

Examen (Durée 1h30)

**Exercice 1 :** Temps estimé (45 mn à 50 mn ) barème : 11 pts

Soit le schéma SQL3 (Base Ex1) d'une base de données RO qui gère les prestations de services et contrôles d'un garage automobile. Donnez les requêtes suivantes : **(-0.25 sur chaque erreur, ajout , syntaxe erronée ou requête mal formulée)**

1. Créer la table **GarageAuto**. 1 pt

Create table **GarageAuto** of TGarage nested table **Controles** store as C  
0.5 0.5

On suppose que la table **Vehicule** est créée et qu'elle contient les voitures : (1, '13 322 14', 'Kia'), (2, '13 419 35', 'Renault')

2. Insérer le garage 'ReparAuto' dont l'adresse et les prestations et le controle sont dans la collection **GarageAuto**. Le contrôle concerne le vehicule 1 ( les infos sont dans la collection **CtrAuto**). 2 pt

insert into garage values (1,'Terki','pasteur',TPrestations('Vida', 'chan'),  
TEnsControles(TControle(1,(select ref(r) from vehicule r where num=1),sysdate,'freinage')  
) ) 0.5 0.5 0.5

3. Modifier par « suspension » le contrôle du véhicule 1 effectué par 'ReparAuto' . 1 pt

Update table(select controles from garage where Nomg =1) c set c.controle =  
'suspension' where c.vehicule.num=1 0.25 0.5 0.25

4. Donner le nombre de véhicules qui ont passé un contrôle de suspension pour chaque garage 1.5pt

select g.nomG, count(\*)  
from garage g, table(g.controles) c  
where c.controle = 'suspension'  
group by g.nomG 0.25 0.25 0.25 0.25

5. Supprimer tous les contrôles (même l'entrée à la table imbriquée) du garage ReparAuto 1pt

UPDATE garage 0.25  
set controles = null 0.5  
where Nomg =1 0.25

6. Ajouter le contrôle Freinage au vehicule 2 effectué aujourd'hui par le garage ReparAuto 1.5 pt

UPDATE garage 0.25  
set controles = TEnsControles(TControle(1,(select ref(r) from vehicule r where  
num=2), sysdate,'freinage')) 1  
where Nomg =1 0.25

7. **Trigger** (4 pt): Soit la base relationnelle qui gère les prestations de services d'un garage automobile

**Garage** (NumG, Responsable, Adresse, nbPr, prixTotal) prixTotal est la somme des prix de toutes les prestations  
**Prestation** (NumPr, nomPr, prix) nomPr peut être Vidange ; Contrôle Freinage, . .  
**CtrVehicule** (NumCtrl, Matricule, NumG# , NumPr#, dateCtr)

Donner le déclencheur qui met à jour automatiquement le champ prixTotal

#### Solution avec declencheur Global

```
create or replace trigger TRGlob
After insert or update or delete on CtrVehicule
Begin
Update garage g
Set prixTotal = (Select sum(prix) from CtrVehicule C, Prestation P where g.NumG = c.NumG
and c.NumPr = P.NumPr);
end;
```

#### Solution avec declencheur Ligne (-1 pt pour ceux qui n'ont pas considéré le delete et update) Declaration sur 1 pt

```
create or replace trigger TRLigne
before insert or update of NumG, NumPr or delete on CtrVehicule
for EACH ROW
declare P number;
```

Begin

/\* je n'ai pas pris en considération le cas de modification de NumPr \*/ 1pt pour  
chaque opération

```
if inserting or updating then
select prix into P from Prestation where NumPr = :new.NumPr;
Update garage g
Set prixTotal = prixTotal + P
where NumG = :new.NumG;
end if;
```

```
if deleting or updating then
select prix into P from Prestation where NumPr = :old.NumPr;
Update garage g
Set prixTotal = prixTotal - P
where NumG = :old.NumG;
end if;
end;
```

#### Solution avec trigger ligne mais avec sum (non recommandée) car elle ne peut être exécutée que pour Insert et pas pour le delete et update (Table mutante) Je l'ai quand même accepté avec restrictions

```
create or replace trigger TRLigne
before insert on CtrVehicule
for EACH ROW
declare P number;
Ptot number;
Begin
select prix into P from Prestation where NumPr = :new.NumPr;
Select sum(prix)+P into Ptot from CtrVehicule C, Prestation P where c.NumG = :new.NumG and
C.NumPr= p.NumPr;
Update garage g
Set prixTotal = Ptot
where NumG = :new.NumG;
end;
```

**Exercice 2 :** Temps estimé ( 35 à 40 mn) barème : 9 pts

Soit la base MongoDB qui gère les prestations de services et contrôles d'un garage automobile.  
Donner les requêtes qui permettent de :

1. Afficher les noms des garages qui ont effectué un contrôle de freinage (sans agrégations) **1.5 pt**  
`db.ctrlAuto.find({"Controles.Controle" : "Freinage"}, {"Garage" : 1, _id : 0})`  
0.25                      0.5                      0.25                      0.25                      0.25
2. Donnez les garages qui proposent à la fois les prestations : "Vidange " et "changement de la courroie" **1.5 pt**  
`db.GarageAuto.find({"Prestations":{$all : [ "Vidange", "Changement de courroie"]}})`  
Toute solution sans le add n'a pas été acceptée
3. Ajouter le champ "HoraireFermeture" : "17h" pour le garage ReparAuto de freinage **1.5 pt**  
`db. GarageAuto.update({"NomGarage":"ReparAuto"},{$set:{ "HoraireFermeture" : "17h"}})` 0.25                      0.5                      0.5                      0.25
4. Donner, pour chaque garage, un tableau de véhicules par contrôle. **(2 pts)**.  
`db.ctrlAuto.aggregate([{$unwind:"$Controles"},{$group:{_id:{garage:"$Garage", control:"$Controles.Controle"}, vehicules:{$addToSet:"$Vehicule"}}}])`  
0.25                      0.5                      0.25
5. Donner l'équivalent de la requête **(1.5 pts)**.  
`Db.ctrlAuto.aggregate([`                      - 0.5 pour ceux qui ont inversé  
`{ $lookup:`  
`{ from: "garageAuto",`  
`localField: "garage",`  
`foreignField: "nomgarage",`  
`as: "info" } ]]`