

# MODÉLISATION ÉLÉMENTAIRE DES DONNÉES ET DES TRAITEMENT AVEC MERISE – L1 (2016-17)

merise

# Outline

## 1 Introduction

## 2 La méthode Merise

## 3 Le modèle conceptuel de données sous Merise : le modèle Entité-Association

- Les dépendances fonctionnelles (*DF*)

## 4 Le modèle relationnel de données : un *MLD* particulier

# INTRODUCTION

## Système d'information d'une entreprise

SI = ensemble des moyens humains, matériels et immatériels permettant de collecter, mémoriser, traiter, distribuer et consulter les informations dont l'entreprise a besoin pour ses différentes activités. Ressource essentielle de l'entreprise.

# INTRODUCTION

## Système d'information d'une entreprise

SI = ensemble des moyens humains, matériels et immatériels permettant de collecter, mémoriser, traiter, distribuer et consulter les informations dont l'entreprise a besoin pour ses différentes activités. Ressource essentielle de l'entreprise.

### Remarque :

A ne pas confondre avec le *système informatique* de l'entreprise : ensemble des moyens informatiques mis en oeuvre pour assurer le traitement automatisé du SI (matériels et logiciels : ordinateurs, réseaux, *SE*, utilitaires, bureautique, *SGBD*, *AGL*...).

# INTRODUCTION

Toute méthode d'analyse et de conception d'un *SI* intègre à des degrés divers les éléments suivants :

- La démarche : processus opératoire permettant de faire le travail de modélisation et de description du *SI*,

# INTRODUCTION

Toute méthode d'analyse et de conception d'un *SI* intègre à des degrés divers les éléments suivants :

- La démarche : processus opératoire permettant de faire le travail de modélisation et de description du *SI*,
- La construction du *SI* via les concepts normalisés sous une forme schématique permettant une représentation simple de la réalité et facilitant le raisonnement

# INTRODUCTION

Toute méthode d'analyse et de conception d'un *SI* intègre à des degrés divers les éléments suivants :

- La démarche : processus opératoire permettant de faire le travail de modélisation et de description du *SI*,
- La construction du *SI* via les concepts normalisés sous une forme schématique permettant une représentation simple de la réalité et facilitant le raisonnement
- Des logiciels d'*AGL* supportant et épaulant la méthode d'analyse (plus ou moins selon les méthodes).

# Outline

## 1 Introduction

## 2 La méthode Merise

### 3 Le modèle conceptuel de données sous Merise : le modèle Entité-Association

- Les dépendances fonctionnelles (*DF*)

### 4 Le modèle relationnel de données : un *MLD* particulier

# La méthode de modélisation Merise

Acronyme pour :

Méthode d'Étude et de Réalisation Informatique des Systèmes  
des Entreprises

# La méthode de modélisation Merise

Acronyme pour :

Méthode d'Étude et de Réalisation Informatique des Systèmes des Entreprises

Historique :

Crée en 1977 par un ensemble de sociétés de services, sous la direction du Centre Technique Informatique du Ministère de l'Industrie. Répond à la volonté des pouvoirs publics de doter les administrations et entreprises publiques d'une méthodologie rigoureuse. Encore utilisée par nombre d'entreprises en France. Malgré son évolution et ses nombreuses variantes et extensions, elle présente une base de principes communs.

# LA MÉTHODE DE MODÉLISATION MERISE

L'approche proposée sépare :

- ① l'étude des données de celle des traitements

# LA MÉTHODE DE MODÉLISATION MERISE

L'approche proposée sépare :

- ① l'étude des données de celle des traitements
- ② pour les données comme pour les traitements, l'étude porte progressivement sur :
  - le niveau conceptuel (le plus abstrait)

# LA MÉTHODE DE MODÉLISATION MERISE

L'approche proposée sépare :

- ① l'étude des données de celle des traitements
- ② pour les données comme pour les traitements, l'étude porte progressivement sur :
  - le niveau conceptuel (le plus abstrait)
  - le niveau organisationnel (intermédiaire)

# LA MÉTHODE DE MODÉLISATION MERISE

L'approche proposée sépare :

- ① l'étude des données de celle des traitements
- ② pour les données comme pour les traitements, l'étude porte progressivement sur :
  - le niveau conceptuel (le plus abstrait)
  - le niveau organisationnel (intermédiaire)
  - le niveau opérationnel (le plus concret)

# LA MÉTHODE DE MODÉLISATION MERISE

L'approche proposée sépare :

- ① l'étude des données de celle des traitements
- ② pour les données comme pour les traitements, l'étude porte progressivement sur :
  - le niveau conceptuel (le plus abstrait)
  - le niveau organisationnel (intermédiaire)
  - le niveau opérationnel (le plus concret)

On a donc trois niveaux de représentation d'un *SI* pour les données comme pour les traitements. Chaque niveau est indépendant du niveau inférieur.

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

## Le niveau conceptuel :

Étape préalable à la construction de la base. Représente l'ensemble des informations et des traitements décrivant le fonctionnement de l'entreprise, compte tenu des choix et objectifs de gestion retenus. Niveau indépendant des choix d'organisation et de techniques. Répond à la question "Quoi ?".

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

## Le niveau conceptuel :

Étape préalable à la construction de la base. Représente l'ensemble des informations et des traitements décrivant le fonctionnement de l'entreprise, compte tenu des choix et objectifs de gestion retenus. Niveau indépendant des choix d'organisation et de techniques. Répond à la question "Quoi ?".

- ❶ Données : en termes d'*entités*, d'*associations* entre entités et de *cardinalités* (pour la représentation des contraintes).
- ❷ Traitements : en termes d'*événements*, d'*opérations*, de *synchronisation* et de *conditions d'émission des résultats*.

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

Le niveau logique (ou organisationnel) :

Exprime la réalité telle qu'elle est perçue par les acteurs de l'entreprise. Aucune différence à ce niveau entre hommes et machines. Répond aux questions "Qui?", "Où?", "Quand?".

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

Le niveau logique (ou organisationnel) :

Exprime la réalité telle qu'elle est perçue par les acteurs de l'entreprise. Aucune différence à ce niveau entre hommes et machines. Répond aux questions "Qui?", "Où?", "Quand?".

- ① Données : structures de données du niveau conceptuel non portables en l'état sur une machine. Suscite la création de *tables*, sans enrichissement de nouvelles informations.  
Simple transformation de la description des entités et associations d'une structure en une autre.
- ② Traitements : répartition hommes/machines, mode de traitement, répartition géographique et temporelle des événements et des opérations.

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

## Le niveau physique :

Représentation des moyens mis en oeuvre pour gérer les données ou activer les traitements. Apporte et décrit les solutions techniques choisies, répondant à la question "Comment ?".

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

## Le niveau physique :

Représentation des moyens mis en oeuvre pour gérer les données ou activer les traitements. Apporte et décrit les solutions techniques choisies, répondant à la question "Comment ?".

- ① Données : se traduit par l'utilisation d'un *SGBD* ainsi que par des choix concernant les méthodes de stockage et d'accès.
- ② Traitements : décrit l'architecture des programmes qui vont activer les différentes tâches de l'ordinateur. La phase de programmation viendra dans la phase de *codage*.

# LES TROIS NIVEAUX DE PRÉSENTATION DU SYSTÈME

## Le niveau physique :

Représentation des moyens mis en oeuvre pour gérer les données ou activer les traitements. Apporte et décrit les solutions techniques choisies, répondant à la question "Comment ?".

- ① Données : se traduit par l'utilisation d'un *SGBD* ainsi que par des choix concernant les méthodes de stockage et d'accès.
- ② Traitements : décrit l'architecture des programmes qui vont activer les différentes tâches de l'ordinateur. La phase de programmation viendra dans la phase de *codage*.

**On n'étudiera pas ce niveau dans le cours.**

# TABLEAU RÉCAPITULATIF DES NIVEAUX D'ÉTUDES

Tableau synthétique :

Niveau	Type de choix	Questions posées	Données	Traitements
Conceptuel	Choix de gestion	Quoi ?	<i>MCD</i>	<i>MCT</i>
Organisationnel	Choix d'organisation	Qui ? Quand ? Où ?	<i>MLD</i>	<i>MOT</i>
Physique	Choix techniques	Comment ?	<i>MPD</i>	<i>MPT</i>

# Outline

- 1 Introduction
- 2 La méthode Merise
- 3 Le modèle conceptuel de données sous Merise : le modèle Entité-Association
  - Les dépendances fonctionnelles (*DF*)
- 4 Le modèle relationnel de données : un *MLD* particulier

# LE MODÈLE ENTITÉ-ASSOCIATION

Un MCD particulier :

Modèle décrivant l'ensemble des données structurées relevant du SI. Se préoccupe de la signification des données, donc de sémantique. Né dans les années 70 en France mais est attribué à l'américain P. Chen qui a participé à sa large diffusion internationale. On développe ici les bases communes à toutes les variantes de ce modèle.

# LE MODÈLE ENTITÉ-ASSOCIATION

Un MCD particulier :

Modèle décrivant l'ensemble des données structurées relevant du SI. Se préoccupe de la signification des données, donc de sémantique. Né dans les années 70 en France mais est attribué à l'américain P. Chen qui a participé à sa large diffusion internationale. On développe ici les bases communes à toutes les variantes de ce modèle.

Concepts propres à ce niveau

On va parler d'entités, de propriétés et d'identifiant. Aussi de type d'association, de cardinalité, de dépendances fonctionnelles, etc.

# CONCEPTS MANIPULÉS AU NIVEAU DU MCD

## Entité :

Représentation de tout objet type, matériel ou non, faisant partie du SI, présentant un intérêt pour le champs d'étude et doté d'une existence propre (i.e. existe indépendamment de toute autre entité du modèle). Décrrite par un ensemble non vide de *propriétés* et possède un *identifiant*. Exemples et formalisme (oeuvre, compositeur,...).

# CONCEPTS MANIPULÉS AU NIVEAU DU MCD

## Entité :

Représentation de tout objet type, matériel ou non, faisant partie du SI, présentant un intérêt pour le champs d'étude et doté d'une existence propre (i.e. existe indépendamment de toute autre entité du modèle). Décrrite par un ensemble non vide de *propriétés* et possède un *identifiant*. Exemples et formalisme (oeuvre, compositeur,...).

## Propriété :

Représentation d'une caractéristique commune à tous les éléments composant une entité (ou une association). Exemples et formalisme (titre-œuvre, catégorie-œuvre, date-œuvre, nom-comp., prénom-comp., nationalité, date-naissance,...).

# CONCEPTS MANIPULÉS AU NIVEAU DU MCD

Identifiant d'une entité  $E$  :

Propriété (ou ensemble de propriétés) dont la connaissance de la valeur permet de distinguer de manière unique une occurrence de  $E$ .

Potentiellement, une entité possède plusieurs identifiants possibles. En pratique, on en retient toujours une seule pour le modèle.

Exemples et formalisme (pour une entité "voiture", le n° d'immatriculation, pour une personne majeure son n° de sécu.,...).

# CONCEPTS MANIPULÉS AU NIVEAU DU MCD

## Association :

Représentation formalisée de l'existence d'une relation entre au moins deux entités du SI. La *collection* de l'association est la liste des entités sur lesquelles l'association est définie et sa *dimension* est le nombre d'entités de sa collection. Elle a éventuellement des propriétés (on la dit alors *porteuse de données*). Son existence est conditionnée par celle des entités de sa collection (sinon elle serait elle-même une entité).

## Exemples d'associations et formalisme (au tableau) :

- ① binaire non porteuse de données :
- ② binaire porteuse de données :
- ③ association  $n$ -aire ( $n > 2$ ) :
- ④ association réflexive :



# CONCEPTS MANIPULÉS AU NIVEAU DU MCD

Cardinalité d'un couple E/A :

Indique pour chaque couple  $(E, A)$  d'entité/association les nombres minimal et maximal d'occurrences de A pouvant exister pour une occurrence donnée de E. Exemples au tableau.

# CONCEPTS MANIPULÉS AU NIVEAU DU MCD

Cardinalité d'un couple E/A :

Indique pour chaque couple  $(E, A)$  d'entité/association les nombres minimal et maximal d'occurrences de A pouvant exister pour une occurrence donnée de E. Exemples au tableau.

Type d'une association binaire entre deux entités  $E$  et  $E'$  :

- ① Association de type **1 à 1** : à une occurrence d'une entité  $E$  correspond *au plus* une occurrence de l'autre entité  $E'$ .
- ② Association de type **1 à plusieurs** : à une occurrence de  $E$  correspond *au plus* une occurrence de  $E'$  et à une occurrence de  $E'$  correspond *au moins* une occurrence de  $E'$ .
- ③ Association de type **plusieurs à plusieurs** : à une occurrence de  $E$  correspond *moins* une occurrence de  $E'$ .

# LES RÈGLES DE GESTION

## Définition :

Elles précisent l'ensemble des contraintes qui doivent être respectées par le modèle. Elles expriment les **contraintes d'intégrité** du modèle et représentent les lois de l'univers réel modélisé dans le SI.

On distingue les contraintes *statiques* et *dynamiques*.

# LES RÈGLES DE GESTION

## Définition :

Elles précisent l'ensemble des contraintes qui doivent être respectées par le modèle. Elles expriment les **contraintes d'intégrité** du modèle et représentent les lois de l'univers réel modélisé dans le SI.

On distingue les contraintes *statiques* et *dynamiques*.

## Les contraintes dynamiques :

Elles expriment les règles d'évolution des éléments du SI et portent sur le passage d'un état donné à un autre (exemple : le salaire d'un salarié de l'entreprise ne peut diminuer, etc...).

# LES RÈGLES DE GESTION

Les contraintes statiques peuvent :

- porter sur le domaine de définition d'une propriété (exemple : l'âge d'une personne),
- traduire une contrainte *entre* propriétés d'une même entité (exemple : pour un salarié d'une entreprise, date-embauche  $\leq$  date-affectation), ou d'entités différentes,
- porter sur les cardinalités,
- traduire des *dépendances fonctionnelles* (on y vient !).

Les dépendances fonctionnelles (*DF*)

# LES DÉPENDANCES FONCTIONNELLES (*DF*)

Notion introduite par Armstrong en 1974 (en mathématiques pures) et utilisée en BD pour formaliser la sémantique.

DF entre propriétés d'une entité  $E$  :

Soit  $E$  une entité et  $X, Y$  deux sous-ensembles de propriétés de  $E$ .  $E$  vérifie la **dépendance fonctionnelle**  $X \rightarrow Y$  ( $X$  détermine fonctionnellement  $Y$ ) si, lorsque deux occurrences de  $E$  ont les mêmes valeurs sur  $X$ , elles ont aussi les mêmes valeurs sur  $Y$ .

Les dépendances fonctionnelles (*DF*)

# LES DÉPENDANCES FONCTIONNELLES (*DF*)

Notion introduite par Armstrong en 1974 (en mathématiques pures) et utilisée en BD pour formaliser la sémantique.

DF entre propriétés d'une entité  $E$  :

Soit  $E$  une entité et  $X, Y$  deux sous-ensembles de propriétés de  $E$ .  $E$  vérifie la **dépendance fonctionnelle**  $X \rightarrow Y$  ( $X$  détermine fonctionnellement  $Y$ ) si, lorsque deux occurrences de  $E$  ont les mêmes valeurs sur  $X$ , elles ont aussi les mêmes valeurs sur  $Y$ .

Remarque :

En d'autres termes, la connaissance des valeurs sur  $Y$  est complètement déterminée par celle des valeurs sur  $X$ .

Les dépendances fonctionnelles ( $DF$ )LES DÉPENDANCES FONCTIONNELLES ( $DF$ )

## Définitions :

- Une  $DF$  est **élémentaire** si elle est de la forme  $X \longrightarrow P$  avec  $P$  une propriété telle que  $P \notin X$  et telle qu'il n'existe pas  $X' \subset X$  avec  $X' \longrightarrow P$  (i.e.  $X$  tout entier est nécessaire pour déterminer la valeur sur  $P$ ).

Les dépendances fonctionnelles ( $DF$ )LES DÉPENDANCES FONCTIONNELLES ( $DF$ )

## Définitions :

- Une  $DF$  est **élémentaire** si elle est de la forme  $X \rightarrow P$  avec  $P$  une propriété telle que  $P \notin X$  et telle qu'il n'existe pas  $X' \subset X$  avec  $X' \rightarrow P$  (i.e.  $X$  tout entier est nécessaire pour déterminer la valeur sur  $P$ ).
- Une  $DF$   $X \rightarrow Y$  est **directe** pour  $E$  s'il n'existe pas d'ensemble  $Z$  de propriétés de  $E$  tel que  $X \rightarrow Z$  et  $Z \rightarrow Y$  existent.

Les dépendances fonctionnelles (*DF*)

# LES DÉPENDANCES FONCTIONNELLES (*DF*)

## Remarques :

La considération de la transitivité permet d'éliminer toutes les *DF* non directes d'une entité sans perdre l'information associée. L'identifiant d'une entité est toujours choisi de manière à ce que toutes les propriétés de  $E$  en dépendent fonctionnellement de façon *directe*. Avec la *normalisation* on impose aussi que ces *DF* soient élémentaires. On rajoute donc le terme *minimal* dans la définition initiale de l'identifiant.

Les dépendances fonctionnelles (*DF*)

# LES DÉPENDANCES FONCTIONNELLES (*DF*)

Dépendances fonctionnelles entre entités :

Il existe une dépendance fonctionnelle de l'entité  $E$  vers l'entité  $E'$  si toute occurrence de  $E$  détermine fonctionnellement une unique occurrence de  $E'$ .

Les dépendances fonctionnelles (*DF*)

# LES DÉPENDANCES FONCTIONNELLES (*DF*)

Dépendances fonctionnelles entre entités :

Il existe une dépendance fonctionnelle de l'entité  $E$  vers l'entité  $E'$  si toute occurrence de  $E$  détermine fonctionnellement une unique occurrence de  $E'$ .

Les cardinalités (0,1) et (1,1) traduisent l'existence d'une *DF* entre deux entités, que l'on exprime par une *DF* entre les deux identifiants correspondants. Les associations concernées sont appelées **contraintes d'intégrité fonctionnelle (CIF)**.

Les dépendances fonctionnelles (*DF*)

# PROPRIÉTÉS DES DÉPENDANCES FONCTIONNELLES

Les *DF* obéissent à un certain nombre de règles d'inférence, appelées les :

## Axiomes d'Armstrong :

Soit  $E$  une entité et  $X, Y, Z$  trois sous-ensembles de propriétés de  $E$ .

- ① Réflexivité :  $Y \subseteq X \implies X \rightarrow Y$  (dépendances *triviales*)
- ② Augmentation :  $X \rightarrow Y \implies XZ \rightarrow YZ$
- ③ Transitivité :  $X \rightarrow Y$  et  $Y \rightarrow Z \implies X \rightarrow Z$

Les dépendances fonctionnelles (*DF*)

## PROPRIÉTÉS DES DÉPENDANCES FONCTIONNELLES

Autres règles déduites des précédents axiomes :

- ① Additivité :  $X \rightarrow Y$  et  $X \rightarrow Z \implies X \rightarrow YZ$
- ② Pseudo-transitivité :  $X \rightarrow Y$  et  
 $WY \rightarrow Z \implies XW \rightarrow Z$
- ③ Décomposition :  $X \rightarrow Y$  et  $Z \subseteq Y \implies X \rightarrow Z$

Les dépendances fonctionnelles (*DF*)

## PROPRIÉTÉS DES DÉPENDANCES FONCTIONNELLES

Autres règles déduites des précédents axiomes :

- ① Additivité :  $X \rightarrow Y$  et  $X \rightarrow Z \implies X \rightarrow YZ$
- ② Pseudo-transitivité :  $X \rightarrow Y$  et  
 $WY \rightarrow Z \implies XW \rightarrow Z$
- ③ Décomposition :  $X \rightarrow Y$  et  $Z \subseteq Y \implies X \rightarrow Z$

Définition :

Soit  $\mathcal{F}$  un ensemble de *DF*. La *DF*  $X \rightarrow Y$  est *impliquée* par  $\mathcal{F}$  si on peut l'obtenir à partir des *DF* de  $\mathcal{F}$  en appliquant les axiomes d'Armstrong. On appelle *fermeture* de  $F$  (notée  $\mathcal{F}^+$ ) l'ensemble des *DF* impliquées par  $\mathcal{F}$ .

## FERMETURE D'UN ENSEMBLE D'ATTRIBUTS

Théorème :

L'ensemble  $\mathcal{F}^+$  est valide et complet.

Les dépendances fonctionnelles ( $DF$ )

# FERMETURE D'UN ENSEMBLE D'ATTRIBUTS

Théorème :

L'ensemble  $\mathcal{F}+$  est valide et complet.

Définition :

La **fermeture** d'un ensemble  $X$  de propriétés relativement à un ensemble  $\mathcal{F}$  de  $DF$  — notée  $X+$  — est l'ensemble des propriétés  $P$  telles que la  $DF X \rightarrow P$  est impliquée par  $\mathcal{F}$ .

Les dépendances fonctionnelles ( $DF$ )

# ALGORITHME DE CALCUL DE LA FERMETURE D'UN ENSEMBLE $X$ DE PROPRIÉTÉS D'UNE ENTITÉ $E$ :

①  $X+ := X$

② Repeat

- $Aux := X+$

- $X+ := X+ \cup \{Z : \exists Y \rightarrow Z \in \mathcal{F} \text{ with } Y \subseteq X+\}$

Until ( $Aux = X+$ ) ou ( $X+ = S$ )

EXERCICES : 1. Calculer la fermeture de  $(AB)$  relativement à  $\mathcal{F} = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E\}$ .

Les dépendances fonctionnelles ( $DF$ )

# ALGORITHME DE CALCUL DE LA FERMETURE D'UN ENSEMBLE $X$ DE PROPRIÉTÉS D'UNE ENTITÉ $E$ :

①  $X+ := X$

② Repeat

- $Aux := X+$

- $X+ := X+ \cup \{Z : \exists Y \rightarrow Z \in \mathcal{F} \text{ with } Y \subseteq X+\}$

Until ( $Aux = X+$ ) ou ( $X+ = S$ )

EXERCICES : 1. Calculer la fermeture de  $(AB)$  relativement à  $\mathcal{F} = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E\}$ .

2. comment savoir si un ensemble  $X$  est une clé de  $E$  ?

Les dépendances fonctionnelles (DF)

# FORMES NORMALES ET RÈGLES DE NORMALISATION

## Objectifs des formes normales :

- Les trois premières *formes normales* ont pour but de permettre la formalisation d'un monde réel en de "bonnes" entités et de "bonnes" associations, en utilisant l'outil des DF.

# FORMES NORMALES ET RÈGLES DE NORMALISATION

Objectifs des formes normales :

- Les trois premières *formes normales* ont pour but de permettre la formalisation d'un monde réel en de "bonnes" entités et de "bonnes" associations, en utilisant l'outil des DF.
- Plus le degré de normalité est haut, plus les redondances et anomalies de mise à jour de la base sont réduites mais plus le temps moyen d'accès aux données augmente car la base est morcelée en de nombreuses entités et associations.

# FORMES NORMALES ET RÈGLES DE NORMALISATION

## Objectifs des formes normales :

- Les trois premières *formes normales* ont pour but de permettre la formalisation d'un monde réel en de "bonnes" entités et de "bonnes" associations, en utilisant l'outil des DF.
- Plus le degré de normalité est haut, plus les redondances et anomalies de mise à jour de la base sont réduites mais plus le temps moyen d'accès aux données augmente car la base est morcelée en de nombreuses entités et associations.
- La *3ème forme normale* est généralement considérée comme un bon compromis.

# NORMALISATION DES ENTITÉS

Définition :

- ① Une entité *E* est en **1ère forme normale (1NF)** si toutes ses propriétés sont *atomiques* (i.e. dont la définition n'est pas décomposables en plusieurs autres propriétés) et si elle possède au moins un identifiant.

Les dépendances fonctionnelles (*DF*)

## NORMALISATION DES ENTITÉS

Définition :

- ① Une entité  $E$  est en **1ère forme normale (1NF)** si toutes ses propriétés sont *atomiques* (i.e. dont la définition n'est pas décomposables en plusieurs autres propriétés) et si elle possède au moins un identifiant.
- ② Une entité  $E$  est en **2ème forme normale (2NF)** si :
  - $E$  est en 1ère forme normale et
  - toute propriété de  $E$  dépend de l'identifiant de façon élémentaire (i.e. ne dépend pas d'une partie stricte de l'identifiant).

Les dépendances fonctionnelles (*DF*)

## NORMALISATION DES ENTITÉS

Définition :

- ➊ Une entité  $E$  est en **1ère forme normale (1NF)** si toutes ses propriétés sont *atomiques* (i.e. dont la définition n'est pas décomposables en plusieurs autres propriétés) et si elle possède au moins un identifiant.
- ➋ Une entité  $E$  est en **2ème forme normale (2NF)** si :
  - $E$  est en 1ère forme normale et
  - toute propriété de  $E$  dépend de l'identifiant de façon élémentaire (i.e. ne dépend pas d'une partie stricte de l'identifiant).
- ➌ Une entité  $E$  est en **3ème forme normale (3NF)** si :
  - $E$  est en 2ème forme normale et
  - toute propriété de  $E$  dépend de l'identifiant de façon directe.

## NORMALISATION DES ASSOCIATIONS

Principe :

Une propriété appartient à une association  $A$  si et seulement si elle dépend fonctionnellement et de façon élémentaire de l'*ensemble entier* des identifiants de la collection de  $A$ .

Les dépendances fonctionnelles (*DF*)

## ELABORATION D'UN *MCD* AVEC LE MODÈLE *E/A* :

La démarche à suivre pour aboutir à un *MCD* utilise :

- le **dictionnaire des données** = document qui recense de façon exhaustive toutes les informations d'intérêt présentes dans le *SI*. Concrètement, ce dictionnaire doit regrouper l'ensemble des propriétés qui devront être modélisées dans le *MCD*, décrites par leur nom, un descriptif et, éventuellement, des informations sur les structures de données (en prévision de la construction du niveau physique).

Les dépendances fonctionnelles (*DF*)

## ELABORATION D'UN *MCD* AVEC LE MODÈLE *E/A* :

La démarche à suivre pour aboutir à un *MCD* utilise :

- le **dictionnaire des données** = document qui recense de façon exhaustive toutes les informations d'intérêt présentes dans le *SI*. Concrètement, ce dictionnaire doit regrouper l'ensemble des propriétés qui devront être modélisées dans le *MCD*, décrites par leur nom, un descriptif et, éventuellement, des informations sur les structures de données (en prévision de la construction du niveau physique).
- les **règles de gestion**.

Les dépendances fonctionnelles (*DF*)

# MODÉLISATION DES HISTORIQUES

Sur des exemples.

# Outline

- 1 Introduction
- 2 La méthode Merise
- 3 Le modèle conceptuel de données sous Merise : le modèle Entité-Association
  - Les dépendances fonctionnelles (*DF*)
- 4 Le modèle relationnel de données : un *MLD* particulier

# LE MODÈLE RELATIONNEL DE DONNÉES (MRD)

Introduit par E. F. Codd (centre de recherche d'IBM de San José, 1970), aujourd'hui l'un des plus utilisés. Modèle logique utilisé dans Merise, le modèle relationnel est fondé sur la théorie mathématique des relations, i.e. la théorie des ensembles. Trois notions fondamentales :

Domaine :

Ensemble de valeurs autorisé pour les données. Défini en *extension* (liste explicite de ses valeurs) ou en *intention* (par une propriété caractéristique des valeurs).

# LE MODÈLE RELATIONNEL DE DONNÉES (MRD)

Introduit par E. F. Codd (centre de recherche d'IBM de San José, 1970), aujourd'hui l'un des plus utilisés. Modèle logique utilisé dans Merise, le modèle relationnel est fondé sur la théorie mathématique des relations, i.e. la théorie des ensembles. Trois notions fondamentales :

Domaine :

Ensemble de valeurs autorisé pour les données. Défini en *extension* (liste explicite de ses valeurs) ou en *intention* (par une propriété caractéristique des valeurs).

EXEMPLES : ENTIER, REEL, BOOLEEN, CARACTERES : définis en intention (qu'on peut préciser en des domaines plus spécifiques : MONNAIE, DATE, TEMPS). Le domaine Couleur-Vin = {Rosé, Blanc, Rouge} : défini en extension.

# DOMAINE ET PRODUIT CARTÉSIEN

Produit cartésien :

Le produit cartésien  $D_1 \times D_2 \times \dots \times D_n$  d'un ensemble de domaines  $D_1, D_2, \dots, D_n$  est l'ensemble des  $n$ -uplets  $(v_1, v_2, \dots, v_n)$  avec  $v_i \in D_i$  pour tout  $i \in \{1, \dots, n\}$ .

# DOMAINE ET PRODUIT CARTÉSIEN

Produit cartésien :

Le produit cartésien  $D_1 \times D_2 \times \dots \times D_n$  d'un ensemble de domaines  $D_1, D_2, \dots, D_n$  est l'ensemble des  $n$ -uplets  $(v_1, v_2, \dots, v_n)$  avec  $v_i \in D_i$  pour tout  $i \in \{1, \dots, n\}$ .

Exemple : Si  $\text{Cru} = \{\text{Volnay}, \text{Sancerre}, \text{Chablis}\}$ , le produit cartésien  $\text{Couleur-Vin} \times \text{Cru}$  donne ?

# RELATION ET ATTRIBUTS

Définition :

Sous-ensemble du produit cartésien d'une liste de domaines. Toute relation est identifiée par un nom et est généralement représentée sous la forme d'un tableau à deux dimensions (Codd). Chaque ligne correspond à un élément de la relation donc à un enregistrement de la base.

Exemple : La relation "Couleur-Cru", dont "Couleur" et "Cru" sont les deux *attributs* qui en décrivent la sémantique.

Couleur	Cru
Rosé	Sancerre
Rosé	Chablis
Rouge	Volnay
Rouge	Sancerre
Rouge	Chablis
Blanc	Sancerre

## EXTENSIONS ET INTENTIONS

Définition en extension d'une relation :

Comme tout modèle de données, le modèle relationnel décrit des données dont les valeurs varient dans le temps. Les relations varient dans le temps en ce sens que des tuples sont ajoutés, supprimés ou modifiés au cours de sa vie. Cependant sa structure est un invariant dans le temps et est capturée dans le concept de *schéma de relation* : nom de la relation suivi de la liste de ses attributs (et éventuellement de la définition de leur domaine).

# EXTENSIONS ET INTENTIONS

Définition en extension d'une relation :

Comme tout modèle de données, le modèle relationnel décrit des données dont les valeurs varient dans le temps. Les relations varient dans le temps en ce sens que des tuples sont ajoutés, supprimés ou modifiés au cours de sa vie. Cependant sa structure est un invariant dans le temps et est capturée dans le concept de *schéma de relation* : nom de la relation suivi de la liste de ses attributs (et éventuellement de la définition de leur domaine).

Notation :  $R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$











