COMP 3005 Project Report

Shaelie Spallin

101064236

Contents

[1. Conceptual Design (25%) 1](#_Toc37342696)

[2. Reduction to Relation Schemas (20%) 1](#_Toc37342697)

[3. Normalization of Relation Schemas (20%) 2](#_Toc37342698)

[4. Database Schema Diagram (10%) 2](#_Toc37342699)

[5. Implementation (25%) 2](#_Toc37342700)

[6. Bonus Features (up to 15%) 2](#_Toc37342701)

[7. GitHub Repo 2](#_Toc37342702)

# Conceptual Design (25%)

A close up of a map

Description automatically generated

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.

# Reduction to Relation Schemas (20%)

* author(a\_id, a\_name)
* genre(g\_id, g\_name)
* book(ISBN, title, p\_id, num\_pages, price, quantity, pub\_percent, year, summary)
* publisher(p\_id, p\_name, email, phone)
* address(addr\_id, country, province, city, street\_name, street\_num, postal\_code)
* purchase(o\_id, tracking\_num, c\_id)
* customer(c\_id, username, password)
* cart(cart\_id, c\_id)
* writes(a\_id, ISBN)
* book\_genre(ISBN, g\_id)
* published(p\_id, ISBN)
* publisher\_addr(p\_id, addr\_id)
* book\_purchased(ISBN, o\_id, quantity)
* customer\_shipping(c\_id, addr\_id)
* customer\_billing(c\_id, addr\_id)
* book\_in\_cart(cart\_id, ISBN, quantity)
* ships\_to(addr\_id, o\_id)

# Normalization of Relation Schemas (20%)

* Author

a\_id → a\_name a\_id+ = R

Therefore, in BCNF

* Genre

g\_id → g\_name g\_id+ = R  
g\_name → g\_id g\_name+ = R

Therefore, in BCNF

* Book

ISBN → title ISBN → quantity ISBN+ = R  
ISBN → p\_id ISBN → pub\_percent  
ISBN → num\_pages ISBN →year  
ISBN → price ISBN → summary

summary → ISBN summary → price summary+ = R

summary → title summary → quantity

summary → p\_id summary → year

summary → num\_pages summary → pub\_percent

Therefore, in BCNF

* Publisher

p\_id → p\_name p\_name → p\_id email → p\_id phone → p\_id  
p\_id → email p\_name → email email → p\_name phone → p\_name  
p\_id → phone p\_name → phone email → phone phone → email

p\_id+ = R p\_name+ = R email+ = R phone+ = 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: Province(postal\_code, province, country)

postal\_code → province

postal\_code → country

Address becomes: Address(addr\_id, city, street\_name, street\_num, postal\_code)

* Purchase

o\_id → tracking\_num tracking\_num → o\_id

o\_id → c\_id tracking\_num → c\_id

o\_id+ = R tracking\_num+ = R

Therefore, in BCNF

* Customer

c\_id → username username → c\_id

c\_id → password username → password

c\_id+ = R username+ = R

Therefore, in BCNF

* Cart

c\_id → cart\_id cart\_id → c\_id

c\_id+ = R cart\_id+ = R

Therefore, in BCNF

* Writes

a\_id → ISBN ISBN → a\_id

a\_id+ = R ISBN+ = R

Therefore, in BCNF

* Book\_genre

ISBN → g\_id

ISBN+ = R

Therefore, in BCNF

* Published

ISBN → p\_id

ISBN+ = R

Therefore, in BCNF

* Publisher\_addr

p\_id → addr\_id addr\_id → p\_id

p\_id+ = R addr\_id+ = R

* Book\_purchased

o\_id → ISBN

o\_id → quantity

o\_id+ = R

* Customer\_shipping

c\_id → addr\_id

c\_id+ = R

Therefore, in BCNF

* Customer\_billing

c\_id → addr\_id

c\_id+ = R

Therefore, in BCNF

* Book\_in\_cart

No FDs break it, so it’s in BCNF

* Ships\_to

o\_id → addr\_id

o\_id+ = R

Therefore, in BCNF

# Database Schema Diagram (10%)

[final schema diagram]

# Implementation (25%)

[describe architecture, what modules there are and how they interact]

[architectural diagram, optional but encouraged]

# Bonus Features (up to 15%)

[list any bonus features implemented]

# GitHub Repo

[url to repo]