

Projet Conception d'une base de donnée

INF-403

Nom prénom: PONS ALFONSO Mislaidis et ROUSSEAU Lénaïc

Description en langage naturel

Une métropole a choisi d'informatiser la gestion de son réseau de trottinette électrique. Elle propose une alternative plus verte pour les petits trajets. Nous avons un magasin, avec des employés, qui gèrent la location de trottinettes. Nous visons une clientèle qui utilise la trottinette pour faire une boucle.

Chaque client est identifié par un numéro, et est décrit par un nom, prénom, date de naissance et adresse. De même pour un employé, avec ses horaires en plus.

Les trottinettes, identifiées par un numéro, et décrites par une date de fabrication et son état, pourront être louées au client, par un employé, à l'aide de la table de location.

Les locations sont identifiées par un numéro et sont décrites par une date de départ et d'arrivée.

Tant que la location n'est pas terminée, la date d'arrivée est nulle.

Un client peut louer plusieurs trottinettes à la fois. Un seul employé s'occupe de faire cette location.

Chaque trottinette aura un certain modèle, identifié par un nom, avec couleur, vitesse et prix de modèle et un prix par minute, et une certaine batterie, identifiée par un numéro.

Toutes les batteries ne seront pas forcément sur une trottinette, et elles auront un certain niveau de charge (petit, moyen, grand).

Règles de traduction de nom

Nom classe au singulier -> nom table au pluriel

CamelCase -> snake_case

attribut -> attribut_nom_classe

Modèle relationnel

Clients (numero_client, nom_client, prenom_client, date_naissance_client, adresse_client)

/* <nc,n,p,d,a> ∈ Clients ⇔ Le numéro client nc, identifié par son nom n et prénom p, est né le d et habite à l'adresse a. */

Employes (numero_employe, nom_employe, prenom_employe, date_naissance_employe, adresse_employe, horaire_employe)

/* <ne,n,p,d,a,h> ∈ Employes ⇔ Le numéro employé ne, identifié par son nom n et prénom p, est né le d et habite à l'adresse a. Il a des horaires h. */

Locations (numero_location, date_dep_location, date_arr_location, numero_client, numero_employe)

/* <nl,dd,da,nc,ne> ∈ Locations ⇔ Une location nl, faite à une date dd et terminée à une date da. Location effectuée par un client nc et un employé ne. */

LocationsTrottinettes (numero_location, numero_trottinette)

/* <nl, nt> ∈ LocationsTrottinettes ⇔ Une location nl contient nt trottinettes et une trottinette nt possède nl locations. */

Trottinettes (numero_trottinette, date_fabrication_trottinette, etat_trottinette, nom_modele, numero_batterie)

/* <nt,nb,nm, d, e> ∈ Trottinette ⇔ La trottinette nt, avec le numéro de batterie nb et le nom du modèle nm et a été fabriquée à la date df et à un état e. */

Modeles (nom_modele, couleur_modele, vitesse_modele, prix_par_minutes_modele, prix_modele)

/* <nm, c, v, ppm, pm> ∈ Modele ⇔ La trottinette du modele nm, de couleur c, avec une vitesse v, est louer à partir du prix pm à un prix par minute ppm. */

Batteries (numero_batterie, charge_batterie)

/* <nb,c> ∈ Batterie ⇔ La batterie identifié par le numéro nb, a une charge c. */

Les domaines sont:

- domaine(numero_client) = domaine(numero_trottinette) = domaine(minutes_location) = domaine(numero_batterie) = domaine(numero_location) = domaine(vitesse_modele) = entier > 0
- domaine(nom_client) = domaine(prenom_client) = domaine(adresse_client) = domaine(nom_employe) = domaine(prenom_employe) = domaine(adresse_employe) = domaine(nom_modele) = domaine(adresse_dep_location) = domaine(adresse_arr_location) = chaîne de caractères
- domaine(couleur_modele) = blanc || rouge || bleu
- domaine(date_naissance_client) = domaine(date_naissance_employe) = domaine(date_fabrication_trottinette) = domaine(date_dep_location) = date
- domaine(prix_modele) = domaine(prix_location) = float > 0

Remarque: Le format d'un horaire:

- ':' separe les jours de la semaine
- '|' separe les horaires du matin et du soir
- ',' separe les heures
- Exemple: "10h30,12h00 | 13h30,17h30 : 13h30,17h30 | : | : | 13h30,22h30 : |13h30,17h30 : |: |"

Les contraintes d'intégrité référentielles:

$\text{Locations}[\text{numero_client}] \subseteq \text{Clients}[\text{numero_client}]$

$\text{Locations}[\text{numero_employe}] \subseteq \text{Employes}[\text{numero_employe}]$

$\text{LocationsTrottinettes}[\text{numero_trottinette}] \subseteq \text{Trottinettes}[\text{numero_trottinette}]$

$\text{LocationsTrottinettes}[\text{numero_location}] = \text{Locations}[\text{numero_location}]$

$\text{Trottinettes}[\text{numero_batterie}] = \text{Batteries}[\text{numero_batterie}]$

$\text{Trottinettes}[\text{nom_modele}] \subseteq \text{Modeles}[\text{nom_modele}]$

$\text{TypePourcentage} \in \{\text{petit, moyenne, grand}\}$

$\text{TypeEtat} \in \{\text{bon, moyen, mauvais}\}$

Remarques:

- Une personne ne peut louer que si elle est cliente
- Une location ne peut pas être faite si le client a une location non fini ($\text{date_arr_location} \neq \text{NULL}$)
- On peut louer plusieurs trottinettes en une seule location essentiellement pour avoir une relation many-to-many
- Nous sommes conscient que notre modèle possède des défauts: obligation de rendre la trottinette dans le magasin où elle a été prise, avec un prix en fonction du temps, ce qui rend la location très peu intéressante