# Editeur

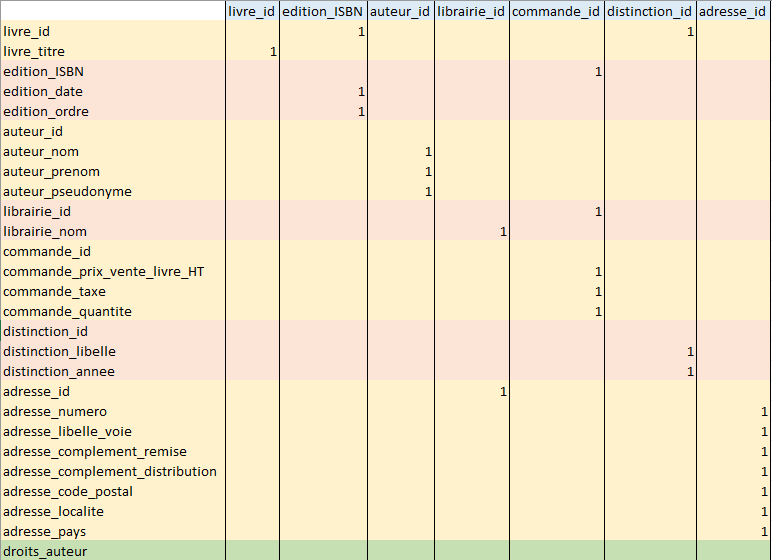
# Récolte des besoins

Un éditeur souhaite installer une base de données pour mémoriser les informations suivantes :   
- Les livres sont identifiés par leur no ISBN. - Un livre possède un titre et un prix de vente.   
- Il est écrit par un ou plusieurs auteurs.   
- Chaque livre est tiré en une ou plusieurs éditions, datées et identifiées par leur ordre (première édition, seconde édition, etc.).   
- Chaque édition comporte un certain nombre d'exemplaires.   
- Un livre peut être primé (Goncourt, Fémina etc...).   
- Les auteurs sont identifiés par leur nom et prénom et peuvent avoir un pseudonyme.   
- Pour chaque livre, un auteur perçoit des droits d'auteur, calculés comme un pourcentage du prix de vente (il est aussi fonction du nombre d'auteurs, du tirage, etc.).   
- Les libraires (identifiés par leur nom et adresse complète) peuvent envoyer des commandes d'un ou plusieurs livres en quantité quelconque.

# Dictionnaire des données

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entité | Mnémonique | Signification | Type(longueur) | Contraintes |
| Livre | livre\_id | identifiant du livre | INT(11) | identifiant, ai |
| livre\_titre | titre du livre | VARCHAR(100) | obligatoire |
| Edition | edition\_ISBN | ISBN de l’édition du livre | CHAR(17) | identifiant format ISBN standard |
| edition\_date | date d’édition du livre | DATE | obligatoire |
| edition\_ordre | ordre d’édition du livre (première édition, seconde édition …) | VARCHAR(50) | obligatoire |
| Auteur | auteur\_id | identifiant de l’auteur | INT(11) | identifiant, ai |
| auteur\_nom | nom de l’auteur | VARCHAR(50) | facultatif |
| auteur\_prenom | prénom de l’auteur | VARCHAR(50) | facultatif |
| auteur\_pseudonyme | pseudonyme de l’auteur | VARCHAR(50) | facultatif |
| Librairie | librairie\_id | identifiant de la librairie | INT(11) | identifiant, ai |
| librairie\_nom | nom de la librairie | VARCHAR(50) | obligatoire |
| Commande | commande\_id | identifiant de la vente | INT(11) | identifiant, ai |
| commande\_prix\_vente\_HT | prix de vente du livre | DECIMAL(6,2) | obligatoire |
| commande\_taxe | TVA | DECIMAL(5,2) | obligatoire |
| commande\_quantite | quantité de livres commandés | INT(10) | obligatoire |
| Distinction | distinction\_id | identifiant de la distinction | INT(11) | obligatoire, ai |
| distinction\_libelle | libellé de la distinction  (Goncourt, Fémina…) | VARCHAR(50) | obligatoire |
| distinction\_annee | année de la distinction | INT(4) | obligatoire |
| Adresse | adresse\_id | identifiant de l’adresse | INT(11) | identifiant, ai |
| adresse\_numero | numéro | VARCHAR(5) | facultatif |
| adresse\_libelle\_voie | libelle de voie | VARCHAR(33) | facultatif |
| adresse\_complement \_remise | complément de remise | VARCHAR(38) | facultatif |
| adresse\_complement\_distribution | complément de distribution | VARCHAR(38) | facultatif |
| adresse\_code\_postal | code postal | CHAR(5) | obligatoire |
| adresse\_localite | localité | VARCHAR(33) | obligatoire |
| adresse\_pays | pays | VARCHAR(38) | facultatif |
|  | droits\_auteur | pourcentage de droits d’auteur sur le livre | DECIMAL(5,2) | obligatoire |

# Matrice



# Règles de gestion

Un livre est écrit par 1 ou plusieurs auteurs  
Un auteur écrit 1 ou plusieurs livres

Un livre a obtenu 0 ou plusieurs distinctions  
Une distinction est obtenu pour 1 seul livre

Un livre concerne en 1 ou plusieurs éditions  
Une édition concerne 1 seul livre

Une commande comprend 1 seule édition  
Un édition est comprise dans 0 ou plusieurs commandes

Une librairie effectue 1 ou plusieurs commandes  
Une commande est effectuée par 0 ou plusieurs librairies

Une librairie est domiciliée a 1 seule adresse  
A une adresse est domiciliée 0 ou 1 librairie

Une commande rémunère 0 ou plusieurs auteurs  
Un auteur est rémunéré par 0 ou plusieurs commandes

# Dépendances fonctionnelles

## Dépendances fonctionnelles simples

livre\_id -> livre\_titre

edition\_ISBN -> edition\_date, edition\_ordre, #livre\_id

auteur\_id -> auteur\_nom, auteur\_prenom, auteur\_pseudonyme

librairie\_id -> librairie\_nom, #ladresse\_id

commande\_id -> commande\_prix\_vente\_HT, commande\_taxe, commande\_quantite, #edition\_ISBN, #librairie\_id

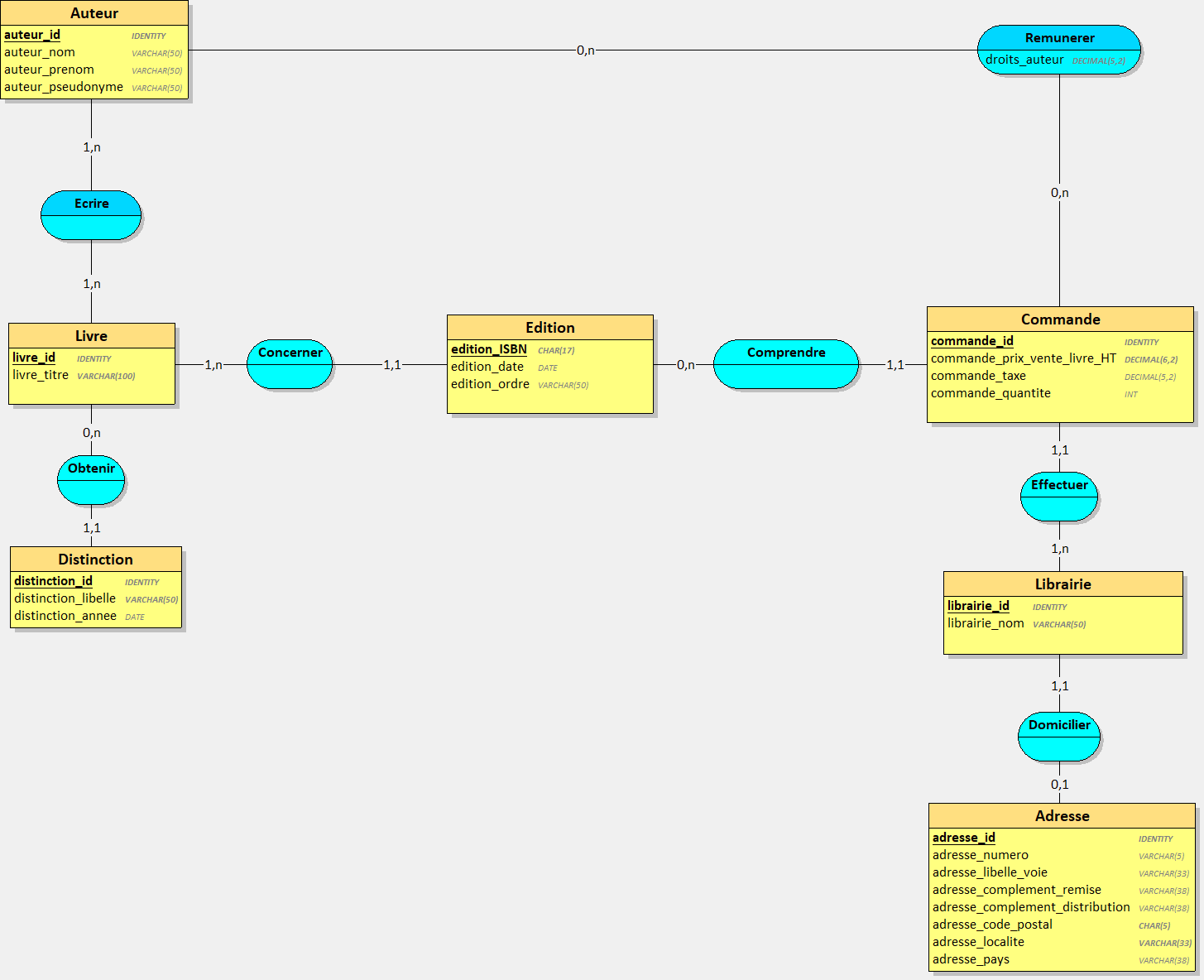
distinction\_id -> distinction\_libelle, distinction\_annee, #livre\_id

adresse\_id -> adresse\_numero, adresse\_libelle\_voie, adresse\_complement\_remise, adresse\_complement\_distribution, adresse\_code\_postal, adresse\_localite, adresse\_pays

## Dépendances fonctionnelles composées

commande\_id, auteur\_id -> droits\_auteur

# Modèle conceptuel des données



# Modèle relationnel des données

Adresse(adresse\_id, adresse\_numero, adresse\_libelle\_voie, adresse\_complement\_remise, adresse\_complement\_distribution, adresse\_code\_postal, adresse\_localite, adresse\_pays)

Librairie(librairie\_id, librairie\_nom, #adresse\_id)

Livre(livre\_id, livre\_titre)

Edition(Edition\_ISBN, edition\_date, edition\_ordre, #livre\_id)

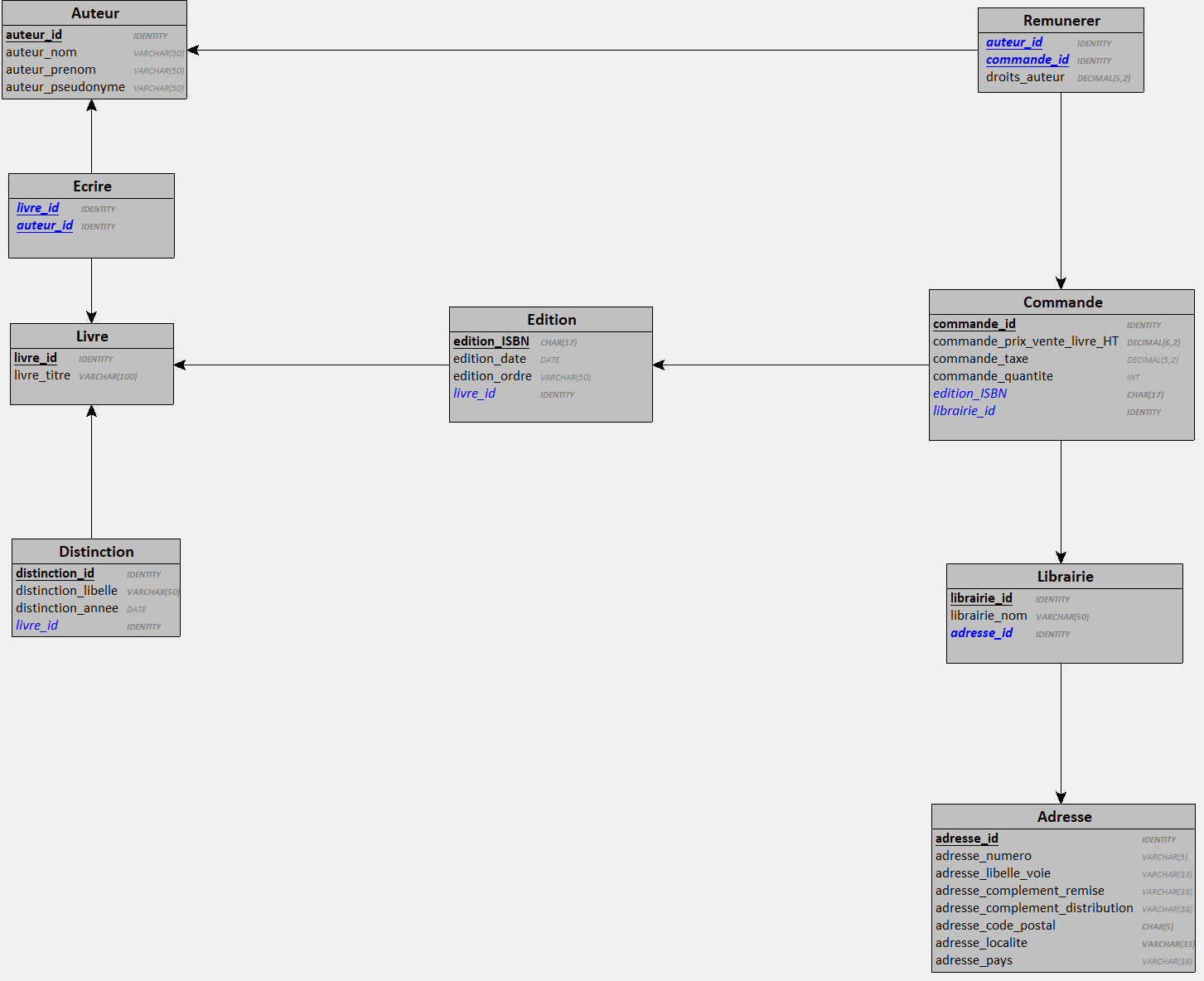
Distinction(distinction\_id, distinction\_libelle, distinction\_annee, #livre\_id)

Commande(commande\_id, commande\_prix\_vente\_HT, commande\_taxe, commande\_quantite, #librairie\_id, #edition\_ISBN)

Auteur(auteur\_id, auteur\_nom, auteur\_prenom, auteur\_pseudonyme)

Auteur\_Livre(#auteur\_id, #livre\_id)

Auteur\_Commande(#auteur\_id, #commande\_id, droits\_auteur)



# Création de la base de données (DDL) Sql Server

drop table if exists Auteur\_Commande;

drop table if exists Auteur\_Livre;

drop table if exists Auteur;

drop table if exists Commande;

drop table if exists Distinction;

drop table if exists Edition;

drop table if exists Livre;

drop table if exists Librairie;

drop table if exists Adresse;

create table Adresse

(

adresse\_id int identity(1,1)

,adresse\_numero varchar(5)

,adresse\_libelle\_voie varchar(33)

,adresse\_complement\_remise varchar(38)

,adresse\_complement\_distribution varchar(38)

,adresse\_code\_postal char(5) not null

,adresse\_localite varchar(33) not null

,adresse\_pays varchar(38)

,primary key (adresse\_id)

)

;

create table Librairie

(

librairie\_id int identity(1,1)

,librairie\_nom varchar(50) not null

,adresse\_id int

,primary key (librairie\_id)

)

;

create table Livre

(

livre\_id int identity(1,1)

,livre\_titre varchar(100) not null

,primary key (livre\_id)

)

;

create table Edition

(

edition\_ISBN char(17)

,edition\_date date not null

,edition\_ordre varchar(50) not null

,livre\_id int

,primary key (edition\_ISBN)

)

;

create table Distinction

(

distinction\_id int identity(1,1)

,distinction\_libelle varchar(50) not null

,distinction\_annee int not null

,livre\_id int

,primary key (distinction\_id)

)

;

create table Commande

(

commande\_id int identity(1,1)

,commande\_prix\_vente\_HT decimal(6,2) not null

,commande\_taxe decimal(5,2) not null

,commande\_quantite int not null

,librairie\_id int

,primary key (commande\_id)

)

;

create table Auteur

(

auteur\_id int identity(1,1)

,auteur\_nom varchar(50)

,auteur\_prenom varchar(50)

,auteur\_pseudonyme varchar(100)

,primary key (auteur\_id)

)

;

create table Auteur\_Livre

(

auteur\_id int

,livre\_id int

,primary key (auteur\_id, livre\_id)

)

;

create table Auteur\_Commande

(

auteur\_id int

,commande\_id int

,droits\_auteur decimal(5,2) not null

,primary key (auteur\_id, commande\_id)

)

;

alter table Librairie

add foreign key (adresse\_id) references Adresse(adresse\_id)

;

alter table Edition

add foreign key (livre\_id) references Livre(livre\_id)

;

alter table Distinction

add foreign key (livre\_id) references Livre(livre\_id)

;

alter table Commande

add foreign key (librairie\_id) references Librairie(librairie\_id)

;

-- Comment faire pour intégrer plusieurs foreign key dans un même alter table ?

alter table Auteur\_Livre

add foreign key (auteur\_id) references Auteur(auteur\_id)

;

alter table Auteur\_Livre

add foreign key (livre\_id) references Livre(livre\_id)

;

alter table Auteur\_Commande

add foreign key (auteur\_id) references Auteur(auteur\_id)

;

alter table Auteur\_Commande

add foreign key (commande\_id) references Commande(commande\_id)

;