CSP362: Database Management Systems - Assignment 1

SAHIL - 2016UCS0008

February 24, 2019

1 Introduction

In this assignment, an ER diagram for library is provided. We have to create a database for **Library Management System** in *mysql* for deployment at **IIT Jammu**.

2 Entities and Relationships

The color scheme followed to list the attributes is as follows:

- blue for **Primary Key**.
- red for **Foreign Key**.
- green for new attributes which are described in later sections.

- The following entities are required:
  - **Book**
    - book_id
    - title
    - year
    - isbn
    - pages
    - user_id
    - add_time
    - publisher_id
    - issue_time
  - **User**

    - user_id
    - name
    - username
    - password
    - email
    - user_type
- **Periodical**
    - periodical_id
    - title
    - year
    - volume
    - isbn
    - user_id
    - publisher_id
    - add_time
    - issue_time
- **Author**
    - author_id
    - name
- **Paper**
    - paper_id
    - name
    - periodical_id
- **Publisher**
    - publisher_id
    - name
- **Tag**
    - tag_id
    - value
- **Message**
    - message_id
    - text
    - user_id

    - time
  - **Issue_Capacity**
    - user_type
    - max_books
    - max_time

- The following relationships are required:
  - **Book_and_Author**
    - book_id
    - author_id
  - **Paper_and_Author**
    - paper_id
    - author_id
  - **Book_and_Tag**
    - book_id
    - tag_id
  - **Periodical_and_Tag**
    - periodical_id
    - tag_id
  - **Book_and_Discipline**
    - book_id
    - discipline
  - **Borrow_Book**
    - borrow_id
    - book_id
    - user_id
    - issue_time
    - return_time
    - due_amount
  - **Borrow_Periodical**
    - borrow_id
    - periodical_id
    - user_id
    - issue_time
    - return_time
    - due_amount

3 Normalizing the database upto BCNF

3.1 1NF

The database design is already in **first normal form** since there is no field which is multi-valued.

3.2 2NF

It is already in **second normal form** as there is no partial dependency of any non-key attribute on any key attribute. This is because all entity tables have single attribute as key, and for the relationship tables, there are only two attributes which themselves are part of the key.

3.3 3NF

To remove the transitive dependencies in the **User** table,

$$user\_id \rightarrow user\_type$$

$$user\_type \rightarrow \{max\_books, max\_time\}$$

A new table **Issue_Capacity** has been created. So, the database is now in **3NF**.

3.4 BCNF

It is already in BCNF since for all functional dependencies $\mathbf{A} \rightarrow \mathbf{B}$, A cannot be a non-prime attribute if B is a prime attribute.

4 Implementing the tables in DBMS

4.1 Inserting at least 100 records for Book and User tables:

Dummy records have been inserted in all the tables using the python library **faker**.

All the code for this is available in *src/generate_data.py* file.

4.2 Modifying User table to add an attribute user_type:

This modification has been done as mentioned in green.

4.3 Maximum capacity of books and time of issuing:

This constraint has been ensured by fixing max_books and max_time in the table **Issue_Capacity**.

5 Performing Queries on LMS

All the following queries have been performed and are available in *src/queries.py* file.

- Listing all the tables and their attributes
- Displaying total inventory in the LMS
- Listing the no. of available books requested by a user
- Listing the author(s) of a given book
- Listing the total no. of books issued for a user
- Checking whether a user is allowed to borrow a book or not
- Listing the no. of books issued and returned on a daily basis (for a given day/period)
- Listing the users with book details if there are any dues
- Listing the newly added book records for a given period
- Displaying the user name and the books issued to them
- Displaying all the books with all the details issued to a user

6 Queries in Relational Algebra Notation

The relational algebra notation for each of the following queries is provided below:

- Listing all the tables and their attributes

$$\sigma_{table\_schema='lms'}(information\_schema.columns)$$

- Displaying total inventory in the LMS

$$G_{count(*)}(table\_name)$$

- Listing the no. of available books requested by a user

$$G_{count(*)}(\sigma_{title='book\_name'}(Book))$$

- Listing the author(s) of a given

$$\pi_{Author.*}(\sigma_{\substack{Book\_and\_Author.book\_id=Book.book\_id \\ \wedge Book\_and\_Author.author\_id=Author.author\_id \\ \wedge Book.title='book\_name'}}(Author \times Book\_and\_Author \times Book))$$

- Listing the total no. of books issued for a user

$$G_{count(*)}(\sigma_{Book.user\_id=User.user\_id \wedge User.name='user\_name'}(Book))$$

- Checking whether a user is allowed to borrow a book or not

$$num\_issued \leftarrow G_{count(*)}(\sigma_{Book.user\_id=User.user\_id \wedge User.name='user\_name'}(Book))$$

$$max\_allowed \leftarrow \pi_{max\_books}(\sigma_{\substack{User.user\_type\ =\ Issue\_Capacity.user\_type \\ \wedge\ User.name='user\_name'}}(Issue\_Capacity \times User))$$

$$\sigma(if(num\_issued < max\_allowed, "Allowed", "Not\ allowed"))$$

- Listing the no. of books issued and returned on a daily basis (for a given day/period)

$$G_{count(*)}(\sigma_{\substack{issue\_time>='start\_daytime' \\ \wedge(return\_time<='end\_daytime' \vee return\_time\ is\ NULL)}}(Borrow\_Book))$$

- Listing the users with book details if there are any dues

$$\pi_{User.user\_id,\ name}(\sigma_{\substack{User.user\_id=Borrow\_Book.user\_id \\ \wedge\ due\_amount\ >\ 0}}(User \times Borrow\_Book))$$

- Listing the newly added book records for a given period

$$\sigma_{\substack{add\_time\ >=\ 'start\_daytime' \\ \wedge\ add\_time\ <=\ 'end\_daytime'}}(Book)$$

- Displaying the user name and the books issued to them

$$\pi_{Book.user\_id,\ User.name,\ book\_id,\ title}(\sigma_{User.user\_id=Book.user\_id}(Book \times User))$$

- Displaying all the books with all the details issued to a user

$$\pi_{Book.*}(\sigma_{\substack{User.user\_id\ =\ Book.user\_id \\ \wedge User.name\ =\ 'user\_name'}}(Book \times User))$$

7 Computing the fine for users and showing all users having dues

The fine to be imposed on a user after the due date is **Rs 1**/day excluding Saturdays and Sundays.

8 Modifying Book table to add the attribute 'Discipline'

Instead of modfiying the book table to add discipline since one book can bu multi-disciplinary, a new table **Book_and_Discipline** has been created.

9 Developing GUI for LMS in Python

The GUI is to be developed using either **PyQt** or **Tkinter**.

10 Reserving the issued books

Once all the requested books are issued, they need to be reserved so that whenever the books is returned, it should be issued to the requested user.