Titre : Développement d’applications Web avancées

**Code officiel:** 420-A16-BB

Examen Final

**Programme d’études:** Programmeur analyste

**Session visée par le cours**: Session 4

### Discipline : Techniques informatiques

**Professeure** Soraya Ferdenache

|  |  |
| --- | --- |
| Examen | Partie de l’examen final |
| **Pondération** | 20% |
| **Durée** | 40mn |
| **Date** | 06/08/2021 |
| **Documents** | NON **autorisés**  SANS Accès Internet |

|  |  |
| --- | --- |
| *Nom et Num Étudiant* | ***Jonathan Tremblay***  ***2095255*** |
| **Note** |  |

1. **Choix multiples (2 pts par question)**
2. **EF Core supporte les approches** 
   1. **Code First**
   2. **DataBase First**
   3. **Model First**
   4. **Toutes les réponses précédentes**
   5. **Aucune des réponses précédentes**
3. Le code

modelBuilder.Entity<Student>()

.HasOne(s => s.Grade)  
 .WithMany(g => g.Students)

.HasForeignKey(s => s.CurrentGradeId);

**Le code ci-dessus configure la relation \_\_\_\_\_\_\_\_ entre les entités Student et Grade**

* 1. **Un à un**
  2. **Plusieurs à plusieurs**
  3. **Un à plusieurs**
  4. **Aucune des réponses précédentes**

1. **Laquelle des plates-formes suivantes, prend en charge EF Core?**
   1. **Mac**
   2. **Windows**
   3. **Linux**
   4. **Toutes les réponses**
2. **Questions à développement (4 pts)**

|  |
| --- |
| 1. Pourquoi utiliserait-on EF Core au lieu de ADO.Net (Vocabulaire technique) |
| Pour s’éviter d’écrire du code exhaustif, qui se répète dans chaque application. (boilerplate code)  Pour une maintenance plus aisée du code d’accès aux données. (DAO)  Pour prioriser / focuser son temps et ses efforts sur la logique d’affaire.  Très pratique aussi pour du prototypage. |
|  |

1. Résumer les étapes de mise en place d’une solution selon l’approche codeFirst **en utilisant le bon vocabulaire technique?**

|  |
| --- |
| **Premièrement, nous devons vérifier que les dépendances liés à EF Core sont installés dans les packages NuGet.**  **Ensuite, il suffit de créer un model (classe C#) représentant l’objet dans la base de donnée. Fields, constructeurs et propriétés seulement.  Maintenant, on crée une classe héritant de DbContext, on implémente ses méthodes, et on ajoute pour chaque modèle créé, un DbSet<Model> (par exemple pour la classe Cours 🡪   DbSet<Cours> Cours {get; set;}**  **Finalement, nous devons faire une migration et mettre à jour la base de donnée  (Add-Migration Initial et Update-Database) dans la console de package**  **Les tables dans la base de données ont étés crées selon le modèle définit plus tôt** |
|  |