An Vu
CSCI 4850 8856 Database Management Systems
November 9 2025

Project Data Definition

I. Description of the Application

This project designs a relational database for Iris ’25, a digital reading platform with user profiles, reading progress tracking, notes, and optional eye - tracking integration. The goal is to build the backend data system that supports user activity, reading analytics, and book management.

Data Stored

The database stores structured information about:

- **Users, roles, and devices** - manage profiles and access.
- **Books, authors, chapters, and genres** - organize and categorize reading content.
- **Reading artifacts** - store user - specific data such as progress, notes, and reading sessions.
- **Optional eye - tracking session summaries** - aggregate metrics for engagement and usability analysis.

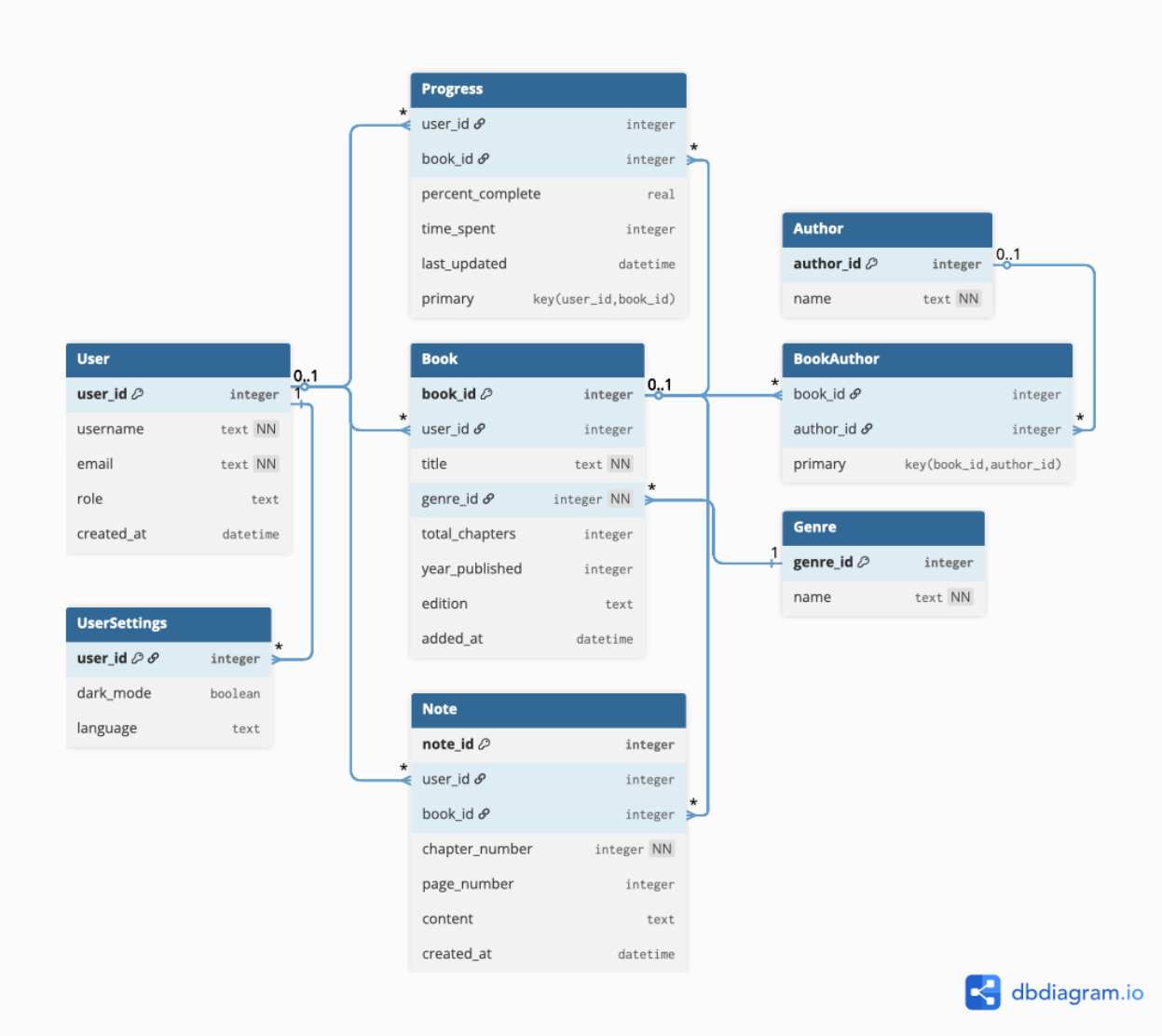
Expected Queries

- **Aggregation**: average completion by genre/author, total read time per day/week/month.
- **Nested/Set queries**: users who started but didn’t finish books, or books read by one user but not another.
- **Joins**: linking user -> progress -> book -> genre, or analyzing author multi - book stats.
- **Constraint validation**: unique usernames/emails, FK enforcement, valid progress percentages and durations.

How Queries Help the Application

- **For users**: generate reading statistics (streaks, reading hours, completion rates) and build personalized book recommendations.
- **For authors**: reveal chapters with high drop-off rates, most highlighted chapters, or frequently revisited content to measure reader engagement.
- **For developers**: support auditing, debugging, and overall system analytics through user activity logs.

II. Entity - Relationship (ER) diagram



- The Iris Reader ’25 database models a digital reading platform where users can upload, read, and annotate books.
- The system includes multiple relationship types and ER features to reflect realistic app behavior.
  - User - Book (1:N): One user can upload many books, but each book belongs to only one uploader.
  - Book - Author (M:N): A book can have several authors, and an author can contribute to many books. The BookAuthor bridge table connects them without duplication.

  - Book - Genre (1:N): Each book is linked to one genre, while each genre can include multiple books.
  - User - Book via Progress (M:N): Tracks what each user reads and how far they've gone. The Progress table records percent complete, time spent, and last update.
  - User - Note - Book (1:N both ways): Users can leave many notes per book, and books can receive notes from many users.
  - User - UserSettings (1:1): Each user has exactly one settings record to store preferences like dark mode and language.
- The model uses different attribute types, including simple attributes (username, title), optional ones (role, user_id in Book), and derived values (percent_complete).
- Cardinalities include 1:1, 1:N, and M:N, while participation constraints are handled through foreign key rules such as [delete: restrict] and [delete: set null].
- Weak entities (Progress and Note) depend on both User and Book for their identity.
- Specialization appears through UserSettings, extending the User entity for individualized app preferences.
- Overall, the design is normalized, easy to maintain, and supports analytics, personalization, and user interaction across the Iris Reader system.

- For information and details, check out **`iris25_schema.dbml`** in the submission folder. Each table and relationship includes comments explaining its purpose and constraints.

III. SQL specification (DDL)

- I used MySQL to implement this database because it's reliable, scalable, and widely used in production environments.
- The SQL DDL file (**`iris25_sql_specification_ddl.sql`**) in the submission folder defines all eight tables, their primary and foreign keys, data types, and relational constraints.
- Each table was successfully created and verified in MySQL, confirming that all relationships and dependencies are valid.

```
-- ============================================================
-- Iris Reader '25 Database Schema
-- Target DBMS: MySQL
-- SQL DDL for tables, keys, and relationships
-- ============================================================

-- Drop existing tables (safeguard for re-runs)
DROP TABLE IF EXISTS UserSettings;
DROP TABLE IF EXISTS Note;
DROP TABLE IF EXISTS Progress;
DROP TABLE IF EXISTS BookAuthor;
DROP TABLE IF EXISTS Book;
DROP TABLE IF EXISTS Genre;
DROP TABLE IF EXISTS Author;
DROP TABLE IF EXISTS User;

-- ============================================================
-- 1. User Table
-- ============================================================
CREATE TABLE User (
  user_id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(100) NOT NULL UNIQUE,
  email VARCHAR(100) NOT NULL UNIQUE,
  role VARCHAR(50),
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

```
-- ============================================================
-- 2. Author Table
-- ============================================================
CREATE TABLE Author (
  author_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(150) NOT NULL
);

-- ============================================================
-- 3. Genre Table
-- ============================================================
CREATE TABLE Genre (
  genre_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL UNIQUE
);

-- ============================================================
-- 4. Book Table
-- ============================================================
CREATE TABLE Book (
  book_id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT,
  title VARCHAR(200) NOT NULL,
  genre_id INT NOT NULL,
  total_chapters INT,
  year_published INT,
  edition VARCHAR(50),
  added_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE SET NULL,
  FOREIGN KEY (genre_id) REFERENCES Genre(genre_id) ON DELETE RESTRICT
);

-- ============================================================
-- 5. BookAuthor (Bridge Table for M:N)
-- ============================================================
CREATE TABLE BookAuthor (
```

```
  book_id INT,
  author_id INT,
  PRIMARY KEY (book_id, author_id),
  FOREIGN KEY (book_id) REFERENCES Book(book_id) ON DELETE CASCADE,
  FOREIGN KEY (author_id) REFERENCES Author(author_id) ON DELETE
CASCADE
);

-- ============================================================
-- 6. Progress (Bridge Table for User ↔ Book)
-- ============================================================
CREATE TABLE Progress (
  user_id INT,
  book_id INT,
  percent_complete FLOAT DEFAULT 0 CHECK (percent_complete BETWEEN 0
AND 100),
  time_spent INT DEFAULT 0,
  last_updated DATETIME DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (user_id, book_id),
  FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
  FOREIGN KEY (book_id) REFERENCES Book(book_id) ON DELETE CASCADE
);

-- ============================================================
-- 7. Note Table
-- ============================================================
CREATE TABLE Note (
  note_id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  book_id INT NOT NULL,
  chapter_number INT NOT NULL,
  page_number INT,
  content TEXT,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
  FOREIGN KEY (book_id) REFERENCES Book(book_id) ON DELETE CASCADE
```

```
);

-- ============================================================
-- 8. UserSettings (1:1 with User)
-- ============================================================
CREATE TABLE UserSettings (
  user_id INT PRIMARY KEY,
  dark_mode BOOLEAN,
  language VARCHAR(50),
  FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
);

-- ============================================================
-- End of Schema
-- ============================================================
```

IV. Sample Data

- The sample data file (**`iris25_sample_data.sql`**) populates the Iris '25 database with example records for testing and demonstration. It includes sample users, authors, genres, and books, along with corresponding reading progress and notes.
- This data helps verify:
  - Foreign key relationships between tables (e.g., User–Book–Progress links).
  - Query functionality and schema integrity.
  - How the app handles typical user scenarios like tracking progress, saving notes, and managing preferences.

```
-- ============================================================
-- Iris '25 Sample Data
-- Run iris25_sql_specification_ddl.sql first to set up the schema.
-- Then execute iris25_sample_data.sql to populate it.
-- ============================================================

-- ===== USERS =====
INSERT INTO User (username, email, role, created_at)
VALUES
('anvu', 'anvu@example.com', 'developer', '2025-11-01 09:45:00'),
('emma', 'emma@example.com', 'tester', '2025-11-02 14:10:00'),
('nori', 'nori@example.com', 'child', '2025-11-03 18:25:00');

-- ===== AUTHORS =====
INSERT INTO Author (name) VALUES
('Jeff Johnson PhD'),
('Abraham Silberschatz'),
('Henry F. Korth'),
('S. Sudarshan'),
('Alice Zhao'),
('Patrick Horvath'),
('Pailin Chongchitnant'),
('Julia Child'),
('Louisette Bertholle'),
('Simone Beck'),
```

```
('J. Kenji López-Alt'),
('George Orwell'),
('Russell Baker'),
('Akira Toriyama'),
('Don Norman');

-- ===== GENRES =====
INSERT INTO Genre (name) VALUES
('Computers & Technology'),
('Cookbooks / Food & Wine'),
('Literature & Fiction'),
('Politics / Social Sciences'),
('Arts & Design'),
('Education / Textbook');

-- ===== BOOKS =====
INSERT INTO Book (user_id, title, genre_id, total_chapters,
year_published, edition, added_at)
VALUES
(1, 'Designing with the Mind in Mind: Simple Guide to Understanding
User Interface Design Guidelines', 1, 12, 2019, '3rd Edition',
'2025-11-04 11:00:00'),
(1, 'Database System Concepts', 1, 16, 2020, '7th Edition',
'2025-11-04 11:10:00'),
(1, 'SQL Pocket Guide: A Guide to SQL Usage', 1, 10, 2021, '4th
Edition', '2025-11-05 09:30:00'),
(1, 'Beneath the Trees Where Nobody Sees', 3, 18, 2024, NULL,
'2025-11-05 10:00:00'),
(2, 'Sabai: 100 Simple Thai Recipes for Any Day of the Week', 2, 10,
2023, NULL, '2025-11-06 14:00:00'),
(2, 'Mastering the Art of French Cooking, Volume 1: A Cookbook', 2,
24, 1983, NULL, '2025-11-07 08:45:00'),
(2, 'Mastering the Art of French Cooking, Vol. 2: A Classic
Continued', 2, 20, 1983, NULL, '2025-11-07 09:15:00'),
(2, 'The Food Lab: Better Home Cooking Through Science', 2, 14, 2015,
NULL, '2025-11-08 10:30:00'),
```

```
(3, 'Animal Farm: 75th Anniversary Edition Mass Market', 4, 10, 2004,
NULL, '2025-11-09 13:00:00'),
(3, 'Dragon Ball, Vol. 1: The Monkey King (Shonen Jump Graphic
Novel)', 3, 8, 2003, NULL, '2025-11-09 13:15:00'),
(3, 'The Design of Everyday Things: Revised and Expanded Edition', 5,
12, 2013, 'Revised Edition', '2025-11-09 13:45:00');

-- ===== BOOKAUTHOR =====
INSERT INTO BookAuthor (book_id, author_id) VALUES
(1, 1),
(2, 2), (2, 3), (2, 4),
(3, 5),
(4, 6),
(5, 7),
(6, 8), (6, 9), (6, 10),
(7, 8), (7, 10),
(8, 11),
(9, 12), (9, 13),
(10, 14),
(11, 15);

-- ===== PROGRESS =====
INSERT INTO Progress (user_id, book_id, percent_complete, time_spent,
last_updated)
VALUES
(1, 2, 85, 320, '2025-11-10 15:30:00'),
(1, 3, 98, 270, '2025-11-10 16:00:00'),
(2, 5, 100, 500, '2025-11-10 18:00:00'),
(2, 6, 1, 5, '2025-11-10 18:10:00'),
(2, 8, 5, 25, '2025-11-10 18:15:00'),
(3, 4, 70, 180, '2025-11-11 09:00:00'),
(3, 10, 100, 240, '2025-11-11 09:15:00');

-- ===== NOTES =====
INSERT INTO Note (user_id, book_id, chapter_number, page_number,
content, created_at)
```

```
VALUES
(1, 2, 6, 254, 'Coding in MySQL is not that bad, but I think the E-R model can make things more complicated. I still forget what the diamond means because I hardly ever use it.', '2025-11-11 10:00:00'),
(1, 2, 9, 412, 'I learned Servlet + SQL combo in Advanced Java Programming. I wonder if Servlet could work with Python because Java is scary.', '2025-11-11 10:15:00'),
(2, 5, 5, 87, 'Tom Yam Pla: clear broth fish soup with lemongrass and kaffir lime leaves. Tom Kha Gai: white broth chicken soup with coconut and galangal. Red snapper is a good choice for Tom Yam Pla. Thai chili powder can get really spicy so start light.', '2025-11-11 10:30:00'),
(3, 10, 3, 58, 'Woof woof woof!', '2025-11-11 10:45:00'),
(3, 10, 3, 69, 'Grrrrrrr', '2025-11-11 10:50:00');

-- ===== USER SETTINGS =====
INSERT INTO UserSettings (user_id, dark_mode, language)
VALUES
(1, TRUE, 'English'),
(2, FALSE, 'Vietnamese, Cantonese, Traditional Chinese'),
(3, TRUE, 'Dognese, Shibanese');

-- ============================================================
-- End of iris25_sample_data.sql
-- ============================================================
```