

WORKSHOP 2



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Members:

Ichel Delgado Morales – 20202020029

Andrés Felipe Salazar Malagón – 20202020043

Professor:

Carlos Andrés Sierra Virguez

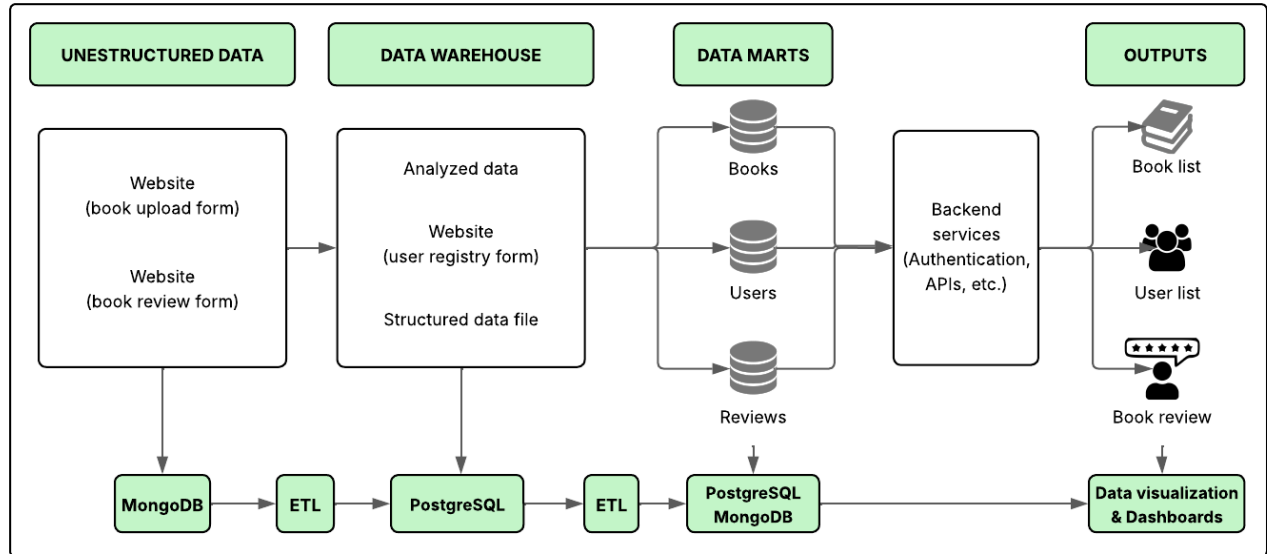
Universidad Distrital Francisco José de Caldas

Faculty of Engineering

Databases 2

Bogotá, 2025

Database Architecture



In the following link you can see the Architecture diagram: [Architecture](#)

1. Initial Data Capture (Unstructured Data Sources)

Website (book upload form) / Website (book review form): These are the user interfaces where data is initially entered. The "book upload" form allows users to upload information about books, which can include details such as title, author, cover, description, etc. The "book review" form allows users to write and submit reviews of books. This data is typically unstructured or semi-structured, so MongoDB will be used because of its flexible with schemas, which is ideal for data that can vary in structure, such as book reviews or initial book information that has not yet been fully processed or validated.

2. Structured Processing and Storage (Data Warehouse)

The ETL process converts unstructured or semi-structured data into a structured format suitable for analysis. The **Data Warehouse**, built using **PostgreSQL**, stores this data due to its strong relational capabilities and support for complex SQL queries.

Key structured data sources include:

- **Analyzed data:** Already processed information ready for advanced querying.
- **Website forms:** Structured input like user registrations (name, email, password).
- **Structured files:** Data imported from CSVs, spreadsheets, or other external structured formats.

3. Data Segmentation (Data Marts)

A second ETL process extracts data from the main PostgreSQL warehouse, transforms it for specific purposes, and loads it into data marts—smaller, focused databases optimized for quick access and specific analyses.

- **Books (MongoDB):** Book data is highly variable (e.g., multiple authors, editions, translators). MongoDB handles this diversity well through flexible JSON documents, avoiding the rigid schema and NULLs typical in relational databases.
- **Users (PostgreSQL):** User data (ID, email, roles, etc.) is structured and relational. PostgreSQL is ideal for this due to its strong ACID guarantees, support for complex relationships, and robust security for sensitive information.
- **Reviews (MongoDB):** Reviews vary in length and structure (text, ratings, tags). MongoDB efficiently stores this unstructured or semi-structured feedback at scale, making it ideal for large volumes of user-generated content.

Information Requirements

Information Type	Description
Book and Audiobook Search	Retrieve books based on filters such as title, author, genre, or popularity.
User Profile Data	Fetch user details including name, profile picture, subscription status, and preferences.
Favorites and Reading Lists	Load a user's saved books, favorite titles, and personalized reading lists.
User Reviews and Ratings	Retrieve reviews and star ratings submitted by users for a specific book or audiobook.
Followed User Activity	Show public lists, favorites, or reviews from users the person follows.
Recommendations	Retrieve book or audiobook suggestions based on reading history, social activity, and preferences.
Subscription status	Check the user's current subscription plan and access level (e.g., Premium).
Administrative Content List	Fetch books uploaded by administrators or creators, including metadata for review or editing.
Audit and Moderation Logs	Access logs of content flagged, edited, or removed for moderation purposes.
Analytics and Reports	(Future) Generate statistics on book popularity, user engagement, and feature usage.

Query Proposals

1. **Requirement FR-1.1 (SQL):** The system must allow book and audiobook searching by title, author, or genre using a search engine with combinable filters.

```
SELECT b.title, a.name AS author, g.description AS genre
FROM book b
JOIN author a ON b.author = a.author_id
JOIN genre_book gb ON b.book_id = gb.book_id
JOIN genre_type g ON gb.genre_type_id = g.genre_type_id
WHERE b.title ILIKE '%magic%'
      OR a.name ILIKE '%rowling%'
      OR g.description ILIKE '%fantasy%';
```

The purpose of this query is to allow users to search by keywords across multiple fields with flexible filters. This would help users to quickly find books or audiobooks of interest, improving discoverability and user experience.

2. **Requirement FR-2.2 (NoSQL):** The system should allow users to write reviews associated with each title and allow editing and deletion if necessary.

```
book_platform.review.insertOne({
  UUID(),
  user_id: "New Horizons",
  book_id: "0304fef4-5e2e-4e1f-9563-f3274dc1fd9b",
  rating: 5,
  comment: " This book was amazing – highly recommended!",
  createdAt: new Date(),
});
```

The purpose of this query is to allow a user to submit a review and rating for a specific book, and it contributes to the collective evaluation of content and improves recommendation quality.

3. **Requirement FR-3.1 (SQL):** The system must allow the creation and editing of a user profile with a photo, display name, and short description.

```
INSERT INTO user (user_id, password, name, email, profile_picture,
is_premium, user_type, created_at)
VALUES (
  gen_random_uuid(),
  '$2b$10$7nDfF3uH1qAn2AymzDqXz.LwPSlYVLz6obGZuhmT.yzGU6/10hzHq',
  'Jane Doe',
  'jane@example.com',
  'https://cdn.app/pics/jane.jpg',
  false,
```

```

        (SELECT user_type_id FROM user_type WHERE description = 'Reader'),
        CURRENT_TIMESTAMP
    );

```

The purpose of this query is to register a new user and create their personalized profile. It enables users to have an account and other user-specific features like reading lists, follows, and recommendations.

- 4. Requirement FR-4.1 (NoSQL):** The system must allow profiles with the Administrator or Creator role to upload book and audiobook files with their metadata.

```

book_platform.book.insertOne({
    UUID(),
    title: "New Horizons",
    author: "9b57fe43-a03f-4a15-8ce5-8e5f7b01b341",
    genres: "57c1488f-d781-4eac-9d6f-7e750675fabf ",
    synopsis: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vestibulum nibh quis turpis feugiat dictum. Etiam eget faucibus sem. Sed non fringilla odio, in sollicitudin risus. Maecenas in neque ut odio iaculis vestibulum at eu tellus. Etiam tempus tincidunt dolor sed euismod. Integer dapibus fringilla ultricies. Sed risus sapien, semper viverra urna vitae, consectetur varius nunc.",
    coverImage: "https://cdn.app/pics/jane.jpg",
    language: "English",
    IsAudiobook: true,
    fileUrl: https://cdn.app/pics/jane.jpg,
    createdAt: new Date(),
    updatedAt: new Date(),
    uploadedBy: "bad0a5df-257b-409c-95a4-f7cc52e5a65d"
});

```

The query adds a new book to the catalog with basic metadata, linked to the creator's user ID. This expands available content and supports the creator-admin workflow for content management.

- 5. Requirement FR-6.1 (SQL):** The system must allow users to purchase a Premium plan with payment management and access control to exclusive content.

```

INSERT INTO subscription (subscription_id, user_id, plan_type, start_date,
end_date, payment_status)
VALUES (
    gen_random_uuid(),
    'bad0a5df-257b-409c-95a4-f7cc52e5a65d',
    (SELECT plan_type_id FROM plan_type WHERE name = 'Premium'),
    CURRENT_DATE,
    CURRENT_DATE + INTERVAL '30 days',

```

```
(SELECT payment_status_type_id FROM payment_status_type WHERE  
description = 'Paid')  
);
```

The query registers a new subscription for a user, specifying the plan and payment status. It also controls access to paid content and helps analyze revenue and user upgrade trends.

```
SELECT plan_type, start_date, end_date, payment_status  
FROM subscription s JOIN plan_type p ON s.plan_type = p.plan_type_id  
JOIN payment_status_type pst ON s.payment_status =  
pst.payment_status_type_id  
WHERE s.user_id = 'current-user-id'  
AND CURRENT_DATE BETWEEN s.start_date AND s.end_date;
```

This query verifies if a user currently has an active Premium subscription and controls access to exclusive content and features based on the user's subscription status.

- **SQL + NoSQL Strategy: Complementary Use**

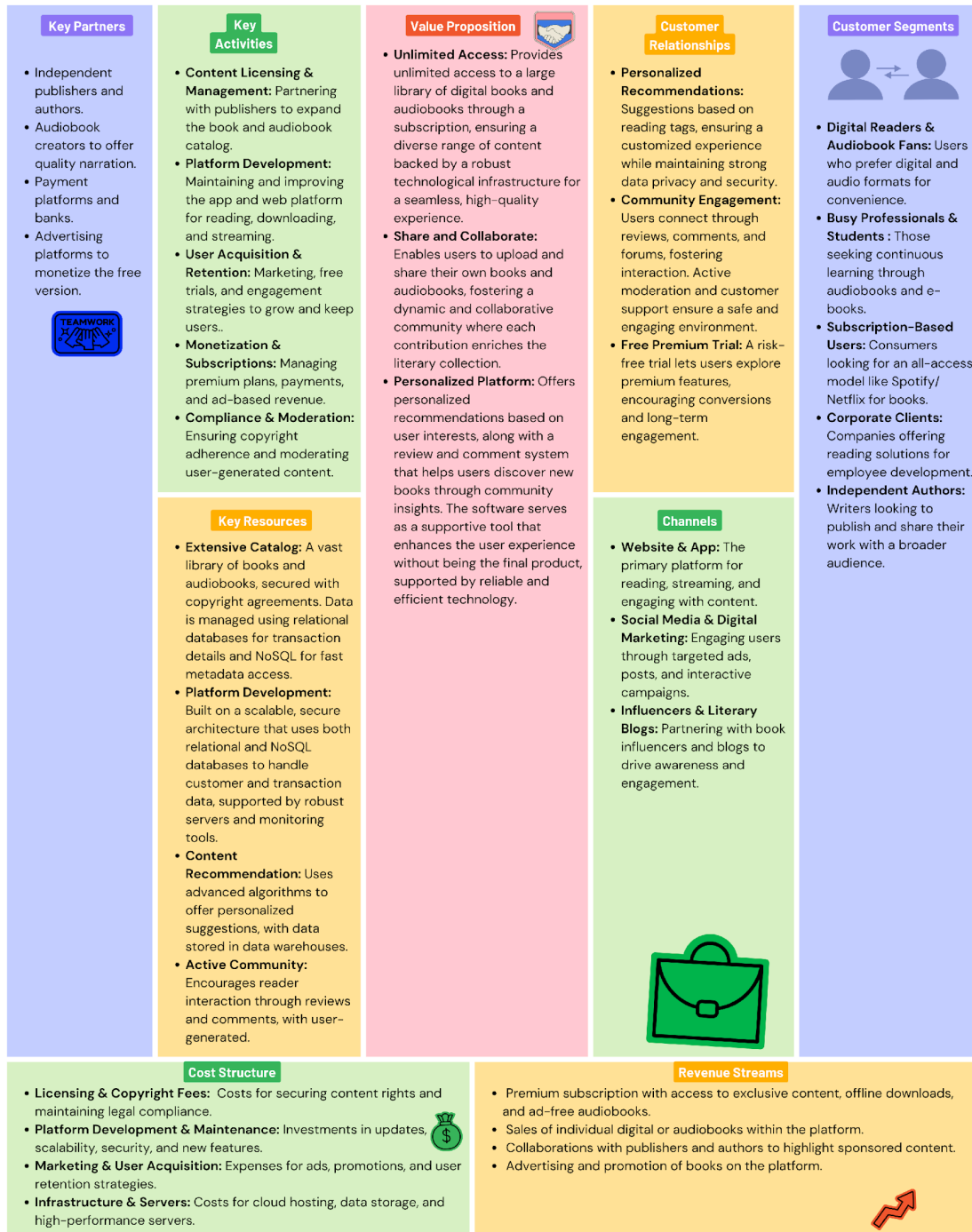
PostgreSQL is used to manage structured, relational data that requires consistency and integrity. Its support for SQL, constraints, and strong ACID properties makes it ideal for securely handling user information, subscriptions, and other well-defined entities.

MongoDB is chosen for storing flexible and variable data structures, such as books and reviews. As a document-oriented NoSQL database, it handles nested and inconsistent data efficiently, allowing rapid development and scalability without enforcing a rigid schema.

Business Model

Reading platform based on Bookmate

Andres Felipe Salazar Malagon - 20202020043
Ichel Delgado Morales - 20202020029



In the following link you can see the Canva Model diagram: [Business model - Bookmate](#)

Functional Requirement

1. Audiobooks and reading module:

Code	Requirement
FR-1.1	The system must allow book and audiobook searching by title, author, or genre using a search engine with combinable filters.
FR-1.2	The system must offer an online reading viewer that loads the text without prior download and quick page turning.
FR-1.3	The system must enable pause, fast-forward, and rewind controls for streaming audiobook playback, ensuring seamless continuity.
FR-1.4	The system must allow downloading complete books and audiobooks for offline reading and listening, storing them locally with space management.
FR-1.5	The reading viewer should allow users to adjust the font size and background theme (light/dark, sepia, etc.) to improve the reading experience.

2. Reviews, favorites and comments:

Code	Requirement
FR-2.1	The system should allow users to rate books and audiobooks using a star system (1-5).
FR-2.2	The system should allow users to write reviews associated with each title and allow editing and deletion if necessary.
FR-2.3	The system must display the list of existing reviews for the user to see and decide whether to read or listen to a title.
FR-2.4	The system must allow titles to be marked as favorites and display a personal list of favorites accessible from the profile.

3. Community and profiles:

Code	Requirement
FR-3.1	The system must allow the creation and editing of a user profile with a photo, display name, and short description.
FR-3.2	The system must allow users to follow other users and view their public activity (reviews and lists).

FR-3.3	The system should display the favorites of followed users to facilitate discovery.
FR-3.4	The system must generate notifications when one user is followed by another.
FR-3.5	The system must maintain a list of followers and followed for each user and show follow/unfollow actions.

4. **Books management and uploading:**

Code	Requirement
FR-4.1	The system must allow profiles with the Administrator or Creator role to upload book and audiobook files with their metadata.
FR-4.2	The system must allow Administrators or Creators to edit metadata (cover, synopsis, author, tags) of any title they have uploaded.
FR-4.3	The system shall allow an Administrator to remove content that violates rules and record the action in an audit log.

5. **Recommendations and discoveries:**

Code	Requirement
FR-5.1	The system should generate personalized recommendations based on the user's reading history and ratings.
FR-5.2	The system should suggest books read by friends (followed users) that the user has not yet read.
FR-5.3	The system should offer curated thematic collections (by the community or administrators) and navigable from a “Discover” section.

6. **Monetization and subscriptions:**

Code	Requirement
FR-6.1	The system must allow users to purchase a Premium plan with payment management and access control to exclusive content.
FR-6.2	For Premium users, the system must enable unrestricted downloading of books and audiobooks.
FR-6.3	For Premium users, the system must play audiobooks without ads or promotional interruptions.

User Stories

1. Reviews, favorites and comments

Title: Book search	Priority: High
User Story: As a user, I want to search for books and audiobooks by their title, author or genre to find content of my interest.	
Acceptance Criteria: The search function should return books and audiobooks that match the parameters sent by the user.	

Title: Online reading	Priority: High
User Story: As a user, I want to be able to read books online without needing to download them.	
Acceptance Criteria: The site should load the content of the book the user selected.	

Title: Audiobook streaming	Priority: High
User Story: As a user, I want to be able to stream audiobooks without interruptions.	
Acceptance Criteria: The site should allow audiobook streaming	

Title: Book download	Priority: Medium
User Story: As a user, I want to be able to download books or audiobooks to read or listen to offline.	
Acceptance Criteria: The downloading function should return books or audiobooks files downloaded.	

Title: Reading experience	Priority: Medium
User Story: As a user, I want to be able to adjust the font size and background to enhance my reading experience.	
Acceptance Criteria: The site should display the personalization features set by the user.	

2. Community and profiles

Title: Rating creation	Priority: High
User Story: As a user, I want to be able to rate books and audiobooks with stars to share my opinion with the community.	
Acceptance Criteria: The application should save and display the ratings made the user.	

Title: Review creation	Priority: High
User Story: As a user, I want to be able to write reviews to recommend (or not) a book to other readers.	
Acceptance Criteria: The application should save reviews made by the user.	

Title: Review reading	Priority: High
User Story: As a user, I want to be able to read reviews from other readers before deciding if I want to read a book or not.	
Acceptance Criteria: The application should display reviews made by other users to see.	

Title: Favorite list	Priority: High
User Story: As a user, I want to be able to save books and audiobooks on my favorites list so I can access them easily.	
Acceptance Criteria: The application should let users create favorite lists and add books and audiobooks to it.	

3. Books management and uploading

Title: Profile creation	Priority: High
User Story: As a user, I want to be able to create a personal profile with my picture, name and a short description to personalize my account.	
Acceptance Criteria: The application should receive the picture, name and short description, save it and show it to the user.	

Title: Following	Priority: Medium
User Story: As a user, I want to be able to follow other readers to see their recommendations and reading lists.	
Acceptance Criteria: The application should add and save followers of any reader and show them to the users.	

Title: Favorite list of followers	Priority: Low
User Story: As a user, I want to be able to see the favorite books and audiobooks of the users I follow to discover new titles.	
Acceptance Criteria: The application should return the list of favorites of the users any other user follows.	

Title: Following notification	Priority: Medium
User Story: As a user, I want to receive notifications when someone follows me.	
Acceptance Criteria: The application should notify any user when another user follows them.	

Title: Follow users	Priority: Medium
User Story: As a user, I want to be able to follow and be followed by other users to connect with other readers.	
Acceptance Criteria: The application should update the list of followers and following of the users.	

4. Recommendations and discoveries

Title: Book uploading	Priority: High
User Story: As an administrator or creator, I want to be able to upload books and audiobooks to share with the community.	
Acceptance Criteria: Only users with administrator or creator roles should be able to upload content on the site.	

Title: Book details editing	Priority: High
User Story: As an administrator or creator, I want to be able to edit the details of the books (cover, synopsis, author and tags).	
Acceptance Criteria: Only users with administrator or creator roles should be able to edit information related to books and audiobooks.	

Title: Content deletion	Priority: High
User Story: As an administrator, I want to be able to delete content that infringes the norms to assure the quality of the platform.	

Acceptance Criteria: Only users with administrator or creator roles should be able to review and delete content.

5. Monetization and subscriptions

Title: Recommendations based on own readings

Priority: Medium

User Story: As a user, I want to receive recommendations of books and audiobooks based on my readings and reviews.

Acceptance Criteria: The application should show recommendations based on the interests of the user.

Title: Recommendations based on friends readings

Priority: Medium

User Story: As a user, I want the platform to suggest new books based on what my friends are reading.

Acceptance Criteria: The application should show recommendations based on the interests of the user's following list.

Title: Exclusive content for premium users

Priority: High

User Story: As a premium user, I want to be able to access a premium plan to unlock exclusive content and advanced functionalities.

Acceptance Criteria: The system should set privileges to users registered as premium.

Title: Downloads for premium users

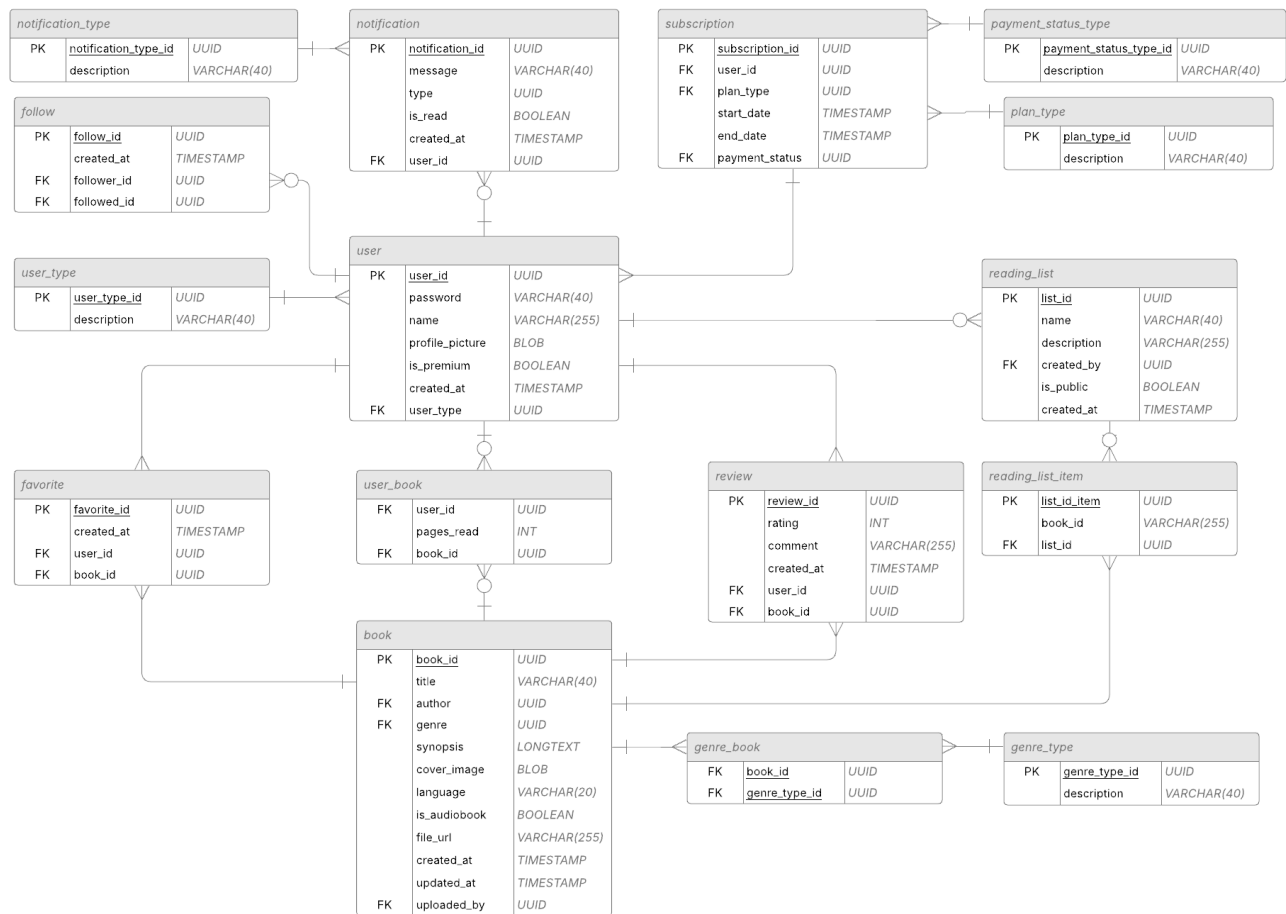
Priority: High

User Story: As a premium user, I want to be able to download books without restrictions to read them offline.

Acceptance Criteria: The application should display the option for premium users to download books and audiobooks.

Title: Advertising for premium users	Priority: High
User Story: As a premium user, I want to be able to enjoy audiobooks without advertising for a better experience.	
Acceptance Criteria: The application should not display any type of advertising only when the user reading is a premium one.	

Entity Relationship Mode



In the following link you can see the Entity Relationship Model: [E-R Model](#)