

E-Library Application

Github source: <https://github.com/rochiekop/RDBMS-E-Library>

➤ Mission Statement

- Create a library application that oversees multiple libraries and gives information about the available books users can loan.
- Provides convenience for users in finding information about books that can be borrowed and can make a booking process if the book to be borrowed is unavailable.

➤ Creating Table Structure

- Identification Actor



Library



Users

- Identification Table

| Table | Description |
|---------|---|
| Library | This table stores details about library data, including their ID, created_date, name, city, address and created_by |
| Users | This table stores details about Users, including ID, created_date, username, fullname, gender, email, no_telephone, address, is_admin, created_by |

- Each library has a variety collection of books (Table Books)
- Each library has own catalog book and information about stock of every book (Table Library_Books)
- Each book in library has its own book details (Table Book_Details)

| Table | Description |
|---------------|---|
| Books | This table stores details about Books data, including their ID, created_date, title, category_id, publisher_id, year, description, created_by |
| Library_Books | This table stores details about Books in each library, including their ID, created_date, library_id, book_id, stock, quantity created_by |
| Book_Details | This table stores details about Books Details, including their ID, created_date, library_book_id, isbn, is_avaliable, created_by |

- Each book has a different category (Table Categories)
- Every book has information about the publisher (Table Publishers)
- Every book has information about the author and in one book can be more than one author (Table Book_Authors, Authors)

| Table | Description |
|--------------|--|
| Categories | This table stores details about Categories data, including their ID, created_date, name, created_by |
| Publishers | This table stores details about Publisher, including their ID, created_date, name, address, created_by |
| Authors | This table stores details about Authors, including their ID, created_date, name, created_by |
| Book_Authors | This table stores details about Mapping Book and Authors, including their ID, created_date, book_id, author_id, created_by |

- Users can borrow the books in library with a maximum loan period of 2 weeks (Table Borrows)

→ Each user can only borrow 2 books (Table Borrow_Details)

| Table | Description |
|----------------|--|
| Borrows | This table stores details about loan book, including their ID, created_date, user_id, start_date, due_date, status_id, and created_by |
| Borrow_Details | This table stores details about details of every loan books, including their ID, created_date, borrow_id, book_detail_id, return_date, status_id, and created_by |

→ If the book to be borrowed is not available then the user takes a hold first with a hold queue (Table Book_Requests)

→ Every transaction status loan and hold queue can be monitored by the application (Table Status)

| Table | Description |
|---------------|---|
| Book_Requests | This table stores details about request book, including their ID, borrow_date, book_detail_id, user_id, status_id, is_taken, and created_by |
| Status | This table stores details about status of every loan and hold queue, including their ID, created_date, name, and created_by |

- **Determine Keys**

| Library | Keys | Description |
|--------------|-------|---|
| id | CK→PK | Unique id that every library data is different |
| created_date | | |
| name | CK→AK | Each library name in every city may be a difference |
| city | | |
| address | | |

| | | |
|------------|--|--|
| created_by | | |
|------------|--|--|

| Users | Keys | Description |
|--------------|-------|---|
| id | CK→PK | Unique ID that every user data is different |
| created_date | | |
| username | CK→AK | The username of each user is a difference |
| fullname | | |
| gender | | |
| email | CK→AK | The email of each user is difference |
| no_telephone | | |
| address | | |
| is_admin | | |
| created_by | | |

| Books | Keys | Description |
|--------------|-------|---|
| id | CK→PK | Unique id that every book data is difference |
| created_date | | |
| title | CK→AK | The title of each book can be many difference |
| category_id | FK | |
| publisher_id | FK | The email of each user is difference |
| year | | |
| description | | |
| created_by | | |

| Library_Books | Keys | Description |
|---------------|-------|---|
| id | CK→PK | Unique id that book in library data is difference |
| created_date | | |
| library_id | FK | |
| book_id | FK | |
| stocks | | |
| quantity | | |
| created_by | | |

| Book_Details | Keys | Description |
|-----------------|-------|--|
| id | CK→PK | Unique id that book details data is difference |
| created_date | | |
| library_book_id | FK | |
| ISBN | CK→AK | Each book has a unique number of ISBN |
| is_available | | |
| created_by | | |

| Categories | Keys | Description |
|--------------|-------|--|
| id | CK→PK | Unique id that category data is difference |
| created_date | | |
| name | | |
| created_by | | |

| Publishers | Keys | Description |
|--------------|-------|--|
| id | CK→PK | Unique id that category data is difference |
| created_date | | |
| name | CK→AK | Each publisher has its own different name. |
| address | | |
| created_by | | |

| Authors | Keys | Description |
|--------------|-------|--|
| id | CK→PK | Unique id that category data is difference |
| created_date | | |
| name | | |
| created_by | | |

| Book_Authors | Keys | Description |
|--------------|-------|---|
| id | CK→PK | Unique id that book author data is difference |
| created_date | | |
| book_id | FK | |
| author_id | FK | |
| created_by | | |

| Borrows | Keys | Description |
|--------------|-------|---|
| id | CK→PK | A unique id that borrows data is a difference |
| created_date | | |
| user_id | FK | |

| | | |
|------------|----|--|
| start_date | | |
| due_date | | |
| status_id | FK | |
| created_by | | |

| Borrow_Details | Keys | Description |
|----------------|-------|--|
| id | CK→PK | A unique id that borrows detailed data is a difference |
| created_date | | |
| borrow_id | FK | |
| book_detail_id | FK | |
| return_date | | |
| status_id | FK | |
| created_by | | |

| Book_Requests | Keys | Description |
|----------------|-------|--|
| id | CK→PK | Unique id that book request data is difference |
| created_date | | |
| borrow_date | | |
| book_detail_id | FK | |
| user_id | FK | |
| status_id | FK | |
| is_taken | | |
| created_by | | |

| Status | Keys | Description |
|--------------|-------|--|
| id | CK→PK | Unique id that book request data is difference |
| created_date | | |
| name | CK→AK | Every status name is difference |
| created_by | | |

➤ Determine Table Relationships

| | Library | Users | Books | Library_Books | Book_Details | Categories | Publishers | Authors | Book_authors | Book_Requests | Borrows | Borrow_details | Status |
|----------------|---------|-------|-------|---------------|--------------|------------|------------|---------|--------------|---------------|---------|----------------|--------|
| Library | | | | 1:N | | | | | | | | | |
| Users | | | | | | | | | | 1:N | 1:N | | |
| Book | | | | 1:N | | 1:1 | 1:1 | | 1:N | | | | |
| Library_Books | N:1 | | N:1 | | 1:N | | | | | | | 1:1 | |
| Book_Details | | | | N:1 | | | | | | | | | |
| Categories | | | 1:1 | | | | | | | | | | |
| Publishers | | | 1:1 | | | | | | | | | | |
| Authors | | | | | | | | | 1:N | | | | |
| Book_author | | | N:1 | | | | | N:1 | | | | | |
| Book_Requests | | N:1 | | | | | | | | | | | 1:1 |
| Borrows | | N:1 | | | | | | | | | | | 1:1 |
| Borrow_details | | | | | | | | | | | | | 1:1 |
| Status | | | | | | | | | | 1:1 | 1:1 | 1:1 | |

➤ Determine Business Rules

1. Table Library

| Library | | |
|----------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "title" varchar(100) | Not Null | |
| "category_id" int4 | | |
| "publisher_id" int4 | | |
| "year" int4 | | |
| "description" text | | |
| "created_by" int4 | | |

2. Table Users

| Users | | |
|----------------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "username" varchar(100) | Not Null | |
| "fullname" varchar(200) | Not Null | |
| "gender" varchar(10) | | |
| "email" varchar(100) | | |
| "no_telephone" varchar(20) | | |
| "address" varchar(200) | | |
| "is_admin" boolean | | |
| "created_by" int4 | | |

3. Table Books

| Books | | |
|----------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "title" varchar(100) | Not Null | |
| "category_id" int4 | | |
| "publisher_id" int4 | | |
| "year" int4 | | |
| "description" text | | |
| "created_by" int4 | | |
| | | |

4. Table Categories

| Categories | | |
|-------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "name" varchar | Not Null | |
| "created_by" int4 | | |

5. Table Publishers

| Publishers | | |
|------------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "name" varchar(100) | Not Null | |
| "address" varchar(100) | | |
| "created_by" int4 | | |

6. Table Authors

| Authors | | |
|---------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "name" varchar(100) | Not Null | |
| "created_by" int4 | | |

7. Table Book Authors

| Book_Authors | | |
|-------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "book_id" int4 | Not Null | FK |
| "author_id" int4 | Not Null | FK |
| "created_by" int4 | | |

8. Table Library Books

| Library_Books | | |
|-------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "library_id" int4 | Not Null | FK |
| "book_id" int4 | Not Null | FK |
| "stocks" int4 | | |
| "created_by" int4 | | |

9. Table Books Details

| Books_Details | | |
|------------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "library_book_id" int4 | | FK |
| "isbn" varchar(100) | | |
| "is_available" boolean | | |
| "created_by" int4 | | |

10. Table Status

| Status | | |
|---|----------|----|
| "id" | Not Null | PK |
| "created_date", "name" varchar(100) "created_by" int4 | Not Null | |

11. Table Borrows

| Borrow | | |
|--------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "user_id" int4, | Not Null | FK |
| "start_date" | | |
| "due_date" | | |
| "status_id" int4, | Not Null | FK |
| "created_by" int4, | | |

12. Table Borrow_Details

| Borrow | | |
|--------------------|----------|----|
| "id" | Not Null | PK |
| "created_date" | | |
| "user_id" int4, | Not Null | FK |
| "start_date" | | |
| "due_date" | | |
| "status_id" int4, | Not Null | FK |
| "created_by" int4, | | |

➤ Implement The Design



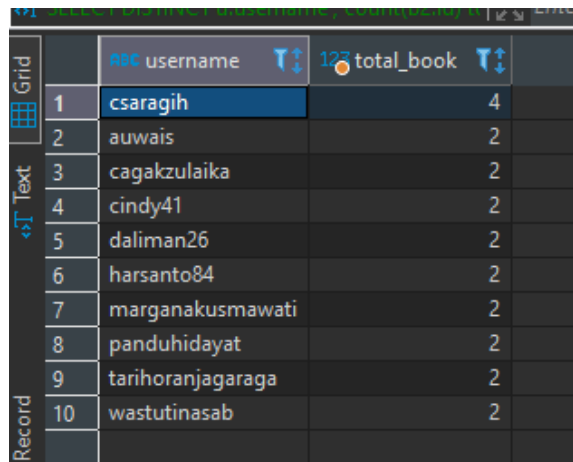
➤ **5 Objectives/questions:**

1. Find the book category that the most borrowed by users?

| | ABC categories | 123 total |
|---|-----------------|-----------|
| 1 | Science Fiction | 6 |
| 2 | Non-Fiction | 4 |
| 3 | Historical | 4 |
| 4 | Fiction | 3 |
| 5 | Romance | 2 |
| 6 | Fantasy | 1 |

```
SELECT
    c."name" categories,
    count(c.id) total
FROM
    "Borrows" b
JOIN "Borrow_Details" bd ON
    b.id = bd.borrow_id
JOIN "Book_Details" bd2 ON
    bd2.id = bd.book_detail_id
JOIN "Library_Books" lb ON
    lb.id = bd2.library_book_id
JOIN "Books" b2 ON
    b2.id = lb.book_id
JOIN "Categories" c ON
    c.id = b2.category_id
GROUP BY
    c."name"
ORDER BY
    count(c.id) DESC
```

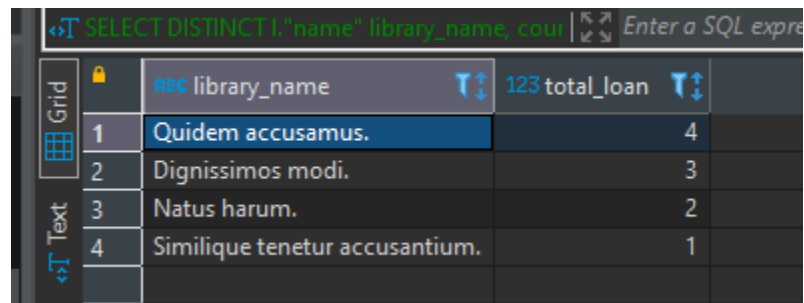
2. Find username that has the highest number of loans in the application?



| | username | total_book |
|----|-------------------|------------|
| 1 | csaragih | 4 |
| 2 | auwais | 2 |
| 3 | cagakzulaika | 2 |
| 4 | cindy41 | 2 |
| 5 | daliman26 | 2 |
| 6 | harsanto84 | 2 |
| 7 | marganakusmawati | 2 |
| 8 | panduhidayat | 2 |
| 9 | tarihoranjagaraga | 2 |
| 10 | wastutinasab | 2 |

```
SELECT
DISTINCT
    u.username ,
    count(b2.id) total_book
FROM
    "Borrows" b
JOIN "Borrow_Details" bd ON
    b.id = bd.borrow_id
JOIN "Book_Details" bd2 ON
    bd2.id = bd.book_detail_id
JOIN "Library_Books" lb ON
    lb.id = bd2.library_book_id
JOIN "Books" b2 ON
    b2.id = lb.book_id
JOIN "Users" u ON u.id = b.user_id
GROUP BY
    u.username
ORDER BY
    count(b2.id) DESC
```

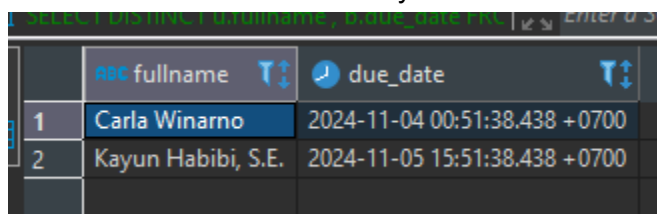
3. Calculate the number of books still on loan for each library?



| | asc library_name | 123 total_loan |
|---|--------------------------------|----------------|
| 1 | Quidem accusamus. | 4 |
| 2 | Dignissimos modi. | 3 |
| 3 | Natus harum. | 2 |
| 4 | Similique tenetur accusantium. | 1 |

```
SELECT
DISTINCT
    l."name" library_name,
    count(b2.id) total_loan
FROM
    "Borrows" b
JOIN "Borrow_Details" bd ON
    b.id = bd.borrow_id
JOIN "Book_Details" bd2 ON
    bd2.id = bd.book_detail_id
JOIN "Library_Books" lb ON
    lb.id = bd2.library_book_id
JOIN "Books" b2 ON
    b2.id = lb.book_id
JOIN "Library" l ON l.id = lb.library_id
WHERE b.status_id = 2
GROUP BY
    l."name"
ORDER BY
    count(b2.id) DESC
```

4. Display users who have not returned books by the due date set



| | asc fullname | due_date |
|---|--------------------|-------------------------------|
| 1 | Carla Winarno | 2024-11-04 00:51:38.438 +0700 |
| 2 | Kayun Habibi, S.E. | 2024-11-05 15:51:38.438 +0700 |

```

SELECT
    DISTINCT
    u.fullname ,
    b.due_date
FROM
    "Borrows" b
JOIN "Borrow_Details" bd ON
    b.id = bd.borrow_id
JOIN "Book_Details" bd2 ON
    bd2.id = bd.book_detail_id
JOIN "Library_Books" lb ON
    lb.id = bd2.library_book_id
JOIN "Books" b2 ON
    b2.id = lb.book_id
JOIN "Users" u ON u.id = b.user_id
WHERE b.status_id = 4

```

5. show users who are still waiting to borrow books and what books will be borrowed

| | abc fullname | abc book_title | abc isbn | |
|---|-------------------------------|--------------------|-------------------|--|
| 1 | Ida Safitri | Doloremque itaque. | 978-1-359-03185-3 | |
| 2 | R.M. Nardi Manullang, S.I.Kom | Ducimus. | 978-0-10-765817-5 | |
| 3 | Kayun Habibi, S.E. | Voluptatum unde. | 978-1-154-07985-2 | |

```

SELECT
    u.fullname,
    b.title book_title,
    bd.isbn
FROM
    "Book_Requests" br
JOIN "Book_Details" bd ON
    bd.id = br.book_detail_id
JOIN "Library_Books" lb ON
    lb.id = bd.library_book_id
JOIN "Books" b ON
    b.id = lb.book_id
JOIN "Users" u ON
    u.id = br.user_id
WHERE
    br.is_taken IS NOT TRUE

```


Reference:

- [Designing a RDB and Creating an ERD - Medium](#)
- [Create and Query a Relational Database - Medium](#)