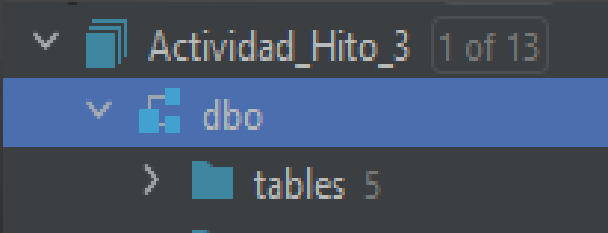



ACTIVIDAD PROCESUAL HITO 3

ESTUDIANTE: EVER TICONA HUALLPA

CREAR UNA BASE DE DATOS

```
CREATE DATABASE Actividad_Hito_3;  
  
USE Actividad_Hito_3;
```



▼  Actividad_Hito_3 1 of 13
▼  dbo
 >  tables 5

The screenshot shows a tree view in SQL Server Enterprise Manager. The 'Actividad_Hito_3' database is expanded, showing the 'dbo' schema. Under 'dbo', there is a folder named 'tables' which contains 5 tables. The 'tables' folder is currently expanded.

TABLA “CATEGORIA”

```
CREATE TABLE categorias  
(  
    id_categoria INT NOT NULL PRIMARY KEY IDENTITY,  
    tipo VARCHAR(20) NOT NULL  
);
```




	categorias
	id_categoria int
	tipo varchar(20)

TABLA “PRODUCTOS”

```
CREATE TABLE productos
(
    id_producto INT NOT NULL PRIMARY KEY IDENTITY,
    nombres VARCHAR(50) NOT NULL,
    stock INT NOT NULL,
    precio_venta INT NOT NULL,
    precio_compra INT NOT NULL,
    id_categoria INT NOT NULL,
    FOREIGN KEY (id_categoria) REFERENCES categorias (id_categoria)
);
```

productos	
id_producto	int
nombre	varchar(100)
stock	int
precio_venta	int
precio_compra	int
id_categoria	int

TABLA “CLIENTE”

```
CREATE TABLE cliente
(
    id_cliente INT NOT NULL PRIMARY KEY IDENTITY,
    nombres VARCHAR(50) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    direccion VARCHAR(100) NOT NULL,
    tipo_cliente VARCHAR(20) NOT NULL
);
```






cliente	
 id_cliente	int
 nombres	varchar(50)
 apellidos	varchar(50)
 direccion	varchar(100)
 tipo_cliente	varchar(20)

TABLA “PEDIDO”

```
CREATE TABLE pedido
(
    id_pedido INT NOT NULL PRIMARY KEY IDENTITY,
    id_cliente INT NOT NULL,
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente)
);
```

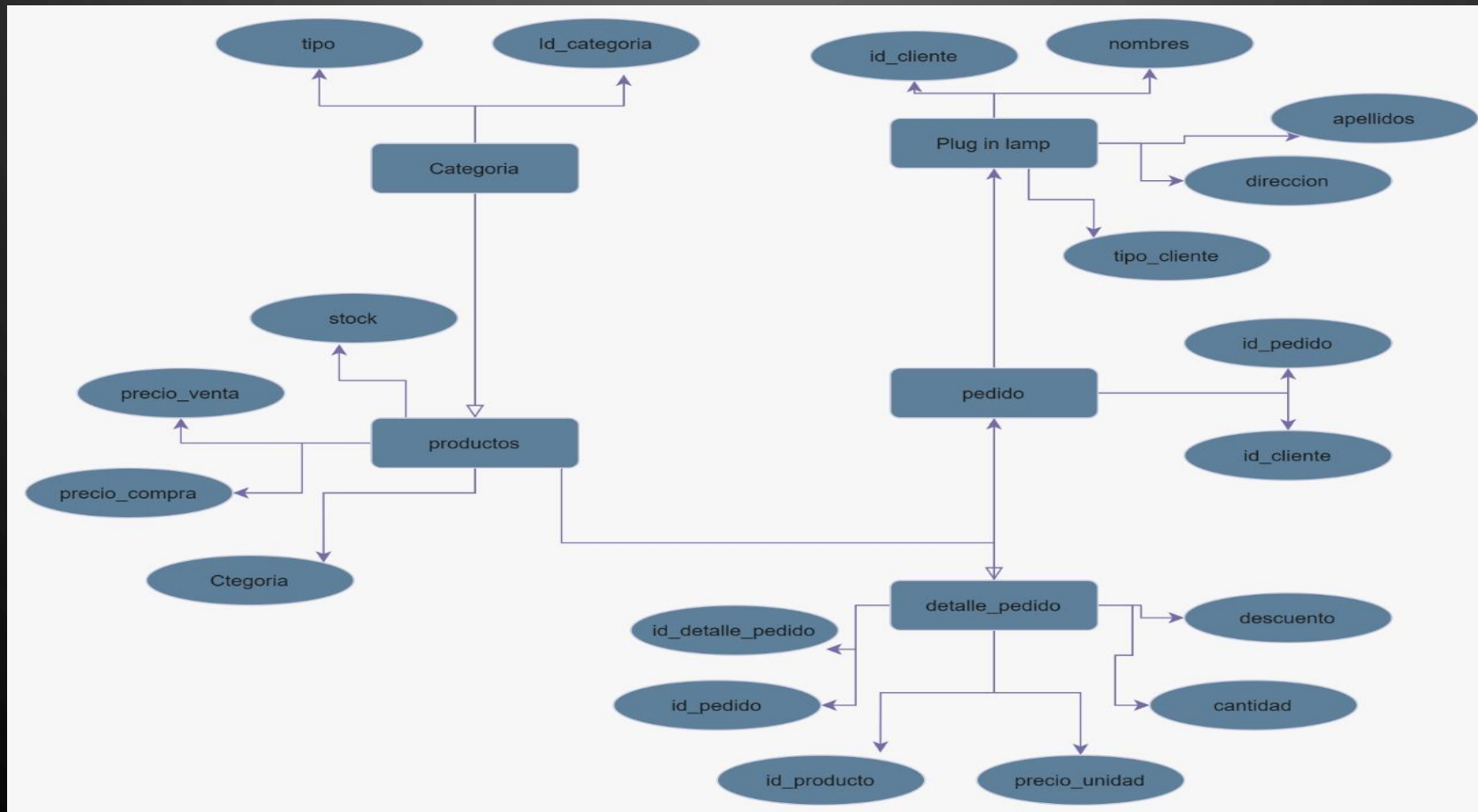


TABLA “DETALLE PEDIDO”

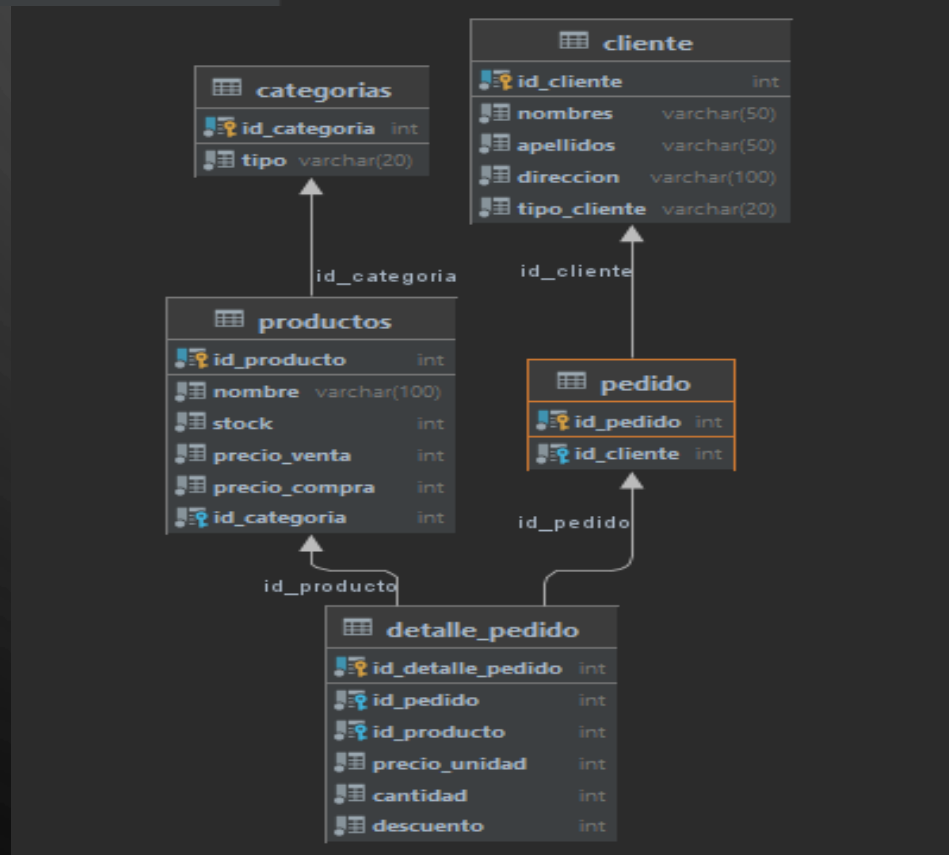
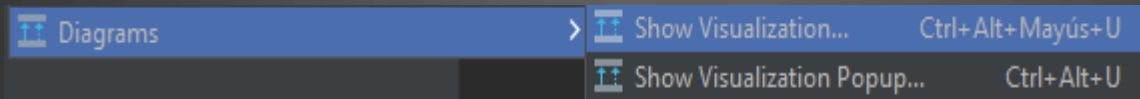
```
CREATE TABLE detalle_pedido
(
    id_detalle_pedido INT NOT NULL PRIMARY KEY IDENTITY,
    id_pedido INT NOT NULL,
    id_producto INT NOT NULL,
    precio_unidad INT NOT NULL,
    cantidad INT NOT NULL,
    descuento INT NOT NULL,
    FOREIGN KEY (id_pedido) REFERENCES pedido(id_pedido),
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
);
```

detalle_pedido	
id_detalle_pedido	int
id_pedido	int
id_producto	int
precio_unidad	int
cantidad	int
descuento	int

BASE DE DATOS REPRESENTADA EN ENTIDAD RELACION



MODELO GRAFICO



INSERTAR REGISTOS A “CATEGORIA”

```
INSERT INTO categorias (tipo)
VALUES ('Electrodomesticos'),
       ('Juguetes'),
       ('Verduras');
```

	id_categoria	tipo
1	1	Electrodomesticos
2	2	Juguetes
3	3	Verduras

INSERTAR REGISTROS A “PRODUCTOS”

```
INSERT INTO productos(nombres, stock, precio_venta, precio_compra, id_categoria)
VALUES ('Refrigerador', 15, 1500, 1000, 1),
       ('Microondas', 4, 800, 500, 1),
       ('Los Vengadores)', 2, 2500, 1700, 2);
```

	id_producto	nombres	stock	precio_venta	precio_compra	id_categoria
1	1	Refrigerador	15	1500	1000	1
2	2	Microondas	4	800	500	1
3	3	Los Vengadores)	2	2500	1700	2

INSERTAR REGISTOS A “CLIENTE”

```
INSERT INTO cliente( nombres, apellidos, direccion, tipo_cliente)
VALUES ('Pedro', 'Mamani Mamani', 'Av.6 de agosto', 'GOLD'),
('juan', 'Sandoval Quispe', 'Av.6 de agosto', 'VIP'),
('Diego', 'Perez Sarmiento', 'Av.6 de agosto', 'VIP'),
('Javier', 'Alanoca Mamani', 'Av.6 de agosto', 'NORMAL'),
('David', 'Ticona Quispe', 'Av.6 de agosto', 'NORMAL'),
('Roly', 'Poma', 'Av.6 de agosto', 'NORMAL');
```

	id_cliente	nombres	apellidos	direccion	tipo_cliente
1	1	Pedro	Mamani Mamani	Av.6 de agosto	GOLD
2	2	juan	Sandoval Quispe	Av.6 de agosto	VIP
3	3	Diego	Perez Sarmiento	Av.6 de agosto	VIP
4	4	Javier	Alanoca Mamani	Av.6 de agosto	NORMAL
5	5	David	Ticona Quispe	Av.6 de agosto	NORMAL
6	6	Roly	Poma	Av.6 de agosto	NORMAL

INSERTAR REGISTOS A “PEDIDO”

```
INSERT INTO pedido(id_cliente)  
VALUES(1),  
      (2);
```

	id_pedido ↕	id_cliente ↕
1	1	1
2	2	2

INSERTAR REGISTROS A “DETALLE PEDIDO”

```
INSERT INTO detalle_pedido (id_pedido, id_producto, precio_unidad, cantidad, descuento )  
VALUES (1, 1, 1000, 2, 0),  
       (1, 2, 800, 1, 0),  
       (2, 2, 800, 1, 10);
```

	id_detalle_pedido	id_pedido	id_producto	precio_unidad	cantidad	descuento
1	1	1	1	1000	2	0
2	2	1	2	800	1	0
3	3	2	2	800	1	10

MANEJO DE CONCEPTOS

- ¿QUE ES EL MODELO ENTIDAD RELACIÓN?

Un modelo entidad relación esta vasado básicamente en entidades de relación de atributos y primary key donde una entidad se ve reflejada atreves de figuras geométricas como rectángulo, rombo y una ovalo

- ¿QUE ES EL MODELO LÓGICO EN BASES DE DATOS RELACIONALES?

El modelo relacional, para el modelado y la gestión de bases de datos, es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados tuplas.

- ¿DESCRIBE Y MENCIONA QUE FORMAS (SHAPES) SE UTILIZA PARA GRAFICAR UN MODELO ENTIDAD RELACIÓN?

Se utilizan el circulo u ovalo, rectángulo, rombo y líneas:

- los círculos u óvalos se utilizan para los atributos de una tabla.
- el rectángulo es para representar las entidades.
- los rombos representa por que se conectan las entidades.
- las líneas se utilizan para conectar entre entidades.

- ¿QUÉ ES UNA FUNCIÓN DE AGREGACIÓN. ?

Es una función invocada por sql que toma valores que dependen de todas las filas que selecciona la consulta y devuelve información sobre estas filas. el servidor de bases de datos da soporte a las funciones de agregación que escribe el usuario, denominadas agregados definidos por el usuario .

- ¿MUESTRE EJEMPLO DEL USO DE 2 FUNCIONES DE AGREGACIÓN. ?

Una función de agregación es una función que resume las filas de un grupo en un solo valor. count, min y max son ejemplos de funciones de agregación.

```
FS SELECT COUNT(*) AS TOTAL_COUNT, COUNT(FRUIT) AS NON_NULL_COUNT,  
MIN(FRUIT) AS MIN, MAX(FRUIT) AS MAX  
FROM (SELECT NULL AS FRUIT UNION ALL  
SELECT "APPLE" AS FRUIT UNION ALL  
SELECT "PEAR" AS FRUIT UNION ALL  
SELECT "ORANGE" AS FRUIT)
```


- ¿MUESTRE UN EJEMPLO DEL USO DE JOINS. ?

```
SELECT TOP 100 P.PRODUCTID,
```

```
P.NAME, P.LISTPRICE, P.SIZE, P.MODIFIEDDATE, SOD.UNITPRICE, SOD.UNITPRICEDISCOUNT, SOD.ORDERQTY, SOD.LINETOTAL FROM  
SALES.SALESORDERDETAIL SOD INNER JOIN PRODUCTION. PRODUCT PON SOD.PRODUCTID = P.PRODUCTID
```

```
WHERE SOD.UNITPRICE > 1000
```

```
ORDER BY SOD.UNITPRICE DESC
```

- ¿QUÉ ES SQL Y NOSQL?

SQL permite combinar de forma eficiente diferentes tablas para extraer información relacionada, mientras que NOSQL no lo permite o muy limitadamente. NOSQL permite distribuir grandes cantidades de información; mientras que SQL facilita distribuir bases de datos relacionales.

- ¿A QUE SE REFIERE CUANDO SE HABLA DE ISO, QUE ES UNA ISO?

Se trata de la organización internacional de normalización o estandarización, y se dedica a la creación de normas o estándares para asegurar la calidad, seguridad y eficiencia de productos y servicios. son las llamadas normas ISOL.

- ¿QUIEN CREO EL MODELO ENTIDAD RELACIÓN O MAS CONOCIDO COMO E-R?

Peter Chen actualmente se desempeña como miembro de la facultad de la universidad carnegie mellon ubicada en pittsburgh y se le atribuye el desarrollo del modelo er para el diseño de bases de datos en los 70.

MANEJO DE CONSULTAS

```
--Mostrar los productos(Nombre y stock) con stock mayor igual a 10.  
--productos  
SELECT pro.nombres, pro.stock  
FROM productos AS pro  
WHERE pro.stock >= 10 ;
```

1 row ▾		Tx: Auto ▾		DDL
!	nombres	!	stock	
1	Refrigerador		15	

MANEJO DE CONSULTAS

```
--Mostrar el nombre del producto y la categoría de los productos pertenecen a la categoría de "electrodomesticos".  
--producto  
--categorias  
✓ SELECT pro.nombres, cat.tipo  
FROM productos AS pro  
    INNER JOIN categorias AS cat ON pro.id_categoria = cat.id_categoria  
WHERE cat.tipo = 'Electrodomesticos';
```

	nombres	tipo
1	Refrigerador	Electrodomesticos
2	Microondas	Electrodomesticos

MANEJO DE CONSULTAS

```
--Que productos(nombre) tiene el pedido con id igual a = 1.  
--Productos  
--detalle_pedido  
--pedido
```



```
SELECT pro.nombres  
FROM productos AS pro  
INNER JOIN detalle_pedido AS dep ON pro.id_producto = dep.id_producto  
INNER JOIN pedido AS ped ON dep.id_pedido = ped.id_pedido  
WHERE ped.id_pedido = 1;
```

|< < 2 rows > >| | ↺ ■ | + - ↶ ↷ ⬆ | Tx: Auto ▾ | DDL

	nombres
1	Refrigerador
2	Microondas

MANEJO DE CONSULTAS



```
SELECT COUNT(pro.nombres) AS pedidos_igual_a_2  
FROM productos AS pro  
INNER JOIN detalle_pedido AS dep ON pro.id_producto = dep.id_producto  
INNER JOIN pedido AS ped ON dep.id_pedido = ped.id_pedido  
WHERE ped.id_pedido = 2;
```



|< < 1 row v > >| | ↺ ■

pedidos_igual_a_2 ↕

1

1

MANEJO DE CONSULTAS

```
6  --Crear una función que permita sumar 3 números.
7
8  CREATE FUNCTION suma_de_3_numeros (@numero_1 INT, @numero_2 INT,@numero_3 INT)
9  RETURNS INTEGER
10 AS
11 BEGIN
12     DECLARE @resultado INT = 0
13     SET @resultado = @numero_1 + @numero_2 + @numero_3
14     RETURN @resultado
15 END;
16
17 ✓ SELECT dbo.suma_de_3_numeros( 8, 9, 10) AS El_resultado_de_la_suma_es;
```

|< < 1 row > >| ↺ ■

El_resultado_de_la_suma_es ↕

1	27
---	----

MANEJO DE CONSULTAS

--Crear una función que permita restar 3 números

```
CREATE FUNCTION resta_de_3_numeros (@numero_1 INT, @numero_2 INT, @numero_3 INT)
RETURNS INTEGER
AS
BEGIN
    DECLARE @resultado INT = 0
    SET @resultado = @numero_1 - @numero_2 - @numero_3
    RETURN @resultado
END;
```

✓ `SELECT dbo.resta_de_3_numeros(5, 12, 7) AS el_resuldado_de_la_resta_es;`



|< < 1 row > >| ↺ ■

el_resuldado_de_la_resta_es ↕

1

-14

MANEJO DE CONSULTAS

```
--Cómo unificaría en una sola función el ejercicio 3.5 y 3.6(los dos anteriores)
CREATE OR ALTER FUNCTION suma_y_resta_de_3_numeros (@numero_1 INT, @numero_2 INT,@numero_3 INT)
RETURNS VARCHAR(100)
AS
BEGIN
    DECLARE @resultado INT = 0
    DECLARE @resultados INT = 0
    DECLARE @mostrar VARCHAR (50) = ''
    DECLARE @a VARCHAR (20) = ''
    DECLARE @b VARCHAR (20) = ''
    SET @resultado = @numero_1 + @numero_2 + @numero_3
    SET @a = @resultado
    SET @resultados = @numero_1 - @numero_2 - @numero_3
    SET @b = @resultados
    SET @mostrar = 'La suma es '+@a+'y' +' la resta es'+@b
    RETURN @mostrar
END;
SELECT dbo.suma_y_resta_de_3_numeros (8, 9, 10) AS suma_y_resta;
```

160 ✓ `SELECT dbo.suma_y_resta_de_3_numeros (8, 9, 10) AS suma_y_resta;`



|< < 1 row > >| | ↺ ■

■ suma_y_resta ▾

1 La suma es 27y la resta es-11