

Contrôle : Administration des Bases de Données

Soit le schéma relationnel de la base de données de gestion des associations :

Association (CodeAss, NomAss, DateCreation, Type, MatriculePresident#, CodeCommune#)

Membre (Matricule, Nom, CodeAss#, CodeCommune#)

Commune (CodeCommune, NomCommune, Daira)

Donner les requêtes SQL suivantes :

1. Décider et justifier la stratégie des clé étrangères correspondantes, en cas de démission d'un président d'association et en cas de dissolution de l'association. **(1 Pts)**
 - Pour **MatriculePresident** → **On Delete Set Null** : Si un président démissionne d'une association, le SGBD va autoriser la suppression de la table MEMBRE, avec remplacement de l'attribut concerné dans la table ASSOCIATION par « NULL », en conservant très naturellement l'association avec toutes ses informations en attendant l'élection d'un nouveau président.
 - Pour **CodeAss (Membre)** → **On Delete Cascade** : Si une association est dissoute, elle va être supprimée de la table ASSOCIATION, en supprimant également tous les membres de cette association de la table MEMBRE du fait que leur présence dans la BD n'a aucun sens puisque leur association n'existe plus.
2. Créer les tables de la BD en respectant l'ordre de création. **(2 Pts)**
 - Create Table COMMUNE (.....) ;
 - Create Table ASSOCIATION (.....) ;
 - Create Table MEMBRE (.....) ;
 - Alter Table ASSOCIATION Add Constraint fk_presid Foreign Key MatriculePresident References MEMBRE (Matricule);
 - Alter Table MEMBRE Add Constraint fk_ass Foreign Key CodeAss References ASSOCIATION (CodeAss);
3. Définir les contraintes qui imposent que : **(2 Pts)**
 - a. L'association doit avoir un Code.
 - Le Code d'association est une clé primaire, Not Null est assuré par le Primary Key : inutile de définir cette contrainte.
 - b. Le Matricule d'un membre doit être unique.
 - Le Matricule est une clé primaire, Unique est assuré par le Primary Key : inutile de définir cette contrainte.
 - c. Le Code d'association dans la table MEMBRE doit exister toujours.
 - Le Code d'association dans la table MEMBRE est une clé étrangère qui référence l'attribut CodeAss dans la table ASSOCIATION qui est clé primaire, Not Null est assuré par le Foreign key: inutile de définir cette contrainte.
 - d. Le Nom de la commune dans la table Commune par défaut est Tlemcen.
 - Alter table COMMUNE Modify Column NomCommune Varchar(15) Default 'Tlemcen' ;

4. Ajouter la contrainte suivante : le président d'une association doit être membre de cette association. **(1.5 Pts)**

- `Alter Table ASSOCIATION Add Constraint Ci_Presid Check (MatriculePresident = (Select Matricule From MEMBRE Where MEMBRE.CodeAss = ASSOCIATION.CodeAss)) ;`

5. Comment limiter l'adhésion d'un membre à une seule association seulement ? **(1.5 Pts)**

- Cette règle est déjà prise en charge par la première forme normale (1FN) du modèle relationnel. Pour un membre : Matricule → CodeAss (un seul Code d'association)

6. Ajouter dans la table Commune la colonne Nombre_Ass puis insérer dans cette colonne le nombre d'associations de cette commune. **(1.5 Pts)**

- `ALTER TABLE COMMUNE
ADD Nombre_Ass Number(3) ;`
- `UPDATE COMMUNE
SET Nombre_Ass = SELECT Count(*) FROM ASSOCIATION GROUP BY CodeCommune
HAVING COMMUNE.CodeCommune = ASSOCIATION.CodeCommune ;`

7. Lister les couples de Noms des membres de la même association avec le Code association correspondant triés sur ce dernier. **(1.5 Pts)**

- `SELECT M1.Nom , M2.Nom, CodeAss FROM MEMBRE AS M1 , MEMBRE AS M2
WHERE M1.CodeAss=M2.CodeAss AND M1.Matricule<>M2.Matricule ORDER BY CodeAss;`

8. Donner les types d'associations qui existent dans toutes les communes. **(1.5 Pts)**

- `SELECT Type
FROM ASSOCIATION
GROUP BY Type
HAVING COUNT(DISTINCT CodeCommune)) = (SELECT COUNT(*) FROM COMMUNE)`

9. Afficher tous les Noms des communes avec les Noms des associations éventuelles. **(1.5 Pts)**

- `SELECT NomCommune , NomAss
FROM COMMUNE LEFT JOIN ASSOCIATION
ON COMMUNE.CodeCommune= ASSOCIATION.CodeCommune;`

10. Créer une vue VUE-MGN, ne permettant de voir que les Noms et les Codes communes des membres résidant à Maghnia (code=04) , Ghazaouet (code=22) ou Nedroma (code=15) sans autoriser la mise à jour à travers cette vue. **(2 Pts)**

```
CREATE VIEW VUE-MGN  
AS SELECT Nom, CodeCommune FROM MEMBRE  
WHERE CodeCommune ='04' OR CodeCommune ='22' OR CodeCommune ='15'  
WITH READ ONLY
```

11. Créer une vue ASS-SCR, qui affiche les noms d'associations et leurs dates de création, de types « Sport », « Cult » ou « Relig », qui va permettre par la suite d'insérer de nouvelles associations de mêmes types. **(2 Pts)**

```
CREATE VIEW ASS-SCR  
AS SELECT NomAss, DateCreation FROM ASSOCIATION  
WHERE Type ='Sport' OR Type='Cult' OR Type='Relig'  
WITH CHECK OPTION
```

12. Créer un Snapshot ASS-1617, qui se renseigne immédiatement, des associations de tous types à l'exception des associations « Quartier » ou « Parent_Elev » créées en 2016 ou 2017, et qui est mis à jour mensuellement. (2 Pts)

```
CREATE MATERIALIZED VIEW ASS-1617  
BUILD Immediate  
REFRESH START sysdate NEXT sysdate+30  
AS SELECT * FROM ASSOCIATION  
WHERE DateCreation > '31/12/2015' AND DateCreation < '01/01/2018'  
AND Type <>'Quart' AND Type<>'Parent_Elev';
```

Mr MATALLAH H