



Documents autorisés – Durée : 2h – Barème fourni à titre indicatif – Questions indépendantes

Question 1 (2 points)

Soit la table de schéma ETIQUETTES (Et1, Et2), ne contenant aucun tuple. Donner le contenu de cette table à l'issue de la séquence de commandes SQL*Plus suivante.

```
INSERT INTO ETIQUETTES VALUES ('A', 'A');
INSERT INTO ETIQUETTES VALUES ('A', 'B');
INSERT INTO ETIQUETTES VALUES ('A', 'C');
INSERT INTO ETIQUETTES VALUES ('B', 'A');
INSERT INTO ETIQUETTES VALUES ('B', 'B');
ROLLBACK;
INSERT INTO ETIQUETTES VALUES ('B', 'C');
COMMIT;
UPDATE ETIQUETTES SET Et2='Z' WHERE Et1='B';
INSERT INTO ETIQUETTES VALUES ('B', 'B');
COMMIT;
INSERT INTO ETIQUETTES VALUES ('C', 'A');
INSERT INTO ETIQUETTES VALUES ('C', 'B');
INSERT INTO ETIQUETTES VALUES ('C', 'C');
DELETE FROM ETIQUETTES;
ROLLBACK;
UPDATE ETIQUETTES SET Et1='Z';
COMMIT;
```

Question 2 (10 points)

Soit la base de données FABRICATION dont le schéma est donné ci-dessous.

```
PIECE (NOP, DESIGNATION, COULEUR, POIDS)
SERVICE (NOS, INTITULE, LOCALISATION)
COMMANDE (NOP, NOS, QUANTITE)
```

Formuler en SQL*Plus les requêtes suivantes.

1. Numéro des services ayant en commande la pièce P1 avec une quantité supérieure à 10, dans l'ordre croissant
2. L'emballage d'une pièce a un poids égal à environ 20 % du poids de la pièce. Poids total emballé pour les différentes pièces, dans l'ordre décroissant.
3. Nombre de services ayant des commandes.
4. Quantité moyenne commandée pour la pièce P3.
5. Liste des pièces commandées par le service S1 avec leur libellé et leurs poids.
6. Numéro des services qui ont commandé la pièce P1 et la pièce P3 en même quantité.
7. Numéro des services qui ont commandé la pièce P3 avec une quantité inférieure à la quantité moyenne

commandée pour cette pièce.

8. Numéro des services ayant commandé au moins une pièce en quantité strictement supérieure à chacune des quantités de pièces commandées par le service S1.
9. Intitulé des services qui n'ont pas commandé de pièce.
10. Quantité totale commandée pour les pièces P1, P2, et P3.

Question 3 (4 points)

On souhaite disposer de statistiques concernant les mises à jour (UPDATE) des tables d'une base de données. Ces statistiques sont stockées dans une table de schéma STATS (NomTable, NbMàJ). Cette table est initialisée avec le nom de toutes les tables de la base de données et le nombre de mises à jour *NbMàJ* correspondant initialisé à 0.

Ex.

NomTable	NbMàJ
EMP	0
DEPT	0
EMP_FR	0

...

Écrire un déclencheur (trigger) après mise à jour de la table EMP et permettant d'incrémenter le compteur statistique *NbMàJ* pour la table EMP dans la table STATS.

Question 4 (4 points)

On suppose que des triggers tels que celui de la question 3 sont définis sur toutes les tables de la base de données. En utilisant la table STATS et le catalogue du système, définir une vue nommée STATOWN permettant, pour chaque propriétaire (OWNER) de table, de connaître le nombre total de mises à jour effectuées sur ses tables.

Ex. Si EMP et EMP_FR appartiennent à l'utilisateur A, DEPT à l'utilisateur B et que la table STATS contient les données suivantes :

NomTable	NbMàJ
EMP	3
DEPT	5
EMP_FR	7

alors le résultat de la requête SELECT * FROM STATOWN; sera le suivant.

OWNER	NbMàJ
A	10
B	5

Donner accès à la vue STATOWN à tous les utilisateurs.

Correction Question 1

Et1	Et2
Z	Z

NB : Il fallait prendre en compte les violations de clé primaire.

Correction Question 2

1. `select nos from commande where nop='P1' and qte>10 order by qte;`
2. `select nop, poids*1.2 emballe from piece order by emballe desc;`
3. `select count(distinct nos) from commande;`
4. `select avg(qte) from commande where nop='P3';`
5. `select desi, poids from piece p, commande c where p.nop=c.nop and nos='S1';`
6. `select distinct c1.nos from commande c1, commande c2
where c1.nos=c2.no and c1.nop='P1' and c2.nop='P3' and c1.qte=c2.qte;`
7. `select nos from commande where nop='P3' and qte<
(select avg(qte) from commande where nop='P3');`
8. `select distinct nos from commande where qte > all
(select qte from commande where nos='S1');`
9. `select inti from service s where not exists
(select * from commande c where c.nos=s.nos);`
10. `select nop, sum(qte) from commande where nop in ('P1','P2','P3') group by nop;`

Correction Question 3

```
create or replace trigger emp_update
after update on emp
for each row

begin
    update stats set nbmaj=nbmaj+1 where nomtable='EMP';
end;
```

Correction Question 4

```
create view statown as
select owner, sum(nbmaj) totmaj
from stats, all_tables
where nomtable=table_name
group by owner;

grant select on statown to public;
```