Ensimag 2ème année
2022-2023

Projet Bases de Donnée: GrenobleEat

1 – 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 essaierez à la programmation d'application utilisant une base de données (à travers des transactions). Le développement du démonstrateur sera fait en Java en utilisant l'API JDBC.

Le projet est à faire en **équipes de cinq à six** 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 uniquement sur les aspects bases de données du projet en l'état au moment de la soutenance.

2 – Description de l'application

Une société de réservation de repas souhaite mettre en place **GrenobleEat**, une base de données afin de permettre la gestion de leur service. Le service proposé est simple : les restaurants de Grenoble et ses environs inscrits sur ce système ont la possibilité de proposer à leur client des réservations de repas à consommer sur place, à emporter ou à faire livrer directement à leur domicile (ou ailleurs). Les restaurants peuvent être classés par catégorie et peuvent recevoir une note des clients commandant chez eux.

Pour simplifier les commandes "sur place", on supposera qu'il n'y a qu'un seul service le midi (12h-14h) et un seul le soir (19h-23h). Il n'y a donc pas à prendre en compte le fait que les tables puissent se libérer pendant un service et de nouveau accueillir des clients: les clients bloquent la table tout le service.

Chaque restaurant est identifié de manière unique par son e-mail et est décrit par son nom, son numéro de téléphone, son adresse, une ou plusieurs catégories, ses horaires d'ouverture (heures d'ouverture des différents créneaux pour chaque jour d'ouverture, on négligera les jours fériés ou autres fermetures exceptionnelles), son nombre de place assises, un texte de présentation, les plats qu'il propose et le type de commande qui peut lui être passé (livraison, à emporter et sur place). Un restaurant possède également des évaluations qui sont associées aux commandes passées chez lui. Le principe des évaluations est décrit plus loin.

Un plat est identifié par un numéro unique au restaurant qui le propose et possède un nom, une description,

un prix et optionnellement une liste des allergènes qu'il peut contenir.

Les catégories de restaurant sont organisées en un arbre avec des sous-catégories, sans profondeur fixée.

Ainsi, la catégorie "cuisine savoyarde" et "cuisine dauphinoise" appartiennent toutes les deux à la catégorie "cuisine des alpes" qui est elle-même une sous-catégorie de la catégorie "cuisine régionale". Un restaurant peut se voir attribuer plusieurs catégories. Il hérite les catégories mères de ses catégories, par exemple un restaurant catégorisé "cuisine savoyarde" est donc catégorisé "cuisine régionale" et "cuisine des alpes" par transitivité.

Les comptes utilisateurs ont un identifiant unique. Les clients sont identifiés par leur adresse email et possèdent un mot de passe (en clair, nous ne regardons pas les aspects sécurité ici), un nom, un prénom et une adresse postale (pour les livraisons). Un client peut parcourir la liste des restaurants en se basant sur l'organisation des catégories et/ou en triant ceux-ci par leur note moyenne, il peut également choisir de se baser sur un ensemble de catégories recommandées en fonction de son historique de notations des commandes.

Une commande doit pouvoir être identifiée de manière unique et est décrite par sa date, son heure, le client qui la passe, le restaurant dans lequel cette commande est passée, le type de commande (livraison, à emporter ou sur place), le contenu de la commande, son prix final et son statut (attente de confirmation, validée, disponible, en livraison, annulée par le client, annulée par le restaurant).

S'il s'agit d'une livraison, on doit également connaître l'adresse de livraison et optionnellement un texte contenant des informations pour aider le livreur. On ajoutera également l'heure de la livraison lorsque celle-ci a été effectuée.

S'il s'agit d'une commande à consommer sur place, on doit connaître le nombre de personnes venant manger, ainsi qu'une heure d'arrivée. Comme il n'y a qu'un seul service le midi et un seul le soir, on se contentera de veiller à ce que le nombre total de personnes ayant commandé "sur place" lors d'un de ces services est inférieur à la capacité maximale du restaurant.

Après une commande, et si elle n'a pas été annulée par l'utilisateur, un utilisateur peut déposer une évaluation sur cette commande. Une évaluation est rattachée à une commande et est décrite par la date d'évaluation, l'heure d'évaluation, un avis sous forme d'un petit texte et une note qui est un entier entre 0 et 5. La note d'un restaurant correspond à la moyenne des notes des évaluations des commandes passées dans ce restaurant.

Les recommandations de catégories se basent sur l'historique de commandes des utilisateurs et les notes qui leur sont reliées. En priorité, les recommandations vont concerner **les catégories** pour lesquelles l'utilisateur a donné la meilleure note moyenne lors de ses commandes. Ensuite, pour départager les catégories se retrouvant à égalité, les recommandations seront triées par ordre décroissant d**es catégories** pour lesquelles il y a eu le plus de notes déposées et ce quel que soit l'utilisateur.

Nous devrons enfin prendre en compte la RGPD, en particulier le droit à l'oubli. De ce fait, si un client demande à ce que l'on supprime ses données personnelles il est important de conserver les commandes qui ont été effectuées. En effet, il reste primordial de savoir qu'un même client a effectué un certain nombre de commandes, même si nous ne savons plus qui il est.

3 – Travail à réaliser (15h encadrées + travail personnel, le tout x5-6)

Le travail se découpe en quatre étapes, décrites ci-dessous, dont seules les deux premières sont obligatoirement séquentielles. L'étape 3 peut être découpée et parallélisée, et l'étape 4 peut être commencée dès le début du projet.

3.1 – 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. Expliquez vos choix de modélisation dans la documentation du projet.

3.2 – Implantation de la base de données

Vous devrez ensuite traduire le schéma Entités/Associations en un **schéma relationnel** décrivant les noms des relations obtenues, les noms et types de leurs attributs, ainsi que les contraintes à vérifier pour chacune des relations. **Vous devrez également justifier vos choix de traduction et expliquer les points difficiles**. Vous préciserez et justifierez la **forme normale de chacune des relations obtenues**.

Vous implanterez ensuite ce schéma relationnel sur le **SGBD Oracle disponible sur le serveur `oracle1`**. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet.

3.3 – Analyse des fonctionnalités

Vous devrez définir les **requêtes SQL2** nécessaires pour réaliser les fonctionnalités suivantes en les **regroupant en transactions** maintenant la cohérence de la base de données conformément au schéma Entités/Associations et aux contraintes associées :

Parcours des restaurants disponibles

Cette fonctionnalité inclura toutes les étapes nécessaires au parcours du catalogue :

1. Connexion du client à l'application avec son email comme identifiant et vérification du mot de passe.
2. Parcours des catégories et sous-catégories, ou des catégories recommandées, au choix. Les restaurants des catégories seront affichés par ordre décroissant de leur note et par ordre alphabétique (un seul tri multicritère).
3. Possibilité de filtrer les résultats par jour/heure d'ouverture des restaurants (pas besoin de gérer l'heure de livraison estimée ni les vacances et jours fériés).

4. Consultation de la fiche complète de chacun des restaurants sélectionnés.

Commande

1. Passage d'une commande auprès d'un restaurant.
2. Si le nombre de places est insuffisant pour venir sur place, prévenir le client pendant la commande.
3. (optionnel) En cas de refus de commande en raison d'un nombre de place insuffisant, proposer au client des restaurants correspondant à ses critères (en se basant sur ses recommandations, ou le nombre de places qu'il demandait si sa commande "sur place" a été refusée)

Droit à l'oubli

Cette fonctionnalité doit permettre à un utilisateur de supprimer ses données personnelles (identifiant de compte, email, mot de passe, nom, prénom et adresse postale). Il ne pourra alors plus se connecter à notre système. Néanmoins les offres et achats déjà effectués par l'utilisateur en question devront être conservés et être associés à un nouvel identifiant de compte créé pour l'occasion.

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

3.4 - Implantation des fonctionnalités

Les fonctionnalités précédemment étudiées devront être implantées dans un **démonstrateur programmé en Java/JDBC**. Nous vous recommandons l'implantation d'une **interface textuelle simple** (quelques menus et saisies utilisateurs/affichages).

L'évaluation ne portera que sur les aspects Base de Données et pas sur le code Java.

4– Déroulement du projet

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

4.1– Séance encadrées

Comme toute épreuve pratique à l'Ensimag, **la présence aux séances encadrées est obligatoire et sera contrôlée**.

La première séance comprendra une présentation du projet, de JDBC et de l'analyse de fonctionnalités. Le travail se fera ensuite par équipe (créées préalablement via Teide). Vous pourrez poser des questions à votre encadrant à tout moment lors des séances, mais il jouera deux rôles alternativement : il sera **soit votre client** (qui connait les besoins applicatifs, mais ne connait rien en base de données), **soit un expert en bases de données** (qui ne connait pas et ne veut pas connaitre les besoins applicatifs).

Posez bien vos questions.

4.2– Outils

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

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

Riot/Element : outil hors-séance privilégié pour interagir avec votre encadrant (http://element.ensimag.fr).

Teide : l'application de gestion de projet. Vous devrez utiliser Teide pour constituer vos équipes, déposer vos rendus (documentation du projet à déposer à chaque séance sans la valider, code source Java et SQL2, et supports pour la soutenance en fin de projet).

4.3– Suivi

Le projet donnera lieu à une **réunion de suivi intermédiaire** entre l'encadrant et chacune des équipes (~4ème séance encadrée). 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**.

4.4– Livrables

Vous aurez trois livrables à fournir pour ce projet :

1. **Documentation du projet** (un fichier PDF): Vous devrez maintenir la documentation du projet tout au long de son déroulement. La documentation doit comprendre :

   L'analyse du problème sous forme de contraintes (DF, de valeur, de multiplicité, contextuelles). Expliquez vos choix sur les points difficiles.

   La conception Entités/Associations noté en UML, comme vu en cours, en expliquant vos choix. Précisez bien les contraintes non représentées dans le schéma. Le schéma peut être réalisé avec l'outil Dia (disponible sur toutes les plateformes), n'importe quel outil de dessin vectoriel, voire même sur papier puis scanné, tant qu'il est lisible et respecte les notations graphiques vues en cours.

   Sa traduction en relationnel en précisant les formes normales des relations et en justifiant vos choix. Les contraintes non implantables en relationnel devront également être listées.

   L'analyse des fonctionnalités (transactions), leur implantation sous forme de requêtes SQL2, le tout bien commenté.

   Un bilan du projet (organisation, points difficiles rencontrés, etc. en prenant du recul sur le travail effectué).

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

   La qualité et la complétude de la documentation seront prises en compte dans la notation.

2. **Sources Java et SQL2** (un fichier ZIP ou TGZ) : Vous devrez rendre en fin de projet un script SQL2 permettant de créer votre schéma relationnel, un script SQL2 permettant de peupler la base de données, l'implantation des fonctionnalités (squelette des transactions et requêtes SQL2), ainsi que le code source Java du démonstrateur.
3. **Supports pour la soutenance** : Les slides prévus pour la soutenance devront être déposés sur Teide (visez 15 à 20 minutes).

4.5 – Soutenance finale

La soutenance **durera en tout 30 minutes** divisées de la façon suivante :

15 à 20 minutes pour que vous présentiez votre projet et faire une démonstration bien préparée (scénario, jeux de données en place, etc.) ;

10 à 15 minutes pour que le jury puisse vous faire un retour sur le projet et vous poser des questions. L'objectif de la soutenance est double : vous devez **convaincre le client que le produit répond bien à ses besoins**, et vous devrez **convaincre l'expert que votre équipe a fait du bon travail**. La soutenance doit donc reprendre les différents points de la documentation en insistant sur les points que vous jugerez les

plus pertinents.

5 – Conclusion

Le projet se déroule en quatre semaines, soutenances incluses. **Cela va aller très vite. Une bonne organisation de l'équipe est primordiale.**

Il est recommandé de diviser les tâches dès le début du projet et de toujours garder les « candides » par tâche (des personnes qui ne participent pas du tout à cette tâche), de façon à ce qu'ils puissent apporter un regard critique sur ce qui a été fait lors des réunions d'équipe.