

# TP\_Junier

May 20, 2020

```
In [2]: %reload_ext sql
        %config SqlMagic.displaycon = False
        %config SqlMagic.autolimit = 100
```

```
In [3]: %sql sqlite:///base-etudiant.db
```

## 1 Exercice 1:

Donner tous les noms des étudiants.

```
In [18]: %%sql

        SELECT NomEt FROM Etudiant;
```

Done.

```
Out[18]: [('Armand A.',),
          ('Berthe B.',),
          ('Cendrine C.',),
          ('David D.',),
          ('Erwan E.',),
          ('Fabien F.',),
          ('Gerald G.',),
          ('Herbert H.',),
          ('Jacques J.',)]
```

```
In [17]: %%sql

        SELECT *
        FROM sqlite_master
```

Done.

```
Out[17]: [('table', 'Etudiant', 'Etudiant', 2, 'CREATE TABLE Etudiant (\n          NumEt INTEGER I
          ('table', 'Enseignant', 'Enseignant', 3, 'CREATE TABLE Enseignant (\n          NumEns I
          ('table', 'UE', 'UE', 4, 'CREATE TABLE UE (\n          NumUE INTEGER PRIMARY KEY,\n          ('table', 'Enseigne', 'Enseigne', 5, 'CREATE TABLE Enseigne (\n          NumEns INTEGER
```



On considère les tables suivantes :

- Etudiant (NumEt, NomEt, Adresse): le numéro, le nom et l'adresse des étudiants;
- Enseignant (NumEns, NomEns): le nom et le prénom des enseignants;
- UE (NumUE, Titre, HCours, HTD, HTP): le numéro et le titre de l'UE, ainsi que le nombre d'heures de cours magistraux, de TD et de TP par groupe d'étudiants;
- Enseigne (NumEns, NumUE, NCours, NTD, NTP): indique dans quelle UE intervient quel enseignant en précisant le nombre de cours magistraux, de groupes de TD et de groupes de TP pour cet enseignant dans cette UE
- Inscrit (NumEt, NumUE): indique quel étudiant est inscrit dans quelle UE

Charger la base dans sqlite3, sqlitebrowser ou DBeaver et traduire les requêtes suivantes en SQL.

## Base étudiant

Done.

```
Out[15]: [('CREATE TABLE Enseigne (\n          NumEns INTEGER REFERENCES Enseignant(NumEns),\n
```

```
In [16]: %%sql
```

```
SELECT sql
FROM sqlite_master
WHERE name = 'Inscrit';
```

Done.

```
Out[16]: [('CREATE TABLE Inscrit (\n          NumEt INTEGER REFERENCES Etudiant(NumEt),\n
```

### 1.1 Exercice 2

Donner les titre des UEs dans l'ordre alphabétique

```
In [19]: %%sql
```

```
SELECT Titre
FROM UE
ORDER BY Titre ASC;
```

Done.

```
Out[19]: [('Algebre',),
          ('Algorithmique',),
          ('Analyse',),
          ('Bases de donnees',),
          ('Programmation',),
          ('Reseaux',)]
```

## 1.2 Exercice 3:

Donner le titre des UEs dont le nombre d'heures total (cours, td et cm) par groupe est au moins 46

```
In [20]: %%sql
```

```
SELECT Titre
FROM UE
WHERE HCours + HTD + HTP >= 46;
```

Done.

```
Out[20]: [('Algorithmique',), ('Bases de donnees',)]
```

## 1.3 Exercice 4:

Donner les noms des étudiants qui ont 'Albert A.' comme enseignant.

```
In [25]: %%sql
```

```
SELECT DISTINCT NomEt
FROM
Etudiant JOIN Inscrit
ON Etudiant.NumET = Inscrit.NumEt
JOIN UE
ON Inscrit.NumUE = UE.NumUE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
WHERE NomEns = "Albert A."
;
```

Done.

```
Out[25]: [('David D.',), ('Gerald G.',), ('Jacques J.',)]
```

## 1.4 Exercice 5:

Donner les titres des cours ayant au moins un étudiant inscrit et dont le nombre d'heures de TD est au moins 18.

```
In [34]: %%sql
```

```
SELECT Titre, HTD
FROM
Etudiant JOIN Inscrit
ON Etudiant.NumET = Inscrit.NumEt
```

```

JOIN UE
ON Inscrit.NumUE = UE.NumUE
WHERE HTD >= 18
GROUP BY Titre
;

```

Done.

Out [34]: [('Algebre', 25.0), ('Analyse', 25.0), ('Bases de donnees', 18.0)]

## 1.5 Exercice 6:

Donner les noms des enseignants qui enseignent dans la même UE que 'Albert A.' (sauf Albert A. lui-même).

In [39]: %%sql

```

SELECT
    DISTINCT NomEns
FROM
    UE JOIN Enseigne
    ON UE.NumUE = Enseigne.NumUE
    JOIN Enseignant
    ON Enseigne.NumEns = Enseignant.NumEns
WHERE NomEns != "Albert A."
AND Titre IN (
    SELECT
        Titre
    FROM
        UE JOIN Enseigne
        ON UE.NumUE = Enseigne.NumUE
        JOIN Enseignant
        ON Enseigne.NumEns = Enseignant.NumEns
        WHERE NomEns = "Albert A."
    );

```

Done.

Out [39]: [('Bertrand B.',)]

In [36]: %%sql

```

SELECT
    NomEns, Titre
FROM
    UE
    JOIN Enseigne

```

```

ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
;

```

Done.

```

Out [36]: [('Albert A.', 'Analyse'),
          ('Albert A.', 'Algebre'),
          ('Bertrand B.', 'Analyse'),
          ('Bertrand B.', 'Algebre'),
          ('Carine C.', 'Programmation'),
          ('Carine C.', 'Algorithmique'),
          ('Carine C.', 'Bases de donnees'),
          ('David D.', 'Programmation'),
          ('David D.', 'Algorithmique'),
          ('David D.', 'Bases de donnees'),
          ('Edgar E.', 'Programmation'),
          ('Edgar E.', 'Algorithmique'),
          ('Edgar E.', 'Bases de donnees')]

```

## 1.6 Exercice 7:

Donner le nombre total d'heures de cours/TD/TP dispensées à l'université. On nommera TOTAL\_HEURES ce nombre.

```

In [4]: %%sql

```

```

SELECT
SUM(HCours * NCours + HTD * NTD + HTP * NTP) AS TOTAL_HEURES
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
;

```

Done.

```

Out [4]: [(436.0,)]

```

## 1.7 Exercice 8:

Donner le nombre d'UE n'ayant pas de TP (on appellera NB\_UES l'attribut donnant ce résultat).

```

In [20]: %%sql

```

```

SELECT

```

```

COUNT(*)
FROM
UE
WHERE HTP = 0
;

```

Done.

Out[20]: [(2,)]

In [19]: %%sql

```

SELECT
COUNT(*)
FROM
(SELECT
UE.NumUE
FROM
UE JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
GROUP BY UE.NumUE
HAVING SUM(HTP * NTP) = 0
)
;

```

Done.

Out[19]: [(2,)]

Plutôt la requête ci-dessous

In [12]: %%sql

```

SELECT
COUNT(*)
FROM(
SELECT Titre
FROM UE

EXCEPT

SELECT Titre
FROM UE JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
GROUP BY UE.NumUE
HAVING SUM(HTP * NTP) > 0
);

```

Done.

Out [12]: [(3,)]

### 1.8 Exercice 9:

Donner le nombre d'étudiants qui suivent le cours d'Analyse (on appellera NB\_ETUDIANTS l'attribut donnant ce nombre).

In [63]: %%sql

```
SELECT COUNT(*) AS NB_ETUDIANTS
FROM
Etudiant JOIN Inscrit
ON Etudiant.NumET = Inscrit.NumEt
JOIN UE
ON Inscrit.NumUE = UE.NumUE
WHERE Titre = "Analyse"
;
```

Done.

Out [63]: [(3,)]

### 1.9 Exercice 10

Donner la moyenne du nombre d'heures de cours, de TD et de TP par UE. On appellera MOY\_COURS la moyenne des heures, MOY\_TD celle des TD et MOY\_TP celle des TP.

In [48]: %%sql

```
SELECT
AVG(HCours) AS MOY_COURS, AVG(HTD) AS MOY_TD, AVG(HTP) AS MOY_TP
FROM UE ;
```

Done.

Out [48]: [(16.5, 16.333333333333332, 8.333333333333334)]

### 1.10 Exercice 11:

Donner pour chaque étudiant le nombre total d'heures qu'il suit. On donnera dans le résultat le numéro de l'étudiant ainsi qu'un attribut HEURES qui indiquera son nombre d'heures.

In [53]: %%sql

```
SELECT NumEt, SUM(HTP + HTD + HCours) AS HEURES
```



```

FROM (
    SELECT DISTINCT UE.NumUE, Etudiant.NumEt, HTP, HTD, HCours --DISTINCT UE
    FROM
        Etudiant JOIN Inscrit
        ON Etudiant.NumET = Inscrit.NumEt
        JOIN UE
        ON Inscrit.NumUE = UE.NumUE
        JOIN Enseigne
        ON UE.NumUE = Enseigne.NumUE
    WHERE NCours+ NTD + NTP != 0
)
GROUP BY NumET
ORDER BY HEURES DESC
;

```

Done.

```

Out [53]: [(1117, 185.0),
           (1111, 149.0),
           (1114, 135.0),
           (1119, 135.0),
           (1112, 104.0),
           (1116, 54.0),
           (1118, 54.0),
           (1115, 50.0)]

```

### 1.11 Exercice 12:

Donner les numéros des enseignants qui effectuent plus de 17 heures de cours magistraux. Attention, ici une clause HAVING est nécessaire.

```
In [49]: %%sql
```

```

SELECT
    Enseignant.NumEns, NomEns, SUM(HCours) AS HEURES_COURS
FROM
    UE
    JOIN Enseigne
    ON UE.NumUE = Enseigne.NumUE
    JOIN Enseignant
    ON Enseigne.NumEns = Enseignant.NumEns
GROUP BY Enseignant.NumEns
HAVING SUM(HCours * NCours) > 17
;

```

Done.

```
Out[49]: [(111, 'Albert A.', 40.0),
          (112, 'Bertrand B.', 40.0),
          (114, 'David D.', 53.0),
          (115, 'Edgar E.', 53.0)]
```

```
In [48]: %%sql
```

```
SELECT
*
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
;
```

Done.

```
Out[48]: [(1, 'Analyse', 20.0, 25.0, 0.0, 111, 1, 1, 1, 0, 111, 'Albert A.'),
          (2, 'Algebre', 20.0, 25.0, 0.0, 111, 2, 0, 1, 0, 111, 'Albert A.'),
          (1, 'Analyse', 20.0, 25.0, 0.0, 112, 1, 0, 1, 0, 112, 'Bertrand B.'),
          (2, 'Algebre', 20.0, 25.0, 0.0, 112, 2, 1, 1, 0, 112, 'Bertrand B.'),
          (3, 'Programmation', 15.0, 15.0, 15.0, 113, 3, 1, 1, 1, 113, 'Carine C.'),
          (4, 'Algorithmique', 20.0, 15.0, 15.0, 114, 4, 1, 1, 1, 114, 'David D.'),
          (5, 'Bases de donnees', 18.0, 18.0, 18.0, 115, 5, 1, 1, 1, 115, 'Edgar E.'),
          (4, 'Algorithmique', 20.0, 15.0, 15.0, 113, 4, 0, 0, 1, 113, 'Carine C.'),
          (5, 'Bases de donnees', 18.0, 18.0, 18.0, 113, 5, 0, 1, 1, 113, 'Carine C.'),
          (3, 'Programmation', 15.0, 15.0, 15.0, 114, 3, 0, 0, 1, 114, 'David D.'),
          (5, 'Bases de donnees', 18.0, 18.0, 18.0, 114, 5, 0, 1, 1, 114, 'David D.'),
          (3, 'Programmation', 15.0, 15.0, 15.0, 115, 3, 0, 0, 1, 115, 'Edgar E.'),
          (4, 'Algorithmique', 20.0, 15.0, 15.0, 115, 4, 0, 1, 1, 115, 'Edgar E.')]

```