

- L3 MIASHS/Ingémath/METIS
- Université Paris Cité
- Course Homepage



## CC2 du 2 décembre 2025

### Exercice 1

On veut concevoir une base de données pour enregistrer des opérations chirurgicales dans une clinique.

Doivent être enregistrés dans la base :

- les soignants avec un numéro, un nom et un prénom.
- les patients avec un nom, un prénom et un numéro de sécurité sociale. Un nouveau patient est enregistré avant la création de l'opération qui le concerne.

La future interface de saisie d'une opération se présentera ainsi :

- Saisie du numéro de patient.
- Saisie de la date-heure et de la durée.
- Saisie de la liste des spécialités nécessaires avec, pour chaque spécialité le nombre d'heures x personnes nécessaires. Au moins une spécialité est requise pour chaque opération.
- Les spécialités sont choisies à chaque fois parmi une liste préalablement enregistrée dans la base de données.

Une fois l'opération enregistrée dans la base de données, une deuxième interface permettra d'affecter les soignants nécessaires à chaque opération :

- Pour chaque opération, on enregistre les soignants qui interviendront et la spécialité dans laquelle chacun interviendra avec le nombre d'heures de son intervention pour cette spécialité.

#### i Question

Ecrire un diagramme E/A, avec les cardinalités, représentant la modélisation conceptuelle des données de ce SI.

**Justifier brièvement** les cardinalités de chaque association (quelques lignes maximum par association).

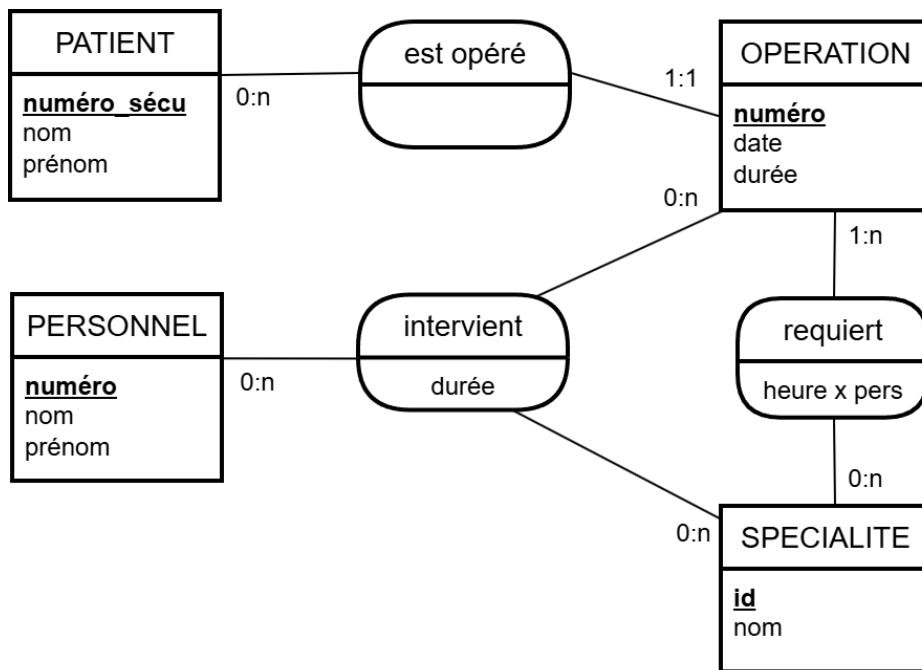
Pas de justifications demandées pour le choix des entités et des associations.

### Solution

Les cardinalités minimales  $0$  et les cardinalités maximales  $n$  représentent une absence de contraintes. Ce sont donc les cardinalités par défaut. Il faut justifier celles qui sont différentes (égales à  $1$ ).

- lors de la création d'une nouvelle opération chirurgicale dans la base, le patient doit être enregistré préalablement et le formulaire demande la saisie du numéro du client opéré donc la cardinalité minimale de *est opéré* est  $1$  du côté de *OPERATION*. La cardinalité maximale est  $1$  car le formulaire prévu ne permet d'enregistrer qu'un seul numéro de client.
- lors de la création d'une opération dans la base, au moins une spécialité est requise donc la cardinalité minimale de l'association *requiert* est  $1$  du côté *OPERATION*.

Au contraire, l'enregistrement des soignants intervenant sur une opération est effectué après la création de l'opération dans la base, donc la cardinalité minimale de l'association *intervient* est  $0$  du côté *OPERATION*.

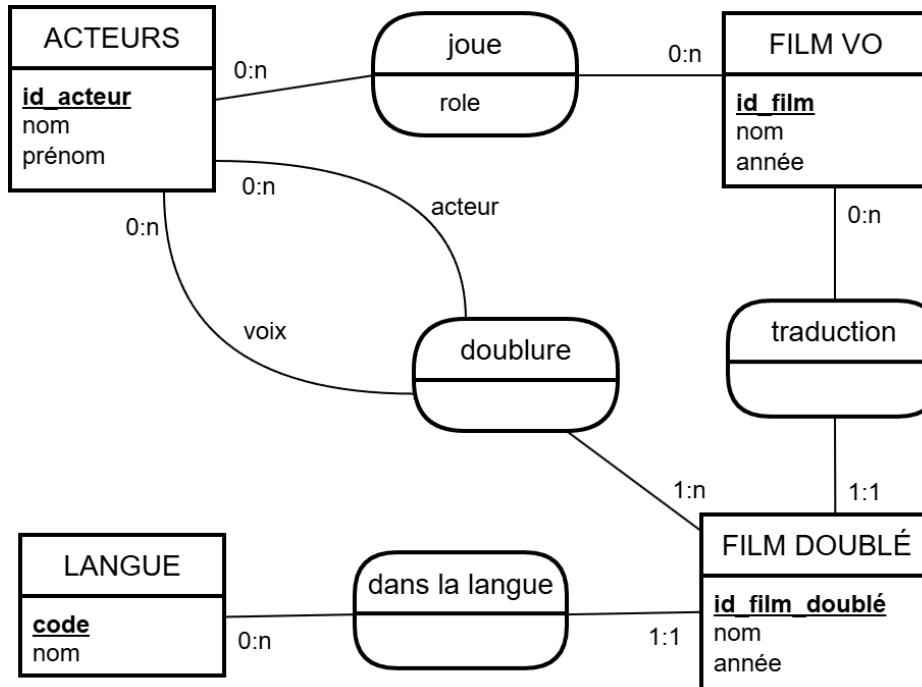


## Exercice 2

On veut créer une BDD pour enregistrer la liste des acteurs ayant joué dans les films en VO et la liste de ceux qui ont fait les voix des versions doublées. On a choisi le modèle conceptuel des données ci-dessous.

Indications complémentaires :

- l'entité *FILM DOUBLÉ* représente les versions non originales des films.
- l'association *joue* enregistre les acteurs de chaque film en VO, en indiquant le nom du rôle joué.
- l'association *doublure* enregistre pour chaque version non originale d'un film, et chaque acteur qui y joue (rôle *acteur*), le nom de celui qui fait sa doublure (rôle *voix*).

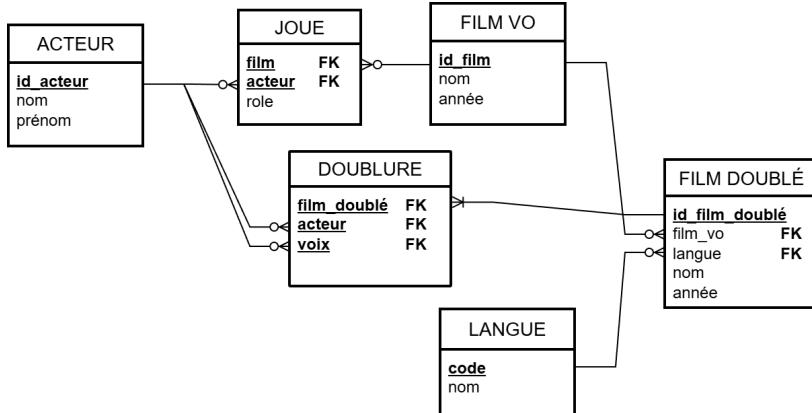


### i Question

Convertir ce diagramme E/A en un modèle relationnel.  
Pas de justifications demandées.

### ?

### Solution



### Question

- i Quelles sont les clés étrangères sur lesquelles il faudra ajouter une contrainte NOT NULL ou UNIQUE lors de la définition des tables en SQL ? **Justifier**.

### ?

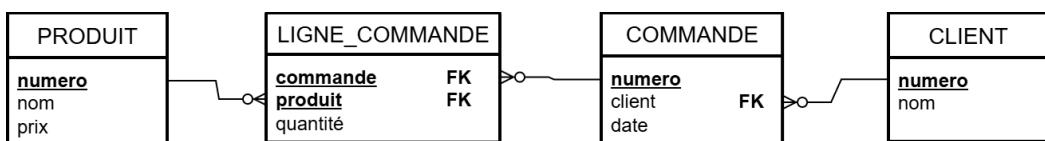
### Solution

Lors de la définition des tables en SQL, il faudra ajouter :

- D'après le MCD, une contrainte NOT NULL sur la colonne `id_film_vo` de la table `FILM_DOUBLE` pour traduire la cardinalité minimale 1 de l'association *traduction* du côté de *FILM DOUBLÉ*.
- D'après le MCD, une contrainte NOT NULL sur la colonne `langue` de la table `FILM_DOUBLE` pour traduire la cardinalité minimale 1 de l'association *dans la langue* du côté de *FILM DOUBLÉ*.

### Exercice 3

Voici le modèle relationnel des données d'un SI :



On connaît de plus les contraintes suivantes :

- les attributs nommés `numéro` sont des entiers.
- un `client` doit avoir un `nom`.
- un `produit` doit avoir un `prix` qui sera de type `money`.
- il n'y pas de `commande` sans numéro de client.
- la `date` d'une `commande` est de type `timestamptz`, elle ne peut pas être vide et elle est définie par défaut à l'heure de création de la commande.
- la `quantité` d'une `ligne de commande` est un entier, elle ne peut pas être vide.

**i Question**

Traduire en SQL ce modèle relationnel en codant toutes les informations connues.  
Pas de justifications demandées.

**Solution**

```
CREATE TABLE client(
    numero INTEGER,
    nom VARCHAR(50) NOT NULL,
    CONSTRAINT client_pk PRIMARY KEY (numero)
);

CREATE TABLE commande(
    numero INTEGER,
    client INTEGER NOT NULL,
    date TIMESTAMPTZ DEFAULT now(),
    CONSTRAINT commande_pk PRIMARY KEY (numero),
    CONSTRAINT commande_client_fk
        FOREIGN KEY (client)
        REFERENCES client(numero)
);

CREATE TABLE produit(
    numero INTEGER,
    nom VARCHAR(50),
    prix MONEY NOT NULL,
    CONSTRAINT produit_pk PRIMARY KEY (numero)
);

CREATE TABLE ligne_commande(
    commande INTEGER,
    produit INTEGER,
    quantite INTEGER NOT NULL,
    CONSTRAINT ligne_commande_pk PRIMARY KEY (commande,produit),
    CONSTRAINT ligne_commande_commande_fk
        FOREIGN KEY (commande)
        REFERENCES commande(numero),
    CONSTRAINT ligne_commande_produit_fk
        FOREIGN KEY (produit)
        REFERENCES produit(numero)
);
```