



OGODJA Joseph  
CHABI Ismaël

Supervisé par : M. MAHMOUD EL HAMLAOUI

Projet de base de données

## 1- Description fonctionnelle :

Pour contrôler le phénomène de fuite des cerveaux, le gouvernement béninois a opté pour un programme contractuel pour les boursiers d'Etat béninois. L'objectif à long terme est de les rapatrier au Bénin pour servir dans la fonction publique. Chaque année en moyenne 5000 étudiants sont boursiers pour des formations très larges et variées sur le territoire béninois ou à l'étranger. Bon nombre de ceux-ci à la fin de leur formation, s'installent à l'extérieur pour acquérir de l'expérience et du savoir-faire et parfois y restent définitivement.

Notre projet, permettra de mettre à disposition de toutes les parties prenantes une base d'informations fiables pour avoir une correspondance entre les profils que l'Etat recherche et les ressources humaines que lui fournit les boursiers.

Un **bénéficiaire** est identifié par son **nom**, son **prénom**, sa **date de naissance**, son **statut matrimonial**, ses **coordonnées**, sa **formation**, la **bourse** dont il bénéficie, son **expérience** professionnelle, son **métier**, son **appartenance politique** et sa **santé**.

La bourse du bénéficiaire est décrite par le **montant alloué** qui est réparti en **frais de vie**, **frais de formation** et le **déplacement**.

Le **déplacement** est décrit par les **frais de déplacement**, la **date de départ** et la **date d'arrivée**.

La **formation** est décrite par son **nom**, le **pays de formation** et la **durée de formation** (incluant la **date de début** et **de fin**) et le diplôme associé.

L'**expérience professionnelle** du boursier est caractérisée par le **poste** et période de l'**année** qu'il y a passé.

Le **métier de prédilection** du boursier est caractérisé par le **nom du métier** et le **salaire associé**.

Les **coordonnées** du boursier sont données par son **adresse**, son **mail**, son **téléphone** et sa **position actuelle**.

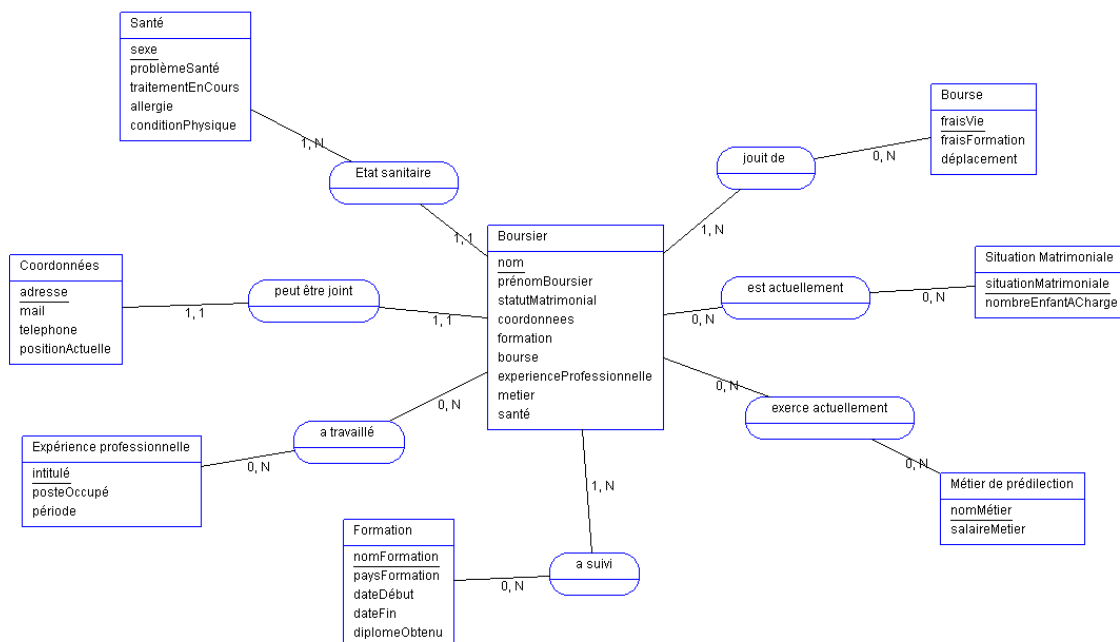
Le **statut matrimonial** comporte sa **situation matrimoniale**, les **enfants** à sa charge.

La **santé** du boursier regroupe son **sexe**, ses **problèmes de santé** nécessitant des soins particuliers, les **traitements** qu'ils suit, ses **allergies**, et sa **condition physique**.

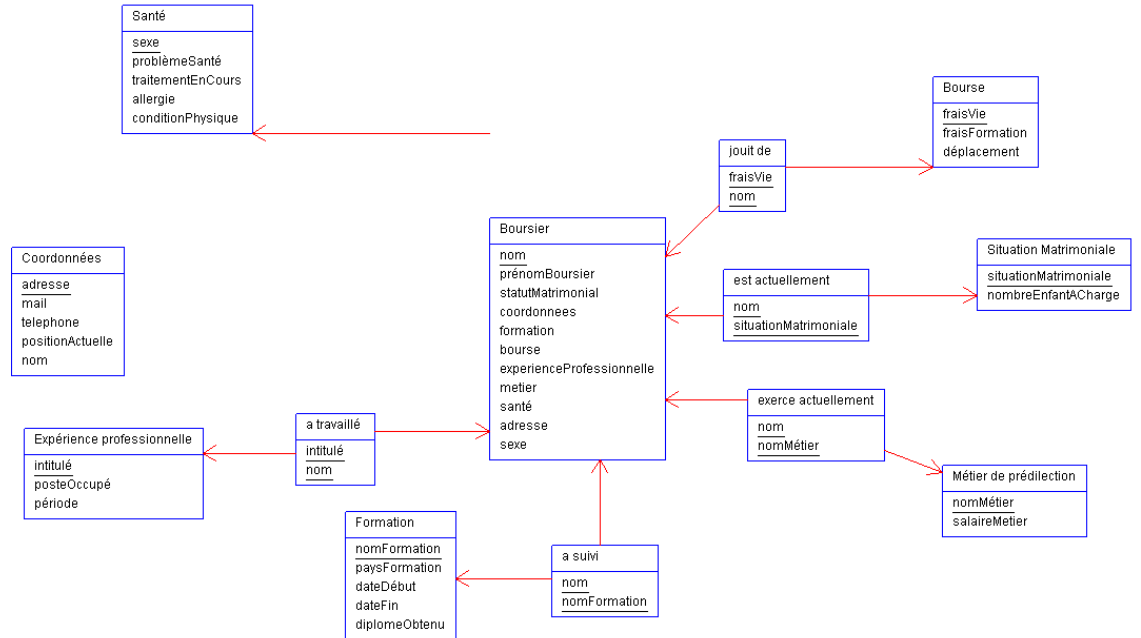
## 2- Dictionnaire de données

Nom	ID	Type	Taille	Utilisé
nom	nom	CHAR	128	✓
nomBoursier	nomBoursier	CHAR	128	✓
prénomBoursier	prénomBoursier	CHAR	128	✓
dateNaissanceBoursier	dateNaissanceBoursier	DATETIME	9	✓
sexe	sexe	CHAR	1	✓
problèmeSanté	problèmeSanté	CHAR	1024	✓
traitementEnCours	traitementEnCours	CHAR	1024	✓
allergie	allergie	CHAR	1024	✓
conditionPhysique	conditionPhysique	CHAR	1024	✓
situationMatrimoniale	situationMatrimoniale	CHAR	128	✓
nombreEnfantACharge	nombreEnfantACharge	BIGINT	8	✓
adresse	adresse	CHAR	128	✓
mail	mail	CHAR	128	✓
telephone	telephone	BIGINT	12	✓
positionActuelle	positionActuelle	CHAR	1024	✓
nomMétier	nomMétier	CHAR	128	✓
saireMétier	saireMétier	BIGINT	128	✓
intitulé	intitulé	CHAR	128	✓
posteOccupé	posteOccupé	CHAR	128	✓
période	période	CHAR	128	✓
nomFormation	nomFormation	CHAR	128	✓
paysFormation	paysFormation	CHAR	128	✓
dateDébut	dateDébut	DATETIME	9	✓
dateFin	dateFin	DATETIME	9	✓
diplomeObtenu	diplomeObtenu	CHAR	128	✓
fraisDéplacement	fraisDéplacement	BIGINT	128	✓
dateDépart	dateDépart	DATETIME	9	✓
dateArrivée	dateArrivée	DATETIME	9	✓
fraisVie	fraisVie	BIGINT	10	✓
fraisFormation	fraisFormation	BIGINT	10	✓
déplacement	déplacement	BIGINT	0	✓
statutMatrimonial	statutMatrimonial	CHAR	128	✓
coordonnees	coordonnees	CHAR	128	✓
formation	formation	CHAR	128	✓
bourse	bourse	CHAR	128	✓


### 3- Modèle EA



### 4- Modèle Relationnel




## 5- Fichier SQL pour la création de la BD

▼ 11 ■■■■■ tables/bourse.sql 

... ... @@ -0,0 +1,11 @@

```
1 + CREATE TABLE bourse (
2 +   ID_bourse SERIAL,
3 +   Nom_de_la_bourse varchar(1024) NOT NULL DEFAULT '',
4 +   Montant_Alloué int DEFAULT NULL,
5 +   Frais_de_vie int DEFAULT NULL,
6 +   Frais_de_formation int DEFAULT NULL,
7 +   Deplacement varchar(1024) DEFAULT NULL,
8 +   PRIMARY KEY (ID_bourse),
9 +   UNIQUE(ID_bourse)
10 + );
11 +
```

▼ 21 ■■■■■ tables/boursier.sql 

... ... @@ -0,0 +1,21 @@

```
1 + CREATE TABLE boursier (
2 +   Id_boursier SERIAL,
3 +   Nom varchar(1024) DEFAULT NULL,
4 +   PrenomBoursier varchar(1024) DEFAULT NULL,
5 +   ID_statu_matrimonial SERIAL,
6 +   Coordonnees SERIAL,
7 +   Formation SERIAL,
8 +   Bourse SERIAL,
9 +   ExperienceProfessionnelle SERIAL,
10 +   Metier SERIAL,
11 +   Sante SERIAL,
12 +   PRIMARY KEY (Id_boursier),
13 +   FOREIGN KEY (ID_statu_matrimonial) REFERENCES statut_matrimonial(ID_statu_matrimonial),
14 +   FOREIGN KEY (Coordonnees) REFERENCES coordonnees(ID_coordonees),
15 +   FOREIGN KEY (Formation) REFERENCES formation(ID_formation),
16 +   FOREIGN KEY (Bourse) REFERENCES bourse(ID_bourse),
17 +   FOREIGN KEY (ExperienceProfessionnelle) REFERENCES experience_professionnelle(ID_experience_pro),
18 +   FOREIGN KEY (Metier) REFERENCES metier(ID_metier),
19 +   FOREIGN KEY (Sante) REFERENCES sante(ID_sante),
20 +   UNIQUE(Id_boursier)
21 + );
```

▼ 8 ■■■■■ tables/coordonnées.sql 

... ... @@ -0,0 +1,8 @@

```
1 + CREATE TABLE coordonnées (
2 +   ID_coordonees SERIAL,
3 +   Adresse varchar(1024) DEFAULT NULL,
4 +   Mail varchar(1024) DEFAULT NULL,
5 +   Telephone int DEFAULT NULL,
6 +   Position_actuelle varchar(1024) DEFAULT NULL,
7 +   UNIQUE(ID_coordonees)
8 + );
```

▼ 9 ■■■■■ tables/deplacement.sql 


... ... @@ -0,0 +1,9 @@

```
1 + CREATE TABLE déplacement (
2 +   ID_deplacement SERIAL,
3 +   Moyen_de_déplacement varchar(1024) NOT NULL DEFAULT '',
4 +   Frais_de_déplacement int DEFAULT NULL,
5 +   Date_de_départ date DEFAULT NULL,
6 +   Date_d_arrivée date DEFAULT NULL,
7 +   PRIMARY KEY (Moyen_de_déplacement),
8 +   UNIQUE(ID_deplacement)
9 + );
```

▼ 9 ■■■■■ tables/experience\_pro.sql 


... ... @@ -0,0 +1,9 @@

```
1 + CREATE TABLE expérience_professionnelle (
2 +   ID_experience_pro SERIAL,
3 +   PosteOccupé varchar(1024) DEFAULT NULL,
4 +   Durée varchar(1024) DEFAULT NULL,
5 +   Période varchar(1024) DEFAULT NULL,
6 +   UNIQUE(ID_experience_pro)
7 + );
8 +
9 + -- Handle experiences by providing them in an arrows and handle the output later in the developpement.
```

▼ 9 ■■■■■ tables/formation.sql 

... ... @@ -0,0 +1,9 @@

```
1 + CREATE TABLE formation (  
2 +   ID_formation SERIAL,  
3 +   Nom_de_la_formation varchar(1024) DEFAULT NULL,  
4 +   Pays_de_formation varchar(1024) DEFAULT NULL,  
5 +   Date_de_début date DEFAULT NULL,  
6 +   Date_de_fin date DEFAULT NULL,  
7 +   Diplôme_associé varchar(1024) DEFAULT NULL,  
8 +   UNIQUE(ID_formation)  
9 + );
```

▼ 6 ■■■■■ tables/metier.sql 

... ... @@ -0,0 +1,6 @@

```
1 + CREATE TABLE metier (  
2 +   ID_metier SERIAL,  
3 +   Nom_du_métier varchar(1024) DEFAULT NULL,  
4 +   Salaire int DEFAULT NULL,  
5 +   UNIQUE(ID_metier)  
6 + );
```

▼ 9	■■■■■	tables/santé.sql	📄
...	...	@@ -0,0 +1,9 @@	
1	+	CREATE TABLE santé (	
2	+	ID_sante SERIAL,	
3	+	Sexe varchar(1024) DEFAULT NULL,	
4	+	Problème_de_santé varchar(1024) DEFAULT NULL,	
5	+	Traitement_suivi varchar(1024) DEFAULT NULL,	
6	+	Allergies varchar(1024) DEFAULT NULL,	
7	+	Conditions_physique varchar(1024) DEFAULT NULL,	
8	+	UNIQUE(ID_sante)	
9	+	);	

▼ 8	■■■■■	tables/statut_matrimonial.sql	📄
...	...	@@ -0,0 +1,8 @@	
1	+	CREATE TABLE statut_matrimonial (	
2	+	ID_statu_matrimonial SERIAL,	
3	+	Statut_Matrimonial varchar(1024) DEFAULT NULL,	
4	+	Conjoint varchar(1024) DEFAULT NULL,	
5	+	Enfants_à_charge varchar(1024) DEFAULT NULL,	
6	+	UNIQUE(ID_statu_matrimonial)	
7	+	);	
8	+	-- We need to provide all available cases.	

## 6- Requêtes SQL nécessaires pour votre projet

Afin de remplir la table boursier nous avons de rempli les tables adjacentes et récupéré leurs clés principales. La fonction implémentée est la même pour les autres tables adjacentes.

Voici ci-contre la fonction de remplissage de la table Statut matrimonial.



```
... @@ -0,0 +1,294 @@
1 + #!usr/bin/python3
2 +
3 + import psycopg2
4 +
5 + from connect import config
6 +
7 + def get_ID_statu_matrimonial():
8 +     ID_statu_matrimonial = None
9 +     Statut_Matrimonial = input("Statut Matrimonial : ")
10 +     Conjoint = input("Conjoint : ")
11 +     Enfants_à_charge = input("Enfants à charge : ")
12 +     query = """INSERT INTO statut_matrimonial (Statut_Matrimonial, Conjoint, Enfants_à_charge)
13 +     VALUES (%s, %s, %s) RETURNING ID_statu_matrimonial;"""
14 +     conn = None
15 +     try:
16 +         # read the connection parameters
17 +         params = config()
18 +
19 +         # connect to the PostgreSQL server
20 +         conn = psycopg2.connect(**params)
21 +         cur = conn.cursor()
22 +
23 +         cur.execute(query, (Statut_Matrimonial, Conjoint, Enfants_à_charge))
24 +
25 +         ID_statu_matrimonial = cur.fetchone()[0]
26 +
27 +         # commit the changes
28 +         conn.commit()
```

La suite est sur le lien github [Database\\_boursier](#).

## 7- Script pour intégrer les requêtes au projet.

Nous avons implémenté les scripts pour ajouter des boursiers en modifier les informations. Le module python utilisé est psycopg2 qui facilite l'envoi des requêtes à la base de donnée PostgreSQL avec Python.

Les scripts sont sur le lien github [Database\\_boursier](#).

## 8- Exploitation des requêtes en python

La suite est sur le lien github [Database\\_boursier](#).