

Gestions de musiciens

D'après 2023, sujet zéro A

Auteurs : Denis Quenton, Nicolas Revéret

On pourra utiliser les mots clés SQL suivants :

AND , SELECT , FROM , WHERE , JOIN , INSERT INTO , VALUES , COUNT , ORDER BY , OR , ON , SET , UPDATE .

On étudie une base de données permettant la gestion de l'organisation d'un festival de musique de jazz, dont voici le schéma relationnel comportant trois relations :

- la relation **groupes**("id_groupe", "nom", "style", "nb_pers")
- la relation **musiciens**("id_musicien", "nom", "prenom", "instru", "#id_groupe")
- la relation **concerts**("id_concert", "scene", "heure_debut", "heure_fin", "#id_groupe")

Dans ce schéma relationnel :

- les clés primaires sont soulignées ;
- les clés étrangères sont précédées d'un #.

Ainsi `concerts.id_groupe` est une clé étrangère faisant référence à `groupes.id_groupe` .

Voici un extrait des tables `groupes` , `musiciens` et `concerts` :

"Extrait de groupes "

<u>id_groupe</u>	<u>nom</u>	<u>style</u>	<u>nb_pers</u>
12	'Weather Report'	'Jazz Fusion'	5
25	'The 3 Sounds'	'Soul Jazz'	4
87	'Return to Forever'	'Jazz Fusion'	8
96	'The Jazz Messenger'	'Hard Bop'	3

"Extrait de musiciens "

id_musicien	nom	prenom	instru	id_groupe
12	'Garrett'	'Kenny'	'saxophone alto'	96
13	'Garrett'	'Kenny'	'flute'	25
58	'Corea'	'Chick'	'piano'	87
97	'Clarke'	'Stanley'	'basse'	87

"Extrait de concerts "

id_concert	scene	heure_debut	heure_fin	id_groupe
10	1	'20 h 00'	'20 h 45'	12
24	2	'20 h 00'	'20 h 45'	35
36	1	'21 h 00'	'22 h 00'	96
45	3	'18 h 00'	'18 h 30'	87

1. Citer les attributs de la table `groupes` .

"Réponse"

Les attributs de la table `groupes` sont : `id_groupe` , `nom` , `style` **et** `nb_pers` .

2. Justifier que l'attribut `nom` de la table `musiciens` ne peut pas être une clé primaire.

"Réponse"

Une clé primaire doit être unique. Le nom `'Garrett'` apparaît plusieurs fois, donc le nom ne peut être une clé primaire.

On notera que le couple `(nom, prenom)` n'est pas non plus une clé primaire de cette table pour la même raison.

De la même façon, le triplet (nom, prenom, intru) ne peut pas non plus être utilisé comme clé primaire : le même musicien peut tout à fait jouer du même instrument dans deux groupes différents.

3. En s'appuyant uniquement sur l'extrait des tables fourni ci-dessus écrire ce que renvoie la requête :

```
SELECT nom
FROM groupes
WHERE style = 'Jazz Fusion';
```

✓ "Réponse"

La requête renvoie : 'Weather Report' et 'Return to Forever' .

4. Le concert dont l' id_concert est 36 finira à 22 h 30 au lieu de 22 h 00.

Recopier sur la copie et compléter la requête SQL ci-dessous permettant de mettre à jour la relation concerts pour modifier l'horaire de fin de ce concert.

```
UPDATE concerts
SET ...
WHERE ... ;
```

✓ "Réponse"

```
UPDATE concerts
SET heure_fin = '22 h 30'
WHERE id_concert = 36;
```

5. Donner une seule requête SQL permettant de récupérer le nom de tous les groupes qui jouent sur la scène 1.

✓ "Réponse"

```
SELECT groupes.nom FROM groupes
JOIN concerts
ON concerts.id_groupe = groupes.id_groupe
WHERE concerts.scene = 1;
```

6. Fournir une seule requête SQL permettant d'ajouter dans la relation `groupes` le groupe 'Smooth Jazz Fourplay', de style 'Free Jazz', composé de 4 membres. Ce groupe aura un `id_groupe` de 15.

✓ "Réponse"

```
INSERT INTO groupes (id_groupe, nom, style, nb_pers)
VALUES (15, 'Smooth Jazz Fourplay', 'Free Jazz', 4);
```

Les données sont ensuite récupérées pour être analysées par la société qui produit les festivals de musique. Pour ce faire, elle utilise la programmation en Python afin d'effectuer certaines opérations plus complexes.

Elle stocke les données relatives aux musiciens sous forme d'un tableau de dictionnaires dans laquelle a été ajouté le nombre de concerts effectués par chaque musicien :

```
>>> print(musiciens)
[{'id_musicien': 12, 'nom': 'Garrett', 'prenom': 'Kenny',
  'instru': 'saxophone alto', 'id_groupe' : 96, 'nb_concerts': 5},
 {'id_musicien': 13, 'nom': 'Garrett', 'prenom': 'Kenny',
  'instru': 'flute', 'id_groupe' : 25, 'nb_concerts': 9},
 {'id_musicien': 58, 'nom': 'Corea', 'prenom': 'Chick',
  'instru': 'piano', 'id_groupe' : 87, 'nb_concerts': 4},
 {'id_musicien': 97, 'nom': 'Clarke', 'prenom': 'Stanley',
  'instru': 'basse', 'id_groupe' : 87, 'nb_concerts': 4},
 ...
]
```

7. Écrire la fonction `recherche_nom` ayant pour unique paramètre un tableau de dictionnaires (comme `musiciens` présenté précédemment) renvoyant une liste contenant le nom de tous les musiciens ayant participé à au moins 4 concerts.

✓ "Réponse"

```
def recherche_nom(musiciens):  
    resultat = []  
    for musicien in musiciens:  
        if musicien['nb_concerts'] >= 4:  
            resultat.append(musicien['nom'])  
    return resultat
```