



ORDER BY en MySQL: DESC y ASC

🕒 Updated October 24, 2021

Clasificación de resultados

Miramos cómo obtener datos de nuestras tablas usando el comando SELECCIONAR. Los resultados se devolvieron en el mismo orden en que se agregaron los registros a la base de datos. Este es el orden de clasificación predeterminado. En esta sección, veremos cómo podemos ordenar los resultados de nuestra consulta. **La clasificación es simplemente reorganizar los resultados de nuestra consulta de una manera específica**. La clasificación se puede realizar en una sola columna o en más de una columna. Se puede hacer en números, cadenas y tipos de datos de fecha.

Orden por cláusula

La cláusula order by se usa para ordenar los conjuntos de resultados de la consulta en orden ascendente o descendente. Se usa junto con la consulta SELECT. Tiene la siguiente sintaxis básica.

```
1 Instrucción SELECT ... [condición WHERE | GROUP BY `field_name` (s)` HAVING condition] ORDER
```

AQUÍ

- “Instrucción SELECT ...” es la consulta de selección regular
- “[” representa alternativas
- “[WHERE condition | GROUP BY `field_name (s)` HAVING condition” es la condición opcional utilizada para filtrar los conjuntos de resultados de la consulta.
- “ORDER BY” realiza la clasificación del conjunto de resultados de la consulta
- “[ASC | DESC]” es la palabra clave utilizada para ordenar los conjuntos de resultados en orden ascendente o descendente. Nota **ASC** se usa como el valor predeterminado.

¿Cuáles son las palabras clave DESC y ASC?



**ASC es la forma abreviada para****ascender**

Se utiliza para ordenar los resultados de la consulta en un estilo de arriba a abajo.

Al trabajar en tipos de datos de fecha, la fecha más temprana se muestra en la parte superior de la lista.

Al trabajar con tipos de datos numéricos, los valores más bajos se muestran en la parte superior de la lista.

Al trabajar con tipos de datos de cadena, el conjunto de resultados de la consulta se ordena desde aquellos que comienzan con la letra A hasta la letra Z.

**DESC es la forma abreviada para****descender**

Se usa para ordenar los resultados de la consulta en un estilo de abajo hacia arriba

. Al trabajar en tipos de fecha, la última fecha se muestra en la parte superior de la lista.

Al trabajar con tipos de datos numéricos, los valores más altos se muestran en la parte superior del conjunto de resultados de la consulta.

Cuando se trabaja con tipos de datos de cadena, el conjunto de resultados de la consulta se ordena de aquellos que comienzan con la letra Z bajando a la letra A.

Las palabras clave DESC y ASC se usan juntas junto con la instrucción SELECT y la cláusula ORDER BY.

Sintaxis DESC y ASC

La palabra clave de clasificación DESC tiene la siguiente sintaxis básica.

```
1  SELECCIONAR {fieldName (s) | *} FROM tableName (s) [WHERE condition] ORDER BY nombre (s) de
```

AQUÍ

- **SELECCIONAR {fieldName (s) | *} FROM tableName (s)** es la declaración que contiene los campos y tablas de los que se obtiene el conjunto de resultados.
- **[WHERE condition]** es opcional pero se puede usar para filtrar los datos de acuerdo con la condición dada.
- **ORDER BY fieldName (s)** es obligatorio y es el campo en el que se realizará la clasificación. La palabra clave DESC especifica que la clasificación debe ser en orden

descendente.

- **[LIMIT]** es opcional, pero se puede usar para limitar el número de resultados devueltos desde el conjunto de resultados de la consulta.

Ejemplos:

Veamos ahora un ejemplo práctico:

```
1 SELECT * FROM miembros;
```

Ejecutar el script anterior en MySQL workbench contra myflixdb nos da los siguientes resultados que se muestran a continuación.

número de socio	nombres completos	género	fecha de nacimiento	dirección física	direccion postal	contct_number	c
1	Janet Jones	Hembra	21-07-1980	Calle Primera Parcela No 4	Bolsa privada	0759 253 542	j
2	Janet Smith Jones	Hembra	23-06-1980	Melrose 123	NULO	NULO	jj
3	Robert Phil	Masculino	12-07-1989	3rd Street 34	NULO	12345	r
4	Gloria Williams	Hembra	14-02-1984	2nd Street 23	NULO	NULO	M
5	Leonard Hofstadter	Masculino	NULO	Woodcrest	NULO	845738767	M
6	Sheldon Cooper	Masculino	NULO	Woodcrest	NULO	976736763	M
7	Rajesh Koothrappali	Masculino	NULO	Woodcrest	NULO	938867763	M

8	Leslie Winkle	Masculino	14-02-1984	Woodcrest	NULO	987636553	N
9	Howard Wolowitz	Masculino	24-08-1981	Parque del Sur	PO Box 4563	987786553	

Supongamos que el departamento de marketing quiere que los detalles de los miembros se organicen en orden decreciente de Fecha de nacimiento. Esto les ayudará a enviar saludos de cumpleaños de manera oportuna. Podemos obtener dicha lista ejecutando una consulta como a continuación –

```
1 SELECCIONAR * DE miembros ORDEN POR date_of_birth DESC;
```

Ejecutar el script anterior en MySQL workbench contra myflixdb nos da los siguientes resultados que se muestran a continuación.

La misma consulta en orden ascendente

```
SELECCIONAR * FROM miembros PEDIR POR date_of_birth ASC
```

Nota: los valores NULL significan que no hay valores (no cero o cadena vacía). Observe la forma en que se han ordenado.

Más ejemplos

Consideremos la siguiente secuencia de comandos que enumera todos los registros de miembros.

1

SELECCIONAR * FROM `miembros`;

Ejecutar el script anterior proporciona los siguientes resultados que se muestran a continuación.

número de socio	nombres completos	género	fecha de nacimiento	dirección física	direccion postal	contct_number	c
1	Janet Jones	Hembra	21-07-1980	Calle Primera Parcela No 4	Bolsa privada	0759 253 542	j
2	Janet Smith Jones	Hembra	23-06-1980	Melrose 123	NULO	NULO	j
3	Robert Phil	Masculino	12-07-1989	3rd Street 34	NULO	12345	r
4	Gloria Williams	Hembra	14-02-1984	2nd Street 23	NULO	NULO	M
5	Leonard Hofstadter	Masculino	NULO	Woodcrest	NULO	845738767	M
6	Sheldon Cooper	Masculino	NULO	Woodcrest	NULO	976736763	M
7	Rajesh Koothrappali	Masculino	NULO	Woodcrest	NULO	938867763	M
8	Leslie Winkle	Masculino	14-02-1984	Woodcrest	NULO	987636553	M
9	Howard Wolowitz	Masculino	24-08-1981	Parque del Sur	PO Box 4563	987786553	M

Supongamos que queremos obtener una lista que ordene el conjunto de resultados de la consulta usando el campo de género, usaríamos la secuencia de comandos que se muestra a continuación.

```
1 SELECCIONE * FROM `members` ORDER BY `gender`;
```

número de socio	nombres completos	género	fecha de nacimiento	dirección física	direccion postal	contct_number	c
1	Janet Jones	Hembra	21-07-1980	Calle Primera Parcela No 4	Bolsa privada	0759 253 542	j
2	Janet Smith Jones	Hembra	23-06-1980	Melrose 123	NULO	NULO	j
4	Gloria Williams	Hembra	14-02-1984	2nd Street 23	NULO	NULO	M
3	Robert Phil	Masculino	12-07-1989	3rd Street 34	NULO	12345	r
5	Leonard Hofstadter	Masculino	NULO	Woodcrest	NULO	845738767	M
6	Sheldon Cooper	Masculino	NULO	Woodcrest	NULO	976736763	M
7	Rajesh Koothrappali	Masculino	NULO	Woodcrest	NULO	938867763	M
8	Leslie Winkle	Masculino	14-02-1984	Woodcrest	NULO	987636553	M
9	Howard Wolowitz	Masculino	24-08-1981	Parque del Sur	PO Box 4563	987786553	M

Los miembros “femeninos” se han mostrado primero seguidos por los miembros “masculinos”, esto es porque cuando se utiliza la cláusula order by sin especificar la palabra clave ASC o DESC, de forma predeterminada, MySQL ha ordenado el resultado de la consulta en orden ascendente.

Veamos ahora un ejemplo que hace la **clasificación usando dos columnas**; el primero está **ordenado** en **orden ascendente** por defecto mientras que la segunda columna está **ordenada** en **orden descendente**.

```
1 SELECCIONAR * FROM `miembros` PEDIR POR `gender`, `date_of_birth` DESC;
```

Ejecutar el script anterior en MySQL workbench contra myflixdb da los siguientes resultados.

La columna de género se ordenó en orden ascendente de forma predeterminada, mientras que la columna de fecha de nacimiento se ordenó en orden descendente explícitamente

¿Por qué podemos usar DESC y ASC?

Supongamos que queremos imprimir un historial de pagos para un miembro de la biblioteca de videos para ayudar a responder las consultas de la recepción, ¿no sería más lógico que los pagos se impriman en orden cronológico descendente comenzando con el pago reciente al pago anterior?

La palabra clave DESC es útil en tales situaciones. Podemos escribir una consulta que ordene la lista en orden descendente utilizando la fecha de pago.

Supongamos que el departamento de marketing desea obtener una lista de películas por categoría que los miembros pueden usar para decidir qué películas están disponibles en la biblioteca cuando alquilan películas. ¿No sería más lógico buscar los nombres y títulos de

las películas en orden ascendente para que los miembros pueden buscar rápidamente la información de la lista?

La palabra clave ASC es útil en tales situaciones; podemos obtener la lista de películas clasificada por nombre de categoría y título de la película en orden ascendente.

Resumen

- La clasificación de los resultados de la consulta está reorganizando las filas devueltas desde un conjunto de resultados de la consulta en orden ascendente o descendente.
- La palabra clave DESC se usa para ordenar el conjunto de resultados de la consulta en un orden descendente.
- La palabra clave ASC se usa para ordenar el conjunto de resultados de la consulta en orden ascendente.
- Tanto DESC como ASC funcionan junto con la palabra clave ORDER BY. También se pueden usar en combinación con otras palabras clave como la cláusula WHERE y LIMIT
- El valor predeterminado para ORDER BY cuando nada se ha especificado explícitamente es ASC.

[< Prev](#)[Next >](#)

Leave a Reply

*Your email address will not be published. Required fields are marked **

Comment *

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment

© Copyright - Guru99
2021

[Casa](#) [Java](#) [SQL](#) [Node.js](#) [AngularJS](#)

[TensorFlow](#) [Contacto](#)