

Funciones en SQL: Guía Teórica y Práctica

En SQL, una **función** es un bloque de código que realiza una tarea específica y retorna un valor. Hay varios tipos:

- **Funciones escalares:** devuelven un valor por cada fila.
- **Funciones de agregado:** procesan un conjunto de filas y devuelven un único valor (p.ej. SUM, COUNT, AVG).
- **Funciones de ventana:** similares a las de agregado, pero no condensan los resultados en una sola fila (p.ej. ROW_NUMBER).
- **Funciones definidas por el usuario (UDF):** creadas por ti para resolver necesidades específicas.

1. Funciones de Agregado

Procesan múltiples filas y devuelven un solo valor. A menudo se usan con GROUP BY y HAVING.

```
-- Contar filas
SELECT COUNT(*) AS total_filas
FROM tabla;

-- Sumar valores
SELECT SUM(precio) AS total
FROM productos;

-- Calcular promedio
SELECT AVG(edad) AS promedio
FROM personas;

-- Encontrar mínimo
SELECT MIN(salario) AS minimo
FROM empleados;

-- Encontrar máximo
SELECT MAX(salario) AS maximo
FROM empleados;
```

Ejemplo adicional:

```
SELECT departamento,
       COUNT(*) AS total_empleados,
       AVG(salario) AS prom_salarial
FROM empleados
GROUP BY departamento
HAVING AVG(salario) > 3000;
```

2. Funciones Numéricas

```
ABS(x)      -- Valor absoluto
CEILING(x)   -- Redondea hacia arriba
FLOOR(x)     -- Redondea hacia abajo
ROUND(x,d)  -- Redondea con d decimales
POWER(x,y)  -- x elevado a y
SQRT(x)     -- Raíz cuadrada de x

-- Ejemplos:
SELECT ABS(-10);      -- 10
SELECT CEILING(4.2);  -- 5
SELECT FLOOR(4.9);    -- 4
SELECT ROUND(3.1415, 2); -- 3.14
SELECT POWER(2, 3);   -- 8
SELECT SQRT(16);      -- 4
```

3. Funciones de Cadena

```
LOWER('TEXT0') / LCASE('TEXT0') -- A minúsculas
UPPER('texto') / UCASE('texto')  -- A mayúsculas
SUBSTRING('Hola Mundo', 1, 4)    -- 'Hola'
LEFT('SQL', 2) / RIGHT('SQL', 2) -- 'Sq' / 'QL'
LTRIM('  Hola') / RTRIM('Hola  ')
REPLACE('Hola Mundo','Mundo','SQL')
CONCAT('Hola',' ', 'Mundo')      -- 'Hola Mundo'
```

4. Funciones de Fecha y Hora

```
NOW() / GETDATE() / CURRENT_TIMESTAMP
-- Retorna fecha y hora actual

CURDATE() / CURRENT_DATE
-- Retorna la fecha actual (sin hora)

DATEADD(...) / DATE_ADD(...)
-- Suma un intervalo de tiempo a una fecha

DATEDIFF(fecha1, fecha2)
-- Diferencia en días entre dos fechas

EXTRACT(...) / DATEPART(...)
-- Extrae parte específica (año, mes, día, etc.)
```

5. Funciones Lógicas y de Control

```
CASE
  WHEN condicion1 THEN ...
  WHEN condicion2 THEN ...
  ELSE ...
END

COALESCE(v1, v2, ..., vN)
-- Devuelve el primer valor NO NULL

NULLIF(exp1, exp2)
-- Devuelve NULL si exp1 = exp2, de lo contrario exp1
```

6. Funciones de Ventana

No agrupan todo en una fila única, sino que aplican cálculos por “ventanas” o particiones. Ejemplo:

```
SELECT empleado_id,
       departamento,
       salario,
       ROW_NUMBER() OVER (PARTITION BY departamento
                          ORDER BY salario DESC) AS posicion
FROM empleados;
```

7. Funciones Definidas por el Usuario (UDF)

Ejemplo en SQL Server:

```
CREATE FUNCTION fn_CalcularIVA
(
    @precio DECIMAL(10,2),
    @tasa   DECIMAL(5,2)
)
RETURNS DECIMAL(10,2)
AS
BEGIN
    DECLARE @iva DECIMAL(10,2);
    SET @iva = @precio * @tasa;
    RETURN @iva;
END;

SELECT dbo.fn_CalcularIVA(100, 0.21); -- 21.00
```

Ejemplo en MySQL:

```
DELIMITER $$
CREATE FUNCTION fn_calcularIVA(precio DECIMAL(10,2), tasa DECIMAL(5,2))
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    DECLARE valorIVA DECIMAL(10,2);
    SET valorIVA = precio * tasa;
    RETURN valorIVA;
END $$
DELIMITER ;

SELECT fn_calcularIVA(100, 0.21); -- 21.00
```

8. Ejemplos Prácticos

Creación de tabla y datos (MySQL):

```
CREATE DATABASE mi_basedatos;
USE mi_basedatos;

CREATE TABLE empleados (
    empleado_id INT PRIMARY KEY,
    nombre VARCHAR(50),
    departamento VARCHAR(50),
    salario DECIMAL(10,2),
    fecha_ingreso DATE
);

INSERT INTO empleados (empleado_id, nombre, departamento, salario, fecha_ingreso)
VALUES
(1, 'Ana', 'Contabilidad', 2500.00, '2022-01-15'),
(2, 'Juan', 'Ventas', 1800.00, '2021-11-10'),
(3, 'Pedro', 'Contabilidad', 3200.00, '2019-06-01'),
(4, 'Maria', 'Ventas', 5000.00, '2020-02-20'),
(5, 'Lucia', 'IT', 4000.00, '2021-08-30');
```

Consultas básicas:

```
-- Funciones de agregado
SELECT COUNT(*) AS total_emp,
       AVG(salario) AS promedio,
       MIN(salario) AS minimo,
       MAX(salario) AS maximo
FROM empleados;

-- Funciones de cadena
SELECT UPPER(nombre) AS mayus,
       LOWER(departamento) AS minus
FROM empleados;

-- Funciones de fecha (MySQL)
SELECT nombre,
       DATEDIFF(CURRENT_DATE, fecha_ingreso) AS dias
FROM empleados;

-- CASE y ROUND
SELECT nombre,
       salario,
       ROUND(salario * 1.10, 2) AS aumento_10,
       CASE
         WHEN salario < 2000 THEN 'Bajo'
         WHEN salario BETWEEN 2000 AND 4000 THEN 'Medio'
         ELSE 'Alto'
       END AS categoria
FROM empleados;

-- Funciones de ventana (si tu SGBD las soporta)
SELECT empleado_id,
       departamento,
       salario,
       ROW_NUMBER() OVER (PARTITION BY departamento ORDER BY salario DESC) AS ranking
FROM empleados;
```

Consejos Generales

- **Compatibilidad:** la sintaxis puede variar según el SGBD (MySQL, SQL Server, Oracle, PostgreSQL...).
- **Manual oficial:** revisa siempre la documentación exacta.
- **Rendimiento:** usar funciones sobre columnas en WHERE puede desactivar índices y ralentizar consultas.
- **Organización:** separa la creación de tablas, inserción de datos y consultas en tu proyecto.
- **Ejemplos reales:** para trabajos académicos, es útil tener datos cercanos a la realidad.