

Bases de Données Avancées

10 décembre 2009

Examen

durée 3h. Documents autorisés, appareils mobiles de communication interdits

INTRODUCTION

Pour cet examen, on veut créer un site de questionnaires, comme ceux présents dans les magazines. Un questionnaire (ou quiz) porte sur un thème, et comporte des questions. Chaque question a des propositions de réponses, qui comportent un symbole (carré, rond, triangle ou étoile). En fonction des propositions que vous avez choisies, on compte le nombre de symboles de chaque type et le symbole majoritaire vous donne le résultat au quiz.

Voici deux exemples de quiz :

Quel genre de casse-pieds êtes-vous ?¹

1. J'ai tendance à me ronger les ongles :
- | | |
|------|---|
| vrai | ★ |
| faux | ○ |

2. Les critiques m'affectent particulièrement :
- | | |
|------|---|
| vrai | △ |
| faux | ★ |

3. ...

Résultat du quiz :

majorité de □ :	vous êtes autoritaire
majorité de △ :	vous êtes victime
majorité de ○ :	vous êtes prétentieux
majorité de ★ :	vous êtes désinvolte

Quelle mangeuse êtes-vous ?

1. Un repas que vous appréciez tout particulièrement

un dîner en tête à tête avec votre amoureux	□
Un repas en famille	○
Un apéro entre copains	★

2. Votre truc pour ne pas grossir

Ecouter sa faim	△
Le dernier régime en vogue	★
Bannir le saucisson et le chocolat	○

3. ...

Résultat du quiz :

majorité de □ :	vous êtes gourmande
majorité de △ :	vous êtes raisonnable
majorité de ○ :	vous êtes traditionnelle
majorité de ★ :	vous êtes obsédée par votre régime

¹merci à Jean-Marie de m'avoir passé ce test très instructif, trouvé dans *Management*.

Exercice 1 : On représente les questionnaire avec le modèle relationnel. Voici le schéma de la base, et le contenu des tables correspondant aux questionnaires donnés en exemples :

Table QUIZ

QUIZ(id_quiz, titre, resultat_carre,resultat_triangle, resultat_rond, resultat_etoile);
id_quiz est la clé primaire de QUIZ

ID_QUIZ	TITRE	RESULTAT_CARRE	...	RESULTAT_ETOILE
10	Quel genre de casse-pieds êtes-vous	Autoritaire	...	Désinvolte
20	Quelle mangeuse êtes-vous	Gourmande	...	Obsédée du régime

Table QUESTION

QUESTION(id_quiz, numero, enonce)
(id_quiz,numero) est la clé primaire de QUESTION.
id_quiz clé étrangère sur la table QUIZ

ID_QUIZ	NUMERO	ENONCE
10	1	J'ai tendance à me ronger les ongles
10	2	Les critiques m'affectent particulièrement
20	1	Un repas que vous appréciez tout particulièrement
20	2	Votre truc pour ne pas grossir
...		

Table PROPOSITION

PROPOSITION(id_quiz, numero, num_prop, texte_prop, item_prop);
(id_quiz,numero,num_prop) est la clé primaire de PROPOSITION
(id_quiz,numero) est une clé étrangère sur QUESTION
on vérifie (item_prop in ('carre', 'rond', 'triangle', 'etoile'))

ID_QUIZ	NUMERO	NUM_PROP	TEXTE_PROP	ITEM_PROP
10	1	1	vrai	carre
10	1	2	faux	rond
10	2	1	vrai	triangle
10	2	2	faux	etoile
20	1	1	un dîner en tête à tête avec votre amoureux	carre
20	1	2	Un repas en famille	rond
20	1	3	Un apéro entre copains	etoile
20	2	1	Ecouter sa faim	triangle
20	2	2	Le dernier régime en vogue	etoile
20	2	3	Bannir le saucisson et le chocolat	rond
...				

Table PERSONNE

PERSONNE(id_pers, nom, prenom, email);
id_pers est la clé primaire de PERSONNE

ID_PERS	NOM	PRENOM	EMAIL
10	caron	anne-cécile	caron@hotmail.com
20	lebbe	jean-marie	super-manager@laposte.net

Table REPONSE

```
REPONSE( id_pers, id_quiz , numero,num_prop);  
id_pers clé étrangère sur PERSONNE  
(id_quiz,numero,num_prop) clé étrangère sur PROPOSITION  
la clé primaire est (id_pers,id_quiz,numero)  
car 1 personne donne 1 seule réponse (num_prop) par question.
```

L'informaticien qui a conçu le schéma a défini une procédure PL/SQL au sein d'un paquetage `paq_quiz` qui permet d'ajouter la réponse d'un utilisateur à une question. Cette procédure déclenche une exception `DONNEE_INCONNUE` si la personne ou la proposition de réponse (triplet (`le_quiz`, `le_numero`, `la_reponse`)) n'existent pas dans la base. Si la personne a déjà répondu à cette question alors l'ancienne réponse est remplacée par la nouvelle.

```
create or replace package paq_quiz as  
  
procedure repondre_question(la_personne personne.id_pers%type,  
                           le_quiz quiz.id_quiz%type,  
                           le_numero question.numero%type,  
                           la_reponse proposition.num_prop%type);  
  
DONNEE_INCONNUE Exception ;  
pragma exception_init(DONNEE_INCONNUE,-20100);  
  
end paq_quiz ;
```

JDBC

On dispose de la classe `Base` utilisée en TP, avec les méthodes suivantes :

- un constructeur sans paramètre qui réalise la connexion.
- `public void fermerBase()`
- `public Statement creerStatement()`
- `public CallableStatement creerAppel(String s)` Cette méthode prend en paramètre la chaîne de caractère qui représente l'appel de procédure ou fonction
- `public PreparedStatement preparerRequete(String s)` Cette méthode prend en paramètre une requête SQL.

On suppose qu'il existe une classe java `Quiz` qui a pour variable d'instance un objet `b` de type `Base` avec une connexion ouverte. Un objet `Quiz` représente un questionnaire, et possède comme variable d'instance un entier `identifiant`, qui correspond à une valeur `id_quiz` de la table `QUIZ`. Par contre, on suppose qu'un objet `Quiz` ne possède pas de `ResultSet` ni aucun objet représentant les questions et les propositions (tout est en base).

Les questions qui suivent concernent des méthodes d'instance de la classe `Quiz`.

- Vous déclarerez en dehors des méthodes toutes les variables de (super-)type `Statement` et vous les initialiserez.
- Vous utiliserez au moins une fois une requête préparée avec paramètre(s).

Question 1.1 : Ecrire dans la classe `Quiz` une méthode java qui renvoie le résultat du quiz pour une personne donnée. Cette méthode est une fonction qui prend en paramètre l'identifiant d'une personne et renvoie la chaîne de caractères contenant la réponse finale, en fonction du symbole majoritaire des réponses de la personne. Cette fonction vérifie que la personne a bien répondu à

toutes les questions du quiz, et renvoie une exception `QuizException` si ce n'est pas le cas.

Question 1.2 : Ecrire dans la classe `Quiz` une méthode java qui permet de répondre à une question. Cette procédure prend en paramètre l'identifiant d'une personne, le numéro de la question, et la réponse de la personne, et appelle la procédure stockée `reponse_question`.

OPTIMISATION PL/SQL ET TRAITEMENT DE REQUÊTE

Le corps de la procédure PL/SQL est le suivant :

```
procedure repondre_question(la_personne personne.id_pers%type,
                           le_quiz quiz.id_quiz%type,
                           le_numero question.numero%type,
                           la_reponse proposition.num_prop%type) is
    cpt NUMBER ;
begin
    -- est-ce que la personne existe ?
    select count(*) into cpt from PERSONNE where id_pers = la_personne ;
    if (cpt=0) then raise DONNEE_INCONNUE ; end if ;

    -- est-ce que la proposition de réponse existe ?
    select count(*) into cpt from PROPOSITION
    where id_quiz = le_quiz and numero = le_numero and num_prop = la_reponse ;
    if (cpt=0) then raise DONNEE_INCONNUE ; end if ;

    -- est-ce que cette personne a déjà répondu à cette question ?
    select count(*) into cpt from REPONSE
    where id_quiz = le_quiz and numero = le_numero and id_pers = la_personne ;

    if (cpt = 0) then -- pas encore de réponse
        insert into reponse(id_pers,id_quiz,numero,num_prop)
        values(la_personne,le_quiz,le_numero,la_reponse) ;
    else -- déjà une réponse que l'on modifie
        update reponse set num_prop = la_reponse
        where id_quiz = le_quiz and numero = le_numero and id_pers = la_personne;
    end if ;
end repondre_question ;
```

Question 1.3 : Cette procédure fait plusieurs requêtes pour vérifier que les valeurs sont correctes, ou pour décider s'il s'agit d'une nouvelle réponse ou de la modification d'une réponse qui existe. Sachant que le serveur teste les clés primaires (erreur de code -1) et les clés étrangères (erreur de code -2291), ces requêtes entraînent un coût inutile. Réécrivez cette procédure pour qu'elle ait le même comportement mais en posant le moins de requêtes possibles.

Question 1.4 : On suppose qu'une ligne de la table `QUESTION` fait 40 octets. que `id_quiz` occupe 1 octet, ainsi que `numero`. Il existe un index B-arbre sur la clé primaire (`id_quiz,numero`). On pose la requête suivante :

```
select * from question where id_quiz = 10;
```

1. Est-ce que le système a intérêt à utiliser l'index B-arbre sur (`id_quiz,numero`) sachant qu'il y a 20 lignes dans la table `QUESTION`
2. S'il y a 5000 questionnaires ayant chacun 20 questions, dessinez l'index B-arbre. Est-ce que le système a intérêt à utiliser cet index pour l'exécution de la requête précédente ?
3. On suppose qu'il y a 100 lignes. Comment le système va traiter la requête :

```
select id_quiz, count(*) from question group by id_quiz;
```

Exercice 2 : On décide d'écrire la base de questionnaires en utilisant le modèle relationnel-objet. Voici les types et tables définis pour modéliser les données :

QUIZ

```
-- un objet pour contenir les résultats selon le symbole majoritaire
create type symboles_type as object(
    carre VARCHAR2(500),
    triangle VARCHAR2(500),
    rond VARCHAR2(500),
    etoile VARCHAR2(500)
);

create or replace type quiz_type as object(
    id_quiz NUMBER(3),
    titre VARCHAR2(50),
    les_resultats symboles_type,
    les_questions ens_questions_type
);
```

QUESTION

```
create type question_type as object(
    le_quiz REF quiz_type,
    numero NUMBER(2),
    enonce VARCHAR2(100),
    les_propositions ens_propositions_type
);

create type ref_question_type as object(
    ref_question REF question_type
);

create or replace type ens_questions_type as table of ref_question_type;
```

PROPOSITION

```
create type proposition_type as object(
    la_question REF question_type,
    num_prop NUMBER(1),
    texte_prop VARCHAR2(100),
    symbole_prop VARCHAR2(8)
    -- on ne gere pas de collection avec toutes les reponses contenant cette proposition
);

create type ref_proposition_type as object(
    ref_proposition REF proposition_type
);
create type ens_propositions_type as table of ref_proposition_type;
```

REPONSE

```
create type reponse_type as object(
```

```

    la_personne REF personne_type,
    la_proposition REF proposition_type
);

create type ref_reponse_type as object(
    ref_reponse REF reponse_type
);
create type ens_reponses_type as table of ref_reponse_type ;

```

PERSONNE

```

create type personne_type as object(
    id_pers NUMBER(4),
    nom VARCHAR2(30),
    prenom VARCHAR2(30),
    email VARCHAR2(50),
    les_reponses ens_reponses_type
);

```

La table QUIZ2 (resp. QUESTION2, PROPOSITION2, REPONSES2, PERSONNE2) contient des objets de type quiz_type (resp. question_type, proposition_type, reponse_type, personne_type). On suppose que pour chaque colonne de type REF, il existe une contrainte de clé étrangère. Par contre, il n'a pas été possible de déclarer une contrainte de clé primaire pour QUESTION2, PROPOSITION2, REPONSES2.

Question 2.1 : Donner les requêtes SQL qui permettent d'ajouter les informations concernant le quiz *quel genre d'emmerdeur êtes-vous ?*. Vous vous limiterez à 1 question de ce quiz, et 1 proposition de réponse à cette question.

Question 2.2 : Ecrivez la procédure `repondre_question` avec cette nouvelle modélisation des données. Elle doit avoir le même comportement quant à la gestion des erreurs que la procédure écrite en relationnel.

Exercice 3 : Voici deux requêtes, l'une sur la version relationnelle (exercice 1), l'autre sur la version relationnelle-objet (exercice 2), qui donnent les réponses de la personne 10 au quiz 10.

- version relationnelle

```

select question.enonce, proposition.texte_prop
from reponse
join question on reponse.numero = question.numero and reponse.id_quiz = question.id_quiz
join proposition on proposition.num_prop = reponse.num_prop
                  and proposition.numero = reponse.numero
                  and proposition.id_quiz = reponse.id_quiz
where reponse.id_pers=10 and reponse.id_quiz=10

```

- version relationnelle-objet

```

select tlr.ref_reponse.la_proposition.la_question.enonce,
       tlr.ref_reponse.la_proposition.texte_prop
from the(select p.les_reponses from personne2 p where p.id_pers=10) tlr
where tlr.ref_reponse.la_proposition.la_question.le_quiz.id_quiz=10;

```

Question 3.1 : Comparez le traitement de ces deux requêtes, en expliquant dans chacun des cas l'algorithme utilisé pour répondre à la requête. Laquelle est la plus efficace ?