

TIn311 - Bases de données - IPSA Toulouse

Alexandre Condette – alexandre.condette@spacebel.fr

2021-2022

4 TP n° 2 : Jointures, Alias

4.1 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 améliorer l'affichage ou faciliter la compréhension. On utilise le mot clef optionnel **AS** pour déclarer des alias. Le mot clef **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.

-- *alias de table*

```
SELECT e.nom FROM Eleve AS e WHERE e.nom LIKE "F%";
```

-- *alias de colonne*

```
SELECT prenom AS given_name FROM Eleve WHERE age = 18;
```

-- *alias de colonne avec une expression*

```
SELECT AVG(e.age) AS moyenne_age FROM Eleve AS e WHERE e.age < 10;
```

-- *le AS est optionnel*

```
SELECT e.nom FROM Eleve e WHERE e.nom LIKE "F%";
```

4.2 Jointures

- L'opération emblématique de SQL
- (C'est une restriction appliquée à un produit cartésien)
- Permet d'associer plusieurs tables en reliant les données par leurs clefs (clef étrangère \Leftarrow clef primaire)
- On peut combiner plusieurs jointures dans une même requête

Nous étudierons 2 principaux types de jointure : **INNER** et **LEFT**. Les autres types de jointures sont soit similaires car symétriques (il y a par exemple **RIGHT**), soit très peu utilisés.

4.2.1 INNER JOIN

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

Donne pour tous les étudiants qui font un stage :

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

4.2.2 LEFT JOIN

```
SELECT * FROM Etudiant AS e
LEFT JOIN Stage AS s ON s.etudiant = e.ine ;
```

Donne pour tous les étudiants :

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

4.3 Exercices : École de rugby

4.3.1 Donnez le prénom des éducateurs qui sont référents.

Correction

```
SELECT ed.prenom
FROM Educateur AS ed
INNER JOIN Categorie AS cat ON ed.numero_licence = cat.referent;
```

4.3.2 Donnez le nombre d'éducateurs qui ne sont pas référents.

Correction

```
SELECT COUNT(*) AS nb_non_referents
FROM Educateur AS ed
LEFT JOIN Categorie AS cat ON ed.numero_licence = cat.referent
WHERE cat.referent IS NULL;
```

4.4 Exercices : Mon Réseau Mobile

Le fichier tp_arcep.zip contient une base de données créée comme ci-dessous. Le schéma est légèrement différent du cas que nous avons étudié. Les données datent de 2019.

```
CREATE TABLE Departement (
    code_insee TEXT NOT NULL,
    nom TEXT NOT NULL,
    PRIMARY KEY (code_insee)
);

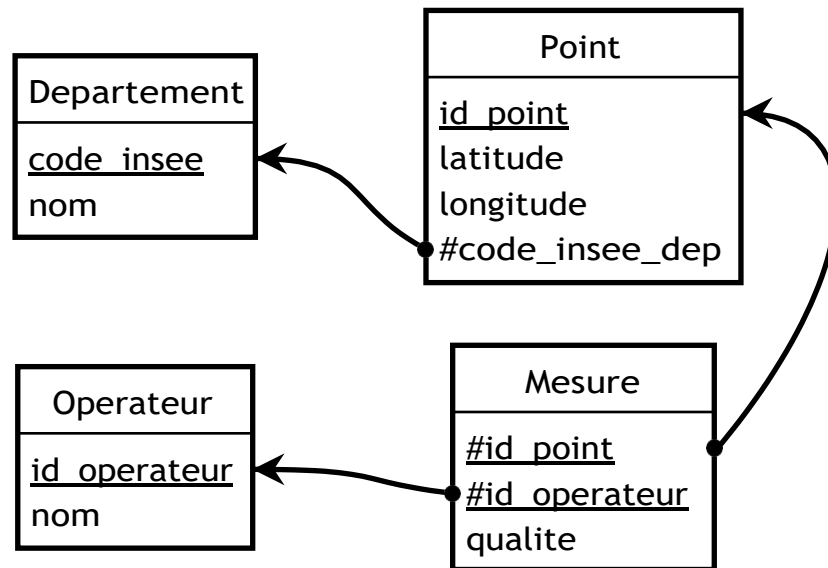
CREATE TABLE Operateur (
    id_operateur INT NOT NULL,
    nom TEXT NOT NULL,
    PRIMARY KEY (id_operateur)
);

CREATE TABLE Mesure (
    id_point INT NOT NULL,
    id_operateur INT NOT NULL,
    qualite REAL NOT NULL CHECK (qualite >= 0), -- un exemple de contrainte
    PRIMARY KEY (id_point, id_operateur),
    FOREIGN KEY (id_point) REFERENCES Point (id_point),
    FOREIGN KEY (id_operateur) REFERENCES Operateur (id_operateur)
);

CREATE TABLE Point (
    id_point INT NOT NULL,
    latitude REAL NOT NULL,
    longitude REAL NOT NULL,
    code_insee_dep TEXT NOT NULL,
    PRIMARY KEY (id_point),
```

```
FOREIGN KEY (code_insee_dep) REFERENCES Departement (code_insee)
);
```

4.4.1 Donnez le MLD sous forme graphique correspondant à cette base de données.



Correction

4.4.2 Donnez la requête pour obtenir la moyenne de la qualité des mesures.

Correction

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

4.4.3 Donnez la requête pour compter le nombre de mesures dont la qualité est supérieure à la moyenne.

Correction

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

4.4.4 Donnez la requête pour compter les points à l'ouest du méridien de Greenwich.

Correction

```
SELECT COUNT(*) AS nb_points_ouest FROM Point WHERE longitude < 0;
```

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

Correction

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

4.4.6 Donnez la requête pour connaître le département la plus à l'ouest sur lequel sont définis des points.

Correction

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

4.4.7 Donnez la liste des départements dans lesquels l'opérateur de votre choix est présent (i.e. il existe des mesures pour cet opérateur), trié par nom de département.

Correction

Exemple avec Orange :

```
SELECT DISTINCT d.nom AS departements_ou_y_a_orange
FROM Departement AS d
INNER JOIN Point AS p ON d.code_insee = p.code_insee_dep
INNER JOIN Mesure AS m ON p.id_point = m.id_point
INNER JOIN Operateur AS op ON m.id_operateur = op.id_operateur
WHERE op.nom = "Orange"
ORDER BY d.nom;
```

4.4.8 Combien y a-t-il de points sans mesures ?

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

4.4.9 Dans quel département n'y a-t-il aucun point de mesure ?

Correction

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