POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

INF3710 - Fichiers et bases de données

Automne 2022

Travail Pratique 4

Normalisation, SQL et Application Web

Durée	9 heures
Session	Automne 2022
Public cible	Étudiants de 1er cycle
Lieu de réalisation	L-4708
Date de Remise	Dimanche 4 décembre 2022 à 23h59 : Groupes B1 – jeudi et vendredi Dimanche 27 novembre 2022 à 23h59 : Groupes B2 – jeudi et vendredi
Taille de l'équipe	**2 personnes**
Pondération	**20%**
Directives particulières	**1.** Tout retard dans la remise du compte-rendu entraîne automatiquement une pénalité comme discuté dans le plan de cours. **2.** Aucun retard de plus de 24 heures ne sera admis, la note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés. **3.** Aucun compte-rendu ne sera corrigé, s'il est soumis par une équipe dont la taille est différente **de deux (2) étudiants** sans l'approbation préalable du chargé de laboratoire. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés. **4.** Soumission du compte rendu par **Moodle** uniquement (https://moodle.polymtl.ca). **5.** Aucune soumission "hors **Moodle**" ne sera corrigée. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés.

Table des matières

1. Travail à remettre

- Le compte-rendu à soumettre est un fichier **pdf** dont le nom est formé des numéros de matricules des membres de l'équipe, séparé par un trait de soulignement (_).
- Utiliser la Page de présentation-Compte rendu disponible sur Moodle, comme page de garde de votre compte rendu.
- Il doit comporter une réponse concise à chacune des questions posées.
- L'application Web développée doit être remise en archive compressée **zip**.
- L'application Web remise **ne doit pas** contenir les **node_modules**.
- Veuillez fournir un **README.txt** pour diverses informations pertinentes pour bien transpiler votre application sans erreur de versions.

2. Évaluation

Rubriques	**Points**
Appréciation générale du rapport	1.00
Normalisation	
Réponses aux questions	19.00
Total de points portion normalisation	19.00
Schéma relationnel	
Transformation du diagramme E-A en un schéma relationnel	15.00
Total de points portion schéma relationnel	15.00
SQL	
Création de la base de données	6.00
Peupler la base de données	3.00
Réponses aux questions des requêtes SQL	23.00
Total de points portion SQL	32.00
Application Web	
Pages	11.00
Requêtes + contrôle de qualité fonctionnelle et performance	22.00
Total de points portion application Web	33.00
Total des points pour l'application Web, la normalisation et SQL	**100.00**
Pondération	**20%**

À noter :

1- **Il faut avoir une exécution fluide de l'application avec gestion de toutes les exceptions (on ne devrait pas avoir à redémarrer l'application pour cause de bogues);**
2- **Une pénalité allant jusqu'à 15 points pourrait être appliquée si ce n'est pas le cas.**

3. Objectifs du laboratoire

Cette séance de laboratoire a pour but de permettre à l'étudiant(e) de :

- **Portion normalisation :**
  - ✓ Vérifier pour chacune des relations dans un schéma relationnel donné si elle est en forme normale.
  - ✓ Normaliser les relations qui ne sont pas normalisées.
  - ✓ Établir un schéma relationnel normalisé.
  - ✓ Expérimenter de manière intuitive les concepts de la normalisation vus dans les séances du cours théorique.
- **Portion SQL :**
  - ✓ Se familiariser avec des requêtes SQL complexes.
  - ✓ Apprendre à modifier intelligemment une base de données en fonction de requis d'affaire.
  - ✓ Faire des liens entre les notions pratiques et théoriques qui lui ont été présentée dans le cadre du cours INF3710.
- **Portion application Web :**
  - ✓ Interagir avec une base de données PostgreSQL à partir d'une API.
  - ✓ Utiliser un guide théorique et pratique pour implémenter une application Web.
  - ✓ Se servir d'une bibliothèque (Angular Material) de design pour édifier un site Web de qualité, notamment pour offrir des vues et modales.
  - ✓ Utiliser un schéma et un script de population SQL pour afficher de l'information au sein de pages.
  - Côté serveur :
    - ✓ Écrire des requêtes SQL (GET, POST, UPDATE, DELETE) au sein d'un service.
    - ✓ Ouvrir des routes pour accéder à une base de données au sein d'un contrôleur.
    - ✓ Gérer toutes les erreurs possibles suite à une requête SQL.
  - Côté client :
    - ✓ Offrir des voies de communication avec le routeur du serveur via HTTP au sein d'un service.

4. Normalisation (19pts)

Étude de cas :

Supposons que le schéma relationnel pour gérer une entreprise de livraison de kits repas qui possède plusieurs plans repas.

Client (nas, nomclient, prénomclient, numérocartedecrédit, cvc, numérodetéléphone)
Téléphone (numérodetéléphone, nas, nomclient)
Cartedecrédit (numérocartedecrédit, cvc, dateexpiration, nas, nomclient)
Planrepas (numéroplan, catégorie, fréquence, prix, numérorepas, descriptionrepas)
Repas (numérorepas, descriptionrepas)

Notez que :

- Chaque client peut avoir différentes cartes de crédit;
- Une carte de crédit n'appartient qu'à un seul client;
- Chaque client peut avoir plusieurs numéros de téléphones;
- Un numéro de téléphone n'appartient qu'à un seul client;
- Un plan repas peut avoir plusieurs repas;
- Un repas appartient qu'à un seul plan repas.

1. Vérifiez chacune des relations mentionnées ci-dessus si elle est en forme normale et justifiez votre réponse. **(9pts)**
2. Normalisez les relations qui ne sont pas normalisées. **(6.5pts)**
3. Présentez le schéma relationnel normalisé. **(3.5pts)**

5. Questions

En partant du **diagramme entité-association du premier exercice du TP2** qui porte sur la Gestion d'une application de livraison de kits repas, allez réaliser les étapes suivantes :

~~Question 1- Transformez le diagramme entité-association (**voir solutionnaire du TP2**) en un schéma relationnel. **(15pts)**~~

~~Question 2- Créez la base de données **TP4_Livraison** en implémentant le schéma relationnel de la **Question 1** en utilisant le système de gestion de base de données PostgreSQL. Enregistrez votre code SQL dans le fichier **TP4_Livraison.sql. (6pts)**~~

Question 3- Peuplez la base de données **TP4_Livraison** en ajoutant des informations pertinentes aux tables soient deux tuples par table. Enregistrez votre code SQL dans **le même** fichier **TP4_Livraison.sql** après les requêtes de création des tables. **(3pts)**

Question 4- Écrivez les requêtes SQL suivantes et les sauvegarder dans le fichier **requetes.sql**. **(23pts)**

4.1- Affichez les numéros (numéroclient) et les noms (nomclient) des clients qui ont commandé un repas avec un prix compris entre 20 dollars et 40 dollars. (**2pts)**

4.2- Afficher les numéros des plans repas (numéroplan) qui ne proviennent pas du fournisseur au nom de '*QC Transport*'. (**2pts)**

4.3- Affichez la liste des numéros des plans Famille (numéroplan) dont la catégorie du plan repas correspond à '*cétogène*'. (**2pts)**

4.4- Affichez le nombre de fournisseurs n'ayant pas de nom dans leur dossier (la valeur de *nomfournisseur* est *NULL*). (**2pts)**

4.5- Affichez les noms des fournisseurs (*nomfournisseur*) ayant fait des livraisons de plans repas dont le montant est supérieur aux livraisons faites par le fournisseur dont le nom est '*AB Transport*'. (**2pts)**

4.6- Affichez les noms des fournisseurs (*nomfournisseur*), les adresses (*adressefournisseur)* et le montant total des prix des livraisons de plans repas des fournisseurs ayant les deux plus larges montants de livraison sur la plateforme. (**2pts)**

4.7- Affichez le nombre de kit repas qui n'ont jamais été réservés chez les fournisseurs. (**2pts)**

4.8- Affichez les numéros (*numéroclient*), les noms (*nomclient*) et les prénoms *(prénomclient)* des clients dont le prénom ne commence pas par une voyelle (en majuscule ou en minuscule) et qu'ils habitent (villeclient) à la même adresse (adressefournisseur) que le fournisseur 'Benjamin'. Ordonnez ces clients alphabétiquement selon le nom. (**2pts)**

4.9- Affichez le pays des ingrédients (*paysingrédient*) et le nombre d'ingrédients par pays dont le *paysingrédient* ne contient pas la lettre g à la troisième position de la fin; triés par ordre décroissant selon le pays de l'ingrédient (*paysingrédient*). (**2pts)**

4.10- Créez une vue *'V_fournisseur'* contenant la catégorie du plan repas *'V_catégorie'*, l'adresse du fournisseur *'V_adresse'* et le total des prix de tous les plans repas desservis par ce fournisseur *'V_tot'*. Cette vue doit uniquement contenir les fournisseurs dont V_tot est supérieur à 12 500$ et dont le nom de la catégorie du plan repas contient la lettre *'e'* et la lettre *'o'* à la troisième position de la fin; triés par ordre

croissant selon le nom de la catégorie du plan repas et par ordre décroissant selon *'V_tot'*. Finalement, afficher le résultat de cette vue. **5pts**

Question 5- **Application Web (33pts)**

En se basant sur Question 2 et Question 3, vous êtes confiés à créer une application Web comme suit. À noter qu'il faut référer au **guide** d'application disponible sur Moodle sur ce lien (Tutoriel de l'application Web).

5.1- Votre application Web doit contenir **une page** qui affiche tous les champs et toutes les entrées de la table **Planrepas. (5pts)**

5.2- Nous voulons pouvoir ajouter (*INSERT*), modifier (*UPDATE*) et supprimer (*DELETE*) n'importe quel **Planrepas** présent au sein de votre base de données par l'intermédiaire d'une modale de votre application Web. Votre application web devrait gérer **toutes les erreurs possibles**, comme les erreurs de référencement de clefs, en offrant à l'utilisateur une modale avec une description appropriée de l'erreur survenue. L'ajout, la modification et la suppression doivent avoir leur propre page. Dans la page d'ajout, on inclut un formulaire avec les informations à saisir pour le **Planrepas**. 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 la catégorie, l'identifiant du fournisseur avec son nom, etc.) obtenue de la base de données. Dans la page de modification, on s'attend à charger toutes les informations reliées au **Planrepas** à partir de la base de données et on permet leur modification, de même pour la suppression. **(22pts)**

5.3- Votre application Web devrait contenir une page d'accueil qui contient des liens directs vers desdites pages à implémenter énumérées dans les deux questions 5.1 et 5.2. **(6pts)**

Nous vous conseillons fortement de vous inspirer du guide d'application vu en laboratoire et disponible sur Moodle sur ce lien (Tutoriel de l'application Web). Enfin, nous recommandons grandement l'utilisation d'Angular Material pour accélérer votre programmation.

Bon travail !