

masquer: 1

## TME 4 : PREMIERS PAS EN SQL

### 1. SGBD H2

Le système utilisé pendant les TME est H2.

1. Commencez par vous connecter au serveur H2 en vous aidant de la documentation suivante: [SGBD H2](#)
2. Créer les tables et charger les données: [BD-JeuxOlympiques-v1](#)

On considère le schéma de la base JeuxOlympique2014 donné en TD où l'on a modifié le schéma de **RangEpreuve**:

- **Athlete** (*nom, prenom, dateNaissance, pays*)
- **RangEpreuve**(*sport, epreuve, categorie, nomAth\*, prenomAth\*, dateDebut, dateFin, rang*)

### 2. PREMIÈRES REQUÊTES SOUS H2

Renvoie la liste des attributs de la relation Athlete.

2.1 Tous les n-uplets stockés dans la relation Athlete.

```
select * from Athlete;
```

2.2 Le nombre de n-uplets stockés dans la relation Athlete.

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

### 3. REQUÊTES

#### Exercice 1:

Exprimer les requêtes suivantes en SQL :

3.1 Les athlètes (nom, prénom) d'Inde

**Résultat :** IQBAL Nadeem, KESHAVAN Shiva, THAKUR Himanshu

3.2 Le prénom des athlètes dont le nom est 'GOLD'.

**Résultat :** Arielle, Gracie, Taylor

3.3 La nationalité de AONO Ryo.

**Résultat :** Japon

3.4 Le gagnant du médaille d'or de chaque épreuve. Triez le résultat par sport, puis par épreuve.

**Résultat :** (74 lignes)

3.5 Toutes les épreuves (sport, épreuve, catégorie) triées par la date de fin de l'épreuve

**Résultat :** (73 lignes)

- 3.6 Les athlètes nés entre 1990 et 1995.

**Résultat :** (87 lignes)

- 3.7 Les athlètes suisses ayant participé au sport 'Biathlon' et disqualifié à au moins une épreuve de ce sport (solution sans jointure, avec une opération ensembliste).

**Résultat :** GASPARIN Elisa

- 3.8 Les épreuves dans lesquelles il n'y a pas eu une médaille d'argent (solution sans jointure, avec une opération ensembliste).

**Résultat :** Ski alpin     Descente     Femmes

## Exercice 2:

On considère maintenant le schéma complet de la base JO2014

**PAYS** (CODEPAYS, NOMP)

**SPORT** (SID, NOMSP)

**EPRUVE** (EPID, SID\*, NOMEP, CATÉGORIE, DATEDEBUT, DATEFIN)

**ATHLETE** (AID, NOMATH, PRENOMATH, DATENAISANCE, CODEPAYS\*)

**EQUIPE** (EQID, CODEPAYS\*)

**ATHLETESEQUIPE** (EQID\*, AID\*)

**RANGINDIVIDUEL** (EPID\*, AID\*, RANG)

**RANGEQUIPE** (EPID\*, EQID\*, RANG)

La relation **PAYS** contient le code et le nom de tous les pays, même si ils n'ont pas participé aux Jeux Olympiques. Les sports (n-uplets de la relation **SPORT**) sont un ensemble d'épreuves (n-uplets de la relation **EPRUVE**). Pour chaque épreuve on connaît son nom et les dates de début et fin de l'épreuve. Les épreuves peuvent être individuelles ou par équipe. Dans le premier cas, la participation des athlètes (n-uplets de la relation **ATHLETE**) est stockée dans la table **RANGINDIVIDUEL** qui contient en plus le rang qu'ils ont obtenu (1 pour la médaille d'or). Pour les épreuves par équipe les résultats sont stockés dans la relation **RANGEQUIPE**, alors que l'information sur le pays de chaque équipe et ses participants est stockée dans les relations **EQUIPE** et **ATHLETESEQUIPE**. Dans les relations **RANGINDIVIDUEL** et **RANGEQUIPE** l'attribut rang est égal à null si l'athlète ou l'équipe a été disqualifiée.

Le schéma E/A pour la BD JO\_v2 se trouve [ici](#).

Créer les tables et charger les données: [BD-JeuxOlympiques-v2](#)

Exprimer les requêtes suivantes en SQL :

- 3.9. Les épreuves (sport, épreuve, catégorie) individuelles.

**Résultat :** (73 lignes)

- 3.10. Les résultats (nom, prénom, pays de l'athlète et rang) de l'épreuve '1000m' du 'Patinage de vitesse' pour les 'Femmes'.

**Résultat :** (36 lignes)

- 3.11. Le nom et prénom des athlètes qui forment l'équipe qui a gagné la médaille d'or dans l'épreuve 'relais 4x6km' de 'Biathlon' de 'Femmes'.

**Résultat :** SEMERENKO Vita, SEMERENKO Valj, DZHYZMA Juliya, PIDHRUSHNA Olena