Systems Design and Databases (CIS1018-N)

Week 9

Using DML to modify Data

Teaching Team

Module Leader & Lecturer: Dr Yar Muhammad
Email: Yar.Muhammad@tees.ac.uk
Office: G0.39 (Greig Building)

Tutor:

- Dr Mengda He
- Mr Mansha Nawaz
- Mr Vishalkumar Thakor

Academic Hub Time Slots, Room IT1.13:
Yar Muhammad
Monday 10:00 - 11:00 and Tuesday 13:00 - 14:00

Mengda He
Wednesdays 1-2 pm and Fridays 11 am - 12 pm

- See Blackboard Ultra for online materials: https://bb.tees.ac.uk/

Lectures & IT Labs

Lectures – Dr Yar Muhammad	Tuesdays @ 2-3 pm	Thursdays @ 1-2 pm
Week 1 – Week 12	CL1.87	

Tutor – Thursday	IT Lab Session Room #: IT2.42
Mr Mansha Nawaz M.Nawaz@tees.ac.uk	Time: 3 – 5 pm

Tutor – Friday	IT Lab Session Room #: OL3
Dr Yar Muhammad Yar.Muhammad@tees.ac.uk	Time: 9 – 11 am & 11 am – 1 pm
Dr Mengda He M.He@tees.ac.uk	Time: 9 – 11 am
Mr Vishalkumar Thakor V.Thakor@tees.ac.uk	Time: 11 am – 1 pm & 1 – 3 pm
Mr Mansha Nawaz M.Nawaz@tees.ac.uk	Time: 1 – 3 pm

Schedule 1/3

Systems Design and Databases CIS1018-N
Weekly Plan for the Activities

Week	Lecturer	Lecture Demo	Lab Exercises & Solutions	ICA Tasks:
01	• Module Introduction, • System Design, • Introduction Databases (DDL, DML, DCL, TCL)	• Requirement List & • MoSCoW Wireframe Design & Templates, • User Stories	• Team Setup, • Hands-on to collect/pick the Requirements from MoSCoW and write Writing User stories on each • **Tutorial 1**	Requirements List & MosCOW, User stories
02	• UML and UML Tool,	• Use Case Diagrams from Requirements List and Wireframe	• Hands-on Use Case Diagrams Activities • **Tutorial 2**	Each Wireframe has associated Use Case Activity Deadline for Team Setup is Week # 2, by Friday 07/10/2022 before 4pm
03	• Sequence Diagrams	• Class Diagrams	• Hands-on Sequence & Class Diagrams Activities • **Tutorial 3**	Each Wireframe has associated Sequence and Class Diagrams
04	• Entity Relationship Diagrams (ERD) A Data Modelling Case Tool for Relational Databases	• Introduction to SQL Server • Walk-through: SQL Quick Guide 1 - How to use SSMS to build Databases	• **Tutorial 4** • Lab Resources: SQL Quick Guide 1	Each Wireframe has associated Class Diagram

Systems Design - UML

Analysis

Design

Schedule 2/3

Databases - TSQL

Week	Lecturer	Lecture Demo	Lab Exercises & Solutions	ICA Tasks:
05	• Querying with Select	**Demo A** – Writing Simple SELECT Statements **Demo B/C** – Eliminating Duplicates with DISTINCT **Demo D** - Writing Simple CASE	• TSQL-Mod03 Lab-Exercise 1-4 • **Tutorial 5**	SQL Task A: TSQL03 Querying with Select • Writing Simple SELECT Statements • Eliminating Duplicates with DISTINCT • Using Column and Table Aliases • Writing Simple CASE Expressions
06	• Querying with Multiple Tables	**Demo B** – Relating 2 or more tables – Joins & Joining multiple tables – inner, outer and cross.	• TSQL-Mod04 Exercise 1-5 • **Tutorial 6**	SQL Task B: TSQL04 – Querying with Multiple Tables • Relating 2 or more tables – Joins • Joining multiple tables – inner, outer and cross.
07	• Sorting and Filtering Data	**Demo A** – Sort with ORDER BY **Demo B** – Filter with WHERE Clause **Demo C** – Filtering with Top OffsetFetch **Demo D** – Handling NULL	• TSQL-Mod05 Exercise 1 – 4 • **Tutorial 7**	SQL Task C: TSQL05 – Sort and Filtering Data • Sort with Order By • Filter with Where By • Filter with top offsetfetch • Handling Nulls
Submission ICA 1 (Group Submission) -> Deadline is Wednesday 16/11/2022 before 4pm				
08	• Working with SQL Server Data	**Demo A** - Conversion in a Query **Demo B** - collation in a query **Demo C** - date and time functions	• TSQL-Mod06 Exercise 1 – 4 • **Tutorial 8**	SQL Task D: TSQL06 – Working with SQL Server Data • Conversion in a Query • collation in a query • date and time functions

Schedule 3/3

Databases - TSQL

09	• Using DML to modify Data	**Demo A** - Adding Data to Tables **Demo B** - Modifying and Removing Data **Demo C** - Generating Automatic Column Values	• TSQL-Mod07 Exercise 1 – 2 • **Tutorial 9**	SQL Task E: TSQL07– Using DML to Modify Data • Adding Data to Tables • Modifying and Removing Data • Generating Automatic Column Values
10	• Using built in Functions	**Demo A** – Scalar Functions **Demo B** – Cast Functions Demo C – If Functions **Demo D** – IsNull Functions	• TSQL-Mod08 Exercise 1 – 3 • **Tutorial 10**	SQL Task F: TSQL08– Using Built-In Functions • Writing Queries with Built-In Functions • Using Conversion Functions • Using Logical Functions • Using Functions to Work with NULL
11	• Walk through SQL Quick Guide 2 - Create a Tables and Relationships via SSMS GUI	• Walk through: • SQL Quick Guide 3 - Create Query, View through Designer	Hands-on: • SQL Server Quick Guide 2	SQL Server – Introduction to SQL Server and SSMS
12	Support	Support	Hands-on: • SQL Server Quick Guide 3	SQL Server – Introduction to SQL Server and SSMS
Submission ICA 2 (Individual Submission) -> Deadline is Wednesday 11/01/2023 before 4pm				

Module Overview

- Adding Data to Tables
- Modifying and Removing Data
- Generating Automatic Column Values

Adding Data to Tables

- Using INSERT to Add Data
- Using INSERT with Data Providers
- Using SELECT INTO
- Demonstration: Adding Data to Tables

Using INSERT to Add Data

The INSERT … VALUES statement inserts a new row

```
INSERT INTO Sales.OrderDetails
       (orderid, productid, unitprice, qty, discount)
VALUES (10255,39,18,2,0.05);
```

Table and row constructors add multirow capability to INSERT … VALUES

```
INSERT INTO Sales.OrderDetails
(orderid, productid, unitprice, qty, discount)

VALUES
(10256,39,18,2,0.05),
(10258,39,18,5,0.10);
```

Using INSERT with Data Providers

- INSERT … SELECT to insert rows from another table:

```
INSERT Sales.OrderDetails
(orderid, productid, unitprice, qty, discount)

SELECT * FROM NewOrderDetails
```

- INSERT … EXEC is used to insert the result of a stored procedure or dynamic SQL expression into an existing table:

```
INSERT INTO Production.Products
(productID, productname, supplierid, categoryid, unitprice)
EXEC Production.AddNewProducts;
```

Using SELECT INTO

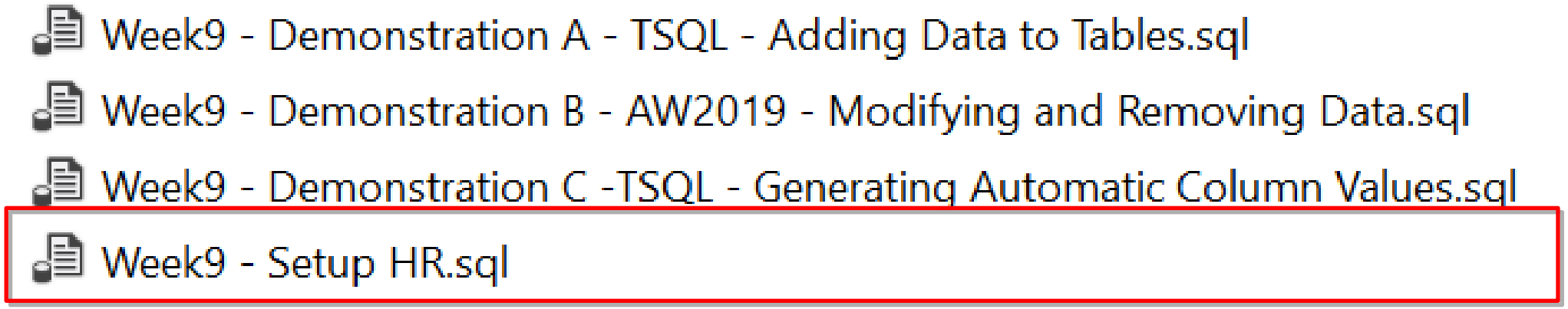
SELECT -> INTO is similar to INSERT <- SELECT

- It also creates a table for the output, fashioned on the output itself

- The new table is based on query column structure
  - Uses column names, data types, and null settings
  - Does not copy constraints or indexes

```
SELECT * INTO NewProducts FROM PRODUCTION.PRODUCTS
WHERE ProductID >= 70
```

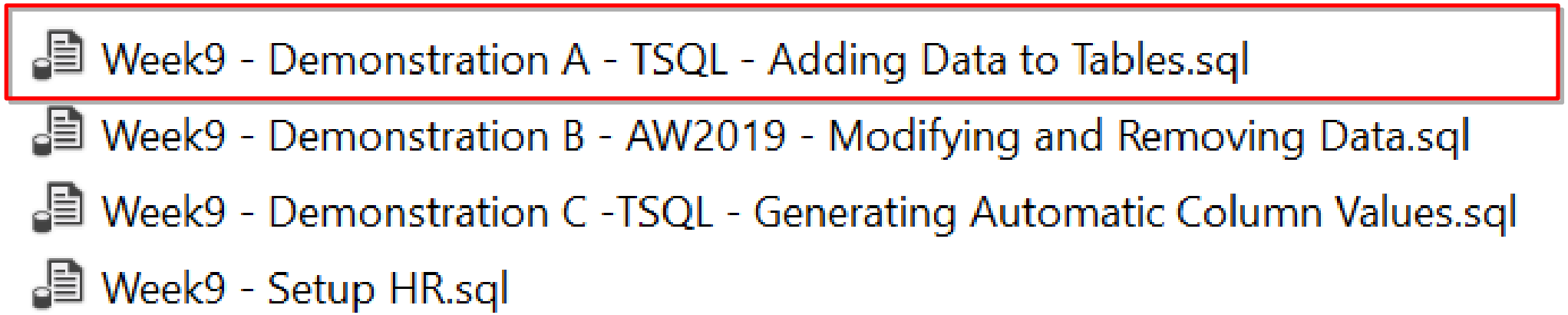
Demonstration: Setup HR



Demonstration A: Adding Data to Tables

In this demonstration, you will see how to:

- Add data to a table using the INSERT statement
- Use the OUTPUT keyword with INSERT
- Use stored procedure output to insert data into a table
- Use SELECT INTO for populating a table with data and create the table structure at the same time



Modifying and Removing Data

- Using UPDATE to Modify Data
- Using MERGE to Modify Data
- Demonstration: Manipulating Data Using the UPDATE and DELETE Statements and MERGING Data Using Conditional DML

Using UPDATE to Modify Data

- UPDATE changes all rows in a table or view
- Unless rows are filtered with a WHERE clause or constrained with a JOIN clause
- Column values are changed with the SET clause

```
UPDATE Production.Products
   SET   unitprice = (unitprice * 1.04)
WHERE   categoryid =  1 AND discontinued = 0
;
```

```
UPDATE Production.Products
   SET     unitprice *= 1.04
                     -- Using compound
                     -- assignment operators
WHERE   categoryid =  1 AND discontinued = 0;
```

Using MERGE to Modify Data

MERGE modifies data based on a condition

- When the source matches the target
- When the source has no match in the target
- When the target has no match in the source

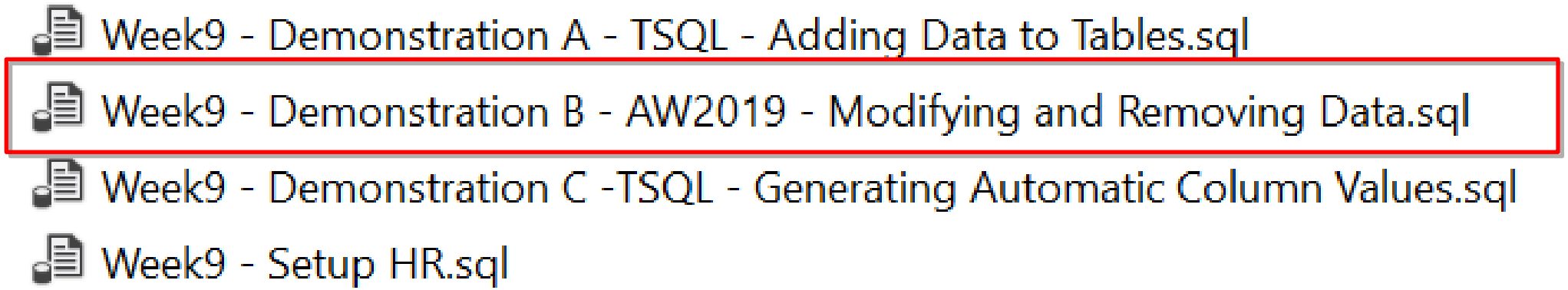
```
MERGE TOP (10)
INTO   Store         AS Destination
USING  StoreBackup   AS StagingTable
       ON(Destination.Key = StagingTable.Key)

WHEN NOT MATCHED THEN
       INSERT (C1,..)
       VALUES (Source.C1,..)
WHEN MATCHED THEN
       UPDATE SET Destination.C1 = StagingTable.C1,..;
```

Demonstration B: Manipulating Data Using the UPDATE and DELETE Statements and MERGING Data Using Conditional DML

In this demonstration, you will see how to:

- UPDATE row, column intersections within tables
- DELETE complete rows from within tables
- Apply multiple data manipulation language (DML) operations by using the MERGE statement
- Understand how to use the OUTPUT clause to monitor data changes during DML operations
- Understand how to access prior and current data elements, in addition to showing the DML operation performed



Generating Automatic Column Values

- Using IDENTITY
- Using Sequences

Using IDENTITY

The IDENTITY property generates column values automatically

- Optional seed and increment values can be provided

```
CREATE TABLE Production.Products
(PID int IDENTITY(1,1) NOT NULL, Name VARCHAR(15),…)
```

- Only one column in a table may have IDENTITY defined
- IDENTITY column must be omitted in a normal INSERT statement

```
INSERT INTO Production.Products (Name,…)
VALUES ('MOC 2072 Manual',…)
```

- Functions are provided to return last generated values
  - SELECT @@IDENTITY: default scope is session
  - SELECT SCOPE_IDENTITY(): scope is object containing the call
  - SELECT IDENT_CURRENT(' tablename'): in this case, scope is defined by tablename

- There is a setting to allow identity columns to be changed manually ON or automatic OFF
  - SET IDENTITY_INSERT <Tablename> [ON|OFF]

Using Sequences

Sequence objects were first added in SQL Server 2012

- Independent objects in database
  - More flexible than the IDENTITY property
  - Can be used as default value for a column

- Manage with CREATE/ALTER/DROP statements
- Retrieve value with the NEXT VALUE FOR clause

```
-- Define a sequence
CREATE SEQUENCE dbo.InvoiceSeq AS INT START WITH 1
INCREMENT BY 1;

-- Retrieve next available value from sequence
SELECT NEXT VALUE FOR dbo.InvoiceSeq;
```

Lab Scenario

You are a database developer need to create DML statements to update data in the database to support the website development team.

The team need T-SQL statements that they can use to carry out updates to data, based on actions performed on the website. You will supply template DML statements that they can modify to their specific requirements.

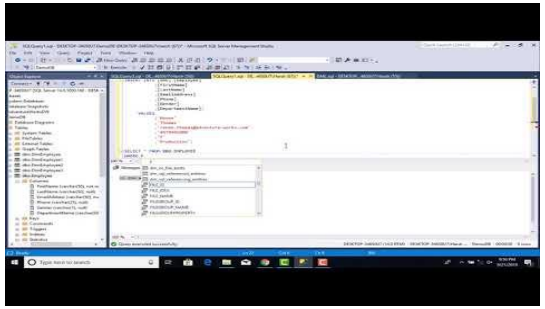
SQL Server - Modifying data

T-SQL DML Statement Web Resource:

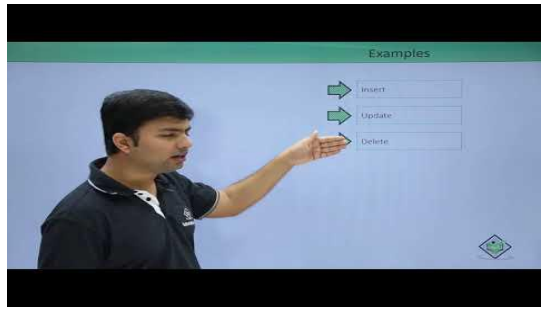
- Microsoft Docs | T-SQL DML statements
- W3Schools | Insert, Update, Delete
- SQL Server Tutorial.net | SQL Server INSERT
- Tutorialspoint | Insert, Update, Delete
- JavaTpoint | SQL Server Insert Data, Update, Delete
- T-SQL Tutorial | T-SQL Statements - DML, DDL, DCL and TCL

T-SQL DML Statement Video link:

SQL DML Statements | Select | Update | Insert | Delete



T-SQL - DML Statements



SQL Server - Modifying data

The SQL commands for modifying data such as insert, delete, and update are referred to as data manipulation language (DML)

- INSERT – insert a row into a table
- INSERT multiple rows – insert multiple rows into a table using a single INSERT statement
- INSERT INTO SELECT – insert data into a table from the result of a query.
- UPDATE – change the existing values in a table.
- UPDATE JOIN – update values in a table based on values from another table using JOIN clauses.
- DELETE – delete one or more rows of a table.
- MERGE – walk you through the steps of performing a mixture of insertion, update, and deletion using a single statement.