

Bases de Données Avancées, Partie 1 : Le modèle relationnel

Thomas Gerald

September 8, 2025

Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS

`thomas.gerald@lisn.upsaclay.fr`

Introduction

Le modèle relationnel : c'est quoi ?

SGBD avant 1970

- Modèle hiérarchique
- Modèle réseaux

Le modèle relationnel:

Proposition d'organisation des données par Codd en 1970

- Une base de données est une collection de **Relations**
- Chaque relation est représentée un tableau avec des colonnes et des lignes

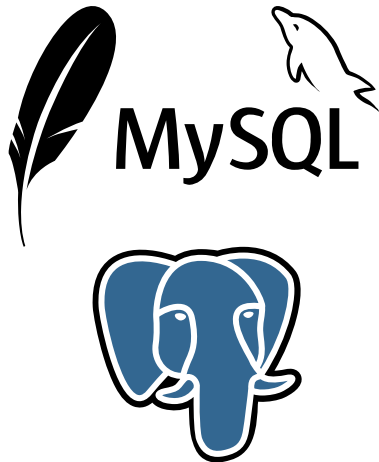
De quel niveau ? **Niveau Logique** (ce n'est pas la structure interne/physique des données)

Le modèle relationnel : lesquelles ?

Le modèle le plus utilisé aujourd'hui

SGBD Post 70

- Oracle
- Microsoft's Access
- MySQL (aussi oracle)
- PostgreSQL
- Et beaucoup D'autres....



Une relation

Une relation est une table (en deux dimensions) définie par un schéma et son instance.

Le schéma et l'instance d'une relation

- **Le schéma de la relation** : la description des colonnes (les noms des colonnes, les types de chaque colonne)
- **Instance d'une relation** : la table contenant les données

Le schéma

Chaque table est décrite par un **schéma**, c'est à dire la définition des colonnes, chaque colonne (ou champ) est associé un "type":

La colonne n'accepte un type unique !!!

- Une chaîne de caractères
- Un entier
- Un flottant
- etc...

Par exemple :

Vendeurs(*vid*: **integer**, *vnom*: **string**,
note: **float**)

Une instance:

Une instance d'une relation correspond aux "données" de la relation, c'est l'ensemble des lignes de la table.

Vendeurs		
vid	vnom	note
5	Jean	12
8	Amine	15
7	Hugo	12
⋮	⋮	⋮

Les données

Un ligne de données s'appelle :

- un n-uplet
- un **enregistrement**
- un tuple (en anglais)

Le schéma de la base de données

- Vendeurs (vid: **integer**, vnom: **string**, note: **float**, mail: **string**)
- Objets (oid: **integer**, onom: **string**, prix: **float**, vid: **integer**)

Une instance de la base de données

S'assurer de la cohérence

On veut pouvoir ajouter des informations :

- On veut que la base de donnée soit cohérente
- On veut que les types soient respectés
- On veut contraindre certaines valeurs par rapport à l'état de la BD (unicité,...)

On peut les définir au moment de la définition du schéma (type, unicité,...)

Considérons la relation

Vendeurs (vid: **integer**, vnom: **string**, note: **float**, mail: **string**)

→ Si *vid* a une contrainte d'unicité alors il s'agit d'une contrainte de type clef.

Définition :

1. Deux tuples distincts dans une instance de relation ne peuvent avoir les mêmes valeurs dans tous leurs champs de la clef
2. Il n'existe pas un sous ensemble de champs de la clef étant un identifieur unique

Par ailleurs il existe toujours au moins une clef dans une relation (potentiellement tous les champs)

Un clef primaire est une clef, elle est unique sur la table elle peut-être :

- Naturelle : Utilisant des informations de la table (plusieurs colonnes)
- Artificielle : une clef dédié à la table sans information

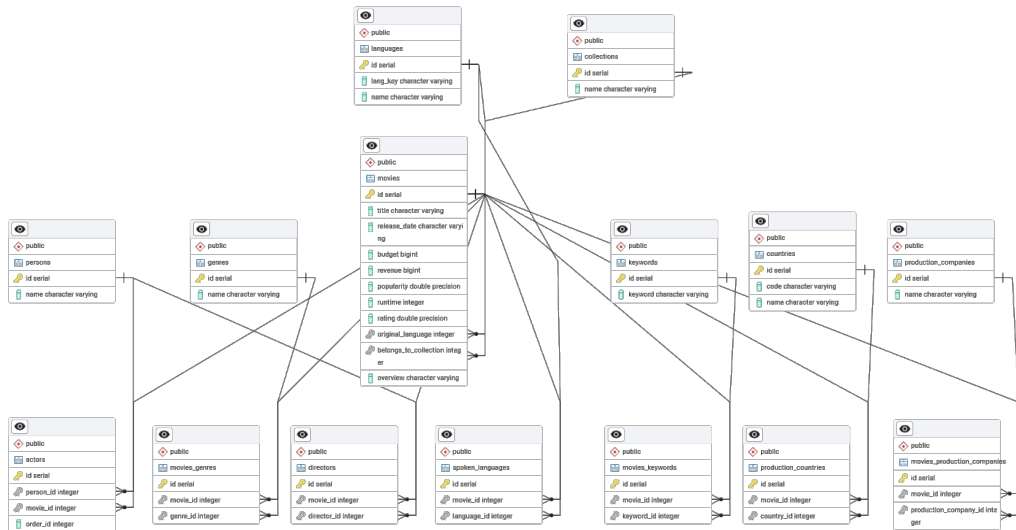
On évite aussi que celle-ci soit mise à jour !!!

Le Modèle relationnel : Les clefs étrangères

Les clefs étrangères sont des références sur des clefs d'une autre relation ( pas forcément une clef primaire). Dans l'exemple :

- Vendeurs (vid: **integer**, vnom: **string**, note: **float**, mail: **string**)
- Objets (oid: **integer**, onom: **string**, prix: **float**, vid: **integer**)

La colonne vid d' Objet pourrait être un clef étrangère sur la clef vid de Vendeurs.



Soit

- Vendeurs (vid: **integer**, vnom: **string**, note: **float**, mail: **string**)
- Objets (oid: **integer**, onom: **string**, prix: **float**, vid: **integer**)

On peut définir la relation Vendeurs avec SQL comme il suit :

```
CREATE TABLE Vendeurs( -- nom de la table
    vid INTEGER, -- un entier
    vnom VARCHAR(32), -- une chaîne de caractère
    note NUMBER, -- Un nombre
    mail VARCHAR(32), -- une chaîne de caractère
    CONSTRAINT vkey PRIMARY KEY(vid) -- une contrainte de clef primaire
)
```

Le Modèle relationnel : Définition SQL

Soit

- Vendeurs (vid: **integer**, vnom: **string**, note: **float**, mail: **string**)
- Objets (oid: **integer**, onom: **string**, prix: **float**, vid: **integer**)

On peut définir la relation Vendeurs avec SQL comme il suit :

```
CREATE TABLE Objet( -- nom de la table
    oid INTEGER, -- un entier
    onom VARCHAR(32), -- une chaîne de caractère
    prix NUMBER, -- Un nombre
    vif INTEGER, -- une chaîne de caractère
    CONSTRAINT okey FOREIGN KEY(oid), -- une contrainte de clef primaire
    CONSTRAINT o_vkey FOREIGN KEY(vid) REFERENCES Vendeurs(vid) -- une
)
```

Il est possible d'ajouter d'autre contrainte comme :

- Autoriser une plage de valeurs
- Une contrainte d'unicité
- Interdire les valeurs nulles
- Des contraintes d'exclusion (contraintes sur différentes colonnes)

Modélisation d'une base de données

On supposera dans la suite de cette section la création d'une application de gestion de bibliothèque où :

- Chaque livre à au moins un auteur qui lui est associé
- Un livre peut exister en plusieurs exemplaires
- Des client ont la possibilité d'emprunter des livres
- Le bibliothécaire peut ajouter/supprimer des livres
- Le bibliothécaire peut consulter les livres empruntés par un client
- Le bibliothécaire et le client peuvent consulter les livres et savoir combien sont disponibles
- Le bibliothécaire peut ajouter/supprimer un emprunt avec la date de rendu
- ...

Description d'une application:

- le nom d'une classe (classe)
- Les attributs (colonnes) des tables/classes
- Les opérations sur la table/classe
- Les relations entre les tables

 **Eviter la redondance des fonctionnalités**

Exemple:

- La classe Livre a des attributs :
 - son titre
 - sa date de publication
 - ...
- Opérations de recherche d'un livre
 - rechercher par date de parution
- Un livre a au moins un auteur...
 - On peut modéliser

Une classe

1. Le nom de la classe
2. Les attributs (colonnes) des tables/classes
3. Les opérations sur la table/classe

Nom de la classe	①
- attribut : type	②
+ operation(paramètre): type	③

Visibilité d'un attribut ou d'une opération

- - l'attribut/opération n'est visible que par la classe (attribut privée)
- + l'attribut/opération n'est visible que par la classe (attribut public)

En général les attributs sont privées et les opérations publics...

Modéliser une base de données : La classe livre (exemple)

Attributs ?

- Le titre du livre (str)
- Le numéro ISBN (int)
- La date de parution (date)
- Le numéro d'exemplaire
 - ❌ → Utilisation d'une classe exemplaire
- La liste des auteurs
 - ❌ → Utilisation d'une classe auteur

Livre
- titre : str - isbn : int - parution : date

Modéliser une base de données : La classe livre (exemple)

Les opérations

- Rechercher un livre selon un intervalle de date
- Ajouter un livre
- ...

Livre
- isbn : int - titre : str - parution : date
+ rechercherDate(début : date, fin : date): Livre + ajouter(isbn : int , titre : str, parution : date) : void + ...

Modéliser une base de données : La classe livre (exemple)

Unicité des enregistrements...

On peut souligner le(s) attribut(s)
représentant la clef primaire...

Quelle est la clef primaire ?

Livre
<ul style="list-style-type: none">- <u>lid</u> : int- isbn : int- titre : str- parution : date
<ul style="list-style-type: none">+ rechercherDate(début : date, fin : date): Livre+ ajouter(isbn : int , titre : str, parution : date) : void+ ...



Livre
<ul style="list-style-type: none">- <u>isbn</u> : int- titre : str- parution : date
<ul style="list-style-type: none">+ rechercherDate(début : date, fin : date): Livre+ ajouter(isbn : int , titre : str, parution : date) : void+ ...



→ Eviter la redondance

Modéliser une base de données : La classe auteur (exemple)

Attributs

- identifiant d'un auteur
- prénom
- nom

Les opérations

- Retrouver la liste des livres d'un auteur
- Ajouter un auteur
- ...

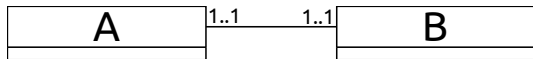
Auteur
- <u>auteur_id</u> : int - prenom : str - nom : str
+ rechercheLivre(Auteur) + ...

Relations

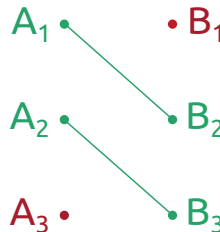
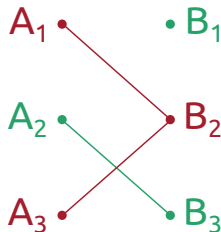
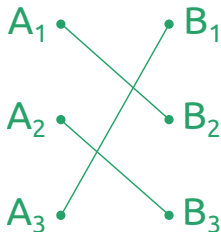
- Un objet d'une classe peut être associé à un autre → Un exemplaire et son numéro d'exemplaire
- Un objet peut être associé à plusieurs autre objet → Un livre a plusieurs auteurs...

Modéliser une base de données : Association (cardinalité/arité)

Relation 1 vers 1 :

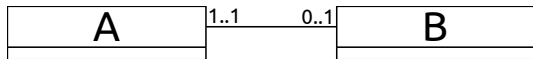


- Tout élément de A est associé à un élément de B
- Tout élément de B est associé à un élément de A

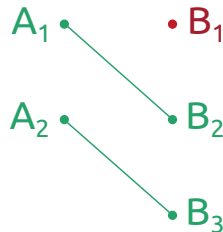
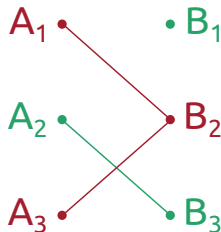
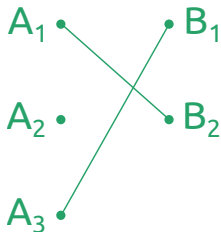


Modéliser une base de données : Association (cardinalité/arité)

Relation 1 vers 0 ou 1:

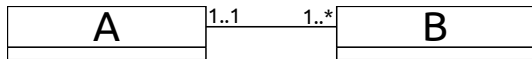


- Tout élément de A est associé au plus à un élément de B
- Tout élément de B est associé à un élément de A

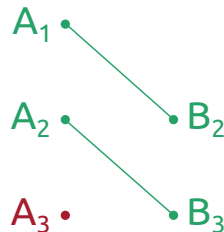
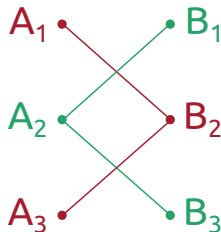
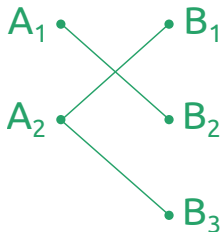


Modéliser une base de données : Association (cardinalité/arité)

Relation 1 vers 0 ou 1:

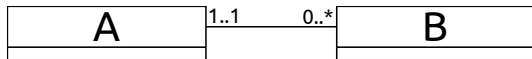


- Tout élément de A est associé à au moins un élément de B
- Tout élément de B est associé à un élément de A

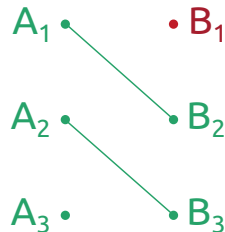
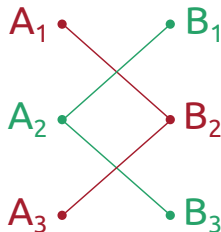
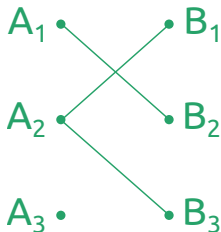


Modéliser une base de données : Association (cardinalité/arité)

Relation 1 vers 1 :



- Tout élément de A peut être associé à plusieurs éléments de B
- Tout élément de B est associé à un élément de A



Modéliser une base de données : Association - exemple 2

Auteur et Livre (écriture) :



Modéliser une base de données : Association - exemple 2

Auteur et Livre (écriture) :



- Combien de livres peut avoir écrit un auteur ?
- Un livre peut avoir été écrit par combien d'auteurs ?

Modéliser une base de données : Association - exemple 2

Auteur et Livre (écriture) :



- Combien de livres peut avoir écrit un auteur ? **1 ou plusieurs**
- Un livre peut avoir été écrit par combien d'auteurs ? **1 ou plusieurs**

Auteur et Livre (écriture) :

Et si un livre ne peut avoir été écrit que par un auteur ?

Modéliser une base de données : Association - exemple 3

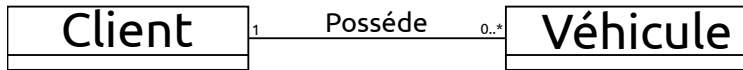
Auteur et Livre (écriture) :

Et si un livre ne peut avoir été écrit que par un auteur ?



Description d'une association :

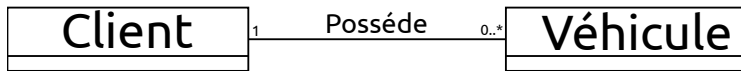
On peut vouloir décrire l'association , par exemple avec une action: contient, emprunte, etc... En considérant les deux classes suivante pour un système de gestion de véhicule on peut définir :



Càd, qu'un client peut posséder 0 ou plusieurs véhicules.

Description d'une association :

On peut vouloir décrire l'association , par exemple avec une action: contient, emprunte, etc... En considérant les deux classes suivante pour un système de gestion de véhicule on peut définir :

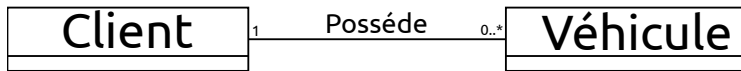


Càd, qu'un client peut posséder 0 ou plusieurs véhicules.

Est-il possible qu'un véhicule n'ai pas de propriétaire ?

Description d'une association :

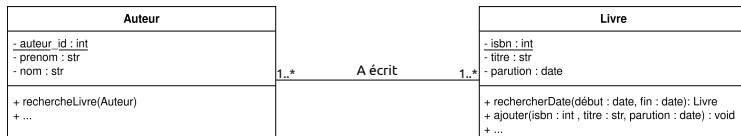
On peut vouloir décrire l'association , par exemple avec une action: contient, emprunte, etc... En considérant les deux classes suivante pour un système de gestion de véhicule on peut définir :



Càd, qu'un client peut posséder 0 ou plusieurs véhicules.

Est-il possible qu'un véhicule n'ai pas de propriétaire ? → **Non**

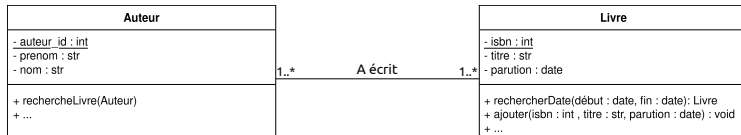
Modéliser une base de données : Table d'association



Être Auteur sera modéliser par une table d'association :

- Contenant une référence sur l'auteur
- Contenant une référence sur le livre

Modéliser une base de données : Table d'association



Être Auteur sera modélisé par une table d'association :

- Contenant une référence sur l'auteur (clef étrangère sur auteur)
- Contenant une référence sur le livre (clef étrangère sur livre)
- Le couple correspond à la clef primaire de la table d'association
- Le couple est unique !!!

Ce que nous avons appris

- Ce qu'est le modèle relationnelle
- Différence entre schéma et instance d'une base de données

Ce que vous devez savoir faire/expliquer

- Définir les différents concepts du modèle relationnel
- Déterminer les contraintes, les types d'une relation (à partir d'un cahier des charges par exemple)

→ Comment manipuler les données d'un SGBD relationnel ?