|  |  |
| --- | --- |
| Group Number | 15 |
| Group Name | Chew Language |
| Names | Leong Chang Yung |

COMP1044 Coursework Assignment 2 report

Group 15: Chew Language

# Lucas Dylan Purnell 20197316

IGNORE THE FIRST TWO PARAGRAPHS AS IT WILL BE REWRITTEN WHEN DATABASE IS EDITED

For every field we chose the datatype depending on what would both match the data we had available and take up the lowest amount of memory possible.

In the actor table this meant that the field was int(3) as every value was a integer with the longest value having a length of 3. The first\_name and the last\_name fields are both varchar with a maximum length of 11 and 12 respectively as all values within these fields were strings with those maximum lengths.

## Reason for the primary keys and foreign keys:

When we were deciding on what to pick for the primary key for each time we understood that it had to be both unique and non-null. This is because the primary key serves as the identifier for each record in the table and if it is either repeated or null it can no longer serve as an identifier. The obvious candidate here was the ID field in every table(for instance the inventory\_id in the inventory table) as it fulfils both of these criteria.

The foreign key has the same requirements. It needs to be both unique and independent in the array it belongs to. With the in mind we can use the same fields that we used as primary keys. Therefore for each of our arrays the id column is used to link the tables together. For instance, customer\_id was chosen as the foreign key to link the payment table with the customer table and address\_id was chosen as the foreign key to link the staff table with the address table.

## Completeness of the dataset:

Probably will edit this later

We found a few different ways that the dataset was incomplete. The first was missing numbers from the auto increment data. For instance, there were missing address\_id’s in Address.csv. This problem was easily remedied by setting it to auto increment.

The next problem was that many values that should not be null were null. For instance, some of the fields of Rental\_id in Payment.csv were null when this should of not been the case. To remedy this we used the notnull function which then prompted the user to provide an input for these null fields(and if they don’t it will keep the field’s null value). This was the solution that we settled on as it allows the null fields to be updated without forcing these fields to contain garbage values.

## Errors in the database:

We also noticed that in country.csv and customer.csv that there wasn’t a consistent delimiter. The file alternated between using commas and semi-colons(to separate its values) depending on the column. This meant that it was impossible to important the data into a database while keeping the data in its intended format. To remedy this we just changed the semi-colons for commas wherever applicable using a notepad application.

The other error that we found was that there was a corrupted picture file within staff.csv. This corruption it unable to view or display the aforementioned picture. We decided that having a picture within a staff database was not of high importance and wouldn’t decrease the completeness or understandability of the dataset so we just removed thee picture. A possible solution to the picture being corrupted would be to have a link that directs to the picture instead of trying to directly attach the picture.