**CS504 FINAL PROJECT REPORT**

KEERTHI REDDY CHINTHA REDDY

G01394014

Prof. Binqian Yin

Principles of Data Management and Mining

George Mason University

**DATABASE MANAGEMENT SYSTEM FOR A PUBLIC LIBRARY:**

**DATABASE DESIGN:**

**SCOPE OF THE PROJECT:**

The goal of the project is to design and set up a database management system for a public library so that it may better manage its resources and serve its patrons. The system will also maintain track of data on library items including books, magazines, e-books, and audiobooks in addition to member data like names, contact information, and borrowing history. The technology will facilitate borrowing by letting users check out items and provide the information library staff needs to manage the circulation of library materials. The project seeks to boost the effectiveness of the library and ensure data integrity while minimizing redundant tasks for both library personnel and consumers. The system will generate reports on library usage, popular titles, and other essential statistics to assist library staff in making knowledgeable decisions about resource management and resource acquisition.

**ENTITIES AND RELATIONSHIPS:**

The project's entities and relationships are listed below.

* ENTITIES:

1. Material:

Constraints: Material\_ID is the primary key and Catalog\_id, Genre\_id are foreign keys.

1. Catalog:

Constraints: Catalog\_ID is the primary key.

1. Genre:

Constraints: Genre\_ID is the primary key.

1. Borrow:

Constraints: Borrow\_ID is the primary key and Material\_ID, Member\_ID, Staff\_ID are the foreign keys.

1. Author:

Constraints: Author\_ID is the primary key.

1. Member:

Constraints: Member\_ID is the primary key.

1. Staff:

Constraints: Staff\_ID is the primary key.

* RELATIONSHIPS:

Many to Many relation:

* Authorship acts as relation and participating fully between Material and Author with Authorship\_ID as its primary attribute.

One to Many and Many to One relations:

* One catalog can have many materials.
* Each material belongs to one catalog.
* One genre can have many materials.
* Each material belongs to one genre.
* Each borrow record belongs to one member and one material.
* Each member can have many borrow records.
* Each material can have many borrow records.
* One author can create many materials.
* Each material can be created by one author.
* Each borrow record belongs to one member.
* Each member can have many borrow records.
* Each borrow record is processed by one staff member.
* Each staff member can process many borrow records.

**ER DIAGRAM:**

**Diagram

Description automatically generated**

**SCHEMA:**

**Diagram

Description automatically generated**

**QUERIES/UPDATES:**

1. Which materials are currently available in the library?
2. Query:

select m.material\_id, m.title

from Material m

where material\_id NOT IN (

select material\_id

from Borrow

where Return\_Date IS NULL

);

Output:

Table

Description automatically generated

1. Which materials are currently overdue? Suppose today is 04/01/2023 and show the borrow date and due date of each material.
2. Query:

select m.material\_id, m.title, b.borrow\_date, b.due\_date

from Material m

inner join Borrow b on m.material\_id = b.material\_id

where b.return\_date is null and b.due\_date < '2023-04-01'

order by material\_id;

Output:

Table

Description automatically generated

1. What are the top 10 most borrowed materials in the library? Show the title of each material and order them based on their available counts.
2. Query:

select material.material\_id,title, count(borrow.material\_id) as count

from material, borrow

where material.material\_id= borrow.material\_id

and borrow.material\_id

not in (select borrow.material\_id from borrow where return\_date is NULL)

group by material.material\_id,title

order by count desc

Output:

Table

Description automatically generated

1. How many books has the author Lucas Piki written?
2. Query:

select count(material\_id) as booksWritten

from Author as a, Authorship as au

where a.name = 'Lucas Piki' and a.author\_id = au.author\_id;

Output:

Graphical user interface, text, application

Description automatically generated

1. How many books were written by two or more authors?
2. Query:

select count(distinct Material\_ID) from Authorship

where Material\_ID in (select Material\_ID from authorship

group by Material\_ID

having count(Author\_ID) > 1);

Output:

Table

Description automatically generated

1. What are the most popular genres in the library?
2. Query:

SELECT g.Name, COUNT(\*) AS Borrow\_Count

FROM Genre g

JOIN Material m

ON g.Genre\_ID = m.Genre\_ID

JOIN Borrow b

ON m.Material\_ID = b.Material\_ID

GROUP BY g.Genre\_ID

ORDER BY Borrow\_Count DESC;

Output:

Table

Description automatically generated

1. How many materials have been borrowed from 09/2020-10/2020?
2. Query:

SELECT COUNT(Material\_ID) AS borrowed\_count

FROM Borrow

WHERE Borrow\_Date BETWEEN '2020-09-02' AND '2020-09-30';

Output:

Graphical user interface, text

Description automatically generated

1. How do you update the “Harry Potter and the Philosopher's Stone” when it is returned on 04/01/2023?
2. Query:

update Borrow

set Return\_Date = '2023-04-01'

where material\_id in (

select material\_id

from Material

where title = 'Harry Potter and the Philosopher''s Stone')

Output:

Graphical user interface, text

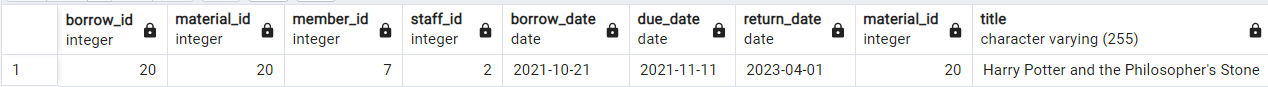
Description automatically generated with medium confidence

Query:

select \* from borrow,Material where title = 'Harry Potter and the Philosopher''s Stone'

and Material.Material\_id= borrow.Material\_id

Output:



1. How do you delete the member Emily Miller and all her related records from the database?
2. Query:

DELETE FROM Member WHERE Name = 'Emily Miller'

Output:

Text

Description automatically generated with medium confidence

delete FROM Borrow WHERE member\_id = '5';

Output:

Text

Description automatically generated with medium confidence

1. How do you add the following material to the database? Title: New book Date: 2020-08-01 Catalog: E-Books Genre: Mystery & Thriller Author: Lucas Pipi
2. Query:

INSERT INTO Material (material\_id,title, publication\_date, catalog\_id, genre\_id)

VALUES (32,

'New book',

'2020-08-01',

(SELECT Catalog.Catalog\_ID FROM Catalog WHERE Catalog.Name = 'E-Books'),

(SELECT Genre.Genre\_ID FROM Genre WHERE Genre.Name = 'Mystery & Thriller')

);

Output:

Text

Description automatically generated with medium confidence

Query;

INSERT INTO Authorship( authorship\_id, author\_id,material\_id)

values( 34,

(select author\_id from Author where name='Lucas Piki'),

(select material\_id from material where title = 'New book'));

Output:

Text

Description automatically generated with medium confidence

**Design Questions:**

* Alert staff about overdue materials on a daily basis?

1. It is possible to build a trigger called "daily\_alert" that runs every day following an INSERT operation on the "Borrow" Table in order to notify workers about overdue materials. A function called alert\_staff was created beforehand, and within of it, a notice alert could be issued to the staff informing them of the overdue, and the trigger operation could then be returned to the alert\_staff function.Automatically deactivate the membership based on the member’s overdue occurrence (>= three times). And reactivate the membership once the member pays the overdue fee.

* Automatically deactivate the membership based on the member’s overdue occurrence (>= three times). And reactivate the membership once the member pays the overdue fee.

1. In order to keep track of member overdue occurrences and automatically deactivate memberships, when necessary, a new table should be created. Additionally, a trigger function can be created to monitor this new table and deactivate a member's membership status if their overdue occurrence count is greater than or equal to 3. To reactivate a member's membership status after they have paid off their overdue fees, a separate trigger can be created to automatically reactivate their membership.

References:

PostgreSQL official documentation for SQL commands: PostgreSQL Documentation. (2023, February 9). SQL Commands. <https://www.postgresql.org/docs/current/sql-commands.html>

PostgreSQL tutorial on tutorialspoint: tutorialspoint. (n.d.). PostgreSQL Tutorial. <https://www.tutorialspoint.com/postgresql/index.htm>

PostgreSQL tutorial on SQLShack: Shrestha, A. (2021, March 5). PostgreSQL tutorial for beginners. SQLShack. <https://www.sqlshack.com/postgresql-tutorial-for-beginners/>

"SQL for Data Analysis" tutorial on Mode Analytics: Mode Analytics. (n.d.). SQL for Data Analysis. <https://mode.com/sql-tutorial/>