

Exercice 1 : Premier exemple de normalisation

1.

Nom	Âge	UE	Jour
Hatem Ben Arfa	19	Programmation fonctionnelle	mardi, jeudi
Sonia Roland	24	Base de données	mercredi-vendredi
Nathan Drake	25	Base de données	mercredi-vendredi
LeBron James	27	Architecture des ordinateurs	lundi, mardi, vendredi
Eva Mendès	21	Pratique des machines	mardi-mercredi
Sidiki Diabaté	21	Programmation fonctionnelle	mardi, jeudi
Novak Djokovic	18	Informatique fondamentale	jeudi
Eikichi Onizuka	22	Méthodologie en programmation	lundi, mardi, vendredi
Whoopi Goldberg	18	Base de données	mercredi-vendredi
Scarlett Johansson	25	Informatique fondamentale	jeudi

2. Par définition, la clé est le minimum d'information qui permet l'identification non équivoque d'un tuple. Un attribut répond à ce critère : (NomEtud). Cela dit, on peut aussi considérer qu'il ne peut pas être choisi comme clé dans le cas d'homonymies entre les différents étudiants. On peut gérer ce problème en ajoutant un compteur après le nom, comme par exemple Sidiki Diabaté 03 pour différencier un nouvel étudiant homonyme.

3. Cette relation telle que définie en 1. n'atteint même pas la première forme normal en raison de l'attribut "Jour" qui n'est pas atomique. De plus, la relation comporte une redondance des valeurs (UE, Jour). Ce qui entraînera une perte de place sur le support physique et des anomalies de stockage dans les bases de données SQL (insertion, suppression et modification).

4. Avant de pouvoir rendre la relation BCNF, il faut d'abord rendre la relation 1NF. Pour cela, il faut rendre les valeurs de (Jour) atomiques. On éclate donc la colonne (Jour) en 5 colonnes pour chaque jour ouvrable de la semaine tel que :

Nom	Âge	UE	L	Ma	Me	J	V
H. Ben Arfa	19	Programmation fonctionnelle		x		x	
S. Roland	24	Base de données			x	x	x
N. Drake	25	Base de données			x	x	x
L. James	27	Architecture des ordinateurs	x	x			x
E. Mendès	21	Pratique des machines		x	x		
S. Diabaté	21	Programmation fonctionnelle		x		x	
N. Djokovic	18	Informatique fondamentale				x	
E. Onizuka	22	Méthodologie en programmation	x	x			x
W. Goldberg	18	Base de données			x	x	x
S. Johansson	25	Informatique fondamentale				x	

La table ci-dessus est visuellement plus parlante pour un humain mais peut avoir une dimension un peu trop imposante. Par souci de place, elle peut être réduite en codifiant les jours de la semaine ouvrable avec un 0 pour les jours non-concernés et avec un 1 (code binaire) pour les jours concernés (plus évident qu'utiliser des x et des o) comme :

Nom	Âge	UE	Jour
H. Ben Arfa	19	Programmation fonctionnelle	01010
S. Roland	24	Base de données	00111
N. Drake	25	Base de données	00111
L. James	27	Architecture des ordinateurs	11001
E. Mendès	21	Pratique des machines	01100
S. Diabaté	21	Programmation fonctionnelle	01010
N. Djokovic	18	Informatique fondamentale	00010
E. Onizuka	22	Méthodologie en programmation	11001
W. Goldberg	18	Base de données	00111
S. Johansson	25	Informatique fondamentale	00010

La relation est maintenant 1NF. Et comme aucun attribut non clé (Âge, UE, Jour) ne peut pas dépendre partiellement de la clé atomique (Nom), la relation est aussi 2NF. En revanche, la nouvelle relation n'est pas encore 3NF. Il faudrait pour cela qu'il n'existe aucune dépendance fonctionnelle entre les attributs n'appartenant pas à la clé. Pour cela, voyons quelles sont les DF de la relation UeEtud. Nous avons :

NomEtud \longrightarrow Âge, UE, Jour (qui est la clé)
UE \longrightarrow jour (l'inverse n'est pas vrai)

Ici, UE pose problème, il faut donc l'isoler en distinguant deux relations issues de UeEtud. Une première relation qu'on peut appeler Etud(NomEtud, Âge) et une seconde relation qu'on peut appeler Ue(UE, Jour). Ce qui nous donne :

TABLE 1 – Ue

UE	Jour
Programmation fonctionnelle	01010
Base de données	00111
Base de données	00111
Architecture des ordinateurs	11001
Pratique des machines	01100
Programmation fonctionnelle	01010
Informatique fondamentale	00010
Méthodologie en programmation	11001
Base de données	00111
Informatique fondamentale	00010

TABLE 2 – Etud

Nom	Âge
H. Ben Arfa	19
S. Roland	24
N. Drake	25
L. James	27
E. Mendès	21
S. Diabaté	21
N. Djokovic	18
E. Onizuka	22
W. Goldberg	18
S. Johansson	25

Par chance, en créant cette décomposition, nous obtenons une forme normale de Boyce-Codd car toutes les parties gauches des DF sont des clés.

Exercice 2 : Deuxième exemple de normalisation

1.

Enseignant	Salaire	UE	Jour
Karim Benzema	2000	Programmation fonctionnelle	01010
Jennifer Lopez	1850	Base de données	00111
Jack Sparrow	1400	Anglais Technique	11011
Dieudonné Mbala Mbala	2000	Architecture des ordinateurs	11001
Will Smith	1500	Pratique des machines	01100
Audrey Lamy	1400	Observations	00100
Son Goku	1850	Informatique fondamentale	00010
Jackie Chan	2200	Méthodologie en programmation	11001
Omar Sy	1800	Programmation Impérative	00111
Eric Judor	2500	Psychologie	00010

2. Par définition, la clé est le minimum d'information qui permet l'identification non équivoque d'un tuple. Un attribut répond à cette spécification : (UE). Il détermine à la fois (Jour), (Enseignant), et par transitivité $UE \rightarrow Enseignant \rightarrow Salaire$, détermine aussi (Salaire). Autrement dit UE est le seul attribut permettant de remonter de façon unique à l'ensemble des autres attributs donc $UE \rightarrow Enseignant, Salaire, Jour$.

3. La forme normale maximale atteinte par la relation est 2NF. D'abord, elle est 1NF car chaque valeur de colonne est atomique. Jour est un agrégat formant un code donc peut être considéré comme atomique contrairement à l'énumération de jours de la semaine standard. Ensuite, elle est 2NF car 1NF et bien que (Salaire) dépende indirectement de la clé (UE) par transitivité, (Salaire) ne peut pas dépendre pas d'une partie de la clé (UE) composée d'un unique attribut. En revanche, le fait que (Salaire) dépende d'un attribut non clé comme (Enseignant), empêche la relation d'atteindre le 3NF.

4. Revenons dans le cas où l'attribut (Jour) n'est pas atomique :

Enseignant	Salaire	UE	Jour
Karim Benzema	2000	Programmation fonctionnelle	mardi, jeudi
Jennifer Lopez	1850	Base de données	mercredi-vendredi
Jack Sparrow	1400	Anglais Technique	lundi, mardi, jeudi, vendredi
Dieudonné Mbala Mbala	2000	Architecture des ordinateurs	lundi, mardi, vendredi
Will Smith	1500	Pratique des machines	mardi, mercredi
Audrey Lamy	1400	Observations	mercredi
Son Goku	1850	Informatique fondamentale	jeudi
Jackie Chan	2200	Méthodologie en programmation	lundi, mardi, vendredi
Omar Sy	1800	Programmation Impérative	mercredi-vendredi
Eric Judor	2500	Psychologie	jeudi

Pour décomposer UeEnseignant en 2NF, il faudrait faire une division en deux relations Ue(UE, Jour) et Enseignant(Enseignant, Salaire) comme suit :

TABLE 3 – Ue

UE	Jour
Programmation fonctionnelle	01010
Base de données	00111
Anglais Technique	11011
Architecture des ordinateurs	11001
Pratique des machines	01100
Observations	00100
Informatique fondamentale	00010
Méthodologie en programmation	11001
Programmation Impérative	00111
Psychologie	00010

TABLE 4 – Enseignant

Nom	Salaire
Karim Benzema	2000
Jennifer Lopez	1850
Jack Sparrow	1400
Dieudonné Mbala Mbala	2000
Will Smith	1500
Audrey Lamy	1400
Son Goku	1850
Jackie Chan	2200
Omar Sy	1800
Eric Judor	2500

Ainsi, toutes les relations issues de la décompositions sont 2NF. Il y a une perte d'information dans ce cas puisqu'il n'est plus possible de reconstituer la jointure s'il n'y a pas une colonne commune aux deux relations. Il vaudrait mieux répéter la colonne (UE) dans la seconde relation issue de la décomposition pour pouvoir faire la liaison telle que :

TABLE 5 – Enseignant

UE	Nom	Salaire
Programmation fonctionnelle	Karim Benzema	2000
Base de donnée	Jennifer Lopez	1850
Anglais Technique	Jack Sparrow	1400
Architecture des ordinateurs	Dieudonné Mbala Mbala	2000
Pratique des machines	Will Smith	1500
Observations	Audrey Lamy	1400
Informatique fondamentale	Son Goku	1850
Méthodologie en programmation	Jackie Chan	2200
Programmation Impérative	Omar Sy	1800
Psychologie	Eric Judor	2500

Par contre, dans ce cas, nous avons une redondance des informations qui prendra de la place sur le support physique et la même clé UE pour les deux sous-relations. De plus, nous nous retrouvons avec la relation Ue(UE, Jour) avec la DF suivante : $UE \rightarrow Jour$. Et la relation Enseignant(UE, Enseignant, Salaire) qui ne sera pas 3NF à cause des DFs : $UE \rightarrow Enseignant$ et $Enseignant \rightarrow Salaire$.

5. Transformer la relation UeEnseignant en BCNF, découle naturellement des essais faits en 3. et 4. Il suffit pour cela d'isoler toutes les DFs en éclatant la relation en 3 sous-relations Jour(UE, Jour), Enseignant(UE, Enseignant) et Salaire(Enseignant, Salaire) à partir desquelles on pourra obtenir la relation UeEnseignant par jointures. pour passer le cap de 3NF et BCNF par la même occasion.

TABLE 6 – Jour

UE	Jour
Programmation fonctionnelle	01010
Base de données	00111
Anglais Technique	11011
Architecture des ordinateurs	11001
Pratique des machines	01100
Observations	00100
Informatique fondamentale	00010
Méthodologie en programmation	11001
Programmation Impérative	00111
Psychologie	00010

TABLE 7 – Salaire

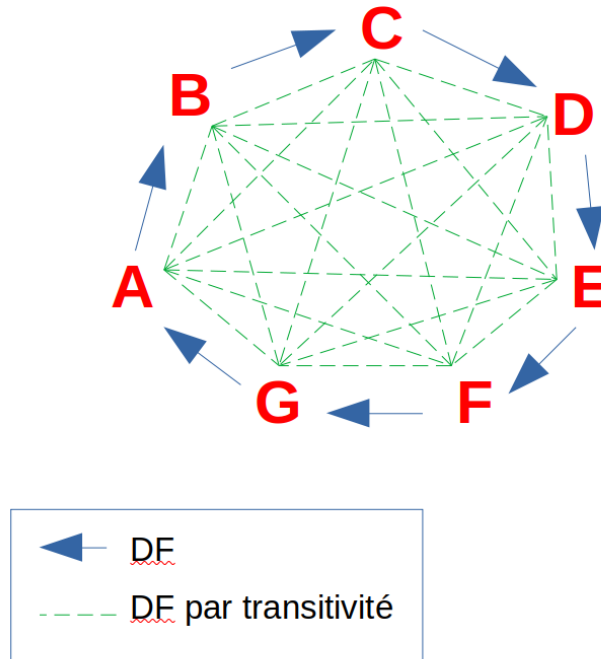
Enseignant	Salaire
Karim Benzema	2000
Jennifer Lopez	1850
Jack Sparrow	1400
Dieudonné Mbala Mbala	2000
Will Smith	1500
Audrey Lamy	1400
Son Goku	1850
Jackie Chan	2200
Omar Sy	1800
Eric Judor	2500

TABLE 8 – Enseignant

UE	Enseignant
Programmation fonctionnelle	Karim Benzema
Base de données	Jennifer Lopez
Anglais Technique	Jack Sparrow
Architecture des ordinateurs	Dieudonné Mbala Mbala
Pratique des machines	Will Smith
Observations	Audrey Lamy
Informatique fondamentale	Son Goku
Méthodologie en programmation	Jackie Chan
Programmation Impérative	Omar Sy
Psychologie	Eric Judor

Exercice 3 : Relation symbolique

1. Oui, il s'agit d'une couverture minimale car aucune des DFE ne peut être éliminée pour obtenir l'ensemble des DF.



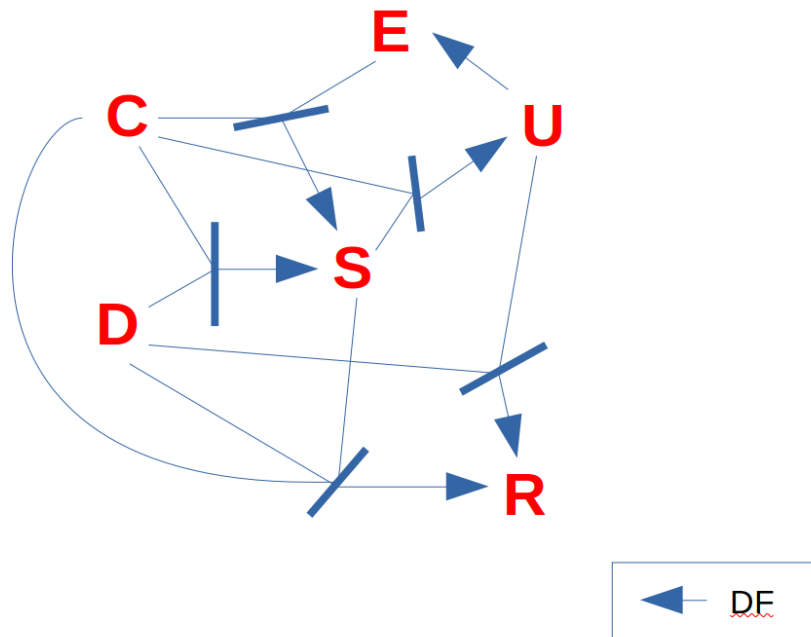
2. Chaque attribut de la relation R est une clé candidate car nous sommes dans une relation circulaire ou chaque élément détermine directement celui qui suit et indirectement tous les autres attributs. D'ailleurs l'identification d'un tuple de la relation se fait donc par n'importe quel point d'entrée, c'est à dire n'importe quel attribut ou groupement d'attribut.

3. La première forme normale est supposée acquise d'après l'énoncé. Ensuite, puisqu'il existe beaucoup de clés candidates équivalentes, il faut d'après le cours, choisir une clé primaire parmi elles avant de déterminer si la relation est 2NF. Une fois ce choix effectué (par exemple clé = attribut A), on voit qu'aucun attribut restant, dit non-clé, ne peut dépendre partiellement de la clé atomique (attribut A) choisie. La relation R est donc 2NF. On voit cependant qu'il y a de nombreuses DFs entre les attributs n'appartenant pas à la clé précédemment choisie. Ainsi la relation ne peut pas être 3NF.

4. Pour répondre à la norme 3NF, puisqu'il y a une dépendance circulaire entre chaque attribut, il faut éclater chaque dépendance en une relation unique, c'est-à-dire en 7 relations différentes pour chaque DF de R. En faisant cela, chaque nouvelle relation créée contient une unique DFE littéralement de forme : clé \rightarrow attribut. Ainsi, toutes les nouvelles relations sont également BCNF.

Exercice 4 : Institut

1.



Si l'on trace un schéma, on se rend plus facilement compte que seuls les attributs C et D n'ont aucune dépendance (aucune flèche ne pointe vers eux). Ils pourraient potentiellement être des clés candidates. En revanche, pris individuellement, ni C, ni D ne permet de d'identifier tous les autres attributs (tuple). En les couplant, on peut voir que l'on peut accéder à n'importe quel attribut de la relation Formation(C, D, E, R, S, U). D'ailleurs, en remplaçant chaque lettre représentant un attribut par sa signification, cela devient assez logique. Le créneau horaire C (sans doute une date du type JJ :MM :AAAA + heure) et l'étudiant D renseignent sur : le créneau horaire et l'étudiant (réflexivité), l'enseignant E, la salle S, l'unité d'enseignement U et enfin sur le résultat R.

2. On suppose la relation 1NF à minima. Conformément au 1., puisque C et D, pris unitairement (clé partielle), ne déterminent aucun des attributs non-clés alors la relation est 2NF. Le schéma indique également clairement qu'il y a d'autres dépendances fonctionnelles entre les attributs non-clés. La relation n'est donc pas 3NF.

3. Pour rendre la relation 3NF, il convient d'isoler toutes les DF entre attributs non-clés. L'éclatement de la relation mère en relations filles dépend de si l'on souhaite que toutes les relations filles soit BCNF et si l'on souhaite préserver un maximum de dépendances. On pourrait très bien créer 6 relations différentes pour chaque dépendance fonctionnelle mais ce n'est pas la configuration optimale. On note que deux dépendances comportent le même élément dans la partie droite : d'une part $UD \rightarrow R$ et $CSD \rightarrow R$, d'autre part $CE \rightarrow S$ et $CD \rightarrow S$. En d'autres termes, nous avons deux possibilités pour trouver R et S. Nous pouvons potentiellement réduire le nombre de relations filles au nombre de 4 en éliminant deux doublons. Naturellement, nous gardons la relation fille de notre clé primaire CD. Reste à trancher entre la relation fille issue de la DF $UD \rightarrow R$ et $CSD \rightarrow R$. Ici, il vaut mieux renoncer à cette dernière car en utilisant les propriétés des DF, notamment la transitivité, on peut reconstituer ladite relation

filles. En effet, CS détermine U qui est un élément de UD. Par substitution de U par CS, on obtient la dernière DF $CSD \rightarrow R$. La relation fille issue de la DF $CE \rightarrow S$ peut également être obtenue par jointures. D'ailleurs $CS \rightarrow E$, par transitivité, alors $C \rightarrow E$ par réflexivité, et donc $CS \rightarrow ES$ par augmentation, équivalant à $CS \rightarrow EE$ puisque $S \rightarrow E$, pouvant donc être réduit à $CS \rightarrow E$. En définitive, nous pouvons obtenir l'ensemble de couverture minimale avec les 4 relations filles suivantes : $R_1(U,E)$, $R_2(C,S,U)$, $R_3(U,D,R)$, $R_4(C,D,S)$.

Exercice 5 : Entreprise

1. Au total, nous avons 13 attributs : NOSER, NOMS, NORES, NOMR, BUSER, NOPRO, NOMP, BUPRO, NOEMP, NOME, NOTEL, NOBUR et NOMB.

On suppose qu'il n'y a pas d'homonymie ou que les conditions posées dans les tables de la base de données empêche l'homonymie (concerne NOMS, NOMR, NOMP, NOME et NOMB). Voici les dépendances fonctionnelles que l'on peut tirer de la compréhension de l'énoncé avec \longleftrightarrow signifiant une interdépendance fonctionnelle :

$NOSER \longleftrightarrow NOMS$

$NORES \longleftrightarrow NOMR$

Deux services peuvent avoir le même budget alloué donc : $NOSER \rightarrow BUSER$

Chaque service gère un ou plusieurs projets, mais un projet est géré par un seul service donc la DF se fait en sens unique tel que : $NOPRO \rightarrow NOSER$

$NOPRO \longleftrightarrow NOMP$

Deux projets peuvent avoir un même budget alloué donc la DF se fait en sens unique tel que : $NOPRO \rightarrow BUPRO$

À un instant t d'interrogation de la base de donnée, nous avons : $NOEMP \rightarrow NOPRO$

$NOEMP \longleftrightarrow NOME$

"Chaque employé peut être joint par l'intermédiaire d'un numéro de téléphone (NOTEL). Un numéro de téléphone peut être partagé entre plusieurs employés". Autrement dit, nous avons la DF à sens unique : $NOEMP \rightarrow NOTEL$.

"Un employé est installé dans un bureau caractérisé par un numéro unique (NOBUR)" donne la DF : $NOEMP \rightarrow NOBUR$

Un bureau peut accueillir plusieurs employés et plusieurs appareils téléphoniques signifie qu'on a : $NOTEL \rightarrow NOBUR$

Enfin, comme la localisation d'un bureau est repérée par le nom de son bâtiment (NOMB), on obtient : $NOBUR \rightarrow NOMB$

On remarquera que certaines DF sont implicites comme $NOME \rightarrow NOTEL$ par substitution de $NOEMP$ par $NOME$ puisque $NOEMP$ et $NOME$ sont interdépendants. Ceci soulève

le choix de clés primaires parmi diverses clés candidates pour déterminer la couverture minimale de dépendances. La couverture minimale est par définition l'ensemble minimum de DFE permettant de générer toutes les autres DF à l'aide de propriété élémentaires. Pour obtenir, cette dernière, il suffit d'avoir un moyen d'accéder à tous les attributs tout en respectant les DF susmentionnées. Cherchons les clés primaires, habituellement, il vaut mieux considérer les numéros d'identification plutôt que les noms. Ainsi pour le minimum répondant à ce critère, nous avons l'ensemble composé des éléments suivants :

NOSER \longrightarrow NOMS, NORES, NOMR, BUSER

NOPRO \longrightarrow NOMP, NOSER, BUPRO

NOEMP \longrightarrow NOPRO, NOME, NOTEL, NOBUR

NOTEL \longrightarrow NOBUR

NOBUR \longrightarrow NOMB

2. Pour atteindre la norme 3NF, il faut supprimer les DFs entre les attributs non-clés. En d'autres termes, il faut pouvoir créer différentes relations ou tables. La logique nous pousse à considérer 5 tables ou relations avec comme clé primaire un identifiant de type numéro : Responsable(NORES, NOMR), Projet(NOPRO, NOMP, BUPRO), Service(NOSER, NOMS, BUSER), Employé(NOEMP, NOME), Installations(NOTEL, NOBUR, NOMB). Ce sont les formes les plus logiques mais elles ne sont ni 3NF, ni BCNF. Pour cela, faudrait encore subdiviser les tables Projet en Projet1(NOPRO, NOMP) et Projet2(NOPRO, BUPRO), ainsi que Service en Service1(NOSER, NOMS) et Service2(NOSER, BUSER), enfin Installations en Installation1(NOTEL, NOBUR) et Installations2(NOBUR, NOMB). Tous les attributs sont placés cependant, on voit que certaines informations sont perdues car les jointures ne permettent pas d'accéder à tous les attributs. Autrement dit la couverture minimale n'est pas encore atteinte. Pour préserver les DFs minimales tout en ayant la norme maximale BCNF, nous devons trouver un moyen d'ajouter des clés étrangères aux tables afin de pouvoir faire toutes les jointures sans perdre la norme BCNF.

Nous aurons au final à compléter les 8 relations existantes, citées en résumé ci-dessous, par 3 relations tampons de type **clé primaire** \longrightarrow **clé étrangère** : Tampon1 (NOEMP, NOPRO), Tampon2 (NOEMP, NOTEL), Tampon3 (NORES, NOSER).

Responsable(NORES, NOMR)

Projet1(NOPRO, NOMP)

Projet2(NOPRO, BUPRO)

Service1(NOSER, NOMS)

Service2(NOSER, BUSER)

Employé(NOEMP, NOME)

Installation1(NOTEL, NOBUR)

Installations2(NOBUR, NOMB)