

Bases de Datos- SQL Server Procesual Hito 3

Estudiante: Karla Belen Díaz Flores
Ingeniera: William Roddy Barra
Paredes
Semestre: Segundo
Gestión:

2021

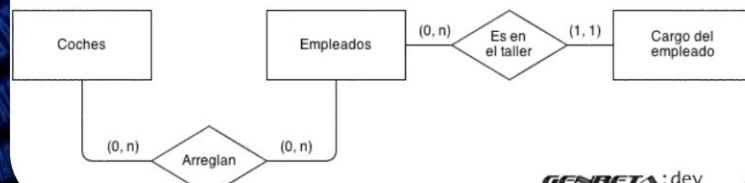


¿Qué es el modelo entidad relación?

Un modelo entidad relación es una herramienta para el diagrama de conceptos de un modelo lógico en nuestra base de datos.

Es el proceso que permite describir el tipo y las relaciones entre los datos.

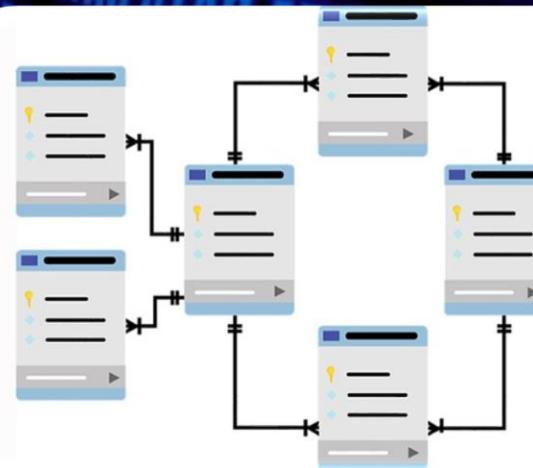
Modelo entidad-relación



GENBETA : dev

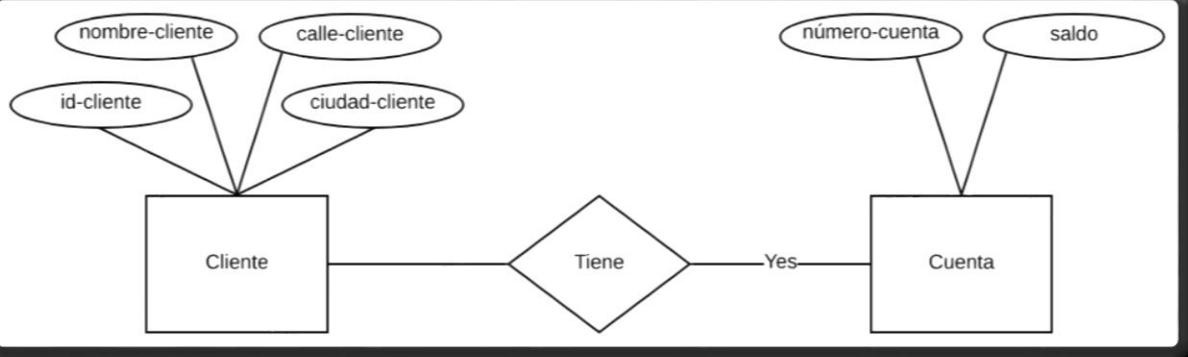
¿Qué es el modelo lógico en bases de datos relacionales?

Un modelo lógico de datos es un modelo que no es específico de una base de datos que describe aspectos relacionados con las necesidades de una organización para recopilar datos y las relaciones entre estos aspectos.



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

En él las entidades se representan utilizando rectángulos, los atributos por medio de círculos o elipses y las relaciones con rombos también líneas que conectan las entidades que tienen algún tipo de vínculo.



¿Qué es una función 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. Cuando se usa junto con una cláusula GROUP BY , los grupos resumidos suelen tener al menos una fila.



Muestre ejemplo del uso de 2 funciones de agregación

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

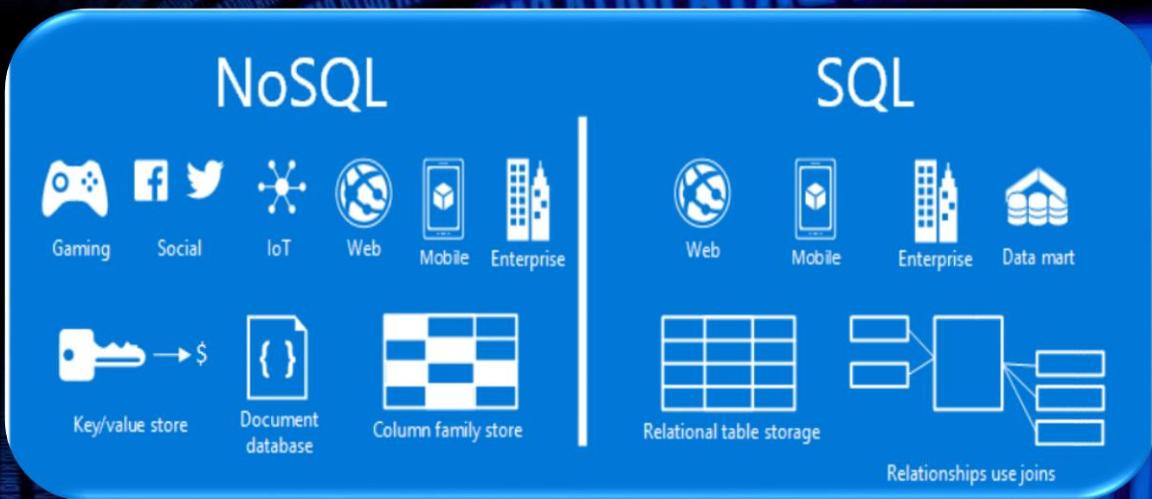
```
SELECT MAX(dp.precio_unidad)  
FROM detalle_pedido AS dp
```

Muestre un ejemplo del uso de JOINS.

```
SELECT prod.nombres  
FROM productos AS prod  
INNER JOIN detalle_pedido AS dp ON prod.id_producto=dp.id_producto  
INNER JOIN pedido AS ped ON dp.id_pedido = ped.id_pedido  
WHERE ped.id_pedido = 1
```

¿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.

ISO se refiere a documentos que especifican requerimientos que pueden ser empleados en organizaciones para garantizar que los productos o servicios ofrecidos por dichas organizaciones cumplen con su objetivo.



¿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.



Crear una función que permita sumar 3 Números.

The screenshot shows a SQL editor interface with a dark theme. The code area contains the following T-SQL script:

```
79  FROM detalle_pedido AS dp
80
81  CREATE OR ALTER FUNCTION suma_tres_numeros(@num1 INT, @num2 INT, @num3 INT)
82      RETURNS INTEGER
83          AS
84              BEGIN
85                  DECLARE @respuesta INTEGER = 0;
86
87                  SET @respuesta = @num1 + @num2 + @num3;
88
89                  RETURN @respuesta;
90              END;
91
92  SELECT dbo.suma_tres_numeros(5,6,2);
93
94
```

The code is highlighted with syntax coloring. A green checkmark icon is located next to line 92, indicating the query was executed successfully. Below the code area, there is an "Output" tab showing the result of the execution:

<anonymous>	1	13
1	13	

Manejo de consultas

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

```
SELECT prod.nombres, prod.stock  
FROM productos AS prod  
WHERE prod.stock >= 10;
```

The screenshot shows a SQL editor interface with two main sections: a code editor at the top and a results viewer at the bottom.

Code Editor (Top):

```
82 RETURNS INTEGER  
83 AS  
84 BEGIN  
85     DECLARE @respuesta INTEGER = 0;  
86  
87     SET @respuesta = @num1 + @num2 + @num3;  
88  
89     RETURN @respuesta;  
90 END;  
91  
92 SELECT dbo.suma_tres_numeros(5,6,2);  
93  
94 ✓ SELECT prod.nombres, prod.stock  
95 FROM productos AS prod  
96 WHERE prod.stock >= 10;
```

Results Viewer (Bottom):

nombres	stock
refrigerador	15

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

```
SELECT prod.nombres, cat.tipo  
FROM productos AS prod  
INNER JOIN categorias AS cat ON  
prod.id_categoria = cat.id_categoria  
WHERE cat.tipo =  
'electrodomesticos'
```

```
END;  
  
SELECT dbo.suma_tres_numeros(5,6,2);  
  
SELECT prod.nombres, prod.stock  
FROM productos AS prod  
WHERE prod.stock >= 10;  
  
SELECT prod.nombres, cat.tipo  
FROM productos AS prod  
INNER JOIN categorias AS cat ON prod.id_categoria = cat.id_categoria  
WHERE cat.tipo = 'electrodomesticos'
```

Output	Result 24
nombres	tipo
	1 refrigerador electrodomesticos
	2 microondas electrodomesticos

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

```
SELECT prod.nombres  
FROM productos AS prod  
INNER JOIN detalle_pedido AS dp  
ON prod.id_producto =  
dp.id_producto  
INNER JOIN pedido AS ped ON  
dp.id_pedido = ped.id_pedido  
WHERE ped.id_pedido = 1
```

```
SELECT prod.nombres, prod.stock  
FROM productos AS prod  
WHERE prod.stock >= 10;  
  
SELECT prod.nombres, cat.tipo  
FROM productos AS prod  
INNER JOIN categorias AS cat ON prod.id_categoria = cat.id_categoria  
WHERE cat.tipo = 'electrodomesticos'  
  
SELECT prod.nombres  
FROM productos AS prod  
INNER JOIN detalle_pedido AS dp ON prod.id_producto = dp.id_producto  
INNER JOIN pedido AS ped ON dp.id_pedido = ped.id_pedido  
WHERE ped.id_pedido = 1
```

Output	the_delicious.dbo.productos
	1 nombres 1 refrigerador
	2 microondas

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

```
SELECT count(prod.id_producto) AS  
Productos  
FROM productos AS prod  
INNER JOIN detalle_pedido AS dp  
ON prod.id_producto =  
dp.id_producto  
INNER JOIN pedido AS ped ON  
dp.id_pedido = ped.id_pedido  
WHERE ped.id_pedido = 2
```

The screenshot shows a database management interface with the following details:

- Code Area:** Displays a block of SQL code. Lines 92 through 13 are visible, including:
 - Line 92: `SELECT dbo.suma_tres_numeros(@num1: 5, @num2: 6, @num3: 2);`
 - Line 94: `SELECT prod.nombres, prod.stock`
 - Line 95: `FROM productos AS prod`
 - Line 96: `WHERE prod.stock >= 10;`
 - Line 98: `SELECT prod.nombres, cat.tipo`
 - Line 99: `FROM productos AS prod`
 - Line 100: `INNER JOIN categorias AS cat ON prod.id_categoria = cat.id_categoria`
 - Line 101: `WHERE cat.tipo = 'electrodomesticos'`
 - Line 103: `SELECT prod.nombres`
 - Line 104: `FROM productos AS prod`
 - Line 105: `INNER JOIN detalle_pedido AS dp ON prod.id_producto = dp.id_producto`
 - Line 106: `INNER JOIN pedido AS ped ON dp.id_pedido = ped.id_pedido`
 - Line 107: `WHERE ped.id_pedido = 1`
 - Line 109: `SELECT count(prod.id_producto) AS Productos` (highlighted with a green checkmark)
 - Line 110: `FROM productos AS prod`
 - Line 111: `INNER JOIN detalle_pedido AS dp ON prod.id_producto = dp.id_producto`
 - Line 112: `INNER JOIN pedido AS ped ON dp.id_pedido = ped.id_pedido`
 - Line 113: `WHERE ped.id_pedido = 2` (highlighted with a green checkmark)
- Output Area:** Shows the results of the query in a table format. The table has one row with the value "1".

Crear una función que permita sumar 3 números.

```
CREATE OR ALTER FUNCTION
suma_tres_numeros(@num1 INT,
@num2 INT, @num3 INT)
RETURNS INTEGER
AS
BEGIN
DECLARE @respuesta INTEGER = 0;

SET @respuesta = @num1 + @num2
    + @num3;

RETURN @respuesta;
END;

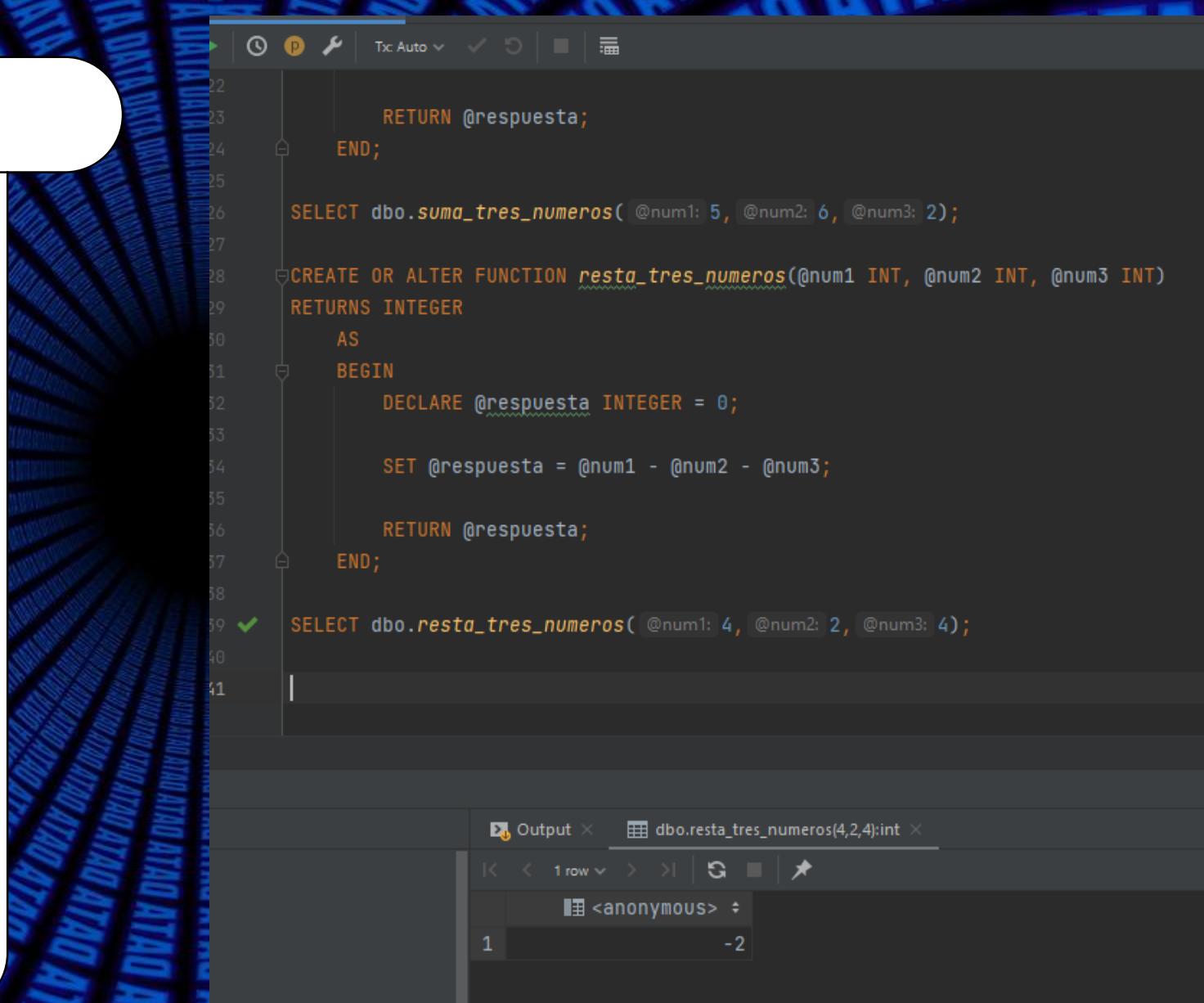
SELECT dbo.suma_tres_numeros(5,6,2);
```

```
104     FROM productos AS prod
105     INNER JOIN detalle_pedido AS dp ON prod.id_producto = dp.id_producto
106     INNER JOIN pedido AS ped ON dp.id_pedido = ped.id_pedido
107     WHERE ped.id_pedido = 1
108
109     SELECT count(prod.id_producto) AS Productos
110     FROM productos AS prod
111     INNER JOIN detalle_pedido AS dp ON prod.id_producto = dp.id_producto
112     INNER JOIN pedido AS ped ON dp.id_pedido = ped.id_pedido
113     WHERE ped.id_pedido = 2
114
115     CREATE OR ALTER FUNCTION suma_tres_numeros(@num1 INT, @num2 INT, @num3 INT)
116     RETURNS INTEGER
117     AS
118     BEGIN
119         DECLARE @respuesta INTEGER = 0;
120
121         SET @respuesta = @num1 + @num2 + @num3;
122
123         RETURN @respuesta;
124     END;
125
126     ✓ SELECT dbo.suma_tres_numeros( @num1: 4, @num2: 2, @num3: 4);
127
```

Output ×		dbo.suma_tres_numeros(4,2,4):int ×
<anonymous>	1	10

Crear una función que permita sumar 3 números.

```
CREATE OR ALTER FUNCTION  
resta_tres_numeros(@num1 INT,  
                     @num2 INT, @num3 INT)  
RETURNS INTEGER  
AS  
BEGIN  
    DECLARE @respuesta INTEGER = 0;  
  
    SET @respuesta = @num1 - @num2  
        - @num3;  
  
    RETURN @respuesta;  
END;  
  
SELECT dbo.resto_tres_numeros(12,6,4);
```



The screenshot shows a SQL code editor with the following content:

```
22  
23         RETURN @respuesta;  
24     END;  
25  
26     SELECT dbo.suma_tres_numeros( @num1: 5, @num2: 6, @num3: 2);  
27  
28     CREATE OR ALTER FUNCTION resta_tres_numeros(@num1 INT, @num2 INT, @num3 INT)  
29         RETURNS INTEGER  
30             AS  
31                 BEGIN  
32                     DECLARE @respuesta INTEGER = 0;  
33  
34                     SET @respuesta = @num1 - @num2 - @num3;  
35  
36                     RETURN @respuesta;  
37                 END;  
38  
39     ✓     SELECT dbo.resto_tres_numeros( @num1: 4, @num2: 2, @num3: 4);  
40  
41
```

The status bar at the bottom indicates the output window is active, showing the result of the last query:

<anonymous>	
1	-2

Crear una función que permita sumar 3 números.

```
CREATE OR ALTER FUNCTION suma_resta_tres_numeros  
    (@num1 INT, @num2 INT, @num3 INT)  
    RETURNS VARCHAR(50)  
    AS  
    BEGIN  
        DECLARE @resultado_suma INT = 0  
        DECLARE @resultado_resta INT = 0  
        DECLARE @respuesta VARCHAR (50) = ""  
        DECLARE @a VARCHAR (30) = ""  
        DECLARE @b VARCHAR (30) = ""  
        SET @resultado_suma = @num1 + @num2 + @num3  
            SET @a = @resultado_suma  
        SET @resultado_resta = @num1 - @num2 - @num3  
            SET @b = @resultado_resta  
        SET @respuesta = 'la suma es: '+@a+' y la resta es: '+@b  
        RETURN @respuesta  
    END;
```

```
SELECT dbo.suma_resta_tres_numeros ( 10, 5, 30);
```

The screenshot shows a SQL editor interface with the following code:

```
SELECT dbo.suma_resta_tres_numeros( @num1: 12, @num2: 6, @num3: 4);  
CREATE OR ALTER FUNCTION suma_resta_tres_numeros (@num1 INT, @num2 INT, @num3 INT)  
RETURNS VARCHAR(50)  
AS  
BEGIN  
    DECLARE @resultado_suma INT = 0  
    DECLARE @resultado_resta INT = 0  
    DECLARE @respuesta VARCHAR (50) = ""  
    DECLARE @a VARCHAR (30) = ""  
    DECLARE @b VARCHAR (30) = ""  
    SET @resultado_suma = @num1 + @num2 + @num3  
    SET @a = @resultado_suma  
    SET @resultado_resta = @num1 - @num2 - @num3  
    SET @b = @resultado_resta  
    SET @respuesta = 'la suma es: '+@a+' y la resta es: '+@b  
    RETURN @respuesta  
END;  
SELECT dbo.suma_resta_tres_numeros ( @num1: 10, @num2: 5, @num3: 30);
```

The output window shows the results of the function execution:

Output	dbo.suma_resta_tres_numeros(10, 5, 30):varchar(50)
<anonymous>	1 la suma es: 45 y la resta es: -25

¡GRACIAS POR SU ATENCIÓN!

LINK DE VIDEO:

<https://drive.google.com/file/d/1SyAXie0YrUA4KS06KZXshK-AWMRS3Iku/view?usp=sharing>

