

Exercice 1

Soit le modèle relationnel suivant relatif à une base de données sur des représentations musicales :

Représentation(N°représentation, Titre_représentation, Lieu)

Musicien(Nom, N°représentation) où N°représentation référence Représentation.N°représentation

Programmer(Date, N°représentation, Tarif) où N°représentation référence Représentation.N°représentation

Donnez les requêtes SQL correspondants aux questions ci-dessous :

1. Quels sont les titres des représentations.

```
SELECT Titre_représentation
FROM REPRESENTATION ;
```

2. Quels sont les titres des représentations ayant lieu à l'opéra Bastille.

```
SELECT Titre_représentation
FROM REPRESENTATION
WHERE Lieu="Opéra Bastille" ;
```

3. Quels sont les noms des musiciens et les titres des représentations auxquelles ils participent.

```
SELECT Nom, Titre_représentation
FROM Musicien, Représentation
WHERE Musicien.N°représentation = Représentation.N°représentation ;
```

4. Quels sont les titres des représentations, les lieux et les tarifs pour la journée du 12 mai 2005.

```
SELECT Titre_représentation, Lieu, Tarif
FROM Représentation, Programmer
WHERE Programmer.N°représentation = Représentation.N°représentation
AND Date='12/05/2005' ;
```

Exercice 2

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

Etudiant(N°Etudiant, Nom, Prénom)

Matiere(CodeMatiere, LibelléMatiere, CoefficientMatiere)

Evaluer(N°Etudiant, CodeMatiere, Date, Note) où N°Etudiant référence Etudiant.N°Etudiant et CodeMatiere référence Matiere.CodeMatiere

Donnez les requêtes SQL correspondants aux questions ci-dessous :

1. Quel est le nombre total d'étudiants de la promotion ?

```
SELECT COUNT(*) FROM Etudiant ;
```

2. Quelle est, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?

```
SELECT MIN(Note), MAX(Note) FROM Evaluer ;
```

3. Il est possible de créer une table en combinant une instruction **CREATE TABLE** et une instruction **SELECT** de la façon suivante : **CREATE TABLE UneTable AS (Instruction SELECT retournant une table)** la table *UneTable* possède alors la même structure et contient les mêmes valeurs que la table retournée par l'instruction **SELECT**. En utilisant cette technique, créez une table *MoyenneEtudiantMatiere*, à partir des tables *Evaluer*, *Matiere* et *Etudiant*, ayant comme attributs un numéro d'étudiant, un nom d'étudiant,

un prénom d'étudiant, le libellé d'une matière, le coefficient de cette matière et la moyenne de cet étudiant dans cette matière.

```
CREATE TABLE MoyenneEtudiantMatiere AS
SELECT Etudiant.N°Etudiant, Nom, Prénom, LibelléMatiere,
       CoefficientMatiere, AVG(Note) AS MoyEtuMat
FROM Evaluer, Matiere, Etudiant
WHERE Evaluer.CodeMatiere= Matiere.CodeMatiere
AND Evaluer.N°Etudiant = Etudiant.N°Etudiant
GROUP BY Etudiant.N°Etudiant, Nom, Prénom,
         LibelléMatiere, CoefficientMatiere ;
```

4. En utilisant la table *MoyenneEtudiantMatiere*, donnez les moyennes de la promotion par matière.

```
SELECT LibelléMatiere, AVG(MoyEtuMat)
FROM MoyenneEtudiantMatiere
GROUP BY LibelléMatiere ;
```

5. En utilisant la table *MoyenneEtudiantMatiere*, afficher pour chaque étudiant, son numéro, son nom, son prénom et sa moyenne générale.

```
SELECT N°Etudiant, Nom, Prénom,
       SUM(MoyEtuMat*CoefficientMatiere)/SUM(CoefficientMatiere)
       AS MoyGeneEtu
FROM MoyenneEtudiantMatiere
GROUP BY N°Etudiant, Nom, Prénom ;
```

Exercice 3

Soit le modèle relationnel suivant relatif à la gestion simplifiée des étapes du Tour de France 2010 :

Equipe(CodeEquipe, NomEquipe, NomDirecteurSportif)

Coureur(NuméroCoureur, NomCoureur, CodeEquipe, CodePays)

où CodeEquipe référence Equipe.CodeEquipe et CodePays référence Pays.CodePays

Pays(CodePays, NomPays)

Type_etape(CodeType, LibelléType)

Etape(NuméroEtape, DateEtape, VilleDépart, VilleArrivée, NbKm, CodeType)

où CodeType référence Type_etape.CodeType

Participer(NuméroCoureur, NuméroEtape, TempsRéalisé)

où NuméroCoureur référence Coureur.NuméroCoureur et NuméroEtape référence Etape.NuméroEtape

Attribuer_bonification(NuméroEtape, km, Rang, NbSecondes, NuméroCoureur)

où NuméroEtape référence Etape.NuméroEtape et NuméroCoureur référence Coureur.NuméroCoureur

Donnez les requêtes SQL correspondants aux questions ci-dessous :

1. Quelle est la composition de l'équipe « Quickstep » (Numéro, nom et pays des coureurs) ?

```
SELECT NumeroCoureur, NomCoureur, NomPays
FROM Coureur, Equipe, Pays
WHERE Equipe.CodeEquipe = Coureur.CodeEquipe
AND Coureur.CodePays = Pays.CodePays
AND NomEquipe = 'Quickstep' ;
```

2. Quel est le nombre de kilomètres total du Tour de France 2010 ?

```
SELECT SUM(NbKm) FROM Etape ;
```

3. Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?

```
SELECT SUM(NbKm)
FROM Etape, Type_etape
WHERE Etape.CodeType = Type_etape.CodeType
AND LibelléType = "Haute Montagne" ;
```

4. Quels sont les noms des coureurs qui n'ont pas obtenu de bonification ?

```
SELECT NomCoureur
FROM Coureur
WHERE NumeroCoureur NOT IN (SELECT NuméroCoureur
FROM Attribuer_bonification) ;
```

5. Quels sont les noms des coureurs dont le directeur sportif a un nom identique à la ville d'arrivée d'une étape où il a réalisé un temps inférieur à 5 heures ?

```
SELECT NomCoureur
FROM Coureur, Equipe, Etape, Participer
WHERE Coureur.CodeEquipe = Equipe.CodeEquipe
AND Participer.NuméroCoureur = Coureur.NuméroCoureur
AND Participer.NuméroEtape = Etape.NuméroEtape
AND NomDirecteurSportif = VilleArrivee
AND TempsRéalisé < 5:00:00 ;
```

6. Quels sont les noms des coureurs qui ont participé à **toutes** les étapes ?

```
SELECT NomCoureur
FROM Coureur, Participer
WHERE Participer.NuméroCoureur = Coureur.NuméroCoureur
GROUP BY Coureur.NuméroCoureur, NomCoureur
HAVING COUNT(NuméroEtape) = (SELECT COUNT(NuméroEtape) FROM Etape) ;
```

7. Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps de la table *Participer* ?

```
SELECT NomCoureur, CodeEquipe, CodePays, SUM(TempsRéalisé) AS Total
FROM Coureur, Participer
WHERE Coureur.NuméroCoureur = Participer.NuméroCoureur
AND NuméroEtape <= 13
AND Coureur.NuméroCoureur IN (SELECT NuméroCoureur
FROM Participer
WHERE NuméroEtape = 13)
GROUP BY Coureur.NuméroCoureur, NomCoureur, CodeEquipe, CodePays
ORDER BY Total ASC ;
```

8. Quels sont les noms des équipes ayant des coureurs de **tous** les pays ?

```
SELECT NomEquipe
FROM Equipe, Coureur
WHERE Equipe.CodeEquipe = Coureur.CodeEquipe
GROUP BY NomEquipe, Equipe.CodeEquipe
HAVING COUNT(DISTINCT CodePays) = (SELECT COUNT(CodePays)
FROM Pays) ;
```

9. Quel est, pour chaque coureur du tour de France 2010 (dont on donnera son numéro et son nom), le nombre total de kilomètres qu'il a parcouru ainsi que le kilométrage moyen des étapes auquel il a participé ?

```
SELECT Coureur.NuméroCoureur, NomCoureur, SUM(NbKm), AVG(NkKm)
FROM Coureur, Participer, Etape
WHERE Coureur.NuméroCoureur = Participer.NuméroCoureur
      AND Participer.NuméroEtape = Etape.NuméroEtape
GROUP BY Coureur.NuméroCoureur, NomCoureur ;
```

10. Pour chaque coureur ayant obtenu au moins une bonification, quel est le temps total de bonification qu'il a obtenu sur le tour de France 2010 ?

```
SELECT NomCoureur, SUM(NbSecondes)
FROM Coureur, Attribuer_bonification
WHERE Coureur.NuméroCoureur = Attribuer_bonification.NuméroCoureur
GROUP BY Coureur.NuméroCoureur, NomCoureur ;
```

Exercice 4

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

Etudiant(NumeroEtudiant, NomEtudiant, DateNaissance, Sexe)

Notes(NumeroEtudiant, NumeroMatiere, Note)

où NumeroEtudiant référence Etudiant.NumeroEtudiant et NumeroMatiere référence Matiere.NumeroMatiere

Enseignant(NumeroEnseignant, NomEnseignant, Grade, Ancienneté)

Matiere(NumeroMatiere, NomMatiere, Coefficient, NumeroEnseignant)

où NumeroEnseignant référence Enseignant.NumeroEnseignant

Donnez les requêtes SQL correspondants aux questions ci-dessous :

1. Quels sont les noms et numéros des étudiants qui n'ont pas de note en Sociologie ?

```
SELECT NumeroEtudiant, NomEtudiant
FROM Etudiant
WHERE NumeroEtudiant NOT IN
      (SELECT NumeroEtudiant
       FROM Notes AS N, Matiere AS M
       WHERE N.NumeroMatiere=M.NumeroMatiere
            AND NomMatiere='Sociologie') ;
```

2. Quel est le nom, le grade et l'ancienneté des enseignants qui enseignent dans plus d'une matière ?

```
SELECT NomEnseignant, Grade, Ancienneté
FROM Enseignant AS E, Matiere AS M
WHERE E.NumeroEnseignant=M.NumeroEnseignant
GROUP BY NomEnseignant, Grade, Ancienneté, Enseignant.NumeroEnseignant
HAVING COUNT(NumeroMatiere)>1 ;
```

3. Quelles sont les matières pour lesquelles la moyenne des notes est inférieure à 10 ? (Afficher le nom de l'enseignant correspondant, la matière et la moyenne des notes)

```
SELECT NomEnseignant, NomMatiere, Avg(Note)
FROM Enseignant AS E, Matiere AS M, Notes AS N
WHERE E.NumeroEnseignant=M.NumeroEnseignant
      AND M.NumeroMatiere=N.NumeroMatiere
GROUP BY NomEnseignant, NomMatiere, Matiere.NumeroMatiere
HAVING AVG(Note)< 10 ;
```

4. Quelle est, pour chaque matière, la meilleure note et quel est l'étudiant qui l'a obtenue ?

```
SELECT NomMatiere, NomEtudiant, Note
FROM Matiere AS M, Notes AS N, Etudiant E AS
WHERE N.NumeroMatiere=M.NumeroMatiere
AND E.NumeroEtudiant=N.NumeroEtudiant
AND Note IN (SELECT MAX(Note)
             FROM Notes AS E1
             WHERE E1.NumeroMatiere=M.NumeroMatiere) ;
```

5. Quelle est, pour chaque sexe (Homme, Femme), la moyenne des notes dans la matière enseignée par Jean Bonnot ?

```
SELECT Sexe, Sum(Note)/Count(Note) AS Moyenne
FROM Etudiant AS E, Notes AS N, Matiere AS M, Enseignant AS G
WHERE E.NumeroEtudiant=N.NumeroEtudiant
AND N.NumeroMatiere=M.NumeroMatiere
AND M.NumeroEnseignant=G.NumeroEnseignant
AND NomEnseignant='Jean Bonnot'
GROUP BY Sexe ;
```

6. Quels sont les noms des étudiants qui ont eu une note d'informatique supérieure à la moyenne générale de la classe ?

```
SELECT NomEtudiant
FROM Etudiant AS E, Notes AS N, Matiere AS M
WHERE E.NumeroEtudiant=N.NumeroEtudiant
AND N.NumeroMatiere=M.NumeroMatiere
AND NomMatiere='Informatique'
AND Note > (SELECT AVG(Note)
            FROM Notes AS N, Matiere AS M
            WHERE N.NumeroMatiere=M.NumeroMatiere
            AND NomMatiere='Informatique') ;
```