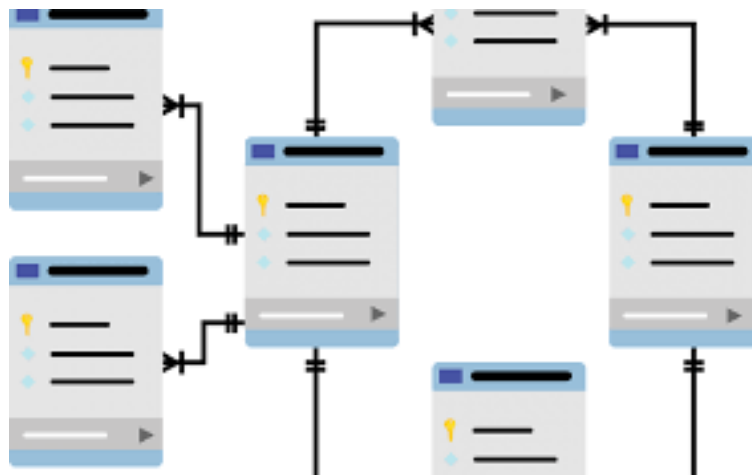


RAPPORT DE PROJET DE BASE DE DONNÉES

Base de donnees de services des citoyens



Diane LEBLANC-ALBAREL
Rand ASSWAD
Génie Mathématique

A l'attention de :
Mme. Nathalie Chaignaud

12 mars 2019

Table des matières

1	Description du projet	2
1.1	Introduction	2
1.2	Questions et requêtes	2
2	Conception	2
2.1	Modèle Entité/Association	2
2.2	Utilisateurs	3
2.3	Liens et entités	3
2.4	Schéma relationnel	4
3	Modifications	4
3.1	Dénormalisation	4
3.2	Tableaux	5
4	Requêtes	5

1 Description 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 a 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 à poser les bases de notre modèle avec les différentes utilisations possibles. Le but de cette partie est de décrire précisément les différentes opérations possibles par les différents acteurs de la plateforme.

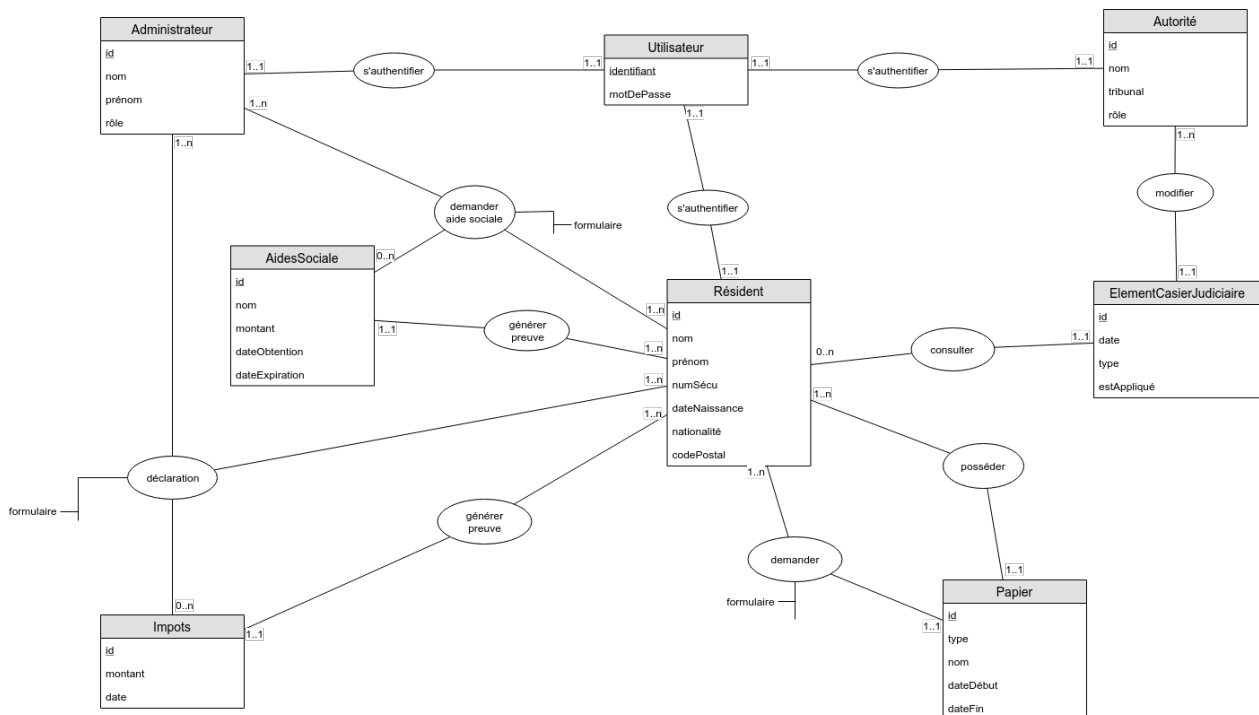


FIGURE 1 – Modèle Entité/Association

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 véracité) 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 paiements passés de ses impôts, de plus il peut également faire une déclaration en remplissant un formulaire qui devra 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)
- Autorite(**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, dateDocument)
- PreuveImpots(**idImpots**, idResident, dateDocument)
- LienJudiciaire(**elementJudiciaire**, residentConcerne, autorite)

2.4.2.2 Associations ternaires

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

2.4.3 Dénormalisation

- La relation *Utilisateur* peut être supprimée en rajoutant ses attributs vers *Administrateur*, *Autorite* et *Resident*.
- La relation *PossessionPapier* peut être simplement inclus dans la relation *Papier*.
- On est tenté 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)
- Autorite(**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, autorite)
- DeclarationImpots(**idImpots**, idResident, idAdministrateur)
- DemandeAideSociale(**idAide**, idResident, idAdministrateur)

3 Modifications

3.1 Dénormalisation

Toutes les relations 1..n – 1..1 peuvent être simplifier en ajoutant dans la table 1..1 une clef pour accéder aux éléments de la table 1..n. (Bonnes pratiques). On peut donc ajouter à **Impot**, à **Papier**, **ElementsJuridique**, et **AidesSociales** une clefs étrangère vers **IdResident** représentant le résident auquel l'Impot, le Papier, l'aide sociales ou l'élément juridique est associé.

3.2 Tableaux

Pour les tables **Papier**, **Impot** et **AidesScociales**: ajout d'un élément **etat** qui prendra les valeurs **EnCoursDeValidation**, **Validé**, **Refusé** ou **Expiré**.

Cela va permettre pour faire une demande de Papier, Aides sociales ou une de déclaration d'impôt, très simplement. Le Résident remplit un formulaire et le soumet sur la plateforme. La requête derrière sera simplement un insert dans la table voulu avec tous les éléments indiqués par le résident, seulement, l'**etat** du Papier, de l'Aide Sociale ou de l'impôt créé sera **EnCoursDeValidation**, pour passer à **Validé** ou **Refusé** il faudrait qu'un administrateur la valide.

Cela permet de supprimer facilement les relations ternaires non souhaitées. Cela permet aussi de garder pendant un temps souhaité les différents éléments du dossier d'un Résident même si ceux-ci sont expirés ou sont des demandes refusées (on peut imaginer qu'après un temps ils sont supprimés de la base de données. Ainsi par exemple si un Résident veut faire une demande de Carte d'identité, il n'a qu'à remplir le formulaire en indiquant **carte d'identité** en nom ainsi que les autres informations utiles, il en découlera un insert dans la table **Papier** avec la création du papier demandé associé au formulaire, seulement l'état du papier sera **EnCoursDeValidation** jusqu'à ce qu'un administrateur le passe en **Validé** ou **Refusé**.

Pour la table **ElementsJudiciaires** : ajout d'un élément **peine** qui pourrait prendre les valeurs **EnCoursDExecution**, **Execute**, **NonEligible**.

La valeur **NonEligible** sera prise s'il n'y a pas lieu d'avoir une peine pour l'élément du casier. Les valeurs **EnCoursDExecution** et **Execute** parlent d'elles-mêmes et permettront ainsi un meilleur suivi de la personne et surtout de pouvoir facilement faire la requête **Est-ce qu'un résident purge actuellement une peine?** puisqu'alors il suffit juste de sélectionner les **ElementsJudiciaires** correspondant à ce résident puis de regarder les valeurs de l'élément **peine** pour chacun d'entre eux, si au moins l'un d'eux a la valeur **EnCoursDExecution** la réponse sera oui.

4 Requêtes

Liste des aides sociales que perçoit un résident **X**. - Reformulation : Liste de toutes les aides où l'IdRésident = <IdX> - Algèbre relationnelle : $\sigma_{IdResident=<IdX>}(AidesSociales)$ - MySQL: `SELECT * FROM AideSociales WHERE IdResident = <IdX>;`

Combien a payé un résident en impôt depuis 10 ans - Reformulation : Liste des montants des impôts où IdRésident = <IdX> ET la date est supérieure ou égale à l'année en cours - 10. (On fait la somme des montants en dehors d'une requête MySQL). - Algèbre relationnelle : $\pi_{montant}(\sigma_{IdResident=<IdX>}(\sigma_{date \geq <anneeEnCours>-10}(Impots)))$ - MySQL : `SELECT montant FROM Impots WHERE IdResident = <IdX> AND date = <AnneeEnCours> - 10;`

* Il n'y a plus qu'à faire une boucle sur les montants obtenus pour les sommer

Est-ce que la demande d'allocation chômage a été validée pour un résident donné. - Reformulation : Etat de de l'Aides Sociales où IdRésident = <IdX> ET où nom = <Chomage> - Algèbre relationnelle : $\pi_{etat}(\sigma_{IdResident=<IdX>}(\sigma_{nom='Chomage'}(AidesSociales)))$ - MySQL : `SELECT etat FROM AidesSociales WHERE IdResident = <IdX> AND nom = 'Chomage';`

Est-ce que le résident purge actuellement une peine ?

- Reformulation : Liste des Elements du Casier Judiciaire où IdRésident = <IdX> ET peine = 'EnCoursDExecution' (Si liste vide alors réponse non sinon réponse oui)
- Algèbre relationnelle : $\sigma_{IdResident=<IdX>}(\sigma_{peine='EnCoursDExecution'}(ElementsCasierJudiciaire))$
- MySQL : `SELECT * FROM ElementsCasiersJudiciaire WHERE IdResident = <IdX> 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 choisi. Afficher tous les éléments si la peine est en cours d'exécution à l'avantage de permettre d'en savoir plus sur la dite peine).

Information sur l'extrait d'état civil d'un résident **X** - Reformulation : Liste des attributs de Papier où IdRésident = <IdX> ET où type = EtatCivile - Algèbre relationnelle : $\sigma_{IdResident=<IdX>}(\sigma_{type='EtatCivile'}(Papier))$ - MySQL : `SELECT * FROM Papier WHERE IdResident = <IdX> AND type = 'EtatCivile';`

Effectuer une demande de renouvellement d'un passeport - Prérequis : Le Résident rempli un formulaire puis le soumet, les valeurs qu'il aurait renseigné seront mises après vérifications dans les variables : **IdDemande**, **IdResident**, **typeDemande**, **nomDemande**, **DateDebutDemande**, **DateFinDemande**. Le résident n'aurait en soit qu'a renseigné le type de demande qu'il souhaite faire ici **Passeport** le reste peut être obtenu sans son intervention. Dans tous les cas, ces informations doivent être disponible avant toute requête. - Reformulation : Créer un nouveau Papier qui a pour valeur les valeurs des variables récupéré. - MySQL :
INSERT INTO Papier (ID, IdResident, type, nom, DateDebut, DateFin) VALUES (IdDemande, IdResident, typeDemande, nomDemande, DateDebutDemande, DateFinDemande);