

Concurso de Funciones en HeidiSQL

Yo hice las funciones SUBSTRING, UNPIVOT y RANK.

- **SUBSTRING**

Se usa para extraer una parte de una cadena de texto pero hay que darle 3 parámetros.

1. La cadena de la que se extraerá el texto: "Hola Mundo".
2. La posición donde empezará a contar será la "H" el número 1.
3. La cantidad de caracteres a extraer.

```
SELECT SUBSTRING ('Hola Mundo', 6, 5);
```

→ **RESULTADO** 'Mundo'

Aquí nos devolverá solamente "Mundo" puesto que empieza a contar desde la sexta posición, es decir la letra "M", (el espacio entre palabras también cuenta como parte de la cadena) y como extraemos 5 caracteres que serían "Hola" + el espacio entre palabras nos quedaría solo "Mundo".

- **UNPIVOT**

Esta función se usa para convertir columnas en filas, es útil cuando tenemos muchos datos en una tabla vertical y nos puede resultar más cómodo pasarla a un formato más horizontal.

Si hacemos una tabla con ventas trimestrales de un portátil:

```
CREATE TABLE ventas
(producto VARCHAR(50),
  Q1 INT,
  Q2 INT,
  Q3 INT,
  Q4 INT
);
INSERT INTO ventas VALUES ('Portátil', 100, 200, 300, 400);
```

PRODUCTO	Columna 1	Columna 2	Columna 3	Columna 4
Portátil	100	200	300	400

Si queremos convertir las columnas **Q1, Q2, Q3, Q4** en filas, usamos **UNPIVOT**:

```
SELECT producto, trimestre, ventas
FROM (SELECT producto, Q1, Q2, Q3, Q4 FROM ventas)
UNPIVOT (ventas FOR trimestre IN (Q1, Q2, Q3, Q4)
) AS unpvt;
```

Resultado:

PRODUCTO	TRIMESTRE	VENTAS
Portátil	Columna 1	100
Portátil	Columna 2	200
Portátil	Columna 3	300
Portátil	Columna 4	400

Convierte los valores de las columnas **Q1, Q2, Q3, Q4** en una sola columna llamada **TRIMESTRE**. Los valores de las ventas se mantienen en la columna **ventas**.

- **RANK**

Esta función sirve para asignar un número de ranking o posición a cada fila en función de un criterio de orden.

También se usa la cláusula OVER() para definir cómo se debe ordenar el ranking.

Ejemplo:

Aquí tenemos una tabla de empleados con sus salarios:

```
CREATE TABLE empleados (  
    nombre VARCHAR(50),  
    salario INT  
);
```

```
INSERT INTO empleados VALUES  
( 'Javier', 1800),  
( 'José', 1800),  
( 'Luis', 1500),  
( 'Pablo', 1320),  
( 'David', 1240);
```

Se puede ordenar por salario y ponerle un ranking o clasificación:

```
SELECT nombre, salario, RANK() OVER (ORDER BY salario DESC) AS ranking  
FROM empleados;
```

Resultado:

NOMBRE	SALARIO	RANKING
Javier	1800	1
José	1800	1
Luis	1500	3
Pablo	1320	4
David	1240	5

Explicación:

- *Javier y José tienen el mismo salario, por lo que ambos reciben el ranking 1.*
- *El siguiente salario más alto es el de Luis con 1500, pero como hay un empate en el primer lugar, el siguiente ranking disponible es 3 (se salta el 2).*
- *Luego iría Pablo, con 1320 en el cuarto lugar y David con 1240 en el quinto lugar.*

Si quieres un ranking sin saltos en los números, usamos DENSE_RANK() en lugar de RANK().

