

POLYTECHNIQUE
MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE



Travail pratique 4 : Applications et Bases de données

École polytechnique de Montréal

Trimestre : Hiver 2020

Nom : Olson Italis

Nom : Charles Bouchard-Levasseur

Matricule : 1898526

Matricule : 1845832

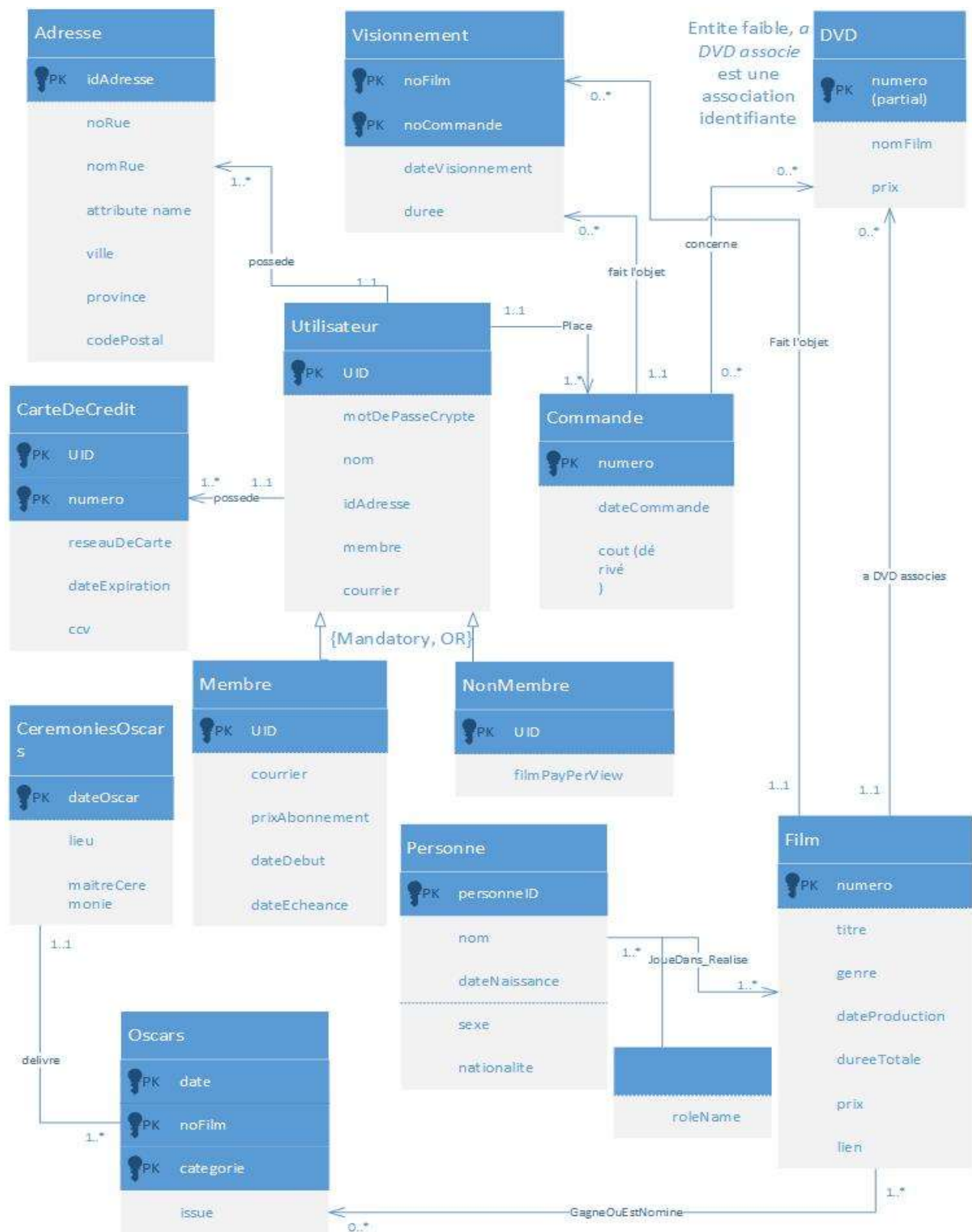
Polytechnique Montréal

Remis le 18 avril 2020

I. Introduction

Cette application a été développée dans le cadre du cours INF3710, donné à Polytechnique, à la session d'hiver 2020. Il fut l'objet des travaux de laboratoire visant à permettre aux étudiants de cerner un cas d'application réelle des connaissances en base de données. Sa conception suit une architecture 3 tiers: le client est le tiers 1, le serveur recevant les requêtes du client est le tiers 2 et une base de données constituant le tiers 3. Angular est utilisé pour le client, Node.js est le serveur et Postgres comme base de données. Le langage Typescript est utilisé pour le développement de l'application. Le projet est accessible sur GitHub sur le lien <https://github.com/nature1996/INF3710.git>

II. Le modèle conceptuel UML



III. Le modèle relationnel en syntaxe abstraite

Adresse(idAdresse, noRue, rue, ville, codePostal, province)

PK : idAdresse;

Utilisateur(UID, motDePasseCrypte, nom, idAdresse, Membre, courrier);

PK: UID

FK: idAdresse References Adresse(idAdresse);

Membre -> bool

Membre(UID, prixAbonnement, dateDebut, dateEcheance)

PK: UID

FK: UID References Utilisateur(UID);

NonMembre(UID, filmPayPerView)

PK: UID

FK: UID References Utilisateur(UID);

CarteDeCredit(UID, reseauDeCarte, numero, titulaire, dateExpiration, CCV)

PK: (UID, numero)

FK: UID References Utilisateur(UID)

Personne(personneID, nom, dateNaissance, sexe, nationalite)

PK: personneID;

Film(numero, titre, genre, dateProduction, dureeTotale, prix, lien)

PK: numero;

RoleFilm(filmID, roleName, personneID, salaire)

PK:(filmID, roleName, personneID),

FK: filmID References Film(filmID),

personneID References Personne(personneID);

CeremoniesOscar(dateOscar, maitre, lieux)

PK: dateOscar;

Oscars(dateOscar, categorie, filmID, issue)

PK: (date, categorie, filmID),

FK: filmID References Film(numero) ,

dateOscar References CeremonieOscars(date) ;

DVD(filmID, numero, nomFilm, prix)

PK: (filmID, numero)

FK: filmID References Film(filmID);

Commande(numero, dateCommande, cout, statut, uid)

PK: numero

FK: uid References Utilisateur(UID);

Visionnement(noCommande, noFilm, dateVisionnement, duree)

PK: (noFilm, noCommande)

FK: noFilm References Film(numero)

FK: noCommande References Commande(numero);

CommandeFilmDVD(noFilm, noCommande, noDVD)

PK: (noFilm, noCommande, noDVD)

FK: noFilm References Film(numero)

FK: noCommande References Commande(numero)

FK: noDVD References DVD(numero);

IV. Les requêtes en algèbre relationnelle

1. Requête 1

Affichez toutes les informations sur un film spécifié par l'utilisateur selon le titre;

Réponse :

$$\sigma_{titre = \$p_titre} (Film)$$

Avec \$p_titre, le titre spécifié par l'utilisateur; \$p_titre peut être 'Matrix', auquel cas le prédicat est $titre = 'Matrix'$.

2. Requête 2

Pour chaque genre de film, listez tous les titres de films ainsi que la dernière date à laquelle un film a été acheté (DVD) ou visionné.

Réponse

2 Algèbre relationnelle pour requête 2

$$CommandesDVD \leftarrow \pi_{noFilm, genre, titre} \left(Film \bowtie_{numero = noFilm} CommandeFilmDVD \right)$$

$$CommandesVisio \leftarrow \pi_{noFilm, genre, titre} \left(Film \bowtie_{numero = noFilm} Visionnement \right)$$

$$ToutesCom \leftarrow CommandesDVD \cup CommandesVisio$$

$$ToutesCommandesDate \leftarrow \pi_{genre, titre, datecommande} \left(ToutesCom \bowtie_{ToutesCom.noFilm = Commande.numero} Commande \right)$$

$$\rho_R (genre, titre, dernière date)_{genre} \mathcal{T}_{max(datecommande)} (ToutesCommandesDate)$$

3. Requête 5

Trouvez les noms des membres dont le coût total d'achat de DVD est plus élevé que la moyenne.

Réponse

$$\begin{aligned}
 & \text{CFD} \leftarrow \text{Commande Film DVD} \quad (\text{renommage de table}) \\
 & \cdot \text{CoûtCommande DVD} \leftarrow \pi_{\text{noCommande}, \text{prix}} \left(\text{CFD} \bowtie \text{DVD} \right) \\
 & \quad \text{(Note : CDU pour tous les dvd achetés par user)} \quad \text{CFD.noFilm = DVD.noFilm} \\
 & \quad \text{CFD.noDVD = DVD.numero} \\
 & \cdot \text{CDU} \leftarrow \pi_{\text{UID}, \text{prix}} \left(\text{Commande} \bowtie \text{CoûtCommande DVD} \right) \\
 & \quad \text{Commande.numero = CoûtCommande DVD.noCommande} \\
 & \cdot \text{CDUTotal} \leftarrow \rho_{\text{User}, \text{coûtTotal}} \left(\text{UID} \right) \text{SUM}(\text{prix}) \left(\text{CDU} \right) \\
 & \cdot \text{CDUTotal-Moy} \leftarrow \text{CDUTotal} \times \left(\rho_{\text{Moy}} \right) \text{AVG}(\text{coûtTotal}) \left(\text{CDUTotal} \right) \\
 & \quad \downarrow \text{produit cartésien} \\
 & \cdot \text{CDURecherche} \leftarrow \sigma_{\text{coûtTotal} > \text{Moy}} \left(\text{CDUTotal-Moy} \right) \\
 & \cdot \text{CDUMembre} \leftarrow \left(\text{CDURecherche} \bowtie \text{Membre} \right) \\
 & \quad \text{CDURecherche.User = Membre.UID} \\
 & \text{Resultat} \leftarrow \pi_{\text{nom}} \left(\text{CDUMembre} \bowtie \text{Utilisateur} \right) \\
 & \quad \text{CDUMembre.user = Utilisateur.UID}
 \end{aligned}$$

V. La présentation de l'application

Notre application se nomme Netflix Polytechnique Montréal. La première page avec laquelle l'utilisateur a accès est la page de login. La raison étant que l'ensemble des autres pages (à l'exception des pages d'administration) sont protégées si l'utilisateur n'est pas connecté.

La page donne un retour d'information si les données entrées ne sont pas valides, mais elle évite volontairement de préciser quelles informations sont problématiques pour protéger la vie privée. Lorsqu'un utilisateur est connecté, il perd le droit d'accéder à cette page et est immédiatement redirigé vers la liste des films.



Vous pourriez remarquer que l'entête affiche maintenant un message à destination de l'utilisateur, servant doublement à conforter l'utilisateur et à confirmer l'identité de l'utilisateur actif. Il y a maintenant aussi l'option de se déconnecter, qui redirige automatiquement à la page de login. Enfin, cliquer sur le nom de l'application (en rouge) permet de revenir à la liste des films. En cliquant sur l'un des films, il est possible d'en voir le détail sur une nouvelle page. Cette page est sur un lien déterminé par l'id unique au film, ce qui veut dire que cette page pourra être retrouvée directement une fois qu'un système d'enregistrement de la session (cookie) sera implémenté. Cette page indique l'année de production ainsi que la durée en minutes et le genre du film. On y retrouve par la suite un bouton permettant de passer au visionnement du film, actif uniquement pour les membres (le visionnement pour les non-membres n'étant pas requis). Pour finir, on y trouve deux tableaux, indiquant les acteurs ayant participé au film qui sont enregistrés dans la base de données, et les nominations aux Oscars si applicable. Plus tard, il pourrait être possible de cliquer sur l'un de ces deux éléments pour voir les détails sur l'acteur ou la cérémonie, mais cela n'a pas été implémenté pour le moment. Lorsqu'on choisit d'écouter un film, l'utilisateur est redirigé vers l'une de deux pages : la page de visionnement s'il possède le film, ou la page d'achat s'il ne l'a jamais



regardé. Ces deux pages possèdent aussi la référence à l'identifiant du film, ce qui pourrait éventuellement servir à la même chose que la page de détail du film. Sur la page d'achat, il n'y a qu'un seul bouton permettant d'acheter un film. Ce bouton n'est pas activé pour les non-membres, mais reste affiché avec le commentaire que le film est gratuit pour les membres (servant de publicité pour les abonnés, et ce même si un non-membre ne devrait jamais



être redirigé vers cette page (il pourrait éventuellement y accéder par lien direct)). Une fois le film acheté, ou s'il le possédait déjà, l'utilisateur est redirigé vers le visionnement du film. Cette page en temps que tel n'a rien de particulier, mais si l'utilisateur quitte la page normalement (en cliquant sur le nom de l'application) la progression du visionnement est enregistrée et restaurée lors de la prochaine lecture.

Il est maintenant temps de regarder les fonctionnalités d'administration. Celles-ci sont accessibles à l'aide du lien relatif « /admin ». Lorsque l'administrateur accède à la

[Retour à l'accueil administrateur](#)

Nom :

E-mail :

Mot de passe :

☐ est membre

Adresse :

Rue :

Ville :

Code Postal :

Province :

Pays :

première page, un message de bienvenue avec des liens vers

les ressources administratives sont affichées. Si celui-ci suit le lien vers l'ajout de profil. Cette page contient une liste des paramètres liés à un compte. Il est de la responsabilité de l'administrateur de s'assurer que les informations entrées sont valides et complètes. S'il choisit « Ajouter/modifier film » il aura été amené à la page de contrôle des films. Sur cette page, les films sont listés, avec le détail d'ajout de leurs IDs. En cliquant sur le X rouge dans la colonne supprimer, il est possible d'enlever un film de la base de

données. Appuyer sur le bouton Ajouter/modifier affiche les champs pour interagir avec les films. Le formulaire permet de créer un film si le premier champ

[Retour à l'accueil administrateur](#)

ID (vide si nouveau) :

Titre :

Genre :

Date de production :

Durée (en minutes) :

Prix :

Lien :

	Genre	Durée	Sortie	Prix	Modifier
1	SCI-FI/ACTION	136	1999-03-31		<input type="button" value="M"/>
2	SCI-FI/ACTION	138	2003-05-07		<input type="button" value="M"/>

(id) est laissé vide, ou d'en modifier un si le champ contient l'identifiant du film. À la fin du champ, un bouton « confirmer » permet d'envoyer l'information au serveur (la validation des informations entrées est la responsabilité de l'administrateur), et un bouton « Reset » qui permet de vider tous les champs. Le bouton « M » dans la colonne Modifier permet aussi d'afficher le formulaire s'il ne l'a pas déjà, et insère les informations du film dans celui-ci pour faciliter la modification des informations. Pour finir, lorsque le formulaire est affiché, un bouton « Masquer » apparaît, qui permet de le cacher afin de faciliter la lecture des données de la table.



Netflix Polytechnique Montreal

[Retour à l'accueil administrateur](#) [Ajouter/modifier](#)

ID	Titre	Genre	Durée	Sortie	Prix	Modifier	Supprimer
1	The Matrix	SCI-FI/ACTION	136	1999-03-31		<input type="button" value="M"/>	<input type="button" value="X"/>
2	The Matrix Reloaded	SCI-FI/ACTION	138	2003-05-07		<input type="button" value="M"/>	<input type="button" value="X"/>
3	The Matrix Revolutions	SCI-FI/ACTION	129	2003-11-05		<input type="button" value="M"/>	<input type="button" value="X"/>
4	War Room	Christian drama	120	2015-08-28		<input type="button" value="M"/>	<input type="button" value="X"/>
5	Overcomer	Christian drama	115	2019-08-23		<input type="button" value="M"/>	<input type="button" value="X"/>
6	Who Framed Roger Rabbit	Live Action	104	1988-06-22		<input type="button" value="M"/>	<input type="button" value="X"/>

VI. Guide d'installation et de configuration

Afin de lancer l'ensemble de l'application, il vous faut :

- Modifiez « connexionConfig » dans le fichier « database.service.ts »
- Ouvrez une invite de commande dans le dossier client et roulez la commande '\$npm install'
- Ouvrez une invite de commande dans le dossier server et roulez la commande '\$npm install'
- Dans le dossier serveur, rouler la commande '\$npm start'
- Dans le dossier client, rouler la commande '\$npm start'