

Rapport de Projet

Projet bases de données : site de préférences vidéos



SOMMAIRE

1. Introduction	3
-----------------	---

2. Modèle entité-associations	3
-------------------------------	---

3. Modèle relationnel	4
-----------------------	---

4. Contraintes d'intégrité	5
----------------------------	---

5. Conclusion personnelle	6
---------------------------	---

1. Introduction

Ce rapport a été réalisé dans le cadre de l'UE Bases de données et programmation web. Il s'agit de représenter un site web de replay vidéo à partir d'une base de donnée. Nous ne traiterons pour l'instant que la partie base de donnée, réalisée en SQL sur un serveur de gestion de bases de données ORACLE qui supporte SQL et PL/SQL. Pour plus d'informations sur le fonctionnement, se référer au sujet du projet.

Les requêtes SQL sont dans un fichier requetes.sql, il faut jouer avec les commentaires pour afficher les requêtes que l'on souhaite utiliser.

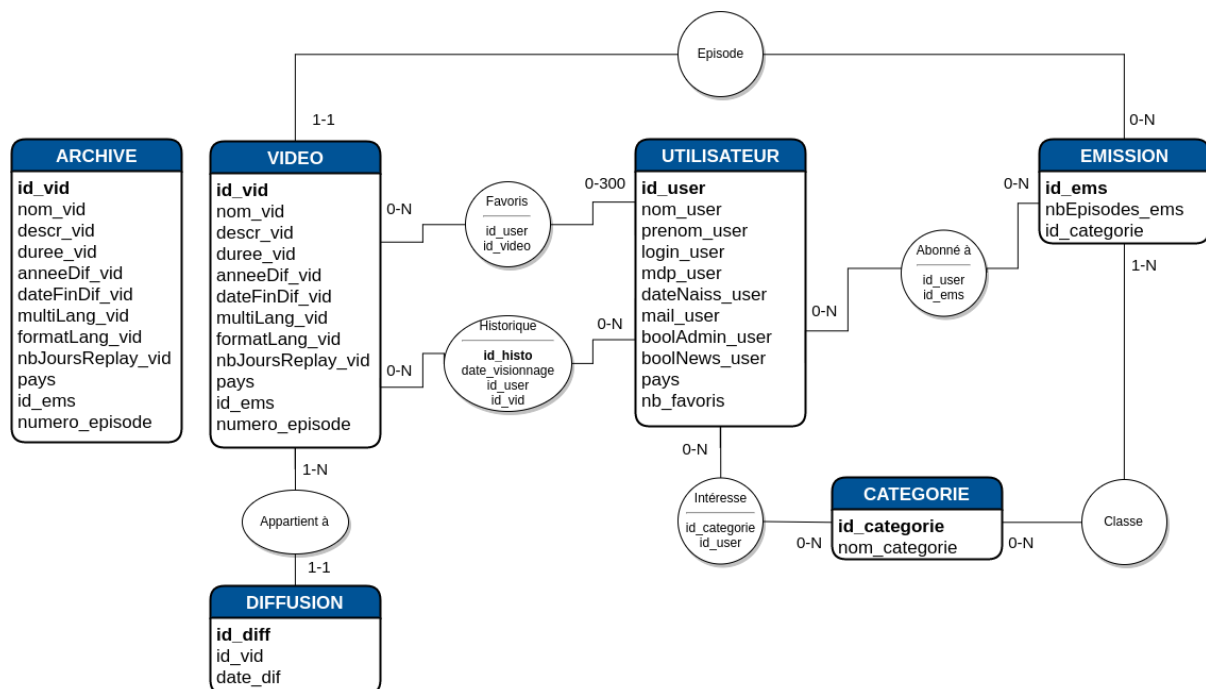
Les procédures et fonctions sont dans des fichiers séparés car plus imposantes que les requêtes : fonction1.sql, procedure2.sql, procedure3.sql et procedure4.sql.

Les triggers sont dans un fichier triggers.sql.

Pour créer la base en ayant les inserts et les triggers, il suffit de lancer @majtables.sql et la base se remplira automatiquement.

Le dossier schéma contient les PDF des deux schémas des deux sections suivantes.

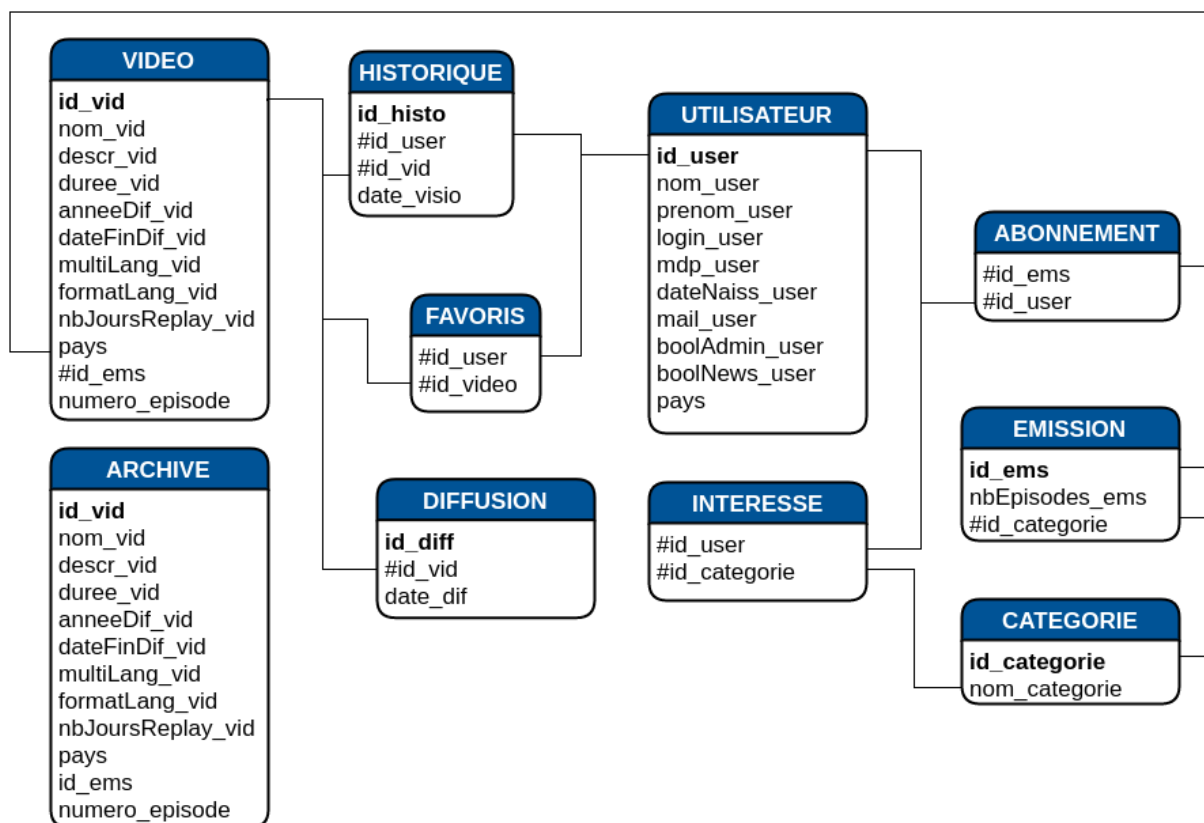
2. Modèle entité-associations



Ce schéma été réalisé sur draw.io en essayant de respecter au maximum les contraintes d'intégrités indiquées dans le sujet. Il est aussi disponible en format PDF. Les clefs primaires sont en gras.

La table archive n'a pas de lien avec les autres tables, excepté le fait qu'elle récupère les copies des champs de la table vidéo lors d'un archivage.

3. Modèle relationnel



Ce schéma a lui aussi été réalisé sur draw.io en essayant de respecter au maximum les contraintes d'intégrités indiquées dans le sujet. Il est aussi disponible en format PDF. Les clefs primaires sont en gras et les clefs étrangères sont précédées du caractère "#".

4. Contraintes d'intégrités

Le sujet indique quelques contraintes d'intégrité et on peut aussi en déduire certaines.

“Un utilisateur aura un maximum de 300 vidéos en favoris.” Cela impose un CHECK dans la création d'un utilisateur qu'on peut aussi faire sous forme de trigger par exemple TRG_FAVORIS.

A chaque insertion d'un épisode favoris associé à un utilisateur, le nombre de favoris est incrémenté avec le trigger TRG_FAVORIS.

Le sujet indique : “Si une diffusion d'une émission est ajoutée, les dates de disponibilités seront mises à jour. La nouvelle date de fin de disponibilité sera la date de la dernière diffusion plus 14 jours.” On suppose qu'il s'agit de l'ajout d'une diffusion d'une vidéo, c'est le trigger DATE_DISPO_DIFF.

Un CHECK dans la table video vérifie aussi que l'intervalle entre la première diffusion d'une vidéo et sa fin de disponibilité est bien supérieur à 7 jour.

La majorité des champs doivent être non nuls, c'est un choix arbitraire qui peut être modifié aisément si besoin.

“La suppression d'une vidéo entraînera son archivage dans une table des vidéos qui ne sont plus accessibles par le site de replay.” Il s'agit d'utiliser le trigger ARCHIVAGE qui copie chaque champ de la table video dans une table archivage solitaire.

Les émissions ne possédant qu'une seule vidéo et voyant leur vidéo supprimée se voient elles aussi supprimée à l'intérieur du trigger. Cette suppression est rendue possible par le champ nbEpisodes_ems indiquant le nombre d'épisodes d'une émission et qui est décrémenté à chaque suppression de vidéo dans ARCHIVAGE.

Inversement, le trigger AJOUT_VIDEO incrémente le nombre d'épisodes d'une vidéo.

Le trigger LIMIT_SPAM limite le nombre de visionnage par utilisateur à 3 vidéo pour chaque minute grâce aux champs de la date des vidéos correspondant à un utilisateur comparés à la date d'ajout. Il faut que tous les champs possibles de la date soient égaux sauf les secondes qui peuvent varier (ou être égales) afin de savoir quelles vidéos on été vues par cet utilisateur dans la minute.

Le dernier trigger DATE_VISIO_HISTO empêche d'insérer des visionnages dans le futur... Il compare la date de la vidéo à insérer avec sysdate.

5. Conclusion personnelle

Cette partie base de données était intéressante, j'ai beaucoup apprécié. C'était peu chronophage mais cela permet d'avoir une première approche très concrète d'un projet de construction de base de données. Il faudra encore modifier et adapter la base et les requêtes pour la partie web. A suivre !