

# IFT3030

## Base de données

### Chapitre 7

## Conception de bases de données

☒ Introduction  
☒ Architecture  
☒ Modèles de données  
☒ Modèle relationnel  
☒ Algèbre relationnelle  
☒ SQL  
☒ **Conception**  
☒ Fonctions avancées  
☒ Concepts avancés  
☒ Modèle des objets  
☒ BD à objets

## Plan du cours

- Introduction
- Architecture
- Modèles de données
- Modèle relationnel
- Algèbre relationnelle
- SQL
- **Conception**
- Fonctions avancées
- Concepts avancés
- Modèle des objets
- BD à objets

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Introduction

- Problème de la conception
  - Associer une structure logique aux données. En d'autres termes, trouver les relations et leurs attributs
  
- Techniques
  - Dépendances fonctionnelles
  - Les formes normales
  - Le modèle Entité-Association

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Dépendances fonctionnelles (DF)

- Définition
  - Un groupe d'attributs Y dépend fonctionnellement d'un groupe d'attributs X, si et seulement si la connaissance d'une valeur de X permet de connaître sans ambiguïté la valeur correspondante de Y
  - On dit également que X détermine Y
  
- Exemple
  - Matricule → Nom
  - (NV, NB) → Quantité
  - NAS → (Nom, Prenom)
  - Nom → ~~Code\_permanent~~

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Dépendances fonctionnelles (DF)

### ■ Règles d'inférence d'Armstrong

- Réflexivité
  - $Y \subseteq X \Rightarrow X \rightarrow Y$
  - $(NV, NB) \rightarrow NB$
- Augmentation
  - $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
  - $Type \rightarrow Puissance \Rightarrow$   
 $(Type, Marque) \rightarrow (Puissance, Marque)$
- Transitivité
  - $X \rightarrow Y$  et  $Y \rightarrow Z \Rightarrow X \rightarrow Z$
  - $Mat \rightarrow Type$  et  $Type \rightarrow Puissance \Rightarrow Mat \rightarrow Puissance$

5

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Dépendances fonctionnelles (DF)

### ■ Règles additionnelles

- Union
  - $X \rightarrow Y$  et  $X \rightarrow Z \Rightarrow X \rightarrow YZ$
  - $NAS \rightarrow Nom$  et  $NAS \rightarrow Prenom \Rightarrow NAS \rightarrow (Nom, Prenom)$
- Pseudo-transitivité
  - $X \rightarrow Y$  et  $WY \rightarrow Z \Rightarrow WX \rightarrow Z$
  - $Mat \rightarrow Type$  et  $(Marque, Type) \rightarrow Puissance \Rightarrow$   
 $(Marque, Mat) \rightarrow Puissance$
- Décomposition
  - $X \rightarrow Y$  et  $Z \subseteq Y \Rightarrow X \rightarrow Z$
  - $NAS \rightarrow (Nom, Prenom) \Rightarrow NAS \rightarrow Nom$

6

## Dépendances fonctionnelles (DF)

### ■ Dépendance fonctionnelle élémentaire

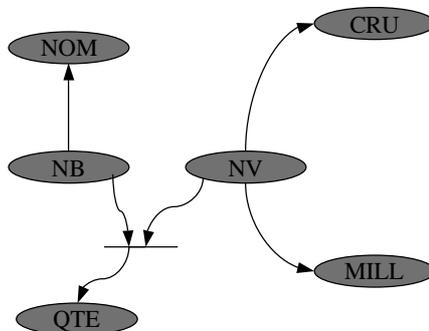
- C'est une dépendance fonctionnelle de la forme  $X \rightarrow A$ , où A est un attribut unique tel que  $A \notin X$  et où il n'existe pas  $X' \subset X$  tel que  $X' \rightarrow A$
- Seule propriété qui s'applique aux dépendances fonctionnelles élémentaires est la transitivité
- Pourquoi pas les autres ?

7

## Dépendances fonctionnelles (DF)

### ■ Graphe de dépendances fonctionnelles élémentaires

- Exemple
  - $NV \rightarrow CRU$
  - $NV \rightarrow MILL$
  - $NB \rightarrow NOM$
  - $(NV, NB) \rightarrow QTE$



8

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Dépendances fonctionnelles (DF)

### Fermeture transitive

- Soit  $D$ , un ensemble des dépendances fonctionnelles élémentaires, la fermeture transitive de  $D$  est l'ensemble  $D$  enrichi de toutes les dépendances fonctionnelles élémentaires obtenues par transitivité
- Exemple
  - $D = \{ NV \rightarrow Type ; Type \rightarrow Marque ; Type \rightarrow Puiss ; NV \rightarrow Couleur \}$ $D^+ = D \cup \{ NV \rightarrow Marque ; NV \rightarrow Puiss \}$
- Conséquence : Deux ensembles de DFs élémentaires sont équivalents s'ils ont la même fermeture transitive

9

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Dépendances fonctionnelles (DF)

### ■ Couverture minimale

- Soit un ensemble de dépendances fonctionnelles élémentaires  $D$  pour un ensemble d'attributs  $A$ ,  $D$  est une couverture minimale si
  - Toute DF  $f$  n'est pas redondante ( $D - f$  n'est pas équivalent à  $D$ )
  - Toute DF élémentaire de  $A$  est dans la fermeture transitive  $D^+$
- Tout ensemble de DFs élémentaires a une couverture minimale
- Cette couverture peut ne pas être unique

10

## Dépendances fonctionnelles (DF)

### ■ Couverture minimale

#### – Exemple

- $D = \{ NV \rightarrow Type ; Type \rightarrow Marque ; NV \rightarrow Couleur ; NV \rightarrow Marque ; NV \rightarrow Puiss ; (Type, Marque) \rightarrow Rabais ; NV \rightarrow Rabais \}$

#### Couverture minimale :

- $CV(D) = \{ NV \rightarrow Type ; Type \rightarrow Marque ; NV \rightarrow Couleur ; NV \rightarrow Marque ; NV \rightarrow Puiss ; (Type, Marque) \rightarrow Rabais ; NV \rightarrow Rabais \}$

11

## Dépendances fonctionnelles (DF)

### ■ DF et notion de clé

#### – Soit $R(A_1, A_2, \dots, A_n)$ une relation, $X \subseteq \{A_1, A_2, \dots, A_n\}$ est une clé de R si

- $X \rightarrow A_1 A_2 \dots A_n$
- Il n'existe pas  $Y \subseteq X$ , tel que  $Y \rightarrow A_1 A_2 \dots A_n$

#### – Conséquence

On peut dériver le schéma d'une relation en partant de la liste des dépendances fonctionnelles sur un ensemble d'attributs

12

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Les trois premières formes normales

### ■ Introduction

- Les trois premières formes normales ont pour objectif de décomposer des relations sans perte d'information. Cette décomposition s'appuie sur la notion de dépendance fonctionnelle

13

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Les trois premières formes normales

### ■ Première forme normale

- Une relation est en première forme normale (1NF) si tous ces attributs ont des valeurs atomiques
- Règles de décomposition
  - $\text{Personne}(\underline{\text{nom}}, \text{prenoms})$   
 $\Rightarrow \text{Personne1}(\text{nom}, \text{prenom1})$   
 $\Rightarrow \text{Personne2}(\text{nom}, \text{prenom2})$
  - Ou  
 $\Rightarrow \text{Personne}(\text{nom}, \text{prenom1}, \text{prenom2})$

14

## Les trois premières formes normales

### ■ Deuxième forme normale

- Une relation est en deuxième forme normale (2NF) si
  - Elle est en première forme normale
  - Tout attribut n'appartenant pas à la clé, ne dépend pas d'une partie de la clé
- Règle de décomposition
  - $R(\underline{K1}, \underline{K2}, X, Y)$
  - DFs :  $(K1, K2) \rightarrow X$  et  $K2 \rightarrow Y$
  - $\Rightarrow R1(\underline{K1}, \underline{K2}, X)$  et  $R2(\underline{K2}, Y)$

15

## Les trois premières formes normales

### ■ Deuxième forme normale

- Exemple
  - Fournisseur(nom, article, adresse, prix)
  - DFs :  $(\text{nom}, \text{article}) \rightarrow \text{prix}$  et  $\text{nom} \rightarrow \text{adresse}$
- $\Rightarrow$ 
  - Fournisseur(nom, adresse)
  - Produit(nom, article, prix)

16

## Les trois premières formes normales

### ■ Troisième forme normale

- Une relation est en troisième forme normale (3NF) si
  - Elle est en deuxième forme normale
  - Tout attribut n'appartenant pas à la clé, ne dépend pas d'un attribut non clé
- Règle de décomposition
  - $R(\underline{K1}, X, Y, Z)$
  - DFs :  $K1 \rightarrow X$ ,  $K1 \rightarrow Y$ ,  $K1 \rightarrow Z$ , et  $X \rightarrow Z$
  - ⇒  $R1(\underline{K1}, X, Y)$  et  $R2(\underline{X}, Z)$

17

## Les trois premières formes normales

### ■ Troisième forme normale

- Exemple
  - Voiture(NV, marque, type, puiss, couleur)
  - DFs :  $NV \rightarrow \text{marque}$ ,  $NV \rightarrow \text{type}$ ,  
 $NV \rightarrow \text{puiss}$ ,  $NV \rightarrow \text{couleur}$ ,  $\text{type} \rightarrow \text{marque}$   
et  $\text{type} \rightarrow \text{puiss}$
- ⇒
  - Voiture(NV, type, couleur)
  - Modele(type, marque, puiss)

18

## Les trois premières formes normales

### ■ Troisième forme normale

#### – Propriété

- Si une relation  $R$  est décomposé en un ensemble de relations  $R_1, R_2, \dots, R_n$ , La fermeture transitive des DFs de  $R$  est a même que celle de l'union des DFs de  $R_1, R_2, \dots, R_n$ .
- On dit que la décomposition préserve les DFs

## Algorithme de décomposition 3FN

- Considérer tout les attributs comme faisant partie d'une seule relation (relation universelle)
- Appliquer les transformations de normalisation pour obtenir des relations en 3FN

## Algorithme de synthèse en 3FN

■ Soit  $A$  un ensemble d'attributs et  $D$  l'ensemble des dépendances fonctionnelles définies sur  $A$

### Synthese

Trouver une couverture minimale  $F$  de  $D$

Conserver dans  $F$  les seules dépendances élémentaires

$AR := A$

**Tant que**  $F$  non vide **Faire**

GenererRelation()

**Fin Faire**

Creer une relation avec les attributs restants dans  $AR$

**Fin Synthese**

## Algorithme de synthèse en 3FN

### GenererRelation

. Trouver dans  $F$  le plus grand ensemble d'attributs  $X$  qui en détermine d'autres

.  $AC := \{A_i \mid X \rightarrow A_i\}$

. Generer la relation  $RC$  ayant pour attributs  $X \cup AC$  et pour clé  $X$

.  $F := F - AC$

.  $AR := AR - AC$

**Fin Synthese**

■ **Exemple :**

$A = \{a, b, c, d, e, f, g, h, i, j, k, l\}$

$D = \{e \textcircled{R} k, (a,d,e) \textcircled{R} i, a \textcircled{R} b, (b,c) \textcircled{R} h, k \textcircled{R} i, a \textcircled{R} f, (b,c) \textcircled{R} f, e \textcircled{R} l, (a,d,e) \textcircled{R} j, (b,c) \textcircled{R} g, c \textcircled{R} h, a \textcircled{R} c\}$

## Forme normale de Boyce-Codd

- Une relation est en forme normale de Boyce-Codd (BCNF) si et seulement si les seules DFs élémentaires sont celles dans lesquelles un clé entière détermine un attribut
- Règle de décomposition
  - $R(\underline{K1}, \underline{K2}, X, Y)$
  - DFs :  $K1, K2 \rightarrow X, K1, K2 \rightarrow Y$  et  $Y \rightarrow K1$
  - ⇒  $R1(\underline{K1}, \underline{K2}, X)$  et  $R2(\underline{Y}, K1)$

23

## Forme normale de Boyce-Codd

- Exemple
  - Localisation(Cru, Pays, Region, Qualite)
  - DFs :
    - Cru, Pays → Region
    - Cru, Pays → Qualite
    - Region → Pays
- ⇒
  - Crus(Cru, Pays, Qualite)
  - Region(Region, Pays)
- Remarque : BCNF ne préserve pas les DFs. Exemple "Cru, Pays → Region" est perdue

24

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Modélisation sémantique

- Permet de modéliser les données
- En utilisant différents formalismes
  - exemples : E/A, UML, ...
- Une conception de bases de données qui commence par la modélisation sémantique est dite conception descendante (conception allant du plus haut niveau vers le plus bas niveau)

25

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Modèle Entité/Association

- Généralités
  - Appelé également Entité/Relation
  - Proposé par Chen en 1976
  - Englobe à l'origine un nombre restreint de concept (entité, association, propriété)
  - Diverses extensions ont été proposé par la suite (contraintes, généralisation agrégation, ...)

26

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Modèle Entité/Association, Concepts

### ■ Entité

- "Une chose qui peut être identifiée distinctement"
- Par abus de langage, on confond *entité* et *type d'entité*
- Exemples :
  - Concrètes : voiture, employé,
  - Abstraites : projet, cours
- Mais, mariage ???

27

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

## Modèle Entité/Association, Concepts

### ■ Entité (suite)

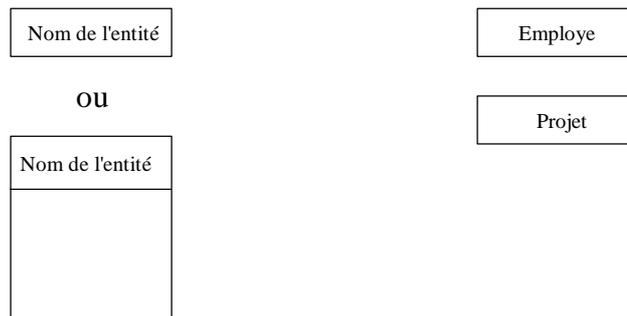
- On distingue souvent
  - Entité régulière
  - Entité faible dont l'existence dépend d'une autre entité
  - Exemple
    - Employé .... Entité régulière
    - Personne\_a\_charge .... Entité faible dont l'existence dépend de celle de l'entité Employé

28

# Modèle Entité/Association, Concepts

## ■ Entité (suite)

– Représentation graphique



29

# Modèle Entité/Association, Concepts

## ■ Propriété

- Les entités possèdent des propriétés qui les décrivent
- Appelées également *Attributs*
- Exemples :
  - Pour Employé, matricule, nom, adresse,
  - mais pas le numéro de département

30

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Modèle Entité/Association, Concepts

## ■ Propriété

- Elle peut être
  - Simple ou composée
    - âge (simple)
    - adresse (composée)
  - Un identificateur (ou clé)
    - code\_permanent
  - mono ou multi-valuée
    - nom (mono-valuée)
    - prénom (multi-valuée)

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Modèle Entité/Association, Concepts

## ■ Propriété

- Elle peut être
  - Manquante
    - valeur inconnue
    - non applicable (date\_mariage pour un célibataire)
  - De base ou dérivée
    - nom (base)
    - ancienneté (dérivée de la date d'embauche et de la date d'aujourd'hui)
    - Quantité totale (dérivé de la somme des quantités)

# Modèle Entité/Association, Concepts

## ■ Propriété

– Représentation graphique

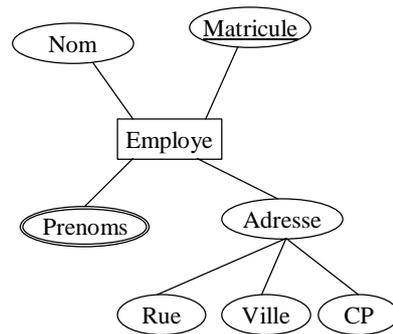
propriété

Parfois, on utilise cette notation pour les propriétés multi-valuées

propriété

Parfois, on utilise cette notation pour les identificateurs

ident



# Modèle Entité/Association, Concepts

## ■ Association

- Est une relation entre entités
- Même abus de langage entre association et type d'associations
- Exemple
  - *Participe\_dans* entre Employé et projet
  - *Récolte* entre Producteur et Vin
- Comme une entité, elle peut posséder des propriétés

## Modèle Entité/Association, Concepts

### ■ Association

- Elle est dite binaire si elle relie uniquement deux entités, ternaire si elle en relie trois, etc.
- Chaque participant (entité) possède un rôle dans l'association
- Elle peut être réflexive
  - Exemple : *est\_mariée\_à* entre personne et personne

35

## Modèle Entité/Association, Concepts

### ■ Association

#### – Cardinalités

- Indiquent le nombre maximal et minimal d'occurrences d'une association pour une occurrence d'un participant.
- Par exemple pour l'association *contrôle* entre département et projet
  - un département contrôle 0 ou plusieurs projet
  - un projet est contrôlé par un et un seul département
- Les différentes possibilités sont
  - 0,1 ; 1,1 ; 0,N ; 1,N

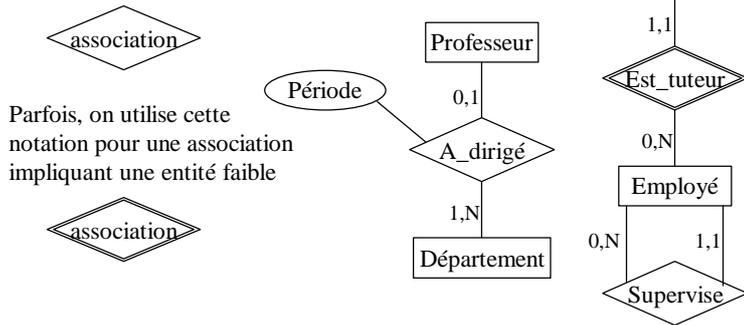
36

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Modèle Entité/Association, Concepts

## ■ Association

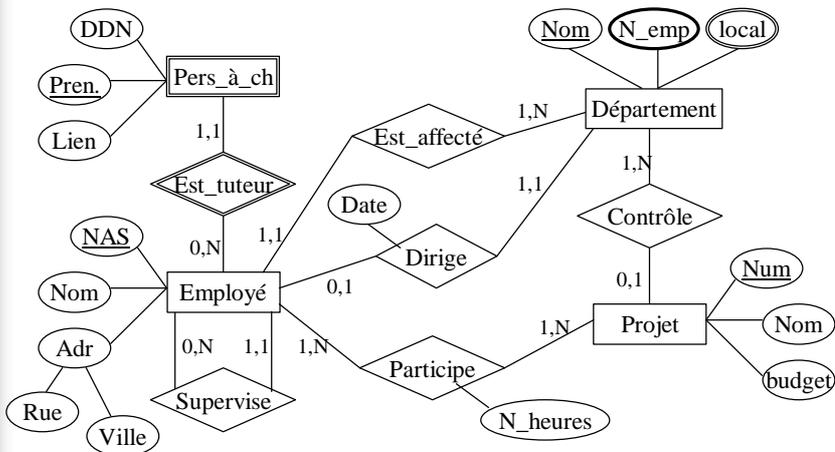
– Représentation graphique



- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Modèle Entité/Association, Concepts

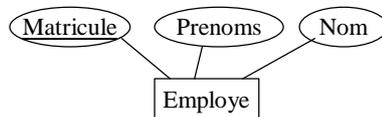
## ■ Un exemple complet



## Conception de la base de données E/A vers relationnel

### ■ Entité

- Chaque type d'entité devient une relation.  
Son identificateur devient la clé et les propriétés deviennent des attributs. Les propriétés multi-valuées sont transformées conformément à la 1NF



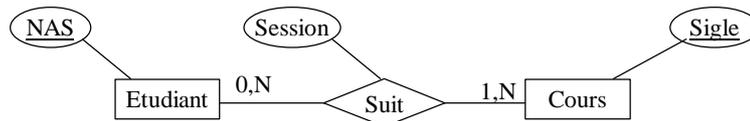
Employe(Matricule, Nom, Prenom)

## Conception de la base de données E/A vers relationnel

### ■ Association plusieurs à plusieurs

{0,N | 1,N} vers {0,N | 1,N}

- Chaque type d'associations devient une relation
- Les identificateurs des participants deviennent la clé de la relation



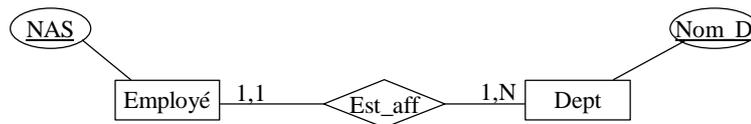
Inscription(NAS, Sigle, Session)

NAS et Sigle sont des clés étrangères également

## Conception de la base de données E/A vers relationnel

### ■ Association un à plusieurs $\{0,1 \mid 1,1\}$ vers $\{0,N \mid 1,N\}$

– Chaque type d'associations se traduit par un ajout d'une clé étrangère



Employé(NAS, ..., Nom\_D)

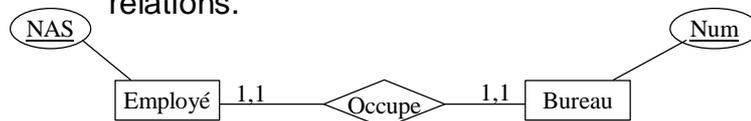
Cas particulier d'une association impliquant une entité faible (contrainte CASCADE)

## Conception de la base de données E/A vers relationnel

### ■ Association un à un $\{0,1 \mid 1,1\}$ vers $\{0,1 \mid 1,1\}$

– Chaque type d'associations se traduit par

- Une fusion des deux relations correspondantes
- Un ajout de clé étrangère dans l'une ou les deux relations.



Option 1 :  
Employé(NAS, ..., Num)

Option 2 :  
Employé(NAS, ..., Num)  
Bureau(Num, ..., NAS)

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ **Conception**
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

# Conception de la base de données

## E/A vers relationnel

■ Convertir le modèle suivant en Relationnel

