Rapport Base De Données

Nom	Prénom	Numéro Etudiant
BOUDEDJA	Naila	22310785
BELGUENBOUR	Manel	22216606

Année universitaire : 2023 / 2024

Table des matières

*

Introduction

CineNet est une plateforme qui combine les fonctionnalités d'une base de données cinématographique participative avec les interactions sociales d'un réseau social et d'un forum de discussion.

La plateforme permet aux utilisateurs d'explorer un vaste catalogue d'œuvres cinématographiques tout en participant activement à des discussions et des événements autour du monde du cinéma.

Les utilisateurs peuvent être des personnes lambda, des acteurs, des réalisateurs, des studios, des organisateurs de festivals, ou encore des clubs et salles de cinéma.

Ils peuvent interagir entre eux en suivant d'autres utilisateurs ou en établissant des relations d'amitié. Ils peuvent également créer et participer à des événements, d'écrire et répondre à des publications, et d'utiliser des mots-clés pour faciliter les recherches et les discussions.

Modèle Entités-Associations et Schéma Relationnel

Pour modéliser Cinenet, nous avons d'abord conçu un modèle Entités-Associations pour représenter les différentes entités du système et leurs relations.

Les principales entités incluent les utilisateurs, les films, les séries, les événements particuliers, les publications, les discussions, les relations sociales et les mots-clés.

Voici le modèle E-A :

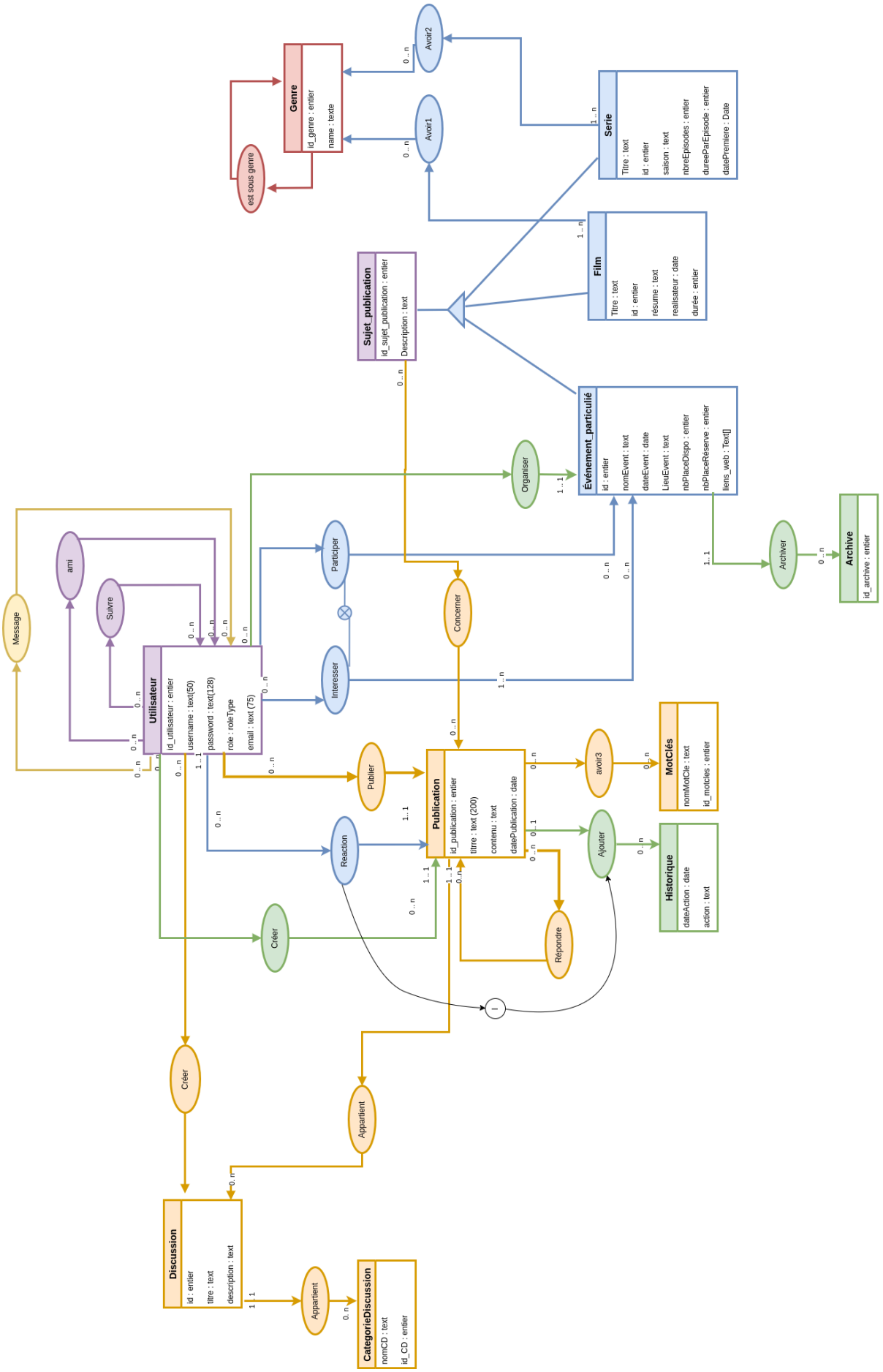
Genre
id_genre : entier
name : texte
est sous genre
Avoir2
Avoir1
Serie
Titre : text
id : entier
saison : text
nbreEpisodes : entier
dureeParEpisode : entier
datePremiere : Date
Film
Titre : text
id : entier
résume : text
realisateur : date
durée : entier
Sujet_publication
id_sujet_publication : entier
Description : text
Événement_particulié
id : entier
nomEvent : text
dateEvent : date
LieuEvent : text
nbPlaceDispo : entier
nbPlaceRéserve : entier
liens_web : Text[]
Organiser
Archiver
Archive
id_archive : entier
Participer
Concerner
Interesser
Message
ami
Suivre
Utilisateur
id_utilisateur : entier
username : text(50)
password : text(128)
role : roleType
email : text (75)
Publier
Publication
id_publication : entier
titre : text (200)
contenu : text
datePublication : date
avoir3
MotClés
nomMotCle : text
id_motcles : entier
Reaction
Ajouter
Historique
dateAction : date
action : text
Répondre
Créer
Créer
Appartient
Discussion
id : entier
titre : text
description : text
Appartient
CategorieDiscussion
nomCD : text
id_CD : entier

Schéma Relationnel

Le passage du modèle E-A au schéma relationnel a été effectué en transformant les entités en tables et les associations en clés étrangères. Voici le schéma relationnel de nos tables :

Utilisateur (*id_utilisateur*, nom_utilisateur, mot_de_passe, id_role, email)

- Utilisateur[id_role] ⊆ Role[id_role]
- Unique email

Role (id_*role*, nom_role)

Ami (*utilisateur_id_1*, *utilisateur_id_2*)

- Ami[utilisateur_id_1] ⊆ Utilisateur[id_utilisateur]
- Ami[utilisateur_id_2] ⊆ Utilisateur[id_utilisateur]
- Contraintes :
  - *utilisateur_id_1 != utilisateur_id_2*
  - Si le couple (a,b) --> pas de (b,a)

Suivre (*utilisateur_id_suiveur*, *utilisateur_id_suivi*)

- Suivre[utilisateur_id_suiveur] ⊆ Utilisateur[id_utilisateur]
- Suivre[utilisateur_id_suivi] ⊆ Utilisateur[id_utilisateur]
- Contraintes :
  - *utilisateur_id_suiveur != utilisateur_id_suivi*

Discussion (*id_discussion*, id_cd)

- Discussion[id_cd] ⊆ CategorieDiscussion[id_cd]

CategorieDiscussion (*id_cd*, nom_cd)

- Unique nom_cd

Publication (*id_publication*, description, utilisateur_id, discussion_id)

- Publication[utilisateur_id] ⊆ Utilisateur[id_utilisateur]
- Publication[discussion_id] ⊆ Discussion[id_discussion]

Reaction (*utilisateur_id*, *publication_id*)

- Reaction[utilisateur_id] ⊆ Utilisateur[id_utilisateur]
- Reaction[publication_id] ⊆ Publication[id_publication]

MotCle (*id_mot_cle*, nom_mot_cle)

- Unique nom_mot_cle

EvenementParticulier (*id_evenement*, date, prix, lieu)

- Unique (titre, date, lieu)

Participer (*utilisateur_id*, *evenement_id*, participation_type)

- Participer[utilisateur_id] ⊆ Utilisateur[id_utilisateur]
- Participer[evenement_id] ⊆ EvenementParticulier[id_evenement]

Archive (*id_archive*, evenement_id)

- Archive[evenement_id] ⊆ EvenementParticulier[id_evenement]
- Unique evenement_id

Genre (*id_genre*, description, id_genre_parent)

- Genre[id_genre_parent] ⊆ Genre[id_genre]
- Unique description

Film (*id_film*,titre, resume, durée, realisation)

Serie (*id_serie*,titre, resume, durée, saison)

FilmGenre (*film_id*, genreId)

- FilmGenre[film_id] ⊆ Film[id_film]
- FilmGenrer[genreId] ⊆ Genre[id_genre]

SerieGenre (*serie_id*, id_genre)

- SerieGenre[*serie_id*] ⊆ Serie[id_serie]
- SerieGenre[id_genre] ⊆ Genre[id_genre]

Message (*utilisateur_id_1*, *utilisateur_id_2, message*)

- Message[utilisateur_id_1] ⊆ Utilisateur[id_utilisateur]
- Message[utilisateur_id_2] ⊆ Utilisateur[id_utilisateur]
- Contraintes :
  - *utilisateur_id_1 != utilisateur_id_2*

Les clés primaires sont soulignées, et les clés étrangères sont ajoutées pour maintenir les relations entre les tables.

Choix de Modélisation

Lors de la modélisation de CineNet, nous avons fait des choix structurants pour assurer la cohérence, la flexibilité et l'efficacité des opérations sur la base de données. Voici une description des principaux choix de modélisation et des raisons qui les motivent :

Utilisateur

- Choix : Nous avons inclus les informations essentielles pour identifier et authentifier les utilisateurs. L'email est unique pour garantir qu'un utilisateur ne peut avoir plusieurs comptes avec la même adresse.
- Raison : il est crucial de pouvoir les identifier de manière unique et sécurisée.

Ami

- Choix : La table Ami modélise les relations d'amitié entre utilisateurs avec des contraintes pour éviter les doublons (a, b) et (b, a).
- Raison : Les relations d'amitié sont symétriques et il est important de maintenir l'intégrité de ces relations.

Suivre

- Choix : Cette table capture les relations de suivi, où un utilisateur peut suivre un autre sans que la réciproque soit vraie.

Discussion et CategorieDiscussion

- Attributs Discussion : id_discussion, id_cd
- Attributs CategorieDiscussion : id_cd, nom_cd
- Choix : Les discussions sont catégorisées pour permettre une organisation et une navigation plus facile.
- Raison : Catégoriser les discussions améliore l'expérience utilisateur en permettant des filtres et des recherches plus efficaces.

Publication

- Choix : Les publications sont liées aux utilisateurs et aux discussions,
- Raison : Il est important de savoir qui a publié quoi et dans quel contexte pour permettant une traçabilité, une organisation claire et modérer et rechercher efficacement.

Reaction

- Attributs : utilisateur_id, publication_id
- Choix : Les réactions (comme, dislike, etc.) sont liées aux publications et aux utilisateurs qui les ont faites.
- Raison : Les réactions sont des interactions essentielles sur les réseaux sociaux, et il est important de pouvoir les suivre.

MotCle

- Attributs : id_mot_cle, nom_mot_cle
- Choix : Les mots-clés sont uniques pour garantir une indexation et une recherche efficaces.
- Raison : Les mots-clés permettent de taguer les publications, facilitant les recherches

EvenementParticulier

- Attributs : id_evenement, date, prix, lieu, nb_places_dispo, nb_places_reservees
- Choix : Chaque événement a un nombre de places disponibles et réservées, et des contraintes uniques sur (titre, date, lieu) pour éviter les doublons.
- Raison : Les événements sont des entités centrales dans CineNet, et il est crucial de gérer les réservations et d'éviter les conflits de planification.

Participer et Intéresser

- Choix : Ces tables capturent la participation et l'intérêt des utilisateurs pour les événements
- Raison : Différencier les rôles de participant et intéresser permet une gestion plus fine des événements.

Film et Serie

- Attributs Film : id_film, titre, resume, duree, realisation
- Attributs Serie : id_serie, titre, resume, duree, saison
- Choix : Les films et séries sont modélisés séparément mais peuvent partager des genres.
- Raison : Bien que similaires, les films et séries ont des caractéristiques différentes (comme les saisons pour les séries).

Limites du Modèle

- Contraintes Complexes : Certaines contraintes d'intégrité nécessitent des vérifications au-delà des capacités des contraintes de clé étrangère et de clés uniques, par exemple :
    - Contraintes sur les Relations d'Amitié : La relation d'amitié doit être symétrique (si A est ami avec B, alors B est ami avec A), et il ne doit pas y avoir de doublons inversés.
    - Exclusion Mutuelle : Entre certaines actions, par exemple, un utilisateur ne peut pas être à la fois intéressé et inscrit à un événement. Assurer ce type de contrainte complexe peut nécessiter des triggers pour vérifier l'intégrité à chaque insertion ou mise à jour, car nous avons choisie de faire deux tables différentes
- Flexibilité vs. Normalisation :
    - Bien que la normalisation soit cruciale pour éviter la redondance et maintenir l'intégrité, elle peut parfois rendre les requêtes complexes et lourdes, par exemple
    - Jointures Multiples : Les tables normalisées nécessitent souvent des jointures complexes pour obtenir des informations liées. Par exemple,
- récupérer tous les détails d'un événement et ses participants peut nécessiter de joindre plusieurs tables (Utilisateur, EvenementParticulier, Participer, etc.).

- Performances : Bien que la normalisation réduise la redondance, elle peut affecter les performances des requêtes en raison des multiples jointures nécessaires pour obtenir les données requises.
- Une solution :
    - Utilisation de vues matérialisées pour stocker les résultats des requêtes fréquentes.
- Scalabilité : Avec l'augmentation du nombre d'utilisateurs et d'événements, certaines tables pourraient devenir très volumineuses.

Passage du Schéma Conceptuel au Schéma Logique

Le passage du modèle conceptuel (E-A) au schéma logique (relationnel) a impliqué plusieurs étapes :

- Identification des Entités et Attributs : Chaque entité du modèle E-A a été transformée en table, et chaque attribut est devenu une colonne, sauf pour la table Role où s'est devenu une énumération

- Définition des Clés Primaires et Étrangères : Les relations entre entités ont été représentées par des clés étrangères. Les identifiants uniques des entités sont devenus des clés primaires pour certain pour d'autre nous avons choisie d'ajouter un id comme clé primaire et garder l'attribut unique,
    - Par exemple dans la table Genre nous avons ajouté un id et garder le nom unique

- Normalisation : Les tables ont été normalisées pour éviter les redondances. Cela a impliqué la décomposition de certaines tables en plusieurs tables.

- Implémentation des Contraintes : Les contraintes d'intégrité référentielle (comme les contraintes de clés étrangères) et les contraintes de validation (comme l'unicité et les vérifications de valeurs) ont été ajoutées pour assurer la cohérence des données.

- Ajout de Triggers : Des triggers ont été ajoutés pour gérer les contraintes complexes et les opérations automatiques qui ne peuvent pas être exprimées directement dans le schéma relationnel.

Requêtes Implémentées

Nous avons implémenté plusieurs requêtes pour répondre aux besoins de recherche des utilisateurs de CineNet, elles sont toutes listées dans le fichier **requete_para.sql** :

- Recherche d'Événements : Requêtes pour rechercher des événements passés ou à venir, des événements organisés par un utilisateur spécifique, ou des événements auxquels les amis d'un utilisateur participent.
- Recherche de Publications : Requêtes pour rechercher des publications par mot-clé ou par utilisateur.

Nous avons rajouté d'autres requêtes pour répondre à la demande de 20 questions sur la base de données. Elles sont toutes listées dans le fichier **requetes.sql**

Indice de recommandation

La requête SQL décrite fait partie d'un algorithme de recommandation avancé pour un système de gestion de contenus médiatiques, structuré autour de trois principaux indices de popularité et de préférences personnelles. Voici le déroulement et la fonction de chaque étape de cet algorithme :

1. Réactions Positives

La première étape, PositiveReactions, calcule le nombre total de réactions positives'Like', 'Fun', 'Love' pour chaque publication, excluant celles déjà vues par l'utilisateur. Ce filtre garantit que les recommandations sont à la fois pertinentes et novatrices pour l'utilisateur.

2. Engagement dans les Événements

L'indice EventInterestParticipation évalue l'engagement autour des événements associés aux publications, en comptant le nombre d'utilisateurs intéressés et participants. Cet indice reflète la popularité de l'événement et aide à identifier les tendances qui captivent l'audience cible.

3. Correspondance avec les Préférences de l'Utilisateur

Ce segment de l'algorithme se décompose en plusieurs sous-étapes :

- Historique Personnel : UserHistory identifie toutes les publications précédemment consultées par l'utilisateur, pour s'assurer que les suggestions soient nouvelles.
- Publications Préférées : UserLikedPublications filtre ces publications pour ne retenir que celles ayant reçu des réactions positives spécifiques de l'utilisateur, indiquant clairement ses goûts.
- Analyse des Contenus Appréciés : UserLikedFilmsSeries extrait les films et séries qui ont été particulièrement appréciés dans les publications aimées, formant une base pour identifier des contenus similaires susceptibles d'intéresser l'utilisateur.

- Genres Préférés : UserLikedGenres utilise cette information pour déterminer les genres favoris de l'utilisateur, ce qui permet de cibler plus précisément les futures recommandations.

4. Propositions de Nouveaux Contenus

La sous-requête RecommendedPublications utilise les genres préférés pour suggérer de nouvelles publications. Elle veille à ce que ces suggestions soient alignées avec les intérêts démontrés de l'utilisateur, augmentant ainsi les chances qu'elles soient bien reçues.

5. Calcul du Score de Recommandation

Enfin, l'algorithme combine ces trois indices pour calculer un score global de recommandation pour chaque publication. Les suggestions sont triées par ce score, présentant en premier les contenus les plus susceptibles de plaire à l'utilisateur.

Conclusion

Cet algorithme sophistiqué vise à personnaliser l'expérience utilisateur en adaptant dynamiquement les recommandations en fonction de ses interactions précédentes avec le système. En intégrant des mesures de réactions positives, d'engagement dans les événements et de préférences personnelles, il optimise les suggestions pour maximiser l'engagement et la satisfaction de l'utilisateur.