**Final Project: Movie Rental Store**

**Group members:**

Ian Duncan

Brandon Shankle

**Abstract**

In summary, our database is built for an entertainment business that is selling movies to rent for a period of days among several store locations and to multiple people. This should keep track of movies, inventory, customers, store locations, rentals, and payment. These records allow the business to keep track of stock, money and customers among several locations. Store A could track 10 customers and the movies it has in stock and who has what compared to overdue fees and expected return dates as well. This in turn can be shared to overall profits throughout the business and stock among all stores as they could easily switch inventory among each other. This framework could be altered to work for other kinds of businesses as well, not just rental focused stores. This means that this format could easily be a great foundation for any business management system.

**Requirement Analysis**

1. The business will store in its database a detailed record of each store, inventory, movies, customer, and payment. One record will include the customer’s first name, middle initial and last name, customer ID, address (includes the city, state, zip code and street). Another record will have a list of movies per store and be tied to the inventory overall with each movie’s release date, director, MPAA rating, runtime, genre and a unique ID to track certain movies. Inventory will have the number of movies overall and each movie’s current location tied with a store location. A rented movie will have a start date for the rental and an end date as well as an id to track it and an overdue fee if the movie is returned past the expected end date. Payment is tracked with how much was paid, the payment method, and the start date and end date of the rental.
2. The IDs for each record will all be tied in a way, from Customer ID being tied to payment ID which is then tied to a rental ID which in turn is tied to an inventory ID to a movie ID so as to track who has paid for what movie among all stores.
3. The inventory location will be tied to certain store locations to track where movies are among the stores.
4. A Store has a unique location and a unique ID.

**Assumptions**

1. We assume that all customers have a unique customer ID, stores have unique store IDs and locations, inventories have unique inventory IDs, payments are tracked through payment IDs, Rentals are tracked through rent IDs, and movies are tracked through Movie IDs.
2. Each customer and payment can have only one unique ID.
3. A store must have a unique location.
4. A movie can be in inventory and share the same inventory ID with movie ID.
5. Rental and payment must share the same start dates and end dates if they share the same IDs.

**Schema Diagram**

**Third Normal Form:**

Table: Customer

Attributes: fname, minit, lname, address, customer\_ID

Relations:

1. Customer\_ID -> fname, minit, lname
2. Customer\_ID -> address
3. Fname, minit, lname -> address

3nf:

1. Fname, lname, minit -> address
2. Customer\_ID -> lname, minit, lname

Table: Movie

Attributes: Movie\_Runtime, Movie\_ID, Movie\_Director, movie\_ReleaseDate, Movie\_Rating, Movie\_Genre

Relations:

1. Movie\_Runtime, Movie\_Director, movie\_ReleaseDate, Movie\_Rating, Movie\_Genre -> movie\_ID

3nf: Already in 3nf

Table: Inventory

Attributes: m\_ID, num\_Of\_Movies, inv\_Locations

Relations:

1. Num\_Of\_Movies -> Inv\_Locations

3nf:

Already in 3rd normal form

Table: Store

Attributes: store\_ID, store\_Location

Relations:

1. Store\_ID -> store\_Location

3nf:

Already in third normal form

Table: Rent

Attributes: rent\_ID, start\_Date, end\_Date. overdue

Relations:

1. Rent\_ID -> start\_Date, end\_Date, overdue

3nf:

Already in 3rd normal form.

Table: Payment

Attributes: Payment\_ID, paid, paymentMethod, startDate, endDate

Relations:

1. Payment\_ID -> paid, paymentMethod, startDate, endDate
2. PaymentMethod -> paid

3nf:

Already in third normal form.

**Query Outputs**

**Display Students Query**

**Add/Display Section Queries**

**Register Student and Display Queries**

**Delete Department Query**

**Edit Department Query**