



# **PROCESUAL HITO 3**

## **BASE DE DATOS I**

**ESTUDIANTE: BRITTANY IBLING MARINO QUISPE**

**UNIVERSIDAD: FRANZ TAMAYO**

**CARRERA:INGENIERIA DE SISTEMAS**

**CODIGO: SIS13181271**

# 1.- DISEÑO DE BASE DE DATOS

## 1.1 DADO EL DETALLE EXPLICADO EN LA PARTE INICIAL DE ESTE DOCUEMENTO DEBERIA GENERAR UNA BASE DE DATOS SIMILAR AL SIGUIENTE.

### 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.

### Detalles puntuales y análisis del problema

The delicious FF	
Problema	<p>En función al escenario se identificó que las posibles entidades son <b>categorías, productos, cliente, pedido y detalle pedido</b>, 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 <b>entidades y/o tablas</b>.</p> <ul style="list-style-type: none"><li>• categorías</li><li>• productos</li><li>• cliente</li><li>• pedido</li><li>• detalle_pedido</li></ul> <p>Detalle de las entidades.</p>

detalle_pedido
<b>id_detalle_pedido</b> => primary key id_pedido => Foreign key con la entidad pedido id_producto => Foreign key con la entidad productos precio_unidad cantidad descuento

categorias
<b>id_categoria</b> => Primary key tipo => posibles valores como (juguetes, verduras, etc)
productos
<b>id_producto</b> => Primary key nombre stock precio_venta precio_compra categoria => Foreign key con la entidad categoria
cliente
<b>id_cliente</b> => primary key nombres apellidos direccion tipo_cliente => posibles valores como (GOLD, VIP y NORMAL)
pedido
<b>id_pedido</b> => primary key id_cliente

Ejecute el comando CREATE DATABASE the\_delicious

Posicionarnos en la base que queremos trabajar. Ejecute el comando: USE the\_delicious

```
console_1 [localhost] x
1 CREATE DATABASE the_delicious;
2
3 USE the_delicious;
4
5 CREATE TABLE categorias
6 (
7     id_categoria INT PRIMARY KEY,
8     tipo VARCHAR(20)
9 );
10
11 CREATE TABLE productos
12 (
13     id_producto INT PRIMARY KEY ,
14     nombre VARCHAR(100),
15     stock INT ,
16     precio_venta INT,
17     precio_compra INT,
18     categoria INT,
19     FOREIGN KEY (categoria) REFERENCES categorias (id_categoria),
20 );
```

```
22 CREATE TABLE cliente
23 (
24     id_cliente INT PRIMARY KEY,
25     nombres VARCHAR(50),
26     apellidos VARCHAR(50) ,
27     direccion VARCHAR(100),
28     tipo_cliente VARCHAR(20)
29 );
30
31
32 CREATE TABLE pedido
33 (
34     id_pedido INT PRIMARY KEY,
35     id_cliente INT,
36     FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente)
37 );
38
39 CREATE TABLE detalle_pedido
40 (
41     id_detalle_pedido INT PRIMARY KEY,
42     id_pedido INT
43 );
```

productos

Para crear una tabla me apoye en la clausula: CREATE TABLE nombre\_tabla(...);  
Como agregar registros a una tabla. Me apoye en la clausula INSERT INTO nombre\_tabla

```
37 );  
38  
39 CREATE TABLE detalle_pedido  
40 (  
41     id_detalle_pedido INT PRIMARY KEY,  
42     id_pedido INT,  
43     id_producto INT,  
44     precio_unidad INT,  
45     cantidad INT,  
46     descuento INT,  
47     FOREIGN KEY (id_pedido) REFERENCES pedido (id_pedido),  
48     FOREIGN KEY (id_producto) REFERENCES productos (id_producto),  
49 );  
50  
51 INSERT INTO categorias (id_categoria, tipo)  
52 VALUES ('1', 'electrodomestico');  
53 INSERT INTO categorias (id_categoria, tipo)  
54 VALUES ('2', 'juguetes');  
55 INSERT INTO categorias (id_categoria, tipo)  
56 VALUES ('3', 'verduras');
```

productos

```
37 );  
38  
39 CREATE TABLE detalle_pedido  
40 (  
41     id_detalle_pedido INT PRIMARY KEY,  
42     id_pedido INT,  
43     id_producto INT,  
44     precio_unidad INT,  
45     cantidad INT,  
46     descuento INT,  
47     FOREIGN KEY (id_pedido) REFERENCES pedido (id_pedido),  
48     FOREIGN KEY (id_producto) REFERENCES productos (id_producto),  
49 );  
50  
51 INSERT INTO categorias (id_categoria, tipo)  
52 VALUES ('1', 'electrodomestico');  
53 INSERT INTO categorias (id_categoria, tipo)  
54 VALUES ('2', 'juguetes');  
55 INSERT INTO categorias (id_categoria, tipo)  
56 VALUES ('3', 'verduras');
```

productos

```

58 INSERT INTO productos (id_producto, nombre, stock, precio_venta, precio_compra, categoria)
59 VALUES ('1', 'refrigerador', '15', '1500', '1000', '1');
60 INSERT INTO productos (id_producto, nombre, stock, precio_venta, precio_compra, categoria)
61 VALUES ('2', 'microondas', '4', '800', '500', '1');
62 INSERT INTO productos (id_producto, nombre, stock, precio_venta, precio_compra, categoria)
63 VALUES ('3', 'los_vengadores', '2', '2500', '1700', '2');
64
65 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
66 VALUES ('1', 'nombre_cliente1', 'apellidos_cliente1', '6 de agosto Avenida', 'GOLD');
67 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
68 VALUES ('2', 'nombre_cliente2', 'apellidos_cliente2', 'Plaza Avaroa', 'VIP');
69 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
70 VALUES ('3', 'nombre_cliente3', 'apellidos_cliente3', 'Plaza del estudiante', 'NORMAL');
71 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
72 VALUES ('4', 'nombre_cliente4', 'apellidos_cliente4', 'Teatro al aire', 'NORMAL');
73
74 INSERT INTO pedido (id_pedido, id_cliente)
75 VALUES ('1', '1');
76 INSERT INTO pedido (id_pedido, id_cliente)
77 VALUES ('2', '2');

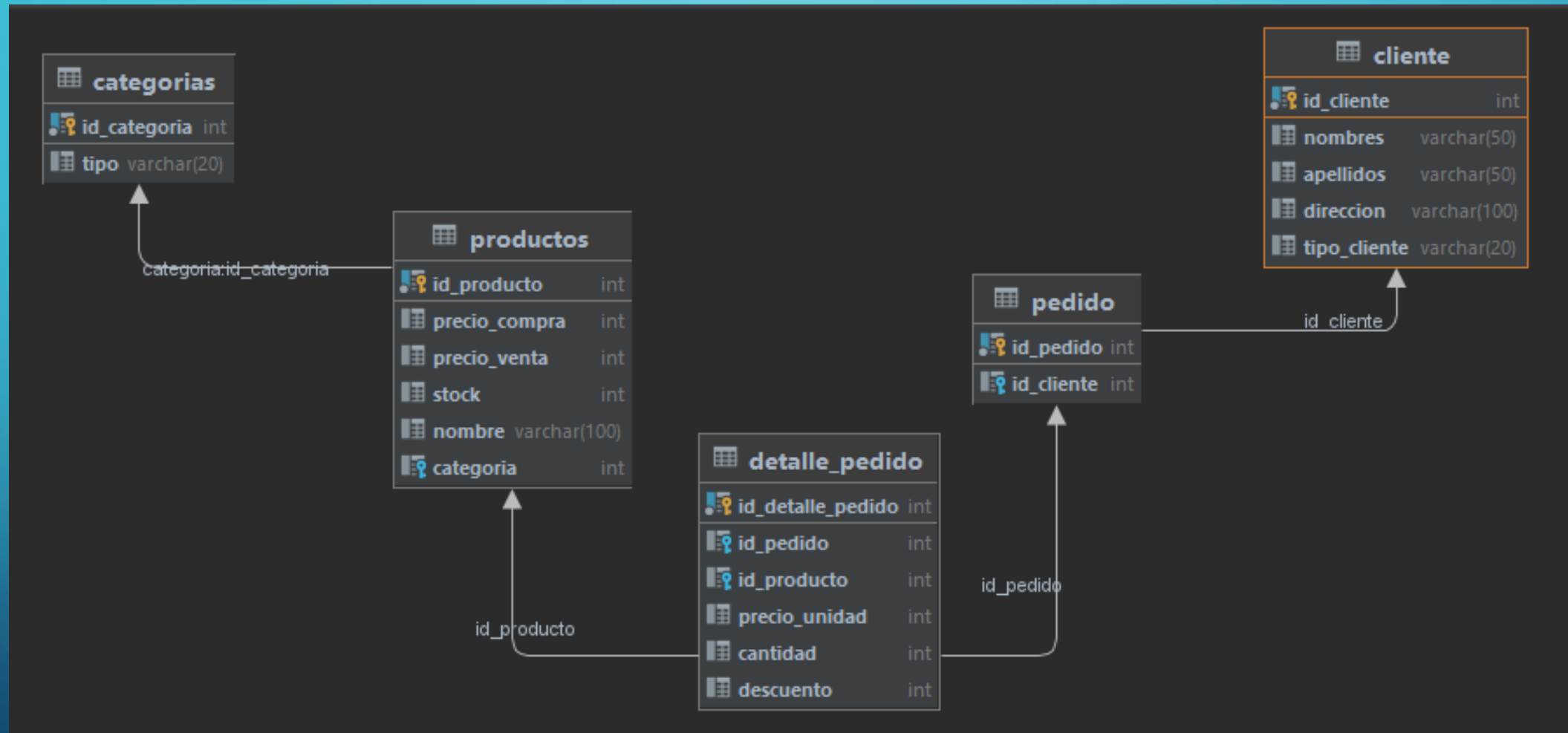
```

```

console_1 [localhost] x
67 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
68 VALUES ('2', 'nombre_cliente2', 'apellidos_cliente2', 'Plaza Avaroa', 'VIP');
69 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
70 VALUES ('3', 'nombre_cliente3', 'apellidos_cliente3', 'Plaza del estudiante', 'NORMAL');
71 INSERT INTO cliente (id_cliente, nombres, apellidos, direccion, tipo_cliente)
72 VALUES ('4', 'nombre_cliente4', 'apellidos_cliente4', 'Teatro al aire', 'NORMAL');
73
74 INSERT INTO pedido (id_pedido, id_cliente)
75 VALUES ('1', '1');
76 INSERT INTO pedido (id_pedido, id_cliente)
77 VALUES ('2', '2');
78
79 INSERT INTO detalle_pedido (id_detalle_pedido, id_pedido, id_producto, precio_unidad, cantidad, descuento)
80 VALUES ('1', '1', '1', '1000', '2', '0');
81 INSERT INTO detalle_pedido (id_detalle_pedido, id_pedido, id_producto, precio_unidad, cantidad, descuento)
82 VALUES ('2', '1', '2', '800', '1', '0');
83 INSERT INTO detalle_pedido (id_detalle_pedido, id_pedido, id_producto, precio_unidad, cantidad, descuento)
84 VALUES ('3', '2', '2', '800', '1', '0');
85

```

## DIAGRAMA





## **2.- MANEJO DE CONCEPTOS**

### **2.1.- ¿Qué es el modelo entidad relación?**

**Un diagrama entidad relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los modelo entidad relación se usan para diseñar bases de datos relacionados en los campos de ingeniería de software, Sistemas de información empresarial, educación e investigación.**

### **2.2.- ¿Qué es el modelo lógico en base de datos relacionales?**

**Un modelo lógico de datos se usa para que implemente el sistema gestión de bases de datos, pero es independiente de los aspectos físicos.**

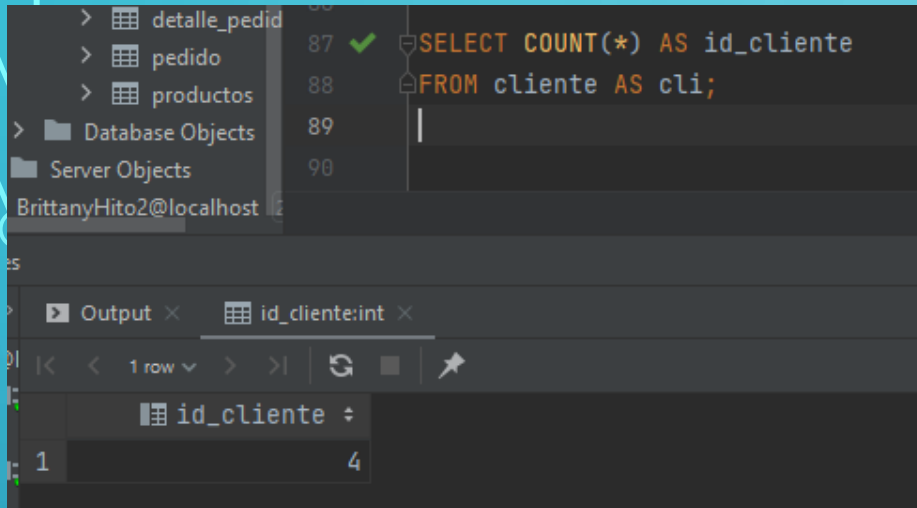
### **2.3.- Describe y menciona que formas (shapes) se utiliza para graficar un modelo entidad relación.**

**La representación grafica de las relaciones se realiza con un rombo al que se le unen líneas que se dirigen a las entidades, Las relaciones tienen nombre. Shape se trata del archivo principal y almacena la información geométrica de los elementos de la capa en formato vectorial. Pueden contener puntos, líneas o polígonos y cada vértice lleva implícitas sus coordenadas en un sistema de referencia concreto. Se compone de una cabecera con información general sobre el tipo de shapefile y un numero variable de registros, que a su vez pueden estar compuestos por varias entidades geométricas independientes.**

### **2.4.- ¿Qué es una función de agregacion?**

**Las funciones de agregación realizan análisis estadísticos sobre números en varios campos de un registro, campos Relacionados tanto si se muestran en un portal como si no, campos repetidos.**

## 2.5.-Muestre ejemplo del uso de 2 funciones de agregación

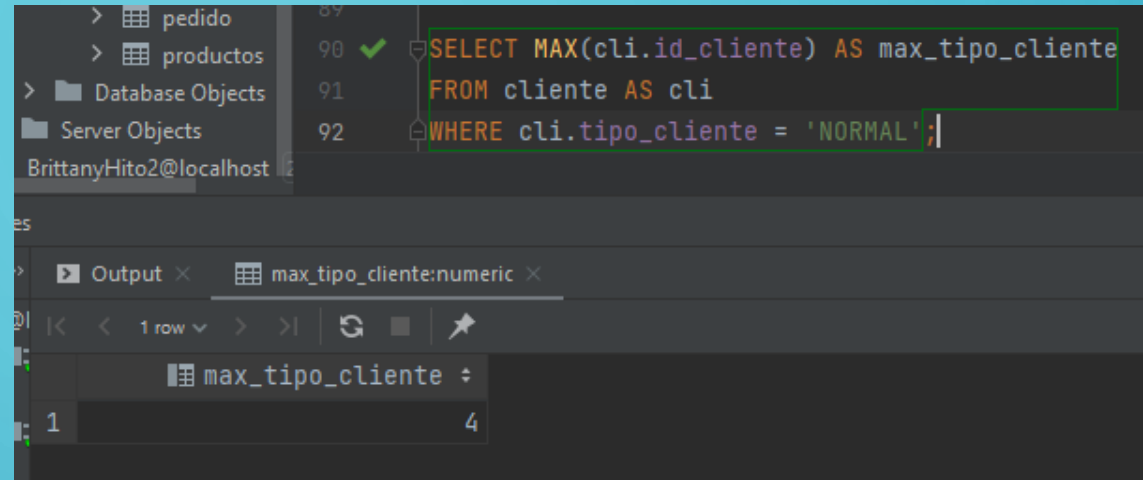


The screenshot shows a SQL query window with the following code:

```
SELECT COUNT(*) AS id_cliente  
FROM cliente AS cli;
```

The query is executed successfully. The output window shows a single row with the value 4 for the column id\_cliente.

id_cliente
4



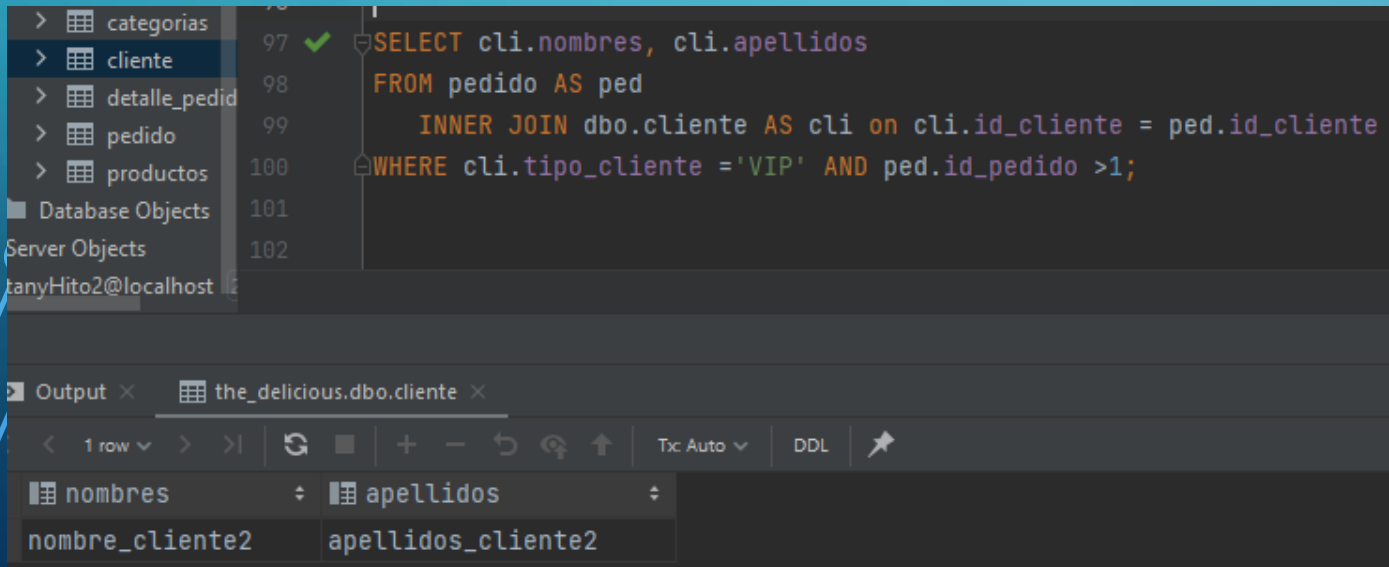
The screenshot shows a SQL query window with the following code:

```
SELECT MAX(cli.id_cliente) AS max_tipo_cliente  
FROM cliente AS cli  
WHERE cli.tipo_cliente = 'NORMAL';
```

The query is executed successfully. The output window shows a single row with the value 4 for the column max\_tipo\_cliente.

max_tipo_cliente
4

## 2.6.-Muestre un ejemplo del uso de JOINS



The screenshot shows a SQL query window with the following code:

```
SELECT cli.nombres, cli.apellidos  
FROM pedido AS ped  
INNER JOIN dbo.cliente AS cli on cli.id_cliente = ped.id_cliente  
WHERE cli.tipo_cliente = 'VIP' AND ped.id_pedido > 1;
```

The query is executed successfully. The output window shows a single row with the values nombre\_cliente2 and apellidos\_cliente2 for the columns nombres and apellidos respectively.

nombres	apellidos
nombre_cliente2	apellidos_cliente2



## 2.7.-¿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 y NoSQL permite distribuir grandes cantidades de información; mientras que SQL facilita distribuir bases de datos relaciones.

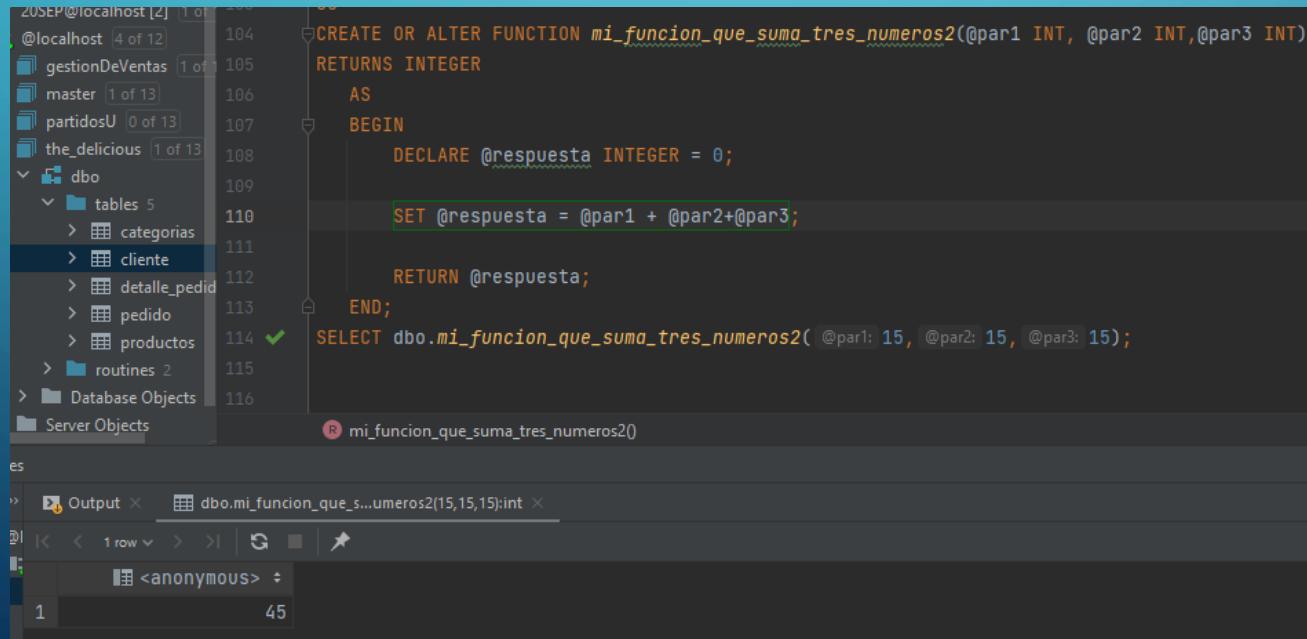
## 2.8.- A que se refiere cuando se habla de ISO, que es una ISO

ISO se refiere al conjunto de normas orientadas a ordenar la gestión de una empresa en sus distintos ámbitos. La alta competencia internacional acentuada por los procesos globalizadores de la economía y el mercado, ISO se trata de un órgano cuya principal función es la de crear normas de carácter internacional. Permite un reconocimiento internacional para así asegurar la homogeneización de aspectos tan relevantes como puede ser el medio ambiente, la seguridad en el entorno laboral o la calidad de los productos.

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

El que creo el modelo entidad relación se llama Peter Chen, también conocido como Peter Pin-Shan Chen.

## 2.10.- Crear una función que permita sumar 3 numeros



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Database Objects' tree is expanded to show the 'dbo' database, which contains several tables and routines. The 'mi\_funcion\_que\_suma\_tres\_numeros2' function is highlighted. The main pane displays the SQL code for creating and executing the function. The code defines a function that takes three integers as input and returns their sum. The function is then executed with the values 15, 15, and 15, resulting in an output of 45.

```
CREATE OR ALTER FUNCTION mi_funcion_que_suma_tres_numeros2(@par1 INT, @par2 INT,@par3 INT)
RETURNS INTEGER
AS
BEGIN
    DECLARE @respuesta INTEGER = 0;

    SET @respuesta = @par1 + @par2+@par3;

    RETURN @respuesta;
END;
SELECT dbo.mi_funcion_que_suma_tres_numeros2( @par1: 15, @par2: 15, @par3: 15);
```

The output window shows the result of the function execution:

1	45

### 3.- MANEJO DE CONSULTAS

#### 3.1.- Mostrar los productos(nombre y stock) con stock mayor igual a 10

```
89 ✓ SELECT pro.nombre, pro.stock
90 FROM productos AS pro
91 WHERE pro.stock >= 10
92
```

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 “electrodomésticos”

```
6 ✓ SELECT pro.nombre, cat.tipo
7 FROM categorias AS cat
8 INNER JOIN dbo.productos AS pro on pro.categoria = cat.id_categoria
9 WHERE pro.id_producto > 0 AND cat.id_categoria > 0 ;
10
```

Output × Result 4 ×

3 rows

	nombre	tipo
1	refrigerador	electrodomestico
2	microondas	electrodomestico
3	los_vengadores	juguetes

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

```
92 ✓ SELECT pro.nombre
93     FROM detalle_pedido AS det
94     INNER JOIN dbo.productos AS pro on pro.id_producto = det.id_pedido
95     WHERE det.id_pedido = 1 ;
96
```

Output × the\_delicious.dbo.productos ×

2 rows

	nombre
1	refrigerador
2	refrigerador

### 3.4.-Cuantos (count) productos tiene el pedido con id igual a=1

```
101 ✓ SELECT COUNT (pro.nombre)
102     FROM detalle_pedido AS det
103     INNER JOIN dbo.productos AS pro on pro.id_producto = det.id_pedido
104     WHERE det.id_pedido = 2 ;
105
```

Output × COUNT (pro.nombre):int ×

1 row

	<anonymous>
1	1

### 3.5.-Crear una función que permita sumar 3 números.

```
103     INNER JOIN dbo.productos AS pro on pro.id_producto = det.id_pedido
104 WHERE det.id_pedido = 2 ;
105
106 GO
107 CREATE OR ALTER FUNCTION mi_funcion_que_suma_tres_numeros2(@par1 INT, @par2 INT, @par3 INT)
108 RETURNS INTEGER
109 AS
110 BEGIN
111     DECLARE @respuesta INTEGER = 0;
112
113     SET @respuesta = @par1 + @par2 + @par3;
114
115     RETURN @respuesta;
116 END;
117 ✓ SELECT dbo.mi_funcion_que_suma_tres_numeros2( @par1: 15, @par2: 13, @par3: 7);
118
```

Output ×    dbo.mi\_funcion\_que\_s...numeros2(15,13,7):int ×

1 row ▾    ↺    ⬛    ⚡

<anonymous> ▾

1	35
---	----

### 3.6.- Crear una función que permita restar 3 números

```
123 GO
124 CREATE OR ALTER FUNCTION mi_funcion_que Resta tres numeros2(@par1 INT, @par2 INT,@par3 INT)
125 RETURNS INTEGER
126 AS
127 BEGIN
128     DECLARE @respuesta INTEGER = 0;
129
130     SET @respuesta = @par1 - @par2 - @par3;
131
132     RETURN @respuesta;
133 END;
134 ✓ SELECT dbo.mi_funcion_que Resta tres numeros2( @par1: 30, @par2: 7, @par3: 7);
135
136
```

mi\_funcion\_que Resta tres numeros2()

Output × dbo.mi\_funcion\_que\_r...\_numeros2(30,7,7):int ×

1 row

<anonymous>

1 16