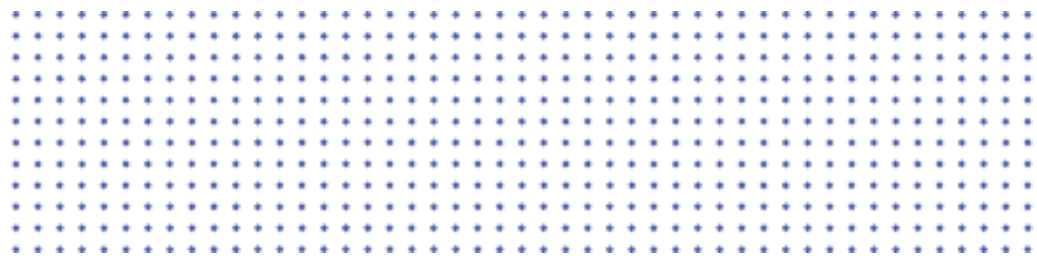




# SAE 2.01

[Conception et implémentation d'une base de données]

---



# TABLE DES MATIÈRES

## Table des matières

Introduction .....	
Méthode .....	
Typage des données .....	
Nettoyage des données .....	
Normalisation .....	
Schéma relationnel .....	
Requêtes .....	
Visualisations .....	

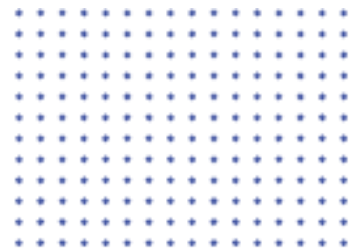
# INTRODUCTION



L'association AVEC, créée en janvier 2015, est une organisation d'intérêt général dont la mission est de favoriser l'émancipation de tous et de lutter contre toutes formes d'exclusion. Elle développe un projet d'accompagnement global, incluant des dimensions culturelles, éducatives et sociales, pour permettre aux habitants d'accéder à l'autonomie et au libre arbitre. Les projets innovants de l'association visent à renforcer le lien social, lutter contre la précarité, favoriser l'accès à la culture et aux pratiques artistiques, encourager l'implication des habitants, et combattre toutes formes de discrimination.

## Objectif du projet

Le projet présent vise à développer une base de données relationnelle pour l'association AVEC afin de suivre les adhésions et les dons effectués par les membres au fil des années. L'objectif principal est de créer un système de gestion des données plus granulaire et efficace que celui actuellement en place, qui repose sur des fichiers Excel. Cette base de données permettra non seulement de gérer les informations de manière plus structurée mais aussi de fournir des analyses et des visualisations détaillées pour faciliter la prise de décision.



## Description de la mission

Pour mener à bien ce projet, plusieurs étapes clés sont nécessaires : Analyse des données existantes :

- Identifier les types de données disponibles et comprendre les besoins de l'association.
- Modélisation conceptuelle et relationnelle : Créer un modèle conceptuel des données (MCD) et le transformer en modèle relationnel (MR) avec des tables normalisées.
- Implémentation de la base de données : Écrire et exécuter un script SQL pour créer et alimenter la base de données.
- Formulation de requêtes SQL : Développer des requêtes SQL pour extraire les informations nécessaires.
- Visualisation des données : Produire des visualisations des résultats des requêtes pour une meilleure compréhension et présentation.

## Importance de la base de données relationnelle

La base de données relationnelle constitue un outil central dans le processus décisionnel de l'association AVEC. En structurant les données de manière optimale et en assurant leur qualité, leur cohérence et leur intégrité, cette base de données facilitera non seulement la gestion courante des adhésions et des dons, mais aussi l'analyse approfondie des tendances et des comportements des membres. Cela permettra à l'association de mieux cibler ses actions et d'optimiser l'utilisation de ses ressources.

# TYPAGE DES DONNEES



Le typage des données est une étape cruciale dans la conception et l'implémentation d'une base de données relationnelle. Il consiste à définir le type de chaque colonne dans les tables de la base de données, ce qui permet d'assurer l'intégrité des données et d'optimiser les performances des requêtes. Les types de données déterminent le type d'informations qu'une colonne peut stocker et les opérations qui peuvent être effectuées sur ces données.

Dans la base de données créée pour l'association AVEC, plusieurs types de données ont été utilisés pour répondre aux différents besoins des tables. Voici une description des principaux types de données utilisés et les raisons de ces choix :

<b>NumeroAdherent</b>	<i>INT</i>
<b>Nom</b>	<i>VARCHAR(255)</i>
<b>Prenom</b>	<i>VARCHAR(255)</i>
<b>DateDeNaissance</b>	<i>DATE</i>
<b>Rue</b>	<i>VARCHAR(255)</i>
<b>Ville</b>	<i>VARCHAR(255)</i>
<b>LatitudeVille</b>	<i>FLOAT</i>
<b>LongitudeVille</b>	<i>FLOAT</i>
<b>Etat</b>	<i>VARCHAR(255)</i>
<b>Telephone :</b>	<i>VARCHAR(20)</i>
<b>DateAdhesion</b>	<i>DATE</i>
<b>Montant</b>	<i>FLOAT</i>
<b>Don</b>	<i>FLOAT</i>
<b>Benevole</b>	<i>VARCHAR(255)</i>
<b>MoyenDePaiement</b>	<i>VARCHAR(50)</i>
<b>Annee</b>	<i>INT</i>

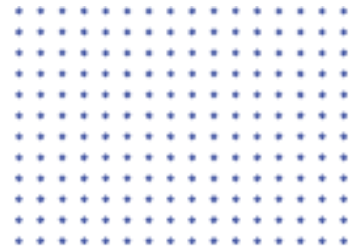
J'ai choisi d'utiliser le format VARCHAR pour le numéro de téléphone car avec un format comme INT, j'aurais eu des soucis avec les 0 au début ou en fin de saisie. 20 car les numéros sont rarement plus longs que ce nombre de caractères.

Les données originales continues dans l'Excel ont été modifiées voir transformées à des fins d'amélioration et d'optimisation pour la base de données Access :

Par exemple j'ai décidé de ne pas conserver la colonne « *expr34* » car elle ne contenait aucunes données, cela aura pour effet d'alléger la BDD d'un attribut « inutile »

J'ai également décidé de créer une nouvelle colonne (ou attribut) « *Année* » afin de ne plus utiliser les différentes colonnes relatives aux années (ex : Montant 2021) et donc tout centraliser. J'ai donc du réorganiser tout le jeu de données autour de ce nouvel attribut, en supprimant les colonnes relatives aux montants, dons et moyens de paiement relatifs aux années et en ne gardant à la place que les colonnes « *Année* », « *Montant* » et « *Don* ». Le but était de simplifier et alléger la BDD.

# NETTOYAGE DES DONNEES



Le nettoyage des données est une étape cruciale dans le processus de création d'une base de données relationnelle. Avant de pouvoir importer et exploiter les données, il est essentiel de s'assurer que celles-ci sont propres, cohérentes et prêtes à être utilisées. Le fichier Excel fourni par l'association AVEC contenait des informations brutes qui nécessitaient plusieurs étapes de nettoyage pour garantir leur intégrité et leur qualité.

## 1 - Validation des formats de données

Les formats de données incohérents peuvent compliquer l'importation et l'analyse. Les colonnes contenant des dates, des montants et des coordonnées etc. ont été vérifiées et corrigées pour assurer un format uniforme.

Pour les coordonnées, remplacement des points par des virgules pour que les données soient bien au bon format :

B	C
Latitude Ville	Longitude Ville
42,65	-73.756232
42,65	-73.756232
42,65	-73.756232

Certains montants n'étaient pas au bon format, j'ai dû m'assurer de les passer au format float pour pouvoir les importer ensuite dans Access :

2020	30,00
2021	30

## 2 – Uniformisation des données

Dans la colonne Bénévole, uniformisation de la modalité de réponse pour éviter les fautes et saisies différentes : Avant/Après :

	Benevole ▾
OUI, pontcuel	Ponctuel(le)

Même chose avec le moyen de paiement qui comportait plusieurs écriture différentes pour une même modalité.

Modification de la date «3017 » en 2017 car c'était probablement une erreur de saisie.

### 3 – Gestion des valeurs manquantes

Suppression lignes vides dans table Adhesion :

	A	B	C	D	E	F	G	H	I	J
1	NUMERO ADHERENT	Année	DATE ADHESION	MONTANT	DON	MOYEN DE PAIEMENT				
415		24								
416		25								
417		26								
418		27								
419		30								
420		31								
421		31								
422		31								
423		32								
424		32								
425		32								
426		33								
427		33								
428		33								
429		35								
430		36								
431		36								
432		41								
433		41								
434		43								
435		43								
436		47								
437		50								
438		51								
439		52								
440		54								
441		55								
442		57								
443		58								
444		60								
445		77								
446		78								
447		79								

Les valeurs « Montant » comportant “?” ont été remplacé par une cellule vide

021	20
021 ?	
021 200	

# NORMALISATION



La normalisation est un processus essentiel dans la conception de bases de données relationnelles. Elle permet de structurer les données de manière à réduire la redondance et à éviter les anomalies de mise à jour. Ce processus implique la division des tables en sous-tables plus petites et la définition de relations entre elles. Pour la base de données de l'association AVEC, la normalisation a été appliquée pour garantir l'efficacité et l'intégrité des données.

**R** = (NumeroAdherent\*, Nom, Prenom, DateDeNaissance, Rue, Ville, LatitudeVille, LongitudeVille, Etat, Telephone, Benevole, Annee\* , DateAdhesion, Montant, Don, MoyenDePaiement)

**DF1** : NumeroAdherent -> DateDeNaissance, Téléphone, Bénévole, Ville, Rue, Nom, Prenom, LatitudeVille, LongitudeVille, Etat

**DF2** : Ville, Etat -> LatitudeVille, LongitudeVille

**DF3** : NuméroAdhérent, Année > Montant, Don, MoyenDePaiement, DateAdhesion

**Clé candidate** : NumeroAdhérent, Année

En DFT par rapport à R (preuve algo diapo 17)



# 1NF

Les données de la table n'étaient pas en 1NF car certaines (comme le numéro de téléphone) n'étaient pas atomiques (il y avait plusieurs données dans une même cellule, donc réductibles)

h Dak	06 61 55 22 49	
nsas	05 61 42 55 24/ 07 61 25 78 37	
iesota		
ca		

Les données de la table n'étaient pas en 1NF car certaines (comme le numéro de téléphone) n'étaient pas atomiques (il y avait plusieurs données, donc réductibles dedans). J'ai remédié à ça en supprimant/modifié les cellules problématiques pour qu'elles ne contiennent qu'une donnée atomique. J'ai décidé de supprimer le numéro de téléphone commençant par 05 car aucun autre numéro n'est en 05 et le deuxième était en 07 comme le reste. J'ai aussi supprimé le contenu de la cellule qui comportait « / » car ce n'est pas un numéro de téléphone.

# 2NF

Une relation R est **2NF** si elle est **1NF** et qu'il n'existe pas d'attribut non primaire qui est en dépendance fonctionnelle avec un sous-ensemble de chaque clé candidate de la relation

A cause de DF1, la 2NF n'est pas respectée, donc on normalise ainsi :

## Décomposition de

**R** (NumeroAdherent\*, Nom, Prenom, DateDeNaissance, Rue, Ville, LatitudeVille, LongitudeVille, Etat, Telephone, Benevole, Annee\*, DateAdhesion, Montant, Don, MoyenDePaiement)

En :

**R1**(NumeroAdherent\*, DateDeNaissance, Téléphone, Bénévole, Ville, Rue, Nom, Prenom, LatitudeVille, LongitudeVille, Etat)

**R2**(NumeroAdherent\*, , Annee\* , DateAdhesion, Montant, Don, MoyenDePaiement)

Pour DF2, la 2NF est respectée

Pour DF3, la 2NF est respectée, grâce à la normalisation issue de la DF1

# 3NF

Une relation est en 3NF si elle est en 2NF et qu'aucun attribut non clé n'est transitivement dépendant de la clé primaire.

Pour que la relation soit en 3NF, il faut supprimer les dépendances transitives. En examinant la relation R1R1 après la décomposition en 2NF :

**R1**(NumeroAdherent, DateDeNaissance, Téléphone, Bénévole, Ville, Rue, Nom, Prenom, LatitudeVille, LongitudeVille, Etat)

**R2**(NumeroAdherent, Annee, DateAdhesion, Montant, Don, MoyenDePaiement) **R2**(NumeroAdherent, Annee, DateAdhesion, Montant, Don, MoyenDePaiement)

Pour **R1R1**, il existe une dépendance fonctionnelle transitive :

NumeroAdherent → Ville, Etat → (LatitudeVille, LongitudeVille, Etat)

Ville, Etat → (LatitudeVille, LongitudeVille)

Pour éliminer cette dépendance transitive, on décompose R1R1 en deux relations :

**R1a**(NumeroAdherent, DateDeNaissance, Téléphone, Bénévole, Ville, Etat, Rue, Nom, Prenom)

**R1b**(Ville\*, LatitudeVille, LongitudeVille, Etat\*)

Ainsi, les relations **R1a** et **R1b** sont en **3NF**.

# BCNF

Une relation est en BCNF si elle est en 3NF et si, pour toute dépendance fonctionnelle  $X \rightarrow Y$  est une super clé.

Pour vérifier que les relations sont en BCNF, nous devons vérifier que chaque dépendance fonctionnelle satisfait la condition de BCNF.

Dans les relations décomposées :

**R1a**(NumeroAdherent, DateDeNaissance, Telephone, Benevole, Ville, Rue, Nom, Prenom)

**R1b**(Ville, LatitudeVille, LongitudeVille, Etat) **R1b**(Ville, LatitudeVille, LongitudeVille, Etat)

**R2**(NumeroAdherent, Annee, DateAdhesion, Montant, Don, MoyenDePaiement)

Toutes les dépendances fonctionnelles ont pour déterminant une super clé, donc elles sont en BCNF.

# 4NF

Une relation est en 4NF si elle est en BCNF et qu'elle ne contient pas de dépendances multivaluées non triviales.

Pour que les relations soient en 4NF, il ne doit y avoir aucune dépendance multivaluée non triviale.

Examinons les relations R1a, R1b, et R2:

- R1a: Pas de dépendances multivaluées, chaque attribut dépend de NumeroAdherent
- R2: Pas de dépendances multivaluées, chaque attribut dépend de NumeroAdherent et Annee.

Les relations sont donc en 4NF.

# 5NF

Une relation est en 5NF si elle est en 4NF et que toute jointure dans la relation est implicite par ses clés candidates.

Pour vérifier la 5NF, nous devons nous assurer qu'aucune perte d'information ne se produit lors de la décomposition de la relation en ses projections.

R1a La décomposition en R1a et R1b ne produit pas de perte d'information et est déjà en 4NF.

R2 : Cette relation ne nécessite pas de décomposition supplémentaire.

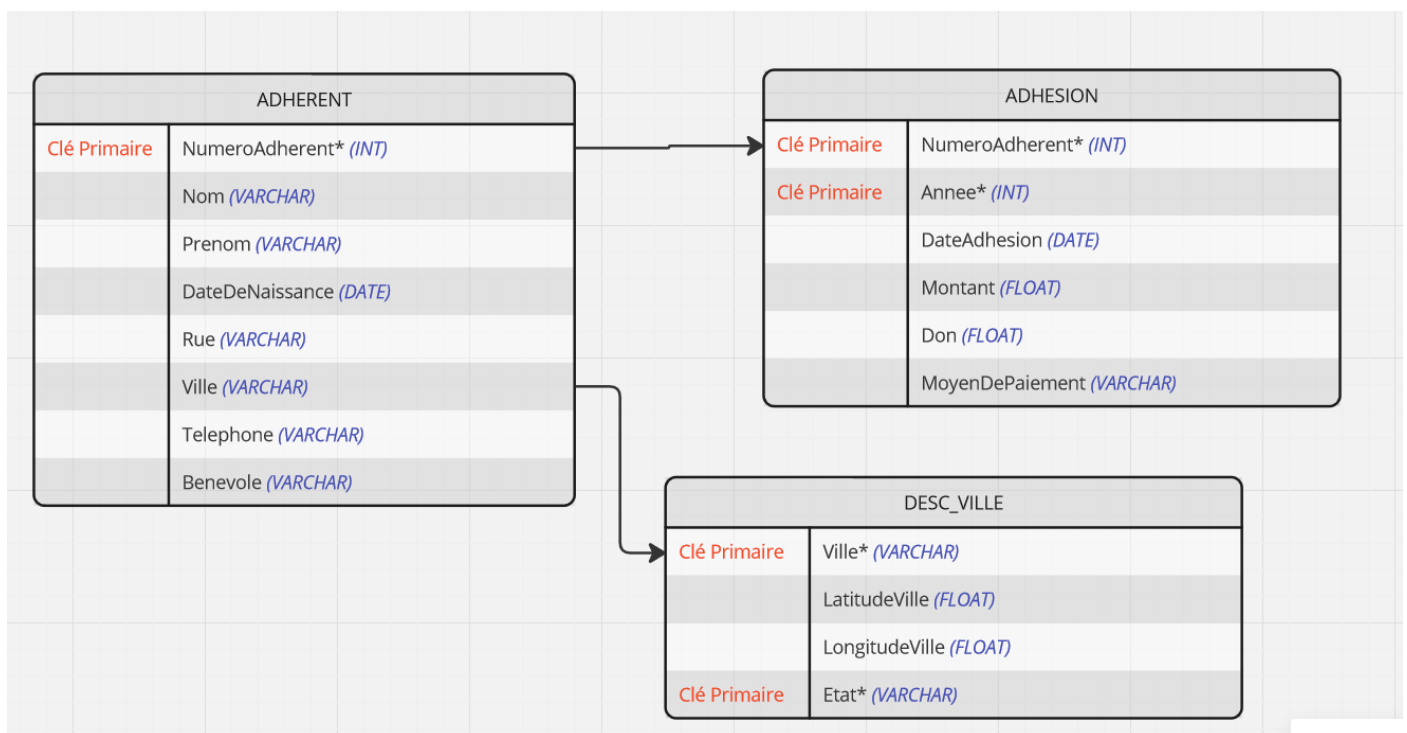
Ainsi, les relations R1a, R1b, et R2 respectent la 5NF car elles ne peuvent pas être décomposées davantage sans perte d'information.

La normalisation a permis de structurer la base de données de l'association AVEC de manière optimale, en réduisant la redondance des données et en améliorant leur intégrité. En suivant les principes de la 1NF, 2NF, et 3NF, nous avons créé une base de données efficace, cohérente et facile à maintenir. Cette structure normalisée facilite les opérations de mise à jour, réduit les anomalies et assure que chaque donnée est stockée de manière unique et précise.

# SCHEMA RELATIONNEL

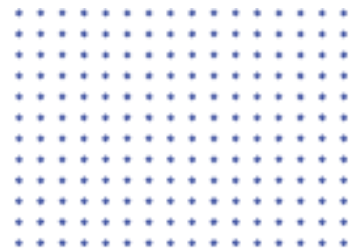


Le schéma relationnel est une représentation formelle de la structure de la base de données. Il définit la façon dont les tables sont organisées et reliées entre elles. Pour l'association AVEC, le schéma relationnel a été conçu en suivant les principes de normalisation, ce qui garantit une structure efficace et cohérente des données. Ce schéma est crucial pour assurer l'intégrité des données et faciliter les opérations de gestion et de requêtes.



Ce schéma relationnel est la base de la base de données relationnelle et assure que les informations sont stockées de manière optimale.

# REQUETES



## Montant moyen des adhésions

```
SELECT AVG(Montant)
```

```
FROM ADHESION;
```

Montant moyen des adhésions	
Expr1000	
	28,6019140

**Forces** : Accès toutes les données concernant les montants, les dons etc car centralisées.

**Faiblesses** : La table ADHESION ne contient pas d'informations sur les villes, requêtes plus lentes car beaucoup de données à traiter à la fois.

## Montant total des adhésions par type de paiement

```
SELECT MoyenDePaiement, SUM(Montant)
```

```
FROM ADHESION
```

```
GROUP BY MoyenDePaiement
```

Montant total des adhésions par type de paiement	
MoyenDePaiement	Expr1001
	398
chèque	1802
Espèce	879,090000145137
HelloAsso	4103
virement bancaire	140

**Forces** : Accès toutes les données concernant les montants, les dons etc car centralisées.

**Faiblesses** : Requêtes plus lentes car beaucoup de données à traiter à la fois.

## Moyenne d'âge des adhérents

```
SELECT AVG((Date() - DateDeNaissance) / 365)
```

```
FROM ADHERENT;
```

Moyenne d'âge des adhérents	
Expr1000	
	41,9245275

**Forces** : En utilisant la méthode Date(), on a toujours des mesures à jour car elle prend la date du jour.

**Faiblesses** : Les informations sur les villes, le dates d'adhésion ne sont pas disponibles dans la table ADHERENT dont est issue la requête.

### Nombre d'adhérents par années

```
SELECT Annee, COUNT(*)  
  
FROM (SELECT DISTINCT NumeroAdherent, Annee FROM ADHESION)  
  
GROUP BY Annee;
```

Nombre d'adhérents par années	
Annee	Expr1001
2015	23
2016	19
2017	14
2018	16
2019	33
2020	58
2021	114

**Forces** : Avec la sous requête SELECT DISTINCT on s'assure de bien distinguer les numéros d'adhérents pour chaque année

### Nombre d'adhérents par ville

```
SELECT Ville, COUNT(*)  
  
FROM (SELECT DISTINCT NumeroAdherent, Ville FROM ADHERENT)  
  
GROUP BY Ville;
```

Nombre d'adhérents par ville	
Ville	Expr1001
Albany	3
Annapolis	2
Atlanta	4
Augusta	4
Austin	1
Baton Rouge	4
Bismarck	5
Boise	2

**Forces** : Assure un comptage précis des adhérents uniques par ville grâce à DISTINCT.

**Faiblesses** : Performance potentiellement faible pour de grandes tables en raison de la sous-requête DISTINCT.

### Nombre d'adhésions par adhérent

```
SELECT a.Nom, a.Prenom, COUNT(*)  
  
FROM ADHESION AS ad INNER JOIN ADHERENT AS a ON ad.NumeroAdherent = a.NumeroAdherent  
  
GROUP BY a.Nom, a.Prenom;
```

**Forces** : Permet de compter les adhésions par adhérent avec précision grâce à la jointure.

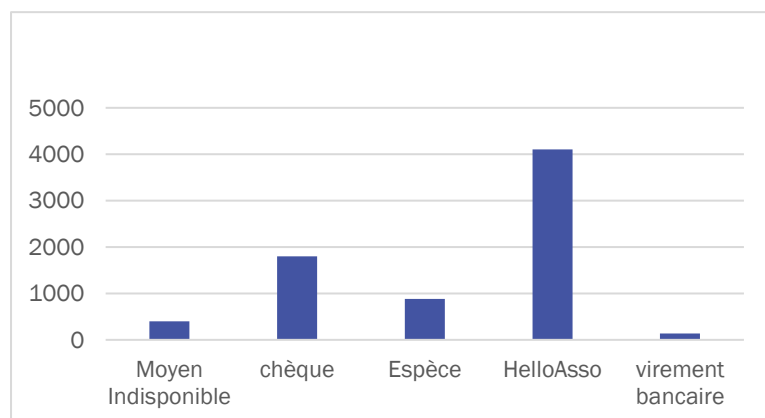
**Faiblesses** : La performance peut être affectée par la jointure si les tables sont volumineuses. La requête pourrait devenir plus complexe avec des tables de grande taille.

Nombre d'adhésions par adhérent		
Nom	Prenom	Expr1002
Adams	Benjamin	1
Adams	Chester	1
Adams	Dwight	1
Adams	James	1
Adams	Jimmy	1

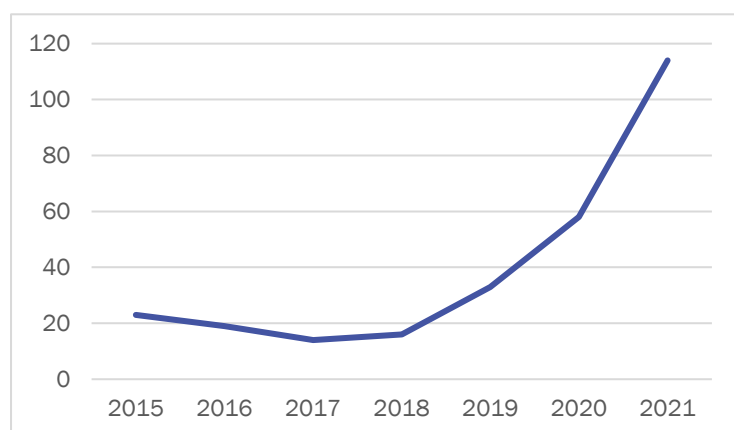
# VISUALISATIONS



Montant total des adhésions par type de paiement



Nombre d'adhérents par années



Nombre d'adhérents par villes

