

# Funciones MySQL

# 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: ', first_name, ' ', last_name)
FROM actors;
```

```
> 'Nombre: Emilia Clarke'
```

# Coalesce

Usamos **COALESCE** para obtener la **primera expresión** que **no sea NULL**:

```
SQL SELECT COALESCE(NULL, 1, 20, 'Digital House');  
> 1
```

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

# Coalesce

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

CLIENTES				
id	nombre	celular	casa	trabajo
1	Juan	124	345	980
2	Rocío		187	243
3	Matías			428

# Coalesce

Usando **COALESCE** podremos obtener el **primer dato no nulo** de cada registro, aclarando las columnas a tener en cuenta.

SQL

```
SELECT id, nombre, COALESCE(celular, casa, trabajo) AS telefono  
FROM clientes;
```

CLIENTES		
id	nombre	telefono
1	Juan	124
2	Rocío	187
3	Matías	428

# Datediff

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

```
SQL SELECT DATEDIFF('2017/08/25', '2017/08/15');
```

```
> 10
```

Devuelve 10 porque es la cantidad de días de diferencia entre el día 25 y el 15.

# 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

```
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 secuencia de caracteres por otra en un string.

```
SQL  SELECT REPLACE('abc abc', 'a', 'B');  
      > Bbc Bbc
```

```
SQL  SELECT REPLACE('abc abc', 'A', 'B');  
      > abc abc  
      -- no se encuentran coincidencias para reemplazar
```

```
SQL  SELECT REPLACE('123 123', '2', '5');  
      > 153 153
```

# Date format

Usamos **DATE\_FORMAT** para que dada una fecha determinada se pueda formatear la misma según deseemos.

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

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

Descubre todas las [posibilidades de formato](#) que posee esta función.

# Case

Usamos **CASE** para **evaluar condiciones** y devolver la primera que se cumpla. En este ejemplo, la tabla resultante tendrá 4 columnas: *id*, *title*, *rating* y *rating\_categories*. Esta última mostrará 'Mala', 'Regular', etc., según el **rating** de la película.

SQL

```
SELECT id, title, rating
CASE
  WHEN rating < 4 THEN 'Mala'
  WHEN rating < 6 THEN 'Regular'
  WHEN rating < 8 THEN 'Buena'
  WHEN rating < 9.5 THEN 'Muy buena'
  ELSE 'Excelente'
END AS rating_categories
FROM movies
ORDER BY rating
```

DigitalHouse>  
Coding School