

Clémence Bocquet  
Ophélie Valin  
Nathan Randrianntsoa  
BUT2 Informatique Orléans

# Rendu SAE Poney

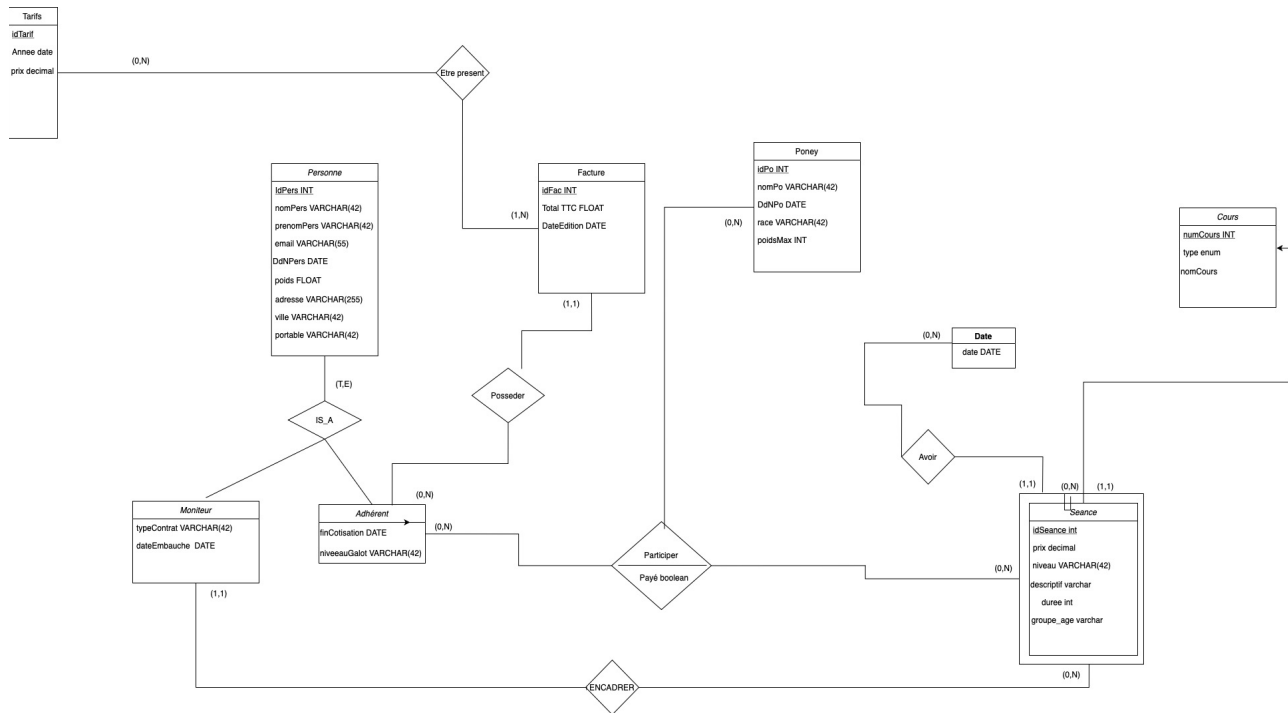
## Table des matières

Fiches individuelles.....	1
MCD.....	2
Dépendances fonctionnelles.....	3
Liste des contraintes.....	3
Scripts.....	4
Script de création.....	4
Script de suppression.....	7
Script d'insertion.....	7
Maquettes.....	7

## Fiches individuelles

# MCD

## MCD SAE Poney Clémence Bocquet, Ophélie Valin, Nathan Randrianstoa



## Dépendances fonctionnelles

L'ensemble des dépendances fonctionnelles est :

- Table PONEY : IDPO → NOMPO, DDNPO, POIDS\_MAX, RACE, SEXE
- Table COURS : NUMCOURS → NOMCOURS, TYPEC
- Table Séance : (IDSEANCE, NUMCOURS) → PRIX, DUREE, NIVEAU, DESCRIPTIF, GROUPE\_AGE, DATE\_SEANCE
- Table PERSONNE : IDPER → NOMPER, PRENOMPER, EMAIL, DDNPER, POIDS, ADRESSE, PORTABLE
- Table MONITEUR : IDMON → TYPECONTRAT, DATEEMBAUCHE
- Table ADHERENT : IDADH → FINCOTISATION, NIVEAUGALOT
- Table FACTURE : IDFACTURE → TOTALTTC, DATEEDITION, IDADH
- Table TARIFS : IDTARIF → ANNEE, PRIX
- Table PARTICIPER : (NUMCOURS, IDADH, DATEPART, IDSEANCE, IDPO) → PAYE

## Liste des contraintes

Les contraintes à implémenter sont :

- Un cours collectif a maximum 10 participants
- Un cours particulier a un seul participant
- Un cours dure entre 1 ou 2 heures
- Un poney doit avoir une heure de repos après deux heures de cours
- Un poney peut porter un cavalier jusqu'à un certain poids
- Un cours est soit collectif soit individuel

Lors de réalisation du script de création de la base de données, nous avons pu implémenter « un cours dure entre 1 ou 2 heures » notamment avec l'utilisation de CHECK `CHECK (DUREE BETWEEN 1 AND 2)`.

De plus, un cours est soit collectif soit individuel. Nous avons donc utilisé un ENUM('collectif', 'individuel') afin de n'autoriser que ces valeurs.

# Scripts

## Script de création

```
-- create new schema  
create database if not exists SAE_PONEY;
```

```
-- Script de création des tables de la base de données SAE_Poney
```

```
CREATE TABLE PONEY(  
IDPO INT PRIMARY KEY AUTO_INCREMENT,  
NOMPO VARCHAR(15),  
DDNPO DATE,  
POIDS_MAX INT(3) NOT NULL, -- un poney supporte un poids max  
RACE VARCHAR(20),  
SEXE enum('F', 'M')  
);
```

```
CREATE TABLE COURS (  
NUMCOURS INT PRIMARY KEY,  
NOMCOURS VARCHAR(42),  
TYPEC enum('Collectif', 'Individuel')  
);
```

```
CREATE TABLE SEANCE (  
IDSEANCE INT,  
NUMCOURS INT,  
PRIX DECIMAL(5,2),  
DUREE DECIMAL(3,2) CHECK (DUREE BETWEEN 1 AND 2), -- le cours dure entre une et deux heures  
NIVEAU INT,  
DESCRIPTIF VARCHAR(50),  
GROUPE_AGE VARCHAR(50),  
DATE_SEANCE DATETIME,  
PRIMARY KEY (IDSEANCE, NUMCOURS)  
);
```

```
CREATE TABLE PERSONNE(  
IDPER INT PRIMARY KEY,  
NOMPER VARCHAR(42),  
PRENOMPER VARCHAR(42),  
EMAIL VARCHAR(50),  
DDNPER DATE,  
POIDS DECIMAL(5,2),  
ADRESSE VARCHAR(100),  
PORTABLE VARCHAR(14)
```

```
);
```

```
CREATE TABLE MONITEUR(  
IDMON INT PRIMARY KEY,  
TYPECONTRAT VARCHAR(42),  
DATEEMBAUCHE DATE  
);
```

```
CREATE TABLE ADHERENT (  
IDADH INT PRIMARY KEY,  
FINCOTISATION DATE,  
NIVEAUGALOT INT  
);
```

```
CREATE TABLE FACTURE ( -- correspond au cotisation de l'adherent chaque annee  
IDFACTURE INT PRIMARY KEY auto_increment,  
TOTALTTC DECIMAL(10,2),  
DATEEDITION DATE,  
IDADH INT  
);
```

```
CREATE TABLE TARIFS (  
IDTARIF INT PRIMARY KEY,  
ANNEE DATE,  
PRIX DECIMAL(7,2)  
);
```

```
-- associations
```

```
-- un moniteur encadre une seance a une certaine date  
CREATE TABLE ENCADRER(  
IDMON INT,  
NUMCOURS INT,  
IDSEANCE INT,  
DATEENC DATETIME,  
PRIMARY KEY (IDMON, NUMCOURS, DATEENC, IDSEANCE)  
);
```

```
-- un adherent et un poney participe a une seance a une certaine date  
CREATE TABLE PARTICIPER(  
NUMCOURS INT,  
IDPO int,  
IDADH INT,  
DATEPART DATETIME,  
PAYE BOOLEAN NOT NULL, -- une fois paye, la seance ne peut pas etre remboursee  
IDSEANCE INT,
```

```
PRIMARY KEY (NUMCOURS, IDADH, DATEPART, IDSEANCE, IDPO)  
);
```

```
-- pour chaque facture, un/des tarif.s sont associes
```

```
CREATE TABLE ETRE_PRESENT(  
IDFACTURE INT,  
IDTARIF INT,  
PRIMARY KEY (IDFACTURE, IDTARIF)  
);
```

```
-- cles etrangeres
```

```
ALTER TABLE MONITEUR ADD FOREIGN KEY (IDMON) REFERENCES PERSONNE(IDPER);  
ALTER TABLE ADHERENT ADD FOREIGN KEY (IDADH) REFERENCES PERSONNE(IDPER);
```

```
ALTER TABLE FACTURE ADD FOREIGN KEY (IDADH) REFERENCES ADHERENT(IDADH);  
ALTER TABLE SEANCE ADD FOREIGN KEY (NUMCOURS) REFERENCES COURS(NUMCOURS);
```

```
ALTER TABLE ENCADRER ADD FOREIGN KEY (IDMON) REFERENCES MONITEUR(IDMON);  
ALTER TABLE ENCADRER ADD FOREIGN KEY (IDSEANCE, NUMCOURS) REFERENCES  
SEANCE(IDSEANCE, NUMCOURS);
```

```
ALTER TABLE PARTICIPER ADD FOREIGN KEY (IDADH) REFERENCES ADHERENT(IDADH);  
ALTER TABLE PARTICIPER ADD FOREIGN KEY (IDPO) REFERENCES PONEY(IDPO);  
ALTER TABLE PARTICIPER ADD FOREIGN KEY (IDSEANCE, NUMCOURS) REFERENCES  
SEANCE(IDSEANCE, NUMCOURS);
```

```
ALTER TABLE ETRE_PRESENT ADD FOREIGN KEY (IDFACTURE) REFERENCES FACTURE(IDFACTURE);  
ALTER TABLE ETRE_PRESENT ADD FOREIGN KEY (IDTARIF) REFERENCES TARIFS(IDTARIF);
```

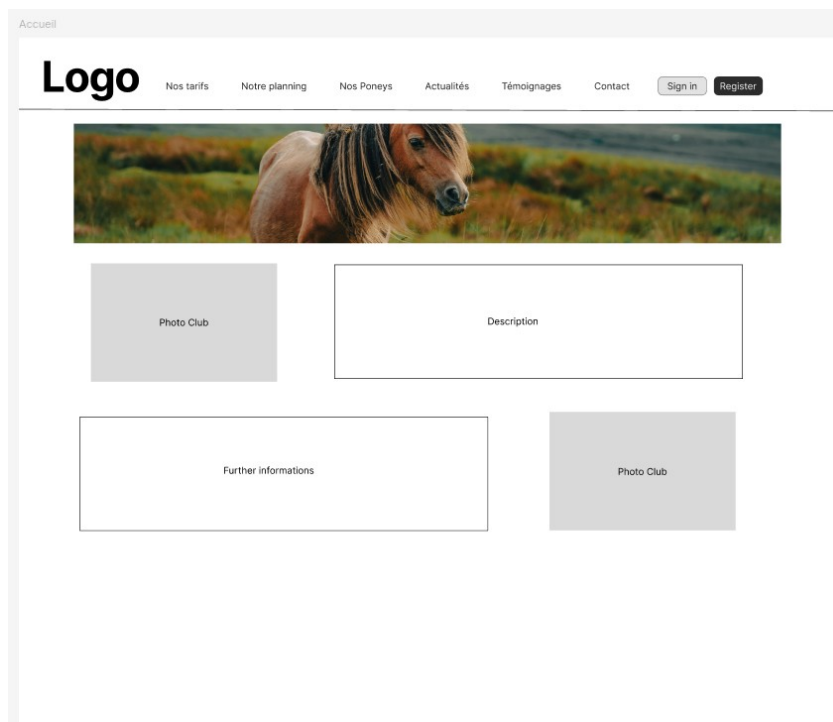
## Script de suppression

```
-- script de suppression des tables de la base de données SAE_PONEY
DROP TABLE IF EXISTS ENCADRER;
DROP TABLE IF EXISTS ETRE_PRESENT;
DROP TABLE IF EXISTS TARIFS;
DROP TABLE IF EXISTS FACTURE;
DROP TABLE IF EXISTS SOLLICITER;
DROP TABLE IF EXISTS PARTICIPER;
DROP TABLE IF EXISTS PONEY;
DROP TABLE IF EXISTS SEANCE;
DROP TABLE IF EXISTS COURS;
DROP TABLE IF EXISTS ADHERENT;
DROP TABLE IF EXISTS MONITEUR;
DROP TABLE IF EXISTS PERSONNE;
DROP TABLE IF EXISTS SEANCE;
```

## Script d'insertion

# Maquettes

## Accueil



## Connexion - Inscription