Justin Phillips
John Walbert
CS340 fall 2019
10/6/19

Project Step 2 (Draft Version)

Feedback

1. Based on the current structure of the database, it is impossible to present many-to-many relationships between checkout and books and checkout and movies. You need to add an associative entity.

   Fixed by adding the relationship tables customer_books and customer_movies.

2. The datatype of phone number should be varchar. The disadvantages of INT here are: you cannot store a dash and the leading zero may be missed.

   Fixed by changing the data type of the phone number attribute to varchar.

3. Please specify the number of characters for all varchars.

   Fixed by specifying varchar(255) for all attributes using the varchar datatype.

b)Updated Outline

Overview

The project that we propose is a database that will handle the checkout process for a library. It will allow the library patron to view the available titles, check items out if they are available, and provide a due date for return.

Entities

The first entity that we will have is ***customers***. Their attributes will include library card number (a unique identifier), first name, last name and a telephone number. The second entity will be the ***checkout***. It is the transaction itself, where the customer borrows books or movies. Its attributes will include transaction ID (a unique identifier) and due date for return. The third entity will be ***books***. Their attributes will include author of the book, the title of the book, a unique ID and if it is already checked out or not. The fourth and final entity is ***movies***. Similar to books, their attributes are title, the year the movie was released, a unique ID and if it is checked out. To support the many to many relationship between books / movies and checkout , the associative entities cust_movie and cust_books will be used.

Relationships

The relationships between these entities are defined as follows. For the first entity, the customer, it will have a one to many relationship with the checkout entity. An individual checkout transaction can only be performed by one customer, but an individual customer can have many checkout transactions. The checkout entity will have a many to many relationship with both the books and movies entities. A checkout transaction can contain multiple books and/or movies, as well as each book or movie being associated with multiple checkout transactions. The customer will not have any direct relationship with the movies or the books.

Constraints

Each entity attribute will have defined data types. For customers the library card number will be an auto-incrementing integer, the first and last names as well as phone number will be varchar(255). The primary key for the customer will be the library card number this will provide a

unique identifier for each customer. For the checkout entity, the transaction ID will be an auto-incrementing integer and also the primary key for that entity. It will also server as a foreign key between the checkout entity and the customer to allow tracking of checkout transaction to a particular customer. The due date will be will be stored as date type. For books and movies, the title will be varchar(255), checked out will be bool data type, and ID will be an auto-incrementing integer (also the primary key for both of those entities). Specifically for movies, the year of release will be stored as an int, and for books the author will be stored as varchar(255). Using these unique attributes the customer will be able to go select the movies and books they wish to checkout from a list that shows what is currently in stock. These items will be added to the checkout then will be assigned a due date and transaction id. After the customer finished the transaction, it will change the checked out attribute for the items that were borrowed from false to true, and will subsequently reflect this on additional searches.

Entities : *Attributes*

Books : *Title, Author, ID, Checked Out*

Check Out : *Transaction ID, Due Date*

Customer : *Library Card Number, First Name, Last Name, Phone Number*

Movies : *Title, Year, ID, Checked Out*

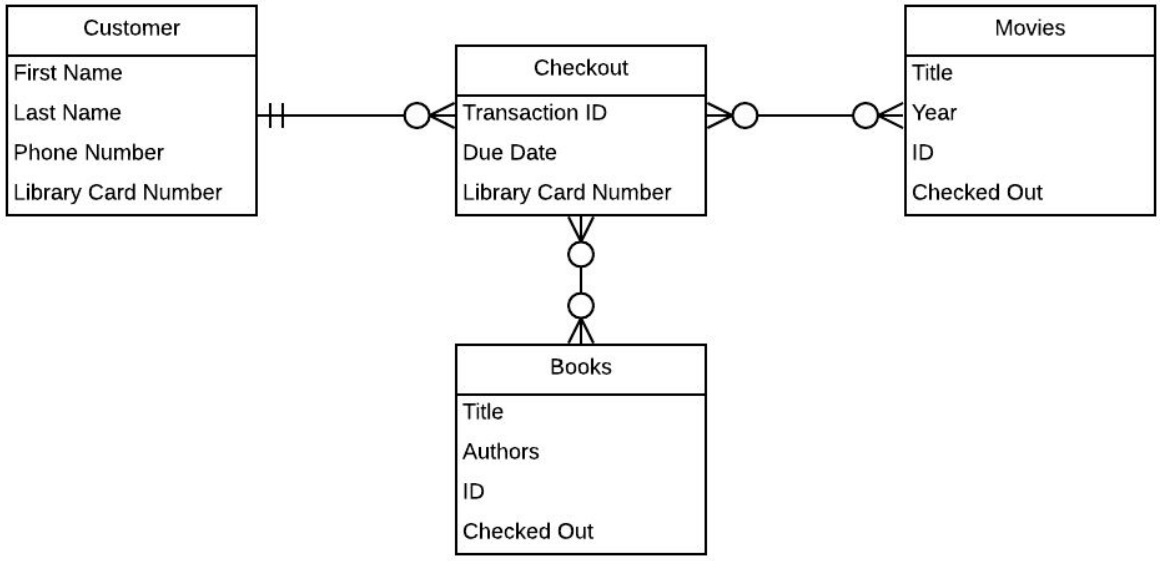
Constraints : *Attributes*

Primary key : *library card number, transaction ID, ID (books / movies)*

Not Null : *All attributes*

Foreign key : *transaction ID*

ER diagram



Schema: