

Hệ thống quản lý Light Novel

Project môn học: Thực hành CSDL

Nhóm 1

Đại học Bách Khoa Hà Nội

Ngày 4 tháng 6 năm 2025

Nội dung

- 1 **Mô tả bài toán**
 - Mở đầu
 - Mục tiêu
- 2 **Thiết kế cơ sở dữ liệu**
- 3 **Câu lệnh SQL**

Nội dung


1 Mô tả bài toán

- Mở đầu
- Mục tiêu

2 Thiết kế cơ sở dữ liệu

3 Câu lệnh SQL

References

-  Xiaoyun Gong, Yuxi Lin, Ye Ding, and Lauren Klein.
Gender and power in japanese light novels.
In *CHR*, pages 256–267, 2022.
-  Sho Hasegawa and Masashi Yamada.
A statistical analysis of history of japanese light novels.
Kansei Engineering, 101(101), 2023.

Bối cảnh văn đề I

Bối cảnh văn đề

- Light Novel là thể loại tiểu thuyết quan trọng trong văn hóa đại chúng Nhật Bản.
- Đặc trưng bởi:
 - Cách viết dễ đọc.
 - Có minh họa theo phong cách anime/manga.
- Tên gọi "light novel" đã tồn tại hơn 25 năm; bối cảnh phát triển đã thay đổi nhiều trong thời gian này. [1, 2]

Bối cảnh vấn đề II

Sự thay đổi gần đây

- Internet có vai trò lớn trong sự thay đổi gần đây của light novel.
- Các nền tảng trực tuyến như **Shōsetsuka ni Narō** và **Kakuyomu** trở nên phổ biến.
 - Là nguồn tuyển chọn tác phẩm light novel mới.
 - Gắn liền với giai đoạn phát triển 2016–2020. [2]

Bối cảnh văn đề III

Ý nghĩa của dự án

- Dự án đọc light novel trực tuyến hưởng lợi từ xu hướng này.
- Công nghệ internet giúp việc tiếp cận và trải nghiệm light novel trở nên thuận tiện.
- Dự án cho phép khám phá:
 - Các thể loại như lãng mạn, phiêu lưu, lát cắt cuộc sống.
 - Sự phát triển của light novel trong bối cảnh văn hóa đại chúng hiện đại.

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 đánh giá về các bộ light novel đó. Hệ thống này sẽ bao gồm các tính năng như:

Đối với người dùng

- 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), thời gian cập nhật 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 cho bộ novel đó.

Yêu cầu hệ thống (continued)

Đối với người quản trị

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

Nội dung

1 Mô tả bài toán

- Mở đầu
- Mục tiêu

2 Thiết kế cơ sở dữ liệu

3 Câu lệnh SQL

Các thực thể

- **User**(user_id, username, password, role, created_at, ...)
- **Author**(author_id, author_name, description, profile_picture)
- **Novel**(novel_id, name, 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)

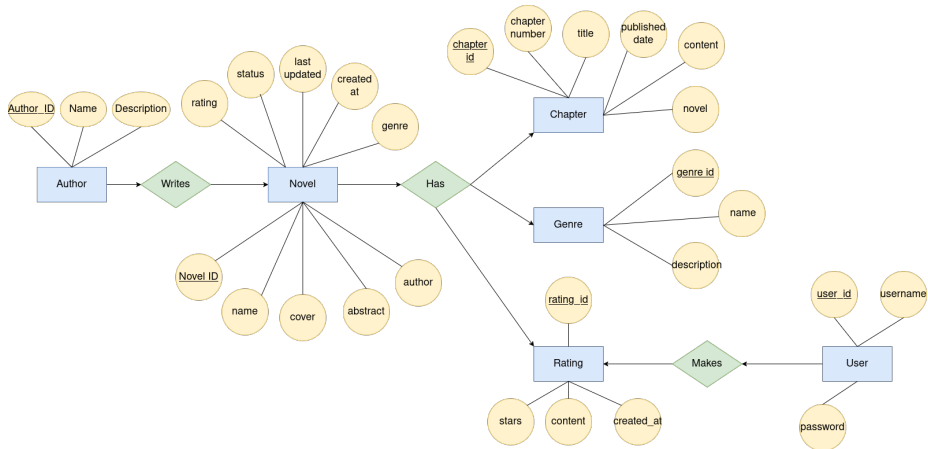
Mối quan hệ

- 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



Nội dung

- 1 Mô tả bài toán
 - Mở đầu
 - Mục tiêu
- 2 Thiết kế cơ sở dữ liệu
- 3 **Câu lệnh SQL**

Bảng auth_user (Người dùng hệ thống)

```
1 -- Django's built-in User table (simplified)
2 CREATE TABLE auth_user (
3     id INTEGER PRIMARY KEY AUTOINCREMENT,
4     username VARCHAR(150) UNIQUE NOT NULL,
5     first_name VARCHAR(150),
6     last_name VARCHAR(150),
7     email VARCHAR(254),
8     is_staff BOOLEAN DEFAULT FALSE,
9     is_active BOOLEAN DEFAULT TRUE,
10    is_superuser BOOLEAN DEFAULT FALSE,
11    date_joined DATETIME DEFAULT CURRENT_TIMESTAMP,
12    last_login DATETIME NULL,
13    password VARCHAR(128) NOT NULL
14 );
```


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 có 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     UNIQUE(novel_id, chapter_number)  
10    -- Ensure chapter_number is unique within a novel  
11 );
```

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    UNIQUE(novel_id, user_id)  
11 );
```


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;
```

Hàm: Định dạng tên tác giả

```
1 -- Create function to format author names
2 CREATE FUNCTION format_author_names(novel_id_param INTEGER)
3 RETURNS TEXT
4 READS SQL DATA
5 DETERMINISTIC
6 BEGIN
7     DECLARE author_list TEXT DEFAULT '';
8
9     SELECT GROUP_CONCAT(a.author_name SEPARATOR ', ') INTO
10    author_list
11    FROM Author a JOIN Novel_Authors na ON a.author_id = na.
12    author_id
13    WHERE na.novel_id = novel_id_param ORDER BY a.author_name;
14
15    RETURN COALESCE(author_list, 'Unknown Author');
16 END;
```

Truy vấn: Lấy thông tin tác giả của tiểu thuyết I

```
1  -- Get novels with their authors
2  SELECT n.name as novel_name,
3         format_author_names(n.novel_id) as authors,
4         n.status,
5         n.last_updated
6  FROM Novel n
7  JOIN Novel_Authors na ON n.novel_id = na.novel_id
8  JOIN Author a ON na.author_id = a.author_id
9  GROUP BY n.novel_id, n.name, n.status, n.last_updated
10 ORDER BY n.last_updated DESC, n.name;
```

Truy vấn: Lấy thông tin tác giả của tiểu thuyết II

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


Truy vấn: Tính điểm đánh giá trung bình I

```
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;
```

Truy vấn: Tính điểm đánh giá trung bình II

```
1 -- Get highly rated novels (4+ stars)
2 SELECT n.name, AVG(r.stars) as avg_rating FROM Novel n
3 JOIN Rating r ON n.novel_id = r.novel_id
4 GROUP BY n.novel_id, n.name
5 HAVING AVG(r.stars) >= 4.0
6 ORDER BY avg_rating DESC;
```

Truy vấn phức tạp: Lấy thông tin tiểu thuyết cụ thể

```
1  -- Get novels with author containing 'Can' and genre 'Fantasy
   ', while excluding those with low ratings
2  SELECT n.name, format_author_names(n.novel_id) as authors,
3         g.genre_name, AVG(r.stars) as avg_rating,
4         n.status, n.last_updated
5  FROM Novel n NATURAL JOIN Novel_Genres ng
6  NATURAL JOIN Genre g NATURAL JOIN Novel_Authors na
7  NATURAL JOIN Author a
8  LEFT JOIN Rating r ON n.novel_id = r.novel_id
9  WHERE a.author_name LIKE '%Can%'
10     AND g.genre_name = 'Fantasy' AND AVG(r.stars) >= 3.0
11  GROUP BY n.novel_id, n.name, g.genre_name, n.status, n.
         last_updated
12  ORDER BY n.last_updated DESC, avg_rating DESC;
```

Truy vấn: Lấy tiểu thuyết đang hot

```

1  -- Get trending novels (recent updates + high ratings + recent 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      (
9          -- new_rating_count in last 7 days * 0.5
10         (
11             SELECT COUNT(*) FROM Rating r2
12             WHERE r2.novel_id = n.novel_id
13                 AND r2.created_at >= datetime('now', '-7 days')
14         ) * 0.5
15         +
16         -- avg_rating * 0.3
17         COALESCE(AVG(r.stars), 0) * 0.3
18         +
19         -- (1 / hours_since_last_update) * 0.2
20         (1.0 / ((julianday('now') - julianday(n.last_updated)) * 24 + 1)) * 0.2
21     ) as trending_score

```

Cách khác để lấy tiểu thuyết đang hot? I

```
1 ALTER TABLE Novel
2 ADD COLUMN trending_score REAL DEFAULT 0;
3
4 CREATE TRIGGER update_trending_score
5 AFTER INSERT OR UPDATE ON Rating
6 BEGIN
7     -- Declare variables
8     DECLARE new_rating_count INTEGER;
9     DECLARE avg_rating REAL;
10    DECLARE hours_since_last_update REAL;
11    DECLARE trending_score REAL;
12
13    -- Calculate new_rating_count in last 7 days
14    SELECT COUNT(*) INTO new_rating_count
15    FROM Rating r2
```

Cách khác để lấy tiểu thuyết đang hot? II

```
16 WHERE r2.novel_id = NEW.novel_id
17     AND r2.created_at >= datetime('now', '-7 days');
18
19 -- Calculate avg_rating
20 SELECT COALESCE(AVG(stars), 0) INTO avg_rating
21 FROM Rating
22 WHERE novel_id = NEW.novel_id;
23
24 -- Calculate hours_since_last_update
25 SELECT (julianday('now') - julianday(last_updated)) * 24 + 1
26 INTO hours_since_last_update
27 FROM Novel
28 WHERE novel_id = NEW.novel_id;
29
30 -- Calculate trending_score
```

Cách khác để lấy tiểu thuyết đang hot? III

```
30  SET trending_score = new_rating_count * 0.5
31                        + avg_rating * 0.3
32                        + (1.0 / hours_since_last_update) * 0.2;
33
34  UPDATE Novel
35  SET trending_score = trending_score
36  WHERE novel_id = NEW.novel_id;
37  END;
```

Cách khác để lấy tiểu thuyết đang hot? IV

```
1 CREATE EVENT update_trending_score_event
2 ON SCHEDULE EVERY 1 DAY
3 DO
4 BEGIN
5     UPDATE Novel
6     SET trending_score = (
7         -- new_rating_count in last 7 days * 0.5
8         (
9             SELECT COUNT(*) FROM Rating r2
10            WHERE r2.novel_id = Novel.novel_id
11                  AND r2.created_at >= datetime('now', '-7 days')
12        ) * 0.5
13     +
```


Cách khác để lấy tiểu thuyết đang hot? V

```
14      -- avg_rating * 0.3
15      COALESCE((SELECT AVG(stars) FROM Rating WHERE novel_id
= Novel.novel_id), 0) * 0.3
16      +
17      -- (1 / hours_since_last_update) * 0.2
18      (1.0 / ((julianday('now') - julianday(Novel.
last_updated)) * 24 + 1)) * 0.2
19  );
20  END;
```

Cách khác để lấy tiểu thuyết đang hot? VI

```
1 -- Get top 10 trending novels
2 SELECT name, status, trending_score
3 FROM Novel
4 ORDER BY trending_score DESC
5 LIMIT 10;
```