



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

INF3710 –Fichiers et Bases de données

Hiver 2020

TP4

Groupe 04

1924705 – Abderrahim Ammour

1955878 - Alexander Vorobev

Soumis à : Amal Zouaq

19 avril 2020

Introduction :

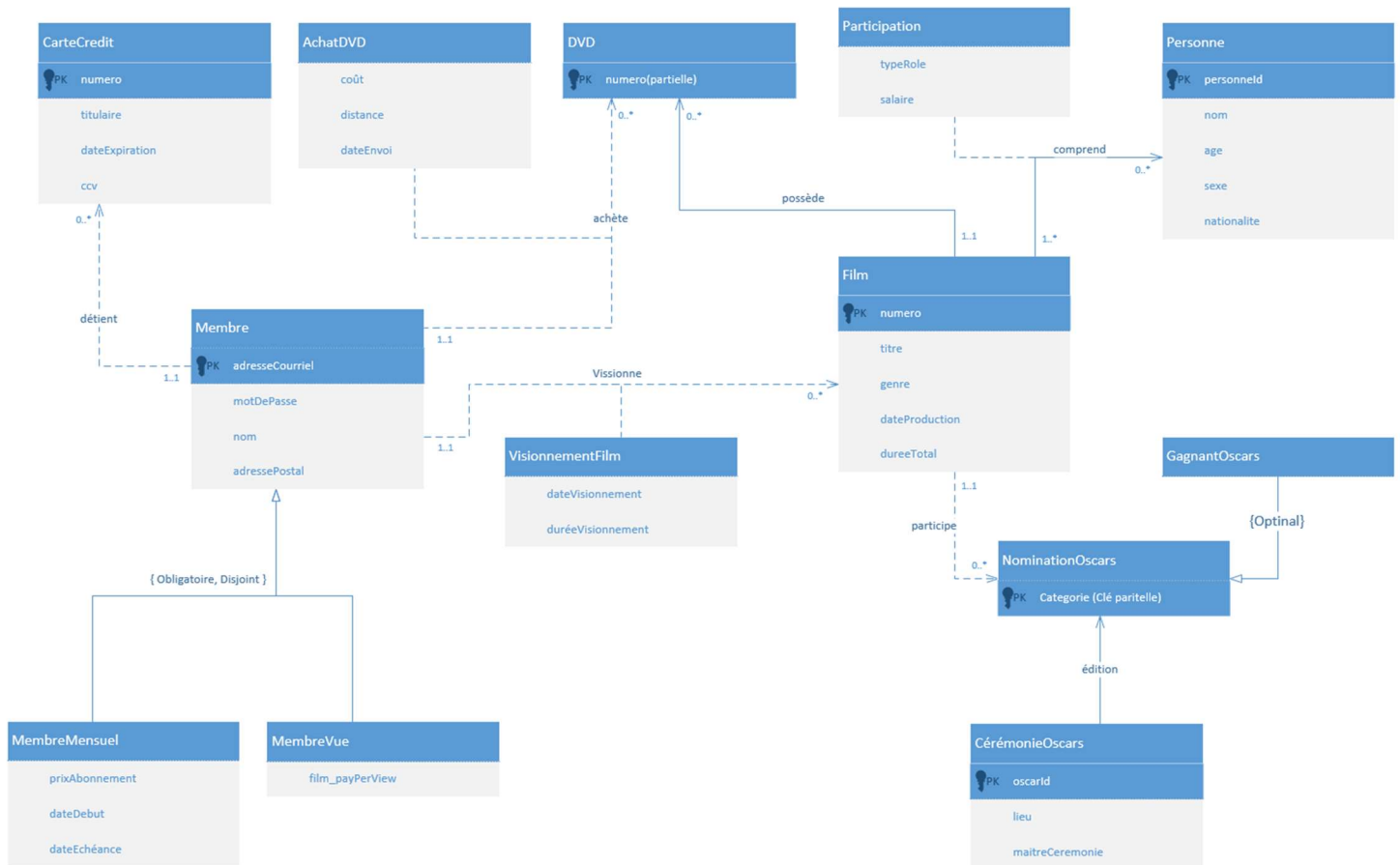
Dans le cadre du projet, il nous a été demandé concevoir une plateforme de location de vidéos en ligne *Netflix Poly*. Cette plateforme permettra à ses membres d'acheter et/ou de télécharger des vidéos. Notre application Web permettra, au moyen d'une interface, d'insérer, supprimer, modifier et d'interroger les données de notre base de données.

D'abord, nous avons établi une hypothèse sur une étude de cas qui nous a été remise. Cette étude de cas est basée sur les requis et exigences d'un client fictif. Certaines informations étant manquantes nous avons dû faire des recherches plus poussées pour mieux comprendre par exemple les particularités d'une cérémonie des Oscars. Nous avons à gérer plusieurs entités importantes telles que Membre, Film et CérémonieOscars. Nous avons dû proposer un modèle entités-associations pour pouvoir créer notre base de données PostgreSQL. Pour ce faire nous avons utilisé le logiciel Visio pour concevoir un modèle UML à partir de l'information fournie. Afin de pouvoir créer notre base de données nous avons créé un schéma et des données représentatives de la réalité sans oublier les contraintes nécessaires dans la représentation de notre modèle tel que *l'intégrité référentielle*.

Par la suite, nous nous sommes assurés de tester notre schéma et nos données sur PGADMIN, afin d'avoir une base de données qui permet d'implémenter les requêtes souhaitées.

Finalement, nous devons par la suite relier notre base de données à une plateforme de location de vidéos en ligne *Netflix Poly* que nous avons à développer à l'aide de Angular et d'un serveur Express. Netflix Poly permet à un nouveau membre de s'inscrire avec un formulaire d'inscription ou tout simplement de se connecter si le membre est déjà enregistré sur la base de données. Cette plateforme permet également de gérer les membres et les films si vous êtes administrateur et donc modifier la base de données avec des requêtes SQL directement de l'application. Il sera aussi possible d'afficher les informations d'un film tel que le titre, le genre, la date de sortie du film, les acteurs, le réalisateur et la durée du film. Enfin, il sera possible de visionner un film tendance ou d'un certain genre en étant membre de Netflix Poly à l'aide d'un simple clic de souris.

Diagramme UML :



Modèle relationnel (syntaxe abstraite) :

Membre(membreID, nom, courriel, motDePasse, adressePostal, isAdmin)

PK: membreID

MembreMensuel(membreID, prixAbonnement, dateDebut, dateEcheance)

PK: membreID

FK: membreID REFERENCES Membre(membreID)

MembreVue(membreID, film_payperview)

PK: membreID

FK: membreID REFERENCES Membre(membreID)

CarteCredit(cartelID, membreID, numero, titulaire, dateExpiration, CCV)

PK: (cartelID, membreID)

FK: adresseMembre REFERENCES Membre(membreID)

FK: membreID REFERENCES Membre(membreID)

Film(filmID, titre, genre, dateProduction, duréeTotal)

PK: filmID

DVD(dvdID, numeroInstance, filmID)

PK: (dvdID, filmID)

FK: filmID REFERENCES Film(filmID)

Personne(personnelID, nom, age sexe, nationalité)

PK: personnelID

Participation(personnelID, filmID, typeRole, salaire)

PK: (personnelID, filmID, typeRole)

FK: personnelID REFERENCES Personne(id)

FK: filmID REFERENCES Film(filmID)

CérémonieOscars(oscarID, lieu, date, maitreCérémonie)

PK: oscarID

NominationOscars(oscarID, filmID, catégorie)

PK: (oscarID, filmID, catégorie)

FK: oscarID REFERENCES Oscar(oscarID)

FK: filmID REFERENCES Film(filmID)

GagnatOscars(oscarID, filmID, catégorie)

PK: (oscarID, filmID, catégorie)

FK: oscarID REFERENCES Oscar(oscarID)

FK: filmID REFERENCES Film(filmID)

VisionnementFilm(membreID, filmID, cout, dateVisionnement, duréeVisionnement)

PK: (membreID, filmID)

FK: filmID REFERENCES Film(filmID)

FK: membreID REFERENCES Membre(membreID)

AchatDVD(achatID, membreID, dvdID, cout, distance, dateEnvoi)

PK: (achatID)

FK: dvdID REFERENCES DVD(dvdID)

FK: membreID REFERENCES Membre(membreID)

Requêtes en algèbre relationnelle:

6) Exprimez en algèbre relationnelle les requêtes 1, 2 et 5. (5 points)

1. Affichez toutes les informations sur un film spécifié par l'utilisateur (selon le titre).

FilmSpecifier = π titre, genre, dateProduction, dureeTotalMinutes (σ Film.titre = 'titleName'(Film))

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

R1 = (π dateVisionnement (VisionnementFilm)) EQUIJOIN (π dateEnvoi (AchatDVD))

VisionnementFilm.filmID = AchatDVD.filmID

R2 = σ VisionnementFilm.dateVisionnement > AchatDVD.dateEnvoi(R1)

R3 = (π filmID (Film)) NATURALJOIN (π filmID (VisionnementFilm)) NATURALJOIN
(π filmID,dvdID (DVD)) (π filmID,dvdID (AchatDVD))

R4 = π genre,titre (R3)

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

R1 = (π membreID (Membre)) NATURALJOIN (π membreID (σ
SUM(AchatDVD.cout)(AchatDVD)))

R2 = σ SUM(AchatDVD.cout)(AchatDVD) > R1

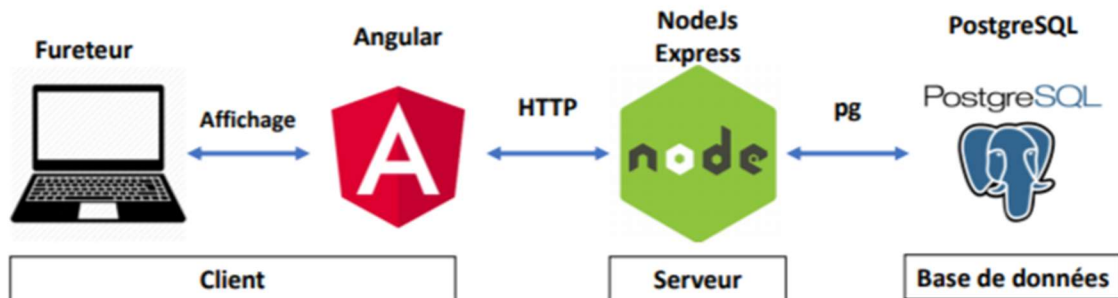
R3 = (π membreID (Membre)) NATURALJOIN (π membreID (R2))

R4 = π nom , SUM(AchatDVD.cout) somme (R3)

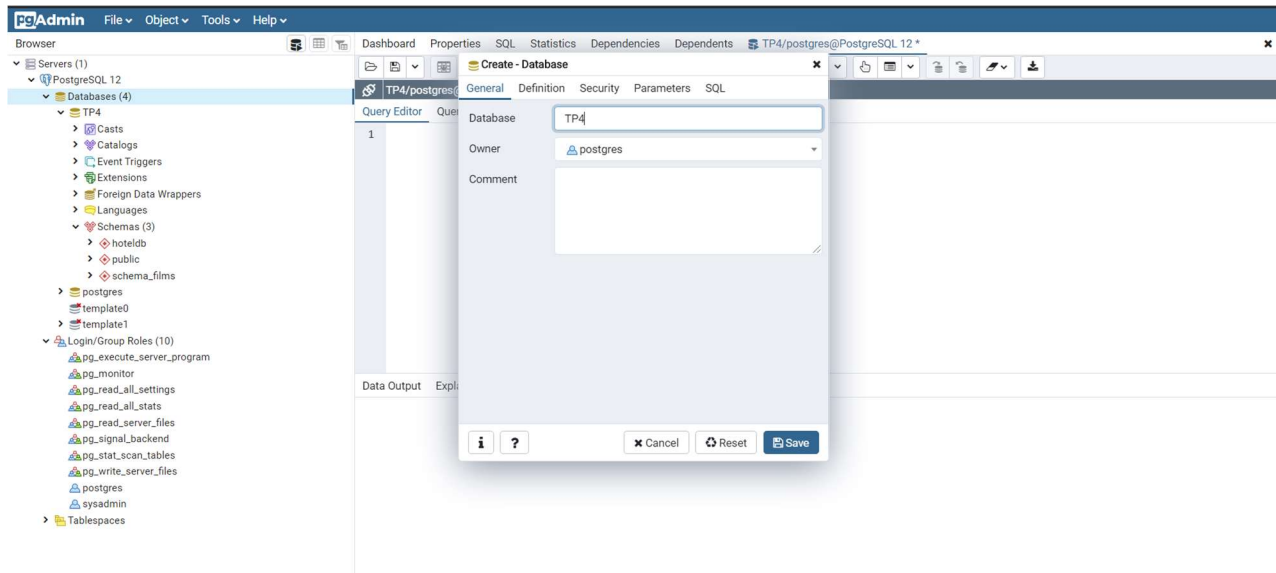
Guide d'installation et de configuration:

Étapes à suivre :

1. Assurez-vous d'avoir Angular et Express installé sur la machine.



2. Créer une Base de donnée et nommée la TP4 dans PGADMIN.



3. Créer un usager nommé "sysadmin" dans l'onglet login/group et configurer les privilèges de celui-ci tel que la représentation suivante:

TP4/postgres@PostgreSQL 12 *

DashboardPropertiesSQLStatisticsDependenciesDependents

Create - Login/Group Role

GeneralDefinitionPrivilegesMembershipParametersSecuritySQL

1

Name

sysadmin

Comments

i?

Cancel

Reset

Save

Help ▾

Create - Login/Group Role

✕

General

Definition

Privileges

Membership

Parameters

Security

SQL

Password

....

Account expires

YYYY-MM-DD HH:mm:ss Z

📅

Connection limit

-1

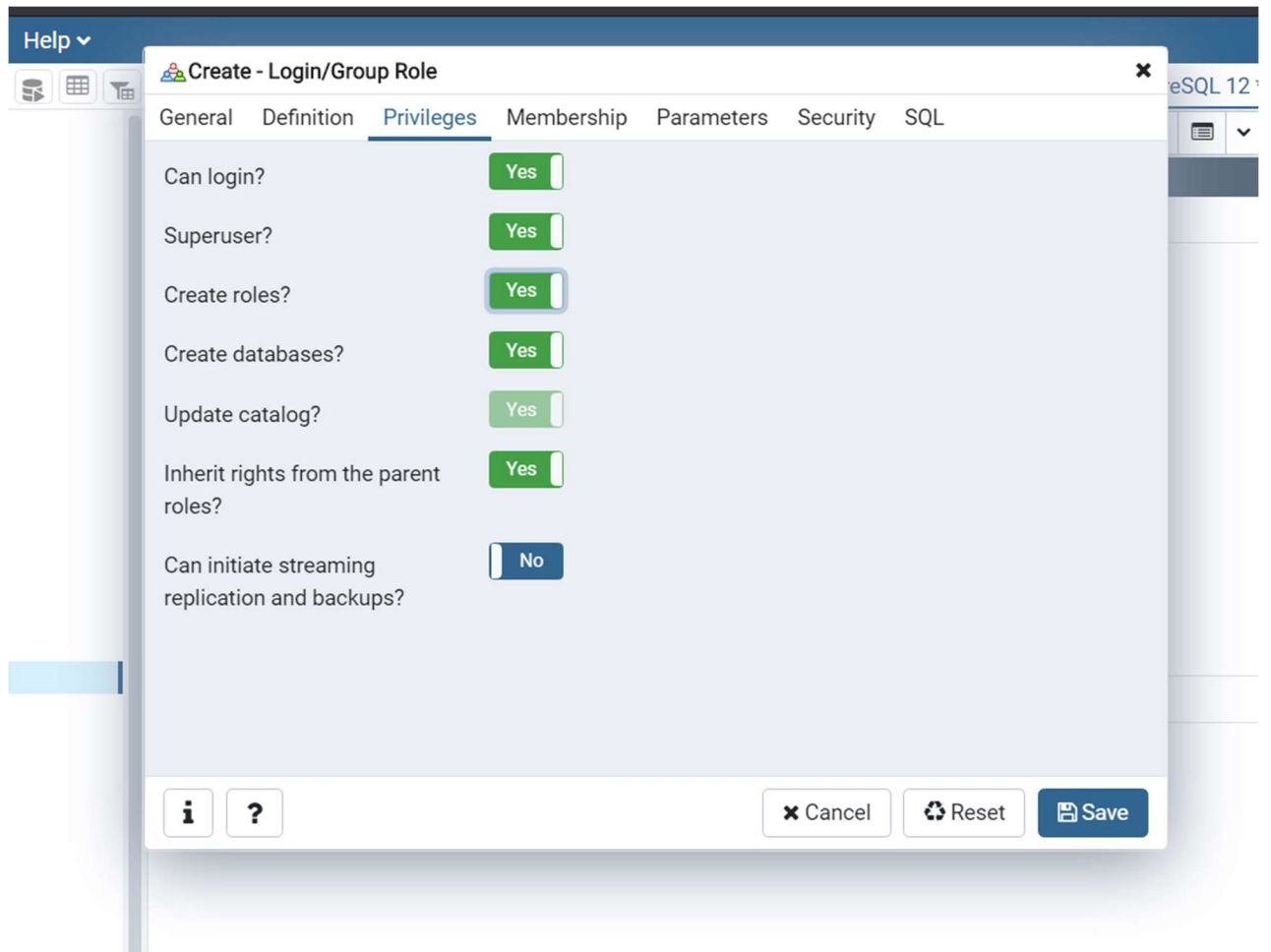
ⓘ

?

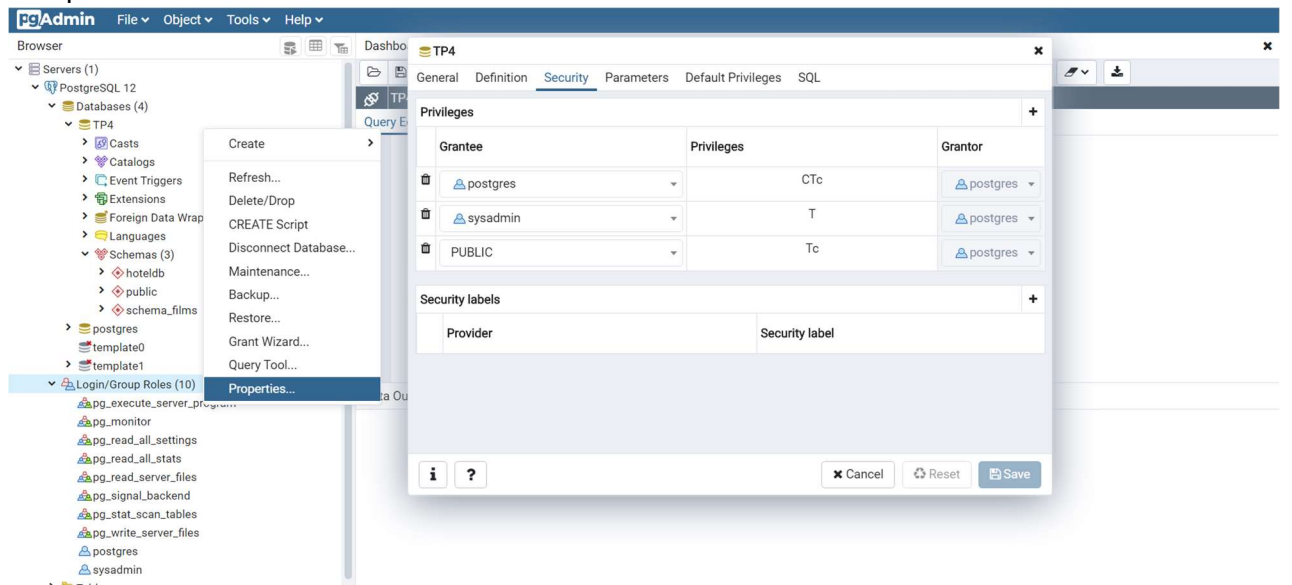
✕ Cancel

🔄 Reset

💾 Save



4. Donner accès à l'utilisateur "sysadmin" à votre base de donnée dans l'onglet *security* tel que la représentation suivante:



5. Assurez-vous d'initialiser correctement les configurations (**ConnectionConfig**) de la base de données dans le fichier `database.service.ts` dans le dossier serveur.

```
public connectionConfig: pg.ConnectionConfig = {  
  user: 'sysadmin',  
  database: 'TP4',  
  password: '1234',  
  port: 5432, // selon vos configurations de PGADIMN  
  host: '127.0.0.1',  
  keepAlive: true,  
};
```

6. Ouvrir un terminal à l'intérieur du dossier `/client` et exécuter la commande **npm install**.
7. Ouvrir un terminal à l'intérieur du dossier `/server` et exécuter la commande **npm install**.
8. Exécuter la commande **npm start** dans le terminal `/client`.
9. Exécuter la commande **npm start** dans le terminal `/server`.
10. Normalement vous devriez être capable d'utiliser l'application.
11. Le schéma et le data se trouvent dans les fichiers **bdschema.sql** et **data.sql**.
12. Avant de commencer, appuyez sur le bouton **initialiser base de données** and <http://localhost:4200/home>.

Présentation de l'application Web :

Dans le cadre du projet Netflix Poly, on a dû concevoir une application permettant d'accomplir certaines tâches dont de pouvoir lister les films gardés en mémoire dans la base de données, pouvoir les supprimer, modifier et visionner. De plus, on a dû concevoir un système d'enregistrement de membres et un système de connexion encore une fois traitant non-seulement la partie client, mais aussi la partie serveur de l'application.

L'enregistrement de membres

Tout d'abord, un utilisateur visitant notre application devra se connecter à la page de connexion. Une fois qu'il entre l'adresse courriel et le mot de passe, cette utilisation sera authentifié sur la base de donnée. Si l'utilisateur est présent dans la base de données, il pourra se connecter en cliquant sur le bouton se connecter, sinon un message indiquera que les informations rentrées sont invalide. Le rôle lui sera attribué selon son rôle dans la base de données, ainsi un utilisateur administrateur continuera en tant qu'admin et ce rôle sera indiqué en haut à droite dans l'interface. Il est aussi à noter que le bouton d'initialisation est présent dans l'objectif de faciliter la correction, mais pourrait être retiré et initialisé au chargement de la page.

The image shows a login form titled "Se connecter" centered on a light blue background. The form is a white rounded rectangle containing two input fields: "Adresse courriel" and "Mot de passe". Below these fields are three buttons: "S'inscrire" (dark grey), "Initialiser base de données" (purple), and "Se connecter" (dark grey).

Si toutefois la personne visitant l'application ne s'est pas encore inscrite, elle pourra le faire en cliquant sur le bouton s'inscrire. Ce bouton mènera vers l'interface indiquée si bas ou il pourra entrer les informations demandées et ainsi le type d'abonnement mensuel ou à la vue. Après validation des entrées par l'application, un ajout d'un nouveau membre sera fait à la base de données. Par la suite, ce nouveau membre pourra se connecter sans problème. Il est important à mentionner qu'ici nous faisons une vérification supplémentaire dans les données qui seront rentrés par l'utilisateur pour vérifier qu'elles sont correctes pour la base de données.

Inscription

| _____ Nom _____

Adresse courriel _____

Adresse postale

Num. _____ Rue _____ App. _____

Ville _____ Code postal _____

Type d'Abonnement

☐ Mensuel

☐ À la vue

Carte de crédit

Numéro _____

Mois _____ Année _____

CCV _____

Mot de passe _____

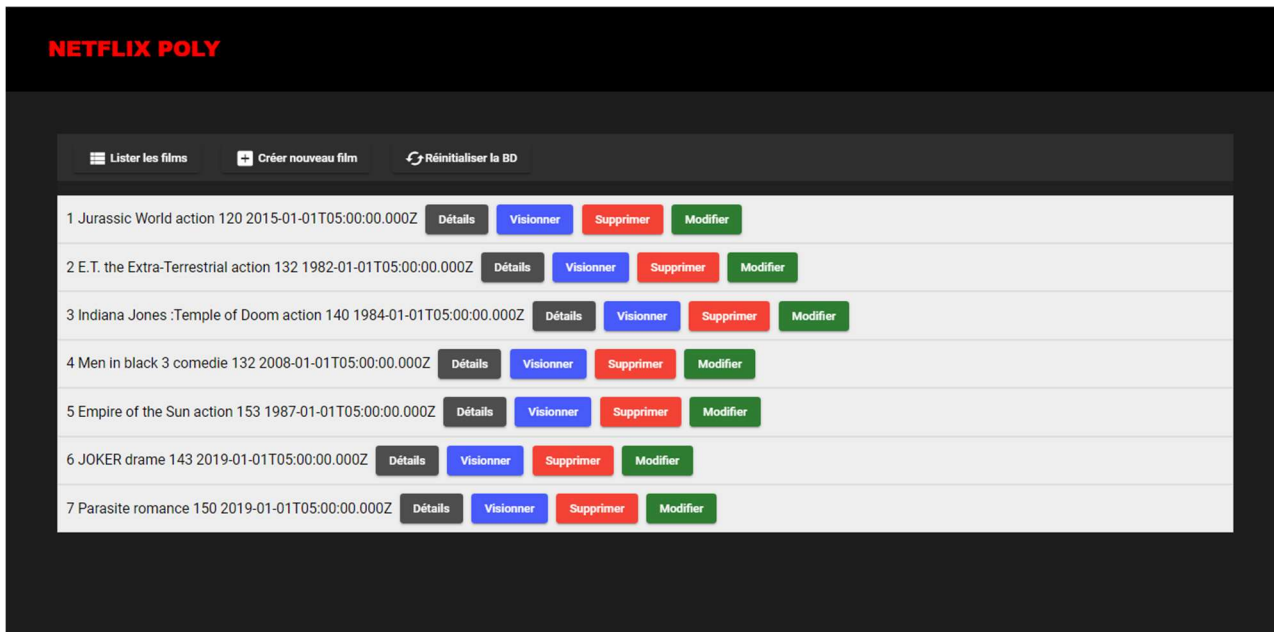
Mot de passe _____

Confirmez votre mot de passe _____

S'inscrire

L'ajout, suppression et modification des films

Une fois l'utilisateur connecté, il pourra se lancer au coeur de l'application ou l'interface présentée sert à ajouter, supprimer et modifier les films. Il était planifié de rendre un meilleur visuel pour le côté de l'utilisateur, mais nous avons manqué de temps donc l'utilisateur a un aperçu des films similaire à celui d'un administrateur pour le but de l'exercice. Nous avons tout de même programmé l'extension pour faire la distinction entre un administrateur et un membre avec l'attribut booléen `isAdmin`, mais nous acheminer tous les deux vers la page suivante.



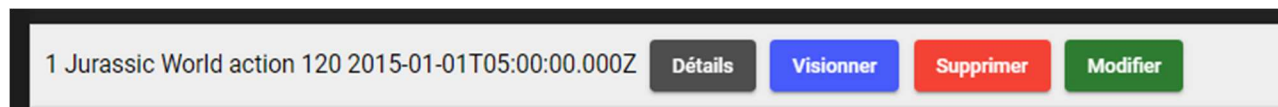
Dans cette interface 3 options principales vous sont présentées:

- 1- Lister les films qui chargent les films de la base de données (elle devra être exécutée en rentrant sur la page). Cette requête interroge la base de données pour avoir un aperçu des films qui y sont présents.
- 2- Créer nouveau film qui redirige l'utilisateur sur une page de création de nouveaux films. Cette option fait l'ajout d'un film dans la base de données.

The screenshot shows the 'NETFLIX POLY' application interface with the 'Ajouter un film' form. The form has the following fields and buttons:

- Ajouter un film:**
- Titre :**
- Genre:**
- Durée:**
- Date:** Choisir la date
- Sauvegarder** **Annuler**

3- Réinitialiser la BD qui réinitialise la BD avec les valeurs initialement chargées au premier chargement.



Pour chaque film, il existe aussi 4 options, dont:

1- Détails: présente une description plus détaillée du film, cette option n'est pas encore implémentée, mais pourra l'être si on en voit l'utilité.

2- Visionner: permet de visionner le film et pourra être changé par continuer si le film à déjà était débuté et non terminé. Pour l'instant cette option, n'est pas complétée, ainsi elle ne fait qu'afficher un message d'erreur.

3- Supprimer: supprime le film de la base de données.

4- Modifié: redirige vers une page de modification du film permettant à l'utilisateur d'apporter des modifications à un film existant dans la base de données.

A dark-themed interface for adding a new film. At the top, the text 'NETFLIX POLY' is displayed in red. Below it, a form titled 'Ajouter un film:' contains several input fields: 'Titre:' with the value 'Indiana Jones :Temple of Doom', 'Genre:' with the value 'action', and 'Duree:' with the value '140'. Below these is a date selection section with the text 'Choisir la date', a date input showing 'Date: 1/1/1984', and a calendar icon. At the bottom of the form are two buttons: 'Sauvegarder' and 'Annuler'.

Aspects novateurs de l'application

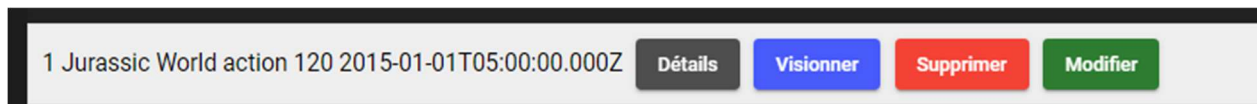
Dans le but d'aller au-delà des exigences, certaines fonctionnalités ont été ajoutées.

1- Gamme de couleurs:

La gamme de couleurs de l'application a été spécifiquement choisie pour pouvoir rendre notre application accessible non seulement à des personnes sans handicaps, mais aussi pour les personnes avec des difficultés visuelles telles que le Daltonisme. On le remarque dans le choix des couleurs des boutons par exemple ou chaque bouton se situe dans le spectre différent de couleurs que les Daltoniens ne voient pas.

2- Réduction de la charge cognitive:

Si vous portez attention à la page des films, il vous sera possible de constater que le fond a été conçu dans les gammes de couleurs pâles, peu excitantes. Par contre, les boutons sont colorés et invitent ainsi l'utilisateur à effectuer des actions dessus.



3- Gestion d'erreurs et vérification:

Dans une application telle que Netflix Poly, la consistance et la validité des données tombent en priorité numéro 1. Ainsi, l'application englobe les fonctionnalités de sauvegarde avec une validation complète des erreurs possibles et de entrées faites par l'utilisateur. En voici un exemple :

```
let isnum = /^\\d+$/\\.test(this.duree);
if (!this.titre || !this.genre || !this.duree || !this.date) {
  alert("Les entrees ne sont pas valides veuillez recommencer!");
  return;
}
if (!isnum) {
  alert("veuillez rentrer un nombre dans la duree");
  return;
}
```

4- Guidage:

Il est à noter que lors de la validation d'erreurs, une attention particulière est attribuée à guider l'utilisateur pour rendre ses données valides. Ainsi les messages d'erreurs sont personnalisés afin de permettre la meilleure expérience possible.

Difficultés rencontrés:

Nous avons eu de la difficulté par moment à comprendre les requêtes SQL demandées. Nous n'étions pas certains si c'était notre modèle relationnel ou notre approche qui était mauvaise. Nous avons dû à plusieurs reprises retourner modifier notre schéma et nos données SQL pour répondre aux requêtes SQL demandées. Nous avons été confus par la formulation de certaine requêtes, après avoir fait des recherches, nous avons trouvé des précisions sur le forum du cours.

Nous avons été dans l'impossibilité d'implémenter l'api de Google maps, puisque nous n'avions pas assez de temps pour le faire. Pour l'exercice nous avons fait un trigger qui vérifie une condition particulière avec le code postal et qui retourne par la suite le coût de la livraison du DVD.

Nous avons dû faire face à plusieurs contraintes de temps, nous avons donc dû faire appel à une bonne communication au sein de l'équipe, afin de séparer le travail. Nous nous sommes davantage concentrés sur la partie conception de la base de données et les requêtes SQL que sur le développement de l'interface et des fonctionnalités de l'application. Nous trouvons plus important de comprendre les notions du cours de ce qui a trait à la base de données. Tel que mentionné sur l'énoncé, nous savions dès le début que l'application nous prendra du temps et nous devons compter sur notre effort personnel et notre expérience en programmation pour la compléter.

Enfin, nous avons trouvé difficile d'avoir des réponses à certaines de nos questions puisque nous n'avions plus accès direct avec un chargé de laboratoire.

Conclusion :

Ce travail nous a permis de mettre en pratique l'ensemble des concepts appris au cours de la session, nous avons été *challengés* par un énoncé réaliste. Enfin, nous avons pu développer une application web permettant à un utilisateur d'interroger une base de données. Tout cela, nous a permis d'approfondir nos connaissances et d'adapter l'utilisation des requêtes SQL pour une application web.