

Introduction

Le modèle relationnel

- ☐ Introduit par Codd dans les années 70
- ☐ C'est un modèle ensembliste simple et rigoureux
- ☐ Une BDR est une BD dont le schéma est un ensemble de schémas de relations et dont les occurrences sont des tuples (ou n-uplets) de ces relations.
- ☐ Les entités et les associations du modèle E/A sont représentées exclusivement par des relations (tables).
- ☐ Une entité est représentée par sa liste d'attributs.
- ☐ Une association est représentée par la liste des clés des entités qu'elle associe et ses propres attributs

Introduction

Le modèle relationnel

- ❑ Un SGBD est dit minimalement relationnel si :
 - Les informations de la base sont représentées par des tables ;
 - Il n'y a pas de pointeurs visibles (pour l'utilisateur) sur les tables ;
 - Le système supporte les opérateurs relationnels :
 - La restriction;
 - La projection ;
 - La jointure.

Introduction

Le modèle relationnel

- ❑ Un SGBD est complètement relationnel :
 - La réalisation de toutes les opérations de l'algèbre relationnel ;
 - Unicité des clés (pas de doublons) ;
 - La contrainte référentielle (ex : pouvoir s'assurer que le produit dont on a passé commande se trouve bien dans la relation PRODUIT).

Les concepts

Le Domaine

- ❑ Un domaine est un ensemble dénombrable de valeurs caractérisé par un nom
- ❑ Exemple :
 $\text{NOM_VILLE} = \{ \text{Rabat, Casablanca, Ouarzazate} \}$
 $\text{COULEUR} = \{ \text{Blanc, Noir, Rose, Gris} \}$
- ❑ Les domaines sont les ensembles de valeurs possibles dans lesquels sont puisées les données
- ❑ Deux ensembles peuvent avoir les mêmes valeurs bien que sémantiquement distincts
- ❑ Exemple. :
 $\text{NUM_ELV} = \{ 1, 2, \dots, 2000 \}$
 $\text{NUM_ANNEE} = \{ 1, 2, \dots, 2000 \}$

Les concepts

La Relation

- ❑ Une relation est un sous-ensemble du produit cartésien d'une liste de domaines caractérisé par un nom
- ❑ $R \subset D1 \times D2 \times \dots \times Dn$
D1, D2, ..., Dn sont les domaines de R
n est le degré de R

Les concepts

La Relation

❑ Les domaines :

- $NOM_ELV = \{ Rachid, Taoufiq \}$
- $PREN_ELV = \{ Amine, Sami \}$
- $DATE_NAISS = \{ 1/1/1990, 2/2/1994 \}$
- $NOM_SPORT = \{ judo, tennis, foot \}$

❑ La relation ELEVE

- $ELEVE \subset NOM_ELV \times PREN_ELV \times DATE_NAISS$
- $ELEVE = \{ (Rachid, Amine, 1/1/1992), (Taoufiq, Sami, 2/2/1994) \}$

❑ La relation INSCRIPT

- $INSCRIPT \subset NOM_ELV \times NOM_SPORT$
- $INSCRIPT = \{ (Rachid, judo), (Rachid, foot), (Taoufiq, judo) \}$

Les concepts

Les N-Uplets

- ❑ Un élément d'une relation est un n-uplet de valeurs (tuple en anglais)
- ❑ Un n-uplet représente un fait
- ❑ Exemple:
 - « Rachid Kamal est un élève né le 1 janvier 1994 »
 - « Rachid est inscrit au judo »

Les concepts

Attribut

- ❑ Un attribut est une variable prenant ses valeurs dans un domaine.
- ❑ Un attribut est une colonne d'une relation caractérisée par un nom. Rôle joué par un domaine dans la relation
- ❑ Analogie avec le modèle entité-association
 - les colonnes (attributs du modèle relationnel) ne sont rien d'autres que les attributs du modèle E-A
 - les lignes correspondent aux occurrences d'une entité ou d'une association du modèle E-A
- ❑ Exemple
 - La relation TRAJET : $\text{TRAJET} \subset \text{NOM_VILLE} \times \text{NOM_VILLE}$
 - Dans laquelle la première composante représente la ville de départ VD, la deuxième composante la ville d'arrivée VA d'un trajet.

Les concepts

Le Schéma d'une relation

- ❑ Le schéma d'une relation est défini par :
 - Le nom de la relation
 - La liste de ses attributs
 - On note : $R (A_1, A_2, \dots, A_n)$
- ❑ Exemple
 - ELEVE (NOM, PRENOM, NAISS)
 - INSCRIPT (NOM_ELIV, SPORT)
 - TRAJET (VD, VA)

Les concepts

Le Schéma d'une relation

Extension et Intension

- ❑ L'extension d'une relation correspond à l'ensemble de ses éléments (n-uplets)
 - le terme **RELATION** désigne une extension
- ❑ L'intention d'une relation correspond à sa signification
 - le terme **SCHÉMA DE RELATION** désigne l'intention d'une relation

Les concepts

Le Schéma d'une BDR

- ❑ Le schéma d'une base de données est défini par :
 - l'ensemble des schémas des relations qui la composent
- ❑ Notez la différence entre :
 - le schéma de la BDR qui dit comment les données sont organisées dans la base
 - l'ensemble des n-uplets de chaque relation, qui représentent les données stockées dans la base

Les concepts

La représentation

- ❑ Relations = Tables
- ❑ Élément ou n-uplet = Ligne
- ❑ Attributs = Colonnes
- ❑ Exemple

Colonne=attribut ou propriété



A1	B1	C1
A2	B2	C2
A3	B3	C3

Ligne=élément
ou n-uplet



Les concepts

La représentation

Exemples :

❑ La relation ELEVE

ELEVE

NOM	PRENOM	NAISSA
Rachid	Adil	11/05/2000
Taoufiq	Amine	12/09/1999
LAFAT	Rajaa	01/02/1996

Les concepts

Les concepts : La représentation

Exemples :

❑ La relation INSCRIPT

INSCRIPT

NOM_EL	SPORT
Rachid	JUDO
Taoufiq	FOOT
LAFAT	JUDO

Les concepts

Les concepts : La représentation

Exemples :

❑ La relation TRAJET

TRAJET

VD	VA
Casablanca	Marrakech
Marrakech	Ouarzazate
Ouarzazate	Casablanca

Règles de passage du modèle entité - association au modèle relationnel

Une entité se transforme en une relation (table)

- ❑ Toute entité du MCD devient une relation du MLDR, et donc une table de la Base de Données.
- ❑ Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante.
- ❑ L'identifiant de l'entité devient la **Clé Primaire** de la relation (elle est donc soulignée), et donc la **Clé Primaire** de la table correspondante.

Règles de passage du modèle entité - association au modèle relationnel

Une entité se transforme en une relation (table)

- ❑ Toute entité du MCD devient une relation du MLDR, et donc une table de la Base de Données.
- ❑ Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante.
- ❑ L'identifiant de l'entité devient la **Clé Primaire** de la relation (elle est donc soulignée), et donc la **Clé Primaire** de la table correspondante.

Exemple

Clients
NumClient
Société
Contact
Fonction
Adresse

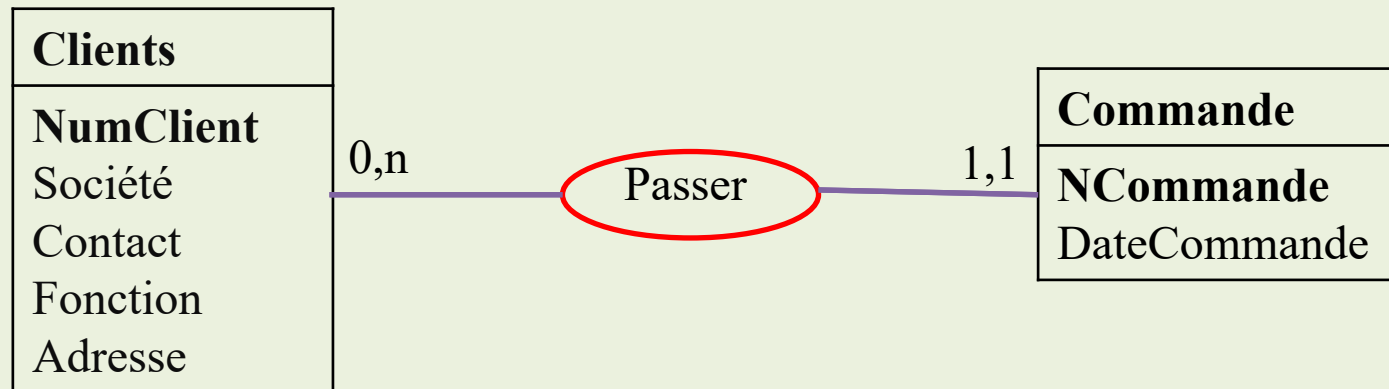
Clients (NumClient, Société, Contact, Fonction, Adresse)



Règles de passage du modèle entité - association au modèle relationnel

Relation binaire aux cardinalités (X,1) - (X,n), $X = \{0 \text{ ou } 1\}$

- ❑ La **Clé Primaire** de la table à la cardinalité (X,n) devient une **Clé Etrangère** dans la table à la cardinalité (X,1) :



MLD : Client (NumClient, Société, Contact, Fonction, Ville)
 Commande (NCommande, DateCommande, #NumClient)

Règles de passage du modèle entité - association au modèle relationnel

Relation binaire aux cardinalités (X,n) - (X,n), X= {0 ou 1}

- ❑ La création d'une table supplémentaire ayant comme **Clé Primaire** une clé composée des **identifiants** des 2 entités.
- ❑ La **Clé Primaire** de la nouvelle table est la **concaténation** des **Clés Primaires** des deux autres tables.

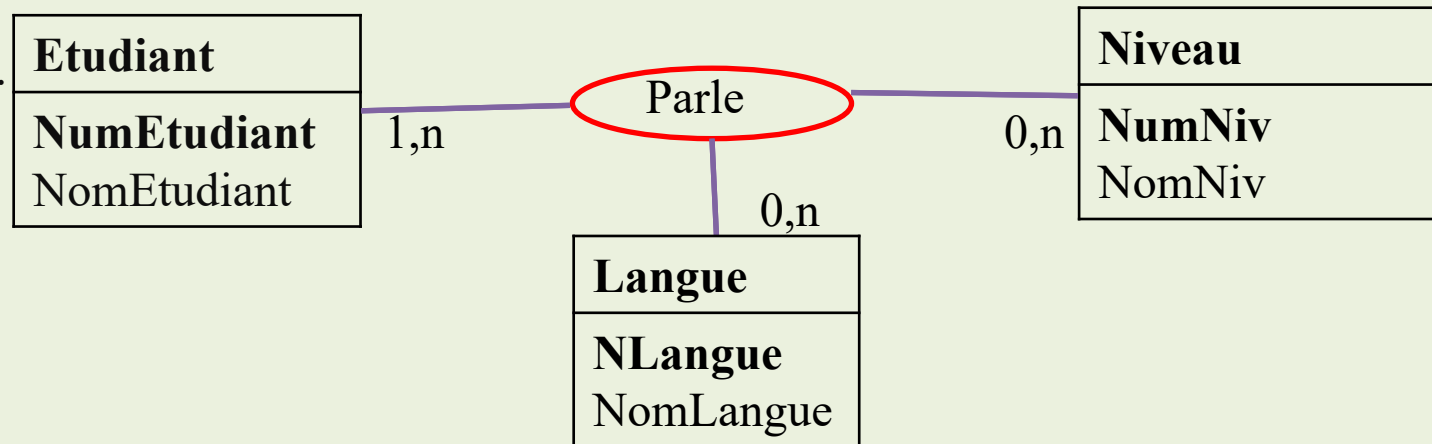


MLD : Commande (NCommande, DateCommande)
 Produit (NumProduit, NomProduit)
 DétailCom (#NCommande, #NumProduit, Quantité, Prix)

Règles de passage du modèle entité - association au modèle relationnel

Relation n-aire (quelles que soient les cardinalités)

- ❑ La création d'une table supplémentaire ayant comme **Clé Primaire** la **concaténation** des **identifiants** des entités participant à la relation.



MLD : Etudiant (NumEtudiant, NomEtudiant)
 Niveau (NumNiv, NomNiv)
 Langue (NLangue, NomLangue)
 Parle (#NEtudiant, #NumNiv, #NLangue)

Règles de passage du modèle entité - association au modèle relationnel

Relation binaire aux cardinalités (0,1) - (1,1)

- ❑ **Clé Primaire** de la table à la cardinalité (0,1) devient une **Clé Etrangère** dans la table à la cardinalité (1,1)



- ❑ **MLD:** Animateur (idAnimateur, NomAnimateur)
Groupe (idGroupe, #NomGroupe, #idAnimateur)

Dépendances fonctionnelles

Dépendance fonctionnelle

- ❑ Introduite par Codd en 1970,
- ❑ La notion de dépendance fonctionnelle (DF) permet de caractériser des relations qui peuvent être décomposée sans perte d'information.
- ❑ On dit qu'un attribut ou un ensemble d'attributs Y est fonctionnellement dépendant d'un autre (ensemble d') attribut(s) X si, à chaque valeur prise par X correspond une valeur unique de Y.
 - Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation
 - Soit X et Y des sous ensembles de $\{A_1, A_2, \dots, A_n\}$
 - On dit que Y dépend fonctionnellement de X ($X \rightarrow Y$) si à chaque valeur de X correspond une valeur unique de Y

Dépendances fonctionnelles

Dépendance fonctionnelle

- ❑ on écrit : $X \rightarrow Y$
- ❑ on dit que : X détermine Y
- ❑ Exemple:
 - PRODUIT (no_prod, nom, prixUHT)
 - $\text{no_prod} \rightarrow (\text{nom}, \text{prixUHT})$
 - NOTE (no_contrôle, no_élève, note)
 - $(\text{no_contrôle}, \text{no_élève}) \rightarrow \text{note}$
- ❑ une dépendance fonctionnelle est une propriété sémantique, elle correspond à une contrainte supposée toujours vrai du monde réel

Dépendances fonctionnelles

La clé d'une relation

- ❑ C'est un attribut qui détermine tous les autres
- ❑ Exemple:
 - PRODUIT (no_prod, nom, prixUHT)
 - $\text{no_prod} \rightarrow (\text{nom}, \text{prixUHT})$
 - no_prod est une clé
- ❑ Une clé détermine un n-uplet de façon unique
- ❑ Pour trouver la clé d'une relation, il faut examiner attentivement les hypothèses sur le monde réel
- ❑ Une relation peut posséder plusieurs clés, on les appelle clés candidates
- ❑ Exemple
 - dans la relation PRODUIT, nom est une clé candidate (à condition qu'il n'y ait jamais 2 produits de même nom)

Dépendances fonctionnelles

La clé d'une relation

- ❑ Clé primaire
 - choix d'une clé parmi les clés candidates
- ❑ Clé étrangère ou clé secondaire
 - attribut (ou groupe d'attributs) qui fait référence à la clé primaire d'une autre relation
 - Ex.: CATEG (no_cat, design, tva)
 - PRODUIT(no_prod, nom, marque, no_cat, prixUHT)
 - no_cat dans PRODUIT est une clé étrangère
- ❑ **CLÉ ÉTRANGÈRE = CLÉ PRIMAIRE dans une autre relation**

Les règles d'intégrité

Définition

- ❑ Une contrainte d'intégrité (CI) est une propriété ou une règle que doivent satisfaire les données de la base pour être considérées comme correctes (sans ambiguïtés ni incohérences).
- ❑ Une base de données est dite intègre ou cohérente si ses contraintes d'intégrité sont satisfaites
- ❑ Le modèle relationnel impose les contraintes structurelles suivantes :
 - Intégrité de domaine
 - Intégrité de clé
 - Intégrité référentielle

Les règles d'intégrité

Intégrité de domaine

- ❑ Les valeurs d'une colonne de relation doivent appartenir au domaine correspondant
 - - contrôle des valeurs des attributs
 - contrôle entre valeurs des attributs
- ❑ La valeur doit respecter le format des données du domaine de l'attribut (entier, réel, date, caractère)

Les règles d'intégrité

Intégrité de clé

❑ Les valeurs de clés primaires doivent être :

- uniques
- non NULL



- Unicité de clé
- Unicité des n-uplets

Les règles d'intégrité

Intégrité référentielle

- ❑ Les valeurs de clés étrangères sont 'NULL' ou sont des valeurs de la clé primaire auxquelles elles font référence



- Relations dépendantes

- ❑ LES DÉPENDANCES : Liaisons de un à plusieurs exprimées par des attributs particuliers: clés étrangères ou clés secondaires

Les formes normales

La théorie de la normalisation

- ❑ La méthode de normalisation consiste à créer des schémas relationnels répondant à un certain standard appelé forme normale qui devront respecter les contraintes de dépendances.
- ❑ La normalisation permet...
 - de guider la structuration des schémas relationnels,
 - de vérifier qu'une décomposition est à jonction conservatrice,
 - d'imposer des contraintes aux relations d'une BD.
- ❑ But
 - rendre une BD la plus efficace possible et sans redondances inutiles.

Les formes normales

La théorie de la normalisation

- ❑ Cette procédure fait subir à une relation une série de tests pour certifier qu'elle appartient à une certaine forme normale.
- ❑ Codd (mathématicien, chercheur de IBM, 1972) a proposé trois formes normales:
 - 1NF : première forme normale,
 - 2NF : deuxième forme normale,
 - 3NF : troisième forme normale.
- ❑ Ces formes normales sont basées sur les dépendances fonctionnelles entre les attributs d'une relation.

Les formes normales

1NF : Première forme normale

- ❑ La première forme normale contient 2 règles:
- R est une relation 1NF ssi les domaines de la relation contiennent uniquement des valeurs atomiques (ne contiennent qu'une seule information).
 - R est une relation 1NF ssi les domaines de la relation ne contiennent pas de relation.


DEPARTEMENT			
DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS
Research	5	333445555	(Bellaire,Sugarland,Houston)
Administration	4	987654321	(Stafford)
Headquarters	1	888665555	(Houston)

DEPARTEMENT			
DNAME	<u>DNUMBER</u>	DMGRSSN	<u>DLOCATION</u>
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Les formes normales

1NF : Première forme normale

- ❑ La première forme normale contient 2 règles:
 - R est une relation 1NF ssi les domaines de la relation contiennent uniquement des valeurs atomiques (ne contiennent qu'une seule information).
 - R est une relation 1NF ssi les domaines de la relation ne contiennent pas de relation.

 Les attributs composites ne sont pas permis !

Exemple:

- a) PROJS est inclus dans EMP_PROJ
- b) Instanciation
- c) EMP_PROJ décomposé

EMP_PROJ			
SSN	ENAME	PROJS	
		PNUMBER	HOURS

EMP_PROJ			
SSN	ENAME	PNUMBER	HOURS

EMP_PROJ1	
<u>SSN</u>	ENAME

EMP_PROJ2		
<u>SSN</u>	<u>PNUMBER</u>	HOURS

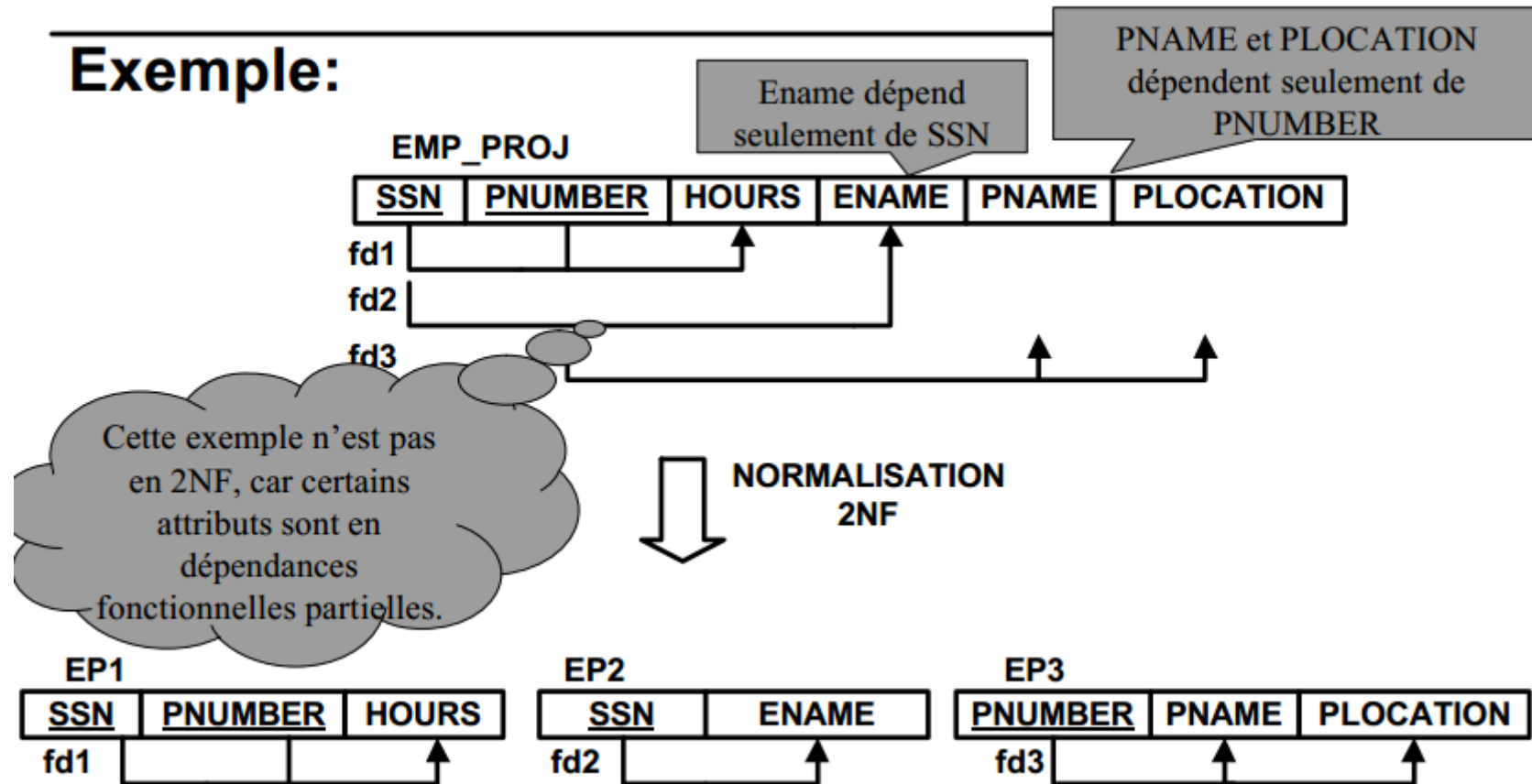
Les formes normales

2NF : Deuxième forme normale

- ☐ Une relation est en 2FN si
 - elle est en 1FN
 - tous les attributs “non clés” sont complètement dépendants de la clé primaire.
- ☐ La **2NF** est basée sur le concept de dépendance fonctionnelle complète.
- ☐ Une dépendance fonctionnelle $X \rightarrow Y$ est complète si l'élimination d'un attribut x_i quelconque de X détruit la dépendance fonctionnelle.

Les formes normales

Exemple:



Les formes normales

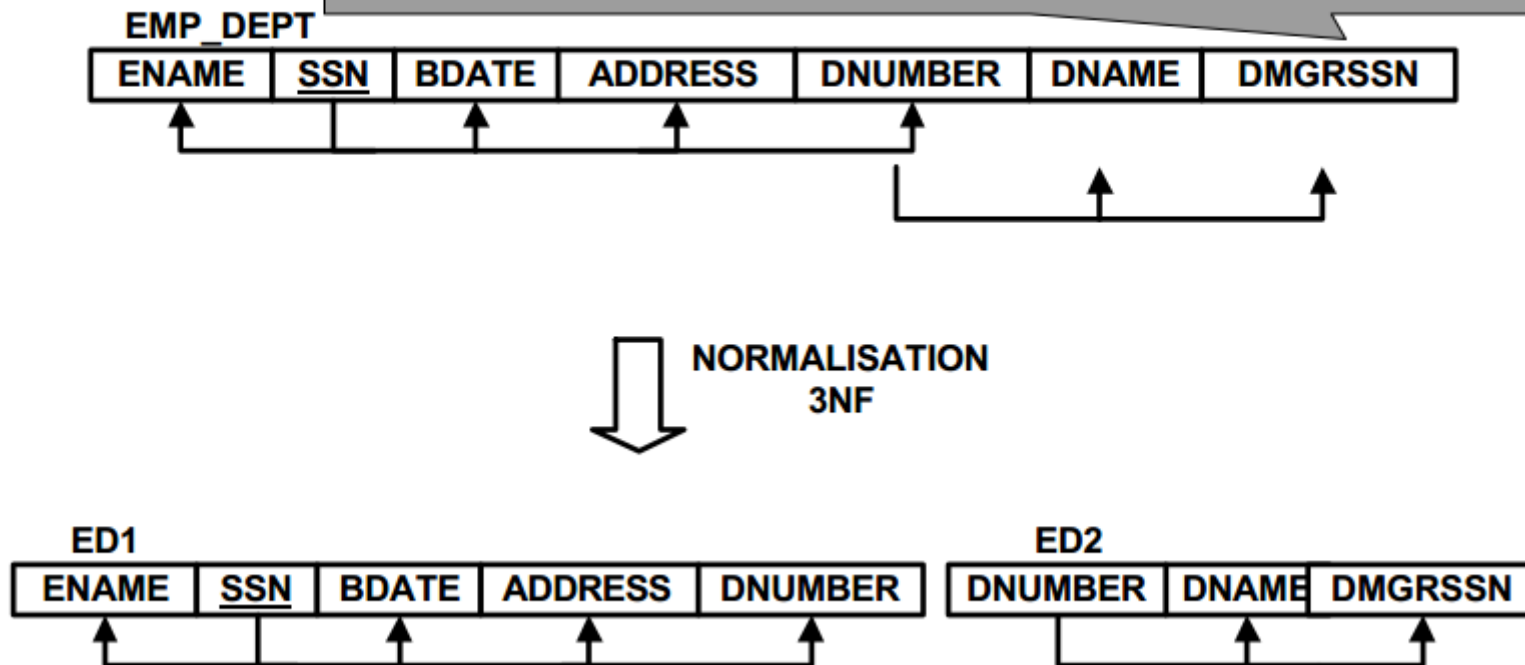
3NF: Troisième forme normale

- ❑ Une relation est en 3FN si
 - elle est en 2FN
 - tous les attributs “ non-clés ” sont dépendants non transitivement (dépendants directement) de la clé primaire.
- ❑ La **3NF** est basée sur le concept de dépendance fonctionnelle transitive.
 - Si les dépendances fonctionnelles $X \rightarrow Z$ et $Z \rightarrow Y$ existent. Alors, la dépendance fonctionnelle $X \rightarrow Y$ est dite transitive.

Les formes normales

Exemple

DMGRSSN est une dépendance fonctionnelle transitive de SSN puisqu'on peut l'obtenir par $SSN \rightarrow DNUMBER$ et $DNUMBER \rightarrow DMGRSSN$



Algèbre relationnelle

Introduction

- ❑ Algèbre relationnelle est une collection d'opérations formelles agissant sur des relations et produisant des relations en résultat.
- ❑ Par analogie, l'algèbre relationnelle est aux relations ce que l'arithmétique est aux nombres entiers.
- ❑ But: spécifier des interrogations sur une base de données pour en tirer des informations.
- ❑ Moyen:
 - combinaison de relations entre elles;
 - sélection de tuples;
 - sélection d'attributs

Algèbre relationnelle

Types d'opérations

Deux types d'opérations:

- ❑ Opérations ensemblistes
 - Union
 - Intersection
 - Différence
- ❑ Opérations spécifiques
 - Projection
 - Restriction
 - Jointure
 - Division

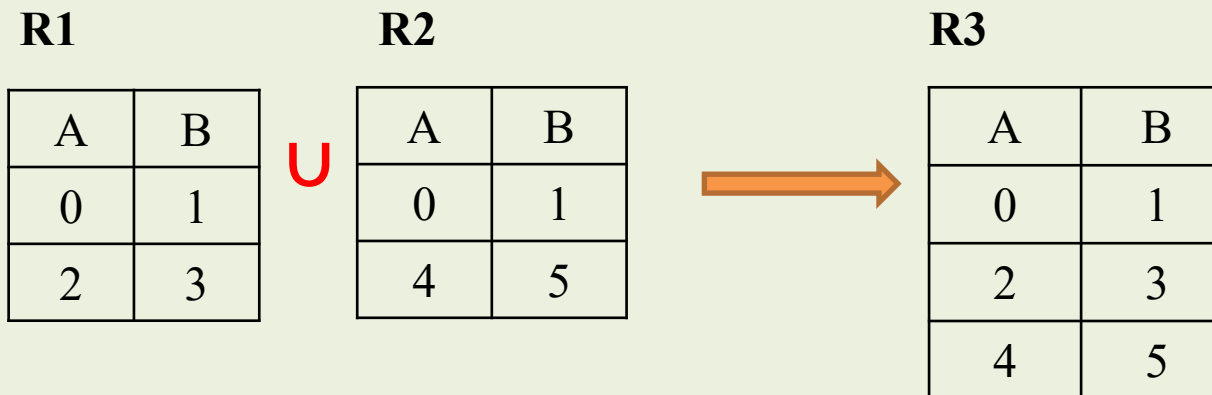
Types d'opérations

Union

- ❑ L'union de deux relations R1 et R2 de même schéma est une relation R3 de schéma identique qui a pour n-uplets les n-uplets de R1 et/ou R2
- ❑ On notera

$$R3 = R1 \cup R2$$

- ❑ Exemple



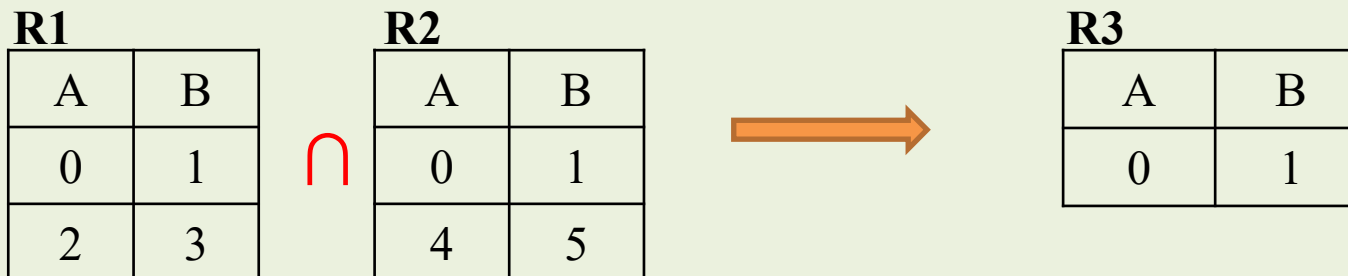
Types d'opérations

Intersection

- ❑ L'intersection entre deux relations R1 et R2 de même schéma est une relation R3 de schéma identique ayant pour n-uplets les n-uplets communs à R1 et R2
- ❑ On notera

$$R3 = R1 \cap R2$$

- ❑ Exemple



Types d'opérations

Différence

- ❑ La différence entre deux relations R1 et R2 de même schéma est une relation R3 de schéma identique ayant pour n-uplets les n-uplets de R1 n'appartenant pas à R2
- ❑ On notera

$$R3 = R1 - R2$$

- ❑ Exemple

R1

A	B
0	1
2	3

-

R2

A	B
0	1
4	5

**R3**

A	B
2	3

Types d'opérations

Produit cartésien

- ❑ Le produit cartésien de deux relations r et s , de schéma quelconque R et S , est une relation ayant pour attributs la concaténation de ceux de R et de S et dont les n -uplets sont toutes les concaténations d'un n -uplet de r à un n -uplet de s
- ❑ On notera

$$R3 = R1 \times R2$$

- ❑ Exemple

R1

No	NP	CP
100	X	B
200	Y	N

R2

X

ND	AD
5	A1
7	W

R3

No	NP	CP	ND	AD
100	X	B	5	A1
100	X	B	7	W
200	Y	N	5	A1
200	Y	N	7	W

Types d'opérations

Projection


- ❑ La projection d'une relation R1 est la relation R2 obtenue en supprimant les attributs de R1 non mentionnés puis en éliminant éventuellement les nuplets identiques
- ❑ On notera
$$\mathbf{R2 = \pi R1 (Ai, Aj, \dots, Am)}$$
la projection d'une relation R1 sur les attributs Ai, Aj, ... , Am
- ❑ La projection permet d'éliminer des attributs d'une relation

Types d'opérations

Projection


❑ **Exemple1** :Requête 1 : « Quels sont les noms et les prix des produits ? »

PRODUIT (IdPro, Nom, Marque, Prix)



IdPro	Nom	Marque	Prix
A	PS1	IBM	1000
B	Mac	Apple	2000
C	PS2	IBM	3000
D	Word	Microsoft	4000

π PRODUIT (Nom, Prix)




Nom	Prix
PS1	1000
Mac	2000
PS2	3000
Word	4000

Types d'opérations

Projection


❑ **Exemple2** :Requête 2 : « Quels sont les marques des produits ? »

PRODUIT (IdPro, Nom, Marque, Prix)



IdPro	Nom	Marque	Prix
A	PS1	IBM	1000
B	Mac	Apple	2000
C	PS2	IBM	3000
D	Word	Microsoft	4000

π **PRODUIT (Marque)**



Marque
IBM
Apple
Microsoft

Il faut éliminer les doublons..

Types d'opérations

Restriction

- ☐ La restriction ou la sélection d'une relation R1 est une relation R2 de même schéma n'ayant que les n-uplets de R1 répondant à la condition énoncée
- ☐ On notera : **$R2 = \sigma R1 (\text{condition})$**
la restriction d'une relation R1 suivant le critère "condition "
- ☐ La restriction permet d'extraire les n-uplets qui satisfont une condition
- ☐ La restriction ou la sélection supprime des lignes dans la table initiale.

Types d'opérations

Restriction

❑ **Exemple** : Requête : « Quels sont les produits de marques « IBM »? »
PRODUIT (IdPro, Nom, Marque, Prix)

	IdPro	Nom	Marque	Prix
→	A	PS1	IBM	1000
	B	Mac	Apple	2000
→	C	PS2	IBM	3000
	D	Word	Microsoft	4000

$\sigma_{\text{PRODUIT (Marque = 'IBM')}} \rightarrow$

IdPro	Nom	Marque	Prix
A	PS1	IBM	1000
C	PS2	IBM	3000

Types d'opérations

Jointure

- ❑ La jointure de deux relations R1 et R2 est une combinaison de tous les n-uplets de R avec ceux de S.

- ❑ On notera :

$$R3 = R1 \times R2 \text{ (condition)}$$

- ❑ la jointure de R1 avec R2 suivant le critère condition:
 - Le schéma de la relation résultat de la jointure est la concaténation des schémas des opérandes (s'il y a des attributs de même nom, il faut les renommer)
 - Les n-uplets de $R1 \times R2 \text{ (condition)}$ sont tous les couples (u1,u2) d'un n-uplet de R1 avec un n-uplet de R2 qui satisfont "condition "
 - La jointure de deux relations R1 et R2 est le produit cartésien des deux relations suivi d'une restriction
 - La condition de liaison doit être du type :
 $\langle \text{attribut1} \rangle :: \langle \text{attribut2} \rangle$
où : attribut1 \in 1ère relation et attribut2 \in 2ème relation :: est un opérateur de comparaison (égalité ou inégalité)

Types d'opérations

Jointure

- La jointure permet de composer 2 relations à l'aide d'un critère de liaison

R1 (A,B,C)

A	B	C
X	P	10
Y	Q	20

R2(U,V)

U	V
10	V1
20	V2

X

R1 × R2 (R1.C = R2.U)



A	B	C	U	V
X	P	10	10	V1
Y	Q	20	20	V2

Types d'opérations

Jointure

❑ **Exemple** :Requête : «Donnez pour chaque vente la référence du produit, sa désignation, son prix, le numéro de client, la date et la quantité vendue »

Vente (V)

IdCli	IdPro	Date	Qte
X	P	1/1/16	1
Y	Q	2/1/16	1
Z	P	3/1/16	1

Produits(P)

IdPro	Désignation	Prix
P	PS	100
Q	Mac	100

X

VENTE × PRODUIT (V.IdPro=P.IdPro)

IdCli	IdPro	Date	Qte	Désignation	Prix
X	P	1/1/16	1	PS	100
Y	Q	2/1/16	1	Mac	100
Z	P	3/1/16	1	PS	100

Types d'opérations

Auto-Jointure

- ❑ Jointure d'une relation par elle-même
- ❑ **Exemple** : Requête : «Quels sont les noms des clients qui habitent la même ville que Taoufiq ? »

Client(C1)

IdCli	Nom	Ville
X	Rachid	Rabat
Y	Adnane	Casa
Z	Taoufiq	Rabat

X

Client(C2)

IdCli	Nom	Ville
X	Rachid	Rabat
Z	Taoufiq	Rabat

Types d'opérations

Auto-Jointure

❑ $R1 = CLIENT \times CLIENT (C1.Ville = C2.Ville)$

C1.IdCli	C1.Nom	Ville	C2.IdCli	C2.Nom
X	Rachid	Rabat	X	Rachid
X	Rachid	Rabat	Z	Taoufiq
Y	Adnane	Casa	Y	Adnane
Z	Taoufiq	Rabat	X	Rachid
Z	Taoufiq	Rabat	Z	Taoufiq

Types d'opérations

Auto-Jointure

❑ $R2 = \sigma_{R1} (C2.Nom = 'Taoufiq')$

C1.IdCli	C1.Nom	Ville	C2.IdCli	C2.Nom
X	Rachid	Rabat	Z	Taoufiq
Z	Taoufiq	Rabat	Z	Taoufiq

❑ $R3 = \pi_{R2} (C1.Nom)$

C1.Nom
Rachid
Taoufiq

Types d'opérations

Division

- ❑ Soit deux relations $R1 (A1, A2, \dots, An, B1, B2, \dots, Bm)$
 $R2 (B1, B2, \dots, Bm)$
- ❑ Si le schéma de $R2$ est un sous-schéma de $R1$.
- ❑ La division de $R1$ par $R2$ est une relation $R3$ dont : -
 - le schéma est le sous-schéma complémentaire de $R2$ par rapport à $R1$
 - un n-uplet $(a1, a2, \dots, an)$ appartient à $R3$ si $(a1, a2, \dots, an, b1, b2, \dots, bm)$ appartient à $R1$ pour tous $(b1, b2, \dots, bm) \in R2$.
- ❑ On notera :
$$\mathbf{R3 = R1 \div R2}$$
 la division de $R1$ par $R2$
- ❑ La division permet de rechercher dans une relation les sous n-uplets qui sont complétés par tous ceux d'une autre relation
- ❑ Elle permet de répondre à des questions qui sont formulées avec le quantificateur universel : "pour tout ..."

Types d'opérations

Division

❑ **Exemple** : Requête : « Quels sont les élèves qui sont inscrits à tous les sports ? »

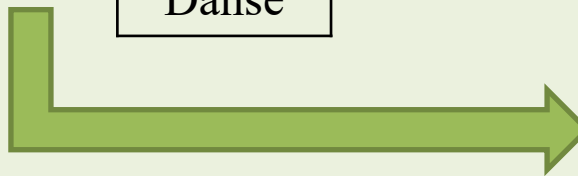
Inscri

Elève	Sport
Ahmed	Judo
Adnane	Foot
Ahmed	Foot
Ahmed	Danse

Sport

Sport
Judo
Foot
Danse

÷

**Res**

Elève
Ahmed

Le langage algébrique

Introduction

- ❑ Le langage algébrique permet de formuler une question par une suite des opérations de base de l'algèbre relationnelle

- ❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

- **Requête 1**

« Donner les no des produits de marque Apple et de prix <5000 dhs »

Le langage algébrique

Introduction

- ❑ Le langage algébrique permet de formuler une question par une suite des opérations de base de l'algèbre relationnelle

- ❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

- **Requête 1**

« Donner les no des produits de marque Apple et de prix <5000 dhs »

$R1 = \sigma_{\text{PRODUIT}} (\text{marque} = \text{'Apple'})$

$R2 = \sigma_{\text{PRODUIT}} (\text{prix} < 5000)$

$R3 = R1 \cap R2$

$\text{RESUL} = \pi_{R3} (\text{IdPro})$

Le langage algébrique

Introduction

- ❑ Le langage algébrique permet de formuler une question par une suite des opérations de base de l'algèbre relationnelle

- ❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

- **Requête 2**

« Donner les no des clients ayant acheté un produit de marque Apple »

Le langage algébrique

Introduction

- ❑ Le langage algébrique permet de formuler une question par une suite des opérations de base de l'algèbre relationnelle

- ❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

- **Requête 2**

« Donner les no des clients ayant acheté un produit de marque Apple »

$R1 = \sigma_{\text{PRODUIT}} (\text{marque} = \text{'Apple'})$

$R2 = R1 \times \text{VENTE} (R1.\text{IdPro} = \text{VENTE}.\text{IdPro})$

$\text{RESUL} = \pi_{R2} (\text{IdCli})$

Le langage algébrique

Introduction

❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

▪ Requête 3

« Donner les no des clients n'ayant acheté que des produits de marque Apple »

Le langage algébrique

Introduction

❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

▪ Requête 3

« Donner les no des clients n'ayant acheté que des produits de marque Apple »

$R1 = \text{VENTE} \times \text{PRODUIT} \text{ (VENTE.IdPro} = \text{PRODUIT.IdPro)}$

$R2 = \sigma R1 \text{ (marque} = \text{'Apple'})$

$R3 = \pi R2 \text{ (IdCli)}$

$R4 = \sigma R1 \text{ (marque} \neq \text{'Apple'})}$

$R5 = \pi R4 \text{ (IdCli)}$

$\text{RESUL} = R3 - R5$

Le langage algébrique

Introduction

❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

▪ Requête 4

« Donner les no des clients n'ayant acheté que des produits de marque Apple »

Le langage algébrique

Introduction

❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

▪ Requête 4

« Donner les no des clients n'ayant acheté que des produits de marque Apple »

$R1 = \text{VENTE} \times \text{PRODUIT} (\text{VENTE.IdPro} = \text{PRODUIT.IdPro})$

$R2 = \sigma R1 (\text{marque} = \text{'Apple'})$

$R3 = \pi R2 (\text{IdCli})$

$R4 = \sigma R1 (\text{marque} \neq \text{'Apple'})$

$R5 = \pi R4 (\text{IdCli})$

$\text{RESUL} = R3 - R5$

Le langage algébrique

Introduction

❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

▪ **Requête 5**

« Donner les no des clients ayant acheté tous les produits de marque Apple »

Le langage algébrique

Introduction

❑ Exemple

Requêtes sur le schéma CLIENT, PRODUIT, VENTE

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, désignation, marque, prix)

VENTE (IdCli, IdPro, date, qte)

▪ Requête 5

« Donner les no des clients ayant acheté tous les produits de marque Apple »

$R1 = \sigma_{\text{PRODUIT}} (\text{marque} = \text{'Apple'})$

$R2 = \pi_{R1} (\text{IdPro})$

$R3 = \pi_{\text{VENTE}} (\text{IdCli}, \text{IdPro})$

$R4 = R3 \div R2$