

# Base de données

---

## Chapitre 3 : Le modèle relationnel

Stéphane Malandain – sbd – 2024

# Introduction aux bases de données relationnelles

---

# Les bases de données relationnelles

---

Première réalisation commerciale d'une bd relationnelle dans les années 80

**Les + populaires :**

- Open source

- MySql
- PostgreSQL
- MariaDB
- SQLite
- Firebird

- Commerciales

- Oracle Database
- DB2 d'IBM
- Microsoft SQL Server
- Microsoft Access

# Les bases de données relationnelles

---

## Base de données relationnelle

Une base de données relationnelle permet de regrouper des données structurées dans un ensemble de **tables** organisées en **lignes** et en **colonnes**. Elle permet de renforcer les contraintes d'intégrité.

### Remarque :

- Le terme relation **ne décrit pas** les associations entre les tables.
- Une **relation** correspond à une **table** avec ses **enregistrements**.
- le terme de relation provient du formalisme mathématique décrit par Codd dans sa formulation du modèle relationnel.
- Des lors, on va préférer le terme d'association (relationship) pour représenter les liens entre les tables.
- Une table est représentée par **un type d'entité ou un type d'association** du modèle EA.

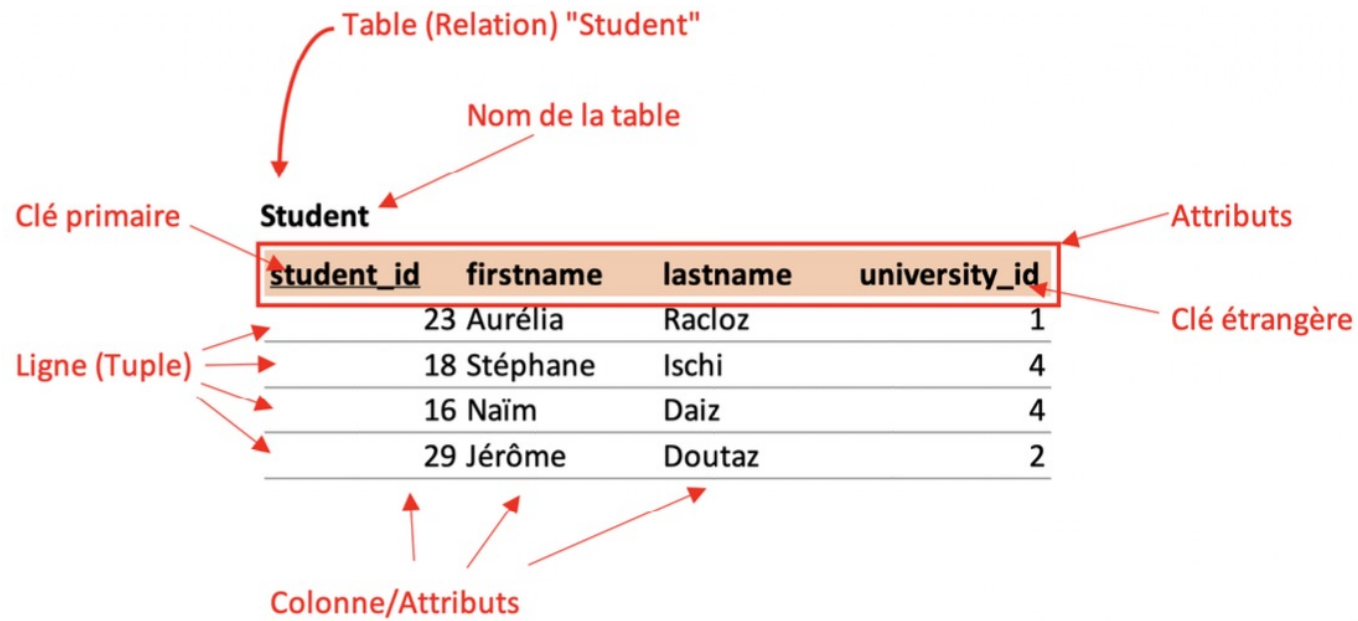
# Les bases de données relationnelles

---

## Caractéristiques :

- Structurée
- **Normalisée**
- Centralisée / Scalabilité verticale
- Cohérence très forte
  - modélisation des CI
  - Transactions (propriétés ACID)
  - Triggers (renforcement des CI)

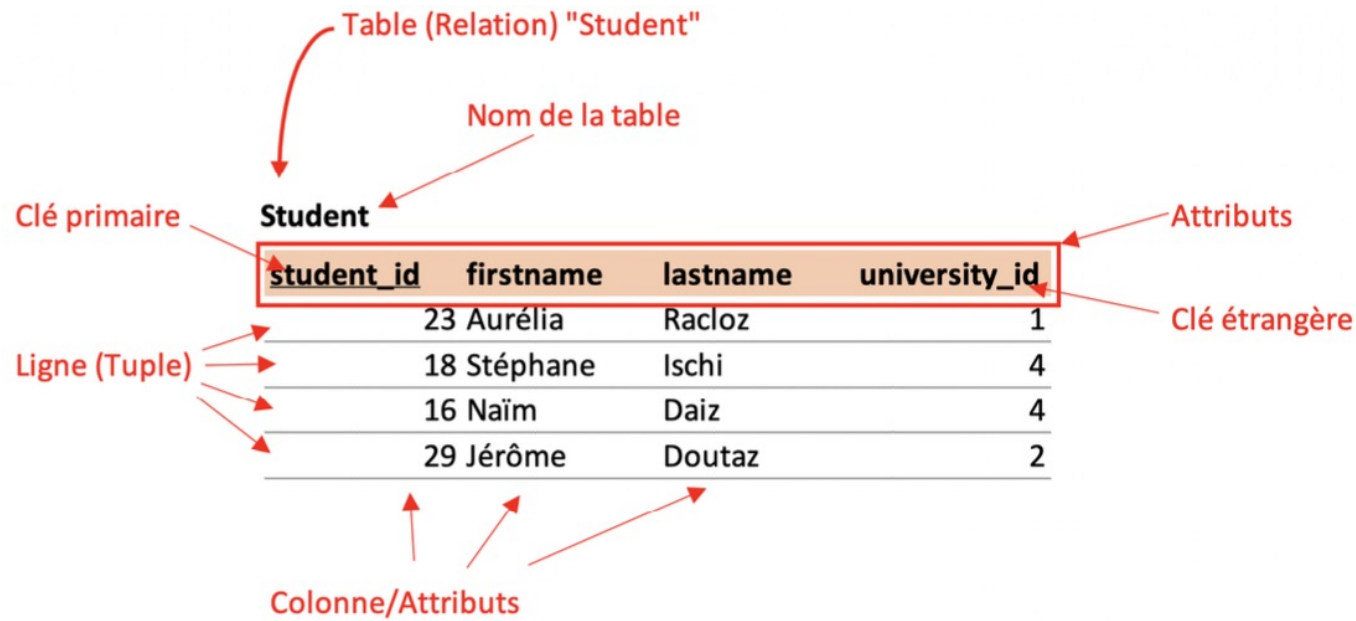
# Structurée



## Terminologie de base

- Une **table** est un ensemble de données organisées sous forme de lignes et de colonnes
- Une **ligne** correspond à un enregistrement, à une entité ou à un tuple.
- Une **colonne** correspond à une propriété d'une entité. Elle possède un type.

# Structurée



## Terminologie de base

- Une **base de données relationnelle** est un ensemble de tables et de contraintes d'intégrité.
- Chaque enregistrement possède un identifiant appelé **clé primaire**. Une telle clé doit être unique.
- Une **clé étrangère** est une colonne utilisée pour référencer une clé (candidate) d'une entité d'une autre table.

# Normalisation

---

## Définition

Une base de données **normalisée** est modélisée de manière à éviter des redondances, des incohérences et des anomalies.

La normalisation repose sur un ensemble de règles à respecter : **les formes normales**.

Grâce à la normalisation, le modèle relationnel permet d'éviter :

- les redondances
- les anomalies de mise-à-jour
- les anomalies d'insertions



# Redondances et anomalies

## Solutions proposées par le modèle relationnel

- Séparer les entités (séparer les concepts)

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique



NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique

# Redondances et anomalies

## Solutions proposées par le modèle relationnel

- Séparer les entités : Attention à la perte des associations

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique



NoEtu	Prénom	Nom
1	Bill	Torvalds
2	Linus	Gates
3	Mark	Jobs
4	Steve	Zuckerberg

ISBN	Titre
2-234432-54-3	Les Aqueducs
1-657879-34-6	Python pour les nuls
4-493828-43-2	Base de données avancée
8-098384-98-7	Algèbre et géométrie

NoThm	Thème
1	Architecture
2	Informatique
3	Mathématique

Echéance
18.06.1999
23.04.2012
19.07.2013
01.05.2011
05.12.2005
06.08.2013

# Redondances et anomalies

## Solutions proposées par le modèle relationnel

- Séparer les entités
- Représenter les associations en utilisant des attributs communs à chaque table associée.

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique



NoEtu	Prénom	Nom
1	Bill	Torvalds
2	Linus	Gates
3	Mark	Jobs
4	Steve	Zuckerberg

ISBN	Titre	NoThm
2-234432-54-3	Les Aqueducs	1
1-657879-34-6	Python pour les nuls	2
4-493828-43-2	Base de données avancée	2
8-098384-98-7	Algèbre et géométrie	3

NoThm	Thème
1	Architecture
2	Informatique
3	Mathématique

Echéance	NoEtu	ISBN
18.06.1999	1	2-234432-54-3
23.04.2012	2	2-234432-54-3
19.07.2013	2	1-657879-34-6
01.05.2011	3	4-493828-43-2
05.12.2005	4	2-234432-54-3
06.08.2013	4	4-493828-43-2

## Contraintes d'intégrité

---

Les contraintes d'intégrité décrivent des règles métiers à respecter. Elles décrivent des situations (ou comportements) admissibles ou non admissibles pour le domaine d'application.

Elles permettent de renforcer la cohérence des données.

Il existe trois moyens de renforcer la cohérence des données

- La modélisation logique
- Les déclencheurs (*triggers*)
- Les transactions

# Classement des contraintes d'intégrité

---

## CI statiques

- décrivent des **états valides**
- modélisables
  1. une personne doit avoir un nom de famille
  2. une conférence doit avoir un thème
- non modélisables
  3. le prix d'une conférence doit être supérieur à 0
  4. les intervenants pour les conférences "blockchain" logent dans des hôtels quatre ou cinq étoiles
  5. les thèmes sont obligatoirement associés à une conférence
  6. les conférences n'ont lieu que dans des capitales européennes

# Classement des contraintes d'intégrité

## CI dynamiques

- décrivent des **changements d'états valides**
- non modélisables

7. Pour un intervenant, il est possible de changer l'attribution d'un hôtel vers un autre de même classe ou de classe supérieure

8. Il est impossible d'augmenter le prix d'une conférence de plus de 10%

9. L'état civil d'une personne peut passer de "célibataire" à "marié" mais pas aux statuts "veuf" ou "divorcé"

	Dynamique	Statique
Modélisable	-	1, 2
Non modélisable	7, 8, 9	3, 4, 5, 6

**Toutes les contraintes non modélisables doivent être décrites en annexe du modèle**

# Les transactions

---

## Transaction

Une transaction est une suite d'opérations réalisées de manière indivisible (c.f. cours de concurrence).

Propriétés d'une transaction : L'acronyme ACID

- Atomicité
- Cohérence
- Isolation
- Durabilité

# Les transactions

---

## Caractéristiques ACID

Ce sont les quatre propriétés d'une transaction

- Atomicité
  - suite d'opérations indivisibles
  - l'opération est faite dans son intégralité ou elle est annulée (commit/rollback)
  - transaction indivisible
- Cohérence
  - la base de données est toujours dans un état cohérent
  - une transaction fait passer d'un état valide à un autre état valide



# Les transactions

---

## Caractéristiques ACID

Ce sont les quatre propriétés d'une transaction

- Isolation

- chaque transaction est indépendante
- une transaction n'a pas accès aux données d'une autre transaction en cours
- une transaction verrouille les ressources

- Durabilité

- une fois qu'une transaction est terminée, elle est persistée même en cas de crash du système
- stockage sur disque non volatile
- résultat d'une transaction persistée à long terme

# Les déclencheurs (triggers)

---

## Trigger

Un **trigger** est une vérification qui est activée lors d'un évènement particulier (insertion, mise à jour, ...). Ils permettent de vérifier des contraintes non modélisables

```
LORS DE <evenement>  
SI <condition>  
ALORS >action/exception>
```

## Le modèle relationnel

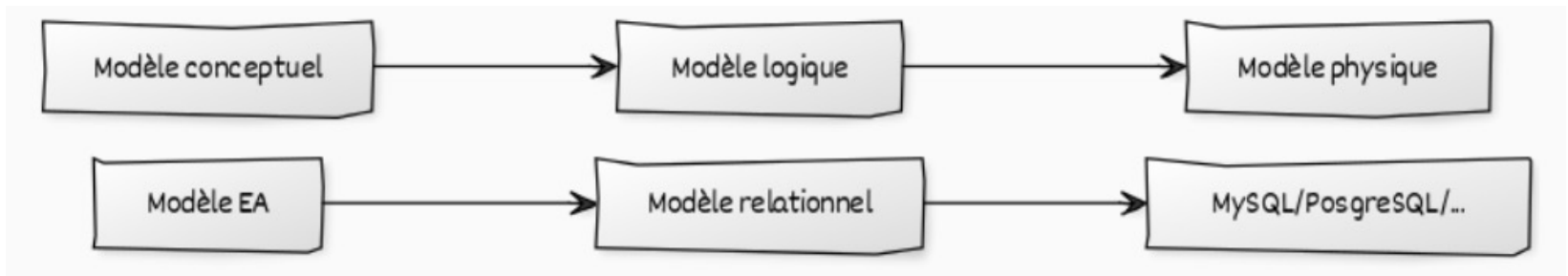
---

# introduction

---

## Objectifs :

- Modélisation logique d'une base de données relationnelle à partir du modèle EA



Choix du modèle logique

# Le modèle relationnel

---

## Le modèle conceptuel

Le modèle Entité-Association est une **représentation graphique** qui permet de visualiser la conception d'une base de données

## Le 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**.
- 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 EA vers le modèle relationnel**

---

# Passage du modèle EA 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 (formel).	Modèle relationnel (courant)
Attribut	Attribut	Attribut
Identifiant	Clé primaire	Clé primaire
Type d'entité	Schéma de relation	Table
Entité	Tuple	Enregistrement
-	Relation	Ens. des enregistrements
Association	représenté par CI réf.	représenté par une clé étrangère



# Type d'association un-à-plusieurs

---

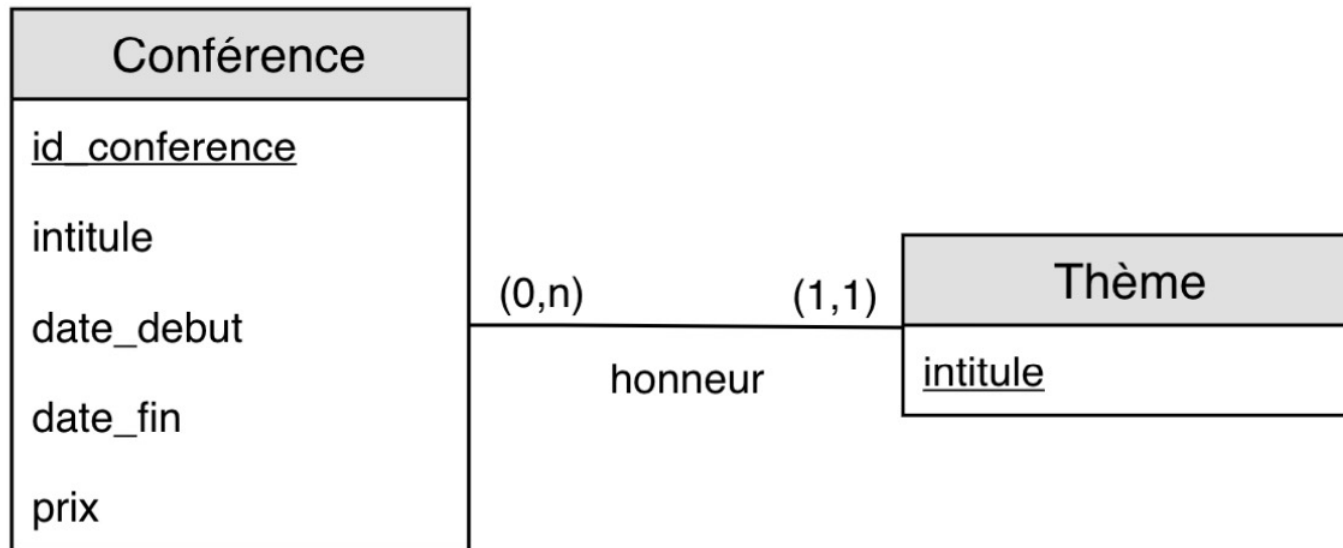
## Variante 1 (standard)

- Chaque type d'entité est transformée 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 est à connaître (quiz slide 25)

## Type d'association un-à-plusieurs

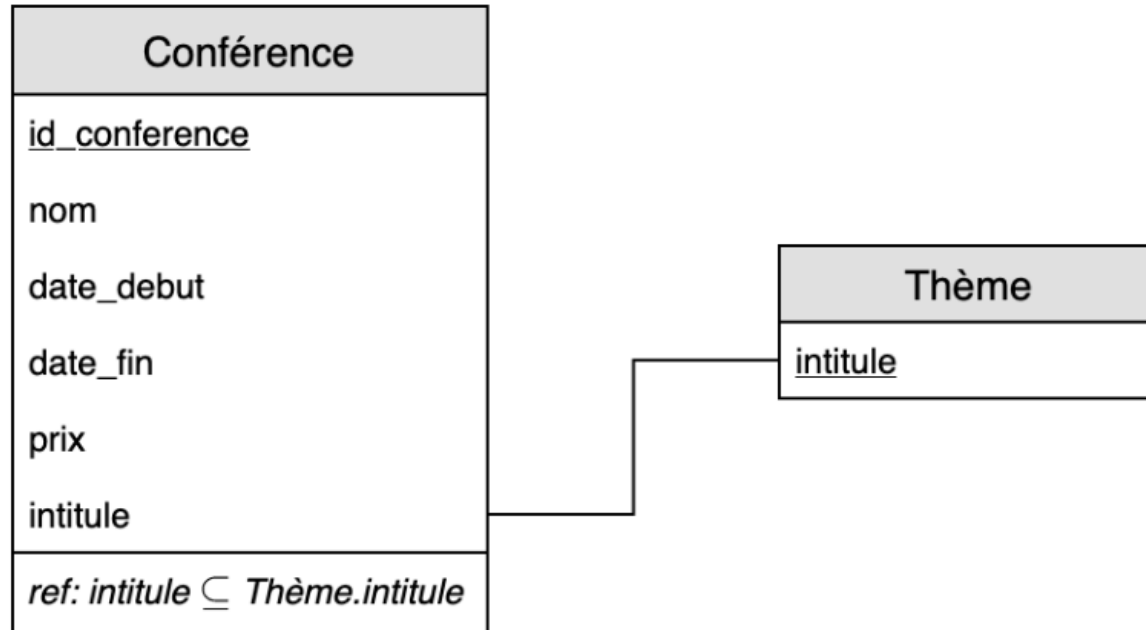
### Modèle EA



Association un-à-plusieurs

## Type d'association un-à-plusieurs

### Diagramme relationnel correspondant :



### Schémas de relations correspondants :

```

Conference(id_conference, nom, date_debut, date_fin, prix, intitule)
  intitule ⊆ 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

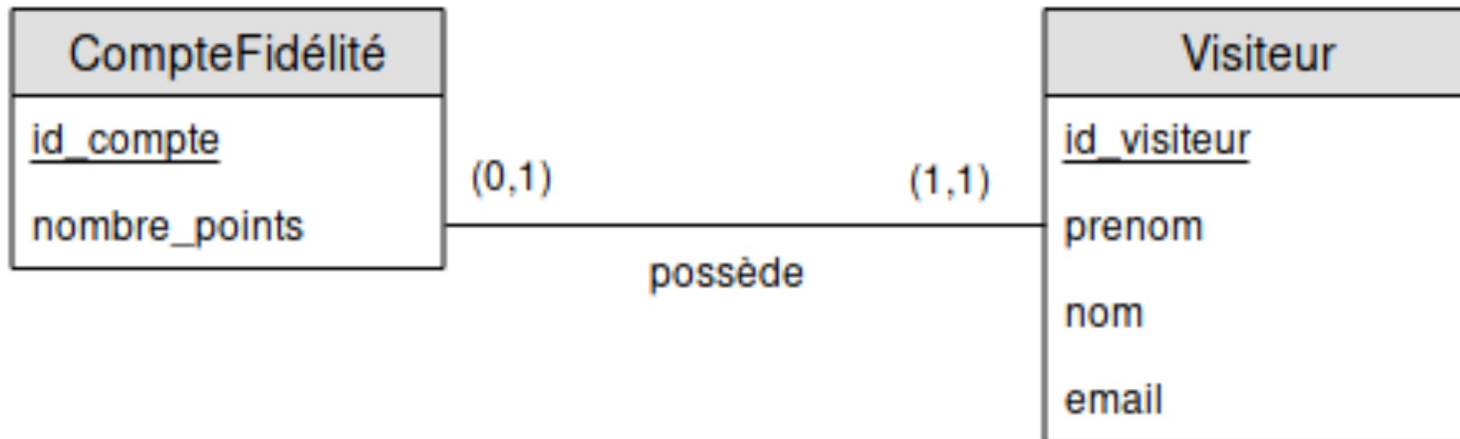
## Variante 2

- 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)

Une autre variante est à connaître (quiz slide 41)

## Type d'association un-à-un

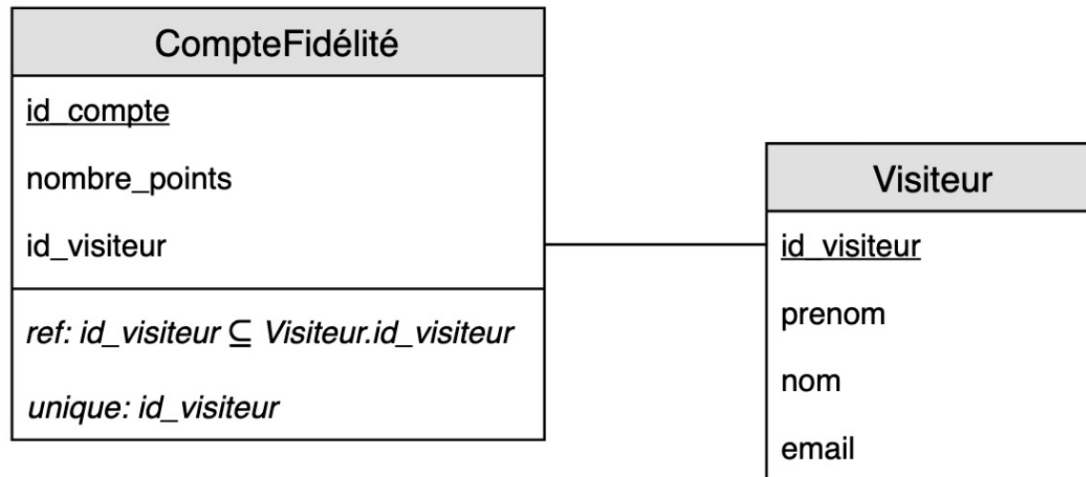
### Modèle EA



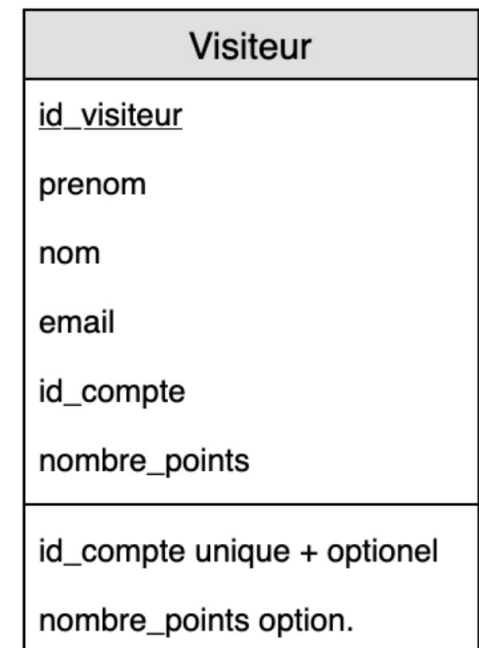
Association un-à-un

## Type d'association un-à-un

Diagramme relationnel correspondant :



Variante 1



Variante 2

## Type d'association un-à-un

### Schémas de relations correspondants :

#### 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

---

### Quizz :

1. Pourquoi la clé étrangère doit-elle être unique ?
2. Quelle variante correspond à un type d'association (1,1) --- (1,1) ?



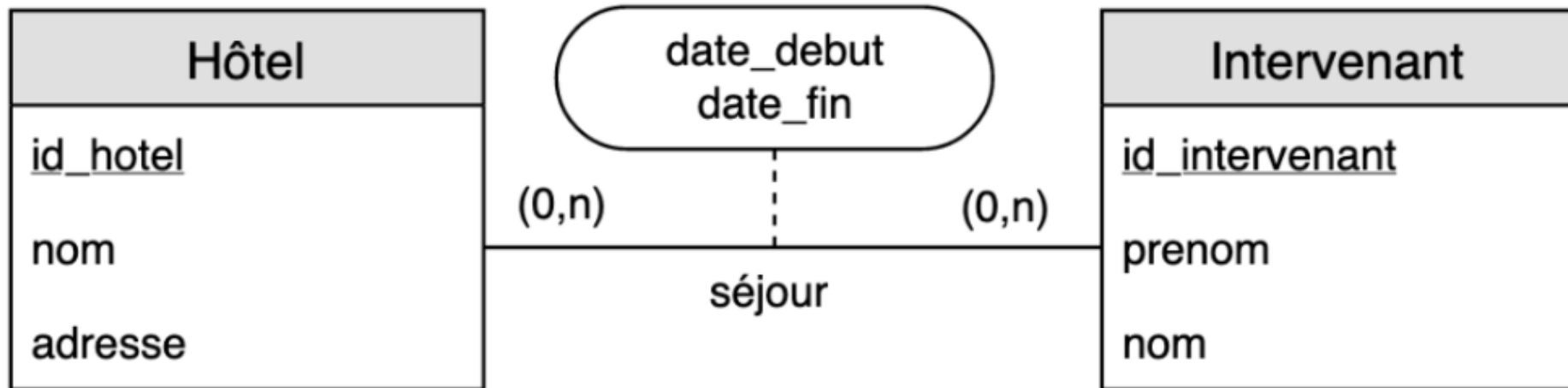
## Type d'association plusieurs-à-plusieurs

---

- Chaque type d'entité est transformé en une table
- Le type d'association devient une table (appelée 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



Association plusieurs-à-plusieurs

# Type d'association plusieurs-à-plusieurs

**Cas restrictif** : clés étrangères comme clés primaires

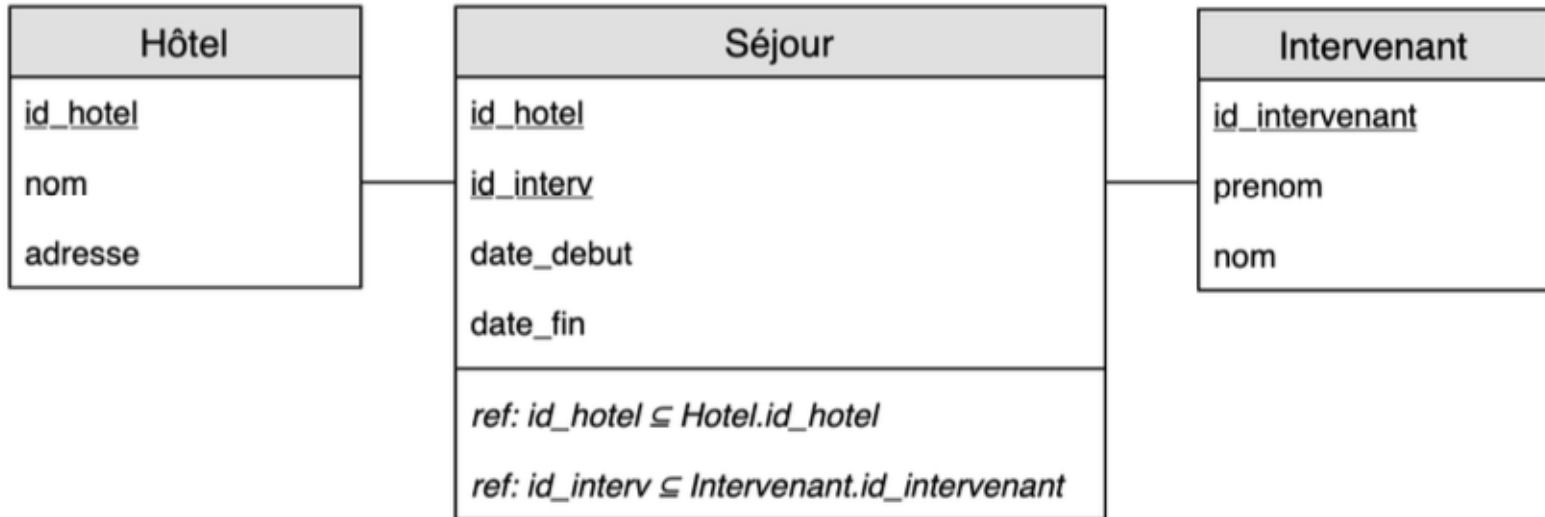


Diagramme relationnel

**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

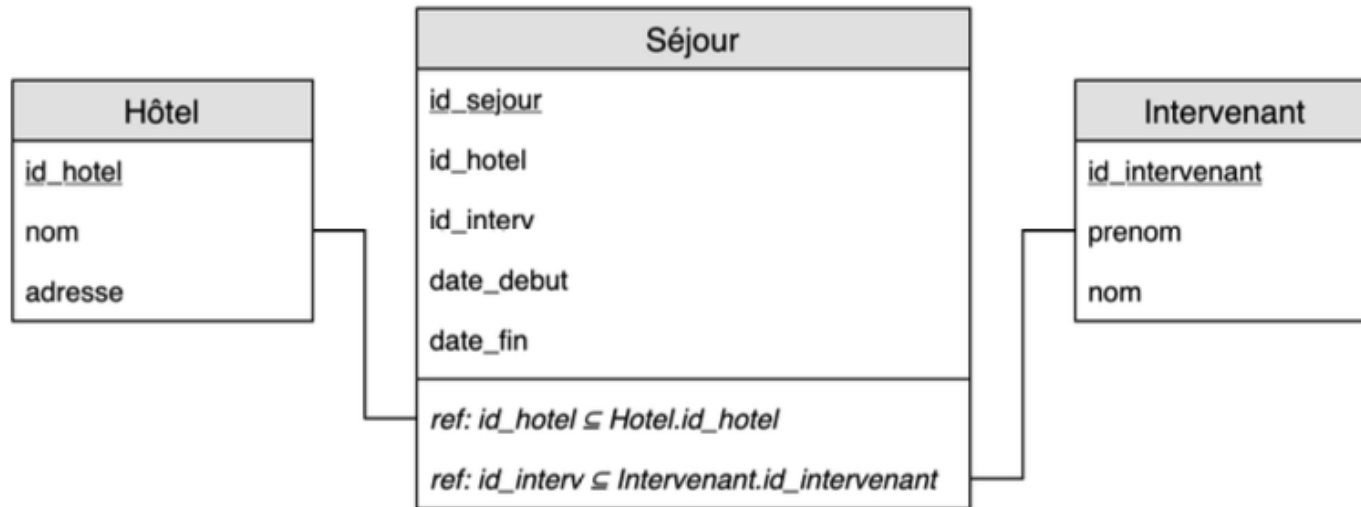


Diagramme relationnel

## 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

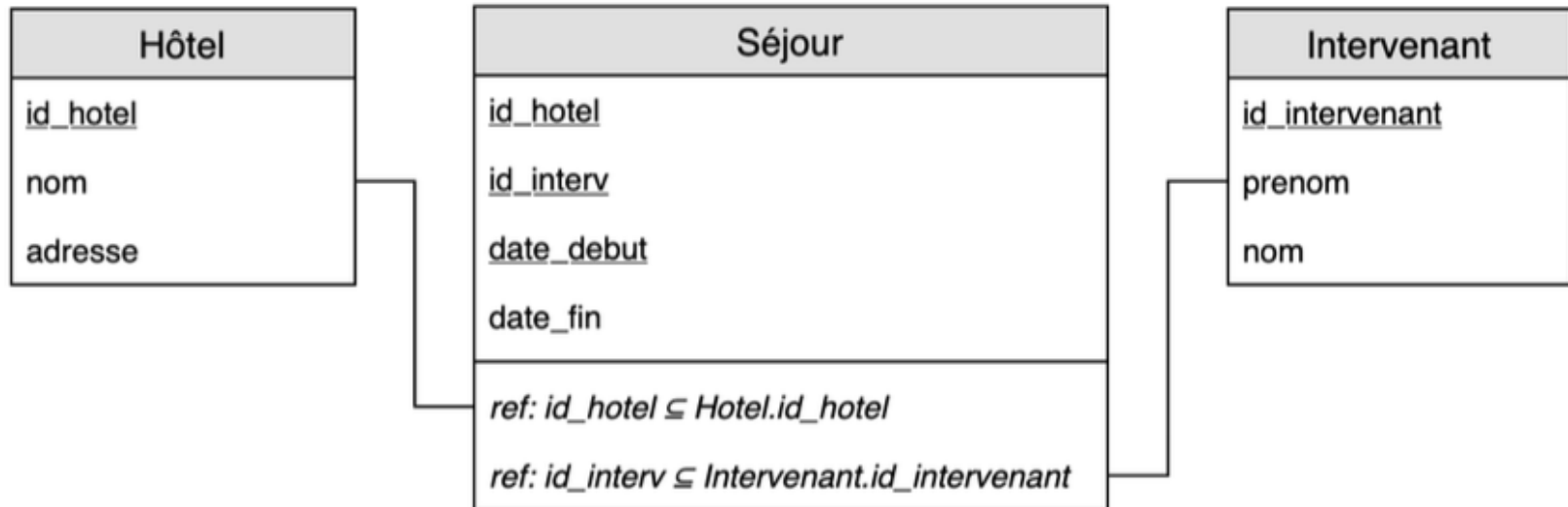


Diagramme relationnel

## 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 contrainte d'unicité

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

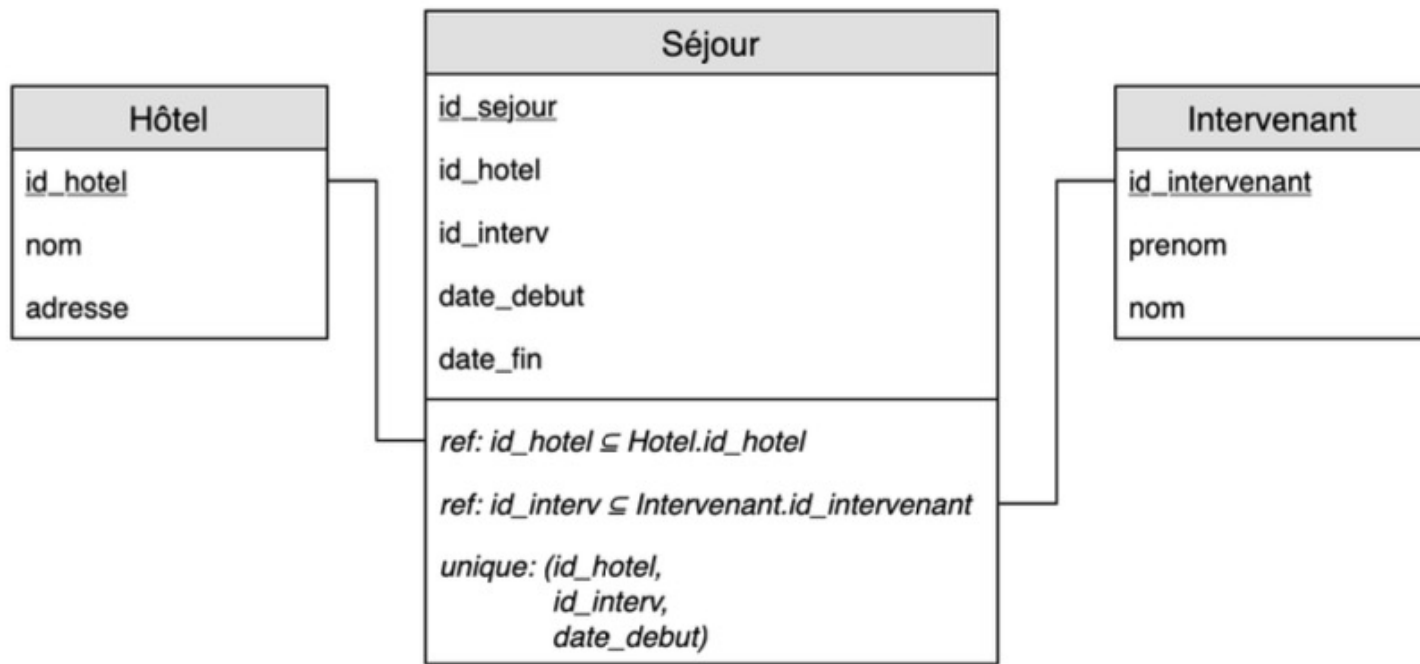


Diagramme relationnel

# Type d'association plusieurs-à-plusieurs

---

## Personnalisé 3 : Quizz

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.

## Remarque

---

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



## Variantes 1-1, 1-n

---

### 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 ?

## Entités faibles

### Exemple 1

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

### Modèle EA



Entité faible

### Schéma relationnel associé

Cinema(nom, adresse)

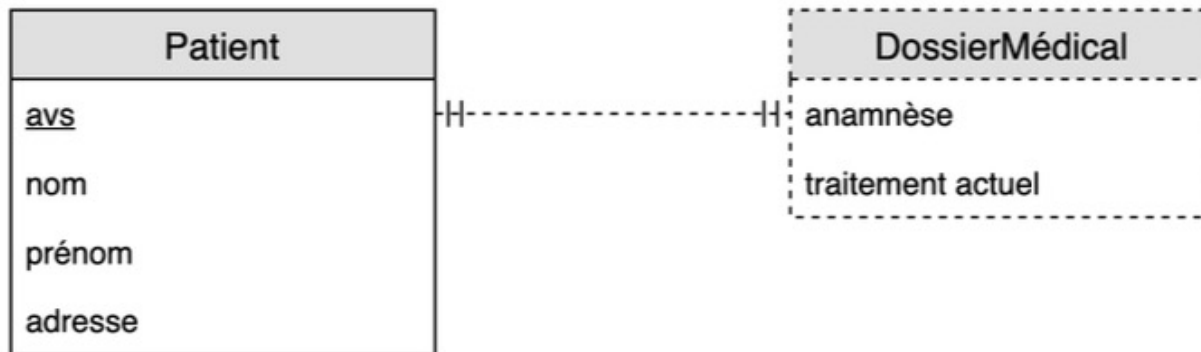
Salle(nom, no, nombre\_places), nom  $\subseteq$  Cinema.nom

## Entités faibles

### Exemple 2

Un dossier patient n'est identifiable que le n°avs du patient associé.

### Modèle EA



Entité faible

### Schéma relationnel associé

Patient(avs, nom, prenom, adresse)

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

## Transformation finale

---

## EA -> schémas relationnels

---

### 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

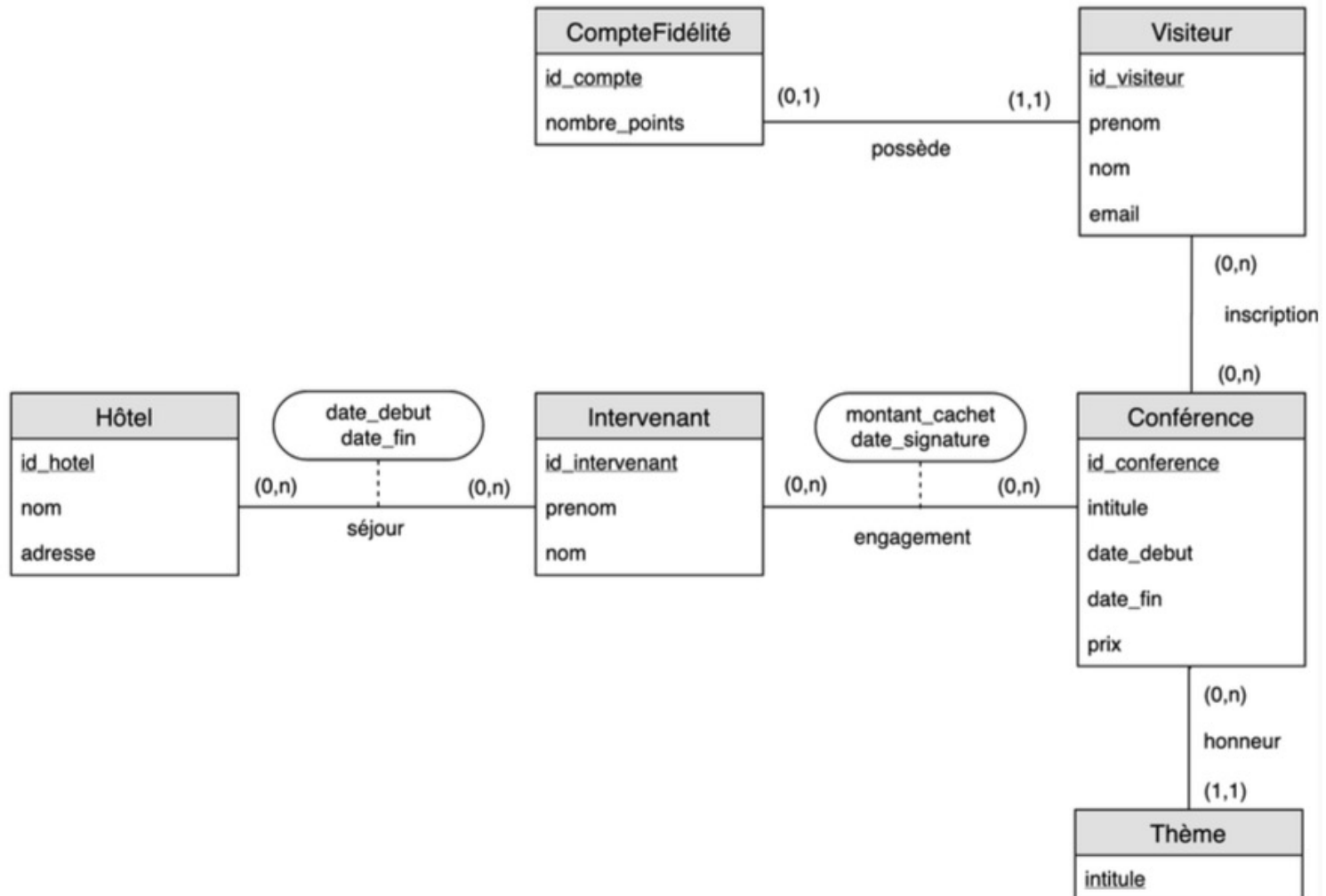


Schéma EA