

## Klex – le Serveur multimédia

### Introduction

Le projet auquel vous allez participer a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles. De plus, vous vous essaieriez à la programmation d'application utilisant une base de données (à travers des transactions). Le développement sera fait en Java en utilisant l'API JDBC.

Le projet est à faire en équipes de cinq et donnera lieu à une réunion de suivi intermédiaire, ainsi qu'à une soutenance en fin de projet. La constitution des équipes et la remise des livrables (rapport final et code source) se fera sur l'application *Teide*. L'évaluation se fera sur les aspects bases de données du projet uniquement en l'état au moment de la soutenance.

### 1. Description de l'application

La société Klex souhaite proposer un serveur multimédia accessible aussi bien par des équipements physiques – téléviseurs, radios, enceintes connectées, etc. – que par des logiciels clients – logiciel dédié, clients web, ou clients standards via le protocole DLNA – comme le sont ses concurrents tels que Emby, Filestage, ou Kodi parmi d'autres. Les utilisateurs doivent pouvoir organiser leur bibliothèque comme ils le souhaitent en affectant des catégories à leurs contenus multimédias (la liste des catégories est facilement extensible).

Votre tâche consiste à concevoir et implémenter la base de données décrivant les fichiers multimédias hébergés sur le serveur (films ou fichiers audio), leurs contenus, ainsi que les caractéristiques des équipements et logiciels clients afin d'optimiser la diffusion des contenus. Quelques fonctionnalités devront également être étudiées et démontrées en fin de projet.

Les utilisateurs sont identifiés par une adresse email et décrits par leur nom, prénom, âge et langue de diffusion préférée. Un code d'accès à quatre chiffres (stocké en clair) est également associé au compte utilisateur. Ces informations seront utiles pour rendre l'interaction plus agréable aux utilisateurs, ainsi que pour respecter les recommandations d'âge minimal pour les contenus.

Les fichiers déposés par les utilisateurs sur le serveur sont décrits par un identifiant unique généré automatiquement, leur taille en Ko, la date de leur dépôt, ainsi que la description du ou des flux multimédias qu'ils contiennent. Chaque flux d'un fichier multimédia est numéroté. Un flux peut être un flux vidéo (ne comprenant que l'image), un flux audio ou un flux texte (paroles d'une chanson ou sous-titre d'un film). La description des flux vidéo comprend le codec utilisé (par exemple : MPEG2, MPEG4, DivX, H264, ...), la résolution de l'image (largeur et hauteur) et le débit en Kbit/s. Les flux audio sont décrits par leur codec (comme MP3, MPEG4, ACC, AC3, ...), leur échantillonnage (16, 24 ou 32 bits), leur débit en Kbit/s et leur langue. Les flux texte sont simplement décrits par leur codec (tels que AQTtitle, DKS, Kate, ...), leur débit en Kbit/s et leur langue.

Les contenus multimédias sont soit des films, soit des pistes musicales. L'association entre fichiers et les contenus est normalement mise en place par un service web faisant intervenir un algorithme de « Deep Learning », mais dans le cadre de ce projet, l'association sera faite à la main. Plusieurs fichiers peuvent être associés à un même contenu. En effet, les sous-titre d'un film peuvent par exemple être séparés du fichier audio-vidéo, ou un même film peut être disponible dans des résolutions et ou débits différents ou encodé avec des codecs différents, de même pour les pistes musicales. L'utilisateur aura la possibilité de corriger l'association entre fichiers et contenus.

Un film est identifié par son titre et son année de sortie. Il est décrit par une ou plusieurs catégories, un résumé, un âge minimum recommandé et l'URL de l'affiche du film. Une ou plusieurs URL de photo extraites du film peuvent également être incluses dans sa description.

Les pistes musicales sont organisées en albums. Un album a un numéro unique, un titre, un nom de groupe ou d'artiste, une date de sortie, l'URL d'une image de sa pochette et une ou plusieurs catégories. Les pistes d'un album, associées à des fichiers, sont identifiées par leur numéro de piste et décrits par leur titre et leur durée (en minutes). Une ou plusieurs catégories peuvent être également associées aux pistes si, par exemple, un même album contient des pistes de style très différents. Les catégories de films et de musique sont bien évidemment distinctes.

Différents artistes interviennent dans des contenus multimédias. Ils peuvent par exemple être acteurs dans un film (et l'on précisera le nom du personnage qu'il joue) ou avoir une autre fonction (par exemple : réalisateur). Ils peuvent être également musicien (en précisant son instrument) ou chanteur sur une piste audio. Les artistes sont identifiés par un numéro unique et décrits par leur nom, leur date de naissance (si elle est connue), l'URL d'une photo et leur spécialité principale. Une courte biographie peut également être donnée. Un même artiste peut être musicien dans des albums et également intervenir dans des films, ou l'inverse. Tout est possible.

Les contenus multimédias seront joués sur des équipements ou des logiciels que nous appellerons des clients. Tous les clients n'ont pas les mêmes caractéristiques et ne peuvent donc pas jouer tous les contenus. La base de données doit donc également comprendre leur description. Celle-ci comprend la marque et le modèle (ou nom de logiciel) qui serviront d'identifiant. Elle comprend également la résolution maximale d'affichage (largeur et hauteur), ainsi que la liste des codecs audio, vidéo et texte qu'il peut interpréter.

## 2. Travail à réaliser

### 2.1. Modélisation du problème

La modélisation se décompose en deux temps. Dans un premier temps, vous aurez à **analyser le problème posé** pour en extraire les propriétés élémentaires, les dépendances fonctionnelles reliant ces propriétés, ainsi que tous les autres types de contraintes (contraintes de valeur, contraintes de multiplicité et contraintes contextuelles). Vous devrez proposer ensuite un **schéma Entités/Associations** représentant les données nécessaires à l'application et leurs liens sémantiques (ce qui correspond à l'état cohérent de la base de données). Le schéma Entités/Associations **doit être totalement justifié par l'analyse**.

### 2.2. Implantation de la base de données

Vous devrez ensuite traduire le schéma Entités/Associations en un **schéma relationnel** que vous implanterez sur le SGBD Oracle disponible sur *oracle1*. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet. **Vous préciserez également la forme normale de chacune des relations obtenues.**

### 2.3. Analyse des fonctionnalités

Vous devrez définir les requêtes SQL2 nécessaires pour réaliser les fonctionnalités suivantes en les regroupant en transactions, ce qui permettra d'assurer la cohérence globale de la base de données, même en cas d'accès concurrents :

- **Insertion d'un nouveau fichier** : ceci comprend également la description de ses flux, son association à un contenu existant ou la création d'un nouveau contenu totalement décrit, voire d'une ou plusieurs nouvelles catégories et éventuellement d'artistes.
- **Sélection de contenus sur un client** : Une première sélection se fait en fonction d'une catégorie donnée, des possibilités du client, ainsi que des préférences et de l'âge de l'utilisateur. Une seconde étape permettra d'afficher la description complète du contenu et du ou des fichiers associés.
- **Suppression d'un contenu** : cette suppression implique la suppression des fichiers correspondants, et éventuellement celle des artistes et/ou albums s'ils ne sont plus référencés par d'autres contenus.

Ces requêtes et transactions peuvent (doivent !) être testées sur Oracle (SQL\*Plus, *Adminer* ne gérant pas les transactions) pour en vérifier leur bon fonctionnement, y compris pour des exécutions concurrentes.

### 2.4. Implantation des fonctionnalités

Les fonctionnalités précédemment étudiées devront être implantées en Java/JDBC. Vous pouvez choisir une interface textuelle (simple et suffisant) ou graphique (attention au temps supplémentaire requis), cela n'a pas d'importance vu que seuls les aspects Bases de Données seront évalués.

## 3. Déroulement du projet

Le projet sera constitué de 15 heures en séances encadrées et 3 heures réservées pour les soutenances.

### 3.1. Séances encadrées

La connexion à Riot/Element lors des séances encadrées est **obligatoire** et **sera contrôlée**. Lors de ces séances, les équipes devront avancer sur le projet et pourront poser des questions à leur encadrant. Attention : posez bien vos questions. L'encadrant jouera **soit le rôle du client** (et, dans ce cas, ne pourra traiter les aspects BD), **soit le rôle d'un expert en bases de données** (mais, dans ce cas, il ne connaît rien à l'application), jamais les deux à la fois.

En fin de chaque séance, chaque équipe devra déposer une **version actualisée de la documentation du projet sans la valider** (validation uniquement en fin de projet). Ceci afin de pouvoir suivre le déroulement du projet et de pouvoir intervenir au plus tôt en cas de grosses erreurs/anomalies/difficultés.

### 3.2. Outils

Vous disposez de quatre outils principaux pour le bon déroulement du projet :

- **Chamillo** : vous y trouverez les documentations techniques pour accéder à Oracle et pour utiliser JDBC, des liens Internet utiles, ainsi qu'un **forum** dans laquelle vous trouverez **les réponses aux questions fréquentes** (à consulter souvent, donc).

- **Zoom** : Chaque séance commence et se termine dans une réunion Zoom (les liens seront affichés sur la page Chamilo du projet). La réunion Zoom reste ouverte durant toute la séance, vous pourrez vous y reconnecter au besoin pour interagir avec vos encadrants.
- **Riot/Element** : outil d'interaction en- et hors-séance privilégié. Un salon sera créé par équipe de projet par vos encadrants. La connexion à ces salons est obligatoire pour toutes les séances, même si vous utilisez d'autres outils en tant que membres d'une même équipe.
- **Teide** : l'application de gestion de projet. Vous devrez utiliser Teide pour constituer vos équipes, déposer vos rendus (**documentation à chaque séance sans la valider**, code source Java et SQL, et supports pour la soutenance en fin de projet).

### 3.3. Suivi

Le projet donnera lieu à une réunion de suivi intermédiaire entre l'encadrant et chacune des équipes (via Zoom). Cette réunion sera provoquée par les équipes elles-mêmes ou par l'encadrant en fonction de l'avancement du travail. Les thèmes discutés lors du suivi concerneront l'**analyse et la modélisation Entités/Associations** (~4<sup>ème</sup> séance encadrée).

### 3.4. Livrables

- **Documentation du projet** : Vous devrez maintenir la documentation du projet tout au long de son déroulement. La documentation doit comprendre :
  - l'analyse du problème,
  - la conception Entités/Associations,
  - sa traduction en relationnel en précisant les formes normales des relations,
  - l'analyse des fonctionnalités (transactions), leur implantation sous forme de requêtes SQL2,
  - ainsi qu'un bilan du projet (organisation, points difficiles rencontrés, etc.)

Un petit mode d'emploi de votre application est également le bienvenu.

La documentation doit comprendre les explications nécessaires à sa compréhension et à la justification de vos choix.

- **Sources java et SQL2** : Vous devrez rendre en fin de projet un script SQL permettant de créer votre schéma relationnel, un script SQL permettant de peupler la base de données, l'implantation des fonctionnalités (squelette des transactions et requêtes SQL), ainsi que le code source Java du démonstrateur.
- **Supports pour la soutenance** : Les slides prévus pour une soutenance devront être déposés sur *Teide* (visez 15 à 20 minutes et écrivez le discours correspondant), ainsi qu'une vidéo de 10 minutes de démonstration de votre prototype, commentée à l'oral.