

Objectifs : - Créer une base de données.
- Relier les tables.
- Créer des requêtes SQL.

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

ETUDIANT (N°Etudiant, Nom, Prénom)

MATIERE (CodeMat, LibelléMat, CoeffMat)

EVALUER (N°Etudiant#, CodeMat#, Date_Evaluation, Note)

Remarques : Les clés primaires sont soulignées et les clés étrangères sont marquées par #

Questions :

Créer les requêtes SQL suivantes :

1. Quel est le nombre total d'étudiants ?
2. Quelle est, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?
3. Quelles sont les notes de chaque étudiant dans chacune des matières ?
4. Quelles sont les moyennes par matière ?

On utilisera la requête de la question 3 comme table source

5. Quelle est la moyenne générale de chaque étudiant ?
On utilisera la requête de la question 3 comme table source

6. Quelle est la moyenne générale de la promotion ?

On utilisera la requête de la question 5 comme table source

7. Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion ?

On utilisera la requête de la question 5 comme table source

Les requêtes SQL

Correction

1 - Quel est le nombre total d'étudiants ?

SELECT COUNT(*) FROM ETUDIANT ;

2 - Quelles sont, parmi l'ensemble des notes, la note la plus haute et la note la plus basse ?

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

3 - Quelles sont les moyennes de chaque étudiant dans chacune des matières?

```
SELECT ETUDIANT.N°Etudiant, Nom, Prénom, LibelléMat, CoeffMat,  
AVG(Note) AS MoyEtuMat  
FROM EVALUER, MATIERE, ETUDIANT  
WHERE EVALUER.CodeMat = MATIERE.CodeMat  
AND EVALUER.N°Etudiant = ETUDIANT.N°Etudiant  
GROUP BY ETUDIANT.N°Etudiant, Nom, Prénom, LibelléMat, CoeffMat;
```

4 - Quelles sont les moyennes par matière ?

```
SELECT LibelléMat, AVG(MoyEtuMat)  
FROM MOYETUMAT  
GROUP BY LibelléMat ;
```

5 - Quelle est la moyenne générale de chaque étudiant ?

```
SELECT N°Etudiant, Nom, Prénom,  
SUM(MoyEtuMat*CoeffMat)/SUM(CoeffMat) AS MgEtu  
FROM MOYETUMAT  
GROUP BY N°Etudiant, Nom, Prénom ;
```

6 - Quelle est la moyenne générale de la promotion ?

Avec la vue MGETU de la question 5 :

```
SELECT AVG(MgEtu)  
FROM MGETU ;
```

7 - Quels sont les étudiants qui ont une moyenne générale supérieure ou égale à la moyenne générale de la promotion ?

Avec la vue MGETU de la question 5 :

```
SELECT N°Etudiant, Nom, Prénom, MgEtu  
FROM MGETU  
WHERE MgEtu >= (SELECT AVG(MgEtu) FROM MGETU) ;
```

Exercice n°1 :

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

REPRESENTATION (n°représentation, titre_représentation, lieu)

MUSICIEN (nom, n°représentation*)

PROGRAMMER (date, n°représentation*, tarif)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par *

Questions :

- 1 - Donner la liste des titres des représentations.
- 2 - Donner la liste des titres des représentations ayant lieu à l'opéra Bastille.
- 3 - Donner la liste des noms des musiciens et des titres des représentations auxquelles ils participent.
- 4 - Donner la liste des titres des représentations, les lieux et les tarifs pour la journée du 14/09/96.

Exercice n°2 :

Soit le modèle relationnel suivant relatif à la gestion simplifiée des étapes du Tour de France 97, dont une des étapes de type "contre la montre individuel" se déroula à Saint-Etienne :

EQUIPE(CodeEquipe, NomEquipe, DirecteurSportif)
COUREUR(NuméroCoureur, NomCoureur, CodeEquipe*, CodePays*)
PAYS(CodePays, NomPays)
TYPE_ETAPE(CodeType, LibelléType)
ETAPE(NuméroEtape, DateEtape, VilleDép, VilleArr, NbKm, CodeType*)
PARTICIPER(NuméroCoureur*, NuméroEtape*, TempsRéalisé)
ATTRIBUER_BONIFICATION(NuméroEtape*, km, Rang, NbSecondes, NuméroCoureur*)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par *

Questions :

- 1 - Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?
- 2 - Quel est le nombre de kilomètres total du Tour de France 97 ?
- 3 - Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?
- 4 - Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?
- 5 - Quels sont les noms des coureurs qui ont participé à toutes les étapes ?
- 6 - 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 réalisés à chaque étape ?
- 7 - Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

Correction de l'exercice n°1

1 - Donner la liste des titres des représentations.

```
SELECT titre_représentation FROM REPRESENTATION ;
```

2 - Donner la liste des titres des représentations ayant lieu à l'opéra Bastille.

```
SELECT titre_représentation FROM REPRESENTATION
WHERE lieu="Opéra Bastille" ;
```

3 - Donner la liste des noms des musiciens et des titres des représentations auxquelles ils participent.

```
SELECT nom, titre_représentation
FROM MUSICIEN, REPRESENTATION
WHERE MUSICIEN.n°représentation = REPRESENTATION.n°représentation ;
```

4 - Donner la liste des titres des représentations, les lieux et les tarifs pour la journée du 14/09/96.

```
SELECT titre_représentation, lieu, tarif
FROM REPRESENTATION, PROGRAMMER
WHERE PROGRAMMER.n°représentation =
  REPRESENTATION.n°représentation
  AND date='14/09/96' ;
```

Correction de l'exercice n°2

1 - Quelle est la composition de l'équipe FESTINA (Numéro, nom et pays des coureurs) ?

```
SELECT NuméroCoureur, NomCoureur, NomPays
FROM EQUIPE A, COUREUR B, PAYS C
WHERE A.CodeEquipe=B.CodeEquipe And B.CodePays=C.CodePays
And NomEquipe="FESTINA" ;
```

2 - Quel est le nombre de kilomètres total du Tour de France 97 ?

```
SELECT SUM(Nbkm) FROM ETAPE ;
```

3 - Quel est le nombre de kilomètres total des étapes de type HAUTE MONTAGNE ?

```
SELECT SUM(Nbkm) FROM ETAPE A, TYPE_ETAPE B
WHERE A.CodeType=B.CodeType And LibelléType="HAUTE MONTAGNE" ;
```

4 - Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

```
SELECT NomCoureur FROM COUREUR
WHERE NuméroCoureur NOT IN (SELECT NuméroCoureur FROM
ATTRIBUER_BONIFICATION) ;
```

5 - Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

```
SELECT NomCoureur FROM PARTICIPER A, COUREUR B
WHERE A.NuméroCoureur=B.NuméroCoureur
GROUP BY NuméroCoureur, NomCoureur
HAVING COUNT(*)=(SELECT COUNT(*) FROM ETAPE) ;
```

6 - 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 réalisés à chaque étape ?

```
SELECT NomCoureur, CodeEquipe, CodePays, SUM(TempsRéalisé) AS Total
FROM PARTICIPER A, COUREUR B
WHERE A.NuméroCoureur=B.NuméroCoureur and NuméroEtape<=13
GROUP BY A.NuméroCoureur, NomCoureur, CodeEquipe, CodePays
ORDER BY Total;
```

7 - Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

```
SELECT NomEquipe, SUM(TempsRéalisé) AS Total
FROM PARTICIPER A, COUREUR B, EQUIPE C
WHERE A.NuméroCoureur=B.NuméroCoureur And
B.CodeEquipe=C.CodeEquipe
And NuméroEtape<=13
GROUP BY B.CodeEquipe, NomEquipe
ORDER BY Total;
```