



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

## Compte rendu compétence 4

Projet : SoundStream

Équipe : Synapse Audio

Rédigé par :

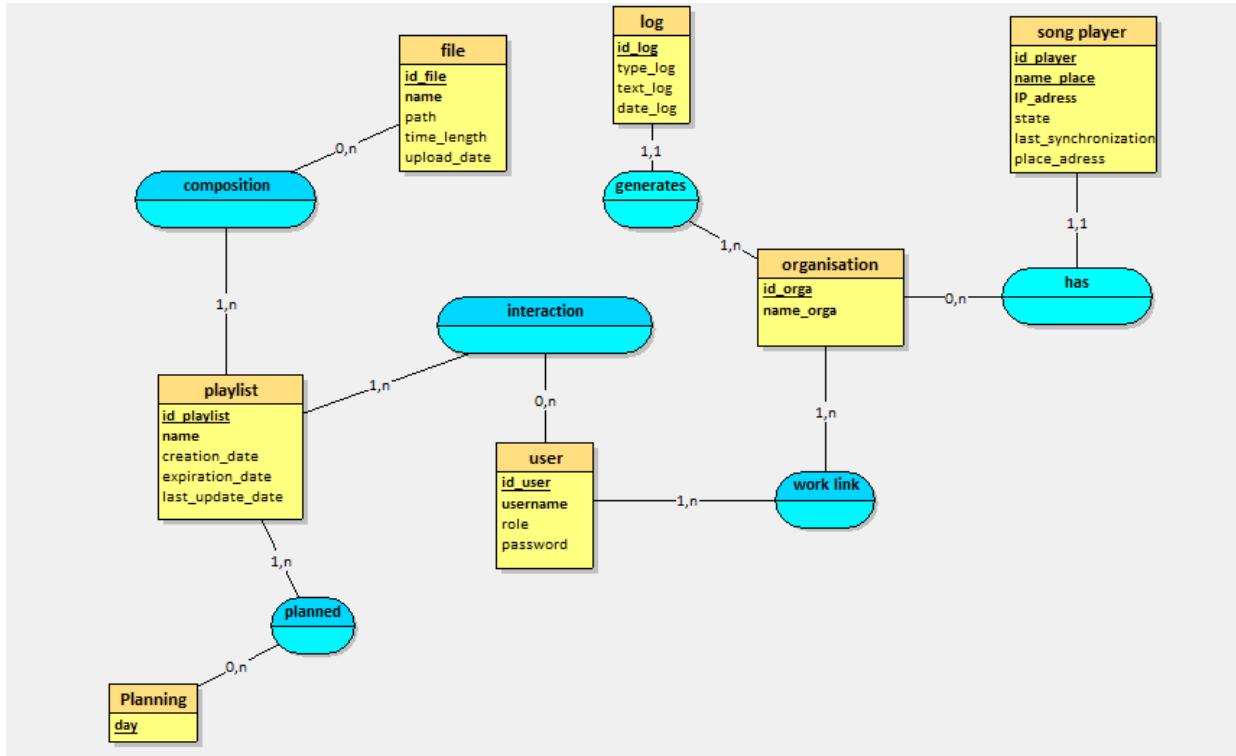
- Kadir CASSEL
- Aboubakry SY
- Saif-eddine ALJANE
- Romain HEURTAUX
- Tristan COLLEN

Client référent : M. Patrice Bringier

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

Date : Novembre 2025

## 1 Image du MCD :



### 1.1 Schema Relationnel :

- **playlist**(id\_playlist, name, creation\_date, expiration\_date, last\_update\_date)
- **user** (id\_user, username, role, password)
- **organisation**(id\_orga, name\_orga, subsidiary)
- **file**(id\_file, name, path, time\_length, upload\_date)
- **song\_player**(id\_player, name\_place, IP\_adress, state, last\_synchronization, place\_adress, id\_orga)
  - *id\_orga est une clé étrangère référençant organisation(id\_orga)*
- **planning**(day\_)
- **log**(id\_log, type\_log, text\_log, date\_log, id\_orga)
  - *id\_orga est une clé étrangère référençant organisation(id\_orga)*
- **work\_link**(id\_user, id\_orga)
  - *id\_user est une clé étrangère référençant user\_(id\_user)*
  - *id\_orga est une clé étrangère référençant organisation(id\_orga)*
- **composition**(id\_playlist, id\_file)
  - *id\_playlist est une clé étrangère référençant playlist(id\_playlist)*
  - *id\_file est une clé étrangère référençant file(id\_file)*
- **interaction**(id\_playlist, id\_user)
  - *id\_playlist est une clé étrangère référençant playlist(id\_playlist)*
  - *id\_user est une clé étrangère référençant user\_(id\_user)*
- **planned**(id\_playlist, day\_)
  - *id\_playlist est une clé étrangère référençant playlist(id\_playlist)*
  - *day\_ est une clé étrangère référençant planning(day\_)*

### 1.2 Script SQL :

C'est un SQL qui répond aux règles de syntaxe de SQLite

```
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 user_(
    id_user INTEGER PRIMARY KEY,
    username VARCHAR(25) NOT NULL,
    role VARCHAR(50) NOT NULL,
    password TEXT NOT NULL,
    UNIQUE(username)
);

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

CREATE TABLE IF NOT EXISTS song_player(
    id_player INTEGER PRIMARY KEY,
    name_place TEXT UNIQUE NOT NULL,
    IP_adress TEXT NOT NULL,
    state VARCHAR(50) NOT NULL,
    last_synchronization DATETIME,
    place_adress TEXT NOT NULL,
    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 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 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),  
    PRIMARY KEY(id_playlist, day_),  
    FOREIGN KEY(id_playlist) REFERENCES playlist(id_playlist),  
    FOREIGN KEY(day_) REFERENCES Planning(day_)  
);
```