



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



MATERIA: Bases de Datos

PRÁCTICA 2B: Funciones en SQL.

ALUMNO: Meza Vargas Brandon David.

BOLETA: 2020630288

PROFESOR: Nancy Ocotitla Rojas.

FECHA PRÁCTICA: 13-01-2021



DESARROLLO

Las funciones revisadas en esta práctica son referentes al uso de cadenas, recordemos que antes de cada función debemos de poner la palabra select como vemos en las capturas.

FUNCIONES:

- **ascii('cadena')** - Devuelve el ascii del carácter más a la izquierda de la cadena, si ponemos un carácter nos devuelve su ascii;

```
mysql> select ascii('pedro');
+-----+
| ascii('pedro') |
+-----+
|          112 |
+-----+
1 row in set (0.00 sec)

mysql> select ascii('p');
+-----+
| ascii('p') |
+-----+
|          112 |
+-----+
1 row in set (0.00 sec)
```

- **char(n)** – Nos devuelve el carácter ascii del numero entero n, en mi caso lo devuelve en hexadecimal por la configuración que tengo;

```
mysql> select char(59);
+-----+
| char(59) |
+-----+
| 0x3B    |
+-----+
1 row in set (0.00 sec)
```

- **char(n, m)** – Devuelve y concatena los caracteres de código ascii n, m... En mi caso lo devuelve en hexadecimal por la configuración que tengo.

```
mysql> select char(59);
+-----+
| char(59) |
+-----+
| 0x3B    |
+-----+
1 row in set (0.00 sec)

mysql> select char(60);
+-----+
| char(60) |
+-----+
| 0x3C    |
+-----+
1 row in set (0.00 sec)

mysql> select char(50, 60);
+-----+
| char(50, 60) |
+-----+
| 0x323C    |
+-----+
1 row in set (0.00 sec)
```

- **bin(n):** Devuelve el numero decimal ingresado en su equivalente en binario:

```
mysql> select bin(10);
+-----+
| bin(10) |
+-----+
| 1010    |
+-----+
1 row in set (0.00 sec)

mysql> select bin(2);
+-----+
| bin(2)  |
+-----+
| 10      |
+-----+
1 row in set (0.00 sec)
```

- **concat('cadena1','cadena2')** – Devuelve la concatenación de dos cadenas dadas;

```
mysql> select concat('hola',' amigo');
+-----+
| concat('hola',' amigo') |
+-----+
| hola amigo              |
+-----+
1 row in set (0.00 sec)
```

- **concat_ws('carácter separador', 'cadena1','cadena2')** – Separa una cadena de acuerdo a un carácter establecido, dicho de otra forma, pone un carácter en medio de dos cadenas;

```
mysql> select concat_ws('*', 'hola', 'amigo');
+-----+
| concat_ws('*', 'hola', 'amigo') |
+-----+
| hola*amigo                      |
+-----+
1 row in set (0.00 sec)
```

- **Hexa(n)** – Devuelve el equivalente en hexadecimal de un entero decimal;

```
mysql> select hex(15);
+-----+
| hex(15) |
+-----+
| F       |
+-----+
1 row in set (0.00 sec)
```

- **oct(n)** -Devuelve el equivalente en octal de un entero decimal;

```
mysql> select oct(15);
+-----+
| oct(15) |
+-----+
| 17      |
+-----+
1 row in set (0.00 sec)
```

- **lower('cadena') / lcase('cadena')** – Ambas funciones regresan una cadena en minúsculas;

```
mysql> select lower('HOLA');
+-----+
| lower('HOLA') |
+-----+
| hola          |
+-----+
1 row in set (0.00 sec)

mysql> select lcase('HOLA');
+-----+
| lcase('HOLA') |
+-----+
| hola          |
+-----+
```

- **upper('cadena') / ucase('cadena')** – Ambas devuelven una cadena en mayúsculas;

```
mysql> select upper('hola');
+-----+
| upper('hola') |
+-----+
| HOLA          |
+-----+
1 row in set (0.00 sec)

mysql> select ucase('hola');
+-----+
| ucase('hola') |
+-----+
| HOLA          |
+-----+
```

- **left('cadena', n)** – Devuelve los primeros n caracteres de una cadena empezando desde la izquierda;

```
mysql> select left('brandon',3);
+-----+
| left('brandon',3) |
+-----+
| bra              |
+-----+
1 row in set (0.00 sec)

mysql> select left('brandon',5);
+-----+
| left('brandon',5) |
+-----+
| brand            |
+-----+
```

- **right('cadena',n)** – Devuelve los primeros n caracteres de una cadena empezando desde la derecha;

```
mysql> select right('brandon',5);
+-----+
| right('brandon',5) |
+-----+
| andon              |
+-----+
1 row in set (0.00 sec)

mysql> select right('brandon',3);
+-----+
| right('brandon',3) |
+-----+
| don                |
+-----+
```

- **length('cadena')** – Devuelve la longitud de una cadena;

```
mysql> select length('brandon');
+-----+
| length('brandon') |
+-----+
| 7                  |
+-----+
```

- **bit_length('cadena')** – Devuelve la longitud en bits de una cadena;

```
mysql> select bit_length('brandon');
+-----+
| bit_length('brandon') |
+-----+
| 56                     |
+-----+
```

- **ltrim('cadena')** – Regresa una cadena quitando los espacios que tenga al inicio;

```
mysql> select ltrim('   brandon');
+-----+
| ltrim('   brandon') |
+-----+
| brandon             |
+-----+
1 row in set (0.00 sec)
```

- **rtrim('cadena')** – Regresa una cadena quitando los espacios a la derecha;

```
mysql> select rtrim('brandon   ');
+-----+
| rtrim('brandon   ') |
+-----+
| brandon              |
+-----+
1 row in set (0.00 sec)
```

- **trim('cadena')** – Regresa la cadena quitando espacios al inicio y al final;

```
mysql> select trim('          brandon ');
+-----+
| trim('          brandon ') |
+-----+
| brandon                    |
+-----+
```

- **trim(Trailing 'prefijo' from 'cadena')** – Quita el prefijo de la última parte de una cadena;

```
mysql> select trim(Trailing 'on' from 'brandon');
+-----+
| trim(Trailing 'on' from 'brandon') |
+-----+
| brand                              |
+-----+
```

- **trim(Leading 'prefijo' from 'cadena')** – Devuelve una cadena quitando el prefijo especificado desde el inicio de la cadena;

```
mysql> select trim(Leading 'bra' from 'brandon');
+-----+
| trim(Leading 'bra' from 'brandon') |
+-----+
| ndon                              |
+-----+
```

- **trim(Both 'prefijo' from 'cadena')**– Devuelve la cadena quitando el prefijo si lo encuentra en el inicio y al final;

```
mysql> select trim(both 'bra' from 'brandonbra');
+-----+
| trim(both 'bra' from 'brandonbra') |
+-----+
| ndon                              |
+-----+
1 row in set (0.00 sec)
```

- **trim('Prefijo' from 'cadena')** – Devuelve la cadena removiendo el prefijo al inicio y al final de la cadena si es que lo encuentra;

```
mysql> select trim('*' from '*brandon*');
+-----+
| trim('*' from '*brandon*') |
+-----+
| brandon                    |
+-----+
1 row in set (0.00 sec)
```

- **repeat('cadena', n)** – Devuelve la cadena repetida n veces;

```
mysql> select repeat('brandon',5);
+-----+
| repeat('brandon',5) |
+-----+
| brandonbrandonbrandonbrandonbrandon |
+-----+
```

- **reverse('cadena')** – Regresa la cadena invertida;

```
mysql> select reverse('brandon');
+-----+
| reverse('brandon') |
+-----+
| nodnarb            |
+-----+
1 row in set (0.00 sec)
```

- **substring('cadena', n)** – Devuelve la cadena cortada desde un n especificado;

```
mysql> select substring('brandon',5);
+-----+
| substring('brandon',5) |
+-----+
| don                    |
+-----+
1 row in set (0.00 sec)
```

- **substring('cadena', n, m)** – Devuelve la cadena cortada desde un n, el m dice el tamaño que tendrá la cadena después de haber sido cortada;

```
mysql> select substring('brandon',5, 1);
+-----+
| substring('brandon',5, 1) |
+-----+
| d                          |
+-----+
1 row in set (0.00 sec)
```

- **substr('cadena', n, m)** – Devuelve la cadena cortada, empieza en n y m es la cantidad de caracteres a quitar;

```
mysql> select substr('brandon',3, 2);
+-----+
| substr('brandon',3, 2) |
+-----+
| an                     |
+-----+
1 row in set (0.00 sec)
```

- **space(n)** – Devuelve una cadena de n espacios;

```
mysql> select space(10);
+-----+
| space(10) |
+-----+
|           |
+-----+
1 row in set (0.00 sec)
```

- **strcmp('cadena1', 'cadena2')** – Devuelve 0 si las cadenas son las mismas, regresa -1 si la cadena 1 es menor que la cadena 2, y 1 si la cadena 1 es mayor que la 2;

```
mysql> select strcmp('hola','hola');
+-----+
| strcmp('hola','hola') |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)

mysql> select strcmp('hola','holaa');
+-----+
| strcmp('hola','holaa') |
+-----+
| -1 |
+-----+
1 row in set (0.00 sec)

mysql> select strcmp('hola','hol');
+-----+
| strcmp('hola','hol') |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

- **locate('subcadena', 'cadena')** – Devuelve la posición donde se encuentra una subcadena en una cadena;

```
mysql> select locate('n','brandon');
+-----+
| locate('n','brandon') |
+-----+
| 4 |
+-----+
1 row in set (0.00 sec)
```

- **insert('cadena', n, m 'nuevacadena')** – Devuelve una cadena insertando los caracteres especificados de la cadena en la nueva cadena;

```
mysql> select insert('hola',1,3,'brandon');
+-----+
| insert('hola',1,3,'brandon') |
+-----+
| brandona |
+-----+
```

- **replace('cadena', de character, por carácter)** - Devuelve una cadena reemplazando caracteres en ella;

```
mysql> select replace('brandaon', 'a','o');
+-----+
| replace('brandaon', 'a','o') |
+-----+
| brondoon |
+-----+
1 row in set (0.00 sec)
```