In this lesson we are looking into SELECT statement in SQL. The Select statement lets us retrieve data from our database and the have a look at it, potentially export it and then use it in our application. It is probably the most popular SQL statement.

The SQL command is simple. It has two parts. It goes like,

SELECT ColumnName FROM TableName;

The only table that we can select from are those tables inside the database that we are using. The only databases we can access with the query are only those databases that we are connected to. All these databases contain tables, and, in these tables, data are entered in each of the columns and rows. Each of the columns has their names on their heading and we use Select statement to select the columns and access the data stored in them.

For example, we have a customers’ table inside our database that contains all the customers’ information. If we want to select all the columns from that table then we write our query as below,

*SELECT \* FROM customers;*

|  |  |  |
| --- | --- | --- |
| first\_name  character varying (100) | id  integer | last\_name  character varying (255) |
| Rolf | 1 | Smith |
| Jose | 2 | Salvatierre |
| Anne | 3 | Watson |
| Craig | 4 | Scott |
| Michael | 5 | Adam |

We see here are all the data those are in the customers’ table. We have a first\_name column, an id column and then a last\_name column. As we get the names of the column, we also get the datatype of the data each column is carrying. We only got two different datatypes in this table, the first\_name and last\_name are both the type called *character varying* and the Id is of type *integer.*

A *character* datatype means letters, numbers, symbols or anything that we can store as a string. The word *varying* with the number *100* means we can have up to 100 characters. If we do not have varying at the end of our column name, then it means all the values in the column should be of 100 characters length and if they do not then empty spaces will be added to make it the length of 100 characters. The id column is of integer, normally if we have an id column, we use the datatype of integers because SQL has a way of automatically incrementing the number as we increase rows by adding new values.

Let’s say we want to select just the first\_name column, we can also do that,

*SELECT customers.first\_name FROM customers;*

|  |
| --- |
| first\_name character varying (100) |
| Rolf |
| Jose |
| Anne |
| Craig |
| Michael |

However, we are not limited to only selecting one column or all columns, we can also select two columns or any number of columns we want.

We can do that by using a comma after the first column name and then adding the second column name that we are looking to select.

*SELECT customers.first\_name, customers.last\_name FROM customers;*

|  |  |
| --- | --- |
| first\_name character varying (100) | last\_name character varying (100) |
| Rolf | Smith |
| Jose | Salvatierre |
| Anne | Watson |
| Craig | Scott |
| Michael | Adam |

We also can customize the column names while displaying them in the table at our output. To customize a column name and set it as the name we want we use command called *Alias.*

We use the letters *AS* for *Alias* command and write our query as below,

Let’s say we want to rename the first\_name column as First Name in our output, then we would code as below,

*SELECT customers.first\_name AS “First name”, customers.last\_name FROM customers;*

|  |  |
| --- | --- |
| First name | last\_name |
| Rolf | Smith |
| Jose | Salvatierra |
| Anne | Watson |
| Craig | Scott |
| Michael | Adam |

*SELECT customers.first\_name AS “First name”, customers.last\_name AS “Last name” FROM customers;*

|  |  |
| --- | --- |
| First name | Last name |
| Rolf | Smith |
| Jose | Salvatierra |
| Anne | Watson |
| Craig | Scott |
| Michael | Adam |