



Synapse Audio : Plus qu'un son, une onde d'excellence.

Compte rendu compétence 4

Projet : SoundStream

Équipe : Synapse Audio

Rédigé par :

- Kadir CASSEL
- Aboubakry SY
- Saif-eddine ALJANE
- Romain HEURTAUX
- Tristan COLLEN

Client référent : M. Patrice Bringuier

Établissement : IUT de Villetaneuse - Université Sorbonne Paris Nord – BUT2 Informatique

Date : Novembre 2025

Image du MCD

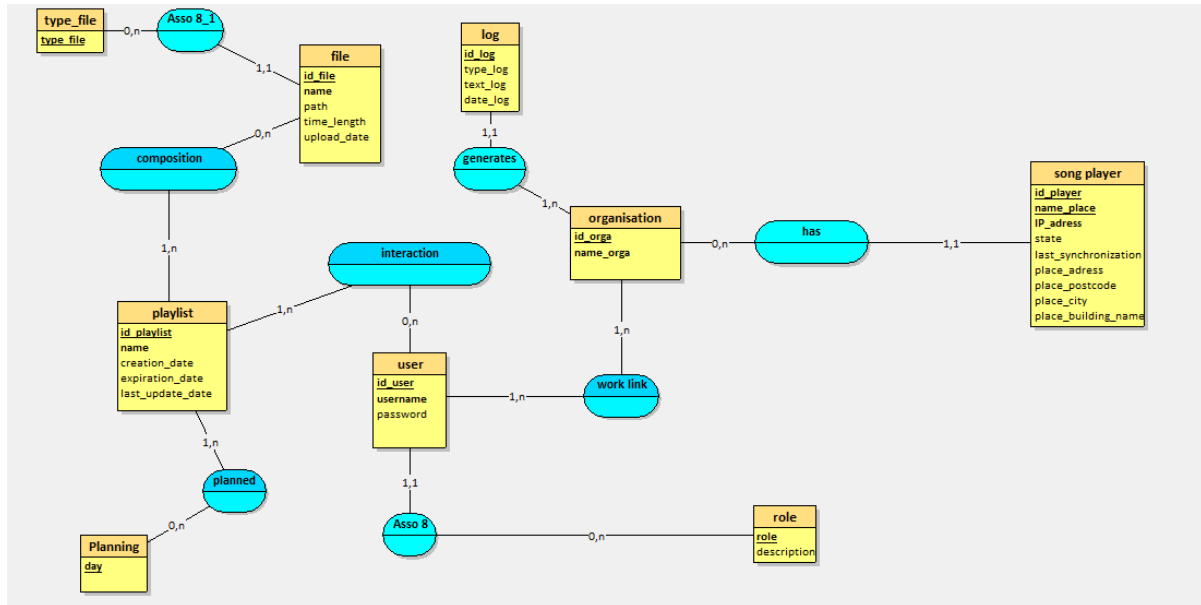


Schéma Relationnel

- **playlist**(id_playlist, name, creation_date, expiration_date, last_update_date)
- **organisation**(id_orga, name_orga)
- **song_player**(id_player, name_place, IP_address, state, last_synchronization, place_address, place_postcode, place_city, place_building_name, id_orga)
où la clé **id_orga** est une clé étrangère qui référence la table **organisation**
- **planning**(day)
- **role**(role, description)
- **type_file**(type_file)
- **user**(id_user, username, password, role)
où la clé **role** est une clé étrangère qui référence la table **role**
- **log**(id_log, type_log, text_log, date_log, id_orga)
où la clé **id_orga** est une clé étrangère qui référence la table **organisation**
- **file**(id_file, name, path, time_length, upload_date, type_file)
où la clé **type_file** est une clé étrangère qui référence la table **type_file**
- **work_link**(id_user, id_orga)
où la clé **id_user** est une clé étrangère qui référence la table **user**
et la clé **id_orga** est une clé étrangère qui référence la table **organisation**
- **composition**(id_playlist, id_file)
où la clé **id_playlist** est une clé étrangère qui référence la table **playlist**
et la clé **id_file** est une clé étrangère qui référence la table **file**
- **interaction**(id_playlist, id_user)
où la clé **id_playlist** est une clé étrangère qui référence la table **playlist**
et la clé **id_user** est une clé étrangère qui référence la table **user**
- **planned**(id_playlist, day, start_time)
où la clé **id_playlist** est une clé étrangère qui référence la table **playlist**
et la clé **day** est une clé étrangère qui référence la table **planning**

Script SQL

Script conforme à la syntaxe SQLite.

```
DROP TABLE IF EXISTS planned;
DROP TABLE IF EXISTS interaction;
DROP TABLE IF EXISTS composition;
DROP TABLE IF EXISTS work_link;
DROP TABLE IF EXISTS log;
DROP TABLE IF EXISTS file;
DROP TABLE IF EXISTS user_;
DROP TABLE IF EXISTS song_player;
DROP TABLE IF EXISTS playlist;
DROP TABLE IF EXISTS organisation;
DROP TABLE IF EXISTS Planning;
DROP TABLE IF EXISTS role;
DROP TABLE IF EXISTS type_file;

CREATE TABLE IF NOT EXISTS playlist(
    id_playlist INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    creation_date DATETIME NOT NULL,
    expiration_date DATETIME NOT NULL,
    last_update_date DATETIME NOT NULL,
    UNIQUE(name)
);

CREATE TABLE IF NOT EXISTS organisation(
    id_orga INTEGER PRIMARY KEY,
    name_orga TEXT NOT NULL,
    UNIQUE(name_orga)
);

CREATE TABLE IF NOT EXISTS song_player(
    id_player INTEGER PRIMARY KEY,
    name_place TEXT NOT NULL,
    IP_adress TEXT NOT NULL,
    state VARCHAR(50) NOT NULL,
    last_synchronization DATETIME,
    place_adress TEXT NOT NULL,
    place_postcode VARCHAR(5) NOT NULL,
    place_city VARCHAR(50) NOT NULL,
    place_building_name TEXT,
    id_orga INT NOT NULL,
    UNIQUE(IP_adress),
    FOREIGN KEY(id_orga) REFERENCES organisation(id_orga)
);

CREATE TABLE IF NOT EXISTS Planning(
    day_ VARCHAR(50),
    PRIMARY KEY(day_)
);

CREATE TABLE IF NOT EXISTS role(
    role TEXT PRIMARY KEY,
    description VARCHAR(50) NOT NULL
);

CREATE TABLE IF NOT EXISTS type_file(
    type_file VARCHAR(50),
    PRIMARY KEY(type_file)
);

CREATE TABLE IF NOT EXISTS user_(
    id_user INTEGER PRIMARY KEY,
```

```
username VARCHAR(25) NOT NULL,
password TEXT NOT NULL,
role TEXT NOT NULL,
UNIQUE(username),
FOREIGN KEY(role) REFERENCES role(role)
);

CREATE TABLE IF NOT EXISTS log(
  id_log INTEGER PRIMARY KEY,
  type_log TEXT NOT NULL,
  text_log TEXT NOT NULL,
  date_log DATETIME NOT NULL,
  id_orga INT NOT NULL,
  FOREIGN KEY(id_orga) REFERENCES organisation(id_orga)
);

CREATE TABLE IF NOT EXISTS file(
  id_file INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  path TEXT NOT NULL,
  time_length TIME NOT NULL,
  upload_date DATETIME NOT NULL,
  type_file VARCHAR(50) NOT NULL,
  UNIQUE(name),
  FOREIGN KEY(type_file) REFERENCES type_file(type_file)
);

CREATE TABLE IF NOT EXISTS work_link(
  id_user INT,
  id_orga INT,
  PRIMARY KEY(id_user, id_orga),
  FOREIGN KEY(id_user) REFERENCES user_(id_user),
  FOREIGN KEY(id_orga) REFERENCES organisation(id_orga)
);

CREATE TABLE IF NOT EXISTS composition(
  id_playlist INT,
  id_file INT,
  PRIMARY KEY(id_playlist, id_file),
  FOREIGN KEY(id_playlist) REFERENCES playlist(id_playlist),
  FOREIGN KEY(id_file) REFERENCES file(id_file)
);

CREATE TABLE IF NOT EXISTS interaction(
  id_playlist INT,
  id_user INT,
  PRIMARY KEY(id_playlist, id_user),
  FOREIGN KEY(id_playlist) REFERENCES playlist(id_playlist),
  FOREIGN KEY(id_user) REFERENCES user_(id_user)
);

CREATE TABLE IF NOT EXISTS planned(
  id_playlist INT,
  day_ VARCHAR(50),
  start_time TIME,
  PRIMARY KEY(id_playlist, day_),
  FOREIGN KEY(id_playlist) REFERENCES playlist(id_playlist),
  FOREIGN KEY(day_) REFERENCES Planning(day_)
);
```