# Systems Design and Databases (CIS1018-N) TSQL Tutorial 10: Using built in Functions

## Before You Start

Finish the previous tutorials before attempting this one. Try to complete this set of tasks before your next tutorial.

**Hint:** It is important before start the lab, you should walk-through lecture(s) and demonstration exercises

## Introduction

You are an Adventure Works business analyst, who will be writing reports using corporate databases stored in SQL Server. You have been provided with a set of business requirements for data and you will write T-SQL queries to retrieve the specified data from the databases. You will need to retrieve the data, convert it, and then check for missing values.

## Sorting and Filtering Data

* Exercise 1: Writing Queries That Use Conversion Functions
* Exercise 2: Writing Queries That Use Logical Functions
* Exercise 3: Writing Queries That Test for Nullability

## Exercise 1: Writing Queries That Use Conversion Functions

**Task 1:**

* Write a SELECT statement against the Production.Products table to retrieve a calculated column named productdesc. The calculated column should be based on the columns productname and unitprice and look like this:
* The unit price for the Product HHYDP is 18.00 $.
* Execute the written statement and compare the results that you got with the possible result(s) of the query.
* Did you use the CAST or the CONVERT function? Which one do you think is more appropriate to use?

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

**Task 2:**

* The US marketing department has supplied you with a start date 4/1/2007 (using US English form, read as April 1, 2007) and an end date 11/30/2007 (using US English form, read as November 30, 2007).
* Write a SELECT statement against the Sales.Orders table to retrieve the orderid, orderdate, shippeddate, and shipregion columns.
* Filter the result to include only rows with the order date between the specified start date and end date and have more than 30 days between the shipped date and order date. Also check the shipregion column for missing values.
* If there is a missing value, then return the value ‘No region’.
* In this SELECT statement, you can use the CONVERT function with a style parameter or the new PARSE function.
* Execute the written statement and compare the results that you got with the possible result(s) of the query.

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

**Task 3:**

* The IT department would like to convert all the information about phone numbers in the Sales.Customers table to integer values. The IT staff indicated that all hyphens, parentheses, and spaces have to be removed before the conversion to an integer data type.
* Write a SELECT statement to implement the requirement of the IT department.
* Replace all the specified characters in the phone column of the Sales.Customers table and then convert the column from the nvarchar datatype to the int datatype.
* The T-SQL statement must not fail if there is a conversion error, but rather it should return a NULL. (Hint: First try writing a T-SQL statement using the CONVERT function and then use the new functionality in SQL Server 2012). Use the alias phoneasint for this calculated column.
* Execute the written statement and compare the results that you got with the possible result(s) of the query.

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

## Exercise 2: Writing Queries That Use Logical Functions

**Task 1:**

* Write a SELECT statement against the Sales.Customers table and retrieve the custid and contactname columns.
* Add a calculated column named segmentgroup using a logical function IIF with the value “Target group” for customers that are from Mexico and have in the contact title the value “Owner”. Use the value “Other” for the rest of the customers.
* Execute the written statement and compare the results that you got with the possible result(s) of the query.

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

**Task 2:**

* Modify the T-SQL statement from task 1 to change the calculated column to show the value “Target group” for all customers without a missing value in the region attribute or with the value “Owner” in the contact title attribute.
* Execute the written statement and compare the results that you got with the possible result(s) of the query.

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

**Task 3:**

* Write a SELECT statement against the Sales.Customers table and retrieve the custid and contactname columns. Add a calculated column named segmentgroup using the logic function CHOOSE with four possible descriptions (“Group One”, “Group Two”, “Group Three”, “Group Four”). Use the modulo operator on the column custid. (Use the expression custid % 4 + 1 to determine the target group.)
* Execute the written statement and compare the results that you got with the possible result(s) of the query.

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

## Exercise 3: Writing Queries That Test for Nullability

**Task 1:**

* Write a SELECT statement to retrieve the contactname and fax columns from the Sales.Customers table. If there is a missing value in the fax column, return the value ‘No information’.
* Write two solutions, one using the COALESCE function and the other using the ISNULL function.
* Execute the written statement and compare the results that you got with the possible result(s) of the query.
* What is the difference between the ISNULL and COALESCE functions?

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

**Task 2:**

* Update the provided T-SQL statement with a WHERE clause to filter the region column using the provided variable @region, which can have a value or a NULL.
* Test the solution using both provided variable declaration cases.

|  |
| --- |
| -- Insert Query here |
| USE TSQL;  DECLARE @region AS NVARCHAR(30) = NULL;  SELECT  custid, region  FROM Sales.Customers;  GO  DECLARE @region AS NVARCHAR(30) = N'WA';  SELECT  custid, region  FROM Sales.Customers; |

|  |
| --- |
| **Possible Result of Query** |
|  |

**Task 3:**

* Write a SELECT statement to retrieve the contactname, city, and region columns from the Sales.Customers table. Return only rows that do not have two characters in the region column, including those with an inapplicable region (where the region is NULL).
* Execute the written statement and compare the results that you got with the possible result(s) of the query.
* Notice the number of rows returned.

|  |
| --- |
| -- Insert Query here |
| USE TSQL; |

|  |
| --- |
| **Possible Result of Query** |
|  |

## Document History

Prepared by Dr Yar Muhammad,   
Revision 0. (08-Sep-22): This is the initial version of the 2022/23 exercise.