

Plan

Bases de Données : 3^{ème} Forme Normale (3NF)

Stéphane Devismes

Université Grenoble Alpes

26 août 2020

- 1 3^{ème} Forme Normale
- 2 Exemple général
- 3 Autres formes normales

Justification

On a vu qu'on sait toujours calculer une décomposition BCNF qui **éviter les redondances** **OU** qui **préserve les dépendances**.

En général il n'existe pas de décomposition BCNF qui satisfait à la fois ces deux propriétés.

Pour obtenir une décomposition qui vérifie à la fois les deux propriétés, on affaiblit la propriété de normalisation de Boyce-Codd.

Définition

Une relation R est en **3^{ème} forme normale** (3NF) si pour tout ensemble d'attributs A on a :

- $A^+ = A$ ou
- $A^+ = S$ ou
- $A^+ = A \cup B$ où chaque élément de B fait partie d'une clé.

Propriété : Soient R une relation et \mathcal{D} une base de dépendances fonctionnelles singletons de R .

R est en 3^{ème} forme normale si et seulement si pour toute dépendance fonctionnelle singleton $A \rightarrow b$ de \mathcal{D} on a :

- A est une superclé ($A^+ = S$) ou
- b fait partie d'une clé.

Preuve de la propriété (1/2)

Soit R est une relation en 3^{ème} forme normale et \mathcal{D} une **base de dépendances fonctionnelles singletons** de R .

Soit $A \rightarrow b \in \mathcal{D}$.

Si $A^+ = A$, alors nécessairement $b \in A$ et $A \rightarrow b \notin \mathcal{D}$, contradiction.

Si $A^+ = S$, alors A est superclé de R , par définition.

Si $A^+ = A \cup B$ où chaque élément de B fait partie d'une clé, alors puisque $b \notin A$, $b \in B$, c'est-à-dire b fait partie d'une clé de R .

Preuve de la propriété (2/2)

Supposons une base de dépendances fonctionnelles singletons \mathcal{D} de R vérifiant : pour toute dépendance fonctionnelle $A \rightarrow b \in \mathcal{D}$ on a

- A est une superclé ($A^+ = S$) ou
- b fait partie d'une clé.

Soit X un ensemble d'attributs.

Il faut montrer que si $X^+ \neq X$ alors $X^+ = S$ ou pour tout $b \in X^+ - X$, b fait partie d'une clé.

Supposons que $X^+ \neq X$ et soit $b \in X^+ - X$.

D'après l'algorithme de clôture, il existe $A \rightarrow b \in \mathcal{D}$ tel que $A \subseteq X^+$.

Par hypothèse, $A^+ = S$ ou b fait partie d'une clé.

Puisque $A \subseteq X^+$, $A^+ \subseteq (X^+)^+ = X^+$, donc si $A^+ = S$, alors $X^+ = S$ aussi.

D'où, $X^+ = S$ ou pour tout $b \in X^+ - X$, b fait partie d'une clé : R est en 3NF.

Propriété

Si R est en BCNF alors R est en 3NF.

L'inverse n'est pas vrai !

Par exemple, soit $R(a, b, c)$ avec l'ensemble de dépendances fonctionnelles $\{a \rightarrow b, b \rightarrow a\}$.

R n'est pas BCNF car $a^+ = ab \neq abc$

Cependant, R a deux clés : ac et bc . Donc, les deux parties droites a et b appartiennent à des clés : R est 3NF !

Définition

Une décomposition $R = R[S_1] * \dots * R[S_n]$ est en 3^{ème} forme normale (3NF) si chaque $R[S_i]$ est en 3NF.

Algorithme de synthèse 3NF

Entrée. Une base **minimale** \mathcal{D}_{min} de R .

Sortie. $Res = \{S_1, \dots, S_p\}$ où les S_i fournissent des relations formant une décomposition de R en 3NF qui préserve les dépendances et qui évite les redondances.

-BCNF- Appliquer l'algorithme de synthèse BCNF. Soit

$Res = \{S_1, \dots, S_n\}$ le résultat obtenu.

-N- Supprimer les redondances :

Tant qu'il y a des inclusions $S_i \subseteq S_j$ avec $i \neq j$

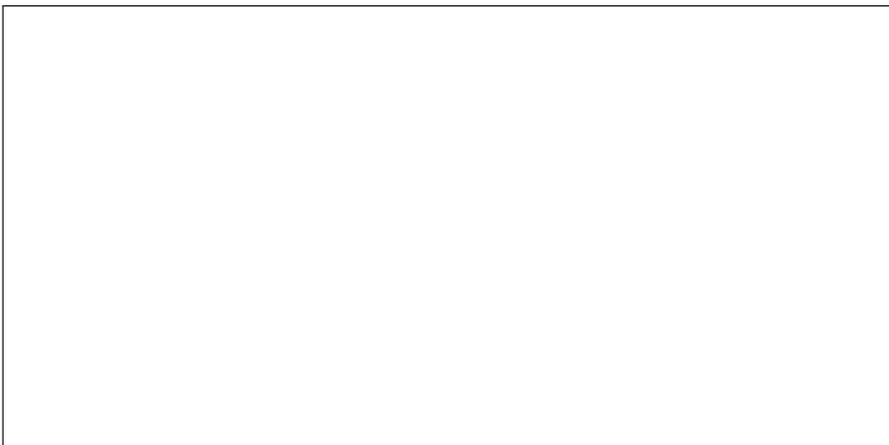
Enlever S_i de Res .

Ajouter les dépendances fonctionnelles de S_i à celles de S_j

-R- Retourner Res .

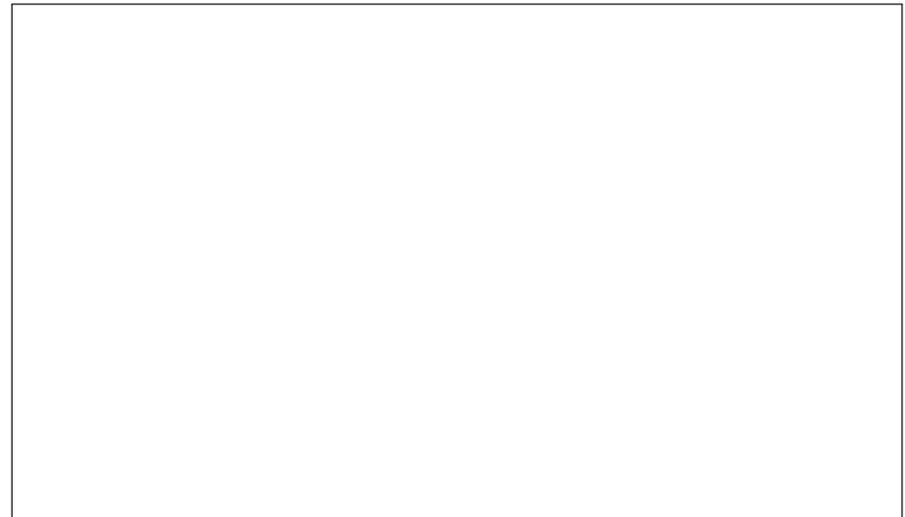
Exemples (2/2)

Soit $R(a, b)$ avec la base minimale $\{a \rightarrow b, b \rightarrow a\}$.



Exemples (1/2)

Soit $R(p, m, e)$ avec la base minimale $\{p \rightarrow m, me \rightarrow p\}$.



Preuve de l'algorithme : terminaison

-BCNF- : **l'algorithme de synthèse BCNF termine** (mais, il est exponentiel — $\leq e^{O(|\mathcal{D}_{min}| * |S|)}$ — cf. cours précédent).

-N- : **polynomial en $|Res|$ et \mathcal{D}_{min} .**

-R- : **temps constant.**

Preuve de l'algorithme : correction (1/5)

On sait déjà que l'ensemble $Res = \{S_1, \dots, S_n\}$ obtenu avant N vérifie : $R[S_1] * \dots * R[S_n]$ est une décomposition BCNF de R .

Ensuite, si on supprime un ensemble S_i de Res pendant N, cela implique qu'il existe $j \neq i$ tel que $S_j \in Res$ et $S_i \subseteq S_j$. Or, puisque $S_i \subseteq S_j$, on a $R[S_j] = R[S_i] * R[S_j]$. Donc, après chaque suppression, on a toujours le produit naturel des projections de R sur les éléments de Res est une décomposition de R .

De plus, on a pas de perte de dépendance fonctionnelle.

Enfin, par N on supprime toutes les redondances.

Preuve de l'algorithme : correction (2/5)

Il nous reste uniquement à démontrer, qu'après R, on a $\forall i \in \{1, \dots, p\}, R[S_i]$ est 3NF.

Soit Res_f la valeur de Res à la fin de l'algorithme de synthèse 3NF. Soit $S_i \in Res_f$. Soit $\mathcal{D}_{min}^{S_i} \subseteq \mathcal{D}_{min}$ le sous-ensemble des dépendances fonctionnelles (singletons) de \mathcal{D}_{min} attachées à $R[S_i]$ à la fin de l'algorithme.

On a deux cas : S_i a été ajouté à Res

- soit lors de l'étape F,
- soit lors de l'étape U

de l'algorithme de synthèse BCNF.

Preuve de l'algorithme : correction (3/5)

Supposons que S_i a été ajouté à Res lors de l'étape F de l'algorithme de synthèse BCNF.

Alors, S_i est une clé de R : $S_i^+ = S$ relativement à \mathcal{D}_{min} et $S_i \rightarrow S$ est une conséquence de \mathcal{D}_{min} .

De plus, $S_i^+ = S_i$ relativement à $\mathcal{D}_{min}^{S_i}$: S_i est une superclé de $R[S_i]$.

Supposons, par contradiction, que S_i n'est pas une clé de $R[S_i]$: il existe $X \subsetneq S_i$ tel que $X^+ = S_i$ relativement à $\mathcal{D}_{min}^{S_i}$.

Alors $X \rightarrow S_i$ est une conséquence de $\mathcal{D}_{min}^{S_i}$ et de \mathcal{D}_{min} car $\mathcal{D}_{min}^{S_i} \subseteq \mathcal{D}_{min}$.

Par transitivité ($X \rightarrow S_i$ et $S_i \rightarrow S$), $X \rightarrow S$ est aussi une conséquence de \mathcal{D}_{min} : $X^+ = S$ relativement à \mathcal{D}_{min} , c'est-à-dire, $X \subsetneq S_i$ est superclé de R et donc S_i n'est pas une clé de R , contradiction.

Donc, S_i est une clé de $R[S_i]$. Or, $\forall A \rightarrow b \in \mathcal{D}_{min}^{S_i}$, on a $b \in S_i$. Donc, $R[S_i]$ est 3NF.

Preuve de l'algorithme : correction (4/5)

Supposons que S_i a été ajouté à Res lors de l'étape U de l'algorithme de synthèse BCNF.

Par construction, il existe $A_i \subsetneq S_i$ tel que : $\forall b \in S_i - A_i, A_i \rightarrow b \in \mathcal{D}_{min}^{S_i}$. Soit $\mathcal{D}' = \{A \rightarrow b \in \mathcal{D}_{min}^{S_i} : A = A_i\}$.

(À cause de la suppression des redondances, on peut avoir $\mathcal{D}_{min}^{S_i} \neq \mathcal{D}'$.)

Par définition, $\mathcal{D}' \neq \emptyset$; de plus, $A_i^+ = S_i$, c'est-à-dire, A_i est une superclé de $R[S_i]$.

Soit $A_i \rightarrow b \in \mathcal{D}' \subseteq \mathcal{D}_{min}^{S_i}$. Posons $A_i = a_1, \dots, a_k$ et supposons, par contradiction, que A_i n'est pas une clé de $R[S_i]$. Il existe donc $K \subsetneq A_i$, tel que $K^+ = S_i$ relativement à $\mathcal{D}_{min}^{S_i}$. Donc $K \rightarrow S_i$ est une conséquence de $\mathcal{D}_{min}^{S_i}$, et par réflexivité et transitivité $K \rightarrow b$ est aussi une

conséquence de $\mathcal{D}_{min}^{S_i}$ et donc de \mathcal{D}_{min} . Soit $x \in \{1, \dots, k\}$ et $a_1 \dots \hat{a}_x \dots a_k$ tels que $K \subseteq \{a_1, \dots, a_k\} - \{a_x\}$.

$$\begin{array}{c} \text{(refl)} \frac{K \subseteq \{a_1, \dots, a_k\} - \{a_x\}}{a_1 \dots \hat{a}_x \dots a_k \rightarrow K} \quad K \rightarrow b \\ \text{(trans)} \frac{a_1 \dots \hat{a}_x \dots a_k \rightarrow K \quad K \rightarrow b}{a_1 \dots \hat{a}_x \dots a_k \rightarrow b} \end{array}$$

Donc, $a_1 \dots \hat{a}_x \dots a_k \rightarrow b$ est une conséquence de \mathcal{D}_{min} : \mathcal{D}_{min} n'est pas une base minimale de R , contradiction. Donc, A_i est une clé de $R[S_i]$.

Preuve de l'algorithme : correction (5/5)

Enfin, montrons que $R[S_i]$ est en 3NF.

Supposons, par contradiction, qu'il existe $A \rightarrow b \in \mathcal{D}_{min}^{S_i}$ telle que $A^+ \neq S_i$ relativement à $\mathcal{D}_{min}^{S_i}$ et b n'appartient à aucune clé de $R[S_i]$.

En particulier, $b \notin A_i$, donc par construction $A_i \rightarrow b \in \mathcal{D}'$.

Tout d'abord, $A_i^+ = S_i - \{b\}$ relativement à $\mathcal{D}' - \{A_i \rightarrow b\}$, donc $A_i \rightarrow S_i - \{b\}$ est une conséquence de $\mathcal{D}' - \{A_i \rightarrow b\}$.

De plus, $A \subseteq S_i$ et $b \notin A$ (car $A \rightarrow b$ est une dépendance singleton) impliquent $A \subseteq S_i - \{b\}$ et par réflexivité $S_i - \{b\} \rightarrow A$.

Par

$$(trans) \frac{A_i \rightarrow S_i - \{b\} \quad S_i - \{b\} \rightarrow A}{(trans) \frac{A_i \rightarrow A \quad A \rightarrow b}{A_i \rightarrow b}}$$

Donc $A_i \rightarrow b \in \mathcal{D}_{min}$ est une conséquence de $\mathcal{D}_{min} - \{A_i \rightarrow b\}$, ce qui implique que \mathcal{D}_{min} n'est pas une base minimale de R , contradiction.

Donc, $\forall A \rightarrow b \in \mathcal{D}_{min}^{S_i}$ telle que $A^+ \neq S_i$, b qui fait partie d'une clé de $R[S_i]$. D'où, $R[S_i]$ est 3NF.

Relations

En algèbre relationnelle, on obtient :

Ecrivains (nom, prenom)

Oeuvres (isbn, titre)

Auteurs (isbn, nom, prenom)

Membres (numM, nom, prenom)

Salles (numS, nomB)

Livres (numL, isbn, numM, dateE, numS, nomB)

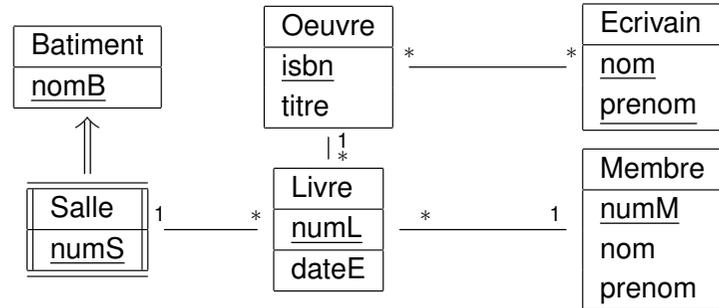
avec des contraintes d'intégrité à écrire ...

Il n'y a pas de contrainte de dépendance fonctionnelle autre que celles donnés par les clés donc c'est **une décomposition BCNF** !

Diagramme conceptuel

Reprenons une variante de l'exemple de la bibliothèque, vu lors du chapitre « Conception ».

La conception avait mené à un diagramme de ce genre (en simplifiant les cardinalités : 1 pour « mono » et * pour « multi »).



Normalisation (1/2)

Maintenant, oublions tout cela et partons de l'ensemble des attributs (les noms sont simplifiés) et d'une base de dépendances entre ces attributs.

Attributs (il y en a 11) :

nom	prenom	isbn	titre	numM	nom	prenom	numS	nomB	numL	dateE
n	p	i	t	m	n'	p'	s	b	l	e

Base de dépendances (ici 9 dépendances singletons) :

$(d1) i \rightarrow t, (d2) m \rightarrow n'p', (d3) l \rightarrow itmesb.$

On peut noter que $l \rightarrow t$ est inutile, on va vérifier que les algorithmes s'en rendent compte.

On peut noter que n et p n'apparaissent pas dans cette base.

Normalisation (2/2)

En appliquant l'algorithme récursif BCNF avec successivement $A = i$, $A = m$, $A = l$, on obtient la décomposition BCNF (sans redondance et sans perte de dépendance) :

$$R = R[\underline{i}, t] * R[\underline{m}, n', p'] * R[\underline{l}, i, m, e, s, b] * R[\underline{l}, n, p]$$

En appliquant l'algorithme de synthèse BCNF on obtient la même décomposition.

En effet la base donnée est minimale (c'est facile à vérifier), d'où les 3 premières relations dans la décomposition.

Mais de plus la seule clé de R est lnp (c'est facile à vérifier), d'où la 4^{ème} relation.

Puisqu'il n'y a pas de redondance dans cette décomposition, l'algorithme de synthèse 3NF ne fait rien de plus.

Discussion

Par définition, la BCNF est « optimale » au sens où elle n'a que les dépendances fonctionnelles dues aux clés.

On a vu deux façons de construire une décomposition BCNF, mais la première ne préserve pas les dépendances et la seconde n'évite pas les redondances.

On a vu un exemple où on trouve une décomposition BCNF avec ces deux propriétés, mais elle n'est pas complètement satisfaisante.

La 3NF autorise certaines dépendances fonctionnelles autres que les clés.

Son avantage est qu'on peut construire une décomposition 3NF qui à la fois préserve les dépendances et évite les redondances.

Malgré cela la 3NF n'est pas toujours satisfaisante non plus.

Comparaison

Relations calculées

O (i, t)
M (m, n', p')
L (l, i, m, e, s, b)
A (l, n, p)

Relations initiales

Ecrivains (nom, prenom)
Oeuvres (isbn, titre)
Auteurs (isbn, nom, prenom)
Membres (numM, nom, prenom)
Salles (numS, nomB)
Livres (numL, isbn, numM, dateE, numS, nomB)

- Les relations Oeuvres, Membres, Livres, correspondent exactement aux relations O, M, L.
- La relation Salles a « disparu », et en effet elle était redondante par rapport à Livres.
- La relation Ecrivains a « disparu », et en effet elle était redondante par rapport à Auteurs.
- La relation Auteurs (i, n, p) ne correspond pas exactement à la relation A(l, n, p) : on n'a pas su traduire par des dépendances fonctionnelles le fait que des livres correspondant à la même œuvre doivent avoir les mêmes auteurs.

Hiérarchie

Il existe toute une hiérarchie de formes normales, de la plus faible vers la plus forte :

$$1NF \Leftarrow 2NF \Leftarrow 3NF \Leftarrow BCNF \Leftarrow 4NF \Leftarrow 5NF \Leftarrow 6NF$$

1NF et 2NF

1NF : la première forme normale impose que les attributs ont un type atomique.

2NF : il s'agit d'une version affaiblie de la 3NF.

Une relation R est en **2^{ème} forme normale** (2NF) si pour tout ensemble d'attributs A on a :

- $A^+ = A$ ou
- $A^+ = S$ ou
- $A^+ = A \cup B$ où pour chaque élément b de B :
 - soit b fait partie d'une clé (comme pour la 3NF)
 - soit b ne fait pas partie d'une clé, mais dans ce cas A n'est pas strictement inclus dans une clé.

Conclusion

Il n'y a pas une forme normale « meilleure » que les autres, ni une décomposition « meilleure » que les autres.

En particulier, il n'est pas vrai que la meilleure forme normale est la plus forte.

En pratique, la 3NF et la BCNF sont souvent un bon compromis.

Noter que la normalisation ne tient pas compte des valeurs absentes, cela peut poser problème.

Lorsqu'on conçoit une base de données en passant par un diagramme de conception « raisonnable », on obtient « souvent » des relations au moins en 3NF.

4NF et 5NF

Bien que la BCNF n'ait que les dépendances fonctionnelles dues aux clés, qui sont « inévitables », on a défini des formes normales plus fines relativement d'autres critères (dépendances multivaluées) ...