

## BDD – DEVOIR SUR TABLE

### EXERCICE 1: MUSIQUE

L'énoncé de cet exercice utilise les mots clefs du langage SQL suivants : SELECT, FROM, WHERE, JOIN ON, UPDATE, SET, INSERT INTO VALUES, COUNT, ORDER BY.

- La clause ORDER BY suivie d'un attribut permet de trier les résultats par ordre croissant de l'attribut ;
- COUNT (\*) renvoie le nombre de lignes d'une requête ;

Un musicien souhaite créer une base de données relationnelle contenant ses morceaux et interprètes préférés. Pour cela il utilise le langage SQL. Il crée une table morceaux qui contient entre autres les titres des morceaux et leur année de sortie :

id_morceau	titre	annee	id_interprete
1	Like a Rolling Stone	1965	1
2	Respect	1967	2
3	Imagine	1970	3
4	Hey Jude	1968	4
5	Smells Like Teen Spirit	1991	5
6	I Want To hold Your Hand	1963	4

Il crée la table interpretes qui contient les interprètes et leur pays d'origine :

id_interprete	nom	pays
1	Bob Dylan	États-Unis
2	Aretha Franklin	États-Unis
3	John Lennon	Angleterre
4	The Beatles	Angleterre
5	Nirvana	États-Unis

id\_morceau de la table morceaux et id\_interprete de la table interpretes sont des clés primaires.

L'attribut id\_interprete de la table morceaux fait directement référence à la clé primaire de la table interpretes.

- Écrire le résultat de la requête suivante :  

```
SELECT titre FROM morceaux WHERE id_interprete = 4;
```
  - Écrire une requête permettant d'afficher les noms des interprètes originaires d'Angleterre.
  - Écrire le résultat de la requête suivante :  

```
SELECT titre, annee FROM morceaux ORDER BY annee;
```
  - Écrire une requête permettant de calculer le nombre de morceaux dans la table morceaux.
  - Écrire une requête affichant les titres des morceaux par ordre alphabétique.
- Citer, en justifiant, la clé étrangère de la table morceaux.
  - Écrire un schéma relationnel des tables interpretes et morceaux.
  - Expliquer pourquoi la requête suivante produit une erreur :  

```
INSERT INTO interpretes VALUES (1, 'Trust', 'France');
```
- Une erreur de saisie a été faite. Écrire une requête SQL permettant de changer l'année du titre « Imagine » en 1971.
  - Écrire une requête SQL permettant d'ajouter l'interprète « The Who » venant d'Angleterre à la table interpretes. On lui donnera un id\_interprete égal à 6.
  - Écrire une requête SQL permettant d'ajouter le titre « My Generation » de « The Who » à la table morceaux. Ce titre est sorti en 1965 et on lui donnera un id\_morceau de 7 ainsi que l'id\_interprete qui conviendra.
- Écrire une requête permettant de lister les titres des interprètes venant des États-Unis.

## EXERCICE 2: METEO

Dans le cadre d'une étude sur le réchauffement climatique, un centre météorologique rassemble des données. On considère que la base de données contient deux relations (tables). La relation `Centres` qui contient l'identifiant des centres météorologiques, la ville, la latitude, la longitude et l'altitude du centre. La relation `Mesures` qui contient l'identifiant de la mesure, l'identifiant du centre, la date de la mesure, la température, la pression et la pluviométrie mesurées.

Le schéma relationnel de la relation `Centres` est le suivant :

`Centres(id_centre: INT, nom_ville: VARCHAR, latitude: FLOAT, longitude: FLOAT, altitude: FLOAT)`

Le schéma relationnel de la relation `Mesures` est le suivant :

`Mesures(id_mesure: INT, id_centre: INT, date: DATE, temperature: FLOAT, pression: INT, pluviometrie: FLOAT)`.

Relation `Centres`

id centre	nom ville	latitude	longitude	altitude
213	Amiens	49.894	2.293	60
138	Grenoble	45.185	5.723	550
263	Brest	48.388	-4.49	52
185	Tignes	45.469	6.909	2594
459	Nice	43.706	7.262	260
126	Le Puy-en-Velay	45.042	3.888	744
317	Gérardmer	48.073	6.879	855

Relation `Mesures`

id mesure	id centre	date	temperature	pression	pluviometrie
1566	138	2021-10-29	8.0	1015	3
1568	213	2021-10-29	15.1	1011	0
2174	126	2021-10-30	18.2	1023	0
2200	185	2021-10-30	5.6	989	20
2232	459	2021-10-31	25.0	1035	0
2514	213	2021-10-31	17.4	1020	0
2563	126	2021-11-01	10.1	1005	15
2592	459	2021-11-01	23.3	1028	2
3425	317	2021-11-02	9.0	1012	13
3430	138	2021-11-02	7.5	996	16
3611	263	2021-11-03	13.9	1005	8
3625	126	2021-11-03	10.8	1008	8

1.

- Proposer une clé primaire pour la relation `Mesures`. Justifier votre choix.
- Avec quel attribut peut-on faire une jointure entre la relation `Centres` et la relation `Mesures` ?

2.

- Qu'affiche la requête suivante ?  
`SELECT * FROM Centres WHERE altitude>500;`
- On souhaite récupérer le nom de la ville des centres météorologiques situés à une altitude comprise entre 700m et 1200m. Ecrire la requête SQL correspondante.
- On souhaite récupérer la liste des longitudes et des noms des villes des centres météorologiques dont la longitude est supérieure à 5. La liste devra être triée par ordre alphabétique des noms de ville. Ecrire la requête SQL correspondante.

3.

- Qu'affiche la requête suivante ?  
`SELECT * FROM Mesures WHERE date= 2021-10-30 ;`
- Écrire une requête SQL permettant d'ajouter une mesure prise le 8 novembre 2021 dans le centre numéro 138, où la température était de 11°C, la pression de 1013 hPa et la pluviométrie de 0mm. La donnée dont l'attribut est `id_mesure` aura pour valeur 3650.

4.

- Expliquer ce que renvoie la requête SQL suivante ?  
`SELECT * FROM Centres WHERE latitude = (SELECT MIN(latitude) FROM Centres);`
- Écrire une requête SQL donnant la liste des villes dans lesquelles on a enregistré une température inférieure à 10°C en octobre 2021. On utilisera le mot clé `DISTINCT` afin d'éviter d'avoir des doublons. On rappelle que l'on peut utiliser les opérateurs de comparaison avec les dates.