SQL et les bases de données relationnelles

Modèle Entité-Association

Guillaume Raschia — Nantes Université

originaux de Philippe Rigaux, CNAM

Dernière mise-à-jour : 4 janvier 2023

Plan de la session

Éléments constitutifs (S6.2)

Concepts avancés (S6.3)

Vers le schéma relationnel (S6.4)

Rétro-ingénierie (S6.5)

Éléments constitutifs (S6.2)

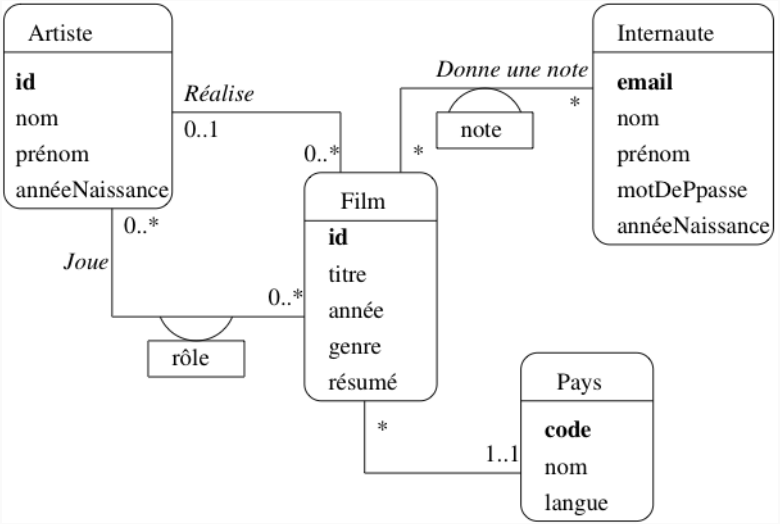
Le modèle Entité-Association

Cette section présente les notions de base du modèle Entité-Association

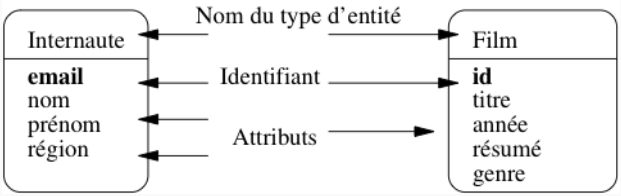
- Présentation commentée du diagramme de la base Films
- Entités
- Association un-à-plusieurs
- Association plusieurs-à-plusieurs
- Réification d'une association

Les conventions graphiques retenues ici sont plus proches du modèle de classe UML que du modèle E-A. Cette liberté permet (i) de respecter les choix de Ph. Rigaux et (ii) une représentation plus compacte, sans pour autant réduire l'expressivité du langage graphique.

Diagramme E-A commenté : la base des films



Les types d'entité

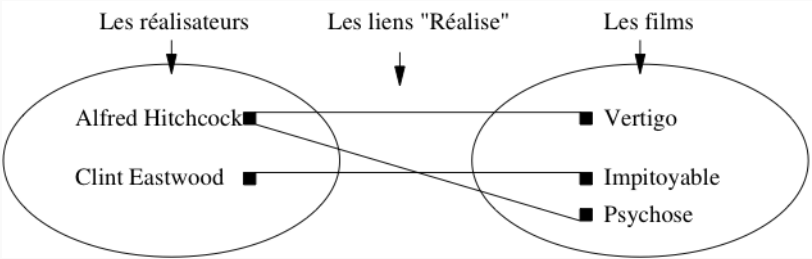


- Tous les attributs sont atomiques
- Clé obligatoire : un attribut non-descriptif `id` peut être ajouté
- Tous les attributs dépendent—fonctionnellement—directement [1] de la clé.

1. DF $K \rightarrow A$ sans transitivité.

Association binaire

Étudions l'association « un artiste réalise un film », avec quelques exemples.



Décisions à prendre :

- certains réalisateurs dirigent *plusieurs* films ;
- inversement, un film est dirigé par *au plus un* réalisateur.

Ces choix sont essentiels (et difficiles à changer après coup).

Cardinalités

On représente les choix de conception sur les associations par des cardinalités.

La cardinalité de l'association pour un type d'entité E est une paire $[\min, \max]$:

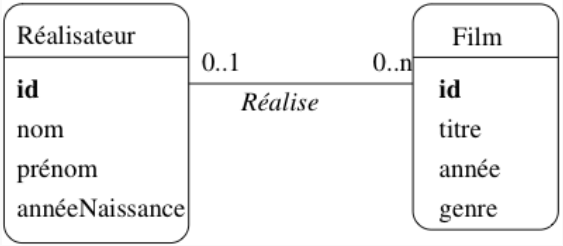
- $\max$ (cardinalité maximale) désigne le nombre maximal de fois où une entité e de E peut intervenir dans l'association : 1 ou * (ou n), nombre indéterminé.
- $\min$ (cardinalité minimale) désigne le nombre minimal de fois où une entité e de E peut intervenir dans l'association : 0 ou 1.

Nommage un/plusieurs-à-un/plusieurs

Exprime les cardinalités maximales de chaque côté.

Association un-à-plusieurs (ou plusieurs-à-un)

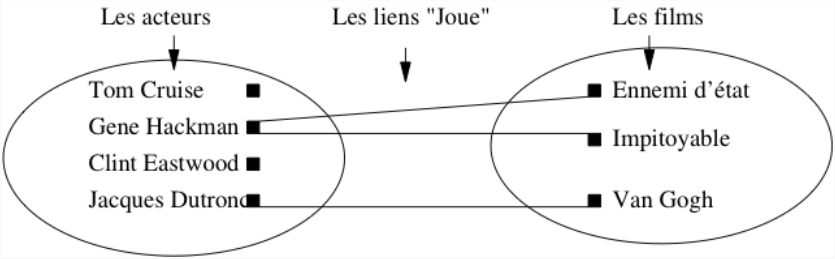
Représentation de l'association « un artiste réalise un film ».



- On interdit donc qu'un film ait deux réalisateurs.
- On autorise qu'un film n'ait pas de réalisateur (ou qu'on ne le connaisse pas).

Autre exemple

Étudions l'association « un artiste joue un rôle dans un film », sur un exemple.

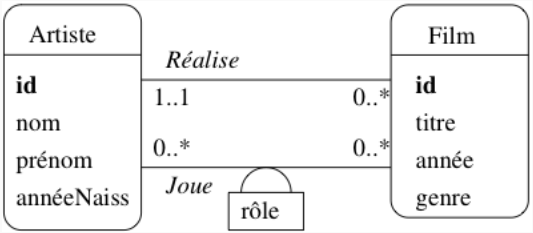


Décisions à prendre :

- certains acteurs jouent dans plusieurs films ;
- la distribution d'un film peut comporter plusieurs acteurs.

Association plusieurs-à-plusieurs

Ajout de l'association « un artiste joue dans un film ».

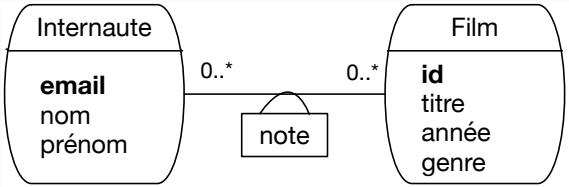


Une association est identifiée par les deux entités qu'elle lie, et donc par la paire des identifiants (idFilm, idArtiste).

De ce fait, un même acteur joue au plus un rôle dans un même film.

Autre exemple

Ajout de l'association « un internaute évalue un film ».



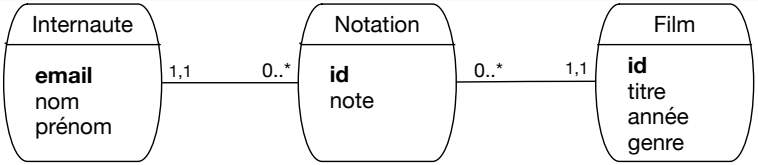
Un même internaute évalue un même film une seule fois au plus.

Rôles

Il est possible de spécifier une étiquette (un **rôle**) pour chaque côté de l'association.

Réification des associations plusieurs-à-plusieurs

On peut remplacer une association plusieurs-à-plusieurs par une entité et des associations un-à-plusieurs.



Souvent un bon choix : il est plus facile d'exprimer des contraintes sur une entité que sur une association.

À retenir

Le modèle Entité / Association

- Entité : objets autonomes, identifiables, persistants.
- Association : lien entre entités, caractérisée par des cardinalités et éventuellement des rôles.
- On peut toujours se ramener à des associations un-à-plusieurs.

Ne représente pas la réalité fidèlement ou complètement mais en fonction d'un besoin.

Concepts avancés (S6.3)

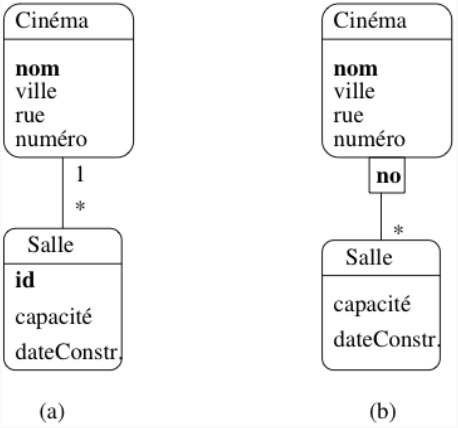
Le modèle E-A : concepts avancés

Cette section présente quelques notions avancées du modèle Entité-Association

- Entités faibles
- Associations généralisées
- Spécialisation orientée-objet

Entités faibles

Identification d'une entité relativement à une autre entité (toujours un-à-plusieurs).



Entités faibles : caractéristiques

La clé de `Salle` est la paire `(idCinéma, noSalle)`

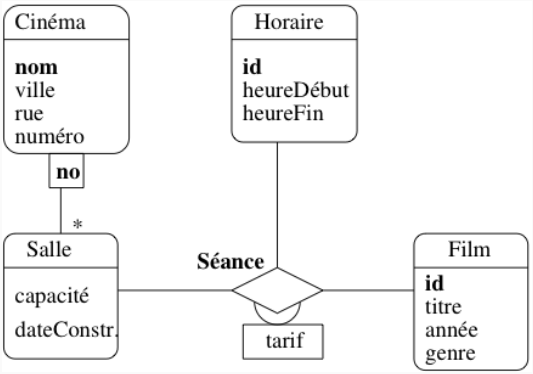
- quand on insère une salle dans la base, on doit toujours l'associer à un cinéma ;
- quand un cinéma est détruit, on doit aussi détruire toutes ses salles.

On peut très bien utiliser une association classique.

Associations généralisées

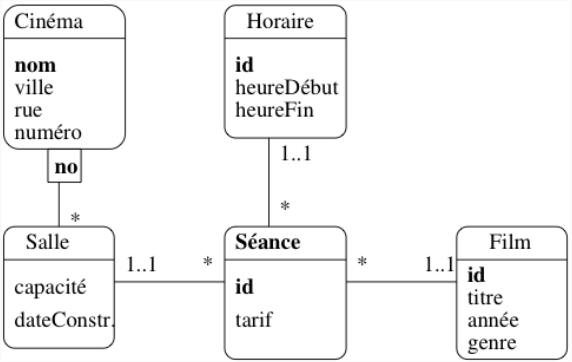
Peut-on avoir des associations entre 3, 4 types d'entité ou plus ?

Oui, mais...



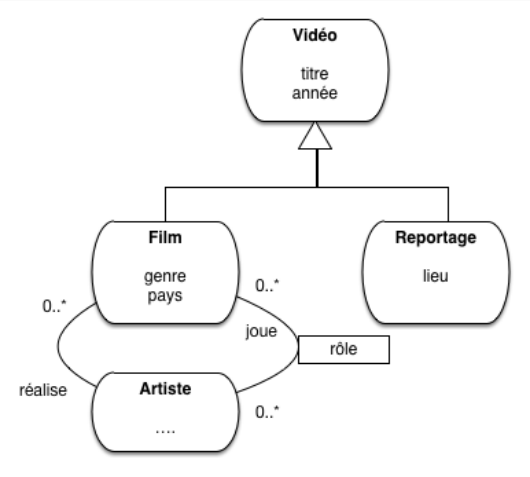
Réification = associations un-à-plusieurs

On peut toujours se ramener aux associations un-à-plusieurs, et ajouter des contraintes ensuite.



Spécialisation orientée-objet

Une structure de modélisation qui va au-delà du relationnel.



À retenir

Constructions avancées du modèle E-A : pas indispensables.

- les entités faibles apportent une certaine clarification sur la modélisation
- Les associations entre plus de deux types d'entités sont à éviter
- La spécialisation est utile quand on souhaite rendre persistant un modèle OO

Pour ce dernier aspect, voir aussi le *mapping* objet-relationnel.

Vers le schéma relationnel (S6.4)

Du diagramme E-A au schéma relationnel normalisé

La modélisation Entité-Association nous donne toutes les informations nécessaires pour obtenir une schéma relationnel normalisé.

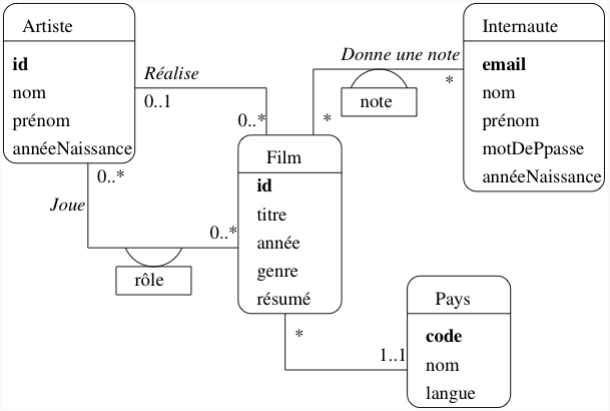
Du Modèle Conceptuel de Données (MCD) au Modèle Relationnel...

Dans cette section :

- Application de l'agorithme de normalisation à un diagramme E-A
- Illustration avec le diagramme et la base des films

Reprenons le diagramme de la base des films

Supposons qu'on ait validé la modélisation suivante



Il n'y a plus qu'à appliquer la normalisation.

Algorithme de normalisation

1. Chaque type d'entité définit une DF de l'identifiant vers les attributs.

   $idFilm \rightarrow titre, année, genre, résumé$

2. Chaque association plusieurs-à-un correspond à une DF entre les identifiants.

   $idFilm \rightarrow codePays$

3. Chaque association (binaire) plusieurs-à-plusieurs correspond à une DF entre l'identifiant de l'association et ses attributs.

   $idFilm, idArtiste \rightarrow rôle$

Il ne reste qu'à appliquer une version simplifiée (sans les vérifications) de l'algorithme de synthèse en 3FN …Et c'est –presque– tout !

Résultat pour la base des films

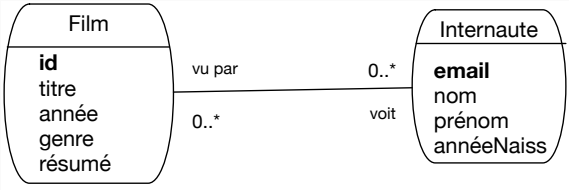
Clés primaires en **gras**, clés étrangères en *italiques*.

- Film (**idFilm**, titre, année, genre, résumé, *idRéalisateur*, *codePays*)
- Artiste (**idArtiste**, nom, prénom, annéeNaissance)
- Rôle (***idFilm***, ***idActeur***, nomRôle)
- Internaute (**email**, nom, prénom, région)
- Notation (***email***, ***idFilm***, note)
- Pays (**code**, nom, langue)

NB : le nommage des attributs est libre.

Attention aux associations plusieurs-à-plusieurs sans attribut

Une association plusieurs-à-plusieurs sans attribut propre.



Ne pas oublier de créer la table `Vu(idFilm, email)`

Autrement, on perd la clé dans la « décomposition » Film–Internaute

Petit exemple

idFilm	titre	année	genre	*idRéalisateur*	*codePays*
20	Impitoyable	1992	Western	100	USA
21	Ennemi d'état	1998	Action	102	USA

La table `Film`

idArtiste	nom	prénom	annéeNaiss
100	Eastwood	Clint	1930
101	Hackman	Gene	1930
102	Scott	Tony	1930
103	Smith	Will	1968

La table `Artiste`

idFilm	***idActeur***	rôle
20	100	William Munny
20	101	Little Bill
21	101	Bril
21	103	Robert Dean

La table `Rôle`

À retenir

À partir du diagramme E-A, on applique l'algorithme de normalisation et on obtient un schéma relationnel en 3FN.

- Pas de magie : le schéma ne sera pas meilleur que la conception
- Attention à ne pas « cacher » de dépendance fonctionnelle dans une entité
- On peut ajouter des contraintes, typages et contrôles quand on crée les tables

Cf. le support de cours pour quelques compléments (spécialisation).

Rétro-ingénierie (S6.5)

Un peu de rétro-ingénierie

La rétro-ingénierie consiste à reconstituer un diagramme E-A à partir d'un schéma relationnel.

Intéressant pour comprendre comment une base a été conçue.

Intéressant aussi pour maîtriser les rapports entre modélisation conceptuelle et modélisation relationnelle.

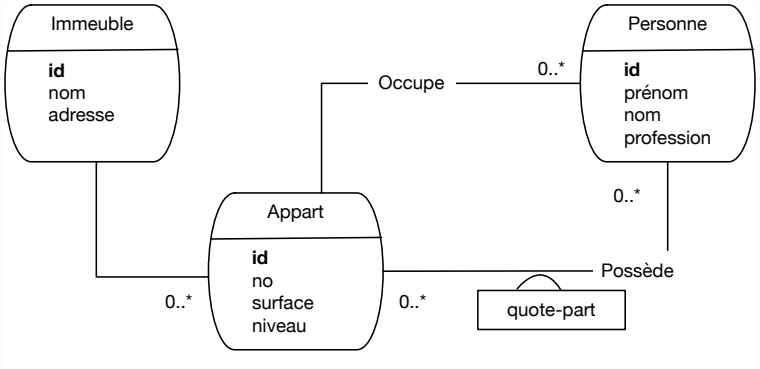
- Rétro-ingénierie de la base des immeubles
- Introduction d'une entité faible
- Introduction d'une réification

Reprenons le schéma de la base des immeubles

- Immeuble(**idImmeuble**, nom, adresse)
- Appart (**idAppart**, no , surface , niveau , *idImmeuble*)
- Personne(**idPersonne**, prénom, nom, profession, *idAppart*)
- Propriétaire(***idPersonne, idAppart***, quotePart)

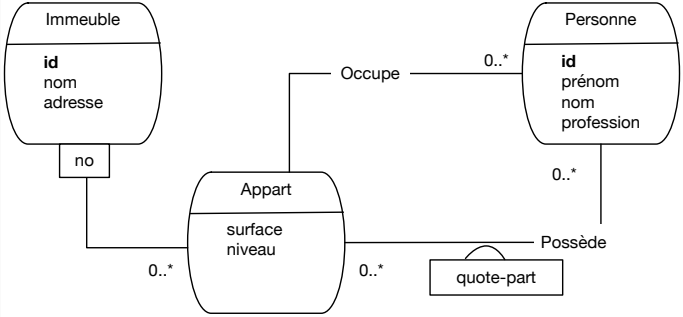
Trouvons les entités, les associations plusieurs-à-plusieurs, et pour finir les associations plusieurs-à-un.

Le diagramme Entité-Association



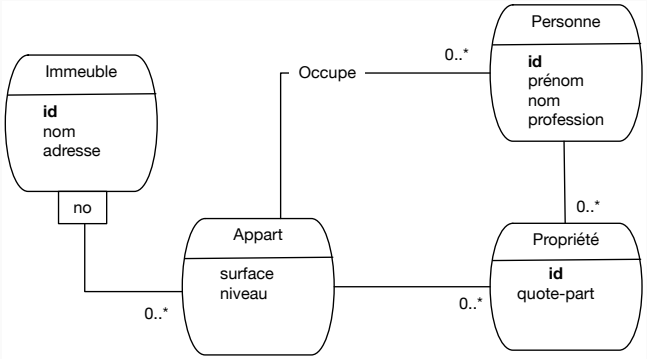
Maintenant on peut se poser des questions sur les choix initiaux

Avec entité faible



- Immeuble(**idImmeuble**, nom, adresse)
- Appart (***idImmeuble*, no** , surface , niveau)
- Personne(**idPersonne**, prénom, nom, profession, *idImmeuble, no*)
- Propriétaire(***idPersonne*, *idImmeuble, no***, quotePart)

Avec réification



- Immeuble(**idImmeuble**, nom, adresse)
- Appart (***idImmeuble*, no** , surface , niveau)
- Personne(**idPersonne**, prénom, nom, profession, *idImmeuble, no*)
- Propriété(**id**, *idPersonne*, *idImmeuble, no*, quotePart)

À retenir

Les modélisations Entité-Association et relationnelle sont très proches l'une de l'autre.

La différence essentielle est la méthode de représentation des associations : directe en E-A, indirecte, par clé primaire et clé étrangère, en relationnel.

Les variantes de modélisation en E-A (typage faible, réification) ont donc un impact en relationnel sur la représentation des clés, et donc sur les requêtes des applications : à maîtriser.