**Série 1 : Les Contraintes d’intégrités**

**Exercice 1 :**

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d’étudiants :

ETUDIANT (***NEtudiant***, Nom, Prénom)

MATIERE (***CodeMat***, LibelleMat, CoeffMat)

EVALUER (**NEtudiant**, **CodeMat**, DateExamen, Note)

Questions :

1. Créer la base de données avec le nom « Ecole »;

create database ecole2;

go

use ecole2;

go

1. Créer les tables avec les clés primaires sans spécifier les clés étrangères ;

create table etudiant

(nEtudiant int identity(1,1) primary key,

nom varchar(30), prenom varchar(30));

create table MATIERE (

CodeMat int identity(1,1) primary key,

LibelleMat varchar(30),

CoeffMat int);

create table EVALUER (

NEtudiant int,

CodeMat int,

DateExamen date,

Note float,

constraint pkEvaluer primary key (NEtudiant, CodeMat))

1. Ajouter les clés étrangères à la table EVALUER ;   
   alter table evaluer add constraint fkNEtudiant foreign key (NEtudiant) references etudiant(nEtudiant);   
     
   alter table evaluer add constraint fkCodeMat foreign key (CodeMat) references matiere(codeMat);
2. Ajouter la colonne Groupe dans la table ETUDIANT: Groupe NOT NULL ;   
   alter table etudiant add groupe varchar(50) not null;
3. Ajouter la contrainte unique pour l’attribut (LibelleMat) ;   
   alter table matiere add constraint matiereUnique unique(LibelleMat);
4. Ajouter une colonne Age à la table ETUDIANT, avec la contrainte (age >16) ;   
   alter table etudiant add age int; --check (age>16);

alter table etudiant add constraint ageplus check (age>16);

ou bien

alter table etudiant add age3 int constraint checkAge3 check (age3 >16);

1. Ajouter une contrainte sur la note pour qu’elle soit dans l’intervalle (0-20) ;

alter table evaluer add constraint noteinterval check (note between 0 and 20);

pour désactiver la check ou la réactiver on utilise nocheck constraint ou check contraint

exemples

alter table evaluer nocheck constraint checknote;

alter table evaluer check constraint checknote;

1. Remplir les tables par les données ;

insert into etudiant values ('a','a','tdi',19);

insert into etudiant values ('b','b','tdi',15);

insert into etudiant values ('c','c','tdi',19);

insert into MATIERE values ('progr', 5);

insert into EVALUER values (1,1,'05/05/2018',12.5);

--probleme de nEtudiant

insert into EVALUER(CodeMat, NEtudiant,DateExamen,Note) values (1,2,'05/05/2018',12.5);

insert into EVALUER(CodeMat, NEtudiant,DateExamen,Note) values (1,3,'05/05/2018',12.5);

--probleme de note

insert into EVALUER(CodeMat, NEtudiant,DateExamen,Note) values (1,4,'05/05/2018',22);

**Exercice 2 :**

Soit le schéma relationnel suivant :

AVION ( **NumAv**, TypeAv, CapAv, VilleAv)

PILOTE **(NumPil**, NomPil,titre, VillePil)

VOL (**NumVol**, VilleD, VilleA, DateD, DateA, NumPil#,NumAv#)

Travail à réaliser :

**À l'aide de script SQL:**

1. Créer la base de données sans préciser les contraintes de clés.
2. Ajouter les contraintes de clés aux tables de la base.
3. Ajouter des contraintes qui correspondent aux règles de gestion suivantes

* Le titre de courtoisie doit appartenir à la liste de constantes suivantes :

M, Melle, Mme.

* Les valeurs noms, ville doivent être renseignées.
* La capacité d’un avion doit être entre 50 et 100.

1. Ajouter la colonne ‘date de naissance’ du pilote : **DateN**
2. Ajouter une contrainte qui vérifie que la date de départ d’un vol est toujours inférieure ou égale à sa date d’arrivée.
3. Supprimer la colonne VilleAv
4. Supprimer la table PILOTE
5. Remplir la base de données pour vérifier les contraintes appliquées.

/\*

AVION ( NumAv, TypeAv, CapAv, VilleAv)

PILOTE (NumPil, NomPil,titre, VillePil)

VOL (NumVol, VilleD, VilleA, DateD, DateA, NumPil#,NumAv#)

Travail à réaliser :

À l'aide de script SQL: \*/

--1. Créer la base de données sans préciser les contraintes de clés.

create database avions;

go

use avions;

go

create table AVION (

NumAv int identity(1,1),

TypeAv varchar(30),

CapAv int,

VilleAv varchar(30)

);

create table PILOTE (

NumPil int identity(1,1),

NomPil varchar(30),

titre varchar(30),

VillePil varchar(30)

) ;

create table VOL (

NumVol int identity(1,1),

VilleD varchar(30),

VilleA varchar(30),

DateD date ,

DateA date,

NumPil int,

NumAv int

);

--2. Ajouter les contraintes de clés aux tables de la base.

alter table pilote add constraint pkPilote primary key (numPil);

alter table avion add constraint pkAvion primary key (numAv);

alter table vol add constraint pkvol primary key (numVol);

alter table vol add constraint fkVolPilote foreign key (numPil) references pilote(numPil);

alter table vol add constraint fkVolAvion foreign key (numAv) references avion(numAv);

--3. Ajouter des contraintes qui correspondent aux règles de gestion suivantes

-- Le titre de courtoisie doit appartenir à la liste de constantes

suivantes :--M, Melle, Mme.

alter table pilote add constraint titrecheck check

(titre in('M', 'Melle', 'Mme'))

insert into pilote values ('a','f','t');

--desactiver une check

alter table pilote nocheck constraint titrecheck;

--activer une check

alter table pilote check constraint titrecheck;

-- Les valeurs noms, ville doivent être renseignées.

alter table AVION alter column VilleAv varchar (20) not null;

alter table PILOTE alter column NomPil varchar (10)not null;

alter table PILOTE alter column VillePil varchar (20) not null;

alter table VOL alter column VilleD varchar (20) not null;

alter table VOL alter column VilleA varchar (20) not null ;

-- La capacité d’une avion doit être entre 50 et 100.

alter table AVION add constraint CapChecker check (CapAv between 50 and 100);

--4. Ajouter la colonne ‘date de naissance’ du pilote : DateN

alter table Pilote add DateN date ;

--5. Ajouter une contrainte qui vérifie que la date de départ d’un vol est toujours inférieure ou égale à sa date d’arrivée.

alter table vol add constraint checkdate check (DateD<DateA);

--6. Supprimer la colonne VilleAv

alter table AVION drop column VilleAv

--7. Supprimer la table PILOTE

alter table VOL drop constraint fkVolPilote;

drop table PILOTE;

--8. Remplir la base de données pour vérifier les contraintes appliquées.