# *Théorie de la Normalisation*



## *La phase de design d'une BD*

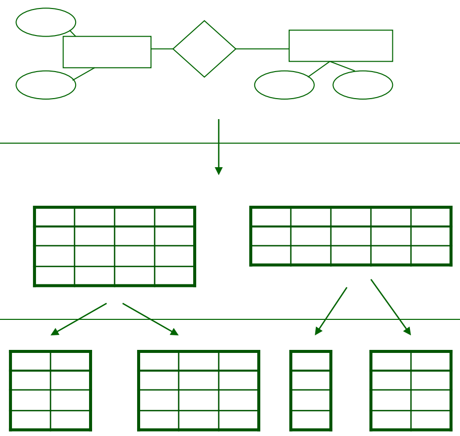
- Analyse des besoins
- Design conceptuel
  - Modèle EA, UML
- Design logique
  - EA vers relations
  - raffinement de schéma: normalisation
- Design physique
  - indexes, etc.

# *Conception de schéma*

Modèle conceptuel:

Modèle relationnel + DFs

Normalisation:
Éliminer les ***anomalies***





# *Théorie de la normalisation*

- Anomalies de conception, redondance
- Dépendances fonctionnelles
- Décomposition de schéma
- Formes normales
- Automatisation de la normalisation
- Conclusion

# *Anomalies de conception de schéma*

VOITURE (N°VEH, TYPE, COULEUR, MARQUE, PUISSANCE)

Un type de voiture peut avoir une seule puissance et une seule marque, mais différentes couleurs, numéros

| VOITURE | NVH | MARQUE | TYPE | PUIS | COULEUR |
|---|---|---|---|---|---|
| | 872RH7 | RENAULT | R21 | 7 | BLEUE |
| | 975AB8 | RENAULT | R21 | 7 | BEIGE |
| | 975AB8 | PEUGEOT | 205 | 8 | ROUGE |

Redondance, anomalies de modification, anomalies d' effacement.



# *Décomposition*

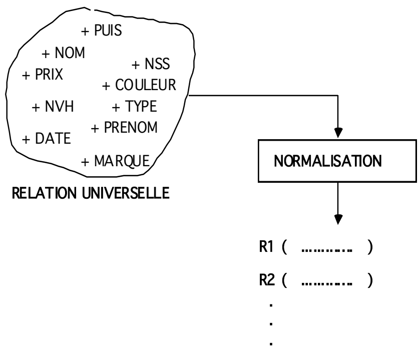
| VOITURE | NVH | MARQUE | TYPE | PUIS | COULEUR |
|---|---|---|---|---|---|
| | 872RH7 | RENAULT | R21 | 7 | BLEUE |
| | 975AB8 | RENAULT | R21 | 7 | BEIGE |
| | 975AB8 | PEUGEOT | 205 | 8 | ROUGE |

| V_TYPE | MARQUE | TYPE | PUIS |
|---|---|---|---|
| | RENAULT | R21 | 7 |
| | PEUGEOT | 205 | 8 |

| V_NVH | NVH | TYPE | COULEUR |
|---|---|---|---|
| | 872RH7 | R21 | BLEUE |
| | 975AB8 | R21 | BEIGE |
| | 975AB8 | 205 | ROUGE |

# *Objectif et principes*

- Problèmes de conception: redondances et anomalies de m.à.j.
- Objectif : Minimiser les redondances pour améliorer les performance de la BD.
  - requêtes d'interrogation et de mise à jour plus rapides
  - moins de valeurs NULL
  - gestion de l'intégrité des données simplifiée
- Principe :






# *Dépendances fonctionnelles*

- Définition :

  *X --> Y, X détermine Y, ou Y dépend fonctionnellement de X,*

  X et Y: ensembles d'attributs de R

  est une assertion sur toutes les instances de la relation R telle que si deux tuples quelconques de R possèdent la même valeur sur X, alors ils doivent avoir la même valeur sur Y.

Formellement :

*X --> Y* ssi quel que soit l'instance *r* de R, pour tout tuple t1 et t2 de *r* on a :

$$\Pi_X(t1) = \Pi_X(t2) \implies \Pi_Y(t1) = \Pi_Y(t2)$$

Utilité :

Contrainte sur les données que peut contenir la relation.

Outil de contrôle d'intégrité des données (niveau schéma).

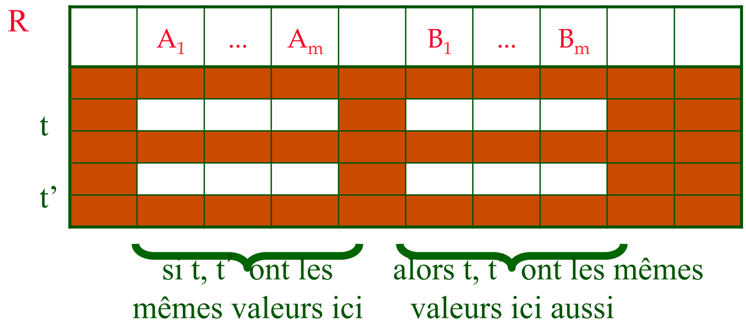
Outil mathématique permettant d'expliquer le processus de la « normalisation »

Vital lorsqu'on veut ré-étudier le schéma d'une BD qui pose problème.

# *Dépendances fonctionnelles*

Definition: $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$ vérifiée dans R si:

$\forall t, t' \in R, (t.A_1=t'.A_1 \wedge \ldots \wedge t.A_m=t'.A_m \Rightarrow t.B_1=t'.B_1 \wedge \ldots \wedge t.B_n=t'.B_n)$






# *Exemples*

| ID | Nom | Tél | Dép |
|---|---|---|---|
| I0 | S | 1234 | Info |
| I3 | M | 9876 | Ventes |
| I1 | S | 9876 | Ventes |
| I9 | M | 1234 | RH |

ID -> Nom, Tél, Dép ?

Dép -> Tél ?

Tél -> Dép ?

## *Observation*

Si ces deux DFs vérifiées:

nom -> couleur
couleur, catégorie -> prix

Alors cette DF vérifiée aussi:

nom, catégorie -> prix

Explication?



## *Axiomes d'Armstrong*

❑ Propriétés de base

**Réflexivité**

Y inclus dans X => X -> Y

**Augmentation**

X -> Y => XZ -> YZ

**Transitivité**

X -> Y et Y -> Z => X -> Z

❑ Propriétés déduites

**Union**

X -> Y
X -> Z
=> X -> YZ

**Pseudo-transitivité**

X -> Y
YW -> Z
=> XW -> Z

**Décomposition**

X -> Y
Z inclus dans Y
=> X -> Z

## *$F^+$ et Couverture minimale*

- Fermeture transitive d'un ensemble F de DF
  - $F^+ = F \cup$ {DF déduites de celles contenues dans F}
- Couverture minimale d'un ensemble F de DF
  - Sous-ensemble minimum de **DF élémentaires** permettant de générer toutes les autres.
  - Théorème : tout ensemble de DF possède une couverture minimale, en général non unique.
  - Formellement, F est une CM ssi :
    - Tout f dans F est élémentaire.
    - Il n'existe pas de f dans F telle que F - {f} soit « équivalent » à F.
- DF élémentaires (les seules utiles)
  - Une **DF élémentaire** est une DF de la forme X -->A, où :
    - A est un attribut, X un ensemble d'attributs, A **n'est pas inclus dans** X
    - **il n'existe pas** X' **inclus dans** X tel que X' -> A dans F+
- Equivalence
  - On dit que deux ensembles de DF sont équivalents s'ils possèdent la même fermeture transitive.



## *Calcul de $F^+$*

- Avec les axiomes de Armstrong
  - Trop compliqué!

## $X^+$: *Fermeture d'un ensemble d'attributs X*

Pour X= $A_1, \ldots, A_n$

La **fermeture,** $\{A_1, \ldots, A_n\}^+$ = l'ensemble des attributs B t.q. $A_1, \ldots, A_n$ -> B

Exemple:

nom -> couleur
catégorie -> département
couleur, catégorie -> prix

Fermetures:

$nom^+$ = {nom, couleur}
{nom, catégorie}$^+$ = {nom, catégorie, couleur, département, prix}
$couleur^+$ = {couleur}



## *Calcul de $X^+$*

$X^+$:=X
do{
pour chaque DF Y->Z dans F tq
Y inclus dans $X^+$
$X^+$:= $X^+$ + Z
}
until ($X^+$ inchangé)

{nom, catégorie}+ = {nom, catégorie, couleur, département, prix}
nom, catégorie -> couleur, département, prix

## *Tester l'appartenance d'une DF à $F^+$*

- Les axiomes d'Armstrong: correctes et complètes (calculent $F^+$ )
- Pour tester l'appartenance de X->Y à $F^+$:
  - calculer la fermeture de X, $X^+=\{A_i \mid X\text{->}A_i\}$
  - tester si Y inclus dans $X^+$

$F^+$: Énumérer les DFs X -> Y t.q. $Y \subseteq X^+$ et $X \cap Y = \emptyset$

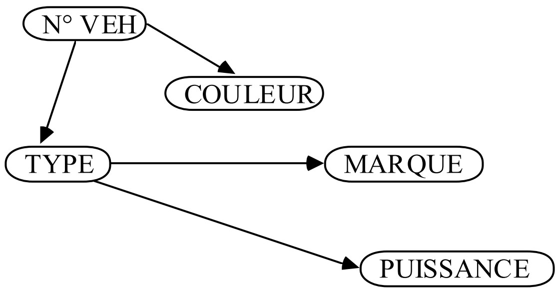


## *Clé d'une relation*

- Définition :
  
  **Ensemble minimum d'attributs qui détermine tous les autres.**
- Formellement :
  
  Soient : R (A1, A2, ..., An) un schéma de relation, F+ l'ensemble des DF associés et X un sous-ensemble de A1, A2, ... , An.
  
  X est une clé de R ssi :
  - X -> A1 A2 ... An est dans F+
  - il n'existe pas de sous-ensemble X' inclus dans X tel que X' -> A1 A2 ... An
- Non unicité :
  - Il peut y avoir plusieurs clés pour une relation (clés candidates)
  - Une est choisie comme clé primaire
- Sur-clé :
  - Sur-clé : X est sur-clé ssi **X->R dans $F^+$**

# Graphe des DF

VOITURE (N°VEH, TYPE, COULEUR, MARQUE, PUISSANCE)



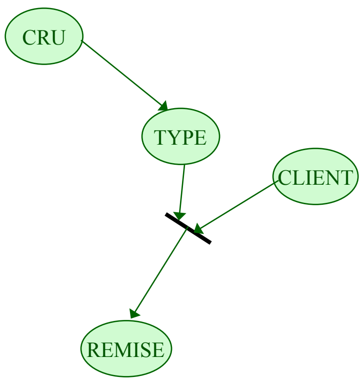



# Autre Exemple

| CRU | TYPE | CLIENT | REMISE |
|---|---|---|---|
| CHENAS | A | C1 | 3% |
| MEDOC | A | C2 | 5% |
| JULIENAS | B | C1 | 4% |

CRU --> TYPE

TYPE, CLIENT --> REMISE



## *Calculer les (sur)clés*

- Calculer $X^+$ pour tous les ensemble X
- Si $X^+$ = {tous les attributs}, alors X est une (sur)clé
- Lister uniquement les Xs minimaux

Les clés décident la «qualité» d'un schéma.



## *Décomposition de schéma*

Définition :

**La décomposition d'un schéma de relation R(A1, ..., An) consiste en son remplacement par une collection de relations (R1,...Rm) tel que**

SCHEM (R) = SCHEM (R1) U SCHEM (R2) U ... U SCHEM (Rm)

But de la décomposition :

**Casser R en de plus petites relations afin d'éviter :**

**les redondances**

**les anomalies en mises à jour**

# *Décomposition Sans Perte d'Information*

Définition

**Une décomposition d'une relation R en m relation (R1, … Rm) est SPI par rapport à un ensemble de DF si et seulement si, quelle que soit I une instance de R, alors**

**$I = I_1 \times I_2 \times \ldots \times I_m$**

**où $I_j = \Pi_{Rj}(I)$**

***c-à-d, Il doit exister une série de jointures sur les instances de R1, …, Rm qui permet de reconstruire I.***

Théorème:

**Une décomposition de R en R1 et R2 est SPI ssi R1 ∩ R2 est sur-clé de R1 et/ou R2.**



# *Décomposition Sans Perte d'Information*

**Preuve :**

On cherche a démontrer que $\Pi_{R1}(r) \times \Pi_{R2}(r) = r$.

- ❖ $r \subseteq \Pi_{R1}(r) \times \Pi_{R2}(r)$ est toujours vrai.

Soient r:

| A | B | C |
|---|---|---|
| a1 | b1 | c1 |
| a2 | b2 | c2 |
| a3 | b1 | c3 |

$\Pi_{R1}(r)$ :

| A | B |
|---|---|
| a1 | b1 |
| a2 | b2 |
| a3 | b1 |

$\Pi_{R2}(r)$ :

| B | C |
|---|---|
| b1 | c1 |
| b2 | c2 |
| b1 | c3 |

$\Pi_{R1}(r) \times \Pi_{R2}(r)$ :

| A | B | C |
|---|---|---|
| a1 | b1 | c1 |
| ***a1*** | ***b1*** | ***c3*** |
| a2 | b2 | c2 |
| ***a3*** | ***b1*** | ***c1*** |
| a3 | b1 | c3 |

# *Décomposition Sans Perte d'Information*

**Preuve :**

On cherche a démontrer que $\Pi_{R1}(r) \times \Pi_{R2}(r) = r$.

- ❖ $\Pi_{R1}(r) \times \Pi_{R2}(r) \subseteq r$ n'est en général pas vrai.

Si $\exists\, t \in \Pi_{R1}(r) \times \Pi_{R2}(r)$ et $t \notin r$

$\Rightarrow$ **R1 ∩ R2 n'est pas sur-clé, ni de R1 ni de R2.**

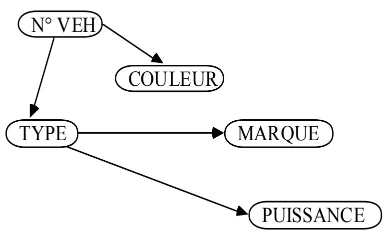


# *Exemple*

| VOITURE | NVH | MARQUE | TYPE | PUIS | COULEUR |
|---|---|---|---|---|---|
| | 872RH75 | RENAULT | R21 | 7 | BLEUE |
| | 975AB80 | RENAULT | R21 | 7 | BEIGE |

| R1 | | | R2 |
|---|---|---|---|
| 8H75 | R21 | BLEUE | R21 RENAULT 7 |
| 97AB80 | R21 | BEIGE | |

VOITURE = R1 X R2




| R'1 | R2 | R'3 |
|---|---|---|
| 872RH75 R21 | R21 7 BLEUE | R21 RENAULT |
| 975AB80 R21 | R21 7 BEIGE | |

VOITURE <> R'1 X R'2 X R'3

R'1 X R'2 X R'3

| | | | | |
|---|---|---|---|---|
| 872RH75 | R21 | 7 | BLEUE | RENAULT |
| 872RH75 | R21 | 7 | BEIGE | RENAULT |
| 975AB80 | R21 | 7 | BLEUE | RENAULT |
| 975AB80 | R21 | 7 | BEIGE | RENAULT |

# Décomposition sans perte de DF

Définition

**Projection de F sur un sous-ensemble X d'attributs: l'ensemble de DFs dans $F^+$ qui utilisent uniquement X.**

Définition

**Une décomposition (R1, …Rm) de R préserve les DFs (SPD) si la fermeture transitive des DFs de R est la même que la fermeture transitive de l'union des projection des DFs sur R1, … Rm.**

Exemple: A->B, B->C, C->A et la décomposition AB, BC

Tester si une décomposition est SPD: par la définition.



# Example (1)

Exemple

1) **R1 (NVH, TYPE, COULEUR) =>F1 = { NVH -> TYPE, NVH -> COULEUR }**

   **R2 (TYPE, MARQUE, PUIS) => F2 = { TYPE -> MARQUE, TYPE -> PUIS }**

   ==> OK

2) **R'1 (NVH, TYPE) => F'1 = { NVH -> TYPE }**

   **R'2(TYPE, PUIS, COULEUR) =>F'2 = {TYPE -> PUIS}**

   **R'3(TYPE, MARQUE) => F'3 = {TYPE -> MARQUE}**

   ==> ON A PERDU LA DF : NVH ->COULEUR

# *Exemple (2)*

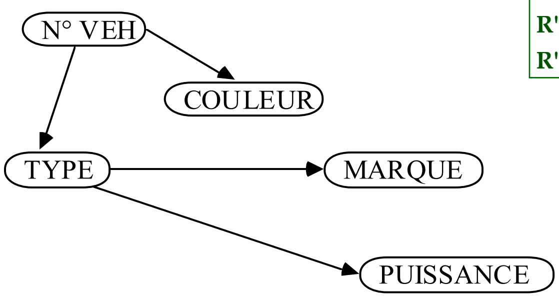
VOITURE (NVH, MARQUE, TYPE, PUIS, COULEUR)

| VOITURE | NVH | MARQUE | TYPE | PUIS | COULEUR |
|---|---|---|---|---|---|
| | 87 2RH75 | RENAULT | R21 | 7 | BLEUE |
| | 97 5AB80 | RENAULT | R21 | 7 | BEIGE |

R1 (NVH, TYPE, COULEUR)
R2 (TYPE, MARQUE, PUIS)

**Décomposition**

**R'1 (NVH, TYPE)**
**R'2 (TYPE, PUIS, COULEUR)**
**R'3 (TYPE, MARQUE)**






# *Formes normales pour éliminer les anomalies*

Deux FN principales, BCNF et 3NF

Idée principale:
X → A est OK si X est un (sur)clé
sinon
X → A mauvaise (BCNF), à moins que A prime (3NF)

A prime: A appartient à une clé.

# *Formes normales*

## Définition 1NF

Une relation est en 1ère forme normale si tout attribut contient une valeur atomique (unique)

## Exemple

| PERSONNE | NOM | PROFESSION |
|---|---|---|
| | DUPONT | Ingénieur, Professeur |
| | MARTIN | Géomètre |

Une telle relation doit être décomposée en répétant les noms pour chaque profession



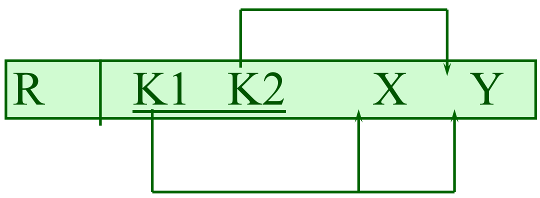
# *2NF*

## Définition

une relation est en 2e forme normale ssi :

- elle est en 1ère forme
- tout attribut non clé ne dépend pas d'une partie de clé

## Schéma




Une telle relation doit être décomposée en
R1(<u>K1,K2</u>,X) et R2(<u>K2</u>,Y)

## *Exemple 2NF*

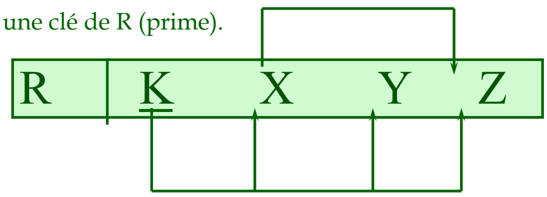
- Exemple 1 :
  - Fournisseur (nom,adresse, article, prix)
  - La clé est (nom, article)
  - Mais nom --> adresse  :  pas en 2e forme

- Exemple 2 :
  - R (cru,type,client,remise)
  - La clé est  (cru, client)
  - Mais cru --> type :  pas en 2e forme



## *3NF*

- Définition
  - une relation est en 3e forme normale ssi :
    - elle est en 1ère forme
    - tout attribut n'appartenant pas a une clé ne dépend pas d'un ensemble d'attributs qui ne sont pas (sur)clé
    - X->A une DF non triviale dans F+ alors :
      - X contient une clé de R ou
      - A fait partie d'une clé de R (prime).
- Schéma




Une telle relation doit être décomposée en

R1(<u>K,</u> X, Y) et R2(X,Z)

## Exemple 3ème Forme

- Exemple
  - Voiture (nvh, marque, type,puissance, couleur)
  - Type --> marque
  - Type --> puissance

  Pas en 3eme forme !

- Caractéristiques de la 3NF :
  - redondance limitée
  - On arrive toujours à une décomposition en 3NF SPI et SPD



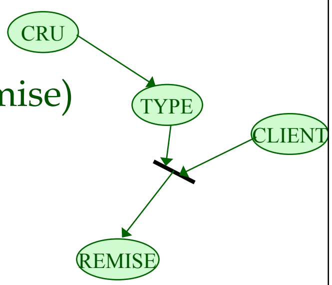
## Exemples de décomposition

- Voiture (nvh, marque, type, puissance, couleur)

  véhicule (nvh, type, couleur)
  Modèle (type, marque, puissance)

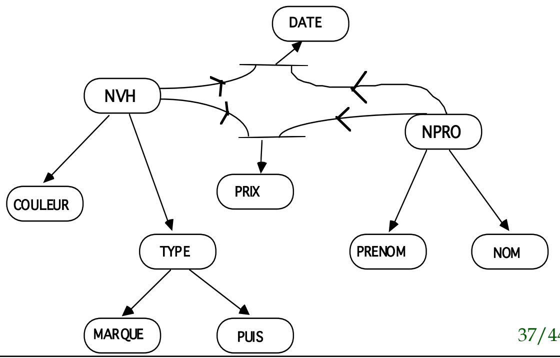
- Réduction (cru, type, client, remise)

  Remise (Type,client,remise)
  Type (cru,type)
  Commande(Cru, Client)



# Algorithme de décomposition

1. A partir du graphe G des DF, calculer une couverture minimale C
2. Rechercher le plus grand ensemble X d'attributs qui détermine d'autres attributs A1, … An
   - Sans DF entre attributs : Ai->Aj
3. Editer la relation (X, A1, … , An) qui est en 3FN
4. Supprimer les DFs (X->A1, … X->An) du graphe de couverture minimale C
5. Supprimer les attributs isolés de C (i.e. les attributs non-cible ni source de DF)
6. Répéter l'opération de réduction du graphe C à partir de l'étape 2, jusqu'à ce que C soit vide.
7. Ajouter une relation dont les attributs forment une clé de la relation universelle



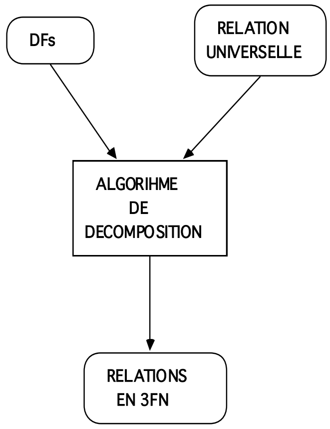



# Comment tester si R en 3NF?

- Calcul des clés (les $X^+$)
- Pour chaque X->A dans F, par la définition
  - X (sur)clé ?
  - A prime?

# *Automatisation de la décomposition*

**AUTOMATISATION DU PROCESSUS DE DECOMPOSITION EN 3NF, A PARTIR DE L'ENSEMBLE DES ATTRIBUTS ET DES DFs**






# *Encore moins de redondances : BCNF*

Définition:

une relation est en BCNF (Boyce-Codd Normal Form) ssi :

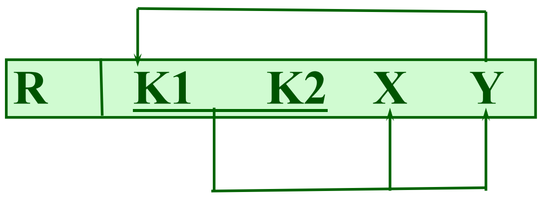
les seules dépendances fonctionnelles sont du type clé détermine attribut

X->A une DF non triviale dans $F^+$ alors :

- **X contient une clé de R**

Plus simple que 3NF, un peu plus fort, mais ne peut pas toujours être obtenue SPD !

**$X^+=\{X\}$ ou $X^+=$tout**




Une telle relation peut être décomposée en

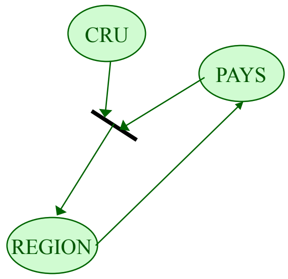
R1(K2, Y, X) et R2(Y,K1)

## *Exemple*

CRU, PAYS --> REGION

REGION--> PAYS

**R (CRU, PAYS, REGION), en 3NF**




**Décomposition BCNF :**

**R1(REGION, PAYS)**
**R2(CRU, REGION)**



## *Décomposition en BCNF*

Algorithme:

(1)Pour R(A) et X t.q. $\mathbf{X^+} \neq X \neq A$

- Décomposer en $R_1(X \cup \{A - \mathbf{X^+}\})$ et $R_2(\mathbf{X^+})$

(2) Répéter le premier pas pour $R_1$ et $R_2$

**Comment tester si R(A) en BCNF?**

**-calcul de $X^+$ pour chaque X->Y dans F,**

**-test si $X^+$=A ou $X^+$ =X**

Exemple: CSJDPQV, clé C, SD->P, J->S, JP->C

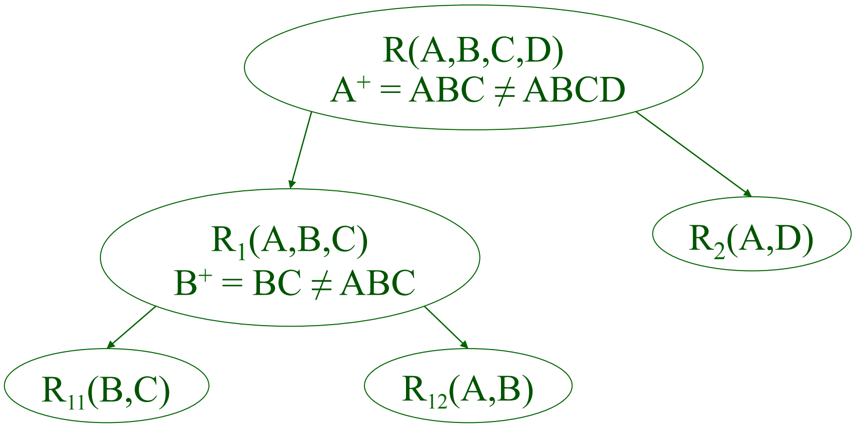
Exemple sans SPD: ABC, AB->C, C->B

On ne peut pas décomposer sans perdre AB->C

R(A,B,C,D)

A ->B
B -> C

## *Exemple*




Si on choisit d'abord $B^+$ (ou $AB^+$) ?



## *Conclusion*

- Normalisation utile
  - pour obtenir une représentation canonique des entités du monde réel
  - pour améliorer la performance de la BD
- Normalisation automatisable
  - Algos 3NF
  - outils du commerce (Power Designer, ER-Win, etc.)