



Procesual Hito 3

Nombre completo: Kevin Michael Apaza Lara

Universidad: Universidad Franz Tamayo (Unifranz el alto)

Paralelo: 3

Semestre: II/2021

Materia: Base de Datos I

Consigna

Diseñe un sistema de Base de Datos Relacional utilizando el gestor de Base de Datos SQL Server teniendo como premisa el uso de buenas prácticas en diseño de la base de datos aplicados al siguiente escenario.

Una pequeña empresa de comida rápida de nombre the delicious desea implementar un nuevo sistema para poder administrar los PEDIDOS de sus productos.

The delicious FF	
Problema	<p>En función al escenario se identificó que las posibles entidades son categorías, productos, cliente, pedido y detalle pedido, pues un cliente realiza un pedido de productos, en donde cada producto tiene una categoría a la cual pertenece y la descripción del pedido debería tener un detalle de pedido.</p> <p>En tal sentido se deberían crear las siguientes entidades y/o tablas.</p> <ul style="list-style-type: none">• categorias• productos• cliente• pedido• detalle_pedido <p>Detalle de las entidades.</p>

creamos la base de datos y la usamos

```
create database the_delicious;  
  
use the_delicious;
```

Creación de tablas

Empezamos con la tabla categoría y las respectivas especificaciones.

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

Procedemos con la creación de la tabla producto con su llave primaria y en este caso al la tabla depender de la tabla categorias creamos nuestra llave foránea y la relacionamos.

```
create table productos
(
    id_producto int primary key identity not null,
    nombre varchar(100) not null ,
    stock int not null,
    precio_venta int not null,
    precio_compra int not null,
    categoria int not null,
    foreign key (categoria) references categorias (id_categoria)
);
```

Creamos una nueva tabla esta ya no depende de otra así que procedemos con la creación de lo especificado.

```
create table cliente
(
    id_cliente int primary key identity not null,
    nombres varchar(50)not null,
    apellidos varchar (50)not null,
    direccion varchar (100)not null,
    tipo_cliente varchar(6)not null
);
```

Seguimos con la creación de la tabla pedido y esta al ser dependiente de la tabla cliente creamos una llave foránea para relacionarla, además de su respectiva llave primaria.

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

Procedemos con la última tabla detalle_pedido creamos su llave primaria, seguimos con las especificaciones, y finalmente ya que la tabla depende de las tablas pedido y productos, creamos las llaves foráneas y con eso concluye la creación de tablas.

```
create table detalle_pedido
(
    id_detalle_pedido int primary key identity not null,
    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)
);
```

1. Diseño de base de datos.

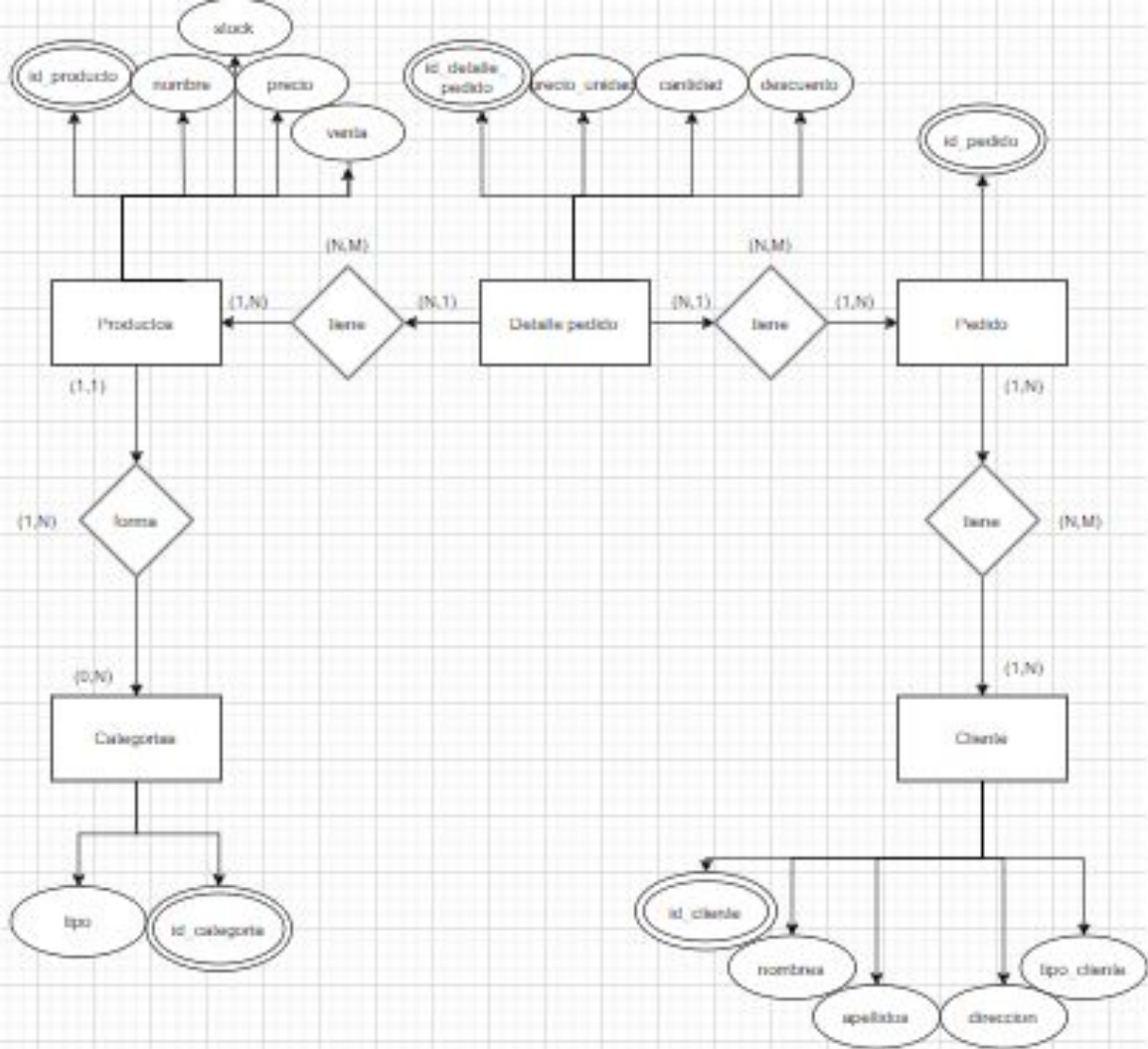
1.1. Dado el detalle explicado en la parte inicial de este documento debería generar el modelo entidad relación.

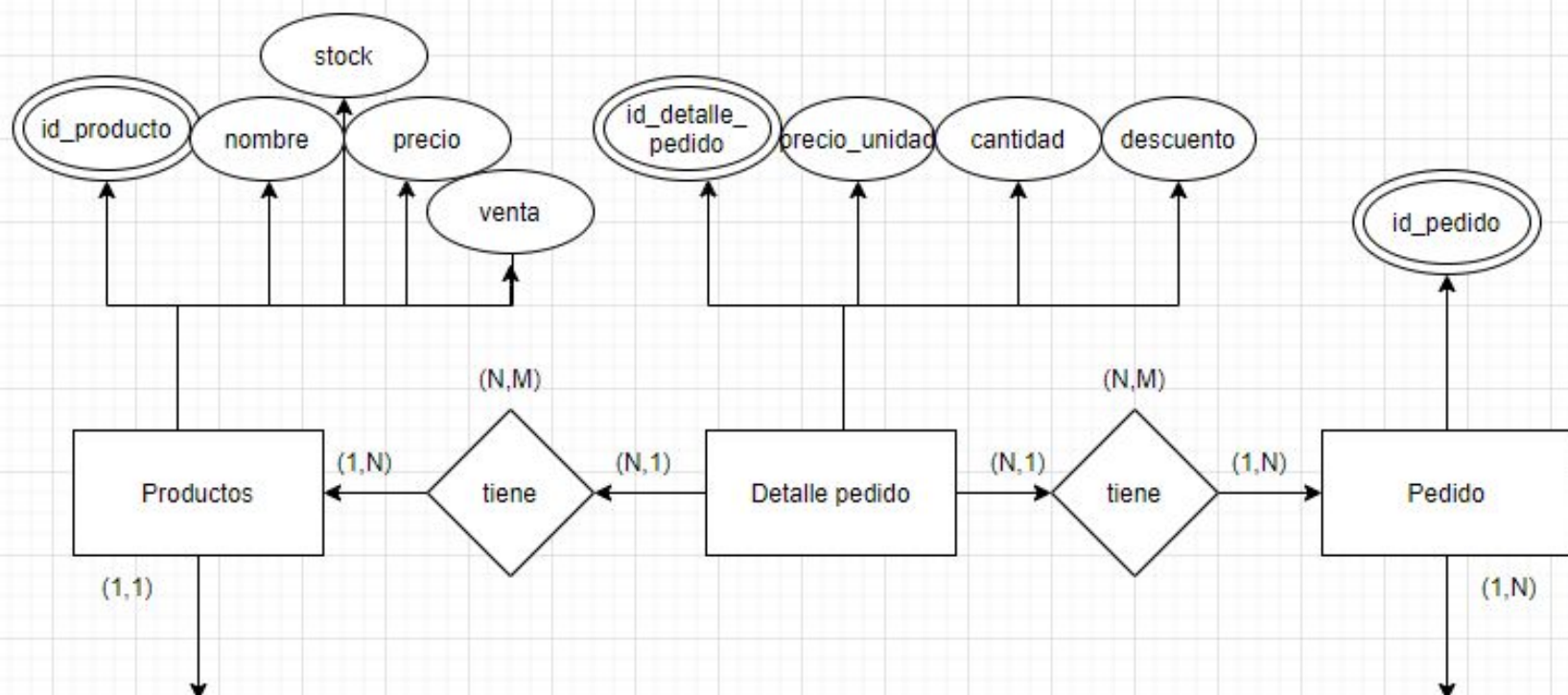
- Para poder generar el diagrama entidad relación, podría utilizar la plataforma Diagrams.

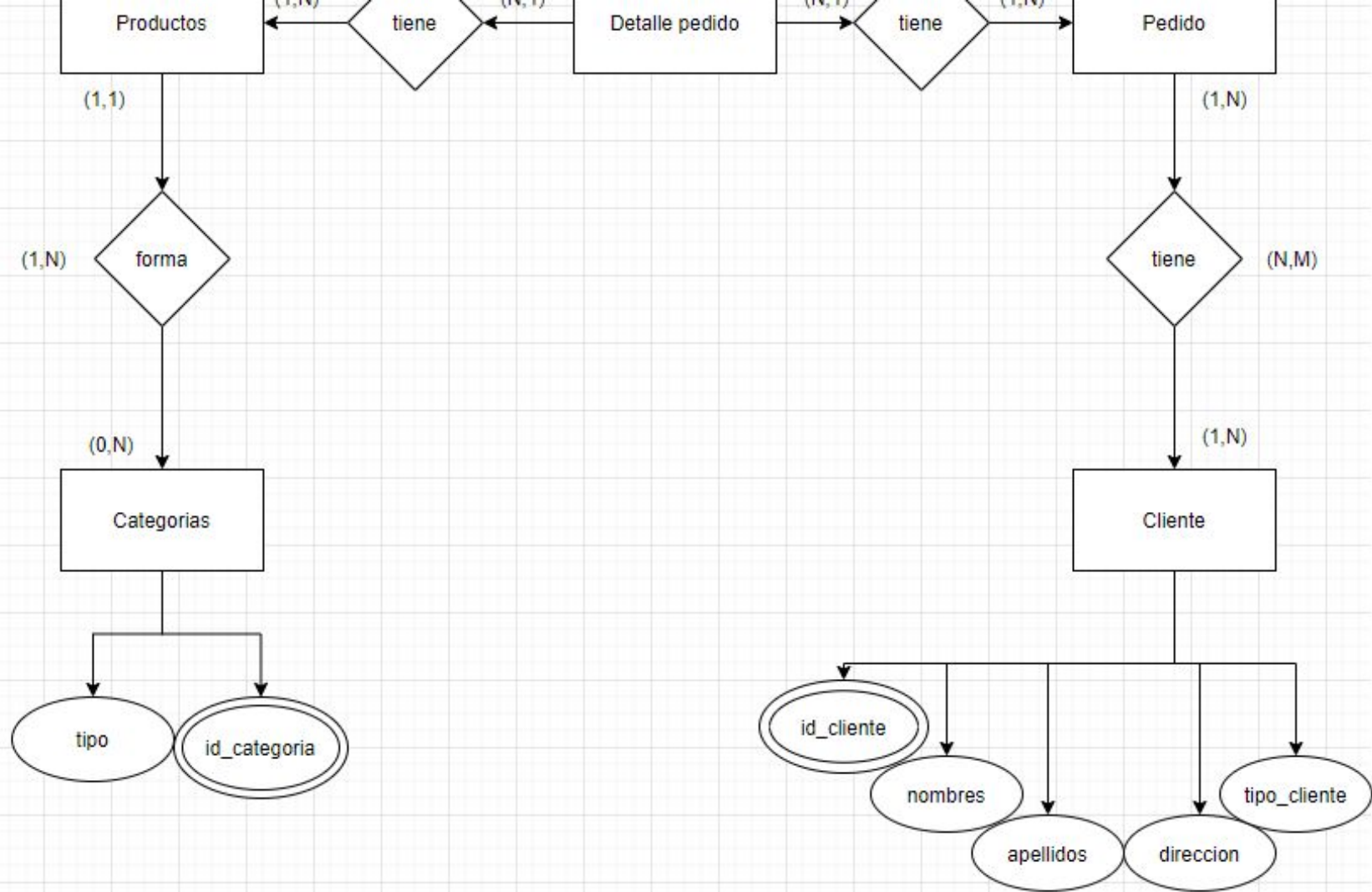
1. Diseño de base de datos.

1.1. Dado el detalle explicado en la parte inicial de este documento debería generar el modelo entidad relación.

■ Para poder generar el diagrama entidad relación, podría utilizar la plataforma Diagrams.







1.2. Después de generar el modelo lógico de la base de datos.

- Utilizar Datagrip para poder generar el diagrama



1.3. Agregar los siguientes registros a las tablas

categorias		
	🔗 id_categoria :	📄 tipo :
1	1	electrodomesticos
2	2	juguetes
3	3	verduras

productos					
🔗 id_prod... :	📄 nombre :	📄 stock :	📄 precio_venta :	📄 precio_compra :	🔗 categoria :
1	refrigerador	15	1500	1000	1
2	microonda	4	800	500	1
3	los vengadores ...	2	2500	1700	2

cliente

id_cliente	nombres	apellidos	direccion	tipo_cliente
1	nombre_cliente1	apellidos_cli...	6 de agosto e...	GOLD
2	nombre_cliente2	apellidos_cli...	Plaza abaroa	VIP
3	nombre_cliente3	apellidos_cli...	Plaza del est...	NORMAL
4	nombre_cliente4	apellidos_cli...	Teatro al air...	NORMAL

pedido

	id_pedido	id_cliente
1	1	1
2	2	2

detalle_pedido

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

Creación de los registros

```
insert into categorias(tipo) values  
('electrodomesticos'),  
('juguetes'),  
('verduras');
```

WHERE		ORDER BY	
	id_categoria	tipo	
1	1	electrodomesticos	
2	2	juguetes	
3	3	verduras	

```
insert into productos(nombre, stock, precio_venta, precio_compra, categoria)values
('refrigerador',15,1500,1000,1),
('microonda',4,800,500,1),
('los_vengadores',2,2500,1700,2);
```

WHERE

ORDER BY

	id_producto	nombre	stock	precio_venta	precio_compra	categoria
1	1	refrigerador	15	1500	1000	1
2	2	microonda	4	800	500	1
3	3	los vengadores	2	2500	1700	2

```
insert into cliente(nombres, apellidos, direccion, tipo_cliente) values  
('nombre_cliente1', 'apellidos_cliente1', '6 de agosto', 'GOLD'),  
('nombre_cliente2', 'apellidos_cliente2', 'Plaza abaroa', 'VIP'),  
('nombre_cliente3', 'apellidos_cliente3', 'Plaza del estadio', 'NORMAL'),  
('nombre_cliente4', 'apellidos_cliente4', 'Teatro al aire libre', 'NORMAL');
```

WHERE		ORDER BY			
	id_cliente	nombres	apellidos	direccion	tipo_cliente
1	1	nombre_cliente1	apellidos_cliente1	6 de agosto	GOLD
2	2	nombre_cliente2	apellidos_cliente2	Plaza abaroa	VIP
3	3	nombre_cliente3	apellidos_cliente3	Plaza del estadio	NORMAL
4	4	nombre_cliente4	apellidos_cliente4	Teatro al aire libre	NORMAL

Manejo de conceptos

2.1. Que es el modelo entidad relación.

2.2. Que es el modelo lógico en bases de datos relacionales.

2.3. Describe y menciona que formas(shapes) se utiliza para graficar un modelo entidad relación.

2.4. Qué es una función de agregación.

2.5. Muestra ejemplo del uso de 2 funciones de agregación.

2.6. Muestra un ejemplo del uso de JOINS.

2.7. Qué es SQL y NoSQL.

2.8. A que se refiere cuando se habla de ISO, que es una ISO.

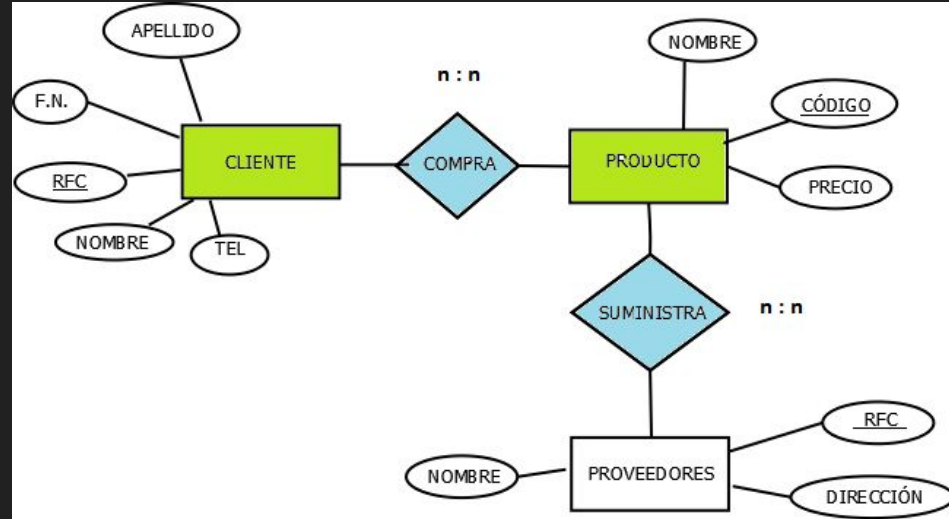
2.9. Quien creo el modelo entidad relación o mas conocido como E-R

2.10. Crear una función que permita sumar 3 números .

2.1. Que es el modelo entidad relación.

Es un sistema de representación de las entidades importantes de una base de datos y cuenta con distintas características como, entidades, atributos, relaciones y cardinalidad

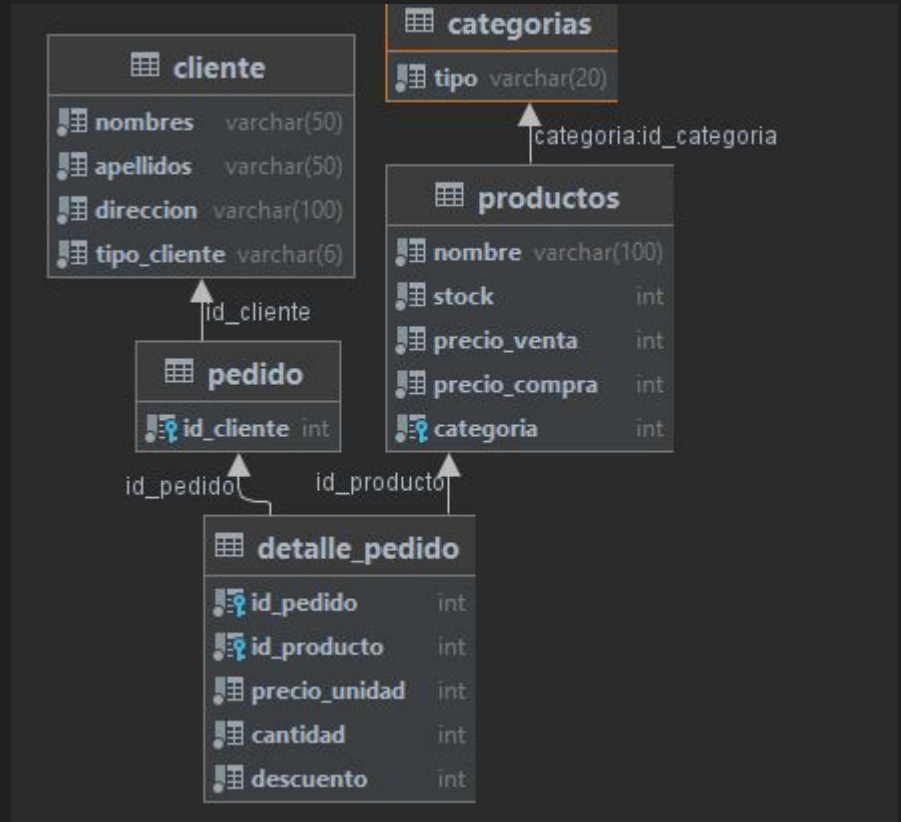
ejemplo



Ejemplo

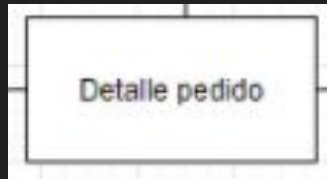
2.2. Que es el modelo lógico en bases de datos relacionales.

Un modelo lógico es el que nos muestra las relaciones entre las distintas entidades buscando mostrar sus diferentes aspectos, además de las dependencias de las diferentes tablas



2.3. Describe y menciona que formas(shapes) se utiliza para graficar un modelo entidad relación.

rectángulo



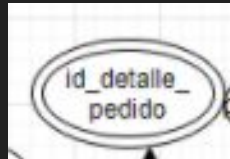
entidad

oval



atributo

doble óvalo



llave primaria

rombo



relaciones

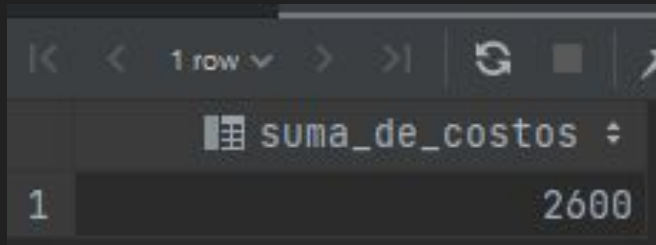
2.4. Qué es una función de agregación.

Aquellas que nos devuelven un valor en base un conjunto de datos enviado que devuelve un único valor.

2.5. Muestra ejemplo del uso de 2 funciones de agregación.

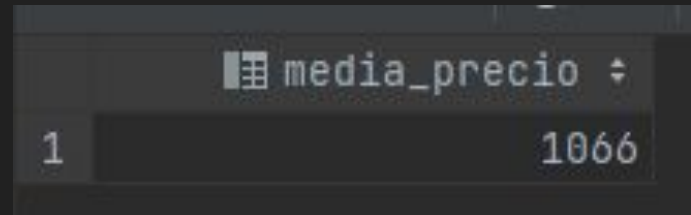
```
select SUM(dp.precio_unidad) as suma_de_costos  
from detalle_pedido as dp
```

```
select AVG(prod.precio_compra) as media_precio  
from productos as prod
```



A screenshot of a database query result. At the top, there is a toolbar with navigation icons and a dropdown menu showing '1 row'. Below the toolbar, the column header is 'suma_de_costos'. The result table has one row with the value '2600'.

	suma_de_costos
1	2600

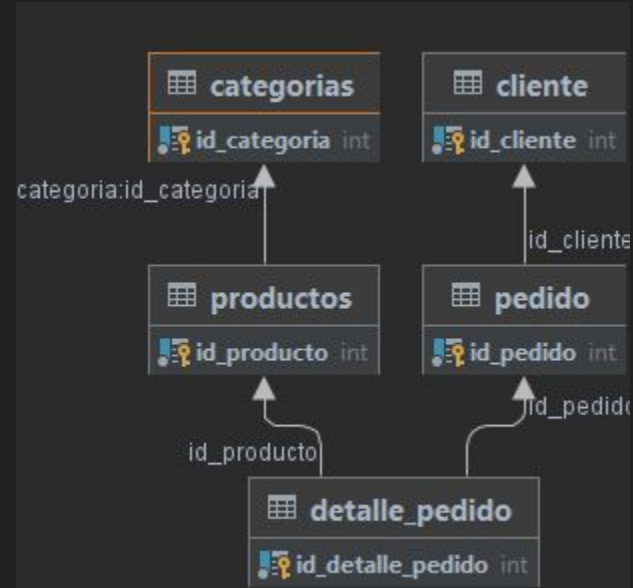


A screenshot of a database query result. The column header is 'media_precio'. The result table has one row with the value '1066'.

	media_precio
1	1066

2.6. Muestra un ejemplo del uso de JOINS.

En base al diagrama una relación y un ejemplo de joint



```
select cat.tipo, prod.nombre
from productos as prod
inner join categorias cat on cat.id_categoria = prod.categoria
where prod.nombre = 'refrigerador'
```

2.7. Qué es SQL y NoSQL.

SQL

Son los que permiten relacionar la información en filas y columnas



NoSQL

Son los que almacenan la información como un documento ideales para grandes volúmenes de información constante



2.8. A que se refiere cuando se habla de ISO, que es una ISO.

Al estándar establecido para facilitar el entendimiento en diferentes aspectos

2.9. Quien creo el modelo entidad relación o mas conocido como E-R

Peter Chen (también conocido como Peter Pin-Shan Chen)

RESEÑA

- **Dr. PETER CHEN**



Creador del modelo E-R. Durante muchos años, se ha utilizado este modelo para representar las interdependencias entre los datos. Desde los 70 hasta ahora, el modelo ha sufrido gran cantidad de cambios y modificaciones, introducidos no necesariamente por su autor, hasta llegar al punto de coexistir múltiples variantes de los diagramas Entidad/Relación.

2.10. Crear una función que permita sumar 3 números .

```
create function suma_de_numeros(@numA integer, @numC integer, @numB integer)
returns integer
begin
    declare @respuesta integer;

    set @respuesta= @numA+@numB+@numC;

    return @respuesta;
end
```

```
select dbo.suma_de_numeros( @numA: 1, @numC: 2, @numB: 10)
```

suma_de_numeros()

Output x

1 row > > | ↺ ■ ↗

<anonymous> ▾

1

13

3. Manejo de consultas

- 3.1. Mostrar los productos(Nombre y stock) con stock mayor igual a 10.
- 3.2. Mostrar el nombre del producto y la categoría de los productos pertenecen a la categoría de "electrodomesticos".
- 3.3. Que productos(nombre) tiene el pedido con id igual a = 1.
- 3.4. Cuantos(count) productos tiene el pedido con id igual a = 2.
- 3.5. Crear una función que permita sumar 3 números.
- 3.6. Crear una función que permita restar 3 números.
- 3.7. Cómo unificaría en una sola función el ejercicio 3.5 y 3.7(los dos anteriores).

3.1. Mostrar los productos(Nombre y stock) con stock mayor igual a 10.

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

Output		the_delicious.dbo.productos	
1 row		nombre	stock
1		refrigerador	15

3.2. Mostrar el nombre del producto y la categoría de los productos pertenecen a la categoría de "electrodomesticos".

```
select prod.nombre, cat.tipo
  from categorias as cat
    inner join productos as prod on prod.categoria = cat.id_categoria
 where cat.tipo='electrodomesticos'
```

2 rows ▾			
	nombre		tipo
1	refrigerador		electrodomesticos
2	microonda		electrodomesticos

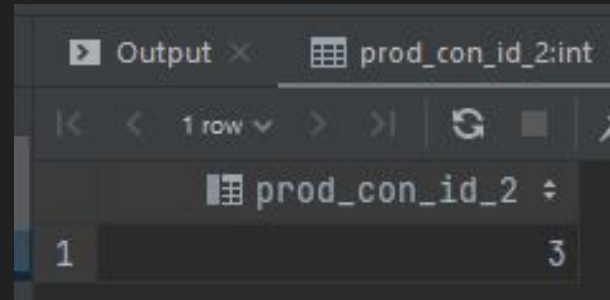
3.3. Que productos(nombre) tiene el pedido con id igual a = 1.

```
select prod.nombre, ped.id_cliente  
from detalle_pedido as dp  
inner join productos as prod on dp.id_producto= prod.id_producto  
inner join pedido as ped on ped.id_pedido= dp.id_pedido  
where ped.id_pedido=1
```

2 rows ▾			
	nombre		id_cliente
1	refrigerador		1
2	microonda		1

3.4. Cuantos(count) productos tiene el pedido con id igual a = 2.

```
select count(ped.id_pedido) as prod_con_id_2  
from detalle_pedido as dp  
inner join productos as prod on dp.id_producto= prod.id_producto  
inner join pedido as ped on ped.id_pedido= dp.id_pedido
```

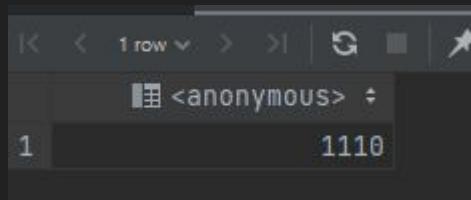


The screenshot shows a database query output window. The title bar indicates the column name 'prod_con_id_2:int'. The window displays a single row with the value 3. The interface includes navigation buttons and a row count indicator.

prod_con_id_2
3

3.5. Crear una función que permita sumar 3 números.

```
--3.5. Crear una función que permita sumar 3 números.  
create function suma_de_numerosA(@numA integer, @numC integer, @numB integer)  
returns integer  
begin  
    declare @respuesta integer;  
  
    set @respuesta= @numA+@numB+@numC;  
  
    return @respuesta;  
end  
  
select dbo.suma_de_numerosA( @numA: 10, @numC: 100, @numB: 1000)
```



The screenshot shows a SQL query result window with a toolbar at the top containing navigation icons and a '1 row' indicator. Below the toolbar, the text '<anonymous>' is displayed. The main area shows a single row with the index '1' and the value '1110'.

1	1110
---	------

3.6. Crear una función que permita restar 3 números.

```
create function resta_de_numeros(@numA integer, @numC integer, @numB integer)
returns integer
begin
    declare @respuesta integer;

    set @respuesta= @numA-@numB-@numC;

    return @respuesta;
end

select dbo.resta_de_numeros( @numA: 10, @numC: 100, @numB: 1000)
```

<anonymous>	
1	-1090

3.7. Cómo unificar en una sola función el ejercicio 3.5 y 3.7 (los dos anteriores).

Fallido

```
create function operacion_entre_numeros(@numA integer, @numC integer, @numB integer)
returns integer
begin

    declare @suma integer;
    set @suma= @numA+@numB+@numC;

    declare @resta integer;
    set @resta=@numA-@numB-@numC;
    return @resta
    return @suma;
end
```

<anonymous>	
1	-1090