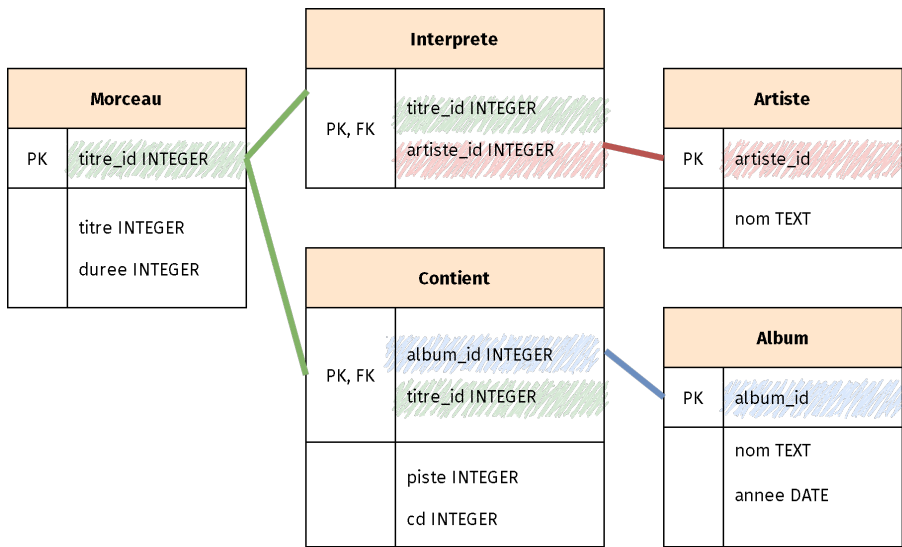


Un ami vous suggère d'utiliser le schéma suivant :

```
Morceau(titre_id, titre, duree, artiste_id, album, piste, cd, annee)
Artiste(artiste_id, nom)
```

4. Expliquer pourquoi cette représentation ne permet toujours pas de gérer les morceaux fait par deux artistes différents.

Finalement, vous arrivez au schéma suivant :



5. Compléter les tables à l'aide des informations déjà disponibles. Un des morceaux n'a pas été intégré, inutile de l'y remettre. Si les noms dépassent, mettre uniquement le début.

titre_id	titre	duree
519	Astronomy	
1219	Astronomy	
316	Stone Cold Crazy	136
1319	Stone Cold Crazy	139
1298		270
1570		
401		
1125	The Outlaw Torn	589
599	The Outlaw Torn	599

titre_id	artiste_id
519	25
1219	154
1319	
1298	154
1570	154
1570	318
1125	154
1591	154
1591	318
316	79
401	79
401	108

artiste_id	nom
	Metallica
318	
25	
79	
108	

6. Comment appelle-t-on les clefs primaires de certaines tables apparaissant dans certaines tables, comme dans Interprete ?

7. Expliquer pourquoi le couple (titre_id, artiste_id) peut servir de clef primaire à Interprete.

8. Traduire en langage naturel les requêtes suivantes :

Code SQL

```
SELECT titre, duree FROM Morceau
WHERE duree > 600 ORDER BY duree DESC;
```


Code SQL

```
SELECT cd, piste, titre FROM Morceau
JOIN Contient ON Contient.titre_id = Morceau.titre_id
JOIN Album ON Contient.album_id = Album.album_id WHERE nom = "Garage Inc."
ORDER BY cd, piste;
```

[illegible]

9. Donner la requête SQL permettant d'obtenir le nom de l'artiste dont l'identifiant est 200.

[illegible]

10. Donner la requête SQL permettant d'obtenir le nom de tous les albums sortis entre 1999 et 2010.

[illegible]

11. Donner la requête SQL permettant d'obtenir le titre et la durée de tous les morceaux, triés par ordre décroissant de durée, de tous les morceaux de l'artiste dont l'identifiant est 200.

[illegible]

12. Les stars étant capricieuses, certaines veulent changer de nom. Donner la requête permettant à 'Maître Gims' de devenir 'Gims' dans la table des artistes.

On rajoute maintenant les tables pour les utilisateurs :

Utilisateur(util_id INTEGER , nom TEXT, e-mail TEXT, adresse TEXT)

Ecoute(id_ecoute INTEGER, titre_id TEXT, util_id INTEGER, date DATE)

13. Expliquer pourquoi le couple (titre_id ,util_id) ne peut pas être une clef primaire.

14. Donner la requête SQL permettant d'ajouter l'utilisateur numéro 2179, qui s'appelle Bob VHS, dont l'email est bob.vhs@hotmail.com et qui habite à New York.

15. Traduire le requête suivante en langage naturel :

Code SQL

```
SELECT COUNT(DISTINCT titre) FROM Morceau
JOIN Ecoute ON Morceau.titre_id = Ecoute.titre_id
WHERE date = "2020-12-12";
```


Files et Piles

Pour les exercices comportant des files et des piles, on utilisera les interfaces suivantes, qu'on suppose programmées en PYTHON :

Structures

FILE :

- *file_vide()* renvoie une file vide;
- *enfiler(file,valeur)* enfila la valeur en fin de file;
- *defiler(file)* enlève la valeur en début de file et la renvoie;
- *est_vide(file)* indique si la file est vide ou non en renvoyant un booléen.

PILE :

- *pile_vide()* renvoie une pile vide;
- *empiler(pile,valeur)* empile la valeur sur la pile;
- *depiler(pile)* renvoie la valeur sur la pile et l'enlève de la pile;
- *est_vide(pile)* indique si la pile est vide ou non en renvoyant un booléen.

Exercice 2

Écrire en PYTHON une fonction `renverse_file` qui

- en entrée prend une file;
- ne renvoie rien mais **utilise une pile** pour renverser la file.

Exemple d'utilisation

```
>>> print(F)
2 -> 3 -> 5 -> 1
>>> renverse_file(F)
>>> print(F)
1 -> 5 -> 3 -> 2
```


3. Dans cette question on a $p = 10 : 20$.

Que se passe t-il lors de l'appel `mystere(p , p)`?

[illegible]

Exercise 4

Écrire en PYTHON une fonction `max_file` qui

- en entrée prend une file **non vide** composée d'**int positifs**;
- renvoie le maximum de cette file. Attention la file doit être remise dans l'état initial et **aucune autre structure de données** (pile, file, liste) ne doit être utilisée.

Exemple d'utilisation

```
>>> print(F)
>>> 2 -> 3 -> 5 -> 1
>>> max_file(F)
>>> 5
>>> print(F)
>>> 2 -> 3 -> 5 -> 1
```

This image shows a full page of blank graph paper. The grid consists of small squares formed by thin, light blue horizontal and vertical lines. There are 20 columns and 20 rows of squares, creating a uniform pattern across the entire page. The background is white, and there are no margins or other markings present.