



ADO.NET: Les transactions

Les transactions (1)

- Une transaction sur une base de données est une suite d'opérations qui font passer la base de données d'un état A à un état B.
- Habituellement, une transaction nécessite plusieurs opérations. La suite de ces opérations est indivisible. Lorsqu'une des opérations échoue pour une raison quelconque, toutes les opérations précédentes qui ont été exécutées doivent également échouer (*rollback*).

Les transactions (2)

o Donner un exemple

Les transactions ADO.NET

- En ADO.NET, il existe essentiellement deux méthodes pour gérer des transactions.
- La 1^{ère} manière est surtout utilisé pour le mode connecté.
- La 2^{ième} manière est utilisé autant pour le mode connecté que le mode déconnecté.

Première méthode

```
SqlTransaction maTransaction = new SqlTransaction(); // Création de la transaction
maConnexion.Open();
maConnexion.BeginTransaction(maTransaction); // Démarrage de la transaction sur la connexion
try
{
    ...
    maCommande1.Transaction = maTransaction; // Cette commande utilise la transaction
    maCommande1.ExecuteNonQuery();
    ...
    maCommande2.Transaction = maTransaction; // Celle-ci aussi
    maCommande2.ExecuteNonQuery();
    ...
    maCommandeN.Transaction = maTransaction; // Celle-ci aussi
    maCommandeN.ExecuteNonQuery();

    maTransaction.Commit(); // Toutes les opérations ont été réussies
}
catch (Exception ex)
{
    maTransaction.Rollback(); // Échec de la transaction. On annule toutes les opérations.
}
maConnexion.Close();
```

Deuxième méthode (connecté)

```
using (TransactionScope maTransaction = new TransactionScope())
{
    maConnexion.Open();
    try
    {
        ...
        maCommande1.ExecuteNonQuery();
        ...
        maCommande2.ExecuteNonQuery();
        ...
        maCommandeN.ExecuteNonQuery();

        maTransaction.Complete(); // Toutes les opérations ont été réussies
    }
    catch (Exception ex)
    {
        // La transaction n'a pas été complétée. Toutes les opérations ont été annulées.
    }
    maConnexion.Close();
}
```

Deuxième méthode(Déconnecté)

```
using (TransactionScope maTransaction = new TransactionScope())
{
    try
    {
        // On suppose ici que toutes les opérations qui sont à faire
        // dans la vraie base de données concernent la même transaction
        tableAdapterManager.UpdateAll(bdVoyagesDataSet);
        maTransaction.Complete(); // Toutes les opérations ont été réussies
    }
    catch (Exception ex)
    {
        // La transaction n'a pas été complétée. Toutes les opérations ont été annulées.
    }
}
```