

Exercice sur les Bases de Données

Enoncé :

Contexte professionnel :

Dans le cadre du CFA INSTA, un professeur connu par un nom, un prénom et un diplôme, peut emprunter un matériel (ordinateur, vidéo projecteur, etc...) décrit par un numéro, une désignation, et une date d'achat et un état. Le matériel peut être emprunter plusieurs fois. A chaque emprunt, on stocke la date de l'emprunt et la date prévu de retour. Un professeur assure une seule matière à une classe. Une classe est connu par un code, une désignation et une salle. Une matière est connue par un code et un libellé. Le professeur est affectée à une classe pour une matière et on enregistre l'année universitaire.

Une classe prépare un diplôme, un diplôme est connu par un numéro et un libellé. Une classe est composé par plusieurs étudiants, connus par leur nom, prenom, e-mails, adresse et date de naissance.

- Elaborer un MCD
- Elaborer le MLDR sous forme textuelle
- Installer la base de donnée sous MySQL
- Réaliser les requêtes suivantes :
 - Ajouter un professeur ben, oka, bac
 - Ajouter un professeur Launay, Fred, BTS
 - Ajouter une matière Français
 - Ajouter une matière Maths
 - Ajouter une matière Informatique
 - Ajouter diplôme License, BTS SIO, Master
 - Ajouter Matériel Vidéos projecteur, 2018-05-15, neuf
 - Ajouter Matériel Ordinateur, 2017-01-02, en bon Etat
 - Ajouter Matériel imprimante, 2015-05-14, en mauvais Etat
 - Ajouter Classe Promo-193, salle 5 et promo 194 salle 3
 - Ajouter Etudiant :

- Pierre Olivier, **p@gmail.com**, 20 rue du Paris, 2000-05-06
- François Marc, **f@gmail.com**, 20 rue de Lyon, 2001-06-05
- Marie France, **m@gmail.com**, 20 rue de Lille, 2002-02-02

◦ Ajouter a Emprunter :

- 2018-10-08, 2018-10-16
- 2018-11-02, 2018-12-02
- 2018-12-05, 2018-12-18

◦ Ajouter a Affecter

- 2018-2019
- 2018-2019
- 2018-2019

Création du MCD



MCD

Création du MLDR



MLD

Version Textuelle

```
MLDR
Professeur(idprof,nom,prénom,diplôme)
Classe(code, désignation, salle, NuméroDiplome#)
Matiere(code, libelle)
Materiel(numero, designation, date_achat, etat)
Etudiant(idEtudiant, nom, prénom, mail, adresse, annee de naissance, code#)
diplôme(numero, libelle)
```

MySQL Version

```
#-----
#      Script MySQL.
#-----
```

```
#-----  
# Table: Professeur  
#-----
```

```
CREATE TABLE Professeur(  
    ID_Prof Int NOT NULL ,  
    Nom      Varchar (50) NOT NULL ,  
    Prenom   Varchar (50) NOT NULL ,  
    Diplome  Varchar (50) NOT NULL  
    ,CONSTRAINT Professeur_PK PRIMARY KEY (ID_Prof)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: Matériel  
#-----
```

```
CREATE TABLE Matériel(  
    Numero      Int NOT NULL ,  
    Designation Varchar (50) NOT NULL ,  
    Date_Achat  Date NOT NULL ,  
    Etat        Varchar (50) NOT NULL  
    ,CONSTRAINT Matériel_PK PRIMARY KEY (Numero)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: Diplôme  
#-----
```

```
CREATE TABLE Diplome(  
    Numero Int NOT NULL ,  
    Libelle Varchar (50) NOT NULL  
    ,CONSTRAINT Diplome_PK PRIMARY KEY (Numero)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: Classe  
#-----
```

```
CREATE TABLE Classe(  
    Code      Int NOT NULL ,  
    Designation Varchar (50) NOT NULL ,  
    Salle      Int NOT NULL ,
```

```

        Numero      Int NOT NULL
    ,CONSTRAINT Classe_PK PRIMARY KEY (Code)

    ,CONSTRAINT Classe_Diplome_FK FOREIGN KEY (Numero) REFERENCES Diplome(Numero)
)ENGINE=InnoDB;

```

```

#-----
# Table: Etudiant
#-----

```

```

CREATE TABLE Etudiant(
    ID_Etudiant      Int NOT NULL ,
    Nom              Varchar (50) NOT NULL ,
    Prenom           Varchar (50) NOT NULL ,
    Email            Varchar (50) NOT NULL ,
    Adresse          Varchar (50) NOT NULL ,
    Date_de_Naissance Date NOT NULL ,
    Code             Int NOT NULL
    ,CONSTRAINT Etudiant_PK PRIMARY KEY (ID_Etudiant)

    ,CONSTRAINT Etudiant_Classe_FK FOREIGN KEY (Code) REFERENCES Classe(Code)
)ENGINE=InnoDB;

```

```

#-----
# Table: Matière
#-----

```

```

CREATE TABLE Matiere(
    Code      Int NOT NULL ,
    Libelle   Varchar (50) NOT NULL
    ,CONSTRAINT Matiere_PK PRIMARY KEY (Code)
)ENGINE=InnoDB;

```

```

#-----
# Table: Emprunter
#-----

```

```

CREATE TABLE Emprunter(
    Numero      Int NOT NULL ,
    ID_Prof     Int NOT NULL ,
    Date_Emprunt Date NOT NULL ,
    Date_Retour  Date NOT NULL
    ,CONSTRAINT Emprunter_PK PRIMARY KEY (Numero,ID_Prof)

```

```

,CONSTRAINT Emprunter_Materiel_FK FOREIGN KEY (Numero) REFERENCES Materiel(Numero)
,CONSTRAINT Emprunter_Professeur0_FK FOREIGN KEY (ID_Prof) REFERENCES Professeur(ID_P
)ENGINE=InnoDB;

#-----
# Table: Affecter
#-----

CREATE TABLE Affecter(
    Code          Int NOT NULL ,
    ID_Prof       Int NOT NULL ,
    Code_Matiere  Int NOT NULL ,
    Annee_Univ    Varchar (50) NOT NULL
,CONSTRAINT Affecter_PK PRIMARY KEY (Code,ID_Prof,Code_Matiere)

,CONSTRAINT Affecter_Classe_FK FOREIGN KEY (Code) REFERENCES Classe(Code)
,CONSTRAINT Affecter_Professeur0_FK FOREIGN KEY (ID_Prof) REFERENCES Professeur(ID_P
,CONSTRAINT Affecter_Matiere1_FK FOREIGN KEY (Code_Matiere) REFERENCES Matiere(Code)
)ENGINE=InnoDB;

# -----
# Instructions
# -----

insert into Professeur values (null, "ben","oka","bac"), (null,"Launay","Fred","BTS");
insert into Matiere values (null, "français"), (null, "Maths"), (null,"Informatique");
insert into Diplome values (null, "bac"), (null,"BTS"), (null,"License");
insert into Materiel values (null, "ordinateur", "2017-01-02", "En bon etat"),
                           (null, "Vidéo Projecteur", "2018-05-15", "Neuf"),
                           (null, "imprimante", "2015-05-14", "En mauvais etat");
insert into Classe values (null, "Promo-193", "Salle-5",1), (null, "Promo-194", "Salle-3
insert into Etudiant values (null, "Pierre","Olivier","p@gmail.com","20 rue de Paris","20
                           (null, "Francois","Marc","f@gmail.com","20 rue de Lyon","200
                           (null, "Marie","France","m@gmail.com","20 rue de Lille","200
insert into Emprunter values (1,1,"2018-10-08","2018-10-16"),
                           (2,1,"2018-11-02","2018-12-02"),
                           (1,2,"2018-12-05","2018-12-18");
insert into Affecter values (1,1,1,"2018-2019"),(1,1,2,"2018-2019"), (2,2,2,"2018-2019").

```

Liste des requêtes de sélection

- Quels sont les professeurs (noms et prénoms) du CFA qui sont enregistrés

- Quels sont les étudiants (nom, prénom, email) qui sont nés en l'an 2000 ?
- Quels sont les classes (nom) qui sont affectés à la "salle 5" ?
- Quels sont les matériels (désignation, état) achetés avant 2018 ?
- Quels sont les professeurs ayant le diplôme "BTS"?

```
SELECT nom, prenom FROM Professeur;
```

```
SELECT nom as NOM_Etudiant, prenom as Prenom_Etudiant, email as Email_Etudiant  
FROM etudiant  
WHERE year(Date_de_Naissance)="2000";
```

```
SELECT Designation as Designation_Classe  
FROM Classe  
WHERE salle = "salle-5"
```

```
SELECT Designation as Designation_Materiel, Etat as Etat_Materiel  
FROM Materiel  
WHERE year(date_achat)<"2018"
```

```
SELECT * from Professeur;  
FROM Professeur  
WHERE diplome = "BTS"
```

Requête de modification de la structure de la table

- Ajouter à la table materiel le prix d'achat du materiel de type float (9.2)
- Ajouter à la table étudiant le montant de la scolarité de type float (5.2)
- Ajouter à la table étudiant un attribut de type booléen (Vrai si alternant Faux si initial)
- Supprimer de la table Etudiant l'attribut Adresse
- Ajouter à la table classe l'attribut nb Etudiant de type entier.

```
alter table Materiel add prix_achat float (9.2);  
alter table Etudiant add prix_scolarité float (5.2);  
alter table Etudiant add type_etudiant boolean;  
alter table Etudiant drop adresse;  
alter table classe add nbr_etudiant int (3);
```

Requêtes de modification de la base de données

- Il y a trois matériels avec les prix d'achat suivant 1200, 500, 900.
- Il y a trois étudiants : 2 sont alternants et leur montant de scolarité est de 0. Le troisième est en initial et son montant est de 4500 euros.
- Les 3 classes ont 25 élèves
- L'état du vidéo projecteur c'est dégradé.
- La classe de la salle 5 est affecté à la salle 7.

```
update Matériel set prix_achat = 1200 where numero = 1;
update Matériel set prix_achat = 500 where numero = 2;
update Matériel set prix_achat = 900 where numero = 3;

update Etudiant set prix_scolarité = 0, type_etudiant = 1 where id_etu <=2;
update Etudiant set prix_scolarité = 4500, type_etudiant = 0 where id_etu =3;

update classe set nbr_etudiant = 25;

update Matériel set etat = "etat dégradé" where numéro = 2;

update Classe set salle = "Salle-7" where salle = "Salle-5";
```

Requête mathématiques

- Quel est le montant le plus élevé des achats de matériels ?
- Quel est le montant moyen des achats de matériels ?
- Quel est le total des montants de scolarité des étudiants initiaux
- Combien y a t'il d'étudiants alternants ?
- Quelle est la capacité totale d'accueil de toutes les classes ?

```
select max(prix_achat) as montant_eleve from materiel;

select AVG(prix_achat) as moyenne_materiels from materiels;

select sum(montant_scolarite) as total from etudiant where type_etudiant = false;

select count(*) as nb_etudiant from etudiant where type
=1;

select sum(nbr_etudiant) as capacity totale from classe;
```

Requêtes entre deux tables :

- Afficher les infos suivantes :
 - nom etudiant, prenom etudiant designation classe, salle
 - Designation classe, salle, libelle Diplome
 - Designation materielle, etat, date emprunt, date retour
 - Nom prof, prénom prof, date emprunt, date date retour
- Lister les étudiants alternants (noms, prénoms) qui sont inscrits dans la classe "promo-193".

```
select e.nom, e.prenom, c.designation, c.salle
from etudiant e, classe c
where e.code_classe = c.code;
```

```
select c.designation, c.salle, d.libelle
from classe c, diplome d
where c.numero_diplome=d.numero
```

```
select m.designation , m.etat, e.date_emprunt, e.date_retour
from materiel m, emprunter e,
where m.numero=e.numero ;
```

```
select p.nom, p.prenom, e.date_emprunt, e.date_retour
from professeur p, emprunter e,
where p.id_prof=e.id_prof;
```

```
select e.nom, e.prenom, c.salle
from etudiant e, classe c
and c.designation="promo-193"
```

Les requêtes de jointure à plusieurs tables

- Afficher les infos suivantes :
 - nom prof, prenom prof, date emprunt, date retour, designation, materiel, etat materiel

```
select p.nom, p.prenom, e.date_emprunt, e.date_retour, m.designation, m.etat
from professeur p, emprunter e, materiel m
where p.id_prof=e.id_prof and e.numero=m.numero ;
```