

Bases de données

Normalisation d'un schéma relationnel

Pourquoi normaliser un schéma de relations?

- **Support théorique d'une « bonne » conception de schémas pour**
 - éliminer les redondances sans perte d'information
 - éviter les incohérences de mise à jour
 - éviter, autant que possible, les valeurs nulles
 - mieux comprendre les relations sémantiques entre les données

Exemple

| <u>NumPropriétaire</u> | Nom | Ville | <u>NumVéhicule</u> | Marque | Date |
|------------------------|-------|--------|--------------------|---------|-----------|
| 1000 | AAAA | PARIS | 90FE75 | PEUGEOT | 10-sep-89 |
| 1500 | BBBBB | NANTES | 43XY97 | RENAULT | 02-fev-96 |
| 1000 | AAAA | PARIS | 56GT98 | FIAT | 06-mar-91 |
| 1350 | CCCC | NICE | 43ZT88 | RENAULT | 28-dec-87 |
| 1500 | BBBBB | NANTES | 57TG92 | PEUGEOT | 26-jui-91 |

- Redondances \Rightarrow risque d'anomalies
- Insertion d'une personne sans voiture \Rightarrow introduction de valeurs nulles
- Suppression de la dernière voiture possédée par une personne \Rightarrow perte d'information

Décomposition

| <u>NumPropriétaire</u> | Nom | Ville |
|------------------------|-------|--------|
| 1000 | AAAA | PARIS |
| 1500 | BBBBB | NANTES |
| 1350 | CCCC | NICE |

| <u>NumVéhicule</u> | Marque |
|--------------------|---------|
| 90FE75 | PEUGEOT |
| 43XY97 | RENAULT |
| 56GT98 | FIAT |
| 43ZT88 | RENAULT |
| 57TG92 | PEUGEOT |

| <u>NumPropriétaire</u> | <u>NumVéhicule</u> | Date |
|------------------------|--------------------|-----------|
| 1000 | 90FE75 | 10-sep-89 |
| 1500 | 43XY97 | 02-fev-96 |
| 1000 | 56GT98 | 06-mar-91 |
| 1350 | 43ZT88 | 28-dec-87 |
| 1500 | 57TG92 | 26-jui-91 |

Comment normaliser un schéma relationnel?

■ Approche par décomposition :

- **Entrées** : une table contenant tous les attributs et des dépendances fonctionnelles
- **Processus** : itératif c'est-à-dire décompositions successives jusqu'à ce qu'il n'y ait plus de redondances
- **Sortie** : un ensemble de tables normalisées

■ Approche par synthèse :

- **Entrées** : une table contenant tous les attributs et des dépendances fonctionnelles
- **Processus** : non itératif basé sur le calcul de la couverture minimale des DFs
- **Sortie** : un ensemble de tables normalisées

Rappel de la définition d'une DF

- Soient
 - $R(A_1, A_2, \dots, A_n)$ un schéma de relation
 - X et Y deux sous-ensembles de $\{A_1, A_2, \dots, A_n\}$
- On dit que :
 - Y dépend fonctionnellement de X ou X détermine Y (**$X \rightarrow Y$**) si quelle que soit l'instance de R , pour tout tuple $T1, T2$ de R on a :
 - **$T1[X] = T2[X] \Rightarrow T1[Y] = T2[Y]$**
 - Ou encore si, à tout moment, lorsque deux tuples ont les mêmes valeurs pour les colonnes de X , ils ont alors nécessairement les mêmes valeurs pour les colonnes de Y
- NB : $Ti[X]$ est la valeur de X pour le tuple Ti

Exemple

| NumPropriétaire | Nom | Ville | NumVéhicule | Marque | Date |
|-----------------|-------|--------|-------------|---------|-----------|
| 1000 | AAAA | PARIS | 90FE75 | PEUGEOT | 10-sep-89 |
| 1500 | BBBBB | NANTES | 43XY97 | RENAULT | 02-fev-96 |
| 1000 | AAAA | PARIS | 56GT98 | FIAT | 06-mar-91 |
| 1350 | CCCC | NICE | 43ZT88 | RENAULT | 28-dec-87 |
| 1500 | BBBBB | NANTES | 57TG92 | PEUGEOT | 26-jui-91 |

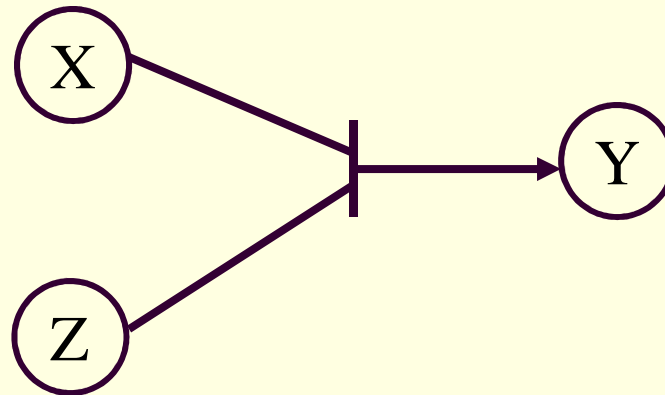
- NumPropriétaire → Nom
- NumPropriétaire → Ville
- NumVéhicule → Marque
- NumPropriétaire, NumVéhicule → Date

Graphe des DFs

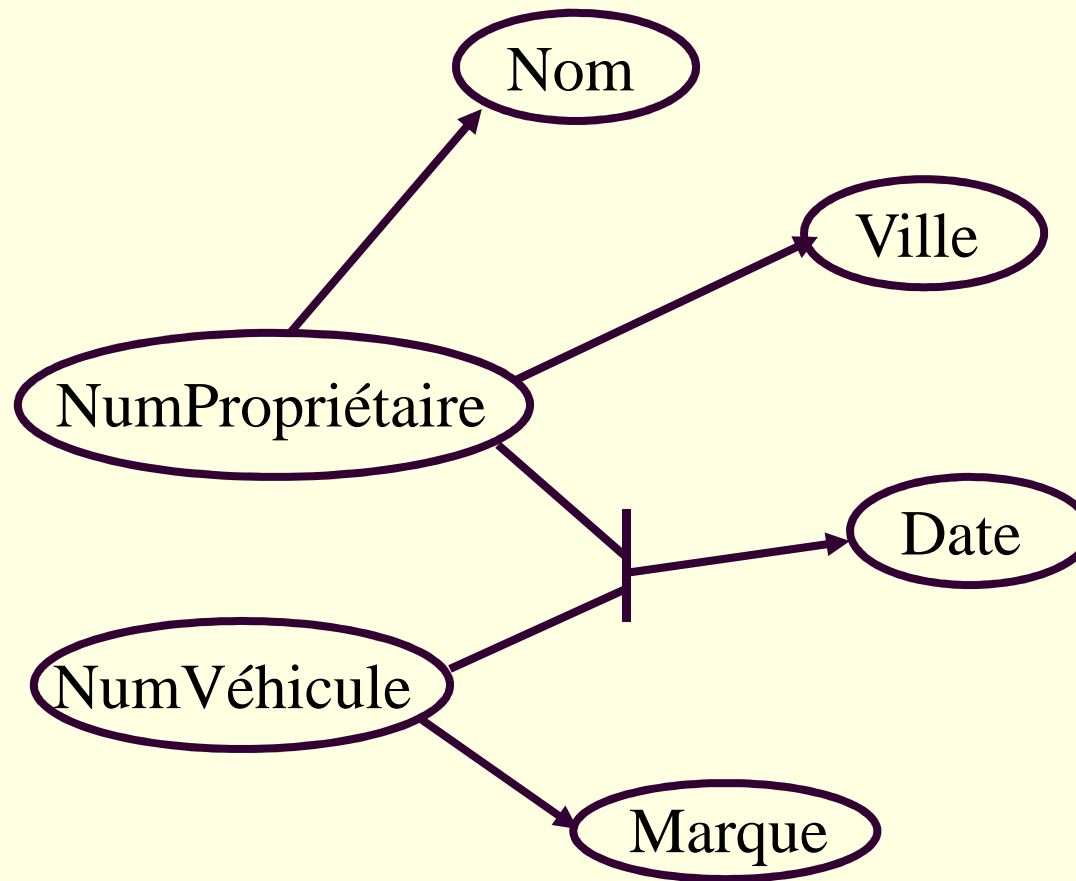
$X \rightarrow Y$



$X, Z \rightarrow Y$



Exemple



Un autre exemple : la table cours

| Nomprof | Ville | Département | Nometud | Age | cours | Note |
|---------|-------|-------------|---------|-----|----------|------|
| Dupont | Lille | 59 | Alfred | 22 | Math | 12 |
| Dupont | Lille | 59 | Arthur | 25 | Math | 05 |
| Martin | Arras | 62 | Alfred | 22 | Anglais | 18 |
| Martin | Arras | 62 | Pierre | 23 | Anglais | 11 |
| Dupont | Lille | 59 | Pierre | 23 | physique | 13 |
| Charles | Lille | 59 | Pierre | 23 | Histoire | 12 |

- **NOMPROF →VILLE**
- **VILLE →DEPARTEMENT**
- **NOMPROF → DEPARTEMENT**
- **NOMETUD → AGE**
- **NOMETUD, COURS → NOTE**
- **COURS → NOMPROF**

Inférence des Dfs : Axiomes d'Amstrong

- Axiome de réflexivité : $Y \subseteq X \Rightarrow X \rightarrow Y$
- Axiome d'augmentation : $X \rightarrow Y \Rightarrow X, Z \rightarrow Y$
- Axiome de transitivité : $X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$
- Axiome de pseudo-transitivité : $X \rightarrow Y$ et $Y, W \rightarrow Z \Rightarrow X, W \rightarrow Z$
- Axiome d'union : $X \rightarrow Y$ et $X \rightarrow Z \Rightarrow X \rightarrow Y, Z$
- Axiome de décomposition : $X \rightarrow Y$ et $Z \subseteq Y \Rightarrow X \rightarrow Z$

Autres définitions

- **Dépendance élémentaire:**

- $X \rightarrow Y$ est élémentaire ssi $Y \notin X$ et $\forall X' \subset X, X' \not\rightarrow Y$

- **Fermeture transitive d'un ensemble F de DFs :**

- Ensemble F^+ de DFs élémentaires obtenu par application de l'axiome de transitivité et de pseudo-transitivité

- **Couverture minimale d'un ensemble F de DFs :**

- Plus petit ensemble de DFs permettant d'obtenir par application successives des axiomes d'inférences la fermeture transitive de F

Exemple

- *COURS* (NOMPROF, VILLE, DEPARTEMENT, NOMETUD, AGE, COURS, NOTE)

La fermeture transitive

1. NOMPROF \rightarrow VILLE
2. VILLE \rightarrow DEPARTEMENT
3. NOMPROF \rightarrow DEPARTEMENT
4. NOMETUD \rightarrow AGE
5. NOMETUD, COURS \rightarrow NOTE
6. COURS \rightarrow NOMPROF

+

7. COURS \rightarrow VILLE
8. COURS \rightarrow DEPARTEMENT

Couverture minimale

1. NOMPROF \rightarrow VILLE
2. VILLE \rightarrow DEPARTEMENT
4. NOMETUDIANT \rightarrow AGE
5. NOMETUD, COURS \rightarrow NOTE
6. COURS \rightarrow NOMPROF

DFs et notion de clé

■ Soient :

$$R(A_1, A_2, \dots, A_n)$$

$$X \subseteq \{A_1, A_2, \dots, A_n\}$$

■ On dit que **X est une clé candidate** de R ssi :

- $X \rightarrow A_1, A_2, \dots, A_n$

- $\forall Y \subset X, Y \not\rightarrow A_1, A_2, \dots, A_n$

Exemple

- R (NumPropriétaire, NumVéhicule, Nom, Ville, Marque, Date)
 - NumPropriétaire \rightarrow Nom
 - NumPropriétaire \rightarrow Ville
 - NumVéhicule \rightarrow Marque
 - NumPropriétaire, NumVéhicule \rightarrow Date



{NumPropriétaire, NumVéhicule} est la seule clé pour R

Formes normales

- 1ère Forme normale 1FN
- 2 ième Forme normale 2FN
- 3ième Forme normale 3FN
- etc.

Relation en 1FN

- Une relation est en première forme normale lorsque aucun attribut n'est un ensemble d'ensembles

ou

- Une relation est en première forme normale si tous ses attributs sont atomiques

EXEMPLE

■ ETUDIANT (Matricule, Nom ,, DIPLOMES)

| | | | |
|----|---|-------|-------------|
| 01 | A | | {Bac, BTS} |
| 02 | B | | {Bac, Deug} |
| 03 | C | | {Bac} |

ETUDIANT n'est pas en 1FN

■ ETUDIANT (Matricule, Nom ,, DIPLOME)

| | | | |
|----|---|-------|------|
| 01 | A | | Bac |
| 01 | A | | BTS |
| 02 | B | | Bac |
| 02 | B | | Deug |
| 03 | C | | Bac |

ETUDIANT est maintenant en 1FN

Relation en 2FN

- Une relation est en deuxième forme normale ssi :
 - 1 - elle est en première forme normale
 - 2 - tout attribut non clé dépend de la totalité de la clé

R (A, B, C, D) en 1FN et $A \rightarrow C \Rightarrow R$ n'est pas en 2FN

Exemple

- R (NumPropriétaire, NumVéhicule, Nom, Ville, Marque, Date

NumPropriétaire \rightarrow Nom

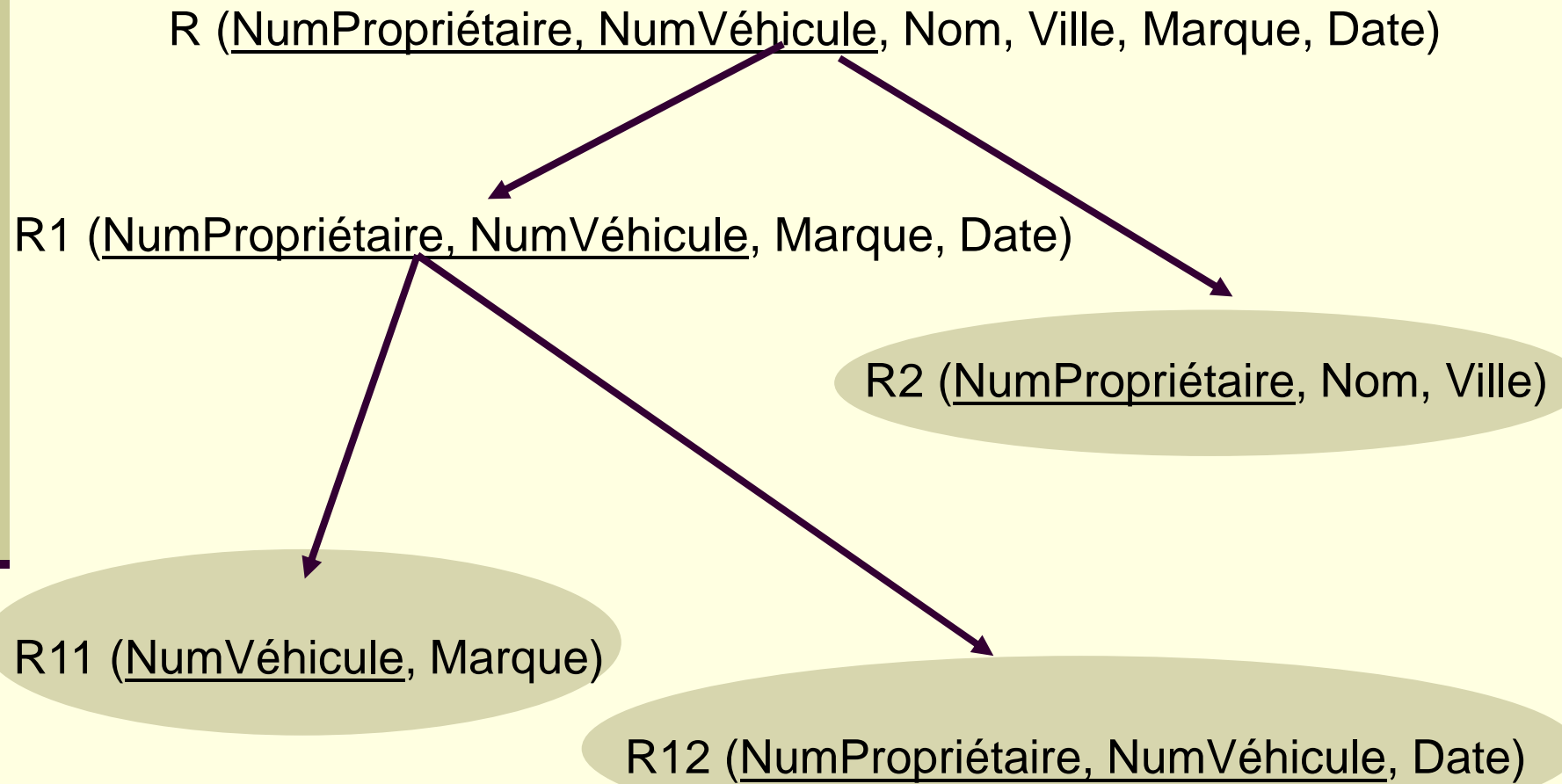
NumPropriétaire \rightarrow Ville

NumVéhicule \rightarrow Marque

NumPropriétaire, NumVéhicule \rightarrow Date

R n'est pas en 2FN

Décomposition de R



Relation en 3FN

- Une relation est en troisième forme normale ssi :
 - 1 - elle est en 2FN
 - 2 - il n'y a pas de dépendance fonctionnelle entre attributs non clé

$R(\underline{A}, C, D)$ en 2FN et $C \rightarrow D \Rightarrow R$ n'est pas en 3FN

Exemple

PRODUIT (NunProduit, Désignation, CodeTVA, TauxTVA)

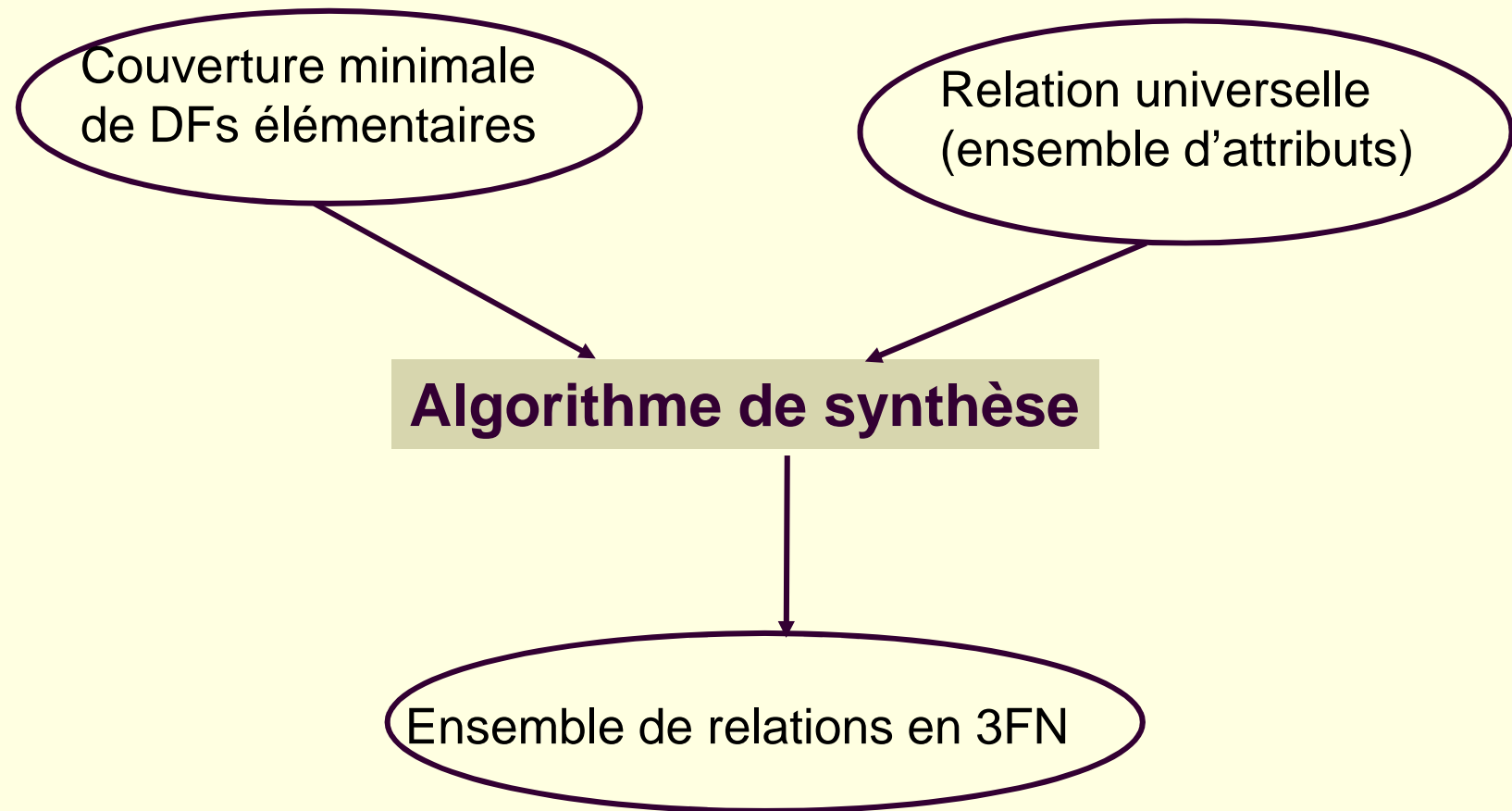
CodeTVA → TauxTVA



PRODUIT (NunProduit, Désignation, CodeTVA)

TVA (CodeTVA, TauxTVA)

Algorithme de synthèse



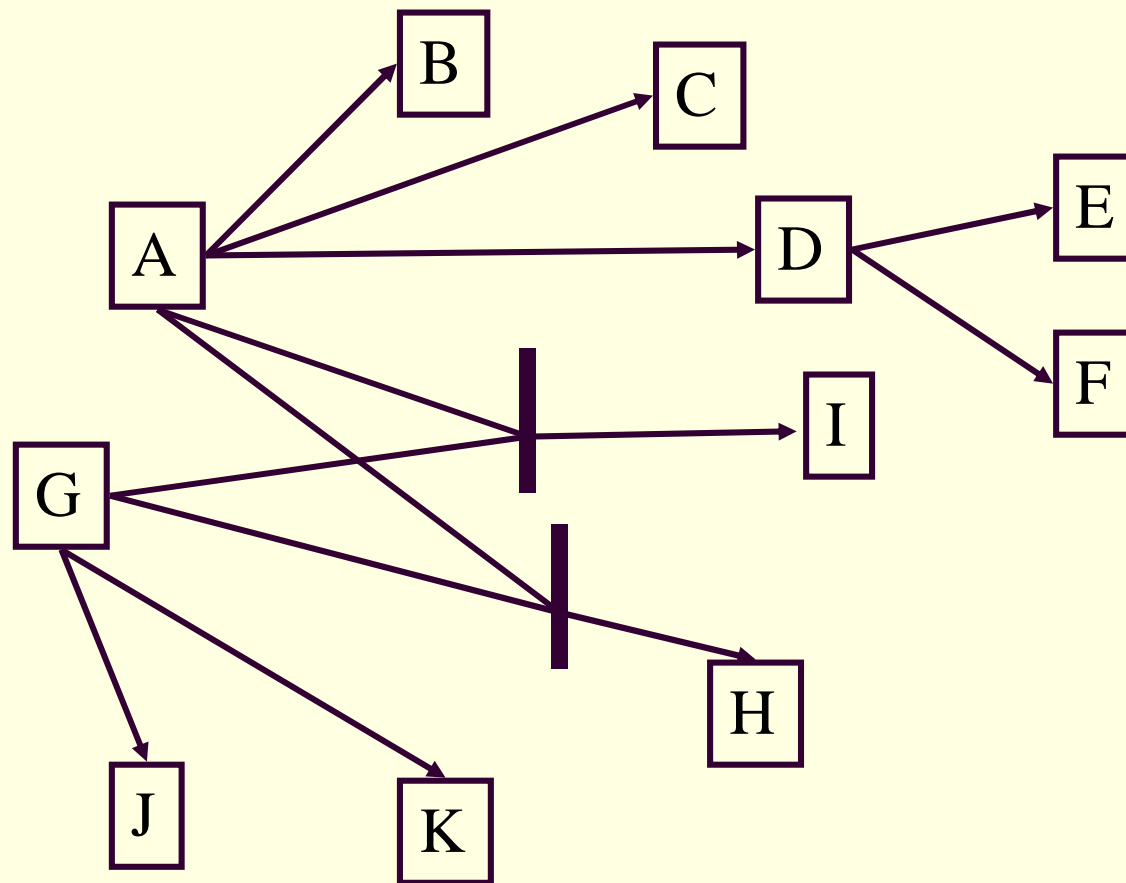
Etape de l'algorithme

1. Regroupement des dépendances de même partie gauche

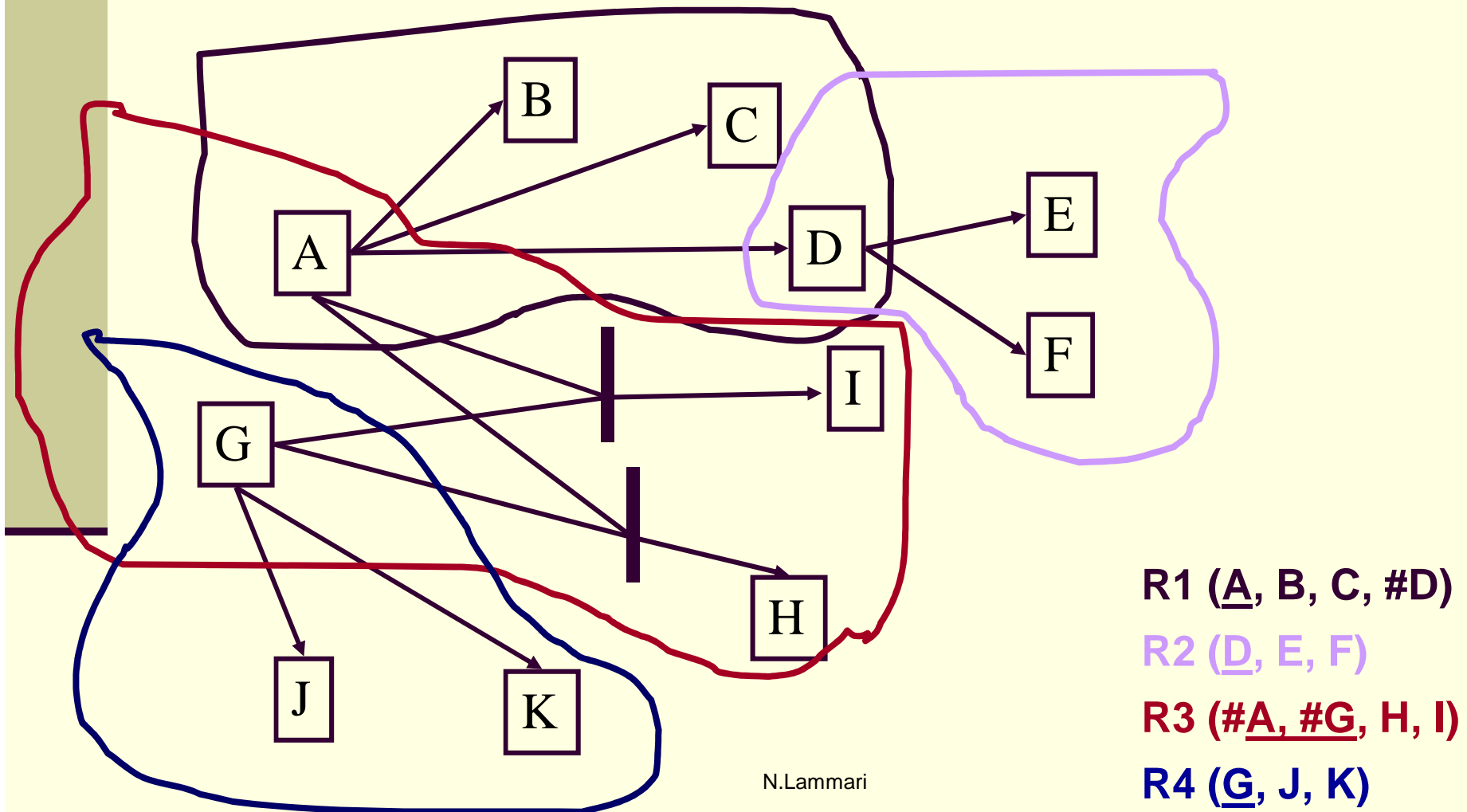
2. Construire une relation pour chaque ensemble.

Chacune des relations a pour clé le groupe d'attribut en
partie gauche

Exemple



Suite exemple



Décomposition versus synthèse

- **Algorithme de décomposition :**

- préserve le contenu
- Conduit à des relations en au moins 3FN

- **Algorithme de synthèse**

- préserve les DFs
- conduit à des relations en 3FN

NB : une décomposition de R en R_1, R_2, \dots, R_n préserve le contenu ssi la jointure des relations de R_1, R_2, \dots, R_n est égale à la relation R