

Document documentant l'ensemble des actions faites pour l'étape 1 de l'ECDF 2.

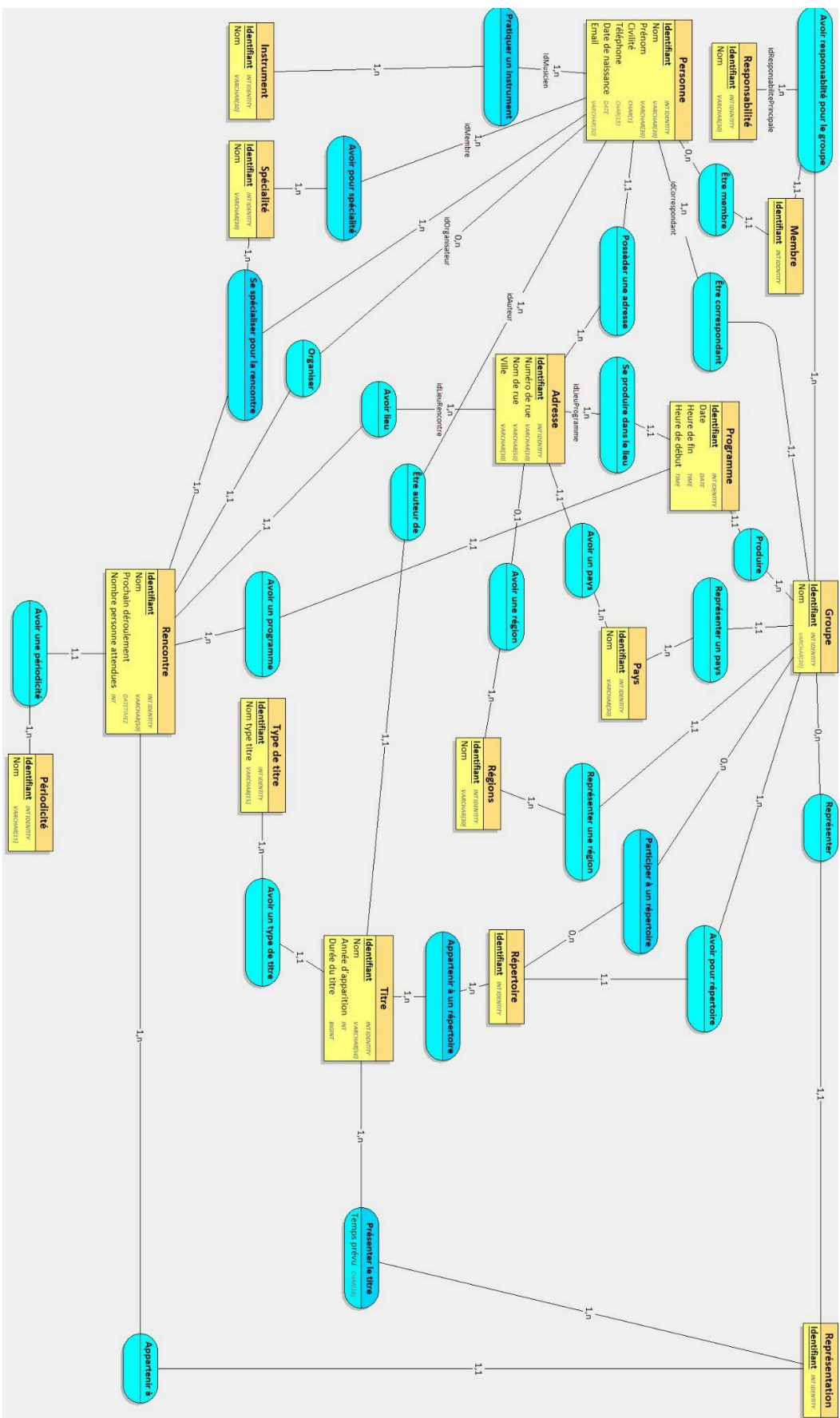
## Dictionnaire des données

Nom donnée	Concerne	Type
nomPays	pays	Chaîne de caractères
nomRegion	regions	Chaîne de caractères
nomSpecialite	specialites	Chaîne de caractères
nomInstrument	instruments	Chaîne de caractères
nomResponsabilite	responsabilites	Chaîne de caractères
nomTypeTitre	typesTitres	Chaîne de caractères
nomPeriodicite	periodicites	Chaîne de caractères
numRue	adresses	Chaîne de caractères
nomRue	adresses	Chaîne de caractères
ville	adresses	Chaîne de caractères
nomPersonne	personnes	Chaîne de caractères
prenomPersonne	personnes	Chaîne de caractères
civilitePersonne	personnes	Caractère
telephonePersonne	personnes	Chaîne de caractères
dateNaissancePersonne	personnes	Date
emailPersonne	personnes	Chaîne de caractères
nomTitre	titres	Chaîne de caractères
anneeTitre	titres	Entier
dureeTitre	titres	Entier
nomRencontre	rencontres	Chaîne de caractères
prochainDeroulement	rencontres	Date & Heure
nbPersonnesAttendues	rencontres	Entier
nomGroupe	groupes	Chaîne de caractères
datePassage	programmes	Date
heureFin	programmes	Heure
heureDebut	programmes	Heure
tempsPrevu	représentations	Chaîne de caractères

## Règles de gestion

N°	Règle
1	Un groupe a un correspondant
2	Un groupe représente un pays
3	Un groupe représente une région
4	Un membre de groupe a une ou plusieurs spécialité(s)
5	Un membre de groupe peut avoir plusieurs spécialité(s) pour une rencontre
6	Membre musicien pratique un ou plusieurs instruments
7	Responsabilité majeur du membre
8	Un membre n'appartient qu'à un seul groupe
9	Un titre a un type
10	Un groupe a un répertoire
11	Un répertoire peut concerner un ou plusieurs groupe(s)
12	Une rencontre a une seule adresse
13	Une rencontre a un organisateur
14	Une rencontre a une périodicité
15	Selon la périodicité, la date et l'heure seront automatiquement changé à la fin d'une rencontre
16	Un programme comprend le passage de plusieurs groupe
17	Une rencontre a plusieurs programmes en fonction des lieux et des dates
18	Un programme peut concerner plusieurs représentations
19	Une adresse comporte un pays et peut comporter une région
20	Une région n'appartient pas forcément qu'à un pays

## MCD



## MLD

pays = (idPays INT IDENTITY, nomPays VARCHAR(30));

regions = (idRegion INT IDENTITY, nomRegion VARCHAR(30));

specialites = (idSpecialite INT IDENTITY, nomSpecialite VARCHAR(30));

instruments = (idInstrument INT IDENTITY, nomInstrument VARCHAR(30));

responsabilites = (idResponsabilite INT IDENTITY, nomResponsabilite VARCHAR(30));

typesTitres = (idTypeTitre INT IDENTITY, nomTypeTitre VARCHAR(15));

periodicites = (idPeriodicite INT IDENTITY, nomPeriodicite VARCHAR(15));

adresses = (idAdresse INT IDENTITY, numRue VARCHAR(10), nomRue VARCHAR(50),  
ville VARCHAR(30), #idRegion\*, #idPays);

personnes = (idPersonne INT IDENTITY, nomPersonne VARCHAR(30), prenomPersonne  
VARCHAR(30), civilitePersonne CHAR(1), telephonePersonne CHAR(15),  
dateNaissancePersonne DATE, emailPersonne VARCHAR(30), #idAdresse);

titres = (idTitre INT IDENTITY, nomTitre VARCHAR(50), anneeTitre INT, dureeTitre BIGINT,  
#idTypeTitre, #idAuteur);

rencontres = (idRencontre INT IDENTITY, nomRencontre VARCHAR(50),  
prochainDeroulement DATETIME2, nbPersonnesAttendues INT, #idPeriodicite,  
#idOrganisateur, #idLieuRencontre);

groupes = (idGroupe INT IDENTITY, nomGroupe VARCHAR(30), #idCorrespondant,  
#idRegion, #idPays);

membres = (idMembre INT IDENTITY, #idGroupe, #idResponsabilitePrincipale,  
#idPersonne);

repertoires = (idRepertoire INT IDENTITY, #idGroupe);

programmes = (idProgramme INT IDENTITY, datePassage DATE, heureFin TIME,  
heureDebut TIME, #idRencontre, #idLieuProgramme, #idGroupe);

representations = (idRepresentation INT IDENTITY, #idRencontre, #idGroupe);

avoirResponsabilite = (#idMembre, #idSpecialite);

pratiquerInstrument = (#idMusicien, #idInstrument);

appartenirRepertoire = (#idTitre, #idRepertoire);

participerRepertoire = (#idGroupe, #idRepertoire);

presenterTitre = (#idTitre, #idRepresentation, tempsPrevu CHAR(10));

specialiserPourRencontre = (#idPersonne, #idSpecialite, #idRencontre);

## Script génération de la base de données

```
CREATE TABLE pays(  
  idPays INT IDENTITY,  
  nomPays VARCHAR(30) NOT NULL,  
  PRIMARY KEY(idPays)  
);
```

```
CREATE TABLE regions(  
  idRegion INT IDENTITY,  
  nomRegion VARCHAR(30) NOT NULL,  
  idPays INT NOT NULL,  
  PRIMARY KEY(idRegion),  
  FOREIGN KEY(idPays) REFERENCES pays(idPays)  
);
```

```
CREATE TABLE specialites(  
  idSpecialite INT IDENTITY,  
  nomSpecialite VARCHAR(30) NOT NULL,  
  PRIMARY KEY(idSpecialite)  
);
```

```
CREATE TABLE instruments(  
  idInstrument INT IDENTITY,  
  nomInstrument VARCHAR(30) NOT NULL,  
  PRIMARY KEY(idInstrument)  
);
```

```
CREATE TABLE responsabilites(  
  idResponsabilite INT IDENTITY,  
  nomResponsabilite VARCHAR(30) NOT NULL,  
  PRIMARY KEY(idResponsabilite)  
);
```

```
CREATE TABLE typesTitres(  
  idTypeTitre INT IDENTITY,  
  nomTypeTitre VARCHAR(15) NOT NULL,  
  PRIMARY KEY(idTypeTitre)  
);
```

```
CREATE TABLE periodicites(  
  idPeriodicite INT IDENTITY,  
  nomPeriodicite VARCHAR(15) NOT NULL,  
  PRIMARY KEY(idPeriodicite)  
);
```

```

CREATE TABLE adresses(
  idAdresse INT IDENTITY,
  numRue VARCHAR(10) NOT NULL,
  nomRue VARCHAR(50) NOT NULL,
  ville VARCHAR(30) NOT NULL,
  idRegion INT,
  idPays INT NOT NULL,
  PRIMARY KEY(idAdresse),
  FOREIGN KEY(idRegion) REFERENCES regions(idRegion),
  FOREIGN KEY(idPays) REFERENCES pays(idPays)
);

```

```

CREATE TABLE personnes(
  idPersonne INT IDENTITY,
  nomPersonne VARCHAR(30) NOT NULL,
  prenomPersonne VARCHAR(30) NOT NULL,
  civilitePersonne CHAR(1) NOT NULL,
  telephonePersonne CHAR(15),
  dateNaissancePersonne DATE,
  emailPersonne VARCHAR(30),
  idAdresse INT NOT NULL,
  PRIMARY KEY(idPersonne),
  FOREIGN KEY(idAdresse) REFERENCES adresses(idAdresse)
);

```

```

CREATE TABLE titres(
  idTitre INT IDENTITY,
  nomTitre VARCHAR(50) NOT NULL,
  anneeTitre INT NOT NULL,
  dureeTitre BIGINT NOT NULL,
  idTypeTitre INT NOT NULL,
  idAuteur INT NOT NULL,
  PRIMARY KEY(idTitre),
  FOREIGN KEY(idTypeTitre) REFERENCES typesTitres(idTypeTitre),
  FOREIGN KEY(idAuteur) REFERENCES personnes(idPersonne)
);

```

```

CREATE TABLE rencontres(
  idRencontre INT IDENTITY,
  nomRencontre VARCHAR(50) NOT NULL,
  prochainDeroulement DATETIME2,
  nbPersonnesAttendues INT NOT NULL,
  idPeriodicite INT NOT NULL,
  idOrganisateur INT NOT NULL,
  idLieuRencontre INT NOT NULL,
  PRIMARY KEY(idRencontre),
  FOREIGN KEY(idPeriodicite) REFERENCES periodicites(idPeriodicite),
  FOREIGN KEY(idOrganisateur) REFERENCES personnes(idPersonne),
  FOREIGN KEY(idLieuRencontre) REFERENCES adresses(idAdresse)
);

```

```

CREATE TABLE groupes(
  idGroupe INT IDENTITY,
  nomGroupe VARCHAR(30),
  idCorrespondant INT NOT NULL,

```

```

idRegion INT NOT NULL,
idPays INT NOT NULL,
PRIMARY KEY(idGroupe),
FOREIGN KEY(idCorrespondant) REFERENCES personnes(idPersonne),
FOREIGN KEY(idRegion) REFERENCES regions(idRegion),
FOREIGN KEY(idPays) REFERENCES pays(idPays)
);

```

```

CREATE TABLE membres(
idMembre INT IDENTITY,
idGroupe INT NOT NULL,
idResponsabilitePrincipale INT NOT NULL,
idPersonne INT NOT NULL,
PRIMARY KEY(idMembre),
FOREIGN KEY(idGroupe) REFERENCES groupes(idGroupe),
FOREIGN KEY(idResponsabilitePrincipale) REFERENCES
responsabilites(idResponsabilite),
FOREIGN KEY(idPersonne) REFERENCES personnes(idPersonne)
);

```

```

CREATE TABLE repertoires(
idRepertoire INT IDENTITY,
idGroupe INT NOT NULL,
PRIMARY KEY(idRepertoire),
FOREIGN KEY(idGroupe) REFERENCES groupes(idGroupe)
);

```

```

CREATE TABLE programmes(
idProgramme INT IDENTITY,
datePassage DATE NOT NULL,
heureFin TIME NOT NULL,
heureDebut TIME NOT NULL,
idRencontre INT NOT NULL,
idLieuProgramme INT NOT NULL,
idGroupe INT NOT NULL,
PRIMARY KEY(idProgramme),
FOREIGN KEY(idRencontre) REFERENCES rencontres(idRencontre),
FOREIGN KEY(idLieuProgramme) REFERENCES adresses(idAdresse),
FOREIGN KEY(idGroupe) REFERENCES groupes(idGroupe)
);

```

```

CREATE TABLE representations(
idRepresentation INT IDENTITY,
idRencontre INT NOT NULL,
idGroupe INT NOT NULL,
PRIMARY KEY(idRepresentation),
FOREIGN KEY(idRencontre) REFERENCES rencontres(idRencontre),
FOREIGN KEY(idGroupe) REFERENCES groupes(idGroupe)
);

```

```

CREATE TABLE avoirResponsabilite(
idMembre INT,
idSpecialite INT,
PRIMARY KEY(idMembre, idSpecialite),
FOREIGN KEY(idMembre) REFERENCES personnes(idPersonne),

```

```
FOREIGN KEY(idSpecialite) REFERENCES specialites(idSpecialite)
);
```

```
CREATE TABLE pratiquerInstrument(
  IdMusicien INT,
  idInstrument INT,
  PRIMARY KEY(IdMusicien, idInstrument),
  FOREIGN KEY(IdMusicien) REFERENCES personnes(idPersonne),
  FOREIGN KEY(idInstrument) REFERENCES instruments(idInstrument)
);
```

```
CREATE TABLE appartenirRepertoire(
  idTitre INT,
  idRepertoire INT,
  PRIMARY KEY(idTitre, idRepertoire),
  FOREIGN KEY(idTitre) REFERENCES titres(idTitre),
  FOREIGN KEY(idRepertoire) REFERENCES repertoires(idRepertoire)
);
```

```
CREATE TABLE participerRepertoire(
  idGroupe INT,
  idRepertoire INT,
  PRIMARY KEY(idGroupe, idRepertoire),
  FOREIGN KEY(idGroupe) REFERENCES groupes(idGroupe),
  FOREIGN KEY(idRepertoire) REFERENCES repertoires(idRepertoire)
);
```

```
CREATE TABLE presenterTitre(
  idTitre INT,
  idRepresentation INT,
  tempsPrevu CHAR(10) NOT NULL,
  PRIMARY KEY(idTitre, idRepresentation),
  FOREIGN KEY(idTitre) REFERENCES titres(idTitre),
  FOREIGN KEY(idRepresentation) REFERENCES representations(idRepresentation)
);
```

```
CREATE TABLE specialiserPourRencontre(
  idPersonne INT,
  idSpecialite INT,
  idRencontre INT,
  PRIMARY KEY(idPersonne, idSpecialite, idRencontre),
  FOREIGN KEY(idPersonne) REFERENCES personnes(idPersonne),
  FOREIGN KEY(idSpecialite) REFERENCES specialites(idSpecialite),
  FOREIGN KEY(idRencontre) REFERENCES rencontres(idRencontre)
);
```