

Module Introduction à la gestion des données

TD 2

Modélisation

Hervé Owsinski
Herve.Owsinski@uphf.fr

Semestre 1



2026-2027

Description globale et vocabulaire à acquérir

Dans cette fiche de TD, nous allons nous focaliser sur une partie des données utiles pour un maraîcher.

Ces données concernent ses commandes et sont stockées dans une base de données relationnelles.

Deux exercices vous permettent de répondre à ces questions :

- 1er exercice : Proposez un **MEA (Modèle Entité-Association)** permettant de répondre au contexte en identifiant les **cardinalités** pour chaque association.
- 2e exercice : Transformez ce **MEA** en **Modèle Relationnel** en précisant les **clés primaires** et les **clés étrangères** de chaque relation.

Le texte descriptif qui sert de base à notre schématisation

Le maraîcher souhaite stocker l'ensemble des produits qu'il propose à la vente. Ces produits sont identifiés de façon unique par un numéro, ils ont un nom, un prix et un descriptif.

Ses clients sont également stockés : ils sont identifiés par un numéro unique, ils ont un nom, une adresse, un numéro de téléphone, une adresse mail et appartiennent à une catégorie de client.

Ces différentes catégories de clients sont représentées par leur nom (unique : par exemple, grossiste, restaurateur, particulier, etc.) et ont chacun un pourcentage maximum de remise qui a été fixé par le maraîcher : par exemple, il n'offrira jamais plus que 10% de remise aux clients particuliers.

Chaque commande est identifiée de façon unique par un numéro de commande, elle aura une date de commande et une date de récupération de la commande. Chaque commande est passée par un client et a un mode de récupération : livraison, retrait, envoi par transporteur etc. Elle correspond à un coût total qui sera précisé ainsi que le mode de paiement utilisé. Pour chaque commande un certain nombre de produits seront commandés avec une quantité spécifiée pour chaque produit.

TD2 Partie I [Cours]

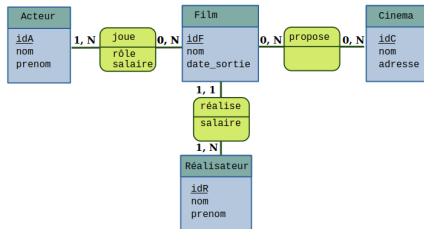
Modèle conceptuel

Modèle Entité - Association

Modèle Entité - Association

Historique et grands principes

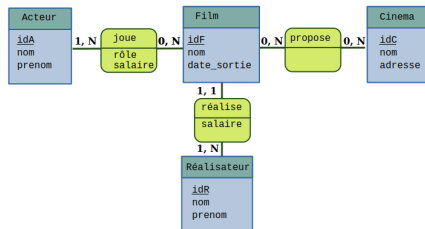
- **Modèle Entité Association** : Peter Chen en **1976** ;
- But : modéliser les besoins de façon abstraite, **indépendamment des technologies** mis en oeuvre ;
- Le **domaine d'application** (la réalité à représenter) est retranscrit sous forme d'un ensemble d'entités reliées entre elles ;
- Plus d'une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



Modèle Entité - Association

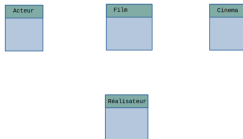
Les concepts de base

- 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.

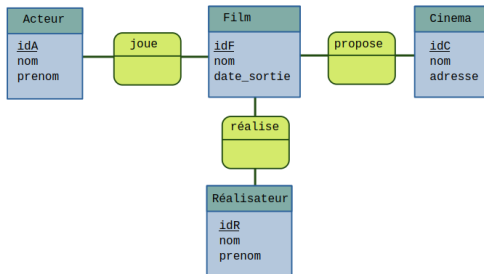


Modèle Entité - Association

Associations

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

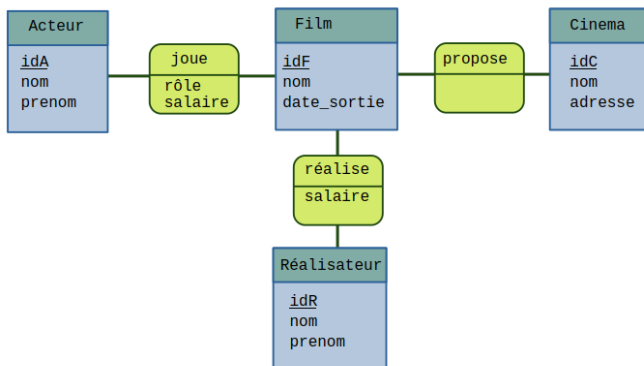
Ces associations sont issues de l'analyse des besoins : "Les films sont **à l'affiche** dans des cinémas. Ils sont **réalisés** par un réalisateur et des acteurs y **jouent** la plupart du temps."



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

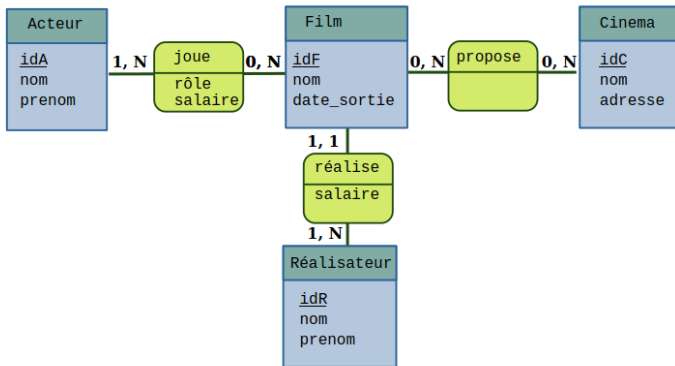
La **cardinalité** exprime le nombre minimum et maximum d'associations d'une entité A vers l'entité B.

- **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

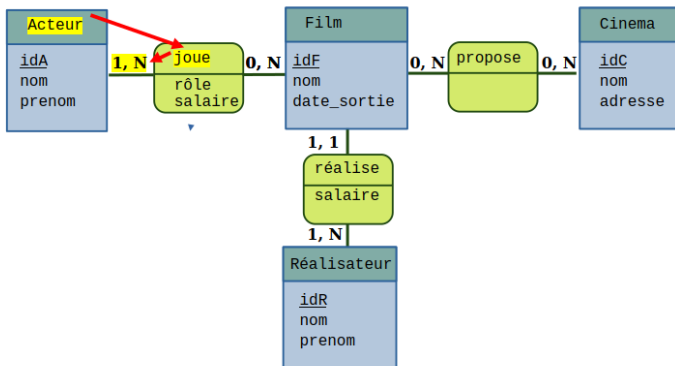
Voici ce qu'on pourrait placer sur notre modèle.



Modèle Entité - Association

Cardinalité

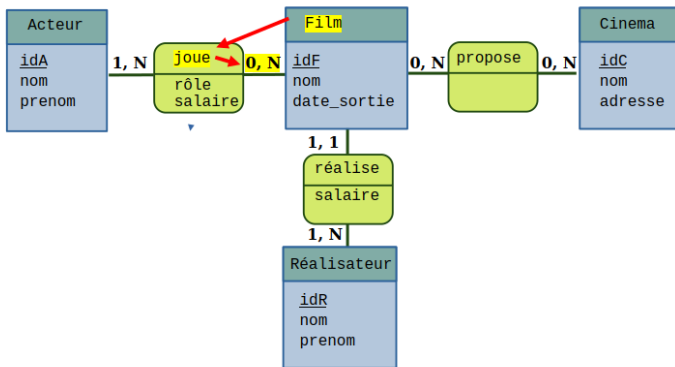
Un acteur peut jouer dans 1 à plusieurs films.



Modèle Entité - Association

Cardinalité

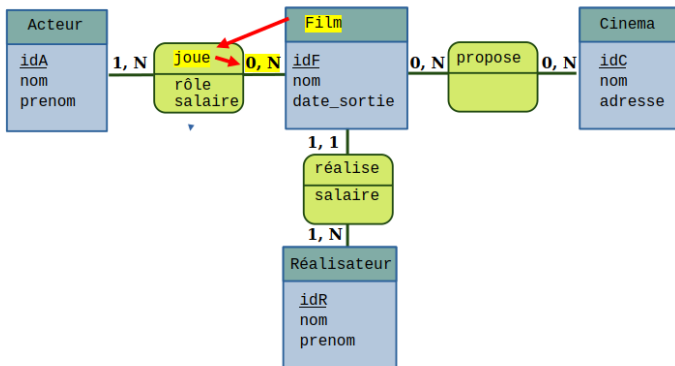
Un film ...



Modèle Entité - Association

Cardinalité

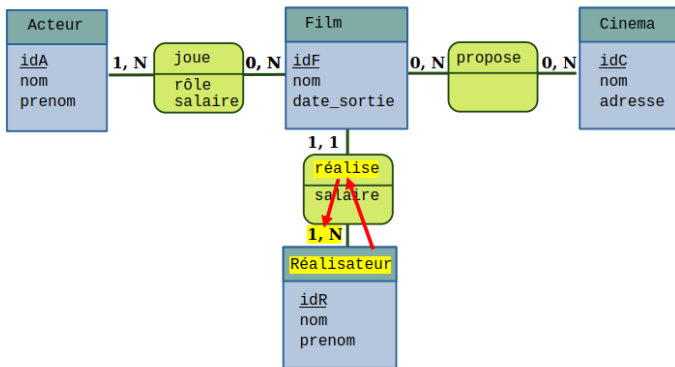
Un film est joué par 0 à plusieurs acteurs.



Modèle Entité - Association

Cardinalité

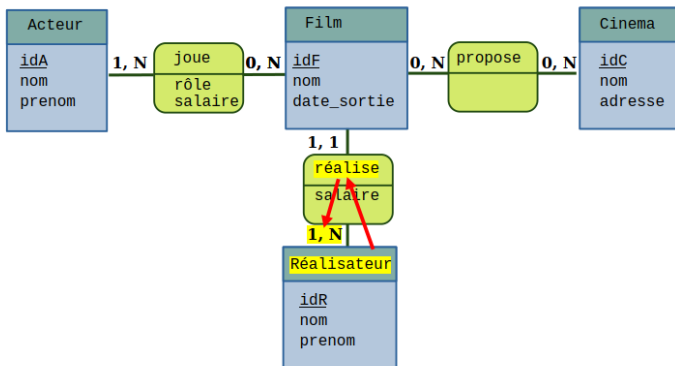
Un ...



Modèle Entité - Association

Cardinalité

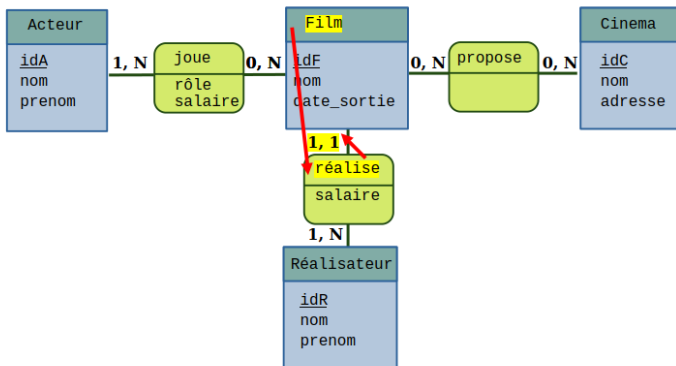
Un réalisateur peut réaliser de 1 à plusieurs films.



Modèle Entité - Association

Cardinalité

Un film ...



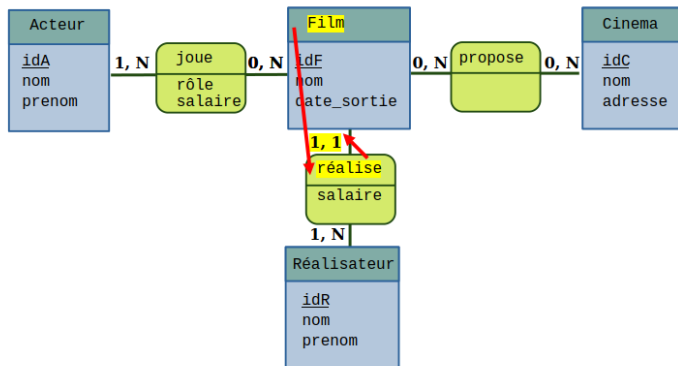
vrai ?

Est-ce toujours

Modèle Entité - Association

Cardinalité

Un film est réalisé par 1 et un seul réalisateur.



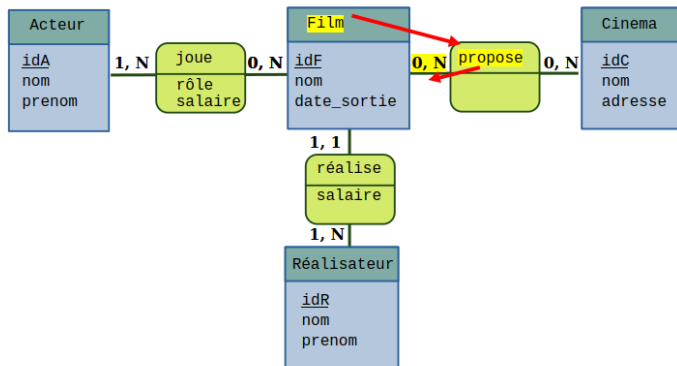
vrai ?

Est-ce toujours

Modèle Entité - Association

Cardinalité

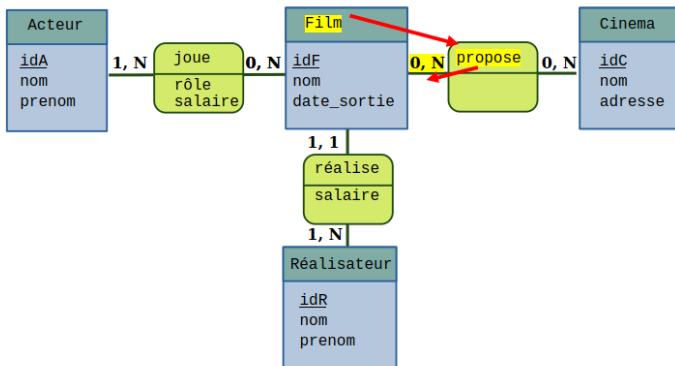
Un film ...



Modèle Entité - Association

Cardinalité

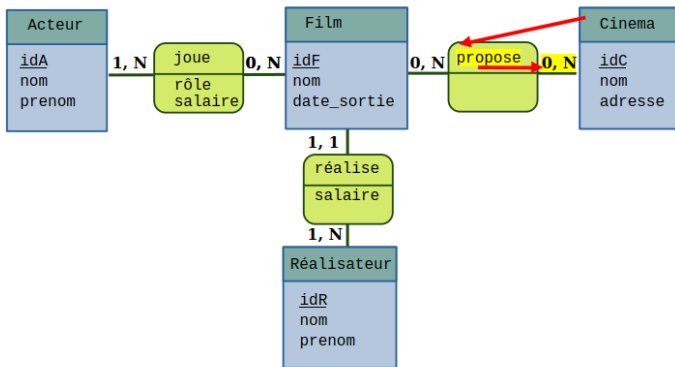
Un film peut être proposé dans 0 à plusieurs cinémas.



Modèle Entité - Association

Cardinalité

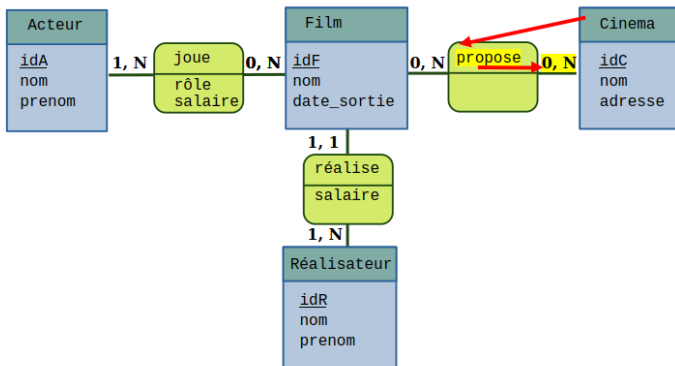
Un cinéma ...



Modèle Entité - Association

Cardinalité

Un cinéma peut proposer de 0 à plusieurs films.



TD2 Partie II

Exercice sur le

Modèle Entité Association

Le texte descriptif qui sert de base à notre schématisation

Le maraîcher souhaite stocker l'ensemble des produits qu'il propose à la vente. Ces produits sont identifiés de façon unique par un numéro, ils ont un nom, un prix et un descriptif.

Ses clients sont également stockés : ils sont identifiés par un numéro unique, ils ont un nom, une adresse, un numéro de téléphone, une adresse mail et appartiennent à une catégorie de client.

Ces différentes catégories de clients sont représentées par leur nom (unique : par exemple, grossiste, restaurateur, particulier, etc.) et ont chacun un pourcentage maximum de remise qui a été fixé par le maraîcher : par exemple, il n'offrira jamais plus que 10% de remise aux clients particuliers.

Chaque commande est identifiée de façon unique par un numéro de commande, elle aura une date de commande et une date de récupération de la commande. Chaque commande est passée par un client et a un mode de récupération : livraison, retrait, envoi par transporteur etc. Elle correspond à un coût total qui sera précisé ainsi que le mode de paiement utilisé. Pour chaque commande un certain nombre de produits seront commandés avec une quantité spécifiée pour chaque produit.

Le texte descriptif qui sert de base à notre schématisation

Le maraîcher souhaite stocker l'ensemble des **produits** qu'il propose à la vente. Ces **produits** sont identifiés de façon unique par un **numéro**, ils ont un **nom**, un **prix** et un **descriptif**.

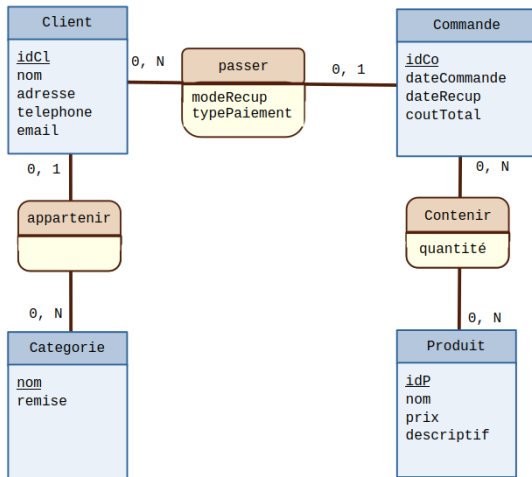
Ses **clients** sont également stockés : ils sont identifiés par un **numéro unique**, ils ont un **nom**, une **adresse**, un numéro de **téléphone**, une adresse **mail** et **appartiennent** à une **catégorie** de client.

Ces différentes **catégories** de clients sont représentées par leur **nom** (unique : par exemple, grossiste, restaurateur, particulier, etc.) et ont chacun un pourcentage maximum de **remise** qui a été fixé par le maraîcher : par exemple, il n'offrira jamais plus que 10% de remise aux clients particuliers.

Chaque **commande** est identifiée de façon unique par un **numéro** de commande, elle aura une **date de commande** et une **date de récupération** de la commande. Chaque **commande est passée** par un **client** et a un **mode** de récupération : livraison, retrait, envoi par transporteur etc. Elle correspond à un **coût total** qui sera précisé ainsi que le mode de paiement utilisé. Pour chaque **commande** un certain nombre de **produits seront commandés** avec une **quantité** spécifiée pour chaque **produit**.

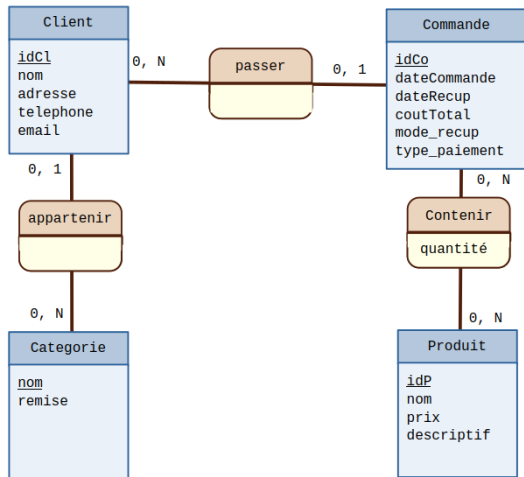
Exemple de schéma Entité Association V1

Ici, on décide de mettre les attributs **modeRecup** **typePaiement** dans l'association.



Exemple de schéma Entité Association V2

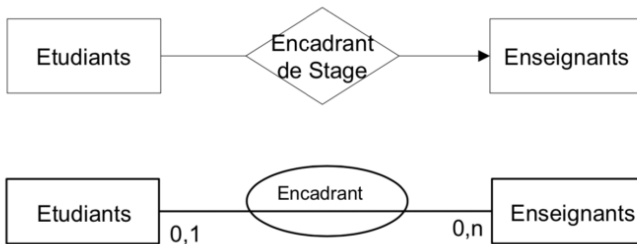
Dans cette version, on les glisse dans l'Entité **Commande**.



Compléments sur le schéma entité-association

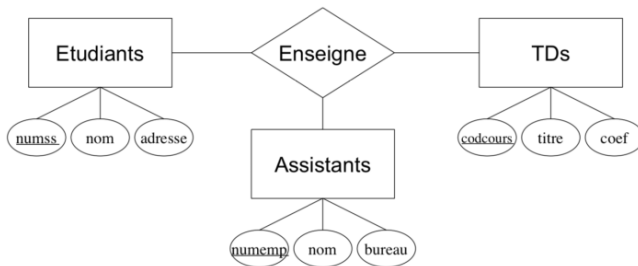
Autres représentations

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



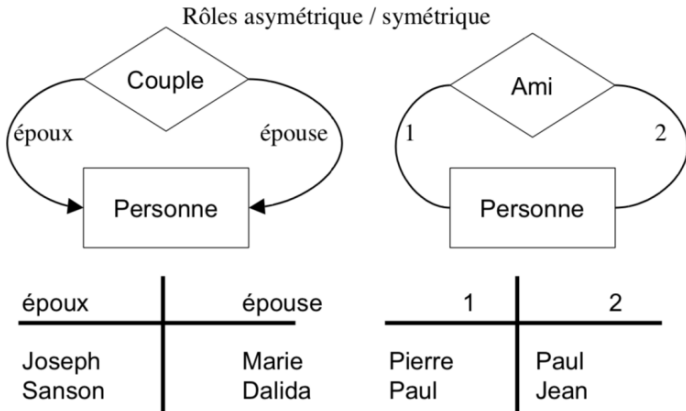
Autres représentations

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

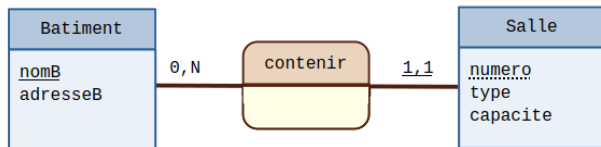
Une association peut relier deux entités de même classe !



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).

Le texte descriptif qui sert de base à notre schématisation

On travaille sur une base de données relative aux modules d'enseignement universitaire.

Chaque étudiant peut suivre plusieurs cours, et chaque cours peut avoir plusieurs étudiants.

Chaque cours est enseigné par un professeur, et chaque professeur appartient à un département.

De plus, chaque étudiant a un ID et un nom, chaque cours a un code et un titre, chaque professeur a un ID et un nom, et chaque département a un ID et un nom. Modélisez ce contexte en utilisant un modèle entité-association.

TD - un dernier exercice du même type

Le texte descriptif qui sert de base à notre schématisation

Nous souhaitons gérer une chaîne d'approvisionnement. Nous tenons une liste de produits et une liste de fournisseurs.

Certains produits peuvent être proposés par plusieurs fournisseurs, à des prix qui peuvent être différents.

Nous garderons l'historique des livraisons en mémoire, avec pour chacune d'entre elle, le fournisseur qui l'a effectué, la date de livraison ainsi que le détail des produits livrés.

TD2 Partie III [Cours]

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 rassemble les données sous forme de **tuples ayant des propriétés communes**.

Qu'est-ce qu'une relation ?

Définitions

def Une **relation** est un **ensemble** dont les éléments sont des **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.
Le schéma spécifie

- le **nom de la relation** et
- **position-nom-domaine** de chaque attribut ;

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

- La plupart du temps, on ne note pas le type sur le schéma :
Exemple 1 : **Personne** (idP, nom, note)
- D'ailleurs, beaucoup de conventions de nommage existent :
Exemple 2 : **PERSONNE** (IdP, Nom, Note)
Exemple 3 : **PERSONNE** (IDP, NOM, NOTE)

...

Qu'est-ce qu'une relation ?

Définitions

def Une **relation** est un **ensemble** dont les éléments sont des **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.
Le schéma spécifie

- le **nom de la relation** et
- **position-nom-domaine** de chaque attribut ;

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

- La plupart du temps, on ne note pas le type sur le schéma :
Exemple 1 : **Personne** (idP, nom, note)
- D'ailleurs, beaucoup de conventions de nommmage existent :
Exemple 2 : **PERSONNE** (IdP, Nom, Note)
Exemple 3 : **PERSONNE** (IDP, NOM, NOTE)

...

Qu'est-ce qu'une relation ?

Définitions

def Une **relation** est un **ensemble** dont les éléments sont des **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.
Le schéma spécifie

- le **nom de la relation** et
- **position-nom-domaine** de chaque attribut ;

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

- La plupart du temps, on ne note pas le type sur le schéma :
Exemple 1 : **Personne** (idP, nom, note)
- D'ailleurs, beaucoup de conventions de nommage existent :
Exemple 2 : **PERSONNE** (IdP, Nom, Note)
Exemple 3 : **PERSONNE** (IDP, NOM, NOTE)

...

Qu'est-ce qu'une relation ?

Définitions

def Une **relation** est un **ensemble** dont les éléments sont des **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.
Le schéma spécifie

- le **nom de la relation** et
- **position-nom-domaine** de chaque attribut ;

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

- La plupart du temps, on ne note pas le type sur le schéma :
Exemple 1 : **Personne** (idP, nom, note)
- D'ailleurs, beaucoup de conventions de nommmage existent :
Exemple 2 : **PERSONNE** (IdP, Nom, Note)
Exemple 3 : **PERSONNE** (IDP, NOM, NOTE)

...

Quelques précisions

- Exemple de relation : **Personne** (idP, nom, note)
- Exemple de tuples : $\{ (5, 'alice', 12.3), (6, 'bob', 9.3) \}$
- Si on représente cet ensemble plutôt comme cela, la notion de tableau 2D semble évidente :
$$\begin{Bmatrix} (5, 'alice', 12.3), \\ (6, 'bob', 9.3) \end{Bmatrix}$$

Quelques précisions

- Exemple de relation : **Personne** (idP, nom, note)
- Exemple de tuples : $\{ (5, 'alice', 12.3), (6, 'bob', 9.3) \}$
- Si on représente cet ensemble plutôt comme cela, la notion de tableau 2D semble évidente :
$$\begin{Bmatrix} (5, 'alice', 12.3), \\ (6, 'bob', 9.3) \end{Bmatrix}$$

Intention et Extension

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

Quelques précisions

- Exemple de relation : **Personne** (idP, nom, note)
- Exemple de tuples : $\{ (5, 'alice', 12.3), (6, 'bob', 9.3) \}$
- Si on représente cet ensemble plutôt comme cela, la notion de tableau 2D semble évidente :
$$\begin{Bmatrix} (5, 'alice', 12.3), \\ (6, 'bob', 9.3) \end{Bmatrix}$$

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

On peut représenter ce schéma et son extension de cette façon :

- **Ligne** du tableau correspond à **tuple** de la relation.
- **Colonne** du tableau correspond à un **attribut** de la relation.

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

Découverte : 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}

Découverte : 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 : idE 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 relation ne possède pas de clé primaire mais reste un ensemble : son triplet est une sorte de PK :

Un étudiant ne pourra jamais s'inscrire deux fois au même cours la même année.

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**.
- 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.

$$\text{Valeurs}(FK) \subseteq \text{Valeurs}(PK)$$

- Une même valeur peut **apparaître plusieurs fois** dans une clé étrangère.
- Les attributs FK sont **préfixés par #**.

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 ? ;
- Rajouter (6, Steven, Spielberg) dans cette table est ? ;
- Rajouter (1, Spielberg, Steven) dans cette table est ? ;
- Rajouter (7, Villeneuve, Denis) dans cette table est ? ;
- Rajouter (1, Villeneuve, Denis) dans cette table est ? ;

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 ? ;
- Rajouter (6, Steven, Spielberg) dans cette table est ? ;
- Rajouter (1, Spielberg, Steven) dans cette table est ? ;
- Rajouter (7, Villeneuve, Denis) dans cette table est ? ;
- Rajouter (1, Villeneuve, Denis) dans cette table est ? ;

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 attribut **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 ?;
- Rajouter (6, Steven, Spielberg) dans cette table est ?;
- Rajouter (1, Spielberg, Steven) dans cette table est ?;
- Rajouter (7, Villeneuve, Denis) dans cette table est ?;
- Rajouter (1, Villeneuve, Denis) dans cette table est ?;

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 ???;
- Rajouter 106, Dune: Part Two (2024), 7) dans la relation de droite est ???;
- Rajouter 107, Dune: Part Two (2024), 1) dans la relation de droite est ???;

Modèle relationnel : exemple de clés, la suite

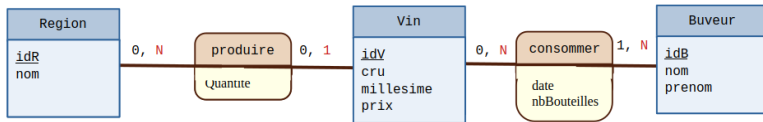
idP (PK)		nom	prénom	idF (PK)		titre	#réalisateur (FK)
1		Villeneuve	Denis	101		Dune : Part Two (2024)	1
2		Gerwig	Greta	102		Barbie (2023)	2
3		Nolan	Christopher	103		Oppenheimer (2023)	3
4		DaCosta	Nia	104		Dune (2021)	1

- 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

- **Entité** : devient une Relation lors du passage E/A vers Relation ;
- **Association** : dépend des cardinalités max. de chaque côté ;
 - Cardinalités max. **1** et **N** : association disparaît, absorbée du côté du **1** ;
 - Cardinalités max. **N** et **N** : association devient une Relation.

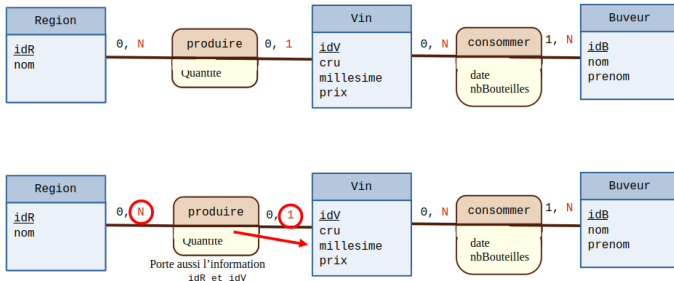


produire : cardinalités max **N** et **1** :
elle disparaît, ses informations absorbées par **Vin**.

produire : cardinalités max **N** et **N** :
elle va devenir une Relation.

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

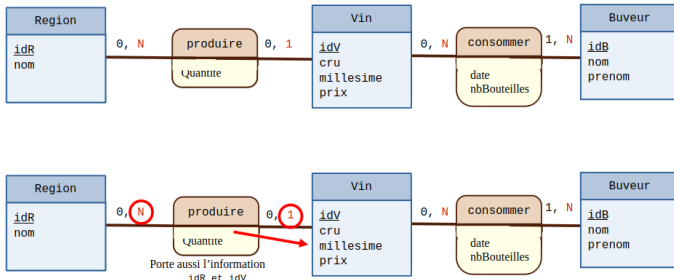
produire : cardinalité max **1** et **N** : One to Many.



- **Vin** absorbe les informations transportées par **produire** :
quantité et le couple d'informations (quelle région **idR**, pour quel vin **idV**).
idV déjà présent dans **Vin** : on ne garde que **quantité** et **idR**.
- Relation **Vin**(idV, cru, millesime, prix, **quantité**, **#idR**)
- Relation **Region**(idR, nom)

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

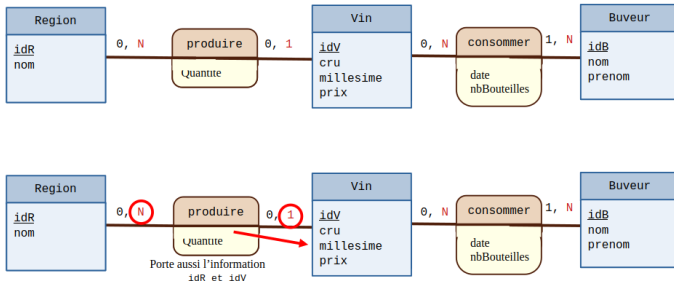
produire : cardinalité max **1** et **N** : One to Many.



- **Vin** absorbe les informations transportées par **produire** :
quantité et le couple d'informations (quelle région **idR**, pour quel vin **idV**).
idV déjà présent dans **Vin** : on ne garde que **quantité** et **idR**.
- Relation **Vin**(idV, cru, millésime, prix, **quantité**, **#idR**)
- Relation **Region**(idR, nom)

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

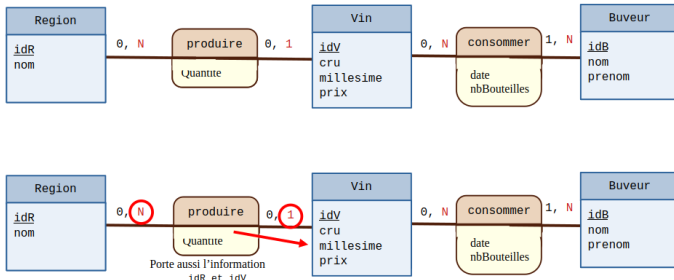
produire : cardinalité max **1** et **N** : One to Many.



- **Vin** absorbe les informations transportées par **produire** :
quantité et le couple d'informations (quelle région **idR**, pour quel vin **idV**).
idV déjà présent dans **Vin** : on ne garde que **quantité** et **idR**.
- Relation **Vin**(idV, cru, millésime, prix, quantité, #idR)
- Relation **Region**(idR, nom)

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

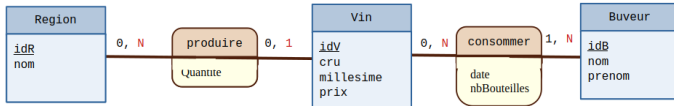
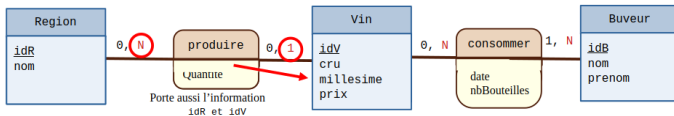
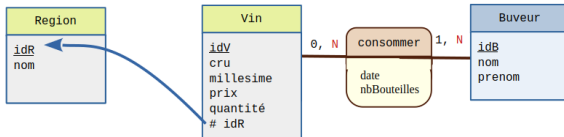
produire : cardinalité max **1** et **N** : One to Many.



- **Vin** absorbe les informations transportées par **produire** :
 quantité et le couple d'informations (quelle région **idR**, pour quel vin **idV**).
 idV déjà présent dans **Vin** : on ne garde que **quantité** et **idR**.
- Relation **Vin**(idV, cru, millésime, prix, **quantité**, **#idR**)
- Relation **Region**(idR, nom)

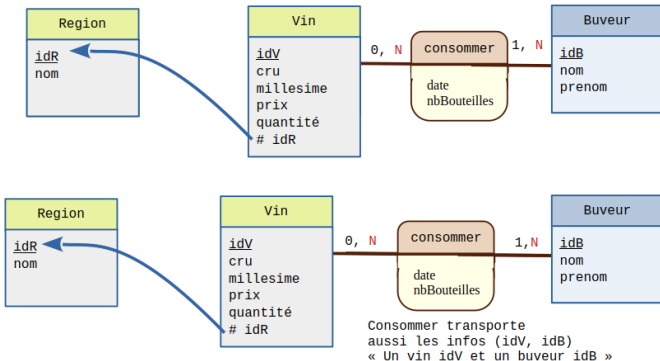
E/A vers relationnel (Many to One) ou (N,1)

BILAN : **produire** a des cardinalités max **1** et **N** → absorbée par **Vin**.



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

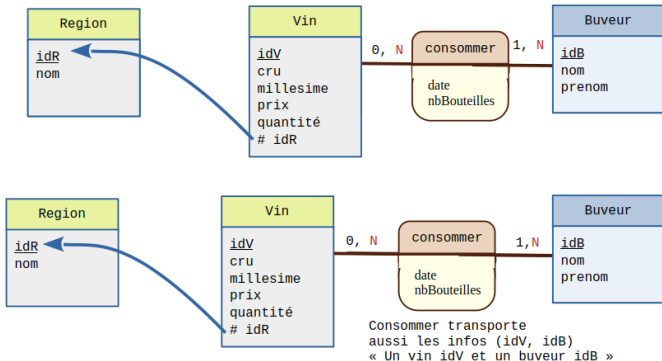
consommer : cardinalités max **N** et **N** → devient une relation.



- Association **consommer** porte les informations **date**, **nbBouteilles** et le couple d'informations (quel vin **idV**, pour quel buveur de vin **idB**).
- Relation **Consommer**(#idV, #idB, date, nbBouteilles)
- Relation **Buveur**(idB, nom, prenom)

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

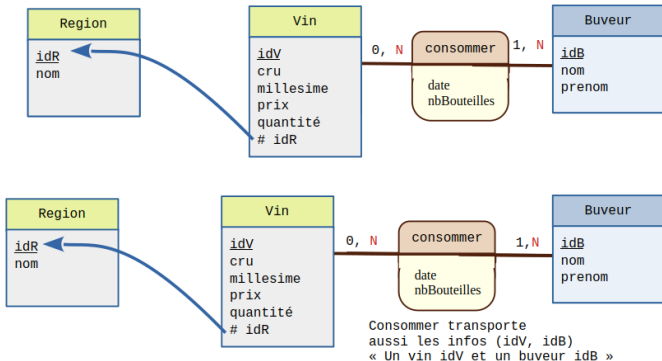
consommer : cardinalités max **N** et **N** → devient une relation.



- Association **consommer** porte les informations **date**, **nbBouteilles** et le couple d'informations (quel vin **idV**, pour quel buveur de vin **idB**).
- Relation **Consommer**(#idV, #idB, date, nbBouteilles)
- Relation **Buveur**(idB, nom, prenom)

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

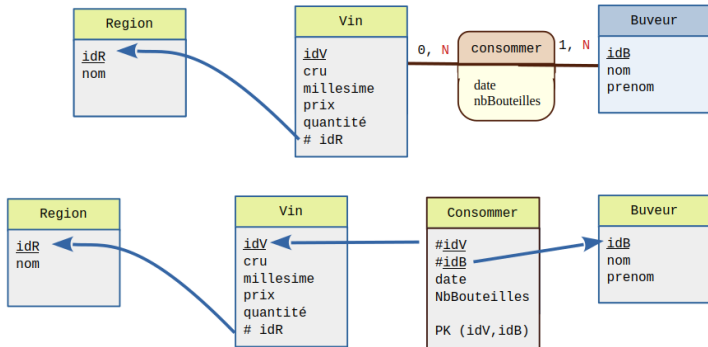
consommer : cardinalités max **N** et **N** → devient une relation.



- Association **consommer** porte les informations **date**, **nbBouteilles** et le couple d'informations (quel vin **idV**, pour quel buveur de vin **idB**).
- Relation **Consommer**(#idV, #idB, date, nbBouteilles)
- Relation **Buveur**(idB, nom, prenom)

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

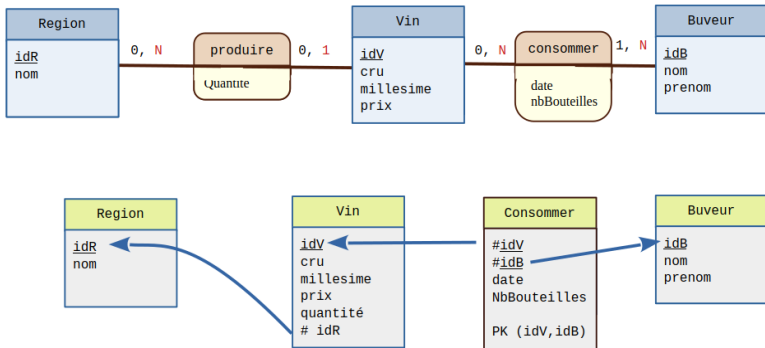
consommer : cardinalités max **N** et **N** → devient une relation.



A COMPRENDRE : **Consommer**(#idV, #idB, date, nbBouteilles) :

- Couple (idV, idB) : clé primaire, chaque valeur du couple est donc UNIQUE ;
- Attribut **idV** : clé étrangère vers la relation **Vin** ;
- Attribut **idB** : clé étrangère vers la relation **Buveur**.

E/A vers relationnel (Many to Many)

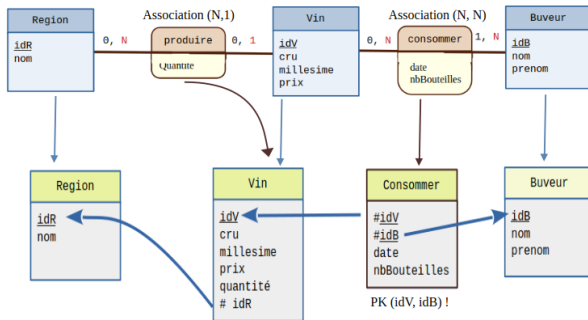


- **Region** (idR, nom)
- **Buveur** (idB, nom, prenom)
- **Vin** (idV, cru, millesime, prix, quantite, #idR)
- **Consommer** (#idV, #idB, date, nbBouteilles)

Modèle E/A vers Modèle relationnel

Bilan à connaître par coeur

- 1 Entité → Relation ;
- 2 Association Many to One → disparaît, infos absorbées du côté 1 ;
- 3 Association Many to Many → Relation,
clé primaire composée de clés des relations qu'elle relie.



TD2 Partie IV

Exercice

EA vers Relationnel

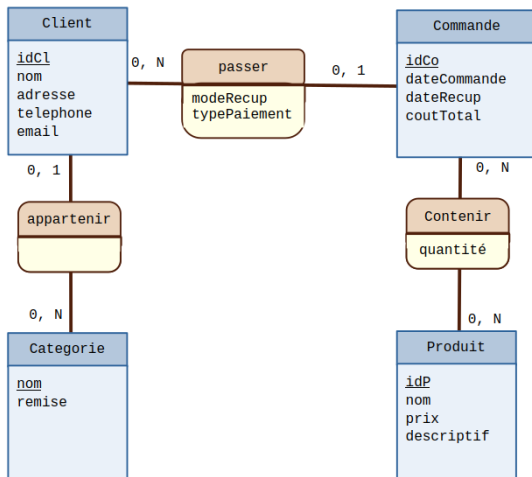
Rappel de l'énoncé

Dans cette fiche de TD, nous allons nous focaliser sur une partie des données utiles pour le maraîcher.

Ces données concernent ses **commandes** et sont stockées dans une **base de données relationnelles**.

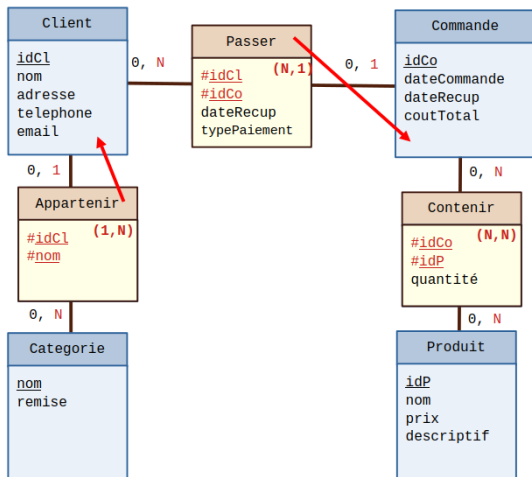
- 1er séance : Proposez un **modèle entité-association** permettant de répondre au contexte ci-dessus en identifiant les **cardinalités** pour chaque association.
- **2nd séance : Transformez ce modèle entité-association en modèle relationnel en précisant les clés primaires et les clés étrangères de chaque relation.**

On décide de prendre la version où l'attribut retrait est dans l'association.



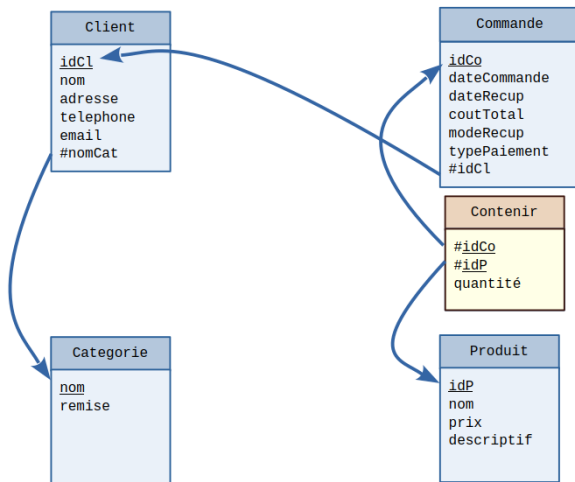
Les entités vont devenir des relations
mais seules certaines associations vont devenir des relations.

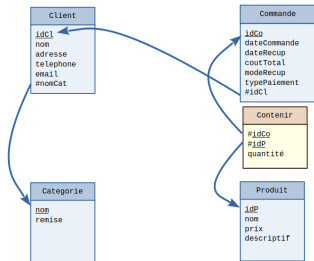
Mentalement, on transforme tout en relation puis on regarde comme agir en réalité :



Notez que nous allons devoir changer le titre de 'nom' de Appartenir car il existe déjà un attribut 'nom' dans Client. Prenons 'nomCat'.

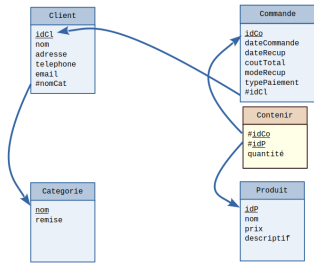
Et au final, nous obtenons ceci :





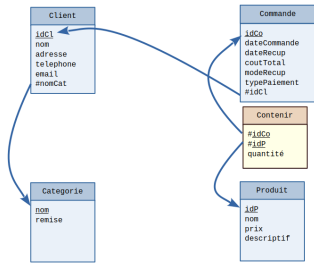
Reste à fournir le schéma relationnel sous la forme textuelle, en commençant par les relations qui ne contiennent pas de clés étrangères vers une autre table :

- Categorie (nom, remise)
- Produit (idP, nom, prix, descriptif)
- Client (idCl, nom, adresse, telephone, email, #nomCat)
- Commande (idCo, dateCommande, dateRecup, coutTotal, modeRecup, typePaiement, #idCl)
- Contenir (#idCo, #idP, quantite)



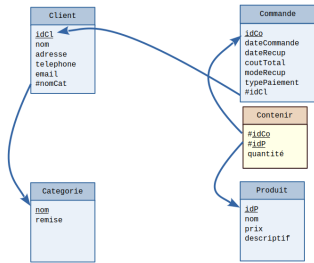
Reste à fournir le schéma relationnel sous la forme textuelle, en commençant par les relations qui ne contiennent pas de clés étrangères vers une autre table :

- Categorie (nom, remise)
- Produit (idP, nom, prix, descriptif)
- Client (idCl, nom, adresse, telephone, email, #nomCat)
- Commande (idCo, dateCommande, dateRecup, coutTotal, modeRecup, typePaiement, #idCl)
- Contenir (#idCo, #idP, quantite)



Reste à fournir le schéma relationnel sous la forme textuelle, en commençant par les relations qui ne contiennent pas de clés étrangères vers une autre table :

- Categorie (nom, remise)
- Produit (idP, nom, prix, descriptif)
- Client (idCl, nom, adresse, telephone, email, #nomCat)
- Commande (idCo, dateCommande, dateRecup, coutTotal, modeRecup, typePaiement, #idCl)
- Contenir (#idCo, #idP, quantite)



Reste à fournir le schéma relationnel sous la forme textuelle, en commençant par les relations qui ne contiennent pas de clés étrangères vers une autre table :

- Catégorie (nom, remise)
- Produit (idP, nom, prix, descriptif)
- Client (idCl, nom, adresse, telephone, email, #nomCat)
- Commande (idCo, dateCommande, dateRecup, coutTotal, modeRecup, typePaiement, #idCl)
- Contenir (#idCo, #idP, quantite)