DNA Project Submission 1: Database Design and Functional Requirements

Sushant Kumar, Aryan Mishra , Manas Mittal , Karthik Malavathula / Group 26

October 24, 2024

1 Abstract

We aim to develop a B2C application for an online bookstore. The database will maintain a large catalog of books, allowing customers to browse and purchase books after authentication via login. Additionally, we facilitate the retailer (also the admin of the database) to add/delete/update books in the dB, view trending books, orders placed by a customer, customer details. The database stores customer details, accounts, transaction history, book details, authors, publishers and shipping details. Future enhancements may include personalized bookshelves for customers, order tracking, search boost requests by authors, trend-based recommendations.

2 Mini-world Description

The mini-world of this project revolves around an online bookstore that serves both customers and retailers. The system's primary users include:

- **Customers:** Can login, browse books, make purchases, and access personalized offers and preferred status.
- **Publishers:** Manage book stock, track sales, and engage in monetary agreements.
- **Authors:** Register, track sales performance, and boost their visibility based on credits.
- **Shipping Agencies:** Handle shipping transactions and fees.
- **Retail Analysts:** Analyze customer reading patterns and trends.

3 Database Requirements

The following are the core data requirements for the online bookstore:

3.1 Entity Types

1. **Customer** (user_id INT PRIMARY KEY AUTO_INCREMENT,
username VARCHAR(20) NOT NULL,
password VARCHAR(20) NOT NULL,
first_name VARCHAR(20) NOT NULL,
last_name VARCHAR(20) NOT NULL,
address **composite** - house_num INT NOT NULL, street VARCHAR(20) NOT NULL, city VARCHAR(20) NOT NULL, country VARCHAR(20) NOT NULL,
age INT NOT NULL **derived**,
gender VARCHAR(20) NOT NULL,
login_status VARCHAR(20) NOT NULL,
wallet DECIMAL(10,2),
customer_status VARCHAR(20) NOT NULL,
customer_phone **multivalued** INT)

2. **Book** (ISBN INT PRIMARY KEY NOT NULL,
type VARCHAR(20) NOT NULL,
status VARCHAR(20) NOT NULL,
availability VARCHAR(20) NOT NULL,
edition INT NOT NULL,
language VARCHAR(20) NOT NULL,
title VARCHAR(20) NOT NULL,
rating VARCHAR(20) NOT NULL)

3. **Publisher** (company_name VARCHAR(20) PRIMARY KEY NOT NULL,
warehouse_zip INT NOT NULL,
agrrement_terms VARCHAR(20) NOT NULL)

4. **History** (transaction_id INT PRIMARY KEY AUTO_INCREMENT)

5. **Author** (login_id VARCHAR(20) PRIMARY KEY NOT NULL,
password VARCHAR(20) NOT NULL,
name VARCHAR(20) NOT NULL,
status VARCHAR(20) NOT NULL)

6. **Accounts (entity with PK consisting of 2 attributes)**
(type VARCHAR(20) NOT NULL,
date DATE NOT NULL,
amount INT NOT NULL,
PRIMARY KEY (type, date))

7. **Shipper** (shipper_id INT PRIMARY KEY NOT NULL,
live_location VARCHAR(20) NOT NULL,
date_of_delivery DATE NOT NULL)

8. **Offers** (offer_id INT PRIMARY KEY NOT NULL,
description VARCHAR(20) NOT NULL)

3.2 Weak Entities

1. **Cart** (**dependent on customer**)
(date DATE,
time TIME,
num_items INT,
total_amount DECIMAL(10,2),
DISCRIMINANT (date, time))

2. **Reviews** (**dependent on book**)
(date DATE,
time TIME,
description VARCHAR(100),
DISCRIMINANT (date, time))

3.3 Relationships

1. **has_bucket**: Between Customer and Cart (Identifying relationship). (One-to-many And total participation of cart)
2. **order_saved**: Between Cart and History. (One-to-one And total participation of History)
3. **views**: Between Customer and History. (One-to-many And total participation of History)
4. **pay**: Between Customer and Accounts. (One-to-Many)
5. **track**: Between Customer and Shipper. (One-to-one And total participation of both)
6. **order_ships**: Between Shipper and History. (One-to-one And total participation of Shipper)
7. **commission**: Between Shipper and Accounts. (One-to-many)
8. **order_ships**: Between Shipper and History. (One-to-one And total participation of Shipper)
9. **order_saved**: Between Cart and History. (One-to-one And total participation of History)
10. **refund**: Between Accounts and History. (Many-to-one)
11. **publisher_share**: Between Publisher and Accounts. (One-to-many)
12. **apply**: Between Customer and Offers. (Many-to-many)
13. **search_view_rate**: Between Customer and Book. (Many-to-many)
14. **contains**: Between Cart and Book. (One-to-many)
15. **adds**: Between Publisher and Book. (One-to-many)
16. **boost_request**: Between Author and Book. (One-to-many)
17. **royalty (ternary (deg 3) relationship)**: Between Author, Book and Accounts. (1, N, 1)

18. **gifts (recursive relation)**: Between Customer (sender) and Customer (receiver). (Many-to-many)

3.4 Special Constraints In addition to the ones already listed

- **Composite Attributes:** "Address" for customers and publishers.
- **Multi-valued Attributes:** "type" for books.
- **Derived Attributes:** "Customer Status" based on purchase history.
- **Subclass:** "Preferred Customer", "General Customer" as a subclass of "Customer".

4 Functional Requirements

4.1 Retrieval Operations

1. **Selection:** Retrieve all customers who are from India
2. **Projection:** title, price from book given ISBN number, description of offers for a particular customer (given customer id using INNER JOIN)
3. **Aggregate Function:** average total amount spent by males and females in their latest purchase provided their average num_items bought were > 2.
4. **Search:** Search for books containing the keyword "Book Title" in their title.

4.2 Analysis Reports

1. **Report 1:** Trending searches for books based on day, month, year; Customer details given age limit, gender, country etc.
2. **Report 2:** transaction details, books purchased, amount spent by a particular customers by performing multiple INNER JOINs

4.3 Modification Operations

1. **Insert:** Adding a new book to the inventory, adding new items in the cart.
2. **Update:** Updating customer wallet, book ratings.
3. **Delete:** Remove a book from the catalog.

5 Possible Upgradations

Future enhancements may include:

- A personalized bookshelf for Kindle customers to maintain reading logs and wishlists.

- A data analysis tool for retail analysts to recommend books based on customer demographics (age, gender, location).

6 Contribution

- Sushant Kumar
- Aryan Mishra
- Manas Mittal
- Karthik Malavathula