Archives of Babel

By: Ryan Reid, Sean Copp, Amy Chen

Project Motivation

The problem being addressed is:

- Want to be able to organize the library (storage of locational information for books, dvds)
- Need better storage of transactional data for the library (returns, loans)
- Database will provide quicker access to book locations and customer/employee information
- Need to keep track of inventory so that librarians can see what the library needs more copies of and what are the most popular books

System Description

System usage and reasons the database solves the issues :

- Organizes data and allows members/librarians to access relevant information to them
  - Members can see the most borrowed books recently, while librarians can see important transactional information
- Utilizes the search bar to allow members and librarians to search through library in ways that solve their problems
  - Members and librarians find the information that is searched for
  - Members don't have to scan through a map to find where materials are stored
- Provides employees inventory information on books and DVDs
  - Provides answers for members if members aren't able to find the materials at the materials' locations

System Architecture

- SQLite as a DBMS
  - Cross-platform; no new hardware required in the library
  - Fast and serverless
  - Database can be stored on computers throughout the library which can be accessed by both members and staff of the library
- Front-End written in Java
  - Graphical User Interface (GUI) written in Java using Swing, which is the built in Java GUI toolkit
- How does everything “talk”?
  - Java Database Connectivity (JDBC) is an application programming interface (API) which allows a Java application to connect to the database, send queries, and receive the results

Database Design - Data Source

Sources for table:

- Tables that have randomly generated data:
  - Member
  - Employee
  - Location
  - Loan
  - Return
  - BookLine
  - DVDLine
- Data obtained from the Thomas Crane Public Library website (located in Quincy): (links are provided in the reference slide)
  - Book
  - DVD

Reports - Query #1

```
/*
    Lists all employees by first name and last name that lent out books
    and how many books they each lent out. Ordered by amount of books
    lent out. This query helps management see which employees have been
    most productive.

    6-pt Query

    1 pt - Motivation/Justification
    2-pts - >2 tables joined
    1-pt - Aggregate function
    1-pt - Grouping
    1-pt - Ordering fields

*/

SELECT
    Employee.EmployeeID AS ID, Employee.FirstName AS First_Name, Employee.LastName AS Last_Name,
    COUNT(BookLine.BookLineID) as Books_Lent
FROM
    (Employee INNER JOIN Loan ON Employee.EmployeeID=Loan.LibrarianID)
    INNER JOIN BookLine ON Loan.LoanID=BookLine.LoanID
GROUP BY
    Employee.EmployeeID
ORDER BY Books_Lent DESC;
```

Reports - Query #1 (Output)

Lists all employee IDs, first names, last names, and how many books they have lent out, ordered from greatest to least.

The report generated tells which employees have been able to loan out the most books.

	ID	First_Name	Last_Name	Books_Lent
1	8	alejandro	osborne	21
2	16	alan	thomas	17
3	11	louise	howard	15
4	7	francis	hernandez	11
5	12	debra	bennett	9
6	5	acora	gray	8
7	1	fernando	stephens	6
8	4	angel	james	6
9	18	roger	king	6
10	15	rose	moore	5
11	17	ashley	russell	5
12	10	randall	sullivan	4
13	9	kay	reid	3
14	14	william	kelly	3
15	19	pamela	washington	3
16	13	juan	bailey	2
17	2	raquel	hunt	1

Reports - Query #2

This query allows employees to find where a book that has been loaned out was located in the library

Benefit:

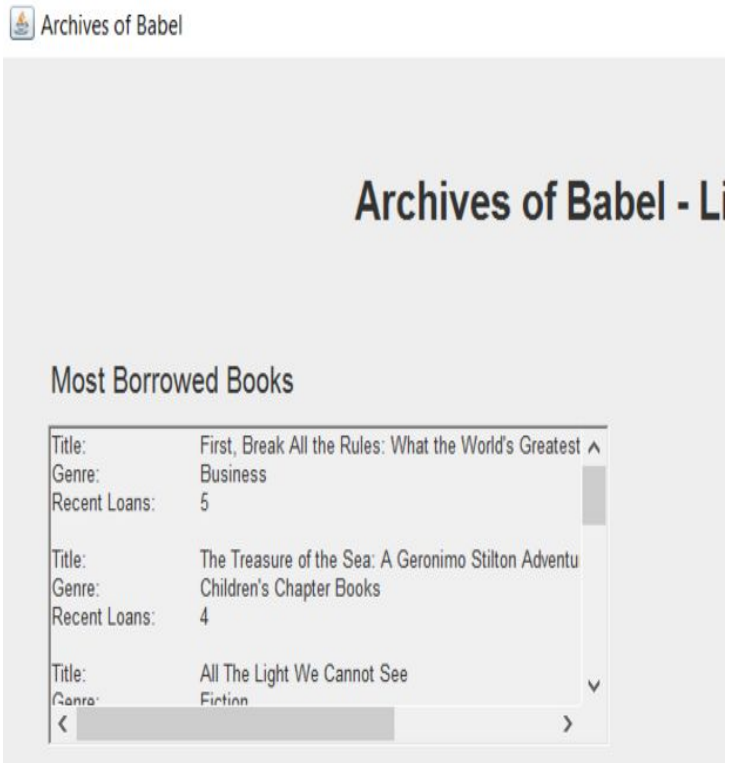
- Helps employees keep better inventory records and organize the library much more easily

```
/*
    Where was the book located in the library that Kendall
    Jenner took out on September 1st, 2016?

    4-pt Query

    1-pt - Motivation/Justification - When a book gets returned, an
    employee can usethis query to place the book back on the proper shelf
    2-pts - >2 tables joined
    1-pt - where/having conditions not for joins
*/

SELECT
    Book.Title AS Book_Title,
    Location.Section AS Section, Location.RowNumber AS Row, Location.Shelf AS Shelf
FROM
    (((Member INNER JOIN Loan ON Member.MemberID=Loan.MemberID)
    INNER JOIN BookLine ON Loan.LoanID=BookLine.LoanID)
    INNER JOIN Book ON BookLine.BookID=Book.BookID)
    INNER JOIN Location ON Book.LocationID=Location.LocationID
WHERE
    Member.FirstName LIKE 'Kendall' AND
    Member.LastName LIKE 'Jenner' AND
    Loan.LoanDate LIKE '2016-09-01';
```

	Book_Title	Section	Row	Shelf
1	The Magic of Gingerbread: 16 Beautiful Pr...	8	4	7

```
Query executed successfully: /*
        Where was the book located in the library that Kendall
        Jenner took out on September 1st, 2016?
```

Reports - Query #3

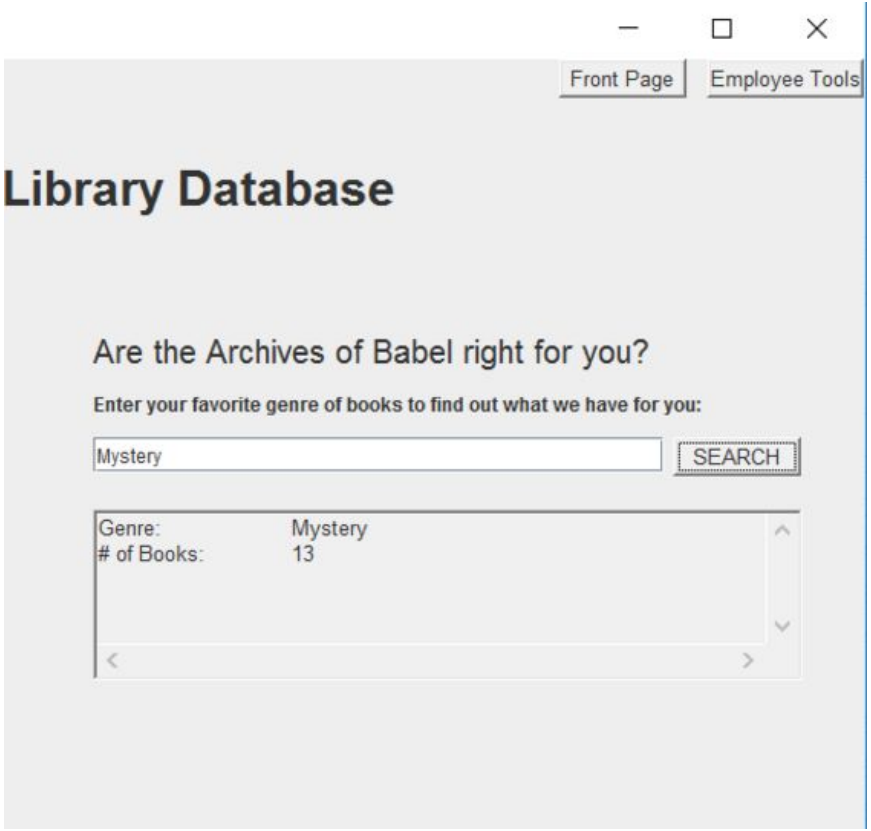


This query shows a list of the books that have been loaned out the most by using the data from the loan entity.

Benefits:

- Quick feeling to what the library is about
- Helps librarians determine what books the inventory should expand on
- Helps the user determine if Archive of Babels is the right database for them to use

Reports - Query #4



This query helps the user see how many of the books Archives of Babel has for a specific genre.

Benefits:

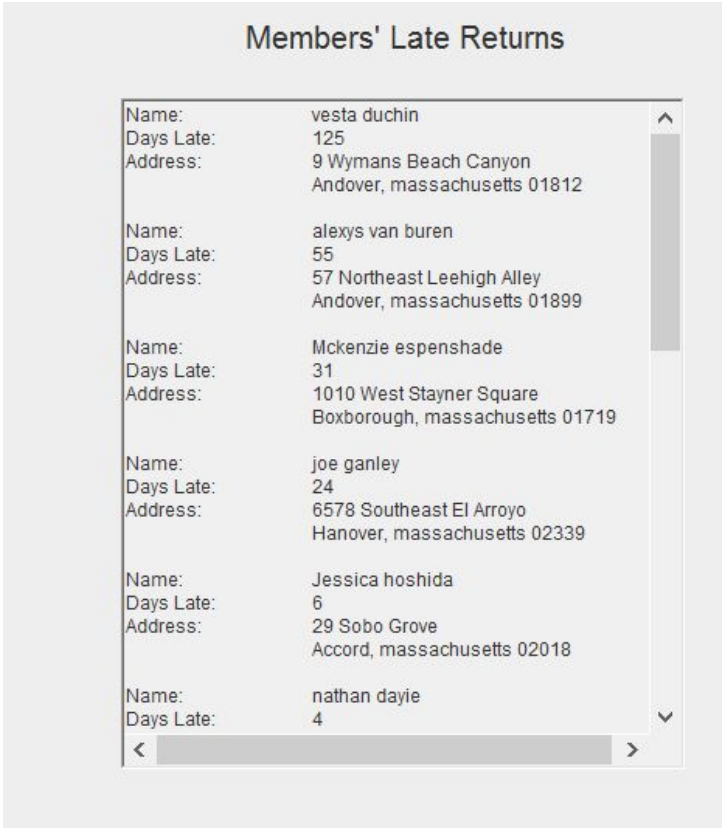
- Shows a member if the library has a large selection of their favorite genre
- Quick feeling to what the library is about
- Helps librarians determine what books the inventory should expand on
- Helps the user determine if Archive of Babels is the right database for them to use

Reports - Query #5

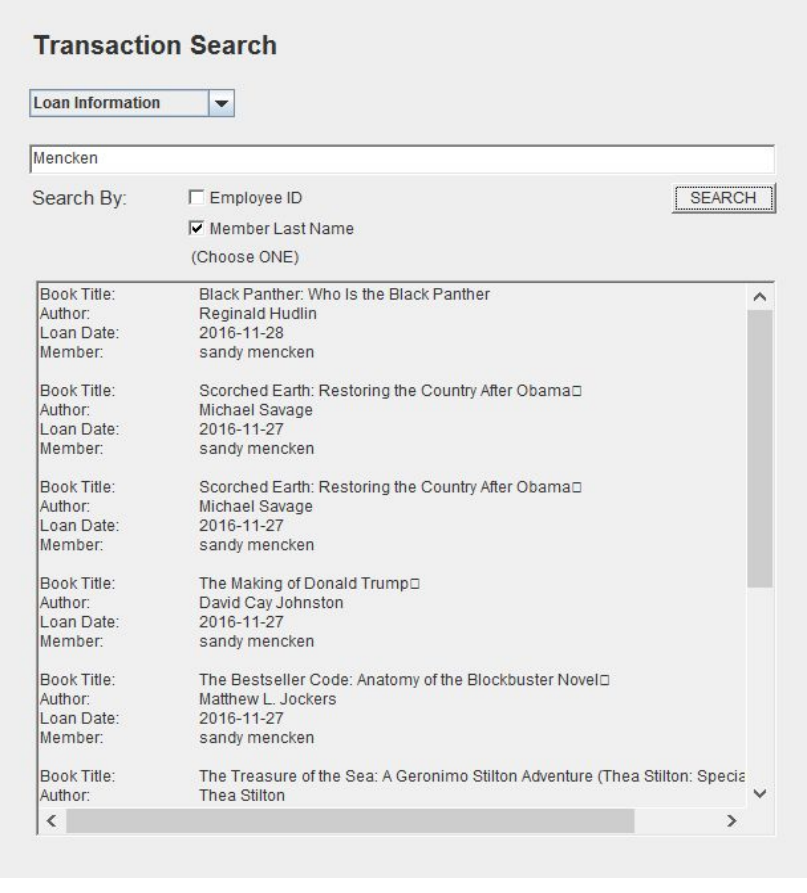
This query runs upon opening the Employee section of the user-interface which gives employees the member's address information so a reminder letter can be sent out for overdue books/DVDs.

Benefits:

- Employees are able to see which members have frequently been late with their returns
- Helps librarians determine who to send our reminder letters to about overdue books/DVDS



Reports - Query #6



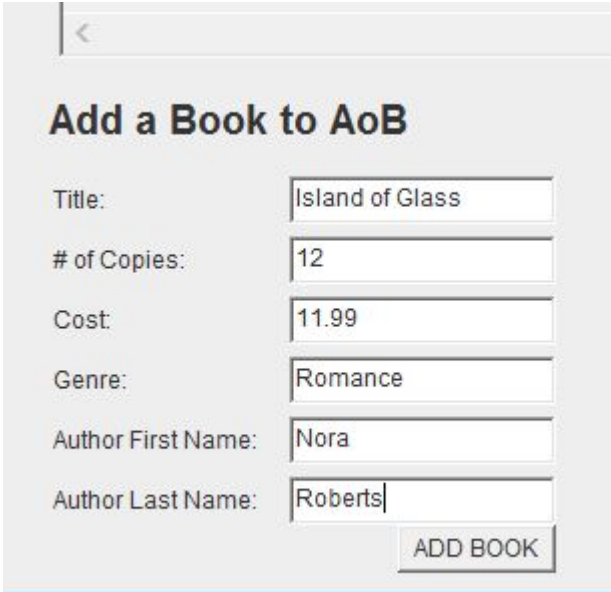
This query shows a search that an employee might do to find out loans of a particular member, by searching by the member's last name.

Benefits:

- Employees checking out past transactions
- Member requesting more information about a particular loan or return

Reports - Query #7

Gives employee users the ability to add books to the database. The query run in the backend automatically updates all corresponding tables with foreign keys.

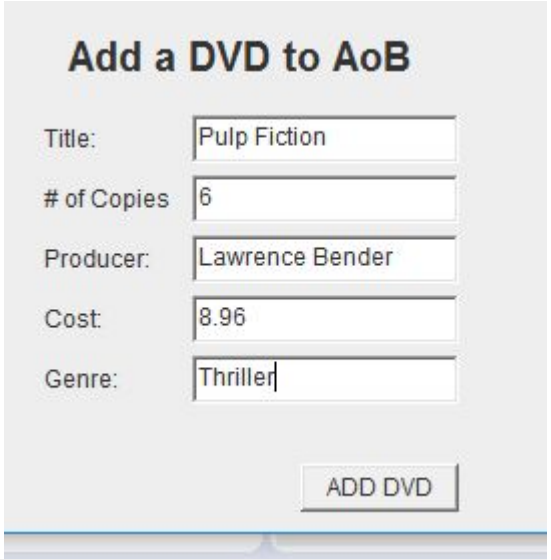


Benefits:

-Gives employees the ability to add new books to the library's database as they are added to the actual physical library

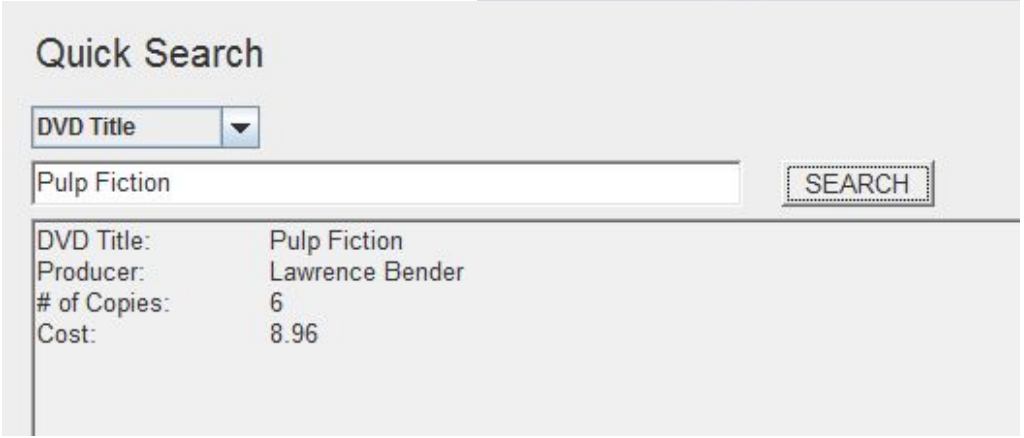
Reports - Query #8

Gives employee users the ability to add a DVD to the database. Query automatically adds foreign keys to the corresponding tables in the backend.



Benefits:

-Gives employees the ability to add new DVDs to the library's database as they are added to the actual physical library



References

- DVD data: http://thomascranelibrary.org/movies/new
- Book data: http://thomascranelibrary.org/books/new