

# COURS BASE DE DONNÉES

## CHAPITRE 1 INITIATION AUX BASES DE DONNÉES

# Problématique

2

Une entreprise ou une organisation doit généralement manipuler **un nombre important de données** (Clients, produits, commandes, factures, .....).

**Exemple** : une facture se présente généralement de la manière suivante :

N° Commande		Date commande		
N° Client				
Nom client				
Coordonnées client				
N° Produit	Libellé Produit	P. U	Quantité	Total
Total commande				

# Problématique



3

Plusieurs problèmes se posent :

1/ Comment **conserver** ces informations dans un ordinateur.

2/ Comment **effectuer des traitements** (recherche, modification, ....)

Plusieurs données sont utilisées dans cette commande :

Données commande

N° Commande		Date commande		
N° Client Nom client Coordonnées client		Données client		
N° Produit	Libellé Produit	P.U	Quantité	Total
Total commande				

Détails commande

Données commande

# Concepts clés

5

A ce niveau il est important de distinguer entre les notions suivantes :

- **Donnée** : client, commande, produit ....etc
- **Valeur** : valeur de la donnée (exemple client n°xxx, commande n° yyyy, ....)
- **Information** : la commande n°xxxx se réfère au client n°yyyy
- **Connaissance** : une commande supérieure à X Dinars obtient une remise de 5%.

# Définition d'une BD

6

Une base de données est un ensemble **structuré de données** (1) enregistrées sur des **supports accessibles par l'ordinateur** (2) pour satisfaire **simultanément plusieurs utilisateurs** (3) de façon **sélective** (4) en **temps opportun** (5)

1: organisation et description des données

2 : stockage sur disque

3 : partage de données

4 : confidentialité (toute donnée doit pouvoir être protégée contre l'accès par les utilisateurs non autorisés en lecture ou en écriture)

5: l'aspect performance est important



## Exemples :

- Gestion du personnel, des étudiants, des cours, des inscriptions, etc., d'une faculté.
- Réservation des billets d'avion dans une agence de voyage
- Gestion des comptes clients d'une banque

# Définition d'un SGBD

7

- **Un SGBD est un logiciel qui permet d'interagir avec une base de données.**
- Un SGBD est un ensemble de programmes permettant de **créer, gérer et manipuler** une base de données **indépendamment** du domaine d'application de la base de données.
- C'est un ensemble de services (applications logicielles) permettant de :
  - ✓ **Définir la structure** d'une BD
  - ✓ **Rechercher des données** dans une BD
  - ✓ **Insérer, modifier, supprimer des données** dans une BD
  - ✓ Assurer la **cohérence** et la **confidentialité** des données

# Fonctionnalités d'un SGBD

Un SGBD assure les fonctions suivantes :

1. Description des données: **LDD**
2. Manipulation des données: **LMD**
3. Organisation efficace des données et accès rapide aux données.
4. Garantie de sécurité et de confidentialité des données.
5. Garantie de la cohérence des données en vérifiant les contraintes d'intégrité

# Fonctionnalités d'un SGBD (suite)

Un SGBD assure les fonctions suivantes :

6. Gestion de la concurrence
7. Résistance aux pannes
8. Administration de la BD

# Fonctionnalités d'un SGBD

10

➤ *Les SGBDs se différencient par :*

- leurs **coûts** respectifs,
- le **volume de données** qu'ils sont capables de gérer;
- le **nombre d'utilisateurs** qui peuvent interroger la base **simultanément**;
- **la facilité** avec laquelle ils s'interfacent avec les autres logiciels d'application utilisés par l'entreprise.

➤ Les SGBDs sont classés selon **leur modèle de données** (relationnel, hiérarchique, réseau, objet, Relationnel Objet).

**Exemples:** Oracle, MySQL, SQLServer, Access, DB2, Informix, etc.

# Modèles de données

C'est l'ensemble des concepts qui **permettent la description de la structure d'une BD et les règles d'utilisation de ces concepts** (Données, type de données, lien entre les données, opérateurs,...).

Les modèles de données présentent 3 niveaux :

1. **Le modèle conceptuel des données** : Description **graphique** des **données et des règles** exprimant les liens entre données
2. **Le modèle logique des données** : Description de **la structure de la base** de données
3. **Le modèle physique des données** : **Données proprement dites** dans leur structure physique.

# Modèles de données:

12

## Le modèle entité/association

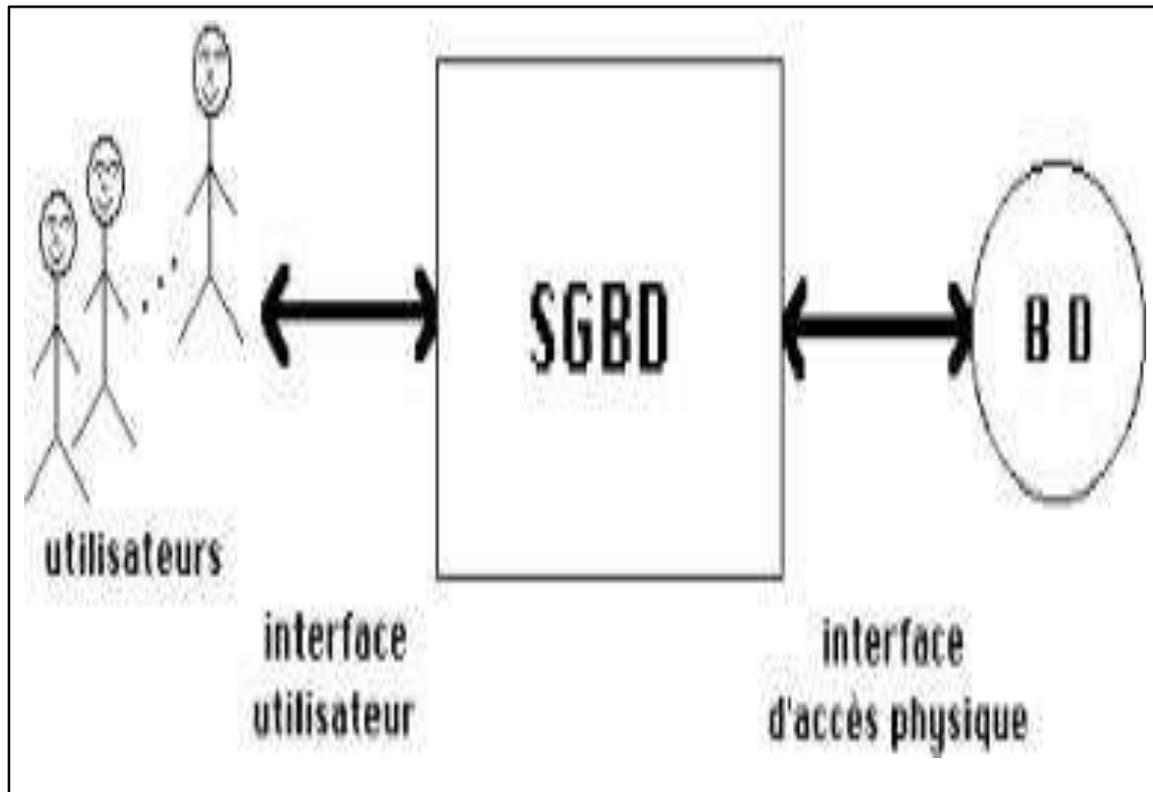
- Il fournit une **description graphique** pour représenter des données sous la forme de **diagrammes** contenant **des entités et des associations**.
- De tels modèles sont utilisés dans les phases amont de conception des systèmes informatiques.

## Le modèle relationnel

- Les données sont enregistrées **dans des tableaux 2D**.
- Une BD relationnelle est un ensemble de relations. **Elle est représentée sous forme de tables** (lignes représentant les enregistrements et colonnes représentant les attributs).

# Niveaux de représentation des données

13



**Figure: Architecture simple d'un SGBD**

# Niveaux de représentation des données

14

## Niveau interne ou physique :

Description de **la façon dont les données sont stockées sur des supports physiques**.  
On appelle cette description **le schéma interne**.

## Niveau conceptuel ou logique :

- Description de la structure de toutes les données de la base, leurs propriétés (c.-à.-d. les relations qui existent entre elles).
- C'est une description abstraite et globale du monde réel : **Schéma conceptuel ou logique**.

**→ Le schéma conceptuel décrit la structure de la BD indépendamment de son implantation.**

## Niveau externe ou vue :

- Le **niveau externe** permet aux différents utilisateurs **d'avoir des vues retréintes (réduites) de la BD** en fonction de leurs responsabilités et leurs besoins.
- Chaque utilisateur ou groupe d'utilisateurs **ne voit pas la totalité de la BD** mais juste une partie appelée : **vue ou niveau externe**. *« Cela dépend de ses droits d'accès (appelés aussi **privilèges**). »*

# Cycle de vie d'une BD

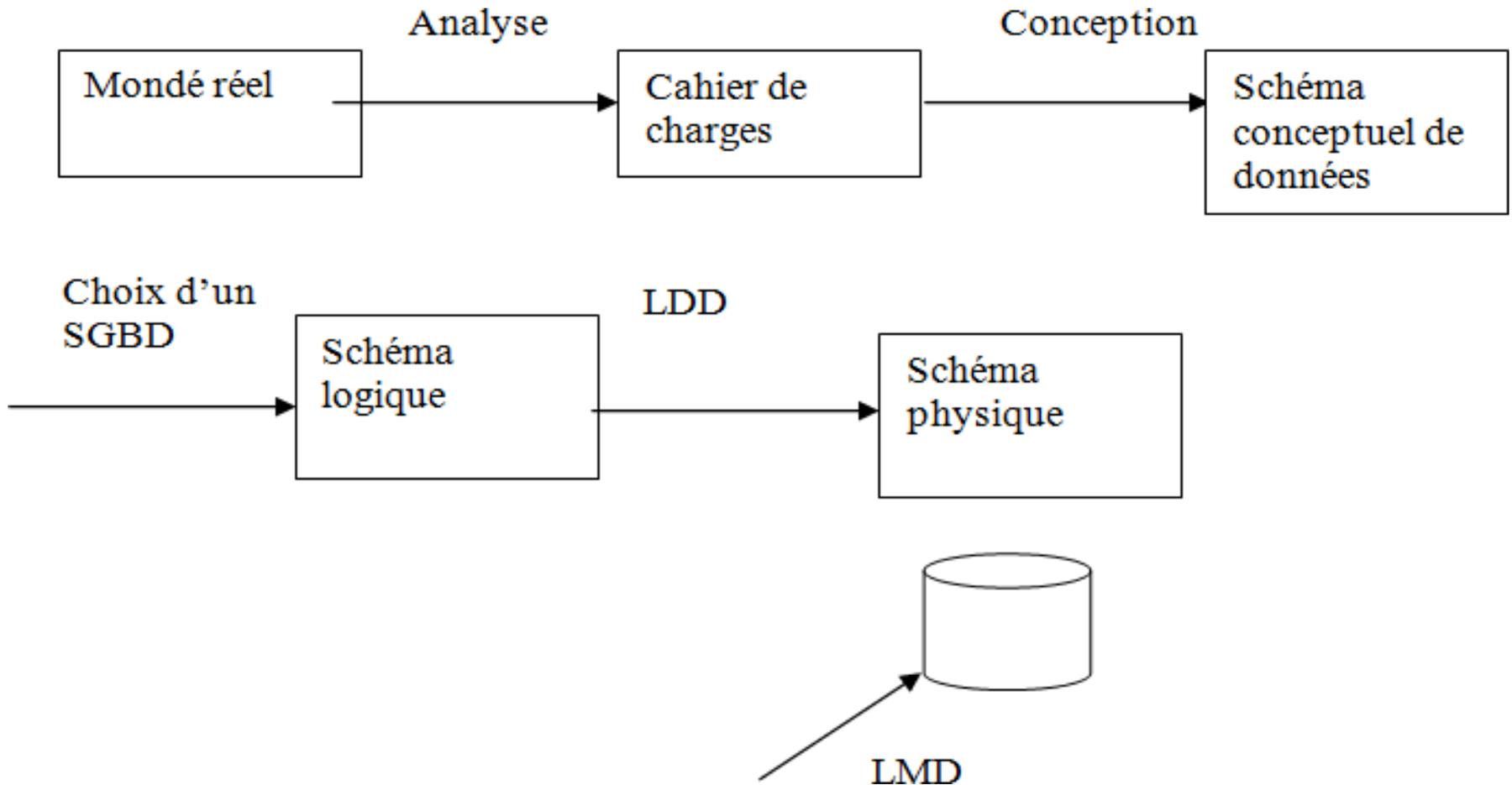
15

Le cycle de vie d'une BD comporte les étapes suivantes:

1. **Analyse des besoins**
2. **Conception de la base**
3. **Implantation des données** grâce à un Langage de Définition de Données (LDD) « (➔ *schéma interne*) »
4. **Utilisation** (interrogation, mises à jour) grâce à un Langage de Manipulation de Données (LMD) « (➔ **schéma externe**) »
5. **Maintenance** (correction, évolution) « (*qui peut affecter les différents schémas*) »

# Cycle de vie d'une BD

16



# CHAPITRE 2

## MODÈLE ENTITÉ/ASSOCIATION

# Introduction

2

## La modélisation des données sur le plan conceptuel

- aboutit à une **représentation schématique** de l'ensemble des données et de leurs liens, au sein du **domaine étudié**.
  - elle est nécessaire pour être en accord avec les différentes perceptions de la réalité et éviter les redondances.
  - il est difficile de construire directement les tables d'une base de données dans un SGBD
- La majorité des conceptions se basent sur le modèle Entité/Association.

# Définition

3

Le MCD modélise la mémoire collective du système.

- Description de la mémoire avec modèle entité association
  - Principe fondamental: Toute information qui circule dans le système doit être mémorisée une et une seule fois. (soit directement soit indirectement)
- *La description du Modèle Conceptuel de Données est non redondante.*
- **AUCUNE REDONDANCE !!!!**

# Définition

4

Son succès est dû au fait qu'il est :

- ✓ un langage graphique
- ✓ il se base sur des concepts simples :
  - le concept « objet » appelé « **entité** »
  - le concept liens entre les objets, appelé « **association** »
  - les propriétés importantes d'une l'entité ou d'une association appelées « **attributs** »
  - le regroupement des objets de même nature : types d'entités, types d'association.

# Principaux concepts du modèle Entité/Association

5

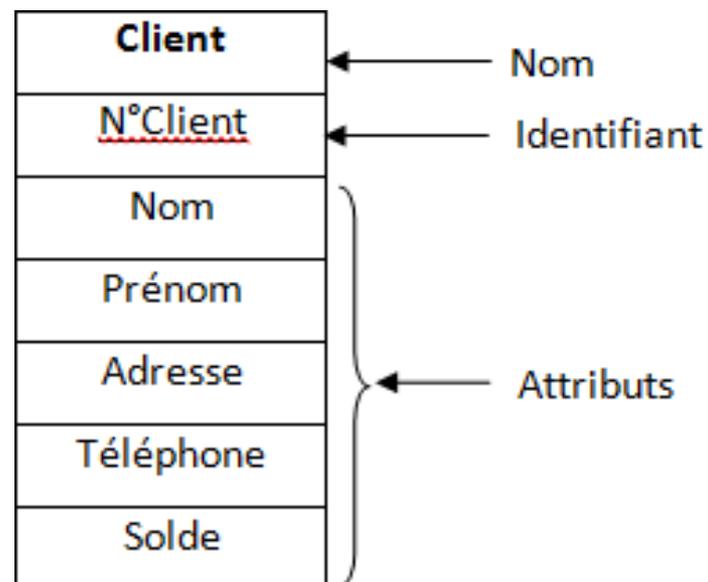
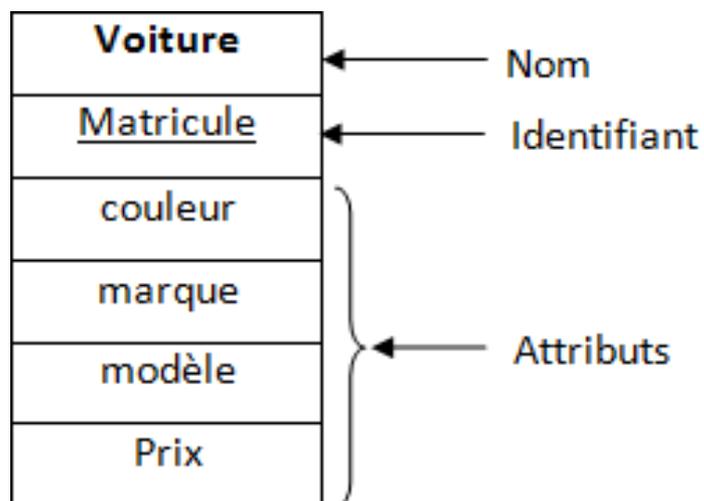
## Entité

- ✓ Représentation formelle de la mémorisation d'une information complexe et cohérente.
- ✓ *C'est un concept qui sert à décrire de façon générale plusieurs occurrences de la même information complexe.*
- ✓ Les entités permettent de décrire :
  - Des objets tangibles : employés, produits ... etc
  - Des objets abstraits : qualité, expertise, ...etc
  - Des transactions : contrat, commandes
- ✓ Une entité est totalement définie par **son nom, son identifiant (unique) et ses attributs (au moins 2).**

# Entité

6

## Exemples d'entités:



# Type d'Entité

7

- Un type d'entité désigne un ensemble d'entités qui ont les mêmes propriétés.
- Une entité fait partie d'un type entité.
- Elle est souvent nommée occurrence ou instance de son type d'entité.
- Il s'agit de l'ensemble des valeurs des attributs qui la caractérisent.
- Pour chaque entité il existe une ou plusieurs occurrences.

## Exemple: Type d'entité CLIENT

Client					
<u>NumClient</u>	Nom	Prénom	Adresse	Téléphone	Solde
133	<u>Sellami</u>	Mohamed	Z.I. Manouba	71 645 765	1.200,00
134	<u>Nebli</u>	Samira	Z.I. Ben Arous	71 380 476	2.800,00
136	<u>Chaieb</u>	Alia	Z.I. <u>Cherguia</u>	71 645 765	5.400,00
139	<u>Sellami</u>	<u>Firas</u>	Z.I. Manouba	71 645 765	3.200,000

Occurrences de l'entité CLIENT

- ✓ Un attribut (ou une propriété) est une caractéristique associée à un type d'entité ou à un type d'association.
- ✓ Chaque type d'entité ou type d'association est caractérisé par un ensemble d'attributs ou propriétés.
- ✓ elle permet de mémoriser une valeur ;
- ✓ elle doit avoir un sens (donc une valeur) pour chacune des occurrences de la composante du modèle (entité ou association) dont elle est un élément.
- ✓ L'ensemble des valeurs possibles d'une propriété constitue son domaine de valeurs.

Il existe différents types d'attributs :

- Simple : non divisible (ex: Nom)
- Composé : subdivisé en attributs simples (ex: Adresse (rue, code postal, ville, pays))
- Dérivé : dont la valeur est calculée (ex: Âge d'après la date de naissance)

# Identifiant d'une Entité

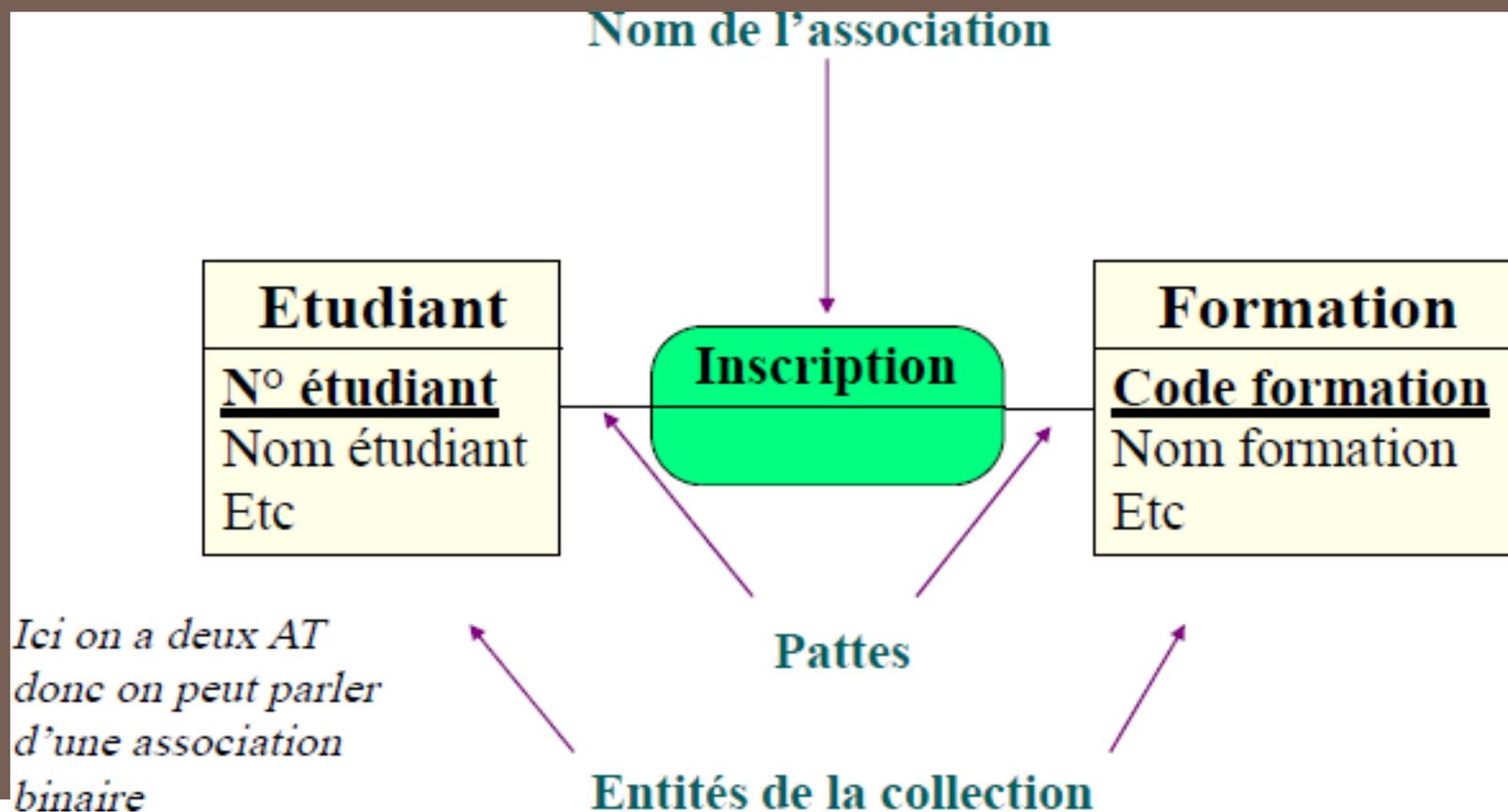
- ✓ Un identifiant (ou clé) d'un type d'entité ou d'un type d'association est constitué par un ou plusieurs de ses attributs qui doivent avoir une valeur unique pour chaque entité ou association de ce type.
- ✓ Une clé peut être :
  - Simple : cin, code
  - Composée : nom + prénom
- ✓ Chaque type d'entité possède un identifiant formé d'un ou plusieurs attributs.
- ✓ Dans la représentation graphique, les attributs qui constituent l'identifiant sont **soulignés et placés en tête**. Ils sont alors appelés **clé primaire**.

# Association

- ✓ **L'association** est la **représentation abstraite de la mémorisation d'un lien** entre des informations complexes (représentées par des entités).
- ✓ On appelle **collection** de l'association l'ensemble des entités types qu'elle relie.
- ✓ Une occurrence de l'association représente un **lien sémantique qui concerne une occurrence de chacune des entités de la collection.**

# Association

11



# Identifiant d'une association

12

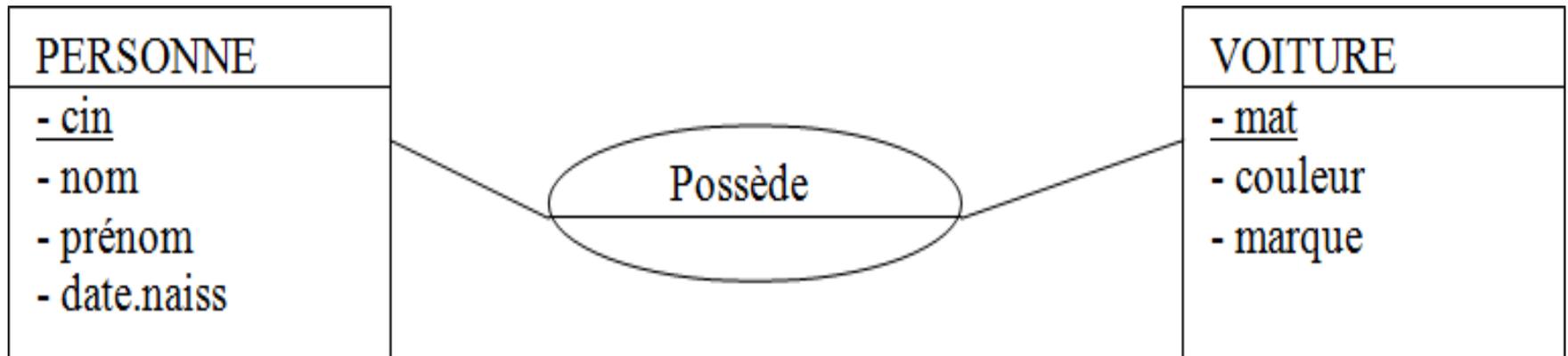
- ✓ Il est implicite
- ✓ C'est un couple composé des identifiants des entités types de sa collection.
- ✓ Ainsi dans l'exemple précédent, l'identifiant de *Inscription* est le couple (*N° étudiant*, *Code formation*).
- ✓ Un couple pour une association binaire ; un triplet pour une association ternaire etc.

# Type d'association

13

## Exemple d'association binaire:

Possède (entre VOITURE et PERSONNE)

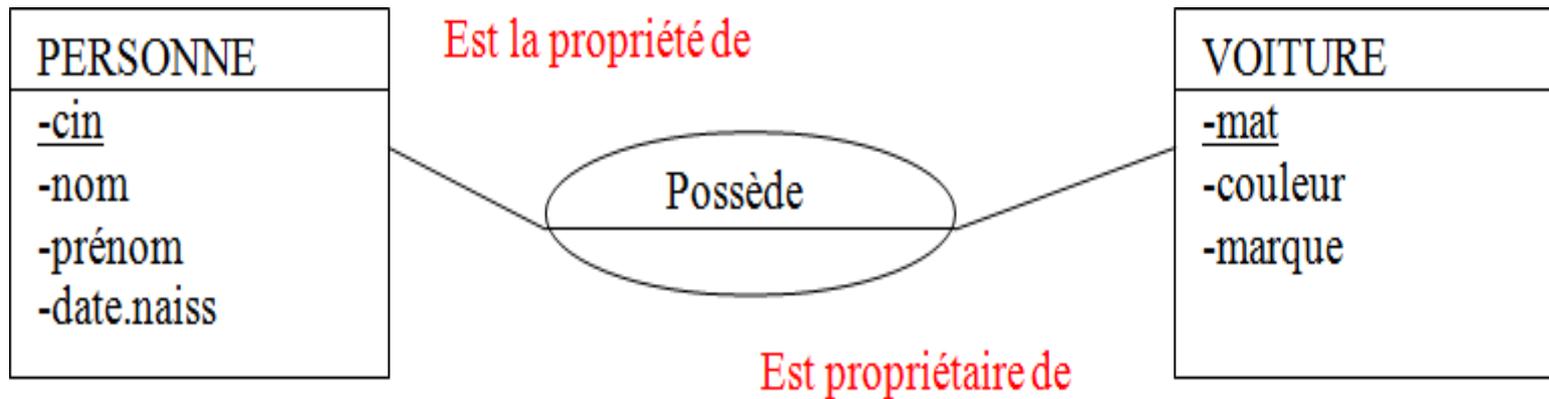


# Type d'association

14

## Remarque :

Le rôle détermine la façon dont les entités sont reliées.

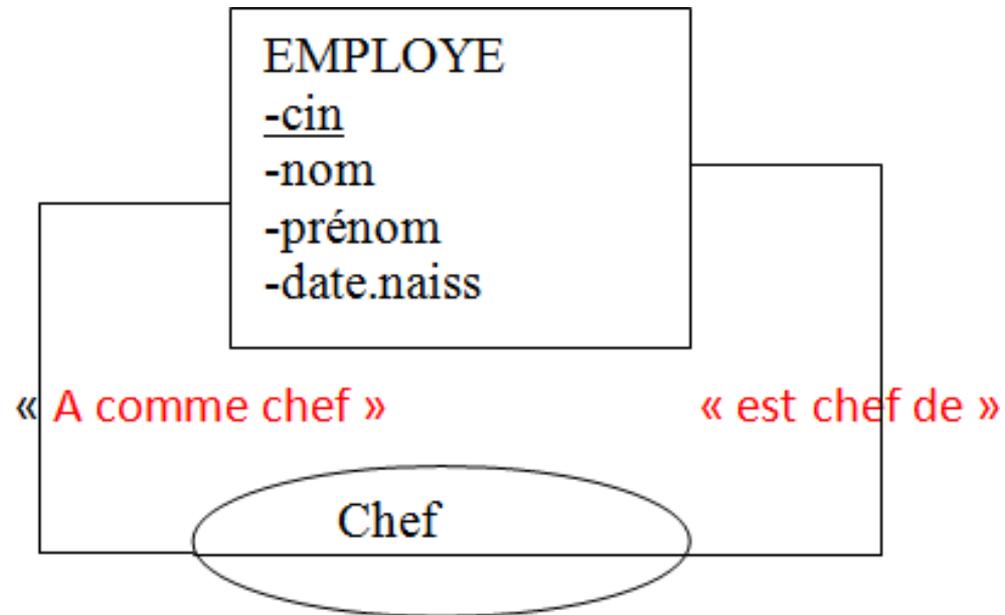


**ON NE PRÉCISE PAS LE RÔLE LORSQU'IL N'Y PAS D'AMBIGUÏTÉ!!!!**

# Type d'association

15

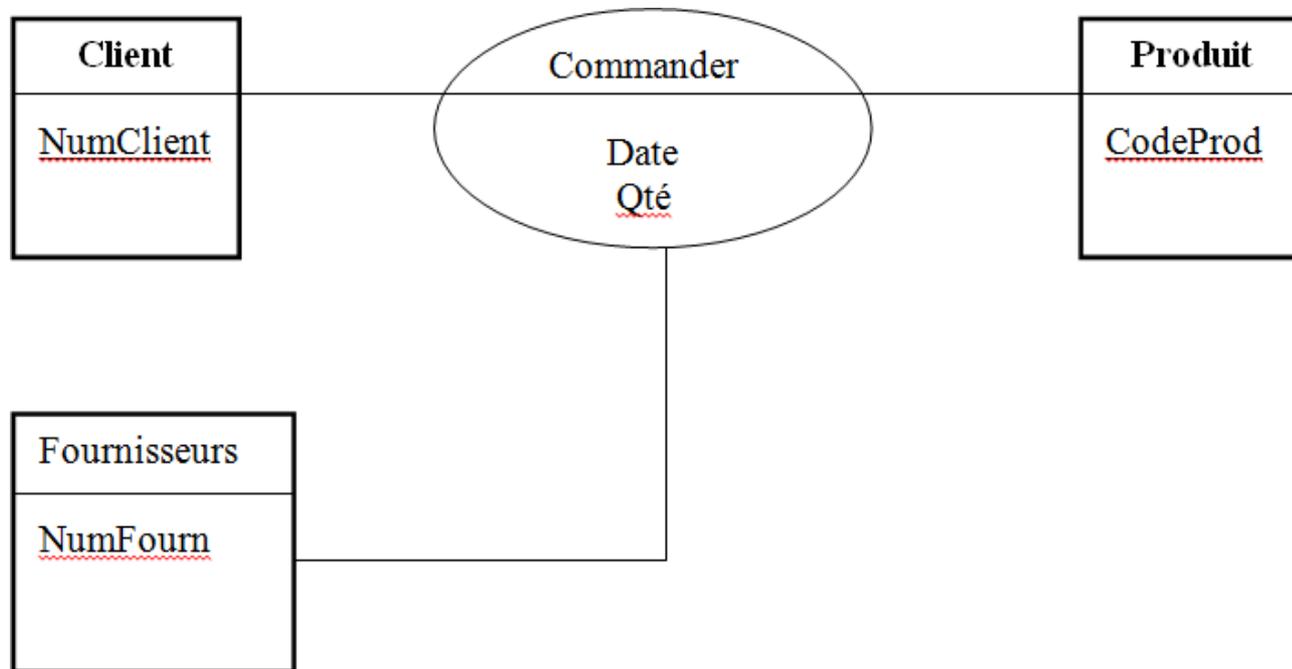
## Association Réflexive



# Type d'association

16

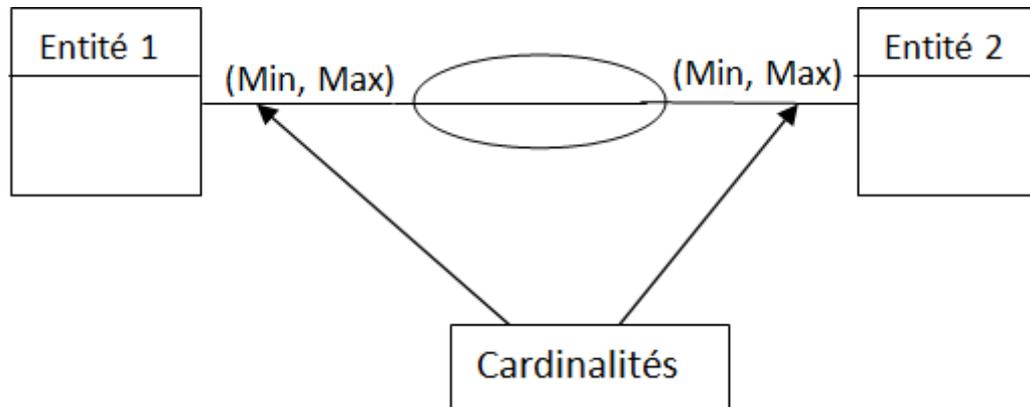
**Exemple d'association n-aire:** Les clients commandent des produits et les produits sont commandés auprès de fournisseurs



# Les contraintes de cardinalité

17

Une cardinalité indique combien de fois l'entité peut intervenir dans l'association



**La contrainte de cardinalité minimale peut prendre deux valeurs : 0 ou 1.**

**0** signifie que l'entité peut ne pas intervenir dans l'association.

**1** signifie au contraire qu'elle intervient obligatoirement une fois.

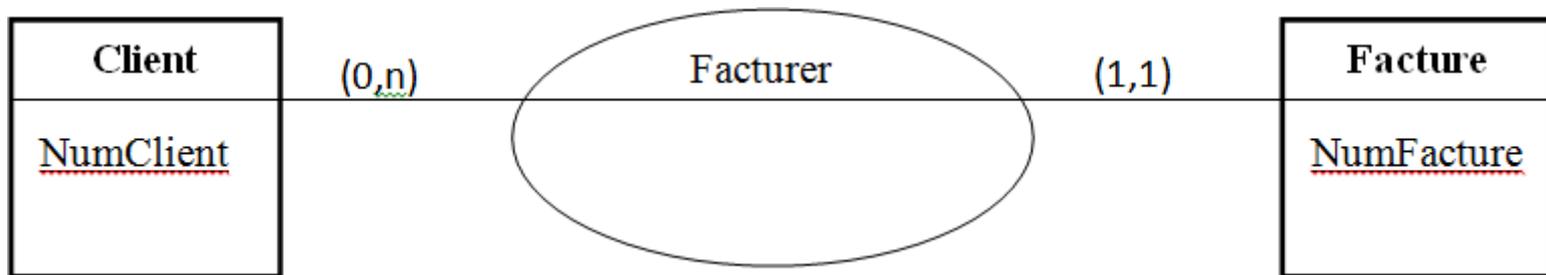
**La contrainte de cardinalité maximale peut prendre deux valeurs : 1 ou n.**

**1** signifie que l'entité ne peut intervenir plus d'une seule fois dans l'association ; **n** signifie au contraire qu'elle peut intervenir plusieurs fois dans l'association.

# Les contraintes de cardinalité

18

## Exemple



Un client peut avoir 0 ou n factures. Une facture ne peut concerner qu'un et un seul client.

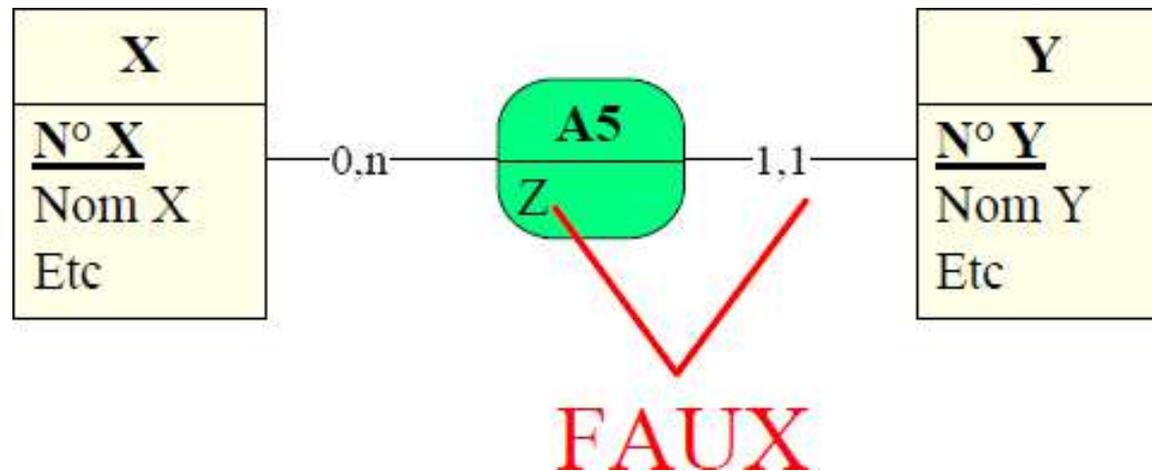
- ✓ **Cardinalité minimale est obligatoirement  $\leq$  cardinalité maximale.**
- ✓ Les cardinalités présentant des valeurs (0,1) ou (1,1) entre une association et une entité, est une association particulière appelée **Contrainte d'Intégrité Fonctionnelle** (CIF).
- ✓ Les autres d'associations (ne présentant pas les cardinalités(0,1) ou (1,1) entre une association et une entité) sont de type **Contrainte d'Intégrité Multivaluée** (CIM).

# Les contraintes de cardinalité

19

## Règle 1 (Association binaire « CIF »)

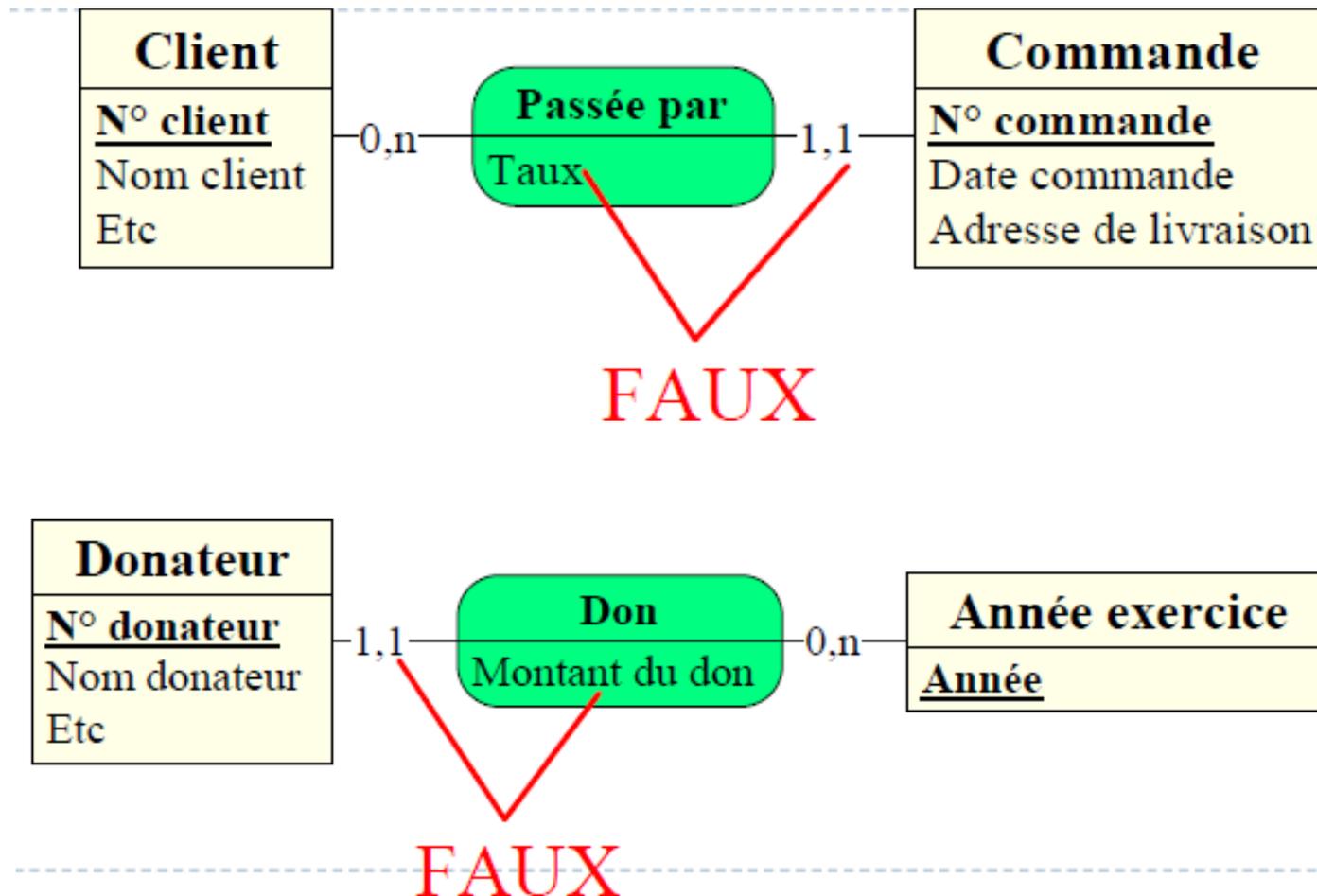
Une association binaire fonctionnelle ne peut en aucun cas porter de propriété.



# Les contraintes de cardinalité

20

## Exemples :

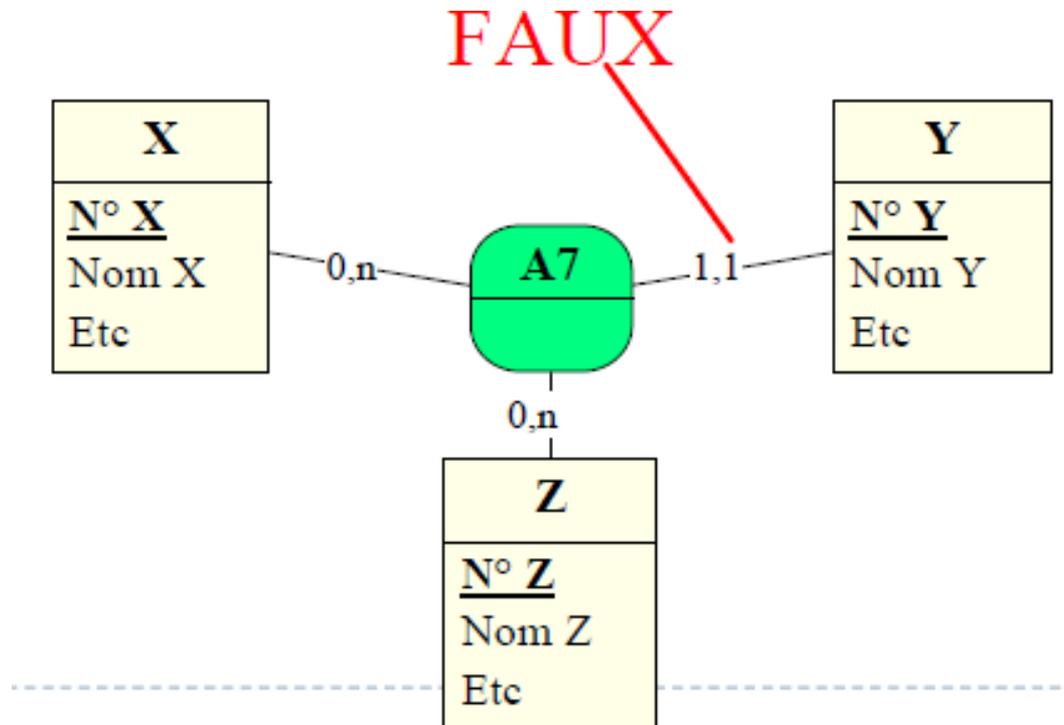


# Les contraintes de cardinalité

21

## Règle 2 (Association ternaire)

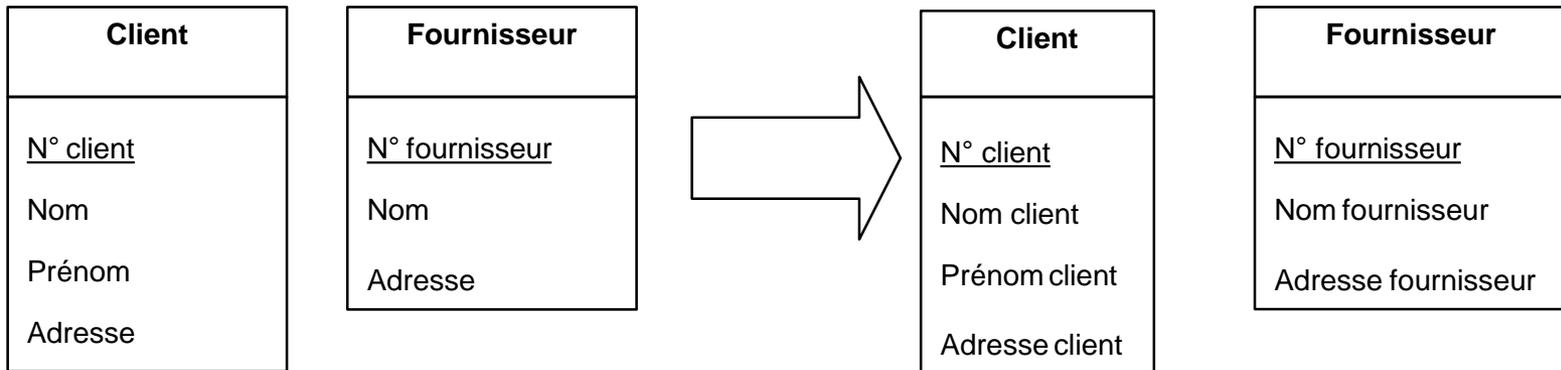
Toutes les cardinalités maximum d'une association ternaire (et au-delà) sont égales à n.



## Règle 1:

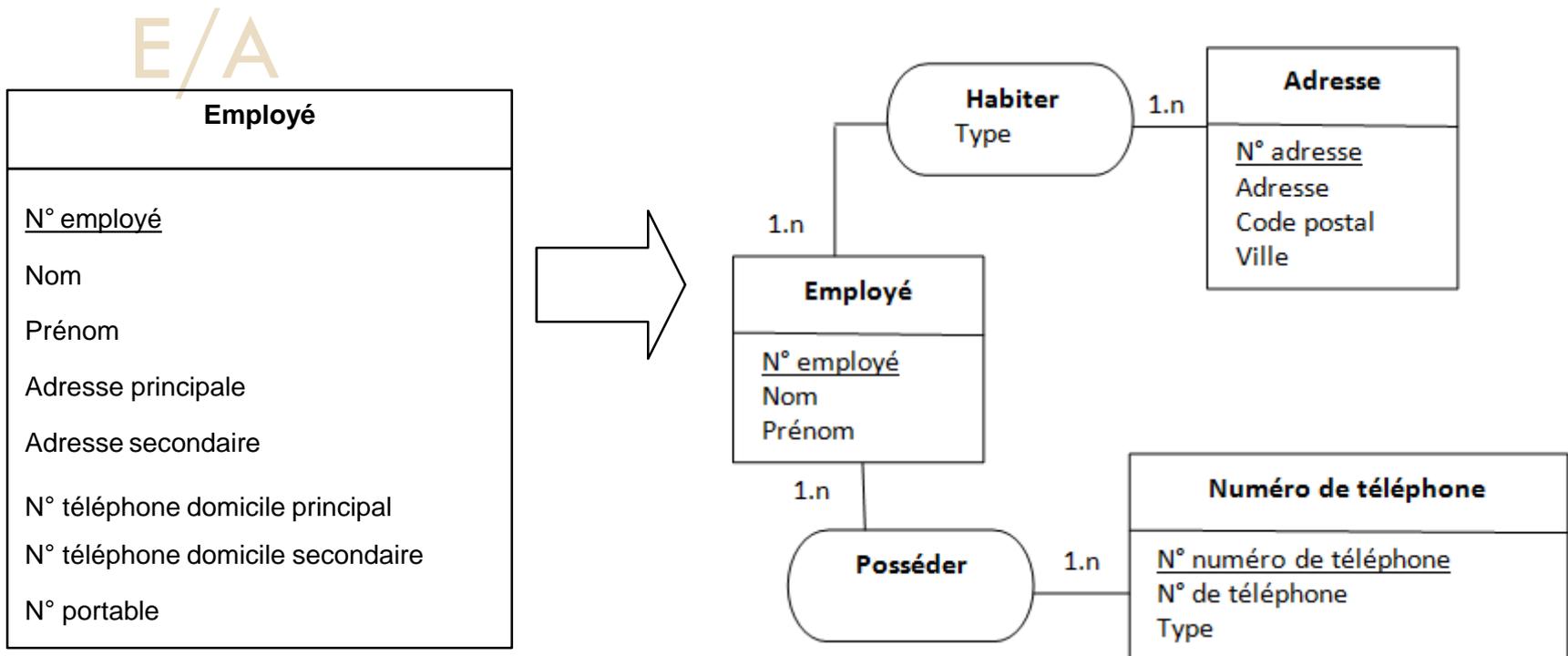
Dans un modèle entités-associations, le nom d'un type d'entité, d'un type d'association ou d'un attribut doit être unique.

E/A



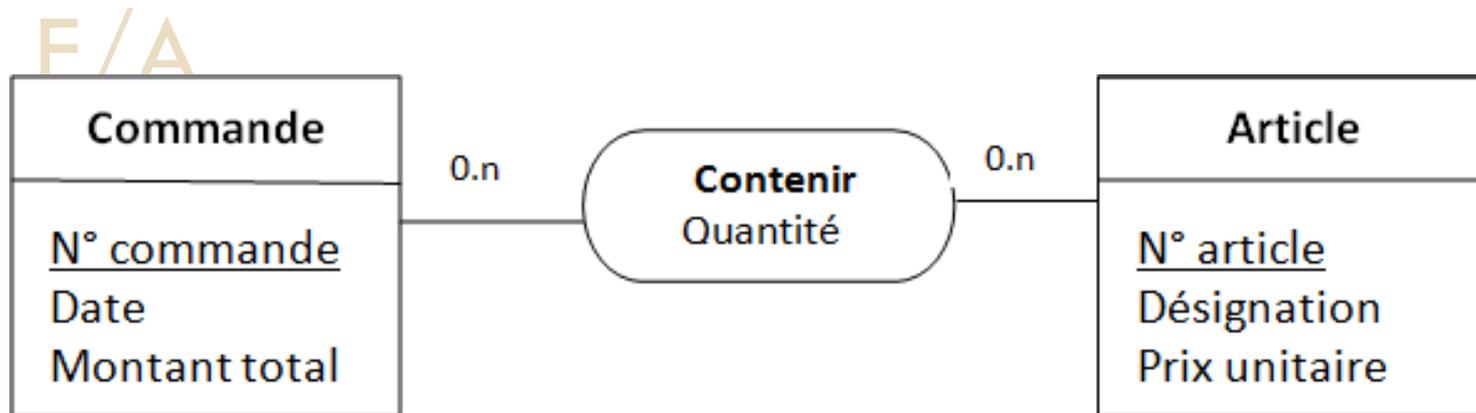
## Règle 2:

Un attribut multiple doit être transformé en un type d'association et un type d'entité supplémentaires.



### Règle 3:

Il ne faut jamais ajouter un attribut dérivé d'autres attributs (que ces autres attributs se trouvent dans le même type d'entité ou pas).

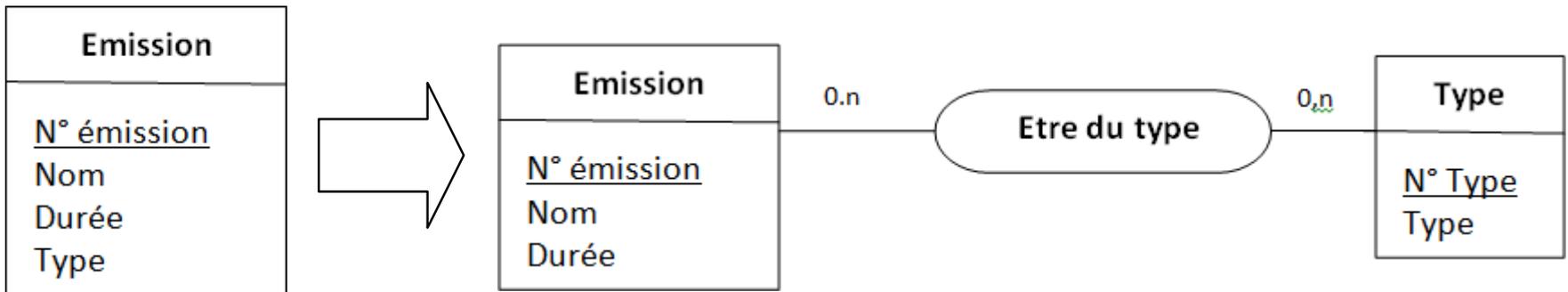


Il faut supprimer l'attribut Montant total du type d'entité Commande car on peut le calculer à partir des attributs Quantité du type association Contenir et Prix unitaire du type d'entité Article.

### Règle 4:

Un attribut correspondant à un type énuméré est généralement remplacé par un type d'entité (éventuellement en introduisant un nouvel attribut).

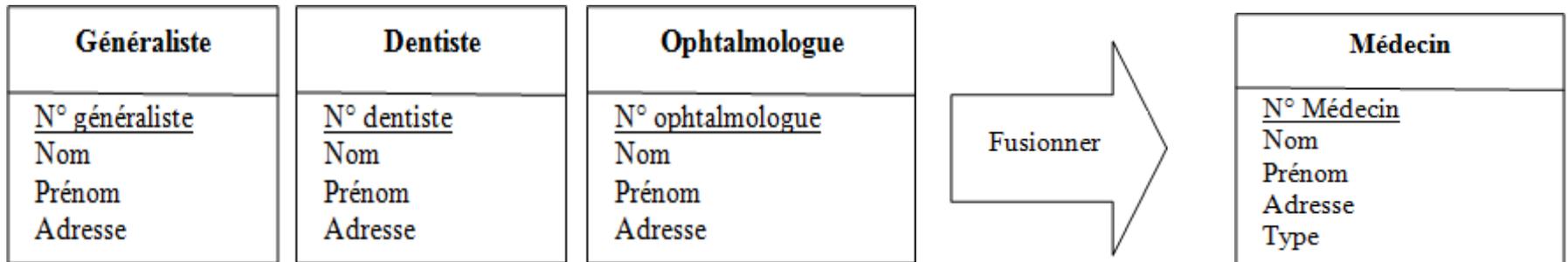
E/A



**Règle 5:**

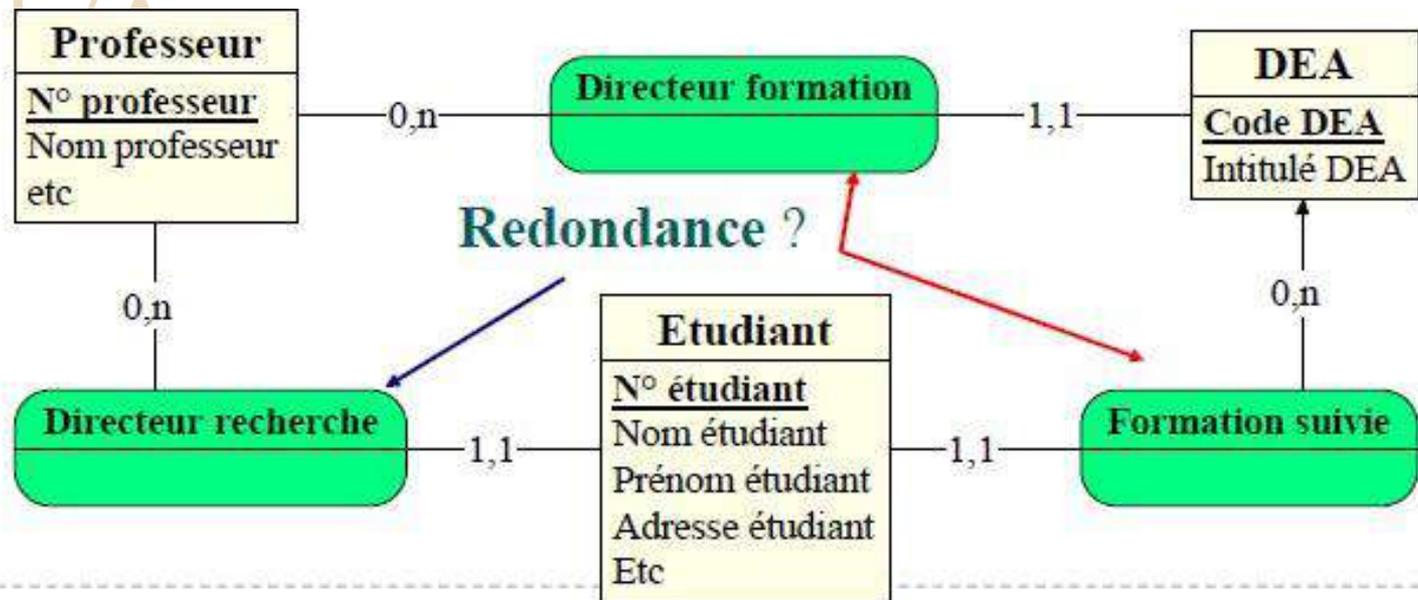
Il faut factoriser les types d'entités quand c'est possible.

Les règles de vérification du M  
E/A



## Règle 6:

Il faut veiller à éviter les types d'associations redondants



# CHAPITRE 3

## MODÈLE LOGIQUE DES DONNÉES

# Rappel: Etapes de création d'une base de données

2

Analyse des besoins de l'entreprise



Dictionnaire de Données

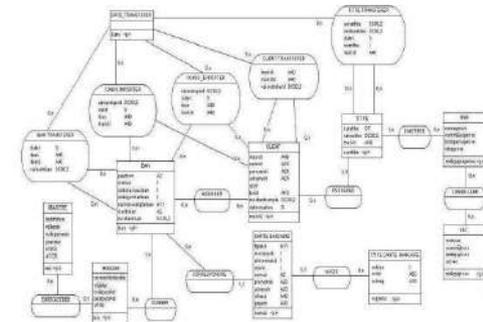


Modèle Conceptuel de Données



**Le dictionnaire de données**

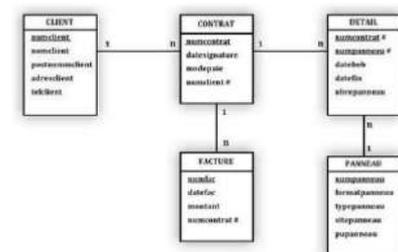
nom	Format	Longueur	Type	Règle de calcul	Règle de gestion	Document
Numéro	Numérique		N	C		Fiche
Nom	Alphabétique	30	X			//
Prénom	Alphabétique	30	X			//
Adresse	Alphabétique	50	X			//
Code Postal	Alphanumérique	10	X			//
Ville	Alphabétique	50	X			//
Téléphone	Alphanumérique	15	X			//
Mail	Alphanumérique	50	X			//
Date d'adhésion	Date		X			//



Modèle Physique de Données



Modèle Logique de Données



# Introduction

3

- Le modèle relationnel a été proposé par CODD en 1970.
- Les données sont représentées par des tables
- Une BD relationnelle est représentée par un ensemble de tables dont les occurrences sont appelées **tuples ou n-uplets** de ces relations.
- Les entités et les associations du modèle E/A sont représentées exclusivement par des relations de la manière suivante :

- **Une entité est représentée par sa liste d'attributs.**

**Exemple** : Schéma de la relation CLIENT  
CLIENT (Numclient, Nom, Adresse, Solde)

- **Une association est représentée par la liste des clés des entités qu'elle associe et ses propres attributs.**

**Exemple** : Schéma de la relation Facturer  
Facturer (Numclient, Numfacture, Période)

# Éléments du domaine relationnel

## ❖ *Attribut*

Un attribut représente une caractéristique d'une table la BD

### Exemple :

L'âge d'une personne, le nom d'une personne, le numéro de sécurité sociale.

## ❖ *Domaine*

Le domaine d'un attribut  $A$ , **noté  $Dom(A)$** , est l'ensemble de ses valeurs possibles.

### Exemple :

L'attribut numéro de la carte d'identité nationale a pour domaine l'ensemble des combinaisons de huit chiffres.

L'attribut nom a pour domaine l'ensemble des chaînes de caractères de longueur 20.

# Éléments du domaine relationnel

5

## ❖ **Relation**

Une relation est représentée sous la forme d'un tableau à deux dimensions dans lequel les  $n$  attributs correspondent aux titres des  $n$  colonnes.

### Exemple :

La relation  
Voiture

nuplets ou  
tuples ou  
lignes

Colonnes ou attributs

mat	Couleur	Energie	marque
1000	Rouge	Essence	peugeot
1001	Bleue	Essence	fiat
1002	Blanche	Diesel	citroen

## ❖ **Degré de relation**

Le degré d'une relation = nombre d'attributs de la relation.

### Exemple :

La table voiture comporte 4 attributs, ainsi, elle a un degré = 4.

# Éléments du domaine relationnel

6

## ❖ **Cardinalité de relation**

La cardinalité (population) d'une relation = nombre de lignes de la relation

Exemple :

La table voiture comporte 3 lignes, ainsi, sa cardinalité = 3

Remarque : Le degré est une constante mais la cardinalité est variable (toujours  $\geq 0$ ).

Une **Occurrence ou n-uplets ou tuple**, , est une ligne de la relation.

## ❖ **Clé candidate**

Une clé candidate d'une relation est un ensemble minimal des attributs de la relation tel que il n'existe pas 2 tuples ayant les mêmes valeurs pour tous ses attributs.

Remarque : Toute relation a au moins une clé candidate et peut en avoir plusieurs. Ainsi, il ne peut jamais y avoir deux tuples identiques au sein d'une relation.

# Éléments du domaine relationnel

7

## ❖ Clé primaire

La clé primaire d'une relation est une de ses clés candidates. Elle doit être soulignée. Tous les attributs de cette clé doivent **avoir toujours une valeur connue**. Chaque table possède une seule clé primaire. Le reste des clés s'appelle **clés secondaires**.

## ❖ Clé étrangère

Une clé étrangère dans une relation est formée d'un ou plusieurs attributs qui constituent une clé primaire dans une autre relation, sachant que chacun des attributs de cette clé est précédé par un « # ».

### Exemple :

Propriétaire (cin, nom, prénom, adresse)

Voiture (mat, couleur, energie, marque, #cin\_prop)

# Éléments du domaine relationnel

8

## ❖ **Schéma de relation**

Un schéma de relation précise le nom de la relation ainsi que la description des attributs de la relation.

### Exemple :

Voiture (mat : entier, couleur : chaîne, énergie : chaîne, marque : chaîne, âge : entier)

## ❖ **Schéma relationnel**

Un schéma relationnel est constitué par l'ensemble des schémas de relation de la BD (c'est-à-dire de toutes les tables de la BD).

## ❖ **Base de données relationnelle**

Une base de données relationnelle est constituée par l'ensemble des n-uplets des différentes relations du schéma relationnel (ensemble des instances de la BD).

# Éléments du domaine relationnel

9

## **Remarques :**

- Toutes les relations dans un schéma relationnel doivent avoir des noms différents.
- Les attributs d'une relation doivent avoir des noms différents.
- L'ordre des colonnes n'est pas important dans une table.
- L'ordre des lignes n'est pas important dans une table.
- Chaque relation doit avoir une clé primaire.



# Les contraintes d'intégrité imposées par le MR

10

Les relations ne sont pas indépendantes l'une de l'autre. Il existe 3 contraintes principales qui permettent d'assurer la cohérence de la BD:

## ❖ **Contrainte d'unicité**

Tous les tuples de la relation doivent avoir des valeurs distinctes pour la clé primaire.

## ❖ **Contrainte d'entité**

La clé primaire ne prend jamais la valeur nulle.

**Remarque (valeur nulle)** : Si on ne connaît pas une valeur d'un attribut alors on dit qu'on a une valeur nulle notée NULL.

## ❖ **Contrainte référentielle (contrainte de clé étrangère) :**

Une colonne **A** d'une relation **R** fait référence à une colonne **B** d'une relation **S** si toutes les valeurs **non nulles** de A figurent dans B.

# Les contraintes d'intégrité imposées par le MR

11

## Exemple

Voiture

Mat	couleur	marque	Cin_prop
1001	rouge	peugoet	1
1002	blanche	citroen	2

Personne

Cin	Nom	Prénom	adresse
1	X	Y	Tunis
2	X	Z	Ariana
3	Y	Y	Tunis

- Toute valeur de la colonne cin\_prop de voiture doit figurer dans la colonne cin de personne. Par contre, **l'inverse n'est pas nécessaire**.  
$$\text{Voiture.cin\_prop} \subseteq \text{Personne.cin}$$
- Il est interdit d'ajouter un tuple dans la table voiture avec un cin\_prop qui n'existe pas dans la table personne.
- Il est interdit de supprimer un tuple de la table personne où sa cin se trouve dans la cin\_prop de la table voiture.

❖ Règle 1 :

Chaque entité du schéma E/A devient une relation ayant les mêmes attributs que l'entité. L'identifiant constitue la clé de la relation.

Exemple:

Employé
<u>Cin</u> nom prénom

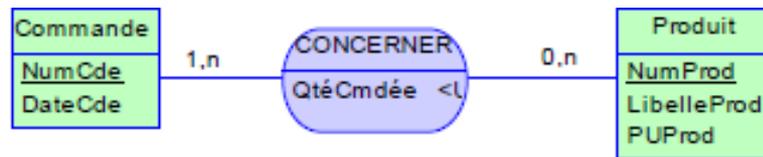


Le TE Employé sera transformé en Employé(cin, nom, prénom) qui est équivalent à :

<u>cin</u>	Nom	Prénom

### ❖ Règle 2:

Si une association ne possède aucune cardinalité maximale égale à 1, cette association devient une relation ayant les mêmes attributs que l'association. L'identifiant de l'association, s'il existe, devient la clé primaire de la relation. Dans le cas où cet identifiant n'existe pas, la clé primaire de l'association devient la concaténation des clés primaires des entités intervenant dans cette association.



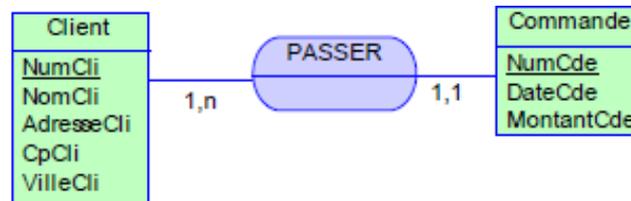
CONCERNER ( #NumCde , #NumProd , QtéCmdée )

2

### ❖ Règle 3:

Lorsqu'une association a une cardinalité associée ayant une cardinalité maximale de 1, ne devient pas une relation. C'est l'entité qui a la cardinalité maximale égale à n qui voit sa clé primaire rajoutée comme attribut dans la seconde entité.

On dit qu'il y a migration de la clé primaire. Cette clé est appelée **clé étrangère**.

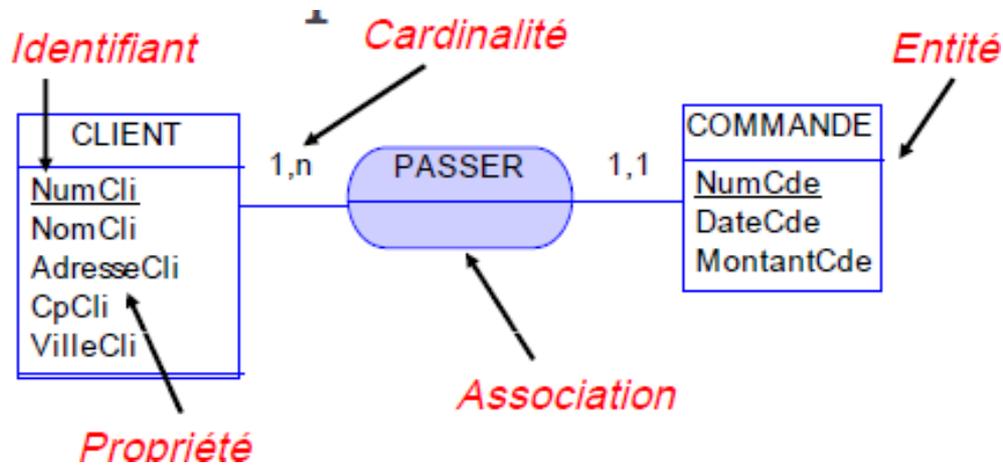


COMMANDE ( NumCde , DateCde, MontantCde, # NumCli )

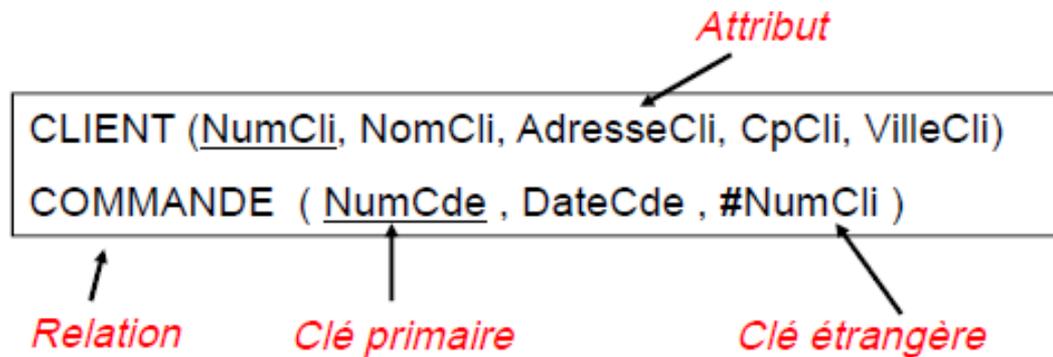


# Récapitulatif du vocabulaire

15



**MCD**



**MLD**

# Récapitulatif du vocabulaire

16

MCD	MLD	MPD
▪Entité	▪Relation	▪Table
▪Association N,N	▪Relation	▪Table
▪Association 1,N	▪Clé étrangère	▪Clé étrangère
▪Identifiants	▪Clé primaire	▪Clé primaire
▪Propriété	▪Attributs	▪Champs

# CHAPITRE4

## *LE LANGAGE SQL*

# PLAN

2

- ▣ Instruction SELECT de base
- ▣ Expressions arithmétiques et valeurs NULL dans l'instruction SELECT
- ▣ Alias de colonne
- ▣ Utilisation de l'opérateur de concaténation, des chaînes de caractères littérales, de l'opérateur alternative quote et du mot clé DISTINCT
- ▣ Commande DESCRIBE

# Capacités des instructions SQL SELECT

3

Projection


Table 1

Selection


Table 1


Table 1

Join


Table 2

# Instruction SELECT de base

4

```
SELECT * | { [DISTINCT] column | expression [alias], ... }  
FROM      table;
```

- **SELECT** identifie les colonnes à afficher.
- **FROM** identifie la table contenant ces colonnes.

# Sélection de toutes les colonnes

5

```
SELECT *  
FROM departments;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	(null)	1700

# Sélection de colonnes spécifiques

6

```
SELECT department_id, location_id  
FROM departments;
```

	DEPARTMENT_ID	LOCATION_ID
1	10	1700
2	20	1800
3	50	1500
4	60	1400
5	80	2500
6	90	1700
7	110	1700
8	190	1700

# Écriture d'instructions SQL

7

- ❑ Les instructions SQL ne sont pas sensibles à la casse.
- ❑ Les instructions SQL peuvent être entrées sur une ou plusieurs lignes.
- ❑ Les mots-clés ne peuvent pas être abrégés ou séparés sur plusieurs lignes.
- ❑ Les clauses sont généralement placées sur des lignes séparées.
- ❑ Les retraits sont utilisés pour améliorer la lisibilité.
- ❑ Dans SQL Developer, les instructions SQL peuvent éventuellement être terminées par un point-virgule (;). Les points-virgules sont requis lorsque vous exécutez plusieurs instructions SQL.
- ❑ Dans SQL \* Plus, vous devez terminer chaque instruction SQL par un point-virgule (;).

# Valeurs par défaut des en-têtes de colonne

8

- SQL Developer:
  - ▣ Alignement de titre par défaut: aligné à gauche
  - ▣ Affichage d'en-tête par défaut: majuscules
- SQL \* Plus:
  - ▣ Les en-têtes de colonne Caractère et Date sont alignés à gauche.
  - ▣ Les en-têtes de colonne nombre sont alignés à droite.
  - ▣ Affichage d'en-tête par défaut: majuscules

# PLAN

9

- ▣ Instruction SELECT de base
- ▣ Expressions arithmétiques et valeurs NULL dans l'instruction SELECT
- ▣ Alias de colonne
- ▣ Utilisation de l'opérateur de concaténation, des chaînes de caractères littérales, de l'opérateur alternative quote et du mot clé DISTINCT
- ▣ Commande DESCRIBE

# Expressions arithmétiques

10

- Créez des expressions avec des données de types nombre et date à l'aide d'opérateurs arithmétiques.

Opérateur	Description
+	Additionner
-	Soustraire
*	Multiplié
/	Divisé

# Utilisation d'opérateurs arithmétiques

11

```
SELECT last_name, salary,  
FROM employees;
```

	LAST_NAME	SALARY	SALARY+300
1	King	24000	24300
2	Kochhar	17000	17300
3	De Haan	17000	17300
4	Hunold	9000	9300
5	Ernst	6000	6300
6	Lorentz	4200	4500
7	Mourgos	5800	6100
8	Rajs	3500	3800
9	Davies	3100	3400
10	Matos	2600	2900

...

# Priorité de l'opérateur

12

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

1

	LAST_NAME	SALARY	12*SALARY+100
1	King	24000	288100
2	Kochhar	17000	204100
3	De Haan	17000	204100

...

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

2

	LAST_NAME	SALARY	12*(SALARY+100)
1	King	24000	289200
2	Kochhar	17000	205200
3	De Haan	17000	205200

...

# Définir une valeur nulle

- Null est une valeur indisponible, non affectée, inconnue ou inapplicable.
- Null n'est pas la même chose que zéro ou un espace vide.

```
SELECT last_name, job_id, salary,  
FROM employees;
```

	LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
1	King	AD_PRES	24000	(null)
2	Kochhar	AD_VP	17000	(null)

...

12	Zlotkey	SA_MAN	10500	0.2
13	Abel	SA_REP	11000	0.3
14	Taylor	SA_REP	8600	0.2

...

19	Higgins	AC_MGR	12000	(null)
20	Gietz	AC_ACCOUNT	8300	(null)

- Les expressions arithmétiques contenant une valeur nulle sont évaluées à null.

```
SELECT last_name,  
FROM employees;
```

## Valeurs nulles dans les expressions arithmétiques

	LAST_NAME	12*SALARY*COMMISSION_PCT
1	King	(null)
2	Kochhar	(null)

...

12	Zlotkey	25200
13	Abel	39600
14	Taylor	20640

...

19	Higgins	(null)
20	Gietz	(null)

# PLAN

15

- ▣ Instruction SELECT de base
- ▣ Expressions arithmétiques et valeurs NULL dans l'instruction SELECT
- ▣ **Alias de colonne**
- ▣ Utilisation de l'opérateur de concaténation, des chaînes de caractères littérales, de l'opérateur alternative quote et du mot clé DISTINCT
- ▣ Commande DESCRIBE

# Définition d'un alias de colonne

- Un alias de colonne:
  - ▣ Renomme une en-tête de colonne
  - ▣ Est utile avec des calculs
  - ▣ Suit immédiatement le nom de la colonne (il peut également y avoir le mot-clé AS facultatif entre le nom de la colonne et l'alias.)
  - ▣ Requiert des guillemets doubles s'il contient des espaces ou des caractères spéciaux ou s'il est sensible à la casse

# Utilisation d'alias de colonne

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

	NAME	COMM
1	King	(null)
2	Kochhar	(null)
3	De Haan	(null)

...

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

	Name	Annual Salary
1	King	288000
2	Kochhar	204000
3	De Haan	204000

...

# PLAN

18

- ▣ Instruction SELECT de base
- ▣ Expressions arithmétiques et valeurs NULL dans l'instruction SELECT
- ▣ Alias de colonne
- ▣ Utilisation de l'opérateur de concaténation, des chaînes de caractères littérales, de l'opérateur alternative quote et du mot clé DISTINCT
- ▣ Commande DESCRIBE

# Opérateur de concaténation

- Un opérateur de concaténation:
  - ▣ Lie les colonnes ou les chaînes de caractères à d'autres colonnes
  - ▣ Est représenté par deux barres verticales (||)
  - ▣ Crée une colonne résultante qui est une expression de caractère

```
SELECT last_name||job_id AS "Employees"  
FROM employees;
```

	Employees
1	AbelSA_REP
2	DaviesST_CLERK
3	De HaanAD_VP
4	ErnstIT_PROG
5	FayMK_REP

...

# Littérales de chaînes de caractères

- Un littéral est un caractère, un nombre ou une date qui est inclus dans l'instruction SELECT.
- Les valeurs littérales date et caractère doivent être placées entre guillemets simples.
- Chaque chaîne de caractères est affichée une fois pour chaque ligne renvoyée.

```
SELECT last_name || ' is a ' || job_id
       AS "Employee Details"
FROM   employees;
```

de caractères

	Employee Details
1	Abel is a SA_REP
2	Davies is a ST_CLERK
3	De Haan is a AD_VP
4	Ernst is a IT_PROG
5	Fay is a MK_REP

...

18	Vargas is a ST_CLERK
19	Whalen is a AD_ASST
20	Zlotkey is a SA_MAN

# Opérateur quote (q)

- ▣ Indiquez votre propre délimiteur de guillemet.
- ▣ Sélectionnez un délimiteur.
- ▣ Augmenter la lisibilité et la convivialité

```
SELECT department_name || ' Department' ||  
       q['s Manager Id: ]'  
       || manager_id  
       AS "Department and Manager"  
FROM departments;
```

	Department and Manager
1	Administration Department's Manager Id:200
2	Marketing Department's Manager Id:201
3	Shipping Department's Manager Id:124
4	IT Department's Manager Id:103
5	Sales Department's Manager Id:149
6	Executive Department's Manager Id:100
7	Accounting Department's Manager Id:205
8	Contracting Department's Manager Id:

# Lignes en double

- L'affichage par défaut des requêtes est toutes les lignes, y compris les lignes en double.

```
SELECT department_id  
FROM employees;
```

1

R	DEPARTMENT_ID
1	90
2	90
3	90
4	60
5	60

...

```
SELECT DISTINCT department_id  
FROM employees;
```

2

R	DEPARTMENT_ID
1	(null)
2	90
3	20
4	110

...

# PLAN

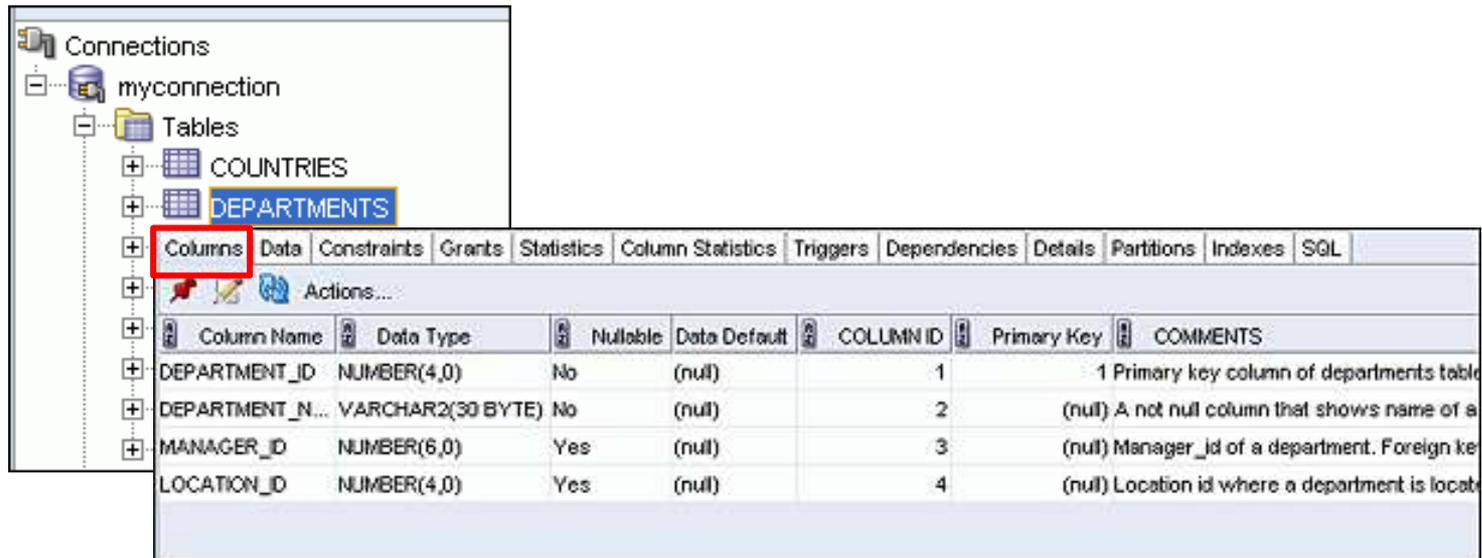
24

- ▣ Instruction SELECT de base
- ▣ Expressions arithmétiques et valeurs NULL dans l'instruction SELECT
- ▣ Alias de colonne
- ▣ Utilisation de l'opérateur de concaténation, des chaînes de caractères littérales, de l'opérateur alternative quote et du mot clé DISTINCT
- ▣ **Commande DESCRIBE**

# Affichage de la structure de la table

- Utilisez la commande DESCRIBE pour afficher la structure d'une table.
- Vous pouvez également sélectionner la table dans l'arborescence Connexions et utiliser l'onglet Colonnes pour afficher la structure de la table.

```
DESC[RIBE] tablename
```



The screenshot shows the Oracle Enterprise Manager interface. In the left-hand tree view, the 'DEPARTMENTS' table is selected under the 'Tables' folder. The 'Columns' tab is active, displaying the table's structure. The table has four columns: DEPARTMENT\_ID (NUMBER(4,0)), DEPARTMENT\_NAME (VARCHAR2(30 BYTE)), MANAGER\_ID (NUMBER(6,0)), and LOCATION\_ID (NUMBER(4,0)).

Column Name	Data Type	Nullable	Data Default	COLUMN ID	Primary Key	COMMENTS
DEPARTMENT_ID	NUMBER(4,0)	No	(null)	1	1	Primary key column of departments table
DEPARTMENT_N...	VARCHAR2(30 BYTE)	No	(null)	2		(null) A not null column that shows name of a
MANAGER_ID	NUMBER(6,0)	Yes	(null)	3		(null) Manager_id of a department. Foreign ke
LOCATION_ID	NUMBER(4,0)	Yes	(null)	4		(null) Location id where a department is locat

# Utilisation de la commande DESCRIBE

```
DESCRIBE employees
```

```
DESCRIBE employees
Name                               Null    Type
-----
EMPLOYEE_ID                       NOT NULL NUMBER(6)
FIRST_NAME                          VARCHAR2(20)
LAST_NAME                           NOT NULL VARCHAR2(25)
EMAIL                               NOT NULL VARCHAR2(25)
PHONE_NUMBER                        VARCHAR2(20)
HIRE_DATE                           NOT NULL DATE
JOB_ID                              NOT NULL VARCHAR2(10)
SALARY                              NUMBER(8,2)
COMMISSION_PCT                      NUMBER(2,2)
MANAGER_ID                          NUMBER(6)
DEPARTMENT_ID                       NUMBER(4)

11 rows selected
```

# TP2

- Ce TP couvre les sujets suivants:
  - ▣ Sélection de toutes les données de différentes tables
  - ▣ Décrire la structure des tables
  - ▣ Effectuer des calculs arithmétiques et spécifier des noms de colonnes

# CHAPITRE 4

## 3. LE LANGAGE SQL: RESTREINDRE ET TRIER SOUS SQL