

1. Alias

Problèmes :

- Le nom d'une table bien nommée est généralement long à taper.
- Comment faire la distinction quand on manipule des tables ayant certains noms de colonnes en commun ?
- Comment expliciter le fait qu'une colonne vient d'une table en particulier dans une requête manipulant plusieurs tables ?

Un alias est un nom qu'on donne à une table ou une colonne dans une requête SQL pour en faciliter l'écriture, en améliorant l'affichage ou en faciliter la compréhension. On utilise le mot clé optionnel "**AS**" pour déclarer des alias. Le mot clé AS ne veut pas dire *alias* mais *comme* ou *tel que* et on le retrouve dans d'autres expressions SQL ne concernant pas les alias.

```
1  -- Alias de table
2  SELECT e.nom FROM Eleve AS e WHERE e.nom LIKE 'F%';
3  -- ici "e" est l'alias de la table Eleve
4
5  -- Alias de colonne
6  SELECT prenom AS given_name FROM Eleve WHERE age = 18;
7
8  -- Alias de colonne avec une expression
9  SELECT (AVG(e.age)) AS moyenne_age FROM Eleve AS e WHERE e.age < 10 ;
10
11 -- AS est optionnel
12 SELECT e.nom FROM Eleve e WHERE e.nom LIKE 'F%'
13
```

2. Jointures

C'est l'opération emblématique du SQL :

- C'est une restriction appliquée à un produit cartésien.
- Elle permet d'associer plusieurs tables en reliant les données par leurs clés.
- On peut combiner plusieurs jointures dans une même requête.

Nous étudierons 2 principaux types de jointures : **INNER** et **LEFT**. Les autres types sont soit similaires car symétriques (ex: **RIGHT**) soit très peu utilisés.

2.1. INNER JOIN

```
1  SELECT * FROM Etudiant AS e
2  INNER JOIN Stage AS s ON s.etudiant = e.ine ;
3
```

La requête ci-dessus donne, pour chaque étudiant faisant un stage :

- Toutes les colonnes d'Etudiant
- Toutes les colonnes de Stage

2.2. LEFT JOIN

```
1 SELECT * FROM Etudiant AS e
2 LEFT JOIN Stage AS s ON s.etudiant = e.id ;
3
```

La requête ci-dessus donne, pour chaque étudiant :

- Toutes les colonnes d'Etudiant
- Toutes les colonnes de Stage pour ceux qui font un stage (sinon NULL)

Exercice 1 : Ecole de rugby

Pour cet exercice, récupérer le fichier *tp_rugby.db*.

Utiliser des alias et des jointures !!

- 1.1. Donner le prénom des éducateurs qui sont référents
- 1.2. Donner le nombre d'éducateurs qui ne sont pas référents

Correction

```
-- 1.1
SELECT DISTINCT prenom FROM Educateur AS ed
INNER JOIN Categorie AS c ON ed.numero_licence = c.referent;

-- 1.2
SELECT DISTINCT prenom FROM Educateur AS ed
LEFT JOIN Categorie AS c ON ed.numero_licence = c.referent
WHERE c.categorie IS NULL ;
```

Exercice 2 : Réseau Mobile

Le fichier *tp_reseauMobile.db* contient une base de données créée comme indiqué ci-dessous :

```
1 CREATE TABLE Departement (
2     code_insee TEXT NOT NULL,
3     nom TEXT NOT NULL,
4     PRIMARY KEY (code_insee)
5 );
6
7 CREATE TABLE Operateur (
8     id_operateur INT NOT NULL,
9     nom TEXT NOT NULL,
10    PRIMARY KEY (id_operateur)
11 );
12
13 CREATE TABLE Mesure (
14     id_point INT NOT NULL,
15     id_operateur INT NOT NULL,
16     qualite REAL NOT NULL CHECK (qualite >= 0), -- un exemple de contrainte
17     PRIMARY KEY (id_point, id_operateur),
18     FOREIGN KEY (id_point) REFERENCES Point (id_point),
19     FOREIGN KEY (id_operateur) REFERENCES Operateur (id_operateur)
20 );
21
22 CREATE TABLE Point (
23     id_point INT NOT NULL,
24     latitude REAL NOT NULL,
25     longitude REAL NOT NULL,
26     code_insee_dep TEXT NOT NULL,
27     PRIMARY KEY (id_point),
28     FOREIGN KEY (code_insee_dep) REFERENCES Departement (code_insee)
29 );
30
```

Question 1 : Donner le MLD sous forme graphique, correspondant à cette base de données

Question 2 : Donner la requête pour obtenir la moyenne de la qualité des mesures

```
SELECT AVG(moyenne) FROM Mesure AS qualite_moyenne ;
```

Question 3 : Donner la requête pour compter le nombre de mesures dont la qualité est supérieure à la moyenne

```
SELECT COUNT(*) FROM Mesure
WHERE Mesure.qualite > (SELECT AVG(moyenne) FROM Mesure AS qualite_moyenne) ;
```

Question 4 : Donner la requête pour compter les points à l'ouest du méridien de Greenwich

```
SELECT COUNT(*) FROM Point AS p
WHERE p.longitude > 0;
```

Question 5 : Donner la requête pour compter les points situés dans les départements pyrénéens(Ariège, Aude, Haute-Garonne, Hautes-Pyrénées, Pyrénées-Atlantiques et Pyrénées-Orientales).

```
SELECT COUNT(*) FROM Point AS p
INNER JOIN Departement AS d ON p.code_insee_dep = d.code_insee
WHERE d.nom IN ('Ariège', 'Aude', 'Haute-Garonne', 'Hautes-Pyrénées', 'Pyrénées-Atlantiques',
'Pyrénées-Orientales');
```

Question 6 : Donner la requête pour connaître le département le plus à l'ouest sur lesquels sont définis des points.

```
SELECT DISTINCT d.nom FROM Departement AS d
INNER JOIN Point AS p ON d.code_insee = p.code_insee_dep
WHERE p.longitude = (SELECT MIN(longitude) FROM Point) ;
```

Question 7 : Donner la liste des départements dans lesquels l'opérateur de votre choix est présent, trié par nom de département.

```
SELECT DISTINCT d.nom FROM Departement AS d
INNER JOIN Point AS p ON d.code_insee = p.code_insee_dep
INNER JOIN Mesure AS m on m.id_point = p.id_point
INNER JOIN Operateur AS op ON m.id_operateur = op.id_operateur
WHERE op.nom LIKE 'ORANGE'
ORDER BY d.nom
```

Question 8 : Combien y a-t'il de points sans mesures ?

```
SELECT COUNT(*) FROM Point AS p
LEFT JOIN Mesure AS m on p.id_point=m.id_point
WHERE m.qualite IS NULL ;
```

Question 9 : Dans quel département n'y a-t 'il aucun point de mesure ?

```
SELECT DISTINCT d.nom FROM Departement AS d
LEFT JOIN Point AS p ON d.code_insee = p.code_insee_dep
WHERE p.longitude IS NULL ;
```

Questions supplémentaires pour l'exercice 1 :

Question 1 : Donner la requête pour obtenir le nombre d'élèves qui ont entre 10 et 12 ans

```
SELECT COUNT(*) FROM Eleve  
WHERE Age >= 10 AND Age <= 12;
```

Question 2 : Donner la requête pour compter le nombre d'élèves qui se prénomment Pierre

```
SELECT COUNT(*) FROM Eleve  
WHERE Prenom LIKE 'Pierre';
```

Question 3 : Donner la requête pour lister les élèves dont le nom ou le prénom est Leroy

```
SELECT * FROM Eleve  
WHERE Prenom LIKE 'Leroy' OR Nom LIKE 'Leroy' ;
```

Question 4 : Donner la requête pour obtenir la moyenne d'âge des élèves dont le nom commence par une voyelle

```
SELECT AVG(age) AS moyenne_age  
FROM Eleve  
WHERE LOWER(SUBSTR(nom, 1, 1)) IN ('a', 'e', 'i', 'o', 'u', 'y');
```

Question 5 : Donner la requête pour obtenir le prénom des élèves triés par âge

```
SELECT prenom FROM Eleve  
ORDER BY Age
```

Question 6 : Donner la requête pour obtenir la liste des élèves qui ont un numéro de licence dont les derniers chiffres sont 42, triés par âge

```
SELECT * FROM Eleve  
WHERE numero_licence LIKE '%42'  
ORDER BY Age ;
```

Question 7 : Donner la requête pour lister les éducateurs qui n'ont pas de date de brevet

```
SELECT * FROM Educateur  
WHERE date_brevet IS NULL ;
```

Question 8 : Donner la requête pour lister les élèves dont le nom est également le prénom d'un autre élève

```
SELECT * FROM Eleve  
WHERE nom in (SELECT prenom FROM Eleve) ;
```