

TP_Junier

May 24, 2020

```
In [21]: %reload_ext sql
         %config SqlMagic.displaycon = False
         %config SqlMagic.autolimit = 100
```

```
In [22]: %sql sqlite:///base-etudiant.db
```

1 Exercice 1:

Donner tous les noms des étudiants.

```
In [23]: %%sql

SELECT NomEt FROM Etudiant;
```

Done.

```
Out[23]: [('Armand A.',),
          ('Berthe B.',),
          ('Cendrine C.',),
          ('David D.',),
          ('Erwan E.',),
          ('Fabien F.',),
          ('Gerald G.',),
          ('Herbert H.',),
          ('Jacques J.',)]
```

```
In [24]: %%sql

SELECT *
FROM sqlite_master
```

Done.

```
Out[24]: [('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
```

```

('index', 'sqlite_autoindex_Enseigne_1', 'Enseigne', 6, None),
('table', 'Inscrit', 'Inscrit', 7, 'CREATE TABLE Inscrit (\n          NumEt INTEGER REF
('index', 'sqlite_autoindex_Inscrit_1', 'Inscrit', 8, None)]

```

In [25]: %%sql

```

SELECT name
FROM sqlite_master
WHERE type = 'table'

```

Done.

Out[25]: [('Etudiant',), ('Enseignant',), ('UE',), ('Enseigne',), ('Inscrit',)]

In [26]: %%sql

```

SELECT sql
FROM sqlite_master
WHERE name = 'Etudiant';

```

Done.

Out[26]: [('CREATE TABLE Etudiant (\n NumEt INTEGER PRIMARY KEY,\n NomEt VARCHAR2(255),\n NumEnseigne INTEGER REFERENCES Enseigne(\n NumEnseigne\n) ON DELETE CASCADE\n)']

In [27]: %%sql

```

SELECT sql
FROM sqlite_master
WHERE name = 'Enseignant';

```

Done.

Out[27]: [('CREATE TABLE Enseignant (\n NumEns INTEGER PRIMARY KEY,\n NomEns VARCHAR2(255),\n NumEnseigne INTEGER REFERENCES Enseigne(\n NumEnseigne\n) ON DELETE CASCADE\n)']

In [28]: %%sql

```

SELECT sql
FROM sqlite_master
WHERE name = 'UE';

```

Done.

Out[28]: [('CREATE TABLE UE (\n NumUE INTEGER PRIMARY KEY,\n Titre VARCHAR2(255),\n NumEnseigne INTEGER REFERENCES Enseigne(\n NumEnseigne\n) ON DELETE CASCADE\n)']

In [29]: %%sql

```

SELECT sql
FROM sqlite_master
WHERE name = 'Enseigne';

```

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

```
In [30]: %sql
```

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

Done.

```
Out [30]: [('CREATE TABLE Inscrit (\n          NumEt INTEGER REFERENCES Etudiant(NumEt),\n
```

1.1 Exercice 2

Donner les titre des UEs dans l'ordre alphabétique

```
In [31]: %sql
```

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

Done.

```
Out [31]: [('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 [32]: %%sql
```

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

Done.

```
Out[32]: [('Algorithmique',), ('Bases de donnees',)]
```

1.3 Exercice 4:

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

```
In [33]: %%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[33]: [('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)]

Plutôt:

In [35]: %%sql

```

SELECT DISTINCT Titre, HTD
FROM Inscrit JOIN UE
ON Inscrit.NumUE = UE.NumUE
WHERE HTD >= 18
;

```

Done.

Out[35]: [('Bases de donnees', 18.0), ('Analyse', 25.0), ('Algebre', 25.0)]

1.5 Exercice 6:

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

In [36]: %%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[36]: [('Bertrand B.',)]

Plutot:

In [37]: %%sql

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

Done.

Out[37]: [('Bertrand B.',)]

In [38]: %%sql

```
SELECT DISTINCT Ens2.NomEns
FROM Enseigne AS E1
JOIN Enseigne AS E2
ON E2.NumUE = E1.NumUE
JOIN Enseignant AS Ens1
ON E1.NumEns = Ens1.NumEns
JOIN Enseignant AS Ens2
ON E2.NumEns = Ens2.NumEns
WHERE Ens1.NomEns = "Albert A." AND E1.NumEns <> E2.NumEns ;
```

Done.

Out[38]: [('Bertrand B.',)]

In [39]: %%sql

```
SELECT
```

```

NomEns, Titre
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
;

```

Done.

```

Out[39]: [('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 [40]: %%sql
```

```

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

```

Done.

```
Out[40]: [(436.0,)]
```

```
In [41]: %%sql
```

```

SELECT

```

```

UE.NumUE, SUM(HCours * NCours + HTD * NTD + HTP * NTP) AS TOTAL_HEURES
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
GROUP BY UE.NumUE
UNION
SELECT NumUE, 0
FROM UE
EXCEPT
SELECT DISTINCT NumUE, 0
FROM Enseigne
;

```

Done.

Out[41]: [(1, 70.0), (2, 70.0), (3, 75.0), (4, 95.0), (5, 126.0), (6, 0)]

Avec LEFT OUTER JOIN (https://www.w3schools.com/sql/sql_join_left.asp) + COALESCE (https://www.w3schools.com/sql/sql_isnull.asp) pour transformer le NULL en 0. Noter que RIGHT OUTER JOIN et FULL OUTER JOIN ne sont pas implémentés par Sqlite : <https://www.sqlite.org/omitted.html>.

Commentaires de Romuald Thion :

Non. Quand on calcule une jointure A JOIN B (NATURAL ou pas) mais qu'on veut aussi les tuples de A qui n'ont pas de liens avec ceux de B on doit prendre une LEFT [OUTER] JOIN. Dans le résultat, ces tuples supplémentaires auront des NULL pour tous les attributs de B 10:06 PM

Si on veut ceux de B sans images dans A, c'est RIGHT [OUTER] JOIN, et si on les veut tous (ceux de A et ceux de B, mais c'est plus rare), c'est FULL [OUTER] JOIN. Mais SQLite ne supporte aucune de ces deux variantes (<https://www.sqlite.org/omitted.html>)

Je préfère r JOIN s USING (atts) que NATURAL JOIN car ça explicite les attributs utilisés et surtout, c'est + robuste au renommage des colonnes : si on renomme Inscrit.NumEt par exemple ça va changer le sens de SELECT * FROM Etudiant NATURAL JOIN Inscrit (qui devient un produit cartésien !) alors que SELECT * FROM Etudiant JOIN Inscrit USING (numEt); produire une erreur (car NumEt n'existe plus) 2:57 PM

Les deux, JOIN USING et NATURAL JOIN, font que ya une colonne NumEt unique dans le résultat de la requête, à la différence de INNER JOIN et ses variantes 2:58 PM

A part les effets de l'implicit du NATURAL... question de goût. Ya un argument en faveur toutefois : c'est que ça colle mieux avec la théorie du modèle relationnel et l'opérateur de jointure (\bowtie en LaTeX)

In [42]: %%sql

```
SELECT
UE.NumUE, COALESCE(SUM(HCours * NCours + HTD * NTD + HTP * NTP), 0) AS TOTAL_HEURES
FROM
UE
LEFT OUTER JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
GROUP BY UE.NumUE
;
```

Done.

Out[42]: [(1, 70.0), (2, 70.0), (3, 75.0), (4, 95.0), (5, 126.0), (6, 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 [43]: %%sql

```
SELECT
COUNT(*)
FROM
UE
WHERE HTP = 0
;
```

Done.

Out[43]: [(2,)]

In [44]: %%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[44]: [(2,)]

Plutôt la requête ci-dessous

In [45]: %%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[45]: [(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 [46]: %%sql

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

Done.

Out[46]: [(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 [47]: %%sql

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

Done.

Out[47]: [(16.5, 16.333333333333332, 8.333333333333334)]

In [48]: %%sql

```
SELECT
  SUM(HCours)/(SELECT COUNT(*) FROM UE WHERE HCours <> 0) AS MOY_COURS,
  SUM(HTD)/(SELECT COUNT(*) FROM UE WHERE HTD <> 0) AS MOY_TD,
  SUM(HTP)/(SELECT COUNT(*) FROM UE WHERE HTP <> 0) AS MOY_TP
FROM
  UE ;
```

Done.

Out[48]: [(16.5, 19.6, 12.5)]

In [49]: %%sql

```
SELECT *
FROM UE;
```

Done.

Out[49]: [(1, 'Analyse', 20.0, 25.0, 0.0),
(2, 'Algebre', 20.0, 25.0, 0.0),
(3, 'Programmation', 15.0, 15.0, 15.0),
(4, 'Algorithmique', 20.0, 15.0, 15.0),
(5, 'Bases de donnees', 18.0, 18.0, 18.0),
(6, 'Reseaux', 6.0, 0.0, 2.0)]

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 [50]: %%sql

```
SELECT Inscrit.NumEt, SUM(HTP + HTD + HCours) AS HEURES
FROM Inscrit JOIN UE
ON Inscrit.NumUE = UE.NumUE
GROUP BY Inscrit.NumEt
ORDER BY HEURES DESC
;
```

Done.

Out [50]: [(1117, 185.0),
(1111, 149.0),
(1114, 143.0),
(1119, 135.0),
(1112, 104.0),
(1116, 62.0),
(1118, 62.0),
(1115, 50.0)]

Plutôt la requête suivante

In [51]: %%sql

```
SELECT NumEt, SUM(HTP + HTD + HCours) AS HEURES
FROM (
    SELECT UE.NumUE, Etudiant.NumEt, HTP, HTD, HCours
    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 Etudiant.NumEt, UE.NumUE
)
GROUP BY NumEt
ORDER BY HEURES DESC
;
```

Done.

Out [51]: [(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 [52]: %%sql
```

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

Done.

```
Out [52]: [(111, 40.0), (112, 40.0), (114, 53.0), (115, 53.0)]
```

```
In [53]: %%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 [53]: [(111, 'Albert A.', 40.0),  
(112, 'Bertrand B.', 40.0),  
(114, 'David D.', 53.0),  
(115, 'Edgar E.', 53.0)]
```

In [54]: %%sql

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

Done.

Out [54]: [(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.')]

1.12 Sandbox

In [55]: %%sql

```
SELECT
Enseignant.NumEns, UE.NumUE, SUM(HTP * NTP + HTD * NTD + HCours * NCours)
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
GROUP BY Enseignant.NumEns, UE.NumUE
;
```

Done.

Out [55]: [(111, 1, 45.0),
(111, 2, 25.0),

```
(112, 1, 25.0),
(112, 2, 45.0),
(113, 3, 45.0),
(113, 4, 15.0),
(113, 5, 36.0),
(114, 3, 15.0),
(114, 4, 50.0),
(114, 5, 36.0),
(115, 3, 15.0),
(115, 4, 30.0),
(115, 5, 54.0)]
```

In [56]: %%sql

```
SELECT
Enseignant.NumEns, UE.NumUE, HCours * NCours
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
GROUP BY Enseignant.NumEns, UE.NumUE
;
```

Done.

Out [56]: [(111, 1, 20.0),
(111, 2, 0.0),
(112, 1, 0.0),
(112, 2, 20.0),
(113, 3, 15.0),
(113, 4, 0.0),
(113, 5, 0.0),
(114, 3, 0.0),
(114, 4, 20.0),
(114, 5, 0.0),
(115, 3, 0.0),
(115, 4, 0.0),
(115, 5, 18.0)]

In [57]: %%sql

```
SELECT
DISTINCT Enseignant.NumEns
FROM
UE
JOIN Enseigne
```

```

ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
WHERE NTD * HTD > 0
;

```

Done.

Out [57]: [(111,), (112,), (113,), (114,), (115,)]

In [58]: %%sql

```

SELECT
Enseignant.NumEns, SUM(NTD * HTD)
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
WHERE NTD * HTD > 0
GROUP BY Enseignant.NumEns
;

```

Done.

Out [58]: [(111, 50.0), (112, 50.0), (113, 33.0), (114, 33.0), (115, 33.0)]

1.13 Sandbox

In [59]: %%sql

```

SELECT
DISTINCT Enseignant.NumEns
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
WHERE NTD * HTD > 0
;

```

Done.

Out [59]: [(111,), (112,), (113,), (114,), (115,)]

In [60]: %%sql

```
SELECT
Enseignant.NumEns, SUM(NTD * HTD)
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
WHERE NTD * HTD > 0
GROUP BY Enseignant.NumEns
;
```

Done.

Out[60]: [(111, 50.0), (112, 50.0), (113, 33.0), (114, 33.0), (115, 33.0)]

In [61]: %%sql

```
SELECT
Enseignant.NumEns, SUM(NTD * HTD) AS HeuresTD
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
GROUP BY Enseignant.NumEns
HAVING HeuresTD > 0
;
```

Done.

Out[61]: [(111, 50.0), (112, 50.0), (113, 33.0), (114, 33.0), (115, 33.0)]

In [62]: %%sql

```
SELECT
Enseignant.NumEns, SUM(NTD * HTD) AS HeuresTD
FROM
UE
JOIN Enseigne
ON UE.NumUE = Enseigne.NumUE
JOIN Enseignant
ON Enseigne.NumEns = Enseignant.NumEns
GROUP BY Enseignant.NumEns
```

```
HAVING HeuresTD > 0  
;
```

Done.

```
Out[62]: [(111, 50.0), (112, 50.0), (113, 33.0), (114, 33.0), (115, 33.0)]
```