Systems Design and Databases (CIS1018-N)

Week 8

Working with SQL Server 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

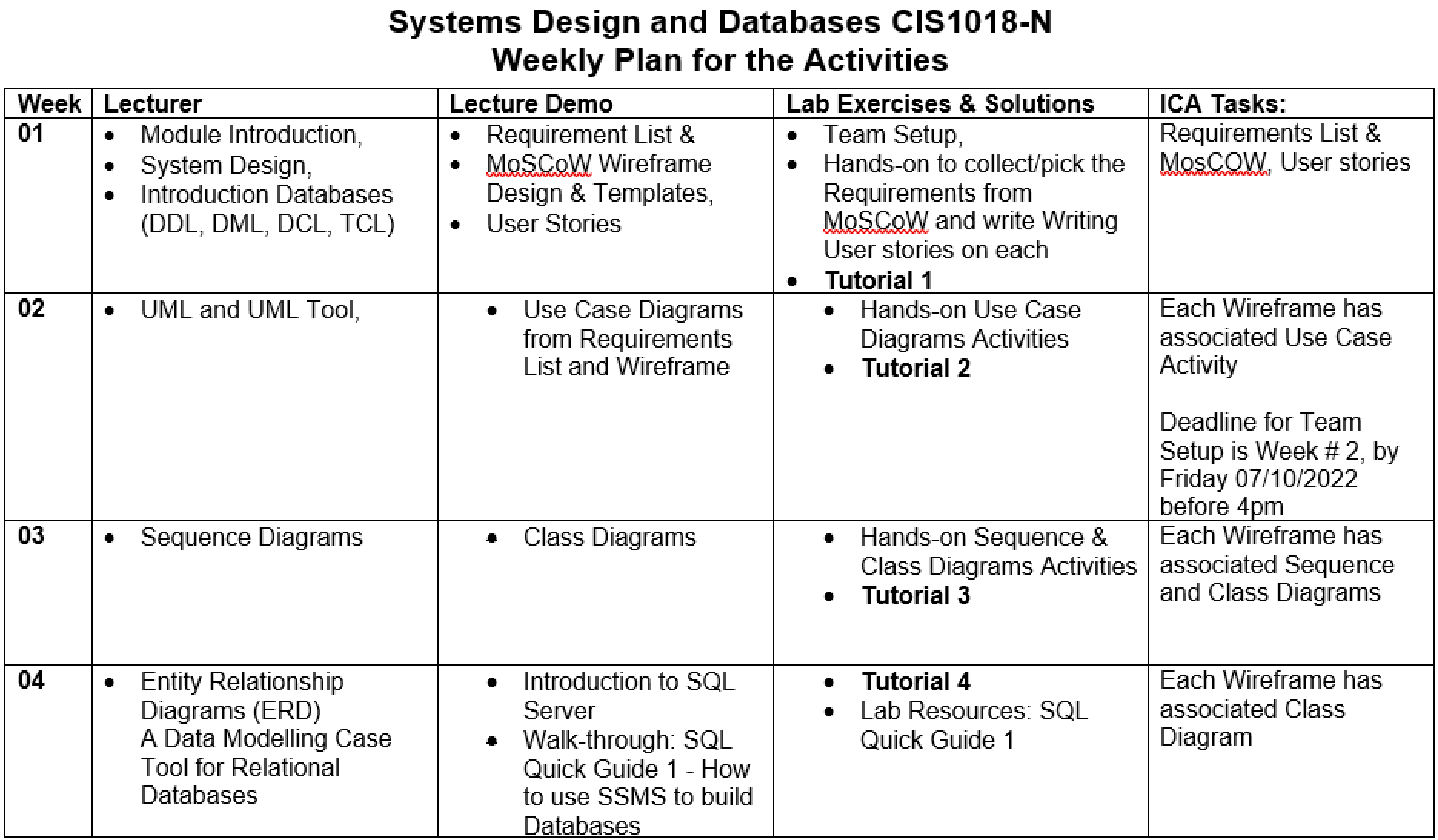
Lectures – Dr Yar Muhammad	Tuesdays @ 2-3 pm	Thursdays @ 1-2 pm
Week 1 – Week 12	CL1.87	

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 - UML



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

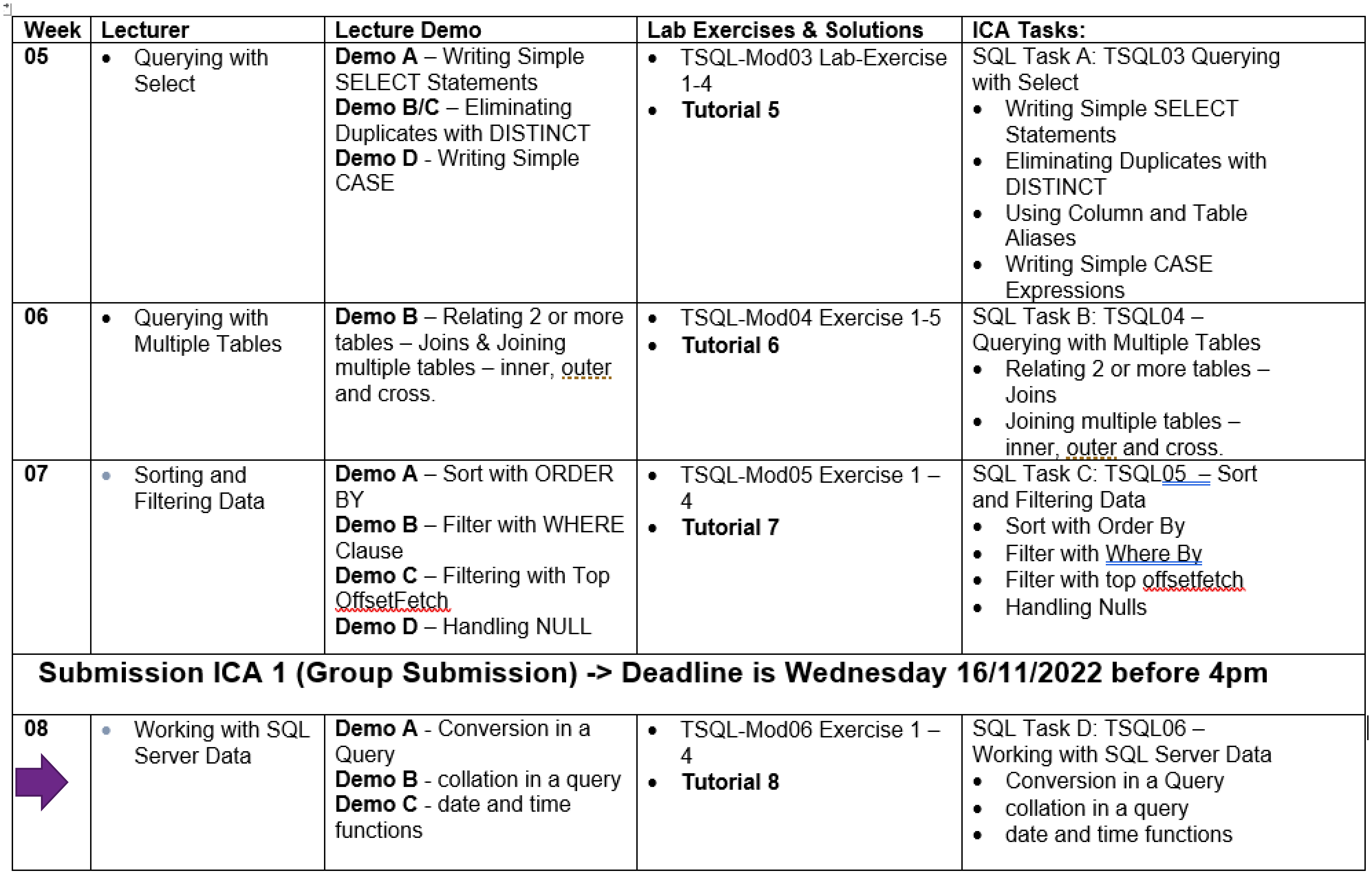
Week	Lecturer	Lecture Demo	Lab Exercises & Solutions	ICA Tasks:
01	- Module Introduction, - System Design, - Introduction Databases (DDL, DML, DCL, TCL)	- Requirement List & - MoSCoW Wireframe Design & Templates, - User Stories	- Team Setup, - Hands-on to collect/pick the Requirements from MoSCoW and 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

Analysis

Design

Schedule 2/3

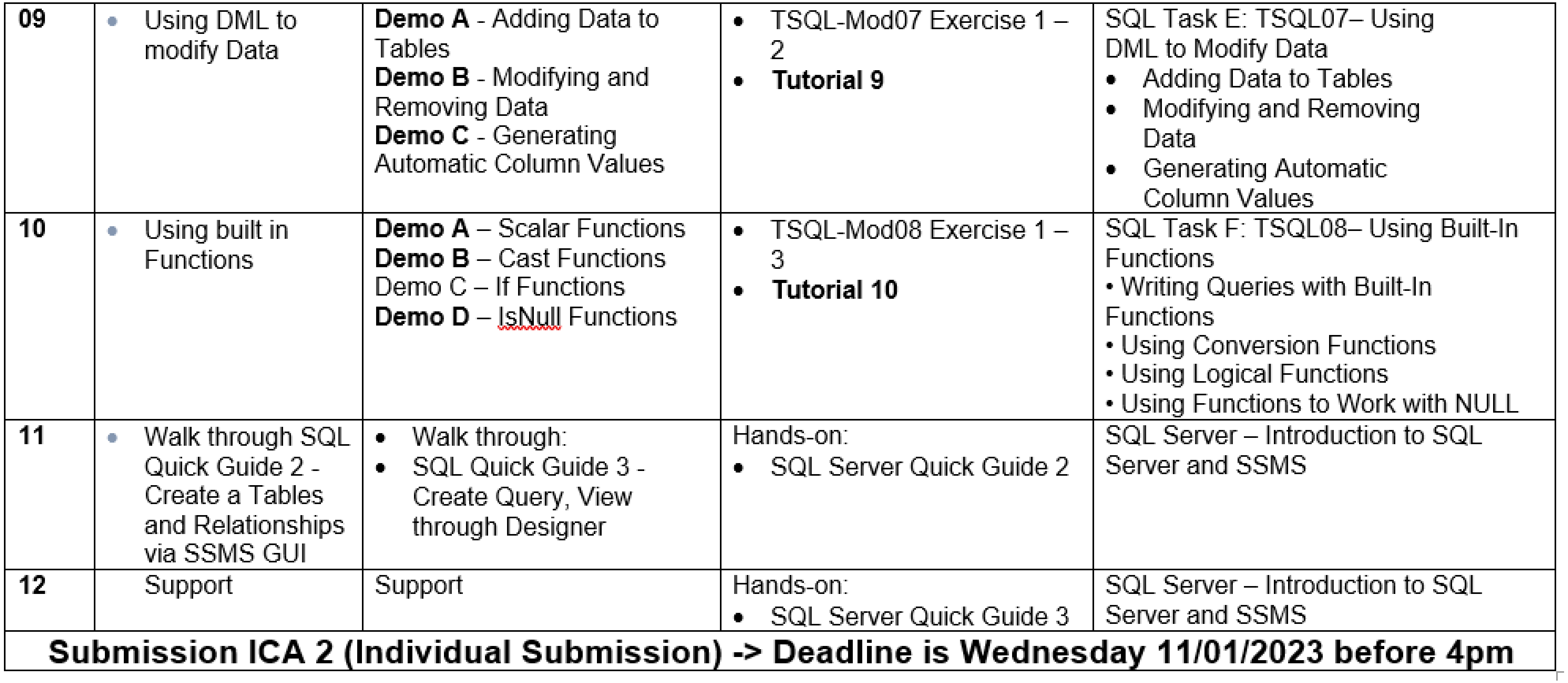
Databases - TSQL



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				

Overview

- Introducing SQL Server 2019 Data Types
- Working with Character Data
- Working with Date and Time Data

Introducing SQL Server 2019 Data Types

- SQL Server Data Types
- Numeric Data Types
- Binary String Data Types
- Other Data Types
- Data Type Precedence
- When are Data Types Converted?
- Demonstration: SQL Server Data Types

SQL Server Data Types

- SQL Server associates' columns, expressions, variables and parameters with data types
- Data types determine the kind of data that can be held in a column or variable
  - Integers, characters, dates, decimals, binary strings, and so on
- SQL Server supplies built-in data types
- Developers can also define custom data types

SQL Server Data Type Categories	
Exact numeric	Unicode character strings
Approximate numeric	Binary strings
Date and time	Other
Character strings	

Numeric Data Types

- Exact Numeric Data Types

Data Type	Range	Storage (bytes)
tinyint	0 to 255	1
smallint	-32,768 to 32,768	2
int	2^{31} (-2,147,483,648) to 2^{31}-1 (2,147,483,647)	4
bigint	-2^{63} - 2^{63}-1 (+/- 9 quintillion)	8
bit	1, 0 or NULL	1
decimal/numeric	-10^{38} +1 through 10^{38} – 1 when maximum precision is used	5-17
money	-922,337,203,685,477.5808 to 922,337,203,685,477.5807	8
smallmoney	-214,748.3648 to 214,748.3647	4

Binary String Data Types

- Binary string data types

Data Type	Range	Storage (bytes)
binary(n)	1 to 8000 bytes	n bytes
varbinary(n)	1 to 8000 bytes	n bytes + 2
varbinary(max)	1 to 2.1 billion (approx.) bytes	n bytes + 2

- The **image** data type is also a binary string type but is marked for removal in a future version of SQL Server; **varbinary(max)** should be used instead

Other Data Types

Data Type	Range	Storage (bytes)	Remarks
xml	0-2 GB	0-2 GB	Stores XML in native hierarchical structure
uniqueidentifier	Auto-generated	16	Globally unique identifier (GUID)
hierarchyid	n/a	Depends on content	Represents position in a hierarchy
rowversion	Auto-generated	8	Previously called timestamp
geometry	0-2 GB	0-2 GB	Shape definitions in Euclidian geometry
geography	0-2 GB	0-2 GB	Shape definitions in round-earth geometry
sql_variant	0-8000 bytes	Depends on content	Can store data of various other data types in the same column
cursor	n/a	n/a	Not a storage datatype—used for cursor operations
table	n/a	n/a	Not a storage data type—used for query operations

Data Type Precedence

- Data type precedence determines which data type will be chosen when expressions of different types are combined
- By default, the data type with the lower precedence is converted to the data type with the higher precedence
- It is important to understand implicit conversions
  - Conversion to a data type of lower precedence must be made explicitly (using CAST or CONVERT functions)
- Example precedence (low to high)
  - CHAR -> VARCHAR -> NVARCHAR -> TINYINT -> INT -> DECIMAL -> TIME -> DATE -> DATETIME2 -> XML
- Not all combinations of data type have a conversion (implicit or explicit)

When are Data Types Converted?

- Data type conversion scenarios
  - When data is moved, compared to or combined with other data
  - During variable assignment
- Implicit conversion
  - When comparing data of one data type to another
  - Transparent to the user

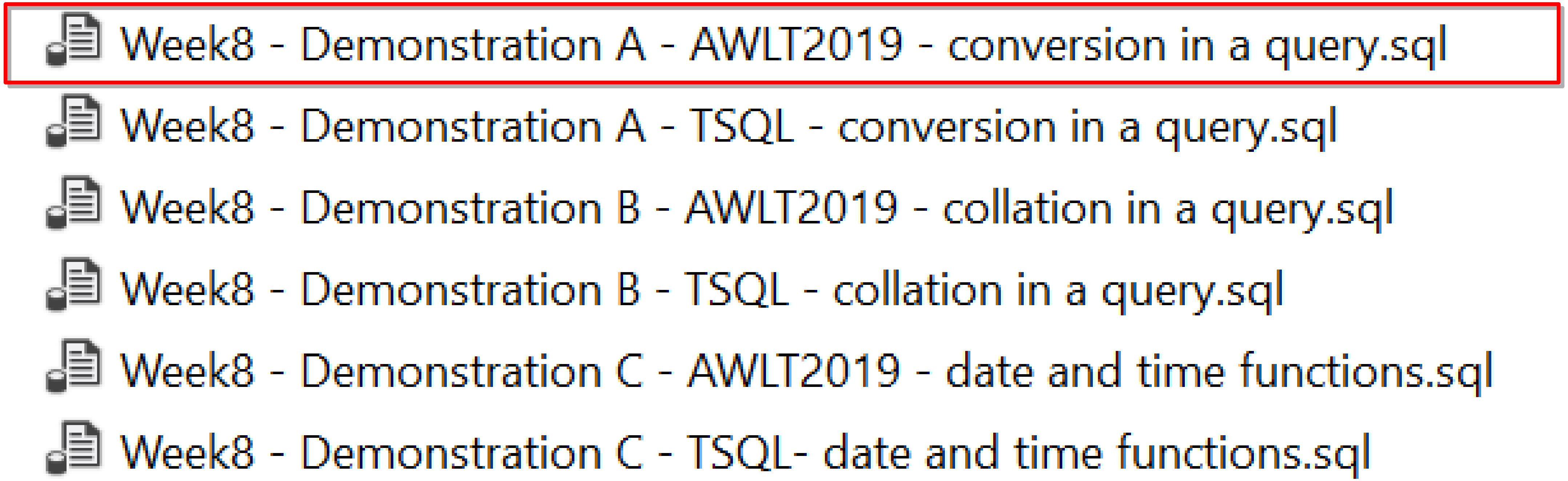
```
WHERE <column of smallint type> = <value of int type>
```

- Explicit conversion
  - Uses CAST or CONVERT functions

```
CAST(unitprice AS INT)
```

Demonstration A withAdventureWorksLT2019 :
SQL Server Data Types

In this demonstration, you will see how to Convert data types



Week8 - Demonstration A - AWLT2019 - conversion in a query.sql

Week8 - Demonstration A - TSQL - conversion in a query.sql

Week8 - Demonstration B - AWLT2019 - collation in a query.sql

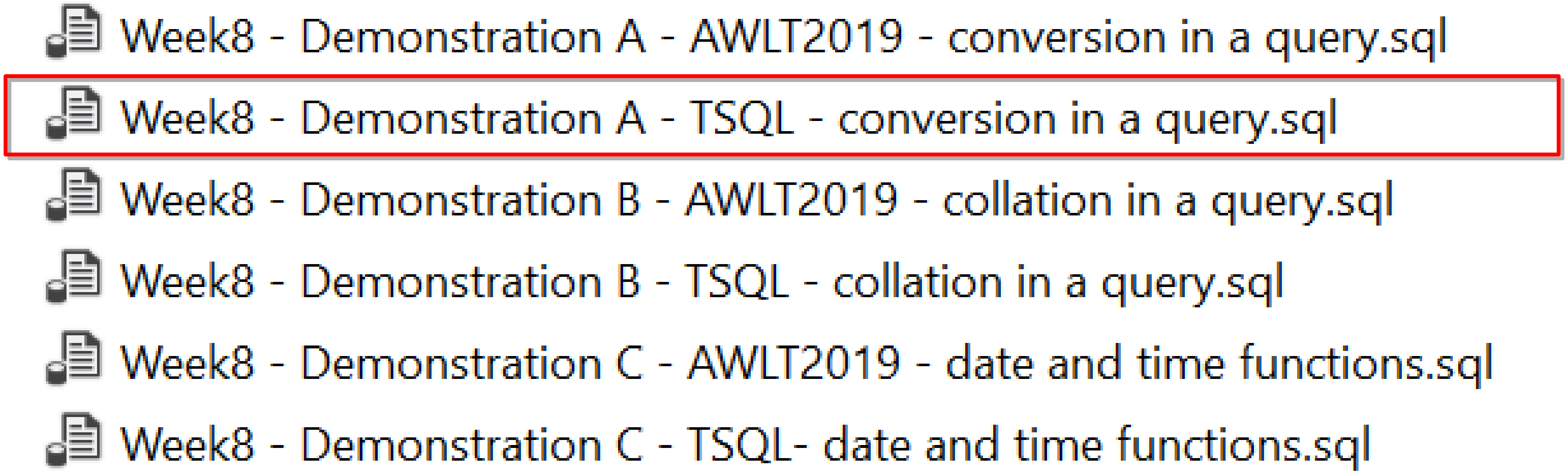
Week8 - Demonstration B - TSQL - collation in a query.sql

Week8 - Demonstration C - AWLT2019 - date and time functions.sql

Week8 - Demonstration C - TSQL- date and time functions.sql

Demonstration A with TSQL: SQL Server Data Types

In this demonstration, you will see how to Convert data types



Week8 - Demonstration A - AWLT2019 - conversion in a query.sql

Week8 - Demonstration A - TSQL - conversion in a query.sql

Week8 - Demonstration B - AWLT2019 - collation in a query.sql

Week8 - Demonstration B - TSQL - collation in a query.sql

Week8 - Demonstration C - AWLT2019 - date and time functions.sql

Week8 - Demonstration C - TSQL- date and time functions.sql

Working with Character Data

- Character Data Types
- Collation
- String Concatenation
- Character String Functions
- The LIKE Predicate
- Demonstration: Working with Character Data

Character Data Types

- SQL Server supports two kinds of character data as fixed-width or variable-width data:
  - Single-byte: **char** and **varchar**
    - One byte stored per character
      - Only 256 possible characters—limits language support
  - Multibyte: **nchar** and **nvarchar**
    - Multiple bytes stored per character (usually two bytes, but sometimes up to four)
      - More than 65,000 characters represented—multiple language support
      - Precede character string literals with N (National)
  - **text** and **ntext** data types are deprecated, but may still be used in older systems
    - In new development, use **varchar(max)** and **nvarchar(max)** instead

Collation

- Collation is a collection of properties for character data
  - Character set
  - Sort order
  - Case sensitivity
  - Accent sensitivity
- When querying, collation awareness is important for comparison
  - Is the database case-sensitive? If so:
    - 'Funk‘ does not equal 'funk‘
    - SELECT * FROM HR.Employee does not equal SELECT * FROM HR.employee
- Add COLLATE clause to control collation comparison

```sql
SELECT empid, lastname
FROM HR.employees
WHERE lastname COLLATE Latin1_General_CS_AS = N'Funk';
```

String Concatenation

- The + (plus) operator and the CONCAT function can both be used to concatenate strings in SQL 2019
  - Using CONCAT
    - Converts input values to strings and converts NULL to empty string

```sql
SELECTcustid, city, region, country,
  CONCAT(city, ', ' + region, ', ' + country) AS location
FROM Sales.Customers;
```

  - Using + (plus)
    - No conversion of NULL or data type

```sql
SELECT empid, lastname, firstname,
firstname + N' ' + lastname AS fullname
FROM HR.Employees;
```

Character String Functions

- Common functions that modify character strings

Function	Syntax	Remarks
SUBSTRING	SUBSTRING (expression , start , length)	Returns part of an expression.
LEFT, RIGHT	LEFT (expression , integer_value) RIGHT (expression , integer_value)	LEFT returns left part of string up to integer_value. RIGHT returns right part of string up to integer value.
LEN, DATALENGTH	LEN (string_expression) DATALENGTH (expression)	LEN returns the number of characters in string_expression, excluding trailing spaces. DATALENGTH returns the number of bytes used.
CHARINDEX	CHARINDEX (expressionToFind, expressionToSearch)	Searches expressionToSearch for expressionToFind and returns its start position if found.
REPLACE	REPLACE (string_expression , string_pattern , string_replacement)	Replaces all occurrences of string_pattern in string_expression with string_replacement.
UPPER, LOWER	UPPER (character_expression) LOWER (character_expression)	UPPER converts all characters in a string to uppercase. LOWER converts all characters in a string to lowercase.

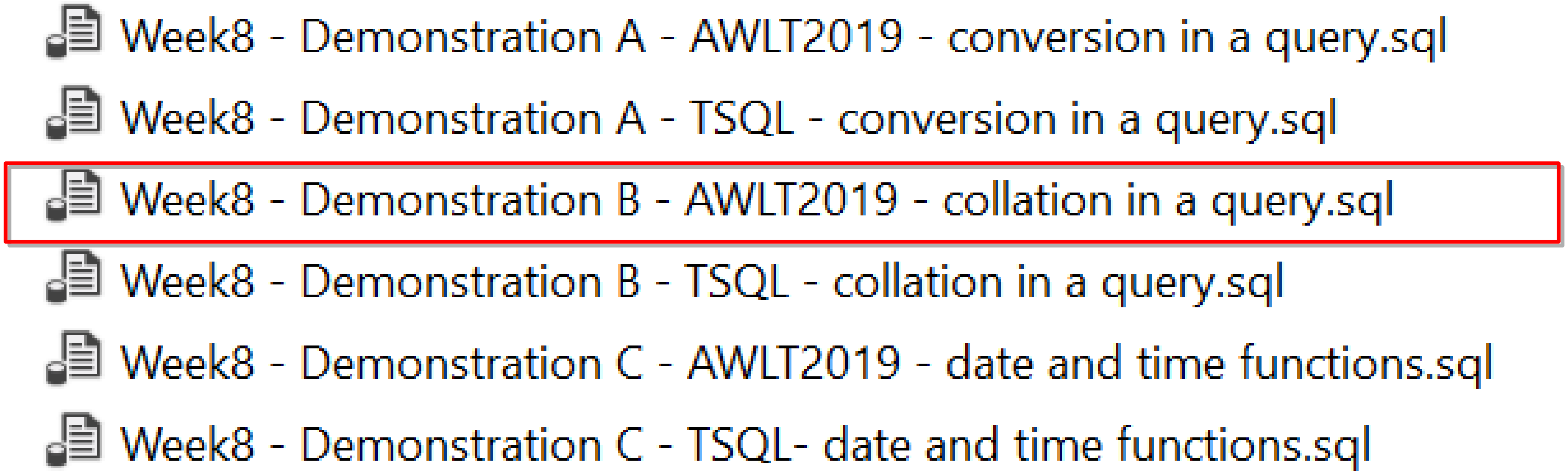
The LIKE Predicate

- The LIKE predicate can be used to check a character string for a match with a pattern
- Patterns are expressed with symbols
    - % (Percent) represents a string of any length
    - _ (Underscore) represents a single character
    - [<List of characters>] represents a single character within the supplied list
    - [<Character> - <character>] represents a single character within the specified range
    - [^<Character list or range>] represents a single character not in the specified list or range
    - ESCAPE Character allows you to search for characters that would otherwise be treated as part of a pattern - %, _, [, and])

```sql
SELECT categoryid, categoryname, description
FROM Production.Categories
WHERE description LIKE 'Sweet%';
```

Demonstration B with AdventureWorksLT2019 : Working with Character Data

In this demonstration, you will see how to Manipulate character data



Week8 - Demonstration A - AWLT2019 - conversion in a query.sql

Week8 - Demonstration A - TSQL - conversion in a query.sql

Week8 - Demonstration B - AWLT2019 - collation in a query.sql

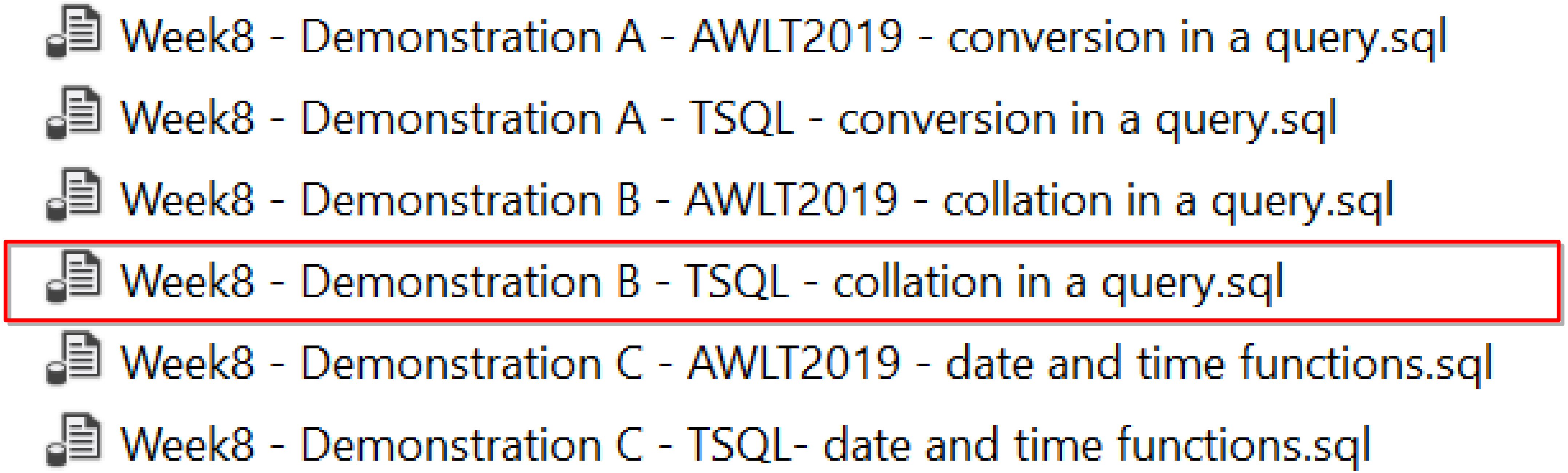
Week8 - Demonstration B - TSQL - collation in a query.sql

Week8 - Demonstration C - AWLT2019 - date and time functions.sql

Week8 - Demonstration C - TSQL- date and time functions.sql

Demonstration B with TSQL : Working with Character Data

In this demonstration, you will see how to Manipulate character data



Week8 - Demonstration A - AWLT2019 - conversion in a query.sql

Week8 - Demonstration A - TSQL - conversion in a query.sql

Week8 - Demonstration B - AWLT2019 - collation in a query.sql

Week8 - Demonstration B - TSQL - collation in a query.sql

Week8 - Demonstration C - AWLT2019 - date and time functions.sql

Week8 - Demonstration C - TSQL- date and time functions.sql

Working with Date and Time Data

- Date and Time Data Types
- Entering Date and Time Data Types Using Strings
- Working Separately with Date and Time
- Querying Date and Time Values
- Date and Time Functions
- Demonstration: Working with Date and Time Data

Date and Time Data Types

- Older versions of SQL Server support only **datetime** and **smalldatetime** data types
- SQL Server 2008 introduced **date, time, datetime2** and **datetimeoffset** data types
- SQL Server 2012 added further functionality for working with date and time data types

Data Type	Storage (bytes)	Date Range (Gregorian Calendar)	Accuracy	Recommended Entry Format
datetime	8	January 1, 1753 to December 31, 9999	Rounded to increments of .000, .003, or .007 seconds	YYYYMMDD hh:mm:ss[.mmm]
smalldatetime	4	January 1, 1900 to June 6, 2079	1 minute	YYYYMMDD hh:mm:ss[.mmm]
datetime2	6 to 8	January 1, 0001 to December 31, 9999	100 nanoseconds	YYYYMMDD hh:mm:ss[.nnnnnnn]
date	3	January 1, 0001 to December 31, 9999	1 day	YYYY-MM-DD
time	3 to 5	n/a – time only	100 nanoseconds	hh:mm:ss[.nnnnnnn]
datetimeoffset	8 to 10	January 1, 0001 to December 31, 9999	100 nanoseconds	YYYY-MM-DDThh:mm:ss[.nnnnnnn][{+|-}hh:mm]

Entering Date and Time Data Types Using Strings

- SQL Server doesn't offer a means to enter a date or time value as a literal value
  - Dates and times are entered as character literals and converted explicitly or implicitly
    - For example, **char** converted to **datetime** due to precedence
  - Formats are language-dependent, and can cause confusion
- Best practices:
  - Use character strings to express date and time values
  - Use language-neutral formats

```sql
SELECT orderid, custid, empid, orderdate
FROM Sales.Orders
WHERE orderdate = '20070825';
```

Working Separately with Date and Time

- **datetime, smalldatetime, datetime2**, and **datetimeoffset** include both date and time data
- If only date is specified, time set to midnight (all zeros)

```
DECLARE @DateOnly AS datetime2 = '20190112';
SELECT @DateOnly AS Result;
```

- If only time is specified, date set to base date (January 1, 1900)

```
DECLARE @time AS time = '12:34:56';
SELECT CAST(@time AS datetime2) AS Result;
```

Querying Date and Time Values

- Date values converted from character literals often omit time
  - Queries written with equality operator for date will match midnight

```sql
SELECT orderid, custid, empid, orderdate
FROM Sales.Orders
WHERE orderdate= '20070825';
```

- If time values are stored, queries need to account for time past midnight on a date
  - Use range filters instead of equality

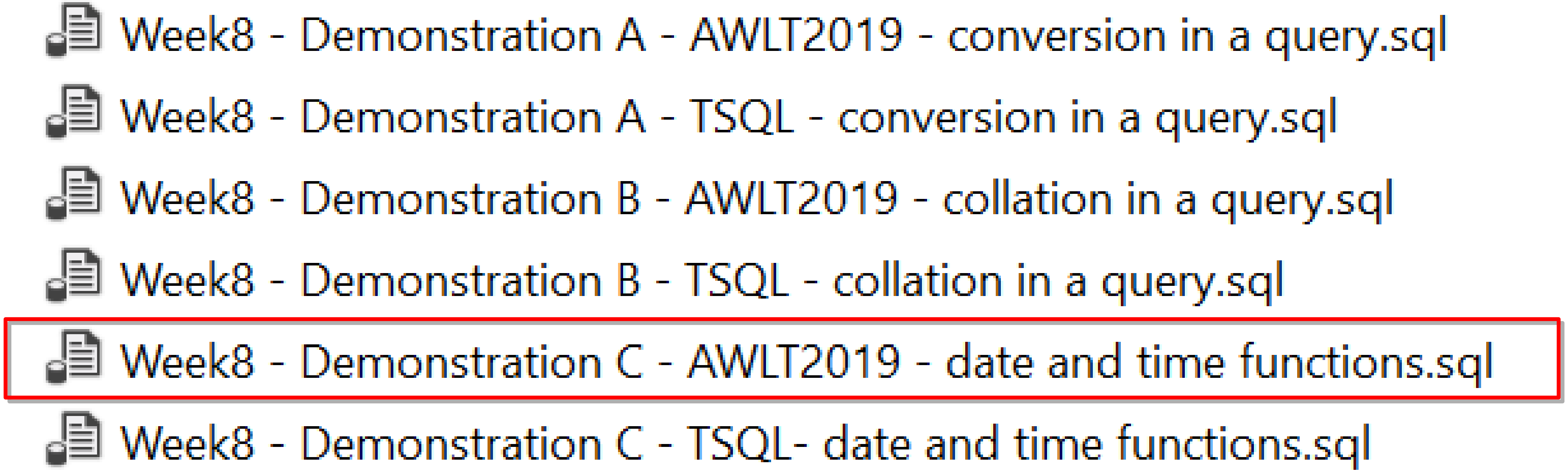
```sql
SELECT orderid, custid, empid, orderdate
FROM Sales.Orders
WHEREorderdate >= '20070825'
ANDorderdate <  '20070826';
```

Date and Time Functions

- To get system date and time values
  - For example, GETDATE, GETUTCDATE, SYSDATETIME
- To get date and time parts
  - For example, DATENAME, DATEPART
- To get date and time values from their parts
  - For example, DATETIME2FROMPARTS, DATEFROMPARTS
- To get date and time difference
  - For example, DATEDIFF, DATEDIFF_BIG
- To modify date and time values
  - For example, DATEADD, EOMONTH
- To validate date and time values
  - For example, ISDATE

Demonstration C with AdventureWorksLT2019: Working with Date and Time Data

In this demonstration, you will see how to Query date and time values



Week8 - Demonstration A - AWLT2019 - conversion in a query.sql

Week8 - Demonstration A - TSQL - conversion in a query.sql

Week8 - Demonstration B - AWLT2019 - collation in a query.sql

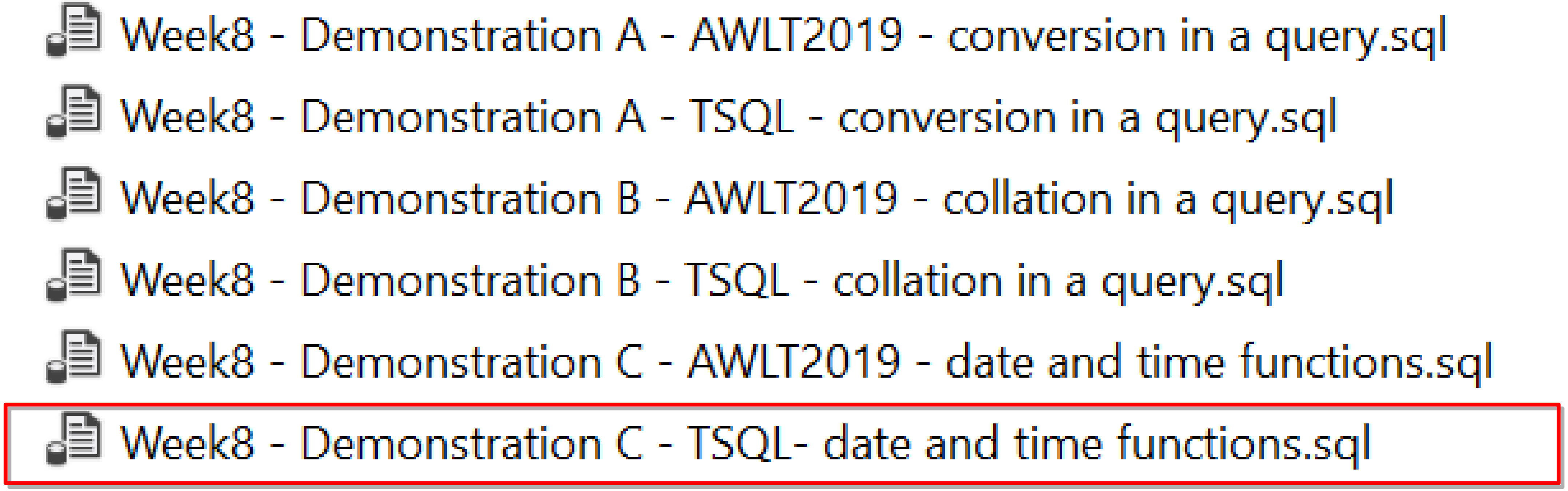
Week8 - Demonstration B - TSQL - collation in a query.sql

Week8 - Demonstration C - AWLT2019 - date and time functions.sql

Week8 - Demonstration C - TSQL- date and time functions.sql

Demonstration C with TSQL: Working with Date and Time Data

In this demonstration, you will see how to Query date and time values



Week8 - Demonstration A - AWLT2019 - conversion in a query.sql

Week8 - Demonstration A - TSQL - conversion in a query.sql

Week8 - Demonstration B - AWLT2019 - collation in a query.sql

Week8 - Demonstration B - TSQL - collation in a query.sql

Week8 - Demonstration C - AWLT2019 - date and time functions.sql

Week8 - Demonstration C - TSQL- date and time functions.sql

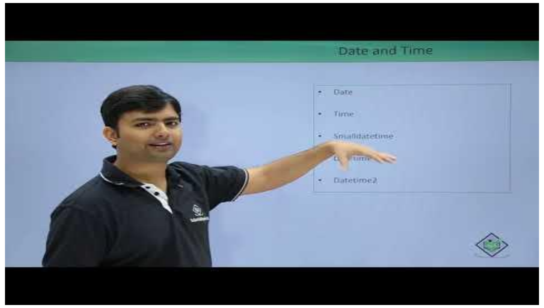
Transact-SQL Data types

SQL Data types (Transact-SQL) web resources:

- Microsoft Docs | Data types (Transact-SQL)
- W3Schools | SQL Data Type
- SQL Server Tutorial.net | SQL Server Data Types
- Tutorialspoint | T-SQL - Data Types
- JavaTpoint | Data types T-SQL

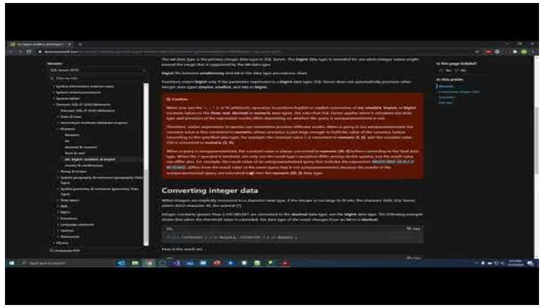
SQL Data types (Transact-SQL) Video link:

T-SQL - Data Types



Data Types in SQL Server

T-SQL Data Types



Converting integer data