

# 1 Production finale d'intégration

## TiGris à l'école des sorciers....



Hiver 2021

- Ce travail sera réalisé individuellement.
- À terminer pour le 21 mai avant minuit pour tous les groupes.
- Le travail compte **pour 30 %** de la note finale.

### Objectifs :

Ce travail vise principalement à vous faire expérimenter les aspects suivants:

- À partir d'un modèle de données, créer et exploiter une base de données relationnelle.
- Normaliser une base de données de petite taille.
- Réaliser une application C# ADO.NET

### Consignes :

1. Vous devez remettre la question de la partie 1 à la date indiquée, c'est-à-dire le vendredi 07 mai avant 16 heures. Sinon vous avez zéro pour cette question
2. Vous devez télécharger le fichier CRUDlard.sql et l'enregistrer sur votre disque. (Le fichier sera disponible sur le site après la remise de la partie 1)
3. Exécuter votre fichier CRUDlard.sql pour commencer à écrire les requêtes de la partie 2
4. Vos requêtes SQL doivent être bien présentées (voir exemple), sinon elles ne seront pas corrigées. **Sachez qu'une longue requête sur une seule ligne est difficile à corriger.**

Exemple :

```
SELECT nom_acteur, prenom_acteur
FROM ((acteurs
INNER JOIN Acteurs_film ON acteurs.id_acteur= Acteurs_film.id_acteur)
INNER JOIN films ON films.id_film= Acteurs_film.id_film)
WHERE titre = 'Pirates des Caraïbes,Jusqu"au bout du monde';
```

Vous pouvez écrire vos requêtes tout en minuscules si vous le souhaitez.

---

<sup>1</sup> Source de l'image

<https://pixabay.com/fr/illustrations/ch%C3%A2teau-de-sorcellerie-vieille-maison-5683182/>

## Mise en contexte :

Nous sommes à l'école des sorciers « CRUdlard » où les élèves apprennent à devenir des sorciers par la maîtrise de certaines techniques décrites dans différents livres, ici appelés *Grimoires*. Un grimoire est un livre de magie.

Un grimoire a un identifiant, un titre et un type de grimoire.

Les élèves sont identifiés par un numéro unique, ils ont un alias unique, un nom, un prénom, ils ont un solde en écus, un nombre de points.

Les élèves appartiennent à des groupes. Dans un groupe nous avons plusieurs élèves.

Pour lire un grimoire, les élèves doivent **payer** un certain montant. Après la lecture d'un grimoire, les élèves obtiennent des **points**.

Le nombre de points et le prix de lecture dépendent du type de grimoire. Nous avons 5 types de grimoires : Air, eau, terre, feu, sort.

Après la lecture d'un grimoire, les élèves peuvent écrire un commentaire et donner une évaluation pour le grimoire. Les commentaires et les évaluations sont stockés dans une table : *evaluationgrimoire*.

De temps en temps, l'école organise des tournois entre les élèves de groupes différents. Le tournoi se déroule entre deux élèves de groupes différents. Lors d'une partie d'un tournoi un élève peut obtenir un maximum 5 points. Exemple

Lorsque l'élève Kiwi affronte Fantomas, Kiwi obtient 3 et Fantomas 2.

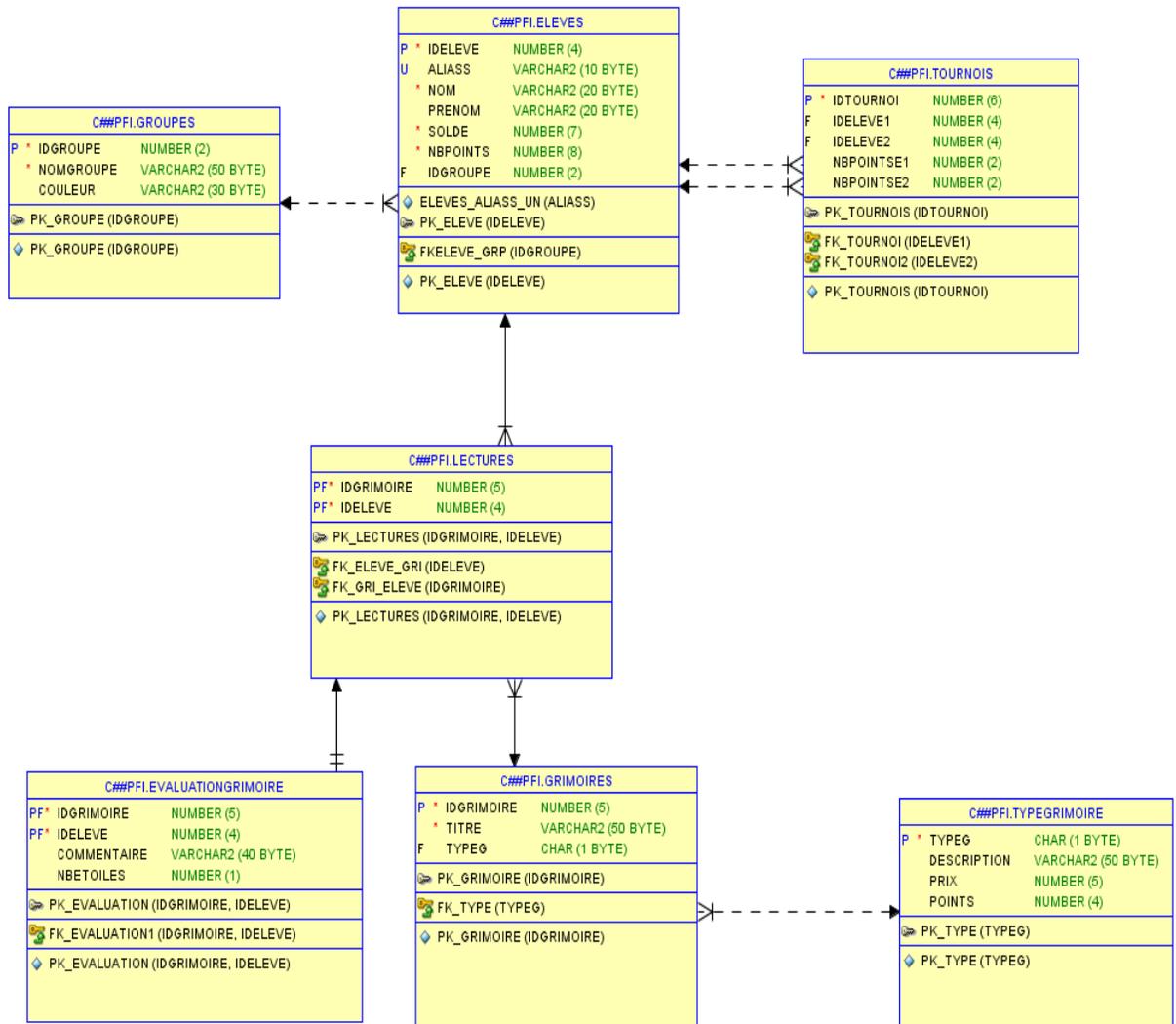
Lorsque l'élève Chubaka affronte Féroce, Chubaka obtient 4 points et Féroce 1 point.

ALIAS_ELEVE1	ALIAS_ELEVE2	NBPOINTSE1	NBPOINTSE2
Kiwi	Fantomas	3	2
Chubaka	Féroce	4	1

Finalement, le nombre total de points d'un élève est égal au nombre de total de points obtenus en lisant des grimoires plus (+) le nombre total de points obtenus dans les tournois.

Le modèle de la base de données De CRUdlard est donné dans le diagramme suivant.

## Modèle relationnel de la base de données de CRUDlard :



## Autres contraintes sur les tables :

En plus des contraintes de primary et de foreign key, voici ce que vous devez savoir pour les différentes tables :

### eleves :

Colonnes	Contraintes
idEleve	Généré par défaut
Solde	Par défaut égal à 10000
NbPoints	Par défaut à 0

### Table grimoires :

lonnes	Contraintes
IdGrimoire	Généré par défaut, commence à 100 et s'incrémente de 1

### Table evaluationgrimoire :

Colonnes	Contraintes
NbEtoiles	Est plus petit ou égal à 5. Plus grand que zéro.

### Table tournois :

Colonnes	Contraintes
IdTournoi	Généré par défaut.
NbPointsE1	Plus grand ou égal à zéro. Plus petit ou égal à 5
NbPointsE2	Plus grand ou égal à zéro. Plus petit ou égal à 5

## Travail demandé : SQL (75 points)



Partie 1 : À partir du modèle de données, créer les tables de la base de données.

Date de remise de cette partie : **le vendredi 07 mai avant 16 heures**

1. Examiner le modèle de données de CRUdlrad puis créer les tables avec toutes les contraintes. Le nom des tables doit être respecté.
  - typegrimoire
  - grimoires
  - groupes
  - eleves
  - lectures
  - evaluationGrimoire
  - tournois

Partie 2 : Écriture de requêtes SQL (exploiter une base de données). **Pour toutes les questions de cette étape, il faudra écrire les requêtes SQL.**

Télécharger le script CRUdlrad.sql puis exécuter le script pour

- a. Créer les tables
- b. Faire les différentes insertions.

1. Afficher le titre et le prix de tous les grimoires. Cette sortie est ordonnée par prix.
2. Afficher les alias des élèves ainsi que le nom de leur groupe. Cette liste doit inclure aussi les élèves qui ne sont dans aucun groupe.
3. Afficher les élèves qui n'ont lu aucun grimoire. Afficher l'alias, le nom et le prénom.
4. Afficher les alias des élèves qui sont dans le même groupe que TiGris
5. Afficher le nombre de grimoires par type de grimoire. Afficher la description ainsi que le nombre.
6. Afficher la moyenne des évaluations pour chaque grimoire. Arrondir ce chiffre à deux chiffres après la virgule. Afficher le titre et la moyenne.
7. Afficher les meilleurs grimoires selon la moyenne des évaluations. Afficher le titre du grimoire, le id du grimoire ainsi que la moyenne d'évaluation.

Vous aurez la sortie suivante.

TITRE	IDGRIMOIRE	MOYENNE
Se déplacer sur l'eau	100	4
Contrôler les rivières	101	4

8. Afficher l'alias, le nom, le prénom et le nombre d'étoiles des élèves ayant donné la même note d'évaluation au grimoire : *Marcher dans le feu*.

Vous aurez la sortie suivante :

ALIAS	NOM	PRENOM	NBETOILES
Panthère	Lavoie	Linna	5
Malin	Gagnon	Nicolas	5
Gribouille	Lapointe	Vanessa	4
PiwiPiwi	Deblanc	Racha	4

9. Afficher l'alias, le nom, le prénom, le nom du groupe et le nombre de grimoires lus des élèves ayant lu 2 grimoires ou plus. Cette sortie est ordonnée par le nombre de grimoires :

ALIAS	NOM	PRENOM	NOMGROUPE	NOMBRE_GRIMOIRE
TiGris	Courtey	Rosa	les poussins	5
Chubaka	Clément	Sylvain	les poussins	3
Malin	Gagnon	Nicolas	les poussins	2
Kiwi	Lépine	Fred	les poussins	2
Gribouille	Lapointe	Vanessa	les chevaux	2

10. Pour chaque groupe, afficher le nombre d'élèves ayant lu des grimoires. Afficher le nom du groupe et le nombre d'élèves. Vous aurez la sortie suivante :

NOMGROUPE	Nombre d'eleves
les poussins	7
les chevaux	2
les renards	1

11. Afficher le nom des groupes ayant le nombre d'élèves ayant lu des grimoires plus grand que 2. (Selon la figure précédente seul le groupe « Les poussins » sortira.)

NOMGROUPE	Nombre d'eleves
les poussins	7

12. Afficher les alias des élèves ayant fait une lecture, mais n'ayant pas fait d'évaluation.

ALIAS
Patoche
Primogene

13. Afficher le coût de lecture de TiGris : à chaque fois que TiGris fait une lecture de grimoire, il paye un montant. On vous demande le coût total de ses lectures.

ALIASS	COUT_DE_LECTURE
TiGris	390

14. Afficher le nombre total des points que TiGris a obtenu grâce à ses lectures. À chaque lecture de grimoire, TiGris obtient des points. On vous demande le total de ses points (de TiGris) :

TOTAL_POINTS
350

15. Mettre à jour le nombre de points de TiGris par le total des points obtenu grâce à ses lectures. (Vous ne devez pas mettre à jour le nombre de points directement par 350. Le nombre de points évolue)
16. Mettre à jour le solde de TiGris après ses lectures. (Après que TiGris ait payé le coût de ses lectures, quel est le solde qui lui reste?)
17. Donner le classement des élèves selon les lectures qu'ils ont réalisés es. Afficher l'alias et le total des points.

ALIASS	TOTAL_POINTS
TiGris	350
Chubaka	150
Gribouille	150
Malin	150
Panthère	100
PiwiPiwi	100
Cacao	100
Kiwi	100
Patoche	50
Primogene	50

Pour les questions suivantes, et pour le classement des élèves, on vous demande d'utiliser la table **tournois**.

18. La table tournois contient les numéros des élèves avec le résultat des tournois. On vous demande d'afficher la sortie suivante : (alias à la place des numéro)

ALIAS_ELEVE1	ALIAS_ELEVE2	NBPOINTSE1	NBPOINTSE2
Kiwi	Fantomas	3	2
Chubaka	Féroce	4	1
TiGris	Gribouille	3	2
Patoche	Barakuda	3	2
Fantomas	Kiwi	4	1
Féroce	Chubaka	4	1
Gribouille	TiGris	1	4
Barakuda	Patoche	2	3

19. Créer la vue Vtournois, qui classe les élèves dans **les tournois**. Le select \* from Vtournois donnera le résultat suivant :

TOTALPOINTS	ELEVE
7	3
6	7
6	4
5	2
5	8
4	12
4	1
3	11

20. Utiliser la vue pour afficher le classement des élèves dans les tournois. Utiliser l'alias de l'élève. Vous aurez la sortie suivante :

TOTALPOINTS	ALIAS
7	TiGris
6	Patoche
6	Fantomas
5	Féroce
5	Chubaka
4	Barakuda
4	Kiwi
3	Gribouille

21. Afficher le classement des élèves du groupes : *les poussins*. Afficher l'alias et le total des points.

ALIAS	TOTALPOINTS
TiGris	7
Patoche	6
Chubaka	5
Kiwi	4

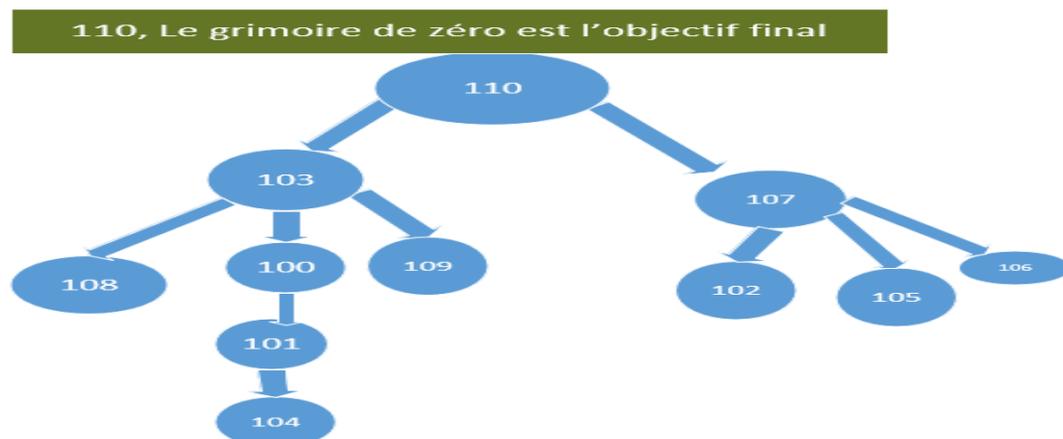
22. Afficher le nombre total des points pour chaque groupe . Afficher le nom du groupe et le total des points.

NOMGROUPE	TOTAL_POINTS
1 les poussins	22
2 les renards	11
3 les chevaux	7

23. On vous demande d'afficher le nombre total de points de TiGris obtenus grâce aux lectures et au tournoi. (Après exécution de votre requête vous obtiendrez 357.)

### Partie 3 : Compréhension

- On veut implémenter un système de cours prérequis comme suit :
  - Un grimoire a **0 ou plusieurs prérequis** (grimoires **qu'il faut avoir lus** pour pouvoir accéder à ce grimoire)
  - Un grimoire a **1 ou zéro « objectif »** (grimoire **qu'on peut atteindre** après avoir lu ce grimoire)
  - Le grimoire ayant **zéro objectif est « l'objectif final »** (une fois qu'on l'a lu, on n'atteint aucun autre grimoire). Dans notre exemple, c'est le grimoire 110.



- a. Comment allez-vous changer la table **grimoiresPlus** (voir le script plus bas) pour représenter ce système de prérequis?
- b. Donner le modèle de données qui représente cette situation.
- c. Écrire la requête qui affiche de manière hiérarchique tous les grimoires prérequis à « Parler aux arbres »

Note : Pour ces questions (a, b, c), afin de ne pas détruire la structure de votre table grimoires, utilisez la table **grimoiresPlus** créé en exécutant le script suivant :

```
create table grimoiresPlus as select * from grimoires;
alter table grimoiresPlus add constraint pk_grimoireplus
primary key(idgrimoire);
```

### Partie ADO.NET (15 points)

Vous devez réaliser une application ADO.NET qui permet d'afficher la liste des grimoires lus par un élève. L'alias de l'élève est entré dans une zone de texte, le résultat est affiché dans un DGV. Le nombre total de grimoires lus doit-être affiché également.

Votre application devra permettre l'inscription d'un élève ainsi que la modification du nombre d'écus ou de points.

**Le nom de votre usager et le mot de passe doivent être entrés dans des zones de texte.**

Il faudra vous déconnecter de manière explicite de la base de données. Toutes les ressources doivent être libérées.

Vous devez faire l'interface graphique suivante :

- Bouton '**Connexion**'  
Établit une connexion à la base de données.
- Bouton '**Quitter**'  
Ferme la connexion et quitte l'application.
- Bouton '**Rechercher**'  
Affiche dans le DataGridView la liste des grimoires lus par un élève. L'alias de l'élève doit être entré dans une zone de texte. Les colonnes qui seront affichées sont : le titre, la description, le prix et le nombre de points du grimoire.
- Bouton '**Créer**'  
Ajoute un nouvel élève avec toutes ses informations (qui seront entrées dans des zones de texte appropriées)
- Bouton '**Modifier**'  
Modifie le solde ou le nombre de points de l'élève. La modification se fait selon l'alias

PFI 2021

Nom usager: c##marcb Mot de passe: ●●●●●●●● [Connecter] [Déconnecter]

Kiwi [Rechercher]

Titre	Description	Prix	Points
Se déplacer sur l'...	Eau	70	50
Contrôler les rivi...	Eau	70	50

Résultats: 2

Alias: Bigl

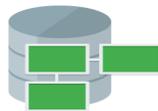
Nom: [Virgule] [Ajouter]

Prénom: [Point] [Modifier]

Solde: 123

Point: 654

## Partie Normalisation (10 points)



CRUDlard souhaite développer une petite application pour la gestion de ses films de magie. Voici certaines données que nous souhaitons stocker dans la base de données MaGilms.

- Les films ont un numéro unique, un titre, une catégorie, l'année de sortie, la durée
- Les films sont classifiés (G, 8+, 13+). Les films sont joués par des acteurs et réalisés par un réalisateur.
- Les catégories ont un code, une description.
- Les acteurs, qui sont identifiés par un numéro unique, ont un nom, un prénom une nationalité, une date de naissance et une photo.
- Les réalisateurs, qui sont identifiés par un numéro unique, ont un nom, un prénom, une nationalité et une date de naissance.

Nous savons que :

- ✓ On peut avoir plusieurs films d'une même catégorie. Un film est dans une seule catégorie. Vous devez avoir une table catégories.
- ✓ Un acteur peut jouer dans plusieurs films. Dans un film nous avons plusieurs acteurs.
- ✓ Un film a un seul réalisateur. Un réalisateur peut réaliser plusieurs films.

Vous devez :

- 1- Faire le modèle relationnel en 3FN dans DataModeler, la remise est en PDF ou PNG.
- 2- Générer le code SQL pour la création de votre modèle. (On ne vous demande pas de créer les tables.) La remise est un simple fichier .sql.

## Modalités de remise :

1. La remise finale du travail est pour mardi le **21 mai avant minuit** sauf pour la question de la partie 1 qui doit être remise le **07 mai avant 16 heures**.
2. Le travail sera remis dans la boîte de remise.
3. Tous les fichiers doivent être clairement identifiés par Votrenom.
4. Pour les fichiers **SQL**, il faudra déposer :
  - a. Un fichier sql correspondant aux réponses aux questions de la partie SQL. À l'intérieur de ce fichier, votre nom doit y figurer. Toutes les requêtes doivent être identifiées par leur numéro.
  - b. Un fichier sql correspondant au code SQL généré par la question 2 de la partie normalisation.
5. Le modèle relationnel sera remis en format **PDF** ou en **PNG** et rien d'autre
6. La partie ADO.NET, vous devez remettre l'application zippée.
7. Tout travail non clairement identifié ne sera pas corrigé
8. Tout travail qui n'est pas dans la boîte de remise ne sera pas corrigé.
9. Jusqu'à 10 % des points seront enlevés si les consignes ne sont pas respectées.

## Évaluation, partie SQL (75% de la PFI)

Éléments évalués	Sur
Partie 1	6
Partie 2 : 1-6 (6 requêtes)	12 (6*2)
Partie 2 :7-22 (17 requêtes)	51(17*3)
Partie 3	6 (2*3)
<b>Total</b>	<b>75</b>

## Évaluation, partie ADO.NET (15% de la PFI)

Éléments évalués	Sur
Connexion et déconnexion	3
Ajouter un élève	2
Modifier	3
Rechercher un élève	5
Afficher le nombre de grimoires lus	2
<b>Total</b>	<b>15</b>

## Évaluation, partie normalisation (10% de la PFI)

Éléments évalués	Sur
Le modèle normalisé	8
Le code SQL généré	2
<b>Total</b>	<b>10</b>