INSTITUT SUPERIEUR D'INFORMATIQUE
Département Ingénierie de Développement du Logiciel

FONDEMENTS DES BASES DE DONNÉES

Mayssa Frikha Mallek
mayssa.frikha@isi.utm.tn

Année universitaire 2023-2024

Plan du cours

Normalisation

Niveau conceptuel

Modèle E-A
Modèle UML

VALIDATION

Modèle
Conceptuel Valide

Niveau logique
(Relationnel)

TRADUCTION

Schéma Relationnel

NORMALISATION

Schéma Relationnel
Normalisé

Qu’est-ce qu’une BD ‘correcte’ ?

Un ensemble de relations tel que :

- Chaque relation décrit un fait élémentaire avec les seuls attributs qui lui sont directement liés
- Il n'y a pas de redondance d’information ► génératrices de problèmes lors des MAJ
- Il n'y a pas de perte d ’information
  - Personne (nom, prénom), Adresse (no, rue, ville) Qui habite où ? Impossible de répondre !

Qu’est-ce qu’une BD ‘incorrecte’ ?

Un ensemble de relations tel que :

- Une relation n’est pas correcte si:
  - elle implique des répétitions au niveau de sa population
  - elle pose des problèmes lors des MAJ insertions / modifications / suppressions
- Les conditions pour qu'une relation soit correcte peuvent être définies formellement ► règles de normalisation
- La nécessité d’une étape d'affinement des schémas relationnels est justifiée pour deux raisons essentielles:
  1. Anomalies de mise à jour
  2. Perte d’informations

Anomalies de mise à jour

- On considère une entité unique PROPRIETAIRE contenant tous les attributs des trois relations PERSONNE, VOITURE et POSSEDE:

NV	Marque	Type	Puiss	Coul.	NSS	Nom	Prénom	Date	Prix
672RH75	Renault	RME8	8	Rouge	142032	Martin	Jacques	100298	70000
800AB64	Peugeot	P206A	7	Bleue	142032	Martin	Jacques	110699	90 000
686HK75	Citroën	BX20V	9	Verte	158037	Dupond	Pierre	200499	120000
72OCD63	Citroën	2CV8	2	Verte	158037	Dupond	Pierre	200278	5000
400W75	Renault	RCL5	4	Verte	275045	Fantas	Julie	110996	20 000
-	-	-	-	-	280037	SchiHer	Claudia	-	-
963TX63	Renault	P306B	7	Bleue	-	-	-	-	-

Anomalies de mise à jour

- Les données sont redondantes:
  - MARTIN Jacques et DUPOND Pierre apparaissent deux fois
  - Une personne apparaît autant de fois qu'elle possède de voitures,
- Ces redondances conduisent à des risques d'incohérences lors des mises à jour.
  - Si le prénom du DUPOND est modifié de Pierre à Jean, il est essentiel de mettre à jour les deux occurrences de DUPOND.
  - Sinon, il y aura un DUPOND Pierre et un DUPOND Jean dans la base de données.
- Il y a tout intérêt à éliminer ces anomalies **d'insertion, de mise à jour et de suppression** afin de faciliter la manipulation des relations.

Dépendance Fonctionnelle

- L’adresse d’un fournisseur ne dépend que du fournisseur.

  ► DÉPENDANCE FONCTIONNELLE (DF)

- Soit une table T (x, y, z)
  - Il existe une DF : $x \rightarrow y$ si et seulement si dans T à une même valeur de x correspond toujours une même valeur de y
  - Un attribut (ou groupe d’attributs) Y dépend fonctionnellement d’un attribut (ou groupe d’attributs) X, si, étant donnée une valeur de X, il lui correspond une valeur unique de Y.

Dépendance Fonctionnelle

T :

X	Y	Z
x1	y1	z1
....		
x2	y2	z2
....		

- X $\rightarrow$ Y : X détermine Y Y dépend de X
- X : source de la DF, Y : cible de la DF
- la source peut être un ensemble d'attributs :

  (nom, prénom) $\longrightarrow$ adresse

Propriétés des DFs

- Les 3 règles suivantes sont connues sous le nom d'axiomes d'Armstrong:
  - **Réflexivité:** Si $Y \subseteq X \Rightarrow X \rightarrow Y$; tout ensemble d'attributs détermine lui-même ou une partie de lui-même
  - **Augmentation:** Si $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$; si X détermine Y, les deux ensembles d'attributs peuvent être enrichis par un même troisième.
  - **Transitivité:** Si $X \rightarrow Y \; et \; Y \rightarrow Z \Rightarrow X \rightarrow Z$

Propriétés des DFs

- Les 3 règles d'inférence d'Armstrong forment un ensemble valide et complet:
  - **Validité:** étant donnée un ensemble F de dépendances vérifiées sur un schéma R, alors toute dépendance inférée de F grâce à ces propriétés est aussi vérifiée sur R.
  - **Complétude:** toute dépendance qui la conséquence d'un ensemble F, peut être obtenue par applications répétées de ces trois règles.

Propriétés des DFs

- Plusieurs autres règles se déduisent de ces axiomes de base:
  - **Union:**

    Si $X \rightarrow Y \; et \; X \rightarrow Z \Rightarrow X \rightarrow YZ$
  - **Pseudo-transitivité**

    Si $X \rightarrow Y \; et \; WY \rightarrow Z \Rightarrow WX \rightarrow Z$
  - **Décomposition**

    Si $X \rightarrow Y \; et \; Z \subseteq Y \Rightarrow X \rightarrow Z$

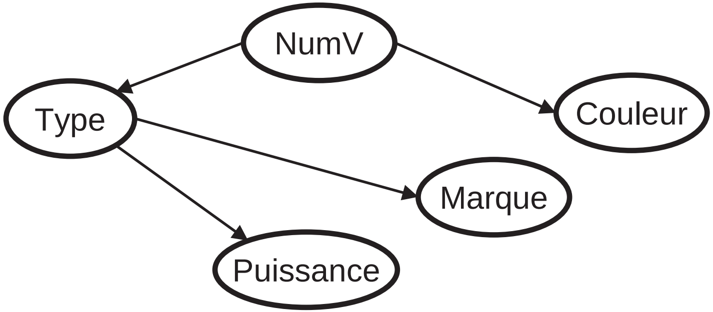
Définitions complémentaires

- La dépendance $X \rightarrow Y$ est élémentaire s'il n'existe pas $X' \subset X$ tel que $X' \rightarrow Y$
- La seule règle d'inférence qui s'applique aux dépendances fonctionnelles élémentaires est la transitivité
- La dépendance $X \rightarrow Y$ est directe s'il n'existe pas Z dans R distinct de X et Y tel que $X \rightarrow Z$ et $Z \rightarrow Y$.
- La dépendance $X \rightarrow Y$ est triviale si $Y–X$ est vide.
- Une dépendance fonctionnelle est simple si elle ne comporte qu'un seul attribut en partie droite et si elle n'est pas triviale.

Graphe des DFs

- Soit un ensemble F de dépendances fonctionnelles élémentaires.
- Si tous les attributs gauches sont uniques, il est possible de visualiser cet ensemble de DF par un graphe appelé graphe des dépendances fonctionnelles.
- Soit la relation **VOITURE** avec les DFs:

NumV → Couleur
NumV → Type
Type → Marque
Type → Puissance



Graphe des DFs

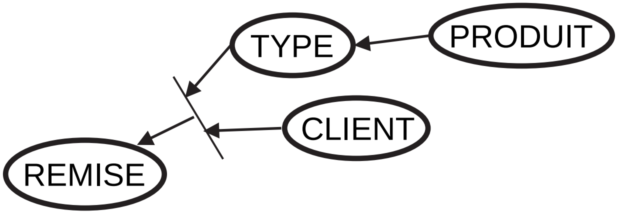
- Si une partie gauche d'une DF comporte plus d'un attribut, on introduit des arcs représentant une association de plusieurs sommets vers un sommet ► Les réseaux de Pétri
- Soit la relation:

  REDUCTION (PRODUIT, TYPE, CLIENT, REMISE)
- Avec
  - (1) Chaque produit est associé à un et un seul type.
  - (2) Les réductions sont effectuées selon le type et le client

PRODUIT	TYPE	CLIENT	REMISE
P1	A	C1	3%
P2	A	C2	5%
P3	B	C1	4%

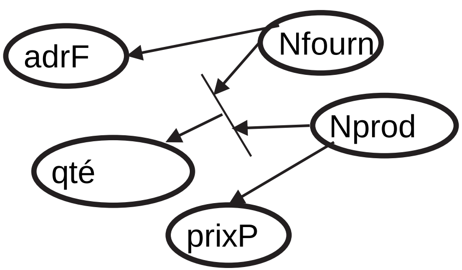
Produit → Type

Type, Client → Remise



Exemple de graphe des DF

- **Livraison (<u>Nfourn</u>, adrF, <u>Nprod</u>, prixP, qté)**
  - Nfourn → adrF
    - l'adresse d'un fournisseur ne dépend que du fournisseur
  - Nprod → prixP
    - le prix d'un produit ne dépend que du produit
  - (Nfourn, Nprod) → qté
    - la quantité livrée dépend du produit et du fournisseur



Fermeture Transitive

- À partir d’un ensemble de DF élémentaires, on peut composer par transitivité d’autres DF élémentaires.
- On aboutit à la notion de fermeture transitive d’un ensemble F des DF élémentaires considérées enrichi de toutes les DF élémentaires déduites par transitivité
- Deux ensembles sont équivalents s’ils ont la même fermeture transitive.

Fermeture Transitive

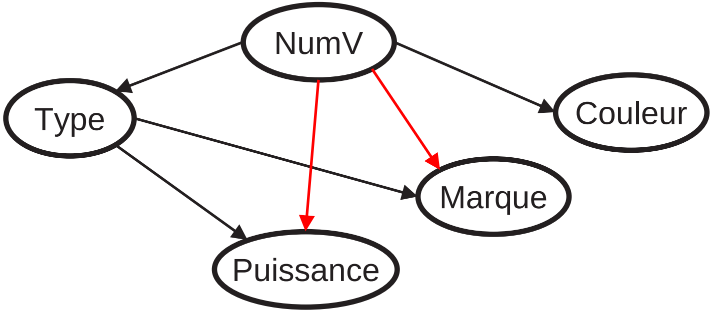
- À partir de l'ensemble de DF:

F = {NumV → Couleur, NumV → Type, Type → Marque, Type → Puissance}

- On déduit la fermeture transitive

F+ = F ∪ {NumV → Marque, NumV → Puissance}

- Le graphe correspondant F+ est:



Couverture Minimale

- Il est intéressant de déterminer un sous-ensemble minimal de DF permettant de générer toutes les autres. C'est la couverture minimale d'un ensemble de DF.
- Une couverture minimale est un ensemble de DF élémentaires associé à un ensemble d'attributs vérifiant les propriétés:
  1. Aucune dépendance dans F n'est redondante, ce qui signifie que pour toute DF f de F, F-f n'est pas équivalent à F.
  2. Toute DF élémentaire des attributs est dans la fermeture transitive de F (notée F+)

- F = {NumV → Couleur, NumV → Type, Type → Marque, Type → Puissance} est une couverture minimale pour l'ensemble des DF de **VOITURE**.

Couverture Minimale

Algorithme pour élaborer une couverture minimale F° d'un ensemble F de dépendances

```
G := F ;
Pour chaque f ϵ G faire
    Si G-{f} est équivalent à G alors
            G := G-{f} ;
    Finsi
Finpour;
F°:= G ;
```

- La couverture minimale est un élément essentiel pour composer des relations sans perte d'informations.

Déterminants et clés

- Constituant = tout sous-ensemble de l'ensemble U des attributs d'une relation R.
- Le constituant Y est totalement dépendant de X si la dépendance fonctionnelle X → Y est élémentaire.
- Déterminant = tout constituant V tel qu'il existe un constituant totalement dépendant de V.
- Clé = tout constituant X de R (A1,A2, …, An) tel que:
  - X→A1, A2, …, An
  - Il n'existe pas Y ⊂ X tel que Y →A1, A1; …, An
- Constituant clé = constituant qui appartient à l'une des clés candidates de R.
- Superclé = tout constituant qui contient une clé de R.
- Une clé est un ensemble minimal d'attributs qui détermine tous les autres.
  - NumV est une clé de la relation VOITURE, alors que (NumV,TYPE) n'est pas une clé mais une superclé.

Normalisation

- Que faire si une table n'est pas « normalisée » ?

  ⇒ ***DECOMPOSITION***

- La table doit être remplacée par un ensemble de tables (plus petites : moins d'attributs)

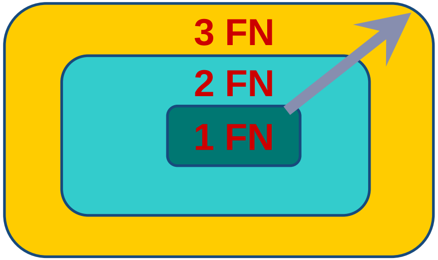
- Une relation universelle est une table unique dont le schéma est composé par l'union de tous les attributs des tables constituant la base.

Qu’est ce que la normalisation ?

- Processus de transformation d'une relation posant des problèmes lors des MAJ en des relations ne posant pas de problèmes
- La normalisation constitue un ensemble de règles introduites dans le modèle relationnel afin de garantir la cohérence de la BD.
- Elle permet d’affecter correctement les attributs dans les différentes relations ce qui permet de donner à la base une sémantique traduisant le monde réel.
- Le processus de normalisation passe par plusieurs étapes constituant les différentes formes normales qui se présentent comme des règles de construction.

Normalisation

- On mesure la qualité d'une relation par son degré de normalisation (1FN, 2FN, 3FN, BCNF, 4FN, etc.).
- Les différentes formes normales sont dépendantes et structurées.
  - En effet une relation ne peut être en 2FN que si elle est déjà en 1FN.
- Les trois premières formes normales ont pour objectif de décomposer une relation sans perdre d'informations, en se basant sur la notion de dépendance fonctionnelle



1ère forme normale : 1FN

- Une relation R est en **1FN** ssi:

  - Elle possède une clé
  - Tous ses attributs sont atomiques

Un attribut atomique → Attribut ayant une valeur simple (ne regroupe pas un ensemble de valeurs).

Employé (<u>code</u>, nom, caractéristiques)

code	nom	caractéristiques
E01	X	analyste,12/01/98
E02	Y	programmeur,13/04/01

SOLUTION →

Employé (<u>code</u>, nom, fonction, date_embauche)

code	nom	fonction	date_embauche
E01	X	analyste	12/01/98
E02	Y	programmeur	13/04/01

2ème forme normale : 2FN

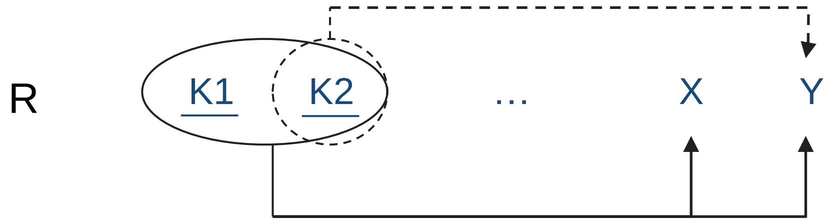
- Une relation R est en **2FN** ssi:

> - Elle est en 1FN
> - Toutes ses DF sont élémentaires

- Une DF est dite élémentaire ($A \rightarrow B$) ssi il n'existe aucun $A' \subset A$ / $A' \rightarrow B$
- Autrement: une table est en 2FN si
  - elle est en 1FN, et
  - chaque attribut qui ne fait pas partie de l'identifiant dépend de l'identifiant entier (et non sous partie d'un identifiant)

2ème forme normale : 2FN

- Le schéma typique d’une relation qui n’est pas en 2ème forme normale est :



- K1 et K2 désignent deux parties de la clé R.
- Le problème est que K2 seulement détermine Y: K2 → Y.
- Donc, Y dépend d’une partie de la clé.

2ème forme normale : 2FN

Exemple 1:

Salarié (CodeS, CodDept, nom, intituléDept, adresse)

- Cette relation est en 1FN mais pas en 2FN car les DF

(CodS, CodDept) → nom, intituléDept, adresse ne sont pas élémentaires

CodS → nom, adresse

CodDept → intituléDept

- La solution réside dans la décomposition de la relation:

R1 (codeS, nom, adresse, #codDept)

R2 (codeDept, intituléDept)

2ème forme normale : 2FN

Livraison (<u>Nfourn</u>, adrF, <u>Nprod</u>, prixP, qté)

3	Lausanne	52	65	10
22	Bienne	10	15	5
22	Bienne	25	10	12
3	Lausanne	25	10	5
3	Lausanne	10	15	20

- L'adresse du fournisseur ne dépend pas du produit
- Le prix du produit ne dépend pas du fournisseur

  $\Rightarrow$ REDONDANCES
- Anomalies de mise à jour

2ème forme normale : 2FN

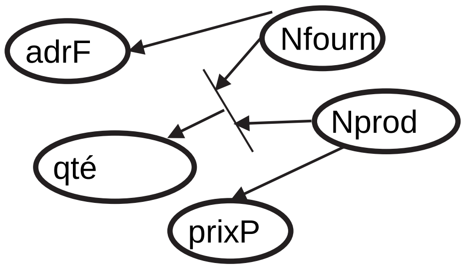
Livraison (<u>Nfourn</u>, adrF, <u>Nprod</u>, prixP, qté)

3	Lausanne	52	65	10
22	Bienne	10	15	5
22	Bienne	25	10	12
3	Lausanne	25	10	5
3	Lausanne	10	15	20

- Si un fournisseur change d'adresse et qu'un seul tuple (une seule ligne) est mis à jour ⇒ incohérence
- Si une nouvelle ligne est insérée pour un fournisseur connu, mais avec une adresse différente ⇒ incohérence

2ème forme normale : 2FN

Livraison (<u>Nfourn</u>, adrF, <u>Nprod</u>, prixP, qté)



- **Identifiant :** Nfourn + Nprod
- **Décomposition :**
  - Nprod → prixP ► Prod (<u>Nprod</u>, prixP)
  - Nfourn → adrF ► Fourn (<u>Nfourn</u>, adrF)
  - (Nfourn, Nprod) → qté ► Commader (<u>#Nfourn</u>, <u>#Nprod</u>, qté)

3ème forme normale : 3FN

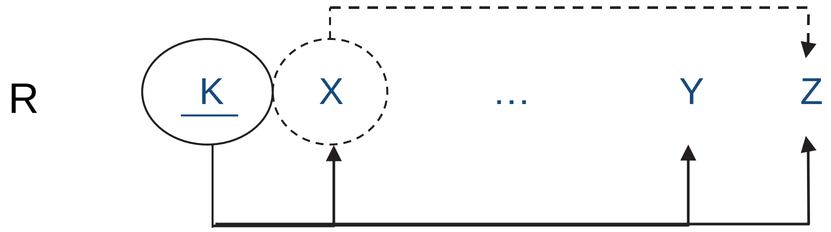
- Une relation R est en **3FN** ssi:

  - Elle est en 2FN
  - Toutes ses DF sont directes

- Une DF est dite directe ssi il n'existe aucun X de R tel que A $\longrightarrow$ X et X $\longrightarrow$ B

3ème forme normale : 3FN

- Le schéma typique d'une relation qui n'est pas en 3ème forme normale est :



- K est la clé de R.
- Le problème est que X à lui seul détermine Z : X → Z.
- Donc Z dépend d'un attribut non clé.
- Une telle relation doit être décomposée en :
  - R1(<u>K</u>,Y,#X)
  - R2(<u>X</u>,Z).

3ème forme normale : 3FN

Exemple 1:

Produit (<u>cod pdt</u>, design, cod_tva, taux_tva)

- Cette relation est en 1FN et 2 FN mais non en 3FN car :
  - la DF cod_pdt → taux_tva n'est pas directe
    - cod_pdt → code_tva
    - cod_tva → taux_tva
- La solution réside dans la décomposition de la relation:

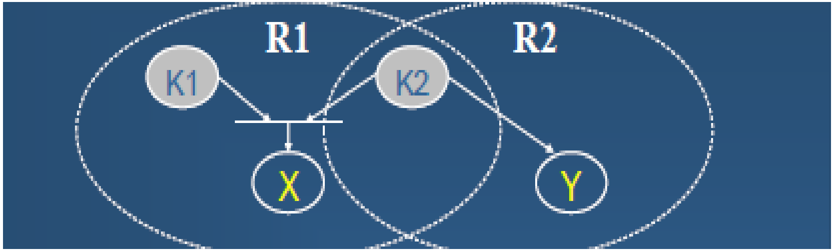
R1 (<u>code_pdt</u>, design, #code_tva)

R2 (<u>code_tva</u>, taux_tva)

Algorithme de décomposition en 3FN

- Pour toute relation, il existe au moins une décomposition en troisième forme normale préservant les DF et sans perte.
- Le but de l'algorithme [Bemstein76] est de convertir un schéma de relation qui n'est pas en 3ème FN en un ensemble de schémas en 3ème FN.
  - Le principe consiste à appliquer récursivement les règles de décomposition déjà énoncées
- Cet algorithme a comme entrées l'ensemble des attributs U ainsi que les DF D. Il consiste tout d'abord à construire une couverture minimale G des DF élémentaires.
- Ensuite, la couverture G doit être partitionnée en groupes Gi tels que les DF dans chaque Gi a le même ensemble d'attributs à gauche
- Chaque groupe produit une relation en 3ème FN.

Algorithme de décomposition en 3FN



- Les cercles correspondant aux relations décomposées R1 (K1,#K2,X) et R2(K2,Y) montrent simplement que l'union des DF de R1 et R2 est bien l'ensemble des DF: cette décomposition préserve les DF.
- Par jointure sur K2, on retrouve bien la relation initiale ► la décomposition est sans perte.

Algorithme de décomposition en 3FN

- Un algorithme de décomposition préserve le contenu si la relation initiale R peut être reconstruite par jointure à partir des relations Ri: $R1 \bowtie R2 \bowtie \ldots \bowtie Rn = R$.
- Un algorithme de décomposition préserve les dépendances si les dépendances initiales de D peuvent être reconstruites à partir des Di.
- Autrement dit, si la fermeture de D est identique à la fermeture de l'union des Di: $(D1 \cup D2 \cup \ldots \cup Dn)^+ = D^+$

Algorithme de décomposition en 3FN

- 1) Rechercher une couverture minimale G de D.
- 2) Partitionner G en groupes de dépendances Gi ayant la même partie gauche.
- 3) Fusionner les groupes Gi et Gj possédant des parties gauches Xi et Xj équivalentes (c'est-à-dire ceux pour lesquels on a $Xi \rightarrow Xj$ et $Xj \rightarrow Xi$, $Xi \in Gi$, $Xj \in Gj$).
- 4)Associer à chaque groupe Gi un schéma Ri=<Ui,Di>. Autrement dit, construire une relation Ri pour chaque Gi dont la clé est la partie gauche des DF de Di et les constituants non clés est la partie droite des DF de Di.

Algorithme de décomposition en 3FN

- 5) Si aucun des schémas précédents ne contient une clé K de R, créer un schéma supplémentaire <K,Ø>.
- Les étapes 2 et 3 permettent de répartir toutes les dépendances fonctionnelles dans des schémas en 3FN. Cet algorithme assure donc bien la préservation des dépendances.
- La préservation du contenu est garantie par l'étape 5

Algorithme de décomposition en 3FN: Exemple

- Considérons le schéma relationnel R=<U,D>.

U={A,B,C,Q,E,F}

D={A→B;B→A;A→F;A,C→Q;C→E}

A : numéro de fournisseur

B : nom de fournisseur

C : numéro de pièce

Q : quantité commandée

E : désignation de la pièce

F : adresse du fournisseur.

Algorithme de décomposition en 3FN: Exemple

- Considérons le schéma relationnel R=<U,D>.
- 1) D est une couverture minimale. G = D.
- 2) L'étape 2 fournit les groupes suivants.
  - G1={C$\rightarrow$E}
  - G2={A,C$\rightarrow$Q}
  - G3={A$\rightarrow$B;A$\rightarrow$F}
  - G4={B$\rightarrow$A}
- 3) L'étape 3 conduit à fusionner G3 et G4. On obtient un nouveau groupe G3.
  - G1={C$\rightarrow$E}
  - G2={A,C$\rightarrow$Q}
  - G3={A$\rightarrow$B;B$\rightarrow$A;A$\rightarrow$F}

Algorithme de décomposition en 3FN: Exemple

- 4) L'étape 4 génère les schémas suivants.
  - R1=<{C,E},{C→E}>
  - R2=<{A,C,Q},{A,C→Q}>
  - R3=<{A,B,F},{A→B;B→A;A→F}>
- 5) Une clé de la relation universelle est A,C. Puisque R2 contient déjà cette clé, le schéma {R1,R2,R3} préserve le contenu.

FORME NORMALE DE BOYCE-CODD

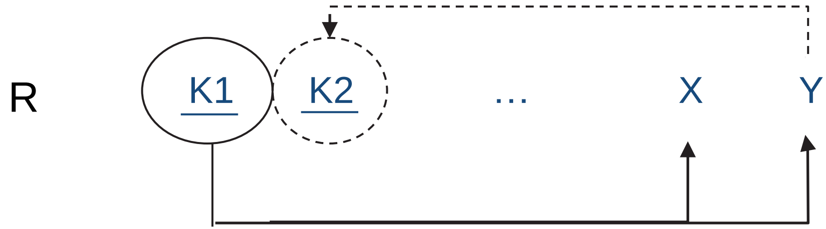
- La 2ème forme normale élimine les anomalies créées par des dépendances entre parties de clé et attributs non clé.
- La 3ème forme normale élimine les anomalies créées par des dépendances entre les attributs non clés.
- Les dépendances de parties de clés entre elles ou d'attributs non clé vers une partie de clé ne sont pas traités? La 3ème FN est donc insuffisante.
- Afin d'éliminer les redondances crées par ces dépendances, Boyce et Codd ont introduit la forme normale qui porte leur nom (en abrégé BCNF)[Codd74].

FORME NORMALE DE BOYCE-CODD

- Une relation R est en BCNF (Boyce-Codd Normal Form) si et seulement si les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé entière détermine un attribut.
- Cette définition a le mérite d'être simple: pas de dépendance autre que K→A, K étant la clé et A un attribut non clé.
- La BCFN généralise donc la 3FN aux relations avec plusieurs identifiants.
- Il a été montré que toute relation a une décomposition en BCNF qui est sans perte. Par contre, une décomposition en BCNF ne préserve en général pas les DF.

FORME NORMALE DE BOYCE-CODD

- La figure illustre le cas typique d'une relation en 3ème forme normale mais non en BCFN puisqu'un un attribut non clé détermine une partie de clé.



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

FORME NORMALE DE BOYCE-CODD

- Considérons la relation :

  Enseignement (NomEtud, Matière, Prof, note)
- Avec les dépendances:

  NomEtud,Matière→prof

  NomEtud,Matière→note

  prof→matière
- Avec un exemple d'extension:

NomEtud	Matière	Prof	Note
Ayari	Bases de données	Ben Saleh	10
Ayari	Réseaux	Ben Ammar	12
Rezgui	Bases de données	Ben Saleh	14
Rezgui	Réseaux	Ben Ammar	08

FORME NORMALE DE BOYCE-CODD

- La relation Enseignement est en 3FN mais n’est pas en FNBC
- Décomposition:
  - R1(<u>nomEtud</u>, matière, note);
  - R2(<u>prof</u>, matière);
- La DF NomEtud, Matière → prof est perdue: La relation ne préserve pas les DF.
- La décomposition est cependant sans perte.