30/03/2023

# Compte rendu

## SAÉ 2.04 Exploitation d'une BD

Arthur KELLER et Maxime MURRAY

# 1) Compréhension des données

1. Combien y a-t-il de lignes dans chaque fichier ?

Pour savoir ceci, nous avons besoin de la commande ‘wc’ avec l’option ‘-l’. Ce qui nous donnerait ici :

- Pour le fichier athlete_events.csv

```
wc -l athlete_events.csv
```

Le fichier contient 271117 lignes

- Pour le fichier noc_regions.csv

```
wc -l noc_regions.csv
```

Nous obtenons 0 lignes mais quand nous lançons le fichier dans nano, il y a 231 lignes

2. Afficher uniquement la première ligne du fichier athlète ?

Pour savoir ceci, nous devons utiliser la commande ‘head’ avec l’option ‘-n’ suivi du nombre de ligne que l’on souhaite afficher

```
Head -n 1 athele_events.csv
```

Cela nous renvoi:
"ID","Name","Sex","Age","Height","Weight","Team","NOC","Games","Year","Season","City","Sport","Event","Medal"

3. Quel est le séparateur de champs ?

Dans ces fichiers, le séparateur utilisé est ‘ , ’.

4. Que représente une ligne ?

Une ligne du fichier correspond aux informations sur un athlète (âge, poids, taille, équipe et noc) et à quelle épreuve il a participé, la date de l’épreuve, et s’il a gagné une médaille

5. Combien y a-t-il de colonnes ?

Pour cela, nous allons utiliser la commande ‘head’ avec l’option -n puis la commande ‘tr’ et la commande ‘wc’ avec l’option ‘-l’

```
head -n 1 athlete_events.csv | tr "," "\n" | wc -l
```

Cela nous renvoi 15 donc il y a 15 colonnes dans le fichier *athlete_events.csv*.

1. Quelle colonne distingue les jeux d'été et d'hiver ?

C'est la colonne « Season » qui distingue les jeux d'été et les jeux d'hiver

Les jeux d'été sont notés « Summer » et pour ceux d'hiver, « Winter »

2. Combien de lignes font référence à Jean-Claude Killy ?

Pour répondre à cela, nous utilisons les commandes 'grep' et 'wc' avec l'option '-l'

```
grep "Jean-Claude Killy" athlete_events.csv | wc -l
```

6 lignes font référence à Jean-Claude Killy

3. Quel encodage est utilisé pour ce fichier ?

Pour savoir l'encodage d'un fichier, il suffit d'utiliser la commande 'file' avec l'option '-i'

```
file -i athlete_events.csv
```

Le fichier *athlete_events.csv* est encodé en us-ascii

4. Comment envisagez-vous l'importation de ces données ?

Nous envisageons ceci :

```
CREATE temporary TABLE import(
  id INT, name VARCHAR(108), sex CHAR(1),  age INT, height INT, weight
FLOAT,
  team VARCHAR(47), noc VARCHAR(3), game VARCHAR(11), year INT,
  season VARCHAR(26), city VARCHAR(22), sport VARCHAR(25),event
VARCHAR(85),
  medal VARCHAR(6));
\copy import from athlete_events.csv delimiter ',' HEADER csv NULL AS
'NA';
```

Le "NULL AS NA" nous permet de passer tous les « NA » à NULL pour nous faciliter l'import de ces données.

# 2) Importer les données

1. Créer une table *import* permettant l'importation de ces données

```
CREATE temporary TABLE import(
    id INT,
    name VARCHAR(108),
    sex CHAR(1),
    age INT,
    height INT,
    weight FLOAT,
    team VARCHAR(47),
    noc VARCHAR(3),
    game VARCHAR(11),
    year INT,
    season VARCHAR(26),
    city VARCHAR(22),
    sport VARCHAR(25),
    event VARCHAR(85),
    medal VARCHAR(6)
);
```

2. Remplir cette table avec les données récupérées

Nous allons cette commande pour remplir la table import :

```
\copy import from athlete_events.csv delimiter ',' HEADER csv NULL AS 'NA';
```

3. Certaines données sont incorrectes. C'est notamment le cas de toutes les informations avant 1920 et de toutes les épreuves "Artistiques". Supprimez toutes ces données douteuses.

```
DELETE FROM import WHERE year < 1920 or sport Like ('Art%');
```

4. Importer tel quel le fichier noc_regions.csv

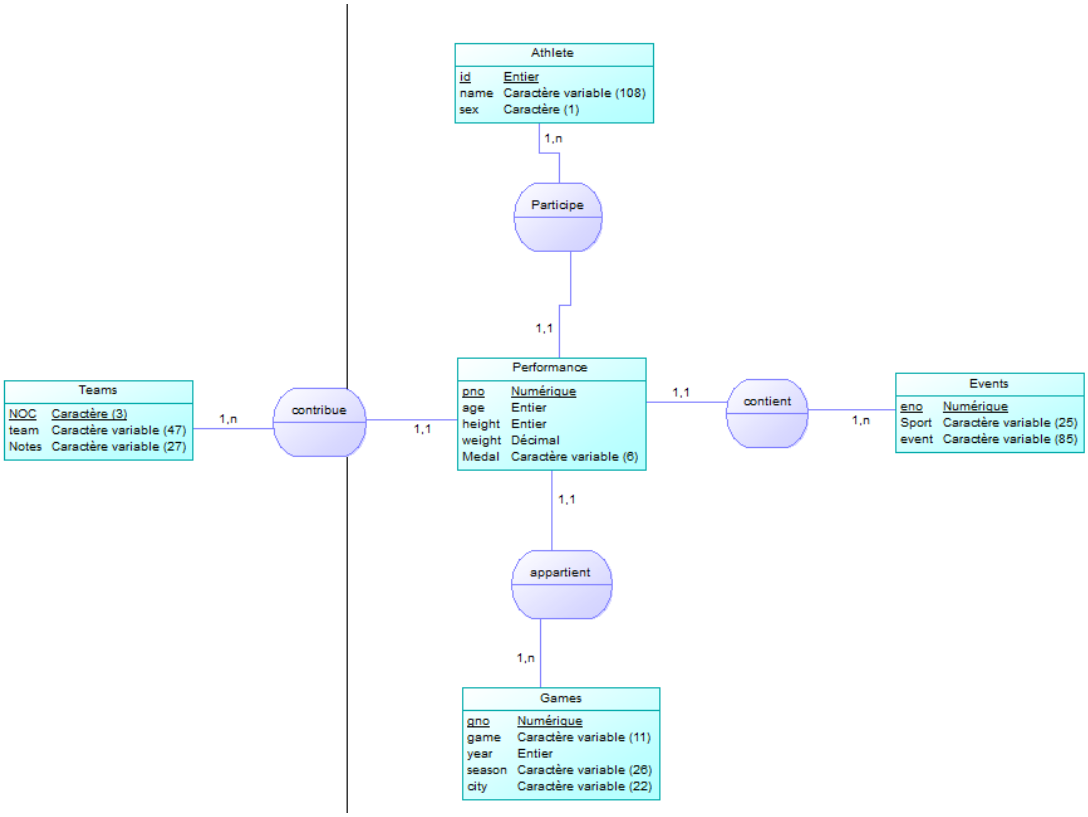
```
-- Creation de la table temporaire region

CREATE temporary TABLE region(
    noc varchar(3) PRIMARY KEY,
    region varchar(32),
    notes varchar(27));

-- Copie de tous du fichier noc_regions.csv dans la table region
\copy region FROM noc_regions.csv delimiter ',' HEADER csv;
```

# 4) Ventiler les données

1. Normalisation des données

Pour répondre à ce sujet, nous avons choisi de partir sur la base ou chaque athlète peut avoir une ou plusieurs performances. Les performances ont l'âge, le poids, la taille et la médaille de l'athlète en question. Une équipe est différenciée par son NOC et elle contribue à la réalisation d'une ou plusieurs performances. Un Events étant un sport avec toutes ses disciplines. Puis pour finir un Game est les jeux qui se sont déroulés depuis 1920. On y retrouve leur année de déroulement avec la ville et la saison associée.

MCD qui correspond à la structure choisie :




MLD :

Athletes(<u>id</u>, name, sex)

Teams(<u>noc</u>, team, notes)

Games(<u>gno</u>, game, year, season, city)

Events(<u>eno</u>, sport, event)

Performance(<u>pno</u>, age, height, weight, medal, #athletes_id, #teams_noc, #games_gno, #events_eno)

2. Une question de taille !

1)Quelle taille en octet fait le fichier récupéré ?

Nous pouvons savoir ceci en utilisant la commande *ls* avec son option *-l* et la commande *cut* avec ses option *-d* et *-f*

- Quand le fichier est en .zip :

```
ls -l data-olympique.zip | cut -d " " -f5
```

Ce qui nous renvoi 5544725 soit 5 544 725 octets

- Quand le fichier est extrait :

```
ls -l data-olympique | cut -d " " -f5
```

Ce qui nous renvoi 41500688 soit 41 500 688 octets

2)Quelle taille en octet fait la table import ?

```
SELECT pg_relation_size('import') AS "Taille de la table import";
```

Ce qui nous donne 47259648 soit 47 259 648 octets

3) Quelle taille en octet fait la somme des tables créées ?

```
SELECT sum(pg_total_relation_size) AS "Somme des tables créées"
FROM (
        SELECT pg_total_relation_size('import')
        UNION SELECT pg_total_relation_size('region')
        UNION SELECT pg_total_relation_size('games')
        UNION SELECT pg_total_relation_size('events')
        UNION SELECT pg_total_relation_size('athlete')
        UNION SELECT pg_total_relation_size('performance')
        UNION SELECT pg_total_relation_size('team')) AS union_tables;
```

Ce qui nous renvoi 82960384 soit 82 960 384 octets

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

Dans un premier nous devons copier toutes nos tables en fichier .csv :

```
COPY performance TO 'performance.csv' WITH DELIMITER ',' CSV HEADER;
COPY events TO 'events.csv' WITH DELIMITER ',' CSV HEADER;
COPY athlete TO 'athlete.csv' WITH DELIMITER ',' CSV HEADER;
COPY games TO 'games.csv' WITH DELIMITER ',' CSV HEADER;
COPY team TO 'team.csv' WITH DELIMITER ',' CSV HEADER;
```

Puis nous exécutons ce script :

```
num=0
for n in $(stat *.csv | grep "Size" | cut -d ' ' -f 4); do
  num=$((num + n))
done
echo $num
```

Ce qui nous donne 95 399 881 octets

# 6) Personnalisation du rapport

Pour cet exercice, nous avons décidé de traiter les Français pratiquant l'athéisme.

- Pour ce faire en première requête nous avons récupéré le nombre de participation total depuis les jeux 1920 :

```
SELECT COUNT(*) AS "Nombre de participation total"
FROM events AS e INNER JOIN performance AS p
    ON e.eno = p.events_eno
WHERE sport = 'Athletics'
AND p.team_noc = 'FRA';
```

Cette requête nous renvoi 1343.

- Ensuite nous allons compter le nombre de participation par épreuve depuis 1920 en aillant l'épreuve avec le plus de participation en première.

```
SELECT e.event, COUNT(*) AS "Nombre de participant par epreuve"
FROM events AS e INNER JOIN performance AS p ON e.eno = p.events_eno
WHERE e.sport = 'Athletics'
AND p.team_noc = 'FRA'
GROUP BY e.event
ORDER BY "Nombre de participant par épreuve" DESC;
```

Avec le résultat de cette requête, nous constatons que les deux premières épreuves avec le plus de participations sont les relais 4*100m et 4*400m ce qui un peu normal car 4 athlètes pratiquent cette discipline par année lorsqu'ils arrivent à se qualifier bien sur

- Pour cette troisième requête, nous allons calculer le poids et la taille au JO de rio (2016)

```
SELECT ROUND(AVG(p.weight)) AS "Moyenne du poids des athletes francais",
ROUND(AVG(p.height), 1) AS "Moyenne de la taille des athletes francais"
FROM performance AS p INNER JOIN events AS e ON p.events_eno = e.eno
      INNER JOIN games AS g ON g.gno = p.games_gno
WHERE e.sport = 'Athletics'  AND p.team_noc = 'FRA' AND g.year = 2016;
```

Lors de cette requête, nous avons rencontré un problème pour le 'ROUND' du poids. Nous n'avons jamais réussi à arrondir à un chiffre après la virgule mais pour arrondir à l'unité près il y avait aucun problème. Du coup nous avons décidé de l'arrondir à l'unité près

- Pour notre dernière requête, nous avons décidé de donner les athlètes avec le plus de titre au JO :

```
SELECT a.name, COUNT(*) AS "Nombre de médaille d'or"
FROM performance AS p INNER JOIN athlete AS a ON p.athlete_id = a.id
    INNER JOIN events AS e ON p.events_eno = e.eno
WHERE e.sport = 'Athletics'
AND p.team_noc = 'FRA'
AND medal = 'Gold'
GROUP BY a.name
ORDER BY "Nombre de médaille d'or" DESC ;
```

Avec cette requête, nous constatons que peu d'athlètes ont obtenu un titre depuis 1920. Seulement 10 sont sur cette liste, l'athlète français le plus titrés est une femme qui est Marie-José Père, la coureuse de 400m la plus connu de France avec 2 titres sur cette discipline. Son premier titre était au JO de Barcelone en 1992 et son second au JO d'Atlanta en 1996 où elle a réussi un doublé sur 400m et 200m.