

UTC 504 - Systèmes d'information et bases de données

Présentation des TP

André Miralles

Sommaire

TP 1. Évolution du schéma de la base de données « Bibliothèque » et introduction de l'Héritage entre tables.....	2
TP 2. Manipulation des données sur hiérarchie de tables.....	4

Nota général :

1. Comme vous aurez à faire plusieurs versions de la même base de données, certains noms de table et de contraintes peuvent être en conflit. Afin d'éviter ces derniers, il est suggéré de préfixer les noms de tables et de contraintes par le numéro de TP (Ex. : **TP01_NomBase**).
2. Dans les diagrammes, remplacer le type **STRING** par des **VARCHAR**.

Travaux pratiques

TP 1. Évolution du schéma de la base de données « Bibliothèque » et introduction de l'Héritage entre tables

Objectif :

Le but de ce TP est de faire évoluer le modèle de la base de données de la Figure 1 afin qu'elle soit plus générique. Pour guider la réflexion, deux évolutions du modèle sont proposées ci-dessous.

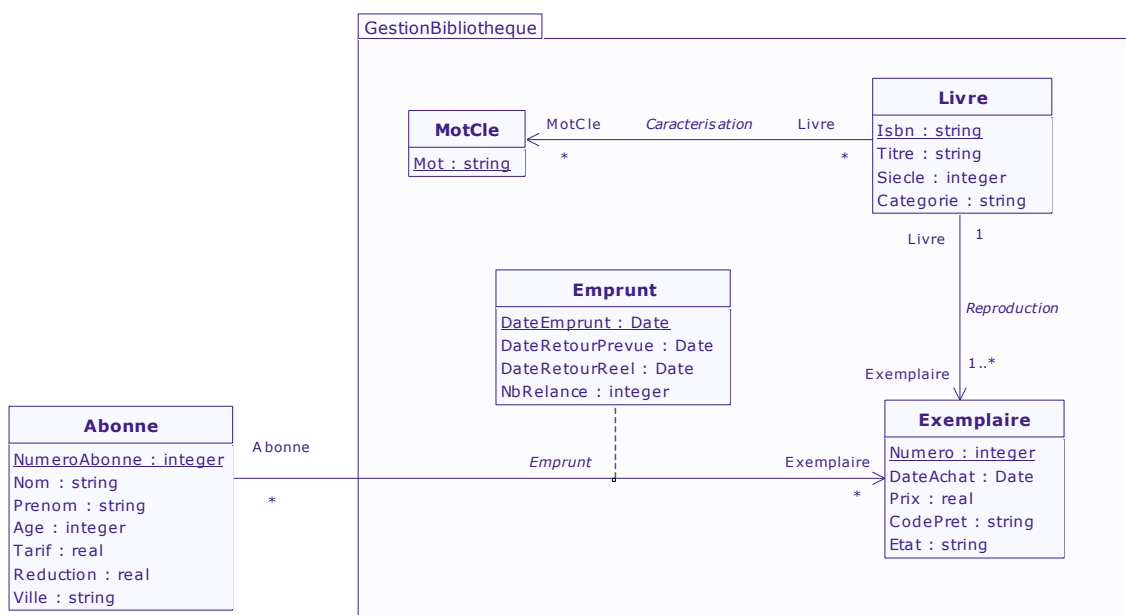


Figure 1. Modèle UML de la base de données Fg01_Bibliotheque

Nota 1 :

Les **attributs soulignés** sont les **clés primaires** des tables auxquelles ils sont attachés.

Nota 2 :

Afin de garantir la qualité des données, des contraintes de domaine ont été ajoutées à certains attributs :

Attributs	Contraintes
Age	0 < Age < 120
Siecle	0 < Siecle < 21
Etat	Etat = 'BON', 'ABIME' ou 'EN_REPARATION'
Nb_Relance	Nb_Relance = 1, 2 ou 3
Code_Pret	Code_Pret = 'EXCLU', 'EMPRUNTABLE' ou 'CONSULTABLE'

Présentation de la 1° évolution :

Cette première évolution porte sur des améliorations techniques et sur des incohérences sémantiques du modèle.

- Lorsqu'on analyse le modèle de la Figure 1, on constate que le nom de la **Ville** de résidence d'un **Abonné** apparaît sous forme d'attribut. Ce n'est pas cohérent, car le concept **Ville** est un objet en soi et indépendant de celui d'**Abonné**. Il faut donc en faire une entité afin d'améliorer la sémantique, la qualité et la stabilité de la base de données.
- Le concept d'**Abonné** introduit implicitement celui de **Bibliothèque** mais aussi celui d'**Abonnement**, qui est la relation (le contrat) liant un **Abonné** à une **Bibliothèque**. Ces deux concepts n'apparaissent pas dans le modèle du TP1. En poursuivant l'analyse, il est possible de constater que les attributs **Numéro Abonné**, **Tarif** et **Réduction** ne sont pas des propriétés propres à l'**Abonné** mais des propriétés du concept d'**Abonnement**, « contrat » liant un **Abonné** à une ou plusieurs **Bibliothèques**.
- Pour améliorer les performances, toutes les clés primaires de type *string* et *date* sont abandonnées et remplacées par des clés de type *integer*.
- Par ailleurs, le choix est fait de ne pas utiliser le **Numéro Abonné** et **Numéro** de la classe **Exemplaire** comme clé primaire.

A faire :

- Réaliser le modèle UML de cette 1° évolution.

Présentation de la 2° évolution :

Cette seconde évolution est effectuée pour répondre à de nouveaux besoins, prendre en compte de nouveaux cas d'utilisation et préparer la base pour d'autres évolutions. Ces nouvelles évolutions sont issues de l'analyse suivante :

- La notion de **Ville** conduit à introduire celle d'**Habitant** et plus généralement celle de **Personne Physique**.
- Souvent, les livres sont traduits dans différentes langues. Les utilisateurs de cette base souhaitent identifier l'original et les traductions associées ce qui induit aux deux évolutions :
 - Le numéro **ISBN** ne peut pas être utilisé comme identifiant du concept livre car chacune des éditions a son propre **ISBN**.
 - La langue du livre d'origine et des traductions étant différente, les utilisateurs souhaitent disposer de cette information. L'attribut **Langue** des classes **Livre Original** et **Livre Traduction** résulte de cette exigence. Une contrainte de domaine est ajoutée à cet attribut pour imposer que la valeur appartienne à la liste 'FR', 'BR' ou 'SP'.

A faire :

Pour réaliser cette évolution de la base Bibliothèque, il faut :

- Réaliser le modèle UML de cette 2° évolution.
- Créer la nouvelle base de données conforme à ce dernier modèle.
- Charger cette nouvelle base en utilisant les données et informations supplémentaires des scripts **3.2.2.1 - TP01 - Données Supplémentaires Hiérarchie Personne Physique.sql**, **3.2.2.2 - TP01 - Données Supplémentaires Hiérarchie**

Livre.sql et **3.2.2.3 - TP01 - Données Supplémentaires Table Bibliothèque.sql** et les données de la base Fg01 de la Figure 1. Il faut au préalable créer et charger la base Fg01 de la Figure 1 en exécutant les scripts **2.3 - Fg1 - Script de Creation Base de données** et **2.4 - Fg1 - Script de Chargement données**.

Nota 3 :

Il est préférable d'introduire les évolutions dans un script de création et un autre d'intégration de données plutôt que de le faire directement dans la base de données. Mettre en entête les instructions DROP et TRUNCATE permettent de relancer le script à la demande.

TP 2. Manipulation des données de hiérarchie de tables

Objectif :

Manipuler des requêtes sur des hiérarchies de tables de données.

A faire :

Faire les requêtes ci-dessous :

- Q1 Quels sont les nom et prénom des abonnés domiciliés à Montpellier ?
- Q2 Donnez la liste des exemplaires empruntés (leur numéro, code prêt et état) du livre de titre "LE MUR"
- Q3 Donnez la liste des exemplaires (leur numéro, code prêt et le titre du livre associé) d'un livre caractérisé par le mot clef "INFORMATIQUE"
- Q4 Quels sont les exemplaires (numéro) reproduisant le même livre que l'exemplaire de numéro 4112 et dont l'état est "BON" ?
- Q5 Quels sont les abonnés (numéro et nom) ayant emprunté un exemplaire du livre "LE MUR" ?
- Q6 Existe-t-il une catégorie pour laquelle aucun livre n'a été emprunté ?
- Q7 Par combien d'abonné le livre "LE MIRACLE DE LA ROSE" a-t-il été emprunté (tous exemplaires confondus) ?