

Travail pratique 2

Application console d'interaction avec la base de données

Consignes

- Le travail doit se faire en équipe de 2.
 - Il faut écrire à l'enseignant pour indiquer les membres de l'équipe avant le cours du 24 février par MIO.
 - La personne seule aura un ajustement sur le travail demandé.
- Le travail doit être fait sur **GitLab**.
 - L'utilisateur **fsthilaire-dm** doit avoir le rôle **Developper**.
- Sur **LÉA**, il faut remettre un fichier **TEXTE** qui contient les éléments ci-dessous.
 - Lien **GitLab**
 - Nom, Prénom, DA de chaque étudiant
- Le fichier **TEXTE** doit être remis dans **Léa** avant le mercredi **29 mars 2023 23:59:59**.
- La date du dernier **COMMIT** dans **GitLab** sera la date de remise du projet.

Énoncé du travail

Vous devez créer une application console pour interagir avec la base de données du magasin d'applications en ligne **FrankStore**. Le script de la base de données et les insertions des pays et des langues sont fournis. Il faut appeler la base de données **eDA_4N1_TP1**. Utilisez le DA d'un des membres de l'équipe. Les scripts de création et des insertions de base sont fournis.

L'application console doit respecter l'architecture par injection de dépendances présentée en classe. La librairie pour la communication avec la base de données doit être l'**ORM Entity Framework Core**. Vous devez utiliser la technique **Database First** pour générer votre contexte et vos modèles. Il faut que le contexte de la base de données utilise la chaîne de connexion du fichier de configuration. L'application doit utiliser le serveur départemental. L'enseignant peut vous créer un accès d'équipe.

La sélection de la fonctionnalité de l'application doit se faire par arguments.

Les interactions avec l'utilisateur doivent se faire dans les classes **Manager**.

Vous devez créer une classe utilitaire pour valider les entrées de l'utilisateur. Par exemple, si la valeur demandée doit être un entier, il faut poser la question de nouveau si l'utilisateur entre une lettre. Le **Manager** doit utiliser cet utilitaire pour optimiser les entrées utilisateurs pour être plus **DRY**.

Votre application doit afficher la liste des arguments avec l'option **-aide**. Cette option doit être par défaut.

Voici la liste des fonctionnalités.

1- Afficher un enregistrement avec identifiant de clé étrangère uniquement

Pour chacune des tables, il faut afficher un enregistrement en fonction de la clé ou les clés primaires. L'application doit demander la clé. Si aucun enregistrement n'est trouvé, il faut l'indiquer à l'utilisateur.

Il faut demander à l'utilisateur s'il veut afficher un autre enregistrement. S'il le désire, il faut lui demander la clé. S'il ne le désire pas, l'application doit se terminer.

Il faut afficher le contenu de l'enregistrement par une classe d'extension.

Pour cette fonctionnalité, si l'enregistrement à une clé étrangère, il faut indiquer le numéro.

2- Afficher un enregistrement avec le détail des clés étrangères

Il s'agit de la même option que la précédente.

Par contre, si l'enregistrement possède une clé étrangère, il faut afficher le numéro de la clé étrangère et un champ significatif de la table parent. Par exemple, pour un **client**, utilisez le champ **NomPays** pour indiquer son pays. Utilisez le **EagerLoading**.

Pensez à réutiliser la même méthode d'extension et détectez le cas de la fonctionnalité 1 et celle-ci.

3- Afficher tous les achats d'un client

Vous devez demander le numéro d'un client. Il faut afficher son information et le détail de tous ses achats. Il faut afficher le titre de l'application, le nom du mode de paiement et le nom du développeur. Il faut trier les achats en date de paiements. Il faut indiquer le total de l'achat à la fin.

Utilisez une méthode d'extension spécifique **AfficherDetailAchats()** pour le client.

Utilisez le **EagerLoading** pour récupérer l'information des autres tables à partir de la table client.

Exemple d'affichage.

Client #3 : François St-Hilaire
123 rue du Cégep
Drummondville (Canada)

2021-01-17, Nintendo, Mario Bros 3, 55.24\$, Mastercard
2022-07-05, Sega, Sonic, 22.01\$, Visa

Total : 77.25\$

4- Afficher le rapport nombre de clients et de développeur par pays

Vous devez afficher le nombre de clients et de développeurs pour chacun des pays. Vous devez demander à l'utilisateur si le tri se fait en ordre alphabétique croissant ou décroissant. Vous devez exclure les pays qui n'ont aucun client et développeur.

Vous devez créer une classe spécifique pour ce rapport et l'utiliser dans la requête **GROUP BY** de LINQ.

Utilisez une méthode d'extension sur la liste et sur la classe spécifique pour faire l'affichage. Pensez à valider le pluriel.

Voici comment la liste doit s'afficher.

Canada	15 clients	1 développeur
Maroc	1 client	5 développeurs

5- Afficher un rapport pour les applications d'un développeur.

Vous devez sélectionner un développeur et pour chacune de ses applications, il faut indiquer le nombre d'achats, le montant total des ventes et la moyenne de sa note.

Vous devez créer une classe spécifique pour ce rapport et l'utiliser dans la requête **GROUP BY** de LINQ.

Voici comment la liste doit s'afficher.

Mario Bros 3, 55 ventes, 1652.35\$, 9.8/10
Mario Bros 2, 15 ventes, 552.35\$, 7.7/10

6- Supprimer un enregistrement

Vous devez supprimer un enregistrement en fonction de sa clé primaire. En premier, il faut afficher l'information de l'enregistrement.

Si l'enregistrement est toujours utilisé par des tables enfants, il faut aviser l'utilisateur que la suppression n'est pas possible et lui indiquer le nombre d'enregistrements enfant par table. Utilisez la technique présentée en classe.

Si l'enregistrement n'est plus utilisé, il faut demander à l'utilisateur la confirmation de le supprimer.

7- Commenter et noter un achat

Vous devez sélectionner un achat et être en mesure de modifier le commentaire et sa note. Avant de le mettre à jour dans la base de données, il devra être validé.

8- Générateur de données

Pour chacune des tables (à l'exception de Pays, ModePaiement et Langue), il faut créer un générateur de données. Il faut demander à l'utilisateur le nombre d'enregistrements à générer. Assurez-vous que le nombre soit plus grand que zéro.

Avant de l'enregistrer, il faut s'assurer que les données sont valides. Si les données ne sont pas valides, il faut ignorer cet objet et en générer un nouveau. Regardez le type des colonnes et leurs contraintes.

Assurez-vous que les données générées soient réalistes. Par exemple, que le nom d'une personne utilise les méthodes de noms. S'il faut indiquer une date, il faut vérifier qu'elle ne soit pas dans le futur, que la date de l'achat d'une application doit être cohérente avec la date de création du client et de publication de l'application...

Pour les champs qui peuvent être **null**, il faut appliquer une probabilité. Les probabilités doivent être différentes pour chacun des champs **null**.

Critères d'évaluation

Le travail pratique vaut pour 35% de la note finale du cours. Ce travail est évalué sur 100 points.

Pour chacune des 8 fonctionnalités, le code doit respecter l'architecture et les techniques présentées en classe.

Évaluation	Pondération
1- Afficher un enregistrement avec identifiant de clé étrangère uniquement	5
2- Afficher un enregistrement avec le détail des clés étrangères	5
3- Afficher tous les achats d'un client	10
4- Afficher le rapport nombre de clients et de développeur par pays	15
5- Afficher un rapport pour les applications d'un développeur	15
6- Supprimer un enregistrement	10
7- Commenter et noter un achat	10
8- Générateur de données	20
Configuration de base de l'application - Enregistrement des dépendances par extension - Fichier configuration - Gestion des arguments du programme - Génération du contexte - Utilitaire des questions pour la console	5
Documentation et mise en forme du code	5
Total	100
Français : Pénalité accordée au français. -1% par faute	1 point par faute jusqu'à 15 points
Retard : 10% par jour de retard pour un maximum de 10 jours.	-10 points par jour