
EXAMEN TECHNOLOGIES .NET

Mercredi 8 Janvier 2014 – Durée 2h
Aucun document autorisé

Partie 1 : Questions de Cours (8 points)

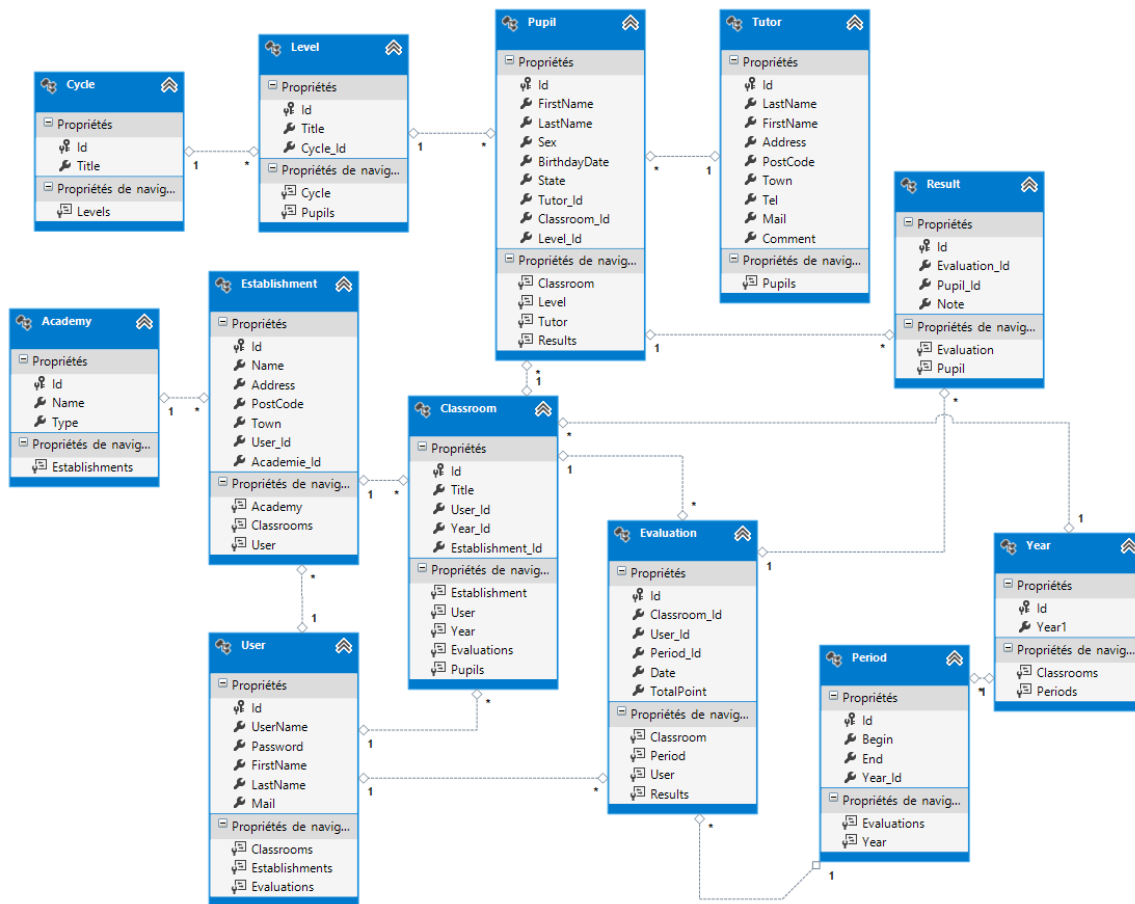
- 1- Citez 4 couches de l'architecture .NET. (0.5 point)
- 2- Donnez une définition de LINQ ? Quelles sont ses différentes fonctionnalités ? (1 point)
- 3- Quelle classe permet de gérer la connexion à une base de données avec Entity Framework ? (0.5 point)
- 4- Qu'est-ce qu'une classe d'extension ? Citez en une avec le nom d'une méthode souvent utilisée. (0.5 point)
- 5- Quel est le nom du fichier de configuration d'une application .NET ? Que contient-il ? (0.5 point)
- 6- Qu'est-ce qu'une annotation pour un modèle ? Citez en trois. (0.5 point)
- 7- Sur le site <http://monSiteDotNet.fr> réalisé en .NET MVC, je souhaite exécuter l'action « Read » de la classe TenderController, qui prend deux arguments : un entier « id » qui vaut 1 et un entier « Year » qui vaut 2014. Quelle adresse dois-je taper dans le navigateur pour réaliser ceci ? (0.5 point)
- 8- Quelles sont les deux objets qui permettent de transmettre des données du contrôleur à la vue sans passer par le modèle ? Donnez un exemple d'utilisation pour chacun d'eux. (0.5 point)
- 9- Quel est le nom de la méthode qui permet d'indiquer l'endroit dans le code HTML où l'on va insérer le contenu de la vue dans la page maitre ? (0.5 point)
- 10- Définissez les trois types de validation possible d'un modèle et donner un exemple pour chacun d'eux. (1,5 point)
- 11- Quelle est la classe qui permet de gérer les utilisateurs et les informations d'identification ? (0.5 point)
- 12- Définissez les deux types d'authentification dans une application .NET ainsi que leurs caractéristiques respectives. (1 point)

Partie 2 : Une application .NET de gestion de notes (12 points)

Pour cette partie, l'idée est la conception et le développement d'une application de gestion d'élèves, de professeurs, de classes et de notes qui pourraient être implantées au ministère de l'éducation nationale. Pour réaliser cette application, nous allons utiliser le modèle de données exposée à la page suivante. Ce modèle est constitué des tables suivantes :

- Users : les professeurs de l'application. Ils sont associés à un établissement (Establishment) et dispensent des enseignements à plusieurs classes (Classrooms).
- Classrooms : les classes d'élèves de l'application. Une classe est associée à une année scolaire (Year), à un établissement (Establishment), un professeur (User), et est constitué d'un ensemble d'élèves.
- Pupils : les élèves de l'application. Ils sont associés à une classe et un niveau d'études (CP, 6^{ème}, Terminal etc...).

- Evaluations : il s'agit d'un examen, associé à une classe, un professeur, une date et un total de point (10, 20...)
- Result : Il s'agit d'une note qui est associée à un examen et un élève avec une note.



Questions :

- 1- On souhaite que la page d'accueil soit la liste de tous les élèves référencés dans l'application. Pour cela vous allez créer une classe `DisplayedPupil.cs`, qui doit reprendre les champs suivants (N'oubliez pas les annotations...) (1 point) :

Nom du champ	Type	Description
Pupil_Id	Guid	Identifiant de l'élève
FirstName	String	Prénom de l'élève
LastName	String	Nom de l'élève
BirthdayDate	DateTime	Date de Naissance de l'élève
Tutor_Id	Guid	Identifiant du tuteur
Tutor_FirstName	String	Prénom du tuteur
Tutor_LastName	String	Nom du tuteur
Classroom_Id	Guid	Identifiant de la classe
Classroom_Title	String	Nom de la classe
Level_Title	String	Le niveau actuel de l'élève
Establishment_Id	Guid	Identifiant de l'établissement
Establishment_Title	String	Nom de l'établissement
Teacher_Id	Guid	Identifiant du professeur
Teacher_FirstName	String	Prénom du professeur
Teacher_LastName	String	Nom du professeur

- 2- Définissez quatre nouveaux champs calculés pour ce modèle (1 point) :
- FullName, Tutor_FullName et Teacher_FullName qui sont la concaténation du prénom et du nom associés.
 - Age : Renvoie l'âge de l'élève en fonction de sa date de naissance. (Date d'aujourd'hui : Datetime.Now pourrait vous être utile...)
- 3- On souhaite maintenant créer une classe dite « Repository » afin de questionner le modèle de données. Voici la classe PupilRepository :

```
public class PupilRepository : IDisposable
{
    NotesEntities entities;

    public PupilRepository(NotesEntities e) { entities = e; }

    public void Dispose() { }
}
```

Définissez trois nouvelles méthodes qui renvoient une IQueryable<Pupil> (1.5 point) :

- GetAllPupils() : Qui renvoie la liste de tous les élèves.
 - GetAllPupilsByClassroom() : Qui prend en argument l'identifiant de la classe et renvoie la liste des élèves associé à cette classe.
 - GetAllPupilsByEstablishment() : Qui prend en argument l'identifiant de l'établissement et renvoie la liste des élèves associé à cet établissement.
- 4- Nous allons maintenant créer une nouvelle classe nommé PupilManager.cs qui va permettre de faire le lien entre la classe Pupil et la classe DisplayedPupil. Voici à quoi elle ressemble :

```
public static class PupilManager
{
    public static List<DisplayedPupil> GetAllDisplayedPupils(NotesEntities entities)
    {
        return new List<DisplayedPupil>() { };
    }
}
```

Réécrivez la méthode GetAllDisplayedPupils pour qu'elle renvoie une liste de DisplayedPupils (qui est la classe que vous avez créé il n'y a pas très longtemps...). Utiliser la projection pour réaliser ceci... (1 point)

- 5- Créez maintenant un contrôleur nommé PupilController.cs, avec une action nommée « Index » qui doit renvoyer à la vue la liste de tous les élèves. (1.5 point)
- 6- Ecrivez maintenant le code de la vue « Index.cshtml » qui permet d'afficher une liste de « DisplayedPupil » à l'écran. Réalisez un simple affichage sous la forme d'un tableau. Ajoutez des liens permettant d'ouvrir les fiches des différentes entités liées à un élève. (Pour cela, on part du principe qu'il existe une méthode « Edit » dans des contrôleurs prenant un identifiant en argument) (2 points)
- 7- Pour aller un peu plus loin maintenant, on souhaite développer un module permettant la saisie des notes d'une évaluation. Pour cela, on souhaite accéder à un formulaire permettant d'ajouter toutes les notes à une évaluation déjà créée. On suppose que l'on a déjà un contrôleur nommé EvaluationController, avec une action Index qui affiche la liste de toutes les évaluations. A vous d'expliquer et d'implanter toutes les étapes nécessaires afin de développer ce module, en partant de la vue Index. Décrivez les différents points à développer et implanter le code nécessaire. (4 points)