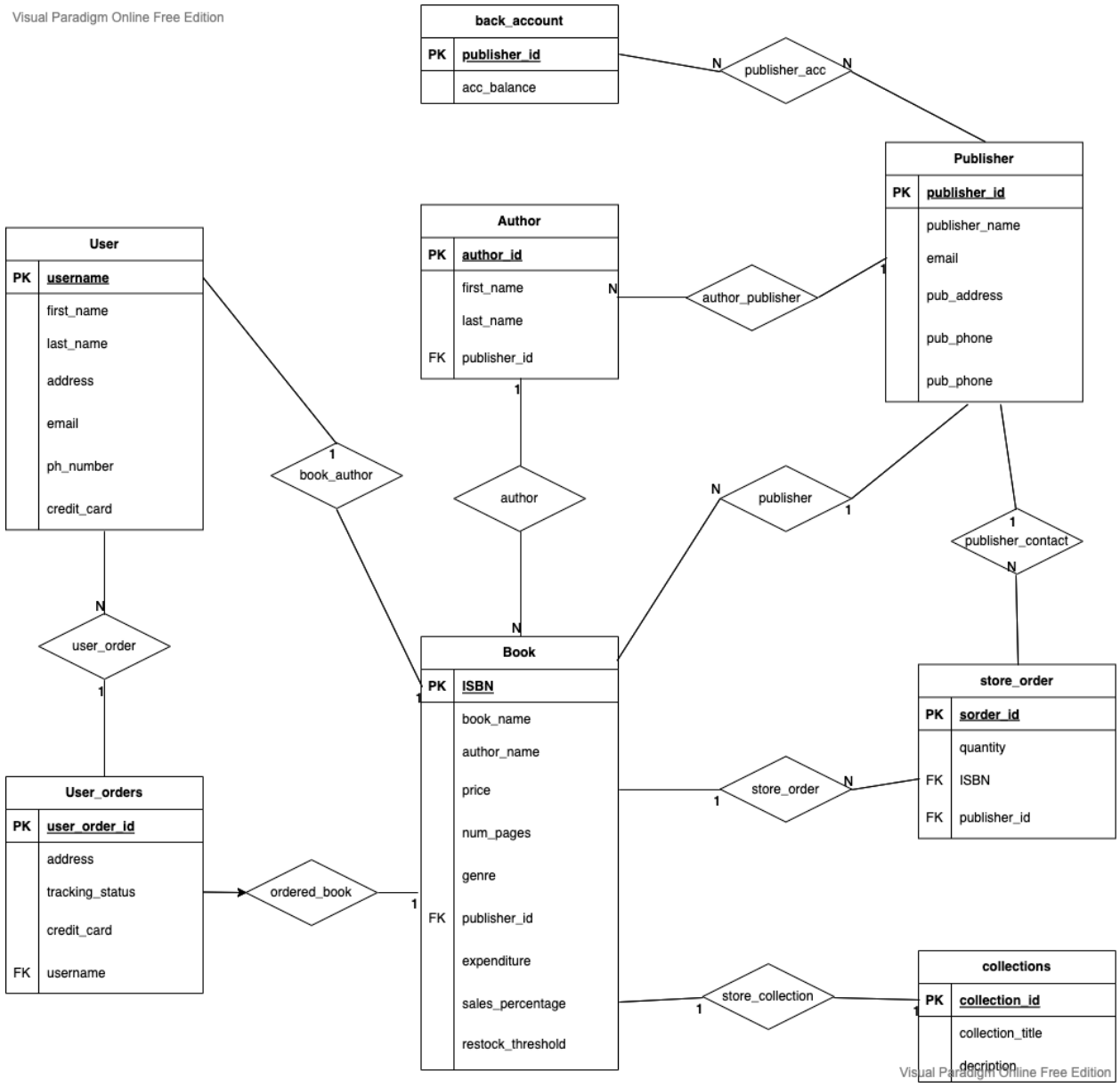
COMP 3005 Project Report

Name: Samarth Wachche
StudentID: 101210870
Date: 11th December 2022

2.1. Conceptual Design



Here i have used the UML diagram to represent the ER diagram for the online book store database software. UMl diagram also know as unified model diagram is a diagram used to unify system models in the form of a diagram. Here i have a Book entity which stores all the information for the book like ISBN, book name, author name for the book, price, pages, genre, publisher id of the book, expenditure for the book, sales percentage sent to the publisher, and the track of the stock. The book has one to one relation with collections as the book can in one collection only. And it has a many to one relation with author, as one author can have many books. I have a User entity which keeps tracks of the registered user and all information for the registered user like first_name, last_name, address, email, phone number, credit card information for the user. And in connection with the user we have a user_orders entity which keeps track of all the orders placed my the user/owner like user_order_id, address, tracking

status and the username. The author entity contains the author_id as the primary key, first_name, last_name and publisher_id as the foreign key. The publisher entity has the publisher_id as the primary key, publisher_name, email, publisher phone number, publisher address. The publisher and the author have the one to many cardinality i.e. the publisher can publish multiple authors books. The book and the publisher have the one to many cardinality i.e. one publisher can publish multiple books and a book can have only on publisher. The store_order cardinality has the store_id as the primary key , quantity of the books the store wants to order, ISBN and publisher_id as the foreign key. The store_order has the one to many cardinality with the books, as the store can order multiple books. The collection entity stores the collection_id(primary key), collection_title and the description. It has one to one cardinality with the book as the book can be a part of one collection only.

2.2. Reduction to Relation Schemas

User

username	first_name	last_name	address	email	ph_numbe r	credit_card

User_order

user_order_id	address	tracking_status	credit_card	username

Checkout

checkout_id	username	ISBN

User_order_books

user_ordered_id	ISBN	user_order_id

Store_order

sorder_id	quantity	ISBN	publisher_id

Book

ISBN	book_ name	author _name	price	num_p ages	genre	publish er_id	expen diture	sales_ percen tage	restock_thre shold

Author

author_id	first_name	last_name	publisher_id

Book_author

book_id	author_id

Publisher

publisher_id	publisher_name	email	publisher_address	publisher_phone

Bank_acc

publisher_id	acc_balance

Collections

collection_id	collection_title	description

Collection_books

collection_id	book_id

2.3. Normalization of Relation Schemas

Normalization is important as we dont want to see too many redundancies and null values in our database. This creates a risk of inconsist values, in order to reduce this risk we need to make sure we get rid of the repeating values, and to do this we sometimes we need to remove some of the entities and merge with the others. Hence, we use the normalization theory to make sure our database is free of the repeated values.

2.3.1. 1st Normal form:

The current relational schema is already in the 1st normal form, as the table to be in first normal there should be no repeating attributes in the table. Hence, there is no repeating attributes in the table the current relational form is in the 1st normal form.

2.3.2. 2nd Normal form:

For the table to be in the 2nd Normal form, the table should be in the 1st normal form, which it is so it can further be normalized in the 2nd form.

1. The book_author entity can be gotten ridd of as it is repeating the values book_id and author_id indirectly. Instead what can be done is have a author_id attribute in the collection_books entity as a foriegn key which is directly connecting to the Author entity as the author_id is the primary key.

Collection_books

collection_id	**author_id**

2. The same can be done with the bank_acc entity as it only holds the account balance, and the account balance can be included in the publisher entity itself.

Publisher

publisher_id	publisher_name	email	publisher_address	publisher_phone	**acc_balance**

3. The user_order_books store the same information as the checkout entity, so to reduce redundancy the the two entities can be merged and create one order_id to keep track of the orders, instead of having two id's one in the user_order_books and one in the user_orders.
   This can be done, by having a order_id in the checkout entity which will be a foriegn key, and this order_id will be the primary key in the user_orders entity and then again storing the ISBN in the checkout entity too. As ISBN in the checkout is a foreign key for book entity, we can reduce redundancy in the schema.

Checkout

checkout_id	username	ISBN	order_id

User_order

order_id	address	tracking_status	credit_card	username

2.3.3. 3rd Normal form:

Now that the table is in its 2nd Normal form, it can be further worked to convert into the 3rd Normal form.

User

username	first_name	last_name	address	email	ph_number	credit_card

User_order

order_id	address	tracking_status	credit_card	username

Checkout

checkout_id	username	ISBN	order_id

Store_order

sorder_id	quantity	ISBN	publisher_id

Book

ISBN	book_name	author_name	price	num_pages	genre	publisher_id	expenditure	sales_percentage	restock_threshold

Author

author_id	first_name	last_name	publisher_id

Book_author

book_id	author_id

Publisher

publisher_id	publisher_name	email	publisher_address	publisher_phone	acc_balance

Collections

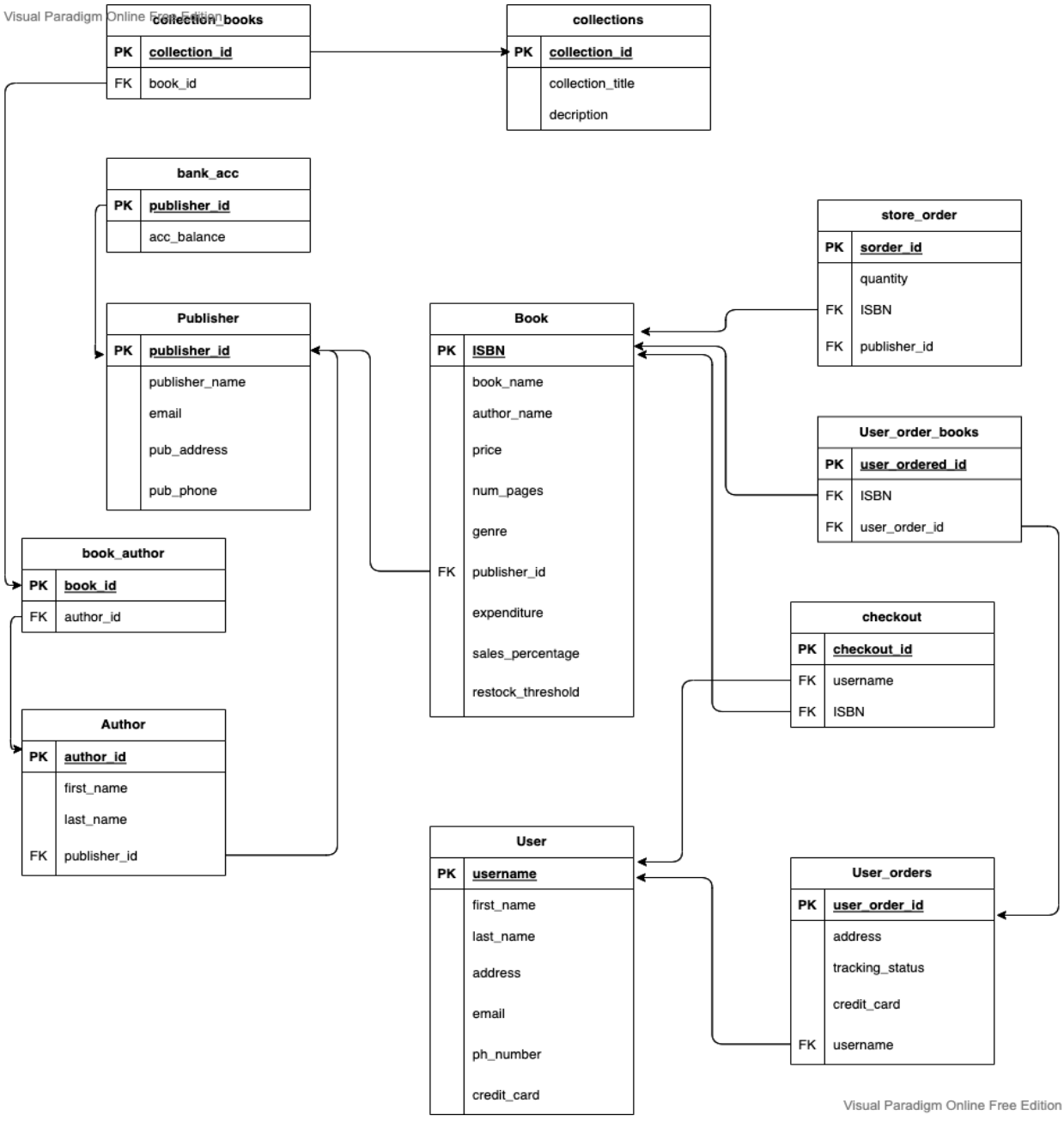
collection_id	collection_title	description

Collection_books

collection_id	author_id

2.4. Database Schema Diagram

This is the database schema before normalization where we can see that there are many repeated values in the tables and un-necesary entities.



After successfully normalizing the table and removing all the repeated and unwanted entities the database schema looks like this.