

Using Advanced T-SQL Techniques



Xavier Morera

HELPING DEVELOPERS UNDERSTAND SEARCH & BIG DATA

@xmorera www.xaviermorera.com



Going Deeper into T-SQL DML Statements

INSERT

UPDATE

DELETE

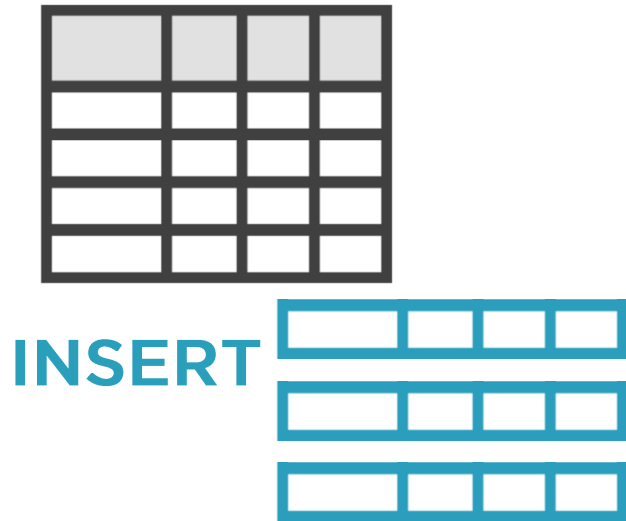
Add Data

Modify Data

Remove Data



Using INSERT with Multiple Values



INSERT

- Provided [columns] and values
- For one record

INSERT multiple records

Provide multiple values

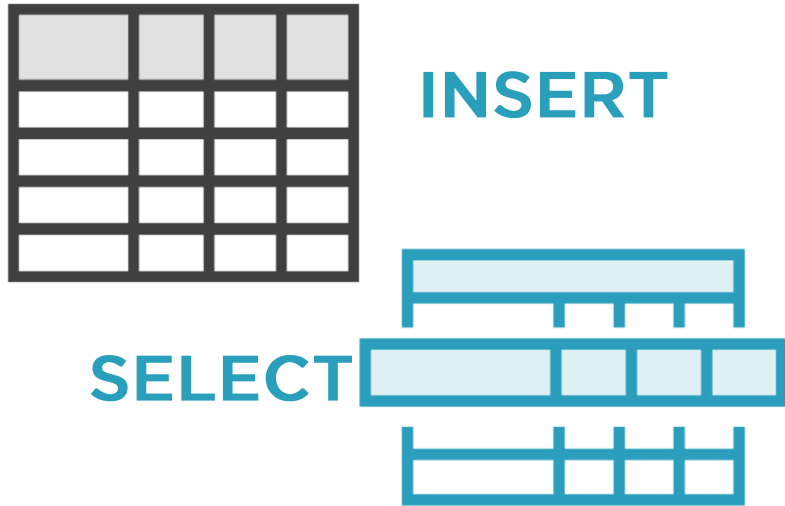
Demo



Using INSERT with Multiple Values



SELECT within an INSERT Statement



Typically we provide values to INSERT

- Know upfront the values
- Sometimes not feasible or practical

Possible to use a SELECT statement

- To retrieve values and use them in the statement
- For example to populate tables

Also works with UPDATE and DELETE

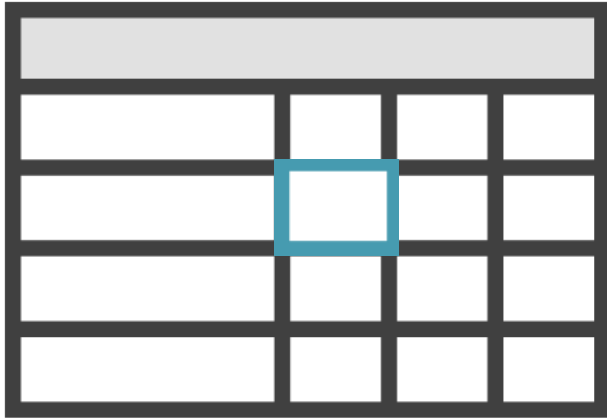
Demo



SELECT within an INSERT Statement



SELECT to Retrieve a Field in an INSERT



INSERT multiple fields

- You can provide all field values
- Or retrieve one field based on another

Can use a SELECT statement

- For a single field
- Advantage: Less error prone

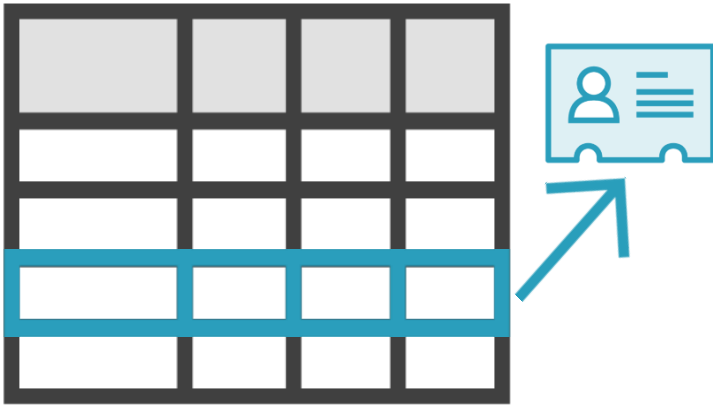
Demo



Using **SELECT** to Retrieve a Specific Field
in an **INSERT**



Retrieving Records on INSERT with OUTPUT



Not all values are known on INSERT

- Auto assigned, calculated, and defaults
- We may need to know them on execution

Use the OUTPUT clause

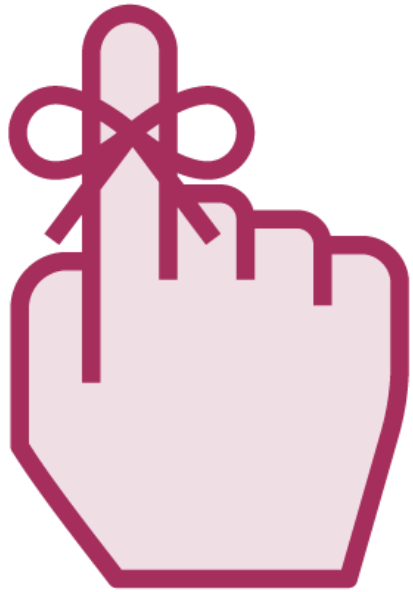
- To retrieve these values

Demo



Retrieving Records on INSERT
Using the OUTPUT Clause





Also works with UPDATE

OUTPUT



Demo



Retrieving Modified Records on UPDATE
Using the OUTPUT Clause





And with DELETE

OUTPUT



Demo

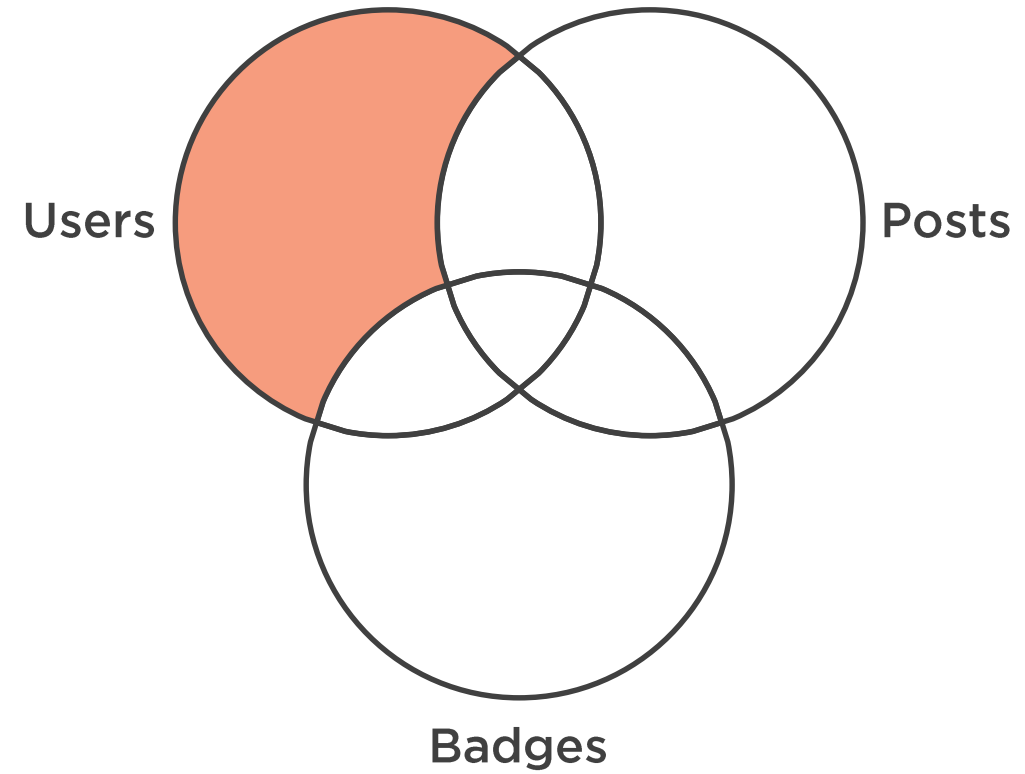


Retrieving Removed Records on DELETE
Using the OUTPUT Clause

Including JOIN and TOP



Users that do not
have posts or badges



MERGE

Used to synchronize tables

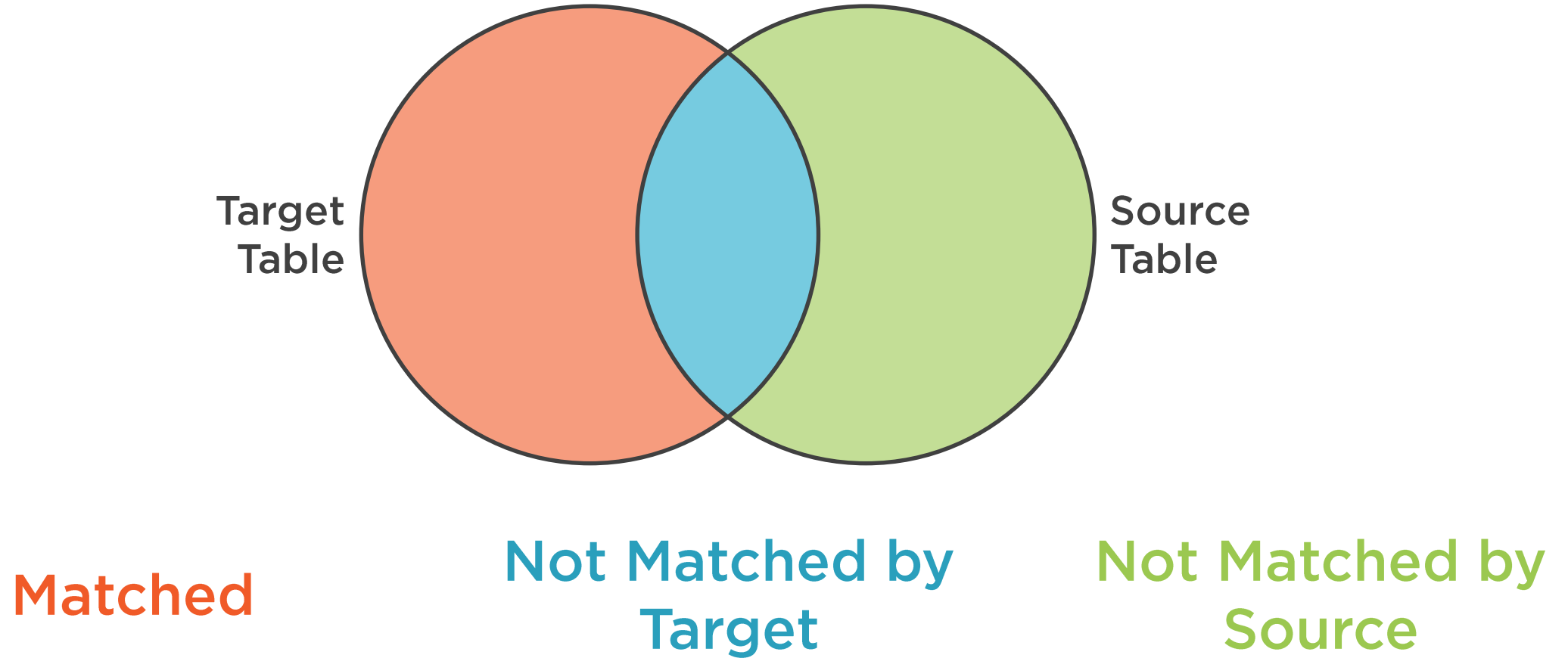
- Mixture of matching characteristics

Runs insert, update, and delete operations

- On a target table
- Based on differences found in a source table



MERGE



MERGE

```
MERGE <targetable>  
USING <sourceTable>  
ON <mergeCondition>  
WHEN MATCHED  
    THEN <updateStatement>  
WHEN NOT MATCHED BY TARGET  
    THEN <insertStatement>  
WHEN NOT MATCHED BY SOURCE  
    THEN <deleteStatement>
```



Demo



Synchronizing Tables Using the MERGE Statement



BULK INSERT Using the bcp Utility



Copying data in large quantities

Utility to import and export data

- Installed by default with SQL Server

Bulk Copy Program

- bcp

Database ↔ Text files

Format file to specify how to import data

Demo



BULK INSERT using a Format File
with `bcp`



BULK INSERT: Using a format file

Version	14.0									
Number of columns	6	Host file data type		Prefix length	Host file data length		Terminator	Server column order	Server column name	Column collation
		1	SQLCHAR	0	12	"\$"	1	Id		" "
Host file field order	2	SQLCHAR	0	255	"\$"	6	UserDisplayName		SQL_Latin1_General_CP1_CI_AS	
	3	SQLCHAR	0	12	"\$"	7	UserId		" "	
	4	SQLCHAR	0	24	"\$"	5	CreationDate		" "	
	5	SQLCHAR	0	0	"\$"	4	Text		SQL_Latin1_General_CP1_CI_AS	
	6	SQLCHAR	0	12	"\r\n"	2	PostId		" "	



Takeaway



INSERT one record at a time

- Or many, providing multiple values

Populate one table based on a query

- SELECT statement

Retrieve specific fields for INSERT

- Using a query
- Also works for DELETE & UPDATE

Takeaway



Retrieve records affected

- OUTPUT clause
- INSERT and UPDATE
 - INSERTED.<column or columns>
- DELETE
 - DELETED.<column or columns>

Takeaway



MERGE clause

- Synchronize tables
 - INSERT, UPDATE, & DELETE
- Scenarios
 - Matched
 - Not matched by target
 - Not matched by source

BULK INSERT

- Using a format file
- bcp