

TD3 : COMMANDES SQL

1. Genres d'albums disponibles (liste unique).

```
SELECT DISTINCT genre
FROM albums;
```
2. Titres des pistes et leur popularité avec popularite ≥ 80 , trié sur popularité décroissante.

```
SELECT pistes.titre, pistes.popularite FROM pistes
WHERE popularite > 80
ORDER BY popularite DESC
```
3. Lister les 10 utilisateurs inscrits les plus récents (nom, pays, date).

```
SELECT nom, pays, date_inscription
FROM utilisateurs
ORDER BY date_inscription DESC
LIMIT 10;
```
4. Nombre d'écoutes totale sur la période 2025-07-01 → 2025-07-31.

```
SELECT COUNT(*) AS nb_ecoutes
FROM ecoutes AS e
WHERE horodatage BETWEEN '2025-07-01' AND '2025-07-31'
```
5. Nombre d'écoutes par appareil (device)

```
SELECT device, COUNT(*) AS nb_ecoutes
FROM ecoutes
GROUP BY device
ORDER BY nb_ecoutes DESC;
```
6. Nombre d'utilisateurs par pays, trié décroissant (afficher uniquement les pays avec ≥ 3 utilisateurs).

```
SELECT pays, COUNT(*) AS nb_utilisateurs
FROM utilisateurs
GROUP BY pays
HAVING COUNT(*) >= 3
ORDER BY nb_utilisateurs DESC;
```
7. Durée moyenne des pistes par genre d'album (JOIN albums ↔ pistes), tri décroissant.

```
SELECT a.genre, AVG(p.duree_sec) AS duree_moyenne_sec
FROM pistes p
JOIN albums a ON p.album_id = a.id
GROUP BY a.genre ORDER BY duree_moyenne_sec DESC;
```
8. Top 20 des pistes les plus populaires (titre piste, album, artiste, popularite).

```
SELECT p.titre, a.titre AS album, ar.nom AS artiste, p.popularite
FROM pistes p
JOIN albums a ON p.album_id = a.id
JOIN artistes ar ON a.artiste_id = ar.id
ORDER BY p.popularite DESC
LIMIT 20;
```
9. Albums sortis entre 2023-01-01 et 2025-01-01 (titre album, artiste, date_sortie).

```
SELECT a.titre, ar.nom AS artiste, a.date_sortie
```

```
FROM albums a
JOIN artistes ar ON a.artiste_id = ar.id
WHERE a.date_sortie BETWEEN '2023-01-01' AND '2025-01-01'
ORDER BY a.date_sortie;
```

10. Utilisateurs sans playlist publique (nom + nombre total de playlists non publiques).

```
SELECT u.nom, COUNT(*) AS nbr
FROM utilisateurs u
JOIN playlists p ON u.id = p.utilisateur_id
WHERE p.publique='0'
GROUP BY u.id
```

11. Nombre de pistes par playlist (titre, nombre de pistes), trie décroissant

```
SELECT COUNT(pl.piste_id) AS nbr_pistes
FROM playlists AS p JOIN playlist_piste AS pl ON p.id=pl.playlist_id
GROUP BY p.id
ORDER BY nbr_pistes DESC
```

12. Nombre moyen de piste par playlist

```
SELECT AVG(nb_pistes) AS nbr_moyenne
FROM (
    SELECT playlist_id, COUNT(*) AS nb_pistes
    FROM playlist_piste
    GROUP BY playlist_id
);
```

13. Top 10 artistes trié par nombre total d'écoute décroissant.

```
SELECT ar.nom AS artiste, COUNT(e.id) AS nb_ecoutes
FROM ecoutes e
JOIN pistes p ON e.piste_id = p.id
JOIN albums a ON p.album_id = a.id
JOIN artistes ar ON a.artiste_id = ar.id
GROUP BY ar.id
ORDER BY nb_ecoutes DESC LIMIT 10;
```

14. Utilisateurs « très actifs » : ceux avec ≥ 260 écoutes au total (afficher le count).

```
SELECT u.nom, COUNT(e.id) AS nb_ecoutes
FROM utilisateurs u
JOIN ecoutes e ON u.id = e.utilisateur_id
GROUP BY u.id
HAVING COUNT(e.id) >= 260 ORDER BY nb_ecoutes DESC;
```

15. Lister les genres et leur durée moyenne, dont leur durée moyenne des pistes est supérieure à la moyenne globale des durées.

```
SELECT a.genre, ROUND(AVG(p.duree_sec), 1) AS duree_moy
FROM albums a
JOIN pistes p ON p.album_id = a.id
GROUP BY a.genre
HAVING AVG(p.duree_sec) > (SELECT AVG(duree_sec) FROM pistes);
```