

# Funciones de alteración

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# CONCAT

Usamos **CONCAT** para **concatenar** dos o más expresiones:

```
SQL SELECT CONCAT('Hola', ' a ', 'todos.');
```

```
> 'Hola a todos.'
```

```
SQL SELECT CONCAT('La respuesta es: ', 24, '.');
```

```
> 'La respuesta es 24.'
```

```
SQL SELECT CONCAT('Nombre: ', apellido, ', ', nombre, '.')  
FROM actor;
```

```
> 'Nombre: Clarke, Emilia.'
```

# COALESCE

Usamos **COALESCE** para sustituir el valor **NULL** en una sucesión de expresiones o campos. Es decir, si la primera expresión es Null, se sustituye con el valor de una segunda expresión, pero si este valor también es Null, se puede sustituir con el valor de una tercera expresión y así sucesivamente.

```
SQL SELECT COALESCE(NULL, 'Sin datos');  
> 'Sin datos'
```

```
SQL SELECT COALESCE(NULL, NULL, 'Digital House');  
> 'Digital House'
```

## COALESCE (cont.)

Los tres clientes de la siguiente tabla poseen uno o más datos nulos:

```
SQL SELECT id, apellido, nombre, telefono_movil, telefono_fijo
FROM cliente;
```

cliente				
id	apellido	nombre	telefono_movil	telefono_fijo
1	Pérez	Juan	1156685441	43552215
2	Medina	Rocío	Null	43411722
3	López	Matías	Null	Null

## COALESCE (cont.)

Usando **COALESCE** podremos sustituir los **datos nulos** en cada registro, indicando la columna a evaluar y el valor de sustitución.

SQL

```
SELECT id, apellido, nombre, COALESCE(telefono_movil, telefono_fijo, 'Sin datos')  
AS telefono FROM cliente;
```

cliente			
id	apellido	nombre	telefono
1	Pérez	Juan	1156685441
2	Medina	Rocío	43411722
3	López	Matías	Sin datos

# DATEDIFF

Usamos **DATEDIFF** para devolver la **diferencia** entre dos fechas, tomando como granularidad el intervalo especificado.

```
SQL SELECT DATEDIFF('2021-02-03 12:45:00', '2021-01-01 07:00:00');
```

```
> 33
```

Devuelve 33 porque es la cantidad de días de la diferencia entre las fechas indicadas.

```
SQL SELECT DATEDIFF('2021-01-15', '2021-01-05');
```

```
> 10
```

Devuelve 10 porque es la cantidad de días de la diferencia entre las fechas indicadas.

# TIMEDIFF

Usamos **TIMEDIFF** para devolver la **diferencia** entre dos horarios, tomando como granularidad el intervalo especificado.

```
SQL SELECT TIMEDIFF('2021-01-01 12:45:00', '2021-01-01 07:00:00');  
> 05:45:00
```

```
SQL SELECT TIMEDIFF('18:45:00', '12:30:00');  
> 06:15:00
```

# EXTRACT

Usamos **EXTRACT** para **extraer** partes de una fecha:

```
SQL SELECT EXTRACT(SECOND FROM '2014-02-13 08:44:21');  
> 21
```

```
SQL SELECT EXTRACT(MINUTE FROM '2014-02-13 08:44:21');  
> 44
```

```
SQL SELECT EXTRACT(HOUR FROM '2014-02-13 08:44:21');  
> 8
```

```
SQL SELECT EXTRACT(DAY FROM '2014-02-13 08:44:21');  
> 13
```



# EXTRACT (cont.)

```
SQL SELECT EXTRACT(WEEK FROM '2014-02-13 08:44:21');  
> 6
```

```
SQL SELECT EXTRACT(MONTH FROM '2014-02-13 08:44:21');  
> 2
```

```
SQL SELECT EXTRACT(QUARTER FROM '2014-02-13 08:44:21');  
> 1
```

```
SQL SELECT EXTRACT(YEAR FROM '2014-02-13 08:44:21');  
> 2014
```

# REPLACE

Usamos **REPLACE** para reemplazar una cadena de caracteres por otro valor. Cabe aclarar que esta función hace distinción entre minúsculas y mayúsculas.

```
SQL SELECT REPLACE('Buenas tardes', 'tardes', 'Noches');
```

```
> Buenas Noches
```

```
SQL SELECT REPLACE('Buenas tardes', 'a', 'A');
```

```
> BuenAs tArdes
```

```
SQL SELECT REPLACE('1520', '2', '5');
```

```
> 1550
```

# DATE\_FORMAT

Usamos **DATE\_FORMAT** para cambiar el formato de salida de una fecha según una condición dada.

```
SQL SELECT DATE_FORMAT('2017-06-15', '%Y');  
    > 2017
```

```
SQL SELECT DATE_FORMAT('2017-06-15', '%W %M %e %Y');  
    > Thursday June 15 2017
```

Para mostrar la fecha escrita en español se debe configurar el idioma con la siguiente instrucción:

```
SQL SET lc_time_names = 'es_ES';  
    SELECT DATE_FORMAT('2017-06-15', '%W, %e de %M de %Y');  
    > jueves, 15 de junio de 2017
```

# DATE\_ADD

Usamos **DATE\_ADD** para sumar o agregar un período de tiempo a un valor de tipo DATE o DATETIME.

```
SQL SELECT DATE_ADD('2021-06-30', INTERVAL '3' DAY);  
> 2021-07-03
```

```
SQL SELECT DATE_ADD('2021-06-30', INTERVAL '9' MONTH);  
> 2022-03-30
```

```
SQL SELECT DATE_ADD('2021-06-30 09:30:00', INTERVAL '4' HOUR);  
> 2021-06-30 13:30:00
```

# DATE\_SUB

Usamos **DATE\_SUB** para restar o quitar un período de tiempo a un valor de tipo DATE o DATETIME.

```
SQL SELECT DATE_SUB('2021-06-30', INTERVAL '3' DAY);  
> 2021-06-27
```

```
SQL SELECT DATE_SUB('2021-06-30', INTERVAL '9' MONTH);  
> 2020-09-30
```

```
SQL SELECT DATE_SUB('2021-06-30 09:30:00', INTERVAL '4' HOUR);  
> 2021-06-30 05:30:00
```

# CASE

Usamos **CASE** para **evaluar condiciones** y devolver la primera condición que se cumpla. En este ejemplo, la tabla resultante tendrá 4 columnas: id, titulo, rating, calificacion. Esta última columna mostrará los valores: Mala, Regular, Buena y Excelente; **según** el **rating** de la película.

SQL

```
SELECT id, titulo, rating,  
CASE  
  WHEN rating < 4 THEN 'Mala'  
  WHEN rating BETWEEN 4 AND 6 THEN 'Regular'  
  WHEN rating BETWEEN 7 AND 9 THEN 'Buena'  
  ELSE 'Excelente'  
END AS calificacion  
FROM pelicula;
```

## CASE (cont.)

A continuación, se muestra la tabla resultante después de haber aplicado la función **CASE**.

pelicula			
id	titulo	rating	calificacion
1	El Padrino	6	Regular
2	Tiburón	4	Regular
3	Jurassic Park	9	Buena
4	Titanic	10	Excelente
5	Matrix	3	Mala

**DigitalHouse** >  
Coding School