

L2 – Bases de données

Projet The Wave

Etape 3

Clément GAUDET / Maxime JAILLARD

08/01/21



Sommaire

Manuel utilisateur	3
Architecture du site	4
Arborescence des pages	4
Requêtes SQL majeures	4
Gestion des sessions	5
Fonctionnalités non-implémentées	5



Manuel utilisateur

Lors de la connexion de l'utilisateur au site TheWave, celui-ci est redirigé vers la page d'authentification s'il n'est pas déjà connecté. Le site demande alors de rentrer l'identifiant et le mot de passe du compte. Si les informations entrées sont correctes, l'utilisateur est envoyé sur la page d'accueil.

Depuis la page d'accueil, on peut consulter les morceaux les plus écoutés de la semaine, les groupes les plus suivis et les derniers albums sortis. Chaque texte écrit en bleu permet de consulter la page associée en cliquant dessus. On accède alors à la page dédiée au morceau, au groupe ou à l'album sélectionné.

- La page de groupe permet à l'utilisateur de consulter les informations (nationalité, date de création, genre, nationalité...) ainsi que la liste des membres du groupe. Elle contient aussi la liste des albums et morceaux publiés par le groupe afin que l'utilisateur puisse naviguer directement vers leur page.
- La page des morceaux contient les informations disponibles sur le groupe qui l'a publié, le genre, l'album d'origine (s'il y en a un) et la durée de l'audio. On peut aussi y trouver la liste des artistes ayant participé au morceau.
- La page des albums montre la couverture et leur description. La liste des morceaux contenus dans l'album permet de consulter leur page respective.

Depuis n'importe quelle page, on peut également accéder à notre profil à l'aide du bouton portant le pseudo du compte, situé en haut à droite dans l'en-tête de page. Cette page répertorie les informations personnelles de l'utilisateur connecté. Il peut aussi y consulter les morceaux les plus écoutés, son historique d'écoute et les dernières activités des comptes suivis.

Pour trouver une page précise sur le site, on peut utiliser la page « Recherche » accessible depuis n'importe quelle page en cliquant sur le bouton correspondant dans l'en-tête. On est redirigé vers une zone à remplir avec les informations voulues.

- On sélectionne tout d'abord le type d'entrées recherchées.
- La case Titre contient les mots-clés à rechercher dans le titre du morceau, de l'album, de la playlist ou le nom du groupe. Si la case reste vide, aucun titre ne sera éliminé lors de la recherche.
- La case Genre contient les mots-clés à rechercher dans le genre des morceaux, de l'album, de la playlist ou du groupe. Si la case reste vide, aucun genre ne sera éliminé lors de la recherche.
- La case Date contient l'année à rechercher dans une liste d'album. Si elle reste vide, aucune contrainte d'année de parution ne sera appliquée. (Remarque : l'année n'est utilisée que pour rechercher des albums)
- La case Artiste contient les noms ou prénoms à rechercher dans les participants aux morceaux, albums, playlists ou dans les membres d'un groupe. Si elle reste vide, tous les artistes seront considérés.

En plus des pages déjà présentées, on peut rechercher les playlists créées par d'autres utilisateurs. Sur ces page, on trouve... (A COMPLETER)



Architecture du site

Arborescence des pages

La page d'authentification est le point d'entrée de l'utilisateur sur le site lors de sa première connexion. Bien que l'en-tête du site soit toujours visible, aucune page n'est accessible à l'utilisateur tant qu'aucune session n'est en cours. Après son authentification, la page d'accueil devient le nouveau point d'entrée de l'utilisateur sur le site.

(A COMPLETER)

Requêtes SQL majeures

Pour la partie recherche, on utilise la méthode de comparaison ILIKE qui ne tient pas compte de la casse des chaînes de caractères. Les requêtes SQL de la page de recherche sont :

➤ Groupes :

```
SELECT DISTINCT idg, nomg, datecrea, nationg, genre
FROM groupe NATURAL JOIN membre NATURAL JOIN artiste
WHERE nomg ILIKE '%' || :nomg || '%'
AND genre ILIKE '%' || :genre || '%'
AND ((noma ILIKE '%' || :artiste || '%' OR prenom ILIKE '%' || :artiste || '%')
OR (:artiste ILIKE '%' || noma || '%' AND :artiste ILIKE '%' || prenom || '%'))
ORDER BY nomg;
```

➤ Morceaux :

```
SELECT DISTINCT morceau.idmo, titrem, duree, genre, nomg
FROM morceau
LEFT JOIN groupe ON groupe.idg = morceau.idg
LEFT JOIN participe ON morceau.idmo = participe.idmo
LEFT JOIN artiste ON participe.ida = artiste.ida
WHERE titrem ILIKE '%' || :titre || '%'
AND genre ILIKE '%' || :genre || '%'
AND ((noma ILIKE '%' || :artiste || '%' OR prenom ILIKE '%' || :artiste || '%')
OR (:artiste ILIKE '%' || noma || '%' AND :artiste ILIKE '%' || prenom || '%')
OR (noma IS NULL AND prenom IS NULL AND :artiste = ''))
ORDER BY titrem;
```

➤ Albums :

Si une date est mentionnée dans la recherche.

```
SELECT idal, titrea, dateparu, couv, genre, nomg
FROM (
```

```
    SELECT DISTINCT album.idal, titrea, dateparu, couv, genre, nomg,
    dateparu - CAST(FLOOR(:date*365.24) AS integer) AS depuis
    FROM album
    NATURAL JOIN albumcontient
    NATURAL JOIN morceau
    LEFT JOIN groupe ON groupe.idg = album.idg
    LEFT JOIN participe ON morceau.idmo = participe.idmo
    LEFT JOIN artiste ON participe.ida = artiste.ida
    WHERE titrea ILIKE '%' || :titre || '%'
    AND genre ILIKE '%' || :genre || '%'
    AND ((noma ILIKE '%' || :artiste || '%' OR prenom ILIKE '%' || :artiste || '%')
    OR (:artiste ILIKE '%' || noma || '%' AND :artiste ILIKE '%' || prenom || '%'))
```



```
OR (noma IS NULL AND prenom IS NULL AND :artiste = ''))  
ORDER BY depuis DESC
```

```
) AS tab1  
WHERE depuis < '0001-01-01' AND depuis > '0010-01-01 BC';
```

Si une date n'est pas mentionnée dans la recherche.

```
SELECT DISTINCT idal, titrea, dateparu, couv, genre, nomg  
FROM album  
NATURAL JOIN albumcontient  
NATURAL JOIN morceau  
LEFT JOIN groupe ON groupe.idg = album.idg  
LEFT JOIN participe ON morceau.idmo = participe.idmo  
LEFT JOIN artiste ON participe.ida = artiste.ida  
WHERE titrea ILIKE '%' || :titre || '%'  
AND genre ILIKE '%' || :genre || '%'  
AND ((noma ILIKE '%' || :artiste || '%' OR prenom ILIKE '%' || :artiste || '%')  
OR (:artiste ILIKE '%' || noma || '%' AND :artiste ILIKE '%' || prenom || '%')  
OR (noma IS NULL AND prenom IS NULL AND :artiste = ''))  
ORDER BY dateparu DESC;
```

➤ Playlists :

```
SELECT DISTINCT idp, titre, pseudo  
FROM playlist  
NATURAL JOIN playlistcontient  
NATURAL JOIN morceau  
LEFT JOIN groupe ON groupe.idg = morceau.idg  
LEFT JOIN participe ON morceau.idmo = participe.idmo  
LEFT JOIN artiste ON participe.ida = artiste.ida  
WHERE titre ILIKE '%' || :titre || '%'  
AND genre ILIKE '%' || :genre || '%'  
AND ((noma ILIKE '%' || :artiste || '%' OR prenom ILIKE '%' || :artiste || '%')  
OR (:artiste ILIKE '%' || noma || '%' AND :artiste ILIKE '%' || prenom || '%')  
OR (noma IS NULL AND prenom IS NULL AND :artiste = ''))  
ORDER BY titre DESC;
```

De manière générale, en utilisant la méthode ILIKE plutôt qu'une égalité entre chaînes de caractères, on cherche une correspondance partielle entre les mots-clés rentrés par l'utilisateur et les caractéristiques des entrées recherchées. Par exemple, pour un album, entrer « Olzon anette » dans le champ Artiste revient à rechercher tous les albums auxquels un artiste ayant participé porte un nom et un prénom contenus dans cette chaîne. S'il existe des artistes nommés Olzon Anette, Anette Olzon, Olzon Anette Olzon...etc, tous leurs albums seront affichés dans les résultats de la recherche.

Gestion des sessions

(A COMPLETER)

Fonctionnalités non-implémentées

(A COMPLETER)