Systems Design and Databases (CIS1018-N)

Week 10

Using built in Functions

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

- Writing Queries with Built-In Functions
- Using Conversion Functions
- Using Logical Functions
- Using Functions to Work with NULL

Writing Queries with Built-In Functions

- SQL Server Built-in Function Types
- Scalar Functions
- Aggregate Functions
- Window Functions
- Rowset Functions
- Demonstration: Writing Queries Using Built-in Functions

SQL Server Built-in Function Types

- SQL Server functions can be categorized by scope of input and type of output:

Function Category	Description
Scalar	Operate on a single row, return a single value
Grouped Aggregate	Take one or more values but return a single summarizing value
Window	Operate on a window (set) of rows
Rowset	Return a virtual table that can be used in a T-SQL statement

Scalar Functions

- Operate on elements from a single row as inputs, return a single value as output
- Return a single (scalar) value
- Can be used like an expression in queries
- May be deterministic or nondeterministic
- Collation depends on input value or default collation of database

Scalar Function Categories

- Configuration
- Conversion
- Cursor
- Date and Time
- Logical
- Mathematical
- Metadata
- Security
- String
- System
- System Statistical
- Text and Image

Aggregate Functions

- Functions that operate on sets, or rows, of data
- Summarize input rows
- Without GROUP BY clause, all rows are arranged as one group
- Will be covered later in the course

```
SELECT COUNT(*) AS numorderlines,
     SUM(qty*unitprice) AS totalsales
FROM   Sales.OrderDetails;
```

```
numorderlines totalsales
------------- -----------
2155          56500.91
```

Window Functions

- Functions applied to a window, or set of rows
- Include ranking, offset, aggregate, and distribution functions
- Will be covered later in the course

```
SELECT TOP(5) productid, productname, unitprice,
    RANK() OVER(ORDER BY unitprice DESC) AS
    rankbyprice
FROM Production.Products
ORDER BY rankbyprice;
```

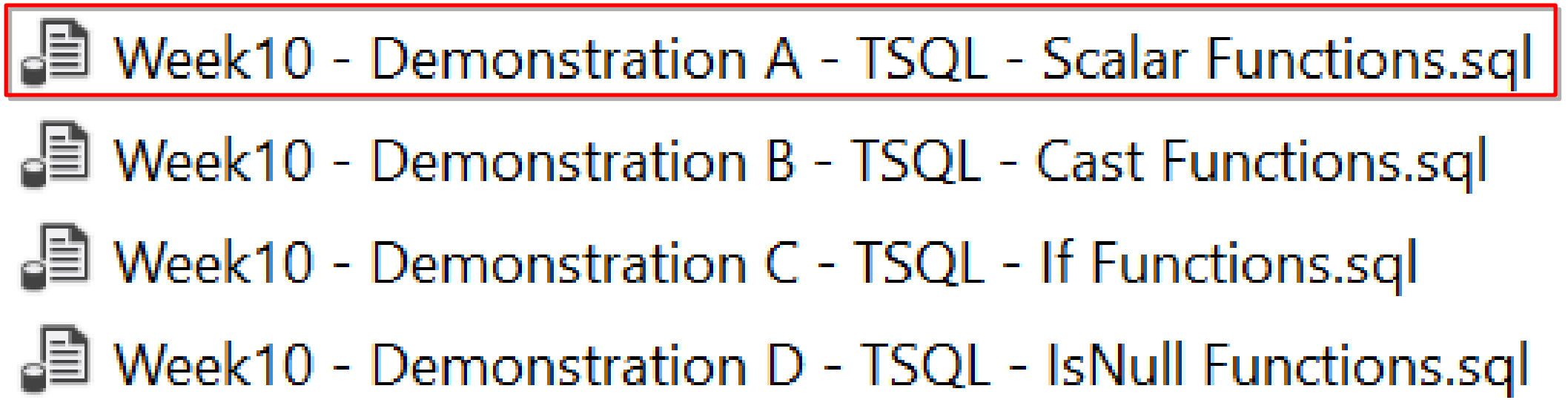
```
productid productname   unitprice rankbyprice
--------- ------------- --------- -----------
8         Product QDOMO 263.50    1
29        Product VJXYN 123.79    2
9         Product AOZBW 97.00     3
20        Product QHFFP 81.00     4
18        Product CKEDC 62.50     5
```

Rowset Functions

- Return an object that can be used like a table in a T-SQL statement
- Include OPENDATASOURCE, OPENQUERY, OPENROWSET, and OPENXML
- Beyond the scope of this course

Demonstration A: Writing Queries Using Built-in Functions

In this demonstration, you will see how to Use built-in scalar functions



Using Conversion Functions

- Implicit and Explicit Data Type Conversions
- Converting with CAST
- Converting with CONVERT
- Converting Strings with PARSE
- Converting with TRY_PARSE and TRY_CONVERT
- Demonstration: Using Conversion Functions

Implicit and Explicit Data Type Conversions

- Implicit conversion occurs automatically and follows data type precedence rules
- Use explicit conversion:
  - When implicit would fail or is not permitted
  - To override data type precedence
- Explicitly convert between types with CAST or CONVERT functions
- Watch for truncation

Converting with CAST

- Converts a value from one data type to another:
  - Can be used in SELECT and WHERE clauses
  - ANSI standard
- CAST syntax:

```
CAST(<value> AS <datatype>)
```

- CAST example:

```
SELECT CAST(SYSDATETIME() AS date);
```

- Returns an error if data types are incompatible:

```
--attempt to convert datetime2 to int
SELECT CAST(SYSDATETIME() AS int);
```

```
Msg 529, Level 16, State 2, Line 1
Explicit conversion from data type datetime2 to int is not allowed.
```

Converting with CONVERT

- Converts a value from one data type to another:
  - Can be used in SELECT and WHERE clauses
  - CONVERT is specific to SQL Server, not standards-based
- Style specifies how input value is converted:
  - Date, time, numeric, XML, and so on
- Syntax:

```
CONVERT (<datatype>, <value>, <optional style no.>)
```

- Example:

```
CONVERT(CHAR(8), CURRENT_TIMESTAMP,112) AS ISO_style;
```

```
ISO_style
---------
20120212
```

Converting Strings with PARSE

- PARSE converts strings to date, time, and number types:

PARSE element	Comment
String_value	Formatted nvarchar(4000) input
Data_type	Requested data type ouput
Culture	Optional string in .NET culture form: en-US, es-ES, ar-SA, and so on

- PARSE example:

```
SELECT PARSE('02/12/2012' AS datetime2
   USING 'en-US') AS parse_result;
```

Converting with TRY_PARSE and TRY_CONVERT

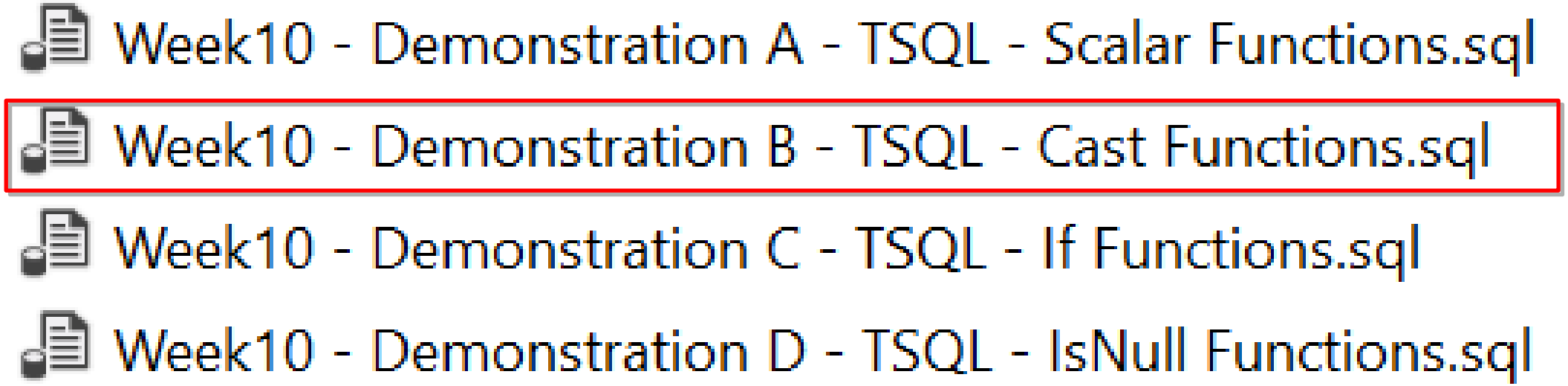
- TRY_PARSE and TRY_CONVERT:
  - Return the results of a data type conversion:
    - Like PARSE and CONVERT, they convert strings to date, time and numeric types
    - Unlike PARSE and CONVERT, they return a NULL if the conversion fails
- TRY_PARSE Example:

```
SELECT TRY_PARSE('SQLServer' AS datetime2
   USING 'en-US') AS try_parse_result;
```

```
try_parse_result
----------------
NULL
```

Demonstration B: Using Conversion Functions

In this demonstration, you will see how to Use functions to convert data



Using Logical Functions

- Writing Logical Test with Functions
- Performing Conditional Tests with IIF
- Selecting Items from a List with CHOOSE
- Demonstration: Using Logical Functions

Writing Logical Test with Functions

- ISNUMERIC tests whether an input expression is a valid numeric data type:
  - Returns a 1 when the input evaluates to any valid numeric type, including FLOAT and MONEY
  - Returns 0 otherwise
- Example:

```
SELECT ISNUMERIC('SQL') AS isnmumeric_result;
```

```
isnmumeric_result
-----------------
0
```

```
SELECT ISNUMERIC('101.99') AS isnmumeric_result;
```

```
isnmumeric_result
-----------------
1
```

Performing Conditional Tests with IIF

- IIF returns one of two values, depending on a logical test
- Shorthand for a two-outcome CASE expression:

IIF Element	Comments
Boolean_expression	Logical test evaluating to TRUE, FALSE, or UNKNOWN
True_value	Value returned if expression evaluates to TRUE
False_value	Value returned if expression evaluates to FALSE or UNKNOWN

- IIF example:

```
SELECT      productid, unitprice,
            IIF(unitprice > 50, 'high','low') AS pricepoint
FROM Production.Products;
```

Selecting Items from a List with CHOOSE

- CHOOSE returns an item from a list as specified by an index value:

CHOOSE Element	Comments
Index	Integer that represents position in list
Value_list	List of values of any data type to be returned

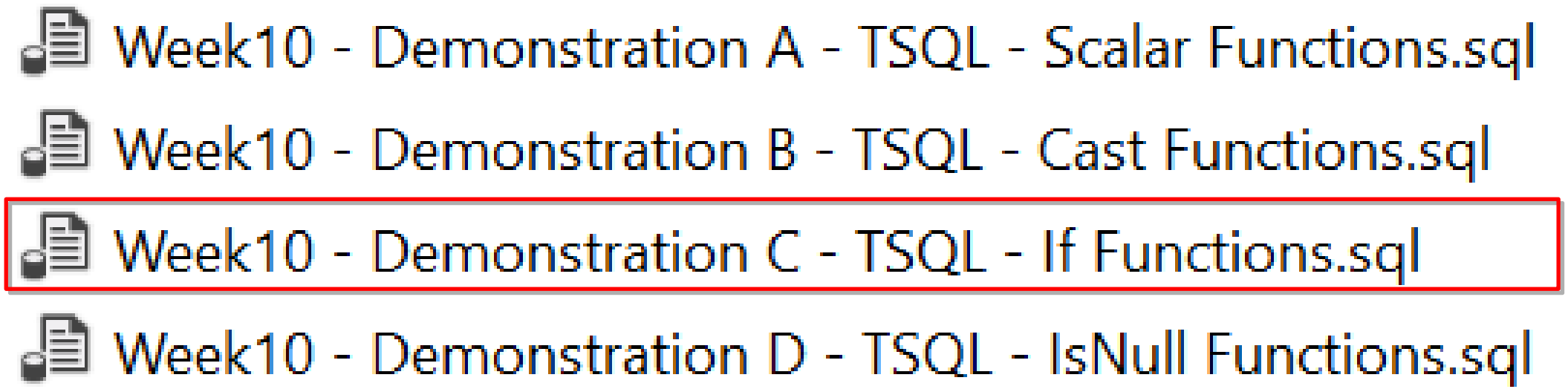
- CHOOSE example:

```
SELECT CHOOSE (3, 'Beverages', 'Condiments', 'Confections')
AS choose_result;
```

```
choose_result
-------------
Confections
```

Demonstration C: Using Logical Functions

In this demonstration, you will see how to Use logical functions



Using Functions to Work with NULL

- Converting NULL with ISNULL
- Using COALESCE to Return Non-NULL Values
- Using NULLIF to Return NULL If Values Match
- Demonstration: Using Functions to Work with NULL

Converting NULL with ISNULL

- ISNULL replaces NULL with a specified value
- Not standard; use COALESCE instead
- Syntax:

ISNULL Element	Comment
expression_to_check	Return expression itself if not NULL
replacement_value	Returned if expression evaluates to NULL

- ISNULL example:

```
SELECT custid, city, ISNULL(region, 'N/A') AS region, country
FROM Sales.Customers;
```

```
custid      city            region          country
----------- --------------- --------------- --------------
7           Strasbourg      N/A             France
9           Marseille       N/A             France
32          Eugene          OR              USA
43          Walla Walla     WA              USA
45          San Francisco   CA              USA
```

Using COALESCE to Return Non-NULL Values

- COALESCE returns the first non-NULL value in a list:
  - With only two arguments, COALESCE behaves like ISNULL
  - If all arguments are NULL, COALESCE returns NULL
- COALESCE is standards-based
- COALESCE example:

```
SELECT custid, country, region, city,
    country + ',' + COALESCE(region, ' ') + ', ' + city as location
FROM Sales.Customers;
```

```
custid country region city        location
------ ------- ------ ----------- --------------------
-
17     Germany NULL   Aachen      Germany, , Aachen
65     USA     NM     Albuquerque USA,NM, Albuquerque
55     USA     AK     Anchorage   USA,AK, Anchorage
83     Denmark NULL   Århus       Denmark, , Århus
```

Using NULLIF to Return NULL If Values Match

- NULLIF compares two expressions:
  - Returns NULL if both arguments are equal
  - Returns the first argument if the two arguments are not equal

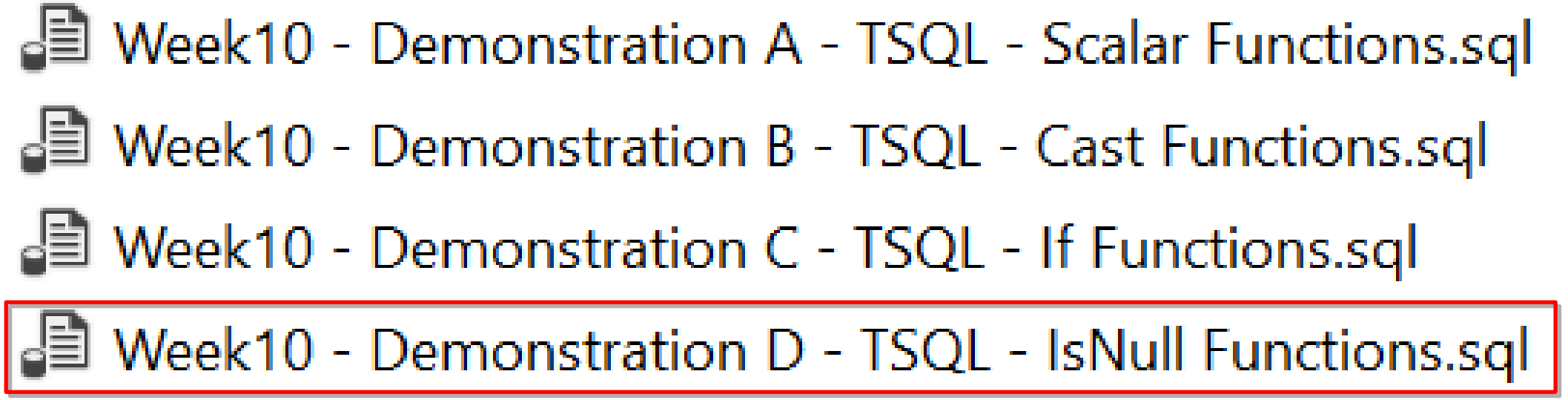
emp_id	goal	actual
1	100	110
2	90	90
3	100	90
4	100	80

```
SELECT emp_id, NULLIF(actual,goal) AS actual_if_different
FROM dbo.employee_goals;
```

```
emp_id      actual_if_different
----------- --------------------
1           110
2           NULL
3           90
4           80
```

Demonstration D: Using Functions to Work with NULL

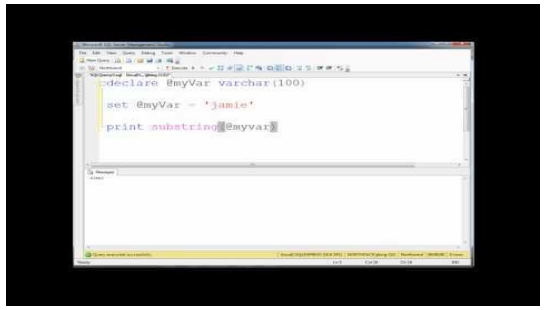
In this demonstration, you will see how to Use functions to work with NULL



Supporting Material - SQL Server Built-In Functions 1/2

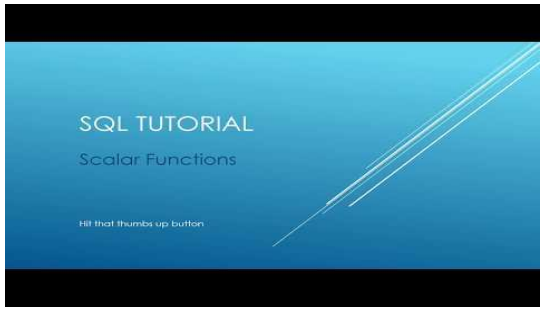
Built-In Functions Video link:

[SQL Built-In Functions]



[Built in string functions in sql]

[SQL Tutorial - Scalar Functions]



[SQL Functions with Examples]

Supporting Material - SQL Server Built-In Functions 2/2

Built-In Functions Web Resource:

- Microsoft Docs | Built-In Functions
- W3Schools | SQL Server Functions
- SQL Server Tutorial.net | SQL Server Scalar Functions
- Tutorialspoint | T-SQL Functions
- JavaTpoint | T-SQL Functions