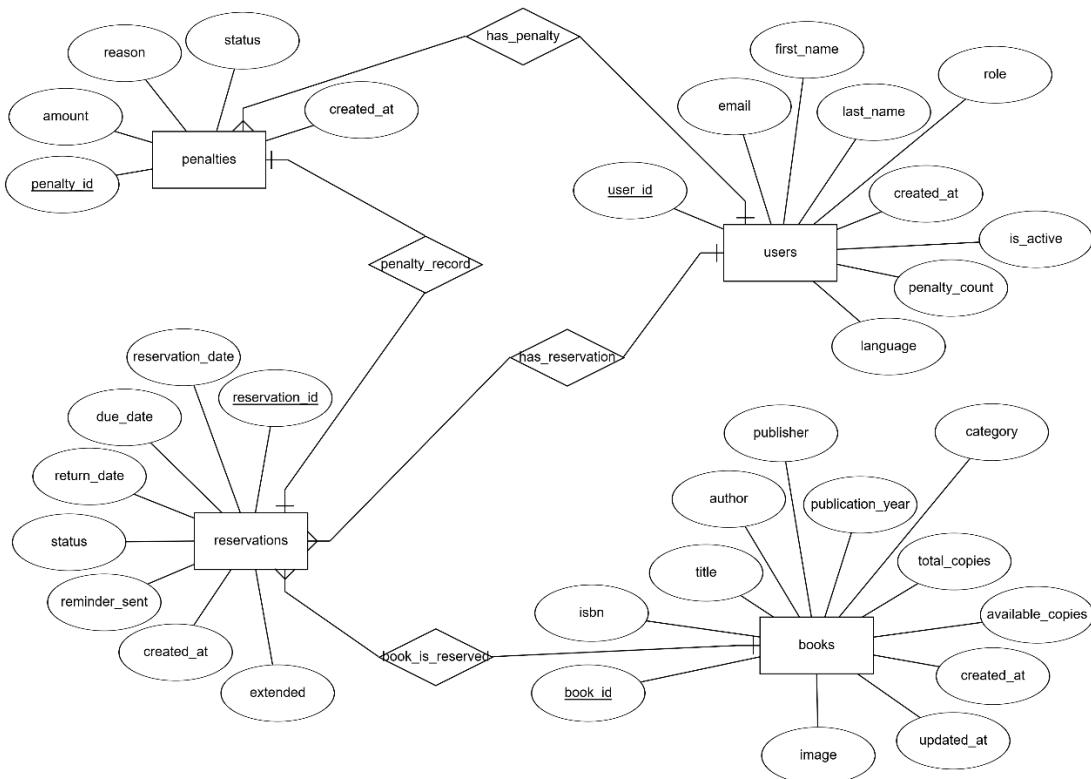


Library Book Reservation System

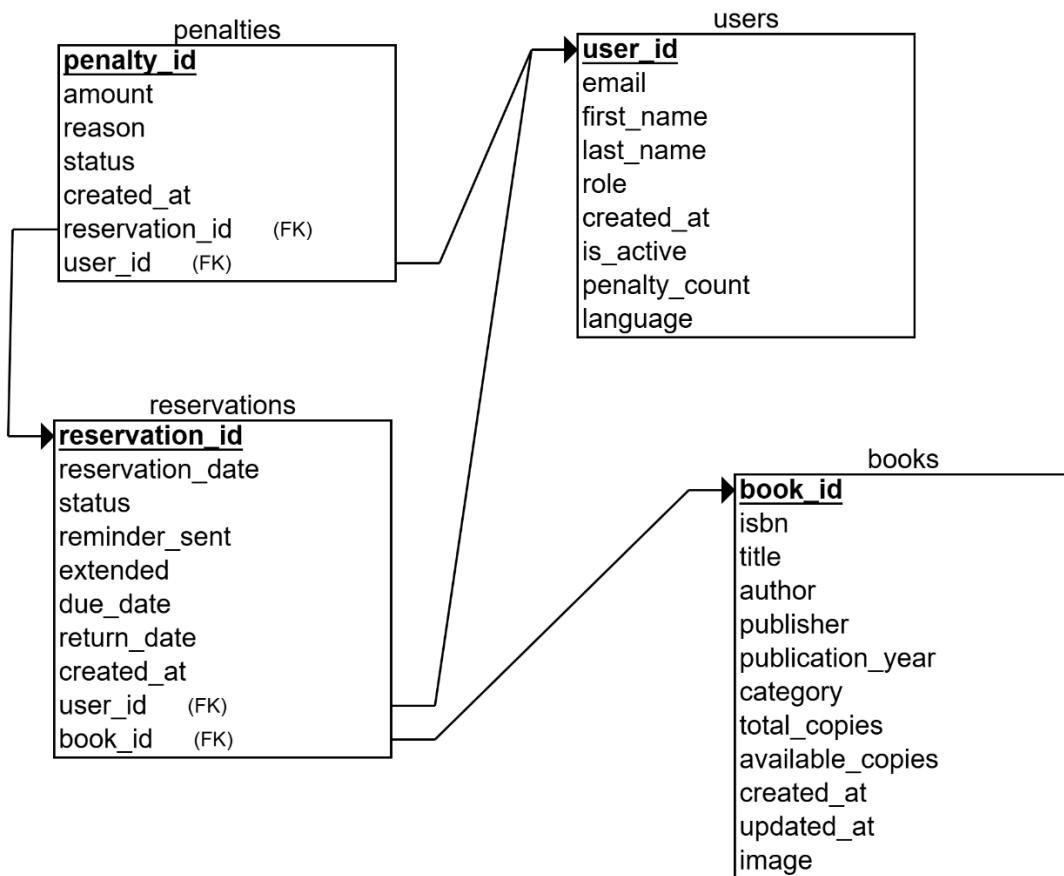
1. Description of the project

Our project aims to develop a modern library management system that digitalizes the library's lending operations. With this system, customers can browse books, make reservations, and manage their borrowed items online. Library staff, on the other hand, can track returns, manage the book collection, and handle overdue returns from customers.

2. ER Diagram



3. Relational database schema



4. Verbal description of each entity type and relationship type in the ER diagram

users

- Purpose:** Describes a library user
- Attributes:**
 - user_id (primary key)
 - email – user's email address
 - first_name – user's first name
 - last_name – user's last name
 - role – user's role (enum: customer, librarian)
 - created_at - timestamp
 - is_active - boolean (true, false), indicates whether the user is active
 - penalty_count - number of penalties accumulated
 - language - saved language for the user (enum: en, ja, ru)
- Relationships:**
 - has_penalty - (users - penalties) links a user to their penalties (1:M)
 - has_reservation - (users - reservations) links a user to their reservations (1:M)

penalties

- **Purpose:** Describes the user's penalties
- **Attributes:**
 - penalty_id (primary key)
 - amount - total penalty amount in currency
 - reason - name of the book that caused the penalty, explanation
 - status - (enum - pending, paid, waived)
 - created_at - timestamp
 - reservation_id (foreign key)
 - user_id (foreign key)
- **Relationships:**
 - has_penalty - (users - penalties) links a user to their penalties (1:M)
 - penalty_record - (penalties - reservations) links a penalty to a reservation (1:1)

reservations

- **Purpose:** Describes a book loan transaction
- **Attributes:**
 - reservation_id (primary key)
 - reservation_date - timestamp
 - status - (enum - active, returned, overdue, cancelled)
 - remainder_sent - boolean (true, false)
 - extended - boolean (true, false), indicates whether the loan was extended
 - due_date – timestamp, due date for returning the book
 - return_date - timestamp, actual return date of the book, NULL if not returned
 - created_at - timestamp
 - user_id (foreign key)
 - book_id (foreign key)
- **Relationships:**
 - penalty_record - (penalties - reservations) links a penalty to a reservation (1:1)
 - has_reservation - (users – reservations) links a user to their reservations (1:M)
 - book_is_reserved - (books - reservations) links books to reservations (1:M)

books

- **Purpose:** Describes books in the library
- **Attributes:**
 - book_id (primary key)
 - isbn - ISBN code
 - title - book title
 - author - book author
 - publisher - book publisher
 - publication_year - year of publication
 - category - category
 - total_copies - total number of copies
 - available_copies - number of available copies
 - created_at - timestamp

- updated_at - timestamp , last update date of the book
- image - URL of the book cover image
- Relationships:
 - book_is_reserved - (books - reservations) links

5. Sample Data

The password is not stored in the project. Supabase handles it securely and stores only a hashed version.

INSERT INTO users (email, first_name, last_name)

The values for the column user_id are generated automatically.

The columns role, created_at, is_active, penalty_count, and language use default values automatically.

VALUES ('customer@library.com', 'John', 'Doe');

user_id	email	first_name	last_name	role	created_at	is_active	penalty_count	language
1	customer@library.com	John	Doe	customer	2025-11-10 12:33:54	1	0	en

INSERT INTO books (isbn, title, author, publisher, publication_year, category, total_copies, available_copies, image)

The values for the column book_id are generated automatically. The columns created_at and updated_at use default values automatically.

VALUES('978-0-547-92822-7', 'The Hobbit', 'J.R.R. Tolkien', 'Houghton Mifflin Harcourt', 1937, 'Fiction', 5, 5, 'https://m.media-amazon.com/images/I/81uEDUfKBZL._SL1500_.jpg'),

VALUES('978-0-452-28423-4', '1984', 'George Orwell', 'Secker & Warburg', 1949, 'Fiction', 5, 5, 'https://www.suomalainen.com/cdn/shop/files/9789510505830_1-vuonna-1984.jpg?v=1743690167&width=950');

book_id	isbn	title	author	publisher	publication_ye...	category	total_copi...	available_copies	created_at	updated_at	image
1	978-0-547-92822-7	The Hobbit	J.R.R. Tolkien	Houghton Mifflin Harcourt	1937	Fiction	5	5	2025-11-07 16:10:03	2025-11-07 16:10:03	https://m.media-amazon.com/images/I/81uEDUfKBZL._SL1500_.jpg
2	978-0-452-28423-4	1984	George Orwell	Secker & Warburg	1949	Fiction	5	5	2025-11-07 16:10:06	2025-11-07 16:10:06	https://www.suomalainen.com/cdn/shop/files/9789510505830_1-vuonna-1984.jpg?v=1743690167&width=950

INSERT INTO reservations (due_date, book_id, user_id)

The values for the column reservation_id are generated automatically. The columns reservation_date, status, reminder_sent, extended, and created_at use default values automatically. The column return_date remains empty (= NULL) until the customer returns the book.

VALUES('2025-11-30', 2, 1);

MySQL stores BOOLEAN as TINYINT(1): TRUE = 1, FALSE = 0

reservation_id	reservation_date	status	reminder_sent	extended	due_date	return_date	created_at	user_id	book_id
1	2025-11-07 16:10:40	active	0	0	2025-11-30 00:00:00	NULL	2025-11-07 16:10:40	1	2

UPDATE reservations SET status = 'returned', return_date = CURRENT_TIMESTAMP
WHERE reservation_id = 1;

reservation_id	reservation_date	status	reminder_sent	extended	due_date	return_date	created_at	user_id	book_id
1	2025-11-07 17:13:18	returned	0	0	2025-11-30 00:00:00	2025-11-07 17:14:02	2025-11-07 17:13:18	1	2

INSERT INTO **penalties** (amount, reason, reservation_id, user_id)

The values for the column penalty_id are generated automatically. The columns amount, status, and created_at use default values automatically.

VALUES('Overdue book: "The Hobbit" (7 days overdue)', '2025-10-30', 1, 1);

penalty_id	amount	reason	status	created_at	reservation_id	user_id
1	0.00	Overdue book: "The Hobbit" (7 days overdue)	pending	2025-10-30 00:00:00	1	1

CREATE DATABASE LibraryHub;

USE LibraryHub;

DROP TABLE IF EXISTS penalties, books, reservations, users;

```
CREATE TABLE users
(
    user_id INT NOT NULL AUTO_INCREMENT,
    email VARCHAR(100) NOT NULL,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    role ENUM('customer', 'librarian') NOT NULL DEFAULT 'customer',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_active BOOLEAN NOT NULL DEFAULT TRUE,
    penalty_count INT NOT NULL DEFAULT 0,
    language ENUM('en', 'ja', 'ru') NOT NULL DEFAULT 'en',
    PRIMARY KEY (user_id)
);
```

```
CREATE TABLE books
(
    book_id INT NOT NULL AUTO_INCREMENT,
    isbn VARCHAR(100) NOT NULL,
    title VARCHAR(100) NOT NULL,
    author VARCHAR(100) NOT NULL,
    publisher VARCHAR(100) NOT NULL,
    publication_year INT NOT NULL,
    category VARCHAR(100) NOT NULL,
    total_copies INT NOT NULL,
    available_copies INT NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
    UPDATE CURRENT_TIMESTAMP,
    image VARCHAR(200) NOT NULL,
    PRIMARY KEY (book_id),
);
```

```
CREATE TABLE reservations
(
    reservation_id INT NOT NULL AUTO_INCREMENT,
    reservation_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    status ENUM('active', 'returned', 'overdue', 'cancelled') NOT NULL DEFAULT 'active',
    reminder_sent BOOLEAN NOT NULL DEFAULT FALSE,
    extended BOOLEAN NOT NULL DEFAULT FALSE,
    due_date TIMESTAMP NOT NULL,
    return_date TIMESTAMP NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    user_id INT NOT NULL,
    book_id INT NOT NULL,
    PRIMARY KEY (reservation_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (book_id) REFERENCES books(book_id)
);
```

```
CREATE TABLE penalties
(
penalty_id INT NOT NULL AUTO_INCREMENT,
amount DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
reason VARCHAR(255) NOT NULL,
status ENUM('pending', 'paid', 'waived') NOT NULL DEFAULT 'pending',
created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
reservation_id INT NOT NULL,
user_id INT NOT NULL,
PRIMARY KEY (penalty_id),
FOREIGN KEY (reservation_id) REFERENCES reservations(reservation_id),
FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

```
INSERT INTO users
(email, first_name, last_name) VALUES ('customer@library.com', 'John', 'Doe');
```

```
INSERT INTO books (isbn, title, author, publisher, publication_year,
category, total_copies, available_copies, image)
VALUES('978-0-547-92822-7', 'The Hobbit', 'J.R.R. Tolkien', 'Houghton Mifflin
Harcourt', 1937, 'Fiction', 5, 5, 'https://m.media-amazon.com/images/I/81uEDUfKBZL.\_SL1500\_.jpg');
INSERT INTO books (isbn, title, author, publisher, publication_year,
category, total_copies, available_copies, image)
VALUES('978-0-452-28423-4', '1984', 'George Orwell', 'Secker &
Warburg', 1949, 'Fiction', 5, 5, 'https://www.suomalainen.com/cdn/shop/files/9789510505830\_1-vuonna-1984.jpg?v=1743690167&width=950');
```

```
INSERT INTO reservations (due_date, book_id, user_id)
VALUES('2025-11-30', 2, 1);
```

```
UPDATE reservations SET status = 'returned', return_date = CURRENT_TIMESTAMP
WHERE reservation_id = 1;
```

```
INSERT INTO penalties(reason, created_at, reservation_id, user_id)
VALUES('Overdue book: "The Hobbit" (7 days overdue)', '2025-10-30', 1, 1);
```

```
SELECT * FROM users;
SELECT * FROM books;
SELECT * FROM reservations;
SELECT * FROM penalties;
```