# 3.4 ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype
```

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name
```

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype
```

If we use CREATE TABLE and the table already exists in the table we will get an error message, so if we combine CREATE TABLE and ALTER TABLE we can create robust database scripts that gives no errors, as the example shown below:

```
if not exists (select * from dbo.sysobjects where id = object_id(N'[CUSTOMER]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
CREATE TABLE CUSTOMER
(
        CustomerId int PRIMARY KEY,
        CustomerNumber int NOT NULL UNIQUE,
        LastName varchar(50) NOT NULL,
        FirstName varchar(50) NOT NULL,
        AreaCode int NULL,
        Address varchar(50) NULL,
        Phone varchar(50) NULL,
)
GO


if exists(select * from dbo.syscolumns where id = object_id(N'[CUSTOMER]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1 and name = 'CustomerId')
ALTER TABLE CUSTOMER ALTER COLUMN CustomerId int
Else
ALTER TABLE CUSTOMER ADD CustomerId int
GO


if exists(select * from dbo.syscolumns where id = object_id(N'[CUSTOMER]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1 and name = 'CustomerNumber')
ALTER TABLE CUSTOMER ALTER COLUMN CustomerNumber int
```

```
Else
ALTER TABLE CUSTOMER ADD CustomerNumber int
GO

...
```

# 4INSERT INTO

The INSERT INTO statement is used to insert a new row in a table.

It is possible to write the INSERT INTO statement in two forms.

The first form doesn't specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name
VALUES (value1, value2, value3,...)
```

Example:

```
INSERT INTO CUSTOMER VALUES ('1000', 'Smith', 'John', 12,
'California', '11111111')
```

The second form specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

This form is recommended!

Example:

```
INSERT INTO CUSTOMER (CustomerNumber, LastName, FirstName, AreaCode,
Address, Phone)
VALUES ('1000', 'Smith', 'John', 12, 'California', '11111111')
```

## Insert Data Only in Specified Columns:

It is also possible to only add data in specific columns.

Example:

```
INSERT INTO CUSTOMER (CustomerNumber, LastName, FirstName)
VALUES ('1000', 'Smith', 'John')
```

**Note!** You need at least to include all columns that cannot be NULL.

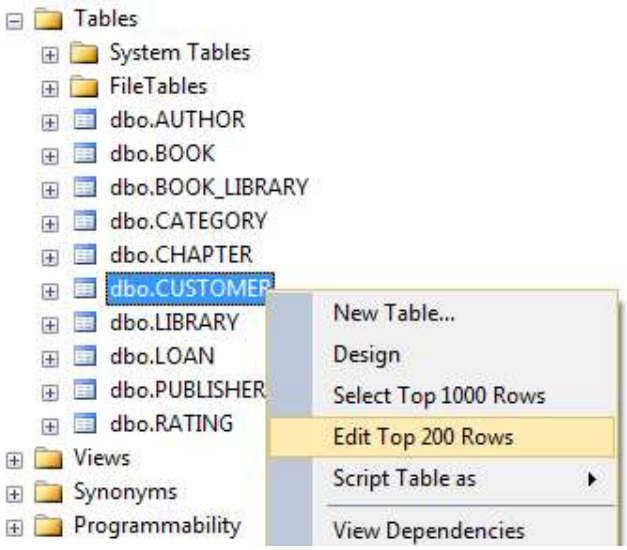
We remember the table definition for the CUSTOMER table:

| Column Name | Data Type | Allow Nulls |
|---|---|---|
| CustomerId | int | ☐ |
| CustomerNumber | int | ☐ |
| LastName | varchar(50) | ☐ |
| FirstName | varchar(50) | ☐ |
| AreaCode | int | ☑ |
| Address | varchar(50) | ☑ |
| Phone | varchar(20) | ☑ |
| | | ☐ |

i.e., we need to include at least “CustomerNumber”, “LastName” and “FirstName”. “CustomerId” is set to “identity(1,1)” and therefore values for this column are generated by the system.

**Insert Data in the Designer Tools:**

When you have created the tables, you can easily insert data into them using the designer tools. Right-click on the specific table and select “Edit Top 200 Rows”:




Then you can enter data in a table format, similar to, e.g., MS Excel:

PC88235\DEVELOP...- dbo.CUSTOMER | Object Explorer Details

| CustomerId | CustomerName | CustomerNu... | Address | Phone | PostCode | PostAddress | EMail | Country |
|---|---|---|---|---|---|---|---|---|
| 1 | Bill Clinton | 1000 | NULL | NULL | NULL | NULL | NULL | NULL |
| 2 | Jens Stoltenberg | 1001 | NULL | NULL | NULL | NULL | NULL | NULL |
| 3 | Barak Obama | 1002 | NULL | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

# 5 UPDATE

The UPDATE statement is used to update existing records in a table.

The syntax is as follows:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

**Note!** Notice the WHERE clause in the UPDATE syntax. The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated!

Example:

```
update CUSTOMER set AreaCode=46 where CustomerId=2
```

Before update:

| | CustomerId | CustomerNumber | LastName | FirstName | AreaCode | Address | Phone |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | Smith | John | 12 | California | 11111111 |
| 2 | 2 | 1001 | Jackson | Smith | 45 | London | 22222222 |
| 3 | 3 | 1002 | Johnsen | John | 32 | London | 33333333 |

After update:

| | CustomerId | CustomerNumber | LastName | FirstName | AreaCode | Address | Phone |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | Smith | John | 12 | California | 11111111 |
| 2 | 2 | 1001 | Jackson | Smith | 46 | London | 22222222 |
| 3 | 3 | 1002 | Johnsen | John | 32 | London | 33333333 |

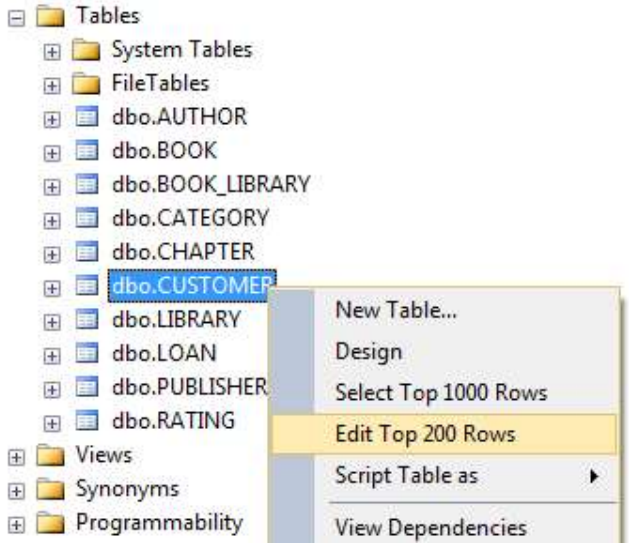
If you don't include the WHERE clause the result becomes:

| | CustomerId | CustomerNumber | LastName | FirstName | AreaCode | Address | Phone |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | Smith | John | 46 | California | 11111111 |
| 2 | 2 | 1001 | Jackson | Smith | 46 | London | 22222222 |
| 3 | 3 | 1002 | Johnsen | John | 46 | London | 33333333 |

→ So make sure to include the WHERE clause when using the UPDATE command!

**Update Data in the Designer Tools:**

The same way you insert data you can also update the data. Right-click on the specific table and select “Edit Top 200 Rows”:




Then you can change your data:

PC88235\DEVELOP...- dbo.CUSTOMER | Object Explorer Details

| CustomerId | CustomerName | CustomerNu... | Address | Phone | PostCode | PostAddress | EMail | Country |
|---|---|---|---|---|---|---|---|---|
| 1 | Bill Clinton | 1000 | NULL | NULL | NULL | NULL | NULL | NULL |
| 2 | Jens Stoltenberg | 1001 | NULL | NULL | NULL | NULL | NULL | NULL |
| 3 | Barak Obama | 1002 | NULL | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

# 6 DELETE

The DELETE statement is used to delete rows in a table.

Syntax:

```
DELETE FROM table_name
WHERE some_column=some_value
```

**Note!** Notice the WHERE clause in the DELETE syntax. The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

Example:

```
delete from CUSTOMER where CustomerId=2
```

Before delete:

| | CustomerId | CustomerNumber | LastName | FirstName | AreaCode | Address | Phone |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | Smith | John | 12 | California | 11111111 |
| 2 | 2 | 1001 | Jackson | Smith | 45 | London | 22222222 |
| 3 | 3 | 1002 | Johnsen | John | 32 | London | 33333333 |

After delete:

| | CustomerId | CustomerNumber | LastName | FirstName | AreaCode | Address | Phone |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | Smith | John | 12 | California | 11111111 |
| 2 | 3 | 1002 | Johnsen | John | 32 | London | 33333333 |

## Delete All Rows:

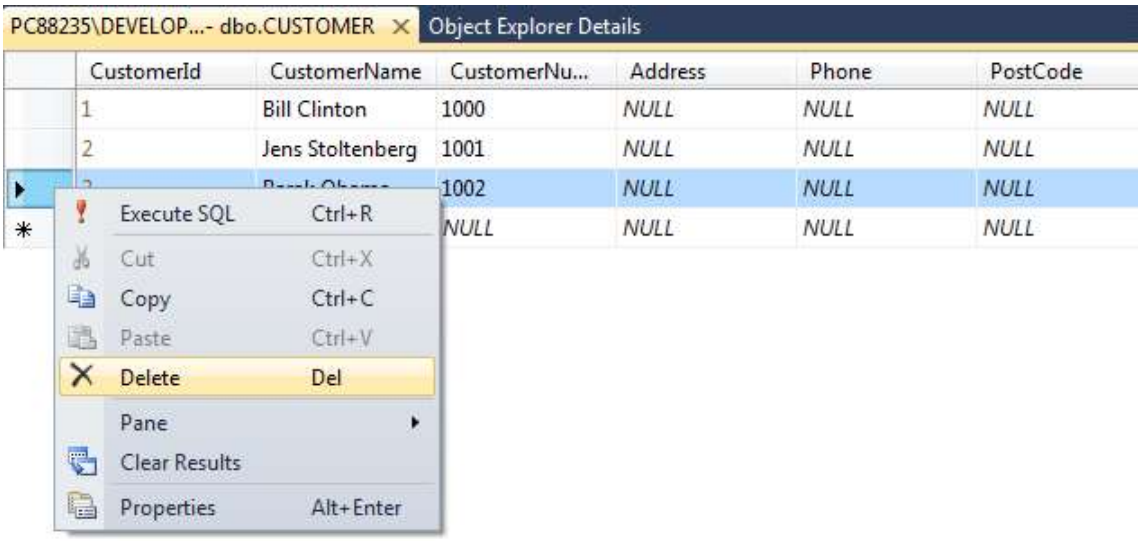
It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact:

```
DELETE FROM table_name
```

**Note!** Make sure to do this only when you really mean it! You cannot UNDO this statement!

### Delete Data in the Designer Tools:

You delete data in the designer by right-click on the row and select “Delete”:


PC88235\DEVELOP...- dbo.CUSTOMER
Object Explorer Details
CustomerId
CustomerName
CustomerNu...
Address
Phone
PostCode
1
Bill Clinton
1000
NULL
NULL
NULL
2
Jens Stoltenberg
1001
NULL
NULL
NULL
1002
NULL
NULL
NULL
NULL
NULL
NULL
NULL
Execute SQL Ctrl+R
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Pane
Clear Results
Properties Alt+Enter