Microsoft SQL Server Express Edition

How to write data from JMdict.txt file into Microsoft SQL Server Express Edition local database.

1) Using Microsoft SQL Server Management Studio 2017, I created a new database. The name of the new database is called JMdict.

The Computer System automatically created the files 'JMdict.mdf' and 'JMdict_log.ldf' in the directory ../users/(MyUserName).

Automatically generated script

In Microsoft SQL Server Management Studio:

```
USE [JMdict]
GO

/****** Object:  Table [dbo].[Table_1]  Script Date: 20.02.2018 11:17:10 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Table_1](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[dbEnt_seq] [nvarchar](50) NOT NULL,
	[dbKeb] [nvarchar](50) NULL,
	[dbReb] [nvarchar](50) NOT NULL,
	[dbGloss] [nvarchar](250) NOT NULL
) ON [PRIMARY]
GO
```

2) C# Program to Insert Data

```
using System.Data.SqlClient;
```

**********MainWindow.xaml.cs**********

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data.SqlClient;

namespace WpfAppJMdictMSSQLLocalDB
{
    /// <summary>
    /// This Program reads data, writes two rows of data, deletes selected row from Microsoft
SQL Server Express Edition Local Database
    /// </summary>
    public partial class MainWindow : Window
    {
        //information aboug connection to Microsoft SQL Server localdb
        const string myConnectionString = @"Data
Source=(LocalDB)¥MSSQLLocalDB;AttachDbFilename=" +
"C:¥Users¥INSERTYOURUSERNAME¥JMdict.mdf; " +
"Integrated Security=True;Connect Timeout=30";
        SqlConnection myConnection = null;
        SqlCommand myCommand = null;
        SqlDataReader mySqlDataReader = null;
        StringBuilder myStringBuilder = new StringBuilder();
        string myQueryAllResultString;


        public MainWindow()
        {
            InitializeComponent();

            //1) READ FROM DATABASE
            //try opening a new connection

            try
            {
                myConnection = new SqlConnection(myConnectionString);
                myConnection.Open();
            }
            catch (Exception myConnectionFailed)
            {
                MessageBox.Show(myConnectionFailed.Message);
                return;
            }

            //try reading data from database
            try
            {
                myCommand = myConnection.CreateCommand();
                myCommand.CommandText = "SELECT dbo.Table_1.* FROM dbo.Table_1";
```

```
                mySqlDataReader = myCommand.ExecuteReader();
                myDatabaseReadEachLineWriteToString();
                myQueryAllResultString = myStringBuilder.ToString();
            }
            catch (Exception myReadingFailed)
            {
                MessageBox.Show(myReadingFailed.Message);
                return;
            }

            //write result into Textbox
            tbx1.Text = myQueryAllResultString;
            mySqlDataReader.Close();
            myConnection.Close();

            //2) INSERT 2 NEW ROWs INTO DATABASE
            //Reference:
www.microsoft.com/en-us/sql-server/developer-get-started/csharp/win/step/2.html

            //Write SQL Statement
            myStringBuilder.Clear();
            myStringBuilder.Append("INSERT dbo.Table_1 (dbEnt_seq, dbKeb, dbReb, dbGloss)");
            myStringBuilder.Append("VALUES ('test1', 'test2', 'test3', 'test4');");
            myStringBuilder.Append("INSERT dbo.Table_1 (dbEnt_seq, dbKeb, dbReb, dbGloss)");
            myStringBuilder.Append("VALUES ('2834160', N'湿布薬', N'しっぷやく', 'poultice;
cataplasm');");
            string sql = myStringBuilder.ToString();

            //try opening connection to database
            try
            {
                myConnection = new SqlConnection(myConnectionString);
                myConnection.Open();
            }
            catch (Exception myConnectionFailed)
            {
                MessageBox.Show(myConnectionFailed.Message);
                return;
            }

            //execute sql command
            using (SqlCommand myInsertCommand = new SqlCommand(sql, myConnection))
            {

                int rowsAffected = myInsertCommand.ExecuteNonQuery();
                tbx1.Text += "¥nNumber of rows inserted: " + rowsAffected.ToString();
            }

            //close connection
            myConnection.Close();


            //3 DELETE ROW FROM DATABASE

            myStringBuilder.Clear();
            myStringBuilder.Append("DELETE FROM dbo.Table_1 WHERE dbENT_seq = 'test1';");
            sql = myStringBuilder.ToString();
            //try opening connection to database
            try
```

```
            {
                myConnection = new SqlConnection(myConnectionString);
                myConnection.Open();
            }
            catch (Exception myConnectionFailed)
            {
                MessageBox.Show(myConnectionFailed.Message);
                return;
            }

            //execute sql command
            using (myCommand = new SqlCommand(sql, myConnection))
            {

                int rowsAffected = myCommand.ExecuteNonQuery();
                tbx1.Text += "¥nNumber of Rows deleted: " + rowsAffected.ToString();
            }

            //close connection
            myConnection.Close();

        }

        public void myDatabaseReadEachLineWriteToString()
        {
            while (mySqlDataReader.Read())
            {
                //mySqlDataReader[]
                myStringBuilder.AppendLine(mySqlDataReader.GetSqlInt32(0).ToString() + "; ");
                myStringBuilder.AppendLine(mySqlDataReader.GetSqlString(1).ToString() + ";
");
                myStringBuilder.AppendLine(mySqlDataReader.GetSqlString(2).ToString() + ";

");
                myStringBuilder.AppendLine(mySqlDataReader.GetSqlString(3).ToString() + ";

");
                myStringBuilder.AppendLine(mySqlDataReader.GetSqlString(4).ToString() + ";

");
            }
        }
    }
}
```

**********End MainWindow.xaml.cs**********

Untested additional information about the installation of SQL Server 2017 Express-Edition:

SQL Server 2017 Express-Edition

Erstellen Sie mit dieser kostenlosen Datenbank der Einsteigerklasse kleine, datengesteuerte Web- und Mobile-Anwendungen von bis zu 10 GB

(...)

Eine einfache Datenbank in Anwendungen einbetten

Erstellen Sie kleine Datenbanken in grundlegenden Desktopanwendungen oder Tools mit SQL Server Express LocalDB – eine einfache Bereitstellungsoption, die zusammen mit Anwendungen innerhalb des Prozesses ausgeführt wird.

(...)

SQL Server 2017 Express LocalDB

Betten Sie SQL Server Express mit LocalDB, einer einfachen Version der Express-Edition, in Anwendungen ein. Beim Herunterladen von SQL Server Express 2017 als kostenlose Option verfügbar.
Installationslink: **https://go.microsoft.com/fwlink/?LinkID=799012**
https://www.microsoft.com/de-ch/sql-server/sql-server-editions-express

(Untested information: Download Link for SQL Server 2017 Express LocalDB:)

https://go.microsoft.com/fwlink/?LinkID=799012

(Untested information: How to install the database management system:)

https://www.microsoft.com/en-us/sql-server/developer-get-started/csharp/win/

(Untested information: How to write data into the database:)

https://www.microsoft.com/en-us/sql-server/developer-get-started/csharp/win/step/2.html