

Annexe 3

Modèle relationnel et normalisation

Cette annexe au chapitre 3 propose un jeu d'exercices, dont la plupart sont accompagnés d'une solution ou de conseils de résolution. Elle complète en outre l'étude de l'algèbre relationnelle par la description de l'opérateur de **division relationnelle**, exprimant le quantificateur logique *pour tout*. On y décrit enfin une nouvelle forme de dépendance, généralisant la dépendance fonctionnelle : la **dépendance multivaluée**. Celle-ci permet de définir la **quatrième forme normale**, généralisant la forme normale de Boyce-Codd.

A3.1 L'OPÉRATEUR ALGÈBRIQUE DE DIVISION

<à rédiger>

A3.2 LES DÉPENDANCES MULTIVALUÉES ET LA 4^e FORME NORMALE

<à rédiger>

A3.3 LES RELATIONS D'ARMSTRONG

<à rédiger>

A3.4 LES RELATIONS NON EN 1^{re} FORME NORMALE

<à rédiger>

A3.5 EXERCICES DU CHAPITRE 3

A3.1 Décomposer si nécessaire la relation ACHAT.

ACHAT(NCOM, NPRO, PRIX)
 NCOM \longrightarrow NPRO
 NPRO \longrightarrow PRIX

Solution

L'identifiant de ACHAT est {NCOM}. La DF NPRO \longrightarrow PRIX est donc anormale. Par décomposition selon cette DF, on obtient le schéma relationnel normalisé :

ACHAT(NCOM, NPRO); PRODUIT(NPRO, PRIX);
 ACHAT[NPRO] \subseteq PRODUIT[NPRO]

A3.2 Décomposer si nécessaire la relation COMMANDE.

COMMANDE(NCOM, NCLI, NOM, DATE, NPRO, LIBELLE)
 NCOM \longrightarrow NCLI, DATE, NPRO
 NCLI \longrightarrow NOM
 NPRO \longrightarrow LIBELLE

Solution

L'identifiant de COMMANDE est {NCOM}. Les DF NCLI \longrightarrow NOM et NPRO \longrightarrow LIBELLE sont donc anormales. Par décomposition selon chacune de ces DF, on obtient le schéma relationnel normalisé :

COMMANDE(NCOM, NCLI, DATE, NPRO);
 CLIENT(NCLI, NOM); PRODUIT(NPRO, LIBELLE);
 COMMANDE[NCLI] \subseteq CLIENT[NCLI]
 COMMANDE[NPRO] \subseteq PRODUIT[NPRO]

A3.3 Décomposer si nécessaire la relation ACHAT2.

ACHAT2(CLI, PRO, MAG, PRIX)
 PRO, MAG \longrightarrow PRIX

Solution

L'identifiant de ACHAT2 est {CLI, PRO, MAG}. La DF PRO, MAG \longrightarrow PRIX est donc anormale. On obtient par décomposition :

ACHAT2(CLI, PRO, MAG); TARIF(PRO, MAG, PRIX);
 ACHAT2[PRO, MAG] \subseteq TARIF[PRO, MAG]

A3.4 Décomposer si nécessaire la relation ACHAT3.

ACHAT3(CLI, PRO, MAG, PRIX)
 CLI, PRO, MAG \longrightarrow PRIX

Solution

L'identifiant de la relation ACHAT3 est {CLI, PRO, MAG}. Celle-ci est donc normalisée.

- A3.5 Décomposer si nécessaire la relation ECRIT (POSITION indique la position de l'auteur dans la liste des auteurs).

ECRIT(AUTEUR, OUVRAGE, POSITION)
 AUTEUR, OUVRAGE \longrightarrow POSITION
 OUVRAGE, POSITION \longrightarrow AUTEUR

Solution

Le graphe ADF comporte un circuit. Les identifiants de la relation ECRIT sont {AUTEUR, OUVRAGE} et {OUVRAGE, RANG}. Celle-ci est normalisée.

- A3.6 Calculer les identifiants de la relation CINE. Décomposer cette relation si nécessaire.

CINE(FILM, VILLE, SALLE, DISTRIBUTEUR, DELEGUE)
 SALLE \longrightarrow VILLE
 FILM, VILLE \longrightarrow SALLE, DISTRIBUTEUR
 DISTRIBUTEUR \longrightarrow DELEGUE

Solution

Le graphe ADF comporte un circuit. Les identifiants sont {FILM, VILLE} et {SALLE, FILM}. Les deux DF suivantes sont donc anormales : SALLE \longrightarrow VILLE et DISTRIBUTEUR \longrightarrow DELEGUE. Cette dernière étant externe, elle permet une première décomposition :

CINE(FILM, VILLE, SALLE, DISTRIBUTEUR);
 DIS(DISTRIBUTEUR, DELEGUE);
 CINE[DISTRIBUTEUR] \subseteq DIS[DISTRIBUTEUR]
 SALLE \longrightarrow VILLE
 FILM, VILLE \longrightarrow DISTRIBUTEUR

La DF FILM, VILLE \longrightarrow DISTRIBUTEUR, non anormale, est externe et ne fait pas partie du noyau irréductible. Elle peut donc faire l'objet d'une décomposition :

CINE(FILM, VILLE, SALLE);
 DISTR(FILM, VILLE, DISTRIBUTEUR);
 DIS_DEL(DISTRIBUTEUR, DELEGUE);
 CINE[FILM, VILLE] \subseteq DISTR[FILM, VILLE]
 DISTR[DISTRIBUTEUR] \subseteq DIS_DEL[DISTRIBUTEUR]
 SALLE \longrightarrow VILLE

Le noyau résiduel {FILM, VILLE, SALLE} est irréductible et non normalisé. Selon le canevas A17.3.8.5, la dernière relation CINE peut être remplacée par un des trois schémas ci-dessous :

1. CINE(FILM, VILLE, SALLE); SALLE \longrightarrow VILLE
2. CINE(FILM, SALLE); LOC(SALLE, VILLE);

CINE[SALLE] = LOC[SALLE]
 CINE*LOC: FILM, VILLE \longrightarrow SALLE

3. CINE(FILM, VILLE, SALLE); LOC(SALLE, VILLE);
 CINE[SALLE, VILLE] = LOC[SALLE, VILLE]

A3.7 Cet exercice est une extension de l'énoncé 3.7 de l'ouvrage (*Démontrer que la règle de pseudo-transitivité est dérivable des autres*).

La version populaire des règles d'Armstrong en comporte six. En réalité, trois d'entre elles (*réflexivité, augmentation, transitivité*) sont suffisantes car elle permettent de dériver les trois autres (*additivité, décomposabilité, pseudo-transitivité*). Démontrer cette dérivation.

Solution

Rappel des trois règles de base (sous une forme un peu plus générale que dans l'ouvrage)

réflexivité : $L \subseteq K \Rightarrow K \longrightarrow L$

augmentation : $K \longrightarrow L \Rightarrow KM \longrightarrow LM$

transitivité : $K \longrightarrow L \wedge L \longrightarrow M \Rightarrow K \longrightarrow M$

Démonstrations

additivité : $K \longrightarrow L \wedge M \longrightarrow N \Rightarrow KM \longrightarrow LN$

1. Par augmentation, $K \longrightarrow L$ donne $KM \longrightarrow LM$
2. Par augmentation, $M \longrightarrow N$ donne $LM \longrightarrow LN$
3. Par transitivité, $KM \longrightarrow LM$ et $LM \longrightarrow LN$ donnent $KM \longrightarrow LN$

Application intéressante : $K \longrightarrow L \wedge K \longrightarrow N \Rightarrow K \longrightarrow LN$

décomposabilité : $K \longrightarrow LM \Rightarrow K \longrightarrow L \wedge K \longrightarrow M$

1. Par réflexivité, on a $L \longrightarrow L$
2. Par augmentation de $L \longrightarrow L$ par M on obtient $LM \longrightarrow L$
3. Par transitivité, $K \longrightarrow LM$ et $LM \longrightarrow L$ donnent $K \longrightarrow L$

De même pour $K \longrightarrow M$

pseudo-transitivité : $K \longrightarrow L \wedge LM \longrightarrow N \Rightarrow KM \longrightarrow N$

1. Par réflexivité, on a $M \longrightarrow M$
2. Par additivité, $K \longrightarrow L$ et $M \longrightarrow M$ donnent $KM \longrightarrow LM$
3. Par transitivité, $KM \longrightarrow LM$ et $LM \longrightarrow N$ donnent $KM \longrightarrow N$

On peut aussi démontrer la version réduite de la propriété d'augmentation :

$K \longrightarrow L \Rightarrow KA \longrightarrow L$

1. Par réflexivité, $KA \longrightarrow K$
2. Par transitivité, $KA \longrightarrow K$ et $K \longrightarrow L$ donnent $KA \longrightarrow L$

A3.8 Décomposer si nécessaire la relation VENTE.

VENTE(NPRO, CLIENT, DATE, QUANTITE, ADRESSE, DELEGUE, REGION)
 NPRO, CLIENT, DATE \longrightarrow QUANTITE
 CLIENT \longrightarrow ADRESSE, DELEGUE
 DELEGUE \longrightarrow REGION

A3.9 Décomposer si nécessaire la relation PRODUIT.

PRODUIT(NPRO, DATE-INTRO, IMPORTATEUR, AGREATION)
 NPRO \longrightarrow IMPORTATEUR
 NPRO \longrightarrow DATE-INTRO
 DATE-INTRO, IMPORTATEUR \longrightarrow AGREATION

A3.10 Décomposer si nécessaire la relation VOYAGE.

VOYAGE(NUMV, NUMC, DATE, MODELE, NOM)
 NUMC \longrightarrow NOM
 NUMV \longrightarrow MODELE

A3.11 Calculer les identifiants de la relation PROJET. Décomposer cette relation si nécessaire.

PROJET(CODE, TITRE, NUM-CONTRAT, BUDGET, RESPONSABLE, UNITE)
 CODE \longrightarrow TITRE, BUDGET
 NUM-CONTRAT \longrightarrow CODE, RESPONSABLE
 TITRE \longrightarrow NUM-CONTRAT, UNITE

Solution

Le graphe ADF comporte un circuit comprenant les attributs {CODE, NUM-CONTRAT, TITRE}. Les identifiants sont {CODE}, {NUM-CONTRAT} et {TITRE}. Chacun des déterminants est un identifiant. La relation PROJET est donc normalisée.

A3.12 Calculer les identifiants de la relation ACHAT4. Décomposer cette relation si nécessaire.

ACHAT4(CLIENT, FOURN, ADR-F, ARTICLE, PRIX, DELAI)
 CLIENT, ARTICLE \longrightarrow FOURN, PRIX
 FOURN \longrightarrow ARTICLE, ADR-F
 ARTICLE, FOURN \longrightarrow DELAI

Solution

Identifiants : {CLIENT, ARTICLE} et {CLIENT, FOURN}. Il existe des DF anormales rendant la relation ACHAT4 non normalisée.

Dépendances de base : on observe que la DF ARTICLE, FOURN \longrightarrow DELAI n'est pas minimale; il faut la réduire à FOURN \longrightarrow DELAI, ce qui va simplifier les choses. On réécrit donc l'énoncé comme suit :

ACHAT4(CLIENT, FOURN, ADR-F, ARTICLE, PRIX, DELAI)
 CLIENT, ARTICLE \longrightarrow FOURN, PRIX
 FOURN \longrightarrow ADR-F, ARTICLE, DELAI

On conserve des contraintes d'égalité lors des décompositions. On rectifiera à la fin si nécessaire.

0) Première passe

R1(CLIENT, ARTICLE, PRIX)
 R2(FOURN, ADR-F)
 R3(FOURN, DELAI)
 R4(CLIENT, ARTICLE, FOURN)
 R4: FOURN \longrightarrow ARTICLE
 R2[FOURN] = R3[FOURN] = R4[FOURN]
 R4[CLIENT, ARTICLE] = R1[CLIENT, ARTICLE]

R4 constitue un noyau irréductible non normalisé.

1) La peste (3FN)

R23(FOURN, ADR-F, DELAI)
 R14(CLIENT, ARTICLE, PRIX, FOURN)
 R14: FOURN \longrightarrow ARTICLE
 R14[FOURN] = R23[FOURN]

2) Le choléra (FNBC)

R1(CLIENT, ARTICLE, PRIX)
 R2(FOURN, ADR-F)
 R3(FOURN, DELAI)
 R4'(FOURN, ARTICLE)
 R4''(CLIENT, FOURN)
 R4'*R4'': CLIENT, ARTICLE \longrightarrow FOURN
 R2[FOURN] = R3[FOURN] = R4'[FOURN] = R4''[FOURN]
 R4'*R4''[CLIENT, ARTICLE] = R1[CLIENT, ARTICLE]
Cette dernière contrainte dérive directement de celle du cas (1)

Les contraintes d'égalité nous autorisent à simplifier ce schéma comme suit :

R1(CLIENT, ARTICLE, PRIX)
 R234'(FOURN, ADR-F, DELAI, ARTICLE)
 R4''(CLIENT, FOURN)
 R4''*R234'': CLIENT, ARTICLE \longrightarrow FOURN
 R234'[FOURN] = R4''[FOURN]
 R234'*R4''[CLIENT, ARTICLE] = R1[CLIENT, ARTICLE]

3) La peste et le choléra (FNCE)

R23(FOURN, ADR-F, DELAI)
 R14(CLIENT, ARTICLE, PRIX, FOURN)

R4'(FOURN, ARTICLE)
 R14[FOURN] = R23[FOURN]
 R14[FOURN, ARTICLE] = R4'[FOURN, ARTICLE]

Les contraintes d'égalité nous autorisent à simplifier ce schéma comme suit :

R234'(FOURN, ADR-F, DELAI, ARTICLE)
 R14(CLIENT, ARTICLE, PRIX, FOURN)
 R14[FOURN, ARTICLE] = R234'[FOURN, ARTICLE]

4) Clôture

Il reste à attribuer des noms significatifs aux relations et à préciser les contraintes d'inclusion.

A3.13 En vous servant des propriétés des contraintes d'inclusion, affinez les définitions suivantes :

OFFRE(PRODUIT, FOURN)
 COMMANDE(CLIENT, PRODUIT, FOURN, DATE, QTE)
 COMMANDE[PRODUIT, FOURN] \subseteq LIVRE[PRODUIT, FOURN]

A3.14 On considère une base de données comportant les deux relations

PAYS(NOM, CAPITALE)
 VILLE(NOM, PAYS)

PAYS reprend pour chaque pays son nom et celui de sa capitale tandis que VILLE reprend pour chaque ville son nom et celui de son pays. Sachant qu'il n'y a pas deux pays de même nom, ni deux villes de même nom dans un même pays, complétez le schéma de cette base de données.

A3.15 Selon les propriétés des contraintes d'inclusion, la relation CLIENT(NCLI, NOM, ADRESSE, LOCALITE) ne contient-elle pas une clé étrangère ?

A3.16 Lorsque vous aurez maîtrisé le langage SQL DDL (chapitre A17.6), si par hasard vous repassez par ici, essayez de traduire en SQL les structures de la solution *c* (*La peste et le choléra*) de la section A17.3.8.5.

A3.17 En analysant les données de la relation ci-dessous, déterminer les dépendances fonctionnelles dont elle est le siège. Parmi celles-ci, quelles sont celles qui semblent pertinentes ?

NumPiece	Description	Fournisseur	AdresseFournisseur	Prix
10010	750 GB Disk	Seagate	Washington	120
10010	750 GB Disk	Samsung	Versailles	120
10220	2 GB RAM card	Kensington	Londres	95
10220	2 GB RAM card	Samsung	Versailles	100
10220	2 GB RAM card	Sun Microsystems	Palo Alto	100
10440	21" LCD Monitor	Samsung	Versailles	310

A3.18 Un jeune lecteur de Carpentras nous écrit pour nous faire part de ses doutes sur le procédé de normalisation décrit à la section A17.3.8.4. Il estime par exemple que le schéma suivant :

```
FAB(USINE, PRODUIT, ADRESSE, DESCRIPTION)
  USINE —→ ADRESSE
  PRODUIT —→ DESCRIPTION
```

peut tout aussi bien se décomposer comme suit¹ :

```
U(USINE, ADRESSE)
P(PRODUIT, DESCRIPTION)
U*P: USINE, PRODUIT —→ ADRESSE, DESCRIPTION
```

Qu'en pensent les autres lecteurs ?

A3.19 On reprend, en le modifiant quelque peu l'exemple vu dans l'ouvrage, au chapitre 3. On admet que le cours que fait un professeur dépend aussi de l'année en cours :

```
INSCR(ETUD,MAT,PROF,ANNEE)
  ETUD,MAT —→ PROF
  PROF,ANNEE —→ MAT
```

Ce schéma comporte un circuit de DF. Il possède deux identifiants : (ETUD,MAT,ANNEE) et (ETUD,PROF,ANNEE). Les deux DF étant anormales, cette relation n'est donc pas normalisée.

On observe qu'il y a deux solutions différentes du *choléra* (FNBC). Concentrons-nous sur la dernière forme, *la peste et le choléra* (FNCE). Elle se présente comme suit :

```
INSCR(ETUD,MAT,PROF,ANNEE)
R1(ETUD,MAT,PROF)
R2(PROF,ANNEE,MAT)
INSCR[ETUD,MAT,PROF] = R1[ETUD,MAT,PROF]
INSCR[PROF,ANNEE,MAT] = R2[PROF,ANNEE,MAT]
```

Cependant, un autre lecteur se demande si la solution suivante, plus simple et plus naturelle à son avis, ne serait pas tout aussi correcte :

```
R1(ETUD,MAT,PROF)
R2(PROF,ANNEE,MAT)
R1[MAT,PROF] = R2[MAT,PROF]
```

Ce lecteur a-t-il raison ?

1. La jointure U*P, pour laquelle on ne précise pas les colonnes de jointure, est un *produit relationnel*. Cet opérateur, qui sera décrit à la section A17.8.4, correspond à une jointure naturelle dans laquelle il n'existerait pas de condition de jointure. Chaque n-uplet de U est associé à chaque n-uplet de P.

Solution

Nous sommes hélas forcé de lui donner tort. En effet, ce dernier schéma est plus général que celui de l'énoncé comme nous allons le montrer.

Supposons que la relation INSCR contienne les deux lignes suivantes :

INSCR: (André,Algèbre,Dupont,2014),
(Élise,Algèbre,Dupont,2015)

Les relations R1 et R2 qui en dérivent par projection contiennent les lignes suivantes :

R1: (André,Algèbre,Dupont), (André,Algèbre,Dupont)
R2: (Algèbre,Dupont,2015), (Algèbre,Dupont,2015)

Si nous reconstruisons par une jointure la relation INSCR d'origine, nous obtenons :

INSCR: (André,Algèbre,Dupont,2014),
(André,Algèbre,Dupont,2015),
(Élise,Algèbre,Dupont,2014),
(Élise,Algèbre,Dupont,2015)

On voit apparaître deux lignes erronées. On ne peut donc pas retrouver la relation d'origine, ce qui invalide la proposition.

Remarque

Il existe d'autres techniques de normalisation, notamment celle dite *par synthèse*, qui construit les relations normalisées à partir des DF d'une couverture minimale. Une des règles de cette technique stipule que chaque identifiant minimal doit se trouver dans au moins une des relations finales. La solution de notre lecteur ne respecte pas cette règle, d'où la perte d'information observée.

