

# Base de données

## Chapitre 4 : Modèle relationnel

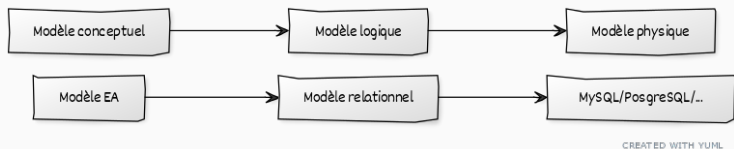
---

Joel Cavat

2022

## Objectifs

- Modélisation logique d'une BD relationnelle à partir du modèle EA



**Figure 1:** Choix du modèle logique

## Modèle conceptuel

- Le modèle entité-association (chapitre 2) est une **représentation graphique** qui permet de visualiser la conception d'une BD

## Modèle relationnel

- Le **modèle relationnel** est une représentation logique de la manière dont est organisée l'information sur le SGBD
  - il peut être représenté textuellement (schéma de relation - par. ex: `Person(login, nom, prenom)`)
  - ou graphiquement (diagramme relationnel)
- Il peut être obtenu
  - à partir du modèle EA en appliquant des **règles de transformations** (chapitre courant).
  - par *décomposition*: en appliquant les règles de normalisation (cf: cours alg. rel.)
  - par *synthèse*, en se basant sur les DF/DI (cf: cours alg. rel.)

# Du modèle conceptuel au modèle relationnel

## Application de règles de transformation

- Un type d'entité devient une **table** (ou schéma de relation)
- Un identifiant devient une **clé**
- Chaque type d'association possède ses propres règles de transformation
  - un-à-plusieurs
  - un-à-un
  - plusieurs-à-plusieurs

# Passage du modèle EA au modèle relationnel

Modèle EA	Modèle relationnel (courant)	Modèle relationnel (formel)
Attribut	Attribut	Attribut
Identifiant	Clé primaire	Clé primaire
Type d'entité	Table	Schéma de relation
Entité	Enregistrement	Tuple
-	Table + ensemble des enregistrements	Relation
Association	<i>représ. par clé étr.</i>	<i>représ. par une CI réf.</i>

# Type d'association un-à-plusieurs

## Variante 1 (standard)

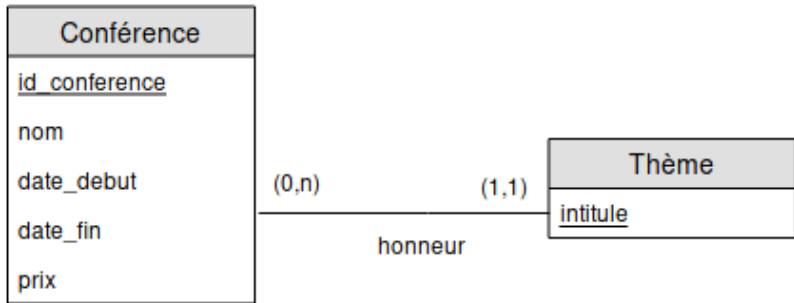
- Chaque type d'entité est transformé en une table
- Un nouveau champ est créé dans la table du côté "plusieurs"
- Ce champ est une **clé étrangère**
- Elle référence la clé primaire de l'autre table
- Si le type d'association a des attributs, ils deviennent des attributs de la table du côté "plusieurs"

Une autre variante à connaître en annexe (quiz slide 25)



# Type d'association un-à-plusieurs

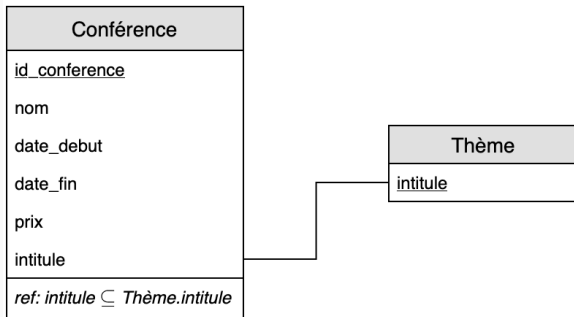
## Modèle EA



**Figure 2:** Association un-à-plusieurs

# Type d'association un-à-plusieurs

## Diagramme relationnel correspondants



## Schémas de relations correspondants

Conference(id\_conference, nom, date\_debut, date\_fin, prix, intitule)  
intitule  $\subseteq$  Theme.intitule  
Theme(intitule)

# Type d'association un-à-un

## Variante 1 (standard)

- Chaque type d'entité est transformé en une table
- Un nouveau champ (clé étrangère) est créé dans une des tables
  - du côté de la table "optionnelle" si existante
  - arbitrairement sinon
- La valeur de cette clé étrangère **doit être unique**
- Meilleure séparation des préoccupations mais jointure nécessaire

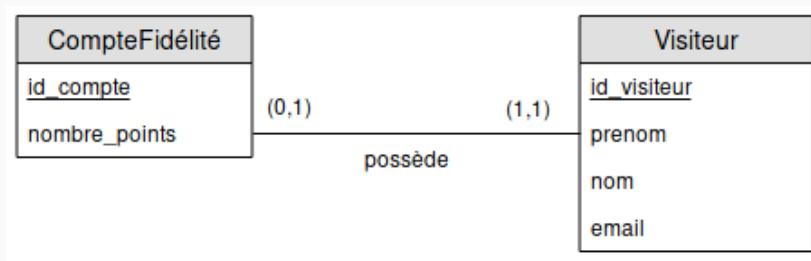
## Variante 2

- Les deux entités sont fusionnées et forment une seule table
- Sélection d'une seule clé primaire
- Simple et efficace, mais implique des champs optionnels (null)

Deux autres variantes à connaître en annexe

# Type d'association un-à-un

## Modèle EA



**Figure 3:** Association un-à-un

# Type d'association un-à-un

## Schémas de relations associés

### Variante 1

```
Visiteur(id_visiteur, prenom, nom, email)  
CompteFidelite(id_compte, nombre_points, id_visiteur)  
    id_visiteur  $\subseteq$  Visiteur.id_visiteur  
    id_visiteur UNIQUE
```

### Variante 2

```
Visiteur(id_visiteur, prenom, nom, email, id_compte, nombre_points)  
    nombre_points OPTIONNEL  
    id_compte OPTIONNEL, UNIQUE
```

# Type d'association un-à-un

## Diagrammes relationnels correspondants

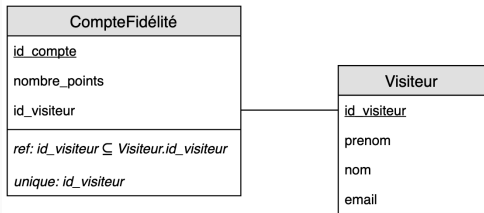


Figure 4: variante 1

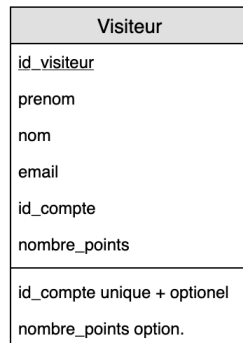


Figure 5: variante 2

### Quiz

Pourquoi la clé étrangère doit-elle être unique ?

## Quiz

- Quelle variante correspond à un type d'association  $(1,1) \cdots (1,1)$  ?
- Quelle variante correspond à un type d'association  $(0,1) \cdots (1,1)$  ?

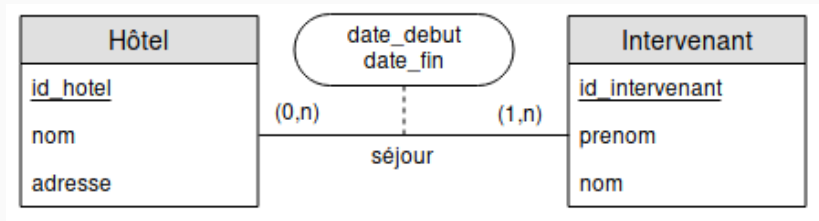


# Type d'association plusieurs-à-plusieurs

- Chaque type d'entité est transformé en une table
- Le type d'association devient une table (appelé parfois table de correspondance)
- Les attributs de l'association se retrouvent dans cette table
- Cette table enregistre les clés étrangères des deux tables associées
- Le choix de la clé primaire pour cette table définit une contrainte (correspond à une DF)
  - Cas **Permissif**: Une clé artificielle est créée
  - Cas **Restrictif**: Les clés étrangères deviennent la clé primaire
  - Cas **Personnalisé**: modélisation d'autres contraintes ; ajout d'un attribut dans la clé par exemple

# Type d'association plusieurs-à-plusieurs

## Modèle EA



**Figure 6:** Association plusieurs-à-plusieurs

# Type d'association plusieurs-à-plusieurs

## Cas restrictif : clés étrangères comme clés primaires

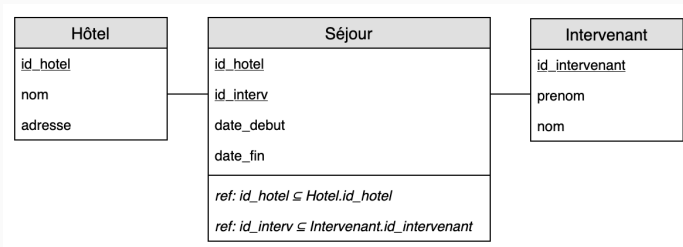


Figure 7: diagramme relationnel

## Schémas de relations associés

```
Hotel(id_hotel, nom, adresse)
Sejour(id_interv, id_hotel, date_debut, date_fin)
    id_hotel  $\subseteq$  Hotel.id_hotel
    id_interv  $\subseteq$  Intervenant.id_intervenant
Intervenant(id_intervenant, prenom, nom)
```

# Type d'association plusieurs-à-plusieurs

## Cas permissif : clé artificielle

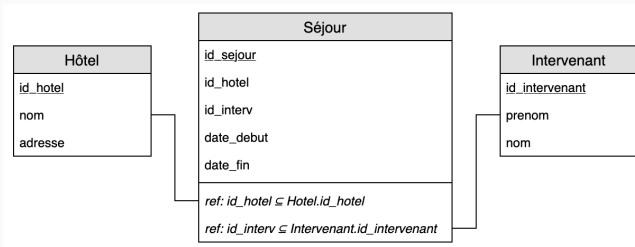


Figure 8: Diagramme relationnel

## Schémas de relations associés

```
Hotel(id_hotel, nom, adresse)
Sejour(id_sejour, id_interv, id_hotel, date_debut, date_fin)
    id_hotel ⊆ Hotel.id_hotel
    id_interv ⊆ Intervenant.id_intervenant
Intervenant(id_intervenant, prenom, nom)
```

# Type d'association plusieurs-à-plusieurs

## Personnalisé 1 : ajout d'un attribut à la clé

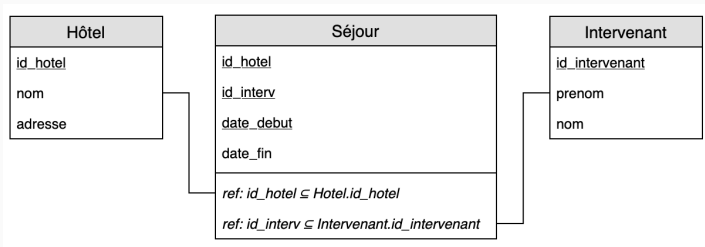


Figure 9: Diagramme relationnel

## Schémas de relations associés

```
Hotel(id_hotel, nom, adresse)
Sejour(id_interv, id_hotel, date_debut, date_fin)
    id_hotel  $\subseteq$  Hotel.id_hotel
    id_interv  $\subseteq$  Intervenant.id_intervenant
Intervenant(id_intervenant, prenom, nom)
```

# Type d'association plusieurs-à-plusieurs

## Personnalisé 2 : clé artificielle avec contraintes d'unicité

- Possible de restreindre d'avantage en appliquant des contraintes d'unicité sur un ensemble d'attributs (en plus d'une clé primaire).
- Interprété comme une deuxième clé

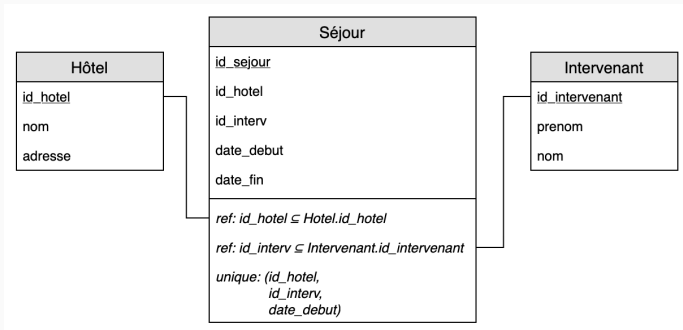


Figure 10: Diagramme relationnel

### Quiz - Personnalisé 3

Trouvez une clé pour la table `Sejour` qui permette d'appliquer la contrainte suivante:

- Un intervenant ne peut se voir attribuer qu'un séjour à une date donnée (`date_debut`). Ne doit pas gérer les chevauchements de dates.

Les contraintes non modélisables (statiques ou dynamiques) peuvent être traitées à l'aide de déclencheurs (**triggers**).



### Quiz

Maintenant que nous avons vu la transformation pour les types d'associations plusieurs-à-plusieurs, nous pouvons imaginer une transformation similaire pour les types d'associations un-à-un et un-à-plusieurs:

- Comment pourrait-on représenter un type d'association  $(0,1) \text{---} (0,1)$  à l'aide d'une table de correspondance ?
- Comment pourrait-on représenter un type d'association  $(0,1) \text{---} (0,n)$  à l'aide d'une table de correspondance ?

## Exemple 1

Un même numéro de salle peut se trouver dans différents cinémas.

## Modèle EA



**Figure 11:** Entité faible

## Schéma relationnel associé

Cinema(nom, adresse)

Salle(nom, no, nombre\_places), nom  $\subseteq$  Cinema.nom

# Entités faibles

## Exemple 2

Un dossier patient n'est identifiable que par le no avs du patient associé.

## Modèle EA

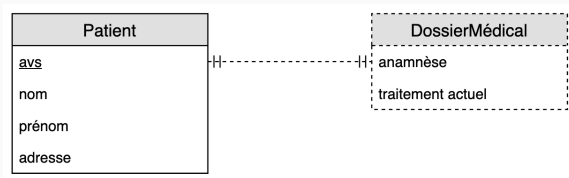


Figure 12: Entité faible

## Schéma relationnel associé

Patient(avs, nom, prenom, adresse)

DossierMedical(avs, anamnese, traitement\_actuel), avs  $\subseteq$  Patient.avs

## Transformation finale

### Exercice

- A partir du modèle EA du chapitre 2, réalisez le modèle relationnel
  - transformez au préalable le compte fidélité en un type d'entité faible
  - représentez le modèle à l'aide d'un diagramme relationnel (illustré)
  - décrivez les CI non modélisables
  - décrivez les propriétés des attributs particuliers
    - clé, unique, optionnel. . .

# EA -> schémas relationnels

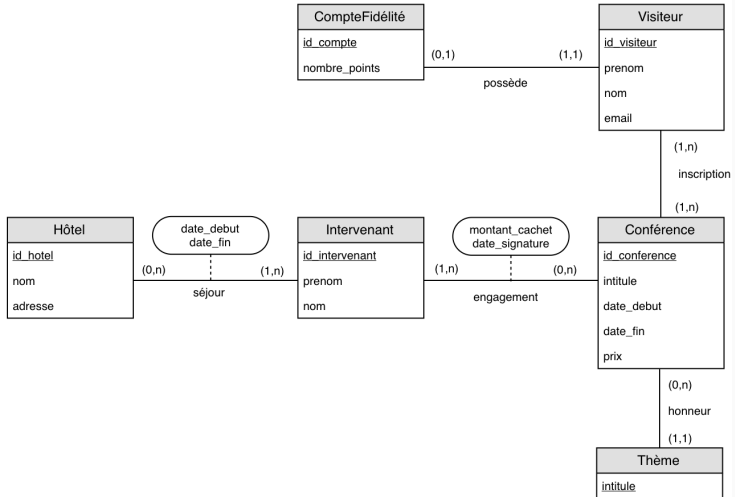


Figure 13: Schéma EA