

Projet NF18: SQL du 27/04/2021

```
CREATE TABLE Association (  
    nom VARCHAR(50) PRIMARY KEY,  
    description TEXT NOT NULL,  
    categorie VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    dateCreation DATE NOT NULL,  
    siteWeb TEXT  
);
```

```
CREATE TABLE Spectacle (  
    idSpectacle INTEGER AUTO_INCREMENT,  
    duree TIME NOT NULL,  
    nomAsso VARCHAR(100),  
    PRIMARY KEY(idSpectacle),  
    FOREIGN KEY (nomAsso) REFERENCES Association(nom)  
);
```

```
CREATE TABLE Personne (  
    idPersonne INTEGER AUTO_INCREMENT,  
    nom VARCHAR(50) NOT NULL,  
    prenom VARCHAR(50) NOT NULL,  
    PRIMARY KEY (idPersonne)  
);
```

```
CREATE TABLE Etudiant (  
    idPersonne INTEGER,  
    numeroCIN INTEGER UNIQUE NOT NULL,  
    PRIMARY KEY(idPersonne),  
    FOREIGN KEY (idPersonne) REFERENCES Personne(idPersonne)  
);
```

```
CREATE TABLE Personnel (  
    idPersonne INTEGER,  
    numeroCIN INTEGER UNIQUE NOT NULL,  
    statut VARCHAR(150) NOT NULL CHECK (statut IN  
    ('enseignant','personnelAdministratif','personnelTechnique')),  
    PRIMARY KEY (idPersonne),  
    FOREIGN KEY (idPersonne) REFERENCES Personne(idPersonne)  
);
```

```
CREATE TABLE Extérieur (  
    idPersonne INTEGER,  
    organisme VARCHAR(50) NOT NULL,  
    contact VARCHAR(10) NOT NULL,  
    PRIMARY KEY (idPersonne),
```

```
        FOREIGN KEY (idPersonne) REFERENCES Personne(idPersonne)
    );

CREATE TABLE ParticipePersonnel(
    idPersonnel INTEGER ,
    idSpectacle INTEGER,
    role VARCHAR(50) NOT NULL,
    FOREIGN KEY (idPersonnel) REFERENCES Personnel(idPersonne),
    FOREIGN KEY (idSpectacle) REFERENCES Spectacle(idSpectacle),
    PRIMARY KEY (idPersonnel, idSpectacle)
);

CREATE TABLE ParticipeEtudiant(
    idEtudiant INTEGER,
    idSpectacle INTEGER,
    role VARCHAR(50) NOT NULL,
    FOREIGN KEY (idEtudiant) REFERENCES Etudiant(idPersonne),
    FOREIGN KEY (idSpectacle) REFERENCES Spectacle(idSpectacle),
    PRIMARY KEY (idEtudiant, idSpectacle)
);

CREATE TABLE GenreConcert(
    genreConcert VARCHAR(50),
    PRIMARY KEY(genreConcert)
);

CREATE TABLE GenreStandUp(
    genreStandUp VARCHAR(50),
    PRIMARY KEY(genreStandUp)
);

CREATE TABLE GenreTheatre(
    genreTheatre VARCHAR(50),
    PRIMARY KEY(genreTheatre)
);

CREATE TABLE Concert(
    idConcert INTEGER,
    compositeur VARCHAR(100) NOT NULL,
    genre VARCHAR(50),
    anneeParution DATE,
    PRIMARY KEY(idConcert),
    FOREIGN KEY (idConcert) REFERENCES Spectacle(idSpectacle),
    FOREIGN KEY (genre) REFERENCES GenreConcert(genreConcert)
);

CREATE TABLE StandUP(
    idStandUp INTEGER,
```

```
    genre VARCHAR(50),
    PRIMARY KEY(idStandUp),
    FOREIGN KEY (idStandup) REFERENCES Spectacle(idSpectacle),
    FOREIGN KEY (genre) REFERENCES GenreStandUp(genreStandUp)
);
```

```
CREATE TABLE Theatre(
    idTheatre INTEGER,
    auteur VARCHAR(100) NOT NULL,
    anneeParution DATE NOT NULL,
    genre VARCHAR(50),
    PRIMARY KEY(idTheatre),
    FOREIGN KEY (idTheatre) REFERENCES Spectacle(idSpectacle),
    FOREIGN KEY (genre) REFERENCES GenreTheatre(genreTheatre)
);
```

```
CREATE TABLE Salle (
    numSalle INTEGER,
    capacite INTEGER NOT NULL,
    PRIMARY KEY (numSalle)
);
```

```
CREATE TABLE Batiment (
    numBat INTEGER,
    PRIMARY KEY (numBat)
);
```

```
CREATE TABLE TypeSalle (
    numBat INTEGER,
    numSalle INTEGER,
    libelleType VARCHAR(50) NOT NULL,
    PRIMARY KEY (numBat, numSalle)
);
```

```
CREATE TABLE Seance (
    idSeance INTEGER PRIMARY KEY AUTO_INCREMENT,
    date DATE NOT NULL,
    horaireDebut TIME NOT NULL,
    idSpectacle INTEGER,
    numBat INTEGER,
    numSalle INTEGER,
    FOREIGN KEY (idSpectacle) REFERENCES Spectacle(idSpectacle),
    FOREIGN KEY (numBat) REFERENCES Salle(numSalle),
    FOREIGN KEY (numSalle) REFERENCES Batiment(numBat)
);
```

```
CREATE TABLE Billet(
```

```
idBillet INTEGER AUTO_INCREMENT,  
categorie VARCHAR(50),  
dateCreation DATE NOT NULL,  
tarif DECIMAL CHECK (tarif BETWEEN 0 AND 9999),  
idSeance INTEGER,  
idAcheteur INTEGER,  
PRIMARY KEY(idBillet),  
FOREIGN KEY (idSeance) REFERENCES Seance(idSeance),  
FOREIGN KEY (idAcheteur) REFERENCES Personne(idPersonne)  
);  
  
CREATE TABLE Reservation(  
    nom VARCHAR(50),  
    numSalle INTEGER,  
    numBat INTEGER,  
    date DATE NOT NULL,  
    heure TIME NOT NULL,  
    FOREIGN KEY (nom) REFERENCES Association(nom),  
    FOREIGN KEY (numSalle) REFERENCES Salle(numSalle),  
    FOREIGN KEY (numBat) REFERENCES Batiment(numBat),  
    PRIMARY KEY(nom, numSalle, numBat)  
);  
  
CREATE TABLE Compose (  
    idEtudiant INTEGER,  
    nomAsso VARCHAR(100),  
    statut VARCHAR(50) NOT NULL CHECK (statut IN ('president','tresorier','membre')),  
    PRIMARY KEY (idEtudiant, nomAsso),  
    FOREIGN KEY (idEtudiant) REFERENCES Etudiant(idPersonne),  
    FOREIGN KEY (nomAsso) REFERENCES Association(nom)  
);  
  
CREATE TABLE Categorie (  
    idSeance INTEGER,  
    idBillet INTEGER,  
    FOREIGN KEY (idSeance) REFERENCES Seance(idSeance),  
    FOREIGN KEY (idBillet) REFERENCES Billet(idBillet),  
    PRIMARY KEY(idSeance,idBillet)  
);
```

Rapport de la séance du 27/04/21:

Nous avons transcrit le mld en requête sql permettant de créer les tables dans notre base de données. Nous avons également identifié les contraintes sur les attributs de nos tables. Suite au retour sur notre MCD, nous avons adapté notre ancien MCD afin de répondre aux attentes: nous avons ainsi ajouter une classe d'association catégorie afin de standardiser le tarif des billets en fonction de leur catégorie et séance.