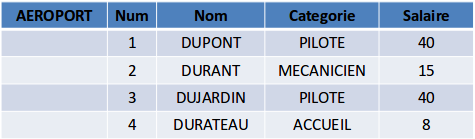
**Dépendances fonctionnelles**

Introduction

Il existe de nombreuses méthodes pour concevoir une base :

* Empirique (DANGER)
* Semi empirique (ex : entité-association)
* Formelle (ex : normalisation relationnelle)

Objectif : trouver un bon schéma relationnel. La qualité d’un schéma se mesure lors des opérations de mises à jour

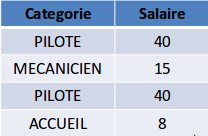
Hypothèse : la catégorie détermine le salaire

* Anomalies de modification : modification du salaire des pilotes : autant de modifications qu’il y a de pilotes
* Anomalies d’insertions : pour stocker le salaire des contrôleurs il faut qu’il y ait au moins une catégorie d’employés dans cette catégorie (pas de clé primaire nulle)
* Anomalies de suppression : si suppression de DURANT, on perd l’information sur le salaire des mécaniciens.

L’objectif est d’éliminer ces anomalies pour obtenir un bon schéma relationnel. La solution : normaliser la relation en la décomposant en plusieurs relations. Il faut répondre aux questions suivantes : s’il y a redondance : comment décomposer la relation ? Y a t’il de l’information perdue lors de la décomposition ? Pour y répondre : les dépendances fonctionnelles

Dépendances fonctionnelles

Soit R(U) une relation avec U l’ensemble des attributs. Soit X, Y ⊂ u, i.e. X et Y sont deux attributs ou ensembles d'attributs de R. Il existe une dépendances fonctionnelle (DF) entre C et Y notée X → Y si et seulement si ∀ t1, t2 ∊ R. si t1(X) = t2(X) alors t1(Y) = t2(Y). Les dépendances fonctionnelles sont des propriétés sémantiques (du schéma de R), donc s’appliquent à toute extension de R.



Dans notre exemple, AEROPORT(num, nom, categorie,salaire), on a categorie → salaire.

∀ t1, t2 projetés sur categorie et salaire, nous avons : si t1(categorie) = t2(categorie) alors t1(salaire) = t2(salaire)

pilote → 40

mécanicien → 15

accueil → 8

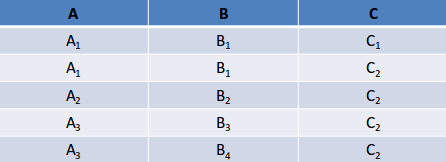
Dans une relation tout attribut est en DF avec la clé primaire

num → nom (à tout numéro correspondant un nom)

num → categorie (à tout numéro correspondant une catégorie)

num → salaire (à tout numéro correspondant un salaire)

Vocabulaire: num détermine nom, catégorie, salaire. Nom est déterminé par num. Catégorie est déterminé par num. Salaire est déterminé par num.

Exemple : soit l'extension suivante de R. quels sont les DF dans R ? 

A→B non car A3 donne B3 et B4

A→C non car A1 donne C1 et C2

B→A oui

B→C non car B1 donne C1 et C2

C→A non car A différents pour C2

C→B non car B différents pour C2

Propriétés des DF

Axiomes d’Amstrong :

P1 : Réflexivité. Si Y ⊆ C alors X→Y (donc Y→X). ex : si A,B→A,B alors A,B→A et A,B→B

P2 : Augmentation. si X→Y alors X,Z→Y,Z ou Z ⊆ C. ex : si A,B→C, alors A,B,C→C

P3 : Transitivité. Si X→Y et Y→Z alors X→Z. ex : si A,B→C et C→D alors A,B→D

Règles d’inférences déduites des axiomes d’Amstrong

P4 : Pseudo-transitivité. Si X→Y et Y,Z→W alors X,Z→W

P5 : Union. Si X→Y et X→Z alors X→Y,Z

P6 : Décomposition.Si X→Y,Z alors X→Y et X→Z

Exemple : soit R(A,B,C,D,E,F,G,H) et F={A,B→C ; B→D ; C,D→E ; G→A ; D→H}. Avec les axiomesd’Amstrong, nous avons :

B→H car B→D et D→H (P3)

B,G→C car G→A et par augmentation (P2) on a B,G→A,B. De plus, A,B→C donc par transitivité (P3) on a B,G→C

A,B→E car B→D et par augmentation (P2) on a A,B→A,D donc par décomposition (P6) on a A,B→D or nous savons que A,B→C par union (P5) on a A,B→C,D et nous savons que C,D→E donc par transitivité (P3) on obtient A,B→E

Fermeture d’un ensemble de DF

Soit F un ensemble de dépendances fonctionnelles sur R(U). Soit X→Y une DF. F implique X→Y, noté : F ⊨X→Y, signifie que toute instance de relation sur R qui satisfait les dépendances dans F satisfait aussi X→Y. Exemple : soit F={B→D ; D→H} sur R(A,B,C,D,E,F,G,H). Soit la dépendance fonctionnelle B→H. F ⊨B→H

Fermeture transitive

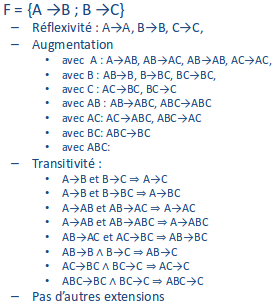
La fermeture transitive (ou clôture) d’un ensemble de dépendances fonctionnelles, F, est ce même ensemble enrichi (F+) de toutes les dépendances fonctionnelles que l’ont peut dériver en appliquant les axiomes d’Amstrong. En d’autres termes, F+ contient toutes les DF impliquées par F : F+ = {X→Y | F ⊨X→Y}

Algorithme de calcul de fermeture des DF

Entrée : F un ensemble de DF

Sortie : Fermeture F+ de F

Algorithme :

1. F+ = F
2. Répéter pour chaque DF f dans F+ :
   1. Appliquer les règles P1 (réflexivité) et P2 (augmentation) sur F+
   2. ajouter les DF résultats à F+
   3. Pour chaque paire de DF f1 et f2 dans F+ :
      1. Si f1 et f2 peuvent être combinésen utilisant la règle P3 (transitivité) ajouter la DF résultat dans F+
   4. jusqu’à ce que F+ ne change plus
3. Retourner F+

Un exemple :

La fermeture contient trop de solutions triviales : calcul des fermetures d'ensembles d’attributs

Fermeture transitive d’attributs

La fermeture transitive d’un ensemble d’attributs X est notée X+. elle contient tous les attributs qui dépendent des attributs dans X.

[X]+F = {A|F|=X→A}

F = {A→B ; B→C}, R=(A,B,C)

[A]+F = {A,B,C}

A partir de A, on peut déterminer l’ensemble des attributs : A→C est dans la fermeture de F (F+)

A→C est dans la fermeture de F

A est une surclé (et clé) de R

Calcul de la fermeture transitive d’un ensemble d’attributs

Entrée : F un ensemble de DF et X un ensemble d’attributs.

Sortie : X+ fermeture transitive de X

Algorithme :

1. Initialise X+ à X
2. Trouver les DF (G→D) ∊ F possédant en partie gauche des attributs inclus dans X+
3. Ajouter dans X+ les attributs situés en partie droite de la DF
4. Répéter 2. et 3.jusqu’à ce que X+ ne puisse plus évoluer

Exemple

Soit F={A→D ; A,B→E ; B,I→E ; C,D→I ; E→C} Calculer la fermeture sous F de AE.

**étape 1** : Initialisation de AE+ avec AE. AE+ = {A,E}

**étape 2** : trouver une DF(G→D) ∊ F possédant en partie gauche des attributs inclus dans X+. Ici, A est inclus dans AE+. Comme A→D, nous pouvons ajouter D à l’ensemble : AE+ = {A,D,E}.

**étape 3** : Ajouter dans X+ les attributs situés en partie droite de la DF. Ici C. Comme E→C, nous pouvons ajouter C : AE+ = {A,C,D,E}

**étape 4** : On peut répéter l’étape 2 et 3 car X+ à été étendu, Ici I. Comme C,D sont dans AE+ et que nous avons C,D→I, alors nous pouvons ajouter I : AE+ = {A,C,D,E,I}

**fin** : pas d’évolution possible. LA fermeture transitive de AE est donc : AE+ = {A,C,D,E,I}

Un autre exemple

F={A→C ; A→D ; B,C→A ; E→B ; E→D} Calculer la fermeture sous F de CE.

On trouve CE+ = {A,B,C,D,E}

Equivalence d’ensemble de DF

Deux ensembles de DF différents peuvent exprimer les mêmes contraintes :

F = {A→B ; B→C ; A→C ; A→D}

G = {A→B ; B→C ; A,B→D}

F et G sont équivalents (expriment les mêmes contraintes). On note : F ☰ G. G est plus “compacte”

Equivalence

F est équivalent à G ( F ☰ G ) ssi F+ = G+. Pour vérifié si F et G sont équivalents :

1. (G ⊃ F ?) Pour chaque df, X→Z dans F: vérifier si X→Z est dans G+ (calculer X+ par rapport à G et vérifier si Z ⊆ X+)
2. (F ⊃ G ?) De facon similaire, pour chaque df X→Z dans G : vérifier si X→Z est dans F+.
3. Si G ⊃ F et F ⊃ G alors F ☰ G

Exemple

F = {A→B ; B→C ; A→C ; A→D} et G = { A→B ; B→C ; A,B→D}

**Etape 1 :vérifier si toutes les DF de G sont présentes dans F**

* A→B et B→C de G sont présentes dans F.
* A,B→D est dans G mais pas dans F : pour F, [AB]+ = {A,B,C,D}. AB détermine fonctionnellement A,B,C et D, donc A,B→D est valable dans F.

Comme toutes les DF de G sont valables pour F, F ⊃ G est vrai.

**Etape 2 :vérifier si toutes les DF de F sont présentes dans G**

* A→B et B→C de F sont présentes dans G.
* A→C est dans F mais pas dans G : pour G, [A]+ = {A,B,C,D}. A→C valable dans G.
* A→D est dans F mais pas dans G : pour G, [A]+ = {A,B,C,D}. A→D valable dans G.

Comme toutes les DF de F sont valables pour G, G ⊃ F est vrai.

**Etape 3 :** comme F ⊃ G et F ⊃ G sont tous deux vrais, alors F ☰ G est vrai.

Dépendance fonctionnelle élémentaire

Soit R(U) une relation, soit X et Y ⊂ U, tels que : X→Y. La dépendance fonctionnelle X→Y est dite élémentaire (ou totale) ssi :

* Y n’est pas inclus dans X, i.e Y=U-X (Y est le complémentaire de X dans U)
* il n’existe pas X’ ⊂ X tel que X→Y

La seconde condition indique que X est “la plus petite quantité d’information donnant Y”. “Il n’y a pas d’attribut inutile dans la partie gauche”.

Dépendance fonctionnelle direct

Soit R(U) une relation, soit X et Y ⊂ U, tels que : X→Y. La dépendance X→Y est directe s’il n'existe pas Z dans R distinct de X et Y tel que X→Z et Z→Y. La dépendance n’est pas obtenue par transitivité.

Dépendance fonctionnelle triviale et simple

Soit R(U) une relation, soit X et Y⊂U, tels que : X→Y. La dépendance X→Y est triviale si Y⊂X est vide. Nom, Numéro→Nom.

Une dépendance fonctionnelle est simple si elle ne comporte qu’un seul attribut en partie droite et si elle n’est pas triviale. X→A1, A2,...,An ⇔ {X→A1 ; X→A2 ; … ; X→An }

Il est toujours possible de présenter les dépendances fonctionnelles sous forme simple (P6)

Couverture minimale

Couverture minimale d’un ensemble de DF : sous ensemble minimum de dépendances fonctionnelles élémentaires qui permettent de générer toutes les autres. Tout ensemble de dépendances fonctionnelles possède une couverture minimale (pas forcément unique).

Algorithme pour couverture minimale

Entrée : F un ensemble de DF et X un ensemble d’attributs

Sortie : M : la couverture minimale de F

Algo :

1. Décomposer chaque DF pour avoir un seul attribut à droite (P6). (les cibles de DF n’ont qu’un attribut - DF simples) - (Right Hand Side - RHS)
2. Supprimer les attributs en surnombre à gauche : pour tout X→Y, s’il existe un Z⊆X tel que Z→Y, alors remplacer X→Y par Z→Y (propriétés P1, P3 et P4) (pas d’attribut inutile dans les DF de M - DF élémentaires) - (“Left Hand Side - LHS”)
3. Supprimer les DF redondantes (qu’on peut obtenir pas les axiomes d’Amstrong -DF directes)

Recherche DF élémentaires

Soit F un ensemble de DF, soit f∊F, la DF A,B,C,D,... →Y. A est un attribut inutile dans f si on peut engendrer B,C,D … → Y à partir des DF de F et des propriétés P1, P3 et P4 (réflexivité, transitivité, pseudo-transitivité)

Exemple : F = {A→B ; A,B→C} B est inutile dans A,B →C car A→C peut être généré

P4 : X→Y Y,Z→W alors X,Z→W

A→B B,A→C alors A,A→C

et donc A→C

Dépendances fonctionnelles et clés

Une clé d’une relation R(A1, … ,An) est un sous ensemble X des attributs de la relation F tel que les deux conditions ci-dessous sont vérifiés :

* X→A1, … , An
* Il n’existe pas de Y ⊂ X tel que Y→A1, … ,An

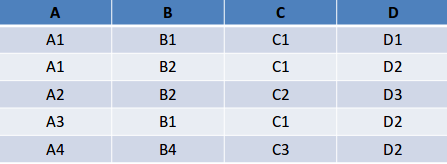
Un attribut clé est un attribut qui appartient à cette clé et un attribut non clé est un attribut qui n’y appartient pas. Super-clé : tout ensemble d’attributs satisfaisant la 1ère propriété constitue une super clé de R.

* Une super clé de R contient donc une clé de R
* Une clé de R est une super clé minimale de R
* Si X=U, la relation est dite “toute clé” : la clé est composée de l’ensemble des attributs

Exemple :

L’un des attributs peut-il jouer le rôle de clé? Quelles associations d’attributs peuvent jouer ce rôle ?

A ne peut pas car A1 détermine plusieurs B (B1, B2)

B ne peut pas car B1 détermine plusieurs A (A1, A3)

C ne peut pas car C1 détermine plusieurs B (B1, B2)

D ne peut pas car D2 détermine plusieurs C (C1, C3)

A,B oui car pas deux fois la même occurrence

A,C non car A1, C1 détermine plusieurs B (B1, B2)

A,D oui car pas deux fois la même occurrence

B,C non car B1, C1 détermine plusieurs A (A1, A3)

B, D oui car pas deux fois la même occurrence

C,D non car C1, D2 détermine plusieurs A (A1, A3)

A,B,C oui car pas deux fois la même occurrence

B,C, D oui car pas deux fois la même occurrence

A,B,C,D oui car pas deux fois la même occurrence ABCD est une super clé

Exemple intuitif

Soit la relation R(A,B,C,D,E) et F={A→C ; A→D ; B,C→A ; E→B ; E→D}. Donner la liste des clés candidates de R.

E doit faire partie de la clé parce qu’il n’apparaît jamais dans une partie droite des DF. Comme A, B, C et D interviennent à droite des DF, il est inutile de calculer A+, B+, C+ et D+. Une clé candidate doit avoir comme fermeture {A,B,C,D,E}. Calcul de la fermeture de E+ = {B,D,E} E ne peux donc pas être seul clé. Il faut ajouter les combinaison d’attributs avec E : Calcul des fermetures de AE+, BE+, CE+ et DE+.

Ne vérifient pas la fermeture : BE+ = {B,D,E} et DE+ = {B,D,E}

Vérifient la fermeture : AE+ = {A,B,C,D,E} et CE+ = {A,B,C,D,E}

Les clés candidates sont : AE+ et CE+

Recherche des clés

Entrée : ensemble des attributs et ensemble des DF

Sortie : ensemble de relations avec leur clés

Algo :

1. Recherche de la couverture minimale M
2. Regroupement des DF de M : réunir dans un même ensemble Ei toutes les DF ayant même source (autant de Ei que de source de DF différentes)
3. regroupement des Ei : on regroupe dans un même ensemble les DF de Ei et Ej s’ils contiennent des DF réciproques (X→Y et Y→X)
4. Création des relations : réunir dans un même ensemble Ei toutes les DF ayant même source (autant de Ei que de source de DF différentes)

Exemple

Soit {A,B,C,D,E,F} un ensemble d’attributs. L’ensemble des DF est composé de : F = { f1 : A,B→C,D ; f2 : C→D ; f3 : E→D ; f4 : F→E,D ; f5 : B→A ; f6 : E,F→F ; f7 : D→E}

Etape 1 : M = {g’1 : B→C ; f2 : C→D ; f3 : E→D ; g’4 : F→E ; f5 : B→A ; f7 : D→E}

Etape 2 : E1 = {B→C ; B→A} E2 = {C→D} E3 = {E→D}

E4 = {F→E} E5 = {D→E}

Etape 3 : E1 ; E2 ; E’3 = E3 E5 ; E4

Etape 4 : R1(B,*C*,A) ; R2(C,D) ; R3(E,D) D clé candidate ; R4(F,*E*)

Retour sur la couverture minimale

A partir de la couverture minimale il est possible de déterminer des relations qui constituent un bon schéma relationnel. Si la couverture minimale n’est pas minimale, il peut y avoir duplication d’informations donc anomalies en opérations de mise à jour.

Vérifier que la couverture est minimale

Remarque : Le calcul de la fermeture transitive d’un attribut est simple à obtenir. L’application des axiomes d'Armstrong est simple à faire mais… En appliquant les axiomes d’Armstrong peut-on être certain que l’on en a pas oublié ? En fait : l’application des axiomes et le calcul de la fermeture est la même chose !

Utilisation de la fermeture

Entrée : F un ensemble de DF et X un ensemble d’attributs

Sortie : M : la couverture minimale de F

Algo :

1. décomposer chaque DF pour avoir un seul attribut à droite (P5). (*les cibles de DF n’ont qu’un attribut. DF simple*)
2. Supprimer les attributs en surnombre à gauche : Pour tout X→Y, s’il existe un Z⊆X tel que Z→Y, alors remplacer X→Y par Z→Y (propriétés P1, P3 et P4) (*pas d’attributs inutile dans les DF de M - DF élémentaires)*.
3. Supprimer les DF redondantes (qu’on peut obtenir par les axiomes d’Armstrong - DF directes)

Les attributs inutiles

Intuition : F = {A→b ; A,B→C}. Si B est inutile, cela revient à dire que sans lui, je peux déterminer l’attribut C donc A→B ; A→C. Si A est inutile, cela revient à dire que sans lui, je peux déterminer l’attribut C donc A→B ; B→C.

On regarde A,B→C. On calcule les fermetures pour tous les sous-ensembles : ici [B]+ et [A]+ sur F.

[B]+F = {B} On ne peut pas l’étendre plus

[A]+F = {A} avec A->B on passe ) {A,B}. Puis avec A,B→C on obtient [A]+F = {A,B,C}

A partir de A on peut directement obtenir C, donc B est inutile.

Les dépendances inutiles

Intuition : A→B ; B→C ; A→C. Si une DF est inutiles, alors en la supprimant on ne doit pas perdre d’information.

* Sans A→B, puis-je obtenir B à partir de A ? [A]+ = {A,C}. Je n’ai pas B donc A→B est utile
* Sans B→C, puis-je obtenir C à partir de B ? [B]+ = {B}. Je n’ai pas C donc B→C est utile
* Sans A→C, puis-je obtenir C à partir de A ? [A]+ = {A,B,C}. J’ai un C donc A→C est inutile

Couverture minimale via les fermetures des attributs

Entrée : F : un ensemble de DF et X un ensemble d’attributs

Sortie : M : la couverture minimale de F

Algo :

1. Décomposer chaque DF pour avoir un seul attribut à droite (P6). (*les cibles de DF n’ont qu’un attribut - DF simples*)
2. Elimination des attributs étrangers dans la partie gauche. Pour chaque DF,