Argentina programa 4.0

Ministerio de Economía
Argentina

Secretaría de
Economía del Conocimiento

primero la gente

UNTREF
UNIVERSIDAD NACIONAL
DE TRES DE FEBRERO

Clase 21: Bases de datos Relacionales

Funciones escalares

Ministerio de
Desarrollo Productivo
Argentina

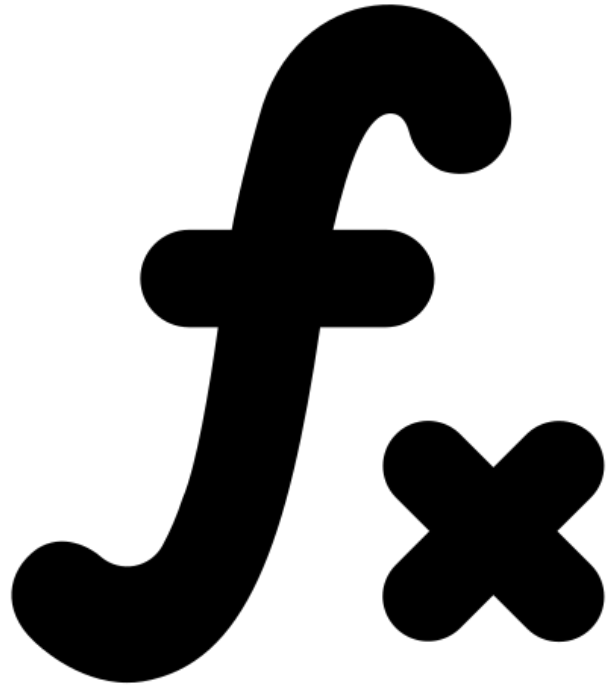
Agenda de hoy

A. Las funciones escalares
   a. Qué es una función
B. Tipos de funciones escalares
   a. Funciones string
   b. Funciones Date
   c. Funciones Number
C. Prácticas intercaladas con la implementación de diferentes funciones escalares

Qué es una función

De acuerdo a lo que aprendimos dentro del mundo de la programación JavaScript, una función es una pieza de código que realiza una operación determinada, por supuesto para lo que fue ideada.

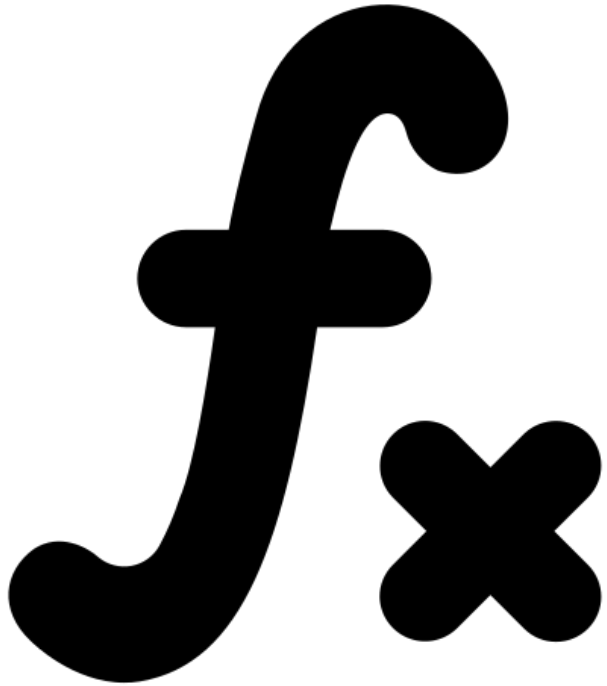
En algunos casos, la función espera uno o más parámetros a procesar, y en otros casos, simplemente se las ejecuta para que devuelvan un dato específico.



Qué es una función

Pensando como un lenguaje de programación, la función puede o no recibir parámetros de entrada, y siempre retornará un valor esperado por el usuario.

Sus nombres son definidos, acorde a lo que deben hacer, y se las utiliza para trabajar con los diferentes tipos de datos que pueden almacenarse en los registros de una tabla.

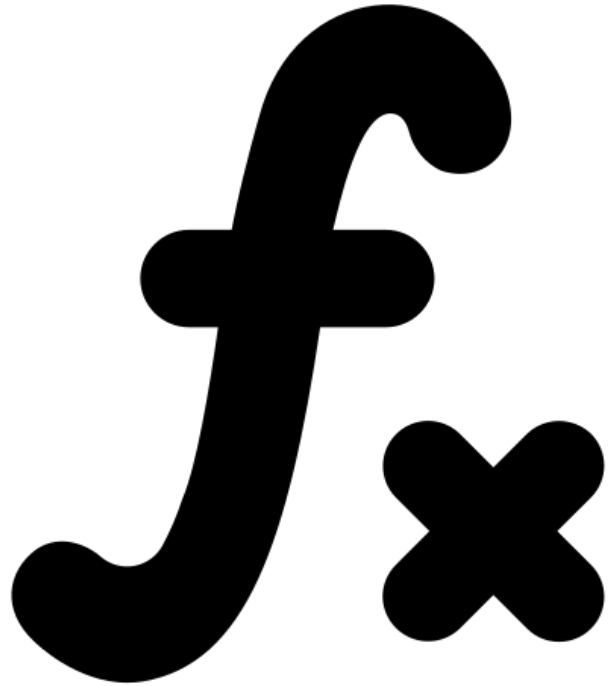


Las funciones escalares

Las funciones escalares

Al igual que los lenguajes de programación en general, SQL incluye una serie de funciones denominadas **Funciones Escalares**.

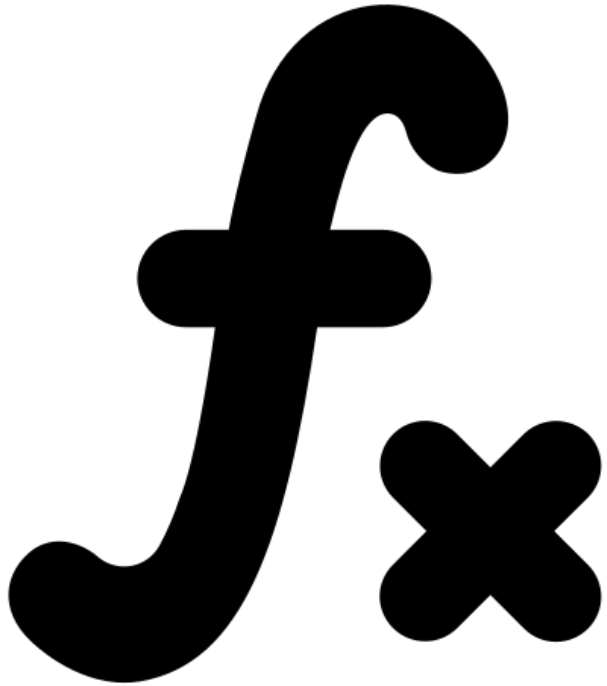
Permiten manipular datos cuando los recuperamos o antes de guardarlos, mediante operaciones predeterminadas, devolviendo un resultado específico, acorde a lo esperado.



Las funciones escalares

Algunas de las ventajas de implementar funciones escalares en nuestras operaciones con SQL, son:

- Reducir el re-trabajo de la lógica comercial
- Evitar la inconsistencia de datos que provenga de un software
- Ayudar a reducir el tráfico de red de aplicaciones cliente/servidor
- Mejorar en gran medida el rendimiento de los sistemas



Las funciones escalares

Y cuando hablamos de funciones escalares en SQL, encontramos que existen dos tipos posibles:

- funciones integradas
- funciones almacenadas

En esta oportunidad trabajaremos con las funciones integradas en el lenguaje de base de datos.

Las funciones escalares

Las funciones almacenadas son habitualmente funciones escalares pero construidas por nosotras mismas.

En determinadas situaciones podemos encontrarnos que sería ideal contar con una función que realice una tarea específica, pero esta no existe.

Lo bueno de esto es que sabiendo programar, podremos construirla nosotras mismas para solventar dicha necesidad dentro de nuestros desarrollos.

Tipos de funciones escalares

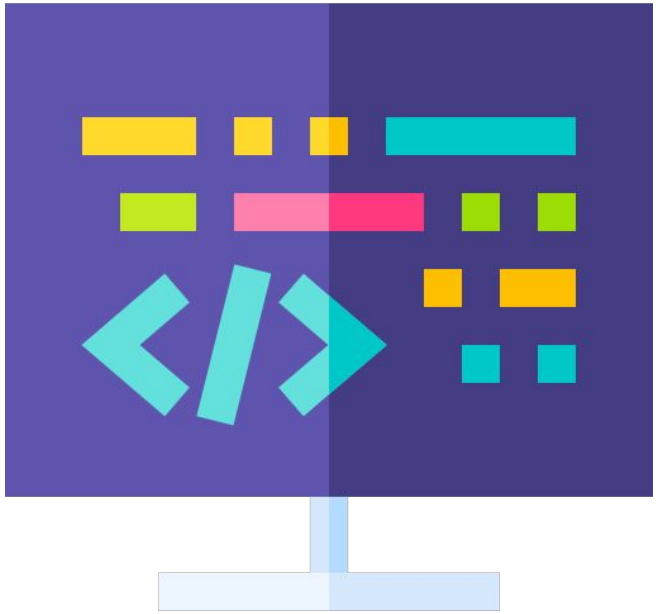
Tipos de funciones escalares

Las funciones escalares, integradas en SQL, se clasifican a su vez bajo las siguientes categorías:

- funciones de cadenas
- funciones numéricas
- funciones de fecha
- funciones de agregación

Tipos de funciones escalares: de cadenas

Las funciones escalares de cadena en SQL son funciones que operan en valores de tipo cadena (*por ejemplo, caracteres alfanuméricos o de texto*) y devuelven, o retornan, un resultado basado en esos valores.



Tipos de funciones escalares: fechas

Las funciones escalares numéricas en SQL son funciones que operan en valores de tipo numéricos.

Usualmente nos acercan una serie de mecanismos para realizar operaciones matemáticas y de cálculo en general, retornando valores de acuerdo a los parámetros indicados.

Tipos de funciones escalares: fechas

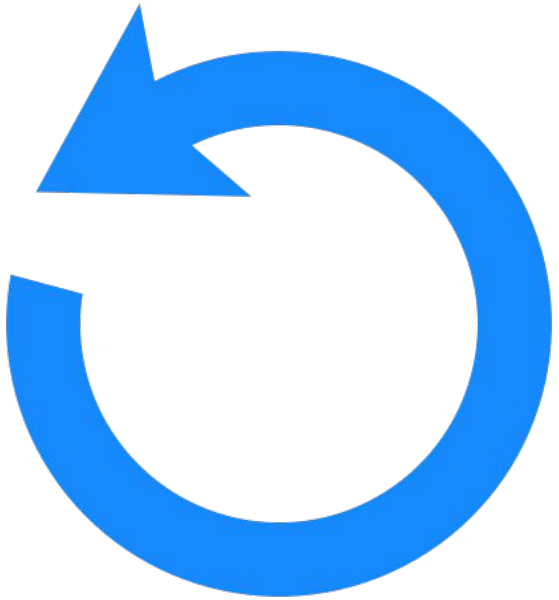
Las funciones escalares de fechas en SQL son funciones que operan en valores de tipo fecha y hora.

Nos acercan una serie de mecanismos para trabajar con operaciones entre fechas y/u horas retornando valores como resultado, de acuerdo a los parámetros indicados.

Tipos de funciones escalares

En todos los casos, las funciones retornan algún valor resultante generalmente identificado como operaciones de transformación o cálculo.

Como podemos apreciar, es el mismo principio que vimos oportunamente cuando aprendimos a utilizar y crear funciones en un lenguaje de programación como lo es JavaScript.



funciones escalares de cadenas (string)

Funciones escalares de cadenas (string)

Permiten operar con cualquier tipo de cadena de caracteres almacenada en una tabla (*o por almacenarse*). Podemos, entre otras cosas:

- convertir el texto a mayúsculas / minúsculas
- concatenar strings
- cortar una porción del texto
- eliminar espacios
- revertir el texto
- contar caracteres

Entre decenas de funciones más.

Funciones escalares de cadenas (string)

Fusiona cadenas de caracteres en un único bloque de datos.

Podemos, por ejemplo, unificar en un campo llamado **nombre_completo**, los campos **nombre** y **apellido** de una tabla.

Función concat()

```
SELECT nombreCompleto,
       concat(telefono, ' - ', tipodetelefono) AS Tel
FROM Ecommerce.contacto;
```

Funciones escalares de cadenas (string)

Convierte un bloque de texto a mayúsculas.
Es ideal para cuando queremos normalizar la visualización de determinados datos, o incluso cuando queremos guardar datos normalizados.

Funciones escalares

```
SELECT UCASE(Title) FROM Employees;
```

Funciones escalares de cadenas (string)

Convierte un bloque de texto a minúsculas.

También cuenta con su nombre de función alternativo: **LOWER()**. Este último funciona de la misma forma que **LCASE()**.

Funciones escalares

```
SELECT LCASE(productName) FROM Products;
```

Funciones escalares de cadenas (string)

Invierte el orden de contenido de un campo específico.

Funciona tanto con textos como también en campos numéricos.

Funciones escalares

```
SELECT REVERSE(productName) AS nombreInvertido
FROM Products;
```

Funciones escalares de cadenas (string)

Elimina espacios innecesarios al inicio o final de un texto.

Es ideal para normalizar contenido que pueda venir con espacios por demás, por ejemplo, en campos de comentarios que deben tener una cantidad de caracteres mínima, y los usuario suelen completarlos con espacios.

Función trim()

```
SELECT TRIM("     Este es un texto más o menos extenso.    ") AS TextoCorrecto;
```

Funciones escalares de cadenas (string)

Existen también **LTRIM()** y **RTRIM()** para eliminar espacios en un texto, del lado izquierdo y derecho, respectivamente.

Funciones escalares

```
SELECT LTRIM('    Texto con espacio a la izquierda')
AS TextoConEspacioIzquierdo;

SELECT RTRIM('Texto con espacio a la derecha      ')
AS TextoConEspacioDerecho;
```

Funciones escalares de cadenas (string)

Retorna una cantidad de espacios determinados, en base al número que recibe como parámetro.

Ideal por si debemos completar con espacios obligatorios un campo específico.

Función space()

```
SELECT SPACE(21)
    AS EspaciosEnTexto;

-- retorna '                     ' espacios
```

Funciones escalares de cadenas (string)

Permite reemplazar una porción de texto por otra, contenida en un bloque de texto determinado. Es ideal para cuando debemos reemplazar posibles caracteres extraños dentro de un bloque de texto, o normalizar quitando acentos, tildes, ñuflos, diéresis, o si debemos eliminar caracteres especiales como ser guiones o espacios en, por ejemplo, un número de teléfono.

Función replace()

```
SELECT REPLACE("SQL SERVER es el mejor", "SQL SERVER", "MySQL") AS ElPreferido;
```

Funciones escalares de cadenas (string)

Permite contar la cantidad de caracteres en un bloque de texto determinado.

Si debemos limitar la cantidad de caracteres a insertar en un determinado campo, podremos validar que esto se cumpla y no sobrepase los caracteres máximos esperados.

Función char_length()

```
SELECT CHAR_LENGTH("Curso de MySQL")
    AS Curso;
```

Funciones escalares de cadenas (string)

Devuelve la posición de un texto o caracter determinado.
Es ideal para identificar si dentro de un bloque de texto hay o no alguna palabra esperada, y en qué posición se encuentra dicha palabra.

- Si no encuentra el término, devolverá 0 (*cero*).
- Si lo encuentra una o más veces, devuelve solo la primera posición encontrada para dicho bloque de texto.

Funciones escalares

```
SELECT INSTR('Programación para servidores (backend)', 'backend')
AS posicion;
```

Funciones escalares de cadenas (string)

Extrae una porción determinada de caracteres de un bloque de texto. Debemos especificar la posición inicial y el total de caracteres a extraer.

Es utilizada de forma frecuente para la limpieza de contenido en textos (*data analytics*), como también para reducir (*o truncar*) a un limitado número de caracteres un bloque de texto, previo a ser almacenado en un campo específico.

Funciones escalares

```
SELECT MID('Programación para servidores (backend)', 30, 7)
AS Palabras;

-- retornará 'backend'
```

Alias en funciones escalares

Alias en funciones escalares

Al ejecutar cualquier función escalar sobre un campo de tabla SQL, veremos que el nombre de la columna en el resultado de la consulta mostrará la función escalar encerrando el nombre del campo.

Para subsanar esto, debemos utilizar la cláusula Alias, la cual nos permite definir otro nombre para el campo. Podremos elegir el mismo o directamente otro diferente, y así conseguir una mejor visualización del resultado de la consulta.

Funciones escalares

```sql
SELECT MID('Programación para servidores (backend)', 30, 7)
AS Palabras;

-- retornará 'backend'
```

Alias en funciones escalares

Incluso el uso de Alias puede aplicarse en campos convencionales, para poder cambiar el nombre de ellos si no es muy claro.

Por ejemplo, podemos ejecutar una consulta de selección sobre una tabla SQL, y redefinir el nombre de sus campos con un alias, para pasar los mismos de inglés a español, tal como muestra el ejemplo contiguo.

Alias

```
SELECT ProductID AS Codigo,
       UCASE(ProductName) AS Descripcion,
       QuantityPerUnit AS Presentacion,
       UnitPrice AS PrecioUnitario
FROM products;
```

Sección práctica

Dado que son muchas las funciones escalares que debemos repasar, haremos una pausa aquí para implementar las funciones escalares del tipo String en algunas consultas de selección.

Veamos a continuación la consigna:

Prácticas

1. Ejecuta una consulta de selección sobre los siguientes campos de la tabla **Products**
   - **productID, productName, QuantityPerUnit, UnitPrice**
   - aplica un alias a cada uno de ellos **(Codigo, Descripcion, Presentacion, PrecioUnitario)**

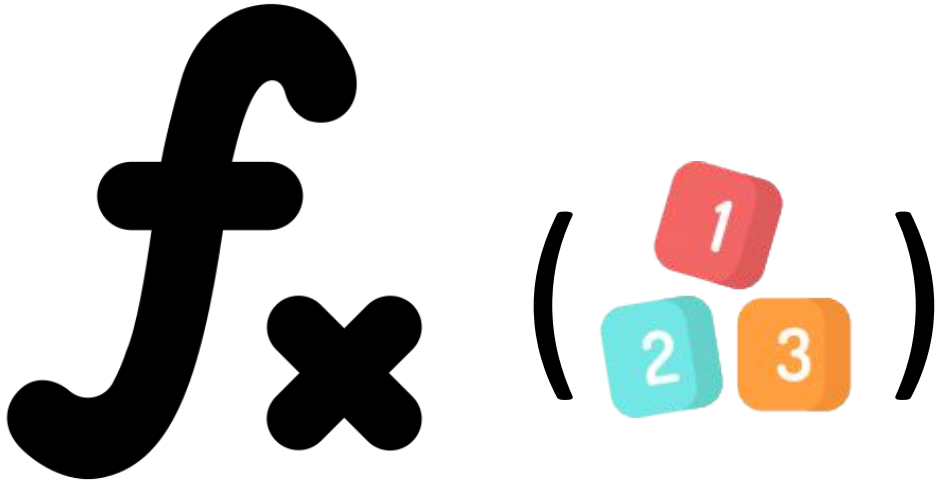
2. Ejecuta una consulta de selección sobre los siguientes campos de la tabla **Products**
   - **ProductID, ProductName, QuantityPerUnit, UnitPrice**
   - Aplica sobre el campo **ProductName** la función escalar que transforma el texto a mayúsculas

3. Ejecuta una consulta de selección sobre los siguientes campos de la tabla **Products**
   - **ProductID, ProductName, QuantityPerUnit, UnitPrice**
   - aplica el mismo alias detallado en el punto uno (1)
   - Aplica sobre el campo **QuantityPerUnit** la función escalar de reemplazo de texto, buscando el texto *'boxes'* y reemplazando el mismo por *'cajas'*
   - la condición WHERE debe filtrar aquellos registros que posean en cualquier parte la palabra *'boxes'* en cualquier parte del campo **QuantityPerUnit**

funciones escalares numéricas

Funciones escalares numéricas

Permiten operar con cualquier tipo de datos numéricos, realizando operaciones matemáticas y aritméticas entre otras, y hasta transformando la forma de mostrar los números.



Funciones escalares numéricas

La función **abs()** recibe un número como parámetro y **retorna el absoluto**.

Función ABS()

```
SELECT ABS(-243.5) AS NroAbsoluto;

-- devuelve el nro. en positivo
```

Funciones escalares numéricas

La función **ceil()** recibe un número como parámetro y **retorna el número entero más próximo**.

Función CEIL()

```
SELECT CEIL(21.375) AS NroEnteroProx;

-- devuelve el nro. entero más próximo
```

Funciones escalares numéricas

La función **floor()** es la función inversa de ceil(). Recibe un número como parámetro y **retorna el número entero anterior, más próximo**.

Función FLOOR()

```
SELECT FLOOR(21.375) AS NroEnteroAnt;

-- devuelve el nro. entero anterior más próximo
```

Funciones escalares numéricas

La función **greatest()** recibe un set de números como parámetro y **retorna el número mayor de dicha lista**.

Función GREATEST()

```
SELECT GREATEST(72, 1, 75, 3, 21, 96, 5, 30) AS Mayor;

-- devuelve el nro. mayor de todo el grupo
```

Funciones escalares numéricas

La función **least()**, es la inversa de greatest(), dado que recibe un set de números como parámetro y **retorna el número menor de dicha lista**.

Función LEAST()

```
SELECT LEAST(72, 1, 75, 3, 21, 96, 5, 30) AS Menor;

-- devuelve el nro. menor de todo el grupo
```

Funciones escalares numéricas

La función **mod()** recibe una división entre dos números, y retorna el módulo, o resto, de dicha operación aritmética.

Función MOD()

```
SELECT MOD(21, 4) AS Modulo;

-- devuelve el resto no divisible (1)
```

Funciones escalares numéricas

La función **mod()** recibe una división entre dos números, y retorna el módulo, o resto, de dicha operación aritmética.

Función MOD()

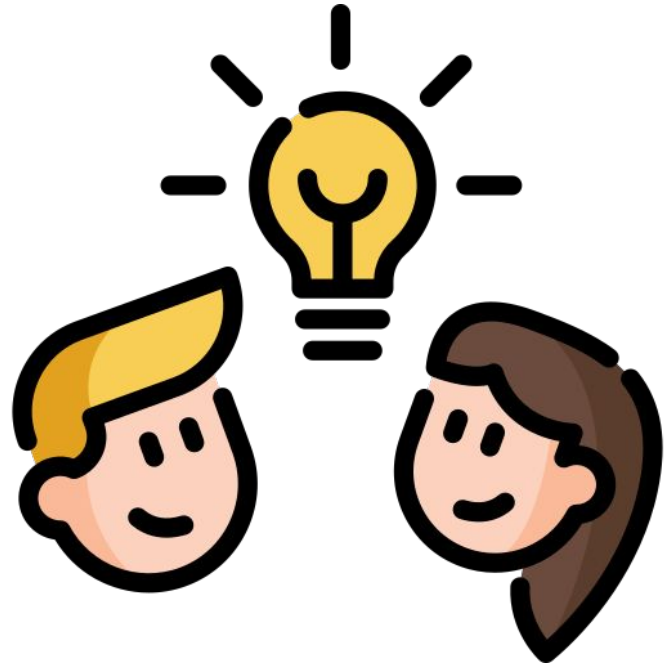
```
SELECT MOD(21, 4) AS Modulo;

-- devuelve el resto no divisible (1)
```

Funciones escalares numéricas

Como bien mencionamos en diferentes instancias, es imposible conocer y tener presente a todas las funciones escalares existentes.

Como siempre, los sistemas de ayuda oficial son los que nos guiarán en conocer la función apropiada y cómo se debe aplicar.

En su mayoría, las funciones escalares trabajan de igual forma, recibiendo uno o más parámetros y siempre retornando una transformación o cálculo como resultado.

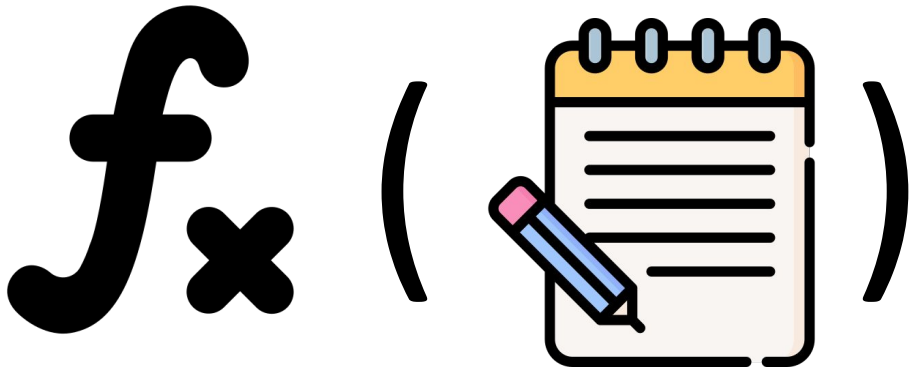


funciones escalares para fechas

Funciones escalares con fechas

Las funciones de fecha y hora, nos permiten trabajar con estas unidades de tiempo, de acuerdo a la necesidad.

En algunos casos, existe más de una función SQL Date, para realizar la misma tarea, tal como vimos oportunamente con las funciones escalares para el manejo de cadenas (string).



Funciones escalares con fechas

NOW() fecha y hora actual

CURDATE() ver el día actual

DAY() retorna el día para una fecha

MONTH() retorna el mes para una fecha

YEAR() retorna el año para una fecha

Funciones Date con fechas

```
SELECT NOW();

SELECT CURDATE();

SELECT DAY();

SELECT MONTH();

SELECT YEAR();
```

Funciones escalares con fechas

DATEDIFF() período en días que pasaron entre dos fechas

DAYOFYEAR() cuántos días transcurrieron en la fecha indicada

WEEKOFYEAR() cuántas semanas transcurrieron en la fecha indicada

Funciones Date con fechas parte 2

```
SELECT DATEDIFF(FECHA1, FECHA2);

SELECT DAYOFYEAR(FECHA);

SELECT WEEKOFYEAR(FECHA);
```

Funciones escalares con fechas

Devuelve el período trimestral de la fecha indicada

Enero a Marzo retorna 1

Abril a Junio retorna 2

Julio a Septiembre retorna 3

Octubre a Diciembre retorna 4

Período trimestral del año

```
SELECT QUARTER("2022-08-03");
```

Esto representa al término **QUARTER** o simplemente **'Q'** mencionado habitualmente en empresas, por las áreas administrativas y gerenciales.

Funciones escalares con fechas

Retorna el nombre del mes definido en una fecha (*en inglés*)

Nombre del mes

```
SELECT MONTHNAME("2022-03-21");
```

Funciones escalares con fechas

Retorna la diferencia entre un período:

- el período se define como año (*4 dígitos*), y el mes (*2 dígitos*)
- la fecha mayor va en primer lugar, la menor en el segundo
- devuelve el resultado en el número de meses transcurridos
- si el número de meses transcurridos ya pasó, lo muestra en positivo, caso contrario en negativo

Tiempo transcurrido en un período

```
SELECT PERIOD_DIFF(202203, 202109);
```

Funciones escalares con fechas

Permite agregar un intervalo de tiempo a una fecha específica.

El intervalo lo podemos definir en:

- YEAR
- MONTH
- DAY

Retorna la fecha resultante.

Intervalo de tiempo

```
SELECT DATE_ADD("2022-08-03", INTERVAL 4 YEAR)
```

Funciones escalares con fechas

CURTIME() devuelve la hora actual

TIME_TO_SEC() cuántos segundos transcurrieron en el tiempo indicado

TIMEDIFF() período de tiempo entre las dos horas indicadas

Funciones con Horas

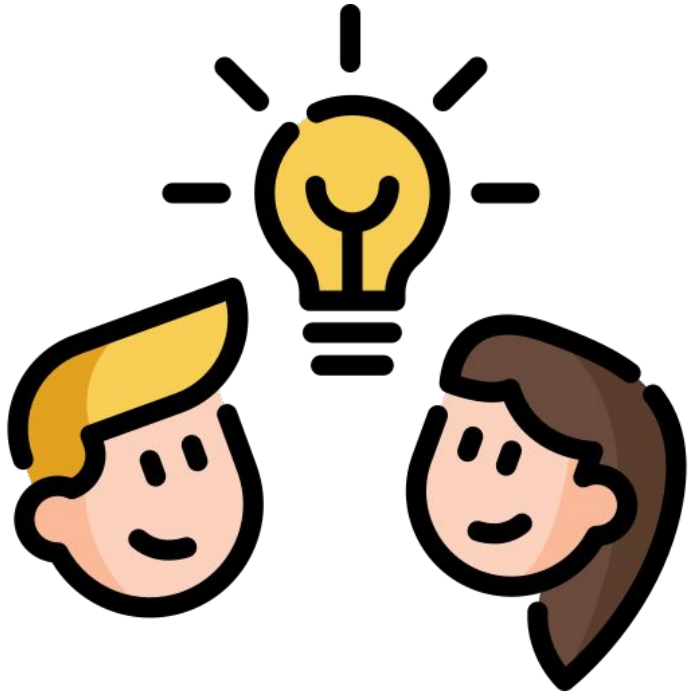
```
SELECT CURTIME();

SELECT TIME_TO_SEC("19:30:10");

SELECT TIMEDIFF(HORA1, HORA2);
```

Funciones escalares con fechas

Muchas de estas funciones escalares que manejan fechas y horas pueden “causarnos ruido” a nivel de no comprender para qué se podrían utilizar.

Si aún no tenemos experiencia en empresas, esto es muy normal. Pero ya, cuando formas parte del mercado IT y comienzas a interactuar con otras áreas corporativas, muchos de estos términos se volverán moneda corriente.

Allí es donde verás, para casos puntuales como ser emitir informes gerenciales, muchas de estas funciones escalares nos ahorrarán tiempos notables en procesar cálculos complejos y agrupar resultados.



Sección práctica

Vayamos con un último ejercicio aplicando funciones escalares.

En esta oportunidad trabajaremos con la función escalar de concatenación de datos y con la función escalar de fecha, aplicándolas sobre la tabla *Employees*, la cual contiene información de los empleados de la bb.dd Northwind.

Prácticas

Necesitamos simplificar la visualización de datos de esta tabla, presentando en una consulta de selección, los siguientes campos:

- **EmployeeID, TitleOfCourtesy, LastName, FirstName, Title, BirthDate, HireDate**

Sobre esta consulta de selección base, realiza las siguientes consignas:

1. En una nueva consulta de selección con la base anterior, concatena los campos:
   a. **(TitleOfCourtesy, LastName, FirstName)** con el alias **NombreCompleto**
   b. respeta los espacios entre los diferentes campos mencionados
2. En una nueva consulta de selección con la base inicial:
   a. elimina el formato fecha y hora sobre el campo **BirthDate**, utilizando la función Date()
   b. aplica un alias a dicho campo para llamarlo **FechaNacimiento**
3. Copia la consulta resultante del punto dos, y modifícala aplicando lo siguiente:
   a. utiliza la función YEAR sobre campo **HireDate**, para mostrar sólo el año de contratación
   b. aplica un alias a dicha campo, para llamarlo **AnioContratacion**

Muchas gracias.

Ministerio de Economía
Argentina

Secretaría de
Economía del Conocimiento

primero
la gente