

BDD

CHEHIRE Marc et ROUYRRE Rodolphe

31 Mai 2020

1 Création de la base

Nous avons créé notre base de données que nous avons nommée "projet". Nous y avons ajouté les tables correspondantes au modèle relationnel.

2 Explication des types

Il est important de bien choisir les types pour que la base soit bien conçue et plus performante.

Tout d'abord, tous les identifiants sont unsigned car ils sont forcément positifs. De plus, ils sont de type smallint. En effet, ils ne peuvent pas être de type tinyint car nous admettons que certaines tables puissent posséder plus de 255 lignes. En revanche, elles ne pourront pas en avoir plus de 65535. Ils sont incrémentés automatiquement et n'acceptent pas NULL comme valeur par défaut.

Les attributs sexe et date_naissance stockent des chaînes de caractères de taille fixe donc ils sont de type char, respectivement de taille 1 et 10. Sexe pourrait être de type enum mais ce type est propre à MySQL et donc fortement déconseillé. Quant aux attributs nom, prenom et nomPays, leur taille peut beaucoup varier donc ils sont de type varchar de taille 30.

InnoDB permet l'utilisation de clés étrangères contrairement à MyISAM.

3 Explication des requêtes

Nous allons présenter les 12 requêtes demandées. Tout d'abord, nous expliquerons les requêtes particulières exigées, puis celles supplémentaires que nous avons trouvées intéressantes.

3.1 Requête avec le mot-clé NOT IN

Question :

Quelles sont les nationalités qui ne sont pas apparues sur au moins un podium du 100m?

Algèbre relationnelle :

Sous-requête :

$$A = \pi_{idSportif}(\sigma_{idDiscipline=1,medaille \neq 'N'}(Performance))$$

Requête principale :

$$\pi_{nomPays}(\sigma_{idSportif \notin A}(Sportif))$$

Requête en SQL :

```
01 |      SELECT DISTINCT nomPays
02 |      FROM Sportif
03 |      WHERE idSportif NOT IN (
04 |          SELECT idSportif
05 |          FROM Performance
06 |          WHERE Performance.idDiscipline = 1
07 |          AND Performance.medaille <> 'N'
08 |      );
```

Explication :

Nous récoltons tous les identifiants des sportifs qui ont enregistré des performances donnant lieu à une médaille. Il nous suffit ensuite de projeter les noms de pays dont les id sportifs associés n'appartiennent pas à cet ensemble.

3.2 Requête avec le mot-clé IN

Question :

Comment s'appellent les sportifs jamaïquains ayant disputé au moins une épreuve d'athlétisme?

Algèbre relationnelle :

Sous-sous-requête :

$$A = \pi_{idDiscipline}(\sigma_{nomD='Athletisme'}(Discipline))$$

Sous-requête :

$$B = \pi_{idSportif}(\sigma_{idDiscipline \in A}(Performance))$$

Requête principale :

$$\pi_{prenom,nom}(\sigma_{nomPays='Jamaïque',idSportif \in B}(Sportif))$$

Requête en SQL :

```
01 |      SELECT prenom, nomS AS nom
02 |      FROM Sportif
03 |      WHERE nomPays='Jamaïque'
04 |      AND idSportif IN (
05 |          SELECT idSportif
06 |          FROM Performance
07 |          WHERE idDiscipline IN (
08 |              SELECT idDiscipline
09 |              FROM Discipline
10 |              WHERE nomD='Athlétisme'
11 |          )
12 |      );
```

Explication :

Nous commençons par récolter les id des disciplines qui font partie de l'athlétisme. Pour trouver les sportifs pratiquant ces épreuves, nous pouvons faire une sélection sur la table Performance avec la condition IN. Pour trouver les sportifs jamaïquains, nous pouvons faire une sélection sur la table Sportif avec la condition nomPays='Jamaïque'. Nous pouvons ensuite ajouter une sélection en utilisant la première requête grâce au mot-clé IN. Si on ne veut que les noms et prénoms des sportifs, on effectue une projection sur ces deux colonnes.

3.3 Jointure

Question :

Comment s'appellent les sportifs jamaïquains ayant pratiqué le 100m, quelle a été leur performance? Lors de quelle édition?

Algèbre relationnelle :

$$\pi_A(\sigma_B(Sportif \bowtie_{idSportif} (Performance \bowtie_{idEdition} Edition)))$$
$$A = prenom, nomS, score, medaille, record, annee, eteOuHiver, ville$$
$$B = nomPays = 'Jamaïque', idDsicipline = 1$$

Requête en SQL :

```
01 |      SELECT prenom, nomS AS nom, score, medaille, record,
02 |             annee, eteOuHiver AS saison, ville
03 |      FROM Sportif, Performance, Edition
04 |      WHERE Sportif.nomPays='Jamaïque'
05 |      AND Sportif.idSportif=Performance.idSportif
06 |      AND Performance.idDiscipline=1
07 |      AND Performance.idEdition=Edition.idEdition;
```

Explication :

Nous voulons enrichir l'information fournie par chaque ligne, et nous procédons à 2 jointures, elles nous permettent d'afficher toutes les informations de A. Nous ajoutons à Sportif chacune des ses performances grâce à Performance, et nous ajoutons à Performance l'édition durant laquelle a été réalisée celle-ci, grâce à Edition.

3.4 Jointure avec trois fois la même table

Question :

Quels ont été les podiums des épreuves du 200m disputées?

Requête en SQL :

```
01 |      SELECT p1.idSportif as 'Or', p2.idSportif as 'Argent',
02 |      p3.idSportif as 'Bronze', p1.idEdition
03 |      FROM Performance p1, Performance p2, Performance p3
04 |      WHERE p1.medaille = 'Or'
05 |      AND p1.idDiscipline = 2
06 |      AND p2.medaille = 'Ar'
07 |      AND p2.idDiscipline = 2
08 |      AND p3.medaille = 'Br'
09 |      AND p3.idDiscipline = 2
10 |      AND p1.idEdition = p2.idEdition
      AND p3.idEdition = p2.idEdition;
```

Explication : de la même façon nous réalisons des jointures, qui enrichissent cette fois la même table pour en comparaison des sportifs et obtenir un podium.

3.5 Requête imbriquée

Question :

Qui sont les sportifs qui ont aujourd'hui moins de 35 ans ?

Algèbre relationnelle :

$$\pi_A(\sigma_{age < 35}(\pi_{A, date_naissance, age}(Sportif)))$$
$$A = sexe, prenom, nom, age, nomPays$$

Requête en SQL :

```
01 |      SELECT sexe, prenom, nom, age, nomPays
02 |      FROM (
03 |          SELECT sexe, prenom, nomS AS nom, date_naissance,
04 |          YEAR(NOW()) - YEAR(date_naissance) AS age, nomPays
      FROM Sportif) AS sportif_et_age
```

```

05 | WHERE age < 35
06 | ORDER BY sexe, age;

```

Explication :

Pour répondre à la question, nous avons trouvé judicieux de diviser la requête en deux. En effet, notre base ne contient pas l'âge que les sportifs ont aujourd'hui. Cependant, nous pouvons trouver celui-ci car nous possédons la date de naissance de chaque sportif. Il suffit d'utiliser la fonction YEAR() qui renvoie l'année d'un attribut au format datetime. Avec YEAR(NOW()) nous obtenons l'année de la date d'aujourd'hui. Ensuite, une simple sélection suivant l'âge, puis une projection donnent le résultat. Bien sûr, cette requête ne prend pas en compte le mois ni le jour de naissance, ce qui serait possible suivant la même logique, mais la requête serait beaucoup plus longue.

3.6 Requête corrélée

Question :

Quels sont les sportifs qui portent le même prénom ?

Algèbre relationnelle :

$$\sigma_{C1.prenom=C2.prenom}(\sigma_{C1.idSportif \neq C2.idSportif}(C2))$$

Requête en SQL :

```

01 | SELECT *
02 | FROM Sportif C1
03 | WHERE EXISTS (
04 |     SELECT *
05 |     FROM Sportif C2
06 |     WHERE C1.idSportif <> C2.idSportif
07 |     AND C1.prenom = C2.prenom
08 | );

```

Explication :

On utilise une requête corrélée pour comparer 2 attributs d'une même table. Il est donc indispensable de passer par des alias. Ici, l'attribut et la table en question sont respectivement 'prenom' et 'Sportif'. Le mot-clé EXISTS permet l'utilisation de la sous-requête.

3.7 Requête avec les mots-clés COUNT et GROUP BY

Question :

Combien de performances a effectuées Usain Bolt ?

Algèbre relationnelle :

$$\pi_{\text{prenom}, \text{nomS}, \text{COUNT}(\text{score})}(\sigma_{\text{Performance.idSportif}=1}(\text{Sportif} \bowtie_{\text{idSportif}} \text{Performance}))$$

Requête en SQL :

```
01 | SELECT prenom, nomS AS nom, COUNT(score) AS nb_perf
02 | FROM Sportif, Performance
03 | WHERE Performance.idSportif=1
04 | AND Sportif.idSportif=Performance.idSportif
05 | GROUP BY prenom, nomS
```

Explication :

Cette requête est une simple sélection sur une jointure entre les tables Sportif et Performance.

3.8 Division

Question :

Quelles sont les disciplines présentes à toutes les éditions d'été ?

Algèbre relationnelle :

Discipline/Propose

$$\pi_{\text{idDiscipline}}(\text{Propose}) - \pi_{\text{idDiscipline}}(\pi_{\text{idDiscipline}}(\text{Propose}) \times \pi_{\text{idEdition}}(\text{Edition}) \\ - \pi_{\text{idDiscipline, idEdition}}(\text{Edition}))$$

Requête en SQL :

```
01 | SELECT DISTINCT idDiscipline FROM Propose
02 | WHERE NOT EXISTS
03 |   (SELECT * FROM Edition
04 |    WHERE NOT EXISTS
05 |      (SELECT * FROM Edition e, Propose p
06 |       WHERE e.idEdition = p.idEdition
07 |       AND Propose.idDiscipline = p.idDiscipline
08 |       AND e.idEdition = Edition.idEdition
09 |      ) AND Edition.eteOuHiver='E'
10 |   );
```

Explication :

Cette question peut-être reformulée ainsi : 'Quelles sont les disciplines telles qu'il n'existe pas d'édition où elles n'ont pas été proposées ?'. Cette question fait intervenir une double négation, elle correspond donc à une division. Cette

double négation peut être modélisée grâce au mot-clé NOT EXISTS. Pour la requête SQL, on reprend la correction de la question 5 du TD5 :

```

01 |      SELECT DISTINCT biere FROM aime
02 |      WHERE NOT EXISTS
03 |          (SELECT * FROM frequente
04 |           WHERE NOT EXISTS
05 |             (SELECT * FROM frequente f, aime a
06 |              WHERE f.buveur=a.buveur
07 |                 AND aime.biere=a.biere
08 |                 AND f.buveur = frequente.buveur
09 |             )
10 |          );

```

Et on remplace biere par idDiscipline, aime par Propose, frequente par Edition et buveur par idEdition.

3.9 Requêtes supplémentaires

3.10 Conjonction avec LIKE

Question :

Qui sont les sportifs Jamaïcains dont le nom commence par un W ?

Algèbre relationnelle :

$$\sigma_{nomPays='Jamaïque', nomS LIKE 'W\%'}(Sportif)$$

Requête en SQL :

```

01 |      SELECT *
02 |      FROM Sportif
03 |      WHERE nomPays='Jamaïque'
04 |      AND nomS LIKE 'W%';

```

Explication :

Cette requête est une simple conjonction, c'est-à-dire plusieurs sélections. La première concerne le nom du pays qui doit être la Jamaïque, la deuxième concerne le nom du sportif, il doit commencer par un W. Cela est possible grâce au mot-clé LIKE, qui porte sur une chaîne de caractères et signifie 'contient'. Le symbole % signifie 'n'importe quelle chaîne de caractères. Ces sélections portent sur la table Sportif.

3.10.1 Requête avec AVG()

Question :

Quelle est la moyenne d'âge des sportifs pour l'épreuve de bobsleigh en 2014 ?

Algèbre relationnelle :

Sous-requête :

$$\pi_{prenom,nomS,date_naissance,age}(\sigma_{nomD='Bobsleigh'}(Sportif \bowtie_{idSportif} Performance \bowtie_{idDiscipline} Discipline))$$

Requête en SQL :

```
01 | SELECT AVG(age)
02 | FROM (
03 |     SELECT prenom, nomS AS nom, date_naissance, annee -
04 |         YEAR(date_naissance) AS age
05 |     FROM Sportif, Performance, Discipline, Edition
06 |     WHERE Discipline.nomD='Bobsleigh'
07 |     AND Sportif.idSportif=Performance.idSportif
08 |     AND Performance.idDiscipline=Discipline.
09 |         idDiscipline
10 |     AND Edition.idEdition = 2
11 |     AND Edition.idEdition = Performance.idEdition
12 | ) AS sportif_et_age;
```

Explication :

On utilise une requête imbriquée faisant intervenir deux jointures. Pour avoir la moyenne d'âge, on utilise la fonction AVG sur l'âge.

3.10.2 Requête avec MIN()

Question :

Quel est le record olympique jamaïquain au 100m ?

Algèbre relationnelle :

Sous-requête :

$$\pi_{prenom,nomS,score}(\sigma_{Sportif.nomPays='Jamaïque',Performance.idDiscipline=1}(Sportif \bowtie_{idSportif} Performance))$$

Requête en SQL :

```
01 | SELECT MIN(score) AS record_olympique_jamaiquain
02 | FROM (
03 |     SELECT prenom, nomS AS nom, score
04 |     FROM Sportif, Performance
05 |     WHERE Sportif.nomPays='Jamaïque'
06 |     AND Sportif.idSportif=Performance.idSportif
07 |     AND Performance.idDiscipline=1
08 | ) AS performances_jamaiquaines;
```


Explication :

La sous requête est une simple sélection sur une jointure entre les tables Sportif et Performance. Elle permet de trouver les sportifs jamaïquains faisant du 100m. Enfin, grâce à la fonction MIN() sur l'attribut score on peut trouver le record olympique jamaïquain.

3.10.3 Union

Question :

Quels sont les pays ayant proposé des sportifs et/ou accueilli une édition ?

Algèbre relationnelle :

$$\pi_{nomPays}(Sportif) \cup \pi_{nomPays}(Edition)$$

Requête en SQL :

```
01 | SELECT nomPays FROM Sportif
02 | UNION
03 | SELECT nomPays FROM Edition;
```

Explication :

L'union permet d'afficher les résultats de deux tables si celles-ci ont le même schéma.