ESIR1 / BDD vendredi 21 mars 2014
Contrôle continu 30 mn

1. **Pourquoi cherche-t-on à normaliser des relations ?**

Pour les rendre accessibles (1NF) aux opérations relationnelles, et plus résistantes aux mises-à-jours (2NF et plus).
Ça n'a rien à voir avec un soucis de simplification (vous trouvez ça vraiment simple ?) ni avec un soucis de standardisation. Cette dernière réponse me fait penser à un cours pas étudié, et à une réponse par association d'idée : standardisation = normalisation, et bien pas ici.

2. **Pourquoi préfère-t-on parfois ne pas normaliser ?**

Parce qu'il y aurait peu de mise-à-jour et parce que la contrepartie de la normalisation est de devoir faire des jointures, qui sont coûteuses.

3. **Pour chacune des formes normales listées dire ce qu'il y a de problématique à ne pas la satisfaire.**

À cette question pourquoi, j'ai essentiellement eu des réponses du genre « 1NF c'est … » ou « Pas 1NF c'est … » alors que la question appelait des réponses « Ne pas être 1NF est problématique parce que … ». Apprenez à répondre à la question qui est vraiment posée ! Qu'est ce que X, Pourquoi X, Quoi faire si pas X, Quand X, Où X, etc, sont autant de questions différentes qui appellent des réponses différentes. Toutes appellent des compréhensions différentes de X. L'autre jour, je suis tombé sur un cours d'anglais de préparation au TOEIC, et c'est exactement un des exercices demandés ! Dites-vous que vous passez le TOEIC de français, et je n'ose même pas parler de l'orthographe. Vous savez que lors de certains entretiens d'embauche on fait écrire quelque chose au candidat ? Ce quelque chose a intérêt à être pertinent et bien écrit. Je n'évalue pas l'orthographe, mais quand l'absence d'orthographe nuit à l'intelligibilité, je ne vais pas inventer du sens là où il n'y en a pas.

- **1NF :** si une relation n'est pas 1NF elle a des attributs non-atomiques, ex. une liste d'auteurs ou une adresse complète qui seraient câblées dans une chaîne de caractères. Comment faites-vous une jointure par le code postal, ou par le premier auteur dans ces conditions. Comment faite-vous un tri, un GROUP BY ? Un attribut pas 1NF s'exclut des opérations relationnelles usuelles**.**
- **2NF, 3NF :** dans les deux cas on crée des redondances. Pour chaque DF fautive A->B on doit répéter la même valeur b de l'attribut B à chaque fois qu'il y a la valeur a de l'attribut A, mais il n'y a rien dans le schéma pour nous y aider. En plus lorsqu'on supprime la dernière valeur a de l'attribut A, on supprime aussi le b de l'attribut B et on perd l'information que c'était b qui était associée à a de l'attribut A. Et finalement, lorsqu'on veut mettre à jour le b pour une nouvelle valeur b', on risque de remplacer des occurrences de b qui n'ont rien à voir avec ce a.

4. **Pourquoi est-il nécessaire de penser en terme de transaction ?**

Il y a peu de chance qu'une opération métier corresponde exactement à une requête, ex. un transfert bancaire, ou une insertion dans une BD éclatée en plusieurs tables (pour normaliser). Le programmeur d'application ou l'administrateur d'application vont donc programmer des opérations plus macroscopiques qui enchaînent des requêtes. Penser en terme de transaction permet de penser ces opérations macroscopiques globalement, et aussi de penser les interactions de ces opérations entre elles.

5. **Que signifie le A de ACID ?**

A comme atomicité, c'est-à-dire la propriété qu'une transaction s'exécute en entier ou pas du tout.
Beaucoup on rajouté « et si une des requêtes de la transaction fait une erreur on efface tout et on recommence ». Ce dernier point est inutile, et faux. Si l'erreur est déterministe (ex. une requête mal formulée qui viole une contrainte d'intégrité) recommencer ne fera que reproduire l'erreur. C'est un réflexe que vous avez souvent, recommencer quand il y a une erreur, mais cela vous met souvent dans des situations tordues où non seulement il y a une erreur, mais à force de recommencer vous avez saturé une ressource système. On l'a vu en TP quand le serveur MySQL s'est mis à refuser les connexions depuis certains postes de travail.

6. **Qu'est-ce que la sérialisabilité ?**

La propriété d'une exécution concurrente de transactions (avec un s) qui fournit le même résultat qu'une exécution séquentielle.
Miracle ! Vous avez presque tous répondu ça au mot près… Ça se gâte après quand certains rajoutent qu'il n'y a qu'un résultat possible, celui de l'exécution séquentielle (notez l'article déterminant). Il peut y avoir plusieurs résultats séquentiels possibles (ex. T1 ; T2 vs T2 ; T1). On demande juste que l'exécution concurrente (T1 // T2) donne un de ceux-là.

7. **Qu'est-ce qu'il y a de problématique à ne pas être sérialisable ?**

Après le bon score de la question 6 on pouvait s'attendre à un bon score ici ; mais non. Cela suggère que beaucoup on répondu correctement à la question 6 sans vraiment comprendre ce qu'ils écrivaient. C'est grave. N'écrivez pas juste pour faire plaisir au professeur. Écrivez pour montrer que vous avez compris la question, que vous connaissez la réponse, et que vous la comprenez. Si vous vous arrêtez à connaître la réponse, mais sans la comprendre, c'est de l'endoctrinement, vous vous auto-endoctrinez. Si vous visez le stade de la compréhension (le nirvana), vous pourrez réutiliser le concept ou le reconnaître ailleurs (ex. en système ou dans un protocole réseau).
Ce qui est problématique à ne pas être sérialisable est que le programmeur d'une transaction ne peut plus penser sa transaction isolément des autres, et non seulement il devrait savoir quelles autres transactions peuvent être exécutées en concurrence avec la sienne, mais en plus il devrait savoir comment elles ont été programmées. C'est ingérable parce que cela peut changer n'importe quand au gré de l'activité des autres programmeurs ou administrateur d'applications !

Soit les relations suivantes :

- **Spectacle(num-spec, titre-spec, salle, metteur-en-scene)**
- **Jouer(nom-acteur, num-spec)**
- **Représentations(date, heure, num-spec, tarif)**

8. **Écrire en SQL la requête qui insert l'acteur `A` dans tous les spectacles du metteur en scène `MS`.**

INSERT INTO Jouer (nom-acteur, num-spec)
SELECT A, num-spec FROM Spectacle WHERE metteur-en-scene = MS
Pas de jointure, pas de UPDATE, pas de SET, pas de INSERT A INTO…

9. **On souhaite apporter les modifications suivantes à la base. Pour chaque modification dire brièvement de quelle façon la réaliser et pourquoi de cette façon. Ne pas chercher à produire le code SQL qui réalise la modification.**

- **Les spectacles ont un nombre limité de spectateurs donné par la salle :**

Il suffit de créer une nouvelle table Salle(num-salle, nom-salle, capacité).
Mettre la capacité dans Spectacle est une erreur car cela serait vraiment anti-3NF ! Comme rien ne parle des réservations dans cette BD, ce n'est pas la peine d'en parler, et surtout ne pas mettre le statut des réservations, qui bouge tout le temps, dans nos tables, qui n'ont pas vocation à beaucoup bouger.

- **Une salle ne peut pas accepter plusieurs spectacles à la même date :**

Déjà se demander pourquoi cette contrainte. Ex. on peut imaginer qu'on n'a pas le temps de changer les décors entre une séance de l'après-midi (une matinée !) et une séance du soir. Toujours essayer de comprendre ce que le client veut dire. Ce n'est pas à formuler dans la réponse, c'est juste pour vous. Rappelez-vous, SYNTAXE et SÉMANTIQUE, et en modélisant vous êtes entre les deux.
Il y a donc une DF salle, date -> spectacle. Le mieux est de créer une nouvelle table Programme(salle, date , spectacle) et de remplacer num-spec par num-salle dans Représentations.
La colonne tarif peut rester dans Représentations ou basculer dans Programme si on apprend que le tarif ne dépend pas de l'heure. Beaucoup de mauvaises solutions impliquaient qu'un spectacle ne peut avoir lieu qu'à une heure donnée ou à une date donnée, ou qu'il ne pouvait y avoir qu'un spectacle par date alors que si il y a plusieurs salles c'est tout à fait possible.

- **Il y a toujours un tarif normal et parfois un tarif réduit :**

Faute d'en savoir plus sur la politique tarifaire il faut faire simple. Le plus simple est de rajouter une colonne tarif-réduit dans Représentations. Celle-ci pourrait être NULL en l'absence de tarif réduit.
On pourrait aussi convenir de ce que le tarif réduit est soit le tarif plein soit un tarif inférieur. Mais ça complique quand même la réponse à la question quels sont les spectacles qui offrent des tarifs réduits. Ne pas se lancer dans des codages bizarres du genre le tarif réduit est 0 quand il n'y en a pas ou un tarif réduit. Cela empêche d'avoir un tarif tellement réduit que gratuit, et ça rend difficile de répondre à la question quelle est la moyenne des tarifs réduits. Ne pas créer de table Tarif qui ferait croire que le tarif réduit dépend fonctionnellement du tarif. Ne pas se lancer non plus dans des taux de réductions ou d'autres éléments de politique tarifaire dont on n'a pas connaissance. Ce n'est pas à l'informaticien de proposer une politique tarifaire à un entrepreneur de spectacle !