

Techniques avancées de bases de données

Implémentation

PREMIÈRE SÉANCE

Introduction

- Application web: quels challenges pour l'accès aux données?

Objectifs

- Exploiter une base de données par une application cliente et par là découvrir les problématiques liées à cette exploitation:
 - Accès aux données
 - Transactions
 - Accès concurrents
 - Impedance mismatch (relationnel/objet)

Organisation

- Création d'un projet ASP.NET MVC selon modalités distribuées séparément
- 4 séances
- Examen oral:
 - Bonne compréhension des problématiques et techniques de résolution
 - Présentation et défense du projet

Modalités d'évaluation

- Examen en janvier 2016
- Délivrables:
 - Remise du projet pour le 5 janvier par voie électronique
samuel.scholtes@henallux.be
 - Pas besoin de manuel utilisateur
- Déroulement de l'examen
 - Vérification de la compréhension des concepts, problématiques, techniques de résolution abordés durant les cours, à travers des questions/réponses, demandes d'exemples, situations problèmes
 - Code review de votre projet
 - Vérification de la maîtrise du projet par les différents intervenants

COUCHE D'ACCÈS AUX DONNÉES

Etape 1: création d'une base de données

- Sur une instance de SQL Server, créez une base de données dont la structure est la suivante.
- Tables:
 - Student
 - Id (long, auto increment, not nullable)
 - Birthdate (date, nullable)
 - FullName (string, not nullable)
 - Remark (string, nullable)
 - Course
 - Id (long, auto increment, not nullable)
 - Description (string, not nullable)
 - StudentCourse: cours suivis par l'étudiant
 - StudentId (long, not nullable)
 - CourseId (long, not nullable)
 - + FK's vers Student et Course
- Nom d'instance: (LocalDb) => v11.0
- Nom de la db: au choix
- Initialisez votre base de données en créant des étudiants, des cours et des associations entre ceux-ci.
- Un des étudiants au moins doit avoir une valeur NULL pour l'attribut Remark. Un autre doit avoir une date de naissance inconnue

Etape 2: exploitation de la base de données

Créez une application console en C# qui liste l'ensemble des étudiants. Pour chacun, imprimez:

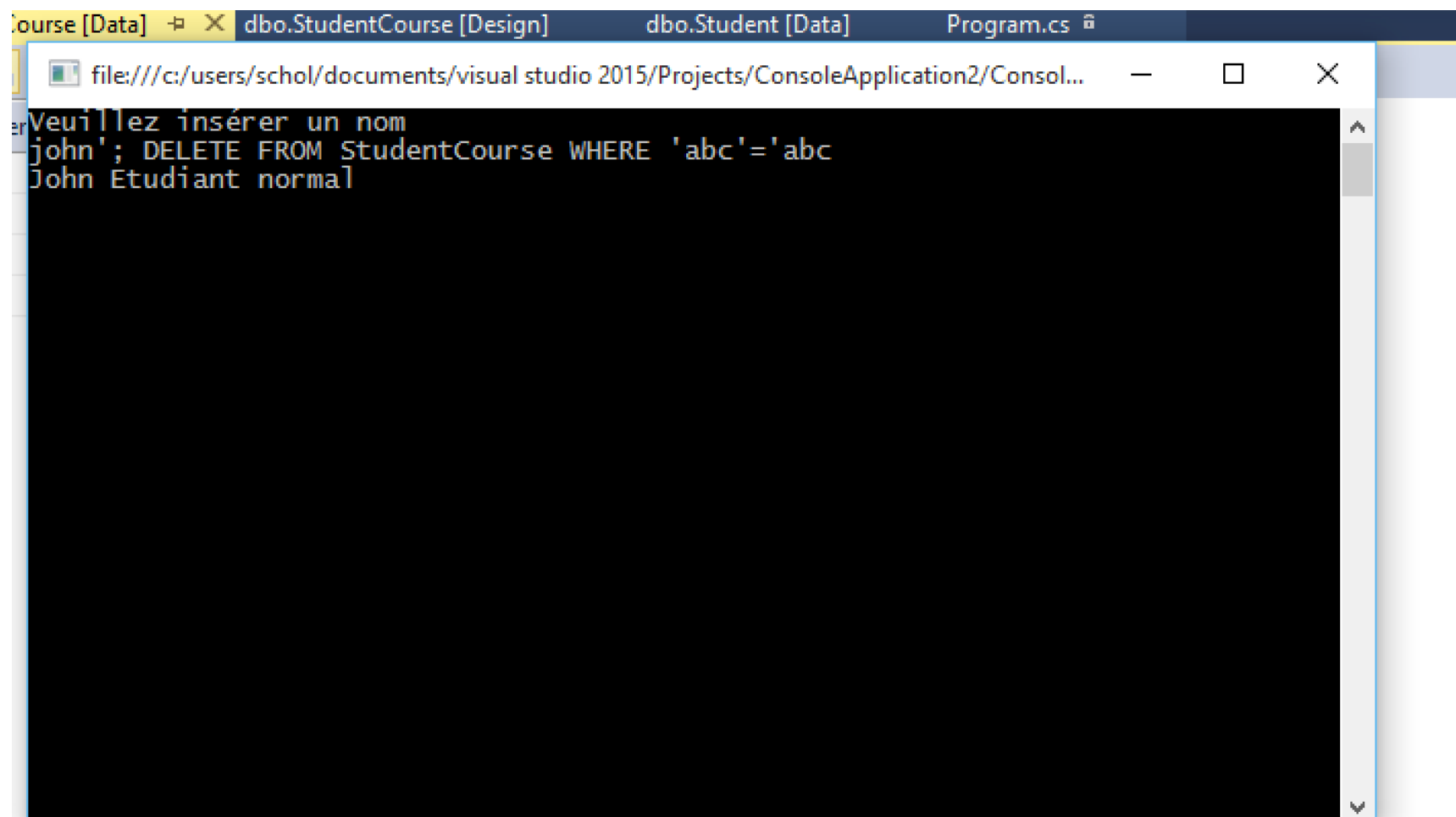
- Nom
- Date de naissance
- Remarque
- L'intitulé de chaque cours auquel l'étudiant est inscrit.
- Remarque: le nom, la date de naissance et la remarque de l'étudiant ne peuvent apparaître qu'une fois.
- Utilisez ADO.NET pour accéder à la base de données (SqlConnection, ...).

Observations et conclusions

- Problématique des jointures (mise à plat)
- Problématique des valeurs NULL
- Paramétrisation des requêtes SQL

Paramétrisation des requêtes SQL?

```
private static void SearchInDatabase(string searchFilter)
{
    SqlConnection cn = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Cat
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = cn;
    cmd.CommandText = "SELECT * FROM Student WHERE FullName='" + searchFilter + "'";
    cn.Open();
    var reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        Console.WriteLine(reader["FullName"] + " " + reader["Remark"]);
    }
    cn.Close();
}
```



DEUXIÈME SÉANCE

Agenda

- Accès concurrents
- Présentation du projet
- Réalisation des maquettes d'écran

Accès concurrents

- Réalisez les manipulations décrites dans le document disponible sur Claroline

Accès concurrents

- Observations, impressions
- C'est ce qu'on nomme le phénomène de "Lost Update". C'est un des cas problématiques rencontré dans le cadre des éditions concurrentes.

Accès concurrents

- Patterns de résolution
 - Pessimistic Offline Lock (<http://martinfowler.com/books/eea.html>)
 - Optimistic Offline Lock (<http://martinfowler.com/books/eea.html>)
- Quelques ressources:
 - <http://www.asp.net/mvc/overview/getting-started/getting-started-with-ef-using-mvc/handling-concurrency-with-the-entity-framework-in-an-asp-net-mvc-application>
 - <http://msdn.microsoft.com/en-us/data/jj592904.aspx>

Accès concurrents

- Essayez de modéliser dans vos tests unitaires cette situation de deux processus qui éditent la valeur du compte en banque de manière concurrente. Inspirez-vous des manipulations réalisées via le client WPF.
- Cette modélisation constituera un test automatisé et réparable qui nous assurera que les éditions concurrentes sont bien détectées.

Accès concurrents

- A l'aide de l'article suivant, implémentez l'optimistic offline lock pour dans votre projet: <http://www.asp.net/mvc/overview/getting-started/getting-started-with-ef-using-mvc/handling-concurrency-with-the-entity-framework-in-an-asp-net-mvc-application>

Présentation du projet

- Prenez connaissance de l'énoncé sur Caroline
- Posez vos questions