In this lesson we are looking at the UPDATE command. The UPDATE command lets us change data that is already in our database. For this lesson we will be using the items’ table.

Let’s see what data is already present in our table.

*SELECT \* FROM items;*

|  |  |  |
| --- | --- | --- |
| name  character varying (255) | id  integer | price  numeric (10,2) |
| Fountain Pen | 2 | 11.30 |
| Laptop | 4 | 899.00 |
| Screen | 5 | 275.50 |
| Hard Drive | 6 | 89.99 |
| Pen | 1 | 4.00 |
| Ink | 3 | 3.50 |

*Numeric* is the datatype that lets us have decimal numbers in our column. It contains two number in brackets. The first number represents the maximum number of digits, in this case it is 10, meaning the value of *price* can be maximum of 10 digits. The second number represents the number of digits after the decimal point, in this case it is 2.

This is all the data we have got and now we want to change the price of Ink from 3.50 to 4.00. We can do that using the UPDATE command. To change the price of Ink from 3.50 to 4.00 we need to code as below.

*UPDATE items SET price=4.00 WHERE id=3;*

*Query returned successfully: one row affected, 22 msec execution time.*

Postgres tells us that it has successfully executed the command and now we need to check our table to see the update.

*SELECT \* FROM items;*

|  |  |  |
| --- | --- | --- |
| name  character varying (255) | id  integer | price  numeric (10,2) |
| Fountain Pen | 2 | 11.30 |
| Laptop | 4 | 899.00 |
| Screen | 5 | 275.50 |
| Hard Drive | 6 | 89.99 |
| Pen | 1 | 4.00 |
| Ink | 3 | 4.00 |

Now, we can see that our Ink is 4.00 instead of 3.50.

In the above query we have updated the price of Ink from 3.50 to 4.00 using the id = 3 in our WHERE clause, but we can also update prices of our items using other columns than the id if we want.

Let’s update the price of items using the price column this time.

*UPDATE items SET price=5.00 WHERE price=4.00;*

*Query returned successfully: 2 items affected, 18 msec execution time.*

Postgres tells us that our query has been executed successfully and now we need to check our items’ table to see the update.

*SELECT \* FROM items;*

|  |  |  |
| --- | --- | --- |
| name  character varying (255) | id  integer | price  numeric (10,2) |
| Fountain Pen | 2 | 11.30 |
| Laptop | 4 | 899.00 |
| Screen | 5 | 275.50 |
| Hard Drive | 6 | 89.99 |
| Pen | 1 | 5.00 |
| Ink | 3 | 5.00 |

If we UPDATE an item but forget to put the WHERE clause in our query, then all the items in our table will get updated. That is why we should always be careful and include the WHERE clause while using the UPDATE command in our query.

*UPDATE items SET price=10.00;*

*Query returned successfully: 5 items affected, 54 msec execution time.*

Postgres lets us know that our query is successful and now we need to check the table to see the update.

|  |  |  |
| --- | --- | --- |
| name  character varying (255) | id  integer | price  numeric (10,2) |
| Fountain Pen | 2 | 10.00 |
| Laptop | 4 | 10.00 |
| Screen | 5 | 10.00 |
| Hard Drive | 6 | 10.00 |
| Pen | 1 | 10.00 |
| Ink | 3 | 10.00 |