🎥Movie Rental Dataset

DBS

August 1 2025

👩‍💻Suganya Maheswaran

👩‍💻Alina Pasat

👩‍💻Ayushkumar Sachchanand Fatwani

📽️Introduction

This movie_rental database stores data related to movie rentals, customers, employee management, and transaction records. It is designed for a single-branch rental company and includes collections such as:

- movies – Metadata about each movie
- customers – Customer personal and contact information
- rentals – Individual rental transactions
- employees – Store employees and their roles
- inventory – Tracks available copies per movie
- orders – Bundled rentals per customer
- payments – Payment records linked to orders

🍿 Assumptions

- Single Branch: No branch or location collections are necessary.
- Customers: Each customer:
  - Can rent multiple movies
  - Has only one account, uniquely identified by their phone number
  - Must be over 18 years old (to be verified via date_of_birth)
- Movies:
  - Can belong to multiple genres (e.g., ["Drama", "Horror"])
  - Include detailed cast with character names and awards
- Orders:
  - Can include multiple rentals
  - Multiple payment methods may be used
- Rentals:
  - Are linked to customers, employees, and specific movie inventory

Criteria for when to embedded and when to create an entity:

We chose to create the following entities for usability and accessibility

Customer

Movies

Rentals/Orders

Employees

Payments

We chose to embed the following

Inventory will be inside movies

🎞️Data Model Overview

Collection	Purpose
`movies`	Stores movie details including cast, genre, ratings
`customers`	Stores customer personal data and address
`employees`	Stores employees data, role, and salary info
`rentals`	
`orders`	**Contains rental arrays (1 order = 1+ rentals)**
`payments`	Records customer payments linked to orders

**not sure if we are making changes to the schema

🎬Collections & Sample Documents

📁Customers

json

```
{
    "_id": 101,
    "first_name": "Jacob",
    "last_name": "Newman",
    "email": "alexandriaphillips@yahoo.com",
    "phone_number": "905-684-3600",
    "address": {
      "street": "3663 Carling Avenue",
      "city": "Toronto",
      "province": "ON",
      "Postal_code": "M1P 1X3"
    },
    "date_of_birth": "1956-03-08"
}
```

📁Employees

json

```
{
    _id: 2,
    first_name: "Gregory",
    last_name: "Lee",
    role: "Associate",
    employment_type: "Full-Time",
    email: "gregory.lee@rentalco.com",
    date_of_birth: "1988-11-22",
    address:{
      street: "27 Nymark Ave",
      city: "Toronto",
      province: "ON",
      postal_code: "M2J 1V2"
   },
    salary: 42000,
    weekly_hours: 40
  },
```

📁Rentals

json


```
{
    "_id": 10002,
    "cust_id": 107,
    "movie_id": 1016,
    "empl_id": 2,
    "rental_date": "2020-07-14",
    "return_date": "2020-07-21",
    "rental_fee": 8.28
  },
```


📁Movies

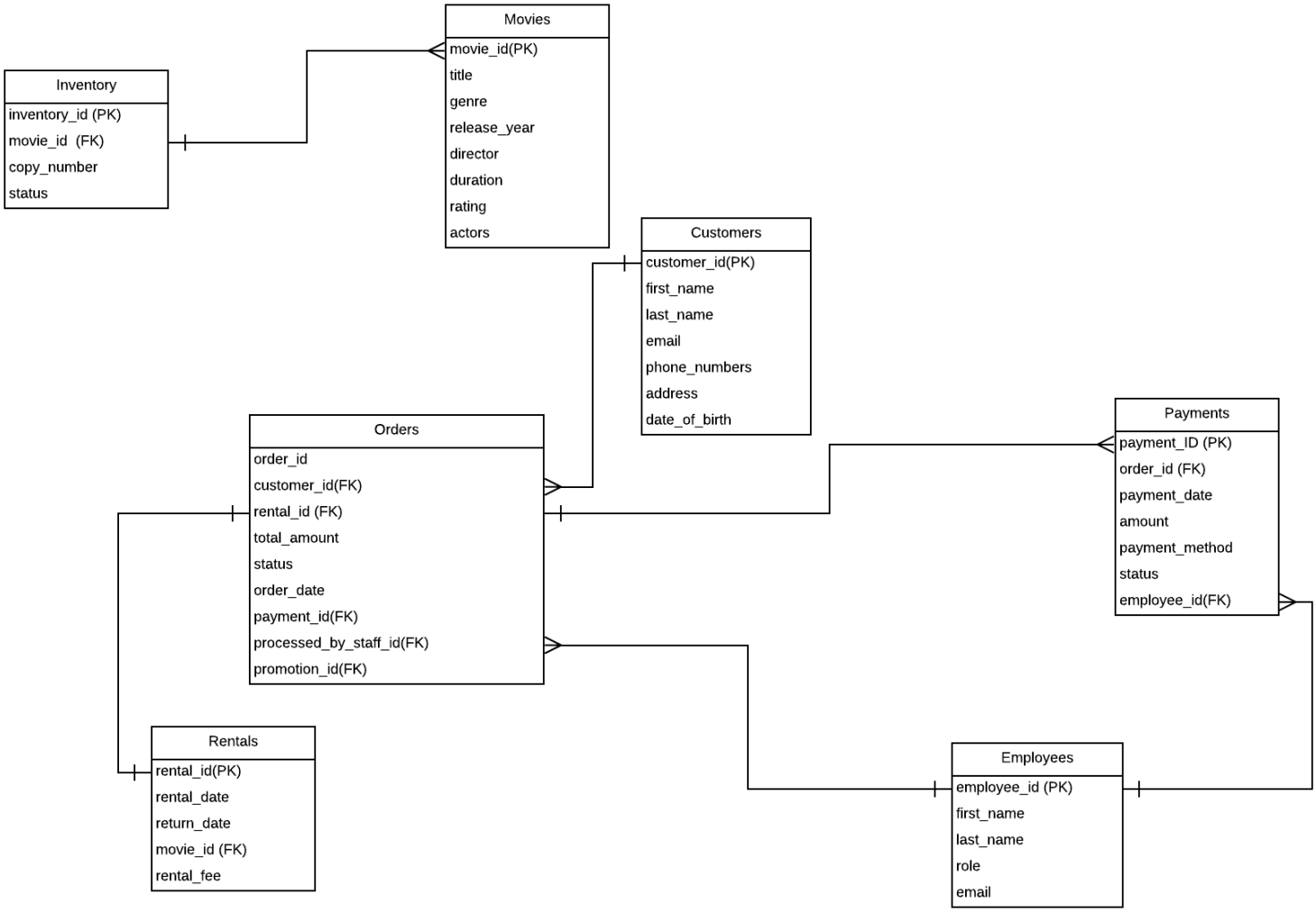
json


```
{
    "_id": 1001,
    "title": "Center word school",
    "genre": [
      "Sport"
    ],
    "release_year": 2019,
    "cast": [
      {
        "actor": "Stacy Fisher",
        "character": "Glenn",
        "awards": []
      },
      {
        "actor": "Sheila Hamilton",
        "character": "David",
        "awards": [
          "Emmy",
          "BAFTA"
        ]
      },
      {
        "actor": "John Houston",
        "character": "Sonya",
        "awards": [
          "Golden Globe",
          "BAFTA"
        ]
```

```
      }
    ],
    "director": "Nicholas Edwards",
    "duration": 128,
    "language": "French",
    "imdb": {
    "rating": {
      "$numberDouble": "7.3"
    },
    "votes": {
      "$numberInt": "5043"
    },
    "id": {
      "$numberInt": "12"
    }
    "copies": 5,
    "available": 4
  },
```

📽️ERD

🔗


Movies
movie_id(PK)
title
genre
release_year
director
duration
rating
actors
Inventory
inventory_id (PK)
movie_id (FK)
copy_number
status
Customers
customer_id(PK)
first_name
last_name
email
phone_numbers
address
date_of_birth
Orders
order_id
customer_id(FK)
rental_id (FK)
total_amount
status
order_date
payment_id(FK)
processed_by_staff_id(FK)
promotion_id(FK)
Payments
payment_ID (PK)
order_id (FK)
payment_date
amount
payment_method
status
employee_id(FK)
Rentals
rental_id(PK)
rental_date
return_date
movie_id (FK)
rental_fee
Employees
employee_id (PK)
first_name
last_name
role
email

🎦Database Creation & Insertion & Querying Scripts

Database Creation Scripts

1. Create database movierentals
    a. Open terminal
    b. Run command **mongosh**
    c. Check if database is created by running **show dbs**
    d. In order to create database run **use movie_rental_database**
2. To Create Collection:
    a. Open powershell or terminal
    b. Run command **mongosh**
    c. Copy the data inside dbCollectionScript.txt and run in terminal

Data Insertion Scripts

1. Insert Customer into Customer collection
    a. Open terminal
    b. Run command **mongosh**
    c. To change database run **use movie_rental_database**
    d. Copy the data inside customerScript.txt and run in terminal

2. Insert Movie data into Movie collection
    a. Open terminal
    b. Run command **mongosh**
    c. To change database run **use movie_rental_database**
    d. Copy the data inside movieScript.txt and run in terminal
3. Insert Employees data into Employee collection
    a. Open terminal
    b. Run command **mongosh**
    c. To change database run **use movie_rental_database**
    d. Copy the data inside employeeScript.txt and run in terminal
4. Insert Rental data into Rental collection
    a. Open terminal
    b. Run command **mongosh**
    c. To change database run **use movie_rental_database**
    d. Copy the data inside rentalScript.txt and run in terminal

Querying Data:

a. Find all movies in a specific genre.

b. Find all movies released in a specific year or range of years.

c. Find all customers who have rented a movie of a specific genre.

d. Find the total rental fees collected from a specific customer.

e. Find the most popular movie(s) based on the number of rentals.

Data Aggregation:

a. Calculate the average rental fee per genre.

b. Calculate the total rental fees collected per month.

c. Find the top 3 most popular genres among customers.

Data Update (CRUD)

a. Update the rental fee for all movies in a specific genre.

***NOT WORKING error: Cannot Convert circular structure to BSON

json

```
// Step 1: Get movie IDs by genre (e.g. Drama)

var movieIds =

// Step 2: Update rental fees in rentals collection

db.rentals.updateMany(
  { movie_id: { $in: db.movies.find({ genre: "Drama" }).map(m
=> m._id)} },
  { $set: { rental_fee: 9.99 } }
);
```

b. Update the contact information (phone number and address) for a specific customer.

json

```
db.customers.updateOne(
  { _id: 101 }, <- insert id of customer you wish to update
  {
    $set: {
      phone_number: "416-123-4567", <- insert new values
      address: {
        street: "123 Queen St",      <- insert new values
        city: "Toronto",             <- insert new values
        province: "ON",              <- insert new values
        postal_code: "M5V 2B1"       <- insert new values

      }
    }
  }
);
```

Conclusion