# Systems Design and Databases (CIS1018-N) TSQL Tutorial 8: Working with SQL Server Data

## 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 a business analyst who will be writing reports using corporate databases stored in SQL Server 2019. You have been given 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 and convert character, and date and time data into various formats.

## Sorting and Filtering Data

* Exercise 1: Writing Queries That Return Date and Time Data
* Exercise 2: Writing Queries That Use Date and Time Functions
* Exercise 3: Writing Queries That Return Character Data
* Exercise 4: Writing Queries That Use Character Functions

## Exercise 1: Writing Queries That Return Date and Time Data

**Task 1:**

* Write a SELECT statement to return columns that contain:
  + The current date and time. Use the alias currentdatetime.
  + Just the current date. Use the alias currentdate.
  + Just the current time. Use the alias currenttime.
  + Just the current year. Use the alias currentyear.
  + Just the current month number. Use the alias currentmonth.
  + Just the current day of month number. Use the alias currentday.
  + Just the current week number in the year. Use the alias currentweeknumber.
  + The name of the current month based on the currentdatetime column. Use the alias currentmonthname.
* Execute the written statement and compare the results that you got with the possible result(s) of the query. Your results will be different because of the current date and time value.
* Can you use the alias currentdatetime as the source in the second column calculation (currentdate)? Please explain.

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

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

**Task 2:**

* Write December 11, 2015, as a column with a data type of date. Use the different possibilities inside the T-SQL language (cast, convert, specific function, etc.) and use the alias somedate.

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

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

**Task 3:**

* Write a SELECT statement to return columns that contain:
  + Three months from the current date and time. Use the alias threemonths.
  + Number of days between the current date and the first column (threemonths). Use the alias diffdays.
  + Number of weeks between April 4, 1992, and September 16, 2011. Use the alias diffweeks.
  + First day in the current month based on the current date and time. Use the alias firstday.
* Execute the written statement and compare the results that you got with the possible result(s) of the query. Some results will be different because of the current date and time value.

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

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

**Task 4:**

* *The IT department has written a T-SQL statement that creates and populates a table named Sales.Somedates.*
* *Execute the provided T-SQL statement.*
* Write a SELECT statement against the Sales.Somedates table and retrieve the isitdate column. Add a new column named converteddate with a new date data type value based on the column isitdate. If the column isitdate cannot be converted to a date data type for a specific row, then return a NULL.
* Execute the written statement and compare the results that you got with the desired results shown in the file 54 - Lab Exercise 1 - Task 4 Result.txt.
* What is the difference between the SYSDATETIME and CURRENT\_TIMESTAMP functions?
* What is a language-neutral format for the DATE type?

|  |
| --- |
| -- Insert Query here |
| USE TSQL;  SET NOCOUNT ON;  DROP TABLE IF EXISTS Sales.Somedates;  CREATE TABLE Sales.Somedates (  isitdate varchar(9)  );  INSERT INTO Sales.Somedates (isitdate) VALUES  ('20110101'), ('20110102'), ('20110103X'), ('20110104'), ('20110105'), ('20110106'),  ('20110107Y'),  ('20110108');  SET NOCOUNT OFF;  SELECT isitdate  FROM Sales.Somedates;  -- Finally, Drop the Table  DROP TABLE Sales.Somedates; |

|  |
| --- |
| **Possible Result of Query** |
| SELECT isitdate FROM Sales.Somedates;    Possible Rresult of your Query    Alternate solution using TRY\_CONVERT |

## Exercise 2: Writing Queries That Use Date and Time Functions

**Task 1:**

* Write a SELECT statement to retrieve distinct values for the custid column from the Sales.Orders table. Filter the results to include only orders placed in February 2008.
* 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:**

* Write a SELECT statement with these columns:
* Current date and time
* First date of the current month
* Last date of the current month
* Execute the written statement and compare the results that you got with the possible result(s) of the query. The results will differ because they rely on the current date.

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

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

**Task 3:**

* Write a SELECT statement against the Sales.Orders table and retrieve the orderid, custid, and orderdate columns. Filter the results to include only orders placed in the last five days of the order month.
* 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 4:**

* Write a SELECT statement against the Sales.Orders and Sales.OrderDetails tables and retrieve all the distinct values for the productid column. Filter the results to include only orders placed in the first 10 weeks of the year 2007.
* 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 Return Character Data

**Task 1:**

* Write a SELECT statement against the Sales.Customers table and retrieve the contactname and city columns. Concatenate both columns so that the new column looks like this:
  + Allen, Michael (city: Berlin)
* Execute the written statement and compare the results that you got with the below given possible result of query.

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

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

**Task 2:**

* Copy the T-SQL statement in task 1 and modify it to extend the calculated column with new information from the region column. Treat a NULL in the region column as an empty string for concatenation purposes. When the region is NULL, the modified column should look like this:
  + Allen, Michael (city: Berlin, region: )
* When the region is not NULL, the modified column should look like this
  + Richardson, Shawn (city: Sao Paulo, region: SP)
* Execute the written statement and compare the results that you got with the below given possible result of query.

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

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

**Task 3:**

* Write a SELECT statement to retrieve the contactname and contacttitle columns from the Sales.Customers table. Return only rows where the first character in the contact name is ‘A’ through ‘G’.
* Execute the written statement and compare the results that you got with the below given possible result of query. Notice the number of rows returned.

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

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

## Exercise 4: Writing Queries That Use Character Functions

**Task 1:**

* Write a SELECT statement to retrieve the contactname column from the Sales.Customers table. Based on this column, add a calculated column named lastname, which should consist of all the characters before the comma.
* Execute the written statement and compare the results that you got with the below given possible result of query.

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

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

**Task 2:**

* Write a SELECT statement to retrieve the contactname column from the Sales.Customers table and replace the comma in the contact name with an empty string. Based on this column, add a calculated column named firstname, which should consist of all the characters after the comma.
* Execute the written statement and compare the results that you got with the below given possible result of query.

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

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

**Task 3:**

* Write a SELECT statement to retrieve the custid column from the Sales.Customers table. Add a new calculated column to create a string representation of the custid as a fixed-width (6 characters) customer code prefixed with the letter C and leading zeros. For example, the custid value 1 should look like C00001.
* Execute the written statement and compare the results that you got with the below given possible result of query.

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

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

**Task 4:**

* Write a SELECT statement to retrieve the contactname column from the Sales.Customers table. Add a calculated column, which should count the number of occurrences of the character ‘a’ inside the contact name. (Hint: Use the string functions REPLACE and LEN.) Order the result from rows with the highest occurrences to lowest.
* Execute the written statement and compare the results that you got with the below given possible result of query.

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

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

## Document History

Prepared by Dr Yar Muhammad,   
Revision 0. (02-Sep-22): This is the initial version of the 2022/23 exercise.