**- Rapport de projet -**

**Réalisation d’une base de données pour une société de location de véhicules**

*Université Cergy Pontoise*

|  |  |
| --- | --- |
| **Auteurs** : | Babaedjou Hanyatou |
|  | Phetramphand Antoine |

Sommaire

[I. Présentation du projet 4](#_Toc437011890)

[1. Contexte 4](#_Toc437011891)

[2. Objet 4](#_Toc437011892)

[3. Organisation 4](#_Toc437011893)

[4. Environnement 4](#_Toc437011894)

[5. Planification 4](#_Toc437011895)

[II. Objectif 5](#_Toc437011896)

[1. Description brève du logiciel 5](#_Toc437011897)

[2. Le but de cette base de données 5](#_Toc437011898)

[3. Liste des fonctionnalités 5](#_Toc437011899)

[4. Contraintes 6](#_Toc437011900)

[III. Livraisons attendues 6](#_Toc437011901)

[IV. Conception du projet 7](#_Toc437011902)

[1. Schéma MCD 7](#_Toc437011903)

[2. Schéma MLD 7](#_Toc437011904)

[V. Description du schéma conceptuel 9](#_Toc437011905)

[1. Description des tables (Entités) 9](#_Toc437011906)

[a) table AGENCE 9](#_Toc437011907)

[b) table CLIENT 9](#_Toc437011908)

[c) table EMPLOYE 9](#_Toc437011909)

[d) table VEHICULE 9](#_Toc437011910)

[e) table UTILITAIRE 10](#_Toc437011911)

[f) table LOUER 10](#_Toc437011912)

[2. Description des relations entre les tables 10](#_Toc437011913)

[VI. Scripts SQL 12](#_Toc437011914)

[1. Création des tables 12](#_Toc437011915)

[a) table AGENCE 12](#_Toc437011916)

[b) table EMPLOYE 12](#_Toc437011917)

[c) table UTILITAIRE 12](#_Toc437011918)

[d) table VEHICULE 12](#_Toc437011919)

[e) table CLIENT 13](#_Toc437011920)

[f) table LOUER 13](#_Toc437011921)

[2. Attribution des clés étrangères 13](#_Toc437011922)

[3. Les requêtes 14](#_Toc437011923)

[a) requête n°1 14](#_Toc437011924)

[b) requête n°2 14](#_Toc437011925)

[c) requête n°3 15](#_Toc437011926)

[d) requête n°4 15](#_Toc437011927)

[e) requête n°5 16](#_Toc437011928)

[f) requête n°6 16](#_Toc437011929)

[g) requête n°7 16](#_Toc437011930)

[h) requête n°8 17](#_Toc437011931)

[VII. Scripts PHP 18](#_Toc437011932)

[1. Quelques illustrations 18](#_Toc437011933)

[2. Codes 20](#_Toc437011934)

[VIII. Remarques sur le déroulement du projet 22](#_Toc437011935)

[1. Méthode de travail au sein du groupe 22](#_Toc437011936)

[2. Difficultés rencontrées 22](#_Toc437011937)

[IX. Conclusion 23](#_Toc437011938)

# I. Présentation du projet

## 1. Contexte

Dans le cadre de notre formation à l’Université de Cergy Pontoise, les étudiants en Licence *Informatique* troisième année réalisent des projets personnels afin de valider leur année scolaire. Ce type de réalisation exige donc un certain temps, un investissement personnel et l’usage de toutes les ressources disponibles. Ici, nous allons réaliser une base de données pour une société de location de véhicules.

## 2. Objet

Le but est la réalisation d’une application qui consiste à créer une base de données, puis manipuler les données à l’aide du langage *SQL*.

Ce document est destiné à identifier et décrire les besoins de la direction de la société de location de véhicules.

## 3. Organisation

Notre équipe est composée de deux membres à la charge du projet : Babaedjou Hanyatou et Phetramphand Antoine.

## 4. Environnement

Les ressources que nous disposons pour le développement de notre projet sont :

* **JMerise**: logiciel permettant de réaliser notre schéma *MCD*.
* **SQL Developper** : logiciel permettant de programmer nos tables, ainsi que nos requêtes *SQL* sous Oracle.
* **Sublime Text** : logiciel permettant d’éditer nos pages *PHP*.
* **Wampserver** : plateforme de développement *Web* de type *WAMP*, permettant de faire fonctionner localement des scripts *PHP*.

Les langages utilisés sont : ***PHP*** et ***SQL****.*

Le SGBDR que nous utilisons est ***Oracle***.

## 5. Planification

**Délai** : le projet doit se finir avant le 04/12/2015.

# II. Objectif

## 1. Description brève du logiciel

L’objectif du projet consiste à créer une base de données ou une application basée qui répond aux exigences du travail demandé.

Cette étude doit permettre à l’entreprise de connaître avec précision les apports et conséquences de la migration de sa base de données dans des conditions réelle afin de pouvoir, à terme, valider ou non la mise en place d’une telle migration de ses serveurs.

## 2. Le but de cette base de données

Notre programme consiste à gérer les employés et les clients selon les critères de cette société.

## 3. Liste des fonctionnalités

Cette application permettra de réaliser les fonctionnalités suivantes :

- Lister la liste des noms et adresses de tous les clients qui ont effectué au moins une location d’une voiture et d’un véhicule utilitaire.

- Lister la liste des modèles de véhicules n’ayant fait l’objet d’aucune location de la part des entreprises, vous indiquerez également la marque de chaque véhicule.

- Afficher les numéros et les noms des clients ayant effectué au moins une location pour laquelle l’agence de restitution est différente de l’agence où le véhicule a été loué.

- Afficher les agences dans lesquelles au moins un véhicule utilitaire de chacune des marques existantes dans l’entreprise est disponible à la location au moment où la requête est exécutée.

- Afficher les noms des responsables des agences dans lesquelles il est impossible de louer un véhicule de catégorie voiture. Aucune voiture n’est disponible au moment où la requête est exécutée.- Afficher, magasin par magasin, le nom du magasin, le nom de son responsable, ainsi que le nombre de locations de plus de trois jours effectuées en 2013.

- Trouver le plus grand nombre de locations effectuées par un client et afficher les noms et adresses des clients qui ont effectué ce grand nombre de locations.

- Afficher, agence par agence, le nom de l’agence, le nom de son responsable, ainsi que le nombre de locations de plus de trois jours effectuées en 2015.

- Pour chaque véhicule de moins de 20 000km (au moment où la requête est effectuée), donner la somme totale des montants de toutes les locations effectuées par des entreprises dont il a fait l’objet au cours du mois de juillet 2015.

- Enregistrement d’une location et la restitution d’un véhicule.

## 4. Contraintes

* La base de données doit être conforme aux normes *SQL*.
* Le code d’interrogation de la base de données doit respecter les règles définies dans les normes *SQL*.

# III. Livraisons attendues

À la fin de notre projet, nous pensons rendre les documents suivants :

• des programmes en langage *SQL* comportant les tables et les requêtes demandées

• des pages *PHP*

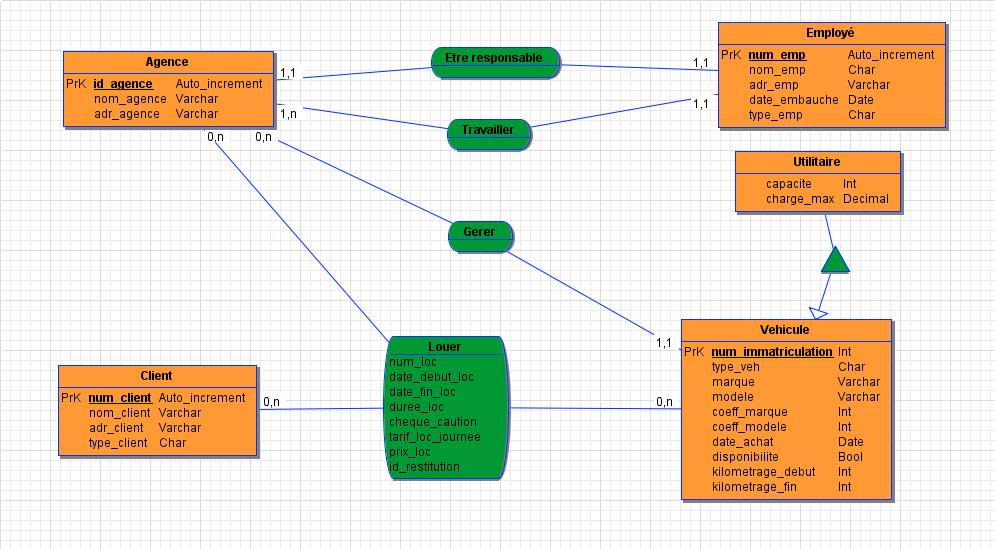
• un rapport

• le schéma *MCD*

# IV. Conception du projet

## 1. Schéma MCD

Voici le schéma MCD :



Légende :

* Les entités sont représentées en orange.
* Les relations sont représentées en vert.

## 2. Schéma MLD

Voici le schéma MLD :

**AGENCE** (id\_agence, nom\_agence, adr\_ agence, #num\_emp)

**CLIENT** (num\_cli, nom\_cli, adresse\_cli, type\_cli)

**EMPLOYE** (num\_emp, nom\_emp, adresse\_emp, date\_embauche, type\_emp, #id\_agence)

**VEHICULE** (num\_immatriculation, type\_veh, marque, modele, coeff\_marque, coeff\_modele date\_achat, disponibilite, kilometrage\_debut, kilometrage\_fin, #id\_agence)

**UTILITAIRE** (capacite, charge\_max, #num\_immatriculation)

**Louer** (num\_cli#, num\_immatriculation#, id\_agence#, num\_loc, date\_debut\_loc, date\_fin\_loc, durée\_loc, cheque\_caution, tarif\_loc\_journée, prix\_loc, id\_restitution)

# V. Description du schéma conceptuel

## 1. Description des tables (Entités)

### a) table AGENCE

* *Id\_agence* : numéro de l’agence
* *nom\_agence* : nom de l’agence
* *adr\_agence* : adresse de l’agence

### b) table CLIENT

* *num\_cli* : numéro du client
* *nom\_cli* : nom du client
* *adresse\_cli* : adresse du client
* *type\_cli* : type du client

### c) table EMPLOYE

* *num\_emp* : numéro de l’employé
* *nom\_emp* : nom de l’employé
* *adresse\_emp* : adresse de l’employé
* *date\_embauche* : date d’embauche de l’employé
* *type\_emp* : type de l’employé

### d) table VEHICULE

* *num\_immatriculation* : numéro d’immatriculation du véhicule
* *type\_veh*: type du véhicule
* *marque* : la marque du véhicule
* *modele* : le modèle du véhicule
* *coeff\_marque* : coefficient de la marque du véhicule
* *coeff\_modele* : coefficient du modèle du véhicule
* *date\_achat* : date d’achat du véhicule
* *disponibilite* : état du véhicule actuel (loué, en panne ou en réparation)
* *kilometrage\_debut* : début du kilométrage du véhicule
* *kilometrage\_fin* : fin du kilométrage du véhicule

### e) table UTILITAIRE

* *capacite* : la capacité du véhicule
* *charge\_max* : la charge maximale du véhicule

### f) table LOUER

* *num\_loc* : numéro de location du véhicule
* *date\_debut\_loc* : la date de début de location du véhicule
* *date\_fin\_loc* : la date de fin de location du véhicule
* *durée\_loc* : la durée de location exprimée en jours
* *cheque\_caution* : chèque de caution
* *tarif\_loc\_journée*: tarif à la journée de location
* *prix\_loc* : prix de location du véhicule
* *id\_restitution* : agence de restitution

## 2. Description des relations entre les tables

**Etre responsable**

Chaque employé de type responsable n’appartient qu’à une seule et une seule agence.

**Gérer**

Un véhicule est au moins restitué par une seule et une seule agence.

Une agence peut gérer soit aucune ou plusieurs véhicules.

**Louer**

Un client peut n’avoir loué aucun véhicule, mais peut en louer plusieurs véhicules dans une agence.

**Travailler**

Un employé travaille dans une et une seule agence.

Une agence regroupe un ou plusieurs employés.

**Véhicule-Utilitaire**

Un véhicule peut être au moins un utilitaire.

# VI. Scripts SQL

## 1. Création des tables

### a) table AGENCE

CREATE TABLE Agence(

id\_agence NUMBER NOT NULL ,

nom\_agence VARCHAR2 (25) ,

adr\_agence VARCHAR2 (40) ,

num\_emp NUMBER(10,0) ,

CONSTRAINT Agence\_Pk PRIMARY KEY (id\_agence)

);

### b) table EMPLOYE

CREATE TABLE Employe(

num\_emp NUMBER NOT NULL ,

nom\_emp CHAR (30) ,

adr\_emp VARCHAR2 (40) ,

date\_embauche DATE ,

type\_emp CHAR (25) ,

id\_agence NUMBER(10,0) ,

CONSTRAINT Employe\_Pk PRIMARY KEY (num\_emp)

);

### c) table UTILITAIRE

CREATE TABLE Utilitaire(

capacite NUMBER(10,0) ,

charge\_max FLOAT ,

num\_immatriculation NUMBER(10,0) NOT NULL ,

CONSTRAINT Utilitaire\_Pk PRIMARY KEY (num\_immatriculation)

);

### d) table VEHICULE

CREATE TABLE Vehicule(

num\_immatriculation NUMBER(10,0) NOT NULL ,

type\_veh CHAR (25) ,

marque VARCHAR2 (25) ,

modele VARCHAR2 (25) ,

coeff\_marque NUMBER(10,0) ,

coeff\_modele NUMBER(10,0) ,

date\_achat DATE ,

disponibilite NUMBER (1) ,

kilometrage\_debut NUMBER(10,0) ,

kilometrage\_fin NUMBER(10,0) ,

id\_agence NUMBER(10,0) ,

CONSTRAINT Vehicule\_Pk PRIMARY KEY (num\_immatriculation) ,

CONSTRAINT CHK\_BOOLEAN\_disponibilite CHECK (disponibilite IN (0,1))

);

### e) table CLIENT

CREATE TABLE Client(

num\_client NUMBER NOT NULL ,

nom\_client VARCHAR2 (25) ,

adr\_client VARCHAR2 (40) ,

type\_client CHAR (40) ,

CONSTRAINT Client\_Pk PRIMARY KEY (num\_client)

);

### f) table LOUER

CREATE TABLE Louer(

num\_loc NUMBER NOT NULL ,

date\_debut\_loc DATE ,

date\_fin\_loc DATE ,

duree\_loc NUMBER(10,0) ,

cheque\_caution FLOAT ,

tarif\_loc\_journee NUMBER(10,0) ,

prix\_loc FLOAT ,

id\_restitution NUMBER(10,0) ,

num\_client NUMBER(10,0) NOT NULL ,

num\_immatriculation NUMBER(10,0) NOT NULL ,

id\_agence NUMBER(10,0) NOT NULL ,

CONSTRAINT Louer\_Pk PRIMARY KEY (num\_client,num\_immatriculation,id\_agence)

);

## 2. Attribution des clés étrangères

ALTER TABLE Agence ADD FOREIGN KEY (num\_emp) REFERENCES Employe(num\_emp);

ALTER TABLE Employe ADD FOREIGN KEY (id\_agence) REFERENCES Agence(id\_agence);

ALTER TABLE Utilitaire ADD FOREIGN KEY (num\_immatriculation) REFERENCES Vehicule(num\_immatriculation);

ALTER TABLE Vehicule ADD FOREIGN KEY (id\_agence) REFERENCES Agence(id\_agence);

ALTER TABLE Louer ADD FOREIGN KEY (num\_client) REFERENCES Client(num\_client);

ALTER TABLE Louer ADD FOREIGN KEY (num\_immatriculation) REFERENCES Vehicule(num\_immatriculation);

ALTER TABLE Louer ADD FOREIGN KEY (id\_agence) REFERENCES Agence(id\_agence);

## 3. Les requêtes

### a) requête n°1

Afficher la liste des noms et adresses de tous les clients qui ont effectué au moins une location d’une voiture et d’un véhicule utilitaire

Select distinct c.nom\_client, c.adr\_client

from CLIENT c, LOUER l, LOUER m, VEHICULE v, VEHICULE u

WHERE c.num\_client=l.num\_client

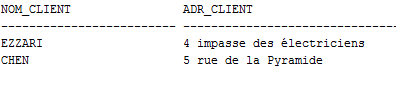
AND l.num\_client=m.num\_client

AND l.num\_immatriculation=v.num\_immatriculation

AND m.num\_immatriculation=u.num\_immatriculation

AND v.type\_veh = 'voiture'

AND u.type\_veh = 'utilitaire';



### b) requête n°2

La liste des modèles de véhicules n’ayant fait l’objet d’aucune location de la part des entreprises, vous indiquerez également la marque de chaque véhicule.

Select distinct v.marque,v.modele

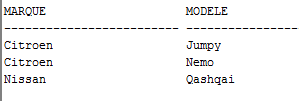
from CLIENT c, LOUER l, VEHICULE v

WHERE c.num\_client=l.num\_client

AND l.num\_immatriculation=v.num\_immatriculation

AND c.type\_client <> 'entreprise'

AND l.num\_loc is NOT NULL;



### c) requête n°3

Afficher les numéros et les noms des clients ayant effectué au moins une location pour laquelle l’agence de restitution est différente de l’agence où le véhicule a été loué.

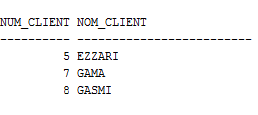
SELECT c.num\_client, c.nom\_client

FROM CLIENT c, LOUER l

WHERE c.num\_client = l.num\_client

AND l.ID\_AGENCE != l.ID\_RESTITUTION

AND l.ID\_RESTITUTION IS NOT NULL;



### d) requête n°4

Afficher les agences dans lesquelles au moins un véhicule utilitaire de chacune des marques existantes dans l’entreprise est disponible à la location au moment où la requête est exécutée.

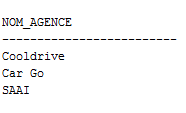
SELECT DISTINCT a.nom\_agence

FROM AGENCE a, VEHICULE v

WHERE a.id\_agence=v.id\_agence

AND v.TYPE\_VEH='utilitaire'

AND v.disponibilite = '1' ;



### e) requête n°5

Afficher les noms des responsables des agences dans lesquelles il est impossible de louer un véhicule de catégorie voiture.

SELECT distinct e.NOM\_EMP

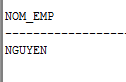
FROM EMPLOYE e, AGENCE a, VEHICULE v

WHERE e.NUM\_EMP = a.NUM\_EMP

AND a.ID\_AGENCE = v.ID\_AGENCE

AND v.TYPE\_VEH = 'voiture'

AND v.DISPONIBILITE='0';



### f) requête n°6

Trouver le plus grand nombre de locations effectuées par un client et afficher les noms et adresses des clients qui ont effectué ce grand nombre de locations.

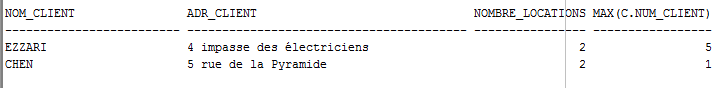
SELECT c.nom\_client , c.adr\_client, COUNT(\*) as nombre\_locations, max(c.NUM\_CLIENT)

FROM CLIENT c, LOUER l

WHERE c.num\_client = l.num\_client

GROUP BY c.nom\_client , c.adr\_client

HAVING COUNT(\*) >= 2;



### g) requête n°7

Afficher agence par agence, le nom de l’agence, le nom de son responsable, ainsi que le nombre de locations de plus de trois jours effectuées en 2015.

SELECT a.NOM\_AGENCE , e.NOM\_EMP , COUNT(\*) nombre\_locations

FROM AGENCE a , EMPLOYE e, LOUER l

WHERE a.ID\_AGENCE = e.ID\_AGENCE

AND a.ID\_AGENCE = l.ID\_AGENCE

AND l.DATE\_DEBUT\_LOC BETWEEN ('01/01/2015') and ('31/12/2015')

AND l.DATE\_FIN\_LOC BETWEEN ('01/01/2015') and ('31/12/2015')

AND e.TYPE\_EMP='responsable'

AND l.DUREE\_LOC > 3

GROUP BY a.NOM\_AGENCE, e.NOM\_EMP;



### h) requête n°8

Pour chaque véhicule de moins de 20 000 km (au moment où la requête est effectuée), donner la somme totale des montants de toutes les locations effectuées par des entreprises dont il a fait l’objet au cours du mois de juillet 2015.

SELECT SUM(l.PRIX\_LOC) montants\_locations

FROM VEHICULE v, LOUER l, CLIENT c

WHERE l.NUM\_IMMATRICULATION = v.NUM\_IMMATRICULATION

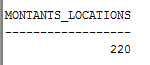
AND c.NUM\_CLIENT = l.NUM\_CLIENT

AND c.TYPE\_CLIENT='entreprise'

AND v.KILOMETRAGE\_FIN < 20000

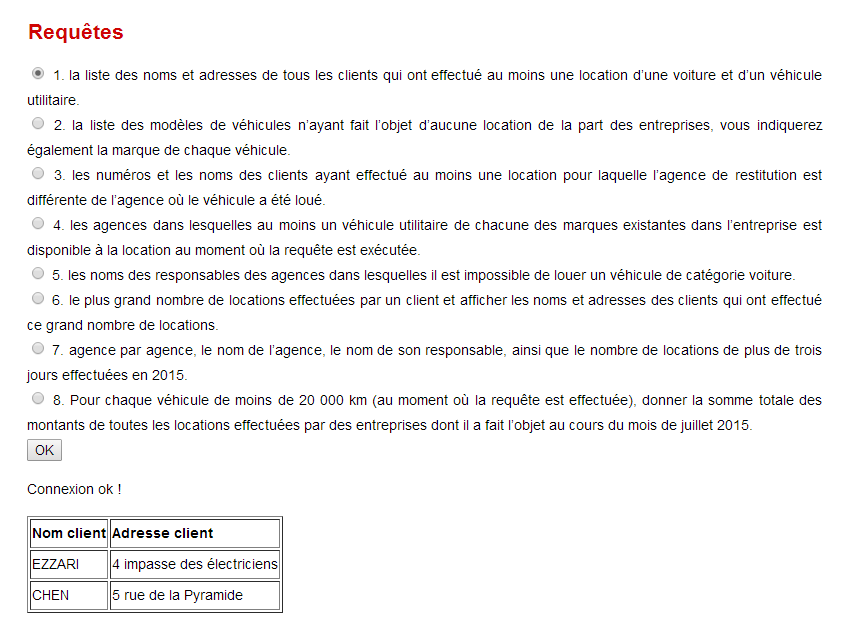
AND l.DATE\_DEBUT\_LOC BETWEEN ('01/07/2015') and ('31/07/2015')

AND l.DATE\_FIN\_LOC BETWEEN ('01/07/2015') and ('31/07/2015');

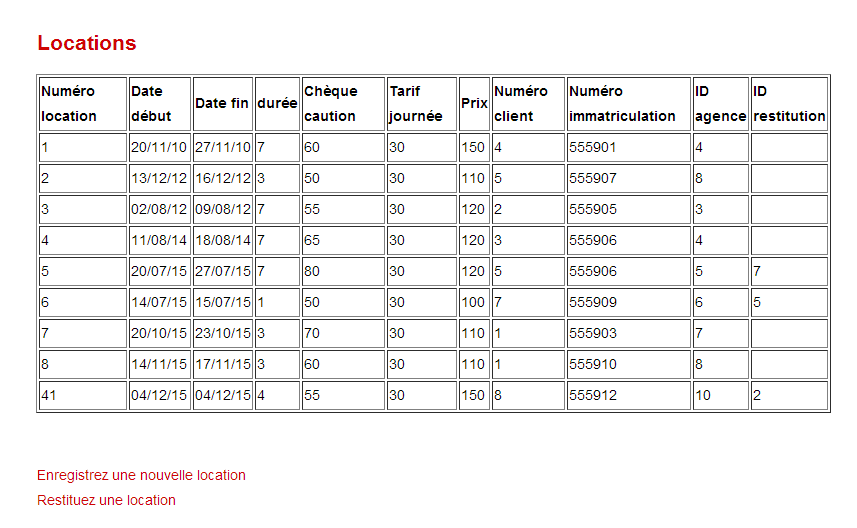


# VII. Scripts PHP

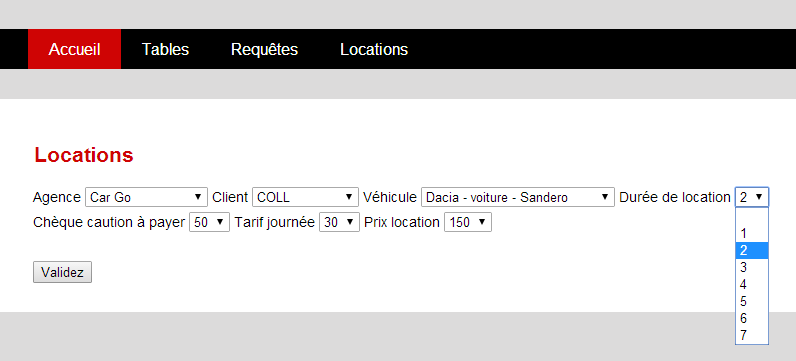
## 1. Quelques illustrations



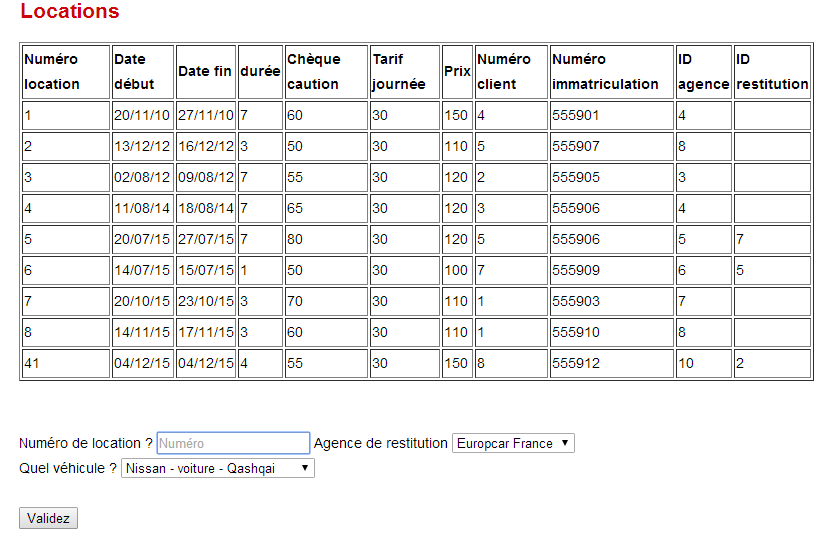
Page qui permet de tester les requêtes demandées au projet.



Page (***locations.php***) qui affiche toutes les locations existantes et offre deux options : soit d’enregistrer une nouvelle location ou de la restituer via des liens *HTML*.



Cette image montre comment on enregistre une nouvelle location d’un véhicule. Ici, il suffit simplement de saisir tous les champs avant de pouvoir valider sinon cette page affichera une erreur.

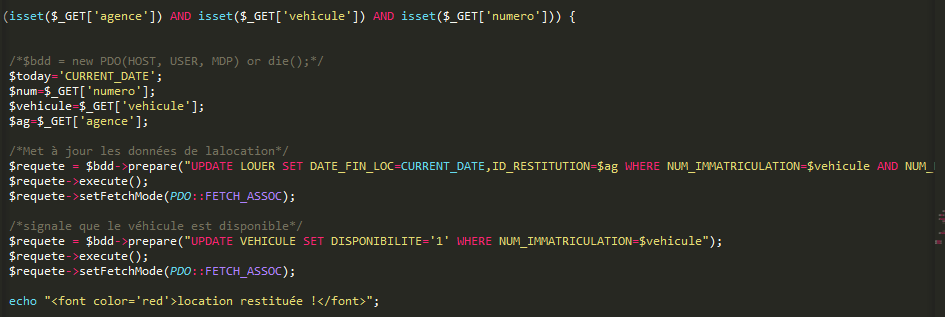


Cette image montre comment on restitue un véhicule à une agence. Ici, on saisit le numéro de location qui existe, ensuite, on choisit le nom de d’agence de retour, ainsi que le nom du véhicule concerné.

## 2. Codes



Code qui permet de réaliser une nouvelle location d’un véhicule.



Code qui permet de réaliser une restitution d’un véhicule à une agence.

# VIII. Remarques sur le déroulement du projet

## 1. Méthode de travail au sein du groupe

Après avoir choisi et analysé le sujet, nous avons rédigé le cahier des charges.

Ensuite, nous nous réunissions pour mettre en commun nos idées, puis nous nous répartissons les tâches pour effectuer chacun une partie du travail.

Puis, nous nous réunissions afin d’effectuer des tests de notre projet et de corriger les erreurs.

Nos travaux étaient enregistrés sur nos ordinateurs personnels pour éviter toute perte de données.

## 2. Difficultés rencontrées

- l’exécution des requêtes.

- la connexion de la base de données en PDO.

- la programmation en PHP.

# IX. Conclusion

Par conséquent, nous avons donc créé notre programme construit avec des fonctionnalités grâce au logiciel SQL Developper.

Pour réaliser notre projet, nous nous sommes fixés des objectifs de programmation. Notre produit doit être simple d’utilisation, les résultats du calcul devaient être clairs et compréhensibles, le programme doit utiliser des méthodes de SQL Developper et le code doit être très commenté pour permettre à l’utilisateur de comprendre le code malgré quelques difficultés rencontrées.

La réalisation de ce projet nous a appris ce qu’est le travail en groupe. Entre autre, nous avons appris à répartir les tâches entre deux personnes, sécuriser les données pour éviter toute perte et mettre en commun le code créé individuellement.

De plus, au niveau informatique nous avons découvert et maitriser de nouvelles techniques de programmation et perfectionné notre programmation SQL.

Ceci nous aura été bénéfique pour des projets futurs que ce soit dans les études ou dans la vie professionnelle.

Si nous devons refaire ce projet, nous déciderons d’optimiser cette fois la simulation des comportements des souris avec plus de fonctionnalités selon nos propres idées.