

Projet : un gestionnaire de listes de musique

UCBL - Département Informatique de Lyon 1 – BDW - automne 2022

- Ce projet est évalué par plusieurs notes : avancement intermédiaire, soutenance et réalisation.
- Cette UE est avant tout une introduction aux **bases de données**. La non-utilisation d'une BD relationnelle dans le projet peut se voir attribuer la note 0.
- Le rendu du projet et la soutenance ont lieu lors de la dernière séance de TP. Tout rendu ne respectant pas les consignes peut se voir attribuer la note 0.
- Le travail d'équipe faisant partie de l'évaluation, le projet est réalisé en binôme (pas de trinôme). Un binôme signifie deux étudiant·e·s **du même groupe de TD**.
- Attention à la gestion de votre temps : concentrez-vous sur l'implémentation des fonctionnalités demandées, et pas plus. Cela ne sert à rien de gagner un point bonus pour quelques améliorations mineures mais de perdre plusieurs points pour des fonctionnalités non développées.



ENTITER



Association

L'objectif du projet est de développer un site web permettant de gérer des **listes de lecture musicales** (playlists).



Attributs



Cardinalites

1 Des spécifications au script SQL

Les spécifications sont les suivantes. **Un groupe de musique** **compose, interprète et enregistre** des musiques. **Un groupe** est caractérisé par son nom, une date de formation et éventuellement une date de séparation. Les **musicien-ne-s**, pour qui on stocke un nom, prénom et nom de scène, **composent les groupes**. **Un-e musicien-ne** peut faire partie de plusieurs groupes, et à différentes périodes. On mémorise le rôle de chaque membre du groupe (guitariste, bassiste, etc.) et si c'est un membre fondateur ou pas. **Une chanson représente l'oeuvre d'un groupe**. Elle possède un identifiant, un titre, une date de création et un ou plusieurs genres. Notez que les genres sont hiérarchisés : le genre *rock alternatif* a pour parent le genre *rock*. Il arrive fréquemment que deux chansons soient en relation (e.g., une reprise d'une chanson existante). On stockera donc des types de relation (e.g., reprise, suite, parodie) et les relations entre les chansons. Enfin, une chanson peut prendre différentes formes (version originale, version réduite pour la radio, versions *live*). Les versions d'une chanson, qui peuvent être vues comme **un fichier audio**, permettent donc de modéliser cet aspect. Chaque version est identifiée par sa chanson et un numéro, et comporte une durée, une date et un nom de fichier. Une version possède d'autres propriétés, mais elles ne sont pas forcément connues en avance (e.g., *bitrate*, description, nombre de lectures) : le modèle doit donc permettre de stocker **un libellé de propriété** et d'associer une valeur pour une propriété à une version de chanson. **Une liste de lecture**, représentée par un identifiant, un titre et une date, inclut au moins une version. **Une chanson n'est composée que par un groupe**, mais **une version peut être interprétée par plusieurs groupes**. Il reste à évoquer **les albums** (titre, date de sortie, producteur), qui se divisent en trois catégories : **les albums studio** pour lesquels on stocke le nom de l'ingénieur·e du son, **ceux de compilation** pour lesquels on conserve une description et **les albums lives (concert)**. Pour ces derniers, **les différents lieux** (identifiant, nom et coordonnées) où ont été enregistrés les chansons doivent être stockés. Attention, il arrive qu'un groupe se produise en live plusieurs fois dans un même lieu. Les albums incluent une ou plusieurs versions (de chanson), avec leur numéro de piste. Un album peut être enregistré par différents groupes (certaines compilations par exemple). Enfin, un groupe et/ou un-e musicien-ne peuvent être invités à participer à un album (duo sur une chanson par exemple), et on fournit alors un commentaire qui détaille cette participation.

Dans un premier temps, nous allons concevoir la base de données à partir des spécifications. Votre modélisation doit respecter au mieux ces spécifications, mais votre site web n'utilisera pas tous les concepts décrits.

1. Produisez un diagramme entité / association pour ces spécifications (par exemple avec [MoCoDo](#), [Looping](#), [AnalyseSI](#) ou [JMerise](#) - attention, chaque outil a ses contraintes et limitations).
2. Produisez le schéma relationnel dérivé de votre diagramme E/A. Si vous générez ce schéma avec un outil de modélisation, il est recommandé de le vérifier et éventuellement de le corriger / compléter.
3. Produisez le script SQL permettant la création de la base de données. Si vous générez ce script avec un outil de modélisation, il sera nécessaire de le vérifier et de le corriger / compléter (ces modifications devraient être stockées dans un autre fichier que celui généré par l'outil de modélisation). Enfin, utilisez PHPMyAdmin pour insérer des instances fictives dans les tables les plus importantes (e.g., quelques chansons/versions, quelques groupes, quelques genres).

2 Design du site et pages statiques

Vous êtes libres d'organiser votre site comme bon vous semble. Mais chacune de vos pages doit avoir les cinq zones suivantes :

- Un entête (<header>, avec un logo (au choix, mais cliquable pour revenir à l'accueil) et un nom de site ;
- Un menu (<nav>), dont les libellés seront explicites ;
- Des statistiques sur la base de données : nombre d'instances pour vos tables les plus importantes, top-5 des genres les plus représentés, groupes favoris (les plus présents, les plus joués), etc ;
- Le contenu de la page, qui correspond aux fonctionnalités développées dans les sections suivantes. Prévoyez également de créer une page d'accueil qui décrit les objectifs de votre site ;
- Un pied de page (<footer>), avec un lien vers le site de l'UCBL, un autre vers la page du BDW, l'année courante et des remerciements (e.g., site [OpenClipart](#) pour les images).

Il ne faut **pas** implémenter un système d'authentification¹ ! La mise en page et mise en forme se feront évidemment avec des styles CSS. L'esthétique est prise en compte lors de la notation, aussi, soignez votre site. Le projet sera évalué avec le navigateur **Mozilla Firefox**, donc vérifiez le rendu de votre site avec ce navigateur !

3 Fonctionnalité 1 : ajouter une chanson

Pour cette première fonctionnalité, nous allons créer une page pour ajouter une nouvelle version d'une chanson. Celle-ci inclut un formulaire avec les informations à saisir pour la chanson : titre, dates, durée, nom de fichier, groupe et genre(s). Le formulaire permet de choisir le groupe qui a composé la chanson ainsi que le(s) genre(s), mais pas d'en créer de nouveaux. Pensez à proposer des valeurs par défaut pour chaque champ de saisie, afin de ne pas avoir à remplir tous les champs du formulaire lors des tests. Quand c'est nécessaire, on utilisera des listes pour sélectionner une valeur (pour le groupe, les genres). Le système effectuera des vérifications avant d'insérer en base et affiche des messages pertinents pour faciliter la correction des erreurs.

Développez ensuite une page qui permet d'afficher les versions (de chanson), avec leurs informations (titre, nom du groupe, durée, genres, nom de fichier).

Lors de la soutenance, il est fort probable que les enseignant-e-s testent votre application en saisissant des valeurs absurdes susceptibles de déclencher des erreurs dans votre application. Alors essayez de penser au pire !

4 Fonctionnalité 2 : intégrer/importer des chansons

Cette fonctionnalité consiste à migrer des données fournies (chansons, groupes et genres, issus d'un *dump* produit par le logiciel [Clementine](#)) dans votre base de données. Ces données ne sont malheureusement pas correctement modélisées, et il est donc nécessaire de les transformer avant de les stocker dans votre base. C'est un processus fréquent appelé *intégration de données*. L'utilisation de ces données est **obligatoire**, et votre script d'intégration doit être **exécutable sur de nouveaux jeux de données** lors de la soutenance.

¹Un système d'authentification se base sur des protocoles standardisés et sécurisés comme SSL, Kerberos ou CAS.

Vous allez donc écrire un script PHP qui interroge le jeu de données fourni pour extraire les informations pertinentes, les transformer si nécessaire, puis les insérer dans votre base. En pratique, vous allez tester votre script à plusieurs reprises, donc prévoyez de vérifier si les chansons, genres ou groupes n'existent pas déjà avant d'insérer !

On peut décomposer l'étape d'intégration en trois étapes :

- Réfléchir aux correspondances entre les données existantes et votre schéma (e.g., à quel attribut de votre schéma correspond tel attribut fourni). Vérifier si les types de données sont cohérents entre attributs correspondants, et quelles sont les transformations nécessaires au niveau des valeurs. N'oubliez pas que le jeu de données fourni n'est pas bien modélisé : si vous modifiez votre schéma de BD pour intégrer plus facilement les données fournies, votre schéma ne respectera plus les spécifications demandées et vous serez pénalisé-e ;
- Pour accéder aux données fournies et récupérer les informations pertinentes, il faut indiquer le nom de la BD (**dataset**) avant la table (**songs2000**)² au niveau du **from** de vos requêtes : **SELECT * FROM dataset.songs2000** ;
- Pour chaque tuple résultat, appliquer éventuellement des transformations sur les valeurs (soit directement en SQL, soit en PHP), puis écrivez une requête **INSERT** pour peupler vos tables. Si certaines données sont manquantes et ne permettent pas de respecter les contraintes de votre schéma, utilisez un **NULL** ou choisissez une valeur par défaut. Enfin, n'oubliez pas de peupler les tables liées !

Toutes les données fournies (i.e., tous les attributs) doivent être stockées dans votre BD. Avant la soutenance, il est possible que l'on supprime les données fournies. Si votre intégration est réussie, le tableau suivant donne les statistiques à obtenir (sur le dataset *songs2000*).

nb_chansons	nb_groupes	nb_genres
2000	720	311

5 Fonctionnalité 3 : créer une liste de lecture aléatoire

La troisième fonctionnalité a pour but de créer une liste de lecture aléatoire selon des paramètres saisis par l'utilisateur-ice. La page comporte donc un formulaire pour renseigner les paramètres suivants :

- Durée (approximative) de la playlist (par défaut, 20 minutes). Les chansons sélectionnées pour la playlist devront totaliser une durée proche de celle saisie (à la minute près) ;
- Genre privilégié (par défaut, aucun). Si un genre est sélectionné, alors les chansons de la playlist devront être en majorité de ce genre (mais pas exclusivement) ;
- Préférence sur les statistiques : les chansons les plus jouées (attribut *playcount*), les plus sautées (attribut *skipcount*) ou celles jouées le plus récemment (attribut *lastplayed*, valeur en secondes). Il n'y aucune valeur par défaut pour ces 3 attributs. Si une valeur est choisie, l'algorithme de sélection choisira des chansons respectant la préférence exprimée.

Le titre de la liste peut être demandé à l'utilisateur-ice ou généré aléatoirement. Quand la liste de lecture est générée, les informations sur sa durée totale, le pourcentage de chansons avec le genre privilégié sont affichées sur la page, ainsi que la liste des chansons sélectionnées. Selon les données présentes et l'algorithme développé, il peut arriver que toutes les contraintes ne soient pas satisfaites. Dans ce cas, il serait bien d'en informer l'utilisateur-ice. Enfin, la liste de lecture sera stockée dans la base de données.

²Pour vos tests, vous disposez d'un jeu de données plus petit avec 100 chansons : **SELECT * FROM dataset.songs100** ; (uniquement pour tester, les données à intégrer sont celles de la table *songs2000*).

6 Fonctionnalité 4 : gérer les listes de lecture

La dernière fonctionnalité consiste à gérer les listes de lecture stockées en base. Vous êtes libres sur la présentation, mais les fonctionnalités suivantes sont attendues :

- Afficher les listes de lecture stockées en base (titre, durée totale, nombre de chansons) ;
- Visualiser le contenu (chansons) d'une liste de lecture ;
- Modifier une liste de lecture en permettant l'ajout ou la suppression de chansons dans la liste. Bien penser à mettre à jour la base de données ;
- Comparer 2 listes de lecture pour évaluer leur degré de ressemblance (par exemple, afin de supprimer des listes très similaires). L'idée est de calculer un score entre 0 (les listes sont totalement différentes) et 1 (les listes sont identiques). À vous d'innover pour définir le calcul de ce score, mais ne comparez pas simplement le titre des chansons : exploitez les genres, les années, les groupes, etc.

7 Préparation des livrables

Trois livrables sont à rendre **le jour de soutenance de votre projet, avant 23h59, sur Tomuss** :

- Une archive contenant votre site web, en zip ou rar (colonne *archive_projet*) ;
- L'URL de votre site, sur le serveur **bdw.univ-lyon1.fr** (colonne *url_projet*) ;
- Les diapositives de soutenance en pdf (colonne *diapositives_projet*).

7.1 Archive avec le site web

L'archive est au format **zip** ou **rar**, et respecte les contraintes suivantes :

- Une **seule archive par binôme**
- L'archive contient au minimum **4 fichiers** :
 - le répertoire contenant les pages de votre site (code commenté et indenté). La page *index* contiendra vos noms et numéros étudiant en commentaire ;
 - un fichier **.txt** ou **.sql** avec le script SQL **"exécutable"** de création de votre base de données ;
 - deux images (ou PDF), l'une de votre diagramme E/A et l'autre de votre schéma Relationnel. Les logiciels utilisés pour générer ces images peuvent être limités (e.g., pas de spécialisation), donc c'est à vous de modifier les images pour rendre un diagramme et un schéma qui correspondent à vos attentes.

7.2 URL du site web

Après le rendu du projet, nous testerons votre application en utilisant l'URL saisie dans Tomuss :

- Elle doit être complète (login et répertoire(s), voire fichier index précisé si nécessaire) et elle doit pointer vers la page d'accueil de votre site (e.g., **http://bdw.univ-lyon1.fr/p1234567/mon_site/**). Merci de la vérifier une fois qu'elle est saisie dans Tomuss !
- Votre site doit déjà avoir des instances stockées en base **en nombre suffisant** pour simplifier les tests.

7.3 Soutenance

La soutenance dure 20 minutes par binôme et se décompose en deux parties : une présentation avec démo et avec diapositives (8 minutes) et une séance de questions (12 minutes). Les conditions suivantes s'appliquent :

- Pour la partie présentation, respectez le temps ! Vous serez interrompus au bout des 8 minutes, et donc pénalisés ;
- Les deux membres du binôme doivent parler lors de la présentation. Entraînez-vous ;

- Vous présentez soit sur votre machine, soit sur une machine de la fac. Dans tous les cas, vérifiez avant que la machine permette de lire vos diapositives, de faire la démo, etc. Prévoyez d'arriver quelques minutes en avance devant la salle, et quand on vous fait entrer, préparez-vous pour la présentation et les questions (i.e., lancement du site dans le navigateur, ouverture des diapositives, ouverture des fichiers source et des modèles) ;
- Vous devez répondre à quatre questions minimum pendant les 12 minutes. Donc donnez des **réponses claires et concises** (pas de baratin). Si vous ne savez pas répondre à une question, dites-le pour ne pas perdre de temps ;
- Si vous arrivez en retard à votre soutenance, vous passez en fin de séance... ou pas du tout selon les disponibilités de votre jury.

Vous pouvez réaliser vos diapositives avec n'importe quel logiciel (e.g., Libre Office, Power Point, Latex Beamer). Vous devrez par contre **déposer un PDF de vos diapositives** sur Tomuss. Les conditions suivantes s'appliquent pour les diapositives :

- Les diapositives seront soignées et numérotées. Évitez les longues phrases !
- Vos 5 diapositives doivent respecter le plan suivant :
 - Diapositive 1 : vos noms, prénoms, numéros étudiant, et si vous participez au concours, le nom et/ou le logo de votre site ;
 - Diapositive 2 : réalisations (fonctionnalités implémentées, avec des détails pour les fonctionnalités avancées) ;
 - Diapositive 3 : difficultés rencontrées au niveau technique, organisationnel, etc. (évitez l'éternel "manque de temps") ;
 - Diapositive 4 : diagramme E/A (prévoir un agrandissement si c'est illisible) ;
 - Diapositive 5 : une capture d'un écran de votre site, qui permet d'annoncer la démo.
- La démo de votre site doit être préparée (déroulement d'un scénario). Vous avez 8 minutes pour les diapositives et la démo, donc focalisez-vous sur les points importants.

Concours (optionnel) : Pour vous détendre (en cas de blocage sur un bug, d'engueulade avec votre binôme, etc.), deux concours sont organisés, celui du **meilleur nom de site** et celui du **plus beau logo**. Les deux binômes gagnants remporteront un paquet de chamallows. Soyez créatifs/ves !

