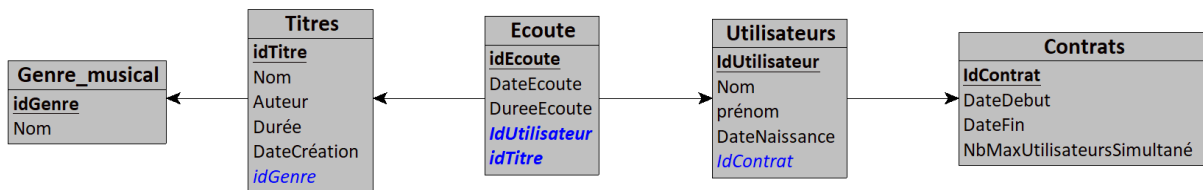


Jalon 2 : projet base de donnée spotifi

Lucas Vade grelet
Matteo Raffin

1 Schéma Relationnel



2 Fonctionnalités Manquantes

Dans le jalon 1, un utilisateur ne pouvait écouter un titre qu’une seule fois. Pour résoudre ce problème, nous avons ajouté l’attribut ‘idEcoute’ dans la table ‘Ecoute’.

Cet ajout permet de distinguer chaque écoute d’un titre par un utilisateur. Ainsi, avec cette modification, un utilisateur peut écouter plusieurs fois le même titre, avec des durées identiques ou différentes. Cela permet également un suivi de l’historique des écoutes d’un utilisateur.

Ce changement améliore la gestion des écoutes répétées et garantit une meilleure flexibilité dans le suivi des actions des utilisateurs.

3 Bugs Référencés

Aucun bug n’a été trouvé lors de l’analyse et du test des différentes fonctionnalités implémentées.

4 Organisation du Travail

Le travail a été réparti comme suit :

- **Lucas** : Rédaction du document, réalisation des requêtes 1, 3, 5, 6, 7, 8, 9, 10.
- **Matteo** : Réalisation des requêtes 2 et 4, remplissage de la base de données, identification des fonctionnalités manquantes.

5 Requetes SQL et Explications

Requête 1 : Combien de titres et de genres sont disponibles dans la base ?

Cette requête calcule le nombre total de titres et de genres dans la base de données en utilisant une union entre les deux informations.

```
SELECT COUNT(*) AS NbTitres FROM Titres
UNION
SELECT COUNT(*) AS NbGenres FROM GenreMusical;
```

Requête 2 : Combien de Titres M. Dupond a-t-il écouté ?

On sélectionne les tables 'Ecoule' et 'Utilisateurs', on réduit la sélection au nom "Dupond", puis on compte le nombre de résultats avec 'COUNT'.

```
SELECT COUNT(*) AS NbrTitreEcoule
FROM Ecoule
INNER JOIN Utilisateurs ON Ecoule.idUtilisateur = Utilisateurs.
    idUtilisateur
WHERE Utilisateurs.Nom = "Dupond";
```

Ce qui nous donne :

NbrTitreEcouté
5

Requête 3 : Quel est le genre musical le plus écouté par M. Dupond ?

Cette requête permet de sélectionner le genre musical le plus écouté par M. Dupond.

```
SELECT GenreMusical.Nom
FROM Utilisateurs
INNER JOIN (GenreMusical
            INNER JOIN (Ecoule
                        INNER JOIN Titres ON Ecoule.idTitre = Titres.
                                idTitre)
            ON GenreMusical.idGenre = Titres.idGenre)
ON Utilisateurs.idUtilisateur = Ecoule.idUtilisateur
WHERE Utilisateurs.Nom = "Dupond"
```

On obtient alors le tableau suivant avec tous les genres musicaux écoutés par M. Dupond :

Nom
Pop
Electro
Country
Electro
Electro

Ce qui nous donne, après un ‘GROUP BY’ et un ‘COUNT’, le genre le plus écouté :

Nom	NbrEcoule
Electro	3

Requête 4 : Quelle est la durée moyenne d'écoute des morceaux ?

Cette requête calcule la durée moyenne d'écoute des morceaux.

On sélectionne d'abord tous les titres de la table 'Titres' avec la requête suivante :

```
SELECT T.Nom
FROM Ecoute AS E
INNER JOIN Titres AS T ON E.idTitre = T.idTitre;
```

Ce qui nous donne :

Nom
Get Lucky
I'm still standing
In The End
...

Ensuite, nous rajoutons le calcul de la moyenne avec la fonction $AVG(E.DuréeEcoute / T.Durée * 100)$ AS Ratio. La requête devient :

```
SELECT T.Nom, AVG(E.DureeEcoute / T.Duree * 100) AS Ratio
FROM Ecoute AS E
INNER JOIN Titres AS T ON E.idTitre = T.idTitre
GROUP BY T.Nom;
```

Ce qui nous donne alors les résultats suivants :

Nom	Ratio
Get Lucky	72.62
I'm still standing	99.63
In The End	85.42
...	...

Requête 5 : Nombre de personnes maximum liées à un même contrat ayant écouté de la musique le même jour ?

Sous-requête 1 : On commence par calculer le nombre d'utilisateurs ayant écouté de la musique pour chaque combinaison de date d'écoute et contrat.

```
SELECT Ecoute.DateEcoute, Utilisateurs.idContrat, COUNT(Utilisateurs.
    idUtilisateur) AS NombrePersonnes
FROM Utilisateurs
INNER JOIN Ecoute ON Utilisateurs.idUtilisateur = Ecoute.idUtilisateur
GROUP BY Ecoute.DateEcoute, Utilisateurs.idContrat;
```

Ce qui nous donne le tableau suivant :

DateEcoute	idContrat	NombrePersonnes
12/10/2024	1	1
12/10/2024	4	1
14/10/2024	3	2
14/10/2024	2	3
...

Sous-requête 2 : Nous cherchons le nombre maximum de personnes ayant écouté de la musique le même jour.

```
SELECT MAX(SousRequete.NombrePersonnes)
FROM (
    -- Sous-requete 1
) AS SousRequete;
```

Ce qui nous donne le résultat suivant :

Max
3

Nous utilisons ensuite ce résultat dans la condition de la requête principale pour récupérer les contrats ayant eu le plus grand nombre de personnes.

```
SELECT MaxContrat.DateEcoute, MaxContrat.idContrat, MaxContrat.
    NombrePersonnes
FROM (
    -- sous-requete 1
) AS MaxContrat
WHERE MaxContrat.NombrePersonnes = (
    -- sous-requete 2
    WHERE SousRequete.idContrat = MaxContrat.idContrat
)
ORDER BY MaxContrat.idContrat;
```

Ce qui nous donne alors les résultats suivants :

DateEcouté	idContrat	NombrePersonnes
16/10/2024	4	2
14/10/2024	4	2
14/10/2024	3	2
14/10/2024	2	3
13/10/2024	1	3

Requête 6 : Nombre de fois où des personnes liées à un même contrat ont écouté de la musique simultanément le 14/10/2024 ?

Sous-requête 1 : Vérification des utilisateurs ayant écouté de la musique en même temps.

```
SELECT 1
FROM Ecoute AS E2
INNER JOIN Utilisateurs AS U2 ON E2.idUtilisateur = U2.idUtilisateur
WHERE E2.DateEcouté = #10/14/2024#
AND U2.idContrat = U1.idContrat
AND E2.idUtilisateur <> E1.idUtilisateur
AND (
    (E1.DateEcouté <= DATEADD("s", TIMEVALUE(E2.DuréeEcouté), E2.
        DateEcouté))
    AND (DATEADD("s", TIMEVALUE(E1.DuréeEcouté), E1.DateEcouté) >=
        E2.DateEcouté)
)
```

Sous-requête 2 : Sélection des utilisateurs et contrats.

```
SELECT DISTINCT E1.idUtilisateur, U1.idContrat
FROM Ecoute AS E1
INNER JOIN Utilisateurs AS U1 ON E1.idUtilisateur = U1.idUtilisateur
WHERE E1.DateEcouté = #10/14/2024#;
```

On obtient les résultats suivants pour les utilisateurs et leurs contrats ayant écouté simultanément :

idUtilisateur	idContrat
1	1
2	2
3	2

Enfin, la sélection finale, en comptant les utilisateurs dans chaque contrat, donne :

```
SELECT idContrat, COUNT(*) AS NombrePersonnes
FROM (
    sous-requete 2
    AND EXISTS (
        sous-requete 1
    )
) AS SousRequete
GROUP BY idContrat;
```

Ce qui nous donne les résultats suivants :

idContrat	NombrePersonnes
1	2
2	3
3	2
4	2

Requête 7 : Playlist de M. Dupond

Cette requête génère une playlist pour M. Dupond en excluant les titres du genre qu'il n'a pas écouté en entier.

```
SELECT Titres.Nom
FROM Titres
WHERE Titres.idGenre = (
    -- Sous-requete pour le genre le plus ecoute
)
AND Titres.idTitre NOT IN (
    -- Titres ecoutes par M. Dupond
);
```

Ce qui nous donne le résultat suivant :

Nom
Electric Feel

Requête 8 : Playlist basée sur le genre le plus écouté de M. Dupond

Cette requête propose une playlist en excluant M. Dupond.
on reprend la requete 3, pour trouver le genre le plus ecouté par M. Dupond

```
SELECT DISTINCT Titres.Nom
FROM Utilisateurs INNER JOIN (Titres INNER JOIN Ecoute ON Titres.
    idTitre = Ecoute.idTitre)
ON Utilisateurs.idUtilisateur = Ecoute.idUtilisateur
WHERE Titres.idGenre = (
    -- Requete pour le genre le plus ecoute
)
AND Utilisateurs.Nom <> "Dupond";
```

Ce qui nous donne le tableau suivant :

Nom
Memories
Get Lucky

Requête 9 : Écrire une requête qui crée une playlist pour M. Dupond en se basant sur les titres les plus écoutés par tous les utilisateurs ayant un écart d'âge de plus ou moins 5 ans

On sélectionne d'abord toutes les musiques :

```
SELECT Titres.Nom
FROM Titres INNER JOIN (Utilisateurs INNER JOIN Ecoule ON Utilisateurs.
    idUtilisateur = Ecoule.idUtilisateur)
ON Titres.idTitre = Ecoule.idTitre;
```

Ce qui nous donne la liste suivante des titres :

Nom
Memories
I'm still standing
Get Lucky
...

Ensuite, on filtre les musiques en fonction de l'âge de l'utilisateur qui l'écoute. On filtre les utilisateurs dont l'année de naissance est dans un écart de plus ou moins 5 ans avec M. Dupond, on distinct :

```
SELECT DISTINCT Titres.Nom
FROM Titres
INNER JOIN (Utilisateurs INNER JOIN Ecoule ON Utilisateurs.
    idUtilisateur = Ecoule.idUtilisateur)
ON Titres.idTitre = Ecoule.idTitre
WHERE Utilisateurs.DateNaissance BETWEEN
    DATEADD("yyyy", -5, (SELECT DateNaissance FROM Utilisateurs WHERE Nom
        = "Dupond"))
AND DATEADD("yyyy", 5, (SELECT DateNaissance FROM Utilisateurs WHERE
    Nom = "Dupond"));
```

On obtient donc les titres suivants :

Nom
Memories
I'm still standing
Get Lucky
Take Me Home, Country Roads

Requête 10 : Playlist basée sur les genres écoutés partiellement par M. Dupond

Cette sous-requête génère une playlist pour M. Dupond en excluant les genres musicaux des titres qu'il n'a pas écouté en entier.

```
SELECT DISTINCT Titres.idGenre
FROM Utilisateurs
INNER JOIN (Titres INNER JOIN Ecoute ON Titres.idTitre = Ecoute.idTitre
) ON Utilisateurs.idUtilisateur = Ecoute.idUtilisateur
WHERE (((Utilisateurs.Nom)="Dupond" And [Ecoute].[DureeEcoute]<[Titres].[Duree]))
```

Exemple de résultats pour les genres écoutés partiellement :

idGenre
4
7

Et la requête finale :

```
SELECT Titres.Nom, COUNT(Ecoute.idTitre) AS NombreEcoutes
FROM Titres INNER JOIN Ecoute ON Titres.idTitre = Ecoute.idTitre
WHERE Titres.idGenre NOT IN (
    -- Genres ecoute partiellement par M. Dupond
)
```

Ce qui nous donne :

Nom
Take on Me
Smooth operator
I'm still standing
In the end
Le bilan