

Contrôle : Administration des BD

Soient les deux tables ETUDIANT et FILIERE définies comme suit :

ETUDIANT (Matricule, Nom, Prénom, Année_Bac, Wilaya, CodeFil*)

FILIERE (CodeFil, Nom_Fil, Description, Matricule_Délégué*)

Les clés primaires sont soulignées, les clés étrangères sont suivies d'une *

Donner les requêtes SQL suivantes :

1. Décider et justifier la stratégie en cas de mise-à-jour ou de suppression des lignes référencées. **(2 Pts)**
 - Pour **CodeFil → On Update Cascade** : Si une filière contenant déjà des étudiants, est modifiée, dans la table filière, tous les étudiants inscrits dans cette filière, doivent suivre le nouveau code filière.
 - Pour **CodeFil → On Delete No Action** : Si une filière contenant déjà des étudiants, veut être supprimée de la table filière, le SGBD doit interdire cette tentative de suppression pour deux raisons : un étudiant ne doit pas se retrouver non inscrit dans aucune filière d'une part, d'autre part on veut conserver l'historique de toutes les filières qui existaient dans la BDD scolarité.
 - Pour **Matricule_Délégué → On Update Cascade** : Si un étudiant est modifié dans la table étudiant, et il s'avère que cet étudiant est un délégué d'une filière, la filière correspondante doit être modifiée également.
 - Pour **Matricule_Délégué → On Delete Set Null** : Si un étudiant est supprimé de la table étudiant, soit pour abandon ou pour d'autres raisons, le SGBD va autoriser la suppression de l'étudiant avec remplacement de l'attribut concerné dans la table filière par « Null », en conservant très naturellement la filière contenant d'autres étudiants sans délégué, en attendant de choisir un nouveau délégué.
2. Créer les relations de la base avec toutes les contraintes d'intégrité. **(2 Pts)**
 - Create Table Etudiant (
Matricule Number(10) Primary Key,
Nom Varchar(20),
Prenom Varchar(30),
Annee_Bac Number(4),
Wilaya Varchar(15),
CodeFil Varchar(10)) ;
 - Create Table Filiere (
CodeFil Varchar(10) Primary Key,
Nom_Fil Varchar(20),
Description Varchar(50),
Matricule_Delegate Number(10),
Constraint fk_del Foreign Key Matricule_Delegate References Etudiant (Matricule) On Update Cascade On Delete Set Null);
 - Alter Table Etudiant Add Constraint fk_fil Foreign Key CodeFil References Filiere (CodeFil) On Update Cascade);

3. Définir les contraintes qui imposent que : **(2 Pts)**

- a. L'étudiant doit avoir un Matricule et un Nom et que la Wilaya par défaut est Tlemcen.
- Le Matricule est une clé primaire, Not Null est assuré par le Primary Key : inutile de définir cette contrainte
 - Alter Table Etudiant Modify [column] Nom Varchar(20) Not Null ;
 - Alter table Etudiant Modify [column] Wilaya Varchar(15) Default 'Tlemcen' ;
- b. Le Matricule de l'étudiant délégué dans la table FILIERE doit exister toujours et que le nom de d'une filière est différent aux autres.

- Matricule_Delegate est une clé étrangère qui référence l'attribut Matricule qui est clé primaire, Not Null est assuré par le Foreign key: inutile de définir cette contrainte.

- Alter Table Filiere Modify [column] Un_NomFil Varchar(20) Unique ;

Ou

- Alter Table Filiere Add Constraint Un_NomFil Unique (Nom_Fil) ;

4. Supprimer la colonne Wilaya dans la table ETUDIANT. Vérifier la suppression. Rétablir cette colonne. **(1 Pt)**

- ALTER TABLE Etudiant
DROP COLUMN Wilaya;

- DESCRIBE Etudiant ;

- ALTER TABLE Etudiant
ADD Wilaya VARCHAR (15) ;

5. Fixer l'intervalle des années de bac acceptées, qui doit être compris entre 2005 et 2015. **(1 Pt)**

- Alter Table Etudiant Add Constraint Ci_AnBac Check (Annee_Bac Between 2005 And 2015) ;

6. On veut limiter les wilayas des étudiants acceptés à : Tlemcen, Ain Temouchent et Naama. Comment faire? **(1 Pt)**

- Alter table Etudiant Add Constraint Ci_Wil Check (Wilaya in ('Tlemcen', 'Ain Temouchent', 'Naama')) ;

Ou

- Alter table Etudiant Add Constraint Ci_Wil Check (Wilaya='Tlemcen' or Wilaya='Ain Temouchent' or Wilaya='Naama') ;

7. Ajouter la contrainte suivante : l'étudiant délégué d'une Filière doit être inscrit dans cette Filière. **(1 Pt)**

- Alter Table Filiere Add Constraint Ci_Deleg Check (Matricule_Delegate= [ou In] (Select Matricule From Etudiant Where Etudiant.CodeFil=Filiere.CodeFil)) ;

8. Supposons qu'un utilisateur veut supprimer la Filière Informatique qui contient 90 étudiants et sachant qu'on n'a pas décidé sur la stratégie, quel serait la réponse du SGBD ? Quelles seront les solutions possibles à ce problème ? **(1.5 Pt)**

Les réponses seront :

- Impossible de supprimer une ligne la table Filière qui est référencée par la table Etudiant

Les solutions possibles à ce problème :

- Supprimer d'abord les lignes de la table Etudiant qui font référence à la filière informatique

Ou

- Désactiver la contrainte de clé étrangère temporairement

Ou

- Changer de stratégie de suppression (On Delete)

9. Une règle de gestion a été dictée par l'administration et qui doit être assurée par le DBA, c'est l'obligation d'inscription d'un étudiant dans une seule filière. Comment la prendre en charge ? **(1 Pt)**
- Cette règle est déjà prise en charge par la première forme normale (1FN) du modèle relationnel. Pour un étudiant : Matricule → CodeFil (un seul code filière)
10. Les agents de saisie n'ont pas pu saisir des étudiants qui n'ont pas encore été affectés à des filières. Comment régler ce problème provisoirement. **(1.5 Pt)**
- Désactivation de la contrainte : `Alter Table Etudiant Disable Constraint fk_fil ;`
 - Saisie des étudiants sans CodeFil : `Insert Into Etudiant.....`
 - Réactivation de la contrainte : `Alter Table Etudiant Enable NoValidate Constraint fk_fil ;`
11. Ajouter dans la table FILIERE l'attribut Nbr_Etud puis insérer dans cet attribut le nombre d'étudiants inscrits dans chaque Filière. **(1.5 Pt)**
- `ALTER TABLE Filiere
ADD Nbr_Etud Number(4) ;`
 - `UPDATE Filiere
SET Nbr_Etud = SELECT Count(*) FROM Etudiant GROUP BY CodeFil
HAVING Filiere.CodeFil = Etudiant.CodeFil;`
12. Supprimer tous les étudiants de la Filière Physiques. **(1 Pt)**
- `DELETE FROM Etudiant
WHERE Etudiant.CodeFil = [ou In] (SELECT Filiere.CodeFil FROM Filiere WHERE
Nom_Fil='Physiques');`
13. Lister les couples de noms d'étudiants de la même Filière. **(1 Pt)**
- `SELECT E1.Nom , E2.Nom FROM Etudiant AS E1 , Etudiant AS E2
WHERE E1.CodeFil=E2.CodeFil AND E1.Matricule< >E2.Matricule;`
14. Afficher tous les noms des étudiants inscrits et non encore inscrits dans une Filière avec leurs noms de Filières correspondantes. **(1 Pt)**
- `SELECT Nom, Nom_Fil
FROM Etudiant JOIN LEFT Filiere
ON Etudiant.CodeFil = Filiere.CodeFil;`
15. Donner la liste des étudiants qui sont inscrits dans la même Filière que l'étudiant « INF144 » **(1.5 Pt)**
- `SELECT *
FROM Etudiant
WHERE CodeFil =(SELECT CodeFil FROM Etudiant WHERE Matricule='INF144');`

Mr MATALLAH H