Projet Bases de Données (BD6)

CineNet : un forum de cinéphiles

Le projet s'effectue par groupe de 2 personnes. **Afin que tout le monde puisse profiter des réponses, les questions concernant le projet doivent être posées sur discord et non par mail. Merci de ne poster aucun diagramme de modélisation sur discord : si vous ne jouez pas le jeu le discord sera supprimé.**

Vous pouvez contacter vos encadrants aux adresses :

Amélie Gheerbrant	:	`amelie@irif.fr`
Sarah Winter	:	`sarah.winter@irif.fr,`
Cristina Sirangelo	:	`cristina@irif.fr`
Raphael Cosson	:	`cosson@irif.fr`
Mouna Safir	:	`safir@irif.fr`

La récupération de données existantes (e.g., noms de films, genre, casting) est non seulement autorisée, mais encouragée. En revanche, les règles anti-plagiat de l'Université s'appliquent à votre travail et toute utilisation d'éléments extérieurs devra être explicitement évoquée et sourcée. Vous pouvez discuter des différents aspects du projet avec les autres groupes, mais toute copie partielle ou totale de code entre plusieurs groupes est interdite.

Rendu 1 : Modélisation conceptuelle

Date	:	TBA (juste avant la soutenance).
Format	:	Diagramme entités-associations au format pdf dessiné à la main ou avec l'outil de votre choix.
Consignes	:	Réalisez le modèle conceptuel de données (diagramme entités-associations) de votre base de données. N'omettez pas de lister les contraintes externes.

Rendu 2 : Code

Date	:	TBA (juste avant la soutenance).
Format	:	Une *unique* archive contenant l'ensemble de vos fichiers (diagramme entités-association, fichier SQL de création du schéma et de peuplement des tables, code SQL des requêtes) et un fichier `README.txt` expliquant brièvement à quoi correspond chaque fichier, déposée sur Moodle. Pour créer une archive `MonArchive.tar` à partir d'un dossier `MonCode` utilisez la commande Linux : `tar cvf MonArchive.tar MonCode`
Consignes	:	Implémentez dans `PostgreSQL` votre base de données et les requêtes permettant de l'utiliser, en utilisant votre schéma entités-associations et les consignes détaillées à partir de la page 2.

Rendu 3 : Rapport

Date	:	TBA (juste avant la soutenance).
Format	:	Rapport entre 4 et 8 pages au format PDF déposé sur Moodle. Vous pouvez utiliser `https://www.freepdfconvert.com/fr` pour convertir en PDF.
Consignes	:	Vous inclurez votre modèle entités-associations et un schéma relationnel de vos tables. Vous expliquerez les choix qui ont motivé votre modélisation, et les limitations de votre modèle. Vous expliquerez le passage du schéma conceptuel au schéma logique, en expliquant les éventuelles étapes intermédiaires de restructuration du schéma et les différents arbitrages que vous aurez été amenés à faire. Si certaines contraintes n'ont pu être implémentées au niveau du schéma logique, vous le soulignerez. Vous décrirez également les requêtes que vous avez choisies d'implémenter, et les choix que vous avez faits pour le calcul de l'indice de recommandation (voir Section 4).

Soutenance

Date	:	TBA (après les examens).
Format	:	Présentation orale par *tous* les membres du groupe.
Consignes	:	Vous débuterez votre soutenance par une présentation de votre travail. Nous vous poserons ensuite des questions sur vos choix de modélisation, les contraintes d'intégrité utilisées, les points forts et les points faibles de votre approche, le fonctionnement de vos requêtes, etc.

1 CineNet : un forum de cinéphiles

Les gens tendent naturellement à se regrouper autour de passions communes. Le cinéma en est un exemple universel, avec ses chapelles (des amateurs de films de kung-fu ou d'animé, aux mordus de séries, en passant par les fans d'art et essai) et ses grand-messes (l'Étrange festival, le festival de Cannes...). Les amateurs se retrouvent lors d'événements (avant-premières de films, festivals, conventions, ...), sur des forums (www.rottentomatoes.com, discord, ...) et des réseaux sociaux (instagram, facebook...). Dans ce contexte, une nouvelle plateforme à visée unificatrice est en projet. Son objectif est de permettre aux cinéphiles de faire des découvertes (nouveaux films, événements...) d'échanger leurs avis et de se connecter avec d'autres passionnés du septième art. A la manière de TMDB, CineNet est conçu comme une base de données participative recensant toutes les oeuvres cinématographiques, mais également comme un réseau social et un forum. Les utilisateurs de la plateforme pourront non seulement être des personnes lambda, mais également des acteurs, des réalisateurs, des studios, des organisateurs de festivals, des clubs ou salles de cinéma... Il sera possible à un utilisateur de suivre ou bien d'être ami avec une autre entité du réseau. Notez que la relation d'amitié est symétrique, mais qu'il est possible que A suive B sans que B suive A (sur le modèle du follows de Instagram).

Un ingrédient essentiel de CineNet sera l'intégration de son forum de discussion, dans lequel les diverses publications pourront concerner un événement particulier, un film, une série... Vous êtes libres d'organiser les publications de la manière qui vous semble la plus pertinente (vous pouvez par exemple vous inspirer vos salons discord préférés). Celles-ci auront toujours un auteur (attention donc à bien modéliser le concept d'utilisateur, sans oublier login, mot de passe, éventuellement rôle comme dans les salons discord...) et pourront être soit des publications de premier niveau, soit des réponses à une autre publication. Elles s'insèreront dans différentes conversations regroupées par catégories. Les publications de premier niveau devront spécifier les sujets sur lesquelles elles portent, par exemples le(s) film(s) dont elles parlent, et les utilisateurs pourront rechercher des discussions en utilisant comme mots clef des titres de films, des noms d'acteur, des genres cinématographiques.... Les genres cinématographiques et leurs sous-genres seront également intégrés comme mots-clés, permettant ainsi aux utilisateurs de faire des recherches en fonction de leurs goûts spécifiques. Attention à bien représenter le concept de sous-catégorie, par exemple, un utilisateur recherchant des films de science-fiction pourra également découvrir des films de sous-genres comme la dystopie ou le space opera en utilisant les mots-clés appropriés. De

plus, les utilisateurs pourront choisir librement d'autres mots-clés, sur le modèle des hashtags utilisés sur les réseaux sociaux tels qu'Instagram ou Twitter.

Tout utilisateur pourra créer une nouvelle discussion sur le forum de CineNet. Il sera possible en particulier d'annoncer des événements, dont les caractéristiques pourront être listées (date, lieu, prix, organisateurs, films au programme, nombre de places disponibles...). Les utilisateurs qui sont des personnes pourront indiquer être intéressés par un événement ou y participer de manière certaine (mais pas les deux à la fois). Pour les événements ayant déjà eu lieu, il sera possible d'archiver un ensemble de données telles que programme, nombre de participants, liens vers des page web évoquant l'événement, ... Comme sur discord, les utilisateurs auront la possibilité de réagir à des publications non seulement en y répondant, mais également en utilisant des emojis.

En plus de permettre des discussions sur le thème du cinéma, CineNet proposera des fonctionnalités avancées de recherche d'événement. Les utilisateurs pourront faire des recherches parmi les événements passés ou à venir. Ainsi, une salle souhaitant organiser une rétrospective du cinéma d'horreur pourra vérifier que les films programmés n'ont pas été projetés ailleurs trop récemment. Les utilisateurs pourront également faire des recherches sur les événements à venir et accéder par exemple à ceux auxquels leurs amis ont prévu d'assister.

Plusieurs services seront proposés aux utilisateurs pour consulter leur historique, ainsi que certaines suggestions (événements auxquels ils pourraient assister, genres cinématographiques qui pourraient leur plaire, personnes avec lesquelles ils pourraient avoir des affinités, villes dans lesquelles ils pourraient trouver à s'amuser,...). Les émojis utilisés par les utilisateurs en réponses aux publications pourront évidemment être utilisés afin d'orienter au mieux les recommandations.

Concevez une base de données pour gérer CineNet. Vous êtes libre de vous inspirer de fonctionnalités de plateformes existantes, tout en ajoutant vos propres fonctionnalités uniques. Vous commencerez par réaliser le modèle conceptuel de données (schéma entités-associations) pour le Rendu 1, puis vous pourrez passer à l'implémentation dans PostgreSQL pour le Rendu 2. Utilisez les différents outils étudiés en cours et en TP (clés primaires/étrangères, contraintes d'intégrité, types adaptés ...). Expliquez dans le rapport (Rendu 2bis) et lors de la soutenance (Rendu 3) comment vous avez conçu cette base de données, les choix que vous avez effectués, les avantages, limites et améliorations possibles de votre modèle, etc. Prenez bien garde à dégager les contraintes externes, i.e., veillez à bien séparer ce qui relèvera du SGBD et ce qui relèvera de l'application et précisez-le dans le rapport. Il est normal que vous ne puissiez pas implémenter toutes les contraintes externes au niveau du SGBD avec ce que nous avons appris pour le moment. Vous êtes libres d'implémenter des triggers si vous le souhaitez, mais ce n'est pas demandé et pas au programme de cette année.

2 Peuplez vos tables

Maintenant que vos tables sont créées, vous pouvez les remplir avec les informations adaptées (films, casting, utilisateurs, cinéma, discussions, publications...). Insérez au moins une centaine de tuples au total dans votre base de données. Évidemment, le contenu textuel des publications ne doit pas nécessairement avoir un sens. Vous pouvez néanmoins utiliser un générateur de contenu textuel du type lorem ipsum ou même chatGPT. Les tables seront alimentées à partir de fichiers csv. Vous mettrez tous les fichier csv utilisés dans un répertoire nommé CSV afin de ne pas les mélanger avec les scripts sql. Vous pourrez alimenter vos tables avec la technique qui consiste à :

- importer les fichier csv avec COPY dans des tables temporaires,
- « oublier » les attributs inutiles en supprimant les colonnes via `ALTER TABLE ... DROP COLUMN`,
- alimenter vos tables à partir de ces tables temporaires, par exemple former des couples nom, prénom à partir de deux tables, une avec des prénoms et une autres avec des noms.

Bien sûr le travail le plus ingrat consiste à produire les fichiers csv avec les données. Vous pouvez par exemple écrire dans votre langage de programmation préféré (Java, Python, ...) une boucle for qui affiche des tuples générés automatiquement (avec la syntaxe attendue par PostgreSQL), et les recopier ensuite dans votre code PostgreSQL.

Vous pouvez aussi utiliser directement les structures de contrôle intégrées à PostgreSQL :

`https://docs.postgresql.fr/current/plpgsql-control-structures.html`

Vous pouvez également utiliser des outils de génération de données en ligne tels que :

— `https://www.mockaroo.com/`,

— `https://generatedata.com/`,

— etc.

Les plus courageux peuvent récupérer des données réelles. Vous pouvez par exemple consulter :

— `https://developer.imdb.com/non-commercial-datasets/`

— `https://www.kaggle.com/code/ruchi798/movies-and-tv-shows-eda`

— etc.

Attention, il est irréaliste de trouver sur internet des fichiers parfaitement prêts à être utilisés tels quels. La récupération et le nettoyage de vraies données peut être un exercice fastidieux et se fait d'habitude au moyen de librairies spécialisées (R et Python sont les languages typiquement utilisés pour ces tâches, voir par exemple `https://www.simplilearn.com/tutorials/data-analytics-tutorial/spotify-data-analysis-project`). Attention notamment au format des données. Si vous souhaitez vraiment récupérer de vraies données, le plus simple sera probablement de récupérer des données au format csv.

3 Effectuez des requêtes

Il est maintenant temps d'utiliser votre base de données pour faire vivre votre réseau social. Imaginez 20 questions sur la base de données que vous avez modélisée, et écrivez des requêtes SQL permettant d'y répondre. L'originalité des questions et la difficulté des requêtes (si tant est que celle-ci soit nécessaire) seront prises en compte dans la notation. Parmi vos requêtes, il faut au minimum :

— une requête qui porte sur au moins trois tables ;

— une 'auto jointure' (jointure de deux copies d'une même table)

— une sous-requête corrélée ;

— une sous-requête dans le FROM ;

— une sous-requête dans le WHERE ;

— deux agrégats nécessitant GROUP BY et HAVING ;

— une requête impliquant le calcul de deux agrégats (par exemple, les moyennes d'un ensemble de maximums)

— une jointure externe (LEFT JOIN, RIGHT JOIN ou FULL JOIN) ;

— deux requêtes équivalentes exprimant une condition de totalité, l'une avec des sous requêtes corrélées et l'autre avec de l'agrégation

— deux requêtes qui renverraient le même résultat si vos tables de contenaient pas de nulls, mais qui renvoient des résultats différents ici (vos données devront donc contenir quelques nulls), vous proposerez également de petites modifications de vos requêtes (dans l'esprit de ce qui a été présenté en cours) afin qu'elles retournent le même résultat

— Une requête récursive (par exemple, une requête permettant de calculer quel est le prochain jour sans événement d'un cinéma) ;

— Une requête utilisant le fenêtrage (par exemple, pour chaque mois de 2023, les dix cinémas dont les événements ont eu le plus de succès ce mois-ci, en termes de nombre d'utilisateurs ayant indiqué y participer)..

4 Vers un algorithme de recommandation

Si vous utilisez Facebook, vous avez peut-être déjà expérimenté sa fonctionnalité de recommandation d'événements. En fonction par exemple de votre lieu de résidence et de l'historique des événements suivis par vous-même et votre réseau, des événements à venir susceptibles de vous intéresser peuvent vous être proposés. De manière similaire, on souhaite ici pouvoir recommander aux utilisateurs des événements,

publications, films... susceptibles de les intéresser. Vous vous renseignerez d'abord sur les techniques de filtrage collaboratif utilisées dans les systèmes de recommandation. Vous concevrez ensuite quelques requêtes sur lesquelles vous pourrez baser vos recommandations (que vous pourrez bien sûr proposer dans la section précédente !). Vous pourrez même proposer un indice de recommandation pour chaque événement à venir, publication ou autre en fonction de ces requêtes. Vous calculerez cet indice selon la méthode et les critères de votre choix. Vous pourrez ensuite l'utiliser pour aider à la recommandation de publications et événements. Ceci pourra par exemple vous permettre de proposer une requête récupérant les n événements ou publications avec l'indice le plus élevé (à suggérer en priorité) pour un utilisateur donné à un moment donné. Expliquez dans le rapport (Rendu 3) le fonctionnement de votre indice et les raisons qui vous ont guidées dans votre choix. Détaillez ses points forts, ses limites et comment l'améliorer.

5 Conseils pour la présentation

Pour rendre votre projet plus interactif, vous pouvez préparer des requêtes dont certains paramètres seront fournis par l'utilisateur. Par exemple, pour obtenir le réalisateur d'un film donné, on pourrait considérer la requête suivante :

```
SELECT realisateur
FROM film
WHERE titre='8 miles';
```

Mais comment faire si le film n'est pas connu d'avance ?

Préparation de requêtes paramétrées Vous pouvez créer une requête paramétrée en utilisant la syntaxe suivante.

```
PREPARE recherche_par_titre(VARCHAR) as
SELECT realisateur
FROM film
WHERE titre=$1;
```

Vous pouvez ajouter autant de paramètres que vous lsouhaitez. Dans la requête, vous utiliserez \$1, \$2... pour en obtenir la valeur. La requête est préparée mais pas exécutée. Pour l'exécuter, on tape sous `psql` :

```
EXECUTE recherche_par_titre('8 miles');
```

La requête est alors exécutée avec la valeur '8 miles' à la place du paramètre \$1.

Variables d'environnement et prompt La commande `prompt` de `psql` permet de faire la saisie au clavier. La valeur est sauvegardée dans une variable d'environnement (`t_titre` dans l'exemple qui suit).

```
\prompt 'Tapez le nom de film -> ' t_titre
SELECT realisateur
FROM film
WHERE titre = :t_titre;
```

Lorsque la requête est exécutée, `:t_titre` prend la valeur de la variable d'environnement `t_titre`.

Tous ces éléments permettent de faire un script interactif que vous exécuterez au cours de votre présentation.

Manipulation des dates Le type Postgres `DATE` permet de stocker une date et le type `TIME` permet de stocker l'heure. Il existe également un type `TIMESTAMP` qui contient la date et l'heure.

Une chaîne peut être convertie en type `DATE` avec la fonction `to_date(chaine_a_convertir, format)`, où `format` est une chaîne de la forme `'YYYYMMDD'` qui indique le format dans lequel est donnée la date. On peut aussi écrire `date '2015-03-05'` pour obtenir une date. *Attention, selon la configuration de postgres, la date '12-11-2015' peut être interprétée comme le 12 novembre, ou le 11 décembre.*

On peut ajouter un entier n à une date pour obtenir la date n jours plus tard, ou soustraire une date à une autre pour connaître le nombre de jours entre les deux dates.

Plusieurs fonctions prédéfinies en postgres permettent de manipuler les données de type `DATE, TIME, TIMESTAMP`.

Fonction	Usage	Exemple
`EXTRACT`	extraire un jour/mois/... d'une date ou d'un timestamp	`EXTRACT (DAY FROM madate)`
`INTERVAL`	préciser une unité de temps (jour/heure/semaine...)	`heureRDV + INTERVAL '2 hours'`
`CURRENT_DATE`	Obtenir la date du jour	
`CURRENT_TIME`	Obtenir l'heure actuelle	

On peut énumérer des valeurs dans un intervalle avec la fonction `generate_series`. Par exemple, `generate_series(1,5)` énumère les valeurs 1, 2, 3, 4, 5. `generate_series(1,5,2)` énumère les valeurs de 1 à 5 avec un pas de 2 : 1, 3, 5. `generate_series(current_date+time '10:00', current_date+time '16:00' , interval '30 minutes')` énumère les créneaux de 30 minutes aujourd'hui entre 10h et 16h.