



MCD

Model Conceptuel De Données (schéma qui va représenter une base de données).

UNE BASE DE DONNÉES

C'est une collection organisée et structurée de données stockées de manière à permettre la gestion, la recherche, la récupération, la modification et l'analyse efficaces de l'information.

Les bases de données peuvent stocker des données de tous types: des textes, des nombres, des images, des dates, et bien d'autres types encore.

Les bases de données sont incontournables dans tous les domaines.

Nous sommes tous dans des BDD sans forcément le savoir (ex : sécurité sociale, banque, etc.).



Dans quel cas va-t-on avoir besoin d'une base de données ?

- Une base de données est nécessaire dans de nombreux cas où il est important de stocker, organiser et récupérer des données de manière efficace.

Exemple :

L'intégrateur va créer un template , avec du HTML et le développeur Backend va, quant à lui, rendre votre site dynamique avec les données issues de la base de données.

De même, toutes les informations des utilisateurs connectés, seront stockées en base de données.

LES CONCEPTS DE BASE

TABLE

Dans une base de données relationnelle, les données sont généralement organisées dans des **tables**. Chaque table est **composée de lignes et de colonnes** (appelées champs).

Chaque colonne représente **un attribut spécifique de l'information stockée**, tandis que chaque ligne contient une **entrée individuelle de données**.

SGBD (Système de gestion de base de données)

Pour gérer efficacement les données, les bases de données sont généralement utilisées avec un **logiciel appelé SGBD**.

Les SGBD permettent **de créer, de lire, de mettre à jour et de supprimer des données** dans la base de données, ainsi que de définir des **règles** pour garantir l'intégrité des données.

Dans notre cas, nous utiliseront **Phpmyadmin**.

CLÉ PRIMAIRE

Une clé primaire est un champ ou un ensemble de champs qui **identifient de manière unique** chaque enregistrement dans une table. Elle garantit **qu'il n'y a pas de doublons** dans la base de données.

REQUÊTES

Les requêtes permettent aux utilisateurs d'extraire des données spécifiques de la base de données en fonction de critères de recherche ou de filtrage. Nous en verrons plus tard dans le détail.

RELATION

Dans les bases de données relationnelles, plusieurs tables peuvent être liées par des relations, ce qui permet de relier une table à une autre.

|  Important !

Avant de créer une base de données, il faut d'abord la schématiser et s'assurer de sa bonne structure. Il existe la méthode **Merise** qui est une méthode d'analyse, de conception et de réalisation de systèmes d'informations.

Nous n'aborderons pas la méthode **dans son intégralité**, elle est bien trop complexe, nous n'en n'utiliserons qu'une partie qui est **suffisante pour la réalisation de projets web**.

VOICI LES TROIS MODÈLES NOUS CONCERNENT

- *Le MCD (Modèle Conceptuel De Données)*
- *Le MLD (Modèle Logique De Données)*
- *Le MPD (Modèle Physique De Données)*

Une fois en possession de notre **MPD**, nous pourrons alors créer la base de données sur **phpmyadmin**.

CRÉATION DU MCD

Un cinéma nous demande de créer sa base de données et nous fourni quelques explications.

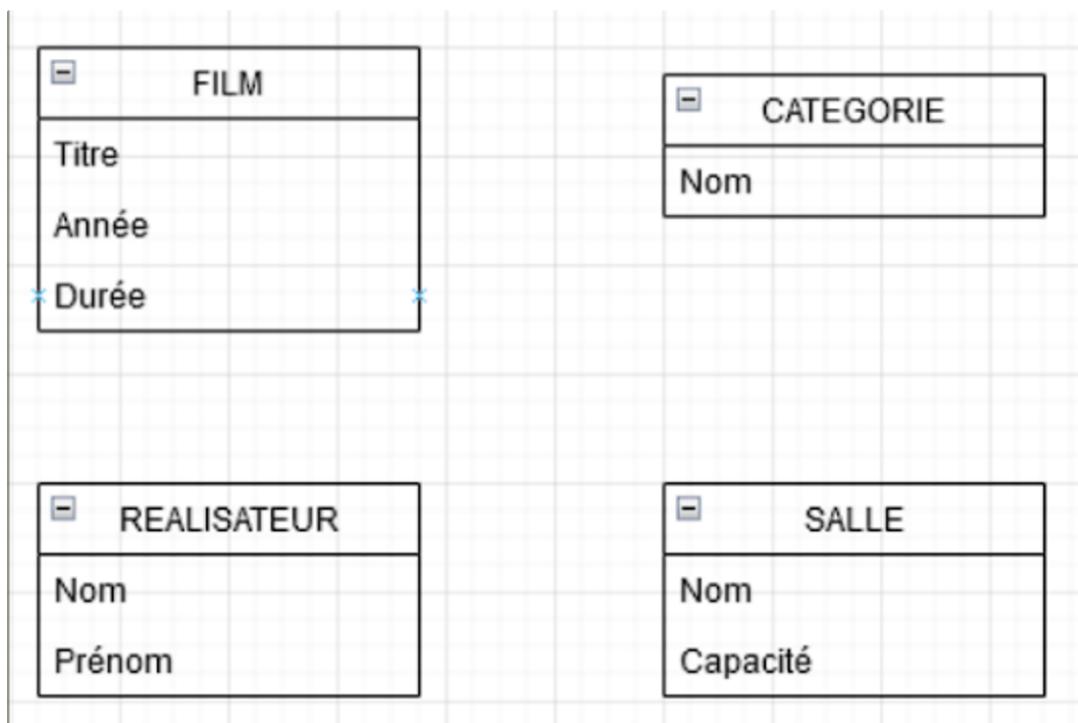
Nous possédons des films qui ont chacun : # une année de sortie # un titre # une durée en minutes # une catégorie (thriller, fantastique, romance) # un réalisateur (nom, prénom). Nous possédons 5 salles qui ont : # une certaine capacité # un nom. Chaque jour, une salle se voit attribuer un film.

1. Définir les tables.
2. Définir les attributs de chaque table.
3. Ensuite nous établirons les relations qui peut y avoir entre ces tables.

Identifions les tables

- Film
- Réalisateur
- Categorie
- Salle

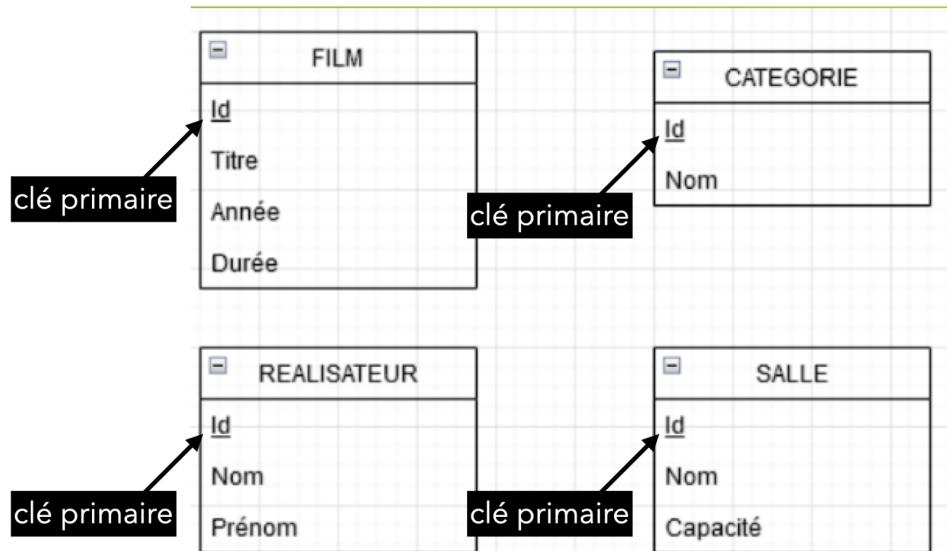
TABLES → ATTRIBUTS



💡 Dans un MCD, il est nécessaire de rajouter des **id** à nos éléments. En effet, pour l'intégrité de la base de données, il est nécessaire **qu'aucun doublon ne soit possible**.

Ainsi, tous les enregistrements auront un **id** unique que l'on appellera **clé primaire**.

CLÉ PRIMAIRE →



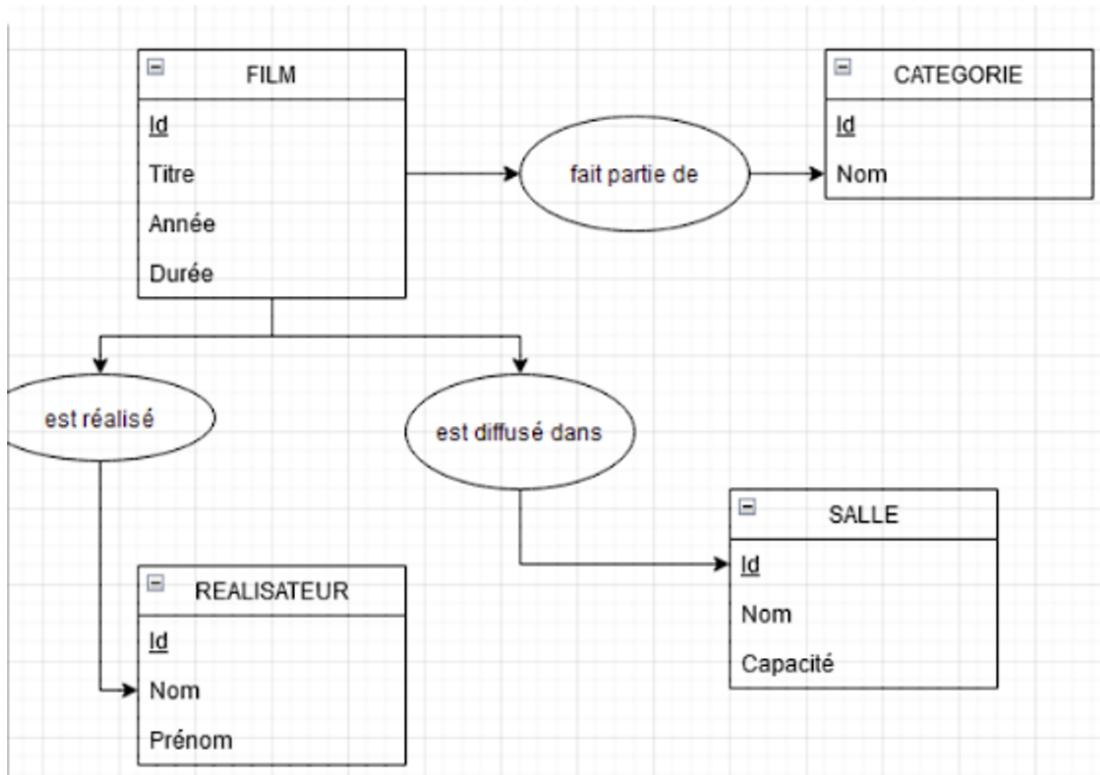
💡 Rajoutons maintenant nos **relations**, un film a été réalisé par un réalisateur, il y a donc un **lien** entre ces deux tables.

Un film fait partie d'une catégorie.

Un film est diffusé.

Un film est réalisé par un réalisateur.

RELATION →



AJOUT DES CARDINALITÉS

Un dernier élément doit être ajouté à notre schéma, ce sont les **cardinalités** qui vont permettre de définir le **type de relation** entre les tables et de définir des **règles de relation**.

Par exemple, entre film et réalisateur, nous allons nous poser la question

- 💡 Un film peut être réalisé par combien de réalisateur minimum, et maximum ?

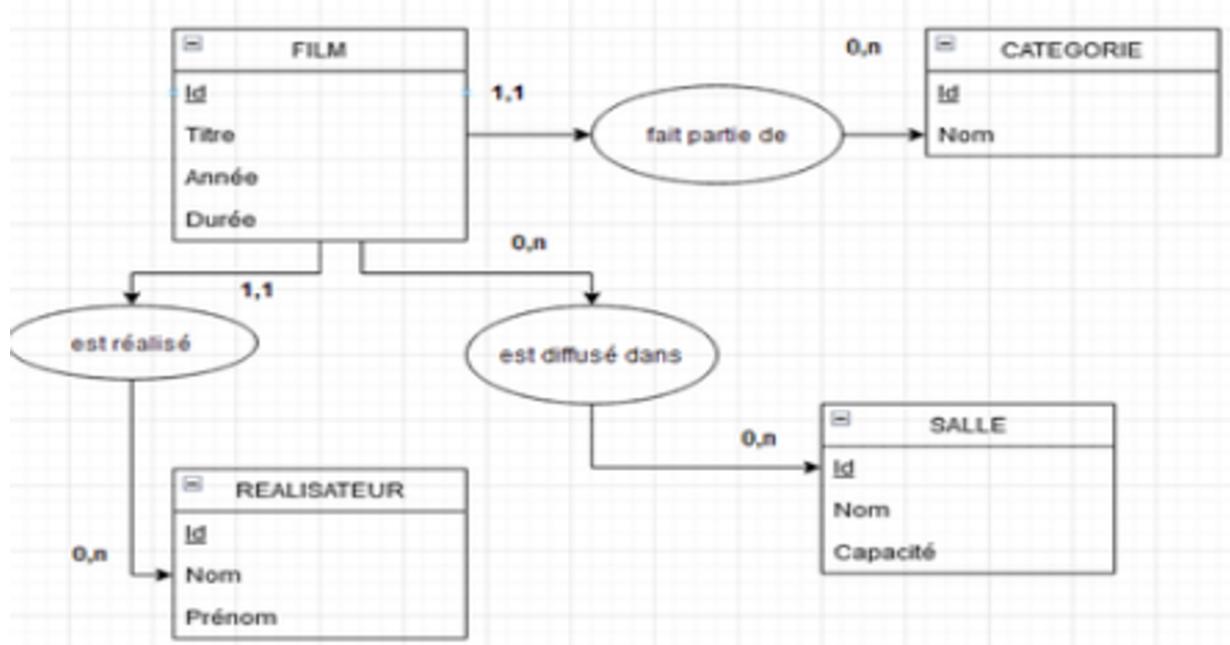
La réponse est minimum **1** réalisateur et maximum **1** réalisateur.

De même, nous allons aussi nous poser cette question dans l'autre sens.

- 💡 Un réalisateur peut avoir réalisé combien de films au minimum et au maximum ?

La réponse est minimum **0** et maximum "**n**" (pour dire plusieurs).

Voici ce que ça donne sur notre MCD, avec les autres cardinalités rajoutées:

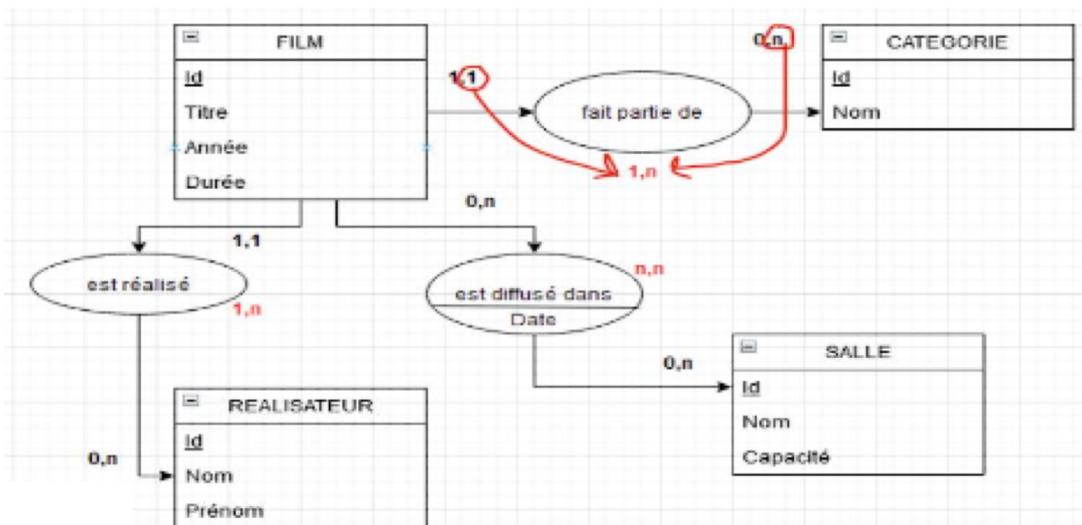


Nous devons encore déterminer comment les relations sont liées les unes aux autres, et cela dépend des deux relations qui les entourent.

Par exemple, entre FILM et CATEGORIE, il y a une relation 1,1 et 0,n.

Nous allons alors prendre les deux valeurs maximales de ces deux cardinalités.

Pour créer la cardinalité de la relation, ce qui nous donne 1, n:



De plus, vous remarquerez que l'on a rajouté "Date".

Le cinéma nous avait bien dit "**Chaque jour, une salle se voit attribuer un film.**

Il fallait rajouter cette notion quelque part et c'est là sa place.

Relation de type 1-1

- C'est une relation dans laquelle chaque occurrence de données de la première table **est en relation avec une (et une seule) occurrence** de données de la deuxième table.

Relation de type 1-N

- C'est une relation dans laquelle une occurrence de données de la première table **est en relation avec plusieurs occurrences** de données de la deuxième table, sachant que chaque occurrence de cette seconde table n'est en relation qu'avec une occurrence de la première.



Exemple : une personne peut emprunter plusieurs livres, alors que un livre ne peut être emprunté que par une personne.

Relation de type N-N

- C'est une relation dans laquelle **chaque occurrence de données de la première table peut être en relation avec une multitude de données de la deuxième table**, et chaque occurrence de données de la seconde table est elle-même en relation avec de nombreuses données de la première table.

REGLES



Si deux tables sont reliées par une seule relation de type 1-1, on cherche à ne faire qu'une seule table à partir des deux.

💡 Si deux tables sont liées par une relation de type **1-N** (un-à-plusieurs), on introduit un champ supplémentaire du côté de la table associée à la relation N.

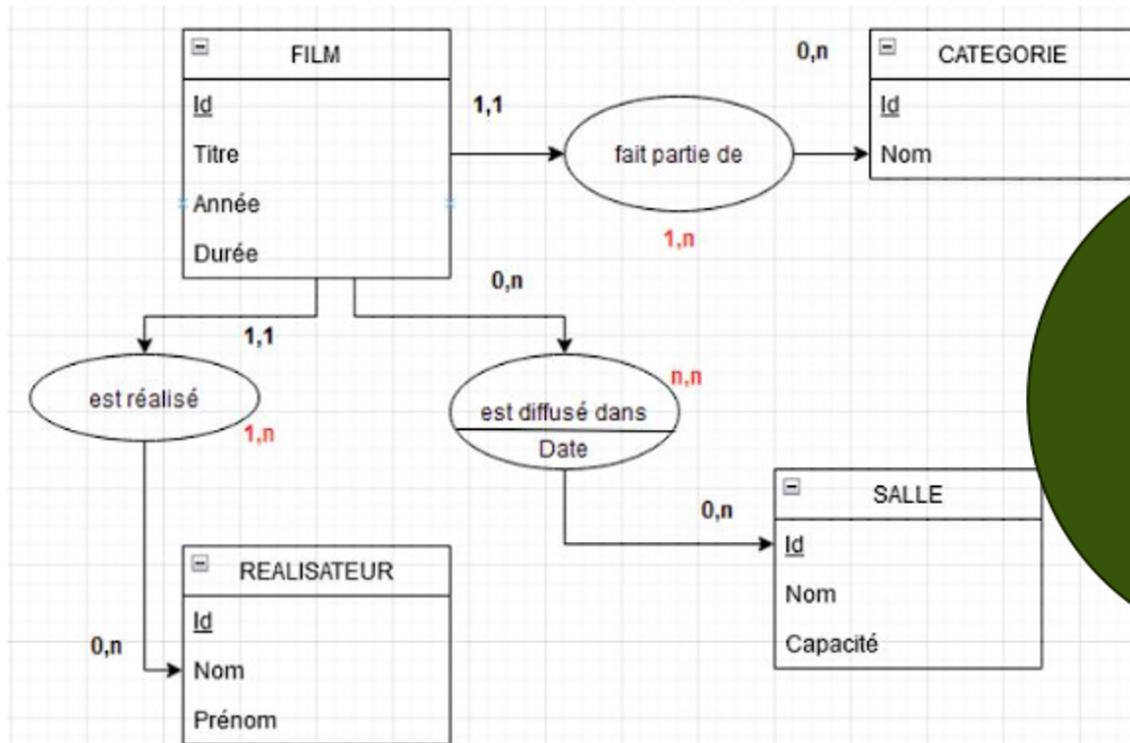
Ce champ supplémentaire est appelé **clé étrangère** et il permet d'établir le lien entre les deux tables en faisant référence à la **clé primaire** de l'autre table.

Cela permet de représenter les relations entre les données et d'assurer l'intégrité référentielle dans la base de données.

💡 Si deux tables sont reliées par une relation de type **N-N**, on va créer une table supplémentaire, appelée **table de relation**, qui sera porteuse des clés étrangères des deux tables.

MLD (Modèle Logique des Données)

NOUS ALLONS TRANSFORMER CE MCD EN MLD



 Le MLD s'écrit sous la forme de fonctions où le nom de la fonction est la "table" et ses arguments sont les "champs".

Dans notre cas, nous n'avons pas de relation 1, 1. Passons donc à la relation 1, n entre *film* et *catégorie*, selon la règle, nous allons ajouter dans l'entité *film* une clé étrangère qui correspondra à la clé primaire de *catégorie*, par convention, nous mettrons un # devant cette clé étrangère, FILM devient :

```
FILM(id, titre, année, durée, #id_categorie)
```

Gérons également et selon la même règle la relation entre *film* et *réalisateur* :

```
FILM(id, titre, année, durée, #id_categorie, #id_realisateur)
```

Les tables *catégorie* et *réalisateur*, elles, n'ont rien de plus que dans le MCD, ce qui nous donne

```
FILM(id, titre, année, durée, #id_categorie, #id_realisateur)
CATEGORIE(id, nom) REALISATEUR(id, nom, prenom)
```

La table *SALLE* est , quant à elle, liée à *FILM* par une relation n, n , ce qui va provoquer la création d'une table supplémentaire, que l'on peut nommer "DIFFUSION", et l'on sait qu'une diffusion a lieu à une date précise, ce qui nous donne :

```
FILM(id, titre, année, durée, #id_categorie, #id_realisateur)
CATEGORIE(id, nom) REALISATEUR(id, nom, prenom) DIFFUSION(#id_salle,
#id_film, da
```

MPD (*Modèle physique de données*)

Un MPD est généralement créé après avoir défini un Modèle Conceptuel de Données (MCD) et un Modèle Logique de Données (MLD). Le MPD se concentre sur les aspects techniques de la base de données et décrit comment les données sont stockées sur le disque.

Reprendons notre MLD et pour chaque entité, nous allons créer une table, lister les champs en prenant soin de respecter ces règles

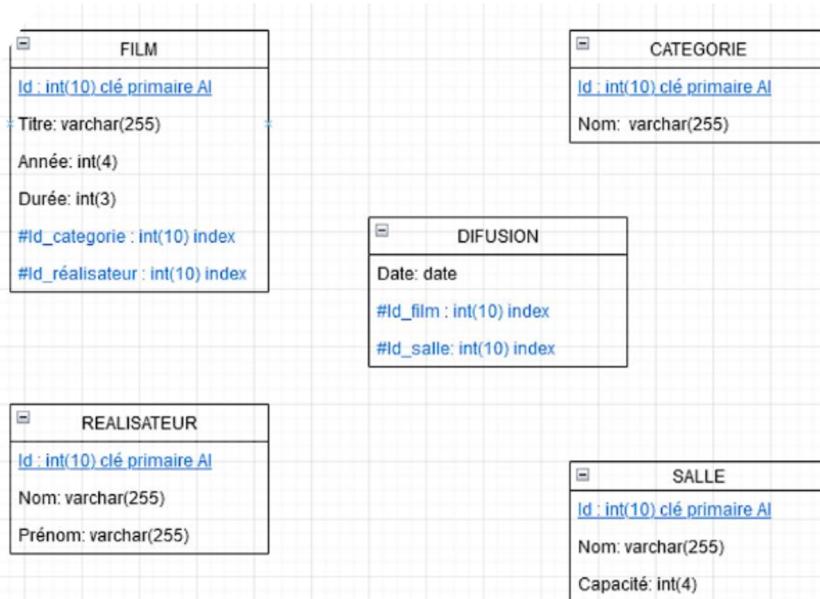
- La clé primaire ou index unique est soulignée.
- Les clés secondaires sont précédées d'une #.

Puis nous allons, pour chaque champ, définir ses contraintes techniques

Le type de données que contiendra le champ, son poids en octets si il y a lieu (*par exemple pour une date ce n'est pas nécessaire puisqu'elle devra respecter le format date*)

Si c'est un index ou une clé primaire

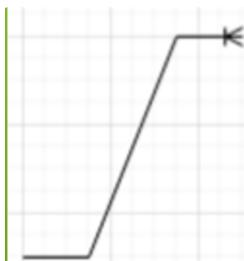
Si c'est un champ qui doit être auto incrémenté



Il ne faut pas oublier les relations, qui sur notre **MCD**, étaient représentées par des cercles avec des mots et des cardinalités.

Nous allons les schématiser un peu différemment sur notre **MPD**, plus de mot, plus de cardinalités mais une jointure à une ou plusieurs branches.

Ce type de relation est 1 à n, en effet on voit sur le schéma que la relation part d'une seule branche à gauche pour se diviser en plusieurs, à droite, donc de 1 à plusieurs.



Remarquez :