

TP2 Reporting SQL PLUS

Q 1

Pour utiliser une sequence qu'on ajout la commande permettant créer la sequence, dans le script qui permet creer lesrelations de notre base des données:

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

Puis on remplace les valeurs d'attribut clé de la relation Clients par :

```
maSequence.nextval
```

et on obtient les requêtes d'insertion suivantes:

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

Tout ces instructions se trouve dans le fichiers *tp2.sql* entre les lignes 38-48.

Q 2

On etait commandé d'implementer la commande suivante

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

La commande suivante nous prepare les commandes pour vider tous les relations qu'on a dans notre bdd, sans supprimer aucun lignes de nos relations;

Le resultats de cette commande est

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

Q 3

Pour mieux utiliser la commande décrit en question 2, on utilise le scrit suivant en ajoutant la commande dernière :

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

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

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

Le resultat de l'execution de cette script enregistre dans le fichier *effacer.sql* le resultat de la commande du question 2. Ensuite on execute *poubelle.sql* pour vider tous les contenus de nos relations.

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 tous les relations presents dans notre base des données.

Pour executer ce script, il faut executer le fichier *question4_ex1.sql*, et puis *Drop_Table.sql*.

- Exemple 2

```

SET ECHO OFF
SPOOL acheteurs2011.sql
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.sql* tous les clients (nom, prenom,tel) de chaque client qui ont acheté la(les) livre(s) durant l'année 2011.

Ce script peut être exécuté en utilisant le fichier *question4_ex2.* et puis *acheteurs2011.sql*.

Q 5

Pour ajouter une nouvelle colonne à la relation 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 83-92

Q 6

Dans cette question il fallait faire le rapport de la moyenne et la somme des prix par client. Avant de faire le rapport, on a d'abord écrit la requête sql qui correspond à la demande. La requête est la suivante :

```

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;

```

Puis on a écrit le script suivant qui permet d'avoir le rapport demandé :

```

set headsep !

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

--definit la colonne des identifiant de client de type entier
column idcl format 999
-- definit la colonne pour la date achat de chaque livre de type date
column DateAchat format a9
-- definit la colonne pour la genre d'un livre de type chaine de caractère de longueur 15
column Genre format a15
-- definit la colonne pour la prix d'un livre d'un type nombre(4,2)
column prix format 99.99

-- permet d'afficher la moyenne et la somme juste avant qu'on change 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

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

--requete qui permet d'avoir le resultat 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

```

Ce script peut etre executer en utilisant le fichier *question6.sql*, le resultat sera enregistrer dans le fichier *2013-01-28-achats.lst*.