

ADO.NET kullanarak Stored Procedure (SP) ve Transaction (Transaksiyon) kullanarak bir para transferi işlemi örneği:

Para Transferi İşlemi Örneği

Bu örnekte, TransferMoney adında bir Stored Procedure kullanarak, iki hesap arasında para transferi yapacağız. Ayrıca bu işlemi bir transaksiyon içinde gerçekleştireceğiz.

Veritabanı Hazırlığı

Öncelikle, aşağıdaki gibi bir SQL Server veritabanı tablosu ve Stored Procedure oluşturun:

1-Hesaplar (Accounts) Tablosu:

```
CREATE TABLE Accounts (  
    AccountId INT PRIMARY KEY,  
    AccountName NVARCHAR(100),  
    Balance DECIMAL(18, 2)  
);  
  
INSERT INTO Accounts (AccountId, AccountName, Balance)  
VALUES (1, 'Hesap A', 1000.00),  
      (2, 'Hesap B', 500.00);
```

2-TransferMoney Stored Procedure:

```
CREATE PROCEDURE TransferMoney  
    @FromAccountId INT,  
    @ToAccountId INT,  
    @Amount DECIMAL(18, 2)  
AS  
BEGIN  
    SET NOCOUNT ON;  
    DECLARE @FromBalance DECIMAL(18, 2);
```

```
DECLARE @ToBalance DECIMAL(18, 2);
```

Hesap bakiyelerini al

```
SELECT @FromBalance = Balance
```

```
FROM Accounts
```

```
WHERE AccountId = @FromAccountId;
```

```
SELECT @ToBalance = Balance
```

```
FROM Accounts
```

```
WHERE AccountId = @ToAccountId;
```

Bakiyeleri güncelle

```
UPDATE Accounts
```

```
SET Balance = @FromBalance - @Amount
```

```
WHERE AccountId = @FromAccountId;
```

```
UPDATE Accounts
```

```
SET Balance = @ToBalance + @Amount
```

```
WHERE AccountId = @ToAccountId;
```

```
END;
```

C# Kodu

Aşağıdaki C# kodu, yukarıda oluşturduğumuz Stored Procedure'ü kullanarak bir para transferi işlemi gerçekleştirir ve bunu bir transaksyon içinde yapar.

```
using System;
```

```
using System.Data;
```

```
using System.Data.SqlClient;
```

```
public class Program
```

```
{
```

```
    // Veritabanı bağlantı dizesi
```

```
private static string connectionString = "Server=myServerAddress;Database=myDataBase;User  
Id=myUsername;Password=myPassword;";
```

```
public static void Main()
```

```
{
```

```
    int fromAccountId = 1; // Transferi yapacak hesap
```

```
    int toAccountId = 2; // Transfer edilecek hesap
```

```
    decimal transferAmount = 200.00M; // Transfer edilecek miktar
```

```
    try
```

```
    {
```

```
        using (SqlConnection connection = new SqlConnection(connectionString))
```

```
        {
```

```
            connection.Open();
```

```
            Console.WriteLine("Bağlantı açıldı.");
```

```
            // Transaksiyon başlatma
```

```
            SqlTransaction transaction = connection.BeginTransaction();
```

```
            Console.WriteLine("Transaksiyon başlatıldı.");
```

```
            try
```

```
            {
```

```
                // Stored Procedure çağırısı
```

```
                using (SqlCommand command = new SqlCommand("TransferMoney", connection))
```

```
                {
```

```
                    command.CommandType = CommandType.StoredProcedure;
```

```
                    command.Transaction = transaction;
```

```
                    // Stored Procedure parametreleri
```

```
                    command.Parameters.AddWithValue("@FromAccountId", fromAccountId);
```

```
                    command.Parameters.AddWithValue("@ToAccountId", toAccountId);
```

```
command.Parameters.AddWithValue("@Amount", transferAmount);

// Stored Procedure çalıştırma

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

    Console.WriteLine("Para transferi başarıyla gerçekleştirildi.");

}

else

{

    throw new Exception("Para transferi işlemi sırasında bir hata oluştu.");

}

}

// Transaksiyonu tamamlama

transaction.Commit();

Console.WriteLine("Transaksiyon başarıyla tamamlandı.");

}

catch (Exception ex)

{

    Console.WriteLine("Hata: " + ex.Message);

    try

    {

        // Hata durumunda transaksiyonu geri al

        transaction.Rollback();

        Console.WriteLine("Transaksiyon geri alındı.");

    }

}
```

```

        catch (Exception rollbackEx)
        {
            Console.WriteLine("Rollback hatası: " + rollbackEx.Message);
        }
    }

    finally
    {
        // Bağlantıyı kapatma
        connection.Close();

        Console.WriteLine("Bağlantı kapatıldı.");
    }
}

catch (Exception ex)
{
    Console.WriteLine("Bağlantı hatası: " + ex.Message);
}
}
}

```

Açıklamalar

*TransferMoney Stored Procedure, @FromAccountId, @ToAccountId ve @Amount parametrelerini alarak veritabanında para transferi işlemini gerçekleştirir.

*C# kodu içinde SqlConnection, SqlCommand ve SqlTransaction sınıfları kullanılarak işlemler gerçekleştirilir.

*BeginTransaction() metodu ile transaksion başlatılır, Commit() metodu ile transaksion tamamlanır veya Rollback() metodu ile geri alınır.

*Her işlem sonucunda konsola bilgi mesajları yazdırılır.

Bu örnekte, ADO.NET kullanarak SQL Server veritabanında bir para transferi işlemi gerçekleştirdik ve bu işlemi bir transaksyon içinde yönettik. Veritabanı bağlantı dizesi ve diğer parametreleri kendi ortamınıza göre ayarlamalısınız (Server, Database, User Id, Password, AccountId vb.).