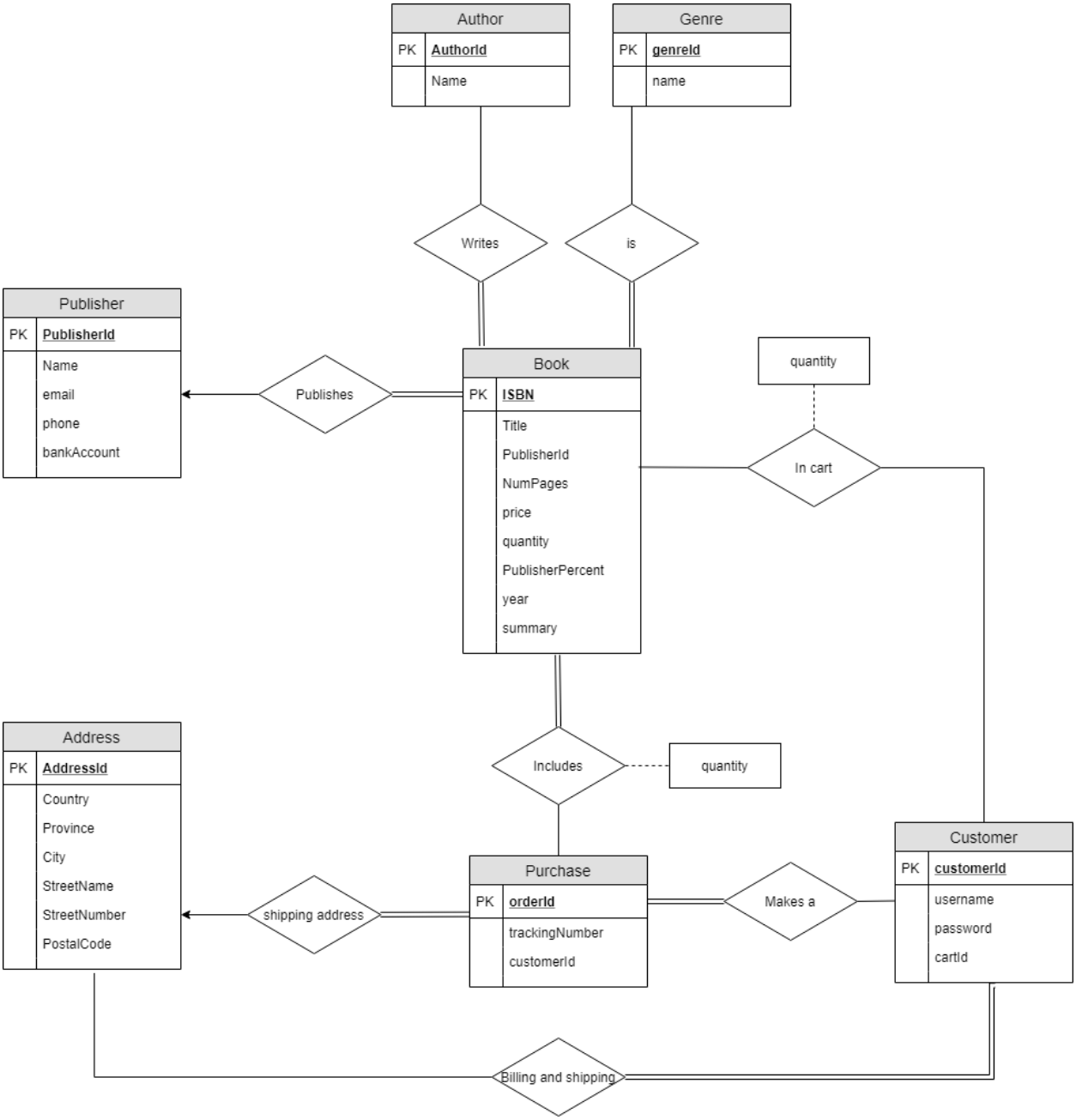
COMP 3005 Project Report

Shaelie Spallin
101064236

Contents

1. Conceptual Design (25%) 2
2. Reduction to Relation Schemas (20%) 3
3. Normalization of Relation Schemas (20%) 3
4. Database Schema Diagram (10%) 6
5. Implementation (25%) 6
6. Bonus Features (up to 15%) 6
7. GitHub Repo 6

1. Conceptual Design (25%)



Assumptions about cardinality:

1) A user will only have 1 cart. (draw.io would not allow me to require total participation and have an arrow for cardinality)
2) A cart will belong to only 1 user. (Same issue with draw.io not allowing an arrow with total participation)
3) A purchase will only have 1 shipping address.

Assumptions about participation type:

1) Authors, publishers, and genres can be added before a book linked to them is added. For example, we could add JK Rowling before adding any of the books she's written.
2) The bookstore only sells books, therefore each purchase must have 1..* books.
3) A user may ship a purchase to an address that is not associated with a customer.

2. Reduction to Relation Schemas (20%)

- author(auth_id, auth_name)
- genre(genre_id, genre_name)
- book(ISBN, title, pub_id, num_pages, price, quantity, pub_percent, year, summary)
- publisher(pub_id, pub_name, email, phone, bank_acct)
- address(addr_id, country, province, city, street_name, street_num, postal_code)
- purchase(order_id, tracking_num, cust_id, addr_id)
- customer(cust_id, username, password, cart_id)
- writes(auth_id, ISBN)
- book_genre(ISBN, genre_id)
- published(pub_id, ISBN)
- book_purchased(ISBN, order_id, quantity)
- customer_shipping(cust_id, addr_id)
- customer_billing(cust_id, addr_id)
- book_in_cart(cart_id, ISBN, quantity)

3. Normalization of Relation Schemas (20%)

- Author

  auth_id → auth_name
  auth_id+ = R
  Therefore, in BCNF

- Genre

  genre_id → genre_name genre_id+ = R
  genre_name → genre_id genre_name+ = R
  Therefore, in BCNF

- Book

  ISBN → title ISBN → quantity ISBN+ = R
  ISBN → pub_id ISBN → pub_percent
  ISBN → num_pages ISBN →year
  ISBN → price ISBN → summary

  summary → ISBN summary → price summary+ = R
  summary → title summary → quantity

summary → pub_id summary → year
summary → num_pages summary → pub_percent
Therefore, in BCNF

- Publisher

pub_id → pub_name pub_name → pub_id email → pub_id
pub_id → email pub_name → email email → pub_name
pub_id → phone pub_name → phone email → phone
pub_id → bank_acct pub_name → bank_acct email → bank_acct
pub_id+ = R pub_name+ = R email+ = R

phone → pub_id bank_acct → pub_id
phone → pub_name bank_acct → pub_name
phone → email bank_acct → email
phone → bank_acct bank_acct → phone
phone+ = R bank_acct+ = R
Therefore, in BCNF

- Address

addr_id → country addr_id → street_name
addr_id → province addr_id → street_num
addr_id → city addr_id → postal_code

postal_code → country BREAKS BCNF
postal_code → province BREAKS BCNF

To make BCNF:
Add a relation: postal_area(<u>postal_code</u>, province, country)
postal_code → province
postal_code → country
Address becomes: address(<u>addr_id</u>, city, street_name, street_num, postal_code)

- Purchase

order_id → tracking_num tracking_num → order_id
order_id → cust_id tracking_num → cust_id
order_id → addr_id tracking_num → addr_id
order_id+ = R tracking_num+ = R
Therefore, in BCNF

- Customer

cust_id → username username → cust_id
cust_id → password username → password
cust_id → cart_id username → cart_id
cust_id+ = R username+ = R
Therefore, in BCNF

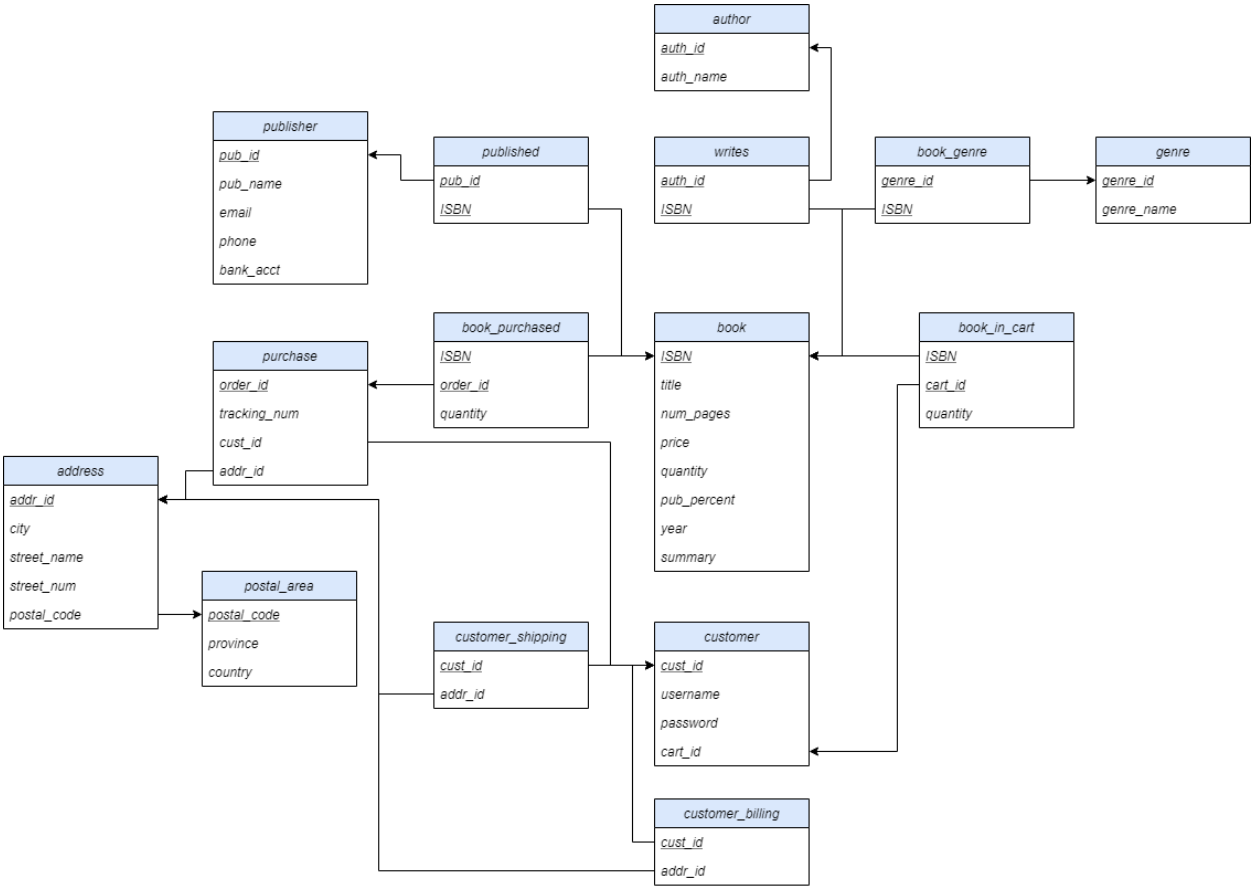
- Writes

auth_id → ISBN ISBN → auth_id
auth_id+ = R ISBN+ = R

Therefore, in BCNF

- Book_genre
  - ISBN → genre_id
  - ISBN+ = R
  - Therefore, in BCNF
- Published
  - ISBN → pub_id
  - ISBN+ = R
  - Therefore, in BCNF
- Book_purchased
  - order_id → ISBN
  - order_id → quantity
  - order_id+ = R
- Customer_shipping
  - cust_id → addr_id
  - cust_id+ = R
  - Therefore, in BCNF
- Customer_billing
  - cust_id → addr_id
  - cust_id+ = R
  - Therefore, in BCNF
- Book_in_cart
  - No FDs break it, so it's in BCNF

4. Database Schema Diagram (10%)



5. Implementation (25%)

I split the code into 2 files, one of which just houses queries in python functions and the other handles all of the UI, and makes calls to the query python functions as needed.

6. Bonus Features (up to 15%)

1) Searches by titles similar to what the user typed in for searching by title
2) Searches by authors with similar names to what the user input for searching by author
3) Searches by genres similar to what the user typed in for searching by genre

I don't believe there are any other bonus features, but I can't remember.

7. GitHub Repo

https://github.com/shaejsp/3005Project