IST 659: Database Management and Concepts

Professor Michael Fudge / Professor Gregory Zink

Library Management System Design

Hernan Orozco

Bailey-Ann Slaughter

Brandon Stanyer

Kory Wakeman

**High-Level Overview:**

The Sim County Library is updating and digitizing its long-time analog bookkeeping system to align with the evolving technological landscape and the changing needs of modern libraries. The library cites issues with the traditional paper-based bookkeeping system as being time-consuming and error-prone. Implementing a digital solution would help automate routine tasks, which will free up staff time typically spent on these tasks to instead focus on increased customer service and customer experience. The transition to a digital management system would also reduce the physical storage required to store this necessary information and allow the library to expand its inventory. With this system, the library will also be able to analyze trends and patterns to execute data-driven strategic planning and proper resource allocation.

**Goal/High-Level Business Requirements:**

The goal for transitioning this paper-based library management system to a digital solution is to create a seamless, simple, and user-friendly platform that staff and patrons are quick to master and recognize its time-saving and efficient behaviors.

Requirements:

* Develop a user interface that is intuitive and easy to navigate.
* Prioritize user experience to ensure both staff and patrons.
* Successfully migrate entire existing paper-based records into the database.
* Implement validation checks to ensure data integrity during migration.
* Create a reporting module that generates accurate and insightful reports for management and decision-making.
* Design the system architecture to be scalable, accommodating future growth and technological advancements.
* Communicate the transition plan, benefits, and impact of the digital system to staff and patrons.
* Address concerns and provide regular updates throughout the transition process.
* Solicit feedback from stakeholders and make necessary adjustments based on their input.

**Glossary:**

* **Library management system**: A software application designed to manage and automate various tasks in a library, such as cataloging, circulation, and patron management.
* **Cataloging**: The process of adding and maintaining records for library materials, including books, journals, and multimedia items.
* **ISBN**: International Standard Book Number; a numeric commercial book identifier that is intended to be unique. An ISBN is assigned to each separate edition and variation of a publication.
* **Database**: A structured collection of data stored electronically, typically organized into tables, rows, and columns.

**Stakeholders:**

* **Library staff:** Librarians and library administrators will be the primary stakeholders and use the database daily to manage library operations, along with the staff responsible for entering and maintaining data in the database, including adding new materials and updating records.
* **Patrons:** Library users who benefit from a user-friendly and efficient system for searching and accessing library materials.
* **Library Board or Trustees:** The governing body of the library, which may have financial and strategic interests in the database's development.
* **Financial Department:** Individuals responsible for budgeting and financial planning, as database development may have cost implications.
* **Marketing Team:** Professionals responsible for promoting and educating patrons about the new database and its features.
* **Software Developers:** Developers who create the software interface that interacts with the SQL database, including the front-end (UI) and back-end (database connectivity).

**Business Rules:**

* Transfer existing paper-based records to digital format.
* Ensure complete accuracy and integrity during data migration of existing catalog.
* Every item in the library collection must have a unique identifier (ISBN).
* Patrons must have a valid library card to borrow an item.
* Each patron has a corresponding mailing address.
* Books will have one or more authors.
* Books will have one or more genres.
* Library items will be borrowed for a finite amount of time.
* Monetary fines will be issued to patrons for items not returned on time.

**Conceptual Model:**

**A screenshot of a computer

Description automatically generated**

**Logical Model:**

**A screenshot of a computer flowchart

Description automatically generated**

**Physical Model:**

**A computer screen shot of a computer

Description automatically generated**

**Data Questions and Answers:**

* What is the current state of the library's collection and inventory?
  + The SQL database now provides real-time data on the library's collection, including the number of books, their titles, authors, current availability, and overdue book status. This information is continuously updated as books are returned and borrowed.
* How has the transition affected the library staff's efficiency?
  + The digital system has streamlined cataloging, check-in/check-out, and inventory management processes, reducing the time and effort required for these tasks. Staff can now focus more on patron assistance and library services.
* How has data accuracy and integrity improved with the SQL database?
  + Data accuracy has significantly improved because the SQL database enforces data validation rules, reducing errors caused by manual data entry. The system also maintains an audit trail of all transactions, enhancing data integrity.
* What is the status of overdue books and fines management?
  + The new SQL system has enabled automatic overdue notifications and fine calculations, resulting in more efficient management of overdue books and timely fine collection.
* How has staff training and patron support been handled during and after the transition?
  + Staff members received training on the new system in lieu of their normal manual data entry tasks, and specific members of the staff with higher comfort on the new systems have been established to assist patrons with any questions or issues they may encounter as a replaced duty to library card management.

**Sample Input/Output:**

Example Input: (‘00123’, ‘1234 Tree Dr’, ‘San Diego, CA 45090’, ‘joesmith@gmail.com’, 1237857645, ‘01/23/1978’, ‘Joe’, ‘Smith’)

Library Card Mockup:

A card with a face and text

Description automatically generatedA screenshot of a computer

Description automatically generated

**Summary/Conclusion:**

Our team project was rooted in a major transformation from a traditional paper-based library management system to a modern and efficient SQL database-driven solution. This upgrade aimed to streamline library operations, enhance user experiences, and improve data management. Despite not being library science majors ourselves, our lifelong experience with libraries at our ages as we watched them transition from quasi-digital to fully digital made it easier for us to conceptualize the requirements of a database upgrade to this type of paper-based system in a tangible and logical way.

**Reflection:**

This project highlighted the significance of technology in modernizing and improving library management. The transition from a paper-based system to an SQL database-driven system was a necessary step to keep up with the evolving needs and expectations of library users. We learned several valuable lessons throughout the process:

* Collaboration: Successful projects like this require strong collaboration among team members, and each brought our unique skills and expertise to the table which guaranteed success of the project.
* Adaptability: Most of our team learned concepts as the class progressed and adapted our approaches on the project after the first half of the course and its concepts passed. Embracing change and learning new technologies on-the-fly is essential in today's digital age.
* User-Centered Design: Prioritizing the needs of library patrons was critical in our conceptual design of the library management system. We ensured that the new system was intuitive and met user expectations.
* Data Integrity: Maintaining data accuracy and security is paramount when handling sensitive information. Regular data backups and security measures will be essential and follow-on support required if this project existed in the real world is evident.

**Team Log:**

* 1 AUG 23: Team 5 group formation; open communication established through Discord
  + Ideas between library management system and car mechanic administration idea debated across the following week
* 9 AUG 23: Rough outline assembled by team for library management design
  + Logical model sketch developed during group call
* 15 AUG 23: Group call for logical model design; required tables established
* 21 AUG 23: Library management system design proposed to professor
* 23 AUG 23: Library management system design approved
* 29 AUG 23: Final paper and PowerPoint initiated concurrently
  + Code syntax discussed and clarified after live lecture
  + Logical model finalized
* 30 AUG 23: SQL code for final project sent to team for review
* 4 SEP 23: SQL code accepted by all team members; finalized

**SQL Code:**

-- Create Database

CREATE LibraryDatabase;

USE LibraryDatabase;

-- Drop tables if they exist

DROP TABLE IF EXISTS patrons

DROP TABLE IF EXISTS adresses;

DROP TABLE IF EXISTS transactions;

DROP TABLE IF EXISTS transaction\_books;

DROP TABLE IF EXISTS books;

DROP TABLE IF EXISTS authors;

DROP TABLE IF EXISTS book\_authors;

DROP TABLE IF EXISTS genres;

DROP TABLE IF EXISTS book\_genres;

DROP TABLE IF EXISTS publishers;

DROP TABLE IF EXISTS fines;

-- Adding in Tables

-- Patrons Table

CREATE TABLE Patron (

patron\_id INT IDENTITY(1,1) PRIMARY KEY,

patron\_first\_name VARCHAR(50) NOT NULL,

patron\_last\_name VARCHAR(50) NOT NULL,

patron\_address\_id VARCHAR(50),

patron\_date\_of\_birth DATE,

patron\_phone\_number CHAR(10),

patron\_email VARCHAR(100)

CONSTRAINT fk\_addresses\_address\_id FOREIGN KEY (patron\_address\_id) REFERENCES addresses(address\_id)

);

-- Addresses Table

CREATE TABLE addresses (

address\_id INT PRIMARY KEY,

address\_street VARCHAR(50),

address\_city VARCHAR(50),

address\_state CHAR(2),

address\_zip\_code CHAR(5)

)

-- Transactions table

CREATE TABLE transactions (

transaction\_id INT IDENTITY(1,1) PRIMARY KEY,

transaction\_date DATE NOT NULL,

transaction\_patron\_id INT NOT NULL

CONSTRAINT fk\_patrons\_patron\_id FOREIGN KEY (transaction\_patron\_id) REFERENCES patrons(patron\_id)

);

-- Transactions Book list table

CREATE TABLE transaction\_books (

transaction\_books\_transaction\_id INT PRIMARY KEY,

transaction\_books\_book\_id INT PRIMARY KEY,

CONSTRAINT fk\_transactions\_transaction\_id FOREIGN KEY (transaction\_books\_transaction\_id) REFERENCES transactions(transaction\_id),

CONSTRAINT fk\_transaction\_books\_book\_id FOREIGN KEY (transaction\_books\_book\_id) REFERENCES books(book\_id)

);

-- Books table

CREATE TABLE books(

book\_id INT IDENTITY(1,1) PRIMARY KEY,

book\_title VARCHAR(50) NOT NULL,

book\_publisher\_id INT,

book\_publication\_year CHAR(4),

book\_isbn CHAR(13),

CONSTRAINT fk\_publishers\_publisher\_id FOREIGN KEY (book\_publisher\_id) REFERENCES publishers(publisher\_id)

);

-- Authors Table

CREATE TABLE authors (

author\_id INT IDENTITY(1,1) PRIMARY KEY,

author\_first\_name VARCHAR(50) NOT NULL,

author\_last\_name VARCHAR(50) NOT NULL,

author\_birth\_date DATE,

author\_nationality VARCHAR(50)

);

-- Bridge Table for Authors & Books

CREATE TABLE book\_authors (

book\_authors\_book\_id INT PRIMARY KEY,

book\_authors\_author\_id INT PRIMARY KEY,

CONSTRAINT fk\_book\_authors\_book\_id FOREIGN KEY (book\_authors\_book\_id) REFERENCES books(book\_id),

CONSTRAINT fk\_book\_authors\_author\_id FOREIGN KEY (book\_authors\_author\_id) REFERENCES authors(author\_id)

);

-- Genres Table

CREATE TABLE Genre (

genre\_id INT IDENTITY(1,1) PRIMARY KEY,

genre\_name VARCHAR(50) NOT NULL

);

-- Bridge Table for Genres & Books

CREATE TABLE book\_genres (

book\_genres\_book\_id INT PRIMARY KEY,

book\_genres\_genre\_id INT PRIMARY KEY,

CONSTRAINT fk\_book\_genres\_book\_id FOREIGN KEY (book\_genres\_book\_id) REFERENCES books(book\_id),

CONSTRAINT fk\_book\_genres\_genre\_id FOREIGN KEY (book\_genres\_genre\_id) REFERENCES genres(genre\_id)

);

-- Publishers Table

CREATE TABLE Publishers (

publisher\_id INT IDENTITY(1,1) PRIMARY KEY,

publisher\_name VARCHAR(50) NOT NULL,

publisher\_address\_id INT,

publisher\_email VARCHAR(100),

publisher\_phone\_number CHAR(10)

CONSTRAINT fk\_publishers\_address\_id FOREIGN KEY (publisher\_address\_id) REFERENCES addresses(address\_id)

);

-- Fines Table

CREATE TABLE fines (

fine\_id INT IDENTITY(1,1) PRIMARY KEY,

fine\_patron\_id INT,

fine\_amount MONEY,

fine\_given\_date DATE,

fine\_due\_date DATE,

FOREIGN KEY (fine\_customer\_id) REFERENCES Patron(patron\_id)

);

GO

-- Sample Date Loading

-- Sample data loading for Author table

INSERT INTO authors (author\_id, author\_first\_name, author\_last\_name, author\_birth\_date, author\_nationality)

VALUES

(1, 'John', 'Doe', '1980-01-15', 'American'),

(2, 'Jane', 'Smith', '1975-05-20', 'British');

(3, 'David', 'Johnson', '1992-03-10', 'Canadian'),

(4, 'Emily', 'Williams', '1985-09-18', 'Australian'),

(5, 'Michael', 'Brown', '1978-07-25', 'Irish'),

(6, 'Sarah', 'Miller', '1989-12-04', 'French'),

(7, 'William', 'Anderson', '1995-02-14', 'German'),

(8, 'Olivia', 'Jones', '1972-06-30', 'Italian'),

(9, 'James', 'Wilson', '1983-04-21', 'Spanish'),

(10, 'Sophia', 'Davis', '1990-11-11', 'Swedish');

-- Insert 10 books with fictitious data

INSERT INTO books (book\_title, book\_author\_id, book\_genre\_id, book\_publisher\_id, book\_publication\_year, book\_ISBN)

VALUES ('The Great Novel', 1, 1, 1, 1995, '978-3-16-148410-0'),

('SciFi Adventure', 2, 2, 2, 2005, '978-1-61-729584-8'),

('Romantic Tales', 3, 3, 3, 2012, '978-1-57-231514-5'),

('History of Empires', 4, 4, 1, 1998, '978-0-07-222680-0'),

('Math for Dummies', 5, 5, 2, 2019, '978-0-07-222535-3'),

('Elixir of Life', 1, 6, 3, 1987, '978-1-58-347562-8'),

('Art of Cooking', 6, 7, 1, 2009, '978-0-19-953589-7'),

('Future of AI', 7, 8, 2, 2021, '978-0-07-882476-0'),

('Calm Your Mind', 8, 9, 3, 2015, '978-0-596-52068-7'),

('Space Exploration', 9, 10, 1, 2003, '978-0-596-10046-9');

-- Insert 10 patrons with fictitious data

INSERT INTO patrons (patron\_first\_name, patron\_last\_name, patron\_address, patron\_date\_of\_birth, patron\_phone\_number, patron\_email)

VALUES ('Alice', 'Smith', '123 Main St, Springfield, IL', '1985-12-01', '+1 555 1234', 'alice@email.com'),

('Bob', 'Johnson', '456 Elm St, Shelbyville, IL', '1990-06-15', '+1 555 5678', 'bob@email.com'),

('Catherine', 'Williams', '789 Oak St, Capital City, IL', '1995-02-20', '+1 555 9876', 'catherine@email.com'),

('David', 'Brown', '101 Maple St, Ogdenville, IL', '1980-08-10', '+1 555 2468', 'david@email.com'),

('Emily', 'Davis', '202 Pine St, North Haverbrook, IL', '1975-03-15', '+1 555 1357', 'emily@email.com'),

('Frank', 'Garcia', '303 Birch St, Seaport, IL', '1987-11-21', '+1 555 8642', 'frank@email.com'),

('Grace', 'Harris', '404 Cedar St, Cypress Creek, IL', '1999-09-30', '+1 555 9753', 'grace@email.com'),

('Henry', 'Clark', '505 Redwood St, Salsbury, IL', '1965-05-25', '+1 555 7891', 'henry@email.com'),

('Ivy', 'Lewis', '606 Sequoia St, Springfield, IL', '1978-04-12', '+1 555 1598', 'ivy@email.com'),

('Jack', 'Young', '707 Palm St, Waverly Hills, IL', '2002-01-17', '+1 555 7531', 'jack@email.com');

-- Sample Queries

-- Query 1: Retrieve books by a specific author

SELECT Book.book\_title

FROM Book

WHERE Book.book\_author\_id = 1;

-- Query 2: Retrieve books borrowed by a specific patron

SELECT patron\_id, transaction\_date, book\_id, book\_title

FROM patrons p

JOIN transactions t on p.patron\_id = t.transaction\_patron\_id

JOIN transaction\_books tb on t.transaction\_id = tb.transaction\_books\_transaction\_id

JOIN books b on tb.transaction\_books\_book\_id = b.book\_id

WHERE patron\_id = 1;

-- Query 3: Calculate the total fines paid by a patron

SELECT patrons.patron\_first\_name, patrons.patron\_last\_name, SUM(fins.fine\_amount) AS total\_fines\_paid

FROM patrons

LEFT JOIN Fine ON patrons.patron\_id = fines.fine\_patron\_id

GROUP BY patrons.patron\_id;

-- Example Trigger

DELIMITER //

CREATE TRIGGER UpdateDueDate

AFTER INSERT ON fines

FOR EACH ROW

BEGIN

IF NEW.fine\_due\_date IS NULL THEN

UPDATE fines

SET borrowed\_return\_date = CURDATE() + 14

WHERE fine\_id = NEW.fine\_id;

END IF;

END;

//

DELIMITER ;

-- Example procedure

CREATE PROCEDURE viewCurrentAddress @patron\_id AS

BEGIN

SELECT patron\_id, address\_street + ', ' + address\_city + ', ' + address\_state + ' ' + address\_zip\_code

FROM patrons

JOIN address on patron\_address\_id = address\_id

WHERE patron\_id = @patron\_id

END;

-- Example View

CREATE VIEW BiographyGenreBookLookup AS

BEGIN

SELECT book\_id, book\_title, book\_publication\_year, book\_isbn

FROM books

JOIN book\_genres ON book\_id = book\_genres\_book\_id

JOIN genres ON genre\_id = book\_genres\_genre\_id

WHERE book\_genre\_id = 'Biography'

END;

-- Example Indexes

CREATE INDEX ix\_genre\_id ON genres(genre\_id)

CREATE INDEX ix\_author\_id ON authors(author\_id)