

# Projet de base de données

Raphael Javaux et Marien Bourguignon

Avril 2013

## 1 Technologies

Nous avons décidé d'utiliser le framework bootstrap (une librairie HTML5 + CSS3 + Javascript) pour la partie HTML de notre site web. Pour les accès à la base de données, nous avons choisi d'utiliser PDO de PHP.

## 2 Fonctionnement du Site Web

Notre projet est accessible à l'adresse suivante :

<http://ks201461.kimsufi.com/bdd/>

### 2.1 Première connexion

Lorsqu'un utilisateur se connecte pour la première fois, il est automatiquement redirigé sur la page de login (signin.php). Si notre script arrive à se connecter à la base de données avec les identifiants qu'il a fourni, une variable de session sera initialisée, qui lui évitera d'être redirigé à nouveau vers la page de connexion par la suite.

Il est possible de se connecter sur notre BDD avec l'utilisateur **ULg** et le mot de passe **batman**.

## 3 Requêtes

### 3.1 Fonctionnalité (a) (global\_search.php)

Nous avons implémenté cette partie totalement génériquement. Il est tout à fait possible de porter ce code dans une autre application sans devoir changer la moindre parcelle de code.

Nous avons découpé le travail de la manière suivante:

- 1 Rechercher la liste des tables de la base de données utilisée ;
- 2 L'utilisateur sélectionne une table ;
- 3 Rechercher, pour la table sélectionnée, toutes les colonnes de la clef primaire ;

- 4 L'utilisateur spécifie les valeurs pour toutes les colonnes de la clé ;
- 5 On crée la requête finale, en prenant en compte le \* ;
- 6 Le résultat est affiché dans un tableau.

La procédure vérifie tous les champs fournis par l'utilisateur pour éviter tout trifouillage dans la base de données.

### 3.1.1 Requetes

#### Recherche des tables

```
SHOW TABLES
```

#### Recherche des clefs primaires

```
SHOW KEYS FROM <tablename> WHERE Key_name = 'PRIMARY'
```

#### Recherche sur les clefs

```
SELECT * FROM <tablename> WHERE <conditions>  
Ou <conditions> est du type "champ = valeur".
```

## 3.2 Fonctionnalité (b) (vehicule\_availability.php)

Sélection des informations des véhicules disponibles entre deux dates pour une catégorie

```
SELECT v.*, m.*  
FROM vehicule AS v  
INNER JOIN modele AS m  
    ON m.id_mod = v.id_mod  
WHERE m.id_cat = <id_cat>  
    AND NOT EXISTS (  
    -- Disponible s'il n'existe pas un contrat  
    -- qui se superpose aux dates souhaitées.  
    SELECT *  
    FROM contrat AS c  
    WHERE      <date_debut> <= c.date_fin  
              AND <date_fin>   >= c.date_debut  
              AND c.id_veh = v.id_veh  
    );
```

## 3.3 Fonctionnalité (c) (new\_transaction.php)

Les requêtes de cette page s'effectuent dans une même transaction pour éviter des lecture non répétables entre la validité des données fournies et l'insertion.

**Sélection de l'ensemble des noms des clients**

```
SELECT p.id_client,  
       CONCAT(p.prenom_part, ' ', p.nom_part) AS nom  
FROM particulier AS p  
UNION  
SELECT s.id_client, s.nom_societe AS nom  
FROM societe AS s;
```

**Sélection de l'ensemble des conducteurs**

```
SELECT cc.no_permis,  
       CONCAT(p.prenom_part, ' ', p.nom_part) AS nom  
FROM particulier AS p  
INNER JOIN cond_client AS cc  
      ON cc.id_client = p.id_client  
UNION  
SELECT cc.no_permis, s.nom_societe AS nom  
FROM societe AS s  
INNER JOIN cond_client AS cc  
      ON cc.id_client = s.id_client  
UNION  
SELECT ca.no_permis,  
       CONCAT(ca.prenom_conducteur_autre, ' ',  
              ca.nom_conducteur_autre) AS nom  
FROM cond_autre AS ca;
```

**Sélection des véhicules disponibles pour la période souhaitée**

```
SELECT v.*, m.*  
FROM vehicule AS v  
INNER JOIN modele AS m  
      ON m.id_mod = v.id_mod  
WHERE NOT EXISTS (  
  -- Disponible s'il n'existe pas un contrat  
  -- qui se superpose aux dates souhaitées.  
  SELECT *  
  FROM contrat AS c  
  WHERE <date_debut> <= c.date_fin  
        AND <date_fin>   >= c.date_debut  
        AND c.id_veh = v.id_veh  
);
```

**Insertion du contrat**

```
INSERT INTO contrat  
(id_veh, id_client, date_debut, date_fin, km_debut, km_fin)  
VALUES (<id_veh>, <id_client>, <date_debut>, <date_fin>, NULL, NULL);
```

**Insertion des conducteurs du contrat**

```
INSERT INTO conduit (id_cont, no_permis)
VALUES (<id_contrat>, <no_permis>);
```

**3.4 Fonctionnalité (d) (finish\_contract.php)**

Les requêtes de cette page s'effectuent dans une même transaction pour éviter de facturer deux fois la même commande.

**Sélection des contrats non facturés**

```
SELECT c.id_cont
FROM contrat AS c
WHERE NOT EXISTS (
    SELECT *
    FROM facture AS f
    WHERE f.id_cont = c.id_cont)
ORDER BY c.id_cont;
```

**Mise à jour du kilométrage du contrat**

```
UPDATE contrat
SET km_debut = <km_debut>, km_fin = <km_fin>
WHERE id_cont = <id_cont>;
```

**Insertion et calcul de la facture**

```
INSERT INTO facture (id_cont, prix)
SELECT con.id_cont,
    DATEDIFF(con.date_fin, con.date_debut) * cat.prix_jour
    + (con.km_fin - con.km_debut) * cat.prix_km AS prix
FROM contrat AS con
INNER JOIN vehicule AS v
    ON v.id_veh = con.id_veh
INNER JOIN modele AS m
    ON m.id_mod = v.id_mod
INNER JOIN categorie AS cat
    ON cat.id_cat = m.id_cat
WHERE con.id_cont = <id_cont>;
```

**3.5 Fonctionnalité (e) (search\_id.php)****Sélection du véhicule ayant le plus voyagé sur une période**

```
SELECT c.id_veh, SUM(c.km_fin - c.km_debut) AS kms
FROM contrat AS c
```

```
WHERE c.date_debut >= <date_debut> AND c.date_fin <= <date_fin>
GROUP BY c.id_veh
ORDER BY kms DESC
LIMIT 1;
```