

Exercice 1

Festival de musique

On dispose d'une base de données d'un festival de musique, où il y a plusieurs représentations. Un ou plusieurs musiciens peuvent participer à une représentation, mais un musicien ne peut participer qu'à une seule représentation.

La structure de la base de données est la suivante :

Représentation (idRep , titreRep , lieu)

Musicien (idMus , nom , # idRep)

Programmer (Date , # idRep, tarif)

Écrire la requête SQL permettant d'afficher :

- 1) La liste des titres des représentations.

```
SELECT titreRep
FROM Représentation ;
```

- 2) La liste des titres des représentations ayant lieu sur le lieu nommé « Dionysos ».

```
SELECT titreRep
FROM Représentation
WHERE lieu = "Dionysos" ;
```

- 3) La liste des noms des musiciens et les titres des représentations auxquelles ils participent.

```
SELECT Musicien.nom , Représentation.titreRep
FROM Musicien AS M
INNER JOIN Représentation AS R
ON R.idRep = M.idRep ;
```

- 4) La liste des titres des représentations, les lieux et les tarifs d'une date précisée.

(Par exemple le 24/12/2020)

```
SELECT Représentation.titreRep , Représentation.lieu , Programmer.tarif
FROM Programmer AS P
INNER JOIN Représentation AS R
ON P.idRep = R.idRep
WHERE P.date = "2020-12-24" ;
```

- 5) Le nombre de musiciens qui participent à la représentation numéro 7.

```
SELECT COUNT(*)
FROM Musicien
WHERE idRep = 7 ;
```

- 6) Les représentations et leurs dates dont le tarif ne dépasse pas 30 €.

```
SELECT Représentation.titreRep , Représentation.idRep , Programmer.Date
FROM Représentation AS R
INNER JOIN Programmer AS P
ON R.idRep = P.idRep
WHERE P.tarif <= 30 ;
```

Exercice 2

Les employés du département

On dispose d'une base de données dont la structure est la suivante :

Departements (DNO, DNOM, DIR, VILLE)

Employes (ENO, ENOM, PROF, DATEEMB, SAL, COMM, DNO)

Donner les requêtes SQL qui permettent d'obtenir :

- 1) La liste des employés ayant une commission (attribut « COMM » déclaré comme booléen).

```
SELECT *  
FROM Employes  
WHERE COMM = 1 ; (remarque : en booleen 1 : vrai -- 2 : faux)
```

- 2) Les noms, emplois et salaires des employés obtenus en classant les emplois dans l'ordre alphabétique, et pour chaque emploi, en classant les salaires du plus grands au plus petit.

```
SELECT ENOM, PROF, SAL  
FROM Employes  
ORDER BY PROF ASC , SAL DESC ;
```

- 3) Le salaire moyen des employés.

```
SELECT AVG (SAL)  
FROM Employes ;
```

- 4) Le salaire moyen dans le département nommé « Production » (DNOM = 'Production').

```
SELECT AVG (Employes.SAL)  
FROM Employes AS E  
INNER JOIN Departements AS D  
ON E.DNO = D.DNO  
WHERE D.DNOM = "Production" ;
```

Exercice 3

Notes annuelles des étudiants

On considère le modèle relationnel suivant concernant la gestion des notes annuelles d'une promotion d'étudiants :

ETUDIANT (NumEtudiant, Nom, Prenom)

MATIERE (CodeMat, LibelleMat, CoeffMat)

EVALUER (NumEval, #NumEtudiant, #CodeMat, Date, Note)

Les clés étrangères sont précédées d'un « » et les clés primaires sont soulignées.

Exprimez en SQL les requêtes suivantes.

- 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) AS minimum , MAX(Note) AS maximum  
FROM EVALUER ;
```

Exercice 4

Gestion d'un hôtel

Un hôtel gère ses chambres avec une bases de données.

La relation **Chambres** permet d'enregistrer : le numéro de la chambre, la date (le jour réservé), l'heure d'arrivée, le nombre de lits, la présence d'un balcon.

L'attribut **date** est au format chaîne de caractères et l'attribut **balcon** au format booléen.

Le tableau ci-dessous précise le contenu d'une ligne :

numéro	date	heure	lits	balcon
108	'18/05/20'	17	3	True

Ecrire les requêtes en SQL qui permettent d'obtenir ce qui est demandé dans chaque question-

- 1) Les numéros des chambres réservées à la date du 20/05/20.

```
SELECT numéro
FROM Chambres
WHERE date = '20/05/20' ;
```

- 2) Les numéros des chambres réservées à la date du 20/05/20 qui ont un balcon.

```
SELECT numéro
FROM Chambres
WHERE date = '20/05/20' AND balcon = True ;
```

- 3) Le nombre de chambres sans balcon réservées à la date du 21/05/20.

```
SELECT COUNT (numéro)
FROM Chambres
WHERE date = '20/05/20' AND balcon = False ;
```

- 4) Le nombre de chambres réservées le 18/05/20 avec deux ou trois lits.

```
SELECT COUNT (numéro)
FROM Chambres
WHERE date = '18/05/20' AND (lits = 2 OR lits = 3);
```

- 5) Les enregistrements des clients qui arrivent le 18/05/20 après 19 heures.

```
SELECT *
FROM Chambres
WHERE date = '18/05/20' AND heure > 19 ;
```

- 6) L'heure d'arrivée du premier client le 18/05/20.

```
SELECT MIN (heure)
FROM Chambres
WHERE date = '18/05/20' ;
```

Exercice 5

Gestion d'une librairie

Une librairie est gérée à l'aide d'une base de données. Le modèle relationnel contient les cinq relations décrites ci-dessous avec leur schéma :

Livres (Id, Titre, PrixHT, Année, # Id-genre, # Id-éditeur)

Auteurs (Id, Nom, Prénom)

Ecrits (#Id-auteur, # Id titre)

Genres (Id, Genre)

Editeurs (Id, Nom)

Les champs **Id** et le couple (**Id-auteur, Id-titre**) sont des **clés primaires**,

Le champ **Id-genre** est une clé étrangère en référence à la **clé primaire Id de la relation Genres**.

Le champ **Id-éditeur** est une clé étrangère en référence à la **clé primaire Id de la relation Editeur**.

Le champ **Id-auteur** est une clé étrangère en référence à la **clé primaire Id de la relation Auteurs**.

Le champ **Id-titre** est une clé étrangère en référence à la **clé primaire Id de la relation Livres**.

Tous les champs dont le nom est Id ou commence par Id et le champ Année sont de type entier.

Le champ PrixHT est de type flottant.

Les autres champs sont du type chaîne de caractères.

Ecrire en langage SQL les requêtes qui permettent d'obtenir ce qui est demandé dans chaque question.

- 1) Les titres et années de parution des livres.

```
SELECT Titre, Année  
FROM Livres ;
```

- 2) Les titres des livres qui commencent par la lettre " A".

```
SELECT Titre  
FROM Livres  
WHERE Titre LIKE 'A%';
```

- 3) Le prix HT maximal d'un livre de la base.

```
SELECT MAX (PrixHT)  
FROM Livres ;
```

- 4) Tous les champs de la table Auteurs.

```
SELECT *  
FROM Auteurs
```

- 5) Le nombre d'auteurs contenus dans la base.

```
SELECT COUNT (id)  
FROM Auteurs
```

- 6) Le nombre de livres référencés dans la base et le prix moyen.

On peut renommer les champs obtenus

```
SELECT COUNT (*) AS 'Nombre de livres' , AVG (PrixHT) AS 'Prix moyen'  
FROM Livres ;
```

7) Les titres des livres coûtant plus de 15 euros HT avec leur prix HT.

```
SELECT Titre, PrixHT
FROM Livres
WHERE PrixHT > 15 ;
```

8) Les titres des livres coûtant moins de 15 euros HT avec leur prix TTC. La TVA est 5,5%.

```
SELECT Titre, PrixHT *1.055
FROM Livres
WHERE PrixHT < 15 ;
```

9) Les titres et années de parution des livres parus de 2010 à 2015 ordonnés suivant l'année de parution de manière décroissante.

```
SELECT Titre, Année
FROM Livres
WHERE Année >= 2010 AND Année <= 2015
ORDER BY Année DESC ;
```

Ou bien :

```
SELECT Titre, Année
FROM Livres
WHERE Année BETWEEN 2010 AND 2015
ORDER BY Année DESC ;
```

10) Les titres des livres avec le nom de leur éditeur.

```
SELECT Titre, Nom
FROM Livres
INNER JOIN Editeurs
INNER JOIN Id-éditeur = Editeurs. Id ;
```

11) Les titres des livres du genre "science".

On renomme la table Genres en table g pour pouvoir abréger Genres. Id en g. Id.

```
SELECT Titre
FROM Livres
INNER JOIN Genres g
ON Id-genre = g. Id
WHERE Genre = 'science' ;
```

12) Les titres et les prix des livres du genre "policier" coûtant moins de 20 euros HT.

```
SELECT Titre , PrixHT
FROM Livres
INNER JOIN Genres g
ON Id-genre = g. Id
WHERE Genre = 'policier' AND PrixHT < 20 ;
```

13) Les années de parution de livres dans le genre science" sans doublons.

```
SELECT DISTINCT Année
FROM Livres
INNER JOIN Genres g
ON Id-genre = g. Id
WHERE Genre = 'science' ;
```

14) Le prix total de tous les livres parus en 2019 dans le genre "science".

```
SELECT SUM (PrixHT)  
FROM Livres  
INNER JOIN Genres g  
ON Id-genre = g. Id  
WHERE Genre = 'science' AND Année = 2019;
```

15) Les titres des livres du genre "science" parus en 2019 contenant la chaîne "nsi" dans le titre.

```
SELECT Titre  
FROM Livres  
INNER JOIN Genres  
ON Id-genre = Genres. Id  
WHERE Genre = 'science' AND Année = 2019 AND Titre LIKE '%nsi%' ;
```