

Plan du document

- **Modèle relationnel**
- **Passage entité association au relationnel**
- **Opérateurs de l'Algèbre Relationnelle**
 - Opérateurs de l'algèbre relationnelle
 - Union
 - Intersection
 - Différence
 - Restriction
 - Projection
 - Jointure
 - Division
 - Produit cartésien
 - Les arbres algébriques



Partie 1 : Le modèle relationnel

Base de données et le modèle relationnel

- Un SGBD est fondé sur un modèle de données
 - ➔ définir la structure de données
 - ➔ définir les liens entre ces données

SGBDR == > utilisation du modèle relationnel

le modèle relationnel

- Le modèle relationnel est basé sur la théorie des relations dérivées de la théorie des ensembles
- Le modèle relationnel a été formalisé par CODD en 1970.
- Exemple d'implantation: SQL-Server, Sybase, ORACLE ...
- Repose sur l'algèbre relationnelle
 - possède des bases solides
- Concepts puissants et simples !
 - succès !
- Dans ce modèle, les données sont stockées dans des tables
- Un ensemble de données sera donc modélisé par un ensemble de tables.

le modèle relationnel

- Les données sont organisées en tables 2D
 - Une table 2D = une « relation »
 - Une ligne = un n-uplet = un « tuple » = (enregistrement)
- Données manipulées uniquement par des opérateurs de l'algèbre relationnelle
- L'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.
- Au modèle relationnel est associée la théorie de la normalisation des relations qui permet de se débarrasser des incohérences au moment de la conception d'une base de données.


modèle relationnel

- Les objets du monde réel sont modélisés par :

Relations / tables

Attributs / champs

Domaines / types



UV	CodeUv	coordonateur
	SE	Arneur
	BD	Ben Salah

- Plus simplement, une relation est un tableaux à deux dimension caractérisé par un nom
- attribut (ou un champ) : nom associé à chaque colonne afin de la repérer indépendamment de l'ordre.
 - ➔ Décrit une information stockée dans une base

Relation

Attribut = nom d'une colonne

- Une ligne est un tuple (n-uplet) correspond à un enregistrement

UV	CodeUv	coordinateur
	SE	Ameur
	BD	Ben Salah

- Chaque relation a un schéma (schéma relationnel), pour la décrire

Relation

- Le degré d'une relation : c'est son nombre d'attributs
- La cardinalité d'une relation : c'est son nombre d'occurrence

Notion de Domaine

■ Définition :

Le type de valeur ou domaine d'un attribut est un ensemble de valeurs qu'on peut lui associer. Il est caractérisé par un nom.

La spécification d'un type de valeur est donnée soit:

- par la liste des valeurs
- par la propriété que doivent vérifier les valeurs

Exemples:

- entier, réel, chaîne de caractères, booléen
- couleur = {rouge, bleu, jaune, vert}
- Numéro étudiant = entier
- nom = chaîne de caractères alphabétiques

Produit cartésien

■ Définition

- Le produit cartésien de D_1, \dots et D_n est l'ensemble des n-uplets (tuples) $\langle V_1, \dots, V_n \rangle$ tel que V_i appartient à D_i

■ Notation

- $D_1 \times \dots \times D_n$

■ Exemple :

$D_1 = \{ 'BD', 'SE' \} \quad / \quad (\text{CodeUV})$
 \times

$D_2 = \{ 'Ben Salah', 'Ameur' \} \quad / \quad (\text{Coordinateur})$



$D_1 \times D_2$	D_1	D_2
	BD	Ben Salah
	BD	Ameur
	SE	Ameur
	SE	Ben Salah

Produit cartésien et relation

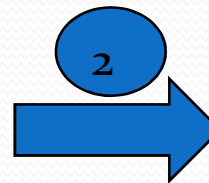
- Une relation est un sous-ensemble du produit cartésien de domaines caractérisé par un nom

Exemple : La relation UV formée à partir du sous-ensemble du produit cartésien de

- $D_1 = \text{CodeUV}$
- et de
- $D_2 = \text{Coordinateur}$

1

$D_1 \times D_2$	D_1	D_2
	BD	Ben Salah
	BD	Ameur
	SE	Ameur
	SE	Ben Salah



Relation UV

UV	D_1	D_2
	SE	Ameur
	BD	Ben Salah

« Les seules lignes qui ont un sens dans le cadre de mon problème »

Schéma de relation

- **Définition** : C'est la description d'une relation.
- Le schéma d'une relation est composé de
 - Son nom
 - La liste des attributs qu'elle comporte (et éventuellement leurs domaines)
 - La liste des attributs composant la clé (la clé est soulignée)

- **Exemple de schéma relationnel**

Etudiant(num : entier, nom : chaîne, adresse : chaîne, âge : entier de 18 à 35)
ou Etudiant(num, nom, adresse, âge)



La table en extension
correspond à la liste
des tuples (ou enregistrements)
== > une table : est la représentation
d'une relation sous forme
de tableau

Étudiant	Num	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tata	RU ₂
	3	Toto	RU ₁

Schéma de relation

- Le schéma de la BD relationnelle est l'ensemble des schémas des relations (tables) composantes et un ensemble de constraints d'intégrité (CI)

Exemple de tables:

films

Titre	Num_realisateur	Année
Alien	1	1979
Vertigo	2	1958
Psychose	3	1960
Kagemusha	4	1980
Volte-face	5	1997
Pulp Fiction	6	1995
Titanic	7	1997
Sacrifice	8	1986

Realisateur

Num_realisateur	nom	prenom	dateNaissance
1	Scott	Ridley	1943
2	Hitchcock	Alfred	1899
3	Kurosawa	Akira	1910
4	Woo	John	1946
5	Tarantino	Quentin	1963
6	Cameron	James	1954
7	Tarkovski	Andrei	1932

Schéma relationnel:

Films(Titre, Num_realisateur, année)

Realisateur(Num_realisateur, nom, prenom, dateNaissance)

■ Définition

Une clé candidate est l'ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un tuple unique dans une relation/table (à tout instant)

- **Clé primaire** : clé choisit parmi les clés candidate et servira d'identification privilégiée des n-uplets (enregistrement). Pour signaler la clé primaire on la souligne (souligner les attributs qui la compose).

■ exemple 1 :

- Clé de Étudiant : **Etudiant**(NumEtudiant, nom, adresse, age)

■ exemple 2: **Salarie**(numSécu, nom, prenom, DateNaissance, matricule)

→ **clés candidates** : numsecu, matricule

Si je choisi numSéc comme clé primaire de ma relation alors on aura :

Salarie(numSécu, nom, prenom, DateNaissance, matricule)

■ Définition

Une relation peut faire référence à une autre en utilisant une **clé étrangère**, qui correspond à la **clé primaire** de la relation référencée
== > c'est un attribut qui est clé primaire d'une autre relation

exemple :

Relaisateurs(numRéalisateur, nom, prénom, dateNaissane)



Filmes(numfilme, #numRéalisateur, titre, année)

■ Rôle

- Les clés étrangères définissent des contraintes d'intégrité **référentielle entre relations**

Les Contraintes d'Intégrité

■ Définition :

Ensemble de règles permettant de garantir la cohérence du schéma relationnel d'une base de données

■ Objectif :

contrôler la cohérence des valeurs prises par :

* Contraintes d'entité :

impose que toute table possède une clé primaire. Cette clé doit être unique (pas de doublons) et non nulle (toujours spécifiées).

ps : si la clé primaire est composée de plusieurs attributs les valeurs de ces attributs doivent être non nulle (null = valeur indéterminée)

Les Contraintes d'Intégrité

■ Objectif (suite) :

* contraintes d'intégrité référentielles :

Contrainte portant sur une relation S, consistant à imposer que la valeur d'un attribut ou groupe d'attributs apparaisse comme valeur de clé dans une autre relation R

* contrainte de domaine :

Les valeurs d'un attribut restent dans son domaine

Exemple : Si l'attribut ' N° Client ' est défini sur un domaine de valeurs numériques , il ne peut pas contenir de lettres

.



REGLES DE PASAGE ENTITE ASSOCIATION (EA) AU MODELE RELATIONNEL

Passage EA au relation

Lien entre le modèle Entité Association et le modèle relationnel :

Modèle EA	Modèle relationnel
Entité, Association Propriété Identifiant	Relation (table) Attribut Clé Primaire

Passage EA au relation

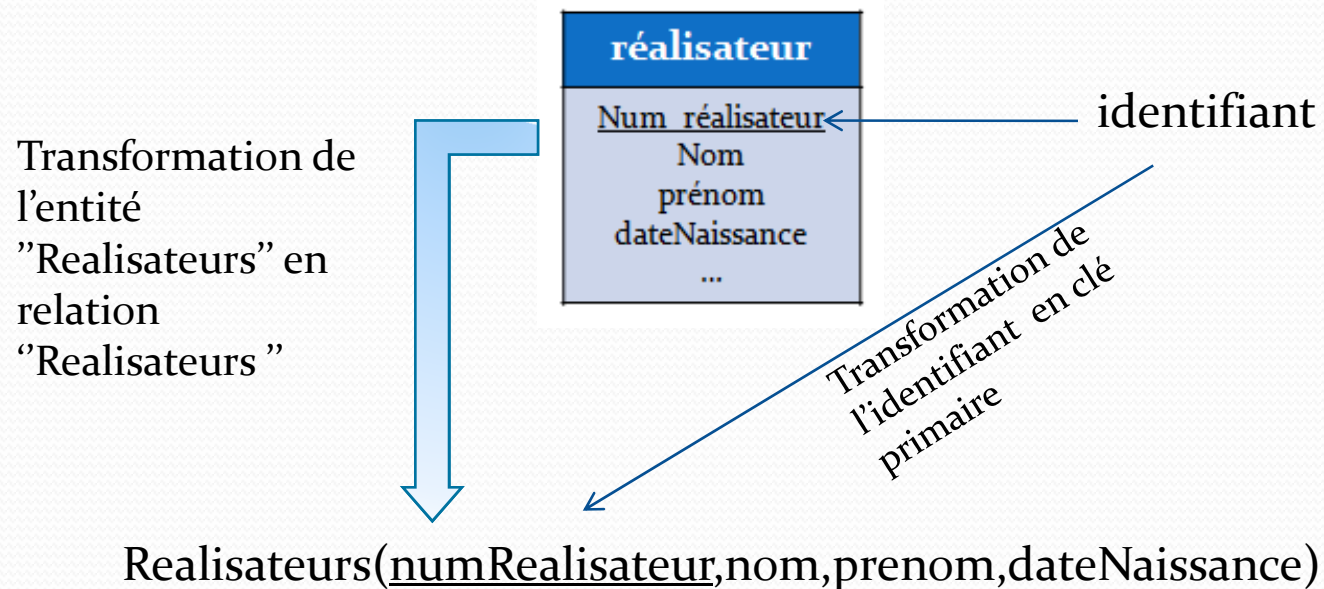
Pour obtenir le modèle relationnel à partir du modèle des entités association, il suffit d'appliquer les règles suivantes :

- Une entité devient une relation (une table),
- Une association devient éventuellement une relation, en fonction de son degré et des cardinalités maximales.

Passage EA au relation

- Transformation de l'entité :

Toute entité devient une relation ayant pour clé primaire son identifiant.
Chaque attribut de l'entité se transforme en attribut de la relation.



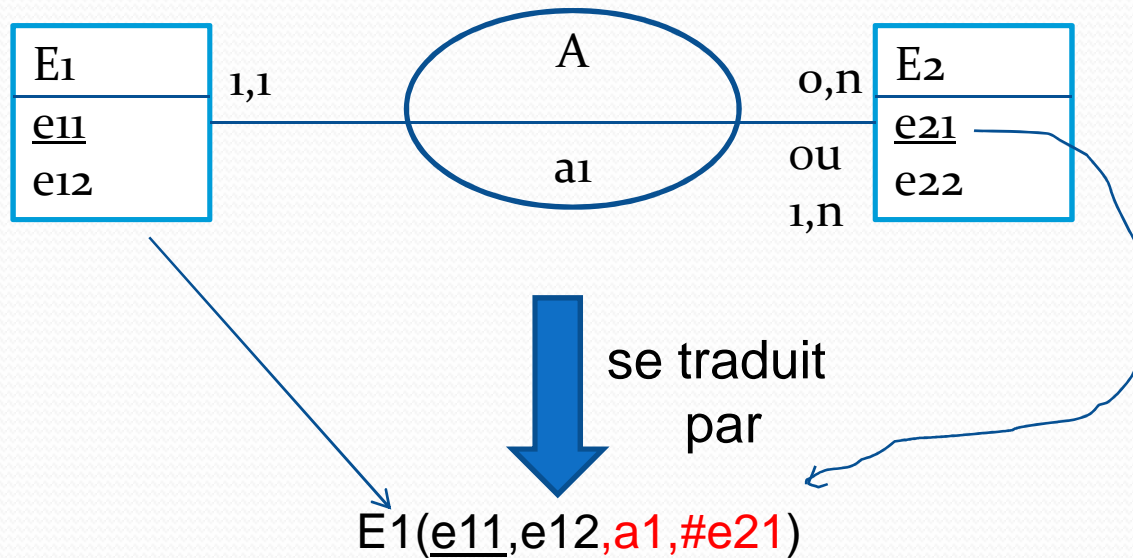
Passage EA au relation

- Transformation d'une association:

Plusieurs cas sont à distinguer selon le degré de la relation et la valeur des cardinalités maximales (1 ou N).

Passage EA au relation

- **Cas E1 (1,1) ou (0,1) – E2 (0-n) ou (1-n)**
l'identifiant de E2 ainsi que les attributs de l'association sont ajoutés à E1.



On aura deux relations (tables) :

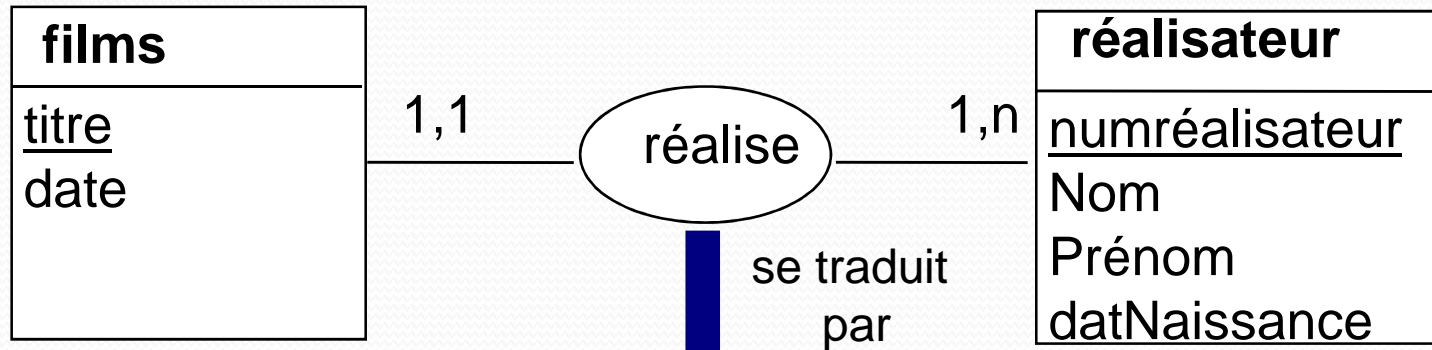
E1(e11, e12, a1, #e21)

E2(e21, e22)

- La clé primaire de E2 (**e21**) devient une clé étrangère dans E1.

Passage EA au relation

Exemple : Association binaire (1,1) - (1,n)



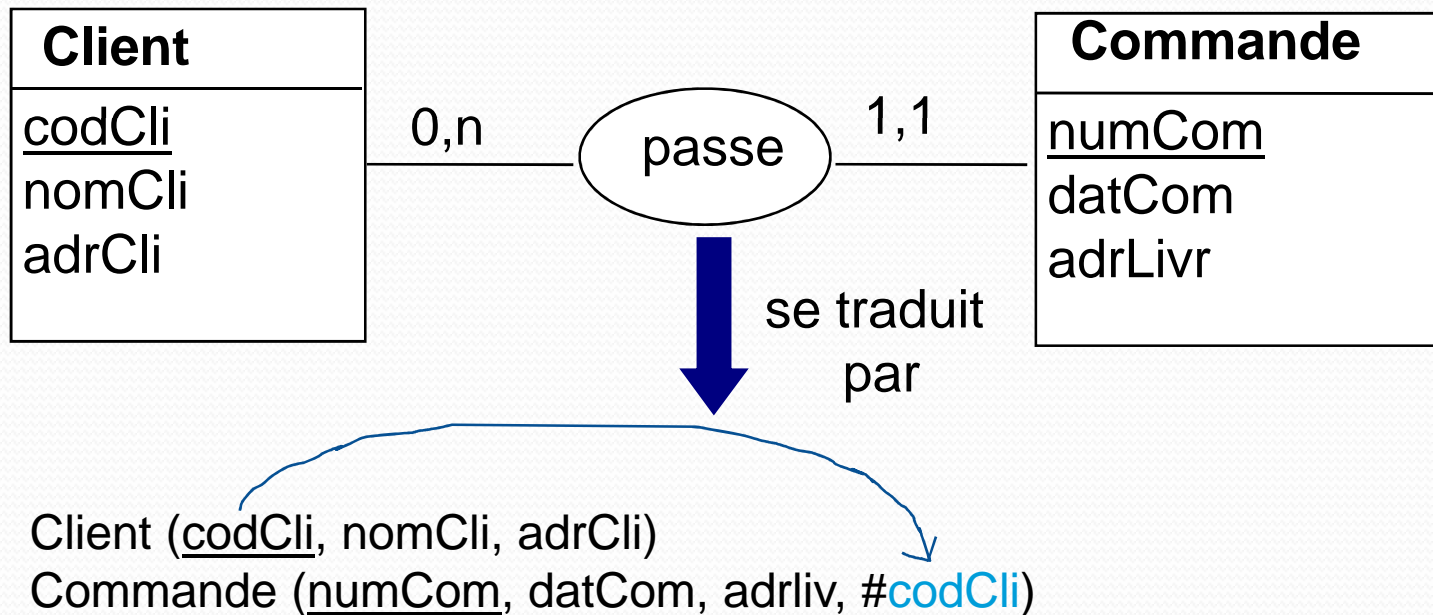
Réalisateur(numRéalisateur, nom, prénom, dateNaissance)
 Films(titre, date, #numRéalisateur)

Titre	Num_realisateur	Année
Alien	1	1979
Vertigo	2	1958
Psychose	3	1960
Kagemusha	4	1980
Volte-face	5	1997
Pulp Fiction	6	1995
Titanic	7	1997
Sacrifice	8	1986

Num_realisateur	nom	prenom	dateNaissance
1	Scott	Ridley	1943
2	Hitchcock	Alfred	1899
3	Kurosawa	Akira	1910
4	Woo	John	1946
5	Tarantino	Quentin	1963
6	Cameron	James	1954
7	Tarkovski	Andrei	1932

Passage EA au relation

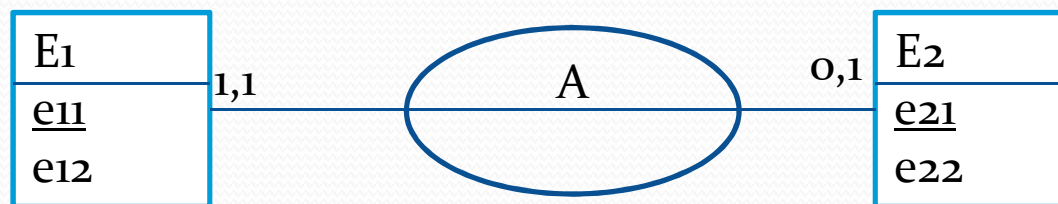
Exemple : Association binaire (1,1) – (0,n)



Passage EA au relation

- **Cas particulier E1 (1,1) – E2 (0,1)**

se traduit en ajoutant une clé étrangère (identifiant de l'entité de cardinalité (0,1)) à la table provenant de l'entité dont la cardinalité est (1,1).

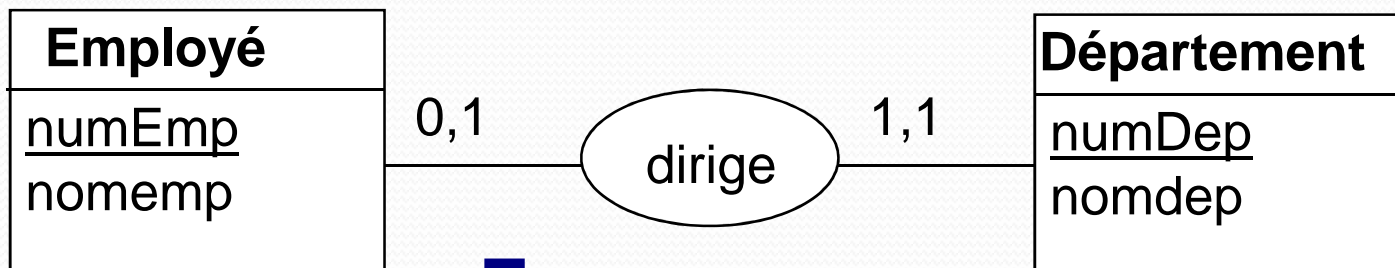


se traduit
par

E2(e21,e22)
E1(e11,e12,#e21)

Passage EA au relation

Exemple : Association binaire (1,1) - (0,1)



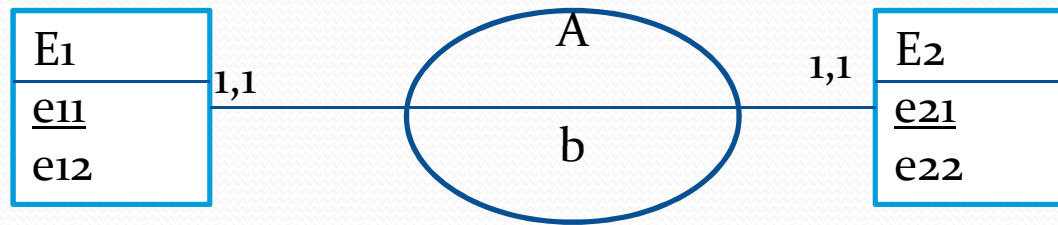
se traduit
par

Employé (numEmp, nomEmp)

Département (**numDep**, nomdep, **#numEmp**)

Passage EA au relation

- **Cas particulier : E1 (1,1) – E2 (1,1)**
ajout à une entité (E1 ou E2) de l'identifiant de l'autre et des attributs de l'association. Possibilité de ne faire qu'une seule relation



se traduit
par

ou

E1(e11, #e21, e12, b)

E2(e21, e22)

Ou

E1(e11, e12)

E2(e21, #e11, e22, b)

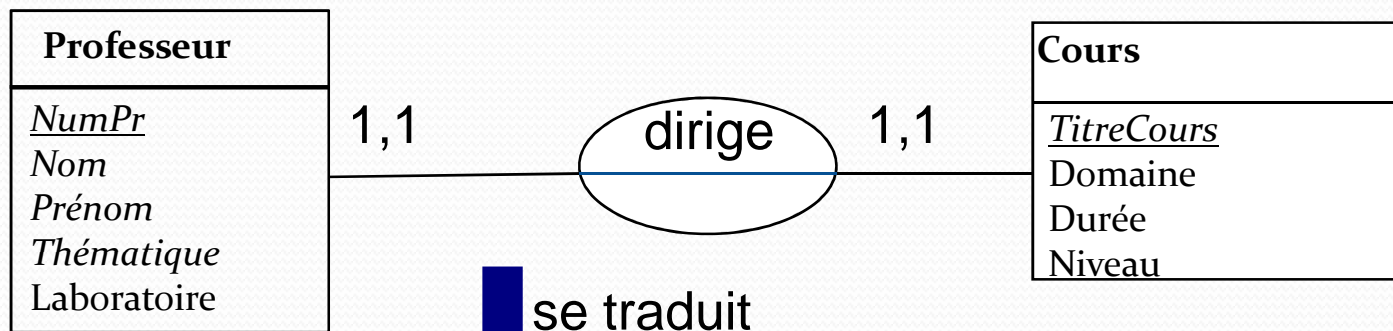
E1(e11, e12, e22, b)

Si cette entité
n'est pas engagée
dans une autre
association

Passage EA au relation

Exemple : Association (1,1) - (1,1)

ajout à une entité (E1 ou E2) de l'identifiant de l'autre et des attributs de l'association



se traduit
par

Professeur(NumPr, Nom, Prénom, thématique, Laboratoire, #titreCours)

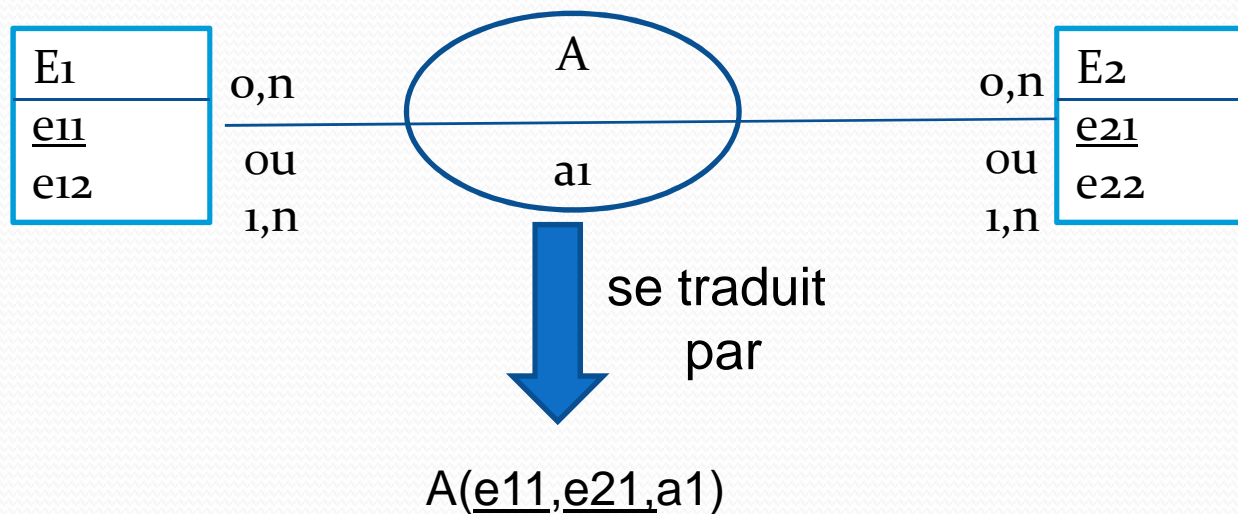
cours(titreCours, domaine, durée, niveau)

Passage EA au relation

- **Cas : E1 (0,N) ou (1,N) – E2 (0,N) ou (1,N)**

l'association devient une nouvelle relation (3 relations au final) :

se traduit par une nouvelle relation dont la clé primaire est composée des identifiants des deux entités. Les éventuelles propriétés de l'association deviennent les attributs de cette nouvelle relation.



On aura donc trois relations (tables) :

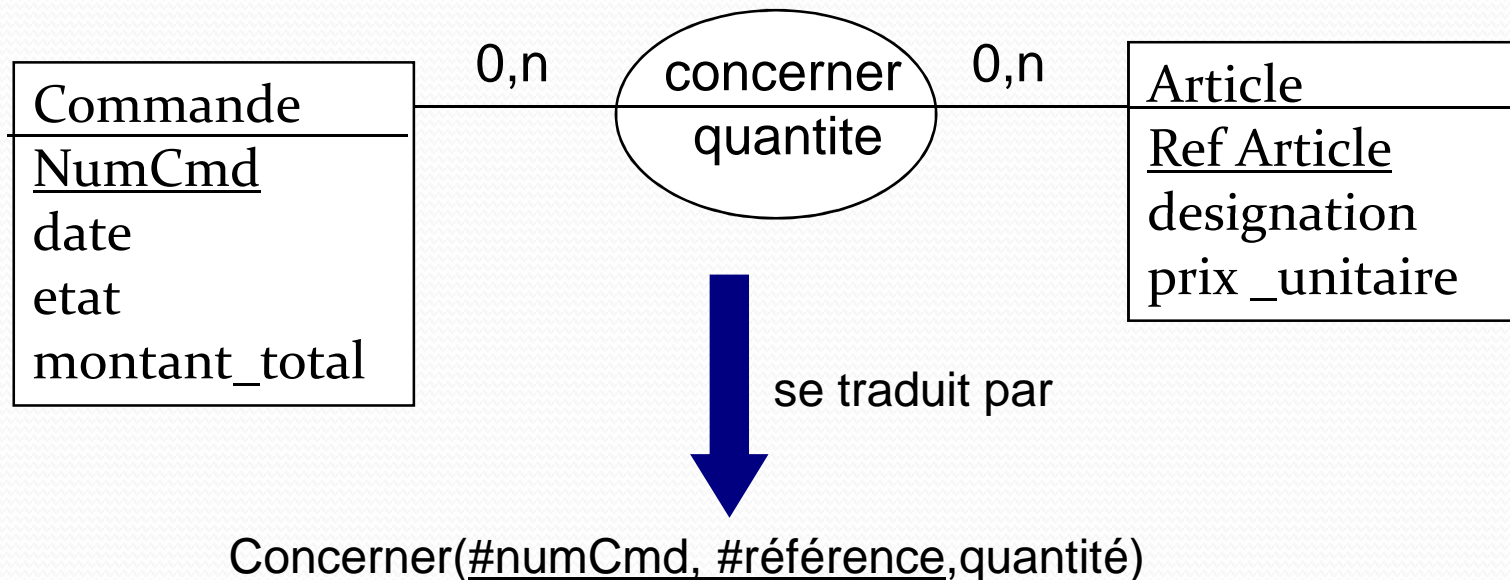
E1(e11, e12)

E2(e21, e22)

A(#e11, #e21, a1)

Passage EA au relation

exemple : association binaire (0,N) ou (1,N) - (0,N) ou (1,N)



On aura donc :

Commande(NumCmd, date, etat, montant_total)

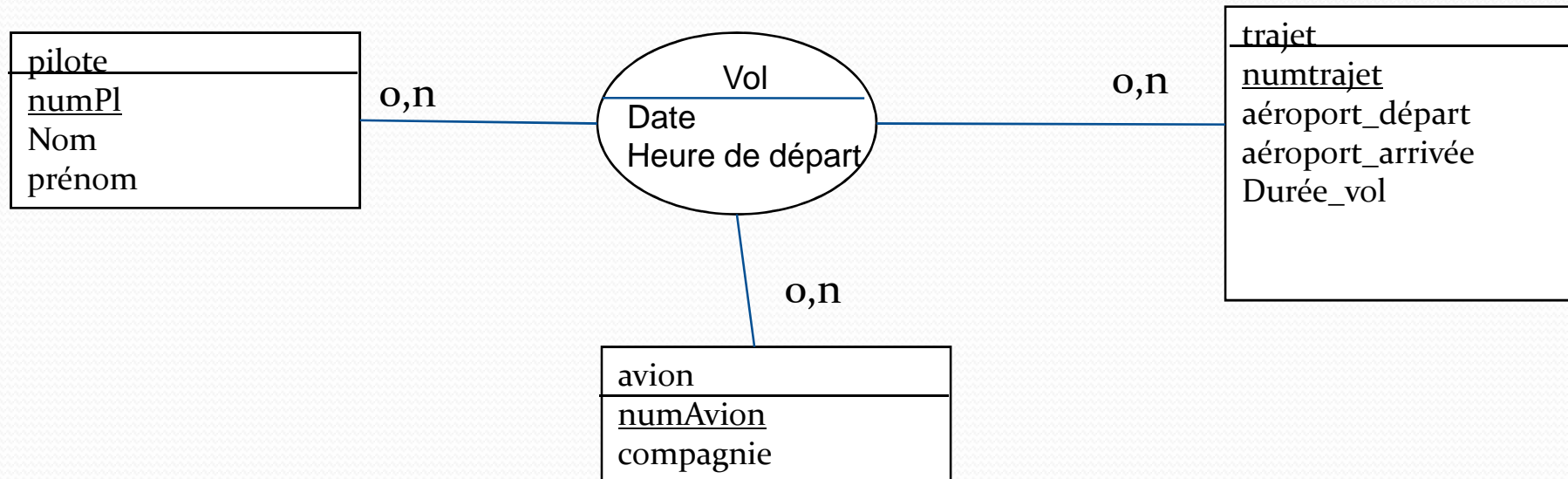
Article(Ref Article, désignation, prix_unitaire)

Concerner(#NumCmd, #RefArticle, quantité)

Passage EA au relation

Association n-aire ($n > 2$) : cardinalité 1,n ou 0,n par tout

on crée une table ayant pour clé primaire les identifiants des différentes entités de l'association. Les éventuelles propriétés de l'association deviennent les attributs de la table.

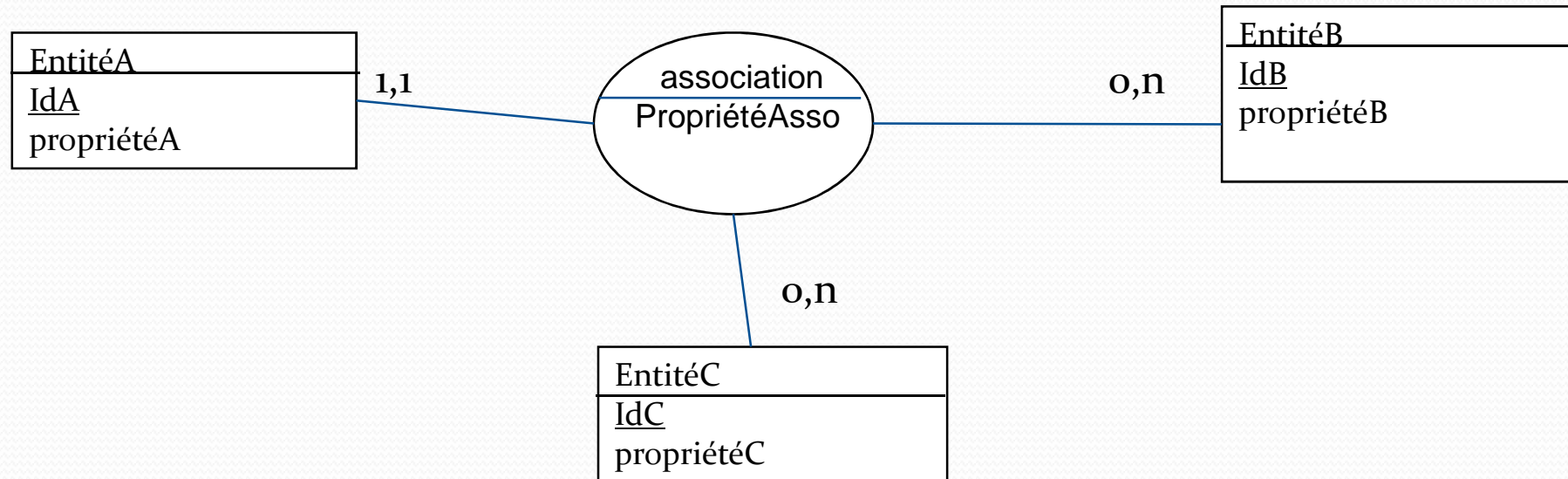


Vol (#numPl, # numAvion, #numtrajet, date, heure départ)

Passage EA au relation

Association n-aire (n>2)

Si une association ternaire possède une branche avec une cardinalité (1,1) on place les références dans la table reliées par (1,1)



Traduction :

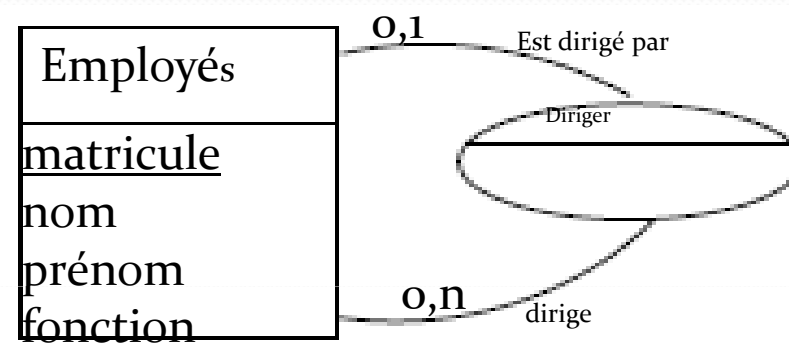
EntitéA (IdA,propriétéA,**#idB**,**#idC**,propriétéAsso)

EntitéB(**idB**,propriétéB)

EntitéC(**idC**,propriétéC)

Passage EA au relation

Cas particulier des associations réflexives



*) Un employé a pour chef 0 ou un seul autre salarié.

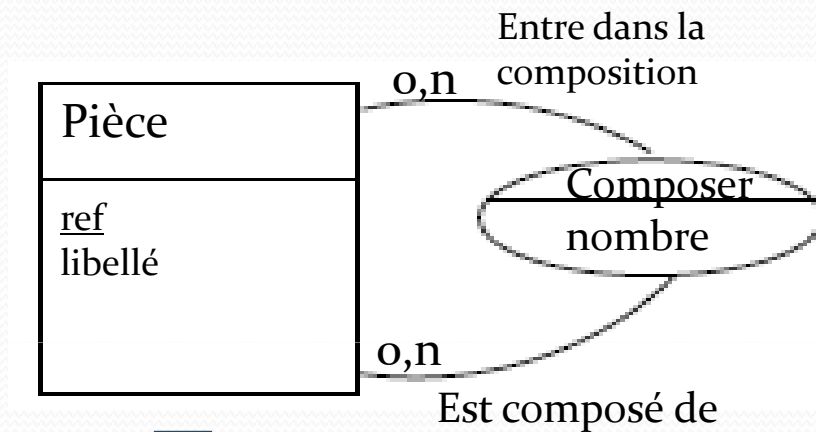
Un salarié est chef de 0 à n autre(s) salariés.

Traduction en modèle relationnel :

Employé(matricul, nom, prénom, fonction, #matricul_chef)

Passage EA au relation

Cas particulier des associations réflexives

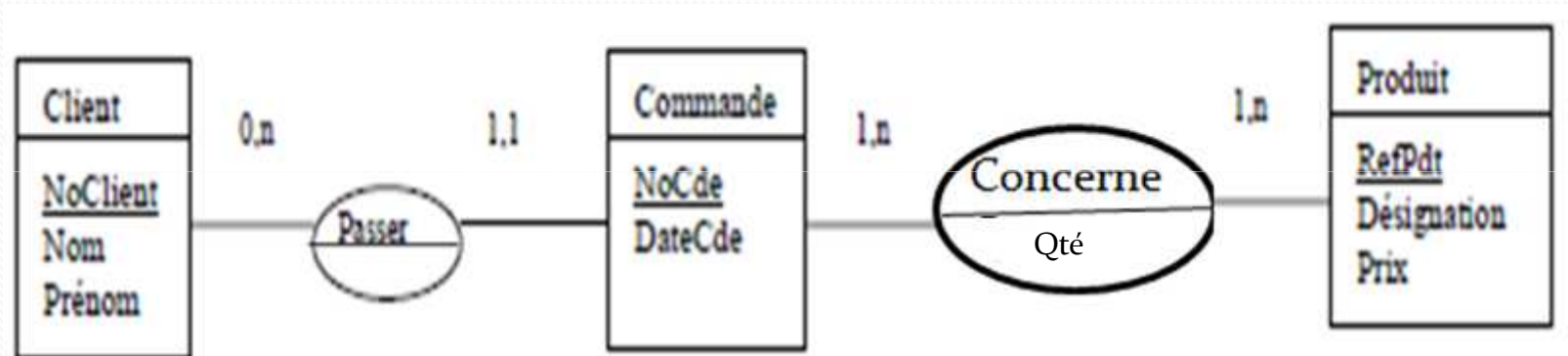


Pièce(ref,libellé)

Composer(#ref composé,#ref composant, nombre)

Exercice 1

A partir de schéma entité association ci-dessous produire le schéma relationnel



Solution exercice 1

== > **Modèle relationnel :**

Client(NoClient, Nom, Prénom)

Commande (NoCde, DateCde, #NoClient)

Produit(RefPdt, Désignation, Prix)

Commande_Produit(#NoCde, #RefPdt, Qté) == > transformation
d'association concernée



Client

<u>NoClient</u>	Nom	Prénom
1	Lassus	Annick
2	Mundubeltz	Armelle
3	Chaulet	Bernadette

Commande

<u>NoCde</u>	DateCde	NoClient#
100	14/04/2001	2
101	14/04/2001	1

Produit

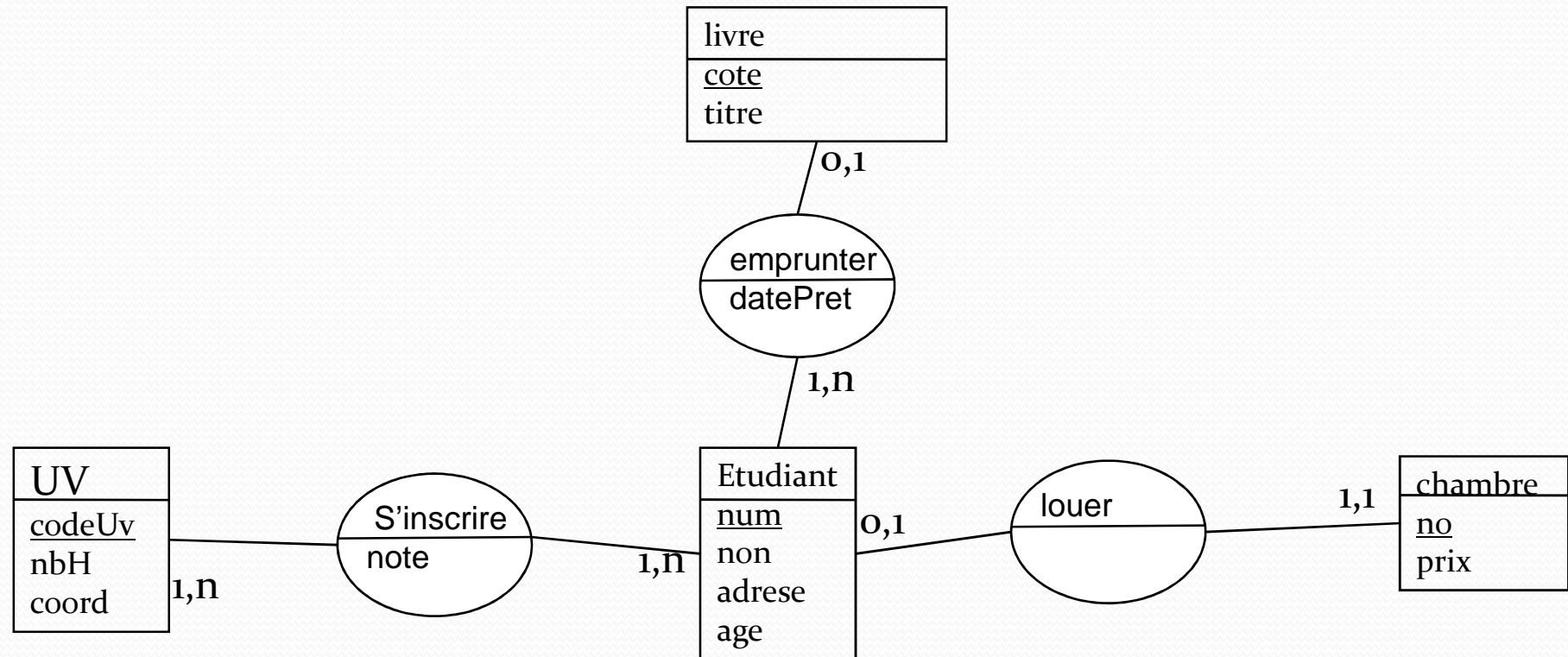
<u>RefPdt</u>	Désignation	Prix
VE45	Vélo	1500 F
VE32	Kit 2 roues arrières	300 F
VE21	Kit éclairage	150 F

Ligne

<u>NoCde#</u>	<u>RefPdt#</u>	Qté
100	VE45	1
100	VE32	1
101	VE21	2

Exercice 2

A partir de schéma entité association ci-dessous produire le schéma relationnel



Correction exercice 2

Étudiant(num, nom, adresse, age)

UV(codeUV, nbH, coord)

Inscrit(#numÉtudiant, #codeUV, note)

Livre(côte, titre, #numÉtudiant, datePrêt)

Chambre(no, prix, #numÉtudiant)

Jeux de données pour l'exercice 2

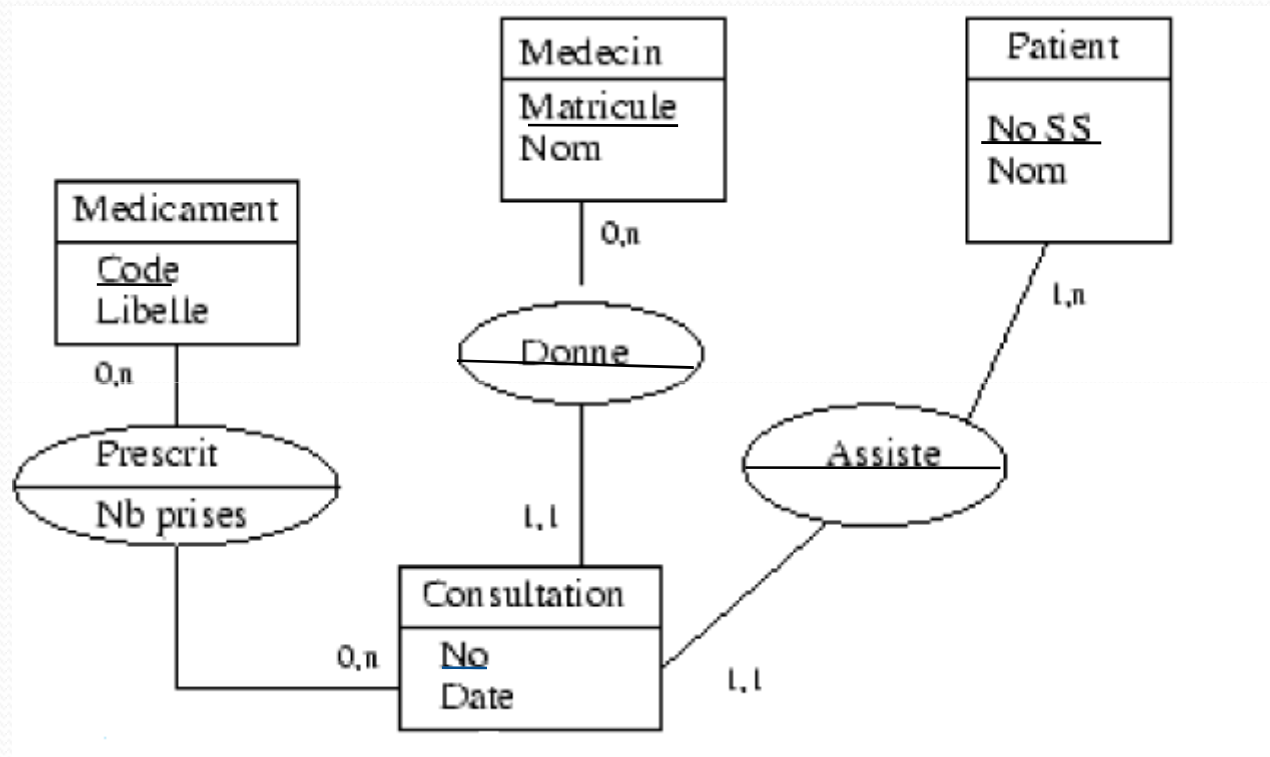
Étudiant	Num	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tata	RU ₂
	3	Toto	RU ₁

Inscrit	NumÉtudiant	CodeUV	Note
	2	BD	10
	1	BD	20
	2	SE	17
	3	SE	18

UV	codeUV	nbH	coord
	SE	45	Ameur
	BD	21	Ben Salah

Exercice 3

A partir de schéma entité association ci-dessous produire le schéma relationnel



Exercice 3

Solution:

Medicament (Code, libelle)

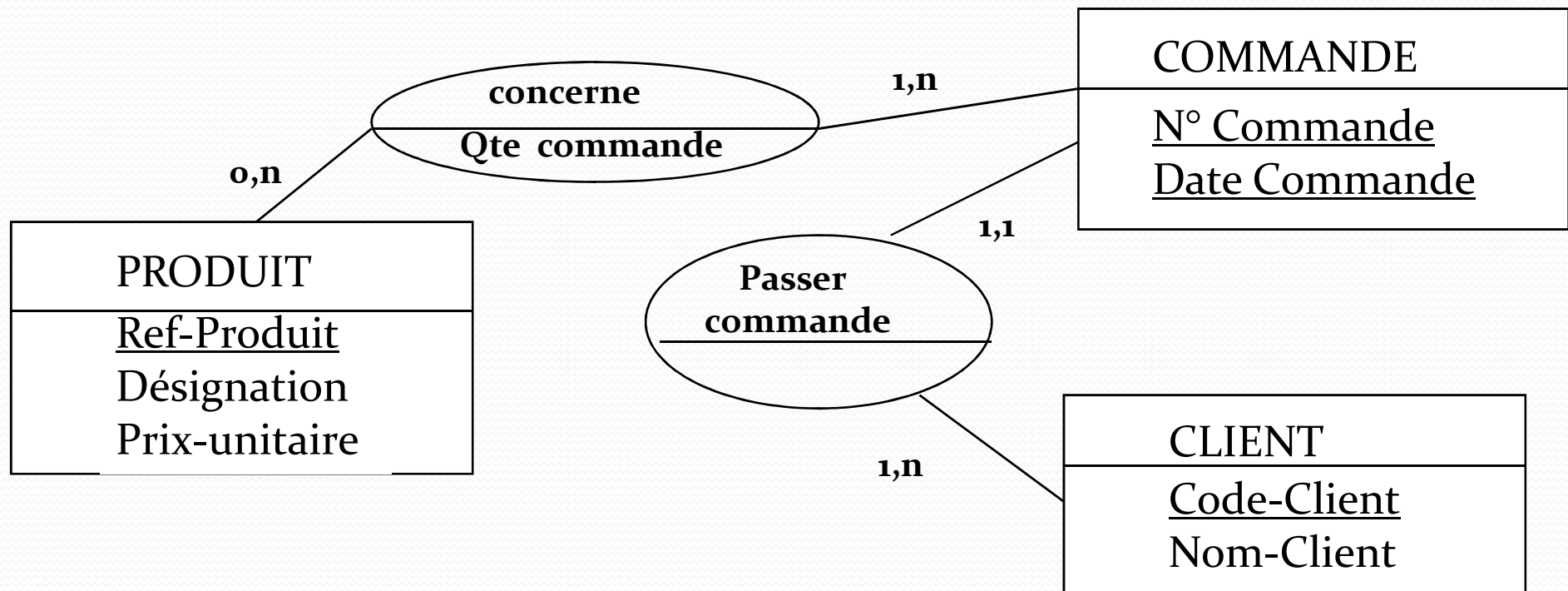
Medecin (matricule, nom).

Patient (NOSS, nom)

Consultation (No, #Matricule, #NOSS, date)

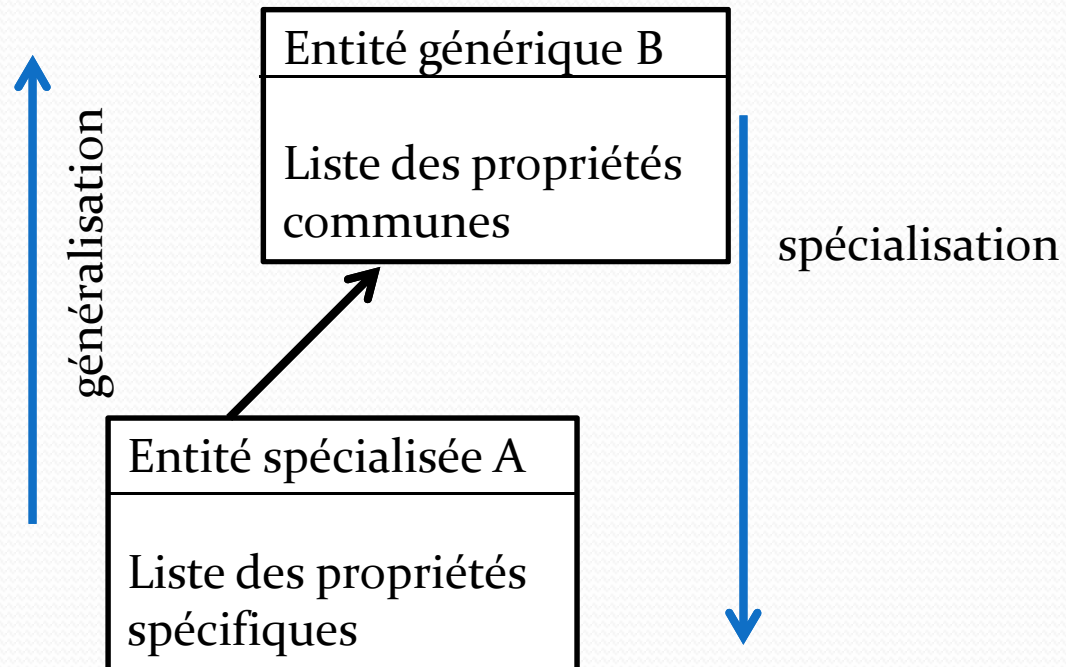
Prescription (#code, #No, nb-prises)

Exercice 4



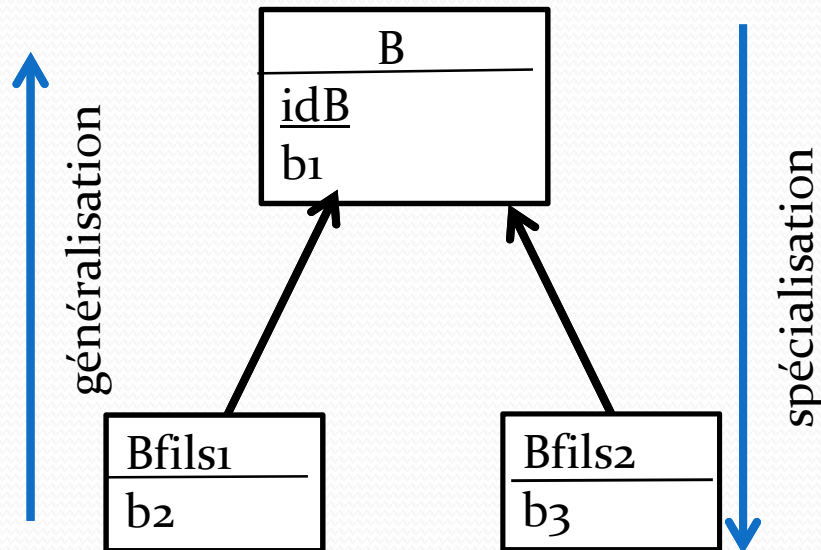
Le concept d'héritage

Entité générique B



Le concept d'héritage

Entité générique B



Sol₁ :

$B(\underline{idB}, b_1, b_2, b_3, statut)$

Sol₂ :

$Bfils1(\underline{idbfils1}, b_1, b_2)$

$Bfils2(\underline{idbfils2}, b_1, b_3)$

Sol₃ :

$B(\underline{idb}, b_1)$
 $Bfils1(\underline{\#idbfils1}, b_2)$ et $Bfils2(\underline{\#idbfils2}, b_3)$

Le concept d'héritage

Sol1 :

Compte(numCompte,solde,taux,Sdecouvert,statut)

Sol2 :

CompteEpargne(numCompte,solde,taux)

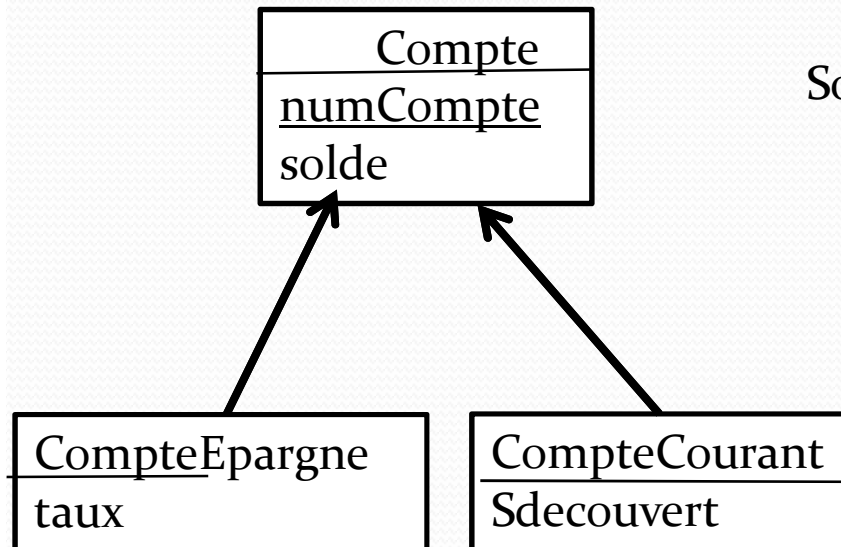
CompteCourant(numCompte,solde,Sdecouvert)

Sol3 :

Compte(numCompte,solde,taux,Sdecouvert)

CompteEpargne(#numCompte,taux)

CompteCourant(#numCompte,Sdecouvert)



Formes Normales

- Objectifs de la normalisation :
 - limiter les redondances de données,
 - limiter les pertes de données,
 - limiter les incohérences au sein des données
 - améliorer les performances des traitements.

Dépendance fonctionnel

- Définition : un attribut Y dépend fonctionnellement d'un attribut X ssi une valeur de X induit une et unique valeur de Y.

Notation : $X \rightarrow Y$

exemple : $\text{numClient} \rightarrow \text{nomClient}$

par contre :

$\text{nomClient} \not\rightarrow \text{numClient}$

car on peut avoir plusieurs client de même nom mais de numClient différents

- Toutes les attributs d'une entité doivent être en dépendance fonctionnelle directe avec l'identifiant de cette Entité

Dépendance fonctionnel

1 - Réflexivité : $Y \subseteq X \Rightarrow X \rightarrow Y$
exemple e: type,marque \rightarrow type

2 - Augmentation : $X \rightarrow Y \Rightarrow X, Z \rightarrow Y, Z$
si X détermine Y, les deux ensemble d'attributs peuvent être enrichis par un même troisième
exemple :
numVoiture \rightarrow couleur
numVoiture,marque \rightarrow couleur, marque

3 - Union : $X \rightarrow Y \text{ et } X \rightarrow Z \Rightarrow X \rightarrow Y, Z$

4 - Projectivité : $X \rightarrow Y, Z \Rightarrow X \rightarrow Y \text{ et } X \rightarrow Z$

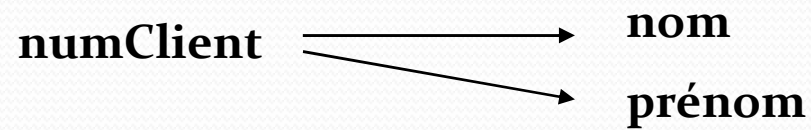
5 - Transitivité : $X \rightarrow Y \text{ et } Y \rightarrow Z \Rightarrow X \rightarrow Z$

6 - Pseudo-transitivité : $X \rightarrow Y \text{ et } W, Y \rightarrow Z \Rightarrow W, X \rightarrow Z$

Dépendance fonctionnel

Exemple 1 : relation client

Client(NumClient, nom, prénom)



On peut écrire : **numClient** \longrightarrow (**nom** , **prénom**)

Toutes les attributs d'une relation sont en dépendance fonctionnelle directe avec identifiant (clé primaire) de cette relation

Première forme normale

- 1ere forme normale (1FN) : un attribut ne peut prendre qu'une valeur (attribut atomique) et non un ensemble ou liste de valeur
 - Si un attribut prends plusieurs valeur == > créer une nouvelle relation

Exemple :

Livre(numLivre, titre, auteurs) n'est pas en 1FN

auteurs : n'est pas un attribut atomique car peut avoir plusieurs valeurs

== >

Livre(numLivre, titre, #numAuteur)

Auteur(numAuteur, nomAuteur)



Deuxième forme normale

- 2 ème forme normale (2FN) :
être en 1ere forme normale
+ si la clé primaire est composée de plusieurs attributs, les autres attributs de la relation doivent dépendre de la totalité de la clé et non d'une partie.

Exemple :

la relation Commande(num_Client, codeProduit, nomClient, designationProduit, Qte)

n'est pas en 2FN car :

DF: $\text{num_Client} \rightarrow \text{nomClient}$ et $\text{codeProduit} \rightarrow \text{designationProduit}$
 $(\text{num_Client}, \text{codeProduit}) \rightarrow \text{Qte}$

Correction :

Client(num_Client, nom)

Article(codeProduit, designationProduit)

Commande(#num_Client, #codePrtilce, Qte)

Troisième forme normale

- 3 ème forme normale :
être en 2FN
- + Tous les attributs n'appartiennent pas à l'identifiant (clé primaire) doivent dépendre fonctionnellement de l'identifiant et d'aucun autre attribut ==> aucun attribut non clé ne dépend fonctionnellement d'un autre attribut non clé. Si ce n'est pas le cas : placer l'attribut pathologique dans une relation séparée

Troisième forme normale

Exemple 1 :

Avions(NumAvion, # type, capacite, poids, propriétaire) n'est pas en 3FN car :

NumAvion	→ propriétaire	== > DF direct
NumAvion	→ type	== > DF direct
type	→ capacite, poids	== > DF direct
NumAvion	→ capacite, poids	== > DF indirect (DF par transitivité)
type	→ NumAvion	== > pas DF

Correction :

Avions(NumAvion, #type, propriétaire)
Type(type, capacité,poids)

Troisième forme normale

Exemple 2 :

R (NumFacture, DateFacture, TotalFacture, NumClient, NomClient)

On a :

NumFacture \rightarrow DateFacture, TotalFacture \implies > DF direct

NumClient \rightarrow NumFacture \implies > pas de DF

NumFacture \rightarrow NumClient \implies > DF direct

NumFacture \rightarrow NomClient \implies > DF indirect (DF par transitivité)

En effet :

La clé est NumFacture

R est en 2NF :

NumFacture \rightarrow NumClient \rightarrow NomClient \implies DF par transitivité.

La solution en 3NF :

Facture(NumFacture, DateFacture, TotalFacture, #NumClient)

Client (NumClient, NomClient)



ALGEBRE RELATIONNEL

Algèbre relationnel : définitions

- Ensemble d'opérateurs qui s'appliquent aux relations
- Résultat : nouvelle relation qui peut à son tour être manipulée
- L'algèbre relationnel permet de faire des recherches dans les relations

Opérations ensemblistes

- Relations ayant le même schéma
 - Union: $T = R \cup S$
 - Intersection: $T = R \cap S$
 - Différence: $T = R - S$
- Opérations n'ayant pas (forcément) le même schéma
 - Produit cartésien : $T = R \times S$



■ But

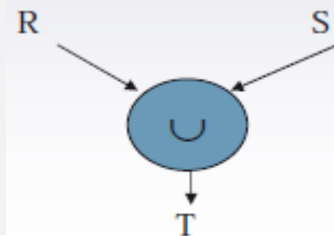
- Permet de fusionner 2 relations : réunit dans une même relation les tuples de R et ceux de S

■ Contraintes

- Opérandes : relations ayant le même schéma

■ Notation

- Notation textuelle : $T \leftarrow R \cup S$
- Notation graphique :



■ Nom des profs, des étudiants

Prof	Nom	Étudiant2	Nom
	Ben Salah		Bélaïd
	Ameur		Tata
	Tata		Toto



Prof ∪ Étudiant2

Resu	Nom
	Ameur
	Ben Salah
	Tata
	Bélaïd
	Toto



Résultat : relation avec les *tuples* de chaque opérande (les doubles sont éliminés)

Intersection



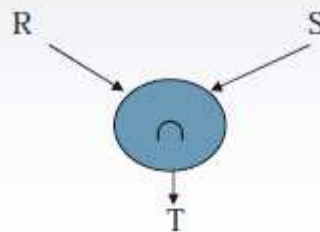
■ But

- Permet d'obtenir l'ensemble des tuples appartenant, à la fois, à deux relations

- Opérandes : relations ayant le même schéma

■ Notation

- Notation textuelle : $T \leftarrow R \cap S$
- Notation graphique :



■ Nom des profs, des étudiants

Prof	Nom	Étudiant2	Nom
	Ben Salah		Bélaïd
	Ameur		Tata
	Tata		Toto

■ Noms communs étudiants-profs : $\text{Prof} \cap \text{Étudiant2}$

Resu	Nom
	Tata

Résultat : relation avec les *tuples* présents dans les deux opérandes à la fois



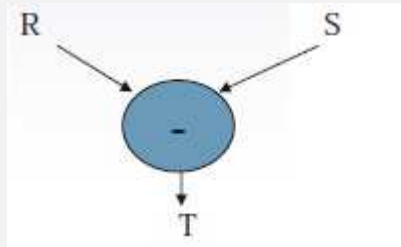
■ But

- Obtenir l'ensemble des tuples d'une relation qui ne figurent pas dans une autre

■ Opérandes : relations ayant le même schéma

■ Notation

- Notation textuelle : $T \leftarrow R - S$
- Notation graphique :



Remarque : $R - S \neq S - R$

■ Nom des profs, des étudiants

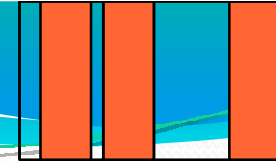
Prof	Nom	Étudiant2	Nom
	Ben Salah		Bélaïd
	Ameur		Tata
	Tata		Toto

■ Noms des étudiants qui ne portent pas le nom d'un prof : Etudiant2 - Prof

Résu	Nom
	Bélaïd
	Toto

Résultat : les tuples de la première relation présent seulement dans la première relation

Projection



■ But

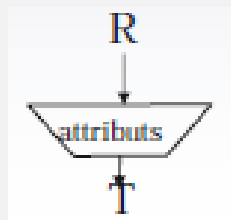
- Permet de "sélectionner" des attributs
- La projection réduit la taille de la relation horizontalement

■ Opérandes :

1 relation et un ou des attributs

■ Notation

- Notation textuelle : $T \leftarrow \Pi_{\text{attributs}}(R)$
- Notation graphique :



Étudiant	Num	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tata	RU ₂
	3	Toto	RU ₁

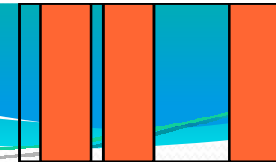
■ Adresses des étudiants : $\Pi_{\text{adresse}}(\text{Etudiant})$

$\Pi_{\text{adresse}}(\text{Etudiant})$	Adresse
	RU ₁
	RU ₂

Pas de doublon

Résultat : la relation restante après conservation des colonnes des seuls attributs indiqués et suppression des *tuples devenus identiques*

Projection



Exemple :

Soit le schéma relationnel :

$\text{voyage}(\text{refV}, \text{dest}, \text{durée}, \text{cout})$



Voyage	refv	dest	duree	cout
	101	tunisie	2	9500
	201	Grèce	2	9000
	109	Tunisie	1	7500
	401	Norvège	4	15000
	304	Espagne	2	9000

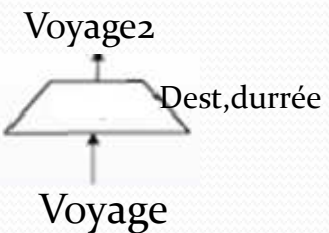
Question :

Quelle est l'opération qui répond à la requête suivante :

Quelles sont les destinations possible et les durées correspondantes ?

Réponse :

$\text{Voyage2} = \Pi_{\text{Dest}, \text{durée}}(\text{Voyage}) \equiv$



Voyage2	dest	duree
	tunisie	2
	Grèce	2
	Tunisie	1
	Norvège	4
	Espagne	2

Restriction

■ But

- Permet de "sélectionner" des tuples
- Réduit la taille de la relation verticalement

■ Notation

- Notation textuelle : $T \leftarrow \sigma_{\text{cond}}(R)$

- Notation graphique :



Restriction

Les étudiants (NumÉtudiant) inscrit dans l'UV BD avec leurs note :

Inscrit	NumÉtudiant	CodeUV	Note
	2	BD	10
	1	BD	20
	2	SE	17
	3	SE	18

■ Inscrits en BD : $\sigma_{\text{codeUV}='BD'}(\text{Inscrit})$

Resu	NumÉtudiant	CodeUV	Note
	2	BD	10
	1	BD	20

Les étudiants inscrit dans l'UV BD et ayant une note >15 (bon étudiant):

■ Bons étudiants (note > 15) de BD :
 $\sigma_{\text{codeUV}='BD' \text{ et note } > 15}(\text{Inscrit})$

Valeur des chaînes de caractères délimitées par des “

Resu	NumÉtudiant	CodeUV	Note
	1	BD	20

Restriction

Exemple :

Soit le schéma relationnel :

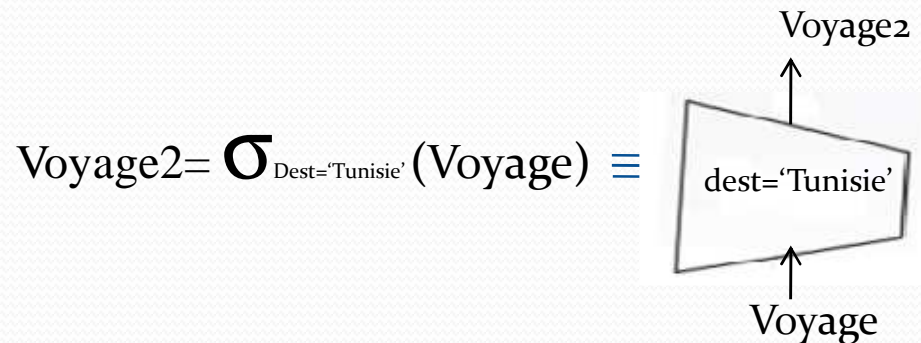
`voyage(refV,dest,durée, cout)`



Voyage	refv	dest	duree	cout
	101	tunisie	2	9500
	201	Grèce	2	9000
	109	Tunisie	1	7500
	401	Norvège	4	15000
	304	Espagne	2	9000

Quels sont les voyages à destination de la Tunisie ?

Réponse :



Voyage	refv	dest	duree	cout
	101	tunisie	2	9500
	109	Tunisie	1	7500

Exercice

Avion(immatrilation,#typeAvion, #compa,prixAchat)

immatriculation	typeAvion	compa	PrixAchat
F-WTS	A320	AF	104500
F-GTMP	A320	AF	104500
M-GTMP	A320	RAM	198000
M-TNP	A330	RAM	204500
F-WXF	A340	AF	120000
F-ITR	A340	CAST	104500
F-LAV	Concord	AF	15600
M-BIO	A340	RAM	198000
M-INR	A330	RAM	204500

les types d'avions que AF exploite

les types d'avions que RAM exploite

TypeAvionAF

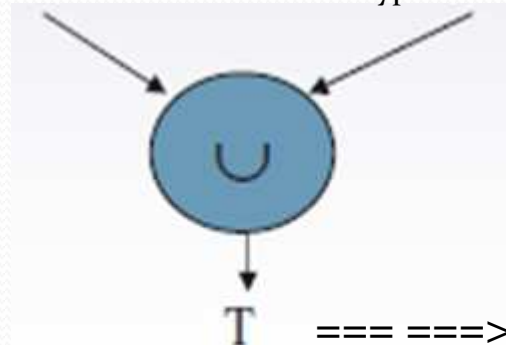
TypeAvion
A320
Concord
A340

TypeAvionRAM

TypeAvion
A320
A330
A340

- Tous les types d'avions exploité par AF ou par RAM

TypeAvionAF TypeAvionRAM



T
A320
A330
A340
Concord

*) $T = \text{TypeAvionAF} \cup \text{TypeAvionRAM}$

Jointure



■ But

- Opération consistant à rapprocher selon une condition les tuples de deux tables (relations) R et S afin de former une troisième table T. Cette opération exploite largement les clé étrangère pour rétablir un lien entre tables.

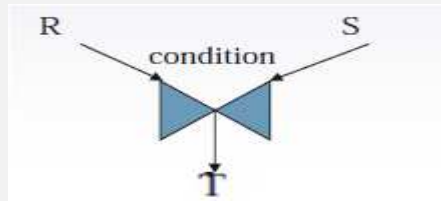
■ Opérandes :

- 2 relations et une condition portant sur un attribut de la première relation et un attribut de la 2eme relation

■ Notation

- Notation textuelle : $T \leftarrow R \bowtie_{\text{condition}} S$

- Notation graphique :

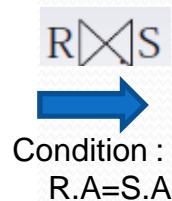


R

A	B	C
a1	b1	c1
a2	b2	c2

S

A	D
a2	d2
a2	d3
a3	d4



A	B	C	D
a2	b2	c2	d2
a2	b2	c2	d3

Jointure



Exemple de Jointure

Étudiant	<u>Num</u>	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tata	RU ₂
	3	Toto	RU ₁

Étudiant.Num=Chambre.numÉtudiant

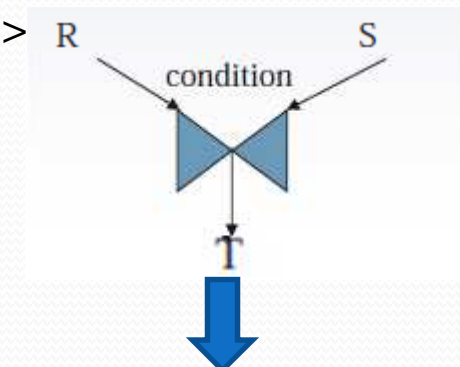
Chambre	<u>No</u>	Prix	numÉtudiant
	10	200	3
	21	150	2

Liste des noms des étudiant ayant louer une chambre ainsi que leurs adresse, age ... >

R = Etudiant

S = Chambre

Condition : Etudiant.num=Chambre.numEtudiant



Étudiant Chambre	<u>Num</u>	Nom	Adresse	<u>No</u>	Prix	numÉtudiant
	2	Tata	RU ₂	21	150	2
	3	Toto	RU ₁	10	200	3

Jointure



Exercice :

Exprimer la requête ci-dessous dans le langage algébrique :

Nom des étudiants inscrit dans l'UV BD

Rappel :

Étudiant	Num	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tota	RU ₂
	3	Toto	RU ₁

Inscrit	NumÉtudiant	CodeUV	Note
	2	BD	10
	1	BD	20
	2	SE	17
	3	SE	18

Division

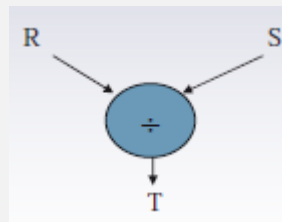
■ But

- Répondre aux requêtes de type « tous les »
- Soit $T = R \div S$: Un tuple t est dans T si et seulement si pour tout tuple s de S , le tuple $\langle t, s \rangle$ est dans R

■ Opérandes : relations ayant le même schéma

■ Notation

- Notation textuelle : $T \leftarrow R \div S$
- Notation graphique :



Division

Division : Exemple

R

A	B	C	D
a	b	c	d
a	b'	c'	d'
a'	b	c	d
a	b	c'	d'
a'	b	c'	d'

S

C	D
c	d
c'	d'

$R \div S$

A	B
a	b
a'	b

ab' n'est dans dans $R \div S$ car
ab'cd n'est pas dans R

Chaque ligne **L1 de $R \div S$** est
telle que pour chaque ligne
L2 de S, **L1L2 est dans R**

Étudiant	Num	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tota	RU ₂
	3	Toto	RU ₁

UV	codeUV	nbH	coord
	SE	45	Ameur
	BD	21	Ben Salah

Inscrit	NumÉtudiant	CodeUV	Note
	2	BD	10
	1	BD	20
	2	SE	17
	3	SE	18

Division

Quels sont les étudiants (numEtudiant) inscrits à toutes les UVs ?

■ Démarche :

- Construire R :
ensemble de toutes les informations dont on a besoin = attributs NumÉtudiant et CodeUV de Inscrit (R)
- Construire S :
ensemble correspondant à « **tous les** », ici UV, **et rien qu'à cet ensemble** => codeUV (S)
- Résultat = $R \div S$
- Vérification :
 - (Résultat \times S) \subseteq R
 - avec Résultat \times S : produit cartésien entre Résultat et S

- $R \leftarrow \Pi_{\text{NumÉtudiant}, \text{CodeUV}}(\text{Inscrit})$

R	NumÉtudiant	CodeUV
	2	BD
	1	BD
	2	SE
	3	SE

- $S \leftarrow \Pi_{\text{Code}}(\text{UV})$

S	CodeUV
	BD
	SE

- Résultat

Resu	NumÉtudiant
	2

Produit cartésien (1)



■ But

- Ensemble de tous les tuples obtenus par concaténation de chaque tuple de R avec chaque tuple de S

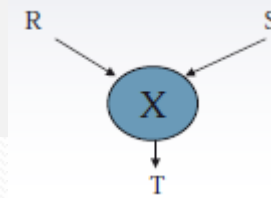
■ Opérandes : 2 relations

■ Schéma du résultat:

- $R(a_1, a_2, \dots, a_n), S(b_1, b_2, \dots, b_p)$
- $T \leftarrow R \times S \implies T(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_p)$

■ Notation

- Notation textuelle : $T \leftarrow R \times S$
- Notation graphique :



Produit cartésien (2)

Étudiant	Num	Nom	Adresse
	1	Bélaïd	RU ₁
	2	Tata	RU ₂
	3	Toto	RU ₁

UV	codeUV	nbH	coord
	SE	45	Ameur
	BD	21	Ben Salah

Étudiant X UV	Num	Nom	Adresse	Code	Nbh	Coord
	1	Bélaïd	RU ₁	SE	45	Ameur
	2	Tata	RU ₂	SE	45	Ameur
	3	Toto	RU ₁	SE	45	Ameur
	1	Bélaïd	RU ₁	BD	21	Ben Salah
	2	Tata	RU ₂	BD	21	Ben Salah
	3	Toto	RU ₁	BD	21	Ben Salah

Résultat : toutes les combinaisons possibles des
tuples des opérandes

Les arbres algébriques

- Arbre dont les nœud correspondent à une opération de l'algèbre relationnel et les arcs à des relations (tables) de base ou temporaires représentant des flots de données entre opérations

- exemple :

Produit(numpro, nomP, qte, couleur)

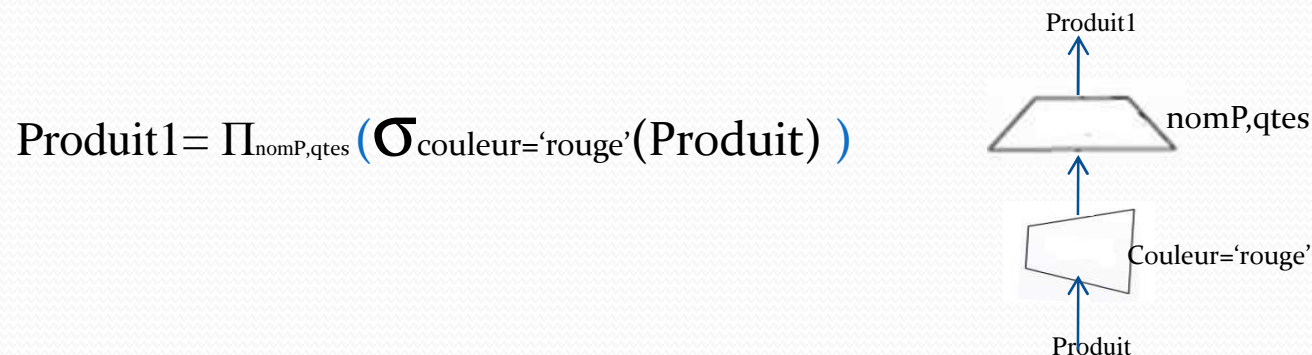
- Requête 1 :

Donner la liste des noms et couleurs de tous les produits



- Requête 2 :

Donner la liste des noms et quantités des produits de couleurs rouge



Les arbres algébriques

Soit les relation suivantes :

employé(NoEmp, non, #NumDept)

Departement(NoDep, intitulé, #noResp)

travaille(#NoEmp, #NoProj, temps)

Exprimer en algèbre relationnel :

- 1- noms des employés du département 03
- 2- nom des employés du département "Achat"
- 3- liste des employés (numéro des employés) consacrant plus de 20 % de leurs temps au projet 'B18'
- 4- nom des employés qui travaillent sur le projet B05 et B06
- 5- nom des employés qui travaillent sur le projet B05 ou B06