Conception Relationnelle Formelle

(Exemples Dirigés & Exercices)

Version 06.10.2013

W. Litwin

Toute version nouvelle a une nouvelle date et actualise toutes celles précédentes. Les MAJ sont sans préavis.

Le but de la méthode est de concevoir formellement la base avec le graphe de références minimal, respectant en pratique l'absence (i) d'anomalies de manipulation et de redondances et (ii) de colonnes nulles ou presque. Il s'agit de la seule méthode à garantir le meilleur schéma relationnel. Contrairement à celles semi-formelles ou empiriques. Basées surtout sur l'approche ER avec ses dérivations : Merise, UML… Pour le modèle relationnel, l'emploi de ces méthodes résulte d'une certaine incompréhension de celle formelle. Tout particulièrement, - des anomalies fréquemment générées par la normalisation d'une table en 0NF en celle en 1NF, discutées en cours. Le travail a pour but la familiarisation avec la méthode. On veut convaincre qu'une fois comprises, les maths constituent une démarche, somme toute simple et naturelle.

On commence par la relation universelle, en 1NF directement, *supposée sans nuls*. On décompose ensuite sans perte de données toute table ayant des anomalies en deux projections recomposables par une jointure interne. On néglige la 5NF où la seule décomposition sans pertes possible est en trois tables au moins. Le besoin est trop rare. Dans la démarche discutée, pour chaque table, celle universelle ou résultant d'une décomposition, en Etape 1, on cherche si la table contient une DM. Auquel cas on l'on élimine du schéma par la décomposition, en vue de la 4NF. Si la table n'a pas de DM, alors en Etape 2, on élimine du schéma de la base par la décomposition de la table, une DF y existante éventuellement, à déterminant non-clé ou super-clé. Aussi bien les DMs que de telles DFs sont responsable d'anomalies, comme on le sait du cours. Le but final est la 4NF pour toute table de la base. Donc la BCNF aussi (revoir la déf. de la 4NF).

Ensuite, on reconsidère la présence de nuls. Le cas échéant, on élimine toute colonne nulle ou presque. On décompose dans ce but toute table du genre en deux tables recomposables sans perte de données par une jointure externe.

Chaque décomposition de la 1ère étape élimine une DM et augmente d'un le nombre de tables du schéma. Le résultat est que pour n DMs éliminées, on finit avec $N_1 = n+1$ tables. D'une manière similaire, pour m anomalies éliminées dans Etape 2, on finit avec $N_2 = N_1 + m$ tables. Enfin, pour q colonnes à nuls éliminées, on aboutit au schéma final à $N_3 = N_2 + q = n+m+q+1$ tables. Ce schéma est optimal, en absence de considérations supplémentaires. Comme celle conduisant à une dé-normalisation discutée en cours. Enfin, comme pour la 5NF, notre démarche néglige quelques autres complications théoriques qui semblent manifestement rares en pratique.

Rappel de la Théorie de la Conception Relationnelle sans Nuls

Dépendance fonctionnelle (DF). Si dans une table R on a deux attributs A,B, peut-être composés, alors on dit que B est en *dépendance fonctionnelle* (DF) sur A ssi B est une fonction monovaluée (mono-valeur), A -> B. L'attribut A est alors le *déterminant* de B. Par exemple, dans la table S, tout fournisseur étant identifié par S# et ayant un seul nom dans NAME, on a S -> NAME. On sait par ailleurs par les *règles d'inférence* d'Armstrong que dans toute R, pour tout A,B,C si A -> B et A -> C, alors A -> BC.

Dépendance multivaluée (DM). Soit R une table avec les attributs A, B et C, peut- être composés. On considère deux fonctions multivaluées de A, notées B(A) et C(A), où pour tout $a \in$ A, B(a) et C(a) sont les ensembles, peut être multivaleurs, des valeurs possibles pour B et C étant donnée a. On suppose que ses ensembles peuvent changer selon le choix de a. On dit alors que B et C *dépendent* de A. On dit ensuite que B est en *dépendance multivaluée* sur A par rapport à (ou étant donné) C si et seulement si B est indépendant de C. Ceci aussi bien dans le sens intuitif habituel que celui formalisé ci-dessous. Le sens intuitif est que pour tout a, à tout choix possible de c pour a, correspond toujours un même ensemble de valeurs de B, B(a) en fait. On exprime la DM comme A ->> B|C. On a à l'évidence toujours la DM *duale* A ->> C|B, avec l'absence de l'incidence de tout choix de $b \in$ B possible pour a sur l'ensemble correspondant des valeurs de C, égal en fait à C(a). On dit aussi que B et C sont (mutuellement) *indépendants*. Egalement, - que A *détermine* B et C dans les deux DMs. On dit enfin que A est le *déterminant multivalué* ou *m-déterminant* de B et de C, si l'on veut insister sur le caractère multivalué des fonctions B(A) et de C(A).

Formellement, l'indépendance entre B et C signifie que pour tout $a \in$ A, $b \in$ B et $c \in$ C, on a : (i) B(a) = B(a,c) et que (ii) C(a) = C(a,b). La condition équivalente à (i) est que quel que soit a, il n'existe pas de c_1, c_2 ; $c_1 \neq c_2$; tels que B (a, c_1) $\neq$ B (a, c_2). Réciproquement pour la condition (ii). La condition équivalente aux (i) et (ii) est aussi que pour tout a, quel que soit (b,c) comme ci-dessus, R contient le tuple (a,b,c). R inclue alors toutes les compositions des valeurs de B et de C pour tout $a \in$ A dans R. Si la condition (i) ou (ii) ou celles équivalentes ne sont pas respectées, il n'y a pas de DMs A ->> B|C et A ->> C|B. On dit pour le non-respect de (i) que B *dépend* de C et vice versa pour (ii). Idem pour leurs équivalences. On dit aussi que B et C sont (mutuellement) *dépendants* ou *en dépendance* si l'on veut simplement dire que B *dépend* de C ou vice versa. Un exercice utile est d'exprimer ce formalisme en algèbre relationnelle du cours ou en utilisant les expressions de sélection de SQL.

Pour illustrer le concept, soit la table P (P#, T, H,…), où une personne identifiée par P peut indiquer plusieurs numéros de téléphone dans T et plusieurs hobbies dans H. L'on peut considérer que les deux attributs sont, comme d'habitude, mutuellement indépendants. Ce qui veut dire en général que pour toute personne, le choix d'un n^o de tél parmi ceux connus pour elle, n'a aucune incidence sur l'ensemble de ses hobbies connus. Vice versa, le choix d'un hobby ne change rien aux téléphones connus de la personne. On a donc deux DMs P# ->> H | T et P# ->> T | H, déterminées par P#. Supposons ensuite que la personne avec P# = P1 ait les téléphones (T1, T2) et les hobbies (H1, H2). On a alors T(P1) = {T1, T2} et H(P1) = {H1, H2}. Ensuite, comme on a discuté en cours, on a dans R toutes les combinaisons de ces valeurs, donc les tuples (P1, T1, H1,…), (P1, T1, H2,…), (P1, T2, H1,…) et (P1, T2, H2,…).

Puis, on voit que T (P1, H1) = {T1, T2} = T(P1) et que T (P1, H2) = {T1, T2} = T(P1) également. Ensuite, on a T (P1, T1) = {H1, H2} = H(P1) et T (P1, T2) = {H1, H2} = H(P1) également. Etc. pour les autres conditions discutées plus haut. Par contre, si jamais l'on savait que pour une activité liée à H1 il ne faut appeler que T1, alors T dépendrait de H etc. Ce qui suffirait pour ne plus avoir de DMs discutées.

Pour rappel succinct de formes normales, on dit qu'une table R est en BCNF si tout déterminant (fonctionnel, monovalué…) est une clé (candidate et/ou primaire) ou une super-clé. On dit que R est en 4NF si elle n'a pas de DMs et est en BCNF. Voir le cours sur la normalisation pour +.

Théorème 1 (Fagin). Soit R (A,B,C) une relation où (i) A,B,C sont des attributs atomiques ou composites et (ii)A ->>B|C (donc B est en DM sur A, étant donné C). Alors, on a la décomposition de R sans perte de données, en projections R' et R'' suivantes:

R (A,B,C) = R' (A,B) Join R'' (A,C)

Th. 1 généralise aux DMs le Th. 2 plus bas, portant sur les DFs et historiquement précédent Th. 1. Les DMs définie de la manière ci-dessus sont dites *non triviales* par certains. Les DFs son dites alors aussi des DMs *triviales*. Voir le cours sur la normalisation ou les réfs du cours pour +. Les DMs et les DFs font partie de la théorie "classique" du relationnel. Dans celle-ci, il n'y a pas de nuls, notamment.

Le but de la décomposition par (1) est la disparition de la DM. Ceci, par la séparation de B et C dans deux tables distinctes. Théorème 1 n'est cependant aisé à appliquer tel quel que dans des cas d'école de la table à 3 attributs atomiques. C'est loin d'être la situation typique où B ou C peuvent être composés de dizaines d'attributs. Il faut un algorithme itératif permettant d'examiner les attributs un à un pour conclure si un choix de A et B et donc de C = R\(A,B) est conforme au Th. 1. La dérivation suivante du Th. 1 sert à ce but. Si aucun choix de A et B n'est trouvé, alors R n'a pas de DM et Th. 1 ne s'applique pas.

Théorème 1bis. Soit R (A,B,C,D) une relation où (i) A,B,C,D sont des attributs atomiques ou composites et on a (a) A ->>B|C. On a la décomposition de R sans perte de données, par les projections R' et R'':

R (A, B, C, D) = R' (A, B) Join R'' (A, C, D)

si R remplit les conditions suivantes :

(i) D est vide ou

(ii) D ne contient pas d'attribut X en dépendance avec un attribut Y de B. X, Y, Z peuvent être composites.

La preuve passe d'abord par l'observation que (i) signifie que Th. 1 s'applique directement. Ensuite (ii) signifie l'existence de la DM A ->> B | D (et donc de A ->> D | B). Alors on peut appliquer la règle d'inférence *d'augmentation* de Fagin & al:

Si A ->> B | C et A ->> B | D alors A ->> B | (C,D).

Th. 1 s'applique alors, en considérant (C, D) comme C de Th. 1. Observez enfin que l'existence de la DM

A ->> B | D (et donc A ->> D | B) résulte en fait formellement aussi de cette règle, appliquée successivement à tout attribut atomique de D et de B.

Exemple 2. ci-dessus illustre l'application de Th. 1bis. En pratique, on peut avoir la situation où le choix de A, B, C et D respecte la condition (a), mais pas (i) ou (ii). Une restructuration mutuelle de B et de D selon Lemme qui suit peut alors aider.

Lemme 1. Supposons que dans la relation R (A,B,C,D) on a A ->> B | C et que D contient un attribut X en dépendance avec un attribut Y de B, X et Y pouvant être composés. Alors, R n'est pas conforme au Th. 1bis. Supposons alors que l'on *restructure* B et D dans la DF, en mettant B := (B, X) et D := D\X. Alors R peut devenir conforme. Plus précisément R devient conforme si (c) X ne dépend pas de C et (d) D restructuré, à son tour ne contient aucun autre X avec les propriétés discutées.

La preuve du lemme passe par l'observation que si D contient X discuté, alors on ne respecte pas la condition (ii) de Th. 1bis. Le Th. 1bis et donc Th. 1 ne s'appliqueraient plus. Sur des exemples faciles à construire, notamment ceux ci-dessous, on peut voir qu'une perte de données pourrait alors résulter de décompositions selon ces théorèmes. Si l'on restructure B en B' = (B,X) et donc D en D' = (D\X), alors D' peut ne plus avoir d'attribut qui dépend de B' ou vice versa. On a alors A ->> B'|D\X. Pour le respect de Th. 1bis, vu la règle de l'augmentation, il faut et il suffit alors que l'on ait A ->> B' | C. Donc il faut vérifier que X et un attribut de C ne soient pas en dépendance. En général, c'est facile.

Il se peut alternativement que D' contient un autre attribut X' qui serait X pour B', D'. Alors il faut appliquer itérativement Lemme 1 pour X' etc. Il est aisé de voir que l'on peut aboutir au respect des conditions du Lemme et donc du Th. 1bis. Voir les exemples plus bas et notamment Exemple 2.4 et Exemple 3 plus loin.

Dans les deux cas, il se peut néanmoins que notre choix ne respecte pas (i), car X s'avère en dépendance avec C aussi. Auquel cas, on peut tenter et réussir un autre choix de B ou C. Certains exemples ci-dessous illustrent ce cas, Exemple 4 et Exemple 5 notamment.

L'impossibilité d'appliquer Th. 1 et Th. 1bis peut concerner des tables même à seulement trois attributs. Il se peut alors que R n'est pas décomposable. C'est le cas d'une relation ternaire où les trois attributs sont interdépendants. Voir Exemple 2.2, 4 et Exemple 5 ci-dessous. Restructurer B et D comme dans le lemme n'aide pas alors (pourquoi ?). Il se peut enfin aussi, nous rappelons, que R est en 4NF, mais pas en 5NF. Des anomalies subsistent dans ce cas.

D'aucuns peuvent finalement trouver pratique une formulation un peu différente de Th. 1bis et du Lemme 1, sous forme d'un lemme unique suivant :

Lemme 2. Supposons que l'on a R (A, B, C, D) avec A ->> B | C. Alors les deux situations suivantes permettent à des décompositions sans perte.

(a) Si A ->> C | D, alors on a R = R'(A, BD) Join R''(A,C)
(b) Si pour un X $\subset$ D on a A ->> C |X et A ->> D\X | X et A ->> B | D\X, alors R = R'(A, BX) Join R''(A,CD\X).

Ici, la notation BD, signifie la même chose que (B,D) etc. Pour la preuve de (a) observez que l'on a aussi la DM : A ->> C | B et donc par la règle de Fagin on a A ->> C | BD donc A ->> BD | C. La décomposition annoncée résulte du Th. 1, appliqué à BD. Pour (b), observez que l'on a d'une manière similaire la DM : A ->> BX | C et celle A ->> D\X | BX. En utilisant la DM duale à la dernière, on a alors, par la même règle, la DM : A ->> BX | CD\X et la décomposition annoncée en vertu du Th. 1.

Comme on voit, (a) remplace l'étude de l'indépendance mutuelle de A et D du Th. 1bis par celle de C et D. La formulation (b) est sous forme d'une *règle d'inférence*. La règle encode le résultat final d'une application peut-être itérative et en tout cas réussite, du Lemme 1. Ceci dit, comme pour Th. 1bis et Lemme 1, on peut analyser la conformité de R au Lemme 2 itérativement aussi. On analyse alors un attribut dans D ou X après l'autre, en les recomposant au fur et à mesure par la règle de Fagin, en D ou X finaux, selon (a) ou (b). Analysez l'emploi de Lemme 2, notamment par cette approche, à travers les exemples qui suivent.

La restructuration de Lemme 1 se généralise au cas de X en DF (Y, Z) -> X, avec Y et Z du Th. 1bis. Idem en fait pour les DFs : (Y, X) -> Z ou (X, Z) -> Y. Voir Exemple 3 déjà mentionné, ainsi que Exemple 5. Il est aisé d'imaginer d'autres exemples correspondants.

Théorème 2 (Heath). Soit R (A, B, C) une relation où A ->B. Alors on a la décomposition de R sans perte de données, en projections R' et R'' suivantes:

R (A, B, C) = R' (A,B) Join R'' (A, C).

Application des Théorèmes

Th. 2 et, dans bien moindre mesure, Th. 1, sont le fondement de la théorie "classique" du relationnel. Sans présence de nuls donc, ils permettent à la décomposition sans pertes de données (voir le cours sur la normalisation, si besoin). Comme on a dit, si une table R a des anomalies, on tente d'abord appliquer Th. 1. On choisit en 1er pour A un attribut identifiant un objet en réalité. D'une manière générale, on choisit pour A et B des attributs atomiques si possible, minimaux autrement. La priorité de la recherche d'une DM résulte du théorème suivant.

Théorème 3. Soit R une relation avec une DM et une DF dont déterminant n'est pas une clé. Soit R1 et R2 la décomposition sans perte de données de R selon la DM et R3 et R4 celle selon la DF. Alors R3 ou R4 peuvent comporter une anomalie impossible à supprimer par la décomposition sans perte. Par contre, R1 et R2 peuvent être libres de toute anomalie.

Preuve. Soit R (A,B,C) avec A ->> B | C et B -> A, ces attributs étant atomiques. Alors, on a R1 (A,B) et R2 (A,C) par Th. 1. Puis, on a R3 (B, A) et R4 (B, C) par Th 2. R4 comporte l'anomalie annoncée. R1 et R2 sont sans anomalies.

Voir aussi Exemple 3 ci-dessous et le cours.

Th. (3) est donc la source de notre règle que l'on pourrait appeler *« DMs d'abord »*. Ainsi, pour toute table R, R = U ou R résultant de décompositions successives de U, si R contient une DM et une DF anormale, alors on décompose R par Th. 1 ou 1bis, non pas par Th. 2. Toujours dans cet ordre et jamais inversement. Ce qui revient à dire que pour toute R, on cherche on en priorité une DM. Seulement si on ne trouve pas, on y cherche une DF. Plus précisément, on ne cherche qu'une DF X -> Y à déterminant X *anormal*, c'est à dire qui ne serait pas une clé ou super-clé. Si on trouve une telle DF, on applique (2) avec A = X et B $\supseteq$ Y, comme l'on discute un peu plus loin. A devient alors la clé (ou super-clé) dans R' et la DF devient *normale*. En même temps, la DF n'est plus un problème pour R'', car A seul y reste. Si on ne trouve aucune, R est considérée comme finale pour la conception relationnelle classique. C'est-à-dire R est sans anomalies que l'on pourrait supprimer par une décomposition sans perte par les projections, recomposables en R par leur jointure naturelle interne. Très rarement néanmoins ceci n'est pas le cas comme on rappelle plus bas.

En appliquant Th. 2, on prend en fait pour B l'attribut composé de tous les attributs que X détermine. On peut le faire, car, pour rappel, (le cours sur la normalisation): une de règles d'Armstrong pour les DFs, dit que pour tout attribut X, Y, V, si X -> Y et X -> V alors X -> YV. Cette règle est aussi dite aussi *d'augmentation*. Si l'on ne procédait pas ainsi, la décomposition aurait généré trop de tables résultant, on rappelle le cours, en graphe qui ne serait pas optimal, à cause de redondances inutiles sur A.

Un tel usage de Th. 2 aboutit à la BCNF pour toute table de la base. Donc - à 3NF, 2NF et 1NF bien sûr (revoir les définitions des NFs concernées). Th. 1 aboutit en plus à 4NF pour toute table.

On peut rappeler encore que le choix de B peut ne pas être unique. Le cas notoire est celui de A –> B_1 où il se trouve aussi que B_1 -> B_2, $B_1 \neq B_2$. Ce qui implique (règle de transitivité d'Armstrong) que A -> B_2 aussi. Dans ce cas, il faut choisir B = B_1. Ce qui est dit la décomposition en projections *indépendantes*. L'autre choix, A -> B_2 donc, est dit en projections *dépendantes*. Il rompe la dépendance B_1 -> B_2, en séparant ces deux attributs entre R2 et R3. Ce qui crée une anomalie de MAJ, illustrée dans le cours sur la normalisation et un peu dans la discussion d'Exemple 3 ci-dessous. A éviter donc.

Le Th. 1bis, le Lemme 1 et Th 3 datent d'il y a que quelques années. Ce qui est très récent par rapport aux 40 ans du modèle relationnel. De ce fait, la règle et la démarche formelle ci-dessus sont encore peu connues. Malgré leur simplicité pratique et la garantie d'optimalité pour le relationnel classique, illustrées par les exemples qui viennent. Les principaux livres et cours en BDs connus n'en parlent pas encore. La plupart proposent même encore la démarche inverse : DFs d'abord, DMs éventuellement ensuite. Ne sachant pas que le résultat erroné peut en résulter. En espérant, semble-t-il, que les DMs avaient été éliminées préalablement par la modélisation conceptuelle semi-formelle, par ex. par UML. Ça peut être le cas, mais sans garantie. Tout bénéfice pour les participants de ce cours après tout, dans ce monde compétitif à outrance.

Enfin, précisons encore une fois que la démarche présentée conduit au graphe de références optimal en général seulement. Déjà on ne traite pas d'anomalies à résoudre par la 5NF seulement. Puis, il y a aussi des cas d'école où des redondances et anomalies subsistent dans une table malgré sa 4NF, sans que l'on puisse les résoudre par une décomposition sans pertes de données par les projections et une jointure naturelle supplémentaires de quelconque manière. Un cas du genre est celui d'un dictionnaire trilingue A, B, C avec les DMs A ->> B|C, B ->> A|Cet C ->> A|B. Voir + dans les références du cours. Plus loin, après les exemples et les exercices qui suivent, nous admettons

aussi des valeurs nulles et les anomalies spécifiques qui peuvent alors apparaitre. Celles-ci ne peuvent plus être éliminées par des décompositions sans perte de données par des jointures naturelles, le fondement de la conception relationnelle classique et des théorèmes ci-dessus. Nous montrons que l'on peut éliminer ces anomalies néanmoins, sur la base d'un théorème employant des jointures dites externes.

Exemples & Exercices

Exemple 1**.**

1. On conçoit par la démarche formelle le schéma de la base S-P du cours.

   U(S#, SName, SCity, Status, P#, PName, Color, Weight, Qty)

   Déterminants: S#, P#, (S#, p#)
   Clé(s) (S#, P#)

   DM(s) ?

Il n'y a pas. Pouvez-vous dire pourquoi au juste ? Sinon voir le paragraphe suivant. En absence de DMs, seul s'applique alors le Th. de Heath.

En ce qui concerne l'absence de DMs. On peut penser d'abord à une DM entre trois attributs déterminés par S#. Par ex. STATUS ->> S# | CITY ? Selon l'extension de S dans base S-P du cours, on aurait alors dans U notamment pour STATUS = 30 les ensembles S# = {S3, S5} et CITY = {Paris, Athens}. La décomposition selon Th. 1 donnerait dans R' les tuples (30, S3), (30, S5) et dans R'' les tuples (30, Paris) et (30, Athens). La décomposition devrait être sans perte. Il faut s'en assurer donc, en appliquant sa définition du cours. La recomposition donnerait pourtant notamment le tuple (30, S3, Athens). Or, il n'y pas de tuple correspondant dans U, car S3 dans S-P n'est qu'à Paris. Cette décomposition est donc à perte. Notez le caractère étrange de ce « perte », car c'est un tuple que serait en fait crée « en trop ». Qui qu'il en soit, la conséquence est que la DM considérée n'existe pas dans U. La raison intuitive est que STATUS et CITY sont en U dans la dépendance à travers S# déterminant les deux. Donc, aussi bien pour S3 que pour S5 il ne peut y avoir qu'une ville. Si S# et CITY avait été par contre indépendants, alors toute ville possible pour S3 devrait être possible pour S5 et vice versa. Ce n'est pas le cas. D'où la perte montrée notamment. Idem pour le choix d'une DM concernant tout autre triplet d'attributs déterminés par S. La situation est similaire pour tout triplet d'attributs déterminés par P#, à l'évidence.

On peut alors penser que les seules DMs A ->> B|C qui pourraient exister seraient entre B déterminé par S# et C déterminé par P# ou (S#, P#), c. à d., entre les attributs déterminés par différent déterminants trouvés. Intuitivement tout choix conduirait alors néanmoins à une dépendance entre B et C due au fait qu'il s'agit toujours de propriétés de pièces fournies par un fournisseur et que les fournisseurs peuvent fournir certaines pièces seulement, pas toutes. Celui à qui cette vue intuitive semblerait peut-être pas vraie, pourrait tenter de nouveau une voie formelle d'un contre-exemple, comme ci-dessus. En commençant par ex. avec la conjecture DM SCity ->> S# | P# ? Alors, selon S-P, on aurait notamment dans l'extension de U les lignes suivantes pour deux fournisseurs S1 et S4 : (S1, Smith, London,…P1,…)… (S1, Smith, London,…P6,…)… et (S4, Clark, London,…P2,…), (S4, Clark, London,…P4,…), puisque S1 fournit les pièces P1…P6 dans S-P, mais S4 ne fournit que P2 et P4. L'application du Th. 1 donnerait la décomposition en projection R' avec notamment les lignes (London, S1), (London, S4) et R'' avec notamment (London, P1), (London, P2), (London, P3), (London, P4). La décomposition devrait être sans perte de nouveau. La recomposition donnerait notamment le tuple (London, S4, P3). Ce qui signifierait que S4 fournit P3. Ce n'est pas vrai dans S-P. La décomposition n'est pas donc sans pertes. La conjecture serait également fausse. Un raisonnement similaire s'appliquerait à toute autre conjecture d'une DM dans U qui aurait pu venir à l'esprit.

DF(s) avec les déterminants qui ne seraient pas les clés de U ?

Il y en a, donnant une suite de décompositions possibles ci-dessous:

A = S#, B = (Sname, Status, SCity) C = (P#, PName, Color, Qty, Weight)

R1 (A,B) = (S#, Sname, Status, SCity)

R2 = (S#, P#, PName, Color, Qty, Weight)

On continue avec la décomposition de R2:

A = P#, B = (Pname, Color, Weight) C = (S#, Qty)

R3 = (P#, Pname, Color, Weight) R4 = (P#,S#, Qty)

Résultat final, après le renommage: S := R1, P := R3 et SP := R4:

S (S#, Sname, Status, SCity)
SP (P#,S#, Qty)
P (P#, Pname, Color, Weight)

Notez à l'occasion la présence de redondances que l'on n'a pas pu éliminer dans le cadre de 1NF. Notamment sur S#. On a introduit aussi une redondance interrelationnelle entre clé primaire et celle étrangère. Ce qui conduit à une possible inconsistance du lien sémantique dans le cas d'une MAJ de S# ou de P#. D'où les deux choix optionnels dans la définition d'une contrainte d'intégrité relationnelle de MsAccess dont on a parlé en cours.

2. On suppose maintenant que dans S aussi bien S# que Sname identifie un fournisseur. Soit maintenant S' (S#, Sname, City, P#, Qty). C'est un ex. de table en 3NF mais pas en de BCNF, pour rappel du cours.

A vous de jouer :

Déterminants, DMs,DFs, Clés, Anomalies ?

Exemple 2.

BDs avec des DM et peut-être des DFs. Les deux théorèmes s'appliquent.

1. U est la table universelle des étudiants à Dauphine dont on a parlé en cours. On suppose que U a néanmoins que trois attributs :

U (E#,Tel,Email)

Tel et Email sont multivalués et indépendants.

Montrez la clé de U et la DM. Décomposez par Th. 1 ou Th 1bis.

2. (a) On suppose maintenant U comme suit:

U (E#,Tel, Email, C#)

Ici, tout étudiant suit plusieurs cours identifiés par C#. Montrez la clé de U et quelques DMs dont E# est le (multi)déterminant et qui permettent d'appliquer Th. 1bis. Montrez votre décomposition. Puis, pourrait-on appliquer Th. 1 directement ?

Réponse partielle. Par ex. on pourrait commencer par prouver la DM :

E# ->> Tel |(Email, C#)

Aussi bien par Th 1 ou Th 1bis, ceci donnerait la 1ère décomposition possible R = R'(E#, Tel) join R''(E#, Email, C#). A vous de continuer si besoin est.

(b) On suppose U (E#, Email, C#). Email contient maintenant les adresses A1…A5 pour communiquer avec les enseignants de cours. On suppose d'abord que pour tout cours, tout Email peut servir à tout étudiant du cours. Tout étudiant communique alors par un même sous-ensemble des adresses qu'il choisit, quel que soit le cours qu'il suit. A-t-on la DM C# → E# | Email dans U ? Puis, on suppose que pour cours C1, Email = A1 est réservé à l'étudiant E1, le délégué de la classe. Tout autre étudiant du cours C1, peut néanmoins inclure A1 dans son sous-ensemble des adresses pour tout autre cours qu'il suit. A-t-on toujours la DM C# → E# | Email dans U ?

(c) On suppose U égal à la table Etud (E#, Tel, Hobby, Dipl, Enfants, Voit) vue en cours. Proposez le schéma

optimal.

Conseil. Observez que seul Th. 1bis suffit. On ne décompose au fur et à mesure que par rapport aux DMs. Pas besoin de lemme ni de Th. 2. Pourquoi au juste ?

3. On suppose maintenant U avec davantage d'attributs, comme ci-dessous. Notamment, un étudiant peut avoir plusieurs n° Tel avec leurs types (domicile, portable…) et plusieurs emails. Les deux sont mutuellement indépendants et tout tél ou email peut être partagé entre des étudiants. Puis, chaque cours C#a un seul Cnom. Pour chaque étudiant et chaque cours il y a une et une seule Note. Ville est une fonction de CP, enfin.

U (E#, ENom, CP, Ville, Tel, TTel, Email, CNom, C#, Note)

On décompose U par le Th. 1bis d'abord et Th. 2 ensuite. Néanmoins, libre à vous de tenter alternativement Th. 1.

Une DM ?

Oui, entre autres : E# ->>(Tél, TTel) | Email). Notez l'application du lemme. En effet, le 1er choix intuitif pourrait être simplement E# ->>Tél | Email. Mais alors la condition (ii) de (1bis) ne serait pas respectée. Pourquoi + précisément ?

Avec cette DM, on a :

A = E#, B = (Tel, TTel) C = Email et D = (ENom, CP, Ville, CNom, C#, Note)

1ère décomp. par Th. 1bis:

Avec U = R, R1 = R' et R2 = R'', on a :

U = R1 join R2

Avec :

R1 (<u>E#</u>, <u>Tél</u>, TTel) et
R2 (<u>E#</u>,<u>Email</u>,ENom, CP, Ville, CNom, <u>C#</u>, Note)

Il n'y a plus de DMs dans R1. Une DM dans R2 ?

Oui, notamment E# ->> Email | C#

2ème décomp., celle de R2 (Th. 1)

Donc A = E#, B = Email C = C# et
D = (ENom, CP, Ville,CNom, Note)

R2 = R3 Join R4 avec

R3 (<u>E#</u>, <u>Email</u>) et
R4 (<u>E#</u>, <u>C#</u>, ENom, CP, Ville, CNom, Note)

R3 et R4 n'ont plus de DMs. On passe aux DFs et Théorème de Heath exclusivement désormais. Y-a-t-il donc, d'abord, un déterminant n'étant pas une clé dans R3 ?

Il n'y a pas.

Les clés dans R4 et un déterminant qui ne serait pas une clé svp?

La clé primaire est déjà soulignée et on a alors E# comme déterminant non-clé. Ce qui conduit à :

DF: A = E# ->B (ENom, CP, Ville)

Donc par le Th de Heath on a :

R4 = R5 Join R6

avec R5 (E#, ENom, CP, Ville) et R6 (E#,CNom, C#, Note)

On a alors dans R5la DF CP -> Ville. Alors, on décompose R5 en ?

R7 (E#, ENom, CP) et R8 (CP, Ville)

C'est une décomposition en projections indépendantes. On pourrait alternativement décomposer R5 en projections dépendantes:

R7' (E#, ENom, CP) et R8' (E#, Ville)

A éviter car on rompe la DF CP -> Ville et il y des ennuis de MAJ (lesquels ?). R5 est un ex. d'une table dite en 2NF, mais pas en 3NF.

Pour une même raison, on décompose R1 en R9 (E#, Tel) et R10 (Tel, TTel).

Enfin, on décompose encore R6 par le Th. de Heath en

R6 = R11 (C#, CNom)JoinR12(C#, E#, Note)

Pourquoi ?

R6 est dite en 1NF et pas en 2NF (voir le cours sur la normalisation).

Il reste à décomposer R1, en R13 (E#, Tel) et R14 (Tel, TTel). Le schéma final est R3, R7…R14. En général, on renomme ces tables d'une manière + parlante, p. ex. :

ET (E#, Tel)
TT (Tel, TTel)
EE (E#, Email)
E (E#, ENom, CP)
CV (CP, Ville)
CN (C#, CNom)
EC (C#, E#, Note)

Q1. Quelle serait R2 si les adresses emails n'étaient pas supposés partagées peut-être entre plusieurs étudiants ?

R1. R3 ne serait pas la même si les adresses émail n'étaient pas partageables. On aurait la DF email –> E#. Email serait un déterminant et la clé. Dans R3 ci-dessus, email est un attribut-clé seulement. Voir le cours sur la normalisation pour + sur cette issue.

Observez notamment que si l'on décomposait R2 par rapport à la DF, en négligeant l'existence de la DM, contrairement à la démarche discutée, on aurait abouti à une anomalie sans issue. Ce qui se passerait si on appliquait en fait la démarche proposée dans les livres sur les BDs. L'observation discutée est plus récente que ces livres, ne datant que d'il y a quelques années.

Q2. Suppose que pour la décomposition de R2 plus haut, on avait observé la DM E# ->> C#| Email au lieu de E# ->> Email| C#. Comment continue-t-on ?

R2.On a alors aussi C# ->CNom et (E#, C#) -> Note. Alors on choisit B = (C#,Cnom, Note).En conséquence on a:

R3' (E#, C#, CNom, Note) et
R4' (E#, Email,ENom, CP, Ville)

Si on avait séparé CNom ou Note de C#, en mettant les 1èrs dans R4' avec le dernier dans R3', alors la décomposition pourrait ne pas être sans perte de données. Un exemple facile à construire par vous-mêmes, peut vous convaincre déjà que la décomposition de R simplifié à R (E#, C#, CNom, Email) avec la DM E# ->> C# | Email et la DF C# ->CNom, en R' (E#, C#) et R'' (*E#*, Email, CNom) pourrait ne pas être sans perte de données.

Q3. A quel moment ci-dessus**,** Lemme 1 aurait pu nous servir ?

R3. Si, à tort, on aurait commencé par la DM: E# ->> Tél | Email.

4. Considère la table Etud (E#, Nom, Tél, Email). On a comme plus haut la DM E# ->> Tél | Email. On a aussi la DF E# -> Nom. (a) Est-ce que Th. 1bis s'applique ? (b) Pouvez-vous proposer une décomposition sans perte ?

Exemple 3. On étend l'exemple du cours comme SP (S#, P#, L#, PL) où PL est le prix unique de livraison de toute pièce P# à partir de la localisation L#. On considère comme en cours, que toute localisation L# déclarée pour un S# peut fournir toute pièce que S# fournit. Ceci étant, PL peut différer pour même L# selon la valeur de S#. A-t-on de DMs dans SP et peut-on le décomposer en conséquence ? Créez un ex. de démonstration dans un sens ou dans l'autre.

Suggestion. Considérez A = S#, B = L#, C = P# et D = PL.

Exemple 4. Dans une clinique, un patient P# est soigné dans certains services S#, par certains, parmi tous, médecins M# attachés à ces services. Indépendamment, tout S# a aussi certaines affections A# soignée chacune par un ou plusieurs médicaments Me#. A titre d'exemple, supposons que le patient Dupont identifié par P# = '123' est soigné pour le troubles cardiaques et mémoire dans le service cardio par Dr. Durant et Dr. Durand et dans le service neuro par le même Dr. Durand et Dr. Plessier. Puis, supposons qu'indépendamment du traitement dans ces services, Dupont prend ses propres médicaments: Fenofib contre son cholestérol et Advi et Dafal contre le mal du dos. Les médicaments sont identifiés chacun par une certaine valeur de Me#, les médecins et les services par des valeurs de M# et de S# respectivement. Proposez le schéma optimal. Illustrez l'emploi éventuel du lemme.

Réponse partielle. Le schéma final est S (P#, S#, M#), A (P#, A#, Me#). S peut être un mnémonique pour Soins, A pour Affections.

Exemple 5. (a) Supposons U (C#, P#, L#, Ch#) où C# désigne un cours, P# un prof qui donne ce cours et L# un livre de support du cours. On suppose la DM C# ->> P# | L# donc U' (C#, P#, L#) est décomposable par Th. (1). En fait c'est l'exemple classique de DM donné par Fagin, voir le cours sur la normalisation. On suppose maintenant que (i) Ch# est un n^{o} de chapitre recommandé dans un livre par un prof. Et que (ii) différents profs d'un même cours peuvent recommander différents chapitres d'un même livre. De même, un même prof peut recommander différents chapitres selon les cours utilisant un même livre. Est-ce que U est décomposable par Th. 1 ou 1bis ?

(b) Supposons maintenant U^2 (C#, P#, L#, Ch#, S#) où S# identifie les salles pour un cours, choisies indépendamment du reste de U^2. Est-ce que Th. 1bis s'applique si l'on explore la DM ci-dessus ? Sinon, est-ce que U est décomposable par (1) ou (1bis) néanmoins ?

(c) Ensuite, soit U^3 (C#, P#, PNom, L#, LPrix). PNom est le nom (unique) d'un prof. Puis, chaque prof peut vendre pour LPrix tout livre d'un cours. Le prix est le même pour tout cours. Mais différents profs peuvent demander différents prix pour un même livre. Proposez le schéma optimal. Que ce que vous avez observez dans votre démarche de particulier par rapport à tous les exemples précédents ?

Conception Formelle en Présence de Nuls

On applique (si besoin) cette conception sur toute table du graphe de références optimal, obtenu par la conception relationnelle classique sans nuls ci-dessus. Toute table considérée ci-dessous est dès lors en 4NF au moins. On alors éliminer les anomalies éventuelles de présence de valeurs nulles « en trop ». En général il s'agit de nuls inapplicables. Leur présence indique en général l'existence de sous-classes. Voir le cours.

Rappel : Jointures externes

Soit R1 (A,B), R2 (B,C) deux relations. Soit R3 = R1 join R2. Soit R.nul un tuple avec tous les attributs de R et un nul pour chacun, c. à d. R.nul = (nul,…,nul). Alors, on a, en utilisant la notation algébrique du cours:

(1) R1 left join R2 = R3 union (R1 diff R3 [A,B] times R2.nul)
(2) R1 right join R2 = R3 union (R2 diff R3 [A,C] times R1.nul)
(3) R1 outer join R2 = R1 left join R2 union R1 right jointures oin R2

Ces jointures sont dites *externes* (et naturelles), respectivement *gauche*, *droite* et *pleine* (ang. full). La jointure (naturelle) algébrique est dite alors *interne*. Par analogie évidente aux equi-jointures pour les jointures internes naturelles, on a aussi les *equi-jointures externes*. Notez que la jointure externe se ramène à celle interne en absence de tuples de R1, respectivement de R2, sans tuple égal sur l'attribut de jointure dans R2, respectivement en R1.

A titre d'illustration, dans la base S-P du cours, S join SP n'aura pas de tuple avec S# = S5. Par contre, S left join SP l'aura. Avec les nuls dans les colonnes P# et Qty. Enfin dans QBE de MsAccess, quand on dessine la ligne de jointure et l'on clique là-dessus, dans l'écran du choix de type de jointure qui s'ouvre alors, (1) est une jointure interne, (2) et (3) sont celles externes. La notion de gauche-droite n'a pas de sens dans cet interface graphique. Donc, parmi les deux tables à joindre, on choisit explicitement celle dont toutes les valeurs des attributs sélectionnés doivent être préservées.

Théorème 3 (Litwin). Décomposition sans perte d'une table avec des nuls (éventuels), en vue d'un schéma optimal, sans anomalie « trop de nuls ».

Soit R1 (A,B,C) une relation en 4 NF au moins, donc après la normalisation habituelle. Les attributs A, B et C peuvent être composés. On dit qu'un tel attribut, soit B, n'est pas nul dans un tuple de R1, si et seulement si au moins l'un des attributs composant B est non-nul. On suppose ensuite que (i) A est une clé (donc tous les attributs composant A sont non-nulles), (ii) B peut contenir des nuls, en principe inapplicables, (iii) dans R2 (A, B) = R1 (A, B) Where Bis not nul, on a A -> B, (iv) C peut être vide. Soit enfin R3 (A, C) égal à R1, mais avec chaque tuple réduit à ses colonnes A et C seulement. Alors on a la décomposition de R1 sans perte de données comme suit:

R1 (A, B, C) <-> R2 (A, B), R3 (A, C)

Avec la recomposition par la jointure externe naturelle droite comme suit :

R1 (A, B, C) = R2 (A, B) right join R3 (A, C)

Note. Th. 3 peut être vue comme une généralisation du Th. 2, de Heath. Ceci vient du fait que sans présence de nuls, les jointures externes naturelles ce réduisent à celles internes.

Exemples et Exercices

Exemple 6

a. Soit Pers0 la table du personnel à Dauphine :

Pers0 (P#, Nom, Prénom, NomJF, S#, Indice)

Pers0 est en 4NF (démontrez svp) et en 5NF en fait. Est-ce que P, V, S, E sont en 4NF ou en BCNF ? Donc libre d'anomalies que dont on pourrait se débarrasser sans perte de données pour le relationnel classique. On s'aperçoit néanmoins que la plupart de personnes à Dauphine n'ont pas de nom de jeune fille. De même, beaucoup de personnes ne sont pas des salariés de Dauphine, n'ayant pas alors d'ID S# et d'indice. Par contre, on connaît le Nom, Prénom de toute personne. Le tout conduit à beaucoup de nuls inapplicables dans toute extension de Pers0. Ils occupent la place dans la base, conduisent au travail inutile pendant l'insertion d'un tuple et enfin conduisent aux problèmes de manipulation que l'on verra dans le cours. Il est utile de s'en débarrasser à condition que l'on ne perde pas de données

au passage. Ce que Th. 3 permet de faire comme suit.

1ère décomposition : R1 = Pers0, A = P#, B = NomJF, C = (Nom, Prénom, S#, Indice), R2 = (P#, NomJF), R3 = (P#, Nom, Prénom, S#, Indice). On a aussi, par Th. 3:

R1 = R2 (P#, NomJF) Right Join R3 (P#, Nom, Prénom, S#, Indice).

Ici R2 ne contient que les tuples de R1 et tous les tuples du genre, où NomJF n'est pas nul. Donc pas d'anomalie discutée. Par contre R3 la présente encore. Donc :

2ème Décomposition, celle de R3 :

A = P#, B = (S#, Indice), C = (Nom, Prénom)

R3 = R4 (P#, S#, Indice) Right Join R5 (P#, Nom, Prénom)

Remarquez que si toutes les personnes dans Pers0 seraient les salariés, alors cette décomposition se réduit à une application du Th. 2. Inutile et même nocive toutefois dans ce cas (pourquoi ?).

Schéma final, après le renommage:

Pers (P#, Nom, Prénom)
PersJF (P#, NomJF)
PersSal (P#, S#, Indice)

Notez que la décomposition montre ici une classe (Pers) avec deux sous-classes. Ces classes reflète la présence de la correspondance bien connue de la littérature, dite ISA, de l'anglais « Is a « et en français « est un(e) ». Ainsi notre décomposition a mise en lumière que dans notre base, toute personne jeune fille *est une* personne et que toute personne salariée *est une* personne aussi. Nos ISA forment une hiérarchie à un niveau sous la racine Pers. La correspondance ISA est néanmoins en général transitive et peut former un treillis. On peut enfin avoir dans notre base peut-être une sous-classe contenant toute personne ayant le nom de jeune fille et étant un(e) salarié(e). Si c'est le cas, alors sur le schéma, une telle classe correspondrait à une jointure interne implicite entre PersJF et PersSal.

b. P (P#, Nom, Prenom, E#, Dipl, S#, Sal, V#, Lab)

On rappelle du cours qu'une personne dans P, identifiée toujours par P#, est en général, mais pas toujours, soit un étudiant ou un salarié soit un visiteur. On reconnait ce fait par des colonnes de nuls correspondants. On considère que pour chaque colonne concernée (lesquelles ?) il y alors dans P « trop » de nuls inapplicables, conduisant aux anomalies. Ce qui indique la aussi la présence dans P de trois sous-classes (lesquelles ?). Enfin, pour cet exemple, on suppose que tout étudiant ne prépare qu'un seul diplôme et que tout étudiant prépare un nécessairement.

Clés et déterminants ?

1ère décomposition :

P = R1 (P#, E#, Dipl) right join R2 (P#, Nom, Prenom, S#, Sal, V#, Lab)

2ème décomposition:

R2 = R3 (P#, S#, Sal) right join R4 (P#, Nom, Prenom, V#, Lab)

3ème :

R3 = R5 (P#, V#, Lab) right join R6 (P#, Nom, Prenom)

Au finale:

P (P#, Nom, Prenom)
V (P#, V#, Lab)
S (P#, S#, Sal)
E (P#, E#, Dipl)

Clé candidates ? Est-ce que P, V, S, E sont en 4NF ou en BCNF ? Y a-t-il d'autres sous-classes que celles, disons, V,S,E ?

Exemple 7. Il s'agit d'une conception d'ensemble d'une BD, suivie de sa création pratique sous MsAccess. A chaque fois il faut proposer un schéma (graphe des références) optimal, prouvé par la démarche formelle. Précisez notamment pour tout arc si c'est un lien référentiel, ou une contrainte d'intégrité. Précisez les jointures implicites.

a. On veut créer la base de mariages monogames homme-femme. On suppose que le mari est identifié par M# et l'épouse par E#. Chacun porte un nom, ces noms pouvant être différents, p.ex. Sarkozy et Bruni-Sarkozy. Il y a aussi une date du mariage.

b. On suppose alternativement qu'il pourrait s'agir d'un mariage polygame usuel. Puis, inversement, comme chez certains peuples. Enfin, des amitiés multigames. Montrez les schémas optimaux.

c. On suppose enfin que l'on pourrait avoir, en plus de mariages comme dans (a), des hommes ou des femmes seul(e)s. Si jamais la démarche usuelle ne suffisait pas à ce cas particulier, alors proposez une solution spécifique de bon sens quand même.

Suite et fin d'une définition de schéma relationnel

Il faut se rappeler que la décomposition ne crée que les nœuds du graphe. Il faut encore créer les arcs. Puis, il faut préciser pour chaque arc si c'est un lien ou une contrainte référentielle. Sous Ms Access et tout autre SGBD le permettant, il est utile de demander la propagation de MAJs en cascade pour ces dernières. Idem enfin, il faut penser aux jointures implicites. Soit accepter celle internes proposées par défaut, soit choisir pour certaines des jointures externes. Ces dernières peuvent être surtout utiles quand on a décomposé des DM. Par exemple pour U (E#, Tel, Email…) quand des étudiants n'ont aucun n^{o} de tél ou aucun email. Revoir le cours pour + sur les bons choix correspondants.

Vos liens référentiels, contraintes d'intégrité et jointures implicites pour tout exemple ci-dessus ?