DBLAB-P081
OTT Service - Programme Catalogue

Name: Gautam Prasanna Kappagal
Roll No: IMT2023082

Database Use Cases

1. Name: Search programmes by attribute
   Actor: Service consumer – OTT Viewer
   Action: Explores programmes filtered by genre, release date, or duration.

2. Name: View programme cast
   Actor: Service consumer – OTT Viewer
   Action: Views the list of cast members for a selected programme.

3. Name: Track programme episodes
   Actor: Service consumer – OTT Viewer
   Action: Lists all released episodes for a particular programme, season-wise.

4. Name: Manage programmes
   Actor: Service provider – OTT admins
   Action: Adds new programmes with details or deletes them using programme ID.

5. Name: Add new genres
   Actor: Service provider – OTT admins
   Action: Inserts new genres into the genre list for future classification.

6. Name: Add episodes to a programme
   Actor: Service provider – OTT admins
   Action: Adds newly released episodes (say weekly) to an existing programme.

DDL Scripts

- ott_db_create.sql:

```
-- Delete the database if it exists and start over
DROP DATABASE IF EXISTS ott_db;
CREATE DATABASE ott_db;
USE ott_db;

-- Create the 'Genres' table to store different genres of the programmes
CREATE TABLE Genres (
   genre_id INT AUTO_INCREMENT PRIMARY KEY,    -- Unique ID for each genre
   genre_name ENUM('Drama', 'Comedy', 'Thriller', 'Action', 'Romance',
'Sci-Fi', 'Documentary', 'Horror', 'Fantasy', 'Mystery', 'Adventure') NOT NULL
UNIQUE   -- Predefined genres
);

-- Create the 'Programme' table to store information about programmes
CREATE TABLE Programme (
   programme_id INT AUTO_INCREMENT PRIMARY KEY,    -- Unique ID for each
programme
   programme_name VARCHAR(255) NOT NULL UNIQUE,    -- Name of the programme
   genre_id INT,   -- Foreign key referencing the Genres table
   release_date DATE,  -- Release date of the programme
   description TEXT,   -- Short description of the programme
   duration INT,   -- Duration in minutes (optional)
);

-- Create the 'Cast' table to store information about the cast
CREATE TABLE Cast (
   cast_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique ID for each cast member
   cast_name VARCHAR(255) NOT NULL UNIQUE, -- Name of the cast member
   date_of_birth DATE, -- Date of birth of the cast member (optional)
   bio TEXT    -- A short biography (optional)
);

-- Create the 'Programme_Cast' table to store the many-to-many relationship
between programmes and cast members
CREATE TABLE Programme_Cast (
   programme_id INT,   -- Foreign key referencing the Programme table
   cast_id INT,    -- Foreign key referencing the Cast table
```

```
    role VARCHAR(255),  -- Role of the cast member in the programme (optional)
    PRIMARY KEY (programme_id, cast_id),    -- Composite primary key for the
many-to-many relationship
);

-- Create the 'Episodes' table for programmes with episodes (like TV shows or
series)
CREATE TABLE Episodes (
    episode_id INT AUTO_INCREMENT PRIMARY KEY,  -- Unique ID for each episode
    programme_id INT,   -- Foreign key referencing the Programme table
    episode_name VARCHAR(255) NOT NULL, -- Name of the episode
    season_number INT,  -- Season number (for series with multiple seasons)
    episode_number INT, -- Episode number within the season
    release_date DATE,  -- Release date of the episode
    description TEXT,   -- Short description of the episode
    duration INT,   -- Duration in minutes
);

-- Creating additional indices for performance optimization
CREATE INDEX idx_programme_name ON Programme (programme_name);
CREATE INDEX idx_cast_name ON Cast (cast_name);
CREATE INDEX idx_genre_name ON Genres (genre_name);
CREATE INDEX idx_episode_name ON Episodes (episode_name);
```

- ott_db_alter.sql:

```
-- Add foreign key from Programme to Genres
ALTER TABLE Programme
ADD CONSTRAINT fk_programme_genre
FOREIGN KEY (genre_id) REFERENCES Genres (genre_id)
ON DELETE SET NULL;

-- Add foreign key from Programme_Cast to Programme
ALTER TABLE Programme_Cast
ADD CONSTRAINT fk_programme_cast_programme
FOREIGN KEY (programme_id) REFERENCES Programme (programme_id)
ON DELETE CASCADE;

-- Add foreign key from Programme_Cast to Cast
ALTER TABLE Programme_Cast
ADD CONSTRAINT fk_programme_cast_cast
```

```
FOREIGN KEY (cast_id) REFERENCES Cast (cast_id)
ON DELETE CASCADE;

-- Add foreign key from Episodes to Programme
ALTER TABLE Episodes
ADD CONSTRAINT fk_episodes_programme
FOREIGN KEY (programme_id) REFERENCES Programme (programme_id)
ON DELETE CASCADE;
```