Manage a chain of Movie Rental Stores

Introduction

In this project you will write more advanced queries on a database designed to resemble a real-world database system - MySQL's Sakila Sample Database.

Development of the Sakila sample database began in early 2005. Early designs were based on the database used in the Dell whitepaper (Three Approaches to MySQL Applications on Dell PowerEdge Servers).

The Sakila sample database is designed to represent a DVD rental store. The Sakila sample database still borrows film and actor names from the Dell sample database.

Problem Description

You're writing SQL to manage a chain of movie rental stores, for example,

- Track the inventory level and determine whether the rental can happen
- Manage customer information and identify loyalty customers
- Monitor customers' owing balance and find overdue DVDs

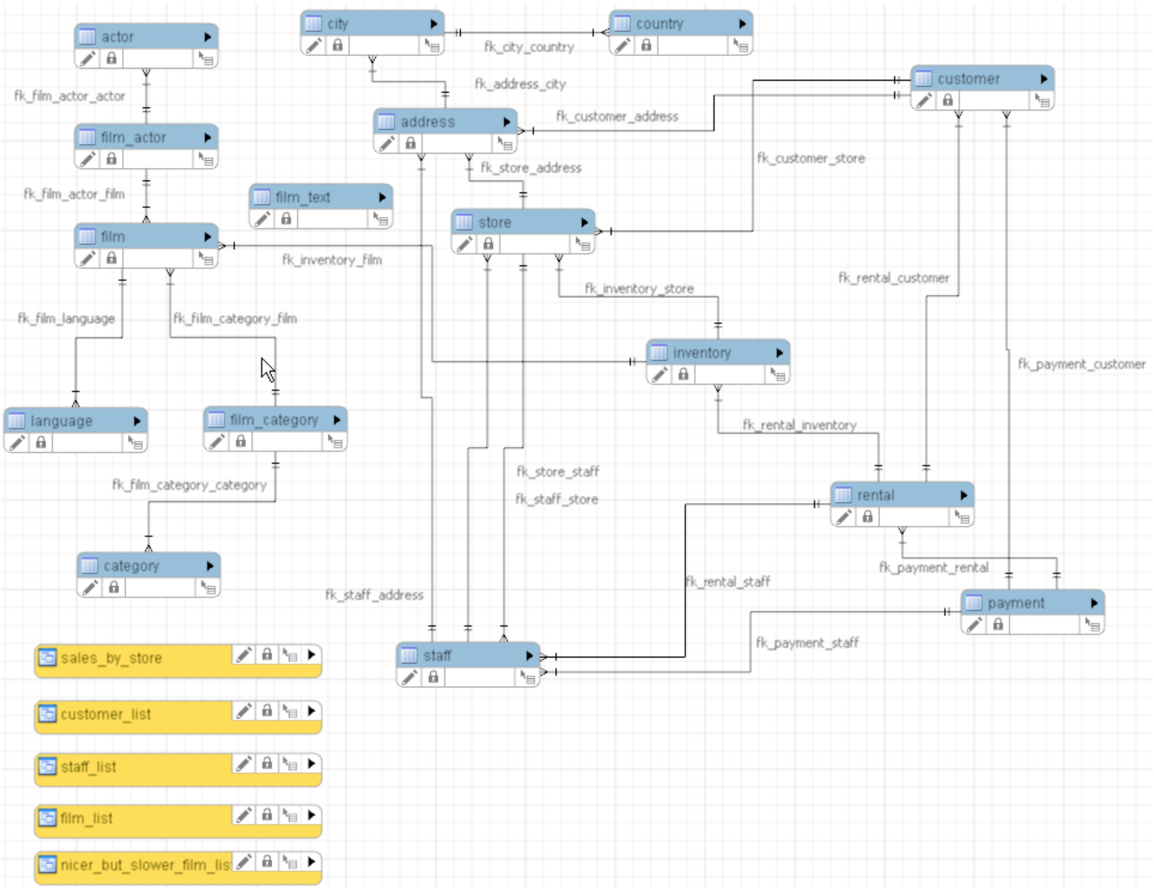
This project can be considered as a typical retail-related business case, because it has the main metrics you can find in any retailer's real database, such Walmart, Shoppers, Loblaws, Amazon…

Key Metrics:

- **Production information (in this project, it is the film)**
- **Sales information**
- **Inventory information**
- **Customer behavior information**

Data Structure

The following diagram provides an overview of the structure of the Sakila sample database.



Exercise 1 (for Day1-Day3):

1. Before doing any exercise, you should explore the data first. For Exercise 1, we will focus on the product, which is the film (DVD) in this project. Please explore the product-related tables (actor, film_actor, film, language, film_category, category) by using 'SELECT *' – do not forget to limit the number of records

Use table FILM to solve questions as below:

2. What is the largest rental_rate for each rating?
3. How many films in each rating category?
4. Create a new column film_length to segment different films by length:
   length < 60 then 'short'; length < 120 then 'starndard'; lengh >=120 then 'long'
   , then count the number of files in each segment.

Use table ACTOR to solve questions as below:

5. Which actors have the last name 'Johansson'
6. How many distinct actors' last names are there?

7. Which last names are not repeated? *Hint: use COUNT() and GROUP BY and HAVING*
8. Which last names appear more than once?

Use table FILM_ACTOR to solve questions as below:

9. Count the number of actors in each film, order the result by the number of actors with descending order
10. How many films each actor played in?