

Cartouche du document

Année : ING 2

Matière : Base de données 2A

Activité : Examen

Objectifs

Cet examen de base de données porte sur tous les TDs de Base de données pratiqués en salle.

Tous les documents papiers sont autorisés.

La durée de l'examen est de 1h00 heure.

Sommaire des exercices

1 - Des Jeux en SQL pur

2 - Des Jeux en SQL et XMLTYPE

Corps des exercices

1 - Des Jeux en SQL pur

Énoncé :

Un centre de vacances organise des jeux. Un jeu est décrit par un nom (différent de ceux des autres jeux), un gain et une date. Des vacanciers peuvent s'inscrire à des jeux. Ils se présentent aux différents jeux avec un pseudo qu'ils ont choisi à leur arrivée dans le centre. Il ne peut pas y avoir plus de 10 joueurs par jeu.

On a donc les trois relations (qui donneront lieu à trois tables).

Jeu(idJeu, nomJeu, gain, dateJeu)

Joueur(idJoueur, pseudo)

JoueurJeu(#idJeu, #idJoueur)

Question 1)

Énoncé de la question

(5pts) Ecrire une procédure qui reçoit un nom de jeu, un gain, une date et qui crée ce jeu. Cette création ne doit se faire que s'il n'existe pas déjà un jeu de même nom.

Solution de la question

```
create or replace procedure creationJeu(pnomJeu varchar2, pgain number, pdateJeu
```

```
date)
as
vidjeu number;
begin
    select seq_idjeu.nextval into vidjeu from dual;
    insert into jeu values(vidjeu,pnomjeu,pgain,pdatejeu);
exception
when dup_val_on_index then
raise_application_error(-20001,'erreur création jeu');
end;
/
```

Question 2)

Énoncé de la question

(4pts pour le trigger et 4pts pour la procédure) Ecrire une procédure qui reçoit le nom d'un jeu et le pseudo d'un vacancier. Cette procédure inscrit le vacancier au jeu. Cette inscription ne peut se faire que si le jeu existe. Le non dépassement de 10 joueurs par jeu doit être géré par un trigger.

Solution de la question

```
create or replace trigger trig_joueur_jeu
before insert on JoueurJeu
for each row
declare
vnbInscrits number;
begin
select count(*) into vnbInscrits from JoueurJeu
where idJeu = :new.idJeu;
if vnbInscrits > 10 then
    raise_application_error(-20001,'jeu plein');
end if;
end;
/
```

```
create or replace procedure inscriptionJoueurJeu(pnomJeu varchar2, ppseudo
varchar2)
as
vidjeu number;
vidjoueur number;
begin
    select idjeu into vidjeu from jeu where nomJeu = pnomJeu;
    select idjoueur into vidjoueur from joueur where pseudo = ppseudo;
    insert into JoueurJeu values(vidJoueur,vidJeu);
exception
when no_data_found then
raise_application_error(-20001,'erreur joueur ou jeu inexistant');
```

```
end;  
/
```

2 - Des Jeux en SQL et XMLTYPE

Énoncé :

On reprend l'exercice précédent mais on désire gérer toutes les informations dans une seule table Jeu en utilisant un arbre XML.

Question 1)

Énoncé de la question

(1pt) Donner un exemple d'arbre XML qui décrit les caractéristiques d'un jeu et tous ses joueurs.

Solution de la question

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<jeu nomJeu="Jeu1" gain="20000" dateJeu="20/01/2008">  
  <joueur pseudo="lionel" id="1"/>  
  <joueur pseudo="hervé" id="2"/>  
  <joueur pseudo="françoise" id="3"/>  
</jeu>
```

Question 2)

Énoncé de la question

(1pt) Donner le script de création de la table Jeu.

Solution de la question

```
create or replace table jeuXML (idJeu number, descriptionJeu XMLTYPE,  
constraint pk_jeu primary key(idJeu));
```

Question 3)

Énoncé de la question

(1pt) Donner le script d'insertion d'un jeu.

Solution de la question

```
insert into jeuXML values(1, XMLTYPE('<?xml version="1.0"  
encoding="iso-8859-1"?><jeu nomJeu="Jeu1" gain="20000" dateJeu="20/01/2008">  
  <joueur pseudo="lionel" id="1"/>
```

```
<joueur pseudo="hervé" id="2"/>  
<joueur pseudo="françoise" id="3"/>  
</jeu>  
'));
```

Question 4)

Énoncé de la question

(2pts) Ecrire un script de mise à jour d'un jeu qui change le gain du jeu connu par son nom.

Solution de la question

```
UPDATE jeuXML SET descriptionJeu =  
UPDATEXML(descriptionJeu, '/jeu[@nomJeu="Jeu1"]/@gain', '30000')
```

Question 5)

Énoncé de la question

(2pts) Ecrire le script d'interrogation de la table Jeu qui affiche tous les joueurs d'un jeu connu par son nom.

Solution de la question

```
SELECT extract(descriptionJeu, 'jeu[@nomJeu="Jeu1"]/joueur') from jeuXML
```