

1 Problématique

Le professeur possède une collection importante de bandes dessinées, qu'il complète régulièrement. Une fois dans sa librairie préférée, il lui arrive de ne plus se souvenir où il en est exactement dans ses séries. De plus il prête régulièrement des livres à ses amis et aimerait pouvoir maintenir à jour l'état de ses étagères.



FIGURE 1 – Bibliothèque

Quelle solution peut-on mettre en place pour gérer efficacement la collection de bandes dessinées ?

2 Première approche

La première approche imaginée consiste en l'utilisation d'un tableau (figure 2) pour stocker les informations.

| | A | B | C | D | E | F | G | H | I |
|----|-----------------|------------------------|-----|------------|-----------------|------------------|------------------|-----------|---------------|
| 1 | Serie | Titre | Tom | ISBN | Genre | Scenariste | Dessinateur | Editeur | Date parution |
| 2 | Aya de Yopougon | Aya de Yopougon, Tom | 1 | 2070573117 | Tranche de vie | Abouet, Marg | Oubrierie, Clé | Gallimard | 2005-11-17 |
| 3 | Kookaburra | Planète Dakota | 1 | 2877645630 | Science-Fiction | Crisse, Didier | Crisse, Didier | Soleil | 1997-02-01 |
| 4 | Lanfeust de Tro | L'ivoire du Magohamoth | 1 | 2877642577 | Héroïc - Fant | Arleston, Chri | Tarquin, Didi | Soleil | 1994-09-01 |
| 5 | Baker Street | Sherlock Holmes n'a pe | 1 | 2840553511 | Humour | Veys, Pierre | Barral, Nicol | Delcourt | 1999-04-01 |
| 6 | Le chant des S | Ombres | 1 | 2840551721 | Aventures | Corbeyran, É | Guérineau, R | Delcourt | 1997-10-01 |
| 7 | Le Messenger | La sainte lance | 1 | 2912715555 | Polar-Thriller | Richez, Hervé | Mig | Bamboo | 2003-05-01 |
| 8 | Blacksad | Quelque part entre les | 1 | 2205049658 | Polar-Thriller | Diaz Canales, | Guarnido, Ju | Dargaud | 2000-11-10 |
| 9 | Le quatrième p | Le quatrième pouvoir | 1 | 2205038680 | Science-Fiction | Gimenez, Jua | Gimenez, Ju | Dargaud | 1989-05-01 |
| 10 | Captain Biceps | L'invincible | 1 | 2723445771 | Humour | Zep | Tébo | Glénat | 2004-04-01 |
| 11 | Caravane (Milh | Mila | 1 | 2756004073 | Aventure | Milhiet, Olivier | Milhiet, Olivier | Delcourt | 2008-04-02 |

FIGURE 2 – Utilisation d'un tableau

Dans ce tableau, l'ajout d'une colonne *emprunteur* permettrait de gérer les prêts.

Activité 1 : Établir les limites de cette approche.

- 2000 entrées = gérables par Python, mais si on veut étendre notre système à structure plus importante ?
- comment gérer modifications efficacement (ex : changement titre d'une série ?)
- pb tableau → article "malades du covid oubliés"
- Si on veut plus d'infos sur l'emprunteur (nom, date anniversaire), les colonnes ajoutées sont-elles pertinentes dans ce tableau ? Quand il rend un livre il faut vérifier/modifier beaucoup de lignes.
- Que se passe-t-il si on change le titre d'une BD ? L'emprunteur a-t-il vraiment pris ce livre ?

3 Modèle de données

3.1 Modèle relationnel

La première étape est de définir la manière dont les données vont être représentées. Le *modèle relationnel* est un des plus populaires. Dans ce modèle théorique :

- Une **entité** est un objet représenté par un n-uplet de valeurs scalaires. Une bande dessinée est une *entité* :

(Captain Biceps, L'invincible, 1, 2723445771, Humour, Zep, Tébo, Glénat, 2004-04-01)

- Une **relation** est l'ensemble des entités. On parle aussi de *table*. La collection de bandes dessinées est une *relation* :

(Captain Biceps, L'invincible, 1, 2723445771, Humour, Zep, Tébo, Glénat, 2004-04-01),
(Caravane, Mila, 1, 2756004073, Aventure, Milhiet Olivier, Milhiet Olivier, Delcourt, 2008-04-02),
(Kick-Ass, Le premier vrai super-héros, 1, 2809409994, Comics, Millar Mark, Romita Jr John, Panini Comics, 2010-03-17)

- Une *relation* possède des **attributs**. La relation des bandes dessinées possède les attributs :

(serie, titre, tome, isbn, genre, scenariste, dessinateur, editeur, date_parution)

- Chacun de ces attributs est défini dans un domaine.

| Bandes__dessinees | |
|-------------------|---------|
| serie | String |
| titre | String |
| tome | Integer |
| isbn | Integer |
| genre | String |
| scenariste | String |
| dessinateur | String |
| editeur | String |
| date_parution | Date |

Code 1 – **Schéma** de la relation bandes__dessinees

Une autre manière de présenter le schéma :

Bandes__dessinees(serie *String*, titre *String*, tome *Integer*, isbn *String*, genre *String*, scenariste *String*, dessinateur *String*, editeur *String*, date_parution *Date*)

scalaire = atomique ; définit par un type "de base" → carte d'identité d'une personne n'est pas une donnée scalaire.

Activité 2 :

1. Sur le même modèle définir la relation des Emprunteurs.
2. Les relations actuelles permettent-elles de gérer les emprunts ? Quelle solution mettre en place ?

table Emprunteurs : impossible de mettre les emprunts dedans → le mettre en évidence questionnement :

- si noms emprunteurs communs (dans Emprunteurs) ? → contrainte d'entité
- comment s'assurer qu'on n'a pas mis 2 fois le même emprunt dans la table (par erreur) ? → contrainte d'entité (isbn = clé primaire Emprunts)
- si on retire un emprunteur (n'est plus mon ami) ? il risque d'y avoir des bd non rendues → contrainte de référence

3.2 Contraintes d'intégrité

Une contrainte d'intégrité est une propriété vérifiée à tout instant et qui garantit *la cohérence des données*.

3.2.1 Contrainte de domaine

Chaque propriété que l'on souhaite renseigner est représentée par un *attribut* dans une relation. Le *domaine de définition* de chaque attribut doit garantir qu'il n'y aura pas de perte de données.

Activité 3 : Ajouter l'attribut *telephone* dans la relation *Emprunteurs*. Quel domaine de définition semble le plus approprié ?

telephone → Integer ? risque de perdre info (le 0 du début) et certains affichages commencent par +33 → String plus approprié
équivalent type de données en Python
autre exemple : une variable binaire (oui/non) → la représenter par un booléen garantit qu'il ne pourra pas y avoir de 3^e choix.

3.2.2 Contrainte d'entité

Elle garantit que chaque entité d'une relation est *unique* et l'identifie de manière non ambiguë. Lors de la modélisation, il faut s'assurer qu'un attribut (ou un ensemble d'attributs) permet d'identifier l'entité de manière unique.

Activité 4 : Dans la relation *Bandes_dessinees*, y-a-t-il un attribut qui garantit cette contrainte ?

dans une première approche isbn, même si : isbn-10 et isbn-13 et certains isbn ont des lettres.
~~De plus le csv fournit ne contient pas tous les isbn~~

À retenir

On appelle **clé primaire** l'attribut qui garantit l'unicité de l'entité.

On indique la *clé primaire* en la soulignant dans le schéma :

Bandes_dessinees(serie *String*, titre *String*, tome *Integer*, isbn *String*, genre *String*, scénariste *String*, dessinateur *String*, éditeur *String*, date_parution *Date*)

Activité 5 :

1. Dans la relation *Emprunteurs* l'attribut *nom* permet-il de garantir cette contrainte ?
2. Quelle solution peut-on mettre en place ?

ensemble nom-prenom pas suffisant non plus : il existe plusieurs Christophe Viroulaud

3.2.3 Contrainte de référence

Elle crée des associations entre deux relations. Elle garantit qu'une entité d'une relation B mentionne une entité existante dans une relation A.

Activité 6 :

1. Dans l'encadré ci-après, écrire à nouveau la relation *Emprunts* en utilisant les clés primaires des autres relations.
2. Définir une clé primaire pour cette relation.

Emprunts(

compléter le schéma dans l'encadré ; les clés étrangères en pointillés

Emprunts(isbn Integer, id_emprunteurs Integer)

Les attributs *isbn* et *id_emprunteurs* sont des **clés étrangères**. Cela signifie que la valeur de l'attribut *isbn* dans la relation *Emprunts* doit être une valeur existante dans la relation *Bandes_dessinees*. De même la valeur de l'attribut *id_emprunteurs* dans la relation *Emprunts* doit être une valeur existante de *id* dans la relation *Emprunteurs*.

il n'est pas obligatoire que la clé étrangère ait le même nom que la clé primaire correspondante
isbn est également clé primaire de la relation : on ne peut prêter 2 fois le même livre ; *id_emprunteurs* pas possible car une même personne peut prendre plusieurs bd

À retenir

Une **clé étrangère** est une référence à une clé primaire d'une autre relation. Elle garantit :

- que la valeur de l'attribut crée dans la relation existe dans la relation liée,
- qu'on ne peut supprimer une entité si elle est liée à une autre relation.

on ne peut supprimer un emprunteur s'il a encore des emprunts

3.2.4 Contrainte utilisateurs

ou contraintes métiers : spécifiques à la modélisation (exemple email ne contient que 1@, âge < 200)

4 Base de données

À retenir

L'ensemble des relations constitue une **base de données**.

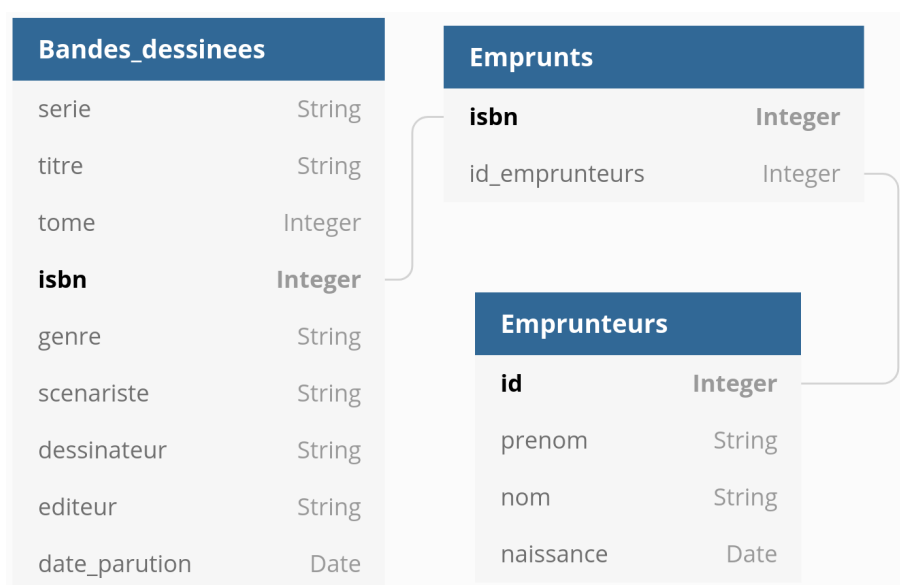


FIGURE 3 – Base de données

Les 2 attributs de Emprunts sont des clés étrangères

Activité 7 : Cette construction peut être améliorée. En effet, un même éditeur publie de nombreuses bandes dessinées.

1. Modifier la base de données pour prendre en compte cette observation.
2. D'autres attributs de la relation *Bandes_dessinees* répondent au même constat. Proposer une nouvelle version de la base de données.

id_dessinateur et id_scenariste sont **des clés étrangères** de l'attribut id de Auteurs (et pas l'inverse) : on peut créer un auteur qui n'a pas encore écrit de bd mais on ne peut pas créer une bd qui n'a pas d'auteur référencé.
Idem pour id_genre et id_editeur