

TP2 Reporting SQL PLUS

Introduction

Tous les scripts utilisés pour réaliser ce TP se trouvent dans le fichier *tp2.sql*, mais on peut également les consulter dans les fichiers identifiés suivant le numéro de question.

Q 1

Pour créer une sequence, on ajoute la commande suivante dans notre script. Comme on nous l'a expliqué lors du TP, pour pouvoir réellement commencer à 1,

nous avons utilisé `START WITH 0` et `MINVALUE 0` :

```
CREATE SEQUENCE idcl_sequence START WITH 0 INCREMENT BY 1 MINVALUE 0 ;
```

on utilise par la suite

```
idcl_sequence.nextval
```

lors de l'insertion de nouvelles valeurs dans la table :

```
INSERT INTO Clients Values (idcl_sequence.nextval, 'toto', 'titi', 'Mars', '123456789123');
INSERT INTO Clients Values (idcl_sequence.nextval, 'tata', 'tete', 'Jupiter', '234567891231');
INSERT INTO Clients Values (idcl_sequence.nextval, 'Potter', 'Harry', 'Poudlard', '345678912312');
INSERT INTO Clients Values (idcl_sequence.nextval, 'Elessar', 'telcontar', 'fennas druinin', '456789123123');
INSERT INTO Clients Values (idcl_sequence.nextval, 'Holmes', 'Sherlock', '221b Baker str', '567891231234');
INSERT INTO Clients Values (idcl_sequence.nextval, 'Dr. Watson', 'John', '221b Baker str', '678912312345');
INSERT INTO Clients Values (idcl_sequence.nextval, 'Pr. Moriarty', 'Jim', NULL, NULL);
```

Toutes ces instructions se trouvent dans le fichier *tp2.sql* entre les lignes 38-50.

Q 2

On nous a demandé de tester la requête suivante :

```
SELECT 'DELETE FROM ' || table_name || ';' FROM user_tables;
```

Cette requête nous prépare les commandes pour vider toutes les tables de notre base de données, sans pour autant les exécuter.

La sortie de cette commande est :

```
DELETE FROM CLIENTS;
DELETE FROM LIVRES;
DELETE FROM ACHATS;
DELETE FROM AVIS;
```

Q 3

Nous avons rajouté quelques lignes au script suggéré afin de rétablir l'affichage d'informations.

```
SET ECHO OFF
SPOOL poubelle.sql
SET ECHO OFF
SET FEEDBACK OFF
SET HEADING OFF
SET PAGESIZE 0

SELECT 'DELETE FROM ' || table_name || ';' FROM USER_TABLES ;

SPOOL OFF
SET PAGESIZE 500
SET FEEDBACK ON
SET HEADING ON
```

Le résultat de l'exécution de ce script s'enregistre dans le fichier *poubelle.sql*. il faut ensuite exécuter *poubelle.sql* pour vider tout le contenu de nos tables.

Ce script se trouve dans le fichier *question3.sql*, puis *effacer.sql*.

Q 4

On propose les exemples suivant:

- Exemple 1

```
SET ECHO OFF
SPOOL Drop_Table.sql
SET ECHO OFF
SET FEEDBACK OFF
SET HEADING OFF
SET PAGESIZE 0

SELECT 'DROP TABLE ' || table_name || ' CASCADE CONSTRAINTS;' FROM USER_TABLES ;

SPOOL OFF
--SET ECHO ON
SET PAGESIZE 500
SET FEEDBACK ON
SET HEADING ON
```

Ce script permet de supprimer toutes les tables présentes dans notre base de données. Il se trouve dans le fichier *question4_ex1.sql*.

- Exemple 2

```
SET ECHO OFF
SPOOL acheteurs2011.lst
SET ECHO OFF
SET FEEDBACK OFF
SET HEADING OFF
SET PAGESIZE 0

SELECT nom,pren,tel || ';' FROM Achats a left outer join clients c on a.idcl=c.idcl where
a.dateachat >= to_date('01-01-2011','DD-MM-YYYY') and a.dateachat <= to_date('31-12-2011','DD-MM-
YYYY') group by nom,pren,tel;

SPOOL OFF
--SET ECHO ON
+SET PAGESIZE 500
SET FEEDBACK ON
+SET HEADING ON
```

Ce script permet d'avoir dans le fichier *acheteurs2012.lst* tous les clients (nom, prenom,tel) de chaque client ayant acheté au moins un livre durant l'année 2011. Il est contenu dans le fichier *question4_ex2*.

Q 5

Pour ajouter une nouvelle colonne dans la table achats, on utilise la commande suivante :

```
alter table achats add prix number(4,2);
```

Ensuite pour remplir cette colonne on utilise les instructions suivantes :

```
update achats set prix = '18,99' where ref1 = '011A';
update achats set prix = '15,99' where ref1 = '011B';
update achats set prix = '23,99' where ref1 = '03A3';
update achats set prix = '21,99' where ref1 = '03B3';
update achats set prix = '22,99' where ref1 = '03AA';
update achats set prix = '27,99' where ref1 = '02A3';
```

Dans le fichier *tp2.sql* on peut trouver ces instructions entre les lignes 153-163.

Q 6

Voici la requête utilisée pour sélectionner les clients ayant effectué un achat avant le 28 janvier 2013.

```
SELECT idcl,dateachat,genre,prix
FROM Achats NATURAL JOIN livres
WHERE dateachat <= to_date('28-01-2013','DD-MM-YYYY')

ORDER BY idcl,dateAchat ASC;
```

On utilise cette requête dans le script suivant afin de générer le rapport demandé :

```
set headsep !

-- Permet de definir le titre du rapport
ttitle 'Achats des clients au 28 janvier 2013'

--définit la colonne des identifiant de client de type entier
column idcl format 999
-- définit la colonne pour le prix d'un livre d'un type nombre(4,2)
column prix format 99.99

-- Permet d'ajouter avant une valeur différente d'idcl
break on idcl skip 1 on report
-- Permet de calculer la somme et la moyenne des achats que chaque client a réalisé
compute avg sum of prix on idcl

set linesize 80
set pagesize 41
set newpage 0
set feedback OFF

SET ECHO OFF

--Écrit le resultat dans le fichier
spool 2013-01-28-achats.lst

--Requête qui permet d'avoir le résultat souhaité
select idcl,dateachat,genre,prix from Achats natural join livres where dateachat <= to_date('28-01-2013','DD-MM-YYYY') order by idcl,dateAchat asc;

spool off
```

Le script se trouve dans le fichier *question6.sql*.