

École Polytechnique de Montréal
Département Génie Informatique et Génie Logiciel
INF3710 – Fichiers et Bases de données

TP 3 – Applications et Bases de données

Objectif:

1. Informations générales

<i>Pondération</i>	20%
<i>Taille de l'équipe</i>	2 personnes

Notez bien:

1. Tout retard dans la remise du TP entraîne automatiquement une pénalité comme discuté dans le plan de cours.
2. Aucun TP 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. Cette approbation ne sera accordée qu'à une seule équipe et qu'en cas de nombre impair d'étudiants dans le laboratoire. Sinon, la note de zéro sera attribuée aux étudiants concernés.
3. Soumission du TP par **Moodle** uniquement (<https://moodle.polymtl.ca>). Aucune soumission "hors **Moodle**" ne sera corrigée. La note de zéro) sera attribuée aux étudiants concernés.

2. Évaluation

Rubriques	Points
Clarté et présentation du rapport	10
Réponses aux questions (voir le détail)	90
<i>Total</i>	100

Critères d'évaluation:

1. Qualité de la modélisation et de la BD correspondante
2. Qualité du code SQL
3. Qualité du code de l'application
4. Exécution fluide de l'application avec gestion de toutes les exceptions (on ne devrait pas avoir à redémarrer l'application pour cause de bogues). Attention, une pénalité allant jusqu'à 15 points pourrait être appliquée si ce n'est pas le cas.
5. Professionnalisme de l'application et du rapport

3. Environnement et outils nécessaires

Vous devez utiliser :

- Le logiciel Visual Studio pour la modélisation UML
- Le SGBD PostgreSQL
- Angular (version 4 et +) et NodeJS pour l'application Web.

4. Etude de cas

Une organisation appelée VetoSansFrontieres fournit des soins de santé privés pour animaux domestiques à travers le Canada. Ce service est offert dans diverses cliniques situées dans les principales villes canadiennes. Le directeur de VetoSansFrontieres craint qu'il y ait un manque de communication au sein de l'organisation, notamment dans le partage de l'information et des ressources à travers les différentes cliniques. Pour résoudre ce problème, le directeur vous demande la création d'un système de base de données centralisée. Le directeur vous fournit la description suivante.

VetoSansFrontieres dispose de nombreuses cliniques vétérinaires situées dans les principales villes canadiennes. Les détails de chaque clinique comprennent le numéro de la clinique, l'adresse de la clinique (constituée de la rue, ville, province et code postal), ainsi que le numéro de téléphone et de télécopieur. Chaque clinique dispose d'un gestionnaire et d'un certain nombre d'employés (par exemple, les vétérinaires, les infirmières, les secrétaires, le personnel d'entretien). Le numéro de la clinique est unique dans l'organisation.

Les détails stockés sur chaque membre du personnel comprennent le numéro du personnel, le nom (nom et prénom), adresse, numéro de téléphone, date de naissance, sexe, numéro de sécurité sociale (NAS), la fonction et le salaire annuel actuel. Le numéro de personnel est unique dans l'organisation.

Quand un propriétaire d'animal contacte une clinique de VetoSansFrontieres pour la première fois, les détails du propriétaire de l'animal sont enregistrés, et comportent un numéro de propriétaire, le nom du propriétaire, son adresse et numéro de téléphone. Le numéro de propriétaire est unique à une clinique particulière. Les informations de l'animal sont également enregistrées, avec un numéro d'animal, un nom, un type (ex : chien, chat, ...), une espèce (ex : berger allemand), une taille, un poids, une description, une date de naissance, une date d'inscription à la clinique, l'état actuel (vivant / décédé), et le propriétaire.

Le numéro des animaux de compagnie est unique à une clinique particulière. Notons qu'un propriétaire peut avoir plusieurs animaux dans la même clinique.

Quand un animal malade est amené à une clinique, le vétérinaire en service procède à un examen de l'animal. Les détails de chaque examen sont enregistrés et comprennent un numéro d'examen, la date et l'heure de l'examen, le nom du vétérinaire, le numéro de l'animal et une description complète des résultats de l'examen. À la suite de l'examen, le vétérinaire peut proposer plusieurs traitements pour l'animal.

VetoSansFrontieres dispose de divers traitements. Ces traitements sont fournis à un taux standard à travers toutes les cliniques. Les détails de chaque traitement comprennent un numéro de traitement, une description complète du traitement, et le coût du traitement. Par exemple, les traitements incluent:

T110	Traitement à la Pénicilline	\$50.00
T112	Vaccination contre la grippe	\$70.00

Un coût de 20,00 \$ est exigé pour chaque examen, qui est enregistré comme un traitement. Le numéro de traitement identifie de manière unique chaque type de traitement et est utilisé par toutes les cliniques de VetoSansFrontieres.

Basé sur les résultats de l'examen d'un animal malade, le vétérinaire peut proposer un ou plusieurs traitements. Pour chaque traitement, l'information enregistrée comprend le numéro et la date d'examen, le numéro de l'animal de compagnie, le numéro de traitement, la quantité de chaque traitement et la date de début et de fin du traitement.

On veut ensuite pouvoir produire une facture au propriétaire, suite à l'ensemble des traitements et examens effectués. On veut ainsi pouvoir générer une facture comprenant les informations du propriétaire et de l'animal, le vétérinaire, la date de la facture, la liste des traitements, ainsi que le total à payer. Vous devez aussi mettre en place une manière de vous assurer que les factures ont été payées ou non, et via quel moyen (carte de crédit, argent comptant, chèque).

5. Travail à faire (100 points)

- 1) Lisez attentivement l'étude de cas. Etablissez vos hypothèses. Discutez-en avec votre chargé de laboratoire. Assurez-vous de bien modéliser ce qui est demandé.
- 2) Proposez un modèle entités-associations (ou entités associations étendu) permettant de répondre aux besoins exprimés ci-dessus et en tenant compte des requêtes auxquelles vous devez pouvoir répondre. N'oubliez aucun composant du modèle. Utilisez le logiciel Visio pour créer le modèle en notation UML. **(15 points)**
- 3) Convertissez votre modèle EA/EAE en modèle relationnel. Listez vos tables, leurs dépendances fonctionnelles et indiquez la forme normale de la base de données relationnelle obtenue. Expliquez en détail et selon une démarche rigoureuse comment vous parvenez à votre conclusion sur la forme normale de votre base de données. **(5 points)**
- 4) Créez la base de données PostgreSQL correspondante. Indiquez vos clés primaires et étrangères. N'oubliez aucune contrainte nécessaire dans votre modèle (exemple : intégrité référentielle, valeurs non nulles, etc.). Notez que votre script doit être fait à la main et non généré avec PGADMIN. Enregistrez votre code SQL dans **bdschema.sql**. **(10 points)**
- 5) Entrez des données dans la base de données et enregistrez vos données dans **data.sql**. Notez que votre script doit être fait à la main et non généré avec PGADMIN. Assurez-vous d'avoir des données qui permettent de répondre aux requêtes tel qu'indiqué dans la liste des requêtes à implanter. **(5 points)**
- 6) Créez les requêtes ci-dessous et enregistrez-les dans un fichier **query.sql**. Notez que chaque requête SQL doit être précédée par un commentaire contenant la requête en Français. **(20 points)**

- 7) Créez un trigger dans un fichier *trigger.sql* qui sauvegarde dans une table historique tous les traitements et examens effectués par un vétérinaire donné sur un animal donné lorsque ce dernier est supprimé **(5 points)**
- 8) Créez une application Web pour que l'utilisateur puisse directement interroger la base de données (voir la section Application Web). **(30 points)**
- 9) Ecrivez un rapport nommé *matricule1_matricule2_TP3_H21.pdf*. Voir la section *Rapport*. **(10 points)**

Liste des requêtes à implanter

- 1) Lister les le numéro et nom des cliniques, leur adresse et leur gestionnaire, ordonnés par le numéro de clinique
- 2) Quels sont les noms des employés de plus de 40 ans ordonnés par nom ?
- 3) Lister les noms des animaux dans toutes les cliniques ainsi que le nombre de fois où ils apparaissent. Par exemple Charlie, 3
- 4) Lister les numéros et noms des propriétaires d'animaux ainsi que les détails de leurs animaux dans une clinique donnée (à vous de la choisir)
- 5) Lister l'ensemble des examens d'un animal donné en utilisant sa clé primaire
- 6) Lister le détail des traitements d'un animal suite à un examen donné
- 7) Lister le salaire total des employés par clinique ordonné par numéro de clinique
- 8) Lister le nombre total d'animaux par type dans chaque clinique. Par exemple : C1, chat, 40.
- 9) Lister le coût minimum, maximum et moyen des traitements
- 10) Quels sont les propriétaires dont le nom contient « blay » ?
- 11) Supprimez le vétérinaire « Jean Tremblay » qui travaille dans la clinique dont l'identificateur est C01.
- 12) Lister les détails des propriétaires qui ont un chat et un chien
- 13) Lister les détails des propriétaires qui ont un chat ou un chien
- 14) Lister les détails des propriétaires qui ont un chat mais pas de chien vacciné contre la grippe (la condition *vacciné contre la grippe* ne s'applique qu'aux chiens)
- 15) Lister tous les animaux d'une clinique donnée avec leurs traitements s'ils existent. Dans le cas contraire, affichez null.

Application Web

Vous devez programmer une application Web en utilisant le stack Node et Angular. Votre application Web doit permettre, au moyen d'une interface, d'insérer, supprimer, modifier et d'interroger les données de votre base de données. En particulier, vous devez permettre :

- 1) D'insérer, supprimer ou de modifier les informations d'un animal. Notez que s'il y a une clé étrangère, comme par exemple un propriétaire, vous devez permettre de sélectionner cette information (par exemple dans une « dropdown list ») et non de l'introduire à la main. Cette liste doit être générée à partir des informations disponibles dans la base de données ; **(12 points)**
- 2) De voir la liste des traitements d'un animal particulier à partir de sa clé primaire ; **(5 points)**

- 3) De rechercher les informations d'un animal en utilisant son nom ou une partie de son nom. Notez que vous pouvez avoir plus d'un résultat pour ces recherches et que vous devez le gérer adéquatement – on doit pouvoir voir tous les résultats correspondants; **(5 points)**
- 4) De générer la facture totale des traitements d'un animal donné ; **(8 points)**

Rapport

Votre rapport doit contenir les informations suivantes :

- La page de garde
- Une brève introduction résumant le projet
- Le modèle conceptuel UML incluant les hypothèses et commentaires si nécessaire
- Le modèle relationnel en syntaxe abstraite
- Les dépendances fonctionnelles et l'explication de la forme normale de votre BD tel que requis dans la liste des tâches à effectuer
- La présentation de l'application développée avec des copies d'écran permettant de démontrer toutes ses fonctionnalités. N'hésitez pas à souligner les aspects novateurs de votre application tel que l'utilisation de certains patrons de conceptions, ou un effort d'ergonomie, etc.
- Un guide d'installation et de configuration qui permette au chargé de laboratoire d'installer et d'exécuter votre application

Informations supplémentaires

- Notez que les projets doivent être faits en groupes de deux. Les projets ne respectant pas cette condition ne seront pas acceptés et mèneront à une note de 0.
- Il est fortement conseillé de compléter les étapes de la base de données et des requêtes le plus tôt possible. L'application **vous prendra du temps** et vous devrez compter sur votre effort personnel et votre expérience en programmation pour la compléter.
- Notez bien que le chargé de laboratoire doit être capable d'installer et d'exécuter votre application sans problèmes en suivant le guide d'installation et d'utilisation que vous préparerez (voir section Rapport).
- Notez également que votre application doit aller ajouter/modifier/chercher les données via des requêtes SQL appropriées. En aucun cas vous ne devez effectuer des traitements sur les données au niveau de l'application (exemple en utilisant des filtres Javascript).

Description des livrables à la fin de la session

- 1) Le rapport matricule1_matricule2_TP3.pdf
- 2) Un modèle relationnel dans un fichier bdschema.sql qui permet de créer votre base de données
- 3) Un fichier data.sql qui ajoute des données à votre BD avec des instructions INSERT (suffisamment pour qu'il y ait au moins deux tuples dans les réponses aux requêtes)
- 4) Un fichier query.sql qui rassemble le code SQL de l'ensemble des requêtes. Notez que chaque requête doit être précédée par un commentaire indiquant le texte de la requête (référez-vous à la liste des requêtes)
- 5) Un fichier trigger.sql
- 6) Le code de votre application Web, soit :

- a. Le code de votre serveur Node dans un répertoire server
- b. Le code de votre client Angular dans un répertoire client
- c. Un ReadMe pour installer et lancer votre application Web

Modalités de remise

Vous devez soumettre sur Moodle à la fin du cours (la date exacte sera spécifiée sur Moodle) un fichier nommé `matricule1_matricule2_TP3.zip` qui contient tous les livrables demandés.