

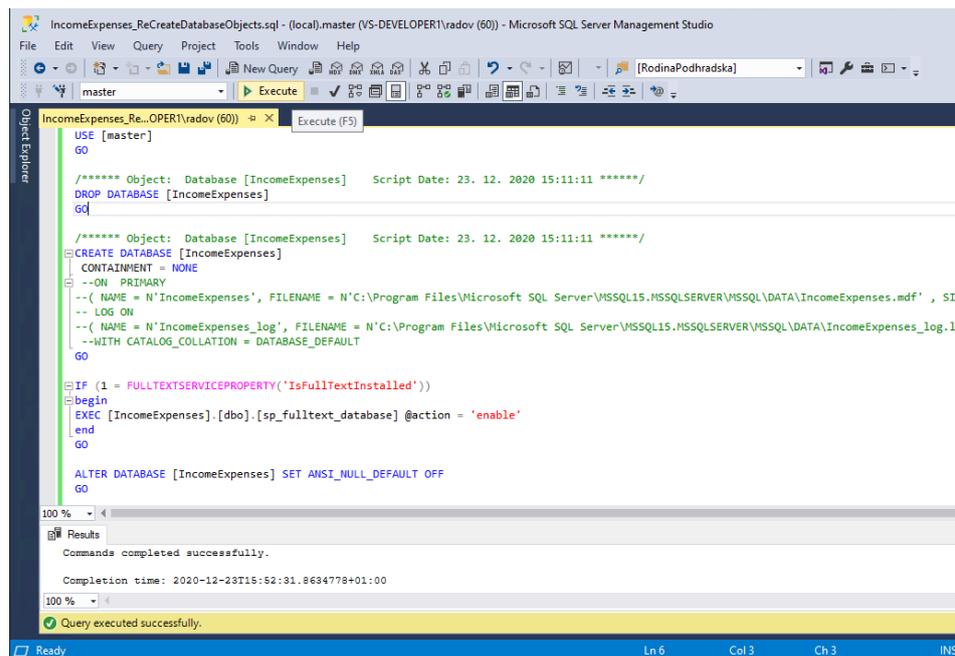
## SQL import for Tatrabanka XML export

### 1. Create SQL database

Create IncomeExpenses database with SQL script file IncomeExpenses\_CreateDatabase.sql using:

#### a) SQL server management studio

In this case simply run file IncomeExpenses\_CreateDatabase.sql included into SqlServerImportTatrabankaXml archive in SQL server management studio environment.



#### b) Sqlcmd utility

You can use sqlcmd utility to run SQL script. Sqlcmd.exe is installed into Client SDK of SQL Server, for example SQL Server 2019 default installation installs this file into directory C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Binn

##### a. Run **SQLCMD** utility from command prompt

```
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Binn>sqlcmd
```

##### b. Log into SQL server instance with **:connect** command, you can omit user name and password from it's parameters in case of use trustor connection for your Windows account on destination SQL server instance

```
1> :connect (local)
Sqlcmd: Successfully connected to server '(local)'.
```

##### c. Add SQL script content into sqlcmd execution cache and execute it with **:r** command

```
1> :r "C:\Users\radov\Documents\SQL Server Management Studio\IncomeExpenses\IncomeExpenses_ReCreateDatabaseObjects.sql"
Changed database context to 'master'.
Changed database context to 'IncomeExpenses'.
```

## **Caution!**

After script completion restart please SQL server instance. My experience says, user defined types of SQL server used in stored procedures works properly after SQL server service restart .

### 2. IncomeExpenses .NET Application

#### 2.1. Download [IncomeExpenses.zip](#) archive with:

IncomeExpenses_CreateDatabase	Microsoft SQL script to create database and it's objects
IncomeExpenses.exe	.NET application
DatabaseWriterDescriptorSale.xml	.NET application SQL import configuration description XML file
Podhradsky2020.xml	Tatrabanka account movements exported XML file

#### 2.2. Place Tatrabanka XML export into IncomeExpenses directory

Place XML export file downloaded from Tatrabanka internet banking application into directory, to which you copy

#### 2.3. Set your Microsoft SQL server connection string

DatabaseWriterDescriptorSale.xml file contains two versions of connection string

Integrated Security = True	To run .NET application on same server as host database or in Windows workgroup or domain security policy, use Windows authentication to database
Integrated Security = False	To run .NET application on same server as host database or in Windows workgroup or domain security policy, use SQL server authentication to database with User and password in XML file, as XML file comment in downloaded file

#### 2.4. Run IncomeExpenses .NET Application

IncomeExpenses.exe TatrabankaExport.xml ImportDescriptor.xml

Command line parameters:

TatrabankaExport.xml	Microsoft SQL script to create database and it's objects
ImportDescriptor.xml	.NET application SQL import configuration description XML file, must be in format downloaded in IncomeExpenses.zip file, only possible change is in XML leaf <b>DatabaseConnection</b> at top of the file, see

Command line parameters and application name are separated with space character. Application find both XML files in same directory as it's located (application working directory).

```
C:\Users\radovan.podhradsky.AF-PREDAJ\Documents\Gifts\IncomeExpenses>IncomeExpenses Podhradsky2020.xml DatabaseWriterDescriptorSale.xml
```

### 3. IncomeExpenses .NET Application Logs

Application logs these informations into command line window:

Application State	Information in console application window
Loaded Descriptor Configuration	Result of load of XML SQL import descriptor file, contains customized DatabaseConnection leaf
Database Connection	Result of connection to SQL server configured in DatabaseConnection leaf of descriptor configuration file
Result of each account movements XML table import	Number of successfully imported files into SQL server, otherwise exception reason.
Overall result of XML import	TRUE – import succeeded FALSE – import failed