

Contents

Exploitation de base de données	1
.....	1
Exercice 1 : Comprendre les données	1
Exercice 2 : Ventiler les données	4
Exercice 3 : Requêtage	5

Exploitation de base de données

Auteurs: Dylan ZHANG, Lorenzo FLAGOTHIER

Exercice 1 : Comprendre les données

Q1. Analyse du fichier récupéré

1. Combien y-a t-il de lignes ? Justifiez !

Le fichier contient 13870 lignes en tout (en comptant l'entête) La commande utilisé pour compter les lignes du fichiers:

```
wc -l nom_du_fichier.csv
```

L'option -l permet de compter que les lignes du fichier

2. Que représente une ligne ?

Une ligne représente une formation spécifique.

3. Combien y-a t-il de colonnes ? Justifiez !

Le fichier contient 118 colonnes en tout on peut le retrouver avec la commande

```
head -n1 nom_du_fichier.csv | tr ';' '\n' | wc -l
```

4. Quelle colonne identifie un établissement ? (numéro et nom de col)

La colonne qui identifie un établissement est la colonne 4, établissement

La commande qu'on a utilisé:

```
head -n1 nom_du_fichier.csv | tr ';' '\n' | cat -n > head.txt
```

On peut renvoyer le résultat de la commande dans un fichier txt pour que ça soit plus pratique.

5. Quelle colonne identifie une formation ? (numéro et nom de col)

La colonne qui identifie une formation est la colonne 10, filière formation
On peut consulter la colonne depuis le fichier txt.

6. Combien de lignes font référence à notre BUT Informatique ?

Une ligne fait référence à notre BUT Informatique,

En utilisant la commande:

```
grep "BUT - Informatique" nom_du_fichier.csv | grep "Villeneuve-d'Ascq"
| wc -l
```

7. Quelle colonne identifie un département ? (numéro et nom)

La colonne qui identifie un département est la colonne 5, code départemental de l'établissement et la colonne 6, Département de l'établissement pour le nom du département. On peut retrouver ces données dans le dossier txt crée.

8. Comment envisagez vous importer ces données ?

On prévoit d'importer ces données dans une grand table tempo SQL.

9. Quels problèmes identifiez vous dans ces données initiales ? (il y en a sûrement plusieurs, expliquez les clairement).

Les différents problèmes qui peuvent être rencontré sont principalement : les doublons, les données null,

Q2 Importer les données

1. Fournir un fichier dico.xls permettant la correspondance entre les numéros de colonnes et les noms du fichier initial. Expliquez comment vous vous y êtes pris pour le constituer.

Pour générer chaque colonnes (118 colonnes en tout) on a la possibilité d'utiliser un fichier csv généré par un simple programme java.

2. Créer une table import permettant l'importation de ces données (fournir le code)

```
CREATE temp TABLE import (
  n1 text,n2 text,n3 text,n4 text,n5 text,n6 text,n7 text,n8 text,n9 text,
  n10 text,n11 text,n12 text,n13 text,n14 text,n15 text,n16 text,n17 text,
  n18 text,n19 text,n20 text,n21 text,n22 text,n23 text,n24 text,n25 text,
  n26 text,n27 text,n28 text,n29 text,n30 text,n31 text,n32 text,n33 text,
  n34 text,n35 text,n36 text,n37 text,n38 text,n39 text,n40 text,n41 text,
  n42 text,n43 text,n44 text,n45 text,n46 text,n47 text,n48 text,n49 text,
  n50 text,n51 text,n52 text,n53 text,n54 text,n55 text,n56 text,n57 text,
  n58 text,n59 text,n60 text,n61 text,n62 text,n63 text,n64 text,n65 text,
  n66 text,n67 text,n68 text,n69 text,n70 text,n71 text,n72 text,n73 text,
  n74 text,n75 text,n76 text,n77 text,n78 text,n79 text,n80 text,n81 text,
  n82 text,n83 text,n84 text,n85 text,n86 text,n87 text,n88 text,n89 text,
  n90 text,n91 text,n92 text,n93 text,n94 text,n95 text,n96 text,n97 text,
  n98 text,n99 text,n100 text,n101 text,n102 text,n103 text,n104 text,n105 text,
  n106 text,n107 text,n108 text,n109 text,n110 text,n111 text,n112 text,n113 text,
```

```
n114 text,n115 text,n116 text,n117 text,n118 text);
```

On peut utiliser un programme java pour écrire tout les colonnes de la tables

```
class tableau {
    public static void main (String[] args){
        String r = "";
        for(int i=1;i<=118-1;i++){
            r+="n"+i+" text,";
        }
        r+="n"+118+" text,\n";
        System.out.print(r);
    }
}
```

3. S'assurer que les types de colonnes soient les plus restrictifs possibles (des int pour les colonnes contenant des entiers, des char(x) pour les données textuelles de taille x etc ...)

```
CREATE temp TABLE import (
n1 int default null,n2 text default null,n3 char(8) default null,n4 text default null,n
n10 text default null,n11 text default null,n12 text default null,n13 text default null
n18 int default null,n19 int default null,n20 int default null,n21 int default null,n22
n26 int default null,n27 int default null,n28 int default null,n29 int default null,n30
n34 int default null,n35 int default null,n36 int default null,n37 text default null,n3
n42 int default null,n43 int default null,n44 int default null,n45 int default null,n46
n50 int default null,n51 real default null,n52 real default null,n53 real default null,
n58 int default null,n59 int default null,n60 int default null,n61 int default null,n62
n66 real default null,n67 real default null,n68 int default null,n69 int default null,n
n74 real default null,n75 real default null,n76 real default null,n77 real default null
n82 real default null,n83 real default null,n84 real default null,n85 real default null
n90 real default null,n91 real default null,n92 real default null,n93 real default null
n98 real default null,n99 real default null,n100 real default null,n101 real default nu
n106 text default null,n107 text default null,n108 text default null,n109 text default
n114 real default null,n115 real default null,n116 real default null,n117 text default
```

4. Remplir cette table avec les données récupérées (fournir le code) Pour remplir cette table avec les données récupérées on utilise la commande suivante.

```
\copy import from fr-esr-parcoursup.csv WITH(delimiter ';',
HEADER)
```

Elle permet de copier les données du fichier csv déjà traité avec unix donc sans le headers etc ..., et en définissant par délimiteur par un ';'.

5. En s'appuyant sur la table import fournir les requêtes et les réponses qui permettent de savoir

1. Combien il y a de formations gérés par ParcourSup ?

Il y a 13869 formations gérés par ParcourSup.

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

2. Combien il y a d'établissements gérés par ParcourSup ?

Il y a 3602 établissements gérés par ParcourSup.

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

3. Combien il y a de formations pour l'université de Lille ?

Il y a 124 formations pour l'université de Lille.

```
select count(*) from import where n4 = 'Université de Lille';
```

4. Combien il y a de formations pour notre IUT ?

Il y a 10 formations pour notre IUT.

```
select count(*) from import where n4 = 'Institut universitaire de technologie de Lille - Université de Lille';
```

5. Quel est le code du BUT Informatique de l'université de Lille ?

Le code du BUT informatique de l'université de Lille est 6888

```
select n110 from import where n4 = 'Institut universitaire de technologie de Lille - Université de Lille' and n10 = 'BUT - Informatique';
```

6. Citez 5 colonnes contenant des valeurs nulles

Les 5 colonnes suivantes contiennent des valeurs nulles.

n23 n38 n39 n55 n71

Exercice 2 : Ventiler les données

Q1 Normalisation des données 1. Fournir le MCD correspondant à votre structuration

Q2

1. Quelle taille en octet fait le fichier récupéré ?
2. Quelle taille en octet fait la table import ?
3. Quelle taille en octet fait la somme des tables créées ?

4. Quelle taille en octet fait la somme des tailles des fichiers exportés correspondant à ces tables ?

Exercice 3 : Requêtage

Q1. Ecrire une requête qui, à partir de import affiche le contenu de la colonne n56 et le re-calcul de celle-ci à partir d'autres colonnes de import (2 cols). ***

```
select n57,(n56-n58-n59) from import;
```

$n57 = n56 - n58 - n59$

car n56 est le total des néo bacheliers admis et n57 n58 b59 sont les néo bacheliers admis rangé par type de bac

Q2 Quelle requête vous permet de savoir que ce re-calcul est parfaitement exact ? ***

```
select count(*) from import where n57<>(n56-n58-n59);
```

Q3 Ecrire une requête qui, à partir de import affiche le contenu de la colonne n74 et le re-calcul de celle-ci à partir d'autres colonnes de import (2 cols). ***

```
select n74,ROUND((n51/n47)*100) from import where n47 <> 0;
```

$n74 = n51/n49$

Car n51 correspond au effectif des admis ayant reçu leur proposition d'admission à l'ouverture de la procédure principale et n49 au Effectif total des admis en phase principale.

Q4 Quelle requête vous permet de savoir que ce re-calcul est parfaitement exact ? ***

```
select count(*) from import where n47 <> 0 and n74<>ROUND((n51/n47)*100);
```

Q5 Ecrire une requête qui, à partir de import affiche le contenu de la colonne n76 et le re-calcul de celle-ci à partir d'autres colonnes de import (2 cols). A partir de combien de décimales ces données sont exactes ? ***

```
select n76,ROUND((n53/n47)*100) from import where n47 <> 0;
```

$n76 = n53/n49$

```
(select count(*) from import where n47 <> 0 and n76<>ROUND((n53/n47)*100);)
```

Sert a tester ça.

Q6 Fournir la même requête sur vos tables ventilées. ***

Q7 Ecrire une requête qui, à partir de import affiche la n81 et la manière de la recalculer. A partir de combien de décimales ces données sont exactes ? ***

```
select n81,round((cast(n55 AS NUMERIC)/n56)*100) from import
where n56 <> 0;
```

$n81 = n55 / n56$

Car n56 correspond au Effectif des admis néo bacheliers et n55 au Effectif des admis boursiers néo bacheliers.

Q8 ***