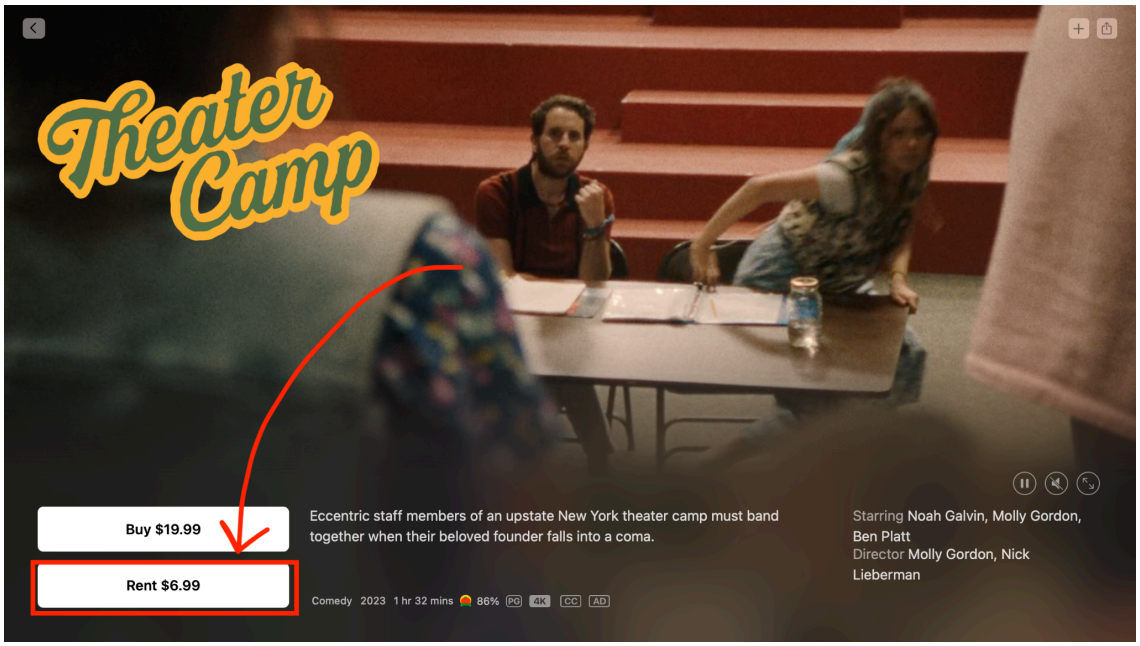
Project 1: Online Movie Rental Platform



Objective:

This project provides hands-on experience in designing, creating, and querying a database for a real-world application. You will design a database for an **online movie rental platform**, where users can rent movies and stream them online. Rentals automatically expire in 28 days unless the user starts watching, in which case the rental expires 48 hours after the start.

Scenario:

You have been hired as a database consultant to design and query a database for an **online movie rental platform** like Apple TV, YouTube, or Prime Video. The platform allows users to browse a catalog of movies, rent them, and stream them online. Rentals automatically expire in 28 days regardless of the user watching them. Once the user starts watching the video, the rental expires 48 hours later.

Your task is to:

1. Design the database schema;
2. Create tables with appropriate constraints;
3. Populate the database with sample data;
4. Write queries to analyze and retrieve information.

Instructions:

Step 1: Database Design

Design the schema for the following entities:

1. **Users**
    - user_id (Primary Key)
    - name (TEXT, Not Null)
    - email (TEXT, Not Null, Unique)
    - subscription_type (TEXT, e.g., "Basic", "Standard", "Premium")

2. **Movies**
    - movie_id (Primary Key)
    - title (TEXT, Not Null)
    - genre (TEXT, Not Null)
    - release_year (INTEGER, Not Null)
    - rating (TEXT, e.g., "PG", "R")
    - rental_price (REAL, Not Null)

3. **Rentals**
    - rental_id (Primary Key)
    - user_id (Foreign Key referencing Users.user_id)
    - movie_id (Foreign Key referencing Movies.movie_id)
    - rental_date (TEXT, Not Null) -- The date the movie was rented.
    - start_date (TEXT) -- The date the user started watching (if applicable).
    - expiry_date (TEXT, Not Null) -- Automatically set as*:
        - 28 days after rental_date if start_date is NULL.
        - 48 hours after start_date if the user begins watching.
    - status (TEXT, Not Null, e.g., "Active", "Expired")

*You will learn how to set the value automatically later this term. In Step 4 below, you can ignore these conditions and insert a date that is 28 days after the rental date.

Step 2: Create the ER Diagram*

*Note: This topic will be covered in Week 6. Skip this section if you are working on this project before Week 6 and come back to it when you are done with the Week 6 materials.

Create the ER Diagram for this database using a tool of your choice. Save (or Export) it as PNG or PDF. Rename the file to **yourfirstname_erd.png** (or .pdf)

Step 3:Create the Database

Using SQLite, create the database named *yourfirstname_online_movie_rental.db* and tables based on the schema above. *(Replace yourfirstname with your first name)*

- Include primary keys, foreign keys, and appropriate constraints.
- Ensure the data types match the real-world requirements.

Step 4: Populate the Database*

*Note: This topic will be covered in Week 5.

Insert at least:

- 5 users (with different subscription types).
- 10 movies (from at least 3 different genres).
- 10 rentals with varying:
    - rental_date (some old, some recent).
    - start_date (some null, some populated).
    - expiry_date (enter the date 28 days after the rental date. No calculation needed for this project).
    - status (some active, some expired).

Step 5: Write SQL Queries

Write and execute SQL queries to answer the following questions (Save all SQL queries in a file named yourfirstname_queries.sql):

1. List all users and their subscription types.
2. Find the titles of all movies in the "Drama" genre.
3. Retrieve all rentals that are still active (status = "Active").
4. Calculate the total revenue generated from all rentals.

5. Display the names of users who rented the most movies.
6. List all movies rented by users with a "Premium" subscription.
7. Display all movies released after 2019, sorted by release year in descending order.
8. Calculate the average rental price for movies in the "Horror" genre.
9. Find all rentals that have expired, along with their rental and expiry dates.
10. List all users who rented a movie that expired before they started watching.

Step 6: Prepare for Submission

1. Export your SQLite database file *(=make a copy of the .db file)* and name it **yourfirstname_online_movie_rental.db**.
2. Save your SQL queries in a file named **yourfirstname_queries.sql**.
    - Each query should be numbered and include comments describing its purpose.
3. Write a brief report (maximum 1 page) in **yourfirstname_report.txt** or **yourfirstname_report.pdf**, describing:
    - Your approach to designing the database.
    - Any challenges faced and how you solved them.
    - Key insights gained during the project.

Note: Replace yourfirstname with your first name.

How to Submit

1. Put all of the following files into a folder and compress it in ZIP format:
    - **yourfirstname_online_movie_rental.db**
    - **yourfirstname_queries.sql**
    - **yourfirstname_report.txt** or **yourfirstname_report.pdf**
    - **yourfirstname_erd.png (or yourfirstname_erd.pdf) ER Diagram**
2. Name the compressed file **project1_firstname_lastname.zip**, replacing firstname and lastname with your name.
3. Submit the .zip file by the deadline.

Grading Rubric (Out of 15 Points)

Criteria	Points	Description

Database Design	3	Tables are correctly structured with appropriate constraints (primary key, foreign key, etc.).
ER Diagram	3	Relationships between entities are clearly visualized using Crows Foot Notation. Column titles, type and constraints are included in each entity.
Data Population	2	Sample data is complete, realistic, and meets the requirements.
SQL Queries	5	Queries are correct, efficient, and provide the expected results.
Report	2	Brief, clear report describing the approach and challenges.