

Melvin NAMONT

Pauline DECOOL

Bases de Données

Mini-projet:

création d'une médiathèque

INFO2

2018-2019

Sommaire

Présentation du projet

Structure de la base de données

Triggers

Procédures

Conclusion

Présentation du projet

Nous avons créé un entrepôt de données regroupant les ressources des médiathèques de différents établissements de la ville de Lannion.

Le but de cet entrepôt est de permettre à tous les utilisateurs d'avoir accès aux contenus des établissements qu'ils ne fréquentent habituellement pas.

Nous avons premièrement défini la structure de notre base de données à l'aide d'un schéma entité association. Nous avons ensuite produit le schéma relationnel associé à l'aide de l'outil **MySQL Workbench**, ce qui a permis à la structure de notre base de données d'être créée numériquement. A partir de ce résultat, nous avons pu créer dans les différentes classes les fonctions et triggers correspondant aux cas d'utilisations de la médiathèque.

Pour tester notre code, nous avons ajouté des données fictives à la base de données. Ainsi, nous avons pu constater visuellement si nos procédures fonctionnaient et retournaient bien le résultat attendu.

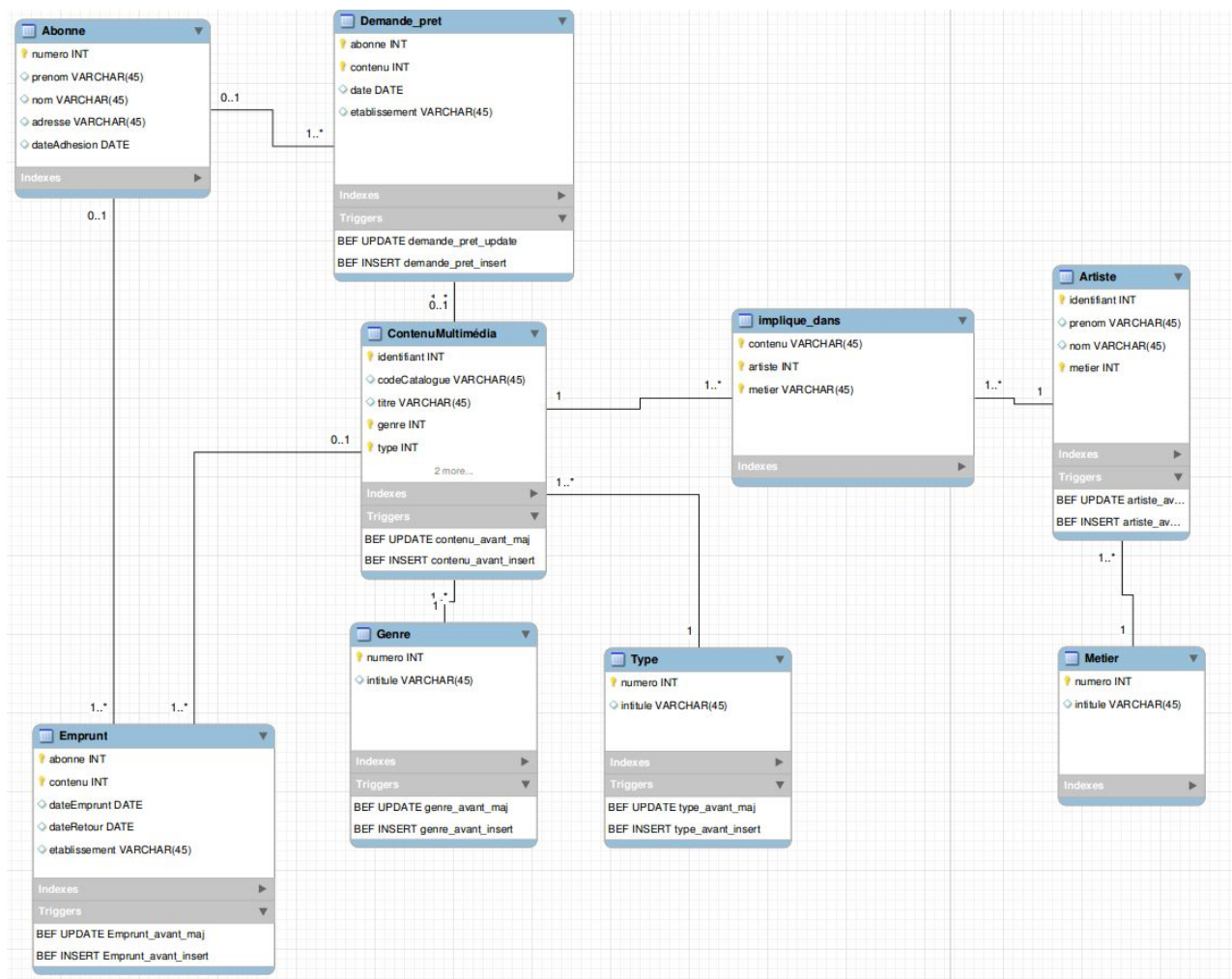
Structure de la base de données

Nous avons choisi de créer les classes **Metier**, **Genre**, et **Type** par souci de normalisation. Lorsqu'une personne voudra ajouter des données, elle devra choisir un numéro de métier, de genre ou de type et non taper l'intitulé au clavier. Cela permet d'éviter les dysfonctionnements dans les requêtes **SQL** au niveau des majuscules ou des accents notamment. De plus, choisir de créer des classes plutôt que des triggers nous permet de pouvoir ajouter de nouvelles possibilités à la liste.

La classe **Emprunt** permet de répertorier les emprunts des utilisateurs; ils sont associés à une date de début et une date de fin, ce qui nous permet de garder un historique de tous les emprunts. La date de fin sera initialisée à **NULL** et ne sera modifiée que lorsque le contenu emprunté sera rapporté à la médiathèque. Cette donnée nous permet donc également de contrôler si un abonné a un retard sur le retour des contenus qu'il a emprunté.

Le classe **Demande_pret** permet aux abonnés de saisir une demande pour un contenu déjà emprunté qui lui sera alors réservé pendant une semaine après son retour.

Un contenu est caractérisé par un numéro, qui correspond au code barre pour un contenu physique et au numéro de licence pour un contenu numérique. On peut différencier les contenus physique et numérique grâce à la colonne **etat**.



Triggers

Un trigger sera déclenché à chaque saisie d'emprunt et de demande de prêt pour vérifier que l'utilisateur a le droit de le faire . Si tous les critères ne sont pas respectés, l'emprunt ou la demande de prêt sera stoppé et un message indiquera le problème. De plus, un trigger empêchera la création d'un **Emprunt** si le contenu fait l'objet de demandes de prêt et que l'abonné qui essaye d'emprunter n'a pas lui-même effectué de demande.

Les retards de retour seront détectés par un trigger qui sera déclenché une fois par jour et qui comparera les dates de demandes de prêt avec la date actuelle et supprimera les demandes de prêt datant de plus de 7 jours.

Nous préférons ne pas supprimer les utilisateurs qui ne sont plus abonnés car cela simplifie le renouvellement d'abonnement et permet de garder une trace d'eux dans la base de données, notamment s'il n'a pas rendu tous ses contenus.

Un trigger sera déclenché à la création d'un nouveau **contenuMultimédia** et empêchera sa création si le champ **etat** n'est pas rempli par 'physique' ou 'numérique' ou que l'établissement ne fait pas partie des 4 établissements autorisés, encore une fois par souci de normalisation. Nous n'avons cependant pas créé de table car ces données sont peu susceptibles de changer.

La clé primaire d'un **Artiste** est (**numero,metier**), car un même artiste peut avoir plusieurs métiers. Nous voulons qu'un artiste soit relié à un numéro donc un trigger permettra d'éviter que deux artistes différents aient le même numéro ou qu'un artiste ait plusieurs numéros.

Dans l'ensemble, nos triggers veillent à ce que des erreurs soient évitées dans la base de données, notamment lors de l'ajout de nouvelles données.

La liste précises des triggers est donnée en annexe.

Procédures

Des procédures permettront aux utilisateurs d'accéder aux données, notamment pour obtenir diverses informations sur les **ContenuMultimédias**. Ils pourront également formuler des demandes de prêt et recevront un message d'erreur explicite si la création est impossible. De même, les abonnés pourront modifier des demandes de prêt, c'est-à-dire remplacer une demande par une autre afin de ne pas dépasser leur nombre limite de demandes.

Des procédures permettront aux administrateurs de répertorier les nouveaux emprunts et enverront un message d'erreur si l'emprunt n'est pas valide. Ils pourront également ajouter des nouveaux **Metiers**, **Types**, **Genres** et créer de nouveaux **Abonnes**.

Les administrateurs pourront de même avoir accès aux informations sur les abonnés, notamment sur les contenus qu'ils ont emprunté et les demandes de prêt qu'ils ont formulées. Des procédures de statistiques donneront des informations globales sur les emprunts et les demandes.

Les données pourront aussi être supprimées grâce à des procédures.

La liste précises des procédures est donnée en annexe.

Conclusion

Nous avons créé une base de données fonctionnelle dans laquelle vous pouvez ajouter des données si vous le souhaitez. Des triggers vous permettront d'éviter de commettre des erreurs et des procédures vous donneront un accès facilité aux données.

Nous n'avons pas trouvé comment insérer un trigger qui se déclenche tous les jours automatiquement, nous l'avons codé à la fin de la fiche Triggers.