Executive Summary:

To: Old Time Movie Rentals
From: JAB
Re: Executive Summary

After an entire month, we have finished the project for Old Time Movie Rentals. Below is an executive summary of the work we have completed. I hope you are satisfied with the solutions we have imposed.

To start the project, we took time to analyze the information provided to us by your company, Old Time Movie Rentals. We separated the related information into unique tables, and determined the primary keys and foreign keys for said tables. After this, we created an entity relationship diagram using an online application and determined the necessary relations, constraints, and cardinalities. Using this diagram, we were able to visualize the database to ensure that it would work efficiently and effectively. For most tables, we created unique surrogate keys as primary keys for the table. These surrogate keys allow the table to be queried easily. After creating the diagram, we examined the functional dependencies within every table, and tested our design using an online tool to ensure that our relationships were in Boyce-Codd Normal Form, which it was. One issue we ran into during this design process was how to represent managers. Technically, managers were also employees, so we added a recursive relationship to the employee table to represent the relationship between employees and managers. We also created a new entity to represent the cash payment type, since we believed it would best prevent any anonymity regarding payment type.

After we created and verified our ER Diagram, we moved onto implementing our design using SQL. We created SQL code to create each table, and specify the primary keys, foriegn keys, and constraints. For the majority of the attributes, we prevented that from being null to prevent any possible anomalies. We also used variable character lengths up to 30 characters to allow a good compromise between efficiency and uniqueness within the database. We also specify the keys within our table creation code to ensure that all of the entities and attributes within the database remain consistent.

Next, we created model data for the database that we used to test our design against possible user queries. The data we used was based on a small company, similar to that of your company. After implementing the SQL to insert the data, we double checked that the data was consistent and did not violate any of the constraints we implemented such as foriegn key possibly violating the referential integrity constraints. To be exact, we created and ran ten queries that would test our design and ensure it worked as intended. All of said queries ran efficiently, and produced the expected result.

To finish off the project, we created two web applications to allow the database to be viewed and used effectively by any user. First, we created an application that would be used at the company level by employees. This application allowed the employee to view most of the database, but

not edit all of it. In some specific scenarios, we used views to restrict what the employee could see, one of these being we did not allow the employee to see credit card or debit card number within the DebitCard and Credit Card entities respectively. This application also allowed employees to create new transactions as MovieRental entities when needed. The second application we created was a catalog application. This application ran on the internet, and would allow any customer to view and query the database limitedly. We used views to allow the customer to see the necessary information they needed, adn to use specific queries, such as seeing what store carries what movie.

If you are satisfied with our proposed solutions, please feel inclined to leave a positive review on our website. If there are any problems with our solution, feel free to reach out to us any time in the future.