

# TD 8 : Modélisation

## Modélisation

2024-11-15

- **L3 MIASHS/Ingémath**
- **Université Paris Cité**
- Année 2024-2025
- [Course Homepage](#)
- [Moodle](#)



### ! Objectifs

L'objectif de cette séance est construire des modèles Entité-Association sur des problèmes miniatures.

## Modélisation Entité-Association (E/A ou E/R)

### Exercice (Supermarché)

#### i Question

Produire un schéma E/R qui décrit des informations concernant les produits d'un supermarché.

Chaque *produit* a un *nom* et un *prix* et appartient à une *catégorie*.

Le supermarché a plusieurs *rayons*, un rayon étant caractérisé par un *étage* et un *numéro de rangée*. On veut maintenir l'emplacement des produits dans les rayons. Les produits d'une même catégorie sont placés dans le même rayon, mais un rayon peut contenir des produits de plusieurs catégories.

#### i Question

Traduire le schéma EA dans le formalisme des pattes de corbeau

#### i Question

Définir le schéma relationnel correspondant en SQL

### Exercice (Location de voitures)

#### i Question

Produire un schéma E/R qui décrit des informations concernant des voitures à louer.

Chaque *voiture* a une plaque d'*immatriculation*, une *couleur* et une *marque*. Le *prix* de la location dépend de la *catégorie*, où chaque catégorie est identifiée par un nom.

**i Question**

Modifier ensuite le schéma pour représenter les modèles de voitures.

Un *modèle* a un *nom*, une *marque* et un *nombre de sièges*.

Toutes les voitures du même modèle doivent appartenir à la même catégorie de prix.

**i Question**

De plus, on veut distinguer les voitures *disponibles* des voitures en *location*. Pour les voitures disponibles on représente l'*emplacement*. Pour les voitures en location on représente la *date* et la *durée de la location*, ainsi que le *nom du client*.

**i Question**

Traduire le schéma EA dans le formalisme des pattes de corbeau

**i Question**

Définir le schéma relationnel correspondant en SQL

### Exercice (Gestion du personnel d'une entreprise)

Dans une entreprise, chaque *employé* (identifié par un *numéro*) est attaché à un *département* de l'entreprise. Il occupe un *bureau* et participe à un ou plusieurs *projets* développés par l'entreprise.

De chaque employé, on connaît : le *nom*, le *prénom*, les *emplois* qu'il a occupés à différentes dates et les *salaires* qu'il a *perçus* dans ces emplois.

Chaque *département* est identifié par un *numéro*, a son *budget* propre et est dirigé par un *directeur* faisant partie du personnel de l'entreprise.

Chaque *bureau* est identifié par un *numéro*, est rattaché à un *département* et est caractérisé par sa *surface* en mètres carrés. Il possède un numéro de *téléphone* associé.

Chaque *projet* est identifié par un *numéro*, possède un certain *budget* et *emploie* plusieurs *personnes* appartenant à différents *départements*. Chaque employé est *affecté* pour un *certain nombre d'heures* à un projet.

**i Question**

Donner un modèle entité-association correspondant à la description ci-dessus.

**i Question**

Modifier votre modélisation pour tenir compte de l'évolution dans le temps de la vie de l'entreprise : les projets ont des durées de vie limitées, chaque employé est affecté à un projet (mais aussi un département) pendant une certaine durée, etc

**i Question**

Traduire le schéma EA dans le formalisme des pattes de corbeau

**i Question**

Définir le schéma relationnel correspondant en SQL

### i Question

Essayer de coder les contraintes externes (exclusion, vérification, unicité, ...)

On s'intéresse d'abord aux contraintes qui pèsent sur les associations entre **employe** et **departement**.

- Un employé ne peut être membre de plusieurs départements simultanément
- Un département ne peut pas être dirigé simultanément par plusieurs employés
- Un employé ne peut pas diriger plusieurs départements simultanément

### i Question

Essayer de coder les contraintes externes (exclusion, vérification, unicité, ...)

On s'intéresse maintenant aux contraintes qui pèsent sur les associations entre **employe** et **projet**.

- Un employé ne peut participer à un projet que pendant la durée de vie du projet
- Un employé ne peut pas travailler plus de 50 heures par semaine

La première contrainte concerne deux tables **projet** et **participe** : il faut que l'intervalle spécifié par **debut\_date**, **fin\_date** dans **participe** soit inclus dans l'intervalle **debut\_date**, **fin\_date** de l'instance de **projet** désignée par **num\_projet**. Cette vérification devrait être effectuée lors des insertions/mises à jour dans **participe** mais aussi lors des mises à jour dans **projet**.

Pour mettre en place de genre de contraintes, SQL et PostgreSQL offre un cadre : celui des gachettes (TRIGGER). Cela va au delà de ce cours. Nous allons essayer de faire avec les moyens dont nous disposons : les fonctions SQL et les contraintes CHECK.

```
CREATE FUNCTION cho.chk_participation_in_project_range(  
    p_num_projet bigint,  
    p_debut_date date,  
    p_fin_date date  
)  
RETURNS integer  
LANGUAGE SQL AS  
$$  
SELECT  
    COUNT(*)  
FROM  
    cho.projet pr  
WHERE  
    p_num_projet=pr.num  
    AND  
    daterange(p_debut_date, p_fin_date) <@ daterange(pr.debut_date, pr.fin_date) ;  
$$ ;
```

```
ALTER TABLE cho.participe  
ADD CONSTRAINT cns_participe_2  
CHECK (  
    1 = cho.chk_participation_in_project_range(  
        num_projet,  
        debut_date,  
        fin_date  
    )  
)  
;
```

Il faudrait créer une fonction et une contrainte CHECK du côté **projet**.

Il faudrait aussi vérifier que la directrice d'un département est membre du département ...