

**Département d’informatique et de mathématiques**

**8TRD157 – Bases de données avancées**

Travail pratique #2 : Conception physique

|  |  |
| --- | --- |
| **Étudiant 1** | Alexandre COX |
| **Étudiant 2** | Valentin BAILLEUL |
| **Date** | 31/10/18 |

|  |  |
| --- | --- |
| **Rapport** |  |
| **Contraintes** | **/ 35** |
| **Procédures** | **/ 35** |
| **Script de création de tables** | **/ 15** |
| **Question théorique** | **/ 15** |
|  |  |
| **Total** | **/ 100** |

# Règles d’affaires (contraintes)

[Décrivez brièvement chacune des contraintes identifiées pour les cas d’utilisation et précisez la stratégie employée pour implémenter cette contrainte (CHECK, FOREIGN KEY ou TRIGGER).]

Tout d’abord, nous avons décidé d’utiliser principalement la contrainte CHECK lorsque nous devions effectuer de « simple » vérification (exemple pour s’assurer que l’inventaire d’un film soit au minimum de 0 ou encore que le matricule d’un employé ne doit contenir seulement 8 caractères).

Ensuite, présentent dans la plupart des classes, les FOREIGN KEY nous ont permis de créer des liens entre les tables assez facilement (les noms donnés aux différentes clés primaires ont également contribué à cette facilité).

Pour finir, l’utilisation de différents TRIGGER a été plus que nécessaire pour différentes contraintes (surtout lorsqu’il fallait établir des comparaisons par exemple quand un prêt est demandé, on effectue une vérification sur le forfait et l’inventaire de la personne qui effectue la demande).

# Opérations à encapsuler

[Décrivez brièvement les principales opérations identifiées pour les cas d’utilisation]

Les différentes opérations à encapsuler que nous avons pu identifier se trouve directement sur le schéma relationnel rendu ci-joint (dans le dossier « doc », nommé « Entity Relationship Diagram1.jpg »).

Pour ce qui est des fonctions (présentent dans le dossier BD/fonctions), nous avons décidé de les créer pour s’en servir dans différents procédures et triggers. Nous avons décidé de donner des noms spécifiques, pour chaque procédure / trigger afin de comprendre facilement l’utilité de chacun.

# Planification des tâches

[Décrivez brièvement comment le travail a été divisé dans votre équipe. Estimez, pour chaque tâche de l’énoncé, le pourcentage du travail effectué par chacun des membres de votre équipe.]

# Question théorique

[Créez une vue (VIEW) permettant d’illustrer le concept d’encapsulation (masquer la complexité d’une opération ou requête) dans votre schéma. Cette vue doit permettre d’afficher l’ensemble de l’information stockée relative à un client. Donnez le code SQL permettant de créer cette vue.]

Nous avons décidé de faire cette vue en fonction de notre schéma relationnel (voir dans le dossier « doc » du zip fourni).

CREATE OR REPLACE VIEW main.info\_client AS

SELECT profile.\*,

adresse.num\_civique, adresse.rue, adresse.ville, adresse.province, adresse.code\_postal,

forfait.nom AS forfait\_nom, forfait.cout, forfait.location\_max, forfait.duree\_max,

carte\_credit.numero, carte\_credit.type\_carte, carte\_credit.expiration, carte\_credit.CVV

FROM main.profile INNER JOIN main.membre ON profile.id = membre.id,

main.adresse INNER JOIN main.membre ON adresse.id = adresse\_id,

main.forfait INNER JOIN main.membre ON forfait.nom = forfait\_nom,

main.carte\_credit INNER JOIN main.membre ON carte\_credit.client\_id = membre.id;

# Création des TRIGGERS

[Insérez ici le code SQL permettant la création des TRIGGERS]

Le code permettant de créer les TRIGGERS se trouve dans le dossier « BD/Triggers ».

# Création des procédures stockées

[Insérez ici le code SQL permettant la création des PROCEDURES stockées]

Le code permettant de créer les PROCEDURES stockées se trouve dans le dossier « BD/Procedures ».