

# Chapter 19: UPDATE

## Section 19.1: UPDATE with data from another table

The examples below fill in a `PhoneNumber` for any `Employee` who is also a `Customer` and currently does not have a phone number set in the `Employees` Table.

(These examples use the `Employees` and `Customers` tables from the Example Databases.)

### Standard SQL

Update using a correlated subquery:

```
UPDATE
  Employees
SET PhoneNumber =
  (SELECT
    c.PhoneNumber
  FROM
    Customers c
  WHERE
    c.FName = Employees.FName
    AND c.LName = Employees.LName)
WHERE Employees.PhoneNumber IS NULL
```

### SQL:2003

Update using `MERGE`:

```
MERGE INTO
  Employees e
USING
  Customers c
ON
  e.FName = c.Fname
  AND e.LName = c.LName
  AND e.PhoneNumber IS NULL
WHEN MATCHED THEN
  UPDATE
  SET PhoneNumber = c.PhoneNumber
```

### SQL Server

Update using `INNER JOIN`:

```
UPDATE
  Employees
SET
  PhoneNumber = c.PhoneNumber
FROM
  Employees e
INNER JOIN Customers c
  ON e.FName = c.FName
  AND e.LName = c.LName
WHERE
  PhoneNumber IS NULL
```

## Section 19.2: Modifying existing values

This example uses the Cars Table from the Example Databases.

```
UPDATE Cars
SET TotalCost = TotalCost + 100
WHERE Id = 3 or Id = 4
```

Update operations can include current values in the updated row. In this simple example the TotalCost is incremented by 100 for two rows:

- The TotalCost of Car #3 is increased from 100 to 200
- The TotalCost of Car #4 is increased from 1254 to 1354

A column's new value may be derived from its previous value or from any other column's value in the same table or a joined table.

## Section 19.3: Updating Specified Rows

This example uses the Cars Table from the Example Databases.

```
UPDATE
  Cars
SET
  Status = 'READY'
WHERE
  Id = 4
```

This statement will set the status of the row of 'Cars' with id 4 to "READY".

**WHERE** clause contains a logical expression which is evaluated for each row. If a row fulfills the criteria, its value is updated. Otherwise, a row remains unchanged.

## Section 19.4: Updating All Rows

This example uses the Cars Table from the Example Databases.

```
UPDATE Cars
SET Status = 'READY'
```

This statement will set the 'status' column of all rows of the 'Cars' table to "READY" because it does not have a **WHERE** clause to filter the set of rows.

## Section 19.5: Capturing Updated records

Sometimes one wants to capture the records that have just been updated.

```
CREATE TABLE #TempUpdated(ID INT)

Update TableName SET Col1 = 42
  OUTPUT inserted.ID INTO #TempUpdated
WHERE Id > 50
```