



Universidad César Vallejo

Pregrado

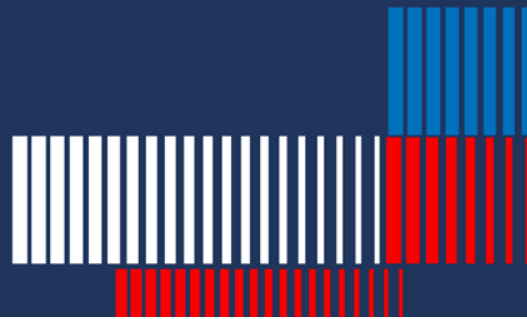
Gestión de Datos e Información II

30
años

Licenciada por
SUNEDU
para que puedas
salir adelante

SESIÓN 03

FUNCIONES EN SQL





Una **función de agregado SQL** acepta un grupo de datos (normalmente una columna de datos) como argumento, y produce un único dato que resume el grupo. **Por ejemplo** la función AVG() acepta una columna de datos numéricos y devuelve la media aritmética (average) de los valores contenidos en la columna.

SINTAXIS: Función ([ALL|DISTINCT] expression)

- ✓ La palabra **ALL** indica que se tiene que tomar en cuenta todos los valores de la columna. Es el valor por defecto.
- ✓ La palabra **DISTINCT** hace que se consideren todas las repeticiones del mismo valor como uno sólo (considera valores distintos).
- ✓ Una función de agregado puede aparecer en la lista de selección en cualquier lugar en el que puede aparecer un nombre de columna.
- ✓ Puede, **por ejemplo**, formar parte de una expresión pero no se pueden anidar funciones de agregado.



COUNT ({[ALL|DISTINCT] expresion | * })

- ✓ **Devuelve el número total de filas seleccionadas por la consulta.**
- ✓ La función cuenta los valores distintos de NULL que hay en la columna.
- ✓ Expresión puede ser de cualquier tipo excepto text, image o ntext.
- ✓ La palabra **ALL** indica que se tienen que tomar todos los valores de la columna, mientras que **DISTINCT** hace que se consideren todas las repeticiones del mismo valor como uno solo. Estos parámetros son opcionales, por defecto se considera ALL.

SELECT COUNT(region) **FROM** oficinas;

SELECT COUNT(DISTINCT region) **FROM** oficinas;

SELECT COUNT(*) **FROM** empleados **WHERE** oficina **IS NOT NULL**;

Si utilizamos * en vez de expresión, devuelve el número de filas del origen que nos quedan después de ejecutar la cláusula **WHERE**.

COUNT(*) no acepta parámetros y no se puede utilizar con **DISTINCT**. **COUNT(*)** no requiere un parámetro expresión porque, por definición, no utiliza información sobre ninguna columna específica. En el recuento se incluyen las filas que contienen valores **NULL**.



SUM ([ALL|DISTINCT] expression)

- ✓ Devuelve la suma de los valores devueltos por la expresión.
- ✓ Sólo puede utilizarse con columnas numéricas.
- ✓ El resultado será del mismo tipo aunque puede tener una precisión mayor.

```
SELECT SUM(importe) FROM pedidos;
```



MAX ([ALL|DISTINCT] expression)

- ✓ Devuelve el valor máximo de la expresión sin considerar los nulos.
- ✓ **MAX** se puede usar con columnas numéricas, de caracteres y de datetime, pero no con columnas de bit. No se permiten funciones de agregado ni subconsultas.
- ✓ Utilizar DISTINCT no tiene ningún sentido con **MAX** (el valor máximo será el mismo si consideramos las repeticiones o no) y sólo se incluye para la compatibilidad con SQL-92.

SELECT

SUM(ventas) **AS** VentasTotales,

MAX(objetivo) **AS** MayorObjetivo

FROM oficinas;



MIN ([ALL|DISTINCT] expression)

- ✓ Devuelve el valor máximo de la expresión sin considerar los nulos.
- ✓ **MIN** se puede usar con columnas numéricas, de caracteres y de datetime, pero no con columnas de bit. No se permiten funciones de agregado ni subconsultas.
- ✓ Utilizar DISTINCT no tiene ningún sentido con **MIN** (el valor mínimo será el mismo si consideramos las repeticiones o no) y sólo se incluye para la compatibilidad con SQL-92.

SELECT

SUM(ventas) **AS** VentasTotales,

MIN(objetivo) **AS** MayorObjetivo

FROM oficinas;



AVG ([ALL|DISTINCT] expression)

- ✓ Devuelve el **promedio** de los valores de un grupo, para calcular el promedio se omiten los valores nulos.
- ✓ El grupo de valores lo determina el resultado de la expresión que será un nombre de columna o una expresión basada en una columna o varias del origen de datos.
- ✓ La función se aplica también a campos numéricos, y en este caso el tipo de dato del resultado puede cambiar según las necesidades del sistema para representar el valor del resultado.

SELECT

AVG(cuota) **AS** [Cuota media],

AVG(ventas) **AS** [Ventas medias]

FROM empleados;



- ✓ **VAR.-** Devuelve la varianza estadística de todos los valores de la expresión especificada.
- ✓ **VARP.-** Devuelve la varianza estadística de la población para todos los valores de la expresión especificada.
- ✓ **STDEV.-** Devuelve la desviación típica estadística de todos los valores de la expresión especificada.
- ✓ **STDEVP.-** Devuelve la desviación estadística estándar para la población de todos los valores de la expresión especificada
- ✓ **COUNT_BIG.-** Funciona igual que la función COUNT. La única diferencia entre ambas funciones está en los valores devueltos, COUNT_BIG siempre devuelve un valor de tipo **bigint** y por lo tanto admite más valores de entrada, no está limitado.



@@ERROR

Devuelve el número de error de la última instrucción Transact-SQL ejecutada. Devuelve 0 si la instrucción Transact-SQL anterior no encontró errores. Devuelve un número de error si la instrucción anterior encontró un error.

@@ROWCOUNT

Devuelve el número de filas afectadas por la última instrucción. Si el número de filas superior a dos mil millones, use ROWCOUNT_BIG.

@@IDENTITY

Se trata de una función del sistema que devuelve el último valor de identidad insertado.



@@TRANCOUNT

Devuelve el número de instrucciones **BEGIN TRANSACTION** que se han producido en la conexión actual.

CHECKSUM

La función **CHECKSUM** devuelve el valor de la suma de comprobación calculado sobre una fila de una tabla o sobre una lista de expresiones. Use CHECKSUM para generar índices hash.

ERROR_MESSAGE

Esta función devuelve el texto del mensaje del error que ha provocado la ejecución del bloque CATCH de una construcción TRY...CATCH.



ERROR_NUMBER

Esta función devuelve el número del error que ha provocado la ejecución del bloque CATCH de una construcción TRY...CATCH.

ISNUMERIC

Determina si una expresión es un tipo numérico válido. devuelve 1 cuando la expresión de entrada se evalúa para un tipo de datos numérico válido; de lo contrario, devuelve 0.

ISNULL

Sustituye el valor NULL por el valor especificado.

ISNULL (check expression , replacement_value) .

check_expression.- Es la expresión que se va a comprobar si es NULL. check_expression puede ser de cualquier tipo.

replacement_value.- Es la expresión que se devuelve si check_expression es NULL. valor_de_reemplazo debe ser de un tipo que se pueda convertir implícitamente en el tipo de expresión_de_comprobación.



HOST_NAME

Devuelve el nombre de la estación de trabajo.

ERROR_STATE

Devuelve el número de estado del error que provocó que se ejecutara el bloque CATCH de una construcción TRY...CATCH.

ERROR_PROCEDURE

Esta función devuelve el nombre del procedimiento almacenado o del desencadenador en el que se produce un error, si ese error ha causado la ejecución de un bloque CATCH de una construcción TRY...CATCH.



FUNCIONES ESCALARES

Una función es una rutina almacenada que recibe unos parámetros escalares de entrada, los procesa según la definición de la función y finalmente retorna un resultado de un tipo específico que permitirá su utilización con un objetivo.

Create Function NombreFunción (@Parámetro1 tipo, @Parámetro 2 tipo...)

Returns Tipo **As**

Begin

Sentencias

End



FUNCIONES DE TABLA

Permiten retornar tablas en los resultados. Esta característica proporciona al programador facilidad a la hora de administrar sus bases de datos.

Create Function NombreFunción(Parámetro1 tipo, Parámetro2 tipo...)

Returns Table As

Return(Sentencias)

/

Sentencias

Return



FUNCION: convierte mayúsculas los campos de una tabla

```
CREATE FUNCTION EnMayusculas
(
    @Nombre Varchar(50),
    @Apellido Varchar(50)
)
RETURNS Varchar(100) AS
--Declarar Variables
BEGIN
    RETURN (UPPER(@Apellido) + ', ' + UPPER(@Nombre))
END
--Ejecutar
Print dbo.EnMayusculas('Geynen','Montenegro')
go
--llamada a la función
Select dbo.EnMayusculas(Nombre,Apellidos) As Nombre
From Cliente
```



FUNCIÓN: ingresar el número del día de la semana y devuelve el nombre del día

```
Create function NombreDia(@Dia int) Returns Varchar(10) AS
Begin
  Declare @Var Varchar(10)
  Select @Var= Case @Dia
    When 1 Then 'Lunes'
    When 2 Then 'Martes'
    When 3 Then 'Miercoles'
    When 4 Then 'Jueves'
    When 5 Then 'Viernes'
    When 6 Then 'Sabado'
    When 7 Then 'Domingo'
  End
  Return @Var
End
--Ejecutar
Print dbo.NombreDia(5)
```




FUNCION: ingresa el id del empleado y devuelve una tabla con los campos id, nombre ,apellidos y el número de pedidos realizados

```
Create Function Tabla(@IdEmpleado int)
Returns Table
AS
    Return (
        Select
            P.IdEmpleado, Nombre,
            Apellidos, Count(P.IdEmpleado) as CantidadPedidos
        From Empleados E inner join Pedidos P On P.IdEmpleado=E.IdEmpleado
        Where E.IdEmpleado=@IdEmpleado
        Group By P.IdEmpleado, Nombre ,Apellidos
    )

--Ejecutar
Select * From dbo.Tabla(1)
```



¿QUÉ HEMOS APRENDIDO HOY?



Para que reflexionen y entiendan la importancia de los temas tratados y el mejoramiento de su propio proceso de aprendizaje.



Universidad **César Vallejo**

Licenciada por Sunedu
para que puedas salir adelante