

ADO.NET – Transakce

**Centrum pro virtuální a moderní metody a formy vzdělávání na
Obchodní akademii T.G. Masaryka, Kostelec nad Orlicí**



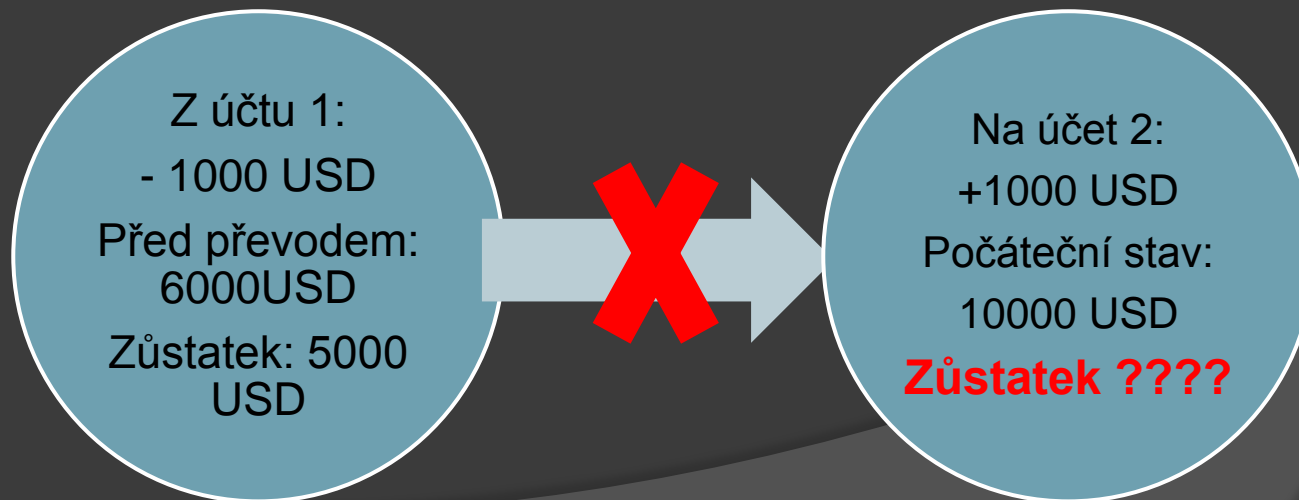
Proč transakce?



- Udržují integritu při modifikaci dat
- Úspěšná transakce musí splňovat **ACID**:
 - Atomicity
 - Consistency
 - Isolation
 - Durability

[ACID on wikipedia](#)

- Představte si manipulaci s platbou mezi dvěma účty :





Transakce

- ⦿ Začni transakci - **Begin**
 - *Něco dělej*
 - *Zase něco udělej*
 - *A zase*
- ⦿ Podařilo se všechno ?

Ano – potvrd' změny
- **commit**

NE – vrat' vše do
původního stavu -
rollback



Druhy transakcí

⦿ *Implicitní*

- Každý SQL příkaz je transakcí

⦿ *Explicitní*

• **Lokální :**

- nad jednou DB

• **Distribuovaná :**

- nad více DB
- Používá dvoufázový Commit (Transakční koordinátor a resource manager)



Local Transactions

- ⦿ Lepší výkon než distribuované
- ⦿ ***object Transaction***
- ⦿ Metody:
 - *BeginTransaction()*
 - *Commit()*
 - *Rollback()*
- ⦿ Savepoints:
 - *tran.Save("Product")*

Ukázka zdrojového kódu



```
public void RunOracleTransaction(string myConnString)
{
    OracleConnection myConnection = new OracleConnection(myConnString);
    myConnection.Open();
    OracleCommand myCommand = new OracleCommand();
    OracleTransaction myTrans;
    // Start a local transaction
    myTrans = myConnection.BeginTransaction(System.Data.IsolationLevel.ReadCommitted);
    // Assign transaction object for a pending local transaction
    myCommand.Transaction = myTrans;
    myCommand.Connection = myConnection;
    try
    {
        myCommand.CommandText = "INSERT INTO Test.Dept (DeptNo, DName) Values (50, 'DEVELOPMENT')";
        myCommand.ExecuteNonQuery();
        myCommand.CommandText = "INSERT INTO Test.Dept (DeptNo, DName) Values (60, 'PRODUCTION')";
        myCommand.ExecuteNonQuery();
        myTrans.Commit();
        Console.WriteLine("Both records are written to database.");
    }
    catch (Exception e)
    {
        myTrans.Rollback();
        Console.WriteLine(e.ToString());
        Console.WriteLine("Neither record was written to database.");
    }
    finally
    {
        myConnection.Close();
    }
}
```



Step by step

- ⦿ Vytvořte novou konzolovou aplikaci s názvem

TransactionExample_01

- ⦿ Využijeme znalostí předchozích kapitol k vytvoření připojení k DB,
(01_ConnectionDemo)



Vytvoříme spojení s DB

- Do metody **Main** sekce **try** vepište následující kód:

```
//vytvoříme spojení s DB
using (SqlConnection cnn = new SqlConnection())
{
    cnn.ConnectionString = @"Data Source=.sqlexpress; Initial Catalog=northwind;Integrated Security=true";
    cnn.Open();
    //vytvořím objekt Command
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = cnn;
```


Transakce



```
//vytvořím objekt transakce
SqlConnection myTrans;
myTrans = cnn.BeginTransaction();
cmd.Transaction = myTrans;

cmd.CommandText = "UPDATE products SET UnitPrice = UnitPrice * 1.1 WHERE CategoryId=1";
Console.WriteLine("První Command");
Console.WriteLine("Records affected {0}", cmd.ExecuteNonQuery());
Console.WriteLine();

cmd.CommandText = "UPDATE products SET UnitPrice = UnitPrice * 1.2 WHERE CategoryId=2";
Console.WriteLine("Druhý Command");
Console.WriteLine("Records affected {0}", cmd.ExecuteNonQuery());
Console.WriteLine();

Console.WriteLine("Chcete potvrdit změny v DB ? Y/N");
Console.WriteLine();

if (Console.ReadLine().ToLower() == "Y".ToLower())
{
    myTrans.Commit();
    Console.WriteLine("Transakce potvrzena");
}
else
{
    myTrans.Rollback();
    Console.WriteLine("Transakce vrácena zpět");
}
```



- ◎ Sekce **catch** metody **Main** pouze odchyťí případnou chybu v kódu

```
catch (SQLException ex)
{
    Console.WriteLine(ex.Number + " " + ex.Message);
    foreach (SqlError item in ex.Errors)
    {
        Console.WriteLine(item.Number + " " + item.Class + " " + item.Message);
    }
}
```



Závěr

- Otevřete tabulku **products** v DB **Northwind** na **SQL Serveru** a ověřte, že Vámi vytvořený program funguje korektně.
- Vylepšení změňte řešení na odrolování transakce tak aby k němu došlo pomocí vyhození výjimky:
throw new Exception("Uživatелеm přerušená transakce");
- V sekci **catch** se provede **rollback**