**SQL Database Project Part 4: The Logical Relational Model**

**Introduction:**

**Project Overview -**

This project involves the design, implementation, and management of a relational database system for a small library. The system is intended to efficiently handle the library's collection of loanable items, manage different types of memberships, enforce borrowing policies, and generate insightful reports. By creating a well-structured database, the library can streamline operations, improve accessibility to resources, and ensure proper tracking of borrowed materials.

**Scope -**

This project will include a user interface for both the library staff and their clients. The library staff interface will include the ability to check out items, process returns, add new items, and manage client accounts. The client side will allow for searching the catalog, reserving items, checking loan status, etc. This project will be structured using SQL RDBMS databases. We will have client IDs, staff IDs, and inventory IDs to keep track of all interactions between users, the material, and the library staff. The database will implement rules for borrowing, due dates, and fines, ensuring compliance with library policies. Additionally, the system will provide reporting features for analyzing borrowing trends and member activity.

Note: We updated the tables from the ER diagram as it wasn't as efficient as we were thinking it would be. Primary keys are underlined and bolded. Some keys are primary keys and foreign keys at the same time.

**Database Schema -**

users(**user_id**, reg_no, name, first_name, last_name, dob, email, phone, reg_date, membership)

authentication(**id**, password)

login(**id**, user_id)
FK: id → authentication.id
FK: user_id → user.user_id

staff(**staff_id**, first_name, last_name, email)

keeps_track_of(**staff_id**, **user_id**)
FK: staff_id → staff.staff_id
FK: user_id → user.user_id

reserves_returns(**user_id**, **media_id**, **date_reserved**, return_date, due_date)
FK: user_id → user.user_id
FK: media_id → media.media_id

media(**media_id**, title, media_type)

books(**media_id**, title, author, pub_date, genre, availability)
FK: media_id → media.media_id

magazines(**media_id**, title, issn, iss_num, publisher, genre)
FK: media_id → media.media_id

movies(**media_id**, title, director, duration, genre, age_rating)
FK: media_id → media.media_id

games(**media_id**, title, developer, console, genre, age_rating)
FK: media_id → media.media_id

reports(**report_id**, staff_id, lib_acquired_date)
FK: staff_id → staff.staff_id

queues(**queue_num**, **media_id**, **user_id**)
FK: media_id → media.media_id
FK: user_id → user.user_id

## Data Dictionary -

| **USER:** | | |
|---|---|---|
| **Attribute Name** | **Data Type** | **Description** |
| User_ID | INT | Unique identifier for each user (positive integers) |
| Reg_No | VARCHAR(50) | Registration number (up to 50 characters) |
| Name | VARCHAR(100) | Full name of the user (alternative to first/last names) |
| First_Name | VARCHAR(50) | User’s first name (up to 50 characters) |
| Last_Name | VARCHAR(50) | User’s last name (up to 50 characters) |
| DOB | DATE | Date of birth (YYYY-MM-DD) |

| | | |
|---|---|---|
| Email | VARCHAR(100) | Email address in a valid format |
| Phone | VARCHAR(15) | Phone number (digits, may include symbols like + or -) |
| Reg_Date | DATE | Date of registration (defaults to current date if not provided) |
| Membership_Type | VARCHAR(50) | Type of membership (e.g., “Standard”, “Premium”, “Student”) |

**AUTHENTICATION:**

| Attribute Name | Data Type | Description |
|---|---|---|
| ID | INT | Unique authentication record ID (positive integers) |
| Password | VARCHAR(128) | Hashed password (up to 128 characters) |

**LOGIN:**

| Attribute Name | Data Type | Description |
|---|---|---|
| ID | INT | Authentication ID; links to Authentication(ID) (positive integers) |
| User_ID | INT | User ID; links to User(User_ID) (positive integers) |

**STAFF:**

| Attribute Name | Data Type | Description |
|---|---|---|
| Staff_ID | INT | Unique staff identifier (positive integers) |
| First_Name | VARCHAR(50) | Staff member’s first name (up to 50 characters) |
| Last_Name | VARCHAR(50) | Staff member’s last name (up to 50 characters) |
| Email | VARCHAR(100) | Staff member’s email address in a valid format |

**KEEPS_TRACK_OF:**

| Attribute Name | Data Type | Description |
|---|---|---|
| Staff_ID | INT | Staff identifier; references Staff(Staff_ID) (positive integers) |

| | | |
|---|---|---|
| User_ID | INT | User identifier; references User(User_ID) (positive integers) |
| | | |
| | | |
| **RESERVES_RETURNS:** | | |
| **Attribute Name** | **Data Type** | **Description** |
| User_ID | INT | User identifier; references User(User_ID) (positive integers) |
| Media_ID | VARCHAR(50) | Identifier for the media item (‘b_0001’, ‘m_1234’, etc.) |
| Date_Reserved | DATE | Date when the item was reserved (YYYY-MM-DD) |
| Return_Date | DATE | Date when the item was returned (YYYY-MM-DD); can be NULL if not returned |
| Due_Date | DATE | Due date for returning the item (YYYY-MM-DD) |
| | | |
| **MEDIA:** | | |
| **Attribute Name** | **Data Type** | **Description** |
| Media_ID | VARCHAR(50) | Identifier for the media item (‘b_0001’, ‘m_1234’, etc.) |
| Title | VARCHAR(100) | Title of the media (up to 100 characters) |
| Media_type | VARCHAR(20) | Type of media from the database (must be of one of the following types: book, magazine, movie, game |
| | | |
| | | |
| **BOOK:** | | |
| **Attribute Name** | **Data Type** | **Description** |
| Media_ID | VARCHAR(50) | Identifier for the media item (books will start with b followed by a string of unique numbers: ‘b_1001’) |
| Title | VARCHAR(100) | Title of the book (up to 100 characters) |
| Author | VARCHAR(100) | Author(s) of the book (up to 100 characters) |
| Pub_Date | DATE | Publication date (YYYY-MM-DD) |
| Genre | VARCHAR(50) | Book genre (e.g., Fiction, Non-Fiction) |
| Availability | BOOLEAN | Availability status (TRUE if available, FALSE otherwise) |

**MAGAZINE:**

| **Attribute Name** | **Data Type** | **Description** |
|---|---|---|
| Media_ID | VARCHAR(50) | Identifier for the media item (magazines will start with mg followed by a string of unique numbers: ‘mg_2001’) |
| Title | VARCHAR(100) | Magazine title (up to 100 characters) |
| ISSN | VARCHAR(20) | International Standard Serial Number; unique if tracked (up to 20 characters) |
| Iss_Num | INT | Issue number (positive integer) |
| Publisher | VARCHAR(100) | Magazine publisher (up to 100 characters) |
| Genre | VARCHAR(50) | Magazine genre (e.g., Lifestyle, News) |

**MOVIE:**

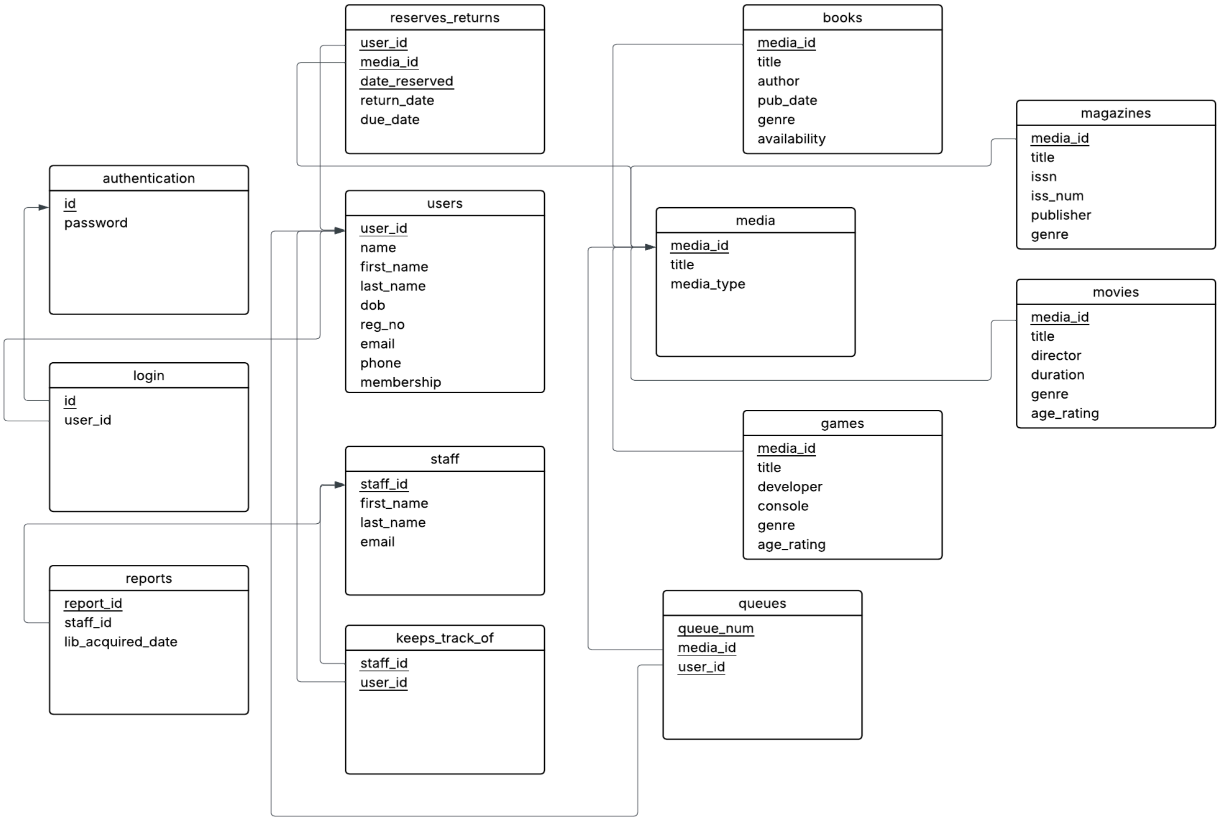
| **Attribute Name** | **Data Type** | **Description** |
|---|---|---|
| Media_ID | VARCHAR(50) | Identifier for the media item (movies will start with mv followed by a string of unique numbers: ‘mv_03001’) |
| Title | VARCHAR(100) | Movie title (up to 100 characters) |
| Director | VARCHAR(100) | Director’s name (up to 100 characters) |
| Duration | INT | Duration in minutes (positive integer) |
| Genre | VARCHAR(50) | Movie genre (e.g., Drama, Comedy) |

**GAME:**

| **Attribute Name** | **Data Type** | **Description** |
|---|---|---|
| Media_ID | VARCHAR(50) | Identifier for the media item (games will start with g followed by a string of unique numbers: ‘g_4001’) |
| Title | VARCHAR(100) | Game title (up to 100 characters) |
| Developer | VARCHAR(100) | Game developer/studio (up to 100 characters) |
| Console | VARCHAR(50) | Supported console (e.g., PS5, Xbox) |
| Genre | VARCHAR(50) | Game genre (e.g., Action, RPG) |
| Age_Rating | VARCHAR(10) | Age rating (e.g., "E", "T", "M") |

| | | |
|---|---|---|
| | | |
| | | |
| **REPORTS:** | | |
| **Attribute Name** | **Data Type** | **Description** |
| Report_ID | INT | Unique report identifier (positive integers) |
| Staff_ID | INT | Staff identifier; references Staff(Staff_ID) (positive integers) |
| Lib_Acquired_Date | DATE | Date library acquired the resource (YYYY-MM-DD) |
| | | |
| | | |
| **QUEUES:** | | |
| **Attribute Name** | **Data Type** | **Description** |
| Queue_Num | INT | Unique queue entry identifier (positive integers) |
| Media_ID | VARCHAR(50) | Identifier for the media item (references the unified media ID) |
| User_ID | INT | User identifier; references User(User_ID) (positive integers) |

## Relational Schema Diagram -



# Meeting notes -

Date: March 28, 2025
Time: 2:30 PM - 3:00 PM
Location: Discord Meeting
Objective: To discuss the 4th part of the project.
Team Members Present: Chris, Christina, Elizabeth, Ryan

Task Completion Confirmation:

Chris: Yes
Carter: Yes
Christina: Yes
Elizabeth: Yes
Jaret: Yes
Ryan: Yes

Brainstorming Session:

- We immediately started discussing what part 4 of the project entailed.
- We then moved onto assigning roles for what we each were going to do, and in this process, we got confused by what the difference between the relational schema mapping and relational schema diagram was. We sent an email to get clarification.
- We wrapped up the meeting, agreeing to do our tasks and let the rest of the team know if we found any information about the difference between the relational schema mapping and diagram.

Tasks Allocated:

Chris: Data Dictionary
Carter: Relational Schema Mapping
Christina: Relational Schema Diagram
Elizabeth: Introduction
Jaret: Relational Schema Mapping
Ryan: Relational Schema Mapping, and finish determining out our attributes

Follow-Up Actions:

- Work on Part 4 of the project.
- We will update the team once we get clarifications on our confusion.

Schedule the next meeting: 4/7/2025 @ 9:00AM