LAB - Alter Table - Alter Column

In this lab, you will learn how to use the SQL Server `ALTER TABLE ALTER COLUMN` statement to modify a column of a table.

SQL Server allows you to perform the following changes to an existing column of a table:

- Modify the data type
- Change the size
- Add a `NOT NULL` constraint

Modify column's data type

To modify the data type of a column, you use the following statement:

```
ALTER TABLE table_name
ALTER COLUMN column_name new_data_type(size);
```

The new data type must be compatible with the old one, otherwise, you will get a conversion error in case the column has data and it fails to convert.

See the following example.

First, create a new table with one column whose data type is `INT`:

```
CREATE TABLE t1 (c INT);
```

Second, insert some rows into the table:

```
    INSERT INTO t1
    VALUES
        (1),
        (2),
        (3);
```

Second, modify the data type of the column from `INT` to `VARCHAR`:

```
ALTER TABLE t1 ALTER COLUMN c VARCHAR (2);
```

Third, insert a new row with a character string data:

```
INSERT INTO t1 VALUES ('@');
```

Fourth, modify the data type of the column from `VARCHAR` back to `INT`:

```
ALTER TABLE t1 ALTER COLUMN c INT;
```

SQL Server issues the following error:

```
Conversion failed when converting the varchar value '@' to data type int.
```

Change the size of a column

The following statement creates a new table with one column whose data type is `VARCHAR(10)`:

```
CREATE TABLE t2 (c VARCHAR(10));
```

Let's insert some sample data into the t2 table:

```
INSERT INTO t2
VALUES
    ('SQL Server'),
    ('Modify'),
    ('Column');
```

You can increase the size of the column as follows:

```
ALTER TABLE t2 ALTER COLUMN c VARCHAR (50);
```

However, when you decrease the size of the column, SQL Server checks the existing data to see if it can convert data based on the new size. If the conversion fails, SQL Server terminates the statement and issues an error message.

For example, if you decrease the size of column `c` to 5 characters:

```
ALTER TABLE t2 ALTER COLUMN c VARCHAR (5);
```

SQL Server issued the following error:

```
String or binary data would be truncated.
```

Add a `NOT NULL` constraint to a nullable column

The following statement creates a new table with a nullable column:

```
CREATE TABLE t3 (c VARCHAR(50));
```

The following statement inserts some rows into the table:

```
INSERT INTO t3
VALUES
    ('Nullable column'),
    (NULL);
```

If you want to add the `NOT NULL` constraint to the column `c`, you must update NULL to non-null first for example:

```
UPDATE t3
SET c = ''
WHERE
    c IS NULL;
```

And then add the `NOT NULL` constraint:

```
ALTER TABLE t3 ALTER COLUMN c VARCHAR (20) NOT NULL;
```

In this lab, you have learned how to use the SQL Server `ALTER TABLE ALTER COLUMN` to modify some properties of an existing column.