Rapport SAE 2.04

Meley Félix

Monnier César

411111111111111111

Sommaire :

I.MCD

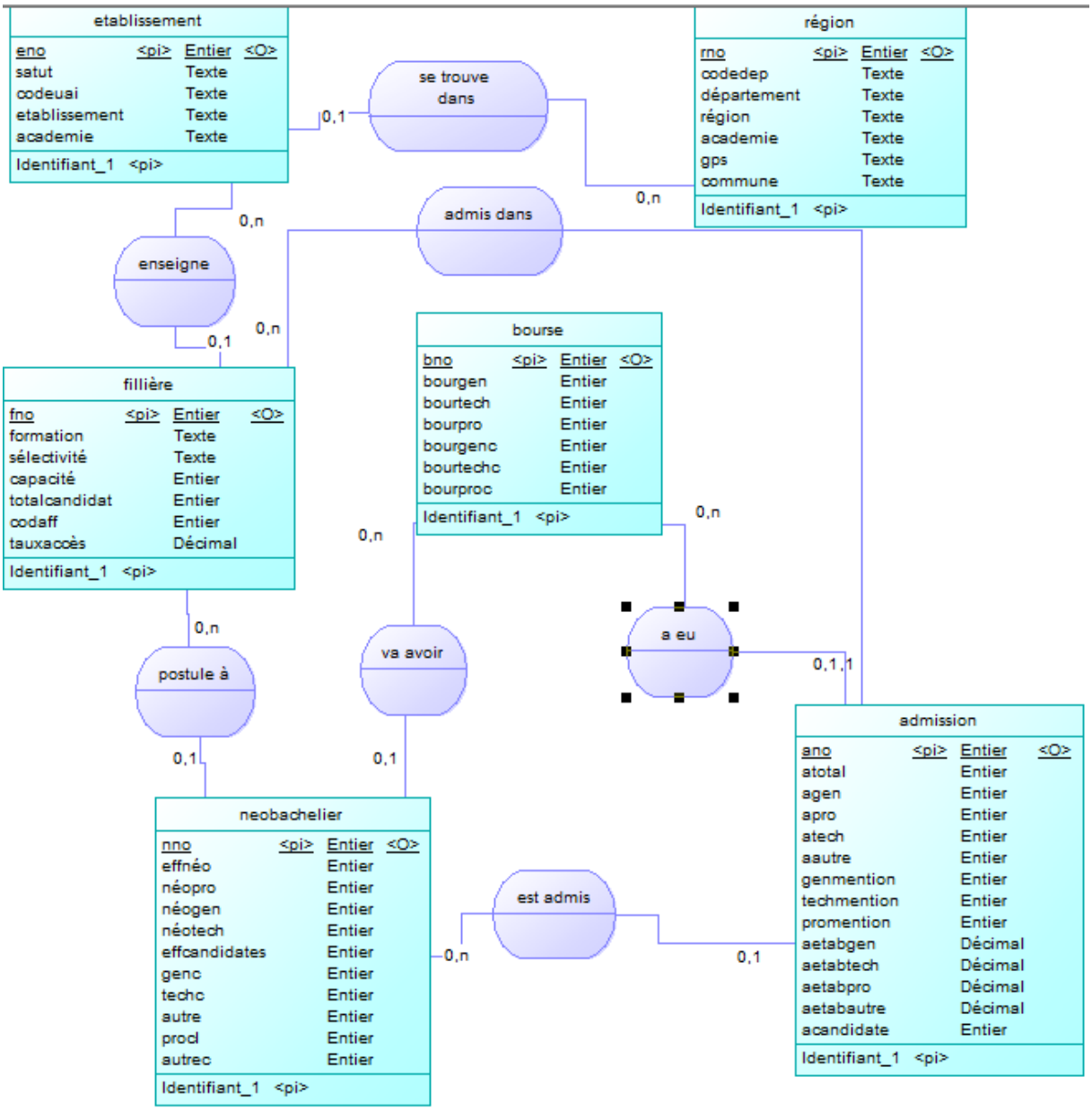

Figure 1MCD

Notre MCD est composé de 6 tables une table regroupent toutes les informations sur les établissements, une table contenant toutes les informations sur la localisation géographique des établissements. Pour les étudiants nous avons découpé leurs informations en trois table une table contenant toutes les informations sur les néo bacheliers, une sur leur admission et enfin une sur les bourses, cette découpe nous permet de plus facilement accéder aux différentes informations concernant les étudiants. Enfin nous avons une table contenant toutes les informations importantes sur les filières et les formations.

II. Exercice 1

Q1.

1. Il y a 13870 lignes. Pour trouver le nombre de ligne nous avons utilisé la commande « wc -| données.csv » ce qui nous à permis de savoir son nombre de ligne.
2. La 1ère ligne du document représente le nom des différentes colonnes contenues dans le document les autres lignes représentes les informations sur les filières telle que le nom de l'établissement, son nom ou encore le nombre d'admis.
3. Il y a 118 lignes pour les obtenir nous avons utilisé la commande bash "wc -c données.csv" qui permet de connaitre le nombre de colonne d'un document.
4. C'est la colonne numéro 3 code UAI qui permet d'identifier un établissement.
5. C'est la colonne numéro 110 cod-aff_form qui permet d'identifier une formation.
6. Il y a une seule ligne.
7. Il y a deux colonnes identifiants les départements la colonne numéro 5 code départemental de l'établissement et la numéro 6 département de l'établissement.
8. Nous envisageons d'importé les données à l'aide de la commande : « `\copy import from fr-esr-parcoursup2.csv with (format csv, delimiter ';', HEADER);` »

```

```

9. Les problèmes que nous voyons sont la redondance des données et des colonnes il y a deux colonnes filière de formation. Il y a beaucoup de colonnes contenant des données calculables comme les pourcentages.

Q2.

5.

a. Il y a 13869 formations gérées par Parcoursup. Nous avons effectué la requête suivante pour le déterminé :

```
select count(n4) from import ;
```

b. Il y a 3602 établissement différent dans la base de données ce fut trouvé à l'aide de la commande `select count(distinct n4) from import;`

En tout la base de données contient 13869 établissements. La commande suivante nous à le déterminer : `select count(n4) from import;`

c. La commande `select count(n10) from import where n4 like '%Université de Lille%';` nous à permis de déterminer qu'il y a 146 formation.

d.

Il y a 10 formations au sein de l'IUT cela peut se trouver grâce à la commande suivante

```
 select n4, count(n10) from import group by n4 having n4 = 'Institut
universitaire de technologie de Lille - Université de Lille';
```

e. Le code du BUT est 0597215X celui -ci s'obtient avec la commande suivante :

```
select n3 from import where n10='BUT - Informatique' AND n9 ='Villeneuve-
d''Ascq';
```

III. Exercice 2

Q1 .

On l'obtient grâce à la commande "wc -c données.csv" nous trouvons 12423586 bytes pour la taille du fichier.

Q2.

Pour trouver le poids de la table import nous avons utilisé la requête suivante :

```
select pg_size_pretty(pg_relation_size('import'));
```

Ce qui nous donne 13 MB.

Q3.

Pour obtenir le poids de toutes les tables ventilées nous avons réutilisé la même commande que pour la table import mais avec les 6 tables ventilées puis nous avons additionné les poids. Vous trouverez sur l'image ci-dessous les différents poids. Nous obtenons un poids total de environ 64 MB.

```
but1=>  select pg_size_pretty(pg_relation_size('etablissement'));
    select pg_size_pretty(pg_relation_size('région'));
    select pg_size_pretty(pg_relation_size('fillière'));
    select pg_size_pretty(pg_relation_size('neobachelier'));
    select pg_size_pretty(pg_relation_size('admission'));
    select pg_size_pretty(pg_relation_size('bourse'));
 pg_size_pretty
----------------
 64 MB
(1 row)

 pg_size_pretty
----------------
 1160 kB
(1 row)

 pg_size_pretty
----------------
 1904 kB
(1 row)

 pg_size_pretty
----------------
 2288 kB
(1 row)

 pg_size_pretty
----------------
 1912 kB
(1 row)

 pg_size_pretty
----------------
 2208 kB
(1 row)
```

IV. Exercice 3

Q1. Nous avons utilisé les colonnes n57, n58 et n59 pour recalculer la colonne 56 à l'aide la commande suivante :

```
select n56, cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+ CAST(n59 as INTEGER)
as somme from import;
```

Nous obtenons le résultat suivant

```
but1=> select n56, cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+ CAST(n59 as INTEGER) as somme from import;
but1=> but1=> select n56, cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+ CAST(n59 as INTEGER) as somme from import limit 10;
 n56 | somme
-----+-------
 7   |     7
 15  |    15
 14  |    14
 39  |    39
 13  |    13
 10  |    10
 16  |    16
 14  |    14
 16  |    16
 4   |     4
(10 rows)
```

Q2. Pour vérifier nous avons cherché à voir s'il y avait des résultats différents entre la colonne n56 et son recalcul avec la commande suivante :

```
 select n56, cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+ CAST(n59 as INTEGER)
as somme from import
    where cast(n56 as INTEGER) <> cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+
CAST(n59 as INTEGER);
```

Nous obtenons zéro ligne comme le montre l'image suivante.

```
but1=> select n56, cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+ CAST(n59 as INTEGER) as somme from import
    where cast(n56 as INTEGER) <> cast(n57 as INTEGER)+ CAST(n58 as INTEGER)+ CAST(n59 as INTEGER) limit 10;
 n56 | somme
-----+-------
(0 rows)
```

Q3. Nous avons utilisé les colonnes suivantes : n51, et n57 ainsi que la commande

```
select n74, round(((cast(n51 as float)/(cast(n47 as float))))*100) as somme
from import where cast(n47 as float) <> 0  group by n74,n47,n51;
```

```
but1=>  select n74, round(((cast(n51 as float)/(cast(n47 as float))))*100) as somme from import where cast(n47 as float) <> 0  group by n74,n47,n51 limit 10;
 n74  | somme
------+-------
 30.0 |    30
 27.0 |    27
 4.0  |     4
 14.0 |    14
 17.0 |    17
 5.0  |     5
 23.0 |    23
 63.0 |    63
 41.0 |    41
 18.0 |    18
(10 rows)
```

Q4. Nous avons utilisé la même méthode que pour la question deux mais nous avons obtenu trois colonnes étant différente cela est dû à l'arrondi. Vous pouvez le constater sur l'image ci-dessous.

```
but1=> select round(cast(n74 as float)), round(((cast(n51 as float)/(cast(n47 as float))))*100) as somme
    from import
    where cast(n47 as float) <> 0 and round(cast(n74 as float)) <> round(((cast(n51 as float)/(cast(n47 as float))))*100)
    group by n74,n47,n51 limit 10;
 round | somme
-------+-------
    58 |    57
    58 |    57
    58 |    57
(3 rows)

but1=>
```

Q5. Nous avons utilisé les colonnes n53 et n47 avec la commande :

```
select n76, round(((cast(n53 as float)/(cast(n47 as float))))*100) as somme
    from import
```

```
    where cast(n47 as float) <> 0  group by n76,n47,n53;
```

Ce qui nous donne :

```
but1=> select n76, round(((cast(n53 as float)/(cast(n47 as float))))*100) as somme
    from import
    where cast(n47 as float) <> 0  group by n76,n47,n53 limit 10;
 n76  | somme
------+-------
 94.0 |    94
 79.0 |    79
 89.0 |    89
 52.0 |    52
 81.0 |    81
 93.0 |    93
 89.0 |    89
 84.0 |    84
 85.0 |    85
 95.0 |    95
(10 rows)
```

Q6. Pour la requête dans la table ventilée nous avons utilisé les colonnes popavproprin et effaccept ce qui donne le résultat suivant :

```
but1=> select round(cast(n76 as float)), round((propavproprin/effaccept)*100)
    from import join admission on cast(n76 as float) = pouradmis
    where effaccept <>0
    group by n76, propavproprin, effaccept limit 10;
 round | round
-------+-------
    82 |    82
    74 |    74
    82 |    82
    96 |    96
    82 |    82
    97 |    97
    54 |    54
    54 |    54
    85 |    85
     0 |     0
(10 rows)
```

Q7.

```
 select round(cast(n81 as float)) as n81, round(((cast(n55 as
float))/(cast(n57 as float)+cast(n58 as float)+cast(n59 as float)))*100) as
somme
    from import
    where cast(n57 as float) <> 0 and cast(n58 as float) <> 0 and cast(n59 as
float) <> 0
    group by n81,n55,n57,n58,n59;
```

```
but1=> select round(cast(n81 as float)) as n81, round(((cast(n55 as float))/(cast(n57 as float)+cast(n58 as float)+cast(n59 as float)))*100) as somme
    from import
    where cast(n57 as float) <> 0 and cast(n58 as float) <> 0 and cast(n59 as float) <> 0
    group by n81,n55,n57,n58,n59 limit 10;
 n81 | somme
-----+-------
  42 |    42
  24 |    24
   6 |     6
  52 |    52
  25 |    25
  29 |    29
  43 |    43
  29 |    29
   9 |     9
  16 |    16
(10 rows)
```

Q8.

```
select round(pouradbour) as pouradbour, round((effadbour/(cast(agen as float)+cast(atech as float)+cast(apro as float)))*100) as somme
    from admission
    where cast(agen as float) <> 0 and cast(atech as float) <> 0 and cast(apro as float) <> 0
    group by pouradbour,agen,apro,atech,effadbour;
```

```
but1=> but1=> select round(pouradbour) as pouradbour, round((effadbour/(cast(agen as float)+cast(atech as float)+cast(apro as float)))*100) as somme
    from admission
    where cast(agen as float) <> 0 and cast(atech as float) <> 0 and cast(apro as float) <> 0
    group by pouradbour,agen,apro,atech,effadbour limit 10;
 pouradbour | somme
------------+-------
         36 |    36
         52 |    52
         71 |    71
         17 |    17
         56 |    56
         23 |    23
         18 |    18
         59 |    59
        100 |   100
         41 |    41
(10 rows)
```