

Module Introduction à la gestion des données

Cours Modélisation

Hervé Owsinski
Herve.Owsinski@uphf.fr

Semestre 1



2026-2027

Partie I

Modèle conceptuel

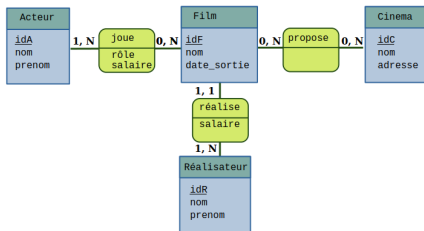
Modèle conceptuel

Modèle Entité - Association

Modèle Entité - Association

Historique et grands principes

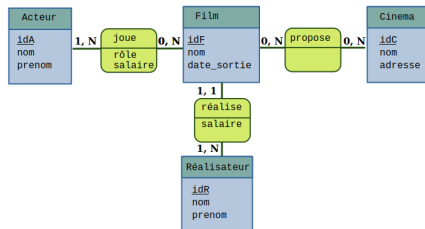
- **Modèle Entité Association** : Peter Chen en **1976** ;
- But : modéliser les besoins de façon abstraite, **indépendamment des technologies** mis en oeuvre ;
- Le **domaine d'application** (la réalité à représenter) est retranscrit sous forme d'un ensemble d'entités reliées entre elles ;
- Plus d'une dizaine de variantes de représentation existent.



Modèle Entité - Association

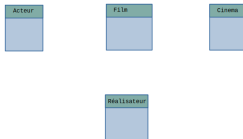
Les concepts de base

- Formalisme **graphique** pour la modélisation des données ;
- Des **entités** ;
- Associées entre elles via des **associations** ;
- Des **attributs** et des **identifiants** ;
- Des **cardinalités**.



Entité

- Analyse de besoins : on met en valeur des "choses" identifiables sans ambiguïté ;
- Une **entité** représente un objet du domaine étudié, identifiable. Elle possède des caractéristiques propres qui la distinguent des autres entités ;
- Exemple : le film Die Hard, l'acteur Bruce Willis, la ville de Valenciennes... ;
- Une **classe d'entités** est un **ensemble d'entités** de même nature. Qui dit ensemble, dit qu'aucun doublon ne peut exister ;
- Par abus : souvent "entité" pour "classe d'entités"..



Modèle Entité - Association

Entités sur un exemple

Tiré de l'analyse des besoins : "Les **films** sont à l'affiche dans des **cinémas**. Ils sont réalisés par un **réalisateur** et des **acteurs** y jouent la plupart du temps."



Modèle Entité - Association

Attributs des entités sur un exemple

Les **entités** possèdent des caractéristiques propres qu'on nommera des **attributs**. L'ensemble des attributs d'une entité doit la décrire entièrement.



Modèle Entité - Association

Identifiants

Un **identifiant** (aussi appelé **clé primaire**) est un ou plusieurs attributs conjoints permettant de distinguer de façon unique chaque occurrence d'une entité : **chaque instance a son identifiant unique.**

Soit on le souligne, soit on le place premier et séparé des autres, soit on place (PK) (Primary Key) à côté de son nom.

Notez bien que nom et prénom ne suffisent pas la plupart du temps.

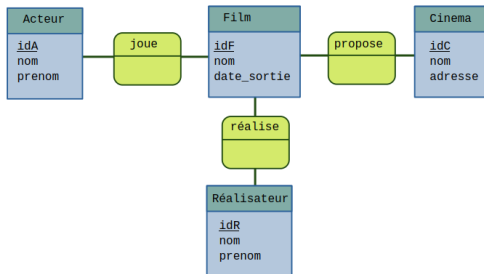


Modèle Entité - Association

Associations

Une **association** relie deux (ou plusieurs) entités pour représenter un lien entre elles..

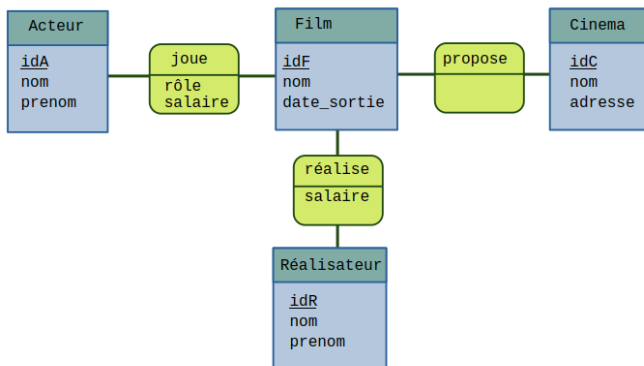
Ces associations sont issues de l'analyse des besoins : "Les films sont **à l'affiche** dans des cinémas. Ils sont **réalisés** par un réalisateur et des acteurs y **jouent** la plupart du temps."



Modèle Entité - Association

Associations

Une **association** peut porter des informations supplémentaires sous forme d'**attributs**. Ce n'est pas obligatoire, parfois seule la connaissance d'une relation entre deux entités est suffisante.



Cardinalité

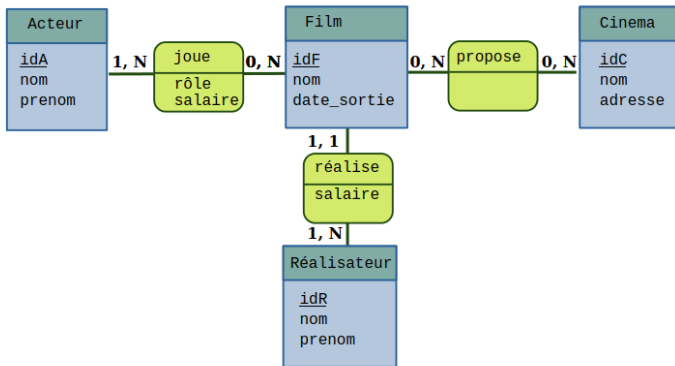
La **cardinalité** exprime le nombre minimum et maximum d'associations d'une entité A vers l'entité B.

- **0,1** : une entité A peut avoir 0 ou 1 association avec des entités B ;
- **0,N** : une A peut avoir de 0 ou plusieurs associations avec des B ;
- **1,N** : une A peut avoir de 1 à plusieurs associations avec des B ;
- **1,1** : une A a obligatoirement une et une seule association avec une B ;
- **N,M** : on doit en avoir plusieurs, entre N et M ;

Modèle Entité - Association

Cardinalité

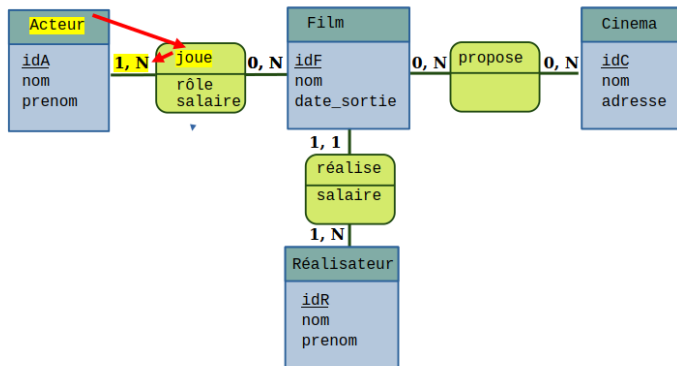
Voici ce qu'on pourrait placer sur notre modèle.



Modèle Entité - Association

Cardinalité

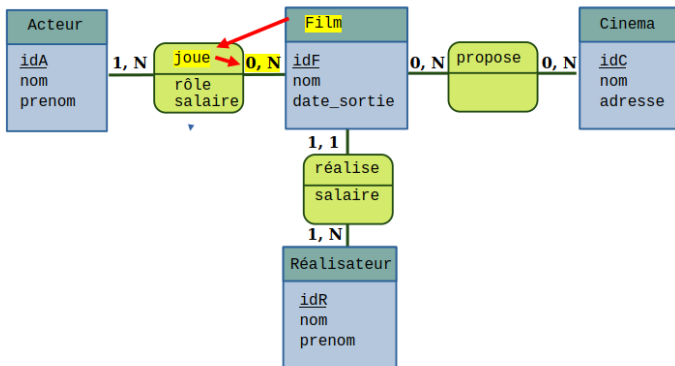
Un acteur peut jouer dans 1 à plusieurs films.



Modèle Entité - Association

Cardinalité

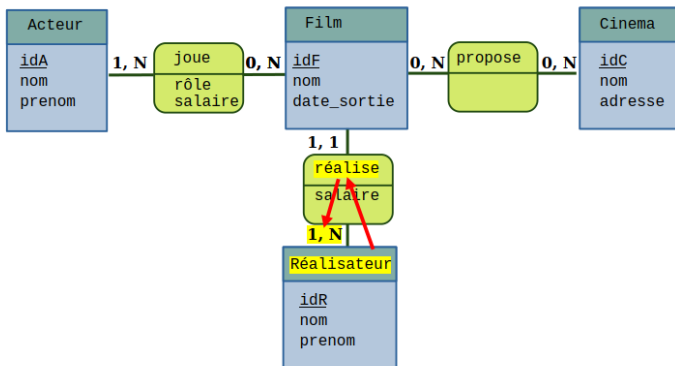
Un film est joué par 0 à plusieurs acteurs.



Modèle Entité - Association

Cardinalité

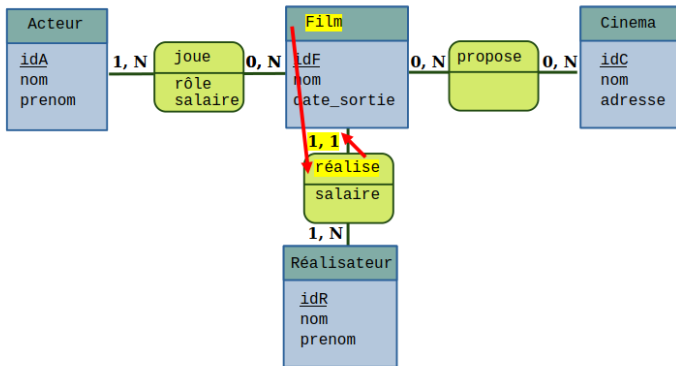
Un réalisateur peut réaliser de 1 à plusieurs films.



Modèle Entité - Association

Cardinalité

Un film est réalisé par 1 et un seul réalisateur.



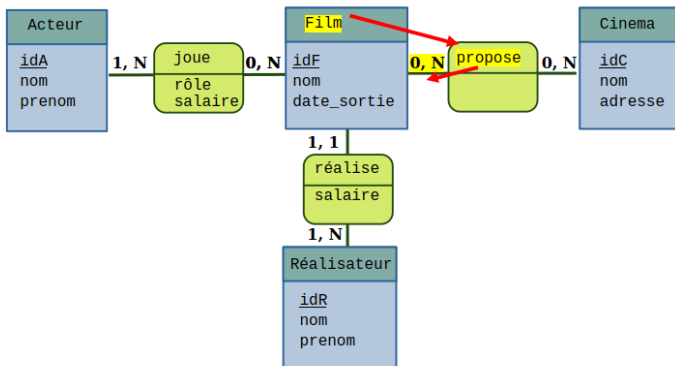
vrai ?

Est-ce toujours

Modèle Entité - Association

Cardinalité

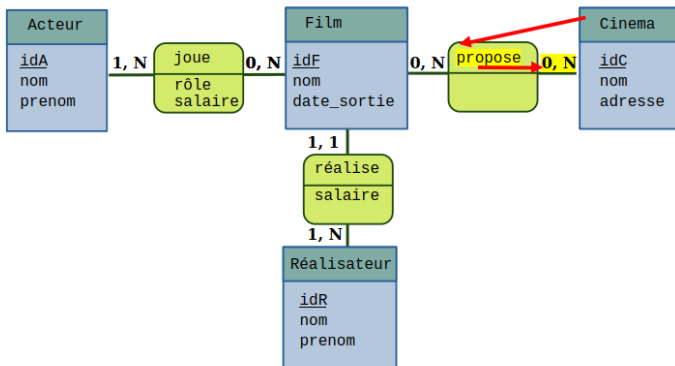
Un film peut être proposé dans 0 à plusieurs cinémas.



Modèle Entité - Association

Cardinalité

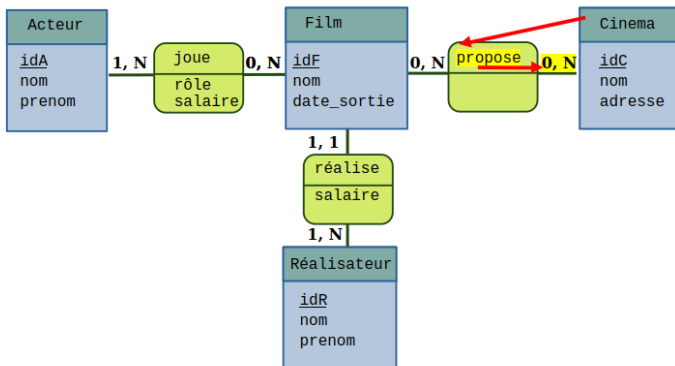
Un cinéma ...



Modèle Entité - Association

Cardinalité

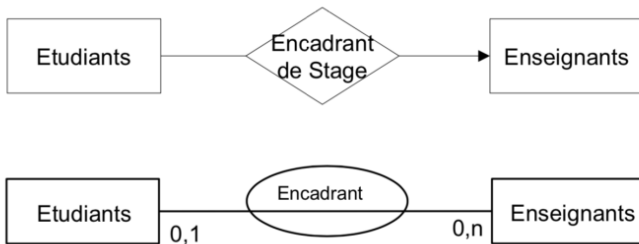
Un cinéma peut proposer de 0 à plusieurs films.



Compléments sur le schéma entité-association

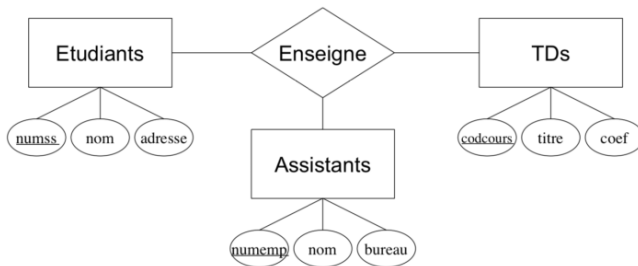
Autres représentations

Il existe de multiples façons de représenter un schéma entité - association.



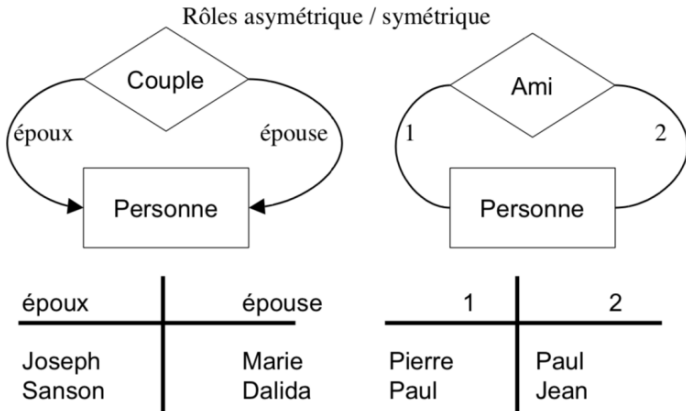
Autres représentations

Il existe des associations ternaires, reliant non pas deux entités mais trois.
On évitera d'utiliser les associations **ternaires**, sauf si nous n'avons pas le choix.



Association réflexive

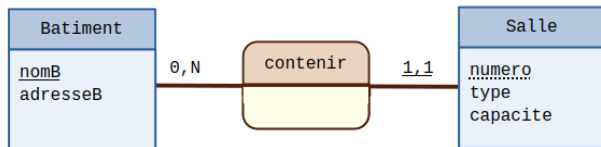
Une association peut relier deux entités de même classe !



Entité faible

Une entité est faible si, pour l'identifier exactement, on a besoin de la localiser par rapport à une autre entité.

Exemple : un appartement dans un immeuble, un dossier dans une armoire...



Pour identifier une salle, son numéro de salle ne suffit pas : il faut l'associer à l'identifiant du bâtiment.

On le signale en soulignant en pointillé l'identifiant. L'association est bien 1,1 du côté de l'entité faible : chaque entité faible est associée à une et une seule entité de référence. D'ailleurs, on la souligne pour indiquer que l'identifiant réel complet d'une Salle est le couple (nomB, salle).

Partie II

Modèle Logique : modèle Relationnel

Modèle logique : Schéma relationnel

Les 3 types de modèles

DEFINITIONS (à connaître)

def Modèle **conceptuel** (abstrait) : les idées sous forme de concepts ;

def Modèle **logique** : choix d'un formalisme mathématique ;

def Modèle **physique** : choix d'une implémentation sur système informatique.

Modèle logique relationnel

Dans le **modèle relationnel**, on choisit de rassembler les données sous forme de **tuples ayant des propriétés communes**.

Qu'est-ce qu'une relation ?

Définitions

def Une **relation** regroupe les données sous forme d'un ensemble de **tuples** possédant tous la même structure :

Exemple : { (5, 'alice', 12.3), (6, 'bob', 9.3) }

def Le **schéma relationnel** est la structure partagée par les tuples d'une relation. Ce schéma spécifie le nom de la relation et la **position**, le **nom** et le **domaine** de chaque attribut ;

Exemple : **Personne** (idP :entier, nom :texte, note :réel)

- Souvent, on n'indique pas le type sur ce schéma :

Exemple 1 : **Personne** (idP, nom, note)

- Il existe beaucoup de conventions de nommmage :

Exemple 2 : **PERSONNE** (IdP, Nom, Note)

Exemple 3 : **PERSONNE** (IDP, NOM, NOTE)

Quelques précisions

- Exemple de relation : **Personne** (idP, nom, note)
- Exemple de tuples : { (5, 'alice', 12.3), (6, 'bob', 9.3) }
- Chaque **tuple** représente un enregistrement (une voiture, un individu...).
- On les représente donc souvent mentalement comme les lignes du tableau formé par les tuples si on les aligne correctement :
$$\begin{Bmatrix} (5, & \text{'alice'}, & 12.3), \\ (6, & \text{'bob'}, & 9.3) \end{Bmatrix}$$
- Les **attributs** représentent les caractéristiques des enregistrements (comme pour le modèle E/A);

Intention et Extension

- Le **schéma relationnel** **Personne** (idP, nom, note) est l'**intention** ;
- Les tuples stockés { (1, 'alice', 12.3), (...), (...) } constituent l'**extension** ;

Relation : analogie avec un tableau 2D

- **Relation** : ensemble de tuples ayant la même structure.
- Soit le schéma **Personne** (idP, nom, note)
- Soit l'extension : { (1, 'alice', 12.3), (2, 'bob', 9.3), (3, 'claire', 16.2) }
On peut représenter cette relation et son extension de cette façon :

<u>idP</u>	nom	note
1	alice	12.3
2	bob	9.3
3	claire	16.2

- Une relation étant un ensemble, chaque tuple ne peut apparaître qu'une unique fois à l'identique.
- Chaque clé primaire doit être unique également : idP ne peut pas prendre deux fois la même valeur.
- Chaque ligne est un **tuple**.
- Chaque colonne correspond à un **attribut**.

Primary Key (PK) et Foreign Key (FK)

Personne (idP, nom, prenom)

Film (idF, titre, #realisateur)

<u>idP</u> (PK)	nom	prénom
1	Villeneuve	Denis
2	Gerwig	Greta
3	Nolan	Christopher
4	DaCosta	Nia

<u>idF</u> (PK)	titre	#réalisateur (FK)
101	Dune : Part Two (2024)	1
102	Barbie (2023)	2
103	Oppenheimer (2023)	3
104	Dune (2021)	1

- PK Clé primaire : Valeurs(idF) = {101, 102, 103, 104}.
- FK Clé étrangère : réalisateur avec Valeurs(réalisateur) $\subseteq$ Valeurs(idP)
- Clé étrangère : réalisateur avec {1, 2, 3} $\subseteq$ {1, 2, 3, 4}

Primary Key (PK) et Foreign Key (FK)

Eudiant (idEtu, nom, prenom, section)

Connexion (idC, cours, date, #idE).

<u>idEtu</u> (PK)	nom	prenom	section
E001	Alice	Dupont	L1 MATH
E002	Bob	Martin	L1 INFO
E003	Claire	Leroy	L1 INFO
E004	David	Moreau	L1 MATH

<u>idC</u> (PK)	cours	date	#idE (FK)
1	Fonctions	23/09	E001
2	Algo	23/09	E001
3	Algo	25/09	E001
4	Base de données	25/09	E003

- PK Clé primaire : Valeurs(idC) = {1, 2, 3, 4}.
- FK Clé étrangère : idE avec Valeurs(idE) $\subseteq$ Valeurs(idEtu)
- Clé étrangère : réalisateur avec {E001, E003} $\subseteq$ {E001, E002, E003, E004}

Modèle relationnel : un cas plus complexe

<u>num_etu</u> (PK)	nom	prénom	anneeNaissance
136	Rochat	Jean	2007
253	Aubry	Annie	2006
101	Duval	André	2006
147	Rochat	Marc	2005

<u>cours</u> (PK)	titre	horaire
BD	Base de données relationnelles	10,5
SI	Système d'Information	12

#num_etu (FK)	#cours (FK)	année
253	SI	2024
253	BD	2024
101	BD	2024
101	SI	2024

La dernière table ne possédant pas de clé primaire, son triplet en fait office "naturellement" puisque c'est un ensemble.

Donc un étudiant ne pourra jamais s'inscrire deux fois au même cours la même année : rajouter (253, SI, 2024) est impossible car ce tuple existe déjà.

Définition clés primaires et étrangères

def Une **clé primaire / primary key / PK** est un tuple d'attributs qui identifie sans ambiguïté l'un des tuples de sa relation.

- Chaque valeur de clé primaire doit être **UNIQUE**.
- Dans une relation, les attributs clés primaires sont soulignés.

def **clé étrangère / foreign key / FK** est un tuple d'attributs dont les valeurs doivent nécessairement correspondre à des valeurs qui apparaissent dans la clé primaire référencée.

$$Valeurs(FK) \subseteq Valeurs(PK)$$

- Une valeur de clé étrangère peut apparaître plusieurs fois.
- Dans une relation, on place # devant le ou les attributs correspondants.

Modèle relationnel : exemple de clés, la suite

Notez que l'attribut **idP** est souligné : c'est une clé primaire.

<u>idP</u> (PK)	nom	prénom
1	Villeneuve	Denis
2	Gerwig	Greta
3	Nolan	Christopher
4	DaCosta	Nia

- Rajouter (5, Abrams, J.J.) dans cette table est possible car le tuple (5) n'est pas encore présent en tant que clé primaire ;
- Rajouter (6, Steven, Spielberg) dans cette table est possible (même si le prénom n'est pas à la bonne place) car le tuple (6) n'est pas encore présent en tant que clé primaire ;
- Rajouter (1, Spielberg, Spielberg) dans cette table est impossible car le tuple (1) est déjà une valeur de clé primaire ;
- Rajouter (7, Villeneuve, Denis) dans cette table est possible car le tuple (7) n'est pas encore une clé primaire et il n'y a aucune contrainte particulière indiquée sur nom et prénom ;
- Rajouter (1, Villeneuve, Denis) dans cette table est impossible car (1) est déjà une valeur de clé primaire ;

Modèle relationnel : exemple de clés, la suite

Notez que l'attribut **idP** n'est pas souligné.

idP	nom	prénom
1	Villeneuve	Denis
2	Gerwig	Greta
3	Nolan	Christopher
4	DaCosta	Nia

- Rajouter (5, Abrams, J.J.) dans cette table est possible car ce triplet n'est pas encore présent ;
- Rajouter (6, Steven, Spielberg) dans cette table est possible car ce triplet n'est pas encore présent ;
- Rajouter (1, Spielberg, Spielberg) dans cette table est possible car ce triplet n'est pas encore présent ;
- Rajouter (7, Villeneuve, Denis) dans cette table est possible car ce triplet n'est pas encore présent ;
- Rajouter (1, Villeneuve, Denis) dans cette table est impossible car le triplet est déjà présent ;

Modèle relationnel : exemple de clés, la suite

Notez que les attributs **nom** et **prénom** sont soulignés : la clé primaire est (**nom**, **prénom**).

idP	<u>nom</u>	<u>prénom</u>
1	Villeneuve	Denis
2	Gerwig	Greta
3	Nolan	Christopher
4	DaCosta	Nia

- Rajouter (5, Abrams, J.J.) dans cette table est possible car le tuple (Abrams, J.J.) n'est pas encore présent en tant que clé primaire ;
- Rajouter (6, Steven, Spielberg) dans cette table est possible (même si le prénom n'est pas à la bonne place) car le tuple (Steven, Spielberg) n'est pas encore présent en tant que clé primaire ;
- Rajouter (1, Spielberg, Spielberg) dans cette table est possible car le tuple (Spielberg, Spielberg) n'est pas encore présent en tant que clé primaire ;
- Rajouter (7, Villeneuve, Denis) dans cette table est impossible car le tuple (Villeneuve, Denis) est déjà présent en tant que clé primaire ;
- Rajouter (1, Villeneuve, Denis) dans cette table est impossible car le tuple (Villeneuve, Denis) est déjà présent en tant que clé primaire ;

Modèle relationnel : exemple de clés, la suite

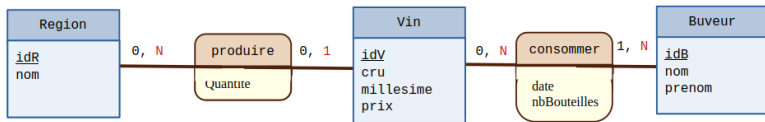
idP (PK)		nom	prénom	idF (PK)		titre	#réalisateur (FK)
1		Villeneuve	Denis	101		Dune : Part Two (2024)	1
2		Gerwig	Greta	102		Barbie (2023)	2
3		Nolan	Christopher	103		Oppenheimer (2023)	3
4		DaCosta	Nia	104		Dune (2021)	1

- Rajouter 101, Dune: Part Two (2024), 1) dans la relation de droite est impossible car ce triplet y est déjà intégralement présent ;
- Rajouter 106, Dune: Part Two (2024), 7) dans la relation de droite est impossible à cause de la clé étrangère qui fait référence à un réalisateur 7 qui n'existe pas ;
- Rajouter 107, Dune: Part Two (2024), 1) dans la relation de droite est possible car la clé primaire (107) n'existe pas encore et la clé étrangère (1) correspond bien à une valeur possible de clé primaire.

Passer du modèle E/A au modèle relationnel

Modèle E/A vers Modèle relationnel

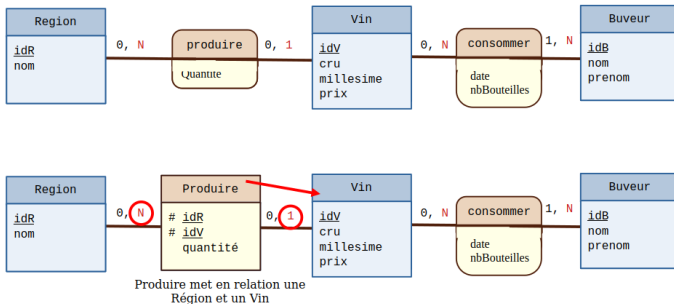
- **Entités** : deviennent toutes des Relations ;
- **Associations** : cela dépend des cardinalités maximales de chaque côté de l'association ;
 - Si on a une cardinalité maximale **1** et **N** : l'association va disparaître, absorbée du côté du **1** ;
 - Si on a deux cardinalités maximales **N** de chaque côté : l'association va devenir une relation.



Ici, produire va disparaître, absorbée par Vin et consommer va devenir une Relation.

E/A vers relationnel (Many to One) ou (N,1)

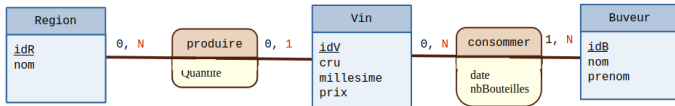
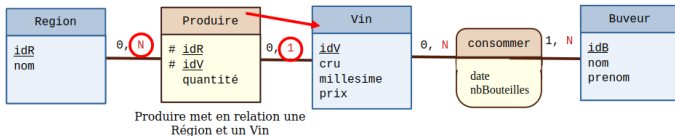
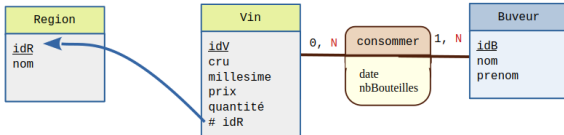
produire a cardinalité max **1** d'un côté : **Vin** l'absorbe car un **Vin** n'a qu'une **Région** à référencer.



- **Vin** va absorber les informations transportées par **produire** : **quantité** et le couple d'informations (quelle région **idR**, pour quel vin **idV**).
- Néanmoins **idV** est déjà présent dans **Vin** : on ne garde que **quantité** et **idR**.
- On obtient une relation **Vin**(idV, cru, millésime, prix, quantité, #idR)
- et la relation **Region**(idR, nom), image directe de l'entité Région.

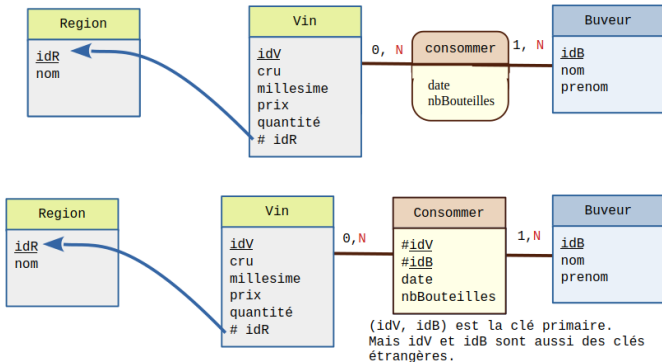
E/A vers relationnel (Many to One) ou (N,1)

produire a cardinalité max **1** d'un côté : **Vin** l'absorbe car un **Vin** n'a qu'une **Région** à référencer.



E/A vers relationnel (Many to Many) ou (N,N)

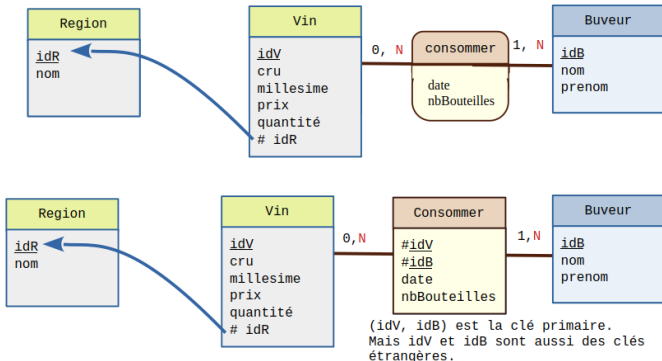
consommer a cardinalité max **N** des deux côtés : elle va devenir une relation.



- L'association **consommer** porte les informations **date**, **nbBouteilles** et le couple d'informations (quel vin **idV**, pour quel buveur de vin **idB**).
- L'association devient la relation **Consommer**(#idV, #idB, **date**, **nbBouteilles**)
- La relation **Buveur**(idB, **nom**, **prenom**) est l'image directe de l'entité.

E/A vers relationnel (Many to Many) ou (N,N)

consommer a cardinalité max **N** des deux côtés : elle va devenir une relation.

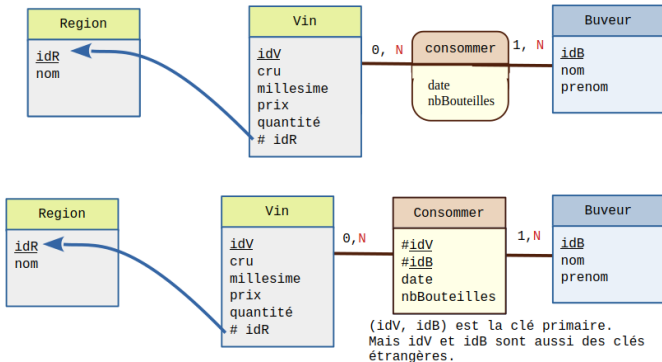


Notez bien ce que signifie **Consommer(#idV, #idB, date, nbBouteilles)** :

- La clé primaire est le couple (idV, idB) : chaque couple ne peut apparaître qu'une fois ;
- L'attribut **idV** pris seul est une clé étrangère vers la relation **Vin** ;
- L'attribut **idB** pris seul est une clé étrangère vers la relation **Buveur**.

E/A vers relationnel (Many to Many) ou (N,N)

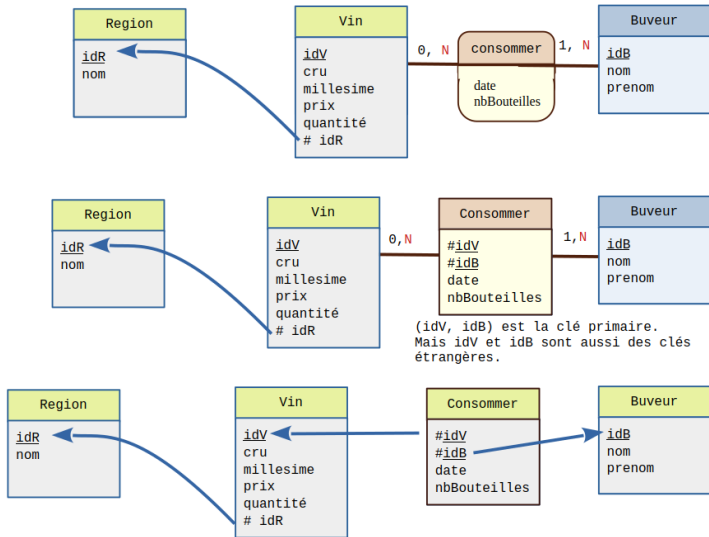
consommer a cardinalité max **N** des deux côtés : elle va devenir une relation.



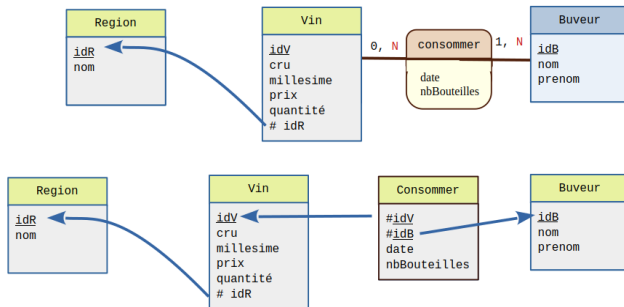
Notez bien ce que signifie **Consommer(#idV, #idB, date, nbBouteilles)** :

- La clé primaire est le couple (idV, idB) : chaque couple ne peut apparaître qu'une fois ;
- Avec ce système, **date** correspondrait par exemple à la date de la dernière dégustation et **nbBouteilles** au nombre total de bouteilles consommées jusqu'à présent.

E/A vers relationnel (Many to Many)



E/A vers relationnel (Many to Many)

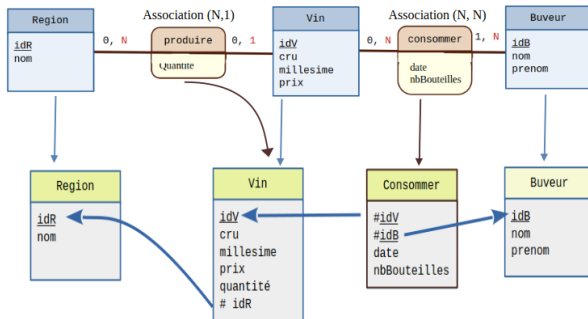


- **Region** (idR, nom)
- **Buveur** (idB, nom, prenom)
- **Vin** (idV, cru, millesime, prix, quantité, #idR)
- **Consommer** (#idV, #idB, date, nbBouteilles)

Modèle E/A vers Modèle relationnel

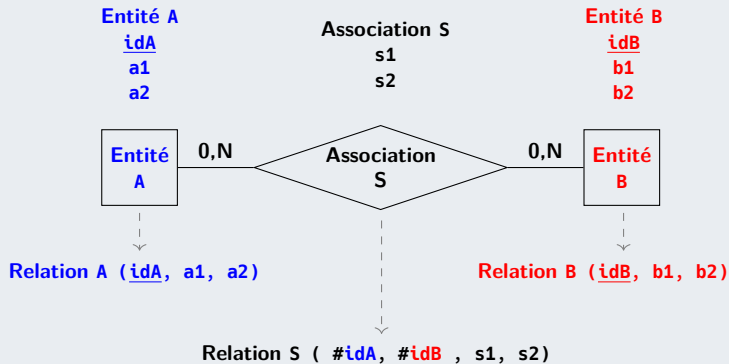
Bilan à connaître par coeur

- 1 Les Entités deviennent des Relations ;
- 2 Une Association Many to One est absorbée par la Relation du côté 1 ;
- 3 Une Association Many to Many devient une Relation : sa clé primaire est composite (composée de clés étrangères référençant les relations qu'elle relie).



Modèle E/A vers Modèle relationnel

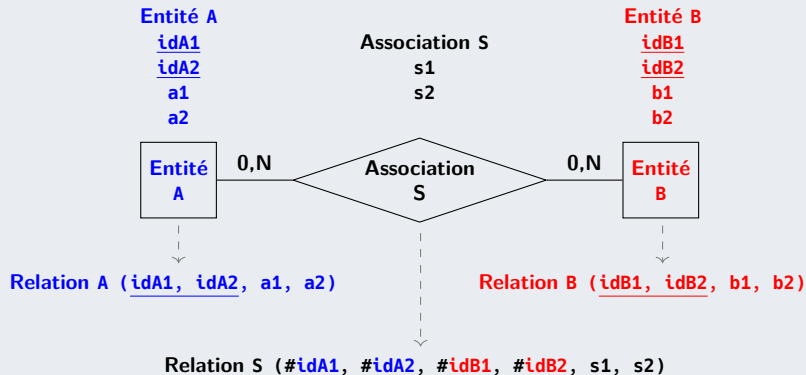
Généralisation : association Many to Many (1pk)



La clé primaire de S est le tuple (idA, idB) et S possède 2 clés étrangères.

Modèle E/A vers Modèle relationnel

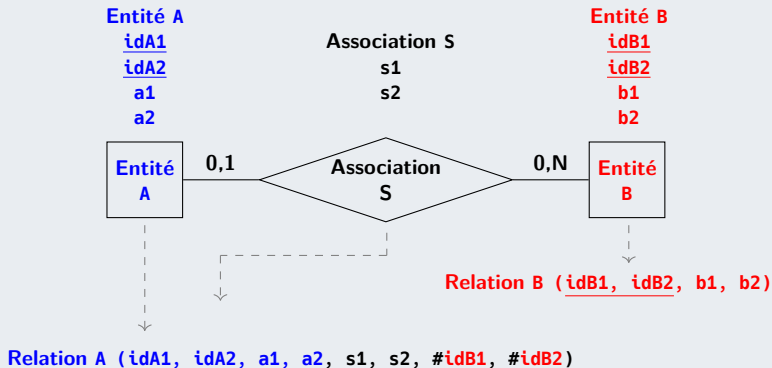
Généralisation : association Many to Many (2pk)



La clé primaire de S est le tuple (idA1, idA2, idB1, idB2) et S possède 4 clés étrangères.

Modèle E/A vers Modèle relationnel

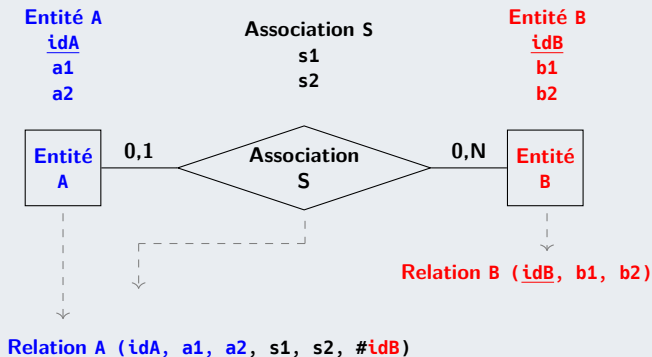
Généralisation : association Many to One (2pk)



La relation A récupère les infos de S (s1 et s2) ainsi que 2 clés étrangères vers B

Modèle E/A vers Modèle relationnel

Généralisation : association Many to One (1pk)



La relation A récupère les infos de S (s1 et s2) ainsi que la clé étrangère vers B