UPDATE

The UPDATE command helps us to change or modify the records that have already been inserted into a table. You can combine it with the WHERE clause to specify the record that is to be updated. Here is the syntax:

```
UPDATE tableName SET field=newValue, field2=newValue2, ...
[WHERE ...]
```

The UPDATE command can also be combined with clauses such as SET, WHERE, LIMIT, and ORDER BY. You will see this shortly:

Consider the table named Price with the following records:

```
<table>
<thead>
<tr>
<th>id</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
</tr>
<tr>
<td>3</td>
<td>220</td>
</tr>
<tr>
<td>4</td>
<td>190</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
</tr>
</tbody>
</table>
```

Let's change the price of the book with an id of 1 from 200 to 250:

```
UPDATE price
SET price = 250
WHERE id = 1;
```

The command ran successfully. You can now query the table to see whether the change took place:
The above screenshot shows that the change has been implemented. Consider the table Book with the following records:

Let us change the name of the book named Book to MariaDB Book1. Notice that the book has an id of 1. Here is the command for this:

```sql
UPDATE book
SET name = "MariaDB Book1"
WHERE id = 1;
```

Check whether the change has been implemented:
The above screenshot shows that the change has been implemented successfully.

In the above examples, we have only changed one column at a time. However, it is possible for us to change multiple columns at a go. Let us demonstrate this using an example.

Let us use the Price table with the following data:

Let us change both the id and the price of the book with an id of 5. We will change its id to 6 and price to 6. Run the following command:

```
UPDATE price
SET id = 6,
price = 280
WHERE id = 5;
```

Now, query the table to check whether the change was made successfully:

```
SELECT * FROM price;
```

The change was made successfully.
Assignment

1. How to update records in table?