

**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).]

Les différentes contraintes sont identifiées dans le diagramme « Entity Relationship Diagram1.jpg », dans le dossier « Doc ».

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, par exemple lorsque plusieurs tables sont impliquées ou lorsqu’une opération ne peut être effectuée dans un CHECK (ex. une opération arithmétique sur une date).

# Opérations à encapsuler

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

Une procédure de création d’entrée pour les tables acteur, scénariste, réalisateur, film, prêt\_courant, profile, client et adresse a été créer pour faciliter les ajouts à la base de donnée. Elle permette de rendre transparente la structure de la BD, tel que l’héritage, les tables servant de type complexe, et les clés primaires auto incrémenté.

Donc, principalement, elles servent à la mise à jour des donnés. Les cas d’utilisations tel que la gestion d’une connexion ou la recherche paramétré par un/des critère(s) seront couvert du côté application.

Une vue, prêts\_client, a été utilisé pour la liste des films loués par un client, plutôt que d’encapsuler un SELECT dans une procédure.

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 DISTINCT 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.client, main.membre, main.profile, main.adresse, main.forfait

WHERE client.id = profile.id AND client.id = membre.id

AND membre.adresse\_id = adresse.id AND membre.forfait\_nom = forfait.nom;

# 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 ».