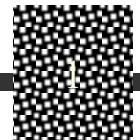




# Résumé

---

- *Schéma conceptuel des données* (SCD) à l'aide du modèle E/A
- Passage au modèle relationnel  $\Rightarrow$  premier *schéma logique des données* (SLD)
- **Problème** : les deux étapes précédentes ne nous garantissent pas l'obtention d'un SLD "optimal"  $\Rightarrow$  un processus de *normalisation* des relations est nécessaire



# Pourquoi normaliser : exemple

Relation Produit

prod_id	libellé	pu	qte	dep_id	adr	volume
p1	K7	23.5	300	1	Nancy	9000
p1	K7	23.5	500	2	Laxou	6000
p3	Vis	10	900	4	Nancy	2000

- **Redondance** : libellé et pu apparaissent pour chaque instance d'un produit
- **Risque d'introduction d'incohérence** lors de l'insertion d'une nouvelle instance de p1
- **Risque de perte d'information** : suppression du produit p3  $\Rightarrow$  on perd son libellé, son pu et les informations relatives au dépôt 4

# Les dépendances fonctionnelles

- Un attribut ou une liste d'attributs  $Y$  *dépend fonctionnellement* d'un attribut ou d'une liste d'attributs  $X$  dans une relation  $R$ , si étant donnée une valeur de  $X$ , il ne lui est associé qu'une seule valeur de  $Y$  dans toute instance de  $R$ . On note  $X \xrightarrow{R} Y$  une telle dépendance.

- Exemples :

$prod\_id \rightarrow libellé$

$prod\_id \rightarrow pu$

$dep\_id \rightarrow adr, volume$

$prod\_id, dep\_id \rightarrow qte$

# Propriétés des dépendances fonctionnelles

Soit  $U$  la liste des attributs de  $R$ , on a alors les propriétés :

$$(F1) \text{ Réflexivité : } Y \subseteq X \wedge X \subseteq U \Rightarrow X \xrightarrow{R} Y$$

$$(F2) \text{ Augmentation : } X \xrightarrow{R} Y \Rightarrow X \cup Z \xrightarrow{R} Y \cup Z$$

$$(F3) \text{ Transitivité : } X \xrightarrow{R} Y \wedge Y \xrightarrow{R} Z \Rightarrow X \xrightarrow{R} Z$$

$$(F4) \text{ Union : } X \xrightarrow{R} Y \wedge X \xrightarrow{R} Z \Rightarrow X \xrightarrow{R} Y \cup Z$$

$$(F5) \text{ Pseudo-transitivité : } X \xrightarrow{R} Y \wedge Y \cup W \xrightarrow{R} Z \Rightarrow X \cup W \xrightarrow{R} Z$$

$$(F6) \text{ Décomposition : } X \xrightarrow{R} Y \wedge Z \subseteq Y \Rightarrow X \xrightarrow{R} Z$$

# Typologie des dépendances fonctionnelles

- Une dépendance  $X \xrightarrow{R} Y$  est :
  - *triviale* si  $Y \subseteq X$
  - *élémentaire* si pour tout  $X' \subset X$ , la dépendance fonctionnelle  $X' \xrightarrow{R} Y$  n'est pas vraie ( $Y$  ne dépend pas fonctionnellement d'une partie de  $X$ )
  - *canonique* si sa partie droite ne comporte qu'un seul attribut
  - *directe* si (i) elle est élémentaire et si (ii)  $Y$  ne dépend pas transitivement de  $X$
- Exemples :
  - $prod\_id, libelle \rightarrow libelle ??$
  - $prod\_id, libelle \rightarrow pu ??$
  - $prod\_id, dep\_id \rightarrow qte ??$



# Clé d'une relation

---

- Un attribut ou une liste d'attributs  $X$  est une *clé* pour la relation  $R(X,Y)$  si  $Y$  dépend fonctionnellement de  $X$  dans  $R$
- $X$  est une *clé minimale* si  $X \xrightarrow{R} Y$  est élémentaire
- Si une relation possède plusieurs clés, nous en choisissons une qui sera appelée *clé primaire* (soulignée dans les schémas de relation). Les autres clés seront appelées *clés secondaires*

Ex : Voiture(immat#, chassis#, type, marque, puissance) admet immat# et chassis# comme clés





# Première forme normale

---

**1NF** Une relation est dite en *première forme normale* si chacun de ses attributs a un domaine atomique mono-valué

■ Exemple :

Personne(id, nom, les\_diplômes) où les\_diplômes est l'ensemble des diplômes obtenus par une personne n'est pas en 1NF

Personne(id, nom) est en 1NF

Diplôme(id, unDiplome) est en 1NF

# Deuxième forme normale

**2NF** Une relation  $R$  est dite en *deuxième forme normale* si  
(i) elle est en 1NF et (ii) tout attribut n'appartenant pas à une clé ne dépend pas d'une partie de la clé de  $R$

■ Exemple :

Stock(prod\_id, dep\_id, libellé, qte) n'est pas en 2NF car

$prod\_id, dep\_id \rightarrow qte$

$prod\_id \rightarrow libellé$

Stock(prod\_id, dep\_id, qte) est en 2NF

Produit(prod\_id, libellé) est en 2NF



# Troisième forme normale

**3NF** Une relation est dite en *troisième forme normale* si (i) elle est en 2NF et (ii) tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé

■ Exemple :

Relation Avion

<u>no_avion</u>	constructeur	type	capacité	propriétaire
AH32	Boeing	B747	C1	Air France
FM34	Airbus	A320	C2	British Airways
BA45	Boeing	B747	C1	Egypt Air

n'est pas en 3NF car *type* → *capacité*, *constructeur* et *type* n'est pas une clé

Avion(no\_avion, type, propriétaire) est en 3NF

Modèle(type, constructeur, capacité) est en 3NF



# Algorithme de normalisation par synthèse *simplifié*

---

- 0 Ecrire les dépendances fonctionnelles (df)
- 1 Rendre canoniques et élémentaires les df qui ne le sont pas
- 2 Représenter les nouvelles df sous forme d'un graphe dont les nœuds sont les attributs impliqués dans les df et les arcs les df elles-mêmes
- 3 Eliminer les df non directes
- 4 Partitionner les df en  $G_1, \dots, G_n$  de façon à ce que toutes les df d'une partition aient la même partie gauche
- 5 Constituer une relation  $R_i$  par  $G_i$ , la partie gauche de  $G_i$  étant clé de  $R_i$  : chaque  $R_i$  est alors en 3NF

# Exemple

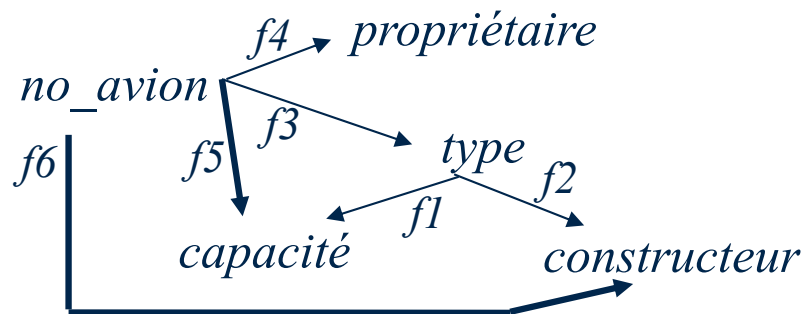
0

$type \rightarrow capacité, constructeur$   
 $no\_avion \rightarrow type, propriétaire, capacité, constructeur$

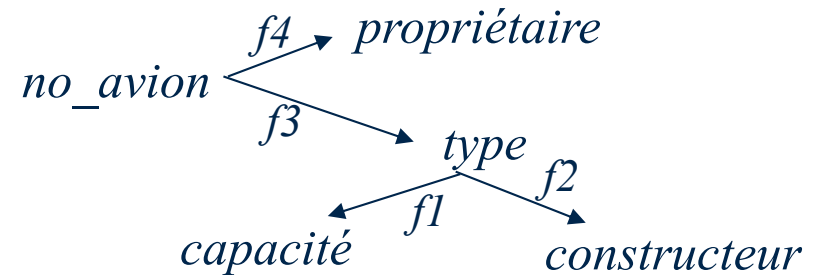
1

$type \rightarrow capacité (f1)$   
 $type \rightarrow constructeur (f2)$   
 $no\_avion \rightarrow type (f3)$   
 $no\_avion \rightarrow propriétaire (f4)$   
 $no\_avion \rightarrow capacité (f5)$   
 $no\_avion \rightarrow constructeur (f6)$

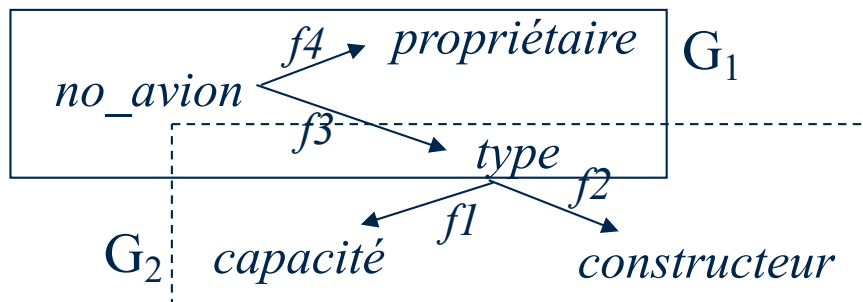
2



3



4



5

Avion(no\_avion, type, propriétaire)  
 Modèle(type, constructeur, capacité)



# Passage du modèle E/A au modèle relationnel

---

- Si règle de construction et de validation du modèle E/A sont respectés alors modèle E/A en troisième forme normale:
  - Les propriétés sont sous forme élémentaire
  - A toute valeur prise par l'identifiant ne correspond qu'une valeur de chaque propriété
  - Chaque propriété d'une relation dépend de la totalité des entités qu'elle relie
  - Toutes les propriétés dépendent directement de l'identifiant