



INF3710 –Fichiers et Bases de données

Hiver 2023

TP No. 4

Groupe 4

Legault, Juliette – 2085041

Ly, Gia-Sherwin – 2137375

Mili, Haroun – 2144744

Soumis à : Charles De Lafontaine.

14 avril 2023

0) Introduction :

Le projet consistait à modéliser une base de données SQL pour une compagnie de location de voitures de co-voiturage. Une application qui permet de voir la liste des membres et des réservations a également été développée. Les contributeurs du projet sont Juliette Legault, Gia-Sherwin Ly et Haroun Mili.

1) Hypothèses :

On suppose que la compagnie de covoiturage est basée au Québec. Les plaques d'immatriculation peuvent donc contenir de 5 à 7 caractères.

On suppose aussi que le numéro d'un permis de conduire peut être de taille variable en fonction de sa provenance.

On suppose que l'identifiant d'un compte bancaire est unique peu importe la banque.

On suppose que l'identifiant d'une assurance est unique à travers toutes les compagnies d'assurance

On suppose qu'une assurance peut être annulée le même jour qu'elle débute. Elle est alors ajoutée dans l'historique.

On suppose qu'un même compte bancaire peut être associé à plusieurs membres de la coopérative.

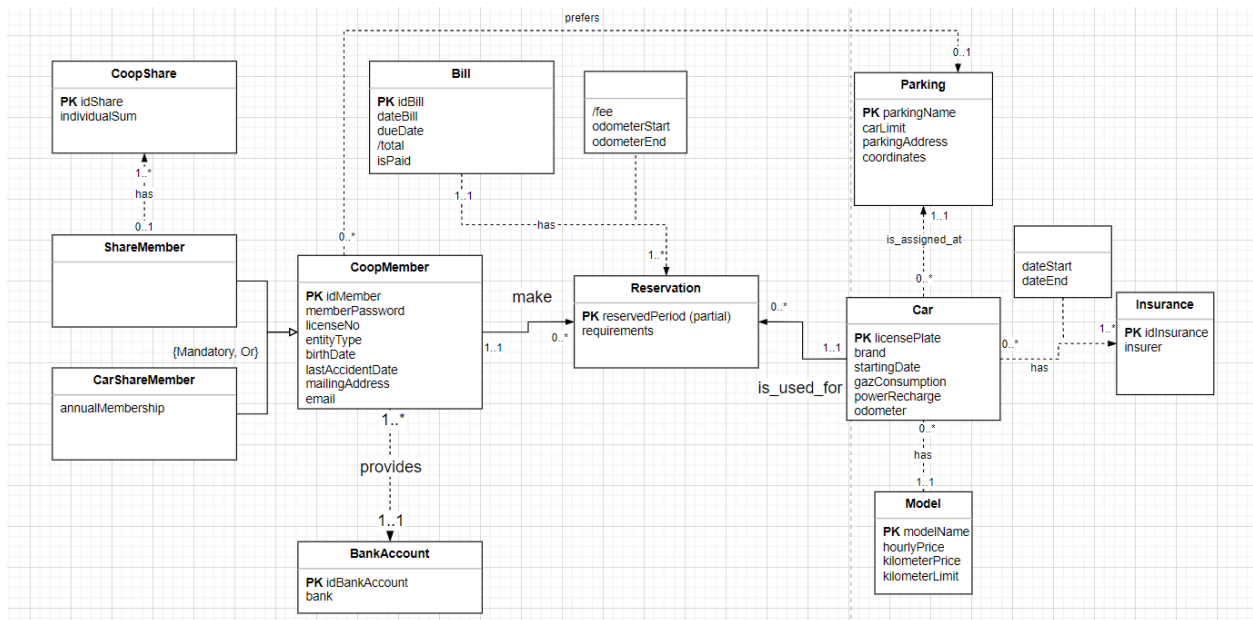
On suppose que les membres qui achètent des parts de la coopérative doivent aussi obligatoirement fournir un email et une adresse de livraison.

On suppose que les membres de la coopérative qui détiennent des parts n'ont pas besoin de payer une cotisation pour louer un véhicule.

On suppose que chaque conducteur peut réserver plusieurs voitures pour une même période.

On suppose qu'un membre se doit de réserver un véhicule un jour à l'avance à cause de délais de traitements et de vérification.

2) Modèle entité association étendu



3) Voir le fichier bdschema.sql pour la création de la base de données

4) Voir le fichier data.sql pour l'insertion des données

À noter que pour insérer les données dans la base de données sans erreurs de console, il faut exécuter les fichiers dans l'ordre suivant :

1. bdschema.sql
2. trigger.sql
3. data.sql

5) Voir le fichier query.sql pour les requêtes

6) Voir le fichier trigger.sql pour l'implémentation du trigger checkForCarSale

À noter que ce fichier contient plusieurs autres triggers afin de bien insérer les données et de calculer les différents attributs dérivés.

7) Modèle relationnel et dépendances fonctionnelles

Parking (parkingName, coordinates, parkingAddress, carLimit)

PK: parkingName

DF1: parkingName -> coordinates, parkingAddress, carLimit

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

Model (modelName, hourlyPrice, kilometerPrice, kilometerLimit)

PK: modelName

DF1: modelName -> hourlyPrice, kilometerPrice, kilometerLimit

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

Car (licensePlate, parkingName, modelName, brand, startingDate, gazConsumption, odometer, powerRecharge)

PK: licensePlate

FOREIGN KEY (modelName) REFERENCES Model(modelName)

FOREIGN KEY (parkingName) REFERENCES Parking(parkingName)

DF1: licensePlate - > parkingName, modelName, brand, startingDate, gazConsumption, odometer)

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

Insurance(idInsurance, insurer)

PK: idInsurance

DF1: idInsurance - > insurer

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

InsuranceHistory(licensePlate, idInsurance, dateStart, dateEnd)

PK: (licensePlate, idInsurance)

DF1: licensePlate, idInsurance -> dateStart, dateEnd

FOREIGN KEY (licensePlate) REFERENCES Car(licensePlate)

FOREIGN KEY (idInsurance) REFERENCES Insurance(idInsurance)

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

CoopMember(idMember, idBankAccount, memberName, preferredParking, memberPassword, entityType, birthdate, lastAccidentDate, mailingAddress, email)

PRIMARY KEY (idMember),

FOREIGN KEY (idBankAccount) REFERENCES BankAccount(idBankAccount),

FOREIGN KEY (preferredParking) REFERENCES Parking(parkingName)

DF1: idMember -> idBankAccount, memberName, preferredParking, memberPassword, entityType, birthdate, lastAccidentDate, mailingAddress, email

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

ShareMember(idMember)

PRIMARY KEY (idMember),

FOREIGN KEY (idMember) REFERENCES CoopMember(idMember)

DF1: idMember

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

CoopShare(idShare, idMember, individualSum)

PRIMARY KEY (idShare),

FOREIGN KEY (idMember) REFERENCES ShareMember(idMember)

DF1 : idShare-> idMember, individualSum

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

CarShareMember(idMember, annualMembership)

PRIMARY KEY (idMember),

FOREIGN KEY (idMember) REFERENCES CoopMember(idMember)

DF1: idMember -> annualMembership

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

Bill(idBill, dateBill, total, dueDate, dateBill, isPaid)

PRIMARY KEY (idBill)

DF1: idBill-> dateBill, total, dueDate, dateBill, isPaid

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

Il n'y a aucune dépendance multivaluée non triviale et la table est en BCNF, donc la table est en 4NF.

Reservation(reservedPeriod, idMember, licensePlate, requirements, idbill, fee, odometerStart, odometerEnd)

DF1: reservedPeriod, idMember, licensePlate -> requirements, idbill, fee, odometerStart, odometerEnd

PRIMARY KEY (reservedPeriod, idMember, licensePlate),

FOREIGN KEY (idMember) REFERENCES CoopMember(idMember),

FOREIGN KEY (licensePlate) REFERENCES Car(licensePlate),

FOREIGN KEY (idBill) REFERENCES Bill(idBill)

Les valeurs sont atomiques, donc la table est en 1NF.

Il y a seulement des dépendances fonctionnelles complètes et la table est en 1NF, donc la table est en 2NF.

Il n'y a aucune dépendance transitive et la table est en 2NF, donc la table est en 3NF.

Tous les déterminants sont des clés candidates et la table est en 3NF, donc la table est en BCNF.

On a idMember->> licensePlate et id->>reservedPeriod, deux dépendances multivaluées non triviales. La table est donc pas en 4NF, elle est en BCNF.

Conclusion sur la base de données :

Tout d'abord, on constate que dans chaque table de la base de données, les attributs de chaque table ont des valeurs atomiques. On peut donc conclure que la BD est en 1NF.

Ensuite, on constate que dans chaque table de la base de données, on a une dépendance fonctionnelle complète avec aucune DF partielle. On peut donc conclure que la BD est en 2NF puisqu'elle est en 1NF et qu'il y a une dépendance fonctionnelle complète avec aucune DF partielle.

Dans chaque table de la base de données, aucun attribut n'est transitivement dépendant de la clé primaire. Ainsi, on peut conclure que la base de données est en 3NF puisque chaque table est en 2NF et qu'aucun attribut n'est transitivement dépendant de la clé primaire.

Dans chaque table de la base de données, on constate que chaque déterminant est une clé candidate. Ainsi, on peut conclure que la base de données est en BCNF puisque chaque table est en 3NF et chaque déterminant est une clé candidate.

Dans la table Reservation, on constate qu'on a deux dépendances multivaluées non triviales. En effet, on a idMember->> licensePlate et id->>reservedPeriod. Ainsi, la base de données n'est pas en 4NF, mais elle est en BCNF.

8) Présentation de l'application



Voici la page principale de l'application Web. À travers la barre de navigation à droite, on peut naviguer entre différentes pages pour voir la liste des membres, afficher les réservations et remplir un formulaire pour réserver une voiture.

Chercher par nom	
Numéro du membre: 1 Nom du membre: Gia-Sherwin Ly Emplacement préféré: Gare d'autocars de Sainte-Catherine Mot de passe: weak password Numéro de permis: L123456789101 Numéro de banque: 123456 Type de personne: Physique Date de naissance: 2002-02-21T05:00:00.000Z Dernier accident: Adresse postale: (1234,'Rue Sherbrooke',Montréal(D1E2F3)) Email: gia@hotmail.com Membre de l'autopartage: Non Cotisation annuelle: N/A Membre de la coopérative: Oui Liste des parts: Part 1 # Montant individuel: 112	Numéro du membre: 2 Nom du membre: Haroun Milli Emplacement préféré: Gare d'autocars de Saint-Sauveur Mot de passe: better password Numéro de permis: L101987654321 Numéro de banque: 234567 Type de personne: Physique Date de naissance: 1997-04-10T04:00:00.000Z Dernier accident: 2012-11-01T04:00:00.000Z Adresse postale: (2374,'Rue Sainte-Catherine',Montréal(E1F2G3)) Email: haroun@hotmail.com Membre de l'autopartage: Oui Cotisation annuelle: 75 Membre de la coopérative: Non Liste des parts: N/A
Numéro du membre: 3 Nom du membre: Juliette Legault Emplacement préféré: Gare d'autocars de Champlain Mot de passe: best password Numéro de permis: L909076978456 Numéro de banque: 345678 Type de personne: Physique Date de naissance: 2001-11-01T05:00:00.000Z Dernier accident: Adresse postale: (5665,'Rue de Carillon',Québec,J9H8N9) Email: juliette@hotmail.com Membre de l'autopartage: Oui Cotisation annuelle: 100 Membre de la coopérative: Non Liste des parts: N/A	Numéro du membre: 4 Nom du membre: Polytechnique Emplacement préféré: Parking Poly Mot de passe: polipoly Numéro de permis: L909076978050 Numéro de banque: 111111 Type de personne: Morale Date de naissance: N/A Dernier accident: 2012-11-01T04:00:00.000Z Adresse postale: (2500,'Chem. de Polytechnique',Montréal(H3T1J4)) Email: poly@hotmail.com Membre de l'autopartage: Oui Cotisation annuelle: 100 Membre de la coopérative: Non Liste des parts: N/A

Voici la page où on peut voir la liste des membres avec leurs informations.

pO

Chercher par nom

Numéro du membre: 4

Nom du membre: Polytechnique

Emplacement préféré: Parking Poly

Mot de passe: polipoly

Numéro de permis: L9090769078050

Numéro de banque: 111111

Type de personne: Morale

Date de naissance: N/A

Dernier accident: 2012-11-01T04:00:00.000Z

Adresse postale: (2500,"Chem. de Polytechnique",Montréal,H3T1J4)

Email: poly@hotmail.com

Membre de l'autopartage: Oui

Cotisation annuelle: 100

Membre de la coopérative: Non

Liste des parts: N/A

Voici la barre de recherche qui permet de chercher un membre en fonction de son nom ou d'une partie de son nom. Il est à noter que la recherche n'est pas sensible à la case pour faciliter l'expérience utilisateur en cas d'erreur de frappe sur une lettre en majuscule.

<div> <div>Créer une réservation</div> </div>			
Période de réservation: ("2023-03-27 12:00:00";2023-03-27 12:30:00") Numéro membre: 1 Plaque: COOKE Requis: Frais: 10.00 Odomètre début: 1000 Odomètre fin:		Période de réservation: ("2023-03-27 14:00:00";2023-03-27 18:00:00") Numéro membre: 1 Plaque: 000002 Requis: Frais: 72.00 Odomètre début: 0 Odomètre fin:	
Période de réservation: ("2023-03-25 09:00:00";2023-03-25 12:00:00") Numéro membre: 1 Plaque: G010A2 Requis: Frais: 66.00 Odomètre début: 30 Odomètre fin:		Période de réservation: ("2023-01-23 09:00:00";2023-01-23 12:00:00") Numéro membre: 1 Plaque: G010A2 Requis: Frais: 66.00 Odomètre début: 30 Odomètre fin:	

Items per page: 41 ~ 4 of 7<>

Voici la page qui permet d’afficher les informations de chacune des réservations. La page possède des flèches qui permet de traverser les fiches de réservations par 4.

Emplacement

Choisissez un emplacement

Date de réservation

Choisissez une date : JJ/MM/AAAA *
14/04/2023

Heure de début:

--:--

Date de réservation

Choisissez une date : JJ/MM/AAAA *
14/04/2023

Heure de fin:

--:--

Voitures disponibles

Choisissez un véhicule

Membres pouvant conduire

1 : Gia-Sherwin Ly

Exigences supplémentaires

(Optionnel)

Envoyer

Voici la page qui permet d’effectuer une réservation pour une voiture. On constate qu’il existe des valeurs par défaut.

Emplacement	
Choisissez un emplacement ▼	
L'emplacement de location est requis	
Date de réservation	Heure de début:
Choisissez une date : JJ/MM/AAAA *	--:-- --
La date de départ est requis	L'heure de départ est requis
Date de réservation	Heure de fin:
Choisissez une date : JJ/MM/AAAA *	--:-- --
La date de retour est requis	L'heure de retour est requis
Voitures disponibles	Membres pouvant conduire
Choisissez un véhicule ▼	Choisissez un membre ▼
Veuillez choisir une plage horaire et un emplacement valide	Le choix de membre est requis
Requis	
(Optionnel)	
Envoyer	

Lorsque l'utilisateur clique sur plusieurs champs sans les spécifier, il obtient une rétroaction lui expliquant que son choix est nécessaire, et parfois, la raison pour laquelle il n'a pas de choix disponibles dans la liste déroulante (dans voitures disponibles par exemple s'il n'a pas choisi de plage horaire et de parking).

La page indique à l'utilisateur quels champs doivent être remplis. Cela permet de guider l'utilisateur dans le remplissage du formulaire de réservation.

Votre réservation a été complétée avec succès!

[▸ Créer une réservation](#)[▸ Afficher les réservations](#)

Un message de succès s'affiche lorsqu'une réservation a été complétée avec succès. L'utilisateur peut décider de choisir de créer une autre réservation ou d'afficher les réservations.

À noter que pour insérer les données dans la base de données sans erreurs de console, il faut exécuter les fichiers dans l'ordre suivant :

4. **bdschema.sql**
5. **trigger.sql**
6. **data.sql**

Guide d'installation

Important : Il faut exécuter les fichiers bdschema.sql, trigger.sql et data.sql dans l'ordre. Sinon, l'application ne pourra pas fonctionner correctement.

Consignes d'installation et démarrage

Veuillez vous assurer d'avoir installer PostgreSQL (la version 15.1.1 est utilisé). Veuillez vous assurer d'avoir installer NodeJS (la version ^18.15.5)

À partir de la racine du projet:

1- Aller dans le sous répertoire /server et lancer `npm install` pour installer les dépendances.

2- Aller dans le sous répertoire /client et lancer `npm install` pour installer les dépendances.

Dans /server/app/services/database.service.ts, il y a un attribut connectionConfig.

```
public connectionConfig: pg.ConnectionConfig = {  
  user: "postgres",  
  database: "TP4",  
  password: "root",  
  port: 5432,      // Attention ! Peut aussi être 5433 pour certains utilisateurs  
  host: "127.0.0.1",  
  keepAlive: true  
};
```

Modifier le user, database et password en conséquence.

Il est nécessaire de lancer pgAdmin4 par la suite et de populer la DB TP4 en exécutant les scripts bdschema.sql, trigger.sql puis data.sql dans. Ces scripts sql se trouvent dans server/app/db.

Dans /server, il faut exécuter la commande `npm start`, cela lancera le serveur sur le port `localhost:3000`.

Dans /client, il faut exécuter la commande `npm start`, cela lancera le client sur le port `localhost:4200`.

Contribution

2085041: 33.33%

2137375: 33.33%

2144744: 33.33%