

Projet de BDD2

Hassan El Sahily - L3 CMI ISR
Numéro étudiant : 22217305

Janvier 2024

Table des matières

Contents

1 Modélisation de la base de données	3
1.1 Interprétation	3
1.2 Modèle Entité-Association	5
1.3 Contraintes d'intégrité	5
1.3.1 Statiques	5
1.3.2 Dynamiques	6
1.4 Modèle logique relationnel	7
1.5 Relations	7
1.6 Implémentation	9

1 Modélisation de la base de données

1.1 Interprétation

1. Les tables :

- (a) **EMISSION**
- (b) **CATEGORIES**
- (c) **VIDEOS** : Normalement, un épisode est une vidéo comme sur Netflix, Max, etc. J'ai considéré aussi que chaque épisode (vidéo) fait référence forcément à une émission.
- (d) **UTILISATEURS** : c'est la table de tous les utilisateurs.
- (e) **COMPTE_CREATED** : contient le numéro de compte pour les utilisateurs qui ont créé un compte.
- (f) **ARCHIVE_VIDEOS**

J'ai choisi que :

- seuls les utilisateurs ayant un compte peuvent marquer des vidéos comme favoris.
- seuls les utilisateurs ayant un compte peuvent regarder des vidéos. Par conséquent, seuls eux possèdent un historique de visionnage. Cela correspond à des plateformes où, comme sur certains sites, il n'est pas possible de regarder une série sans avoir un compte ou payer, mais on peut tout de même consulter les catégories disponibles.
- **DATE_HISTORIQUE** dans la table **HISTORIQUE_VISIONNAGE** ainsi obtenu dans le modèle relationnel soit une des clés primaires du fait que un utilisateur peut regarder la même vidéo plusieurs fois.

2. Les relations :

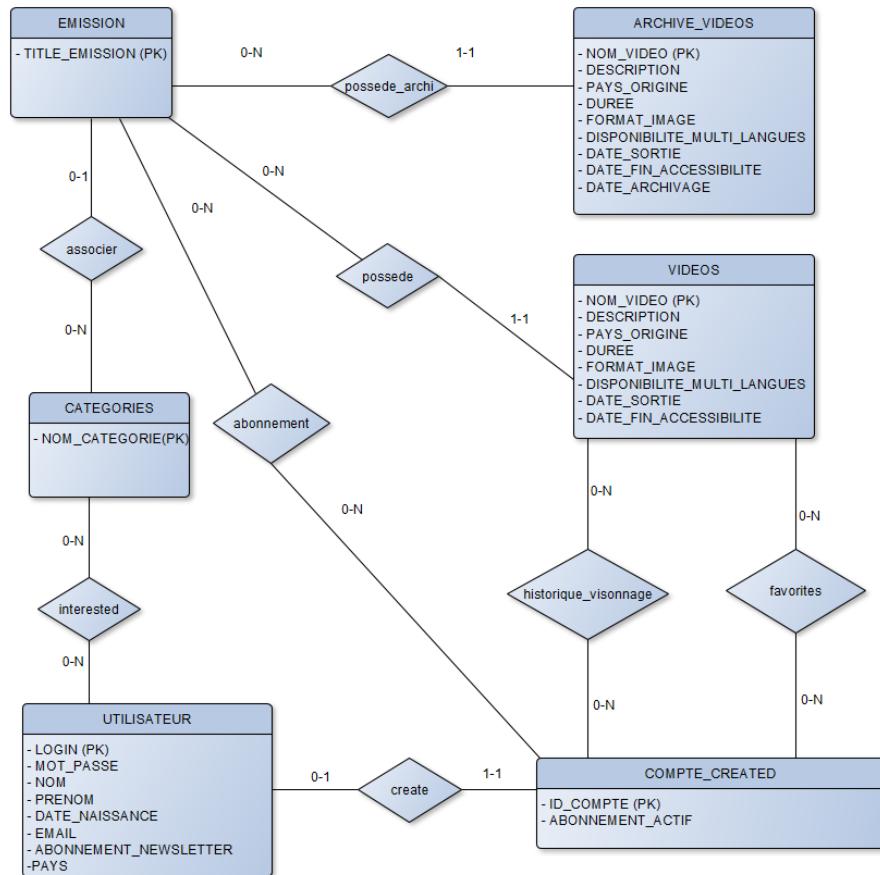
- (a) **associer (EMISSION - CATEGORIES)** : Pour chaque émission, on peut associer au plus une et une seule catégorie. Une catégorie peut appartenir à plusieurs émissions, d'où une cardinalité de 1-N.
- (b) **possede (EMISSION - VIDEOS)** : Chaque émission peut contenir plusieurs épisodes, donc plusieurs vidéos, mais une vidéo appartient à une et une seule émission, d'où une cardinalité de 1-N.
- (c) **possede_archi (EMISSION - ARCHIVE_VIDEOS)**
- (d) **interested (CATEGORIES - UTILISATEURS)** : Un utilisateur peut indiquer les catégories qui l'intéressent, et une catégorie peut intéresser plusieurs utilisateurs, d'où une cardinalité N-N.
- (e) **create (UTILISATEURS - COMPTE_CREATED)** : J'ai considéré qu'un utilisateur peut créer un et un seul compte et que le compte appartient forcément à un et un seul utilisateur.

- (f) **abonnement (EMISSION - COMPTE_CREATED)** : Chaque utilisateur ayant un compte peut s'abonner à plusieurs émissions. Cardinalité: N-N.
- (g) **favorites(COMPTE_CREATED - VIDEOS)** : Cardinalité N-N.
- (h) **historique_visionnage (COMPTE_CREATED - VIDEOS)** : Chaque utilisateur peut aussi voir des vidéos. Cardinalité N-N.

3. Vues :

- (a) **Suggestions générales de vidéos** : Elles doivent toujours être basées sur les visionnages des deux dernières semaines. Une vue garantit qu'on n'a pas à mettre à jour manuellement les données, et il n'y a absolument aucun intérêt à les stocker physiquement.
Obtenue par une jointure entre **EMISSION**, **VIDEOS** et **HISTORIQUE_VISIONNAGE**.
- (b) **Suggestions personnalisées de vidéos** : Elles doivent inclure les nouveaux épisodes des émissions auxquelles l'utilisateur est abonné. Elles ont un caractère évolutif et doivent être recalculées en temps réel.
Obtenues par une jointure entre **UTILISATEUR**, **COMPTE_CREATED**, **ABONNEMENT** et **VIDEOS**.

1.2 Modèle Entité-Association



1.3 Contraintes d'intégrité

1.3.1 Statiques

1. EMISSION

- (a) TITLE_EMISSION : clé primaire.

2. VIDEOS

- (a) NOM_VIDEO : clé primaire.
- (b) NOM, DATE_SORTIE, DISPONIBILITE_MULTI_LANGUES, DATE_FIN_ACCESSIBILITE, FORMAT_IMAGE, DUREE, PAYS_ORIGINE : NOT NULL.
- (c) DATE_FIN_ACCESSIBILITE - DATE_SORTIE >= 7.
- (d) DESCRIPTION : par défaut ''.

3. **ARCHIVE_VIDEOS** : C'est le même que la table VIDEOS mais avec DATE_ARCHIVAGE NOT NULL aussi.

4. CATEGORIES

(a) NOM_CATEGORIE : clé primaire.

5. UTILISATEURS

(a) LOGIN : clé primaire.

(b) MOT_PASSE, NOM, PRENOM, EMAIL, PAYS : NOT NULL.

(c) ABONNEMENT_NEWSLETTER : par défaut FALSE.

(d) DATE_NAISSANCE <= SYSDATE and NOT NULL.

6. COMPTE_CREATED

(a) ID_COMPTE : clé primaire, auto-incrémenté.

(b) ABONNEMENT_ACTIF : par défaut TRUE (quand on crée un compte, on paye, mais à tout moment on peut annuler l'abonnement).

(c) la clé étrangère LOGIN obtenue de la relation `create` sera non pas juste NOT NULL mais aussi UNIQUE(un utilisateur peut avoir 1 et 1 seul compte)

1.3.2 Dynamiques

1. VIDEOS

Gestion de la disponibilité et de l'archivage des vidéos : "Après la sortie de la vidéo, une vidéo sera accessible sur le site en replay pendant au moins 7 jours. Après un certain temps, certaines vidéos ne seront plus disponibles au visionnage en replay. Elles seront alors archivées".

Pour gérer cela :

(a) Je voulais initialement définir un job qui s'exécute périodiquement pour vérifier la disponibilité des vidéos et les archiver si nécessaire. Cependant, en tant qu'utilisateur Oracle, je n'ai pas les privilèges suffisants pour créer un job avec DBMS_SCHEDULER.

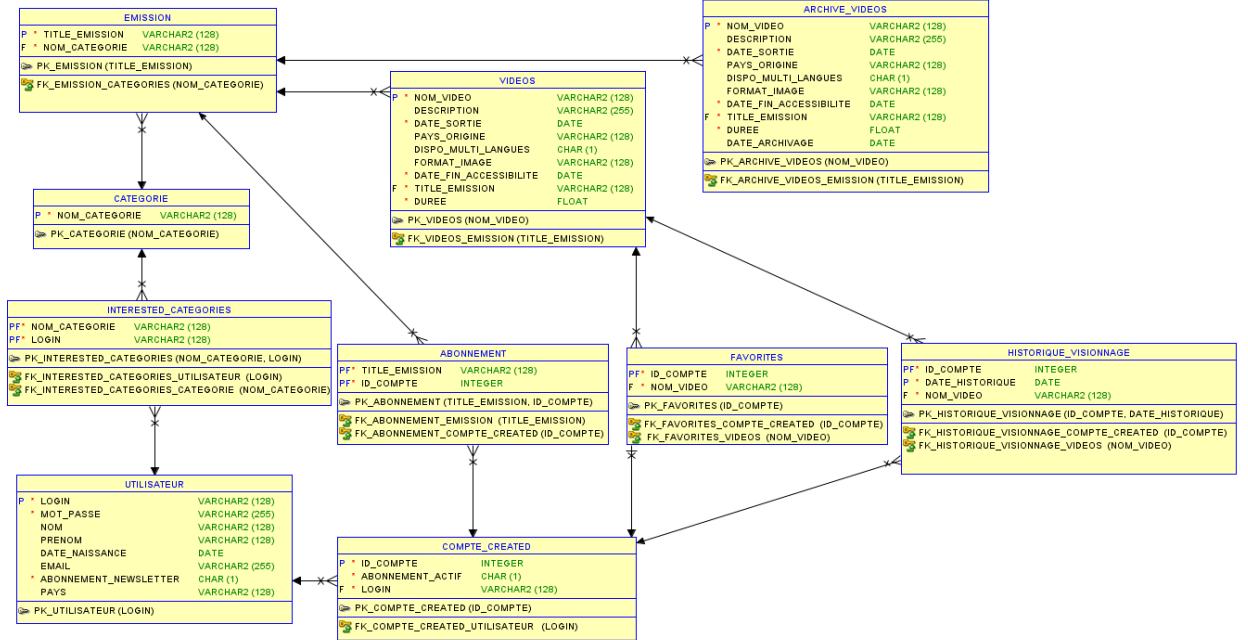
(b) En alternative, j'ai :

- Documenté cette logique à la fin du fichier `creation_table.sql`(en commentaire).
- Défini un trigger sur la table VIDEOS `trg_auto_archive_videos` contenant un curseur pour récupérer les données à archiver suite à une insertion ou une mise à jour des données. Ce dernier supprime le video de la table VIDEOS, ce qui provoque l'appel de trigger `trg_archive_video` qui insère ce dernier dans la table ARCHIVE_VIDEOS juste avant la suppression.

2. ABONNEMENT, FAVORITES et HISTORIQUE_VISIONNAGE

Avant chaque insertion dans ces tables, il faut s'assurer que l'abonnement du compte est actif.

1.4 Modèle logique relationnel



1.5 Relations

EMISSION(TITLE_EMISSION : STRING, NOM_CATEGORIE : STRING)

- Clé primaire : TITLE_EMISSION.
- Clé étrangère : NOM_CATEGORIE référence à CATEGORIE(NOM_CATEGORIE).

CATEGORIE(NOM_CATEGORIE : STRING)

- Clé primaire : NOM_CATEGORIE.

INTERESTED_CATEGORIES(LOGIN : STRING, NOM_CATEGORIE : STRING)

- Clé primaire : (NOM_CATEGORIE : STRING, LOGIN : STRING)
- Clé étrangère :
 - NOM_CATEGORIE référence à CATEGORIE(NOM_CATEGORIE)

– LOGIN réfère à UTILISATEUR(LOGIN)

UTILISATEUR(LOGIN : STRING, MOT_PASSE : STRING, NOM : STRING, PRENOM : STRING, DATE_NAISSANCE : DATE, EMAIL : STRING, ABONNEMENT_NEWSLETTER : BOOLEAN, PAYS : STRING)

- Clé primaire : LOGIN

VIDEOS(TITLE_EMISSION : STRING, NOM_VIDEO : STRING, DESCRIPTION : STRING, DATE_SORTIE : DATE, PAYS_ORIGINE : STRING, DISPO_MULTI_LANGUES : BOOLEAN, FORMAT_IMAGE : STRING, DATE_FIN_ACCESSIBILTE : DATE, DUREE : FLOAT)

- Clé primaire : NOM_VIDEO

- Clé étrangère : TITLE_EMISSION réfère à EMISSION(TITLE_EMISSION)

ARCHIVE_VIDEOS(NOM_VIDEO : STRING, TITLE_EMISSION : STRING, DESCRIPTION : STRING, DATE_SORTIE : DATE, PAYS_ORIGINE : STRING, DISPO_MULTI_LANGUES : BOOLEAN, FORMAT_IMAGE : STRING, DATE_FIN_ACCESSIBILTE : DATE, DUREE : FLOAT, DATE_ACRHIVAGE : DATE)

- Clé primaire : NOM_VIDEO

- Clé étrangère : TITLE_EMISSION réfère à EMISSION(TITLE_EMISSION)

ABONNEMENT (TITLE_EMISSION : STRING, ID_COMPTE : INT)

- Clé primaire : (TITLE_EMISSION, ID_COMPTE)

- Clé étrangère :

– TITLE_EMISSION réfère à EMISSION(TITLE_EMISSION)

– ID_COMPTE réfère à COMPTE_CREATED(ID_COMPTE)

COMPTE_CREATED (ID_COMPTE : INT, LOGIN : STRING, ABONNEMENT_ACTIF : BOOLEAN)

- Clé primaire : ID_COMPTE

- Clé étrangère : LOGIN réfère à UTILISATEUR(LOGIN)

FAVORITES (ID_COMPTE : INT, NOM_VIDEO : STRING)

- Clé primaire : (ID_COMPTE,NOM_VIDEO)

- Clé étrangère :

– ID_COMPTE réfère à COMPTE_CREATED(ID_COMPTE)

– NOM_VIDEO réfère à VIDEOS(NOM_VIDEO)

HISTORIQUE_VISIONNAGE(ID_COMPTE : INT, NOM_VIDEO : STRING, DATE_HISTORIQUE : DATE)

- Clé primaire : (ID_COMPTE, NOM_VIDEO, DATE_HISTORIQUE)
- Clé étrangère :
 - ID_COMPTE réfère à COMPTE_CREATED(ID_COMPTE)
 - NOM_VIDEO réfère à VIDEOS(NOM_VIDEO)

1.6 Implémentation

- Pour générer la liste des vidéos populaires recommandées à un utilisateur (question 3 de Procédures et fonctions PL/SQL) , j'ai choisi d'utiliser une procédure. Cette procédure identifie le vidéo le plus regardé dans chaque catégorie suivie par l'utilisateur et les affiche.