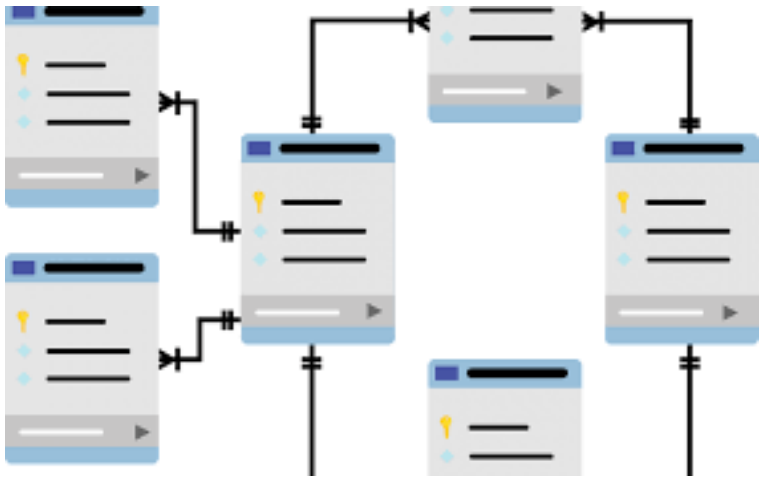
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

Rapport de projet de base de données

Base de données de services des citoyens



Diane LEBLANC-ALBAREL
Rand ASSWAD
Génie Mathématique

A l'attentien de :
Mme. Nathalie Chaignaud

12 mars 2019

Table des matières

1 Déscription du projet

1.1 Introduction

Notre projet est une base de données à destination de citoyens et le cadre administratif d'un pays. Le but est de pouvoir répertorier et d'avoir accès aux différents papiers et informations relatifs à un citoyen donné et de pouvoir également faire les démarches en lignes afin d'obtenir des aides, des papiers ou autres, le tout sur la même plateforme.

Actuellement, les plateformes pour percevoir des aides, payer des impôts ou demander certains papiers ou justificatifs sont rarement les mêmes ce qui peut compliquer les tâches des citoyens et des autorités qui pourraient avoir besoin de certaines informations sur certains citoyens. Le modèle nous a donc semblé pertinent et intéressant, de plus il semblait répondre aux critères obligatoires à savoir au moins 7 entités et au moins une relation ternaires.

1.2 Questions et requêtes

La base de données est adaptée pour deux applications web: l'une est mise à dispositions des résidents dans un pays et l'autre pour le corps administratif qui traitera les demandes effectuées sur le premier site. Voici quelques exemples de requêtes auxquelles notre base de données pourra effectuer:

- Quelles sont les aides sociales dont profite tel résident ?
- Combien a payé un tel résident en impôts depuis 10 ans ?
- Est-ce que la demande d'allocation chômage à été validée pour un résident donné ?
- Est-ce que un tel résident est en libre juridiquement ?
- Les informations de l'extrait d'état civil d'un résident (pour générer le document automatiquement).
- Effectuer une demande de renouvellement d'un passeport.

2 Conception

2.1 Modèle Entité/Association

Après plusieurs discussions nous avons réussi a poser les bases de notre modèle avec les différentes utilisations possibles. Le but de cette partie et de décrire précisément les différentes opérations possibles par les différents acteurs de la plateforme.

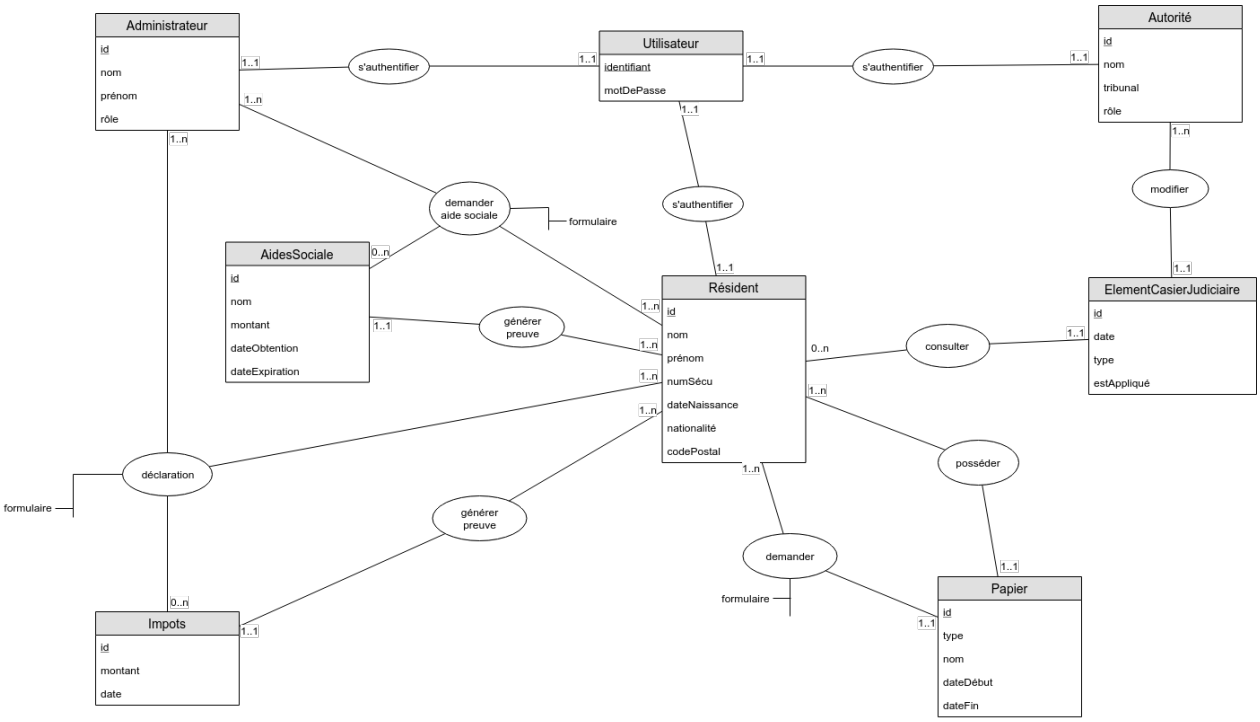


FIGURE 1 – Modèle Entité/Association

2.2 Utilisateurs

Trois types d'utilisateur différents pourraient utiliser la base de données : un résident, un administrateur (chargé de vérifier les différentes déclarations des résidents afin de vérifier leur véridicité) et un représentant de l'autorité pouvant consulter les données relative à chaque habitant.

Résident

Le résident doit tout d'abord pouvoir s'authentifier puis avoir accès à plusieurs services. Il doit pouvoir générer des justificatifs des différents papiers en sa possession (justificatifs de passeport, carte de séjour, carte d'identité, permis de conduire etc.) ainsi que des justificatifs des impôts qu'il auraient pu voir déjà versé ainsi qu'un justificatif des aides sociales qu'il aurait pu avoir perçu. De plus il doit également être en mesure d'effectuer des démarches pour l'obtention de ces différents papiers et la régularisation de sa situation en faisant différentes déclarations lui permettant de payer ses impôts ou de percevoir des aides. Ainsi le résident peut par exemple effectuer une déclaration de ses revenus qui devra être validé par un administrateur afin qu'il puisse payer les impôts correspondant a sa situation, de même pour les aides sociales.

Administrateur

L'administrateur a pour unique rôle de vérifier les déclarations faites par le résident. Les déclarations concernant ses revenus pour les impôts et les déclarations qu'il auraient effectué pour percevoir certaines bourses ou autres aides sociales.

Autorité

Cette entité peut consulter les informations relatives aux habitants et compléter le casier judiciaire des habitants. Hormis le casier judiciaire des habitant, il ne peut rien modifier mais il a un droit de regard sur toutes les informations concernant les utilisateurs enregistrés en tant que résidents.

2.3 Liens et entités

Papiers

Un résident peut posséder déjà certains papiers, il peut en posséder plusieurs et en possède au moins un : sa carte d'identité. De plus un papier ne peut appartenir qu'à une seule personne. Un résident peut demander un ou plusieurs papiers et sa demande sera muni d'un formulaire. D'où le lien « demander » entre l'entité « Résident » et l'entité « Papier » doublé d'un formulaire. Enfin lors de sa demande de papier ou lors ce qu'il consulte les papiers qu'il possède, un résident peut générer une preuve temporaire. Par exemple s'il demande une carte européenne d'assurance maladie il peut demander de générer un papier pouvant remplacer la carte pendant une certaine durée par exemple un mois. Une preuve temporaire ne peut correspondre qu'à un seul papier mais un résident peut demander plusieurs exemplaires de preuves temporaires de papiers.

Impôts et aides sociales

Un résident peut, tout d'abord, générer une preuve ou plusieurs preuves du payements passés de ses impôts, de plus il peut également faire une déclaration en remplissant un formulaire qui devrai être validé par un administrateur afin de pouvoir payer ses impôts. Cela explique donc la relation ternaire entre l'entité « Impôt », l'entité « Résident » et l'entité « Administrateur » : Pour établir une déclaration afin de payer des impôts il faut qu'un Résident remplisse un ou plusieurs formulaires correspondant chacun a un impôt précis (par exemple impôt sur le revenu), formulaires qui devront être validé par un Administrateur pour que la déclaration existe ; Pour les aides sociales on retrouve exactement le même principe qu'avec les impôts pour comprendre cette partie il suffit donc se référer à l'explication ci-dessus

Casier judiciaire

Chaque résident possède un casier judiciaire (celui peut être vide ou non). Il peut consulter son casier judiciaire mais ne peut le modifier. Les autorités peuvent consulter le casier judiciaire de chaque habitant (comme toutes les autres informations) mais peuvent également le compléter ci besoin en renseignant de nouvelles informations ou en supprimant certaines.

2.4 Schéma relationnel

2.4.1 Entités

- Utilisateur(**identifiant**, motDePasse)
- Administrateur(**id**, nom, prenom, role)
- Authorite(**id**, nom, prenom, role, tribunal)
- Resident(**id**, nom, prenom, numSecu, dateNaissance, nationalite, codePostal)
- Papier(**id**, type, nom, dateDebut, dateFin)
- Impots(**id**, montant, date)
- AidesSociale(**id**, nom, montant, dateObtention, dateExpiration)
- ElementCasierJudiciaire(**id**, date, type, estApplique)

2.4.2 Associations

2.4.2.1 Associations binaires

- PossessionPapier(**idPapier**, idResident)
- DemandePapier(**idPapier**, idResident, formulaire)
- PreuveAideSociale(**idAide**, idResident, dateDocuemnt)
- PreuveImpots(**idImpots**, idResident, dateDocument)
- LienJudiciaire(**elementJudiciaire**, residentConcerne, authorite)

2.4.2.2 Associations ternaires

- DeclarationImpots(**idImpots**, idResident, idAdministrateur)
- DemandeAideSociale(**idAide**, idResident, idAdministrateur)

TABLE 1 – résultats 1..6 (total: 6)

id	typeAide	frequence	montant	dateObtention	dateExpiration	etat	idResident	idAdmin
119	CROUS	m	null	2019-05-19	null	EnCoursDeValidation	44	14
223	CAF	m	null	2015-09-03	null	Refuse	44	3
292	RSA	m	227	2014-02-21	2015-02-16	Expire	44	39
397	CMU	a	337	2018-08-07	2019-08-02	Valide	44	33
483	RSA	m	360	2017-11-02	2018-10-28	Expire	44	2
488	RSA	m	140	2019-04-22	2020-04-16	Valide	44	7

2.4.3 Dénormalisation

- La relation *Utilisateur* peut être supprimée en rajoutant ses attributs vers *Administrateur*, *Authorite* et *Resident*.
- La relation *PossessionPapier* peut être simplement inclus dans dans la relation *Papier*.
- On est tempté de supprimer les relations *PreuveAideSociale* et *PreuveImpots*, mais il est important de garder les numéros et dates de ces documents afin de pouvoir y revenir en cas de problèmes, et ce n'est pas cohérent d'avoir ses informations dans les relations *AideSociale* et *Impots* car ce ne sont que des justificatifs et le résident peut en prendre autant qu'il lui faut.

2.4.4 Schéma relationnel final

- Administrateur(**id**, motDePasse, nom, prenom, role)
- Authorite(**id**, motDePasse, nom, prenom, role, tribunal)
- Resident(**id**, motDePasse, nom, prenom, numSecu, dateNaissance, nationalite, codePostal)
- Papier(**id**, idResident, type, nom, dateDebut, dateFin)
- Impots(**id**, montant, date)
- AidesSociale(**id**, nom, montant, dateObtention, dateExpiration)
- ElementCasierJudiciaire(**id**, date, type, estApplique)
- DemandePapier(**idPapier**, idResident, formulaire)
- PreuveAideSociale(**idAide**, idResident, dateDocument)
- PreuveImpots(**idImpots**, idResident, dateDocument)
- LienJudiciaire(**elementJudiciaire**, residentConcerne, authorite)
- DeclarationImpots(**idImpots**, idResident, idAdministrateur)
- DemandeAideSociale(**idAide**, idResident, idAdministrateur)

3 Requêtes

```
library(DBI)
pays <- dbConnect(RMySQL::MySQL(), dbname="pays", username="root", password="")
```

On initialise la valeur test suivante:

```
X <- 44
Y <- 77
```

Liste des aides sociales que perçoit un résident donné.

- **Reformultion :** Liste de toutes les aides **où** idResident $= X$
- **Algèbre relationelle :**

$$\sigma_{\text{idResident}=<X>}(\text{AideSociale})$$

- **SQL:**

```
SELECT * FROM AideSociale WHERE idResident = ?X;
```

TABLE 2 – résultats 1..25 (total: 25)

id	nom	prenom	nationalite	SUM(montant)
1	Caron	Maëlle	France	6364
2	Herve	Kimberley	France	3282
3	Pons	Quentin	France	1200
4	Lemoine	Maïlé	France	974
5	Julien	Éloïse	France	3330
6	Le gall	Simon	France	5376
7	Daniel	Cédric	Northern Mariana Islands	1656
8	Guillot	Adam	France	11184
9	Renard	Yanis	France	5592
10	Menard	Yasmine	France	8880
11	Roussel	Corentin	France	17729
12	Marchand	Félix	Slovenia	6280
13	Gauthier	Gabin	France	2779
14	Gonzalez	Robin	France	17240
15	Brunet	Jade	Venezuela	14150
16	Francois	Gaspard	France	1776
17	Gautier	Salomé	France	19046
19	Lefebvre	Alexia	Åland Islands	894
20	Henry	Dorian	France	9045
21	Noel	Loane	Holy See (Vatican City State)	2818
23	Bouvier	Maxence	France	7412
24	Leclerc	Félix	France	2181
25	Millet	Nolan	France	4328
26	Huet	Jeanne	France	7570
27	Lucas	Gabin	France	7247

Combien chaque résident a payé en impôt sur les derniers 10 ans ?

- **Reformulation :**
  - Liste d'impôts *Validés* **où** la date est $\geq$ année en cours -10 (sélections)
  - Le montant payé (projection)
  - La somme des montants de la liste par résident (fonction agrégation)
- **Algèbre relationnelle :**

$$I = \pi_{\text{montant, idResident}}(\sigma_{\text{date}\geq<\text{AnnéeEnCours}>-10 \quad \wedge \quad \text{etat="valide"}}(\text{Impots}))$$

  Afin d'obtenir la somme, nous n'avons qu'à faire: ${}_{\text{idResident}}\gamma_{\text{somme(montant)}}(\sigma_{\text{idResident}=<X>}(I))$
- **SQL :**

```
SELECT R.id, R.nom, R.prenom, R.nationalite, SUM(montant) FROM Impots, Resident R
    WHERE etat = "Valide"
    AND idResident = R.id
    AND dateDeclaration >= YEAR(CURDATE()) - 10
    GROUP BY idResident;
```

Si on cherche pour un résident en particulier, il suffit de faire

$$\gamma_{\text{somme(montant)}}(\sigma_{\text{idResident}=<\text{X}>}(I))$$

```
SELECT SUM(montant) FROM Impots
    WHERE etat = "Valide"
    AND idResident = ?X
    AND dateDeclaration >= YEAR(CURDATE()) - 10;
```

Est-ce que la demande d'allocation RSA a été validé pour un résident donné ?

- **Reformulation :** état de l'aide sociale **où** idResident $= X$ **et** typeAide = "RSA"

TABLE 3 – résultats 1..1 (total: 1)

x
2779

TABLE 4 – résultats 1..3 (total: 3)

x
Expire
Expire
Valide

— **Algèbre relationnelle :**

$$\pi_{\text{etat}}(\sigma_{\text{idResident}=<X>}(\sigma_{\text{typeAide}=\text{"RSA"}}(\text{AideSociale})))$$

— **SQL :**

```
SELECT etat FROM AideSociale
    WHERE idResident = ?X
    AND typeAide = "RSA";
```

Est ce que tel résident purge actuellement une peine ?

— **Reformulation :**
  — liste des elements du casier judiciaire **où** idResident = X **et** peine est en cours d'exécution.
  — si la liste est vide alors la réponse est "non" sinon "oui"

— **Algèbre relationnelle :**

$$\sigma_{\text{idResident}=<X>}(\sigma_{\text{peine}=\text{"EnCoursDExecution"}}(\text{ElementJudicaire})$$

— **SQL :**

```
SELECT * FROM ElementJudiciaire
    WHERE idResident = ?X
    AND peine = "EnCoursDExecution";
```

* (Possibilité de faire une projection sur une peine si on veut simplement avoir la liste des peines 'EnCoursDExecution' cela dépend de l'usage choisit. Afficher tous les éléments si la peine est en cours d'éxécution à l'avantage de permettre 'en savoir plus sur la dite peine).

Effectuer une demande de renouvellement d'un passeport

Dans notre vision de cette application il s'agit d'un procédure qui ne demande pas l'intervention d'un *Administrateur* car toutes les informations nécessaires sont déjà dans cette base de donnée. Par exemple, en supposant qu'un citoyen qui purge une peine n'a pas le droit d'obtenir un passeport il s'agit donc de faire la requête précédente. Pour le cas d'un passeport, le tarif et la durée sont déjà fixés, il reste plus qu'à remplir la base de donné par ces informations récupérées. Une instance sera donc créée avec l'état "EnCoursDeValidation" jusqu'à la vérification du paiement.

```
INSERT INTO Papier (idResident, typePapier, dateDebut, dateFin, etat, tarif)
VALUES (?X, "passeport", CURRENT_DATE(), CURRENT_DATE() + 10, "EnCoursDeValidation", 86);
```

Lister tout les administrateur dans l'ordre de nombre de tâches dont ils sont chargé

On voudrait étendre la relation Administrateur en y rajoutant le nombre de tâches dont chaque administrateur est chargé actuellement, cette requête est très utile dans le cas où on voudrait automatiser l'attribution des tâches.

TABLE 5 – résultats 1..1 (total: 1)

id	dateElement	typeElement	peine	idResident	idAuthorite
117	2006-01-20	null	EnCoursDExecution	44	9

Petite remarque: on aurait pu rendre l'application plus réaliste en rajoutant un champs *disponibilité* qui indiquera si l'employé est disponible ou pas (en arrêt maladie, congé maternel/paternel, vacances, etc), ce qui exclura les employés non disponible des résultats de cette requête.

Plongeons nous alors dans cette requête !

- Un administrateur est chargé d'examiner les demandes d'aides sociales et des déclarations d'impots, on voudrais donc d'abord séléctionner tous les éléments de ces deux relations dont l'état est *en cours de validation*.
- Ensuite, on fait une jointure externe gauche entre Administrateur et la relation obtenue.
- A l'aide de la fonction d'agrégation de comptage, compte le nombre d'occurances par administrateur.

$$T = \pi_{\text{id,idAdmin}}(\sigma_{\text{etat="EnCoursDeValidation"}}(\text{AideSociale})) \cup \pi_{\text{id,idAdmin}}(\sigma_{\text{etat="EnCoursDeValidation"}}(\text{Impots}))$$

$$_{\text{id}}\gamma_{\text{compter(T.id)}}(\text{Administrateur} \underset{\text{id=T.idAdmin}}{\bowtie} T)$$

```
SELECT Administrateur.id, email, nom, prenom, COUNT(ALL T.id) as nbTaches from Administrateur
LEFT OUTER JOIN (
    SELECT id, idAdmin FROM AideSociale WHERE etat = "EnCoursDeValidation"
    UNION ALL SELECT id, idAdmin FROM Impots WHERE etat = "EnCoursDeValidation"
) AS T ON T.idAdmin = Administrateur.id
GROUP BY Administrateur.id
ORDER BY nbTaches;
```

Donner pour chaque aide sociale attribuée le montant total reçu

```
DROP VIEW IF EXISTS AideTotale;
```

```
CREATE VIEW AideTotale AS
SELECT id,
    CASE frequence
        WHEN 'a' THEN montant * (timestampdiff(year, dateObtention, dateExpiration))
        WHEN 'm' THEN montant * (timestampdiff(month, dateObtention, dateExpiration))
        ELSE montant
    END AS montantTotal
FROM AideSociale;
```

```
SELECT AideSociale.id, typeAide, frequence, montant, montantTotal, dateObtention, dateExpiration, idRe
FROM AideSociale, AideTotale
WHERE AideSociale.id = AideTotale.id;
```

Table 6 – résultats 1..30 (total: 50)

id	email	nom	prenom	nbTaches
16	ante.lectus.convallis@CurabiturdictumPhasellus.ca	Dupont	Tatiana	0
30	diam@eu.net	Bonnet	Malik	0
1	auctor@turpisvitaepurus.org	Charpentier	Maéva	0
40	Aliquam.erat.volutpat@asollicitudinorci.edu	Gauthier	Amine	0
47	nonummy.ac.feugiat@ipsum.edu	Nicolas	Arthur	0
22	cursus.vestibulum.Mauris@dictumPhasellus.co.uk	Gauthier	Paul	0
35	risus.at.fringilla@hendrerit.ca	Bourgeois	Gaspard	0
17	eu.eleifend@nullaDonecnon.edu	Simon	Marine	0
49	mauris.Integer@erat.ca	Laine	Élise	0
38	ullamcorper.Duis.cursus@purusinmolestie.co.uk	Dupuy	Esteban	0
45	malesuada@vitae.net	Philippe	Clémence	0
34	euismod.enim.Etiam@massaQuisque.com	Chevalier	Juliette	1
41	auctor.Mauris.vel@cursuspurus.co.uk	David	Thibault	1
23	neque@Aliquamerat.edu	Collin	Yanis	1
37	elit@pedeNuncsed.ca	Carre	Gabriel	1
5	fames.ac@enim.co.uk	Gay	Syrine	1
44	semper.pretium.neque@mollisDuis.edu	Louis	Catherine	1
12	dictum.placerat.augue@vitaeerat.ca	Pereira	Jade	1
26	arcu@sitametdapibus.ca	Pierre	Elsa	1
8	magna@tempus.com	Meyer	Alexandre	1
36	Nullam.velit.dui@aliquameu.edu	Boyer	Maéva	1
4	enim@mollisduiin.com	Dumont	Renaud	1
43	justo.Proin@cursuseteros.edu	Albert	Lamia	1
25	fames.ac@dignissimtempor.ca	Dumas	Carla	1
39	Nunc.mauris.sapien@dictumPhasellusin.net	Renault	Lauriane	1
21	odio@ametante.edu	Roger	Justine	1
3	nunc.Quisque.ornare@pharetra.org	Nicolas	Noë	1
42	cursus.purus@congue.ca	Dupuis	Lilou	1
31	ullamcorper.Duis@velitegestaslacinia.com	Masson	Sara	1
6	ipsum@sed.net	Girard	Hugo	1

TABLE 7 – résultats 1..30 (total: 500)

id	typeAide	frequence	montant	montantTotal	dateObtention	dateExpiration	idResident
1	RSA	m	246	2706	2017-03-05	2018-02-28	90
2	PrimeNaissance	p	562	562	2017-04-14	2017-04-14	18
3	CMU	a	302	0	2016-08-08	2017-08-03	72
4	CMU	a	401	0	2015-08-09	2016-08-03	36
5	RSA	m	284	3124	2016-03-24	2017-03-19	60
6	CROUS	m	402	4422	2017-08-06	2018-08-01	22
7	CAF	m	62	682	2011-04-28	2012-04-22	59
8	CAF	m	null	null	2019-02-22	null	4
9	RSA	m	164	1804	2013-12-27	2014-12-22	19
10	CROUS	m	178	1958	2017-02-02	2018-01-28	43
11	CAF	m	180	1980	2014-05-04	2015-04-29	77
12	CROUS	m	294	3234	2013-04-29	2014-04-24	30
13	CAF	m	156	1716	2013-02-10	2014-02-05	68
14	RSA	m	null	null	2014-10-18	null	17
15	PrimeNaissance	p	529	529	2012-07-02	2012-07-02	27
16	RSA	m	158	1738	2014-08-11	2015-08-06	38
17	CROUS	m	null	null	2017-02-14	null	29
18	PrimeNaissance	p	747	747	2014-11-04	2014-11-04	7
19	CMU	a	258	0	2012-11-13	2013-11-08	38
20	CAF	m	null	null	2016-05-23	null	69
21	CMU	a	403	0	2018-10-21	2019-10-16	95
22	RSA	m	397	4367	2016-07-07	2017-07-02	11
23	CMU	a	431	0	2011-03-21	2012-03-15	89
24	PrimeNaissance	p	null	null	2019-03-11	null	75
25	CROUS	m	null	null	2012-10-24	null	50
26	PrimeNaissance	p	657	657	2012-04-04	2012-04-04	99
27	CROUS	m	432	4752	2016-10-03	2017-09-28	87
28	CAF	m	null	null	2019-03-28	null	20
29	RSA	m	490	5390	2013-08-26	2014-08-21	78
30	CMU	a	null	null	2013-11-22	null	1