

Bases de Données TD2

Algèbre Relationnelle, Modélisation UML, Passage au relationnel

1. Montrez la même table (relation) parmi celles ci-dessous.

S#	P#	QTY
S1	P1	300
S2	P2	200
S1	P3	400

P#	S#	QTY
P3	S1	400
P2	S2	200
P1	S1	300

P#	S#	QTY
P1	S1	300
P3	S1	200
P2	S2	400

2. Une opération appliquée aux relations (tables relationnelles) donne comme résultat une table. De ce fait les relations manipulées doivent être *union-compatibles*. Ainsi, elles doivent avoir d'abord le même nombre des attributs. Puis, les domaines (ensembles des valeurs) des attributs, soit A et B, à comparer, doivent être aussi compatibles ($\text{dom}(A) \approx \text{dom}(B)$). En pratique, soit ils doivent être du même type, numérique par exemple, soit le SGBD doit être capable de les convertir dans un type commun. Chaque SGBD a ses règles pour les conversions acceptables. Est-ce que les opérations suivantes sont possibles, toujours ou sous certaines conditions, à préciser alors:
- R (A, B) UNION S (D, E, F) puis R (A, B, C) DIFF S (D, E, F)
 - R (A) UNION S (D) en sachant que l'attribut A est (sous MsAccess) de type Numérique Octet ($\text{dom } A = \text{Int}(256)$) et que D est de type Texte ($\text{dom } D = \text{Char}(256)$).
 - R (A) UNION S (D) en sachant que sous MsAccess 2007, A est de type Numérique Entier ($\text{dom } A = \text{Int}(64K)$) et que D est de type Memo ($\text{Char}(64KO)$).
 - R (A) UNION S (D) où A est de type quelconque sous MsAccess et $\text{dom } D = \text{OLE}$ (tout objet Windows sous MsAccess), ou PJ, ou de type quelconque déclaré en plus à valeurs multiples.
 - R (A) UNION S (D) où A est un texte et D est un hyperlien.
3. Expliquez les requêtes suivantes et montrez l'effet de leurs opérations successives sur la base S-P:
- S [CITY]; S WHERE CITY = 'Paris', S JOIN P; S JOIN P ON S.CITY = P.CITY;
 - ((S JOIN SP) WHERE P# = 'P4') [SNAME]; (S JOIN (SP WHERE P# = 'P4')) [SNAME]
 - ((P WHERE COLOR = 'Red') [P#] JOIN SP) [S#] JOIN S [SNAME];
 - ((P WHERE COLOR = 'Red') [P#, PNAME] JOIN SP) [S#, PNAME] JOIN S [SNAME];
 - ((SP [S#, P#] DIVIDEBY P [P#]) JOIN S) [SNAME];
 - SP [S#, P#] DIVIDEBY ((SP WHERE S# = 'S2') [P#]);
4. Est-ce que (R1 JOIN R2) WHERE R2.a = C est toujours égal à (R1 JOIN (R2 WHERE R2.a = C)) comme dans 3.b ci-dessus ? Si oui, commentez sur ce que faut en déduire sur le fonctionnement de SGBD, en supposant qu'une jointure examine pour chaque tuple de R1 tous les tuples de R2 et qu'une restriction examine tous les tuples de la table concernée. Prendre comme exemple numérique l'exécution de 3.b sur la base S-P du cours.
5. Formulez les requêtes suivantes et montrez le résultat sur S-P:
- Noms des fournisseurs de la pièce 'P2'
 - Noms des pièces fournies par S1 et S2.
 - Couleurs des pièces fournies par S1 et non-fournies par S2.
 - Noms de pièces fournies par tout fournisseur.
 - Noms et ville communes de fournisseurs dans une même ville, en évitant la répétition de tuples du genre (x, y, z) et (y, x, z). Si besoin de renommer un attribut A, utilisez la déclaration (RENAME A AS B).
6. On crée la BD de personnes identifiées chacune par un Nom et ayant plusieurs Emails et plusieurs Tél.
- Montrer l'extension pour TOTO avec e1, e2, e3 et t1, t2 en supposant d'abord 0 NF puis 1 NF
 - Montrer une conceptualisation UML souhaitable pour le modèle relationnelle.
7. Indiquez les attributs clés, les clés candidates, les clés minimales, les clés étrangères et une sur-clé dans S-P.
8. On a la table suivante des étudiants sans que la clé primaire soit encore choisie:
E (E#, Nom, Email, Tél, Tuteur, Adresse)
- L'étudiant est identifié par E# et, alternativement, par les couples (Nom, email) et (Nom, tél). Le tuteur est étudiant (plus expérimenté). Indiquez les attributs clés, les clés candidates, les clés minimales, les clés étrangères, s'il y en a, et une sur-clé. Proposez la clé primaire.
9. Proposez le modèle conceptuel pour la base S-P.
10. A la compagnie d'assurances FIAM, chaque client a un et un seul conseiller (chargé de clientèle). Chaque client est identifié par un numéro par conseiller. Donc le client 1 chez le conseiller A n'est pas le même que le

client 1 chez le conseiller B. Les autres attributs du client sont le nom et l'adresse. Le conseiller a le nom et le tél., mais aussi le nombre de clients gérés (2 à 7). Proposez une modélisation UML et une réification.

11. Considérez alternativement, que chaque client a un numéro unique d'assuré. Quel sera le modèle UML et les tables ?

12. Proposez un schéma relationnel avec les liens sémantiques et contraintes d'intégrité réf. pour un Trombinoscope M1 avec photo, adresse perso et prof. avec code postal et ville, impliquée par celui-ci, ID d'un étudiant, peut-être fictif, un email, peut-être plusieurs tél de type : domicile, travail, mobile, un ou plusieurs hobbies.

13. **Proposez le schéma relationnel pour l'école de conduite du cours sur la conception UML.**

14. A Dauphine on a les personnes nommées et identifiées par P# qui peuvent être, notamment, des étudiants avec leur E# et/ou les salariés avec leur S#. Proposez le schéma UML et celui relationnel.

Corrigé TD2

1. Montrez la même table (relation) parmi celles ci-dessous. La 1^{ère} et la 2^{ème}.

S#	P#	QTY
S1	P1	300
S2	P2	200
S1	P3	400

P#	S#	QTY
P3	S1	400
P2	S2	200
P1	S1	300

P#	S#	QTY
P1	S1	300
P3	S1	200
P2	S2	400

2. Une opération appliquée aux relations (tables relationnelles) donne comme résultat une table. De ce fait les relations manipulées doivent être *union-compatibles*. Ainsi, elles doivent avoir d'abord le même nombre des attributs. Puis, les domaines (ensembles des valeurs) des attributs, soit A et B, à comparer, doivent être aussi compatibles ($\text{dom}(A) \approx \text{dom}(B)$). En pratique, soit ils doivent être du même type, numérique par exemple, soit le SGBD doit être capable de les convertir dans un type commun. Chaque SGBD a ses règles pour les conversions acceptables. Est-ce que les opérations suivantes sont possibles, toujours ou sous certaines conditions, à préciser alors:
- R (A, B) UNION S (D, E, F) **non** puis R (A, B, C) DIFF S (D, E, F) **oui sous conditions évidentes**
 - R (A) UNION S (D) en sachant que l'attribut A est (sous MsAccess) de type Numérique Octet ($\text{dom } A = \text{Int}(256)$) et que D est de type Texte ($\text{dom } D = \text{Char}(256)$). **Oui**
 - R (A) UNION S (D) en sachant que sous MsAccess 2007, A est de type Numérique Entier ($\text{dom } A = \text{Int}(64K)$) et que D est de type Memo ($\text{Char}(64KO)$). **Oui, mais pas sous MsAccess 2003**
 - R (A) UNION S (D) où A est de type quelconque sous MsAccess et $\text{dom } D = \text{OLE}$ (tout objet Windows sous MsAccess), ou PJ, ou de type quelconque déclaré en plus à valeurs multiples). **Non**
 - R (A) UNION S (D) où A est un texte et D est un hyperlien. **Oui, mais hyperlien est converti en texte.**

3. Expliquez les requêtes suivantes et montrez l'effet de leurs opérations successives sur la base S-P:

- S [CITY]; S WHERE CITY = 'Paris', S JOIN P; S JOIN P ON S.CITY = P.CITY;
- ((S JOIN SP) WHERE P# = 'P2') [SNAME]; (S JOIN (SP WHERE P# = 'P2')) [SNAME]
- ((P WHERE COLOR = 'Red') [P#] JOIN SP) [S#] JOIN S [SNAME];
- ((P WHERE COLOR = 'Red') [P#, PNAME] JOIN SP) [S#, PNAME] JOIN S [SNAME];
- ((SP [S#, P#] DIVIDEBY P [P#]) JOIN S) [SNAME];
- SP [S#, P#] DIVIDEBY ((SP WHERE S# = 'S2') [P#]);

- Toutes les villes de fournisseurs (sans doublons) : Tout sur tout fournisseur à Paris ; Tout sur tout fournisseur et toute pièce qui est produite dans la même ville qu'un fournisseur ; Même chose, mais avec deux colonnes CITY, par exemple pour vérifier que la requête a été correctement exécutée.
 - Nom de tout fournisseur de la pièce P2
 - Nom de tout fournisseur d'une pièce rouge.
 - Même chose, mais formulé autrement. La 1^{ère} jointure prendra moins de place mémoire. Est-ce que PNAME est une nécessité ? Sinon, comment optimiser encore cette requête ?
 - Le nom de tout fournisseur de toutes les pièces
 - Toutes les pièces fournies par un fournisseur de toutes les pièces fournies par le fournisseur 'S2'
4. Est-ce que (R1 JOIN R2) WHERE R2.a = C est toujours égal à (R1 JOIN (R2 WHERE R2.a = C)) comme dans 3.b ci-dessus ? Si oui, commentez sur ce que faut en déduire sur le fonctionnement de SGBD, en supposant qu'une jointure examine pour chaque tuple de R1 tous les tuples de R2 et qu'une restriction examine tous les tuples de la table concernée. Prendre comme exemple numérique l'exécution de 3.b sur la base S-P du cours.

Oui. Il faut que SGBD en général effectue la restriction d'abord. Dans notre exemple, on examine plusieurs fois moins de tuples. Donc en général on exécute la requête plusieurs fois plus vite.

5. Formulez les requêtes suivantes et montrez le résultat sur S-P:

- Noms des fournisseurs de la pièce 'P2' : ((S JOIN SP) WHERE P# = 'P2') [SNAME]
 - Noms des pièces fournies par S1 et S2 : (S JOIN SP) WHERE S# = 'S1' [P#, PNAME] JOIN ((S JOIN SP) WHERE S# = 'S2')
 - Couleurs des pièces fournies par S1 et non-fournies par S2. (S JOIN SP JOIN P) WHERE S# = 'S1' [COLOR]
- DIFF
((S JOIN SP JOIN P) WHERE S# = 'S2') [COLOR]
- Noms de pièces fournies par tout fournisseur. **Voir (e) plus haut.**

- e. Noms et ville communes de fournisseurs dans une même ville, en évitant la répétition de tuples du genre (x, y, z) et (y,x, z).

S RENAME CITY AS SCITY RENAME SNAME AS SNAME1 TIMES S WHERE SCITY = CITY
WHERE SNAME1 > SNAME

6. On crée la BD de personnes identifiées chacune par un Nom et ayant plusieurs Emails et plusieurs Tél.
a. Montrer l'extension pour TOTO avec e1, e2, e3 et t1, t2 en supposant d'abord 0 NF puis 1 NF
b. Montrer une conceptualisation UML souhaitable pour le modèle relationnelle.

Voir le cours

7. Indiquez les attributs clés, les clés candidates, les clés minimales, les clés étrangères et une sur-clé dans S-P.
8. On a la table suivante des étudiants sans que la clé primaire soit encore choisie:

E (E#, Nom, Email, Tél, Tuteur, Adresse)

L'étudiant est identifié par E# et, alternativement, par les couples (Nom, email) et (Nom, tél). Le tuteur est étudiant (plus expérimenté). Indiquez les attributs clés, les clés candidates, les clés minimales, les clés étrangères, s'il y en a, et une sur-clé. Proposez la clé primaire.

9. Proposez le modèle conceptuel pour la base S-P.

10. A la compagnie d'assurances FIAM, chaque client a un et un seul conseiller (chargé de clientèle). Chaque client est identifié par un numéro par conseiller. Donc le client 1 chez le conseiller A n'est pas le même que le client 1 chez le conseiller B. Les autres attributs du client sont le nom et l'adresse. Le conseiller a le nom et le tél., mais aussi le nombre de clients gérés (2 à 7). Proposez une modélisation UML et les tables relationnelles.

11. Considérez alternativement, que chaque client a un numéro unique d'assuré. Quel sera le modèle UML alors et les tables ? **L'entité client ne sera pas faible. Le schéma de la Clients sera le même.**

12. Proposez un schéma relationnel avec les liens sémantiques et contraintes d'intégrité réf. pour un Trombinoscope M1 avec photo, adresse perso et prof. avec code postal et ville, impliquée par celui-ci, ID d'un étudiant, peut-être fictif, un email, peut-être plusieurs tél de type : domicile, travail, mobile, un ou plusieurs hobbies.

13. **Proposez le schéma relationnel pour l'école de conduite du cours.**

Cl (Cl#, Nom, Prenom, Adr, D_Naiss)

S_Code (S#, Date, Heure, Cd#, S#)

A (Cl#, S#, N_Fautes)

E (PC#, Date, Heure, Lieu)

E-R (PC#, Cl#, N_Fautes)

CD (Cd#, Edit)

SRQ (Cd#, S#, Q#, P#)

Q (Q#, Int, Rep, Diff, Theme)

Les noms de tables et des attributs sont abrégés par rapport au schéma UML. C'est une pratique courante. La table S inclut l'info sur la série diffusée, car il y a une série par séance. A noter le traitement de l'identité faible. Il n'y a pas de table pour la série toute seule, elle ne serait pas utile. Si la série avait en plus d'autres attributs, par exemple l'attribut Descr, donnant sa description verbale, alors cette table serait justifiée (pourquoi au juste ?). La table SRQ réifie l'association Position_Question. Le choix de la clé signifie que dans une série, une question ne peut être plus qu'une fois.

14. A Dauphine on a les personnes nommées et identifiées par P# qui peuvent être, notamment, des étudiants avec leur E# et/ou les salariés avec leur S#. Proposez le schéma UML et celui relationnel.

Le schéma UML est celui d'une classe avec les sous-classes. Le schéma relationnel peut être :

P (P#, Nom, Adresse, ...)

/* Données de toute personnes

E (E#, P#, Cycle, ...)

/* Données spécifiques aux étudiants

S (S#, P#, Sal, ...)

/* Données spécifiques aux salariés

A noter que P# est une clé candidate dans E et S. Elle pourrait être une clé primaire dans ces tables. Elle est aussi une clé étrangère dans E et dans S. On a l'association de type 1:1 avec la contrainte d'intégrité référentielle (où toute valeur de la clé étrangère doit être celle de P.P# d'un tuple dans P).

Observez que formellement, ce schéma correspond à la décomposition sans perte de la table universelle soit U comme Université :

U (P#, Nom, Adresse... E#, Cycle..., S#, Sal...),

Ceci, en utilisant une généralisation de Th. de Heath, (proposée en premier par votre prof., semble-t-il). Pour la table R1 (A,B), l'attribut A est d'abord E#. L'attribut B se définit comme d'habitude, par la règle d'union de tous les attributs fonctionnellement dépendant sur E#, mais cette fois-ci, avec la condition supplémentaire. A savoir, qu'il ne s'agit que des attributs qui sont nuls si E# est nul, ou de la clé de P, P# en l'occurrence. Puis, l'on ne retient pour la table (A,B) que les tuples où E# n'est pas nul. On peut dire que l'on projète U sur (A,B), en y choisissant E# comme clé. Donc en y retient nécessairement que les tuples où E# n'est pas nul. Ceci donne la table E. La table (A,C) c'est tous les tuples de U avec les attributs AC. Ensuite, d'une manière similaire, on projète ce (A,C) sur le nouveau (A,B) = S cette fois-ci. Ce qui reste de U comme (AC), donne P final ci-dessus.

Pour la recomposition sans perte, on peut observer qu'il suffit d'utiliser la jointure externe gauche (Left Join, voir mon cours sur SQL), avec la projection éliminant les attributs E.P# et S.P# à la fin (inutiles). Par exemple :

$$U = P \text{ Left Join } E \text{ On } P.P\# = E.P\# \text{ Left Join } S \text{ on } P.P\# = S.S\# [P.P\#, \text{Nom}, \dots, E\#, \text{Cycle}\dots, S\#, \text{Sal}\dots]$$

Pour rappel, le Th. de Heath originel ne connaît pas de concept de valeur nulle et utilise la jointure naturelle pour la recomposition sans perte. Le procédé expliqué donne une justification formelle (mathématique) à la conception empirique de classes et de sous-classes d'UML. On peut créer dès lors des assistants de conception automatique qui ne semblent pas encore exister. Donc si vous voulez gagner un peu de d'argent....