II. Modélisation relationnelle des données

1) Modèle relationnel

2) Contraintes d'intégrité

a. Contrainte de domaine :

> Définition : Les contraintes de domaine restreignent les valeurs d'un attribut à celles d'un domaine donné et évitent que l'on puisse donner des valeurs illégales.

Exemple :
Dans notre exemple de la bibliothèque, en plus du prénom, du nom et d'un code, on peut souhaiter enregistrer l'adresse physique, l'adresse email et la date de naissance pour la relation **Usager**.
La contrainte de domaine sur l'adresse email sera "Chaine de caractères" : String
La date de naissance sera contrainte à être une "Date" : Date

Exemple :
La contrainte de domaine sur l'adresse physique est plus problématique.
Comment doit-on stocker l'adresse physique ? **Adresse + Ville + Code Postal** car deux villes peuvent porter le même nom (homonyme). Par contre, Ville + Code Postal sont uniques.
Quels sont les contraintes de domaine sur ces quatre attributs ? Ce sont toutes des **strings** car le code postal peut commencer par un 0 (comme l'Ain ;-))

b. Contrainte d'entité :

> Définition : Les contraintes d'entité garantissent l'unicité de chaque entité d'une relation. La **clé primaire**, garante de ces contraintes d'entité, est un ensemble d'attributs identifiant chaque entité de la relation de manière unique.

Exemple :
Dans notre bibliothèque, les usagers ont un nom, un prénom et un code barre.

- Le nom (ou le prénom) ne peut pas être une clé primaire car deux personnes d'une même ville peut avoir le même nom (ou pire le même prénom!) ;
- Pourquoi le nom et le prénom ensemble ne peuvent pas constituer une clé primaire ?

Car s'il existe deux personnes nommées Jean Dupont, la clé n'est pas unique.

- ❖ Expliquez pourquoi seul un code barre unique peut constituer une bonne clé primaire et garantir les contraintes d'entité.

Clairement, le nom et le prénom ne vont pas marcher. Nom+Prénom ne fonctionnent pas non plus. Selon l'énoncé, le code barre est généré de manière unique. Cela semble être une bonne clé primaire.

- ❖ Expliquez pourquoi nous ne pouvons pas nous passer du code barre unique, même si nous rajoutons des informations sur l'usager comme dans l'exemple du paragraphe a). Dans chacun des cas, expliquez la limitation liée aux contraintes d'entité.

Nous pouvons rajouter l'adresse, le CP, ou la ville mais dans ce cas on pourrait se retrouver dans la situation où deux Jean Dupont, habitent tous les deux au 7 Rue des plantes, 75012, Paris. Cela arrive !

Notation : La clé primaire est soulignée d'un trait plein. Par exemple :
Usager(*nom* **`String`**, *prénom* **`String`**, *adresse* **`String`**, *cp* **`String`**, *ville* **`String`**,
email **`String`**, *code_barre* **`String`**)

Exercice :

a) Quelle est la clé primaire du schéma **Livre** ?

L'ISBN est un code unique créé pour chaque version d'un livre. Cela semble être une excellente clé primaire.

Livre(*titre* **`String`**, *auteur* **`String`**, *éditeur* **`String`**, *année* **`Int`**, *isbn* **`String`**)

b) Quelle pourrait être la clé primaire du schéma **Emprunt** défini précédemment ? On se réfèrera à des situations concrètes pour trouver la clé primaire de ce schéma.

On rappelle : Emprunt(*titre* **`String`**, *usager* **`String`**, *date_emprunt* **`Date`**)
Le titre peut-il être suffisant ? Non, car deux livres peuvent avoir le même titre.
L'usager peut-il être suffisant ? Non, car Jean Dupont…
La date ? Non, toujours pas, deux livres peuvent être empruntés en même temps.
Titre + Usager ? Non car un des deux Jean Dupont peut emprunter un des deux livres intitulés "Possession".
Rien de fonctionne dans ce cas. La relation a en fait été mal définie dès le début…

Nous avons vu précédemment que les clés primaires de Livre et de Usager sont l'isbn et le code_barre.
Ainsi, au lieu de définir le livre par son **titre**, définissons le avec son **isbn**.
Pour l'usager, au lieu de le définir par son **nom**, utilisons son **code_barre**.

La relation est réécrite : Emprunt(*isbn* **String**, *code_barre* **String**, *date_emprunt* **Date**)
Le code_barre est-il une bonne clé primaire ? Non car un usager peut emprunter plusieurs livres.
L'isbn est-il une bonne clé primaire ? Oui ! Car il n'existe qu'un seul livre avec un isbn donné.

c. Contrainte de référence :

Définition : Les contraintes de référence assurent la cohérence du lien entre deux relations A et B. La **clé étrangère**, garante de ces contraintes de référence, est un ensemble d'attributs d'une table A qui sont aussi une **clé primaire** d'une autre table B.

Exemple :
Comme vous l'avez sans doute compris dans l'exercice précédent, notre première approche de la relation **Emprunt** était très naïve. Il n'existe pas de bonne clé primaire. On va donc redéfinir cette relation en proposant le schéma suivant :

Emprunt(*code_barre* **String**, *isbn* **String**, *retour* **Date**)

code_barre et *isbn* sont définies comme des clés étrangères : *code_barre* doit être une **valeur valide** de la relation Usager.
Cela garantit deux propriétés :
a) la relation Emprunt ne mentionne que des usagers présents dans la base de données.
b) on ne peut pas supprimer une entité de la relation Usager si celle-ci n'a pas rendu tous ses livres.

L'usager est défini comme une clé secondaire et non primaire. Un usager va donc pouvoir emprunter une liste de plusieurs livres représentés par leurs isbn.

Exercice 1 :
a) Expliquez pourquoi l'isbn est défini comme la clé primaire de la relation Emprunt.

voir exemple précédent

b) Avec vos propres mots, expliquez précisément la propriété b) de l'exemple ci-dessus.

L'isbn constitue une clé primaire, garante de la contrainte d'entité. En effet, l'isbn est unique : toutefois, attention… Cela ne sera pas suffisant pour gérer la bibliothèque car cela ne prend pas en compte l'existence de plusieurs copies du même livre (donc avec le même isbn).

Exercice 2 :
La relation Livre écrite comme Livre(*titre* **String**, *auteur* **String**, *éditeur* **String**, *année* **Int**, *isbn* **String**) est très mal définie.

En effet, si un livre a plusieurs auteurs, on peut se retrouver avec une liste d'auteurs dans une seule chaine de caractères. Par exemple, deux livres d'Axel Kahn :

Livre = {

```
('Copies conformes, le clonage en question', 'A. Kahn et F. Papillon', 'Nil', 1998, '2702816738'),
('Entre deux mers', 'A. Kahn', 'Stock', 2015, ' 9782234079168'),
}
```

On va donc créer deux nouvelles relations **Auteur** et **Auteur_de**. La relation **Auteur_de** relie un auteur au livre dont celui-ci est un des auteurs.

❖ Donnez le schéma de la relation **Auteur**. On soulignera en trait plein la clé primaire.

Auteur(*auteur_id* **Int**, *nom* **String**, *prénom* **String**)

Deux auteurs peuvent avoir le même nom et le même prénom, donc ce ne sont pas de bonnes clés primaires. Par contre l'id, géré automatiquement, sera une bonne solution.

❖ Expliquez pourquoi la relation **Auteur_de** a pour schéma :

Auteur_de(*auteur_id* **Int**, *isbn* **String**)

Les attributs *auteur_id* et *isbn* sont des clés étrangères faisant référence aux relations Auteur et Livre respectivement.

Auteur_id peut-il être une clé primaire ? Non car un auteur peut avoir écrit plus d'un livre.

isbn peut-il être une clé primaire ? Non car un livre peut avoir été écrit par plusieurs auteur.

Il faut donc prendre les deux attributs comme clé primaire.

d. Contraintes utilisateurs (ou métier) :

> Définition : Les contraintes utilisateurs sont toutes les contraintes que l'on ne peut exprimer par les trois autres contraintes.

Exercice :

Quelle serait la contrainte utilisateur associée à l'âge d'un usager ? L'age doit être compris entre 0 et 120 ans.

Quelle serait la contrainte utilisateur associée à un code postal ? Le code postal doit avoir une longueur de 5 caractères.