THE COPPERBELT UNIVERSITY

RENT A MOVIE DATABASE SYSTEM

CS 235 ASSIGNMENT

Lecturer: Mr. Chibuluma

GROUP MEMBERS

BRIGHT SIMUMBWE	22105961
Alex HIMOONGA	22179213
MUTUMWA MTONGA.	22110590
DALITSO TEMBO.	22107126
GIVEN CHISANGA.	22102416
EMMANUEL SIAMOONGA	22107671
EMELDAH MIYANDA.	21164180
GIFT M SHIMATUBULU	2102262
GIFT KAWAMWILU	22108950

LUKONDE MULENGA. 22109721

CONTENTS PAGE

INTRODUCTION

This project involves designing a relational database for Rent-A-Movie, aimed at streamlining customer and rental management while bolstering marketing efforts. Utilizing MySQL Work Bench, the database will feature customer and movie searches, zip code-driven location services, rental tracking, personalized communications and actor-based movie queries.

We will define the data schema thereby constructing tables with keys, enforce referential integrity and input sample data. Queries will extract customer and transaction data, and tools like SSRS will report on transactions. Our goal is to deliver a robust, intuitive database that elevates Rent-A-Movie's customer service and marketing capabilities.

BACKGROUND

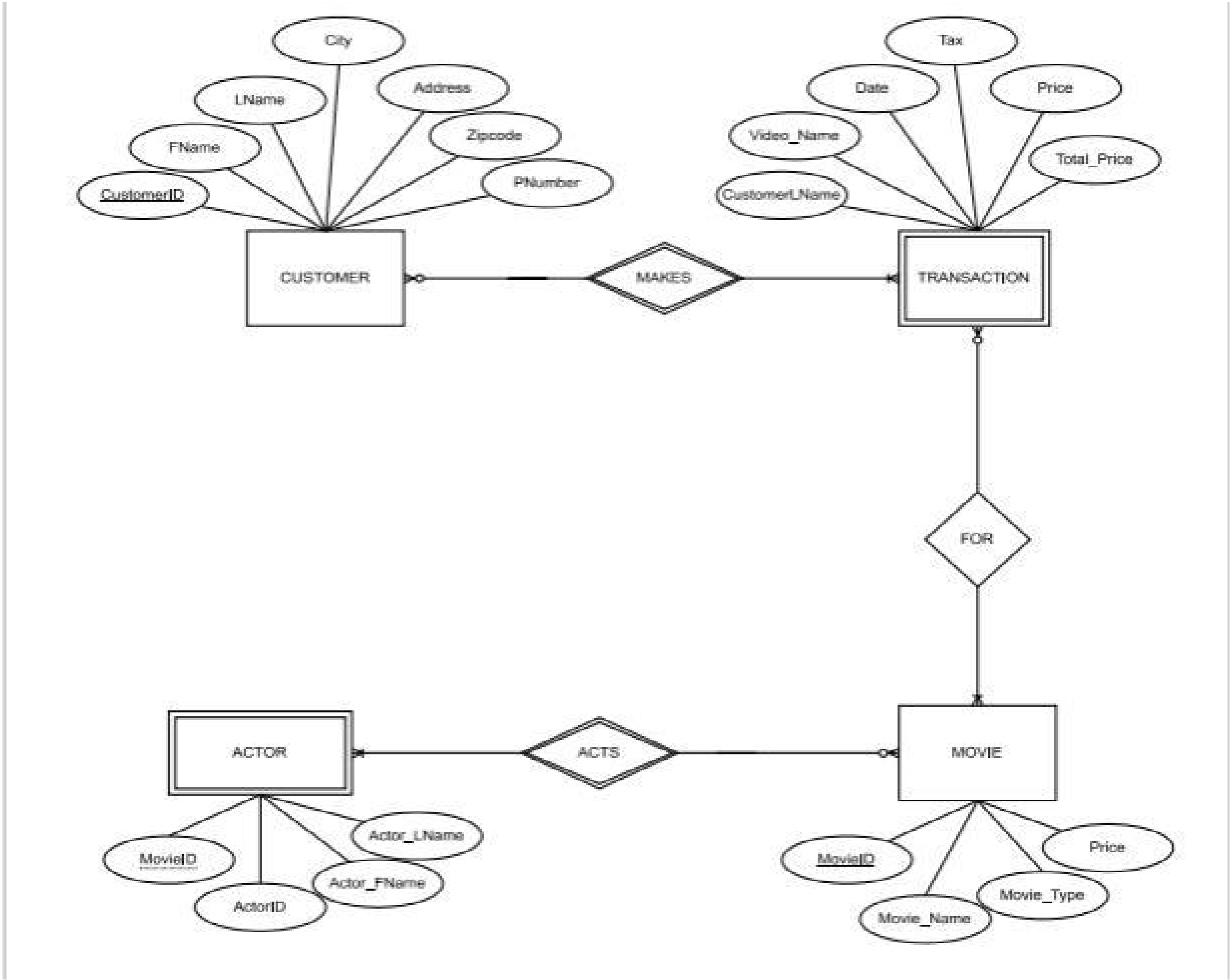
Rent-A-Movie, a customer video rental store, seeks to enhance operational efficiency and customer service. The store plans to leverage a structured database system to manage customer details and movie inventory effectively. The anticipated database will support strategic marketing and improved service delivery by utilizing data on customer preferences and rental patterns. This initiative is expected to boost customer loyalty and store profitability through tailored services and marketing.

PROBLEM STATEMENT

Rent a movie is a system which designs a database system that efficiently manages movie rentals, customer information, available movie inventory and transactions to ensure smooth operations and accurate tracking of records. Tables are created to store information about customers (customer ID, customer name, phone number, cities), movies (movie ID, movie type, movie price), actors (actor name, actor ID) and so on and so forth. Each table will have a primary key to uniquely identify each record and there will be relationships between tables. The database will handle queries like finding available movies, tracking customer rentals and generating reports on popular movies.

DISCUSSION OF SOLUTIONS

W e constructed an ER-diagram using the information provided in the question paper. The following is the ER-diagram. ER-diagram is shown below.

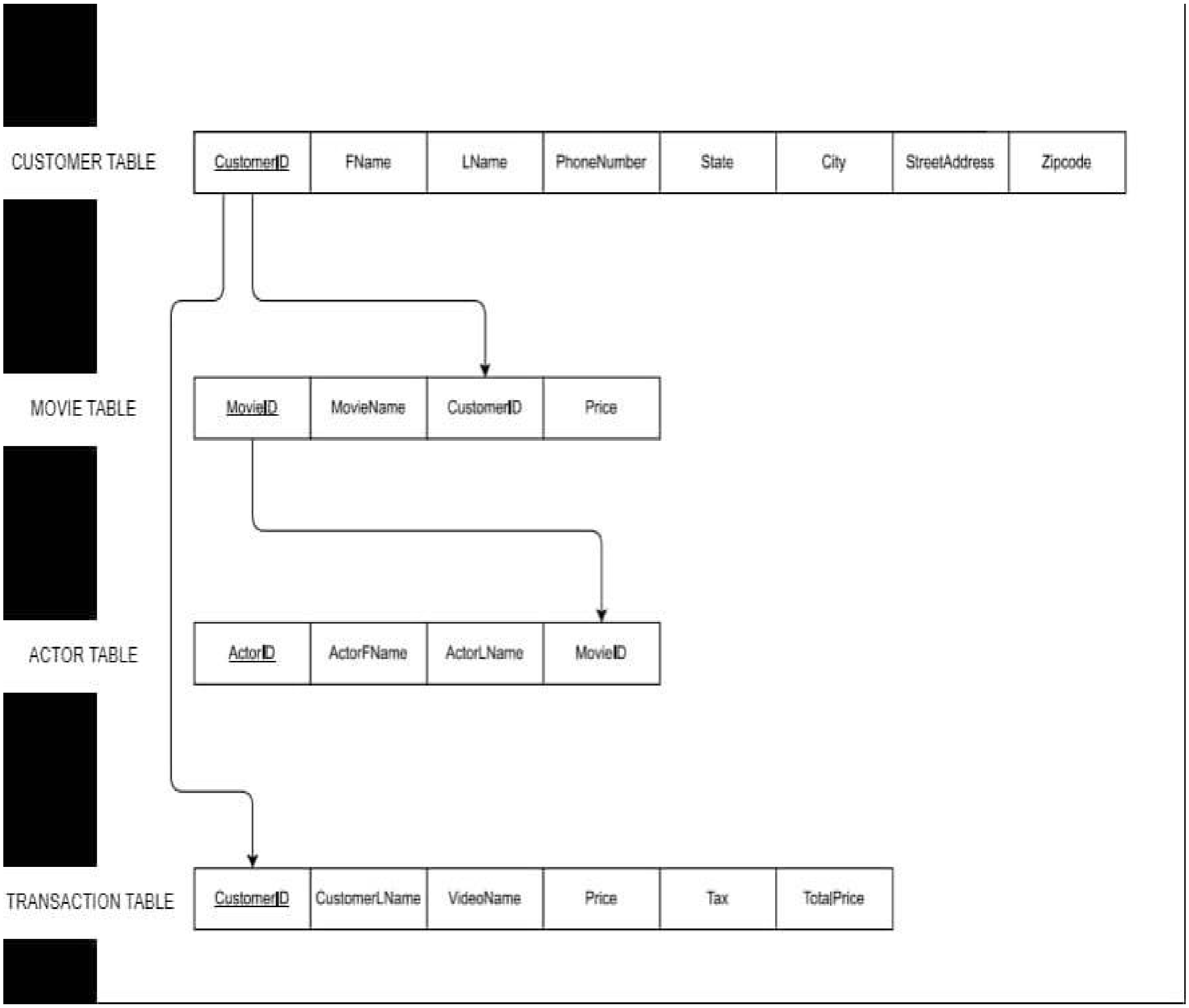


From the entity relational diagram (ERD), the following information was captured. They are four entities which are; customer, movie, transaction and actor.

- Customer entity with its attributes: customer ID as the primary key, phone number, street address, first name, last name, zip code and city.

- Movie entity with its attributes: Movie ID as the primary key, movie name, movie type and price.
- Transaction with its attributes: Customer name, video name, tax and price.
- Actor with its attributes: Actor ID as the primary key and actor name.

Therefore the ER diagram was converted into relational schemas using my SQLite capturing all the foreign keys.



CUSTOMER TABLE

CustomerID	FName	LName	PhoneNumber	State	City	StreetAddress	Zipcode

MOVIE TABLE

MovieID	MovieName	CustomerID	Price

ACTOR TABLE

ActorID	ActorFName	ActorLName	MovieID

TRANSACTION TABLE

CustomerID	CustomerLName	VideoName	Price	Tax	TotalPrice

OBJECTIVES

- Develop a database system to store and organize customer information, including personal details, rental history and preferences.
- Enable quick and accurate customer searches by last name or phone number.
- Create a comprehensive movie catalogue with details such as title, genre, release date, and actors.
- Implement efficient search functionality for movies based on name, type, or actor.
- Record rental transactions, including rental dates and amounts spent.
- Calculate total spending per customer and overall revenue
- Utilize customer data to send personalized letters or emails, addressing customers by their first names.
- Implement targeted promotions based on rental history and preferences.
- Generate transaction reports for management and marketing purposes.
- Analyse trends, popular movies, and customer behaviour to inform business decisions.

LIMITATIONS

Lack of mobile optimization, the website is not optimized foe mobile for mobile devices, providing a poor user experience. The website does not send email notifications for movie availability, rentals, or returns. No integration with social media, the website does not allow users to share movie reviews or ratings on social media platforms. The database may become slow and unresponsive with large number of users and movie records.

CONCLUTION

The Rent-Me-Movie website and MySQL database project demonstrates a comprehensive online movie rental system. A user-friendly website with robust search function and intuitive interface, efficient data retrieval and manipulation using SQL queries. The website allows users to browse, search, and rent movies while the MySQL database efficiently stores and manages movies information, user data and rental transactions. A scalable and reliable database management system.

REFERENCES

-SSRS (SQL Server Reporting Services): https://learn.microsoft.com/en-us/sql/reporting-services/install-windows/install-reporting-services?view=sql-server-ver16

MYSQL Workbench- https://www.mysql.com/products/workbench/

ERD and Relational Model - https://lechaamwe.weebly.com/cs235.html