

Chapitre 1 4 et 15.1 – Objectifs

|9:0| : Quand Cole a inventé le modèle relationnel, c’est des relations normalisé, Qu’est-ce que la normalisation?

* But de la normalisation
* Problèmes potentiels liées aux données redondantes
* Dépendance fonctionnelle (DF) décrit associations entre attributs
* Caractéristiques des DF utilisées dans la normalisation
* Identifier les DF dans une relation donnée
* Identifier les formes normales les plus utilisées : la première (1NF), la deuxième (2NF) et la troisième (3NF) formes normales
* Utiliser les règles d’inférence pour identifier l’ensemble de toutes les DF d’une relation
* Utiliser les axiomes d’Armstrong pour identifier un ensemble minimal de DF utiles à partir de l’ensemble de touts les DF dans une relation.

Normalisation

Objectif modèle \_\_\_logique\_\_\_\_ données

Créer représentation exhaustive

|9:6| : On veut sa représente les données,

|9:6| : On veut aussi tout les liens entre ces données la

|9:6| : Et on veut avoir aussi les contraintes.



Identifier ensemble approprié \_\_\_relations\_\_\_\_

|9:7| : On veut avoir la bonne liste des relation, relation étant une table ici, relation filiale, personnel etc. LE no de l’employé est en relation avec le prenom de l’employé le nom son no de telephone.

Formes normales + utilisées -- \_\_\_1NF, 2NF, 3NF\_\_\_\_

|9:7| : C’est le processus de normalisation, on applique la 1ere forme normale 2e 3e

|9:8| : Pour appliquer les étapes ils faut toutes les étapes une apres l’autre.

Basées sur (\_\_\_dependance fonctionelle , DF\_\_\_\_ ) entre attributs relation

|9:8| : On va regarder tous les attributs et on va chercher quel attribut détermine les autres, on a déjà parlé de superclé clé candidate. Le no employé détermine le prenom nom.

|9:8| : Quand on dit détermine, comme le no assurance social correspond a un courriel etc sa détermine a une dépendance fonctionnel. La détermination, c’est une dépendance. Le nom de la personne dépend de son numéro d’assurance social. Si je vous donne un NAS je suis capable de retrouver les autres valeurs. Les dépendants.

Prévenir

|9:9| : On veut prevenir des anomalies de mise à jours. Grace à la normalisation.

|9:10| : Sa c’est provqué par la redondance des données

Redondance de données

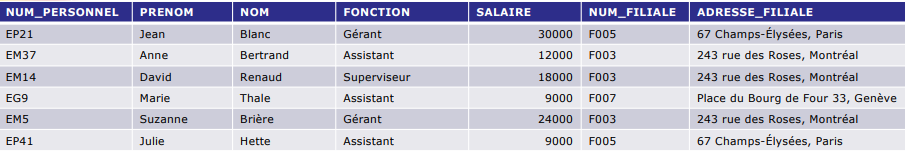
But majeur conception BD relationnelle

Grouper attributs relations minimisent \_\_\_redondance\_\_\_\_

|9:10| : On ne veut pas de donnée qui se répète.

Relation PERSONNEL\_FILIALE contient données redondantes

Lesquelles?



|9:11| : Pourquoi on aurais des anomalies de mise à jour dans cette table

|9:12| : L’adresse se répète inutilement.

|9:12| : Le no filiale va faire partit du minimum de redondance nécessaire. On verra pourquoi plus tard.

|9:15| : Il y a 3 anomalies de mise à jour possibles :

|9:15| : la première insertion

|9:16| : suppression

|9:16| : Modification.

|9:16| : Si j’insère un employé il faut que jaille absolument la bonne adresse de filiale

|9:16| : Si je veux crée une nouvelle filiale, tant que j’aurais pas d’employé associé ce sera difficile.

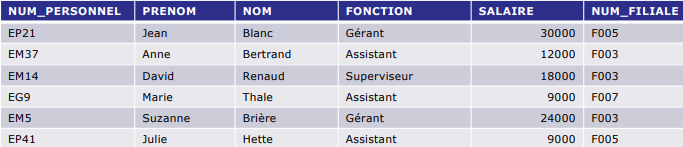
|9:17| : Anomalie pour les suppression, quest-ce qui arrive si je supprime l’employé Marie Thale? On perd la filiale 7.

|9:17| : Anomalie de modification. Si je change un employé de succursale je change son numéro de filiale, il faut aussi que je change son adresse!

|9:19| : insertion, suppression, modification (répété environ 5 fois).

Redondance de données

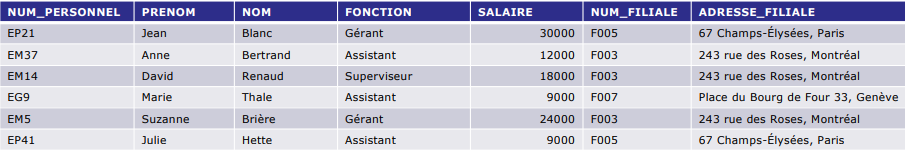
Relation PERSONNEL



Relation FILIALE



Anomalies de mise à jour



Dépendance fonctionnelle

Concept \_\_\_principal\_\_\_ normalisation

Décrit \_\_\_liens\_\_\_ entre attributs relation

|9:34| : Le NAS est en relation avec son nom prenom etc.

A et B attributs relation R

|9:35| : A et B peuvent etre plusieurs attributs, et B peut être un groupe de 1 ou plusieurs attriuts

B est dépendant fonctionnel de A

|9:35| : On va le noté A fleche B. B dépend de A.

|9:35| : Pourquoi on dit que B depend A ou A détermine B

|9:35| : Pour chaque valeur de A je n’ai qu’une seule valeur de B correspondante

|9:35| : Une dépendance fonctionnel a lieu si et seulement si pour chaque propriété de A je trouve une seule valeur de B. (ACSI!)

Noté A → B

|9:37| : y = f(x). On dit que c’est une fonction pour une valeur de x j’ai une seule valeur d’y.

Dépendance fonctionnelle

|9:39| : Un ensemble A détermine un ensemble B ou B est dépendant de A

|9:39| : A fleche b.



Caractéristiques principales DF pour normalisation

|9:40| : Lien 1 pour 1. Important Flche droite.

Lien \_\_1\_→ 1\_\_ entre attributs gauche vers droit

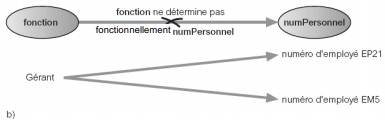


Valide en tout \_\_\_temps\_\_\_

|9:40| : C’est valide maintenant, avant et après.

|9:41| : Il faut pas ex dire que le nom de famille de la personne détermine l’employé, ex Tremblay détermine l’Employé. Il ne sera pas possible d’engager d’autre personne avec le nom de famille tremblay.

|9:42| : Est-ce que un employé peut travailler dans plusieurs filiale? Sa change les dépendances fonctionnel.



Dépendance fonctionnelle complète

|9:47| : Il y a deux type de dépendances,

|9:47| : dépendance complète

|9:47| : Quand mon déterminant, donc quand mon déterminant est unique, quand du coté gauche de ma flèche il a un seul attribut, c’est implicitement une dépendance fonctionnel complète.

|9:48| : Lorsque quil y a deux attribut du coté gauche on peut avoir des dépendance fonctionnel partielle.

DF \_\_\_complète\_\_\_

|9:45| : Variation dans les dépendances fonctionnel,

A et B 2 ensembles 1 + attributs

A → B

|9:48| : Il le ne faut pas que B dépend d’un sous-ensemble de A mais seulement d’un attribut.

B ne dépend d’aucun \_\_\_sous-ensemble\_\_\_ de A

Sinon dépendance \_\_\_partielle\_\_\_ (DP)

Par exemple :

R(a, b, c, d) avec a, b → c, d où

|9:49| : Les deux premier attributs lorsque que je les met ensemble sa détermine les autres.

A = (a, b) et B = (c, d)

Donc A → B

Si DF b → c alors

b → c est \_\_\_DP\_\_\_ de A → B

Dépendance fonctionnelle transitive

DF \_\_\_transitive\_\_\_

A, B et C 3 ensembles 1.. n attributs tels que

|9:52| : Sa prend 3 attributs au moins pour qu’il une dépendance transitive.

A → B

B → C

\_\_\_A → C\_\_\_

|9:52| : Ce n’est pas suffisant pour que déterminer une transitivité

\_\_\_transitif\_\_\_ sauf si

|9:53| : Il ne faut pas qu’il y ait une boucle, Il faut pas qu’il ait de retour en arrière. Il faut pas que B détermine A ou C détermine A.

B → A ou C → A

Exercices

Que veut prévenir la normalisation?

A-DF B-Anomalies de mise à jour C-Relation D-Transitivité

E-Formes normales F-Toutes ces réponses G-Aucune de ces réponses

Quel est le concept principal de la normalisation?

A-DF B-Formes normales C-Unicité D-Transitivité

E-Toutes ces réponses F-Aucune de ces réponses

Qu’est-ce qu’une DF partielle?

A-DF dont tous les attributs de droite déterminent les attributs de gauche

B-DF dont les attributs de droite sont tous partiels

C-DF dont les attributs de gauche font partie des déterminants d’une autre DF

D-DF dont tous les déterminants déterminent les déterminés

E-DF dont les attributs de gauche déterminentn partiellement les attributs de droite

F-Aucune de ces réponses

Identifier les dépendance fonctionnelles

Identification toutes DF assez simple \_\_\_si\_\_\_ bien compris

\_\_\_signification\_\_\_ attributs

|10:0| : établir les besoins en information : le nom, le prénom? oui

|10:1| : Connaître la définition de chaque attribut.

\_\_\_Liens\_\_\_ attributs

|10:3| : Les associations.

Informations fournies par

|10:3| : L’entreprise / Organisation

|10:4| : Fait par des rencontre, questionnaires.

|10:4| : Documentation en ligne papier.

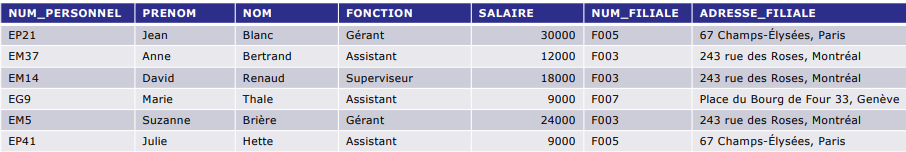


Information \_\_\_Incomplète\_\_\_

|10:5| : Le spécialiste va devoir se fier a son expérience à ce moment (lorsque l’information est incomplète)



Identifier les dépendances fonctionnelles



\_\_\_NUM\_PERSONNEL\_\_\_ → PRENOM, NOM, FONCTION, SALAIRE, NUM\_FILIALE, ADRESSE\_FILIALE

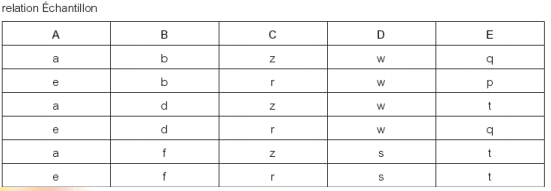
NUM\_FILIALE → ADRESSE\_FILIALE

\_\_\_ADRESSE\_FILIALE\_\_\_ → NUM\_FILIALE

NUM\_FILIALE, FONCTION → SALAIRE

\_\_\_ADRESSE\_FILIALE, FONCTION\_\_\_ → SALAIRE

Identifier les dépendances fonctionnelles à partir d’un échantillon



A → C (fd1)

C → A (fd2)

B → D (fd3)

A, B → E (fd4)

Identifier la clé primaire

Identification DF?

Spécifier ensemble \_\_\_contraintes\_\_\_ d’intégrité

|10:15| : Les Dependance fonctionnelle va obliger les données, par exemple on peu pas avoir deux nom de famille pour un employé. C’est une contrainte.

Identification CK

|10:15| : Clé candidate, Candidate Key.

Contraintes d’intégrité importantes

|10:15| : Trop souvent les clé candidate sont oublié.

|10:16| : C’est important sa rajoute des contraintes d’intégrité

Choisir ensuite clé \_\_\_primaire\_\_\_

Déterminer clé(s) candidates(s)

Repérer attributs(s) identifie manière \_\_\_unique\_\_\_ tous tuples relation.

|10:16| : Des groupes d’attribut peuvent devenir des clés candidate.

Identifier la clé primaire

\_\_\_NUM\_PERSONNEL\_\_\_ → PRENOM, NOM, FONCTION, SALAIRE, NUM\_FILIALE, ADRESSE\_FILIALE

NUM\_FILIALE → ADRESSE\_FILIALE

\_\_\_ADRESSE\_FILIALE\_\_\_ → NUM\_FILIALE

NUM\_FILIALE, FONCTION → SALAIRE

\_\_\_ADRESSE\_FILIALE, FONCTION\_\_ → SALAIRE

Déterminants

|10:17| : Les attributs a gauche.

NUM\_PERSONNEL

NUM\_FILIALE

ADRESSE\_FILIALE

NUM\_FILIALE, FONCTION

ADRESSE\_FILIALE, FONCTION

|10:17| : Lesquels détermine les autres attributs de toutes les relations.

|10:18| : La seule qui défini tout c’est NUM\_PERSONEL.

|10:18| : Celle-ci deviens automatiquement la clé primaire.

Le processuss de normalisation

|10:19| : C’est formelle c’est structurée, quand vous savez pu quoi faire, revenez au processus de normalisation.

Technique \_\_\_formelle\_\_\_ pour analyser relation basée sur

PK

|10:20| : Analyer les clé primaire, les Dependance fonctionnel.

DF ente attributs

Souvent \_\_\_executé\_\_\_ comme série étapes

|10:20| : 1Nf, 2NF, 3NF, BCNF.

Chaque étape correspond à 1 \_\_\_FN\_\_\_

|10:20| : Forme normale.

Normalisation progresse

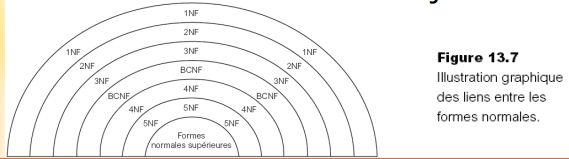
Relation + \_\_\_restreinte\_\_\_ (+ forte) en format

|10:20| : Le données sont de plus en plus valide, de plus en plus struturée.

|10:21| : Quand on fait du transactionnel c’est tres bon d’avoir un modèle restreint fort.

* Vulnérable Anomalies maj

|10:21| : plus on normalise moin on est vulnérable aux anomalies de mise à jour.



Forme non normalisée (UNF) et (1NF)

|8:32| : quand on veut normaliser on peut prendre une approche descendante ou ascendante,

|8:33| : Dans notre cas on pars de la liste d’attribut et appliquer les formes.

UNF

Table contient groupes (s \_\_\_répétitif(s)\_\_\_ )

Créer table non normalisée

Transformer données source information (formulaire \_\_\_papier\_\_\_ )

→ 1 table avec colonnes et rangées

1NF

Relation dont chaque \_\_\_cellule\_\_\_ contient 1..1 valeur.

|8:37| : Coté atomique, 1 valeur par cellule.

UNF vers 1NF

Identifier \_\_\_attributs\_\_\_

|8:39| : sa peut être des attributs explicitement nommé comme sa peut être des attributs sous-entendu ou nécessaire.

Identifier \_\_\_DFs\_\_\_

Choisir attribut(s) pour agir comme \_\_\_clé\_\_\_ table non normalisée

|8:40| : souvent a cette étape si cette DF est composé de deux ou trois autres DF.

Identifier groupes(s) répétitif(s) qui se répète pour attribut(s) clé

\_\_\_Enlever\_\_\_ groupes répétitifs

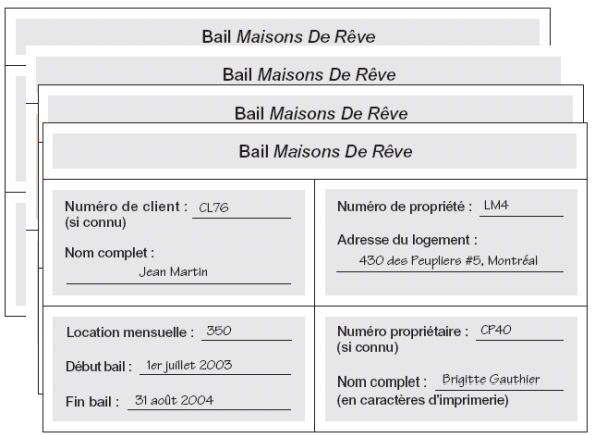
Entrer données dans colonnes vides rangées contenant données répétitives (‘mise a \_\_\_plat\_\_\_ ‘ de la table – fait dans le cours)

Ou

~~Déplacer données répétitives avec copie clé dans relation distincte.~~

|8:41| : c’est mieux la formule de mise à plat. En commençant (1 valeur par cellule)

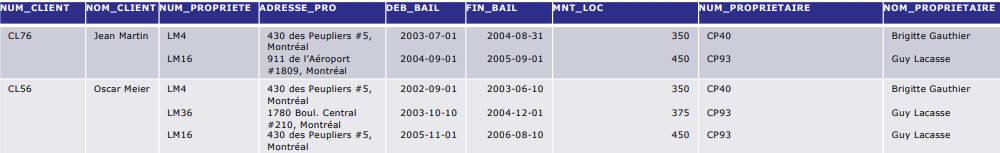
Normalisation de UNF a 1NF



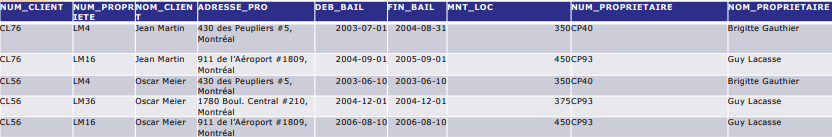
Normalisation de UNF a 1NF – Mise a plat

CLIENT\_BAIL \_\_\_UNF\_\_\_

|8:43| : On veut éviter de répéter plusieurs valeur dans la même cellule



CLIENT\_BAIL \_\_\_1NF\_\_\_



|8:43| : Il y a eu une mise à plat mais cela crée de la redondance.

|8:44| : Dans de grand modèle on travail juste avec la liste des attributs.

Seconde forme normale (2NF)

|8:48| : on passe à la 2NF après avoir choisi la première clé primaire.

Basée sur concept DF \_\_\_complète\_\_\_

|8:48| : On ne veut que des DF complète. On compare les DFS pour ne pas avoir de partielle.

A et B 2 ensemble de 1+ attributs

A → B

B ne dépend d’aucun \_\_\_sous-ensemble\_\_\_ de A

Dépendance \_\_\_Partielle (DP)\_\_\_

a, b → c, d

b → c \_\_\_DP\_\_\_

|8:49| : Quand on dit Dépendance partielle c’est toujours par rapport a une autre dépendance.

2NF

Relation 1NF

Chaque attribut non PK est en DF \_\_\_complète\_\_\_ avec PK

→ Aucune \_\_\_DP\_\_\_ sur la PK.

|8:50| : Dans une table si ma clé primaire n’est pas composé de 2 attributs, il est certain que la PK n’a pas de Dépendance partielle

1NF vers 2NF

1, identifier les DF

2, identifier \_\_\_PK\_\_\_ de relation 1NF

|8:51| : Une grosse table 1 NF, ou on identifie une PK.

3, Si \_\_\_DP\_\_\_ sur PK

|8:54| : (Dépendance partielle)

Déplacer déterminer dans nouvelle relation avec copie \_\_\_déterminant\_\_\_

|8:54| : On les met dans une nouvelle table.

Créer nouvelle relation avec la DP

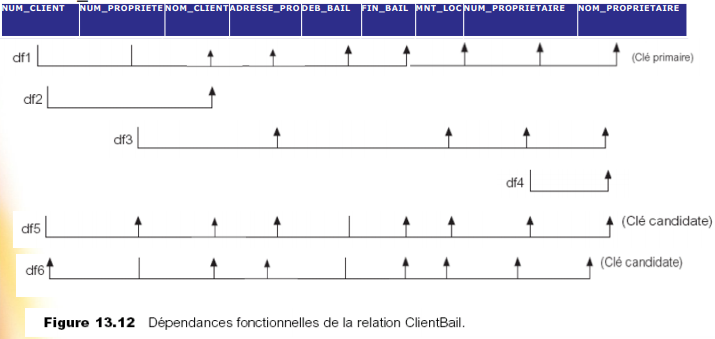
Enlevé attributs déterminés de relation 1NF

|8:55| : On enlève le c de la relation 1NF et copie b dans l’autre.

4, Recommencer avec les tables créées

1NF vers 2NF

CLIENT\_BAIL



|9:4| : Ce n’est pas les DF qui sont clé primaire mais leurs déterminants.

1NF vers 2NF

1, Identifier DF → df1 a df6 page précédente

2, PK = déterminant de la df1 (donc AK …)

|9:5| : Alternative Key.

3, Identifier DP sur la PK

|9:6| : Dépendance partielle)

df2 est DP sur la PK

(Créer une) Nouvelle relation CLIENT avec df2

\_\_\_Sort\_\_\_ NOM\_CLIENT

\_\_\_Copie\_\_\_ NUM\_CLIENT dans BAIL

Df3 est DP sur la PK

Nouvelle relation PROPRIETAIRE avec df3

\_\_\_Sort\_\_\_ ADRESSE\_PRO, MTN\_LOC, NUM\_PROPRIETAIRE, NOM\_PROPRIETAIRE

\_\_\_Copie\_\_\_ NUM\_PROPRIETE dans BAIL.

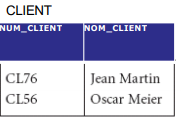
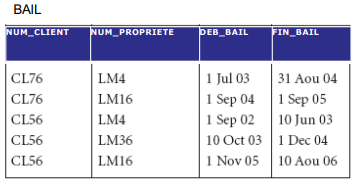
1NF vers 2NF

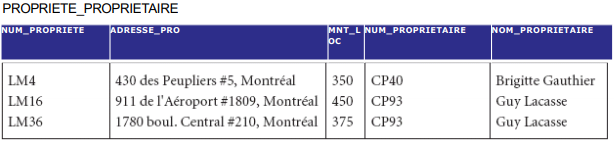
|9:8| : Bail est le restant de la DF1

|9:8| : Client est la DF2

|9:8| : PROPRIÉTÉ et Propriétaire est la DF3.

|9:9| : Num Client est devenu une clé étrangère dans CLIENT.



Troisième forme normale (3NF)

Basée sur le concept de \_\_\_DT\_\_\_

|9:11| : Dépendance Transitive

|9:11| : On veut au moins 3 attributs pour faire une dépendance transitive.

A, B et C 3 ensembles de 1+ attributs tels que

A → B

B → C

\_\_\_A --­» C\_\_\_

Sauf

~~B → A ou C → A~~

3NF

Relation \_\_\_1NF\_\_\_ et en \_\_\_2NF\_\_\_

Aucun attribut non PK est en \_\_\_DT\_\_\_ sur la PK.

|9:12| : Dois être brisé doit, etre sorti, je dois enlever ces cas là.

2NF vers 3NF

Pour chaque relation

1, Identifier les DF dans la relation

2, Identifier \_\_\_PK\_\_\_ dans relations 2NF

|9:13| : JE vérifie si il y a des Dependance transitive sur la clé primaire. Primary Key

3, Si DT sur la PK

Supposons A la PK

A → B, B → C et A → C ou A → B → C

Déplacer B → C dans une novuelle relation avec une copie du \_\_\_déterminant\_\_\_

Créer nouvelle relation avec B et C

Conserver B dans relation 2NF

2NF vers 3NF

CLIENT

Df2 NUM\_CLIENT → NOM\_CLIENT (PK)

BAIL

Df1 NUM\_CLIENT, NUM\_PROPRIETE → DEB\_BAIL, FIN\_BAIL

Df5’ NUM\_CLIENT, DEB\_BAIL → NUM\_PROPRIETE, FIN\_BAIL (\_\_\_CK\_\_\_ )

|9:14| : Clé candidate et même clé alternative.

Df6’ NUM\_PROPRIETE, DEB\_BAIL → NUM\_CLIENT, FIN\_BAIL ( \_\_\_CK\_\_\_ )

PROPRIETE\_PROPRIETAIRE

Df3 NUM\_PROPRIETE → ADRESSE\_PRO, MNT\_LOC, NUM\_PROPRIETAIRE, NOM\_PROPRIETAIRE

Df4 NUM\_PROPRIETAIRE → NOM\_PROPRIETAIRE (\_\_\_DT\_\_\_ )

Car NUM\_PROPRIETE → NUM\_PROPRIETAIRE → NOM\_PROPRIETAIRE.

|9:15| : 2 DF identifie un autre attribut.

2NF vers 3NF

1, DF (1 a 6)

2, \_\_\_PK\_\_\_ (Déterminants de df2, df1 et df3)

3, Si DT sur la PK

PK NUM\_PROPRIETE

Df3 NUM\_PROPRIETE → NUM\_PROPRIETAIRE

Df4 NUM\_PROPRIETAIRE → NOM\_PROPRIETAIRE

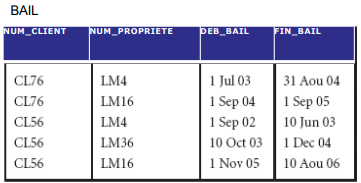
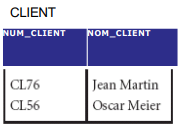
Donc NUM\_PROPRIETE → NOM\_PROPRIETAIRE

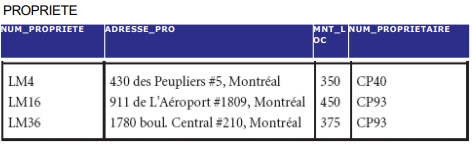
Déplacer df4 dans une nouvelle relation avec une copie du \_\_\_déterminant\_\_\_

Créer relation PROPRIETAIRE avec df4

Conserver NUM\_PROPRIETAIRE dans PROPRIETE.

3NF





Définition générales de la 2NF et la 3NF

|9:20| : C’est générale qui est important.

|9:21| : La règle s’applique aussi aux clés candidates pas juste aux clés primaires.

Seconde forme normale (2NF)

Relation 1NF

Chaque attribut non PK est en dépendance fonctionnelle complète sur toute \_\_\_CK\_\_\_

Troisième forme normale (3NF)

Relation 1NF

\_\_\_2NF\_\_\_

Aucun attribut non PK est en DT sur toute \_\_\_CK\_\_\_ .

Exemple récapitulatif (vide?)

|9:37| : UNF, on vois un tableau, avec plusieurs valeurs dans les meme cellule.

|9:37| : On fait une mise a plat : il y a une valeur par cellule. La table est devenu en 1NF

|9:38| : On énumère les attributs, Num\_client Num\_propriete etc.

|9:38| : On identifie les DF parmis les attributs. qu’elles sont les dépendance fonctionnel?

|9:38| : On veux garder un nombre de DF manipulable

|9:39| : DF7 et DF8 ne vont pas ajouter beaucoup au modèle

|9:39| : On identifie les Clés candidates. On choisit idéalement la plus courte comme clé primaire.

|9:40| : On regarde la clé primaire, et on vérifie si il y a des dépendance partielles.

|9:41| : On crée une table avec les dépendance partielle. Les relation seront en 2NF.

|9:42| : Les clés étrangère on les nommes avec les dièse # on peut aussi mettre #2 pour montrer une 2e clé étrangère dans une relation.

|9:43| : On veut passer à 3NF

|9:43| : On identifie les DF dans chaque tables.

|9:43| : On vérifie si il y a transitivité, par rapport à la clé primaire PK.

Règles d’inférence

|9:46| : Comment choisir les DF utile et moin utile?

Ensemble complet DF relation → tres \_\_\_vaste\_\_\_

|9:46| : Permutable (il a parlé vite)

Trouver approche grandeur gérable

|9:46| : Regarder des DF avec une grandeur gérable)

Identifier ensemble DF < ensemble complet DF (Y)

|9:47| : Je veux un ensemble plus petit que l’ensemble complet.

Chaque DF dans Y est \_\_\_implicite\_\_\_ par DF de X

\_\_\_Fermeture\_\_\_ X (noté X+) → Ensemble DF implicites partir de DF X

Règles \_\_\_d’inférence\_\_\_

Axiomes d’Armstrong

|9:47| : Pour définir ce qui est possible comme DF

|9:48| : Cette DF je peux la réduire a cause de cela… etc.

Comment DFs inférées à partir celles données

|9:48| : Qu’est-ce que je peut inférer et ce que je peux pas inférer.

A, B, et C sous-ensembles attributs de R

Règles d’inférence

Axiomes d’armstrong

1, \_\_\_Réflexivité\_\_\_ B sous-ensemble de A, alors A → B

|9:48| : Si B fait partit de A, par reflexivité, A détermine B.

2, Augmentation A → B, alors A,C → B,C

|9:49| : Si j’ajoute un attribut d’un coté je peux l’ajouter de l’autre coté.

3, \_\_\_Transivité\_\_\_ A → B et B → C, alors A → C

|9:49| : J’ai pas besoin d’écrire A -» séparément. Attention on parle de la règle d’amstrong ce n’est pas une DF transitive.

Règles supplémentaires dérivées, simplifient calcul de X+

D, sous ensemble attributs de R

4, \_\_\_Autodétermination\_\_\_ A → A

5, Décomposition A → B, C alors A → B et A → C

6, \_\_\_Union\_\_\_ A → B et A → C, alors A → B, C

|9:51| : Vous devriez trouver 5 DF dans le laboratoire Utilisez l’union.

7, Composition A → B et C → D, alors A, C → B, D .

|9:51| : Si je prend 2 déterminant je vais pouvoir les déterminés ensemble aussi. Attention par contre je peux pas faire l’inverse nécessairement. On ne peut pas aller de droite vers la gauche.

Ensembles minimaux de dépendances fonctionnelles

|9:53| : Couverture minimale

|9:53| : L’idée je trouve mon ensemble minimum de DF et je fais un union à la fin.

|9:54| : Essayer d’Avoir le moin de DF possible a exprimer.

Ensemble Y de DF est (fonctionnellement) \_\_\_couvert\_\_\_ par un ensemble de DF X si toutes DF de Y est aussi dans X+

Si toute DF dans Y peut être inférée à partir de X,

Ensemble X de DF \_\_\_minimal\_\_\_

|9:55| : Pourquoi c’est minimal parce qu’on peut pas briser aucune DF.

Toute DF dans X a un seul attribut à droite

Ne peut remplacer aucune A → B par C → B où C est un \_\_\_sous-ensemble\_\_\_ propre de A, et obtenir encore un ensemble équivalent à X

Ne peut \_\_\_supprimer\_\_\_ aucune DF de X et conserver un semble équivalent à X …

Ensembles minimaux de dépendnaces fonctionnelles

Converture minimale de DF X

Ensemble minimal de dépendances X min \_\_\_équivalent\_\_\_ à X \_\_\_plusieurs\_\_\_ couvertures minimales possibles pour 1 ensemble DF

Soit les DF de R

1, a → b, d, e, f

2, b → \_\_\_e, f\_\_\_

3, a, b → d, e, f

Alors

1’. a → b; a → d; a → e; a → f par règle \_\_\_5\_\_\_

Ensembles minimaux de dépendances fonctionnelles

2’. b→e b→f par règle \_\_\_5\_\_\_

3’. a, b → d a, b → e a, b → f par règle \_\_\_5\_\_\_

Avec règle \_\_\_7\_\_\_ , combiner 1 et 2 donne 3

Enlever les DF obtenues en 3’

Comme a → b, b → e et b → f donnent a → e et a → f avec règle \_\_\_3\_\_\_

Enlever a → e, a → f

Donc, couverture minimale de R

a → b a → d

b → e b → f

On peut retrouver R, on ne peut enlever DF.

Exercice

Quelle est cette règle? A → B,C, alors A → B et A → C?

A-Réflexivité B-Augmentation C-Transitivité D-Décomposition E-Union

F- Aucune de ces réponses

Quelle est cette règle? A → B, alors A, C → B,C

A-Réflexivité B-Augmentation C-Union D-Décomposition E-Composition

F-Aucune de ces réponses

Qu’est-ce que la couverture minimale?

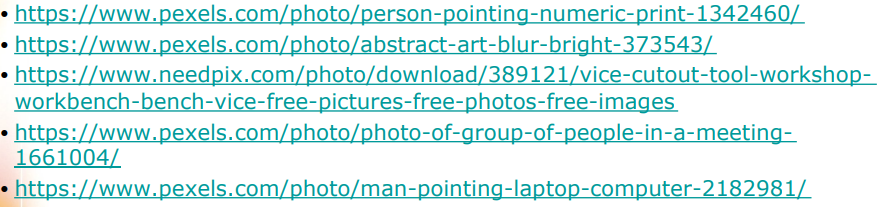
A-Toutes les DF que l’on peut déduire d’un ensemble X

B-Le minimum d’axiome à utiliser pour couvrir X

C-L’ensemble, irréductible, des DF permettant de déduire X

D-Aucune de ces réponses

Images



|10:2| : Laboratoire apres