

UNIVERSIDAD PRIVADA
FRANZ TAMAYO



Laboratorio Procesual Hito 3

Estudiantes:

- Mijael Jhonatan Rojas Arias

Docente:

Ing. William Roddy Barra Paredes

Asignatura:

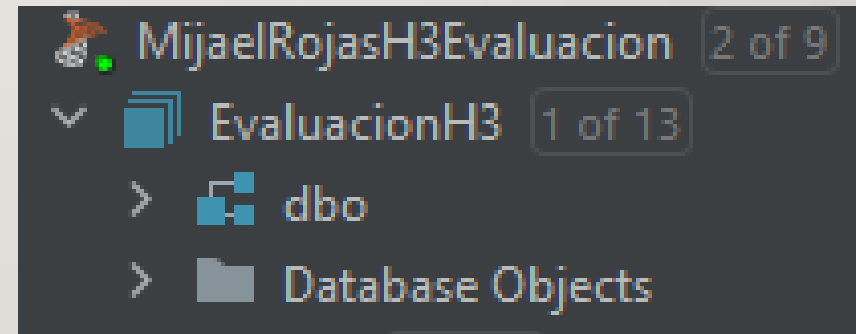
Base de Datos

CREACIÓN DE LA BASE DE DATOS

```
Create database EvaluacionH3;
```

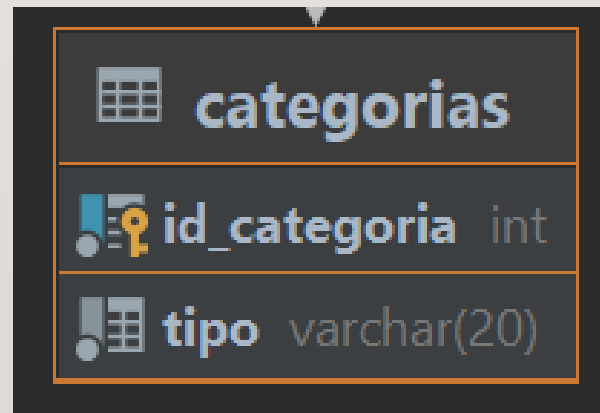
ENTRANDO A LA BASE DE DATOS

```
use EvaluacionH3;
```








CREANDO LA TABLA “CATEGORIAS” CON SUS RESPECTIVAS COLUMNAS.

```
create table categorias (  
    id_categoria int not null primary key identity,  
    tipo varchar (20) not null  
);
```



CREANDO LA TABLA “CLIENTE” CON SUS RESPECTIVAS COLUMNAS.

```
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)







CREANDO LA TABLA “PEDIDO” CON SUS COLUMNAS RELACIONADA CON OTRA TABLA.

```
create table pedido (  
    id_pedido int not null primary key identity,  
    id_cliente int not null,  
    foreign key (id_cliente) references cliente(id_cliente)  
);
```



CREANDO LA TABLA “PRODUCTOS” CON SUS COLUMNAS RELACIONADA CON OTRA TABLA.

```
create table productos (  
    id_producto int not null primary key identity,  
    nombre varchar (100) 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

CREANDO LA TABLA “EQUIPO” CON SUS COLUMNAS RELACIONADA CON OTRA TABLA.

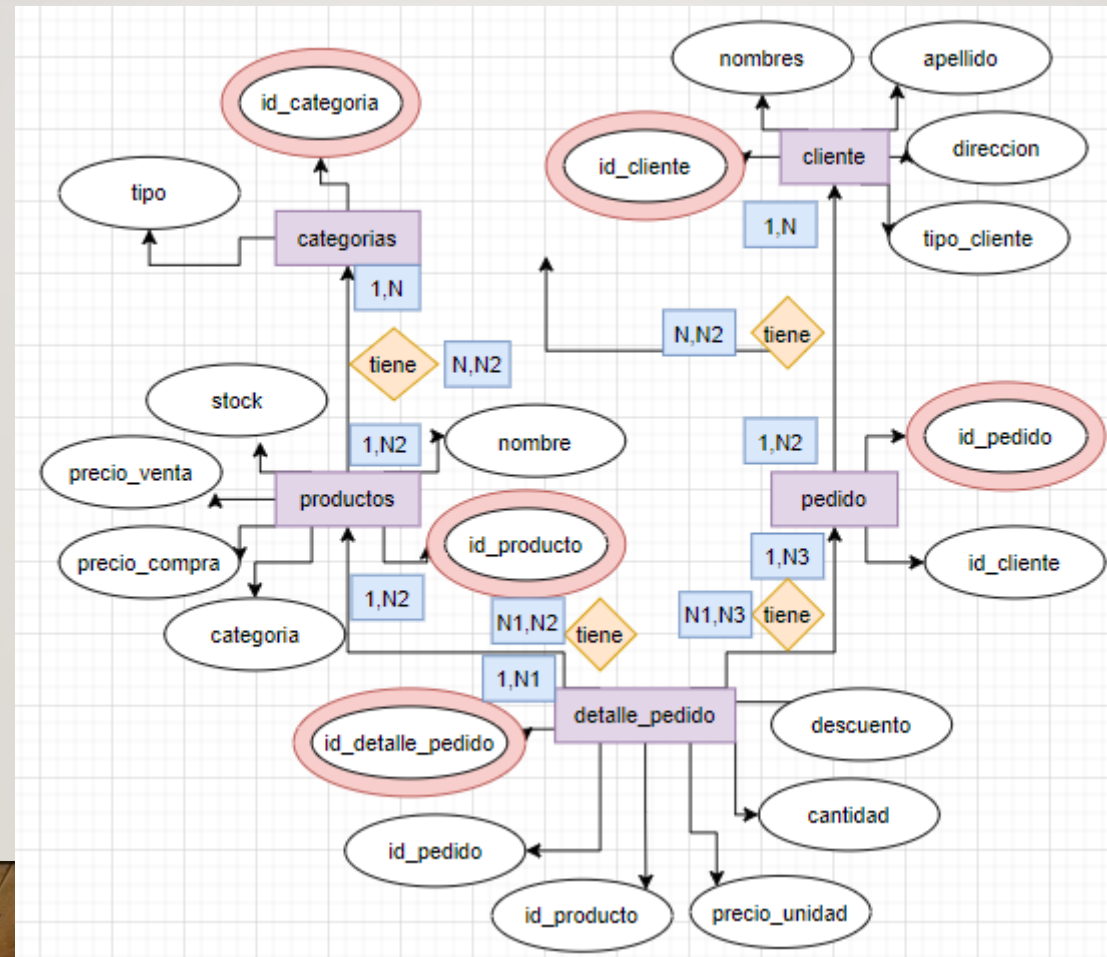
```
create table detalle_pedido (  
    id_detalle_pedido int not null primary key identity,  
    id_pedido int not null,  
    id_producto int not null,  
    precio_unidad int not null,  
    cantidad int not null,  
    descuento int not null,  
    foreign key (id_pedido) references pedido (id_pedido),  
    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

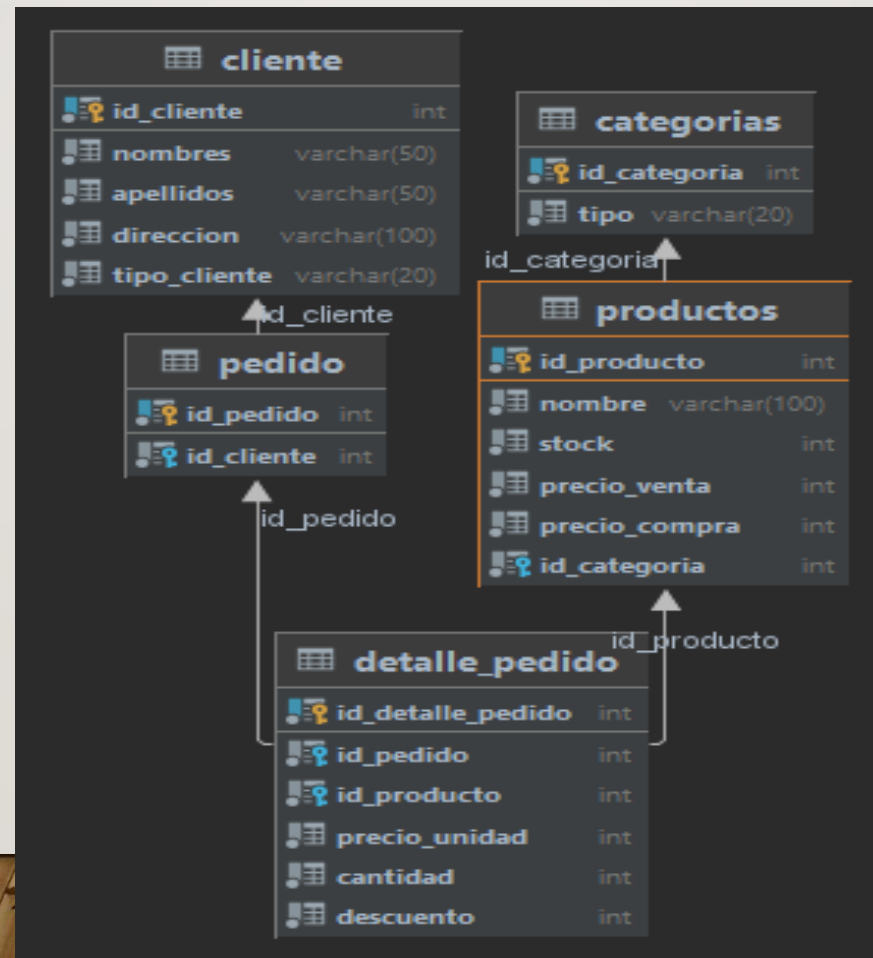
DISEÑO DE LA BASE DE DATOS



BASE DE DATOS REPRESENTADA EN ENTIDAD RELACIÓN.



BASE DE DATOS REPRESENTADA EN MODELO GRAFICO



AGREGAMOS REGISTROS A LA TABLA “CATEGORIAS”.

```
insert into categorias (tipo)
values
('Electrodomesticos'),
('Juguetes'),
('Verduras');
```

	id_categoria	tipo
1	1	Electrodomesticos
2	2	Juguetes
3	3	Verduras



AGREGAMOS REGISTROS A LA TABLA “CLIENTE”.

```
insert into cliente (nombres, apellidos, direccion, tipo_cliente)
values
('nombre_cliente1', 'apellidos_cliente1', 'ó de agosto edificio fernandez', 'GOLD'),
('nombre_cliente2', 'apellidos_cliente2', 'Plaza Avaroa', 'VIP'),
('nombre_cliente3', 'apellidos_cliente3', 'Plaza del estudiante', 'NORMAL'),
('nombre_cliente4', 'apellidos_cliente4', 'Teatro al aire libre', 'NORMAL');
```

	id_cliente	nombres	apellidos	direccion	tipo_cliente
1	1	nombre_cliente1	apellidos_cliente1	ó de agosto edificio fernandez	GOLD
2	2	nombre_cliente2	apellidos_cliente2	Plaza Avaroa	VIP
3	3	nombre_cliente3	apellidos_cliente3	Plaza del estudiante	NORMAL
4	4	nombre_cliente4	apellidos_cliente4	Teatro al aire libre	NORMAL

AGREGAMOS REGISTROS A LA TABLA “PEDIDO”.

```
insert into pedido (id_cliente)  
values  
(1),(2);
```

	 id_pedido :	 id_cliente :
1	1	1
2	2	2

AGREGAMOS REGISTROS A LA TABLA “PRODUCTOS”.

```
insert productos (nombre, stock, precio_venta, precio_compra, id_categoria)
values
('refrigerador', 15, 1500, 1000, 1),
('microondas', 4, 800, 500, 1),
('Los vengadores (incluye Thanos)', 2, 2500, 1700, 2);
```

	🔑 id_producto ↕	📄 nombre ↕	📄 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 (incluye Thanos)	2	2500	1700	2

AGREGAMOS REGISTROS A LA TABLA “PRODUCTOS”.

```
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	5	1	1	1000	2	0
2	6	1	2	800	1	0
3	7	2	2	800	1	10

MANEJO DE CONCEPTOS



MANEJO DE CONCEPTOS

1. ¿Qué es el modelo entidad relación?

Es la representación grafica de datos, se utiliza para facilitar las entidades de una base de datos.

2. ¿Qué es el modelo lógico en Base de Datos relaciones?

Es un modelo de una Base de Datos, esta te facilita el trabajado para la representación de entidades.

3. Describe que formas (shapes) se utilizan 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.

MANEJO DE CONCEPTOS

4. ¿Qué es una función de agregación?

Una función de agregación, estas funciones se encuentran ya integradas en SQL, una cuantas de ellas serian el: Count, Max, Min

5. Muestre ejemplo del uso de funciones de agregación.

Un ejemplo de funciones de agregación es:

```
Select count (cli.tipo_cliente)
from cliente as cli
where cli.tipo_cliente = 'VIP'
```

Esta función “count” te cuentas las filas (rows) que tiene una tabla, en este caso cuenta las filas que tiene agregado “VIP”

MANEJO DE CONCEPTOS

6. Muestre un ejemplo de uso de JOINS.

Un ejemplo de uso de JOINS es:

```
select ins.id_insc, est.id_est, mat.id_mat  
from Inscripcion as ins
```

```
inner join Estudiantes as est on ins.id_est = est.id_est
```

```
inner join Materias as mat on mat.id_mat = ins.id_mat
```

Este ejemplo muestra como la tabla Inscripcion relaciona a la tabla Estudiantes por su `ins.id_est = est.id_est` y la tabla Materias relaciona con la tabla Inscripcion por su `mat.id_mat = ins.id_mat` con el uso de **JOINS**

7. ¿Que es SQL y NoSQL?

SQL hace referencias a las bases de datos relacionales, estas utilizan filas y columnas (tabla)

NoSQL hace referencias a las bases de datos no relacionales, estas utilizan documentos

MANEJO DE CONCEPTOS

8. ¿A que se refiere cuando hablan de ISO, que es una ISO?

Una ISO son normas que describe como gestionar la información de algún centro que tiene información masiva.

Esta explica una buena protección de los datos para evitar el robo de identidades o información de una Base de Datos

9. ¿Quién creo el modelo entidad relación o mas conocido como E-R?

Peter Pin- Shan es el que creo el modelo entidad relación en el año 1968.

MANEJO DE CONCEPTOS

10. Crea una función que permita sumar 3 números

```
create function suma_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.suma_de_3_numeros(4, 6, 7 ) as suma_de_3_números
```

MOSTRAR LOS PRODUCTOS(NOMBRE Y STOCK)
CON STOCK MAYOR IGUAL A 10.

```
select pro.nombre, pro.stock  
from productos as pro  
where pro.stock >= 10;
```

	nombre	stock
1	refrigerador	15

MOSTRAR EL NOMBRE DEL PRODUCTO Y LA CATEGORÍA DE LOS PRODUCTOS PERTENECEN A LA CATEGORÍA DE "ELECTRODOMESTICOS".

```
select pro.nombre, cat.tipo
from productos as pro
inner join categorias as cat on pro.id_categoria = cat.id_categoria
where cat.tipo = 'Electrodomesticos'
```

	nombre	tipo
1	refrigerador	Electrodomesticos
2	microondas	Electrodomesticos

QUE PRODUCTOS(NOMBRE) TIENE EL PEDIDO CON ID IGUAL A = 1

```
select pro.nombre  
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
```

	nombre
1	refrigerador
2	microondas

CUANTOS(COUNT) PRODUCTOS TIENE EL PEDIDO CON ID IGUAL A = 2.

```
select count(pro.nombre) as Cantidad_de_productos
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
```

	Cantidad_de_productos	
1		1

CREAR UNA FUNCIÓN QUE PERMITA SUMAR 3 NÚMEROS.

```
create function suma_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
```

Consulta para ejecutar la función

```
select dbo.suma_de_3_numeros( @numero_1: 4, @numero_2: 6, @numero_3: 7 ) as suma_de_3_numeros
```

	suma_de_3_numeros
1	17

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
```

Consulta para ejecutar la función

```
select dbo.resta_de_3_numeros( @numero_1: 4, @numero_2: 6, @numero_3: 7 ) as resta_de_3_numeros
```

	resta_de_3_numeros
1	-9

CÓMO UNIFICARIA EN UNA SOLA FUNCIÓN EL EJERCICIO 3.5 Y 3.6(LOS DOS ANTERIORES).

```
create or alter function suma_resta_de_3_numeros(@numero_1 int, @numero_2 int, @numero_3 int)
returns varchar (100)
as
begin
    declare @resultado int = 0
    declare @resultadores int = 0
    declare @mostrar varchar (100) = ''
    declare @a varchar (100) = ''
    declare @b varchar (100) = ''
    set @resultado = @numero_1 + @numero_2 + @numero_3
    set @a = @resultado
    set @resultadores = @numero_1 - @numero_2 - @numero_3
    set @b = @resultadores
    set @mostrar = 'El resultado de la suma es '+@a+' y el resultado de la resta es '+@b
    return @mostrar
end
```

Consulta para ejecutar la función

```
select dbo.suma_resta_de_3_numeros ( @numero_1: 4, @numero_2: 6, @numero_3: 7) as Suma_y_resta
```

Suma_y_resta

1 El resultado de la suma es 17 y el resultado de la resta es -9

GRACIAS

