

Utiliser les transactions

Une transaction est un élément utilisé dans une base de données afin de définir qu'un ensemble d'opérations doit être exécuté de manière atomique. Le groupe de requêtes est considéré comme indivisible, de telle sorte que si l'une d'elles échoue, la totalité est annulée et la base retrouve l'état qu'elle avait avant le début de la transaction. Cette **opération d'annulation** est souvent nommée **Rollback**, tandis que l'opération de **validation finale** résultant de l'exécution correcte de chacune des requêtes est appelée **Commit**.

Les transactions sont donc très souvent utilisées lorsque plusieurs requêtes de modification de données doivent être effectuées mais que l'échec de l'une d'elles entraîne une incohérence des données. Un exemple pourrait être la gestion des réservations dans un avion ou un train. Lorsque toutes les places sont réservées, les compagnies aériennes maintiennent une liste d'attente pour pouvoir revendre aisément une place pour laquelle un passager annule sa réservation. Au moment de l'annulation, on se retrouve donc avec deux actions à effectuer :

- Annulation de la réservation du passager A sur le vol 1234, siège 45.
- Création de la réservation pour le client B, premier sur la liste d'attente, sur le vol 1234, siège 45.

Si la première opération échoue mais que la seconde se passe correctement, l'état des réservations sur ce vol devient incohérent. En effet, deux passagers auront une réservation valide pour le même vol et le même siège, tandis que tous les autres sièges seront occupés, ce qui posera probablement quelques soucis aux passagers et à la compagnie aérienne.

Avec ADO.NET, une transaction est utilisée au niveau de la connexion par laquelle transitent les commandes que l'on souhaite grouper. L'appel de la méthode `BeginTransaction` d'un objet `SqlConnection` renvoie un objet de type `SqlTransaction` qui encapsule l'utilisation des transactions en base de données. Cette transaction peut ensuite être fournie aux commandes par l'intermédiaire de leur propriété `Transaction`. En cas d'erreur lors de l'exécution d'une des requêtes, il faut appeler la méthode `Rollback` de l'objet `SqlTransaction`. Lorsque toutes les requêtes ont été exécutées avec succès, la méthode `Commit` de l'objet `SqlTransaction` valide définitivement les modifications apportées à la base de données.

```
private void UtilisationTransaction()
{
    SqlConnection connexion = new
SqlConnection("Server=(LocalDb)\\v11.0;Initial
Catalog=ReservationsAvion;Integrated Security=True");

    SqlTransaction transaction = connexion.BeginTransaction();

    try
    {
        SqlCommand commandeAnnulation =
connexion.CreateCommand();
        commandeAnnulation.CommandText = "DELETE FROM RESERVATION
WHERE NumeroVol=1234 AND NumeroSiege=45";
        commandeAnnulation.Transaction = transaction;

        SqlCommand commandeReservation =
connexion.CreateCommand();
        commandeAnnulation.CommandText = "INSERT INTO RESERVATION
(NumeroVol, NumeroSiege, IdentifiantClient) VALUES (1234, 45,
4765169";
        commandeReservation.Transaction = transaction;
```

```

        commandeAnnulation.ExecuteNonQuery();
        commandeReservation.ExecuteNonQuery();

        transaction.Commit();
    }
    catch (Exception ex)
    {
        transaction.Rollback();
    }

    connexion.Close();
}

```

Ce code peut également être réécrit de la manière suivante en utilisant des structures using.

```

private static void UtilisationTransaction()
{
    using (SqlConnection connexion = new
SqlConnection("Server=(LocalDb)\\v11.0;Initial
Catalog=ReservationsAvion;Integrated Security=True"))
        using (SqlTransaction transaction =
connexion.BeginTransaction())
        {
            SqlCommand commandeAnnulation = connexion.CreateCommand();
            commandeAnnulation.CommandText = "DELETE FROM RESERVATION
WHERE NumeroVol=1234 AND NumeroSiege=45";
            commandeAnnulation.Transaction = transaction;

            SqlCommand commandeReservation = connexion.CreateCommand();
            commandeAnnulation.CommandText = "INSERT INTO RESERVATION
(NumeroVol, NumeroSiege, IdentifiantClient) VALUES (1234, 45,
4765169";
            commandeReservation.Transaction = transaction;

            commandeAnnulation.ExecuteNonQuery();
            commandeReservation.ExecuteNonQuery();

            transaction.Commit();
        }
}

```

En cas d'erreur, l'instruction Rollback est appelée par la méthode Dispose de l'objet SqlTransaction. Celle-ci est appelée implicitement par la structure using.