

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**(id_rep, titre_rep, lieu)
- **Musicien**(id_mus, nom, #id_rep)
- **Programmer**(date, #id_rep, tarif)

✍ Écrire la requête SQL permettant d'afficher :

- La liste des titres des représentations.

```
SELECT id_rep  
FROM Représentation;
```

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

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

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

```
SELECT Musicien.nom, Représentation.titre_rep  
FROM Musicien AS M  
JOIN Représentation AS R  
ON M.id_rep = R.id_rep;
```

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

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

✍ Que permettent d'afficher les requêtes SQL suivantes :

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

La liste des titres des représentations, le lieu et le tarif pour la date du 24/12/2021.

```
SELECT Représentation.titre_rep, Programmer.date  
FROM Représentation AS R  
JOIN Programmer AS P  
ON R.id_rep = P.id_rep  
WHERE P.tarif <= 30;
```

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

2. Notes annuelles des étudiants

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

- ETUDIANT(num_etudiant, nom, prenom)
- MATIERE(code_mat, libelle_mat, coeff_mat)
- EVALUER(num_eval, #num_etudiant, #code_mat, date, note)

✎ Exprimez en langage SQL les requêtes suivantes :

- Quel est le nombre total d'étudiants ?

```
SELECT COUNT(*)
FROM ETUDIANT;
```
- 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. 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.

numéro	date	heure	lits	balcon
108	'18/05/21'	17	3	True

✎ Écrire les requêtes en langage SQL qui permettent d'obtenir ce qui est demandé :

- Les numéros des chambres réservées à la date du 20/05/21.

```
SELECT numéro
FROM Chambres
WHERE date = '20/05/21';
```
- Les numéros des chambres réservées à la date du 20/05/21 qui ont un balcon.

```
SELECT numéro
FROM Chambres
WHERE date = '20/05/21' AND balcon = True;
```
- Le nombre de chambres sans balcon réservées à la date du 20/05/21.

```
SELECT COUNT(numéro)
FROM Chambres
WHERE date = '20/05/21' AND balcon = False;
```
- Le nombre de chambres réservées le 18/05/21 avec deux ou trois lits.

```
SELECT COUNT (numéro)
FROM Chambres
WHERE date = '18/05/21' AND (lits = 2 OR lits = 3);
```
- Les enregistrements des clients qui arrivent le 18/05/21 après 19 heures.

```
SELECT *
FROM Chambres
WHERE date = '18/05/21' AND heure > 19;
```
- L'heure d'arrivée du premier client le 18/05/21.

```
SELECT MIN (heure)
FROM Chambres
WHERE date = '18/05/21';
```

4. Les employés du département

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

- **Departements**(dep_num, dep_nom, direct, ville)
- **Employes**(empl_num, empl_nom, prof, date_emb, sal, comm, #dep_num)

✍ Donner les requêtes SQL qui permettent d'obtenir :

- 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

- 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 empl_nom, prof, sal  
FROM Employes  
ORDER BY prof ASC, sal DESC;
```

- Le salaire moyen des employés.

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

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

```
SELECT AVG (Employes.sal)  
FROM Employes AS E  
JOIN Departements AS D  
ON E.dep_num = D.dep_num  
WHERE D.dep_nom = "Production";
```