

Materia 1

Materia 2

Materia 3

Materia 4

Materia 5

Materia 6



Horario

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-9:00					
9:00-10:00					
10:00-11:00					
11:00-12:00					
12:00-1:00					
1:00-2:00					

Evaluación	
Tareas	50%
Examen	30%
Asistencia	10%
Participación	10%
Total	100%

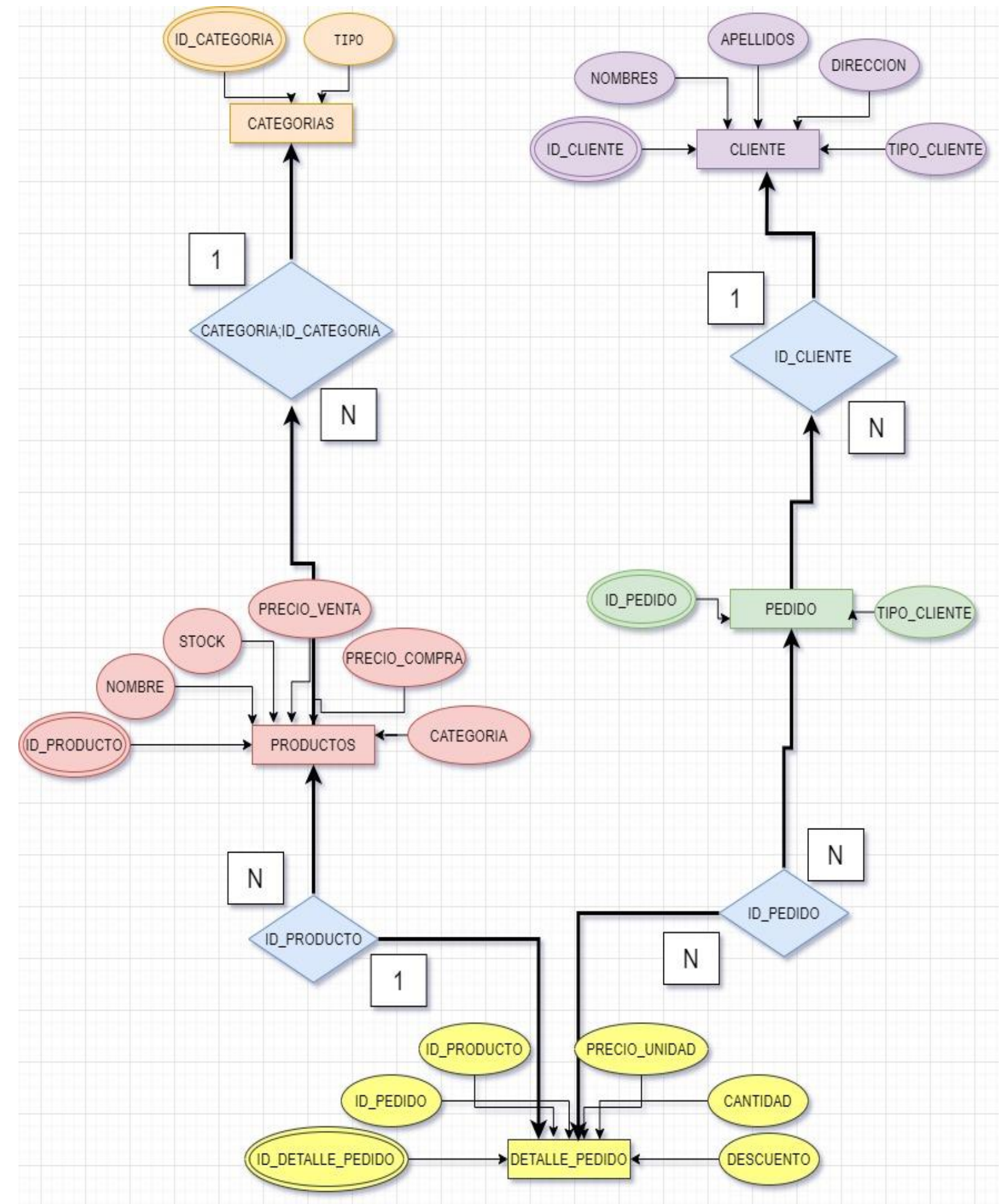


Ana Cristina Calderón Ortega
Segundo Semestre
Ingeniería de Sistemas

1. Diseño de base de datos.

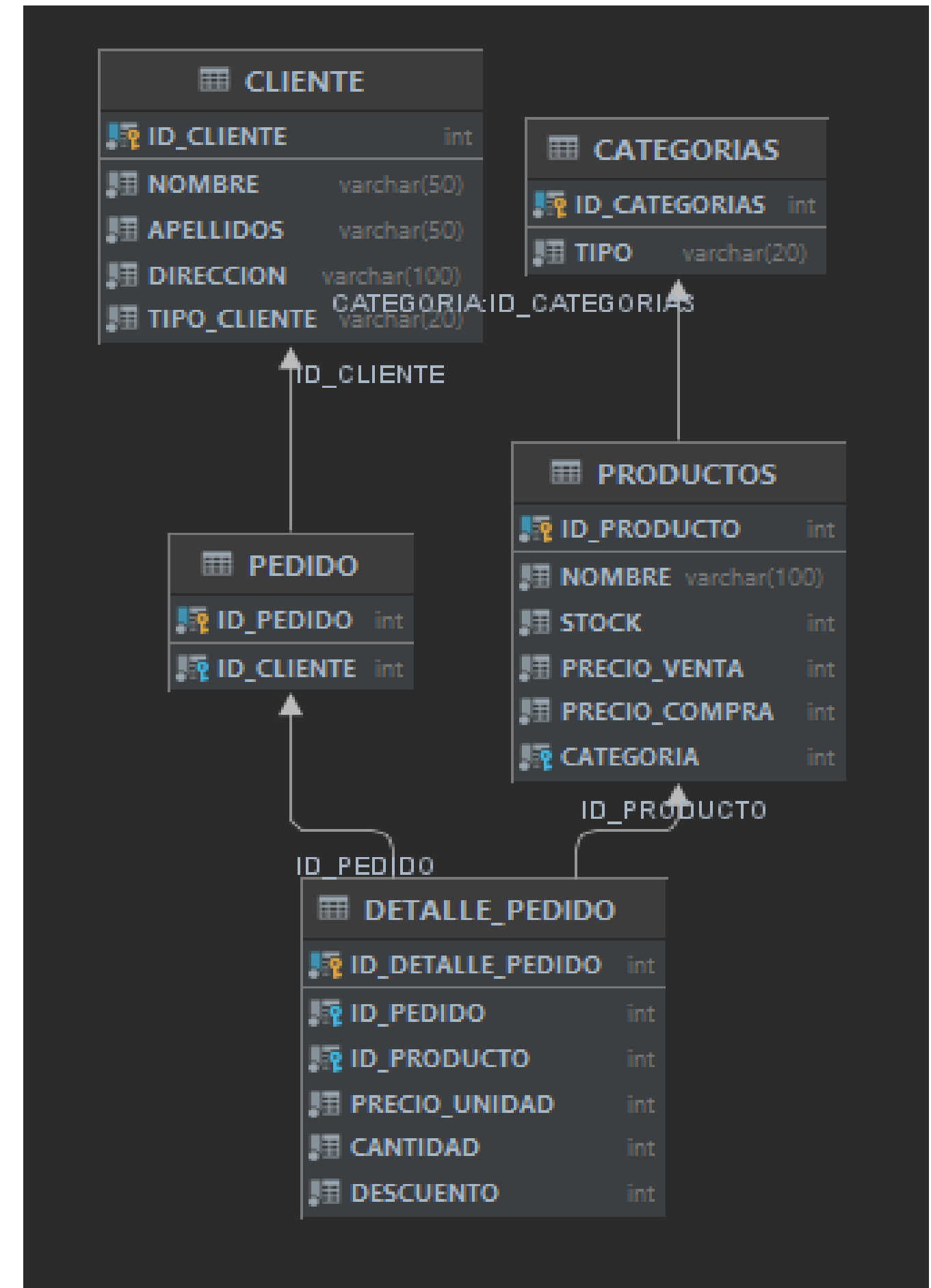
1.1. Dado el detalle explicado en la parte inicial de este documento debería generar el modelo entidad relación.

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



Después de generar el modelo lógico de la base de datos. El mismo debería quedar similar a lo siguiente:

- **Utilizar Datagrip para poder generar el diagrama**



1.3. Agregar los siguientes registros a las tablas

```
CREATE DATABASE THE_DELICIOUS_FF;
```

```
USE THE_DELICIOUS_FF;
```

```
CREATE TABLE CATEGORIAS
```

```
(
  ID_CATEGORIAS INT NOT NULL PRIMARY KEY IDENTITY ,
  TIPO VARCHAR(20) NOT NULL
);
```

```
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,
  CATEGORIA INT NOT NULL,
  FOREIGN KEY (CATEGORIA) REFERENCES CATEGORIAS(ID_CATEGORIAS)
);
```

```
CREATE TABLE CLIENTE
```

```
(
  ID_CLIENTE INT NOT NULL PRIMARY KEY IDENTITY ,
  NOMBRE VARCHAR(50) NOT NULL,
  APELLIDOS VARCHAR(50) NOT NULL,
  DIRECCION VARCHAR(100) NOT NULL,
  TIPO_CLIENTE VARCHAR(20) NOT NULL
);
```

```
CREATE TABLE PEDIDO
```

```
(
  ID_PEDIDO INT NOT NULL PRIMARY KEY IDENTITY ,
  ID_CLIENTE INT NOT NULL,
  FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTE(ID_CLIENTE)
);
```

```
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)
);
```

```
INSERT INTO CATEGORIAS(TIPO) VALUES ('ELECTRODOMESTICOS');
```

```
INSERT INTO CATEGORIAS(TIPO) VALUES ('JUGUETES');
```

```
INSERT INTO CATEGORIAS(TIPO) VALUES ('VERDURAS');
```

```
INSERT INTO PRODUCTOS(NOMBRE, STOCK, PRECIO_VENTA, PRECIO_COMPRA, CATEGORIA) VALUES
('REFRIGERADOR',15,1500,1000,1),
('MICROONDA',4,800,500,1),
('LOS VENGADORES (INCLUYE TANOS)',2,2500,1700,2);
```

```
INSERT INTO CLIENTE(NOMBRE, APELLIDOS, DIRECCION, TIPO_CLIENTE) VALUES
('NOMBRE_CLIENTE1','APELLIDO_CLIENTE_1','6 DE AGOSTO EDIFICIO FERNANDEZ','GOLD'),
('NOMBRE_CLIENTE2','APELLIDO_CLIENTE_2','PLAZA ABAROA','VIP'),
('NOMBRE_CLIENTE3','APELLIDO_CLIENTE_3','PLAZA DEL ESTUDIANTE','NORMAL'),
('NOMBRE_CLIENTE4','APELLIDO_CLIENTE_4','TEATRO AL AIRE LIBRE','NORMAL');
```

```
INSERT INTO PEDIDO(ID_CLIENTE) VALUES (1);
```

```
INSERT INTO PEDIDO(ID_CLIENTE) VALUES (2);
```

```
INSERT INTO DETALLE_PEDIDO(ID_PEDIDO, ID_PRODUCTO, PRECIO_UNIDAD, CANTIDAD, DESCUENTO)
VALUES
(1,1,1000,2,0),
(1,2,800,1,0),
(2,2,800,1,0);
```


Horario

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-9:00					
9:00-10:00					
10:00-11:00					
11:00-12:00					
12:00-1:00					
1:00-2:00					

Evaluación	
Tareas	50%
Examen	30%
Asistencia	10%
Participación	10%
Total	100%

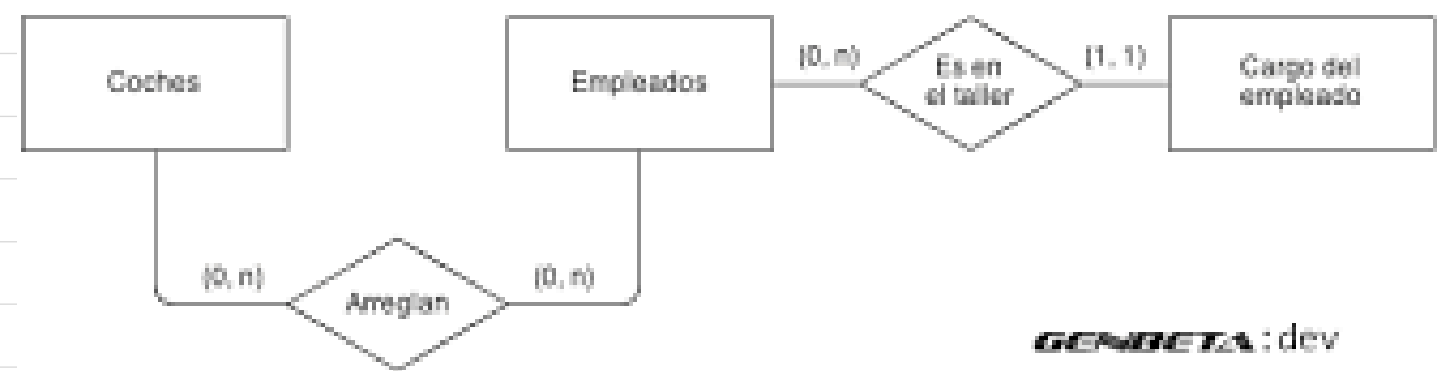


Ana Cristina Calderón Ortega
Segundo Semestre
Ingeniería de Sistemas

2. Manejo de conceptos
2.1. Que es el modelo entidad relación.

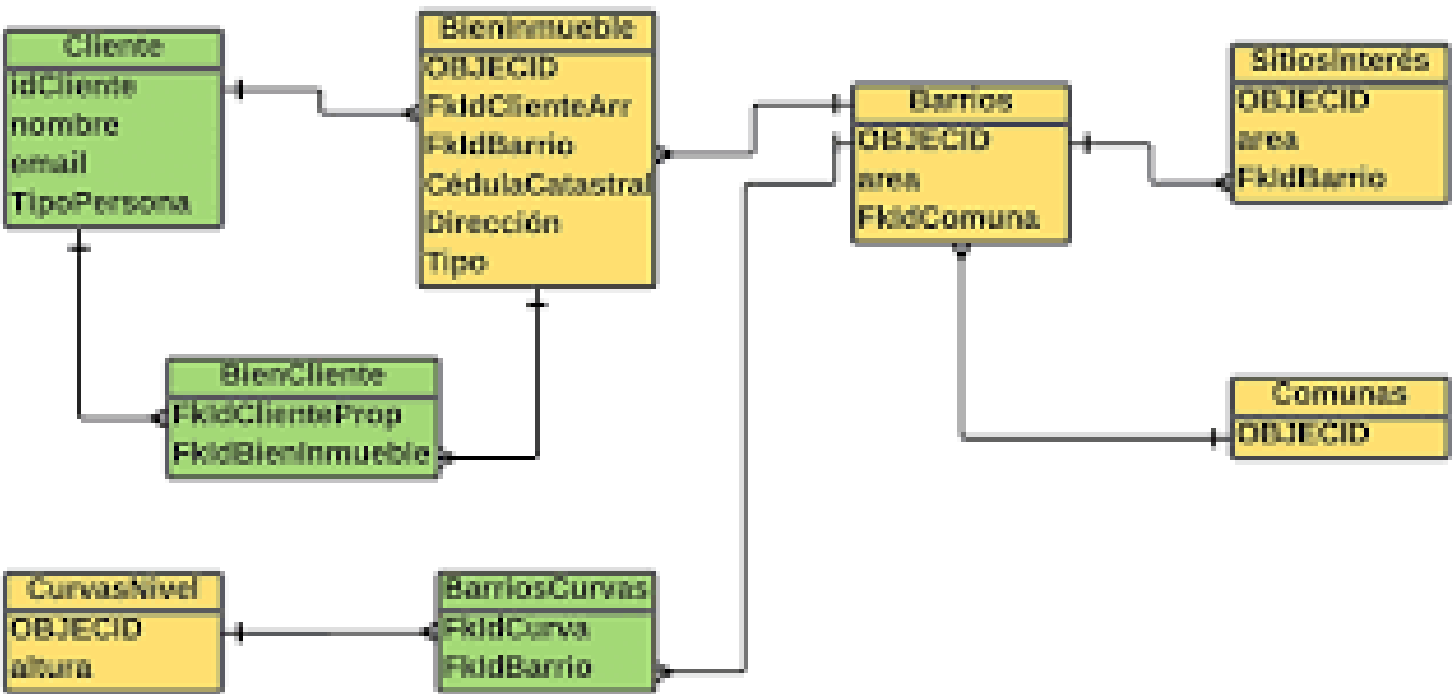
Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra as "entidades". Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales.

Modelo entidad-relación



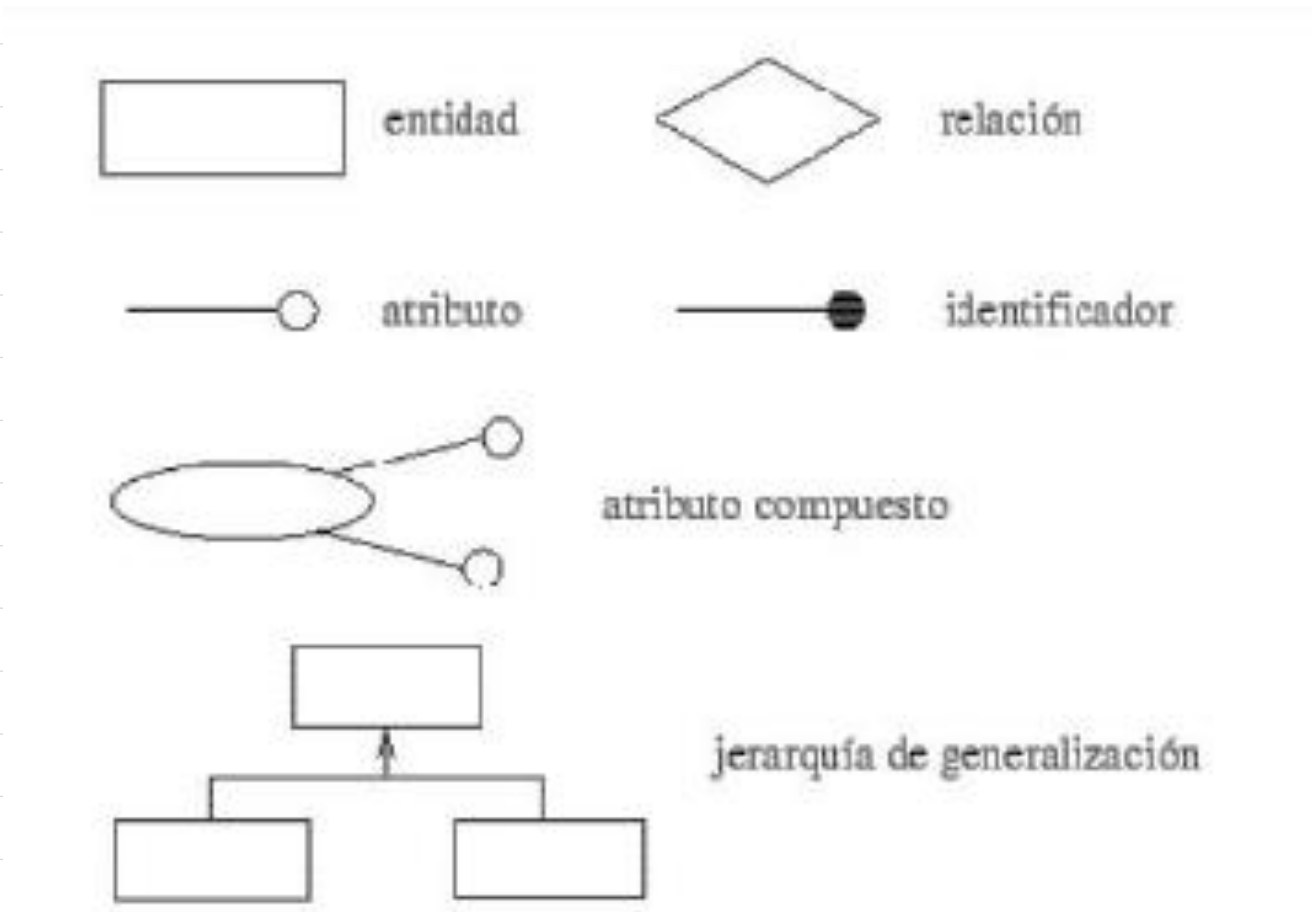
2.2. Que es el modelo lógico en bases de datos relacionales.

El modelo relacional es un modelo lógico que sólo sirve para SGBD relacionales (y no para jerárquicos, o Codasyl, por ejemplo). En el modelo relacional una relación es una tabla mientras que en el entidad/relación es la asociación que se produce entre dos entidades.



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

El diagrama entidad relación es la expresión gráfica del modelo entidad relación. En él las entidades se representan utilizando rectángulos, los atributos por medio de círculos o elipses y las relaciones como líneas que conectan las entidades que tienen algún tipo de vínculo.



2.4. Qué es una función de agregación.

Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos.

Funciones de Agregado	
Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

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

--Maximo monto del precio de compra de los productos registrados en la tabla

```
SELECT max(PRO.PRECIO_COMPRA) AS  
MAXIMO_PRECIO_COMPRA  
FROM PRODUCTOS AS PRO;
```



```
71 (2,2,800,1,0);
72
73 --Maximo monto del precio de compra de los productos registrados en la tabla
74
75 SELECT max(PRO.PRECIO_COMPRA) AS MAXIMO_PRECIO_COMPRA
76 FROM PRODUCTOS AS PRO;
77
78 --Minimo monto del precio de venta de los productos que sean de la categoria 1
79 -- o la categoria electrodomestico
80
81 SELECT MIN(PRO.PRECIO_VENTA) AS MINIMO_PRECIO_COMPRA
82 FROM PRODUCTOS AS PRO
```

MAXIMO_PRECIO_COMPRA
1700

2.6. Muestre un ejemplo del uso de JOINS.

```
--Mostrar el nombre, stock y la categoría a la que pertenecen los productos
-- con una categoría igual a 'ELECTRODOMESTICOS'

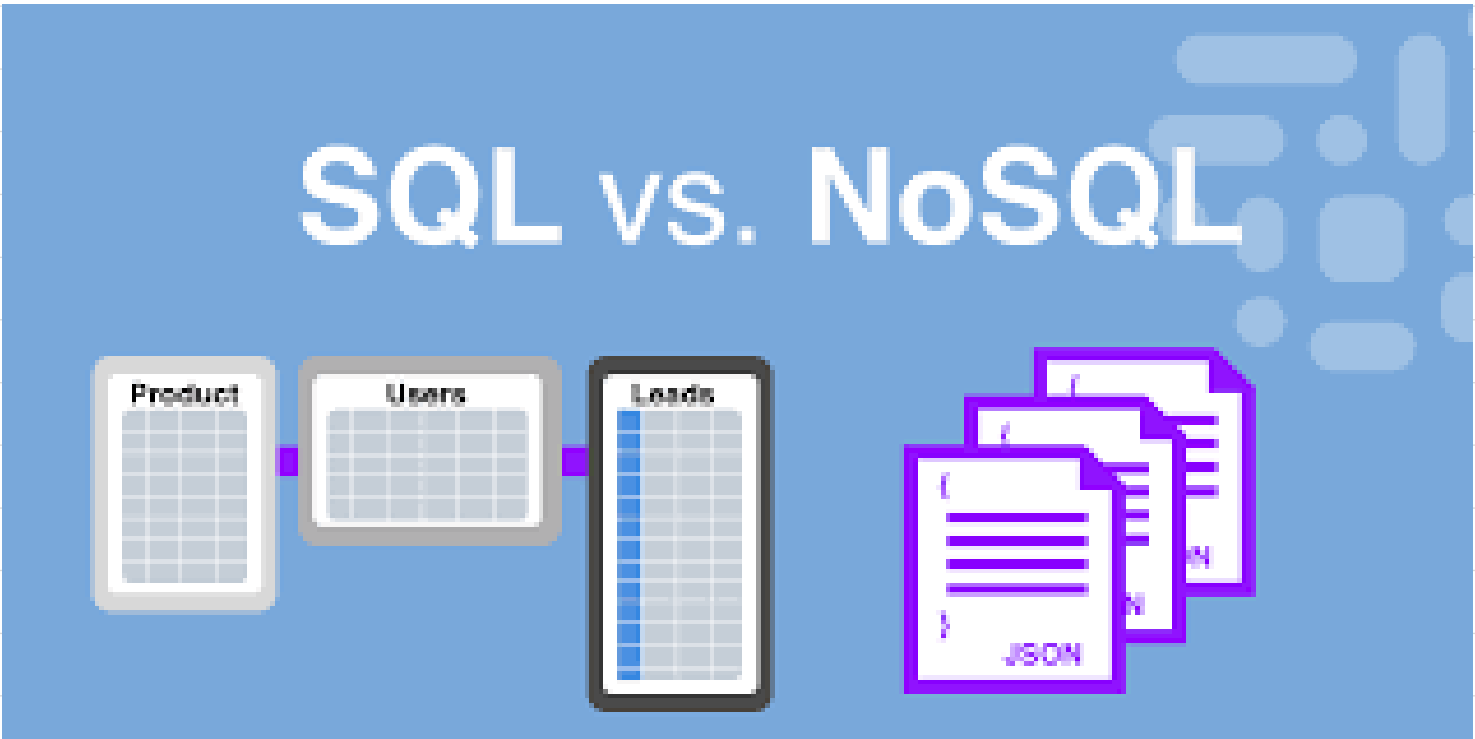
SELECT PRO.NOMBRE, PRO.STOCK, CAT.TIPO
FROM PRODUCTOS AS PRO
INNER JOIN CATEGORIAS AS CAT ON
CAT.ID_CATEGORIAS=PRO.CATEGORIA
WHERE CAT.TIPO='ELECTRODOMESTICOS';
```

```
82 FROM PRODUCTOS AS PRO
83 WHERE PRO.CATEGORIA=1;
84
85 --Mostrar el nombre, stock y la categoría a la que pertenecen los productos
86 -- con una categoría igual a 'ELECTRODOMESTICOS'
87
88 SELECT PRO.NOMBRE, PRO.STOCK, CAT.TIPO
89 FROM PRODUCTOS AS PRO
90 INNER JOIN CATEGORIAS AS CAT ON CAT.ID_CATEGORIAS=PRO.CATEGORIA
91 WHERE CAT.TIPO='ELECTRODOMESTICOS';
92
93
```

	NOMBRE	STOCK	TIPO
1	REFRIGERADOR	15	ELECTRODOMESTICOS
2	MICROONDA	4	ELECTRODOMESTICOS

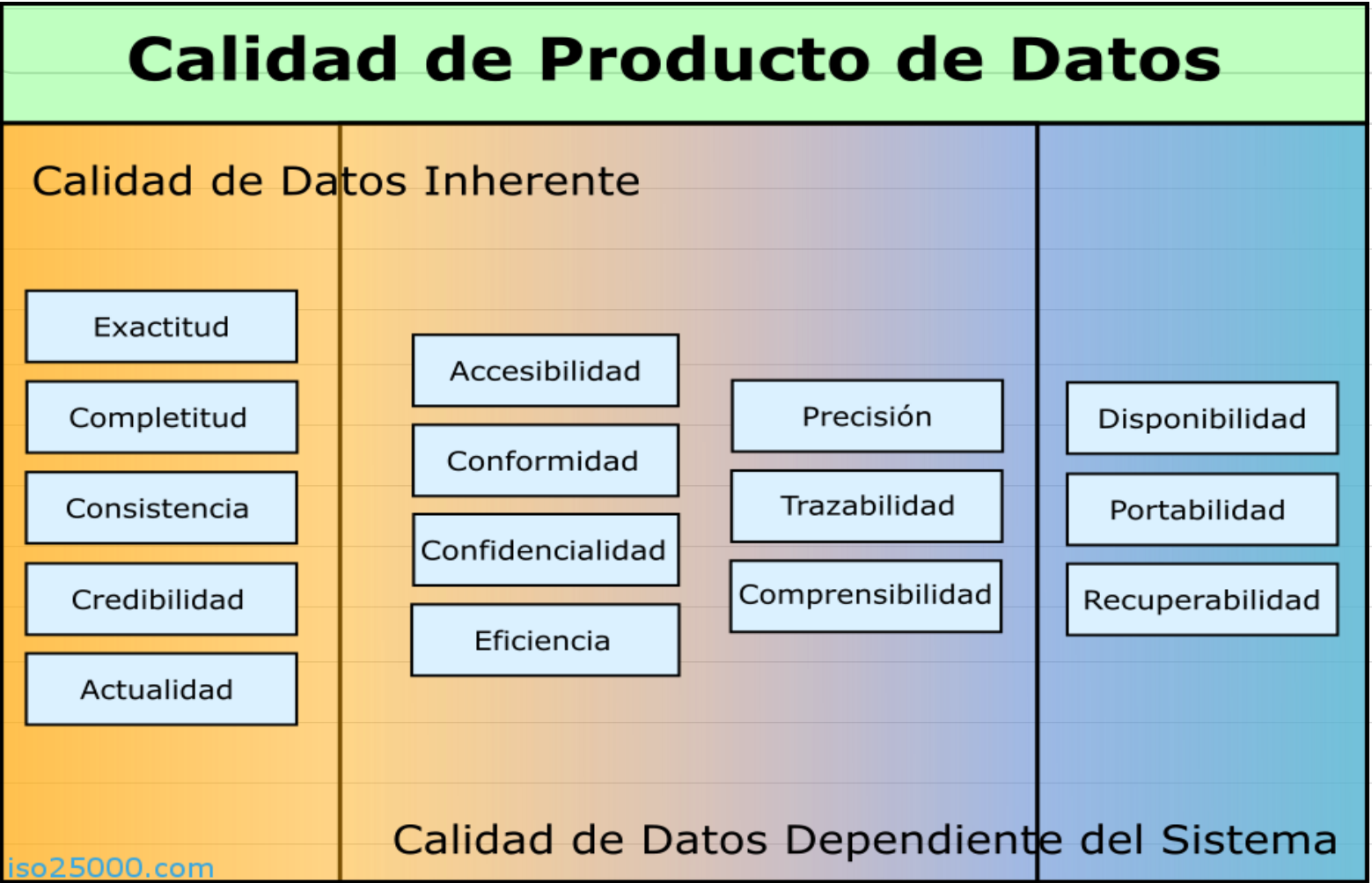
2.7. Qué es SQL y NoSQL.

Una base de datos SQL es una base de datos relacional, escrita en el lenguaje de consulta estructurado SQL . Una base de datos NoSQL o *Not Only SQL* es una base de datos no relacional que no cuenta con un identificador que relacione un conjunto de datos con otro.



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

Al modelo de Calidad de Datos representa los cimientos sobre los cuales se construye un sistema para la evaluación de un producto de datos. En un modelo de Calidad de Datos se establecen las características de Calidad de Datos que se deben tener en cuenta a la hora de evaluar las propiedades de un producto de datos determinado. Las normas ISO son un conjunto de normas orientadas a ordenar la gestión de una empresa en sus distintos ámbitos



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

Peter Chen (también conocido como Peter Pin-Shan 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.



PETER CHEN (PETER
PIN SHAN CHEN)

2.10. Crear una función que permita sumar 3 números.

```
--CREAR UN FUNCION QUE PERMITA SUMAR 3 NUMEROS  
CREATE OR ALTER FUNCTION SUMA_TRES_NUMEROS(@NUM1  
INTEGER, @NUM2 INTEGER, @NUM3 INTEGER)  
RETURNS INTEGER  
BEGIN  
    DECLARE @RESPUESTA INTEGER;  
  
    SET @RESPUESTA=@NUM1 + @NUM2+@NUM3;  
  
    RETURN @RESPUESTA;  
end  
  
SELECT dbo.SUMA_TRES_NUMEROS(12,4,5) AS  
SUMA_TRES_NUMEROS;
```

A screenshot of the SQL Server Enterprise Manager interface. The top pane shows the SQL code for creating the function SUMA_TRES_NUMEROS. The bottom pane shows the execution result of the function call SUMA_TRES_NUMEROS(). The result is displayed in a table with one row and one column, showing the value 21.

```
--CREAR UN FUNCION QUE PERMITA SUMAR 3 NUMEROS  
CREATE OR ALTER FUNCTION SUMA_TRES_NUMEROS(@NUM1 INTEGER, @NUM2 INTEGER, @NUM3 INTEGER)  
RETURNS INTEGER  
BEGIN  
    DECLARE @RESPUESTA INTEGER;  
  
    SET @RESPUESTA=@NUM1 + @NUM2+@NUM3;  
  
    RETURN @RESPUESTA;  
end  
  
SUMA_TRES_NUMEROS()
```

SUMA_TRES_NUMEROS
21

Horario

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-9:00					
9:00-10:00					
10:00-11:00					
11:00-12:00					
12:00-1:00					
1:00-2:00					

Evaluación	
Tareas	50%
Examen	30%
Asistencia	10%
Participación	10%
Total	100%



Ana Cristina Calderón Ortega
Segundo Semestre
Ingeniería de Sistemas

--3. Manejo de consultas

--3.1. 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;
```

--3.2. 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.CATEGORIA = CAT.ID_CATEGORIAS  
WHERE CAT.TIPO='ELECTRODOMESTICOS';
```

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

```
SELECT PRO.NOMBRE, PE.ID_PEDIDO  
FROM PRODUCTOS AS PRO  
INNER JOIN DETALLE_PEDIDO AS DET ON PRO.ID_PRODUCTO =  
DET.ID_PRODUCTO  
    INNER JOIN PEDIDO AS PE ON DET.ID_PEDIDO = PE.ID_PEDIDO  
WHERE PE.ID_PEDIDO=1;
```

```
SELECT PRO.NOMBRE, DET.ID_PEDIDO  
FROM PRODUCTOS AS PRO  
INNER JOIN DETALLE_PEDIDO AS DET ON PRO.ID_PRODUCTO =  
DET.ID_PRODUCTO  
    WHERE DET.ID_PEDIDO=1;
```

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

```
SELECT COUNT(DET.CANTIDAD) AS CUANTOS_PRODUCTOS  
FROM DETALLE_PEDIDO AS DET  
WHERE DET.ID_PEDIDO=2;
```

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

```
CREATE OR ALTER FUNCTION SUMA_TRES_NUMEROS(@NUM1  
INTEGER, @NUM2 INTEGER, @NUM3 INTEGER)  
RETURNS INTEGER  
BEGIN  
    DECLARE @RESPUESTA INTEGER;  
  
    SET @RESPUESTA=@NUM1 + @NUM2+@NUM3;  
  
    RETURN @RESPUESTA;  
end  
  
SELECT dbo.SUMA_TRES_NUMEROS(12,17,9);
```

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


```
CREATE OR ALTER FUNCTION RESTA_TRES_NUMEROS(@NUM1 INTEGER,
@NUM2 INTEGER, @NUM3 INTEGER)
RETURNS INTEGER
BEGIN
    DECLARE @RESPUESTA INTEGER;

    SET @RESPUESTA=@NUM1 - @NUM2-@NUM3;

    RETURN @RESPUESTA;
end

SELECT dbo.RESTA_TRES_NUMEROS(17,12,9);

--3.7. Cómo unificaría en una sola función el ejercicio 3.5 y 3.7(los dos anteriores).
CREATE OR ALTER FUNCTION FUSION(@OPER VARCHAR(20), @NUM1
INTEGER, @NUM2 INTEGER, @NUM3 INTEGER)
RETURNS INTEGER
BEGIN
    DECLARE @RESPUESTA INTEGER;
    IF @OPER='SUMA'
        BEGIN
            SET @RESPUESTA=@NUM1 + @NUM2 +@NUM3;
        end
    ELSE
        BEGIN
            SET @RESPUESTA=@NUM1 - @NUM2-@NUM3;
        end
    RETURN @RESPUESTA;
end
```

```
SELECT dbo.FUSION('SUMA',1,2,3) AS LA_SUMA_ES_ ;
SELECT dbo.FUSION('RESTA',3,2,1) AS LA_RESTA_ES_;
```

The screenshot shows a SQL query in a dark-themed editor. The query is as follows:

```
--Minimo monto del precio de venta de los productos que sean de la categoria 1
-- o la categoria electrodomestico

SELECT MIN(PRO.PRECIO_VENTA) AS MINIMO_PRECIO_COMPRA
FROM PRODUCTOS AS PRO
WHERE PRO.CATEGORIA=1;

--Mostrar el nombre, stock y la categoria a la que pertenecen los productos
-- con una categoria igual a 'ELECTRODOMESTICOS'

SELECT PRO.NOMBRE, PRO.STOCK, CAT.TIPO
```

Below the editor, the 'Output' window is open, showing the result of the query. It displays a single row with the value 800 for the column MINIMO_PRECIO_COMPRA.

MINIMO_PRECIO_COMPRA
800

