# **Sujet BAC 3:** Langage SQL



# 1. Métropole J1 : Base de données cinématographique

### \* Métropole J1 : Base de données cinématographique

- 3 relations dans une base de données sur le cinéma
- 2 tables: individu et realisation

On pourra utiliser les mots clés SQL suivants : SELECT, FROM, WHERE, JOIN, ON, INSERT, INTO, VALUES, UPDATE, SET, AND.

Nous allons étudier une base de données traitant du cinéma dont voici le schéma relationnel qui comporte 3 relations :

- la relation individu (id ind, nom, prenom, naissance)
- la relation realisation (<u>id rea</u>, titre, annee, type)
- la relation emploi (<u>id emp</u>, description, #id\_ind, #id\_rea)

Les clés primaires sont soulignées et les clés étrangères sont précédées d'un # .

Ainsi emploi.id\_ind est une clé étrangère faisant référence à individu.id\_ind.

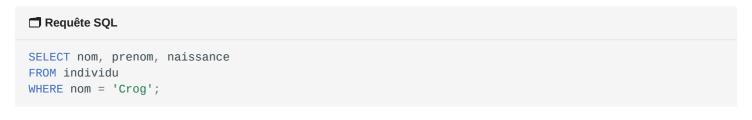
Voici un extrait des tables individu et realisation :

## **Extrait de individu**

id_ind	nom	prenom	naissance
105	'Hulka'	'Daniel'	'01-06-1968'
403	'Travis'	'Daniel'	'10-03-1968'
688	'Crog'	'Daniel'	'07-07-1968'
695	'Pollock'	'Daniel'	'24-08-1968'

<b>Extrait de </b> realisation			
titre	annee	type	
'Casino Imperial'	2006	'action'	
'Ciel tombant'	2012	'action'	
'Fantôme'	2015	'action'	
'Mourir pour attendre'	2021	'action'	
	'Casino Imperial'  'Ciel tombant'  'Fantôme'	titre annee  'Casino Imperial' 2006  'Ciel tombant' 2012  'Fantôme' 2015	

- 1. On s'intéresse ici à la récupération de données dans une relation.
- 1.a. Décrire ce que renvoie la requête ci-dessous :





**1.b.** Fournir une requête SQL permettant de récupérer le titre et la clé primaire de chaque film dont la date de sortie est strictement supérieure à 2020.



- 2. Cette question traite de la modification de relations.
- **2.a.** Dire s'il faut utiliser la requête 1 ou la requête 2 proposées ci-dessous pour modifier la date de naissance de Daniel Crog. Justifier votre réponse en expliquant pourquoi la requête refusée ne pourra pas fonctionner.



**2.b.** Expliquer si la relation **individu** peut accepter (ou pas) deux individus portant le même nom, le même prénom et la même date de naissance.



- 3. Cette question porte sur la notion de clés étrangères.
- **3.a.** Recopier sur votre copie les demandes ci-dessous, dans leur intégralité, et les compléter correctement pour qu'elles ajoutent dans la relation emploi les rôles de Daniel Crog en tant que James Bond dans le film nommé 'casino Impérial' puis dans le film 'Ciel tombant'.





**3.b.** On désire rajouter un nouvel emploi de Daniel Crog en tant que James Bond dans le film 'Docteur Yes'. Expliquer si l'on doit d'abord créer l'enregistrement du film dans la relation realisation ou si l'on doit d'abord créer le rôle dans la relation emploi.



- **4.** Cette question traite des jointures.
- **4.a.** Recopier sur votre copie la requête SQL ci-dessous, dans son intégralité, et la compléter de façon à ce qu'elle renvoie le nom de l'acteur, le titre du film et l'année de sortie du film, à partir de tous les enregistrements de la relation **emploi** pour lesquels la description de l'emploi est 'Acteur (James Bond)'.

```
SELECT ...
FROM emploi
JOIN individu ON ...
JOIN realisation ON ...
WHERE emploi.description = 'Acteur(James Bond)';
```

>



**4.b.** Fournir une requête SQL permettant de trouver toutes les descriptions des emplois de Denis Johnson (Denis est son prénom et Johnson est son nom).

On veillera à n'afficher que la description des emplois et non les films associés à ces emplois.



## D'après 2022, Métropole, J2

!!! exo D'après 2022, Métropole, J2"

#### ⟨ Texte

- 2 relations dans une base de données sur la musique
- 2 tables : \*\*`morceaux`\*\* et \*\*`interpretes`\*\*

On pourra utiliser les mots clés SQL suivants : SELECT, FROM, WHERE, JOIN, ON, INSERT, INTO, VALUES, UPDATE, SET, AND.

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

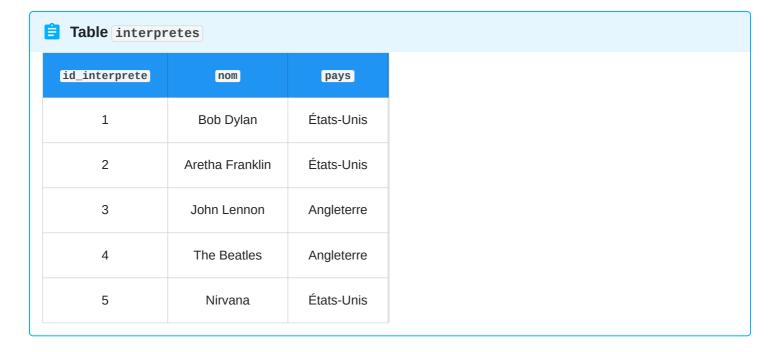
Un musicien souhaite créer une base de données relationnelle contenant ses morceaux et interprètes préférés. Pour cela il utilise le langage SQL.

Il crée une table morceaux qui contient entre autres attributs les titres des morceaux et leur année de sortie :

#### Table morceaux id\_morceau titre annee id\_interprete 1 Like a Rolling Stone 1965 1 2 2 Respect 1967 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

>

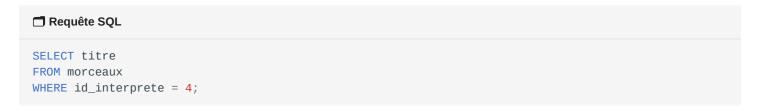
Il crée la table interpretes qui contient les interprètes et leur pays d'origine :

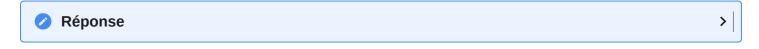


id\_morceau de la table morceaux et id\_interprete de la table interpretes sont des clés primaires.

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

1.a. Écrire le résultat de la requête suivante :





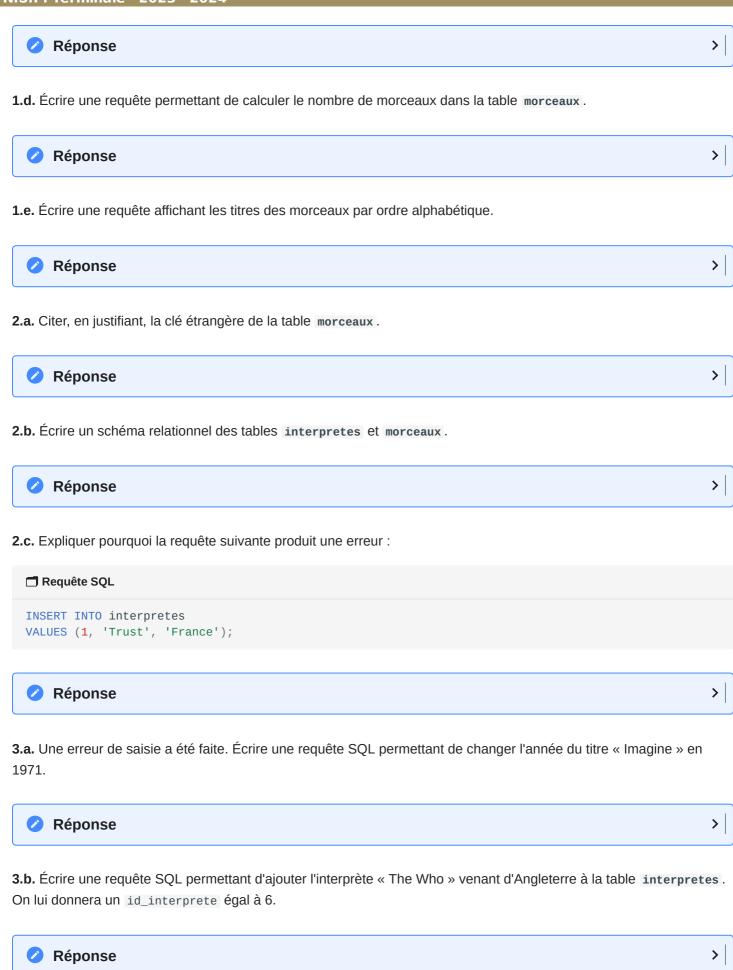
1.b. Écrire une requête permettant d'afficher les noms des interprètes originaires d'Angleterre.



1.c. Écrire le résultat de la requête suivante :

```
Requête SQL

SELECT titre, annee
FROM morceaux
ORDER BY annee;
```



**3.c.** 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.



4. Écrire une requête permettant de lister les titres des interprètes venant des États-Unis.

# 2. Métropole, Candidats libres, J2 2021

### 🗱 Exercice n°3 : Métropole, Candidats libres, J2 2021

- 2 relations dans une base de données sur un CDI
- 3 tables: Livres, Emprunts et Eleves

L'énoncé de cet exercice utilise les mots du langage SQL suivants :

```
SELECT FROM, WHERE, JOIN ON, INSERT INTO VALUES, UPDATE, SET, DELETE, COUNT, AND, OR.
```

On considère dans cet exercice une gestion simplifiée des emprunts des ouvrages d'un CDI. La base de données utilisée sera constituée de trois relations (ou tables) nommées Eleves, Livres et Emprunts selon le schéma relationnel suivant :

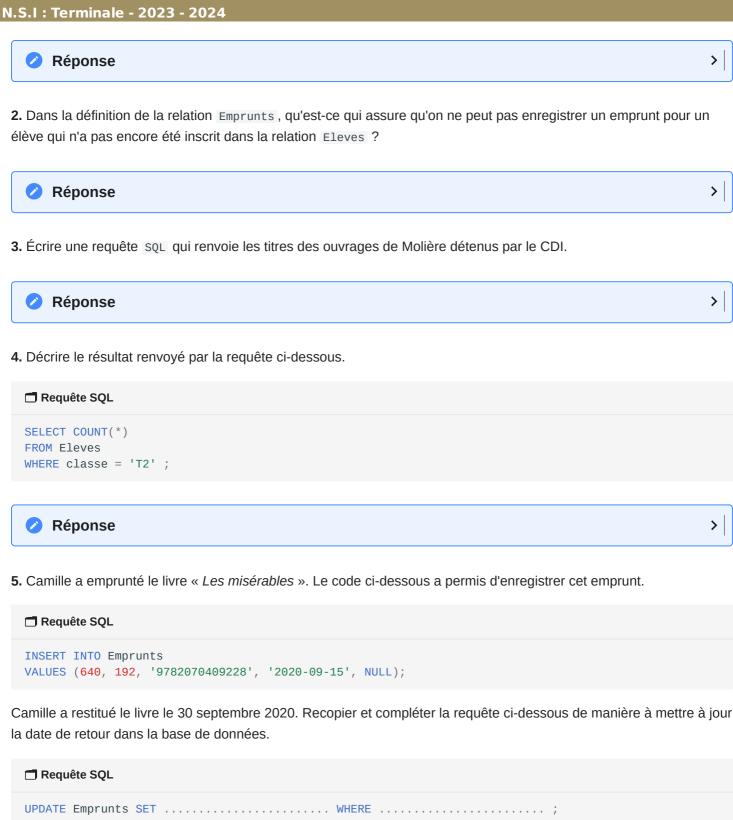
- Livres (<u>isbn (CHAR 13)</u>, titre (CHAR), auteur (CHAR))
- Emprunts (<u>idEmprunt (INT)</u>, #idEleve (INT), #isbn (CHAR 13), dateEmprunt (DATE), dateRetour (Date))
- **Eleves** (<u>idEleve (INT)</u>, nom (CHAR), prenom (CHAR), classe (CHAR))

Dans ce schéma relationnel, un attribut souligné indique qu'il s'agit d'une clé primaire.

Le symbole # devant un attribut indique qu'il s'agit d'une clé étrangère. Ainsi, l'attribut ideleve de la relation Emprunts est une clé étrangère qui fait référence à la clé primaire ideleve de la relation Eleves. De même l'attribut isbn de la relation Emprunts est une clé étrangère qui fait référence à la clé primaire isbn de la relationcompléter Livres.

**1.** Expliquer pourquoi le code SQL ci-dessous provoque une erreur.

```
INSERT INTO Eleves VALUES (128, 'Dupont', 'Jean', 'T1');
INSERT INTO Eleves VALUES (200, 'Dupont', 'Jean', 'T1');
INSERT INTO Eleves VALUES (128, 'Dubois', 'Jean', 'T2');
```





6. Décrire le résultat renvoyé par la requête ci-dessous.

```
☐ Requête SQL

SELECT DISTINCT nom, prenom
FROM Eleves, Emprunts
```

WHERE Eleves.idEleve = Emprunts.idEleve
AND Eleves.classe = 'T2';



>

>

**7.** Écrire une requête SQL qui permet de lister les noms et prénoms des élèves qui ont emprunté le livre « *Les misérables* ».

