BLAGA Alexandru, grupa 1A, anul 3
DĂNILĂ Ionuț, grupa 1A, anul 3

COMPARAȚIE

ADO.NET vs. Entity Framework *(tranzacții)*

ADO.NET

- În **ADO.NET** lansarea unei tranzacţii se face prin apelul metodei **BeginTransaction()** din obiectul **IDbConnection**, returnând un obiect de tipul **IDbTransaction**;
- Metoda **Commit()** identifică faptul că tranzacţia a reuşit şi că modificările trebuie salvate în baza de date;
- Metoda **Rolback()** indică faptul că tranzacţia nu a reuşit, iar modificările trebuie anulate;
- Pentru salvarea stării într-un anumit punct se utilizează metoda **SqlTransaction.Save()**;
- Revenirea la starea salvată se face prin **SqlTransaction.Rollback()**;
- Adăugarea operaţiilor la lista tranzacţiei se face explicit prin setarea **IDbCommand.Transaction** cu obiectul **IDbTransaction** rezultat din urma apelului metodei **BeginTransaction()** asupra obiectului **Connection**;

Entity Framework (EF)

- Se folosește o instanță a clasei **EntityConnection**, care reprezintă o cale spre conexiunea la baza de date;
- Metoda **EntityConnection.Open()** apelează **Connection.Open()** din **ADO.NET,** iar **EntityConnection.Close()** închide conexiunea;
- Obiectul **ObjectContext** execută o cerere, se creează intern un obiect **EntityConnection** și un obiect **EntityCommand**, apoi se execută comanda;
- Eliberarea unui obiect **ObjectContext** are ca efect eliberarea conexiunii la baza de date. **ObjectContext** poate fi eliberat în mod explicit sau preluat de GC;
- Pentru a realiza rollback se folosește o componentă din **System.Transaction**, numită **Windows Distributed Transaction Coordinator (DTC)**, care supervizează acțiunile care sunt mult mai complexe decât **DbTransaction**;
- Crearea unei noi tranzacții se poate face prin folosirea unui obiect **TransactionScope**.
- Putem folosi o tranzacție pentru citire sau scriere în baza de date, se va lucra cu **ObjectContext** și **EntityClient**.

*Utilizarea **tranzacțiilor***

```
string conString =
(string)ConfigurationManager.ConnectionStrin
gs["northwind"].ConnectionString;
SqlConnection con = new
SqlConnection(conString);
SqlCommand cmd1 = new SqlCommand(
"insert INTO Customers (CustomerID,
ContactName, CompanyName)VALUES
('JHAPP','John','Apple')" , con);
SqlCommand cmd2 = new SqlCommand(
"INSERT INTO Customers (CustomerID,
ContactName,CompanyName) VALUES
('INMAR','Ion','Marul')" ,con);
SqlTransaction tran = null;
try{ con.Open();
tran = con.BeginTransaction();
//încep tranzacţia, specific ce
comenzi fac parte din lista de
operaţii care compun tranzacţia
cmd1.Transaction = tran;
cmd2.Transaction = tran;
cmd1.ExecuteNonQuery();
cmd2.ExecuteNonQuery();
tran.Commit();//commit
Console.WriteLine("Tranzactie reusita");}
catch (Exception ex){
Console.WriteLine("Tranzactie
nereusita, cauza:" + ex.Message);
tran.Rollback();}
finally{con.Close();}
```

- Modificările realizate în entitățile de context nu pot fi "roll back". O posibilitate este de a reîmprospăta datele apelând metoda **ObjectContext.Refresh()**. Putem, de asemenea, crea un nou obiect pentru a reîmprospăta datele.

*Utilizarea **EntityConnection***

```
<connectionStrings>
<add name="BreakAwayEntities"
connectionString=
"metadata=res://*/BAModel.csdl|res://
*/BAModel.ssdl|res://*/BAModel.msl;
provider=System.Data.SqlClient;
provider connection string='Data
Source=myserver;
Initial Catalog=BreakAway;
Integrated Security=True;
MultipleActiveResultSets=True'"
providerName="System.Data.EntityClient"/>
</connectionStrings>
```

*Utiliarea **tranzacțiilor***

```
using (var connection = new
EntityConnection("name=BAEntities"))
{connection.Open();
EntityTransaction transaction =
connection.BeginTransaction(Isolation
Level.ReadUncommitted);
var command = connection.CreateCommand();
command.CommandText = "SELECT
c.contactID FROM BAEntities.Contacts AS c";
var dataReader = command.ExecuteReader
(CommandBehavior.SequentialAccess |
CommandBehavior.CloseConnection);
while (dataReader.Read())
{//do something with the data;}
transaction.Commit();}
```