INFO-H303 - Projet - Partie 1

Wets Loukas, Alfaro Perez Victor, Muller Noëmie

12 mai 2019

1 Modèle entité-association

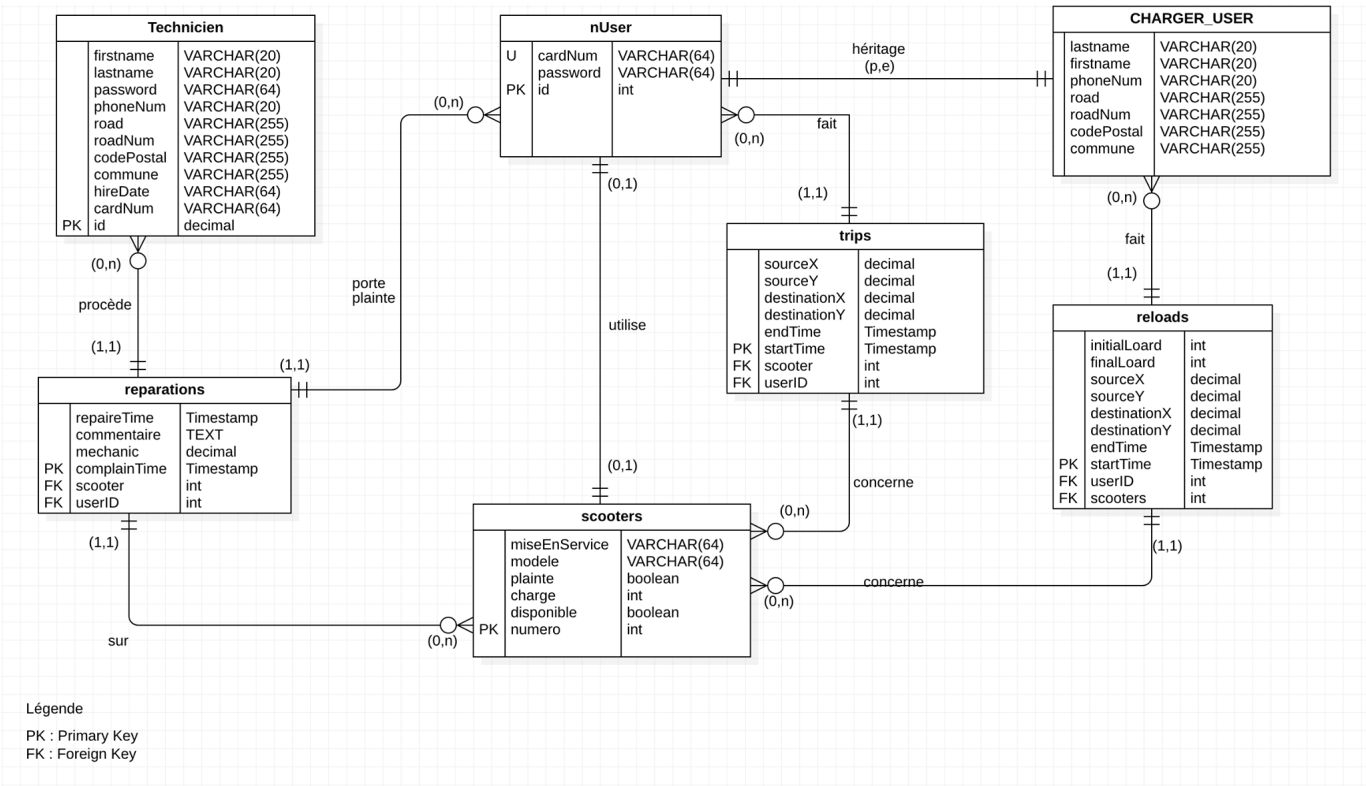

FIGURE 1 – Modèle entité-association

Contraintes d'intégrité

- La *position initiale* de la *trottinette* doit être à Bruxelles.
- La *position finale* de la *trottinette* doit être un numéro unique.
- L'*indentifiant* de chaque *trottinette*, de chaque *utilisateur* et de chaque *technicien* doit etre un numéro unique.
- Les *trottinettes* ne doivent pas être tout à fait chargées pour être rechargées.
- La *recharge* doit être éffectué entre 22h et 7h du matin.

- L'*inspection* des trottinettes doit être effectué entre 22h et 7h du matin.
- Une *trottinette* ne peut être utilisé que par un *utilisateur* à la fois.
- Une *trottinette* ne peut être réparé que par un *technicien* à la fois.
- Une *trottinette* ne peut être rechargé que par un *utilisateur qui recharge* à la fois.
- Une *trottinette* ne peut pas être *utilisé* et être en même temps en train de *chargé* ou en train d'être *réparé*.
- La *Date de réparation* doit être supérieur à la *date de dépot de plainte*.
- La *Date de mise en service* doit être inférieur à la *date de dépot de plainte*.
- La *Date d'intervention* doir être supérieur ou égale a la *date de dépot de plainte*.

2 Modèle Relationnel

Trottinette(N°Trottinette, DateMiseService, N°Modèle, Etat, Identifiant)

Utilisateur(Identifiant, MotDePasse, N°Carte)

Technicien(N°Employé, Nom, Prenom, N°Téléphone, DateEmbauche

Trajet (HeureDeVerrouillage, N°Trottinette, Identifiant, HeurVerouillage, PositionInitiale, PositionFinale)
N°Trottinette référence Trottinette.N°Trottinette
Identifiant référence Utilisateur.Identifiant

UtilisateurQuiRecharge(Identifiant, Nom, n°Téléphone, AdresseRue, AdresseNumero, AdresseCommune, AdresseCodePostale)
Identifiant référence Utilisateur.idnetifiant

Recharge(Identifiant, N°Trottinette, Date, ChargeInitiale, ChargeFinale, Position, Heure)
Identifiant référence UtilisateurQuiRecharge.Identifiant
N°Trottinette référence Trottinette.n°Trottinette

Intervention(N°Trotinette, N°Employe, DateReparation, DateDéppotPlainte, Utilisateur, Note, RetirerTrottnette)
N°Trottinette référence Trottinette.N°Trottinette
N°Employé référence Technicien.N°Trottinette

TrottinetteUtilisateur(N°Trottinette, Identifiant)
N°Trottinette référence Trottinette.N°Trottinette
Identifiant référence Utilisateur.Identifiant

UtilisateurTechnicien(Identifiant, N°Employé)
Identifiant référence Utilisateru.Identifient
N°Employe référence Technicien.N°Employé

Contraintes du modèle relationnel

Trottinette.DateMiseEnService <= Intervention.DateDeDépot
Intervention.DateRéparation >= Intervention.DateDeDépotPlainte

3 Requêtes algèbre relationnel

-R1 : $\pi_{scooter,destinationX,destinationY}(\sigma_{endTime>=endTime}(Trips) \cap \sigma_{disponible='t'}(Scooter))$

-R2 : $\sigma_{scooter=scooter}(Reloads \cap Trips)$

4 Requêtes calcul relationnel tuple

-R1 {trips.scooter, trips.destinationX, trips.destinationY | Trips(trips)$\wedge scooters(scooter) \wedge trips.endTime >= \forall trips.endTime \wedge scooter.numero = trips.scooter \wedge scooter.disponible =' t'$}

-R2 {i.userID | Reloads(r)$\wedge i(Reloads(r) \wedge Trips(t) \wedge r.scooter = t.scooter) \wedge \forall i.scooter = \forall r.scooter$}

5 Choix d'implémentation

5.1 Méthode d'extraction des données

L'extraction des données ce fait à partir d'un programme python extractData.py ce trouvant dans le dossier base de donnée qui sert donc a extraire les donnée des fichiers contenant les données a inseré dans la base de donnée. L'extraction des donnée se fait de deux manières différentes l'une pour les fichiers .xml dont la fonction xml.etree.ElementTree est utilisé pour extraire les données. Pour les fichiers .csv on ouvre simplement le fichier et on extrait les donnée a l'aide de la fonction split en fonction du caractère utilisé pour les séparations. Le programme s'occupe ensuite d'écrire les inserts automatiquement en fonction des données à introduire en prenant compte les noms avec des "'" dedans sensible a la casse. Toutes ces requêtes sont ensuite inséré dans le fichier data.sql.

5.2 Outils utilisé

Serveur

Le serveur est un serveur python facile à utilisé nommé Flask, plutôt intéressant car facile à utilisé et offre largement assez d'outils pour ce dont nous avons besoin. Session pour contenir les données du client connecté, request pour récupérer les données des forms et render_template pour générer les pages html (dont les fichiers ce trouvent dans le dossier templates) avec les résultats des requêtes sql. Pour lancer le serveur il vous faut exécuter la commande $python2 server.py (nom de votre base de donnée) (nom du propriétaire de la base de donnée).

Base de donnée

Pour le base de donnée on a choisit postgreSQL car la sémantique est facile à comprendre et agréable a utilisé. Il possède également un large épouvantail de type de données. La création des tables ce fait a l'aide du fichier init.sql et l'insertion des données de la consigne ce fait à l'aide du fichier data.sql. Ces fichiers doivent être exécuté à l'aide de la commande $psql -d (nom de votre base de donnée) -f (nom du fichier). Pour exécuté les requêtes et récupéré les données ou les commits on utilise la librairie psycopg2 qui offre une bonne interface pour communiqué avec posgreSQL.

6 Hypothèses

-Un mécanicien ne peut pas être un utilisateur et vice versa.
-Une plainte ne peut pas être introduite si il y a déjà une plante en cours.
-Toutes les trottinettes insérer à partir des données de issue des consignes sont disponibles.

-Lorsqu'un utilisateur qui a pris une trottinette pour la recharger et la rends on considère que la trottinette est rechargé au maximum.