MERGE

Performs insert, update, delete operations on a target table based on the results of a join with a source table. For example, two tables can be synchronized by inserting, updating, deleting rows in one table based on differences found in the other table.
The conditional behavior described for the MERGE statement works best when the two tables have a complex mixture of matching characteristics. For example, inserting a row if it does not exist, or updating the row if it does match.

The MERGE statement is used to perform the following operations:

- Conditionally insert or update rows in a target table.
  If the row exists in the target table, update one or more columns; otherwise, insert the data into a new row.
- Synchronize two tables.
- Insert, update, or delete rows in a target table based on differences with the source data.

The MERGE syntax contain five primary clauses:

- The MERGE clause specifies the table or view that is the target of the insert, update, or delete operations.
- The USING clause specifies the data source being joined with the target.
- The ON clause specifies the join conditions that determine where the target and source match.
- The WHEN clauses (WHEN MATCHED, WHEN NOT MATCHED BY TARGET, and WHEN NOT MATCHED BY SOURCE) specify the actions to take based on the results of the ON clause and any additional search criteria specified in the WHEN clauses.
- The OUTPUT clause returns a row for each row in the target that is inserted, updated, or deleted.

```
--Create a target table
CREATE TABLE Products (ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Rate MONEY)
GO
--Insert records into target table
INSERT INTO Products VALUES (1, 'Tea', 10.00), (2, 'Coffee', 20.00), (3, 'Muffin', 30.00), (4,
'Biscuit', 40.00)
GO
--Create source table
CREATE TABLE UpdatedProducts (ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Rate MONEY)
GO
--Insert records into source table
INSERT INTO UpdatedProducts VALUES (1, 'Tea', 10.00), (2, 'Coffee', 25.00), (3, 'Muffin', 35.00), (5,
'Pizza', 60.00)
GO
SELECT * FROM Products
SELECT * FROM UpdatedProducts
```

Results | Messages

	ProductID	ProductName	Rate
1	1	Tea	10.00
2	2	Coffee	20.00
3	3	Muffin	30.00
4	4	Biscuit	40.00

	ProductID	ProductName	Rate
1	1	Tea	10.00
2	2	Coffee	25.00
3	3	Muffin	35.00
4	5	Pizza	60.00

```sql
--Synchronize the target table with refreshed data from source table
MERGE Products AS TARGET
USING UpdatedProducts AS SOURCE
ON (TARGET.ProductID = SOURCE.ProductID)
--When records are matched, update the records if there is any change
WHEN MATCHED AND TARGET.ProductName <> SOURCE.ProductName
OR TARGET.Rate <> SOURCE.Rate THEN
UPDATE SET TARGET.ProductName = SOURCE.ProductName,
TARGET.Rate = SOURCE.Rate
--When no records are matched, insert the incoming records from source table to target table
WHEN NOT MATCHED BY TARGET THEN
INSERT (ProductID, ProductName, Rate) VALUES (SOURCE.ProductID, SOURCE.ProductName, SOURCE.Rate)
--When there is a row that exists in target table and same record does not exist in source table
--then delete this record from target table
WHEN NOT MATCHED BY SOURCE THEN
DELETE
--$action specifies a column of type nvarchar(10) in the OUTPUT clause that returns one of three
--values for each row: 'INSERT', 'UPDATE', or 'DELETE', according to the action that was performed on
that row
OUTPUT $action,
DELETED.ProductID AS TargetProductID,
DELETED.ProductName AS TargetProductName,
DELETED.Rate AS TargetRate,
INSERTED.ProductID AS SourceProductID,
INSERTED.ProductName AS SourceProductName,
INSERTED.Rate AS SourceRate;
SELECT @@ROWCOUNT;
GO
```

Results | Messages

	$action	TargetProductID	TargetProductName	TargetRate	SourceProductID	SourceProductName	SourceRate
1	UPDATE	2	Coffee	20.00	2	Coffee	25.00
2	UPDATE	3	Muffin	30.00	3	Muffin	35.00
3	DELETE	4	Biscuit	40.00	NULL	NULL	NULL
4	INSERT	NULL	NULL	NULL	5	Pizza	60.00

	(No column name)
1	4

```sql
SELECT * FROM Products
SELECT * FROM UpdatedProducts
```

Now it is:

Results | Messages

	ProductID	ProductName	Rate
1	1	Tea	10.00
2	2	Coffee	25.00
3	3	Muffin	35.00
4	5	Pizza	60.00

	ProductID	ProductName	Rate
1	1	Tea	10.00
2	2	Coffee	25.00
3	3	Muffin	35.00
4	5	Pizza	60.00

Before it was:

Results | Messages

	ProductID	ProductName	Rate
1	1	Tea	10.00
2	2	Coffee	20.00
3	3	Muffin	30.00
4	4	Biscuit	40.00

	ProductID	ProductName	Rate
1	1	Tea	10.00
2	2	Coffee	25.00
3	3	Muffin	35.00
4	5	Pizza	60.00

References:

https://docs.microsoft.com/en-us/sql/t-sql/statements/merge-transact-sql?view=sql-server-2017

https://www.mssqltips.com/sqlservertip/1704/using-merge-in-sql-server-to-insert-update-and-delete-at-the-same-time/