

Hệ thống quản lý Light Novel

Project môn học: Thực hành CSDL

Nhóm 1

Đại học Bách Khoa Hà Nội

Ngày 25 tháng 5 năm 2025

Nội dung

- 1 Mô tả bài toán
- 2 Thiết kế cơ sở dữ liệu
- 3 Câu lệnh SQL

Mục tiêu

Xây dựng một hệ thống web cho phép người dùng đọc light novel, tiếp cận với những bộ truyện mới nhất, và có thể bình luận về các bộ truyện mà mình đã đọc.

Yêu cầu hệ thống

Người dùng có thể:

- Xem tổng quan về các bộ light novel, bao gồm tên, tác giả, thể loại, mô tả, ảnh bìa, đánh giá, trạng thái (đang ra hay đã hoàn thành), tóm tắt nội dung, thời gian cập nhật gần nhất và thời gian tạo.
- Xem danh sách các chương của một bộ light novel, bao gồm tên chương, thời gian xuất bản và nội dung.
- Đọc các chương của bộ light novel.
- Đưa ra đánh giá và bình luận cho bộ novel đó.

Yêu cầu hệ thống (continued)

Đối với tác giả

Tác giả có thể:

- Thêm và sửa đổi các thuộc tính như thể loại, mô tả, trạng thái của một LN mình viết
- Tạo và chỉnh sửa các chương của một LN

Đối với người quản trị

- Có thể thêm và xóa các tác giả, LNs và thể loại
- Quản lý đánh giá (chỉ có thể xóa hoặc ẩn, không thể chỉnh sửa nội dung!)

Các thực thể

- **User**(user_id, username, password, role, created_at)
- **Author**(author_id, author_name, description, profile_picture)
- **Novel**(novel_id, name, author, genre, description, cover, status, last_updated, created_at)
- **Genre**(genre_id, genre_name, genre_description)
- **Chapter**(chapter_id, novel_id (fk), chapter_number, title, published_date, content)
- **Rating**(rating_id, user_id (fk), novel_id (fk), stars, content, created_at)

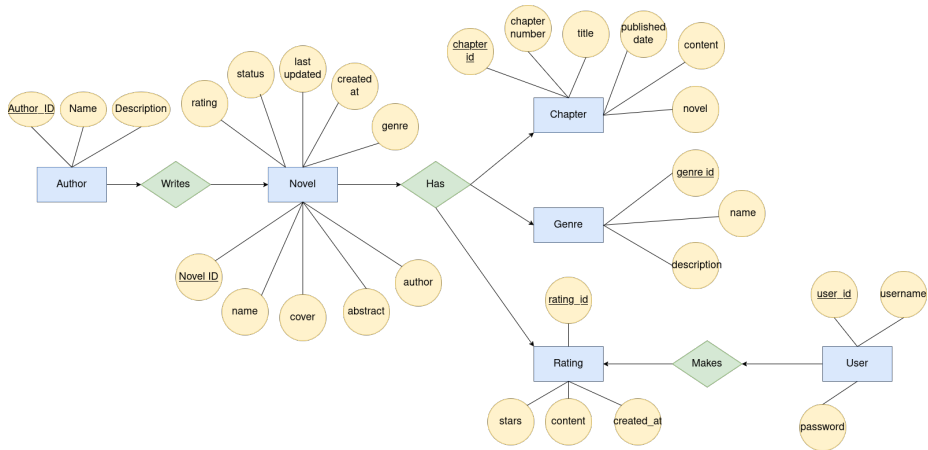
Mối quan hệ

- Một bộ light novel có thể có nhiều chương.
- Một bộ light novel có thể thuộc nhiều thể loại.
- Một bộ light novel có thể có nhiều tác giả.
- Một đánh giá thuộc về một người dùng và một bộ light novel.

Liên kết giữa các thực thể

Thực thể	Liên kết
Novel - Chapter	1 - N
Novel - Genre	N - N
Novel - Author	N - N
Novel - Rating	1 - N
User - Rating	1 - N

Sơ đồ ER



Bảng Author (Tác giả)

```
1 CREATE TABLE Author (  
2     author_id INTEGER PRIMARY KEY AUTOINCREMENT,  
3     author_name VARCHAR(255) NOT NULL,  
4     description TEXT DEFAULT 'The author has not disclosed any  
5     information about themselves.',  
6     profile_picture VARCHAR(255) NULL  
7 );
```

Bảng Genre (Thể loại)

```
1 CREATE TABLE Genre (  
2     genre_id INTEGER PRIMARY KEY AUTOINCREMENT,  
3     genre_name VARCHAR(100) NOT NULL,  
4     genre_description TEXT NOT NULL  
5 );
```

Bảng Novel (Tiểu thuyết) I

```
1 CREATE TABLE Novel (  
2     novel_id INTEGER PRIMARY KEY AUTOINCREMENT,  
3     name VARCHAR(255) NOT NULL,  
4     description TEXT NOT NULL,  
5     cover VARCHAR(255) NULL,  
6     status VARCHAR(50) DEFAULT 'ongoing'  
7     CHECK (status IN ('ongoing', 'completed', 'hiatus', '  
dropped'))),  
8     last_updated DATETIME DEFAULT CURRENT_TIMESTAMP,  
9     created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
10 );
```

Bảng quan hệ Novel-Author (Tiểu thuyết - Tác giả)

```
1  -- Many-to-Many relationship table
2  CREATE TABLE Novel_Authors (
3      id INTEGER PRIMARY KEY AUTOINCREMENT,
4      novel_id INTEGER NOT NULL,
5      author_id INTEGER NOT NULL,
6      FOREIGN KEY (novel_id) REFERENCES Novel(novel_id),
7      FOREIGN KEY (author_id) REFERENCES Author(author_id),
8      UNIQUE(novel_id, author_id)
9  );
```

Bảng quan hệ Novel-Genre (Tiểu thuyết - Thể loại)

```
1  -- Many-to-Many relationship table
2  CREATE TABLE Novel_Genres (
3      id INTEGER PRIMARY KEY AUTOINCREMENT,
4      novel_id INTEGER NOT NULL,
5      genre_id INTEGER NOT NULL,
6      FOREIGN KEY (novel_id) REFERENCES Novel(novel_id),
7      FOREIGN KEY (genre_id) REFERENCES Genre(genre_id),
8      UNIQUE(novel_id, genre_id)
9  );
```

Trigger: Cập nhật `last_updated` khi thay đổi

```
1 CREATE TRIGGER update_last_updated
2 AFTER INSERT OR UPDATE ON Chapter
3 BEGIN
4     UPDATE Novel
5     SET last_updated = CURRENT_TIMESTAMP
6     WHERE novel_id = NEW.novel_id;
7 END;
```

Bảng Chapter (Chương)

```
1 CREATE TABLE Chapter (  
2     chapter_id INTEGER PRIMARY KEY AUTOINCREMENT,  
3     novel_id INTEGER NOT NULL,  
4     chapter_number INTEGER NOT NULL,  
5     title VARCHAR(255) NOT NULL,  
6     published_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
7     content TEXT NOT NULL,  
8     FOREIGN KEY (novel_id) REFERENCES Novel(novel_id)  
9 );
```

Bảng Rating (Đánh giá)

```
1 CREATE TABLE Rating (  
2     rating_id INTEGER PRIMARY KEY AUTOINCREMENT,  
3     novel_id INTEGER NOT NULL,  
4     user_id INTEGER NOT NULL,  
5     stars INTEGER NOT NULL CHECK (stars BETWEEN 1 AND 5),  
6     content TEXT NULL,  
7     created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
8     FOREIGN KEY (novel_id) REFERENCES Novel(novel_id),  
9     FOREIGN KEY (user_id) REFERENCES auth_user(id)  
10 );
```


Truy vấn: Tìm tác giả theo tên

```
1 -- Find authors by name pattern
2 SELECT author_id, author_name, description
3 FROM Author
4 WHERE author_name LIKE '%Nguyen%';
5
6 -- Get all authors with their profile pictures
7 SELECT author_name, description, profile_picture
8 FROM Author
9 WHERE profile_picture IS NOT NULL
10 ORDER BY author_name;
```

Truy vấn: Lấy thông tin thể loại

```
1  -- Get all genres ordered by name
2  SELECT genre_name, genre_description
3  FROM Genre
4  ORDER BY genre_name;

5
6  -- Count novels in each genre
7  SELECT g.genre_name, COUNT(ng.novel_id) as novel_count
8  FROM Genre g
9  LEFT JOIN Novel_Genres ng ON g.genre_id = ng.genre_id
10 GROUP BY g.genre_id, g.genre_name
11 ORDER BY novel_count DESC;
```

Truy vấn: Lọc tiểu thuyết theo trạng thái

```
1 -- Get ongoing novels ordered by last update
2 SELECT name, status, last_updated
3 FROM Novel
4 WHERE status = 'ongoing'
5 ORDER BY last_updated DESC;
6
7 -- Get completed novels with their creation date
8 SELECT name, description, created_at
9 FROM Novel
10 WHERE status = 'completed'
11 ORDER BY created_at DESC;
```

Truy vấn: Lấy thông tin tác giả của tiểu thuyết

```
1 -- Get novels with their authors
2 SELECT n.name as novel_name, a.author_name
3 FROM Novel n
4 JOIN Novel_Authors na ON n.novel_id = na.novel_id
5 JOIN Author a ON na.author_id = a.author_id
6 ORDER BY n.name;
7
8 -- Get all novels by a specific author
9 SELECT n.name, n.status, n.last_updated
10 FROM Novel n
11 JOIN Novel_Authors na ON n.novel_id = na.novel_id
12 JOIN Author a ON na.author_id = a.author_id
13 WHERE a.author_name = 'Nguyen Van A';
```

Truy vấn: Lấy thể loại của tiểu thuyết

```
1 -- Get novels with their genres
2 SELECT n.name as novel_name, g.genre_name
3 FROM Novel n
4 JOIN Novel_Genres ng ON n.novel_id = ng.novel_id
5 JOIN Genre g ON ng.genre_id = g.genre_id
6 ORDER BY n.name;
7
8 -- Get all Fantasy novels
9 SELECT n.name, n.description
10 FROM Novel n
11 JOIN Novel_Genres ng ON n.novel_id = ng.novel_id
12 JOIN Genre g ON ng.genre_id = g.genre_id
13 WHERE g.genre_name = 'Fantasy';
```

Truy vấn: Lấy chương của tiểu thuyết

```
1 -- Get chapters of a specific novel
2 SELECT chapter_number, title, published_date
3 FROM Chapter
4 WHERE novel_id = 1
5 ORDER BY chapter_number;
6
7 -- Get latest chapter of each novel
8 SELECT n.name, c.title, c.published_date
9 FROM Novel n
10 JOIN Chapter c ON n.novel_id = c.novel_id
11 WHERE c.published_date = (
12     SELECT MAX(published_date)
13     FROM Chapter c2
14     WHERE c2.novel_id = n.novel_id
```

Truy vấn: Tính điểm đánh giá trung bình

```
1  -- Get average rating for each novel
2  SELECT n.name,
3         AVG(r.stars) as average_rating,
4         COUNT(r.rating_id) as total_ratings
5  FROM Novel n
6  LEFT JOIN Rating r ON n.novel_id = r.novel_id
7  GROUP BY n.novel_id, n.name
8  ORDER BY average_rating DESC;
9
10 -- Get highly rated novels (4+ stars)
11 SELECT n.name, AVG(r.stars) as avg_rating
12 FROM Novel n
13 JOIN Rating r ON n.novel_id = r.novel_id
14 GROUP BY n.novel_id, n.name
```

Truy vấn phức tạp: Thông tin đầy đủ tiểu thuyết

```
1 -- Get comprehensive novel information
2 SELECT
3     n.name as novel_name,
4     n.status,
5     n.last_updated,
6     GROUP_CONCAT(DISTINCT a.author_name) as authors,
7     GROUP_CONCAT(DISTINCT g.genre_name) as genres,
8     AVG(r.stars) as avg_rating,
9     COUNT(DISTINCT r.rating_id) as total_ratings,
10    COUNT(DISTINCT c.chapter_id) as total_chapters,
11    MAX(c.published_date) as latest_chapter_date
12 FROM Novel n
13 LEFT JOIN Novel_Authors na ON n.novel_id = na.novel_id
14 LEFT JOIN Author a ON na.author_id = a.author_id
15 LEFT JOIN Novel_Genres ng ON n.novel_id = ng.novel_id
16 LEFT JOIN Genre g ON ng.genre_id = g.genre_id
17 LEFT JOIN Rating r ON n.novel_id = r.novel_id
18 LEFT JOIN Chapter c ON n.novel_id = c.novel_id
19 GROUP BY n.novel_id, n.name, n.status, n.last_updated
20 ORDER BY avg_rating DESC, total_ratings DESC;
```


Truy vấn: Lấy tiểu thuyết đang hot

```
1 -- Get trending novels (recent updates + high ratings)
2 SELECT
3     n.name,
4     n.status,
5     AVG(r.stars) as avg_rating,
6     COUNT(r.rating_id) as rating_count,
7     n.last_updated,
8     (AVG(r.stars) * (1.0 / (julianday('now') - julianday(n.last_updated) + 1)))
9     as trending_score
10 FROM Novel n
11 LEFT JOIN Rating r ON n.novel_id = r.novel_id
12 WHERE n.status IN ('ongoing', 'completed')
13 GROUP BY n.novel_id, n.name, n.status, n.last_updated
14 HAVING COUNT(r.rating_id) > 0
15 ORDER BY trending_score DESC
16 LIMIT 10;
```