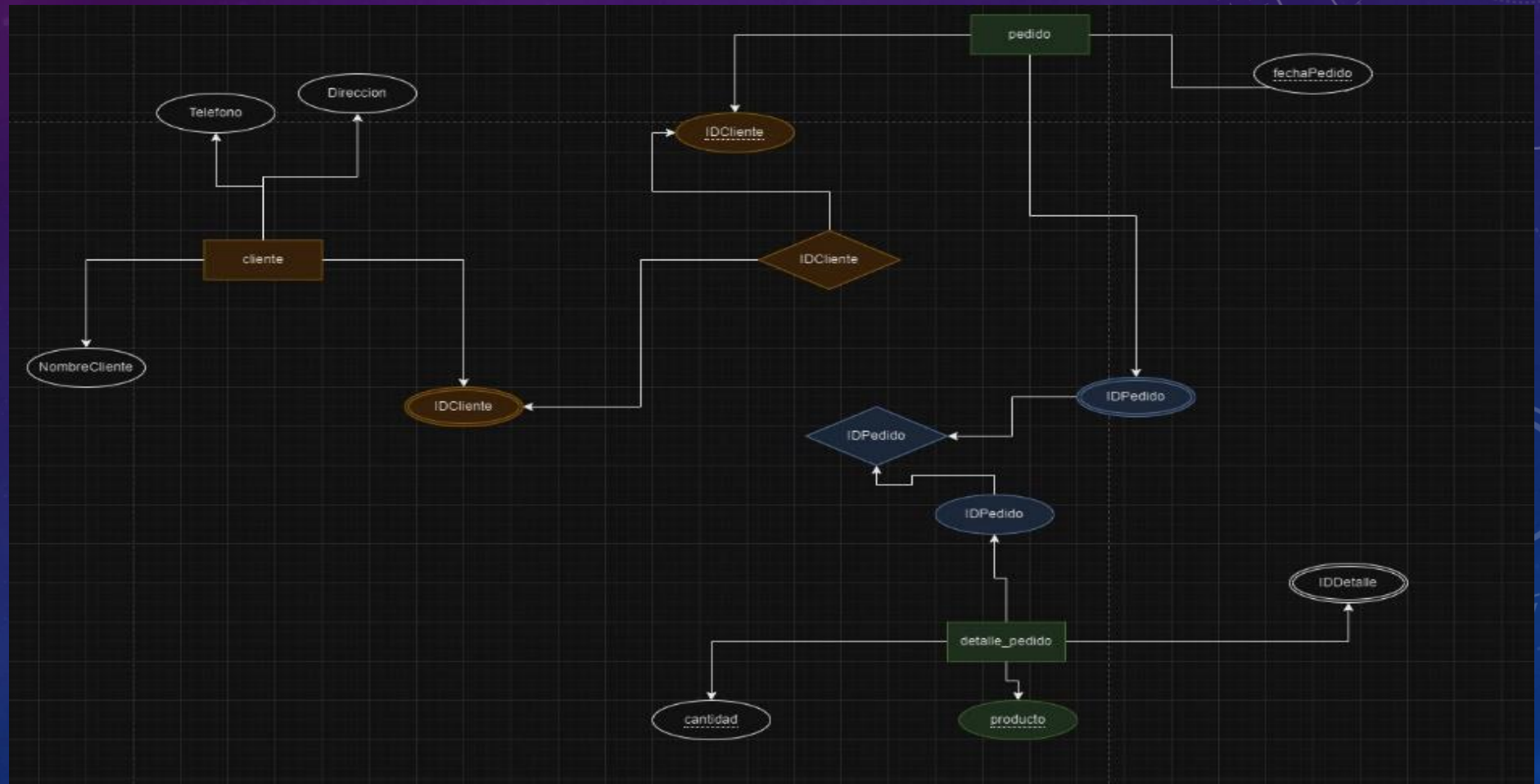
	IDUniversidad	NombreUniversidad	Ciudad	Ranking
1	4578018	unifranz	la paz el alto	5
2	456487481	University of San Andrés	la paz	1
3	874848169	Autonomous University of Baja California Sur	la paz	3
4	984478018	Bolivian Catholic University	la paz	2



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

Se **sugiere** crear una base de datos de nombre **POLLOS\_COPA** y en ella crear las tablas:

- **cliente**
- **detalle\_pedido**
- **pedido**



# ADJUNTAR EL CÓDIGO SQL GENERADO.

```
create database pollos_copa

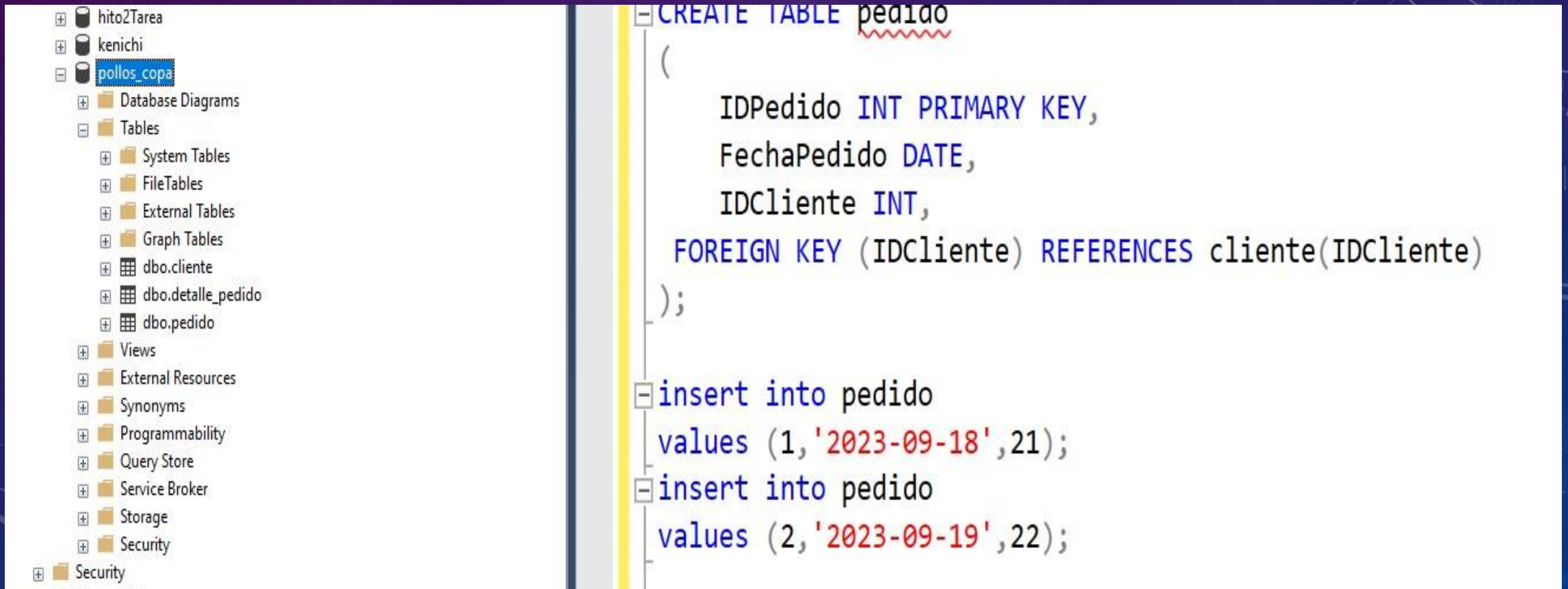
use pollos_copa

CREATE TABLE cliente
(
    IDCliente INT PRIMARY KEY,
    NombreCliente VARCHAR(100),
    Direccion VARCHAR(200),
    Telefono VARCHAR(15)
);

insert into cliente
values (21, 'juanita', 'calle lazaro puerta 6', 15487245);

insert into cliente
values (22, 'juan', 'calle laro puerta 8', 24874687);
```

# CREAMOS LAS DEMAS TABLAS Y LE DAMOS UNA RELACION ENTRE ELLAS



The image shows a screenshot of a SQL Server environment. On the left, the SQL Server Enterprise Manager tree is visible, showing the database structure. The database 'pollos\_copa' is selected, and the 'Tables' folder is expanded, showing the following tables:

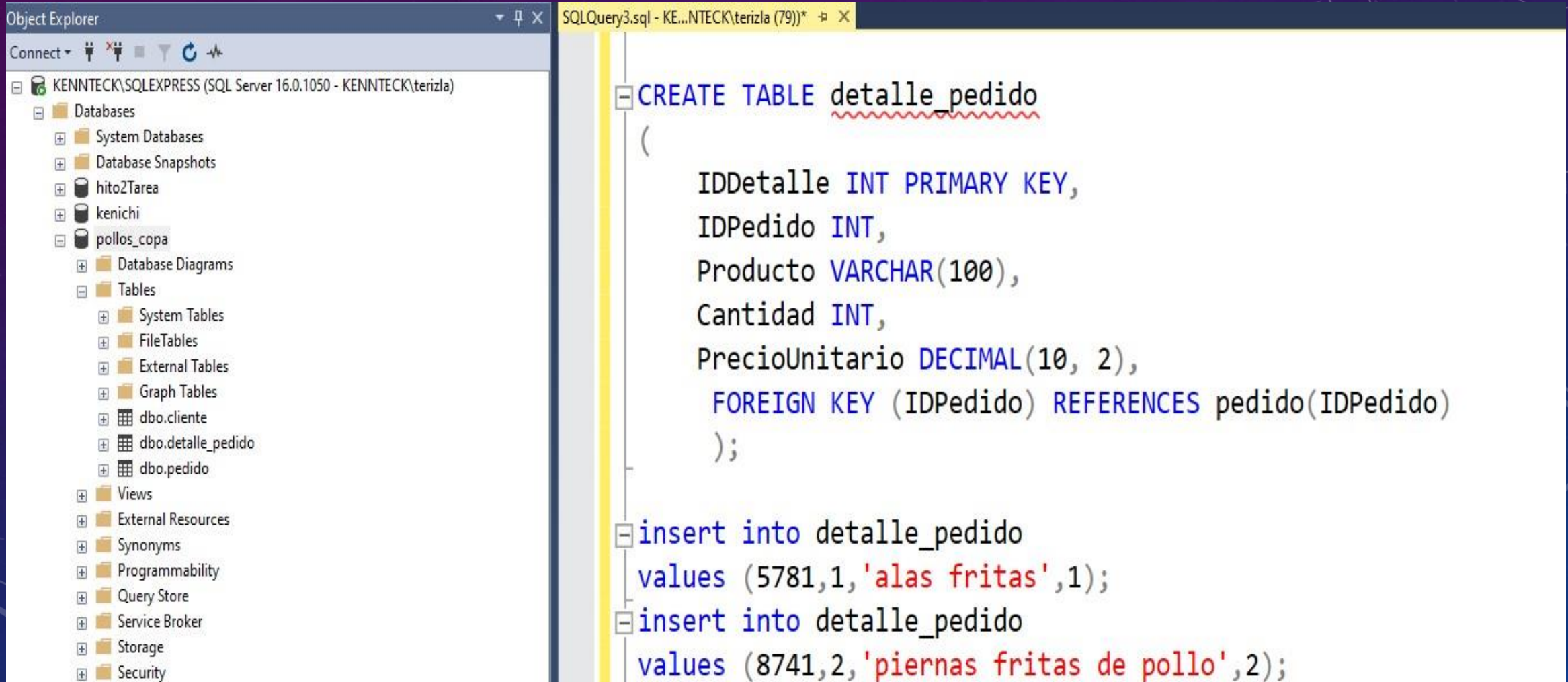
- dbo.cliente
- dbo.detalle\_pedido
- dbo\_pedido

On the right, the SQL Server Query Editor is open, showing the following SQL code:

```
CREATE TABLE pedido  
(  
    IDPedido INT PRIMARY KEY,  
    FechaPedido DATE,  
    IDCliente INT,  
    FOREIGN KEY (IDCliente) REFERENCES cliente(IDCliente)  
);  
  
insert into pedido  
values (1, '2023-09-18', 21);  
  
insert into pedido  
values (2, '2023-09-19', 22);
```



# LA ULTIMA TABLE



The screenshot displays the SQL Server Enterprise Manager interface on the left and the SQL Query Editor on the right. The Enterprise Manager shows the database structure for 'KENNTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTECK\terizla)', including databases, tables, and views. The 'detalle\_pedido' table is listed under the 'dbo' schema. The Query Editor shows the SQL script for creating and inserting data into this table.

**Object Explorer**

- KENNTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTECK\terizla)
  - Databases
    - System Databases
    - Database Snapshots
    - hito2Tarea
    - kenichi
    - pollos\_copa
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.cliente
        - dbo.detalle\_pedido
        - dbo.pedido
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Query Store
      - Service Broker
      - Storage
      - Security

**SQLQuery3.sql - KENNTECK\terizla (79)\*\***

```
CREATE TABLE detalle_pedido
(
    IDDetalle INT PRIMARY KEY,
    IDPedido INT,
    Producto VARCHAR(100),
    Cantidad INT,
    PrecioUnitario DECIMAL(10, 2),
    FOREIGN KEY (IDPedido) REFERENCES pedido(IDPedido)
);

insert into detalle_pedido
values (5781,1,'alas fritas',1);

insert into detalle_pedido
values (8741,2,'piernas fritas de pollo',2);
```

## PARA VER LOS DATOS

```
SELECT c.*, p.*, dp.*  
FROM cliente c  
JOIN pedido p ON c.IDCliente = p.IDCliente  
JOIN detalle_pedido dp ON p.IDPedido = dp.IDPedido;
```

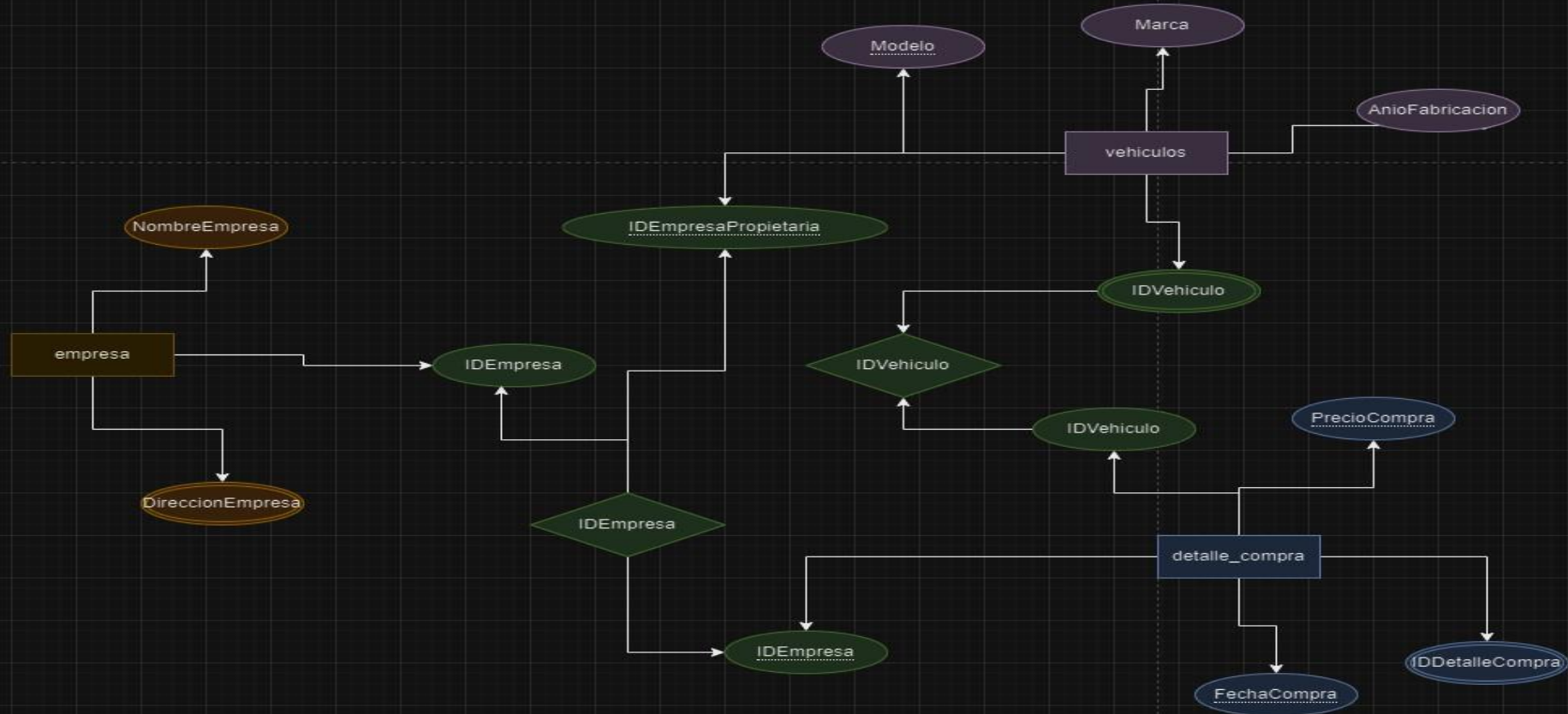
# EL RESULTADO

Results Messages

	IDCliente	NombreCliente	Direccion	Telefono	IDPedido	FechaPedido	IDCliente	IDDetalle	IDPedido	Producto	Cantidad	PrecioUnitario
1	21	juanita	calle lazaró puerta 6	15487245	1	2023-09-18	21	5781	1	alas fritas	8	1.00
2	22	juan	calle laro puerta 8	24874687	2	2023-09-19	22	8741	2	piemas fritas de pollo	7	2.00

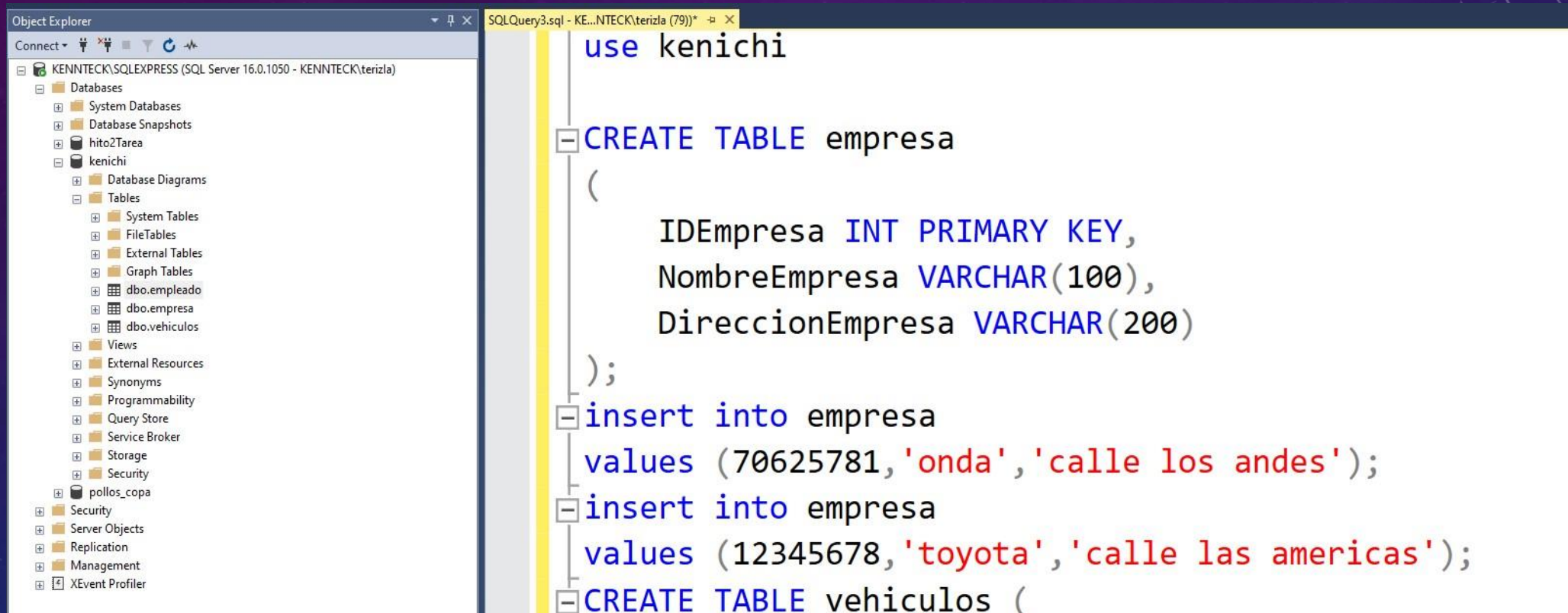


# 16.CREAR EL MODELO ENTIDAD RELACIÓN ER Y SU CÓDIGO SQL.



# EL CODIGO SQL

## CREACION DE LA TABLA EMPRESA



The image shows a screenshot of a SQL Server environment. On the left is the 'Object Explorer' window, which displays the database structure for 'KENNTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTECK\terizla)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', 'hito2Tarea', and 'kenichi'. The 'kenichi' database is expanded, showing 'Database Diagrams', 'Tables', 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', 'dbo.empleado', 'dbo.empresa', and 'dbo.vehiculos'. The 'dbo.empresa' table is highlighted. On the right is the 'SQLQuery3.sql - KENNTECK\terizla (79)' window, which contains the following SQL code:

```
use kenichi

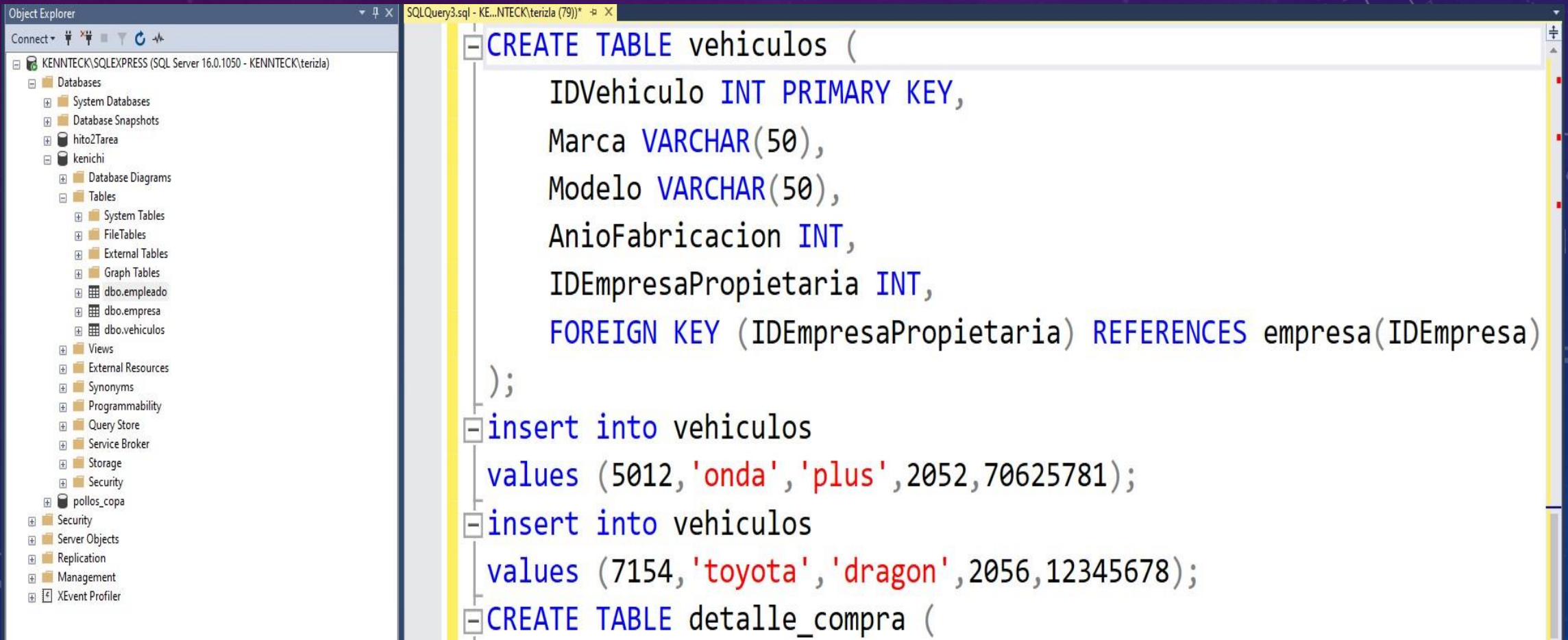
CREATE TABLE empresa
(
    IDEmpresa INT PRIMARY KEY,
    NombreEmpresa VARCHAR(100),
    DireccionEmpresa VARCHAR(200)
);

insert into empresa
values (70625781, 'onda', 'calle los andes');

insert into empresa
values (12345678, 'toyota', 'calle las americas');

CREATE TABLE vehiculos (
```

# CREACION DE LA TABLA VEHICULO

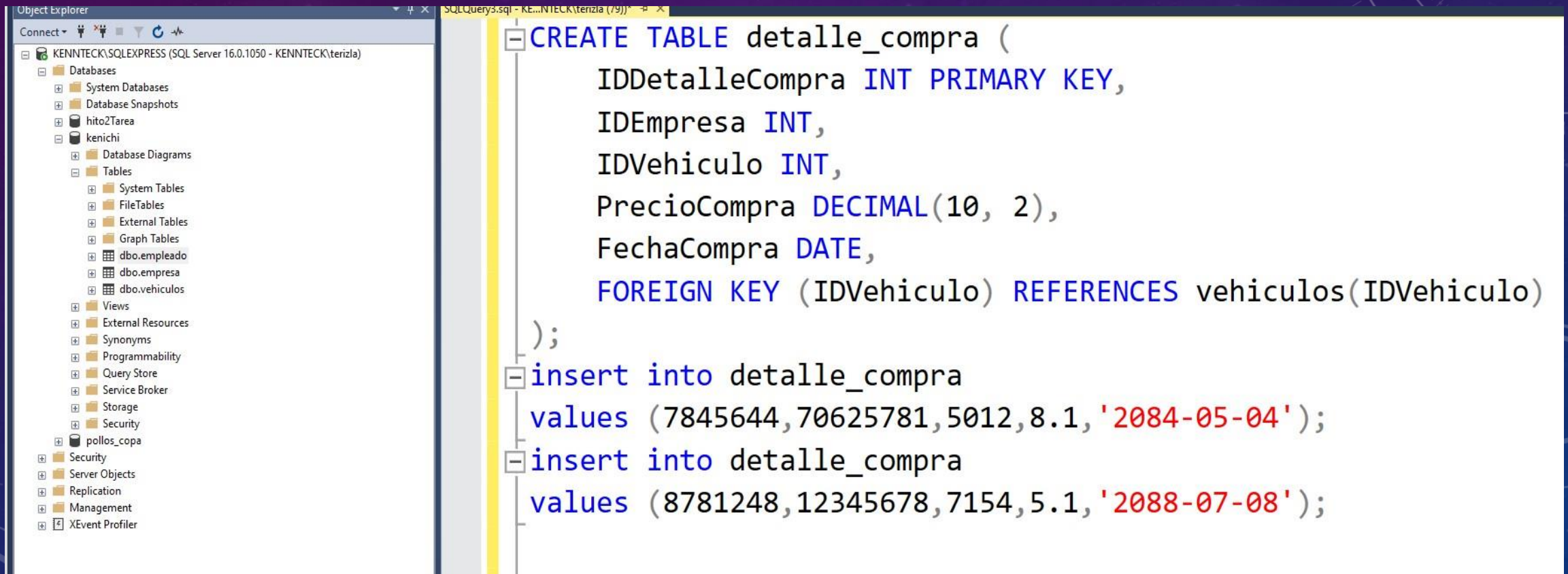


The screenshot displays the SQL Server Enterprise Manager interface on the left and the SQL Query Editor on the right. The Enterprise Manager shows the 'kenichi' database with tables 'dbo.empleado', 'dbo.empresa', and 'dbo.vehiculos'. The Query Editor shows the following SQL script:

```
-- CREATE TABLE vehiculos (  
    IDVehiculo INT PRIMARY KEY,  
    Marca VARCHAR(50),  
    Modelo VARCHAR(50),  
    AnioFabricacion INT,  
    IDEmpresaPropietaria INT,  
    FOREIGN KEY (IDEmpresaPropietaria) REFERENCES empresa(IDEmpresa)  
);  
  
-- insert into vehiculos  
values (5012, 'onda', 'plus', 2052, 70625781);  
  
-- insert into vehiculos  
values (7154, 'toyota', 'dragon', 2056, 12345678);  
  
-- CREATE TABLE detalle_compra (
```



# CREACION DE LA TABLA COMPRA\_DETALLE

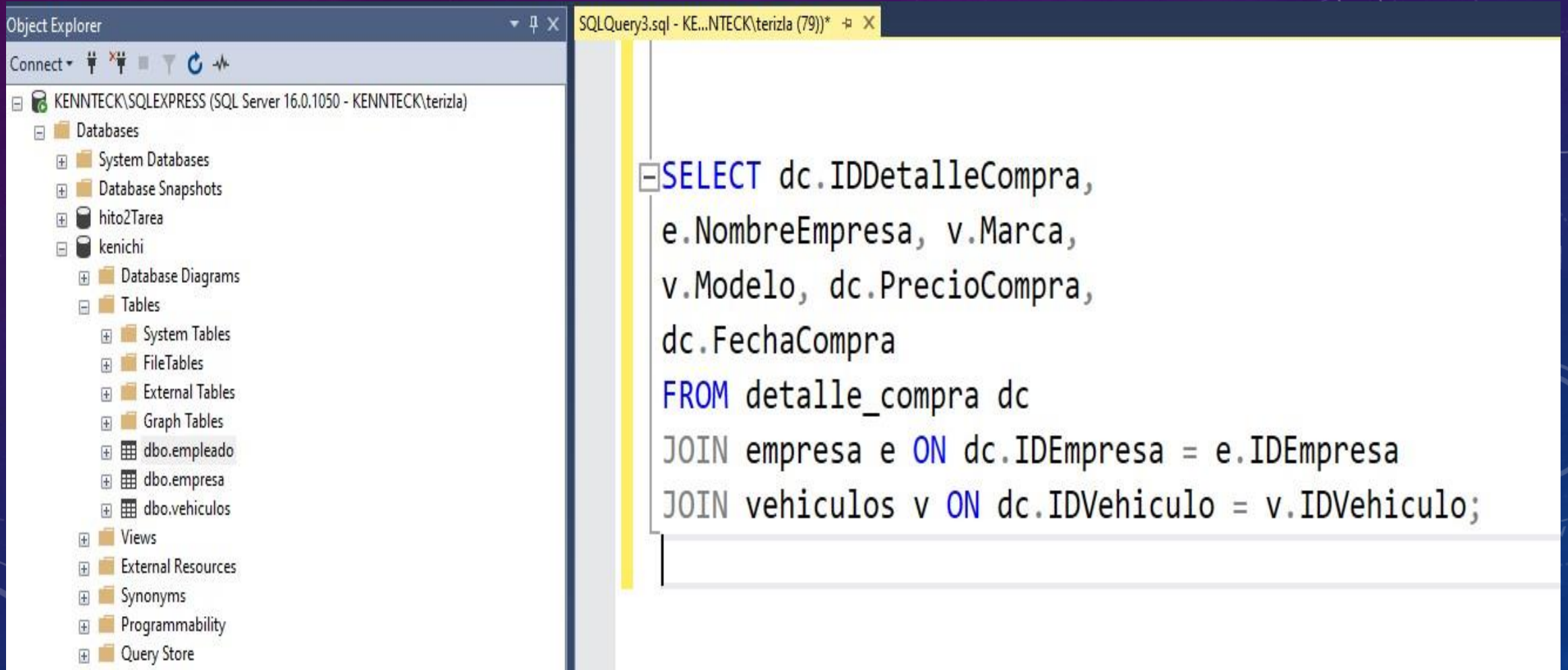


The screenshot displays the SQL Server Enterprise Manager interface on the left and a SQL Query window on the right. The Enterprise Manager shows the 'KENNTECK\SQLSERVER (SQL Server 16.0.1050 - KENNTECK\terizla)' server with a tree view of databases, including 'kenichi' and 'pollos\_copa'. The 'kenichi' database is expanded, showing 'Tables' and 'Views'. The 'pollos\_copa' database is also expanded, showing 'Security', 'Server Objects', 'Replication', 'Management', and 'XEvent Profiler'.

The SQL Query window shows the following SQL script:

```
CREATE TABLE detalle_compra (  
    IDDetalleCompra INT PRIMARY KEY,  
    IDEmpresa INT,  
    IDVehiculo INT,  
    PrecioCompra DECIMAL(10, 2),  
    FechaCompra DATE,  
    FOREIGN KEY (IDVehiculo) REFERENCES vehiculos(IDVehiculo)  
);  
  
insert into detalle_compra  
values (7845644,70625781,5012,8.1,'2084-05-04');  
  
insert into detalle_compra  
values (8781248,12345678,7154,5.1,'2088-07-08');
```

# FINALMENTE SABREMOS SI FUNCIONA CON LOS COMMANDOS



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows the database structure for 'KENNTECK\SQLEXPRESS (SQL Server 16.0.1050 - KENNTECK\terizla)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', 'hito2Tarea', and 'kenichi'. Under 'kenichi', the 'Tables' folder is expanded, listing 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', 'dbo.empleado', 'dbo.empresa', and 'dbo.vehiculos'. The 'Views' folder is also expanded, showing 'External Resources', 'Synonyms', 'Programmability', and 'Query Store'. The main pane on the right shows the 'SQLQuery3.sql' file with the following SQL query:

```
SELECT dc.IDDetalleCompra,  
e.NombreEmpresa, v.Marca,  
v.Modelo, dc.PrecioCompra,  
dc.FechaCompra  
FROM detalle_compra dc  
JOIN empresa e ON dc.IDEmpresa = e.IDEmpresa  
JOIN vehiculos v ON dc.IDVehiculo = v.IDVehiculo;
```

# FINAL EXITOSO

Results		Messages				
	IDDetalleCompra	NombreEmpresa	Marca	Modelo	PrecioCompra	FechaCompra
1	7845644	onda	onda	plus	8.10	2084-05-04
2	8781248	toyota	toyota	dragon	5.10	2088-07-08



# CONCLUSION

- en todo el contenido tanto teorico como practico fue de utilidad para reforzar lo aprendido en clases
- Dando un refuerzo positivo sobre la materia base de datos
- Me a servido para reforzar todo lo aprendido en clases y aumentar algunas cosas a mi conocimiento sobre SQL server