DÉPENDANCES FONCTIONNELLES

code	titre	duree	annee	nom	adresse
A	Aliens	137	1982	Clean Kill Movies	45, walker street, houston
SF	Aliens	137	1982	Clean Kill Movies	45, walker street, houston
SF	Blade Runner	117	1982	SF Movies	13, Champs Elysee, Paris
CD	Casablanca	102	1942	Classique Film	2, Place Kleber, 67000 Strasb
W	Dances with Wolves	180	1990	Constance Film	Gumpendorferstrasse 17, A-106

- Un film est stocké plusieurs fois s'il possède plusieurs catégories
- l'adresse d'un distributeur est stockée plusieurs fois s'il distribue plusieurs films
- ….

La mise à jour d'une valeur d'un attribut doit être propagée à plusieurs endroits.

Dépendance Fonctionnelle DF

L'identification des DF est fondamentale pour éliminer les redondances dans une relation.

Les DF sont associées au schéma d'une relation (et non à une instance particulière).

Dans une relation, certains attributs peuvent en « déterminer » d'autres, i.e., il n'y a pas deux tuples ayant les mêmes valeurs pour un premier ensemble d'attributs sans avoir également les mêmes valeurs pour un deuxième ensemble.

Dépendance fonctionnelle

Soient r une instance de la relation R et
X et Y deux sous-ensembles d'attributs de R.

On dit que r satisfait la DF X $\rightarrow$ Y

ssi $\forall\, t1, t2 \in r$ (t1.X = t2.X $\Rightarrow$ t1.Y = t2.Y)

Dépendance fonctionnelle triviale

Une DF X $\rightarrow$ Y est triviale si Y $\subset$ X

Une DF triviale ne nous apprend rien sur les informations stockées.

Dépendance fonctionnelle élémentaire

X $\rightarrow$ Y est une DF élémentaire ssi

- $X \cap Y = \emptyset$
- Pour tout sous-ensemble strict Z de X, on n'a pas Z $\rightarrow$ Y

Dépendance fonctionnelle et clé candidate

Soient r une instance de la relation R et
X et Y deux sous-ensembles d'attributs de R.

Si r satisfait la DFE $X \rightarrow Y$, avec X U Y = Attr(R),
alors X est une clé candidate.

Déterminer les DF permet de déterminer les clés candidates d'une relation.

Théorème de Heath

Si une relation r satisfait la DFE $X \rightarrow Y$
alors r = $\prod_{XUY}(r) \bowtie \prod_{Z}(r)$, avec Attr(r) - Y = Z

(les deux relations ont en commun les attributs de X)

Déterminer les DF permet de décomposer l'information dans une BD de manière normalisée pour éviter les redondances dans une relation.

Dépendances fonctionnelles déduites

ENREG={NumE, Pays, NomM, Classe, Date, IdDep}

Si on connaît les dépendances fonctionnelles suivantes :
- df1 : NumE,Pays $\rightarrow$ NomM,Date
- df2 : NumE,Pays $\rightarrow$ Classe,IdDep
- df3 : NomM,Pays,Classe $\rightarrow$ NumE

alors on peut en déduire :
- df4 : NumE,Pays $\rightarrow$ NomM,Date,Classe,IdDep
- df5 : NomM,Pays,Classe $\rightarrow$ NumE,Date,IdDep

Implication et équivalence entre DF

Soient DF et DF' deux ensembles de DF définies sur un schéma de relation R.
DF implique DF' ssi pour toute instance r de la relation R, r satisfait DF $\Rightarrow$ r satisfait DF'

Exemple : $\{A \rightarrow B,\ A \rightarrow C\}$ implique $\{A \rightarrow BC\}$

DF et DF' sont équivalents s'ils s'impliquent mutuellement.

Règles d'implication entre DF

Réflexivité	$\emptyset \Rightarrow X \rightarrow X$
Augmentation	$X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
Transitivité	$X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$
Addition	$X \rightarrow Y, X \rightarrow Z \Rightarrow X \rightarrow YZ$
Projection	$X \rightarrow YZ \Rightarrow X \rightarrow Y, X \rightarrow Z$
Pseudo transitivité	$X \rightarrow Y, YZ \rightarrow W \Rightarrow XZ \rightarrow W$

Clé candidate d'une relation

{A1,…,An} est une clé candidate d'une relation r ssi
- {A1,…,An} déterminent fonctionnellement tous les autres attributs de la relation r,
- aucun sous-ensemble strict de {A1,…,An} ne détermine fonctionnellement tous les autres attributs de r (la clé doit être minimale).

Clés candidates d'une relation

On parle de clé *candidate*, car il peut y en avoir plusieurs.

En SQL une seule clé candidate sera déclarée comme clé primaire.

Mais pour normaliser, il est important de considérer toutes les clés candidates.

Superclé d'une relation

ensemble d'attributs qui contient une clé candidate

SC $\rightarrow$ E avec SC U E = Attr(R)

Attribut primaire, non primaire

Un attribut A est primaire dans R ssi
A appartient à au moins une clé candidate de R.

Un attribut A est non primaire dans R ssi
A n'est élément d'aucune clé candidate de R.

Fermeture d'un ensemble d'attributs

Soient {A1,…,An} un ensemble d'attributs et DF un ensemble de DF.

La fermeture de {A1,…,An} par DF notée {A1,…,An}+ est l'ensemble d'attributs B *maximal* tel que {A1,…,An} $\rightarrow$ B

Exemple:

DF={A$\rightarrow$ D; A$\rightarrow$ E; E$\rightarrow$ C}

{A}+ = {A,D,E,C}

Calcul d'une fermeture

Algorithme de saturation

1. X = {A1,…,An}
2. Rechercher une DF B $\rightarrow$ C t.q. B $\subset$ X et B $\not\subset$ C.
   X = X $\cup$ C
3. Répéter l'étape 2 jusqu'au point fixe de X.
   (plus rien ne peut être ajouté à X)
   X = {A1,…,An}+

Calcul d'une fermeture

Exemple

DF = {A $\rightarrow$ D; AB $\rightarrow$ E; BI $\rightarrow$ E; CD $\rightarrow$ I; E $\rightarrow$ C}

Calcul de {A,E}+ :

- initialisation: X = {A,E}
- A $\rightarrow$ D permet d'ajouter D : X = {A,E,D}
- E $\rightarrow$ C permet d'ajouter C : X = {A,E,D,C}
- CD $\rightarrow$ I permet d'ajouter I : X = {A,E,D,C,I}
- il n'est plus possible d'augmenter X, donc {A,E}+ = {A,E,D,C,I}

Fermeture et clé

{A1,…,An} est une superclé de R ssi
{A1,…,An}+ = Attr(R)

On peut tester si {A1,…,An} est une clé candidate d'une relation R en vérifiant que {A1,…,An} est une superclé et qu'aucun sous-ensemble strict de {A1,…,An} n'est une superclé de R.

Couverture minimale

Couverture minimale d'un ensemble de DF
- Ensemble minimum de DF permettant de reconstruire toutes les autres

Algorithme de calcul de la couverture minimale
1. Minimiser à droite
   remplacer chaque DF avec une partie droite non atomique par plusieurs DF avec un seul attribut à droite (et même partie gauche)
2. Minimiser à gauche
   parmi les DF ayant la même partie droite, supprimer celles qui ont une partie gauche incluse dans la partie gauche d'une autre
3. Eliminer les redondances
   si on peut déduire une DF des autres on la supprime

Couverture minimale

Exemple de calcul de couverture minimale
A → B ; B C → D ; A C → B D E ; D → E
Etape 1 : minimiser à droite
- A → B ; B C → D ; A C → B D E ; D → E
- A → B ; B C → D ; A C → B ; A C → D ; A C → E ; D → E

Etape 2 : minimiser à gauche
- A → B ; B C → D ; A C → B ; A C → D ; A C → E ; D → E
- A → B ; B C → D ; A C → D ; A C → E ; D → E

Etape 3 : éliminer les redondances
- A → B ; B C → D ; A C → D ; A C → E ; D → E
- A → B ; B C → D ; D → E ;

FORMES NORMALES

Formes Normales

- Décomposer les relations d'un schéma en des relations plus simples et plus « indépendantes »
- Faciliter la compréhension
- Éliminer les redondances
- Améliorer les aspects incrémentaux
- Faciliter la distribution sur des sites répartis

Première Forme Normale 1NF

Un schéma relationnel R est en 1NF ssi les valeurs de ses attributs sont atomiques i.e. chaque attribut contient une seule valeur.

Exemple de schéma non en 1NF :

Titre	Acteurs
Casablanca	Humphrey Bogart, Ingrid Bergman
Perfect World	Kevin Costner, Clint Eastwood
The Terminator	Arnold Schwarzenegger, Linda Hamilton, Michael Biehn
Die Hard	Bruce Willis

Deuxième Forme Normale 2NF

Un schéma relationnel R est en 2NF ssi
- R est en 1NF,
- Toute DF issue d'une clé, i.e. de la forme *clé candidate* $\rightarrow$ *attribut,* est élémentaire.

Autrement dit : Aucun attribut non primaire ne dépend fonctionnellement d'un sous-ensemble strict d'attributs d'une clé candidate.

Deuxième Forme Normale 2NF

Exemple de schéma relationnel en 2NF :
joueur(Personne, Taille, Poids)
clé candidate unique : Personne

joueur est en 2NF car toute DF issue de la clé est élémentaire, autrement dit tout attribut non primaire ne peut dépendre d'un sous-ensemble strict de la clé qui est un singleton.

Plus généralement, toute relation avec une clé candidate unique réduite à un attribut unique est en 2NF.

Deuxième Forme Normale 2NF

Exemple de schéma relationnel **non** en 2NF :
joueur(Personne, Sport, Taille, Poids)
Personne $\rightarrow$ Taille
Personne $\rightarrow$ Poids
clé candidate unique : {Personne, Sport}

joueur n'est pas en 2NF car :
- La DF issue de la clé: {Personne, Sport} $\rightarrow$ {Taille, Poids} n'est pas élémentaire car on a par addition des 2 DF {Personne} $\rightarrow$ {Taille, Poids}
- Autrement dit les attributs non primaires Taille et Poids dépendent de Personne uniquement, Sport n'est pas nécessaire pour les déterminer.
- Autrement dit la table joueur contient des redondances car le même triplet de valeurs (personne, taille, poids) apparaitra autant de fois que de sports pratiqués par la personne.

Deuxième Forme Normale 2NF

Décomposition de joueur

Application du théorème de Heath avec la DFE Personne $\rightarrow$ Taille, Poids

joueur = pratique ⋈ physique avec

physique = $\prod_{\text{Personne, Taille, Poids}}$ (joueur)

pratique = $\prod_{\text{Personne, Sport}}$ (joueur)

Troisième Forme Normale 3NF

Intuition

Une relation est en 3NF ssi elle est en 2NF et les attributs non primaires sont mutuellement indépendants.

Troisième Forme Normale 3NF

Un schéma relationnel R est en 3NF ssi

chaque DF X $\rightarrow$ {A} vérifie au moins une de ces conditions :
- A $\in$ X (i.e. DF triviale),
- X est une supercl é,
- A est primaire.

Notez que pour utiliser cette définition, il faut se ramener à un ensemble de DF qui ont un seul attribut à droite (e.g. X $\rightarrow${A,B} remplacé par X $\rightarrow$A et X $\rightarrow$B)

Troisième Forme Normale 3NF

Définition équivalente

Un schéma relationnel R est en 3NF ssi chaque DF X → A vérifie au moins une de ces conditions :

- A ⊂ X (i.e. DF triviale),
- X est une superclé,
- Tous les attributs de A-X sont primaires.

Notez bien qu'ici A est un *ensemble* d'attributs.

Troisième Forme Normale 3NF

Exemple de relation en 2NF mais pas en 3NF

voiture(<u>NVH</u>, TYPE, MARQUE, PUISS, COULEUR)

clé candidate unique NVH

TYPE → MARQUE, PUISS

voiture est en 2NF car clé candidate unique singleton.

voiture n'est pas en 3NF car TYPE → MARQUE, PUISS :

- Cette DF est non triviale
- {TYPE} n'est pas une superclé
- MARQUE et PUISS ne son pas des attributs primaires.

Troisième Forme Normale 3NF

Décomposition de voiture

Application du théorème de Heath avec la DFE TYPE → MARQUE, PUISS

voiture = modele ⋈ vehicule avec

- modele(TYPE, MARQUE, PUISS)
- vehicule(NVH, TYPE, COULEUR)

Troisième Forme Normale 3NF

3NF n'élimine pas toutes les redondances

Exemple :

vin(<u>CRU, PAYS</u>, REGION)

DF1: CRU,PAYS → REGION

DF2: REGION → PAYS

CRU	PAYS	REGION
Chenas	France	Beaujolais
Julienas	France	Beaujolais
Chablis	France	Bourgogne
Chablis	USA	Californie

vin est en 3NF car les DF vérifient bien les conditions :

- DF1 clé candidate
- DF2 : PAYS est un attribut primaire

Il reste cependant des redondances (couples (région, pays))

Forme Normale de Boyce-Codd BCNF

Intuition

Une relation R est sous BCNF ssi chacun de ses attributs ne dépend fonctionnellement que des clés entières.

Forme Normale de Boyce-Codd BCNF

Un schéma relationnel R est en BCNF ssi chaque DF X → {A} vérifie au moins une de ces conditions :

- A ∈ X (i.e. DF triviale),
- X est une superclé.

3NF et BCNF

Toute relation en BCNF est en 3NF

Les conditions de BCNF font partie des conditions de 3NF :

- A ∈ X (i.e. DF triviale),
- X est une superclé,
- A est primaire.

Toute relation en 3NF avec une unique clé est en BCNF

Car alors une DF vérifiant la 3ème condition (A primaire) vérifie aussi la 1ère (DF triviale).

Donc pour qu'une relation soit en 3NF et pas en BCNF il faut qu'elle ait au moins deux clés candidates.

Forme Normale de Boyce-Codd BCNF

Exemple de relation en 3NF mais pas en BCNF
vin(CRU, PAYS, REGION)
DF1 : CRU,PAYS → REGION (clé candidate)
DF2 : REGION → PAYS

vin n'est pas en BCNF car DF2 ne vérifie aucune des conditions

- REGION → PAYS non triviale,
- REGION n'est pas une superclé.

vin a une seconde clé candidate
REGION,CRU → PAYS, DF non élémentaire

Décomposition de vin selon REGION → PAYS
vin = regions(REGION, PAYS) ⋈ crus(CRU, REGION)

Forme Normale de Boyce-Codd BCNF

Il n'est pas toujours possible de décomposer un schéma en un schéma équivalent composé de relations en BCNF sans perte d'information sur les dépendances fonctionnelles.

Exemple :
Décomposition de vin selon REGION → PAYS
vin = regions(REGION, PAYS) ⋈ crus(CRU, REGION)
On perd la DF CRU, PAYS → REGION

Normalisation

Objectif :

Décomposer une relation non normalisée en sous-relations normalisées, si possible sans perte

- la jointure naturelle des sous-relations est la relation de départ
- on préfèrerait ne pas perdre les dépendances fonctionnelles

Normalisation en 3NF

Rappel

Une relation R est en 3NF ssi chaque DF {X1,….Xn} → A vérifie au moins l'une des conditions suivantes :

- Il existe i tel que A = Xi (DF triviale)
- {X1,…,Xn} contient une clé candidate (superclé)
- A appartient à au moins une clé candidate (primaire)

Normalisation en 3NF

Algorithme

Soit R(A1,…,An) satisfaisant l'ensemble de DF S

1. Calculer S_{min} la couverture minimale de S
2. Regrouper toutes les DF de S_{min} ayant la même partie gauche en une seule DF (addition)
3. Créer une relation par DF dont les attributs sont ceux de la DF
4. Si aucune de ces relations ne contient une des clés candidates de R, ajouter une dernière relation dont les attributs sont ceux d'une clé candidate de R

Normalisation en 3NF

Algorithme

L'étape 4 assure que la jointure des tables produites est bien égale à la table initiale.

Il suffit qu'une des clés candidates soit incluse entièrement dans une des tables du résultat pour que ce soit le cas.

L'algorithme de normalisation en 3NF est sans perte d'information, sans perte de DF.

Normalisation en 3NF

Exemple

R(Cours, Prof, Heure, Salle, Eleve, Note)

Cours → Prof

Heure, Salle → Cours

Heure, Prof → Salle

Cours, Eleve → Note

Heure, Eleve → Salle

clé candidate unique : {Eleve, Heure}

Normalisation en 3NF

Exemple (suite)

Algorithme

1. Les 5 DF forment une couverture minimale
2. Il n'y a pas de DF à regrouper
3. On décompose la relation en 5 relations issues chacune
4. Pas besoin d'une relation supplémentaire car la 5ème table contient les attributs de l'unique clé candidate

Résultat

R = R1(Cours,Prof) ⋈ R2(Heure,Salle,Cours) ⋈ R3(Heure,Prof,Salle) ⋈ R4(Cours,Eleve,Note) ⋈ R5(Heure,Eleve,Salle)

Normalisation en 3NF

Exemple 2

R(A,B,C)

AB → C et B → C

Une unique clé candidate : AB

Algorithme

1. Couverture minimale des DF: B → C
2. Il n'y a pas de DF à regrouper
3. On crée R1(B,C) selon B → C
4. On ajoute R2(A,B) car R1 ne contient pas la clé candidate AB.

Résultat

R = R1(B,C) ⋈ R2(A,B)

Normalisation en BCNF

Rappel

Une relation R est en BCNF ssi chaque DF {X1,….Xn} → A vérifie au moins l'une des conditions :

- Il existe i tel que A = Xi (DF triviale)
- {X1,….Xn} contient une clé candidate (superclé)

Normalisation en BCNF

Algorithme

D←{R}

Tant qu'il existe Ri dans D qui n'est pas en BCNF

- Trouver X → Y une DF non triviale sur Ri qui viole la BCNF
- Décomposer Ri en Ri1(X U Y) et Ri2(Attr(Ri)-Y)
- D←D-{Ri} ∪ {Ri1, Ri2}

Décomposition avec possible perte de DF

Plusieurs résultats possibles selon l'ordre choisi

Normalisation en BCNF

Algorithme avec perte d'information possible

- Comme on décompose toujours selon une DF, la jointure des nouvelles relations est toujours égale à la relation initiale.
- Mais il est possible qu'il n'y ait plus de relation pour « porter » une DF, c'est-à-dire plus de relation comportant tous les attributs de la DF.

Normalisation en BCNF

Exemple
R(A,B,C,D,E)
A → B ; A → C ; CD → E ; B → D
Clé A

R non en BCNF (pas en 3NF en fait)
car ni CD → E ni B → D ne satisfont les conditions : non triviales et ni CD ni B ne sont des superclés.

Exemple de normalisation en BCNF

Exemple (suite)
Décomposition de R(A,B,C,D,E) selon B → D
R = R1(B,D) ⋈ R2(A,B,C,E)

R1 : B → D ; clé B ; en BCNF
R2 : A → B ; A → C ; B → D ; clé unique A ; non en BCNF
car B → D ne satisfait pas les conditions

Exemple de décomposition en BCNF

Exemple (suite)
Décomposition de R2(A,B,C,E) selon BC → E :
R2 = R21(B,C,E) ⋈ R22(A,B,C)

R21 : B → D ; clé BC (issue de A → B et A → C) - en BCNF
R22 : A → B ; A → C ; clé A - en BCNF

Finalement
R = R1(B,D) ⋈ R21(B,C,E) ⋈ R22(A,B,C)
A → B ; A → C ; CD → E ; B → D
CD→ E est perdue

Exemple de normalisation en BCNF

Exemple (suite)
Autre décomposition de R en considérant d'abord CD → E
R = R'1(C,D,E) ⋈ R'2(A,B,C,D)

R'1 : CD → E ; clé CD – en BCNF
R'2 : A → B ; A → C ; B → D ; clé A – non en BCNF (car B → D)

Décomposition de R'2 selon B → D :
R'2 = R'21(BD) ⋈ R'22(ABC)

R'21 et R'22 sont en BCNF

Finalement R = R'1(C,D,E) ⋈ R'21(BD) ⋈ R'22(ABC)
A → B ; A → C ; CD → E ; B → D
Cette décomposition ne perd aucune DF.
C'est aussi celle qu'aurait donné l'algo de normalisation en 3NF.