

# Clausula ORDER BY en PostgreSQL

Los resultados devueltos por una consulta con `SELECT` generalmente no guardan un orden específico. La clausula `ORDER BY` ofrece la posibilidad de ordenar los resultados de una consulta siguiendo unos criterios definidos.

La sintaxis de la clausula `ORDER BY` es como sigue:

```
SELECT
    lista_de_columnas
FROM
    tabla
ORDER BY
    expresión_1 [ASC | DESC] [NULLS FIRST | NULLS LAST],
    ...
    expresión_n [ASC | DESC] [NULLS FIRST | NULLS LAST]
```

Como puedes ver, para cada expresión de ordenación, se puede definir el orden, `ASC` para ascendente y `DESC` para descendente. Si se omite el orden, entonces se usará `ASC` por defecto. Cada una de las expresiones de ordenación debe separarse de una coma (,), pues se trata de una lista.

Las opciones `[NULLS FIRST | NULLS LAST]` hacen referencia al modo en que serán tratados los valores nulos.

A continuación se muestran algunos ejemplos de uso de la clausula `ORDER BY`.

## Ordenar una consulta por una columna

Uno de los ejemplos mínimos del uso de la clausula `ORDER BY` es la ordenación de los resultados por una sola columna. Veamos el siguiente ejemplo:

```
SELECT
    ship_name,
    ship_address,
    ship_city
```

```

FROM
    orders
ORDER BY
    ship_name ASC;

```

El resultado devuelto por esta consulta esta ordenado por `ship_name`:

	<b>ship_name</b> character varying (40) 🔒	<b>ship_address</b> character varying (60) 🔒	<b>ship_city</b> character varying (15) 🔒
1	Alfred's Futterkiste	Obere Str. 57	Berlin
2	Alfred's Futterkiste	Obere Str. 57	Berlin
3	Alfred's Futterkiste	Obere Str. 57	Berlin
4	Alfred's Futterkiste	Obere Str. 57	Berlin
5	Alfred's Futterkiste	Obere Str. 57	Berlin
6	Alfreds Futterkiste	Obere Str. 57	Berlin
7	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
8	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
9	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
10	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
11	Antonio Moreno Taquería	Mataderos 2312	México D.F.
12	Antonio Moreno Taquería	Mataderos 2312	México D.F.
13	Antonio Moreno Taquería	Mataderos 2312	México D.F.
14	Antonio Moreno Taquería	Mataderos 2312	México D.F.

La consulta a continuación es análoga a la anterior y produce el mismo resultado:

```

SELECT
    ship_name,
    ship_address,
    ship_city
FROM
    orders
ORDER BY
    ship_name;

```

Recordemos que si omitimos los complementos `ASC` y `DESC` en la clausula `ORDER BY` por defecto se usará `ASC` por defecto.

Del mismo modo, si definimos el ordenamiento en `DESC`, los resultados se ordenarán en sentido descendente:

```

SELECT
    ship_name,
    ship_address,
    ship_city
FROM
    orders
ORDER BY
    ship_name DESC;

```

La consulta anterior produce el siguiente resultado:

	<b>ship_name</b> character varying (40) 🔒	<b>ship_address</b> character varying (60) 🔒	<b>ship_city</b> character varying (15) 🔒
1	Wolski Zajazd	ul. Filtrowa 68	Warszawa
2	Wolski Zajazd	ul. Filtrowa 68	Warszawa
3	Wolski Zajazd	ul. Filtrowa 68	Warszawa
4	Wolski Zajazd	ul. Filtrowa 68	Warszawa
5	Wolski Zajazd	ul. Filtrowa 68	Warszawa
6	Wolski Zajazd	ul. Filtrowa 68	Warszawa
7	Wolski Zajazd	ul. Filtrowa 68	Warszawa
8	Wilman Kala	Keskuskatu 45	Helsinki
9	Wilman Kala	Keskuskatu 45	Helsinki
10	Wilman Kala	Keskuskatu 45	Helsinki
11	Wilman Kala	Keskuskatu 45	Helsinki
12	Wilman Kala	Keskuskatu 45	Helsinki
13	Wilman Kala	Keskuskatu 45	Helsinki
14	Wilman Kala	Keskuskatu 45	Helsinki

## Ordenar una consulta por múltiples columnas

La cláusula `ORDER BY` puede ordenar los resultados por múltiples columnas. Para ello la cláusula `ORDER BY` acepta una lista de columnas para ordenar. Veamos un ejemplo:

```

SELECT
    ship_name,
    ship_address,
    ship_city,
    shipped_date

```

```
FROM
    orders
ORDER BY
    ship_name DESC,
    shipped_date ASC;
```

Cuyo resultado es como sigue:

	ship_name character varying (40) 🔒	ship_address character varying (60) 🔒	ship_city character varying (15) 🔒	shipped_date date 🔒
1	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1996-12-09
2	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1997-08-01
3	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1997-12-31
4	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1998-02-13
5	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1998-03-03
6	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1998-04-17
7	Wolski Zajazd	ul. Filtrowa 68	Warszawa	1998-05-01
8	Wilman Kala	Keskuskatu 45	Helsinki	1997-08-06
9	Wilman Kala	Keskuskatu 45	Helsinki	1997-09-19
10	Wilman Kala	Keskuskatu 45	Helsinki	1997-10-14
11	Wilman Kala	Keskuskatu 45	Helsinki	1998-02-09
12	Wilman Kala	Keskuskatu 45	Helsinki	1998-02-12
13	Wilman Kala	Keskuskatu 45	Helsinki	1998-03-04
14	Wilman Kala	Keskuskatu 45	Helsinki	1998-04-10

En este ejemplo, hemos ordenado, en primer lugar todos los resultados por la columna `ship_name` en orden descendente. A continuación, todos los resultados que tienen el mismo valor, son ordenados a su vez por la columna `shipped_date` en orden ascendente.

Esto significa que la cláusula `ORDER BY`, realiza la ordenación por grupos de igual valor en la columnas, según el orden en el que aparecen en la lista de izquierda a derecha.

## Ordenar una consulta usando expresiones

La cláusula `ORDER BY` puede también ordenar el resultado de una expresión. Un ejemplo puede ser el uso de la función `LENGTH()` y ordenar por la longitud de la cadena:

```
SELECT
    ship_name,
    ship_address,
    ship_city,
    LENGTH(ship_name) longitud
FROM
```

```
orders
ORDER BY
    longitud ASC;
```

El resultado obtenido es el siguiente:

	ship_name character varying (40) 🔒	ship_address character varying (60) 🔒	ship_city character varying (15) 🔒	longitud integer 🔒
1	Bon app'	12, rue des Bouchers	Marseille	8
2	Bon app'	12, rue des Bouchers	Marseille	8
3	Bon app'	12, rue des Bouchers	Marseille	8
4	Bon app'	12, rue des Bouchers	Marseille	8
5	Bon app'	12, rue des Bouchers	Marseille	8
6	Bon app'	12, rue des Bouchers	Marseille	8
7	Bon app'	12, rue des Bouchers	Marseille	8
8	Bon app'	12, rue des Bouchers	Marseille	8
9	Bon app'	12, rue des Bouchers	Marseille	8
10	Bon app'	12, rue des Bouchers	Marseille	8
11	Bon app'	12, rue des Bouchers	Marseille	8
12	Bon app'	12, rue des Bouchers	Marseille	8
13	Bon app'	12, rue des Bouchers	Marseille	8
14	Bon app'	12, rue des Bouchers	Marseille	8

## Ordenación de columnas con valores nulos

Dentro de PostgreSQL un valor nulo representa el estado de una celda cuyo valor no ha sido definido o es desconocido en un momento dado. La opción de tratamiento de valores nulos permite definir el modo en el que queremos que los valores nulos sean tratados. Tenemos dos opciones, `NULLS FIRST` ordenará los resultados con valores nulos primero y `NULLS LAST` los posicionará al final de los resultados. En el caso que la opción de nulos se omita, por defecto se usa `NULLS FIRST`.

Vamos a crear una tabla e insertar unos registros para entender mejor la ordenación de nulos:


```
CREATE TABLE demo_nulos (numero INT);

INSERT INTO demo_nulos(numero) VALUES (11),(22),(33),(44),(null);
```

Veamos el resultado de un primer ejemplo, en el que ordenamos los resultados de tal forma que los nulos sean posicionados al final:


```
SELECT numero FROM demo_nulos ORDER BY numero ASC NULLS LAST;
```

Cuyo resultado será:

	numero integer 
1	11
2	22
3	33
4	44
5	[null]

Ahora vamos a modificar la consulta para posicionar los nulos al inicio de los resultados:

```
SELECT numero FROM demo_nulos ORDER BY numero ASC NULLS FIRST;
```

	numero integer 
1	[null]
2	11
3	22
4	33
5	44

---

Revisión #1

Creado 19 diciembre 2023 13:06:04 por Guillermo

Actualizado 19 diciembre 2023 13:14:41 por Guillermo