

Modélisation

Table des matières

1.	Pourquoi modéliser ?	2
2.	Un point vocabulaire	2
3.	À quoi ça ressemble ?	3
4.	Comment traduire ça en base de données ?	4
	Exercices :	6

1. Pourquoi modéliser ?

Les bases de données peuvent très vite atteindre un degré de complexité difficile à transmettre avec des mots à un collègue développeur. Pour éliminer le risque d'oubli d'informations cruciales, et pour simplifier leur transmission, deux méthodologies ont été mises au point :

- La modélisation UML, plutôt orientée objet, comprenant de nombreux diagrammes de classes et autres joyeusetés
- La méthode Merise, qui comprend trois diagrammes principaux qui se réalisent dans un ordre logique :
 - Le **MCD** ou modèle **conceptuel** de données, qui permet une représentation détachée de la logique informatique. Il est celui qui est le plus compréhensible, même pour des non-développeurs.
 - Le **MLD**, ou modèle **logique** de données, découle du MCD. Il est une représentation de la logique du MCD : les **entités** deviennent des **tables**, les **associations** deviennent des clés étrangères ou des tables... Il permet de démontrer ce dont aura besoin le développeur pour intégrer la base.
 - Le **MPD**, ou modèle **physique** de données, est la représentation la plus aboutie de la base, elle précède l'intégration par le développeur. Toutes les clés étrangères sont notées au bon endroit, les associations n'existent plus, seulement les relations et les **cardinalités**

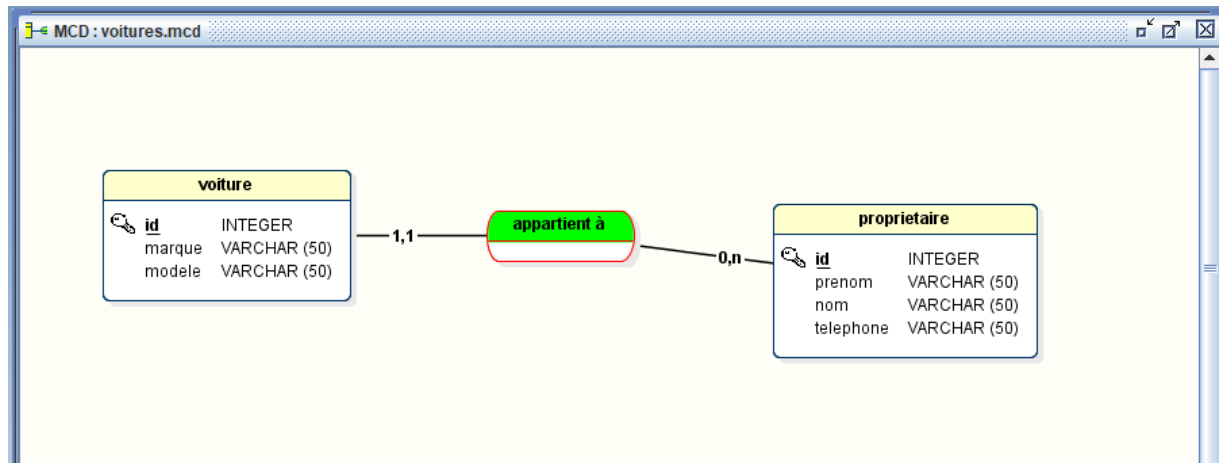
2. Un point vocabulaire

Selon l'étape de l'analyse on utilise un vocabulaire particulier.

MCD	SGBD
Entité	Table
Association	Liaison (peut être une table ou une clé étrangère)
Identifiant	Clé primaire
	Clé étrangère
Attribut, propriété	Champs, colonne
Occurrence, instance	Occurrence, enregistrement

3. À quoi ça ressemble ?

Voici un MCD reprenant les informations des voitures que nous avons vues dans le premier cours sur la BDD :

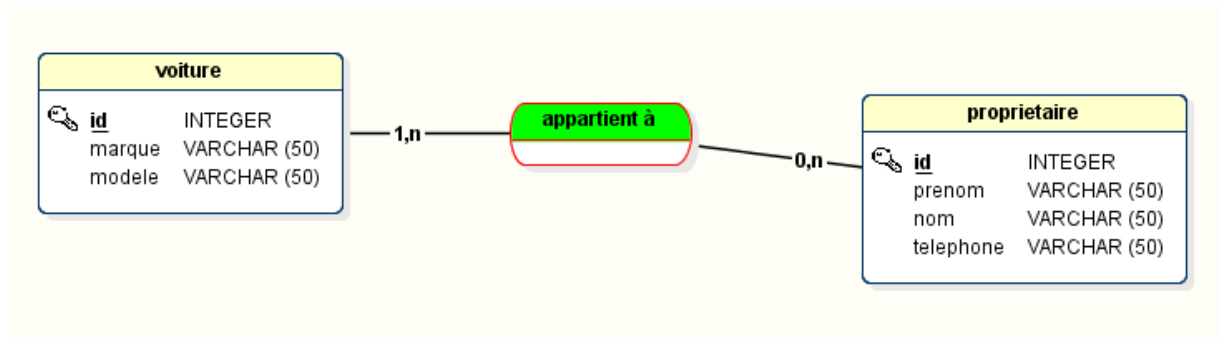


On peut y voir beaucoup de choses très intéressantes :

Entité		Ceci s'appelle une entité . Elle est la représentation de la structure de la table dans laquelle on va stocker les informations nécessaires pour gérer les voitures dans notre base.
Clé primaire		On peut voir que cette entité en question a un de ses attributs en gras, souligné, marqué d'une icône de clé : c'est l' identifiant de cette entité, la clé primaire donc.
Association		Cette association, le plus souvent notée avec un verbe , décrit le lien qui unit deux entités. Elle peut être porteuse de données ou non, comme ici.
Cardinalités		Elles complètent les descriptions de l' association en indiquant le nombre de liens possibles entre une occurrence d'une entité et les autres. Elles comportent deux nombres, un minimum et un maximum. Ici on peut donc lire qu'une voiture n'a qu' UN propriétaire, et qu'un propriétaire peut avoir ZÉRO ou PLUSIEURS voitures

Il est intéressant de noter que les **clés étrangères** ne sont pas représentées ici. C'est à moitié vrai car même si elles ne sont pas nommées, elles sont en réalité représentées par les **cardinalités**. Une cardinalité **1,1** de la table voiture vers la table propriétaire indique que l'on aura une clé étrangère **id_proprietaire** dans la table voiture.

Et du coup, que se passe-t-il si on a une cardinalité **1,n** au niveau de la voiture, comme ceci ?



Alors dans ce cas l'association veut dire autre chose, à savoir qu'une voiture peut avoir **UN** ou **PLUSIEURS** propriétaires, et qu'un propriétaire peut avoir **ZÉRO** ou **PLUSIEURS** voitures.

4. Comment traduire ça en base de données ?

Chaque entité va devenir une **table** en base de données, chaque **attribut** va devenir un champ. Sous cette apparente simplicité il y a quelques règles cachées en plus que nous allons voir ici, notamment par rapport aux associations.

Nous l'avons brièvement vu plus haut, mais lorsque qu'une association porte les cardinalités **0,1** ou **1,1**, alors un champ **clé étrangère** sera créé dans la table qui se trouve du côté où est la cardinalité en question. Mais alors que se passe-t-il si jamais on a deux cardinalités dites « n-max » (C'est-à-dire avec n en valeur maximale) ?

Alors dans ce cas cela se matérialisera par ce que l'on appelle une **table de liaison**, qui portera les deux clés étrangères des tables de l'associations.

Exemple avec la table de liaison **voiture_proprietaire** :

id_voiture	id_proprietaire
2	1
1	2
3	2
4	1
4	3

Seule cette structure de table nous permet de gérer efficacement le fait qu'un propriétaire puisse avoir plusieurs voitures **ET** qu'une voiture puisse avoir plusieurs propriétaires. Cependant ici je n'ai pas mis de clé primaire... Car la clé primaire va être en réalité sur les **DEUX CHAMPS** à la fois. C'est l'association des deux clés étrangères qui va permettre d'identifier la ligne.

Comme je l'évoquais plus haut, une table de liaison peut également être porteuse de données, comme par exemple la date d'inscription sur la carte grise du propriétaire. Cette donnée doit respecter les trois formes normales bien sûr, mais c'est le cas ici puisque :

- La donnée sera atomique, il s'agira simplement d'une date
- Les données dépendront de la clé primaire, la date d'inscription du propriétaire sur la carte grise dépendra bien **ET** de la voiture **ET** du propriétaire
- Aucune donnée ne dépendra d'autre chose que de la clé primaire (qui est en fait deux champs, pour rappel)

On aurait du coup cette structure :

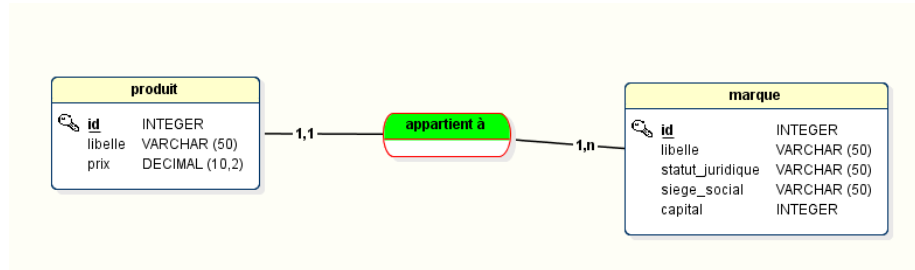
id_voiture	id_proprietaire	date_entree_certificat
2	1	2025-06-30
1	2	2014-05-14
3	2	2001-05-22
4	1	1994-04-20
4	3	2003-09-08

Dans ce cas, on qualifie le champ « **date_entree_certificat** » de **donnée portée**.

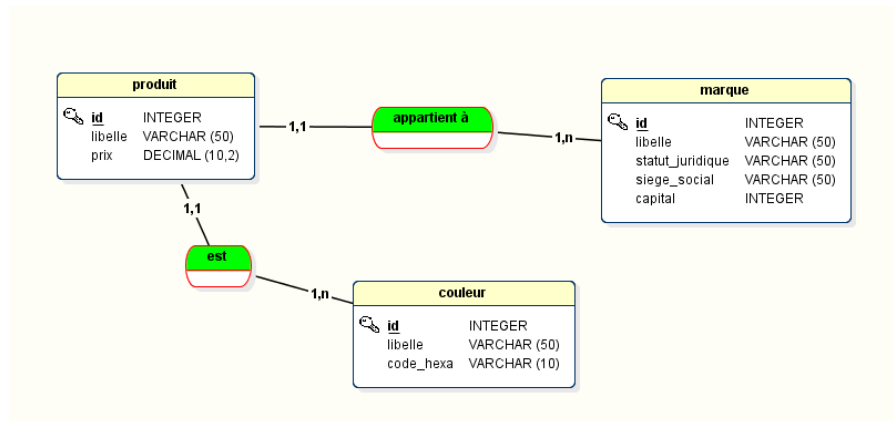
Exercices :

Écrire le tableau de la ou des entités représentées dans ces schemas

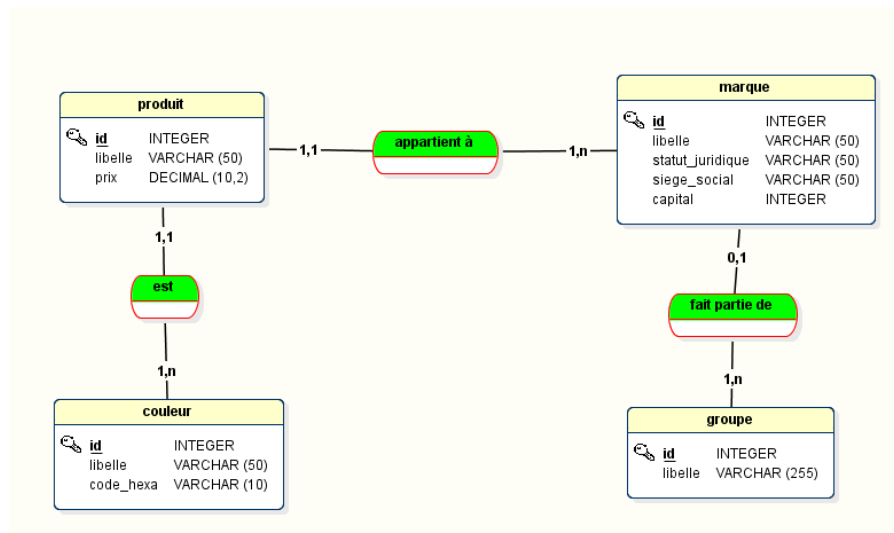
1. Il faut tout modéliser



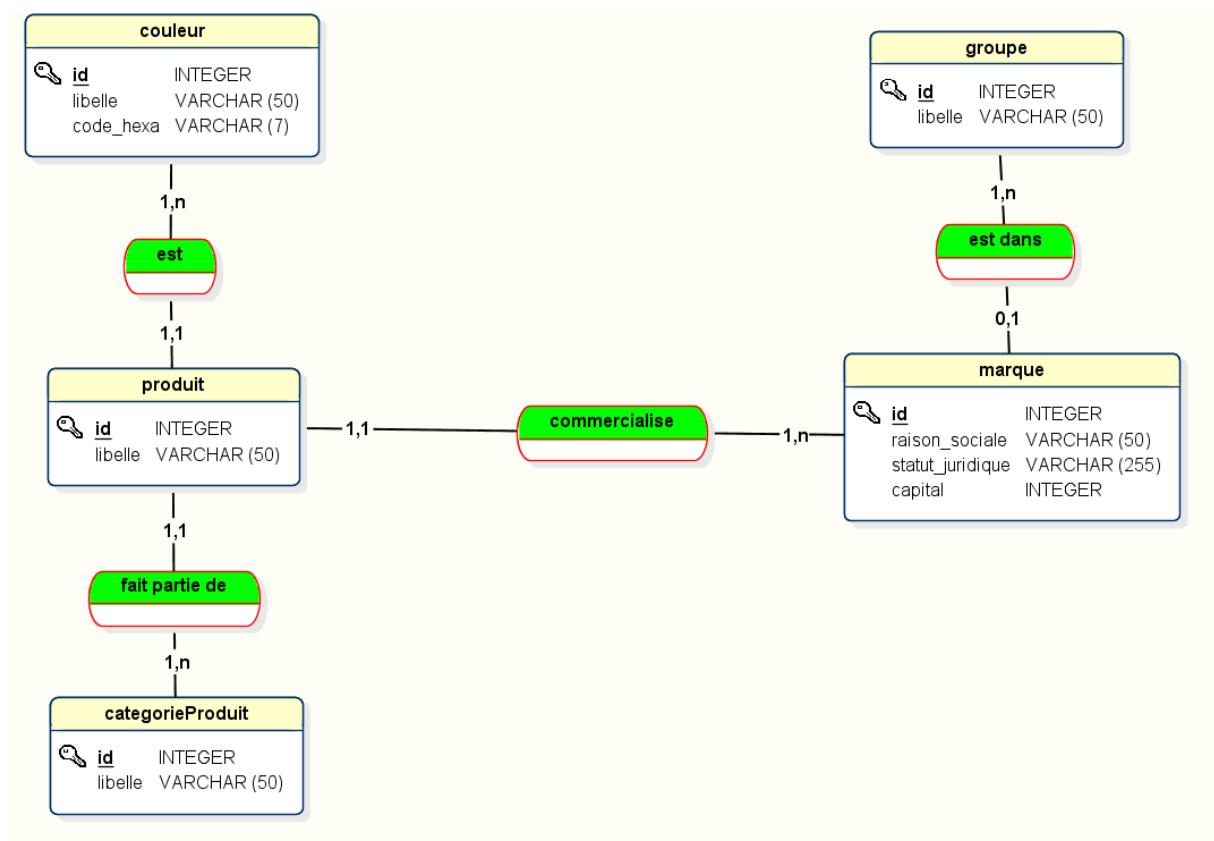
2. Modifier les entités précédentes pour inclure la nouvelle entité couleur



3. Modifier les entités précédentes pour inclure la nouvelle entité groupe



4. Modifier les entités précédentes pour inclure la nouvelle entité categorieProduit et l'association qui la lie à produit. Un produit peut faire partie de plusieurs catégories de produit



5. Modéliser les tableaux suivants :

PROFESSEUR

id	nom	prenom	email	id_matiere
1	SAVOCA	Nicolas	n.savoca@osengo.fr	2
2	DOE	John	j.doe@gmail.com	1

MATIERE

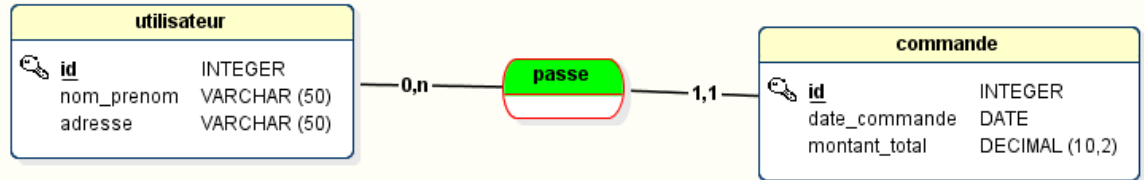
id	libelle
1	Pétanque
2	Développement Web

6. Modéliser les tableaux suivants, et adapter l'existant : un professeur enseigne dans un seul établissement :

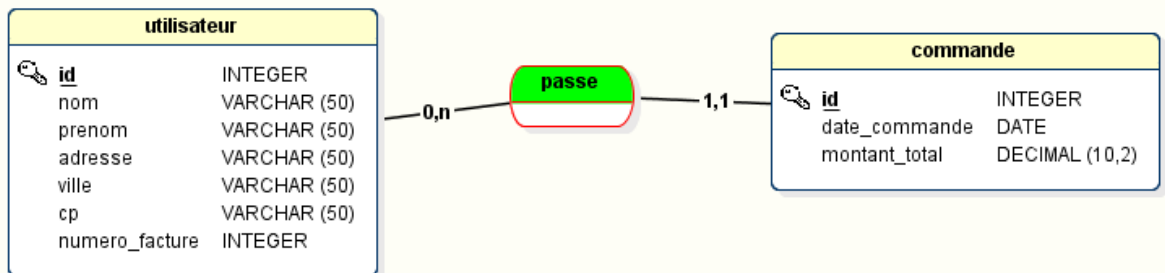
ETABLISSEMENT

id	libelle	adresse	ville	cp
1	OSENGO	2 Av . des Combattants en Afrique du Nord	Moulins	03000
2	HumanBooster	Chez toi	Chez toi	012345

7. Adapter les tableaux existants : un professeur peut enseigner sa matière dans plusieurs établissements...
8. Adapter les tableaux existants : un professeur peut enseigner plusieurs matières
9. Quelle forme normale n'est pas respectée ici ?



10. Et ici ?



11. Et là ?

