

Conception d'un schéma relationnel

1. Bons et mauvais schémas

FilmSimple

Titre	Annee	NomMES	PrenomMES	AnneeNaiss
Hana-bi	1997	Kitano	Takeshi	1947
Big fish	2003	Burton	Tim	1958
Edward aux mains d'argent	1990	Burton	Tim	1958
Sonatine	1993	Kitano	Takeshi	1947
Pulp Fiction	1995	Tarantino	Quentin	1963
Play Time	1967	Tati	Jacques	1907
Vertigo	1958	Hitchcock	Alfred	1899
Psychose	1960	Hitchcock	Alfred	1899
Parle avec elle	2002	Almodovar	Pedro	1949
Mon oncle	1958	Tati	Jacques	1907
Volver	2006	Almodovar	Pedro	1949
La mauvaise éducation	2004	Almodovar	Pedro	1949

Grave défaut : il est possible de représenter la même information plusieurs fois.

■ **Anomalie d'insertion**

Possibilité d'avoir plusieurs fois le même film avec des données différentes. Qu'est-ce qui distingue 2 films l'un de l'autre ? Peut-il y avoir 2 films différents avec le même titre ?

■ **Anomalie de modification**

Problèmes de mise à jour. Par ex., la modification de l'année de naissance d'Hitchcock pour Vertigo et pas pour Psychose entraîne des informations incohérentes. Il n'y a qu'un seul réalisateur nommé Hitchcock et il ne doit y avoir qu'une seule année de naissance possible.

■ **Anomalie de suppression**

Impossibilité de supprimer un film sans supprimer son metteur en scène.

La bonne méthode

1. représenter individuellement les films et les réalisateurs, de manière à ce qu'une action sur l'un n'entraîne pas systématiquement une action sur l'autre,
2. définir une méthode d'identification d'un film ou d'un réalisateur, qui permette d'assurer que la même information est représentée une seule fois,
3. préserver le lien entre les films et les réalisateurs mais sans introduire de redondance.

Films

Titre	Annee
Hana-bi	1997
Big fish	2003
Edward aux mains d'argent	1990
Sonatine	1993
Pulp Fiction	1995
Play Time	1967
Vertigo	1958
Psychose	1960
Parle avec elle	2002
Mon oncle	1958
Volver	2006
La mauvaise éducation	2004

Films Réalisés

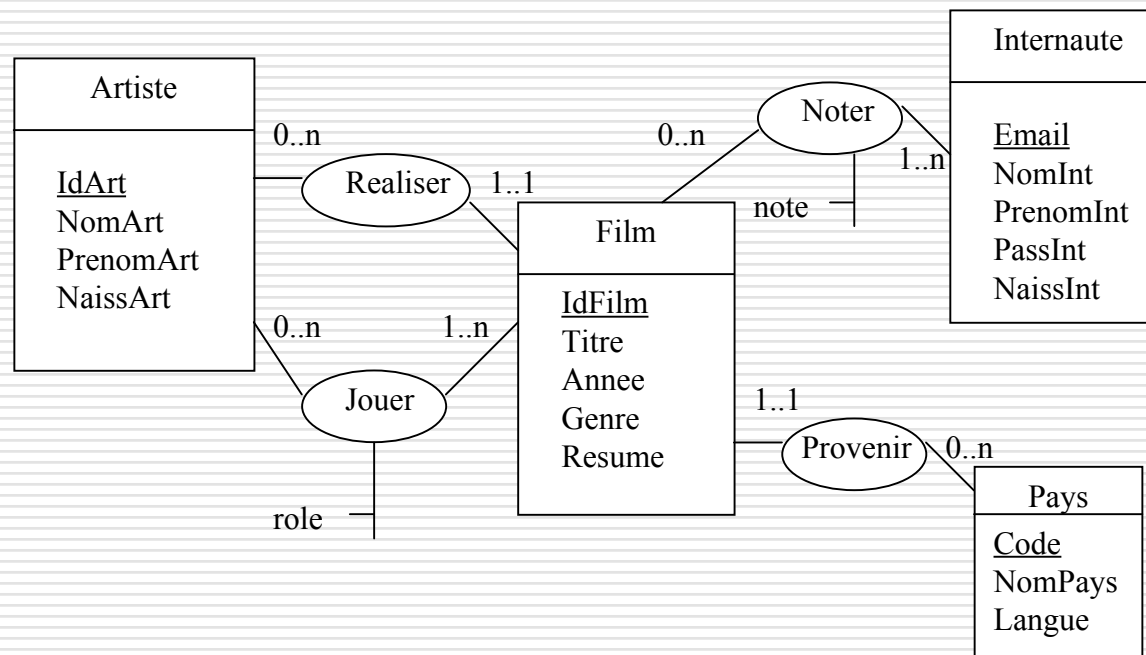
Titre	IdMES
Hana-bi	1
Big fish	2
Edward aux mains d'argent	2
Sonatine	1
Pulp Fiction	3
Play Time	4
Vertigo	5
Psychose	5
Parle avec elle	6
Mon oncle	4
Volver	6
La mauvaise éducation	6

Réalisateurs

idMES	NomMES	PrenomMES	AnneeNaiss
1	Kitano	Takeshi	1947
2	Burton	Tim	1958
3	Tarantino	Quentin	1963
4	Tati	Jacques	1907
5	Hitchcock	Alfred	1899
6	Almodovar	Pedro	1949

2. Le modèle Entité/Association

- Conçu en 1976 et utilisé presque universellement pour la conception des bases de données relationnelles.
- Conception d'un schéma correct essentielle pour le développement d'une application viable.
- Abstraction d'un domaine d'étude relativement aux objectifs fixés.
- Choix de certains aspects de la réalité perçue en fonction des besoins qui doivent être définis.



Entités

Définition 3

On désigne par *entité* tout objet identifiable et pertinent pour l'application.

Notion primordiale qui permet de distinguer les entités les unes des autres.

identifiant ou *clé* permet de faire cette distinction.

La première étape d'une conception consiste à identifier les entités.

La deuxième est de regrouper les entités en ensembles.

Les attributs

Les entités sont caractérisées par des propriétés appelées attributs.

Un attribut est désigné par un nom et prend ses valeurs dans un domaine énumérable.

Un attribut prend une valeur et une seule, il est atomique.

Types d'entités

Elle exprime un type, une classe, un ensemble dont les éléments sont les entités (occurrences).

Définition 4

Le type d'une entité est composé de son nom, de ses attributs et de son identifiant.

Une entité e est une instance du type E .

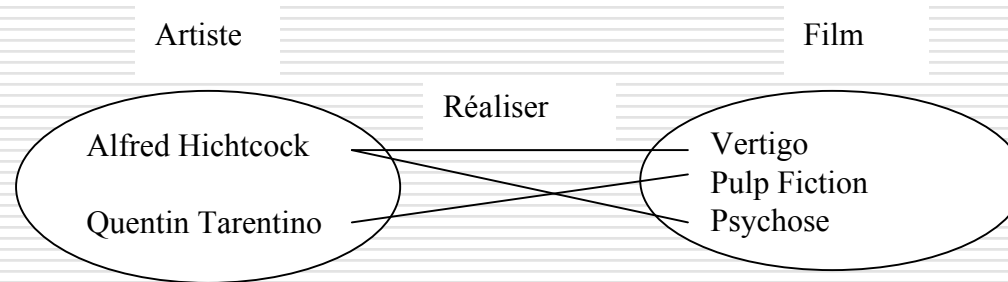
Un ensemble d'entités $\{e_1, e_2, \dots, e_n\}$ est une extension de E .

Définition 5

Soit E un type d'entité et A l'ensemble des attributs de E , une clé de E est un sous-ensemble minimal de A permettant d'identifier de manière unique une entité parmi n'importe quelle extension de E .

-
- Possibilité d'avoir plusieurs clés pour un même type d'entité. L'une est clé primaire et les autres sont clés secondaires. Le choix de la clé primaire est déterminant pour la qualité du schéma.
 - Un identifiant doit être
 - *univalué* : à une occurrence correspond une seule valeur pour un identifiant donné,
 - *discriminant* : à une valeur correspond une seule occurrence de l'individu,
 - *stable* : pour une occurrence donnée d'individu, une fois définie une valeur à son identifiant, cette valeur doit être conservée jusqu'à la destruction de l'occurrence,
 - *minimal* : s'agissant d'un identifiant composite, la suppression d'un de ses composants lui ferait perdre son caractère discriminant.
 - On peut créer un identifiant abstrait appelé id indépendant de tous les autres attributs. De plus, sa taille de stockage est la plus petite possible.

Association binaire



Elle modélise un ensemble d'associations de même nature entre 2 occurrences d'individus (différents ou non).

Définition 6

Une association binaire entre les types d'entités E_1 et E_2 est un ensemble de couples (e_1, e_2) avec $e_1 \in E_1$ et $e_2 \in E_2$.

Définition 7

Soit une association entre 2 types d'entités, la cardinalité est une paire $\{\text{min}, \text{max}\}$ où min (resp. max) désigne le nombre minimal (resp. maximal) de fois où une entité peut intervenir dans l'association.

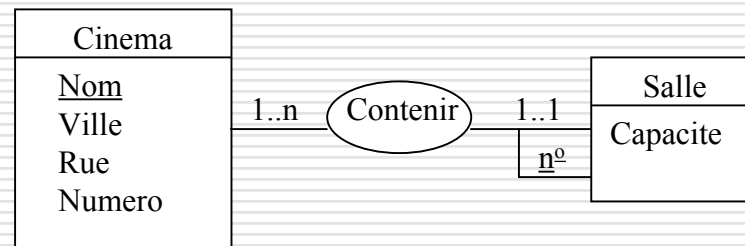
- Dans le cas de cardinalités multiples, on peut avoir des attributs affectés à l'association elle-même. Information ne prenant de sens qu'avec les 2 types d'entités.
- **Remarque :** toute association binaire avec cardinalité (1,1) ne peut être porteuse d'attribut (migre dans l'entité portant cette cardinalité).

Définition 8

La clé d'une association (binaire) entre un type d'entité E_1 et un type d'entité E_2 est le couple constitué de la clé c_1 de E_1 et de la clé c_2 de E_2 .

Entité faible

- Une entité ne peut exister qu'en association avec une autre et est identifiée relativement à cette autre entité.
- Avec une entité faible on obtient un identifiant composé qui est souvent plus pratique à gérer et plus facile qu'avec une association classique.
- La présence d'un type d'entité faible B associé à un type d'entité A implique aussi des contraintes fortes sur les créations, modifications et destructions des instances de B et A.



Exemple : un cinéma et ses salles

Si association classique entre cinéma et salles, créer un identifiant **id** pour le type d'entité salle.

Sinon, numéroter les salles par un numéro interne à chaque cinéma.

La clé d'une salle est constituée de la clé du cinéma et du numéro de la salle.

L'entité salle ne dispose pas d'une identification absolue mais d'une identification relative à une autre entité.

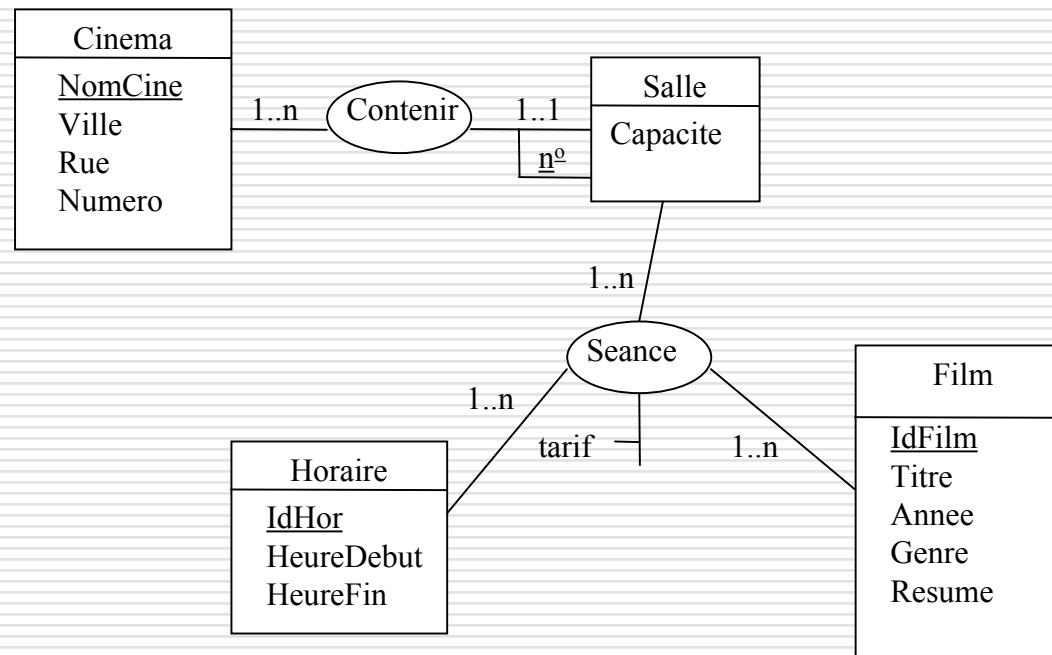
Association généralisée

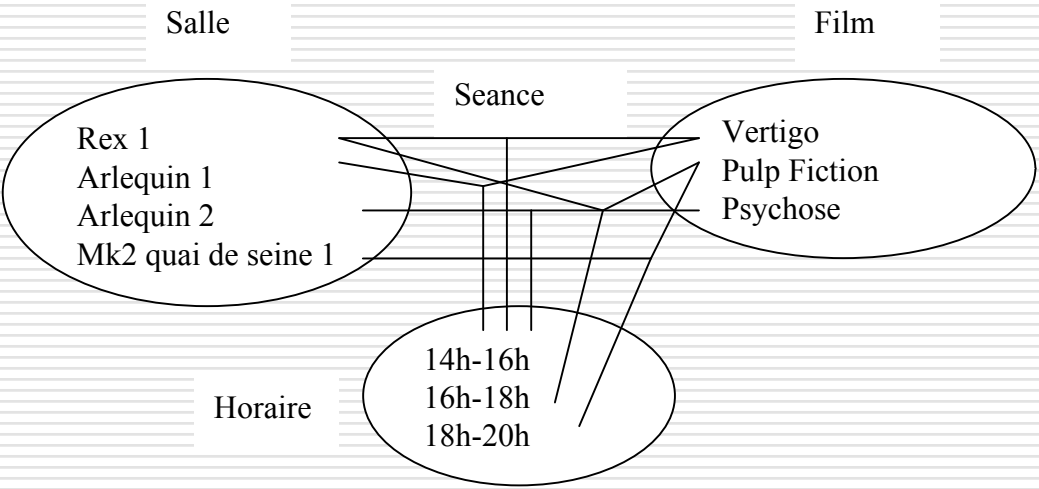
□ La définition d'une association n-aire est une généralisation de celle des associations binaires.

□ **Définition 9**

Une association n-aire entre n types d'entités E_1, E_2, \dots, E_n est un ensemble de n-uplets (e_1, e_2, \dots, e_n) où chaque e_i appartient à E_i .

□ La *dimension* est le nombre d'entités de l'association. La *collection* est la liste de ces entités. L'occurrence d'une association est celles des entités de sa collection. La dimension est en pratique limitée à 4.

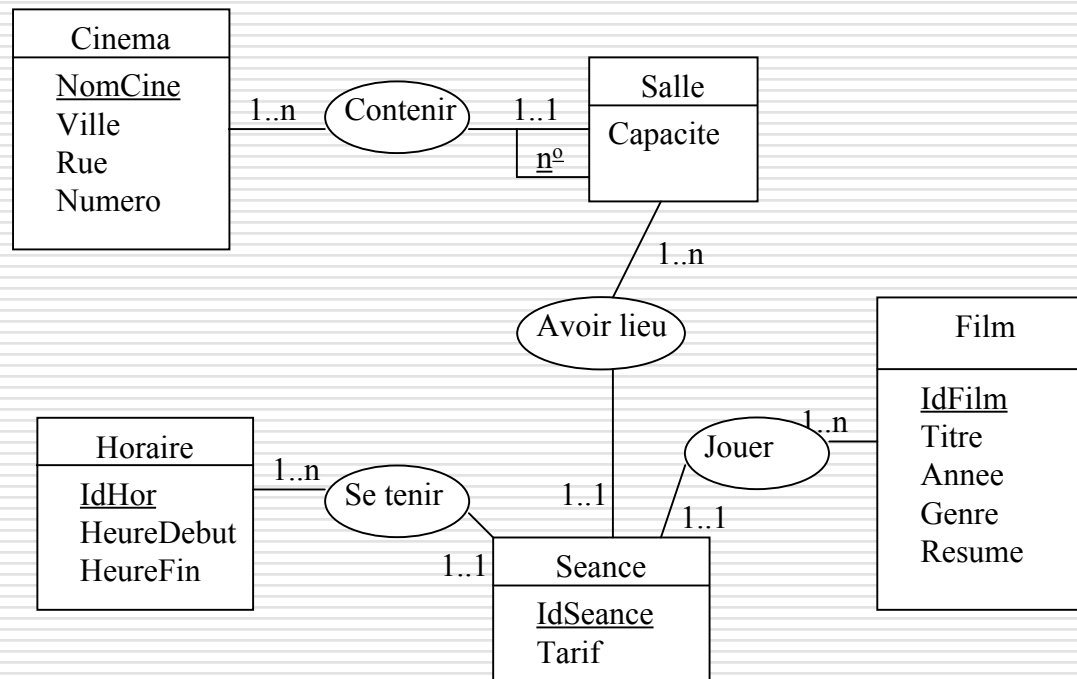




-
- La clé de Seance doit être (NomCiné, noSalle, IdFilm, IdHor)
Une telle clé est assez volumineuse et elle ne permet pas la contrainte « dans une salle, pour un horaire donné, il n'y a qu'un seul film »

 - Il est possible de remplacer l'association par un type d'identité.
Mais cela ne règle pas toujours le problème !!

 - **Règle :** soit A une association entre les types d'entité (E_1, E_2, \dots, E_n), pour sa transformation en type d'entité, il faut
 - attribuer un identifiant autonome à A,
 - créer une association A_i entre A et chacun des E_i . de cardinalité 1..1 (du côté de A) et 1..n de l'autre.



Le problème de la salle persiste : dans une salle, pour un horaire donné, il est possible d'avoir plusieurs films !

-
- Le modèle E/A permet de spécifier la structure des informations de la base et d'offrir une représentation abstraite indépendante du modèle logique.

 - En général, on essaie
 - d'éviter toute redondance,
 - de privilégier la simplicité et la lisibilité en ne représentant que ce qui est strictement nécessaire.

3. Le modèle relationnel

- ❑ Totale indépendance entre les représentations logiques et physiques. Très simple : une seule structure la *relation* représentée par une *table*.
- ❑ Une relation possède un nom, et se compose de colonnes désignées par un nom d'attribut prenant valeur dans un domaine.
- ❑ Chaque ligne (tuple) correspond à une entité. Le degré (arité) est le nombre d'attributs et le cardinal est le nombre de tuples.
- ❑ Un schéma relationnel est constitué d'un ensemble de schémas de relations.

□ **Domaine**

Ensemble d'instances d'un type élémentaire (\neq type structuré). Les types sont figés par le système.

□ **Attribut**

Sert à indiquer le contenu d'une colonne et à la référencer quand on effectue des opérations. Le nom d'un attribut peut apparaître dans plusieurs schémas de relations.

□ **Schéma de relation**

Nom suivi de la liste des attributs, chaque attribut n'étant présent qu'une seule fois et étant associé à son domaine.

$$R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$$

□ Instance d'une relation

Sous-ensemble fini de produit cartésien des domaines des attributs de R
l'ordre des lignes n'a pas d'importance,

- on ne peut pas trouver deux fois la même ligne,
- et il n'y a pas de case vide.

□ Clé primaire d'une relation

Plus petit sous-ensemble des attributs qui permet d'identifier chaque ligne de manière unique. La clé primaire est mise **en gras**. Le choix de la clé est très important pour la qualité du schéma.

□ Clé étrangère d'une relation

Attributs d'une relation qui correspondent à la clé primaire d'une autre table.

Base de données

- ❑ Ensemble fini de relations.
- ❑ Le schéma de la base est l'ensemble des schémas des relations.
- ❑ Le choix des relations est primordial car il détermine en grande partie la qualité de la base.
- ❑ On conçoit le schéma à l'aide d'un modèle de données conceptuel (par ex, le modèle E/A), puis en transcrivant ce schéma en un schéma relationnel.

Contraintes d'intégrités : conditions à vérifier par ttes les relations

- clé primaire : tte relation possède une clé primaire
- clé étrangère : même sémantique que la clé primaire correspondante
- intégrité référentielle : l'ensemble des valeurs d'une clé étrangère est inclus ou égal à celui des valeurs de la clé primaire correspondante
- existentielle : les valeurs d'un attribut sont définies pour tous les tuples de la relation
- de domaine : précision sur le domaine des valeurs d'un attribut
- de valeur par défaut : nécessité d'une valeur à affecter à un attribut lors de l'insertion d'un tuple
- sur un tuple : prédicat mettant en jeu plusieurs attributs
- de dépendance fonctionnelle (DF) : vérification des DF entre attributs
- contrainte de normalité : vérification du respect d'une forme normale
- de cardinalité : nécessité de bornes pour le nombre de tuples d'une relation
- pré-post condition : condition à vérifier par le SGBD lors de l'évolution de la BD, sous forme d'une condition explicite avant, pendant ou après
- règle actives : actions à exécuter suite à un changement d'état de la BD

4. Passage d'un schéma E/A à un schéma relationnel

Dépendances fonctionnelles (DF)

Définition

Soient X et Y deux ensembles d'attributs.

On dit que X détermine Y (ou Y dépend de X), noté $X \rightarrow Y$ lorsque deux tuples qui ont une même valeur sur les attributs de X implique qu'ils ont même valeur sur les attributs de Y

= à toute valeur de X n'est associée qu'une et une seule valeur de Y , dans toute extension de la relation.

Propriétés des DF

- réflexivité : $X \rightarrow X'$ avec $X' \subseteq X$
- projection : $X \rightarrow Y, Z \Rightarrow X \rightarrow Y$ et $X \rightarrow Z$
- additivité : $X \rightarrow Y$ et $X \rightarrow Z \Rightarrow X \rightarrow Y, Z$
- augmentation : $X \rightarrow Y \Rightarrow X, Z \rightarrow Y$
- transitivité : $X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$
- pseudo-transitivité : $X \rightarrow Y$ et $Y, W \rightarrow Z \Rightarrow X, W \rightarrow Z$

Théorème

Toutes les DF peuvent être générées à partir d'un ensemble d'entre elles par les propriétés de réflexivité, augmentation et transitivité.

Typologie des DF : $X \rightarrow Y$

- canonique : Y est réduit à un seul attribut
- triviale : $Y \subseteq X$
- élémentaire : il n'existe pas $X' \subset X$ tel que $X' \rightarrow Y$
- directe : il n'existe pas Z tel que $X \rightarrow Z$, $Z \rightarrow Y$ et $Z \rightarrow X$ n'existe pas

Hyper-graphe des DF

- Sommets : attributs
 - Hyper-arcs : DF
-
- La fermeture transitive d'un ensemble de DF consiste à appliquer systématiquement tant que cela est possible la propriété de transitivité.
 - La couverture minimale d'un hyper-graphe de DF est un ensemble de DF F vérifiant qu'aucune DF n'est redondante (\forall DF f , $F - \{f\} \neq F$) et que toute DF est dans la fermeture transitive de F .

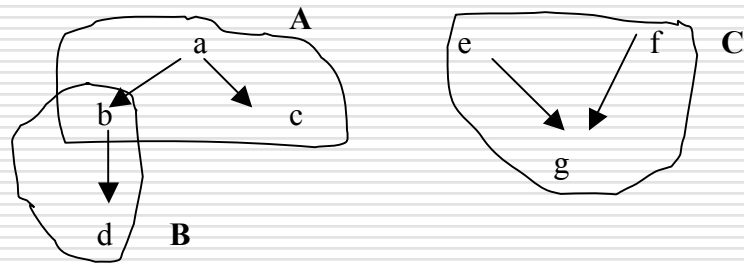
Formes normales

Notation : $R (C_1, \dots, C_k, A_1, \dots, A_n)$ avec (C_1, \dots, C_k) clé primaire de R .

- 1FN : les attributs d'une relation hors clé primaire sont fonctionnellement dépendants de la clé primaire
 $C_1, \dots, C_k \rightarrow A_1, \dots, A_n$
- 2FN = 1FN + élémentarité : les attributs d'une relation hors clé primaire sont en DF élémentaire avec la clé primaire
 $\neg \exists C \subset \{C_1, \dots, C_k\}$ et A_j tel que $C \rightarrow A_j$
- 3FN = 2FN + non transitivité : les attributs d'une relation hors clé primaire sont en DF élémentaire directe avec la clé primaire (\neg DF entre attributs hors clé)
 $\neg \exists A', A'' \subset \{A_1, \dots, A_n\}$ et $A' \cap A'' = \emptyset$ tel que $A' \rightarrow A''$
- 3FN de Boyce-Codd-Kent
 $\neg \exists A \subset \{A_1, \dots, A_n\}, C \subset \{C_1, \dots, C_k\}$ tel que $A \rightarrow C$

Théorème : toute relation admet une décomposition sans perte ni redondance en troisième forme normale.

- Une fois l'ensemble des DF établi, le schéma relationnel se déduit de l'hyper-graphe des DF de couverture minimale.
- Pour chaque sommet interne de l'hyper-graphe, on obtient une relation dont la clé primaire correspond au sommet interne et dont les attributs hors clé correspondent aux sommets fils du sommet interne
- Les attributs correspondant à des sommets fils qui sont eux-mêmes des sommets internes sont des clés étrangères.



devient

A (a, b, c)

B (b, d)

C (e, f, g)

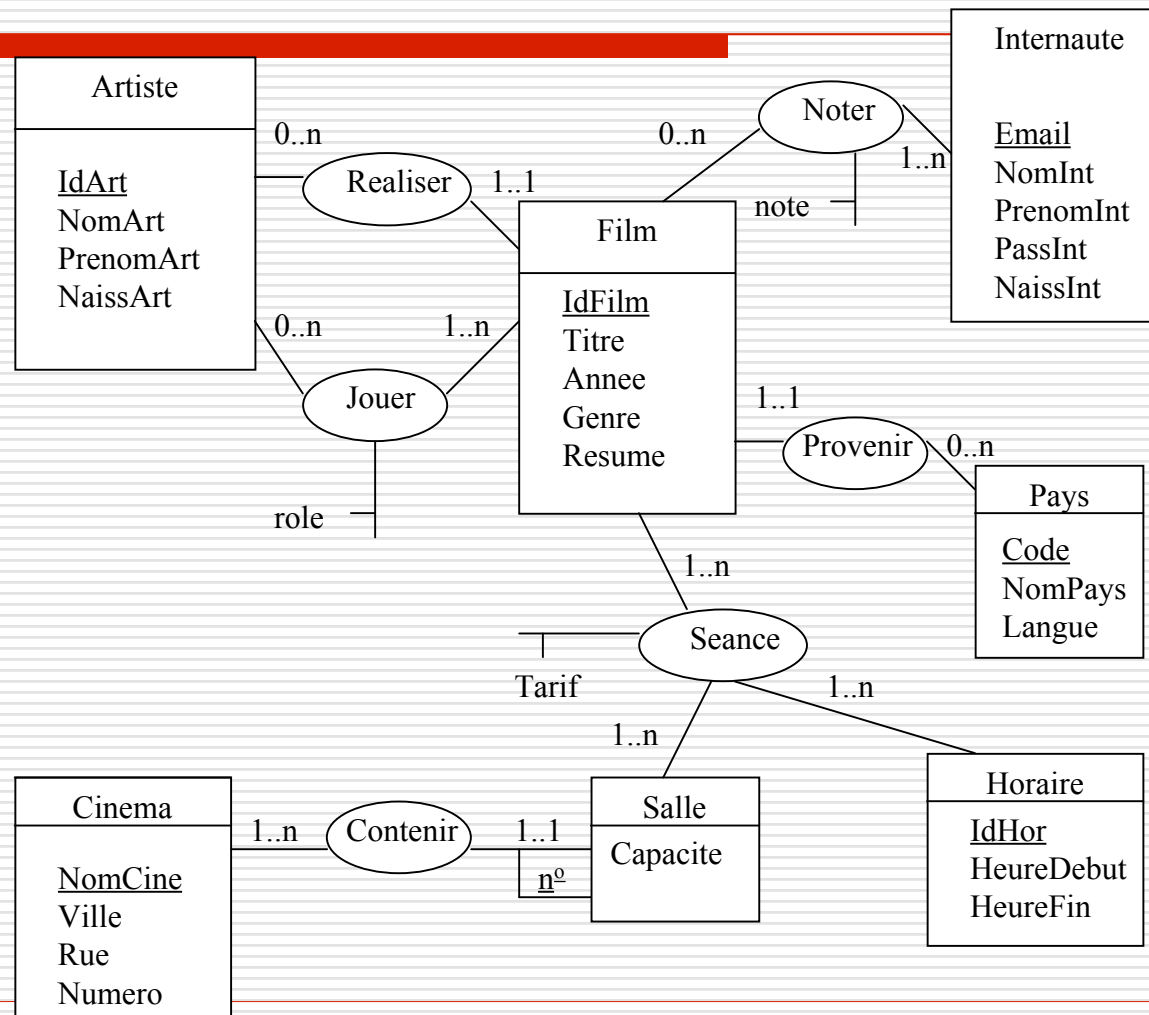
A est la clé de la relation A

B est la clé de relation B

(e, f) est la clé de la relation C

valeurs (A.b) \subseteq valeurs (B.b)

Règles de passage d'un schéma E/A à un schéma relationnel en 3FN



Règles (1/3)

- **Entité** : pour chaque entité, on crée une relation de même nom et chaque propriété, y compris l'identifiant, devient un attribut de la relation et l'identifiant constitue la clé.

Film (**IdFilm**, Titre, Annee, Genre, Resume)

Artiste (**IdArt**, NomArt, PrenomArt, NaissArt)

Internaute (**Email**, NomInt, PrenomInt, PassInt, NaissInt)

Pays (**Code**, NomPays, Langue)

Cinéma (**NomCine**, Ville, Rue, Numero)

Horaire (**IdHor**, HeureDebut, HeureFin)

- **Entité faible** : on crée un mécanisme de clé étrangère pour référencer l'entité forte dans l'entité faible (partie de l'identifiant de l'entité faible).

Salle (**NomCine**, No, Capacite)

Règles (2/3)

- **Association binaire 1:n** entre A et B : on crée les relations R_A et R_B correspondant aux entités A et B, l'identifiant de B devient un attribut de R_A . Une occurrence de A référence l'occurrence de B qui lui est associée à l'aide d'une clé étrangère.

Film (**IdFilm**, Titre, Année, Genre, Resume, IdMES, Code)

Le rôle tenu par l'artiste disparaît. Les clés étrangères peuvent ne pas avoir le même nom que ceux de la clé primaire.

- **Association binaire n:n** entre A et B : on crée les relations R_A et R_B correspondant aux entités A et B, on crée une relation R_{AB} pour l'association, la clé de R_A et la clé de R_B deviennent des attributs de R_{AB} , la clé de cette relation est la concaténation des clés de R_A et R_B et les propriétés de l'association deviennent des attributs de R_{AB} .

Role (**IdFilm**, **IdArt**, NomRole)

Notation (**IdFilm**, **Email**, note)

Règles (3/3)

- **Association ternaire** : limite du modèles E/A en matière de contraintes. On peut appliquer la règle énoncée pour les associations binaires généralisée.

Seance (**IdFilm**, **NomCine**, **No**, **IdHor**, Tarif)

La même salle peut présenter deux films différents au même horaires. Pour éviter cette situation, la clé devient (**NomCine**, **No**, **IdHor**), mais on ne respecte plus la règle de passage.

Si on utilise la représentation d'une séance par une entité et des associations binaires, le problème reste le même.

Seance (**IdSeance**, IdFilm, NomCine, No, IdHor, Tarif)

En cas d'association entre plus de 2 entités, la clé de la table représentant l'association est un sous-ensemble de la concaténation des clés.

Il faut se poser soigneusement la question de la clé au moment de la création de la table car elle ne peut plus être déduite du schéma E/A.

Choix des identifiants

- ❑ Choisir un identifiant « neutre » qui ne soit pas une propriété de l'entité.
 1. Chaque valeur de l'identifiant doit caractériser de manière unique une occurrence.
 2. Si on utilise un ensemble de propriétés comme identifiant, la référence à une occurrence est très lourde.
 3. L'identifiant sert de référence externe et ne doit donc jamais être modifiable.
- ❑ Le problème de l'identifiant « neutre » est qu'il ne donne pas d'indication sur l'occurrence qu'il réfère.

Dénormalisation du modèle logique

- Deux objectifs principaux :
 1. Simplifier le schéma relationnel en réduisant le nombre d'éléments qui le composent.
 2. Faciliter l'accès aux données en introduisant un certain degré de redondance.
- Ces techniques reviennent à introduire des anomalies dans le schéma. Il faut donc comparer le gain attendu avec les risques courus.

-
- Suppression de relations : on peut supprimer les entités qui portent peu d'attributs en les déplaçant vers une autre relation.

Exemple 1 : on supprime l'entité Cinéma.

Salle (**NomCine**, **No**, Ville, Rue, Numero)

Exemple 2 : on supprime Horaire qui gère juste un couple d'heures.

Seance (**NomCine**, **No**, **HeureDebut**, HeureFin, IdFilm, Tarif)

Pas de création d'un type d'entité Horaire ou Date → attribut dans le schéma.

- Introduction de redondance : l'accès à une information peut être long ou compliqué.

Exemple : à partir de Seance, il faut consulter Salle puis Cinema pour connaître l'adresse du cinéma et faire un calcul pour obtenir le nombre de salles.

Seance (**NomCine**, **No**, **HeureDebut**, HeureFin, IdFilm, Ville, Rue, Numero, Tarif)

Cinema (**NomCine**, Ville, Rue, Numero, NbSalles)

Mécanisme de triggers pour effectuer automatiquement la répercussion de la modification d'une donnée sur ses autres versions dans la base.