



---

# Flotte de vélos

---

Filière informatique 2<sup>ème</sup> année  
Projet SGBD S7

***Auteurs :***

Nicolas DO  
Othmane MANSOURI  
Aymane LAMHAMDI

***Encadrants :***

M. Sylvain LOMBARDY  
M. Mohamed MOSBAH

***Directeur de projet :***

M. Sylvain LOMBARDY

# Table des matières

<b>I</b>	<b>Présentation du sujet</b>	<b>3</b>
I.1	Objectif du projet . . . . .	3
<b>II</b>	<b>Modélisation des données</b>	<b>4</b>
II.1	Présentation globale de la base . . . . .	4
II.2	Modèle conceptuel . . . . .	4
II.2.1	Les entités mises en jeux . . . . .	4
II.2.2	Modèle entité-association . . . . .	7
II.3	Modèle relationnel . . . . .	9
II.3.1	Du conceptuel au relationnel . . . . .	9
II.3.2	Les opérations prévues sur la base . . . . .	10
<b>III</b>	<b>La base de données</b>	<b>10</b>
III.1	Implémentation de la base . . . . .	10
III.1.1	Création de la base . . . . .	10
III.1.2	Insertion de données . . . . .	11
III.2	Implémentation des requêtes SQL . . . . .	12
III.2.1	Consultation . . . . .	12
III.2.2	Mise à jour . . . . .	12
III.2.3	Suppressions . . . . .	13
III.2.4	Statistiques . . . . .	14
<b>IV</b>	<b>L'interface</b>	<b>15</b>
IV.1	Outils utilisés . . . . .	15
IV.2	Installation et configuration . . . . .	15
IV.3	Utilisation . . . . .	15
IV.4	Affichage de tables . . . . .	16
IV.5	Affichage des statistiques . . . . .	16
IV.6	Mise à jour des données . . . . .	17
IV.6.1	Démonstration . . . . .	19
<b>V</b>	<b>Conclusion</b>	<b>21</b>

## Table des figures

1	Logo Vélo . . . . .	3
2	Schéma conceptuel initial . . . . .	7
3	Schéma conceptuel final . . . . .	8
4	Schéma relationnel . . . . .	9
5	La page d'accueil . . . . .	16
6	La table des stations . . . . .	16
7	La liste des statistiques . . . . .	17
8	Classement des vélos les plus chargés par station . . . . .	17
9	La liste des opérations . . . . .	18
10	Début d'emprunt . . . . .	18
11	Fin d'emprunt . . . . .	19
12	Suppression d'historique . . . . .	19
13	État de la table avant l'emprunt . . . . .	19
14	Démonstration d'un début d'emprunt . . . . .	20
15	État de la table au cours de l'emprunt . . . . .	20
16	Démonstration d'une fin d'emprunt . . . . .	20
17	État de la table après l'emprunt . . . . .	20

# I Présentation du sujet

Ce rapport a pour objectif de décrire l'ensemble du processus de réalisation de projet du système de gestion de base de données, un projet de programmation proposé en 2ème année d'informatique à l'ENSEIRB-MATMECA. Il détaille nos choix de modélisation et d'implémentation de la base de données en différentes étapes :

1. La conception des données
2. L'implémentation de la base et des requêtes
3. L'utilisation de l'interface graphique

## I.1 Objectif du projet

Ce projet consiste à réaliser une base de données pour gérer les emprunts de vélos électriques. Il démarre par une modélisation de données, et aboutit à la création d'une base de données relationnelle et à l'implémentation d'un certain nombre d'opérations (consultations, mises à jour, etc).

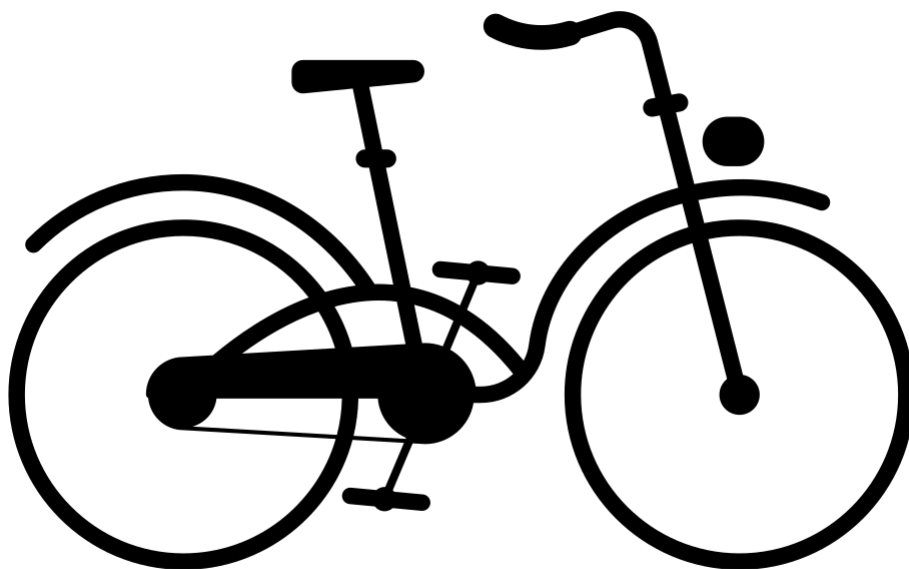


FIGURE 1 – Logo Vélo

## II Modélisation des données

### II.1 Présentation globale de la base

La base gère le service des emprunts ou de location de vélos électriques. Un adhérent souscrit à un abonnement peut emprunter un vélo d'une station, faire son trajet puis le déposer à une autre station en fonction des places disponibles.

### II.2 Modèle conceptuel

Dans cette partie, nous discuterons des choix faits pour le modèle conceptuel, il nous permettra d'établir une représentation claire des données et de définir les dépendances fonctionnelles de ces données entre elles.

#### II.2.1 Les entités mises en jeux

Les entités qu'on a jugées utiles pour la conception de notre bases sont les suivantes :

##### **Communes :**

Une commune est un élément fondamental de notre base. Chaque commune peut contenir plusieurs ou aucune station. Elle est représentée par un `ID_commune`, son nom et son code postal.

COMMUNES
id_commune
nom_commune
code_postal

##### **Stations :**

Les stations électriques sont des éléments fondamentaux pour l'implémentation de notre base aussi. Une station de vélo est caractérisée par le numéro de station et son adresse. Une association **être éloigné** entre la table et elle-même permet de déterminer la distance entre chacune des différentes stations. Une station se trouve dans une et une seule commune et contient au moins une borne.

STATIONS
numéro_station
adresse_station

### **Bornes :**

Les bornes sont des éléments nécessaires pour les stations. Elles permettent de stationner et ranger les vélos électriques tout en rechargeant leurs batteries. Elles sont caractérisées par leur numéro de borne et leur état (en service/hors service). Une borne est contenue dans une et une seule station, et elle est occupé par un vélo ou est libre.

BORNES
numéro_borne
état

### **Vélos :**

Un vélo est attaché à une borne en service. Il est caractérisé par une référence, sa marque, son kilométrage, son état, le niveau de charge de sa batterie et la date de mise en service. Il peut être emprunté par un adhérent sans limite de distance ou de temps.

VÉLOS
référence
marque
kilométrage
état
batterie
date_mise_en_service

### **Adhérents :**

Une personne est considéré comme adhérent que si elle s'est déjà abonnée au moins une fois au service. Un adhérent est caractérisé par un numéro d'adhérent, son nom et prénom, son adresse et sa commune, et sa date d'adhésion.

ADHÉRENTS
numéro_adhérent
nom
prénom
adresse
commune
date_adhésion

### Historique des emprunts :

Cette table nous sera utile pour stocker les opérations des emprunts des vélos. Elle est caractérisée par un ID\_historique, la date de début d'emprunt, la date de fin d'emprunt et la distance parcourue.

HISTORIQUE
id_historique
date_début_emprunt
date_fin_emprunt
distance

### Abonnements :

Un abonnement est caractérisé par un ID\_abonnement, le type d'abonnement et le tarif associé par heure d'utilisation du service. Un adhérent peut souscrire à un ou plusieurs abonnements.

ABONNEMENTS
id_historique
date_début_emprunt
date_fin_emprunt
distance

### Historique des abonnements :

Cette table nous sera utile pour stocker l'historique des souscriptions au service. Elle est caractérisée par un ID\_historique\_abonnement, une date de début d'abonnement et une autre de fin d'abonnement.

HISTORIQUE_ABONNEMENTS
id_historique_abonnement
date_début_abonnement
date_fin_abonnement

## II.2.2 Modèle entité-association

Modèle initial :

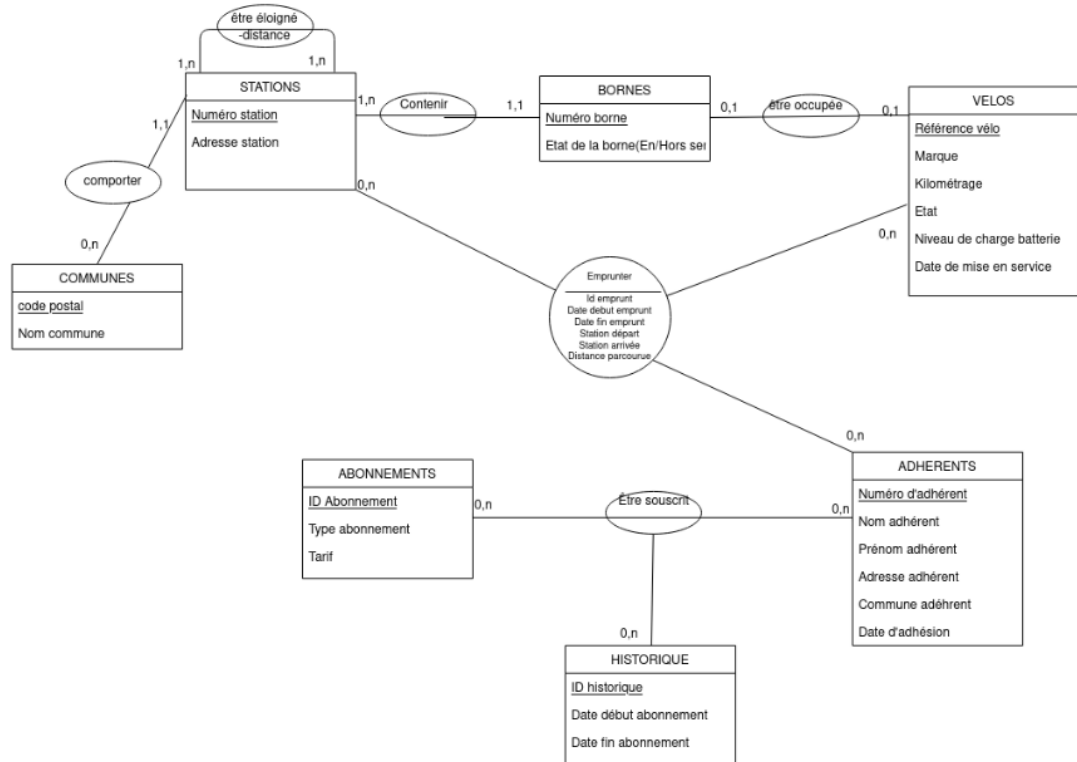


FIGURE 2 – Schéma conceptuel initial

**Explications et justifications des changements effectués sur le schéma conceptuel durant le projet :**

Pendant l'étape de l'implémentation de la base sur machine, il s'est avéré que le schéma conceptuel qu'on a proposé pour modéliser les emprunts des vélos n'était pas suffisant, et qu'il faudrait effectuer quelques modifications sur les entités pour faire fonctionner la base de données de manière cohérente. Principalement, l'ajout d'une entité **historique** indépendante pour enregistrer l'historique des emprunts. Cette entité est liée avec la table **bornes** avec deux associations, qui représentent les bornes/stations de départ et d'arrivée. Ce changement dans le schéma conceptuel avait pour but d'avoir l'information de la borne de départ et la borne d'arrivée ( les stations pourront être



tirées des numéros des bornes ). Cela n'était pas possible dans le schéma initial vu que, d'un coté, l'ancienne relation ternaire **Emprunter** (**stations**, **adherents**, **velos**) ne permettait de contenir qu'une seule station, et donc on perd l'information sur l'une des deux stations. D'un autre coté, vu que les trois clés étrangères qui font référence aux trois tables **stations**, **adherents**, et **velos** forment une clé primaire de la table **Emprunter** après la transformation du schéma conceptuel en un schéma relationnel. Cela imposait l'unicité du triplet (**numero\_station**, **numero\_adherent**, **reference\_velo**) ce qui met une contrainte non nécessaire sur l'emprunt des vélos en supposant qu'un même adhérent ne peut faire deux emprunts différents du même vélo de la même station de la même borne.

### Modèle final :

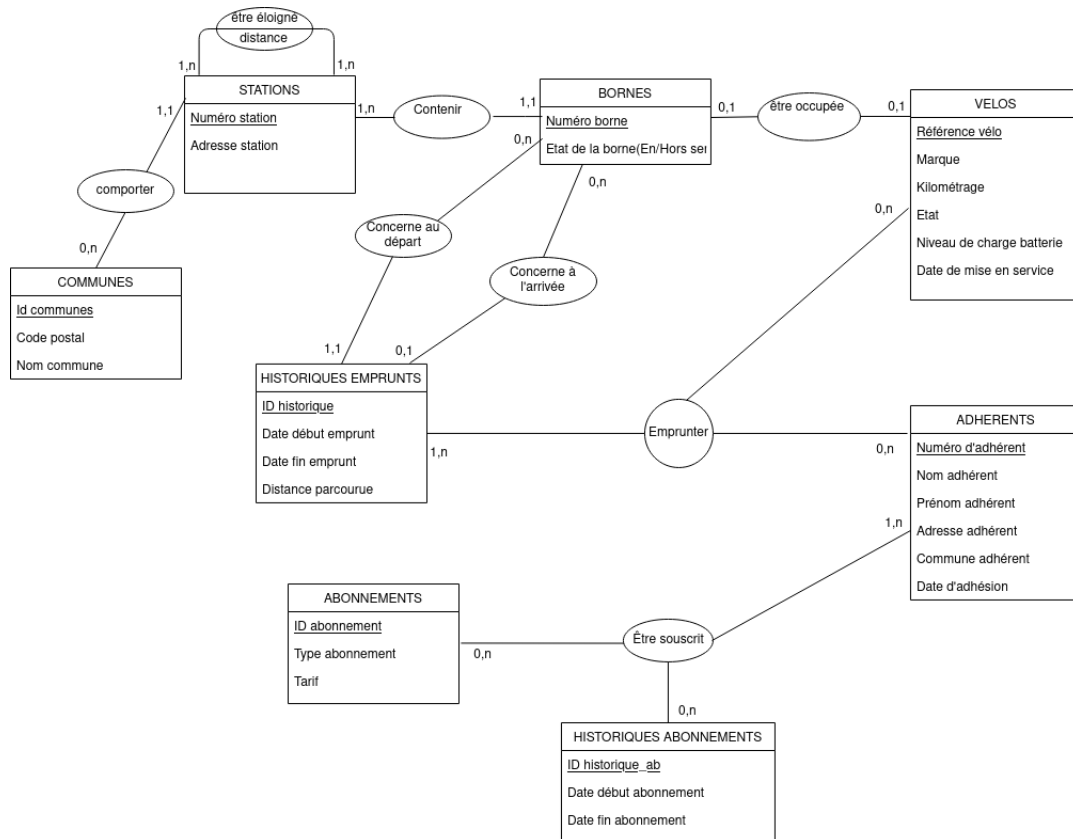


FIGURE 3 – Schéma conceptuel final

## II.3 Modèle relationnel

Nous expliquerons dans cette partie comment on est passé de notre schéma conceptuel au schéma relationnel tout en citant les opérations prévues sur la base.

### II.3.1 Du conceptuel au relationnel

Après avoir lister les entités de la base. Nous passons à la traduction de notre schéma conceptuel en schéma relationnel qui doit être conforme en 3ème forme normale. La figure représente les différentes tables présentes dans la base.

```
COMMUNES(id_commune, Code postal, Nom commune)
STATIONS(Numéro station, Adresse station, #id_commune )
BORNES(Numéro borne, #Numéro station, État Borne)
VELOS(Référence Velos, #Numéro borne (unique), Marque, Kilométrage, Etat, Niveau de charge batterie, Date de mise en service)
ADHERENTS(Numéro adhérent, Nom adhérent, Prénom adhérent, Adresse adhérent, Commune adhérent, Date adhésion)
HISTORIQUE(ID historique, Date début emprunt, date fin emprunt, distance parcourue, #id_borne_depart (non nulle), #id_borne_arrivée)
EMPRUNTS(#Numéro d'adhérent, #Référence vélo, #ID_historique)
DISTANCE(#numero_station_A, #numero_station_B, distance)
SOUSCRIPTION(#Numéro d'adhérent, #ID abonnement, #ID historique)
ABONNEMENTS(ID abonnement, Type abonnement, Tarif)
HISTORIQUE_AB(ID historique_ab, Date début abonnement, Date fin abonnement)
```

FIGURE 4 – Schéma relationnel

Dans ce schéma, nous pouvons voir les différentes contraintes d'intégrité et les dépendances fonctionnelles :

```
id_commune → code postale, nom commune
numéro station → adresse station, id_commune
numéro bornes → numéro station, état borne
référence velo → marque, kilométrage, état, niveau de charge batterie,
date de mise en service, numéro borne
numéro adhérent → nom adhérent, prénom adhérent, adresse adhérent,
commune adhérent, date adhésion
id_historique → date début emprunt, date fin emprunt, distance parcourue,
numéro borne départ, numéro borne arrivée
id_abonnements → type abonnement, tarif
id_historique_ab → date début abonnement, date fin abonnement
```

### II.3.2 Les opérations prévues sur la base

3 types d'opérations sont prévus pour notre base de données :

**Consultation** : pour consulter et afficher les données de la base (liste des communes, liste des stations, liste des adhérents, liste des vélos par station/en cours d'utilisation, liste des stations pour une commune donnée, liste des adhérents ayant emprunté au moins deux vélos pour un jour donné...)

**Statistiques** : pour calculer la moyenne du nombre d'utilisateurs par vélo par jour, la moyenne des distances parcourues par les vélos sur une semaine, le classement des stations par nombre de places disponibles par commune et enfin le classement des vélos les plus chargés par station.

**Mise à jour** : Pour mettre à jour, supprimer, ajouter ou modifier les données.

## III La base de données

Pour ce projet, la base SQL choisi s'est porté sur MySQL/mariaDB

### III.1 Implémentation de la base

#### III.1.1 Création de la base

Le fait d'avoir établi au préalable le schéma relationnel a été très bénéfique car cela nous a permis dans un premier de savoir les différentes tables à créer, les différents champs que chacune d'elles possède et leurs contraintes (unique, non nulle, clé primaire, clé secondaire...) Ainsi, la création des tables pour chacune des entités se traduisait par exemple pour la table bornes :

```
1 CREATE TABLE 'bornes '  
2 (  
3     numero_borne INT AUTO.INCREMENT,  
4     etat_borne CHAR(50) ,  
5     numero_station INT NOT NULL, # Ne peut pas être nulle  
6     PRIMARY KEY (numero_borne)   # Déclaration de numéro_borne  
7                                   # en tant que clé primaire  
8 );
```

Jusque là, nous n'avons pas encore tenu comptes des potentielles clés étrangères. Ces contraintes seront mis en place une fois que toutes les tables soient créées de la façon suivante :

```
1 ALTER TABLE 'bornes '  
2 ADD CONSTRAINT 'fk1_bornes' FOREIGN KEY (numero_station)  
3 REFERENCES 'stations' (numero_station) ON DELETE CASCADE;
```

---

On remarque ici que nous avons utilisé des options en cas de suppression d'une référence vers la table mère que nous détaillerons plus en détail dans la partie III.2.3.

### III.1.2 Insertion de données

Pour insérer des données dans la base de données, cela se fait de la manière suivante :

- Pour ajouter une nouvelle stations dans la table des stations

```
1 INSERT INTO stations
2 (numero_station , adresse_station , id_commune)
3 VALUES (10, '3 avenue bardanac', 33)
```

- Pour ajouter un nouvel adhérent dans la table des adhérents

```
1 INSERT INTO adherents (numero_adherent , nom_adherent ,
2   prenom_adherent , adresse_adherent , commune_adherent ,
3   date_adhesion)
4 VALUES (777, 'Dupont', 'Tom', '5 avenue Collegno', 'Talence',
5   '2021-05-24 11:30:00')
```

- Pour ajouter un nouvel emprunt dans la table des **emprunts** et enregistrer l'emprunt dans la table **historique**

```
1 INSERT INTO historique (date_debut_emprunt , date_fin_emprunt
2   , numero_borne_depart , numero_borne_arrivee ,
3   distance_parcourue)
4 VALUES ('2021-02-02 17:30:00', '2021-02-02 18:30:00', 5, 6,
5   30);
6
7 INSERT INTO emprunts
8 (numero_adherent , reference_velo , id_historique)
9 VALUES (777, 19, 70);
```

D'autres exemples d'ajout sont visibles dans le fichier `requetes_update.sql`

## III.2 Implémentation des requêtes SQL

### III.2.1 Consultation

La consultation d'une table de la base de données se fait par la requête suivante :

```
1 SELECT * from nom_de_la_table ;
```

D'autres requêtes de consultation comme le nombre de vélos en cours d'utilisation, la liste des stations par communes, etc, sont disponibles dans le fichier `requetes_consult.sql`

### III.2.2 Mise à jour

Nous allons à présent voir tous les mécanismes que nous avons mis en place afin d'automatiser en quelque sorte les changements à apporter sur l'entièreté de la base lors de l'ajout, la modification ou la suppression d'une ligne dans une table en particulier. Sans automatisation, il aurait fallu, pour commencer un emprunt par exemple, d'une part créer manuellement une nouvelle ligne dans la table `historique`, associer cette ligne avec un vélo et un adhérent dans la table `emprunts`, recueillir la borne où était placé le vélo afin de l'insérer dans l'historique et pour finir dissocier le vélo de sa borne. Cela aurait été très contraignant à gérer au vu des nombreuses opérations à effectuer pour un seul emprunt (risque d'erreur accrue). Afin de faciliter la gestion des emprunts, nous avons voulu faire en sorte que lors du début d'un emprunt (ajout d'une ligne dans la table `emprunts`), seuls le numéro de l'adhérent et la référence du vélo étaient nécessaires. Ensuite, si une erreur de saisie a été commise et que l'on essaie d'emprunter un vélo déjà en cours d'utilisation, nous renvoyons un message d'erreur et aucune ligne n'est ajoutée à la table `emprunts`. Pour finir, dans le cas où le vélo est disponible, nous allons automatiquement créer une ligne dans la table historique avec la date actuelle renseignant le début de l'emprunt et la borne de départ, c'est à dire d'où le vélo a été pris. Le vélo, lui, est bien retiré de la borne (mise à nulle du champ "borne"). Tout cela est géré grâce au déclencheur "`debut_emprunt`" que nous avons implémenté et dont le code se trouve dans le fichier `trigger.sql`. D'autres déclencheurs se trouvant dans ce même fichier ont été mis en place pour gérer la remise du vélo. On va notamment interdire le fait de déposer un vélo sur une borne "`hors service`". Enfin, pour signaler la fin d'emprunt, il ne faut renseigner que la référence du vélo, la distance parcourue avec ce dernier et la borne de remise. Le déclencheur "`fin_emprunt`" va s'occuper de mettre à jour la ligne

d'historique correspondant à l'emprunt du vélo avec les informations données et associer le vélo à sa borne de dépôt.

### III.2.3 Suppressions

Ensuite, concernant la suppression de données, nous avons fait les choix suivants :

- Lors de la suppression d'une commune, nous mettons à nulle les clés étrangères qui référencent cette dernière dans la table **stations** car il se pourrait qu'on veuille les ré-attribuer plus tard à une nouvelle commune (lors d'une fusion entre communes par exemple)
- La suppression d'une station entraîne la suppression des bornes contenues dans cette dernière.
- La suppression d'une borne entraîne la mise à nulle des clés étrangères qui référencent cette dernière dans les tables **historique** et **velos** car on voudrait tout de même garder nos vélos et les ré-attribuer à d'autres bornes plus tard et garder l'historique même si on n'a plus l'information sur l'une ou les deux bornes.
- En ce qui concerne la table **emprunts**, étant donné qu'elle a pour clés étrangères un id historique, un numéro d'adhérent et une référence de vélo, nous avons décidé que lorsque l'on supprimait un historique, les lignes ayant comme référence l'id de l'historique supprimé seraient également supprimées car l'intérêt de la table **emprunts** est d'associer un emprunt à ses informations. En revanche, si on supprimait un adhérent ou un vélo, on voudrait garder l'association entre l'historique et le vélo (ou l'adhérent) afin de pouvoir par exemple faire des statistiques par rapport à ce vélo (où à cet adhérent).
- De la même façon, la table **etre\_souscrit** a pour clés étrangères un numéro d'adhérent, un id historique d'abonnement et un id abonnement. Lorsque l'on supprime un historique, on supprime les lignes dans la table **etre\_souscrit** ayant pour référence ce dernier. On garde la ligne si uniquement si on supprime un adhérent ou un abonnement pour avoir une trace des souscriptions des différents adhérents.

Ces différents mécanismes de suppressions sont précisés lorsque l'on pose la contrainte de clé étrangère sur une table avec pour référence une autre table. On utilise alors "ON DELETE SET NULL" pour la mise à nulle du champ lors de la suppression et "ON DELETE CASCADE" pour la suppression de la ligne ayant en référence l'élément supprimé.

```

1 ALTER TABLE 'bornes'
2 ADD CONSTRAINT 'fk1_bornes' FOREIGN KEY (numero_station)
3 REFERENCES 'stations' (numero_station) ON DELETE CASCADE;

```

Le fichier suppression.sql permet de supprimer toutes les données de la base.

### III.2.4 Statistiques

En plus des requêtes de mise à jour et la suppression des données, la base de données fournit des statistiques avec les requêtes implémentées dans le fichier requetes\_stat.sql. On mentionne :

- la moyenne du nombre d’usagers par vélo par jour, en calculant le nombre total d’emprunts pour chaque vélo de la base puis en faisant la moyenne sur les vélos.
- la moyenne des distances parcourues par les vélos sur une semaine, qu’on obtient en calculant pour chaque vélo la distance totale parcourue (la somme des distances parcourues pendant les emprunts du vélo) dans la durée de temps donnée sous forme d’une date qui indique le début de la semaine, puis on calcule la moyenne de ces distances.
- le classement des stations par nombre de places disponibles par commune. On commence par calculer, pour chaque station, le nombre de bornes  $n$  disponibles en suivant la formule :

$$n = N - a - b$$

avec

- $N$  : le nombre total des bornes dans la station.
- $a$  : le nombre de bornes ”hors service” dans la station.
- $b$  : le nombre de bornes associées à un vélo.

Finalement, on classe les stations suivant les valeurs obtenues.

- le classement des vélos les plus chargés par station, en tirant le niveau de batterie de chaque vélo puis les ordonner du vélo le plus chargé à celui le moins chargé. Les stations sont classé par leur identifiant du plus petit au plus grand.

## IV L'interface

Dans cette partie, nous allons présenter notre interface qu'on a implémenté pour faciliter l'interaction avec l'utilisateur. Nous citerons ici les étapes pour le bon fonctionnement de l'environnement d'exécution.

### IV.1 Outils utilisés

Pour l'implémentation de l'interface, nous avons utilisé principalement les langages **php** et **html**. Le principe était de créer une interface dynamique qui réponds aux besoins des utilisateurs.

Nous avons besoin aussi de mettre en place un serveur web local offrant une bonne souplesse d'utilisation. Pour ceci, nous avons utilisé l'outil **XAMPP** qui est un ensemble de logiciels permettant de configurer un serveur de test local avant la mise en oeuvre d'un site Web. **phpMyAdmin** a été également utilisé pour gérer notre base de donnée sur un serveur PHP.

Le php nous a permis de mettre le lien entre notre base de donnée et l'interface graphique. En fonction des actions déclenchées par l'utilisateur, notre script se connecte à notre base et la charge. Ensuite, en fonction de la requête indiquée, l'affichage du tableau est réalisée d'une manière dynamique.

### IV.2 Installation et configuration

Sous Linux, télécharger la package XAMPP pour linux depuis <https://www.apachefriends.org/fr/index.html>. Une fois le package installé, lancez la commande `sudo /opt/lampp/lampp start` pour lancer le service. Pour l'arrêter, lancez la commande `sudo /opt/lampp/lampp stop`. Récupérer le répertoire php de notre dépôt et placer le dans le répertoire `/opt/lampp/htdocs`. Depuis votre navigateur préféré, connectez-vous à votre localhost d'adresse IP 127.0.0.1 et accédez ensuite à notre interface via `127.0.0.1/php/` ou `localhost/php/`.

### IV.3 Utilisation

Une fois l'interface ouverte, la page d'accueil est lancée par défaut. Vous y trouverez une brève description de la base et un menu en haut avec différentes sections.





FIGURE 5 – La page d'accueil

#### IV.4 Affichage de tables

Pour consulter et faire afficher une table de la base, il suffit d'en sélectionner une parmi celles qui sont proposées dans la liste à la section "afficher une table". L'affichage de la table des stations nous donne le résultat suivant :

numero_station	adresse_station	id_commune
1	5 rue de la liberté	1
2	8 avenue collegno	2
3	3 rue de de la mort	3
4	4 rue de paradis	4
5	5 rue saint-louis	5
6	8 avenue joyeuse	6
7	3 rue de Marie	7
8	4 rue de alos	8
9	3 avenue bardanac	7

[Retourner à la page d'accueil](#)

FIGURE 6 – La table des stations

#### IV.5 Affichage des statistiques

Les statistiques introduites dans la partie III.2.4 peuvent être consultées depuis la section "statistique" du menu. Une liste déroulante vous proposera les statistiques à afficher.

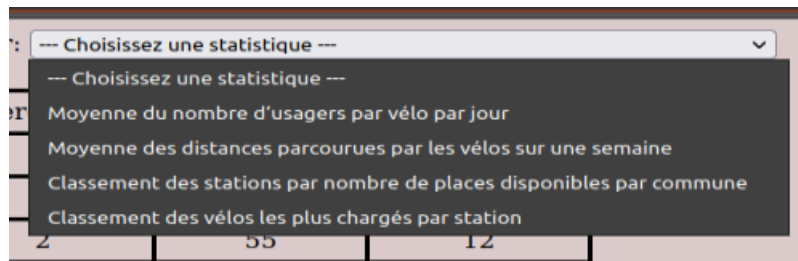


FIGURE 7 – La liste des statistiques

L’affichage du classement des vélos chargées par station par exemple nous donne le résultat suivant :

statistique à afficher:

numero_station	reference_velo	niveau_batterie
1	22	90
1	11	45
2	55	12
3	66	90
3	77	10
4	88	90
4	99	40
9	33	15

[Retourner à la page d'accueil](#)

FIGURE 8 – Classement des vélos les plus chargés par station

## IV.6 Mise à jour des données

La rubrique "Mettre à jour une table" dans le menu permet de mettre à jour les données, insérer des nouvelles ou en supprimer. Le système de mise à jour présenté dans la partie III.2.2 vérifie que les données peuvent être ajoutées à la base. On s’est principalement intéressé à la fonctionnalité de l’emprunt sur l’interface. Elle permet à l’utilisateur de commencer un emprunt, le finir et enfin de supprimer les données de l’emprunt de l’historique.

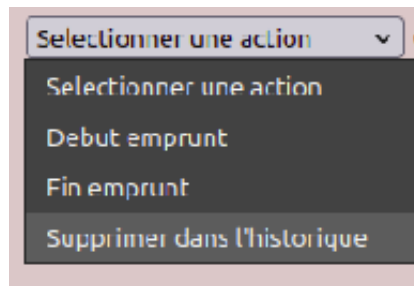


FIGURE 9 – La liste des opérations

### Récupération des données utilisateurs

Pour recueillir et utiliser les entrées de l'utilisateur dans nos requêtes pour l'interface graphique, nous avons utilisé des requêtes préparées :

```
1 $requete = mysqli_prepare($connection, "DELETE FROM historique
   WHERE id_historique = ?");
2 mysqli_stmt_bind_param($requete, "i", $_POST['id_historique']);
3 mysqli_stmt_execute($requete);
```

### Insertion des données

La 1ère action est déclenchée par le choix de "début emprunt" proposé dans la liste déroulante et en saisissant le numéro d'adhérent et la référence du vélo emprunté. La date de début d'emprunt est l'heure lors de la saisie des données.

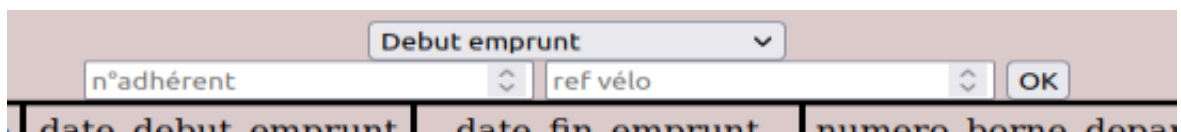


FIGURE 10 – Début d'emprunt

Le choix de "fin emprunt" permet de finir un trajet, l'utilisateur y indique la référence du vélo, la borne où il le dépose ainsi que la distance parcourue lors du trajet. Comme précédemment, l'heure de fin de l'emprunt est celle lors de la saisie des données.

FIGURE 11 – Fin d’emprunt

## Suppression des données

La dernière action ”supprimer dans l’historique” permet de supprimer une ligne de la table des historiques des emprunts en indiquant l’id historique.

FIGURE 12 – Suppression d’historique

### IV.6.1 Démonstration

Voici une démonstration sur la mise en place d’un emprunt, la fin de ce dernier et l’action qu’elle mène sur les tables emprunts et historique.

id_historique	numero adherent	reference velo	date debut emprunt	date fin emprunt	numero borne depart	numero borne arrivee	distance parcourue
1	11	11	2021-01-01 12:00:00	2021-01-01 12:30:00	2	3	10
2	13	11	2021-01-01 12:00:00	2021-01-01 12:30:00	2	4	10
3	12	11	2021-01-01 12:35:00	2021-01-01 13:05:00	4	5	20
4	11	22	2021-01-01 13:10:00	2021-01-01 13:40:00	3	6	6
5	13	11	2021-01-01 13:45:00	2021-01-01 13:15:00	1	2	13
6	11	22	2021-01-01 16:00:00	2021-01-01 16:15:00	1	1	3
7	2	33	2021-02-01 11:00:00	2021-02-01 12:00:00	1	2	20
8	2	11	2021-02-01 13:00:00	2021-02-01 15:00:00	2	3	40
9	3	22	2021-11-30 15:00:00	2021-11-30 18:00:00	2	3	15
10	5	33	2021-04-10 10:00:00	2021-04-10 11:00:00	2	5	9
11	4	11	2021-04-10 12:00:00	2021-04-10 15:00:00	3	4	10
12	4	11	2021-04-10 16:00:00	2021-04-10 18:00:00	5	4	10
13	5	44	2021-05-01 13:00:00		4		
14	5	11	2021-12-07 10:32:12	2021-12-07 10:41:53	1	14	100

[Retourner à la page d'accueil](#)

FIGURE 13 – État de la table avant l’emprunt

FIGURE 14 – Démonstration d'un début d'emprunt

id_historique	numero_adherent	reference_voie	date_debut_emprunt	date_fin_emprunt	numero_borne_depart	numero_borne_arrivee	distance_parcourue
1	11	11	2021-01-01 12:00:00	2021-01-01 12:30:00	2	3	10
2	13	11	2021-01-01 12:00:00	2021-01-01 12:30:00	2	4	10
3	12	11	2021-01-01 12:35:00	2021-01-01 13:05:00	4	5	20
4	11	22	2021-01-01 13:10:00	2021-01-01 13:40:00	3	6	6
5	13	11	2021-01-01 13:45:00	2021-01-01 13:15:00	1	2	13
6	11	22	2021-01-01 18:00:00	2021-01-01 18:15:00	1	1	3
7	2	33	2021-02-01 11:00:00	2021-02-01 12:00:00	1	2	20
8	2	11	2021-02-01 13:00:00	2021-02-01 15:00:00	2	3	40
9	3	22	2021-11-30 15:00:00	2021-11-30 18:00:00	2	3	15
10	5	33	2021-04-10 10:00:00	2021-04-10 11:00:00	2	5	9
11	4	11	2021-04-10 12:00:00	2021-04-10 15:00:00	3	4	10
12	4	11	2021-04-10 16:00:00	2021-04-10 18:00:00	5	4	10
13	5	44	2021-05-01 13:00:00		4		
14	5	11	2021-12-07 10:32:12	2021-12-07 10:41:53	1	14	100
15	1	33	2021-12-07 12:45:16		3		

FIGURE 15 – État de la table au cours de l'emprunt

FIGURE 16 – Démonstration d'une fin d'emprunt

id_historique	numero_adherent	reference_voie	date_debut_emprunt	date_fin_emprunt	numero_borne_depart	numero_borne_arrivee	distance_parcourue
1	11	11	2021-01-01 12:00:00	2021-01-01 12:30:00	2	3	10
2	13	11	2021-01-01 12:00:00	2021-01-01 12:30:00	2	4	10
3	12	11	2021-01-01 12:35:00	2021-01-01 13:05:00	4	5	20
4	11	22	2021-01-01 13:10:00	2021-01-01 13:40:00	3	6	6
5	13	11	2021-01-01 13:45:00	2021-01-01 13:15:00	1	2	13
6	11	22	2021-01-01 18:00:00	2021-01-01 18:15:00	1	1	3
7	2	33	2021-02-01 11:00:00	2021-02-01 12:00:00	1	2	20
8	2	11	2021-02-01 13:00:00	2021-02-01 15:00:00	2	3	40
9	3	22	2021-11-30 15:00:00	2021-11-30 18:00:00	2	3	15
10	5	33	2021-04-10 10:00:00	2021-04-10 11:00:00	2	5	9
11	4	11	2021-04-10 12:00:00	2021-04-10 15:00:00	3	4	10
12	4	11	2021-04-10 16:00:00	2021-04-10 18:00:00	5	4	10
13	5	44	2021-05-01 13:00:00	2021-12-07 10:41:53	4	10	250
14	5	11	2021-12-07 10:32:12	2021-12-07 10:41:53	1	14	100
15	1	33	2021-12-07 12:45:16		3		

FIGURE 17 – État de la table après l'emprunt

## V Conclusion

Ce projet nous a permis de comprendre l'intérêt de la méthodologie vu en cours pour réaliser une base de données : partir d'un schéma conceptuel, passer à un schéma relation pour enfin aboutir à l'implémentation de notre base et tout cela à partir d'un exemple concret. Bien que le sujet semblait simple aux premiers abords, il a permis énormément de débats au sein du groupe sur ce qu'on voulait mettre en place pour la base. De plus, le fait que nous devions proposer une interface graphique pour le projet nous a permis de manipuler de nouveaux outils permettant un affichage de nos tables comme php bien que la prise en main de ces outils a été assez laborieuse dans un premier temps. Pour finir, le sujet étant vaste et à cause de problèmes techniques avec MySQL sur plusieurs de nos machines, nous n'avons pas eu le temps d'implémenter toutes les fonctionnalités que nous avions envisagé au départ notamment la partie concernant l'abonnement de l'adhérent.