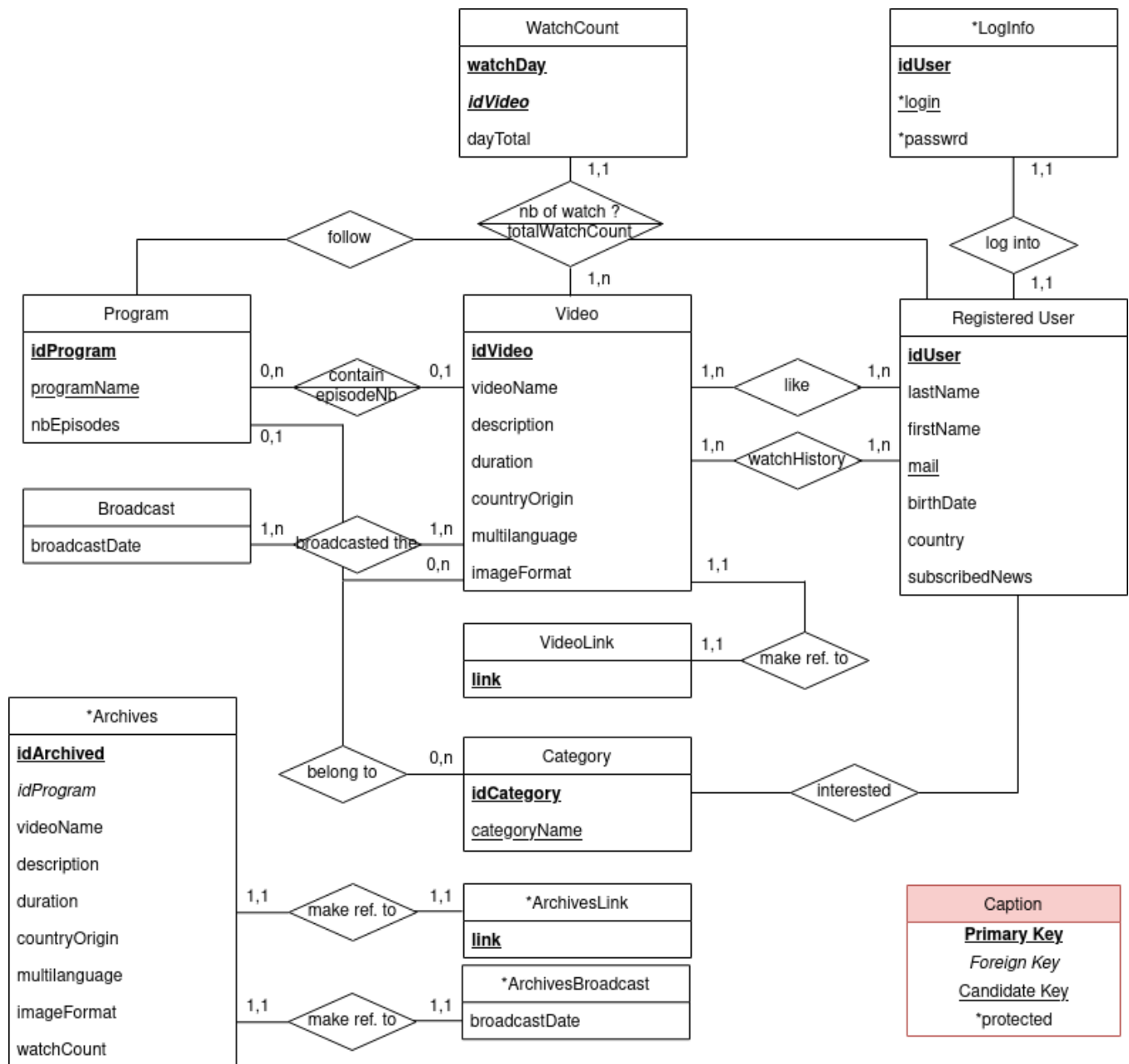


Rendu de projet BDD2 (part1)

1) Schema entite-association :



2) Formes Normales:

Notre schema relationnel suit la 1NF car chaque attribut possede une valeur atomique et les tables une cle primaire.

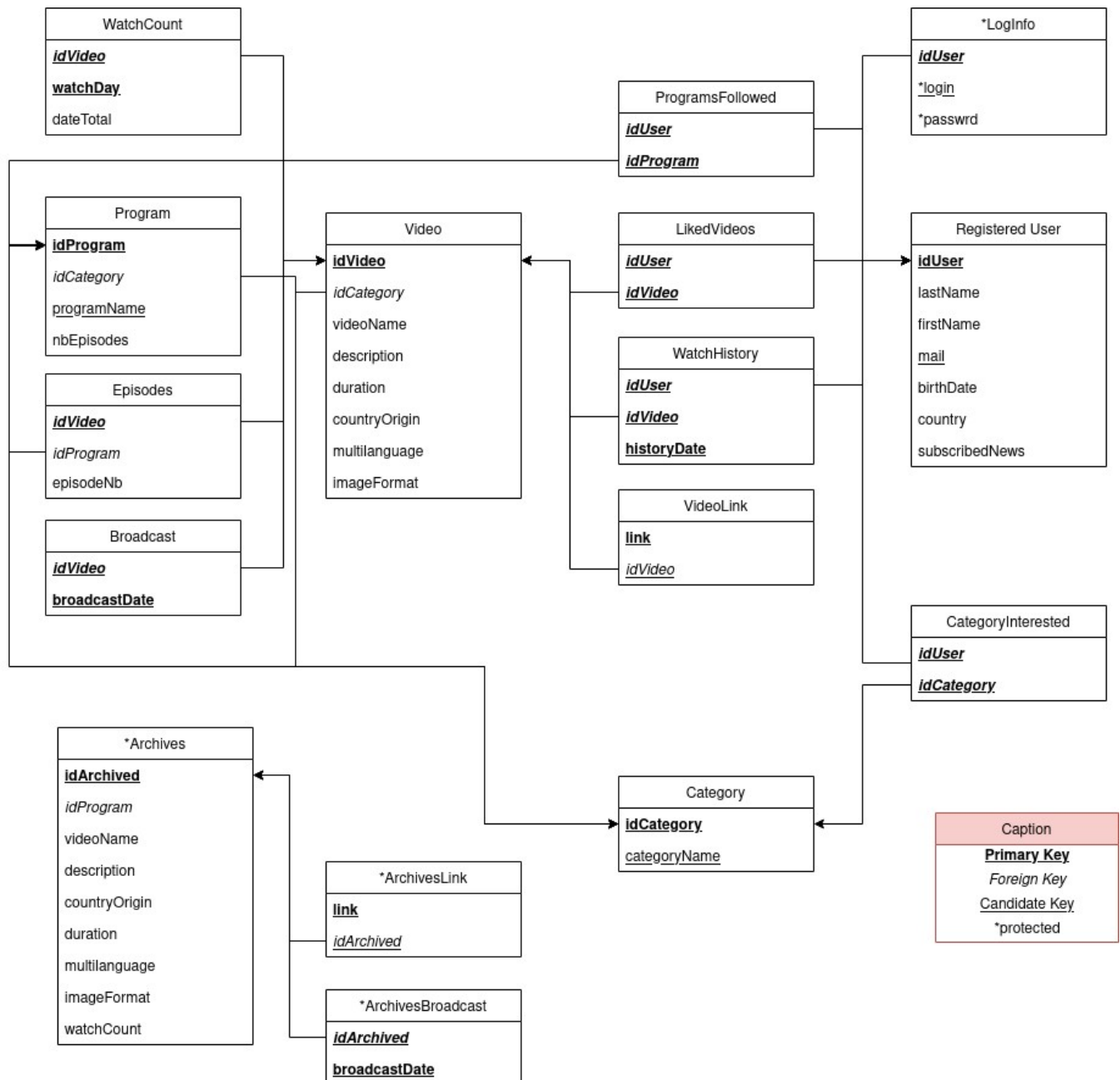
Il suit aussi la 2NF car tout attribut n'appartenant pas une cle ne depend d'une partie de la cle.

Il suit aussi la 3NF car tout n'appartenant pas a une cle, il ne depend pas d'un autre attribut non cle.

La BCNF est aussi respectee car aucune partie de cle ne depend d'une autre ou d'un autre attribut non cle.

Je n'ai pas cherche a conformer mes tables aux autres formes normales, car il me semblaient qu'elles n'etaient pas suffisamment nombreuses et complexes (ou trop remplies) pour justifier d'appliquer la 4Nf ou la 5Nf. De plus elles peuvent provoquer des pertes de dependances si on cherche a les appliquer.

3) Schema des tables :



4) Choix des tables:

Je vais revenir sur mes choix table par table dans cette section.

- **Video** : J'ai choisi de séparer le comptage de vues de la table video car pour pouvoir ne prendre en compte que les vues récentes il faut compter le nombre de vues selon les jours. Pour respecter la 1NF je dois donc faire une autre table.
J'ai décidé d'autoriser plusieurs vidéos à avoir le même nom en m'inspirant de la plateforme youtube qui l'autorise. Cela permet de garder une certaine liberté sur le nom de la vidéo sans avoir à forcément choisir un nom différent et en réalité, avec toutes les autres informations (image, category, description ...) il y a très peu de chances d'avoir exactement deux fois (ou plus) la même video.
- **Program** : J'ai choisi d'interdire de créer des programmes avec le même nom car sinon l'utilisateur risque de confondre les programmes en question et de plus cela traduit une réalité qui est que les programmes TV n'ont pas deux fois le même nom.
J'ai séparé la table épisodes de program encore une fois pour respecter la 1NF dans le cas (très courant) ou un program à plusieurs épisodes.
- **Episodes** : La colonne 'episodeNb' permet de trier les vidéos dans l'ordre logique de visionnage.
- **Broadcast** : Cette table est aussi séparée de Video pour respecter la 1NF.
- **Registered User** : On interdit à deux utilisateurs d'avoir la même adresse mail pour des raisons évidentes d'identification du compte.
- **Loginfo** : Cette table est faite à part de Registered_User pour permettre de restreindre les droits sur toutes les opérations à l'admin. La table est sensible car personne d'autre ne doit y avoir accès et les informations forment un bloc logique qui est mieux séparé du reste des informations sur l'utilisateur.
- **Programs Followed | Liked Videos | Watch History | Category Interested** : Ces tables servent à traduire les relations homonymes de la table Registered_User vers Program | Video | Category (respectivement).
- **Category** : Table qui est référencée par Video et Program qui ne peuvent avoir qu'une seule category.
- **Archives*** : La table Archives reçoit toutes les entrées de la table Vidéo en plus de la colonne 'idProgram' et 'nbViews' pour pouvoir restaurer les vidéos archivées après qu'elle soit repassée à la télé. Les autres tables remplacent aussi les autres tables de nom semblable.

J'ai fait des séquences sur les id de la table Video, Registered_User et Program pour gérer automatiquement les id de ces tables. IdVideo sera unique parmi Video et Archives pour faciliter la remise en ligne des vidéos. Il faudra par contre gérer quelques cas où la séquence fait n'importe quoi.

Pour faciliter l'archivage j'ai mis des 'ON DELETE CASCADE' sur les tables référençant la table Video.

Pour faciliter la mise en ligne de nouvelles/anciennes vidéos j'ai défermé les contraintes sur les FOREIGN KEY des tables référençant la table Video et la table Archives. Les seules exceptions étant les tables qui ne peuvent pas avoir d'entrées avant que la vidéo soit en ligne (Liked_videos et Watch_history)

Toutes mes tables ont été traduites en anglais et la majorité de ce qui est fait aussi (sauf oublis) pour que cela soit plus simple à maintenir par une équipe internationale ce qui risque d'être le cas pour un site de replays TV.

Je précise que les timestamp pour les inserts de la table video ont été mis à jour et **fonctionneront jusqu'au 09/11 environ**.

5) Choix des vues:

J'ai choisi de modéliser les RECOMMENDATIONS par des vues parce que je ne les actualiserais qu'une seule fois par jour, et que ces vues seront appelées à chaque retour sur la page d'accueil du site (comme pour youtube). De plus la vue contient des informations redondantes qu'on peut obtenir par des jointures.

On a choisi de faire la vue FULL_TOTAL pour afficher pour chaque video le nombre total de vue et le nombre de vue sous 14 jours utilise pour les suggestions. Il s'agit basiquement d'une jointure de RECENT_WATCH_COUNT sur TOTAL_WATCH_COUNT qui ont leurs attributs respectifs.

6) Création des roles:

Les droits sont répartis sur 4 rôles :

- **Admin** : Nombre restreint de personnes qui a total accès à la base de données
- **Employe** : Rôle de base des employés qui permet de faire la majorité des opérations de tous les jours tant qu'elles ne sont pas sensibles (cad pas la table LOGINFO car un employe ne devrait pas avoir accès à ces informations à moins d'avoir les responsabilités et le contrat qui va avec)
- **Users** : Rôle pour tous les utilisateurs enregistrés de la plateforme. Ce rôle doit permettre de gérer son propre profil et d'accéder en select à la majorité des tables sauf les tables archives* et loginfo.
- **Public** : Rôle pour tous les utilisateurs non-enregistrés de la plateforme. Ce rôle doit permettre d'accéder en select à tout ce qu'un utilisateur enregistré à accès.

Je précise que 'CREATE ROLE xxxx' nous est interdit donc tout ce qui est au-dessus a été implémenté dans un fichier mais n'a pas pu être testé