



≡ Menu

# APRENDE SQL EN 2 HORAS

Cortesía de FuturisticLab



# Aprenda a realizar consultas en SQL en 2 horas. 1ra parte.

October 6, 2019 by user

## Primera Hora:

(SQL )Structured Query Language o en español lenguaje de consulta estructurada.

Es un lenguaje por medio del cual se extrae y analiza información de bases de datos relacionales.

Las bases de datos relacionales a su vez son bases de datos que almacenan información en una estructura de tabla (similar a una tabla de Excel por ejemplo). Las tablas se componen de columnas, que van a ser atributos o propiedades de la información a ser guardada y en filas como componente de información.

Las bases de datos se llaman relacionales porque además de guardar la información en tablas pueden generar relaciones y cruces de información entre ellas. Estas relaciones se pueden dar por medio de consultas o por medio de enlazamiento por llaves primarias y foráneas. Esto por el momento es un poco abstracto pero a medida que vayamos tratando el tema se ira aclarando.

## ¿Por qué es importante aprender SQL?

Más del 90% de las bases de datos incluso muchas NoSQL utilizan un formato de consultas muy similar a este. Luego toda persona interesada en el análisis de datos debe conocer al menos un poco de este lenguaje.

## ¿Por qué no solamente me quedo con Excel y ya?

Excel e incluso Power Query para manejar archivos de más de 2GB no es una herramienta apropiada. Por lo que necesario almacenar nuestros datos en plataformas que después nos permitan consumirlos mediante lenguajes de consulta como SQL.

## Consultas de información en una tabla única:

Vamos a aprovechar la base de datos que nos proporciona W3Schools para realizar nuestras primeras consultas para tal fin, vayamos a la siguiente dirección: [https://www.w3schools.com/sql/trysql.asp?filename=trysql\\_select\\_all](https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all) adicionalmente en : <https://www.w3schools.com/sql> Encontrara un manual detallado.

Allí tenemos una base de datos con 8 tablas.

### Your Database:

Tablename	Records
<a href="#">Customers</a>	91
<a href="#">Categories</a>	8
<a href="#">Employees</a>	10
<a href="#">OrderDetails</a>	518
<a href="#">Orders</a>	196
<a href="#">Products</a>	77
<a href="#">Shippers</a>	3
<a href="#">Suppliers</a>	29

Restore Database

Tablas de la base de datos con la que vamos a trabajar

### SELECT, FROM WHERE y ORDER BY.

Las consultas en SQL tienen la siguiente estructura (la información entre paréntesis es información propia de la consulta)

*SELECT (nombres de las columnas que aparecerán en la consulta al ejecutarse)*  
*FROM (Nombre de la tabla que quiero consultar) WHERE (Filtro que le deseo*

*aplicar a una o más columnas ) ORDER BY (Columna por la cual quiero que se ordenen los datos en la visualización)*

Vamos a ver algunos ejemplos

Para ver todos los registros y columnas de una tabla por ejemplo utilizara la consulta:

```
SELECT * FROM Customers;
```

\* este símbolo de asterisco en SQL hace referencia a todas las columnas y todos los registros de la tabla Customers.

#### SQL Statement:

```
SELECT * FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

#### Result:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Bergsgatan 2	Luleå	S-971 08	Sweden

Consulta SQL que retorna todas las filas y columnas de una Tabla

Supongamos que ahora quiero solo las columnas CustomerID, CustomerName y Country. Entonces la consulta que tengo que ejecutar es:

```
SELECT CustomerID, CustomerName , Country FROM Customers;
```

Así obtenemos

### SQL Statement:

```
SELECT CustomerID, CustomerName, Country FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

### Result:

Number of Records: 91

CustomerID	CustomerName	Country
1	Alfreds Futterkiste	Germany
2	Ana Trujillo Emparedados y helados	Mexico

**Ejercicio:** Para practicar realice una consulta en la tabla Employees, en las que únicamente me retorne las columnas LastName y BirthDay.

Para filtrar datos utilizamos la palabra WHERE por ejemplo quiero todos los clientes cuya ciudad es Berlin, entonces:

```
SELECT * FROM Customers WHERE City ="Berlin";
```

```
SELECT * FROM Customers WHERE City ="Berlin";
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 1

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany

Cuando se filtra con WHERE se ponen comillas si el valor en la columna es texto ,si el valor es número es sin comillas. Por ejemplo en la tabla ordenes filtremos todas las ordenes realizadas con el empleado con id igual a 4 , y mostrando en la salida de la consulta unicamente los campos EmployeeID y OrderID luciría de la siguiente forma.

```
SELECT EmployeeID,OrderID FROM Orders WHERE EmployeeID =4;
```

Tambien se pueden aplicar dos filtros en simultanea por ejemplo que la orden sea despachada oir el empleado con id 4 y que sea de la fecha 1996-07-08. La consulta se vería así:

```
SELECT * FROM Orders WHERE EmployeeID =4 AND OrderDate ="1996-07-08";
```

En el caso de **AND** se cumplen las dos condiciones, si solo necesitamos que se cumpla una de ellas utilizamos la palabra **OR**.

**Ejercicio:** De la tabla Suppliers obtenga todos los proveedores que son de Estados Unidos, Japón o Brasil.

Si queremos que los datos de la consulta (SELECT CustomerID, CustomerName , Country FROM Customers) nos los retorne en orden alfabético la consulta del nombre se modificaría de la siguiente forma:

```
SELECT CustomerID, CustomerName , Country FROM Customers ORDER BY CustomerName;
```

Si queremos que el orden sea inverso (De Z a A) utilizamos el DESC al final:

```
SELECT CustomerID, CustomerName , Country FROM Customers ORDER BY CustomerName DESC;
```

```
SELECT CustomerID, CustomerName , Country FROM Customers ORDER BY CustomerName DESC;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 91

CustomerID	CustomerName	Country
91	Wolski	Poland
90	Wilman Kala	Finland
89	White Clover Markets	USA

**Ejercicio:** Vaya a la tabla products y ordene la tabla con precios del más pequeño al más grande, en caso de empate en el precio , ordene por descendentemente por nombre.

**CONTAR, SUMAR, SACAR PROMEDIOS , MÁXIMOS Y MÍNIMOS CON SQL y (GROUP BY Y HAVING)**

Ahora puede que quiera saber cuántos clientes tenemos en un país en este caso necesitaríamos **la función contar COUNT aplicada a una columna y con un filtro de país**. La función COUNT lo que va hacer es contar cada registro de nombre del cliente que cumpla una condición. Ejemplo:

```
SELECT COUNT(CustomerName) FROM Customers WHERE Country="USA";
```

El resultado de esta consulta es : 13

Ahora queremos una consulta que nos arroje el nombre del país y en otra columna el número de clientes existentes en cada país una consulta como esas se vería de la forma. En este caso la condición bajo la cual se produciría el conteo de los elementos depende de GROUP BY , en el siguiente query se contarán teniendo en cuenta si el nombre de Country es el mismo.

```
SELECT Country, COUNT(CustomerName) FROM Customers GROUP BY  
Country;
```

En caso de que necesitemos que nos ordene esta información de forma que los países con más clientes queden en la parte superior de la consulta modificamos la misma de la siguiente forma:



**SQL Statement:**

```
SELECT Country, COUNT(CustomerName) FROM Customers GROUP BY Country ORDER BY COUNT(CustomerName) DESC ;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

**Result:**

Number of Records: 21

Country	COUNT(CustomerName)
USA	13
France	11
Germany	11
Brazil	9

Es decir hemos prácticamente creado una columna nueva a partir de **COUNT(CustomerName)** si queremos asignarle un nombre por cuestiones de organización de la consulta lo podemos hacer con la palabra **AS**, quedando la consulta de la siguiente forma:

```
SELECT Country, COUNT(CustomerName) AS TotalClientes FROM Customers  
GROUP BY Country ORDER BY TotalClientes DESC ;
```

Ejecuten esa consulta y verán como el nombre de la columna en el resultado cambia. Esto es necesario ya que hace más legible la consulta (en consultas largas en especial).

Modifiquemos la consulta anterior para que solo me muestre los países con 7 clientes o menos .

```
SELECT Country, COUNT(CustomerName) AS TotalClientes FROM Customers  
GROUP BY Country HAVING TotalClientes <= 7 ORDER BY TotalClientes DESC  
;
```

**SQL Statement:**

```
SELECT Country, COUNT(CustomerName) AS TotalClientes FROM Customers GROUP BY Country HAVING  
TotalClientes <= 7 ORDER BY TotalClientes DESC ; |
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

**Result:**

Number of Records: 17

Country	TotalClientes
UK	7
Mexico	5

*En este caso cuando utilizamos GROUP BY en vez de utilizar WHERE se utiliza HAVING después del GROUP BY.*

**Ejercicio:** En la tabla OrdenDetails haga una consulta que en la primera columna muestre el ProductID y en la segunda columna sea la suma de todas las cantidades que se han ordenado para cada producto, dejando solo aquellos productos que en total tienen más de 50 unidades, esa segunda columna se debe llamar Total, también deben estar ordenados de mayor cantidad a menor. TIP: En vez de utilizar COUNT utilice SUM.

Obtener el producto con el menor precio

La consulta seria con MIN y seria del tipo :

```
SELECT ProductName, MIN(Price) FROM Products ;
```

**Ejercicio:** en la tabla OrderDetail encuentre la orden con el mayor número de unidades. Tip: MAX.

Si les gusto y desean aprender más los espero para la segunda hora de SQL en la [segunda parte](#).

📁 Inteligencia de Negocios, IT

🔖 sql

- ◀ Aplicación de análisis de sentimientos a encuestas de satisfacción
- ▶ Aprenda a realizar consultas en SQL en 2 horas. 2da Parte

## Leave a Comment

Post Comment

## Recent Posts

[Materiales Curso de Power BI](#)

[Aprenda a realizar consultas en SQL en 2 horas. 2da Parte](#)

[Aprenda a realizar consultas en SQL en 2 horas. 1ra parte.](#)

[Aplicación de análisis de sentimientos a encuestas de satisfacción](#)

[Análisis histórico de eventos de minas antipersonas y desminado utilizando POWER BI](#)

## Recent Comments

[erotik izle](#) on [Materiales Curso de Power BI](#)

[Edward](#) on [Tendencias para INNOVAR en la industria de BEBIDAS en 2017 y 2018](#)

[Antonella - clima laboral](#) on [2 CASOS EXITOSOS DE ESTRATEGIAS DE FELICIDAD CORPORATIVA.](#)

## Archives

[February 2020](#)

[October 2019](#)

[March 2019](#)

[November 2018](#)

[June 2018](#)

[May 2018](#)

[March 2018](#)

[February 2018](#)

[January 2018](#)

December 2017  
November 2017  
October 2017  
August 2017  
July 2017  
June 2017  
May 2017  
March 2017  
February 2017  
January 2017  
November 2016  
August 2016  
July 2016  
June 2016  
May 2016  
April 2016  
February 2016  
January 2016

## Categories

Creatividad  
Cultura Organizacional  
Estrategia  
Innovación  
Inteligencia de Negocios  
IT  
Líderes  
Prospectiva  
Tendencias

[Uncategorized](#)

**Meta**

[Log in](#)

[Entries feed](#)

[Comments feed](#)

[WordPress.org](#)

## Estrategias innovadoras

Genere innovación y diferenciación tanto en el negocio tradicional de su compañía, como para nuevas unidades de negocios.

© 2023 FuturisticLab • Powered by GeneratePress