



DATABASE

Projet fil rouge – 1^{er} rendu

Le 26-03-2018

QUIBEL Florian - ROBIC Gaëtan
TD N

Sommaire :

| | |
|--|---|
| Introduction : | 2 |
| Les schémas de conception et relationnel de la base de données | 2 |
| La construction des tables..... | 4 |
| Le peuplement des tables | 4 |
| Les requêtes | 4 |

Introduction :

L'objet de ce projet fil rouge est de créer une base de données pour l'agence Escapade. Cette agence propose à ses clients différents séjours de courtes durées dans Paris intra-muros avec la location d'une voiture et du logement intégrés au forfait. Il faut donc aider cette société à gérer à la fois les différents séjours mais aussi la disposition de voitures présentes dans des places de parkings réservés dans chaque arrondissement sans oublier la vérification de ses dernières après chaque fin de séjour. Nous allons alors permettre de gérer au mieux cette gestion complexe avec la mise en place d'une base de données contenant plusieurs tables et qui sera mise à jour en continu.

Les schémas de conception et relationnel de la base de données

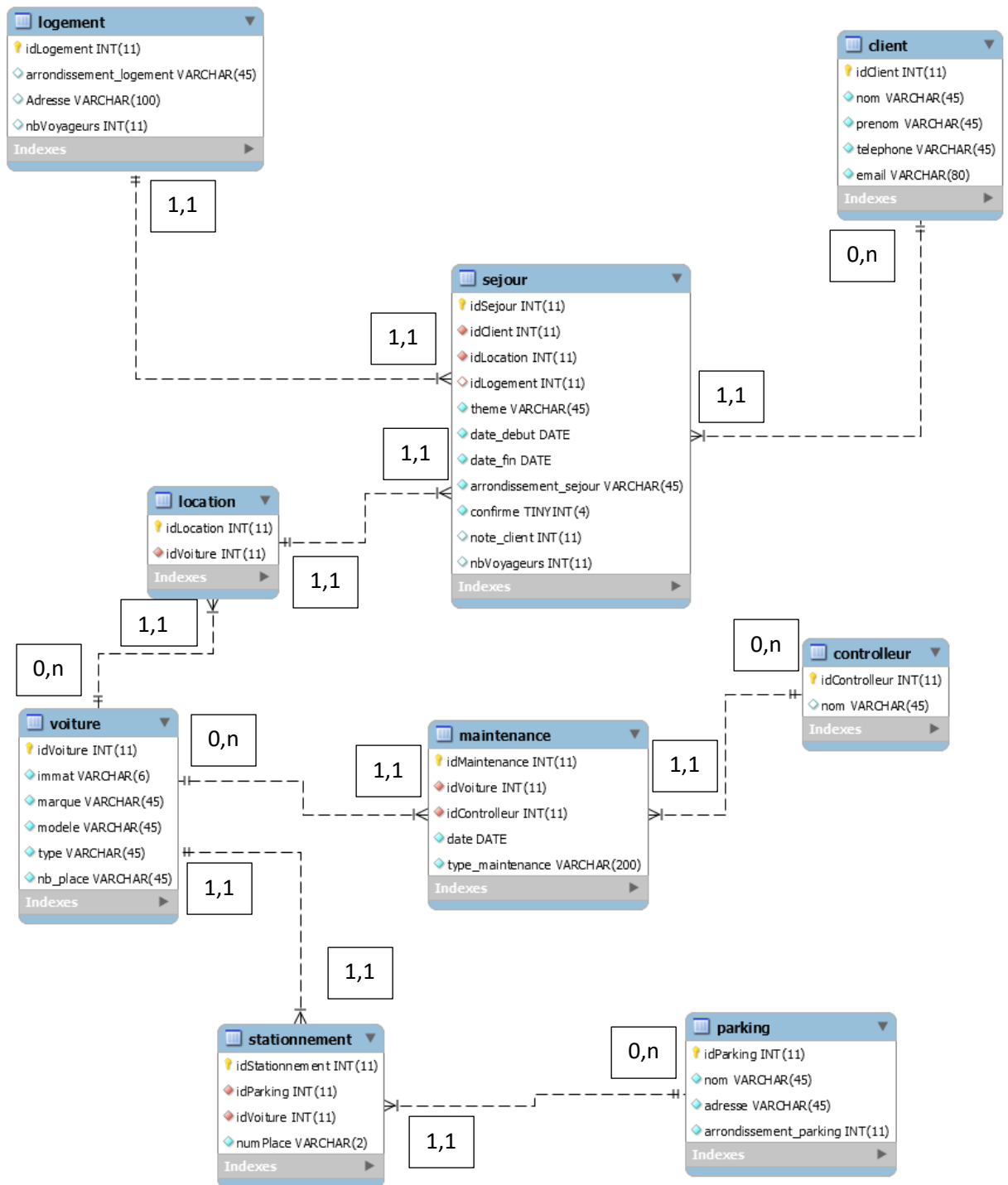
La première chose à faire a été de faire le schéma afin de mieux visualiser le problème. On a alors créé 9 tables différentes :

| | | | | |
|----------------|----------|--------------|--------------|-----------|
| _Client | _Sejour | _Location | _Voiture | _Logement |
| _Stationnement | _Parking | _Maintenance | _Controlleur | |

Toutes ses tables sont reliées entre elles grâce à leurs clés d'indentification et des associations tels que idVoiture, idSejour, location, stationnement etc...

Il a ensuite fallu remplir les cardinalités entre chaque table : Une voiture ne peut être stationnée qu'à une place de parking au maximum à un instant et une place de parking n'est pas forcément utilisée mais elle ne sera jamais utilisée par plusieurs voitures en même temps.

De même il y a 3 contrôleurs qui s'occupent de l'état de toutes les voitures mais une voiture n'est vérifiée que par un seul contrôleur après un séjour.



La construction des tables

Ensuite, il a fallu créer chaque table une par une, certaines possédant une foreign-key comme c'est le cas des tables Séjour, Location, Maintenance et Stationnement. En revanche chaque table possède son propre « id » (ex : idVoiture). Chaque table possède ses propres attributs.

Le peuplement des tables

On a alors initialisé les tables, on y a ajouté les 40 voitures de la société. Pour les voitures, on a décidé de mettre des Renault Clio, des Renault Scénic et des Citroën C4 pour les berlines 4 places ainsi que des Toyota GT86 et des Audi A3 pour les cabriolets 2 places. Chaque voiture possédant sa propre plaque d'immatriculation et son propre « id ».

De même, nous avons initialisé les 3 contrôleurs de la société et les 20 parkings dans lesquels la société loue des places de stationnement.

On a aussi peuplé les tables différentes autres tables comme Client et Maintenance avec certaines personnes ou voitures selon le cas afin de pouvoir tester toutes les différentes requêtes demandées.

Les requêtes

Il fallait tester 10 différentes requêtes différentes et nous allons expliquer les résultats obtenus un par un.

#1 Liste des clients :

Il s'agit simplement d'afficher tous les clients de la société, c'est-à-dire les personnes que nous avons créé lors du peuplement de la table Client

#2 Saisie d'un nouveau client :

Il suffit d'ajouter un client, c'est-à-dire de faire un insert into comme lors du peuplement, en y mettant les informations de cette personne (nom, prénom, téléphone, email).

#3 Liste des voitures, de leur position et de leur disponibilité :

Il faut ici s'occuper de toutes les voitures, qu'elles soient garées dans un parking, louées pour un séjour ou bien en maintenance. On précise alors lors de l'affichage la position/disponibilité de la voiture.

#4 Sélection d'une voiture disponible dans un arrondissement :

Peu importe le nombre de voitures disponible dans l'arrondissement choisi (on prend arbitrairement la valeur de l'arrondissement ici), on se limite à une seule voiture grâce à « limit 1 ».

#5 Requête de mise à jour de la place de parking d'un véhicule identifié par son immatriculation :

On utilise « Update » afin de modifier la place de stationnement sur laquelle est située la voiture. Il ne s'agit pas de modifier le nom de cette place de stationnement mais de changer l'endroit où est garée la voiture correspondant à la plaque d'immatriculation donnée.

#6 Combien d'opérations de maintenance sur une voiture identifiée par son immatriculation :

On cherche ici à remonter l'historique d'une voiture. Grâce à son immatriculation on peut alors savoir l'ensemble des maintenances effectuées depuis que la voiture est dans la société.

#7 Enregistrement du retour d'une voiture :

C'est-à-dire qu'une voiture revient d'un séjour. Avant de passer à la maintenance, elle est stationnée sur une place de parking. Il faut donc la supprimer de la table des locations et la rajouter à celle des stationnements.

#8 Nombre de voitures contrôlées par chacun des contrôleurs :

On choisit pour chacun des contrôleurs les maintenances où son id est inscrit. On affiche ensuite somme des voitures contrôlées.

#9 Liste des voitures indisponible et du motif correspondant :

On a considéré qu'une voiture est indisponible si elle est en location ou bien en maintenance, il n'y a donc pas forcément de motif.

#10 Enregistrement d'une opération de maintenance par un des contrôleurs sur une voiture identifiée par son immatriculation :

Il s'agit simplement d'ajouter une nouvelle maintenance, comme lors du peuplement de la table.