

# Elementos Fundamentales de SQL

Primera parte

**DETECTAR SI LA BASE DE DATOS  
EXISTE**

Podemos detectar si existe el registro en la tabla sysdatabases de la Base de Datos master.

```
if (select name from master.sys.sysdatabases
    where name='Agenda') is null
    select 'no existe';
else
    select 'existe';
```

También tenemos una función del sistema que lo valida

```
if DB_ID('Agenda2') is null  
    select 'no existe';  
else  
    select 'existe';
```

## Ejemplo

Colocar las instrucciones que permitan que al ejecutar el archivo sql de creación y carga de la Base de datos del Jardín Botánico, permita el borrado cuando exista y la creación posterior.

# **TIPO DE DATO FECHA Y HORA**

Para almacenar valores de tipo FECHA Y HORA SQL Server dispone de dos tipos:

- datetime: puede almacenar valores desde 01 de enero de 1753 hasta 31 de diciembre de 9999. (De 1787 es la Constitución USA)
- smalldatetime: el rango va de 01 de enero de 1900 hasta 06 de junio de 2079.

Las fechas se insertan entre comillas simples.

Para almacenar valores de tipo fecha se permiten como separadores "/", "-" y ".".

SQL Server reconoce varios formatos de entrada de datos de tipo fecha.

Para establecer el orden de las partes de una fecha (día, mes y año) empleamos "set dateformat". Estos son los formatos:

- mdy: 4/15/96 (mes y día con 1 ó 2 dígitos y año con 2 ó 4 dígitos),
- myd: 4/96/15
- dmy: 15/4/1996
- dym: 15/96/4
- ydm: 96/15/4
- ydm: 1996/15/4

Podemos determinar el idioma que usaremos por defecto, que tendrá múltiples efectos en la denominación y formatos de la fecha.

@@language nos da el lenguaje actual

sp\_helplanguage nos muestra los lenguajes disponibles

set language nos permite cambiarlo

```
select @@language;  
go  
exec sp_helplanguage;  
go  
set language us_english;  
go  
set language Español;  
go
```



## Ejemplo de salida de sp\_helplanguage

Resultados Mensajes										
	langid	dateformat	datefirst	upgrade	name	alias	months	shortmonths	days	lcid
1	0	mdy	7	0	us_english	English	January,February,Mar...	Jan,Feb,Mar,Apr,...	Monday,Tuesday,Wednesday,...	1033
2	1	dmy	1	0	Deutsch	German	Januar,Februar,März,...	Jan,Feb,Mär,Apr,...	Montag,Dienstag,Mittwoch,Do...	1031
3	2	dmy	1	0	Français	French	janvier,février,mars,av...	janv,févr,mars,avr,...	lundi,mardi,mercredi,jeudi,vendr...	1034
4	3	ymd	7	0	日本語	Japanese	01,02,03,04,05,06,07...	01,02,03,04,05,06...	月曜日,火曜日,水曜日,木曜...	1041
5	4	dmy	1	0	Dansk	Danish	januar,februar,marts,a...	jan,feb,mar,apr,ma...	mandag,tirsdag,onsdag,torsdag...	1035
6	5	dmy	1	0	Español	Spanish	Enero,Febrero,Marzo,...	Ene,Feb,Mar,Abr,...	Lunes,Martes,Miércoles,Jueve...	3081

Todos los valores de tipo "datetime" se muestran en formato "año-mes-día hora:minuto:segundo .milisegundos", independientemente del formato de grabación.

Hay que usar funciones específicas para presentarlo en otro formato.

Por ejemplo:

`convert(varchar,campoFechaHora,103)`

Nos lo muestra en formato dd/mm/aaaa

## Usaremos la base de datos alquilerCochesBasicas

```
use alquilercochesbasicas;
go

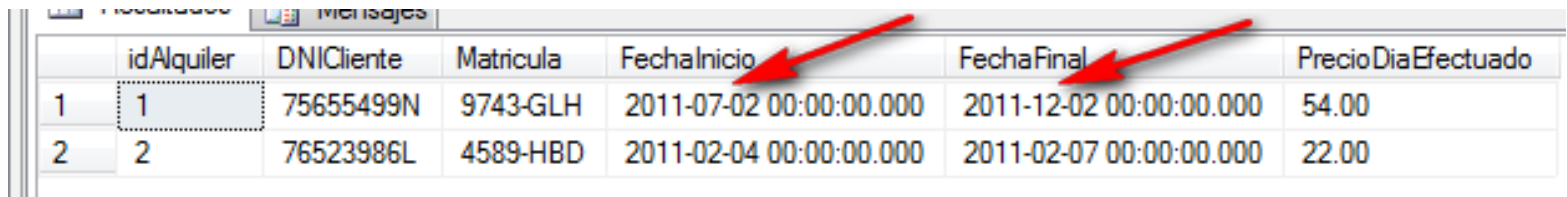
if object_id('ALQ_Alquiler2') is not null
    drop table ALQ_Alquiler2;
go

Create table ALQ_Alquiler2
(idAlquiler int identity primary key,
 DNICliente varchar(9),
 Matricula varchar(10),
 FechaInicio datetime,
 FechaFinal datetime,
 PrecioDiaEfectuado numeric(10,2)
);
go
```

Si tenemos la configuración en inglés nos pedirá formato de fechas mdy, salvo que le indiquemos lo contrario.

Si el nº del día no es mayor de 12 no dará error pero lo almacenado será incorrecto.

```
set language us_english;
go
insert ALQ_Alquiler2
( DNICliente,Matricula,FechaInicio,FechaFinal,PrecioDiaEfectuado )
values ( '75655499N','9743-GLH','7/2/2011','12/2/2011',54.00 );
set dateformat dmy;
insert ALQ_Alquiler2
( DNICliente,Matricula,FechaInicio,FechaFinal,PrecioDiaEfectuado )
values ( '76523986L','4589-HBD','4/2/2011','7/2/2011',22.00 );
go
select
idAlquiler,DNICliente,Matricula,FechaInicio,FechaFinal,PrecioDiaEfectuado
from ALQ_Alquiler2;
go
```



	idAlquiler	DNICliente	Matricula	FechaInicio	FechaFinal	PrecioDiaEfectuado
1	1	75655499N	9743-GLH	2011-07-02 00:00:00.000	2011-12-02 00:00:00.000	54.00
2	2	76523986L	4589-HBD	2011-02-04 00:00:00.000	2011-02-07 00:00:00.000	22.00

Tendremos que cuidar también las cláusulas where

```
set language us_english;  
go  
select idAlquiler, FechaInicio  
from ALQ_Alquiler2  
where FechaInicio='04/02/2011';  
go  
set dateformat dmy;  
go  
select idAlquiler, FechaInicio  
from ALQ_Alquiler2  
where FechaInicio='04/02/2011';  
go
```

Resultados		Mensajes
idAlquiler		FechaInicio
1	2	2011-02-04 00:00:00.000
2	4	2011-02-04 00:00:00.000

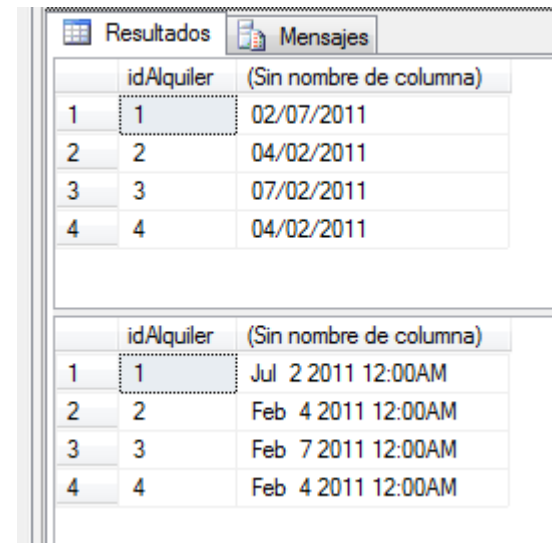
Si tenemos el lenguaje español activado no tendremos que colocar nada para entrar las fechas en formato dmy

```
set language Español;
go
insert ALQ_Alquiler2
( DNICliente,Matricula,FechaInicio,FechaFinal,PrecioDiaEfectuado )
values ( '75655499N','9743-GLH','7/2/2011','12/2/2011',54.00 );
insert ALQ_Alquiler2
( DNICliente,Matricula,FechaInicio,FechaFinal,PrecioDiaEfectuado )
values ( '76523986L','4589-HBD','4/2/2011','7/2/2011',22.00 );
go
select
idAlquiler,DNICliente,Matricula,FechaInicio,FechaFinal,PrecioDiaEfectuado
from ALQ_Alquiler2;
go
```

3	3	75655499N	9743-GLH	2011-02-07 00:00:00.000	2011-02-12 00:00:00.000	54.00
4	4	76523986L	4589-HBD	2011-02-04 00:00:00.000	2011-02-07 00:00:00.000	22.00

El formato de salida será siempre el mismo y tendremos que aplicar funciones para especificar otro.  
Las veremos en detalle más adelante.

```
select  
idAlquiler, convert (varchar, FechaInicio, 103)  
from ALQ_Alquiler2;  
go  
select  
idAlquiler, convert (varchar, FechaInicio, 100)  
from ALQ_Alquiler2;  
go
```



The screenshot shows the 'Resultados' (Results) pane in SQL Server Enterprise Manager. It displays two query results side-by-side. The top result shows the date in YYYY-MM-DD format (103), and the bottom result shows the date in full English format (100).

	idAlquiler	(Sin nombre de columna)
1	1	02/07/2011
2	2	04/02/2011
3	3	07/02/2011
4	4	04/02/2011

	idAlquiler	(Sin nombre de columna)
1	1	Jul 2 2011 12:00AM
2	2	Feb 4 2011 12:00AM
3	3	Feb 7 2011 12:00AM
4	4	Feb 4 2011 12:00AM

# Ejemplo

Usar la base de datos RestauranteBasicas.

Crear la Base de Datos, sus tablas y añadir los registros que contiene el archivo sql.

Ajustar el lenguaje a inglés.

Entrar dos comidas con los id 22 y 23, con fecha 13 de marzo de 2013 y 26 de diciembre de 2013.

Validar que ha entrado la fecha correctamente.

Ajustar el lenguaje a español.

Entrar dos comidas con los id 24 y 25, con fecha 18 de marzo de 2013 y 25 de julio de 2013.

Validar que ha entrado la fecha correctamente.

Hacer un select con filtro a través del campo fecha. Ajustarlo para que filtre correctamente.

Mostrar la fecha de la tabla en un formato más claro.



# **VALORES POR DEFECTO (DEFAULT)**

Cuando no asignamos valor a un campo se le pondrá el valor NULL. Salvo el uso del identity.

Dará error si no permitimos null en ese campo.

Podemos establecer un valor que se usará en el caso que no especifiquemos valor al campo en el insert. Se denominará valor por defecto.

La cláusula en la creación de la tabla será:

```
create table nombredetabla
(
    campo tipo,
    campo tipo default valorpordefecto,
    ...
);
```

Creamos la base de datos especificando dos valores por defecto.  
DescripcionEstado será Bueno si no se especifica.  
codTipo será 1 si no se especifica.

```
use AlquilerCochesBasicas;
go
if object_id('ALQ_Coche2') is not null
    drop table ALQ_Coche2;
go
create table ALQ_Coche2
(Matricula varchar(10) primary key,
 DescripcionEstado varchar(100) default 'Bueno',
 codTipo integer default 1
);
go
```

Si no se especifica DescripcionEstado se pondrá Bueno. Si no se especifica codTipo se pondrá 1, si se pone en la lista de valores default en el correspondiente a DescripcionEstado se pondrá Bueno.

```
insert ALQ_Coche2
( Matricula,codTipo )
values ( '2354-HBC',2 );
insert ALQ_Coche2
( Matricula,DescripcionEstado)
values ( '3216-BHF','Seat Ibiza 3 puertas');
insert ALQ_Coche2
( Matricula )
values ( '3256-GDF' );
insert ALQ_Coche2
( Matricula,DescripcionEstado,codTipo )
values ( '4589-HBZ',default,3 );
go
insert ALQ_Coche2
( Matricula,DescripcionEstado,codTipo )
values ( '4589-HBD','Volkswagen Polo 3 puertas',3 );
go
```

Resultados		Mensajes	
	Matricula	DescripcionEstado	codTipo
1	2354-HBC	Bueno	2
2	3216-BHF	Seat Ibiza 3 puertas	1
3	3256-GDF	Bueno	1
4	4589-HBD	Volkswagen Polo 3 puertas	3
5	4589-HBZ	Bueno	3

Para grabar en un registro un valor NULL en un campo con default definido tendremos que poner explícitamente NULL en el insert como valor para ese campo.

```
insert ALQ_Coche2
( Matricula, DescripcionEstado, codTipo )
values ( '9999-HBD', null, 3 );
go
select Matricula, DescripcionEstado, codTipo
from ALQ_Coche2
where matricula='9999-HBD';
go
```

Resultados		Mensajes	
	Matricula	DescripcionEstado	codTipo
1	9999-HBD	NULL	3

## Ejemplo

Usar la base de datos RestauranteBasicas.

Crear la Base de Datos, sus tablas y añadir los registros que contiene el archivo sql.

Crear una tabla TipoPlato2 igual que la TipoPlato, dando como valor por defecto del campo TipoPlato a Bebidas básicas y en el campo Agrupa a Bebida.

Probar insertando varios registros a TipoPlato2 que obliguen a usar los 2 valores por defecto y cada uno por separado.

Añadiendo otro registro usando la cláusula default en la lista de valores.

Mostar los datos para comprobar que la tabla actuó como se definió.

# **COLUMNAS CALCULADAS (OPERADORES ARITMÉTICOS Y DE CONCATENACIÓN) Y ALIAS**



Los operadores son símbolos que permiten realizar distintos tipos de operaciones.

Podemos usarlos en vez de columnas.

Por ejemplo permitirá realizar operaciones con campos en el select, mostrando el resultado de la operación para cada registro seleccionado.

Los operadores aritméticos permiten realizar cálculos con valores numéricos.

Son:

- cambio de signo con un – delante de un campo o valor

- multiplicación \*

- división /

- módulo % (el resto de dividir números enteros)

- suma +

- resta -

Los paréntesis harán que las operaciones se realicen primero.

## Algunos ejemplos:

```
-- precio semanal
select DescripcionTipo, PrecioDia *7
from ALQ_tipoCoche;
go
--sumarle 3 euros al preciodia
select DescripcionTipo, Preciodia +3
from ALQ_tipoCoche;
go
--Precio con descuento del 10 %
select DescripcionTipo, PrecioDia - (PrecioDia*10/100)
from ALQ_tipoCoche;
go
--Billetes de 10 euros y resto
select DescripcionTipo, cast(PrecioDia/10 as int),
Preciodia % 10
from ALQ_tipoCoche;
go
```

Se puede usar en otros lugares, por ejemplo en el SET del UPDATE:

```
select CodTipo, PrecioDia
from ALQ_tipoCoche
where CodTipo=2;
go
update ALQ_tipoCoche
    set PrecioDia=PrecioDia * 1.07
    where CodTipo=2;
go
select CodTipo, PrecioDia
from ALQ_tipoCoche
where CodTipo=2;
go
```

Resultados		Mensajes
	CodTipo	PrecioDia
1	2	56.00

	CodTipo	PrecioDia
1	2	59.92

La precedencia entre operadores define qué operaciones se realizarán primero.

Primero se ejecuta lo situado entre paréntesis

Después los cambios de signo

Después las multiplicaciones, divisiones y el módulo.

Después las sumas y restas

Dentro de cada grupo se ejecutarán de izquierda a derecha. Salvo el cambio de signo que se valida de derecha a izquierda.

El sistema es como en los lenguajes de programación.

```
-- primero 4-2 y el resultado +27
select 4 - 2 + 27;
go
--primero 2*4 y el resultado +5
select 2 * 4 + 5;
go
--primero 4*5 y el resultado +2
select 2 + 4 * 5;
go
--primero 2+4 y el resultado *5
select (2 + 4) * 5;
go
```

El operador de concatenación permite pegar cadenas, mediante el símbolo más (+).

Se pueden concatenar tanto campos de texto como cadenas de caracteres entre comillas simples.

## Concatenando (pegando) cadenas de caracteres:

```
--apellidos y nombre juntos
select DNICliente, Apellidos+', '+Nombre
from ALQ_Cliente;
go

--nombre, apellidos y datos juntos
select DNICliente,
Nombre+', '+Apellidos+' con datos: '+DatosCliente
from ALQ_Cliente;
go
```

Resultados			Mensajes		
	DNICliente	(Sin nombre de columna)			
1	05679340L	pérez, david			
2	07865988B	hemández, pedro			
3	13455656N	ruiz, maría			
4	23432988J	ramos, maría			
5	23446456A	gonzález, josé			
	DNICliente	(Sin nombre de columna)			
1	05679340L	david pérez con datos: c/galicia nº 32			
2	07865988B	pedro hemández con datos: Rambla nº 7			
3	13455656N	maría ruiz con datos: c/fuerteventura nº 9			

Para hacer más comprensible el resultado de una consulta, sobre todo cuando realizamos cálculos o concatenaciones podemos cambiar el nombre de la columna mediante un **alias**.

```
select OperaciónEntreCampos as NombreDeLaColumna ...
```

Se puede poner entre comillas simples si el alias tiene espacios en blanco.



## Ejemplos de uso de los alias.

```
select DNICliente,Apellidos+', '+Nombre as ApeNom
from ALQ_Cliente;
go
select DNICliente,Apellidos+', '+Nombre as 'Apellidos y nombre'
from ALQ_Cliente;
go
select DescripcionTipo,
       PrecioDia - (PrecioDia*10/100) as PrecioDescuento
from ALQ_tipoCoche;
go
```

Resultados		Mensajes
DNICliente		ApeNom
1	05679340L	pérez, david
2	07865988B	hemández, pedro
3	13455656N	ruiz, maría
DNICliente		Apellidos y nombre
1	05679340L	pérez, david
2	07865988B	hemández, pedro
3	13455656N	ruiz, maría
DescripcionTipo		PrecioDescuento
1	Económico	22.500000
2	Ejecutivo	50.400000
3	Furgoneta	63.000000
4	Todo terreno	67.500000

## Ejemplo

Usar la base de datos RestauranteBasicas.

Crear la Base de Datos, sus tablas y añadir los registros que contiene el archivo sql.

Cuánto costaría 5 platos de todos los que tenemos. Ponerle nombre a la columna calculada.

Mostrar los precios de cada comida con el 5% de descuento. Ponerle nombre a la columna calculada.

Modificar el precio del plato 4 sumándole 3 euros.

Calcular y explicar las siguientes operaciones matemáticas, poniéndole nombre al cálculo:

- $(4+5)*6$
- $3+4*2$
- $-4*5+2$
- $22\%5$

De todos los tipos de plato, mostrar el campo Agrupa seguido por dos puntos y un blanco y el campo TipoPlato. Darle el nombre "Su plato" a la columna.

# **FUNCIONES**

SQL Server nos permite realizar cálculos más complejos de manera automática, sin que nosotros tengamos que realizar programas que permitan resolver el problema.

Esto se realiza mediante funciones.

Cada función está definida para un cálculo específico, le pasaremos un conjunto de valores y devolverá un valor calculado a partir de los mismos.

Podremos usarla en cualquier situación en la que usemos un campo.

Son específicas de cada sistema gestor de bases de datos, por lo que tendremos que revisarlas en cada caso.

El formato será:

`NombreFunción(Parámetro1, parámetro2,...)`

Se podrá usar en el select, por ejemplo con el formato

```
select campo, NombreFunción(campo1,campo2)
from NombreTabla;
```

# Funciones de cadenas de caracteres

<https://docs.microsoft.com/es-es/sql/t-sql/functions/string-functions-transact-sql?view=sql-server-2017>

**Ver el documento sobre FuncionesSQLServer**

## Ejemplos

```
select substring('Buenas tardes',8,6);
--extraer una subcadena dando inicio y longitud
--retorna "tardes".
select str(-123.456,7,3);
--convierte el valor numérico "123.456" a cadena,
--especificando 7 de longitud y 3 decimales
--retorna '-123.46';
  select stuff('abcde',3,2,'xyyyyz');
--inserta la cadena enviada como cuarto argumento,
--en la posición indicada en el segundo argumento,
--reemplazando la cantidad de caracteres indicada
--por el tercer argumento en la cadena que es primer
--parámetro.
--retorna "abxyyyyze"
```

```
select len('Hola');  
--retorna la longitud de la cadena  
--devuelve 4.  
select char(65);  
--retorna un carácter en código ASCII del entero  
--enviado  
--retorna "A".  
select left('buenos dias',8);  
--retorna la cantidad (longitud) de caracteres  
--de la cadena comenzando desde la izquierda  
--retorna "buenos d".  
select right('buenos dias',8);  
--igual pero por la derecha  
--retorna "nos dias".
```

```
select lower('HOLA ESTUDIAnTe');  
--pasa la cadena a minúsculas  
--retorna "hola estudiante".  
select upper('HOLA ESTUDIAnTe');  
--pasa la cadena a mayúsculas  
--retorna "HOLA ESTUDIANTE".  
select ltrim('      Hola      ');  
--retorna la cadena con los espacios de la  
--izquierda eliminados  
--retorna "Hola      ".  
select rtrim('      Hola      ');  
--retorna la cadena con los espacios de la  
--derecha eliminados  
--retorna "      Hola".
```



```
select replace('xxx.elmundo.es','x','w');
--cambia lo indicado en el segundo parámetro
--por el tercero en la cadena primera
--retorna "www.elmundo.es".
select reverse('Hola');
--cambia el orden de la cadena
--retorna "aloH".
select charindex('or','Jorge Luis
Borges',5);
--devuelve dónde comienza la subcadena del
--primer parámetro dentro de la del
--segundo parámetro, comenzando en la
--posición del tercer parámetro.
--retorna 13.
select replicate ('Hola',3);
--repite la cadena el nº de veces indicado en
--el segundo parámetro
--retorna "HolaHolaHola";
```

```
select 'Hola'+space(7)+'que tal';  
--cadena con el nº de espacios indicados  
--retorna "Hola          que tal".  
go
```

Se pueden emplear estas funciones enviando como argumento el nombre de un campo.

```
use alquilercochesbasicas;
go
--sacar los cinco primeros caracteres de
--DescripcionTipo y el preciodia
select left(DescripcionTipo,5),PrecioDia
from ALQ_tipoCoche;
go
--mostrar el DNI y las tres primeras letras
--del nombre y las tres primeras de los
--apellidos de todos los clientes
select
DNICliente,left(Apellidos,3)+left(Nombre,3)
from ALQ_Cliente;
go
```

```

--mostrar todos los clientes cuyo nombre
--termine en a
select
DNICliente,Apellidos,Nombre,DatosCliente,
FechaNacimiento,FechaCarnet
from ALQ_Cliente
where right(Nombre,1)='a';
go
--Pasar nombre y apellidos a mayúsculas
select DNICliente,upper(Apellidos) as
ap,upper(Nombre) as nom
from ALQ_Cliente;
go
--Mostrar matricula, una coma y un blanco,
--descripcionEstado, un blanco y entre
--paréntesis el codTipo pasado a cadena de
--caracteres y sin blancos a la izquierda
select Matricula+', '+DescripcionEstado+
' ('+ltrim(STR(codTipo))+') '
from ALQ_Coche;
go

```

## Ejemplo

Crear la base de datos PrestamoLibros.

Crear la tabla y cargarla con datos:

```
if object_id ('libros') is not null
```

```
drop table libros;
```

```
create table libros(
```

```
codigo int identity,
```

```
titulo varchar(40) not null,
```

```
autor varchar(20) default 'Desconocido',
```

```
editorial varchar(20),
```

```
precio decimal(6,2),
```

```
cantidad tinyint default 0,
```

```
primary key (codigo)
```

```
);
```

Cargar con estos datos:

```
insert into libros (titulo,autor,editorial,precio)
```

```
values('El aleph','Borges','Emece',25);
```

```
insert into libros
```

```
values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
```

```
insert into libros (titulo,autor,editorial,precio,cantidad)
```

```
values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);
```

Mostrar las 3 primeras letras de todos los títulos.  
Mostrar el precio como cadena de caracteres.  
Mostrar la cadena con el titulo, un guión, el autor un guión y el precio.  
Mostrar las seis últimas letras del titulo y del autor.  
Mostrar el nombre del autor en mayúscula.  
Indicar el número de letras del autor y del título.  
Mostrar los caracteres del 4 al 10 del autor  
Cambiar arroba por el carácter arroba y punto por el carácter punto en el texto correoarroba@hotmailpuntocom

# Funciones matemáticas

<https://docs.microsoft.com/es-es/sql/t-sql/functions/mathematical-functions-transact-sql?view=sql-server-2017>

**Ver el documento sobre FuncionesSQLServer**

## Ejemplos:

```
select abs(-23);  
--valor absoluto del argumento  
--retorna 23.  
select ceiling(12.34);  
--redondea hacia arriba  
--retorna 13.  
select floor(12.34);  
--redondea hacia abajo  
--retorna 12.  
select power(2,3);  
--primer argumento elevado al segundo argumento  
--retorna 8.
```



```
select round(12.35,1);  
--redondea a la longitud de decimales especificada  
--en el segundo argumento  
--retorna 12.40  
go
```

Se pueden emplear estas funciones enviando como argumento el nombre de un campo.

```
use alquilercochesbasicas;
go
--sacar precio redondeado
select CodTipo,PrecioDia,
       round(PrecioDia,0) as precioRedondeado
from ALQ_tipoCoche;
go
--elevar al cubo el precio
select CodTipo,PrecioDia,
       power(PrecioDia,3) as precioAlCubo
from ALQ_tipoCoche;
go
```

## Ejemplo

Redondear 4567.345 con 2 decimales

Truncar 4567.356 con 1 decimal

Raíz cuadrada de 625

Resultado y resto de dividir 16 entre 3

Cuadrado de 12

Valor absoluto de la diferencia 23-56

Usar la base de datos RestauranteBasicas.

Crear la Base de Datos, sus tablas y añadir los registros que contiene el archivo sql.

Dar el precio del plato redondeado sin decimales.

Resultado de dividir el precio entre diez, dando el cociente entero y el resto por separado.

# Funciones de fecha y hora

<https://docs.microsoft.com/es-es/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-2017>

**Ver el documento sobre FuncionesSQLServer**

## Ejemplos

```
--tener en cuenta ejecutar el formato para
--asegurarnos que las fechas se entiendan
--adecuadamente por el SGBD
set dateformat ymd;
go
select getdate();
go
--retorna la fecha y hora actuales
select datepart(month,getdate());
go
--Retorna el número de mes actual
select datepart(day,getdate());
go
--Retorna el día actual
```

```

select datepart(hour,getdate());
go
--Retorna la hora actual
select datename(month,getdate());
go
--Retorna el nombre del mes actual
select datename(dw,getdate());
go
--Retorna el nombre del día de la semana de la fecha
actual
select dateadd(day,3,'1980/11/02');
go
--agrega el número de intervalos a la fecha del tercer
--parámetro
--(intervalo es primer parámetro y el número el segundo)
--retorna "1980/11/05", agrega 3 días.

```

```
select dateadd(month,3,'1980/11/02');
go
--retorna "1981/02/02", agrega 3 meses.
select dateadd(hour,2,'1980/11/02');
go
--retorna "1980/02/02 2:00:00", agrega 2 horas.
select dateadd(minute,16,'1980/11/02');
go
--retorna "1980/02/02 00:16:00", agrega 16 minutos.
```

```
select datediff (day, '2005/10/28', '2006/10/28');  
--calcula los intervalos (primer parámetro)  
--entre las dos fechas (segundo y tercer parámetro)  
--Cuidado con el tratamiento de los meses y años  
--ya que no tiene en cuenta que haya transcurrido  
--el último  
--retorna 365 (días).  
select datediff(month, '2005/10/28', '2006/11/27');  
--retorna 13 (meses), ya que calcula la diferencia  
--en meses, sin tener en cuenta si ha transcurrido  
--o no el último mes
```



```
--todas se puede aplicar a cualquier fecha  
select day(getdate());  
--día de la fecha actual  
select month(getdate());  
--mes de la fecha actual  
select year(getdate());  
--año de la fecha actual
```

Se pueden emplear estas funciones enviando como argumento el nombre de un campo fechaHora donde corresponda.

```
use alquilercochesbasicas;
go
--sacar el nombre del mes en que nacieron los clientes
select DNICliente,Apellidos,Nombre,
       datename(month,FechaNacimiento) as mesNacimiento
from ALQ_Cliente;
go

--Mostrar los clientes nacidos en el día 12
select DNICliente,Apellidos,Nombre,FechaNacimiento
from ALQ_Cliente
where day(fechanacimiento)=12;
```

```
--Mostar los alquileres que comiencen en Miércoles
select idAlquiler,DNICliente,Matricula,FechaInicio,
FechaFinal,PrecioDiaEfectuado
from ALQ_Alquiler
where datename(dw,fechainicio)='Miércoles';
go
```

```
--Mostrar el número de días de cada alquiler
select idAlquiler,DNICliente,Matricula,FechaInicio,
FechaFinal,PrecioDiaEfectuado,
datediff(day,FechaInicio,fechaFinal) as Ndias
from ALQ_Alquiler;
go
```

```
--Días de diferencia entre la fecha de inicio de
--cada alquiler y la fecha actual
select idAlquiler,DNICliente,Matricula,FechaInicio,
FechaFinal,PrecioDiaEfectuado,
datediff(day,FechaInicio,GETDATE()) as Ndias
from ALQ_Alquiler;
go
```

## Ejemplo

Mostrar la fecha actual

Indicar el nombre del día de la semana de hoy

Indicar el nº del mes de la fecha actual.

Calcular el número de días de diferencia entre el 25/12/2010 y la fecha actual.

Dar el nº del año actual

Sumar 35 días a la fecha actual.

Usar la base de datos RestauranteBasicas.

Crear la Base de Datos, sus tablas y añadir los registros que contiene el archivo sql.

Calcular el número de días transcurridos entre la fecha actual y la fecha de cada comida.

Dar las comidas efectuadas en Domingo.

Dar el número del mes de cada comida.