


# TD 9 : Modèle relationnel des données

## Conversion d'un modèle E/A en un modèle relationnel

2025-11-28

 Avec solutions

L3 MIASHS  
Université Paris Cité  
Année 2025  
[Course Homepage](#)  
[Moodle](#)



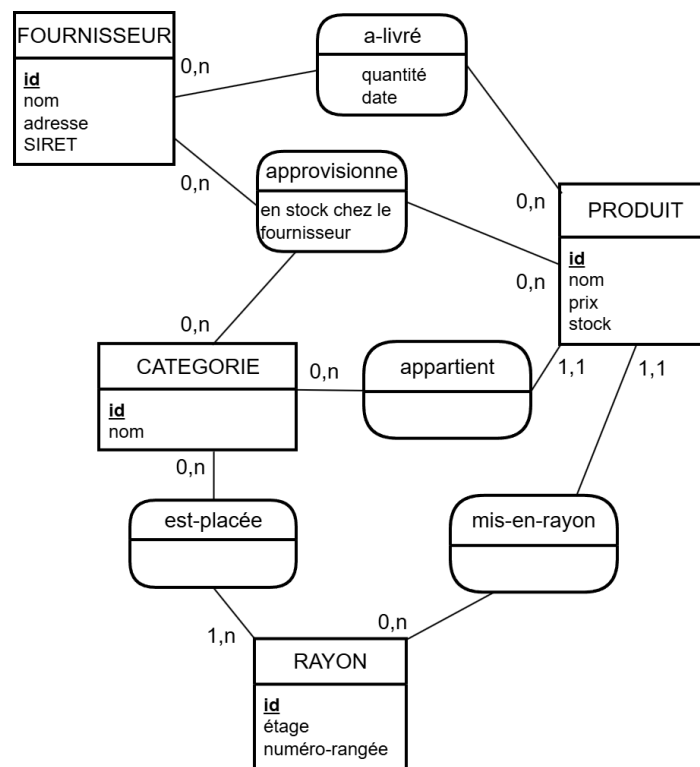
### ! Objectifs

L'objectif de cette séance est de traduire un diagramme E/A (MCD) en un modèle relationnel (MRD).

### Exercice (Supermarché)

L'équipe de conception a étudié le SI d'un supermarché et a fourni le MCD suivant pour la partie approvisionnement et mise en rayon.

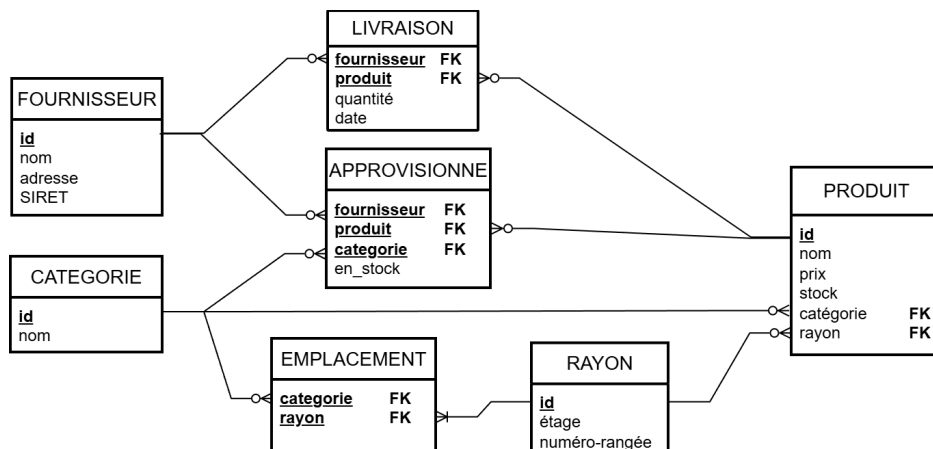
La propriété "en stock chez le fournisseur" est un booléen.



### i Question

Convertir ce diagramme E/A en un modèle relationnel.

### Solution



### i Question

Quelles sont les clés étrangères sur lesquelles il faut ajouter une contrainte NOT NULL ou UNIQUE ?

### Solution

D'après le MCD, il faut définir une contrainte NOT NULL sur les clés étrangères **rayon** et **catégorie** de la table **produit** pour traduire les cardinalités minimales 1 des associations *mis-en-rayon* et *appartient* du côté de *PRODUIT*.

### i Question

Ecrire en SQL la définition des données de la partie *PRODUIT*, *CATEGORIE*, *RAYON*, en essayant d'implémenter toutes les contraintes du diagramme E/A.

Une cardinalité n'est pas codable par une contrainte de table ou de champ, laquelle ? Comment peut-on généraliser ?

### Solution

```

CREATE TABLE RAYON(
    id INT PRIMARY KEY,
    etage INT NOT NULL,
    rangee INT NOT NULL
);

CREATE TABLE CATEGORIE(
    id INT PRIMARY KEY,
    nom VARCHAR(50)
);

CREATE TABLE PRODUIT(
    id INT PRIMARY KEY,
    nom VARCHAR(50),
    prix MONEY NOT NULL,
    stock INT NOT NULL DEFAULT 0,
    categorie INT NOT NULL,
    rayon INT NOT NULL
);
    
```

```
rayon INT NOT NULL,  
CONSTRAINT PRODUIT_categorie_fk  
    FOREIGN KEY (categorie)  
    REFERENCES CATEGORIE(id),  
CONSTRAINT PRODUIT_rayon_fk  
    FOREIGN KEY (rayon)  
    REFERENCES RAYON(id)  
);  
  
CREATE TABLE EMPLACEMENT(  
    categorie INT,  
    rayon INT,  
    PRIMARY KEY (categorie, rayon),  
    CONSTRAINT EMPLACEMENT_categorie_fk  
        FOREIGN KEY (categorie)  
        REFERENCES CATEGORIE(id),  
    CONSTRAINT EMPLACEMENT_rayon_fk  
        FOREIGN KEY (rayon)  
        REFERENCES RAYON(id)  
);
```

🔥 La cardinalité minimale 1 de l'association *est-placée* du côté de *RAYON* n'est pas codable par une contrainte de table ou de champ.  
On peut généraliser à **toute cardinalité minimale 1 qui se trouve sur un côté *plusieurs* n'est pas codable par une contrainte de table ou de champ.**

### **i** Question

Ce modèle E/A présente une erreur de conception, indépendamment de la connaissance du SI qu'il modélise. Laquelle ?  
Comment faut-il normaliser ce modèle ?

### Solution

D'après la cardinalité *1 :1* entre les entités *PRODUIT* et *CATEGORIE*, il y a une dépendance fonctionnelle *produit.id* → *categorie.id*.

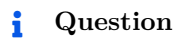
Dans la table *approvisionnement*, on a donc la dépendance fonctionnelle *produit* → *categorie* qui contrevient aux règles de normalisation que nous allons voir en cours. L'idée est qu'il y a redondance d'information avec la table *produit* dans laquelle cette dépendance fonctionnelle est déjà enregistrée.

Il faut décomposer l'association *approvisionnement* en :

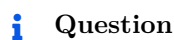
- une association binaire *appartient* qui existe déjà,
- une association binaire *approvisionne* entre *fournisseur* et *produit*.

### Exercice

On reprend un MCD élaboré dans un TD précédent :



### Solution



### Solution

- D'après le MCD, une contrainte NOT NULL sur la clé étrangère **type** de la table **véhicule** pour traduire la cardinalité minimale 1 de l'association *est de* du côté de *véhicule*.

- D'après le MCD, une contrainte NOT NULL sur la clé étrangère `service` de la table `emprunteur` pour traduire la cardinalité minimale 1 de l'association *attaché* du côté de *EMPRUNTEUR*.

### **i** Question

Ecrire en SQL la définition des données du modèle relationnel en essayant d'implémenter les contraintes du diagramme E/A.

### Solution

```
CREATE TABLE TYPE (  
    numero INT PRIMARY KEY,  
    Nom VARCHAR(25) NOT NULL  
);  
  
CREATE TABLE VEHICULE (  
    id INT PRIMARY KEY,  
    immat VARCHAR(10) NOT NULL UNIQUE,  
    type INT NOT NULL,  
    date_mes DATE NOT NULL,  
    km INT NOT NULL,  
    FOREIGN KEY (type) REFERENCES TYPE(numero)  
);  
  
CREATE TABLE GARAGE (  
    numero INT PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE SERVICE (  
    id INT PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL UNIQUE  
);  
  
CREATE TABLE EMPRUNTEUR (  
    numero INT PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL,  
    service INT NOT NULL,  
    FOREIGN KEY (service) REFERENCES SERVICE(id)  
);  
  
CREATE TABLE EMPRUNT (  
    vehicule INT,  
    emprunteur INT,  
    date DATE,  
    essence MONEY,  
    km INT NOT NULL,  
    durée DATE NOT NULL,  
    PRIMARY KEY (vehicule, emprunteur, date),  
    FOREIGN KEY (vehicule) REFERENCES VEHICULE(id),  
    FOREIGN KEY (emprunteur) REFERENCES EMPRUNTEUR(numero)  
);  
  
CREATE TABLE REPARATION (  
    vehicule INT,  
    garage INT,  
    date DATE,  
    prix MONEY NOT NULL,  
    durée DATE,
```

```
PRIMARY KEY (vehicule, garage, date),  
FOREIGN KEY (vehicule) REFERENCES VEHICULE(id),  
FOREIGN KEY (garage) REFERENCES GARAGE(numero)  
);  
  
CREATE TABLE REVISION (  
    vehicule INT,  
    garage INT,  
    date DATE,  
    prix MONEY NOT NULL,  
    durée DATE,  
    PRIMARY KEY (vehicule, garage, date),  
    FOREIGN KEY (vehicule) REFERENCES VEHICULE(id),  
    FOREIGN KEY (garage) REFERENCES GARAGE(numero)  
);
```

Des contraintes NOT NULL ont été ajoutées aux attributs qui ne sont pas des clés étrangères en fonction du “bon sens”.

On a créé des clés primaires entières pour les tables **vehicule** et **service** car il est plus sûr d'utiliser des clés primaires internes sans signification externe à la BDD. Sinon il y a un risque de modification dans le temps qui imposerait une mise à jour en cascade des clés étrangères. Une telle mise à jour aurait une complexité importante et toute requête devrait cesser pendant cette mise à jour.

Par conséquent, les clés primaires récusées **immat** de **vehicule** et **nom** de **service** doivent avoir les contraintes NOT NULL UNIQUE.