COMP 3005 Winter 2020 Project

Sharjeel Ali

March 28th, 2020

# 1. Conceptual Design

# 2. Reduction to Relation Schemas

Relations:

1. *author(id, name)*
2. *genre(id, name, type)*
3. *publisher(id, phone, email, address\_id, dd\_id)*
4. *book(isbn, title, description, price, page\_count, published\_date, add\_date, stock, rating, rating\_count, sale\_percent, author\_id, genre\_id, publisher\_id)*
5. *orders(order\_number, username, order\_date, status\_id, card\_number)*
6. *status(status\_id, name, description)*
7. *order\_book(order\_number, isbn, warehouse\_id, quantity)*
8. *cart(username, isbn, quantity)*
9. *client(username, email, first\_name, last\_name, password)*
10. *admin(email, first\_name, last\_name, password)*
11. *sales\_report(report\_number, admin\_email, start\_date, end\_date, file)*
12. *report\_transaction(report\_number, transaction\_id)*
13. *transaction(transaction\_id, transaction\_name, transaction\_type, amount, date)*
14. *warehouse(id, address\_id)*
15. *warehouse\_books(warehouse\_id, isbn, stock)*
16. *address(id, country, state, city, code, street, apartment\_number)*
17. *client\_address(username, address\_id)*
18. *direct\_deposit(dd\_id, transit\_number, account\_number, institution\_number)*
19. *payment\_info(card\_number, name, expiry\_date, security\_code)*
20. *client\_billing(username, card\_number)*
21. *restock(restock\_number, isbn, warehouse\_id, quantity, restock\_date)*
22. *request\_book(request\_number, username, status, request\_name, request\_isbn, date)*
23. *admin\_decides(email, request\_number, decision)*
24. *history(username, isbn, rank)*

test

# 3. Normalization of Relation Schemas

For this section, before I reduce, and normalize my relations, the following relations from section 2 above cannot be normalized any further due to them containing only two attributes, where the primary key can determine the other attribute. However, some of the following do not determine each other, but this is still normalized due to the following rule:

*For every set of attributes a ⊆ Ri , check that a+ (the attribute closure of a) either includes no attribute of Ri - a, or includes all attributes of R­i*

Therefore, if the relation does not include the attribute from Ri-a (Which in the following cases is just the other attribute itself), it is normalized:

1. *author(id, name)*
2. *cart(username, isbn, quantity)*
3. *report\_transaction(report\_number, transaction\_id)*
4. *warehouse(id, address\_id)*
5. *client\_address(username, address\_id)*
6. *client\_billing(username, card\_number)*

Ddd

Mention discussion with email and username and how it is normalized for admin and client. Assumption that admin emails are constant and cannot be changed, but client ones can.

# 4. Database Schema Diagram

# 5. Implementation

# 6. Bonus Features

# 7. Github Repository

<https://github.com/SharjeelAliBCS/comp3005W20-project>