**Library Book Management System**

***Group: 10 (Abhishek Aggarwal, Yashan Wasson, Ayush Kapoor)***

*Section: 1 (Introduction)*

1. In our library book management system, we've set up a database to keep things organized and running smoothly. We've got different tables for books, reservations, students, issued books, and book returns. Each table has its own job, like keeping track of who borrowed a book, when it's due back, and more. This way, our database helps us easily manage and retrieve information, making sure everything in the library stays well-organized.
2. Students:
   * Holds vital information about Students, including their first name, last name, Email.
   * Primary key: StudentID
3. Books:
   * Contains comprehensive details about ISBN, Name, Author, Publisher, Copies.
   * Primary key: BooksNO
4. Issues:
   * Compiles pertinent data about StudentId, BookNo, IssueDate, DueDate.
   * Primary key: IssueID
   * Foreign keys: Bookno, StudentID
5. IssueReturns:
   * Serves as a repository for invoices and encompasses BookNo, StudentID, IssueDate, DueDate, ReturnDate.
   * Primary key: ReturnID
   * Foreign keys: Bookno, StudentID
6. Reservations:
   * Compiles pertinent information including StudentID, BookNo, ReservationDate, Status.
   * Primary key: ReservationID
   * Foreign keys: Bookno, StudentID

*Explanation of the Data (*library book management system*):*

The database structure allows for efficient organization and retrieval of information related to students, books, issued books, book returns, and reservations.

Relationships between tables are established using primary and foreign keys, ensuring data integrity and consistency.

This system facilitates tracking the borrowing and returning of books, managing student information, and handling reservations, providing a robust foundation for library management.

*Relationships between the tables in the database:*

1. One-to-Many Relationship between Students and Issues:

One student can issue multiple books.

One issue can be associated with only one student.

1. One-to-Many Relationship between Books and Issues:

One book can be issued multiple times.

Each issue is associated with only one book.

1. Many-to-One Relationship between Issues and Students:
   * Many issues can be associated with one student.
   * Each issue is associated with only one student.
2. Many-to-One Relationship between Issues and Books:

* Many issues can be associated with one book.
* Each issue is associated with only one book.

1. Many-to-One Relationship between IssueReturns and Students:

* Many issue returns can be associated with one student.
* Each issue return is associated with only one student.

1. Many-to-One Relationship between IssueReturns and Books:

* Many issue returns can be associated with one book.
* Each issue return is associated with only one book.

1. Many-to-One Relationship between Reservations and Students:

* Many reservations can be associated with one student.
* Each reservation is associated with only one student.

1. Many-to-One Relationship between Reservations and Books:

* Many reservations can be associated with one book.
* Each reservation is associated with only one book.

*Section 2: Group Membership and Tasks*- Abhishek Aggarwal: [agga0042@algoquinlive.com](mailto:agga0042@algoquinlive.com)

- Yashan Wasson: [wass0025@algonquinlive.com](mailto:wass0025@algonquinlive.com)

- Ayush Kapoor: [kapo0040@algonquinlive.com](mailto:kapo0040@algonquinlive.com)

*Section 3: Historical Data Framework and Multi-valued Fields*

In our database system, we plan to introduce multi-valued historical data fields in the following tables and relationships:

1. Student - Is-a (relationship)
2. Reservation - Is-related-to (relationship)
3. issues - Contains (relationship)

For each of these fields, we will develop a set of triggers and stored procedures designed to manage INSERTS, UPDATES, and DELETES while concurrently maintaining a comprehensive history of changes through timestamps. This approach enables us to monitor and record alterations and updates made to our database system, providing a chronological record of modifications over time.

**Database Schema**

**Entity Descriptions**

* **Students**: Stores Student information.
* **Books**: Contains details about Books, their author, publisher etc.
* **issues**: This table compiles pertinent data about issued books.
* **Issue Returns**: This table serves as a repository for information related to book returns.
* **Reservations**: This table compiles information about book reservations made by students.
* **Student history, Reservation history, issues history**: Tables for tracking historical data changes.

Steps:  
  
First, Creating the Tables:

A screenshot of a computer

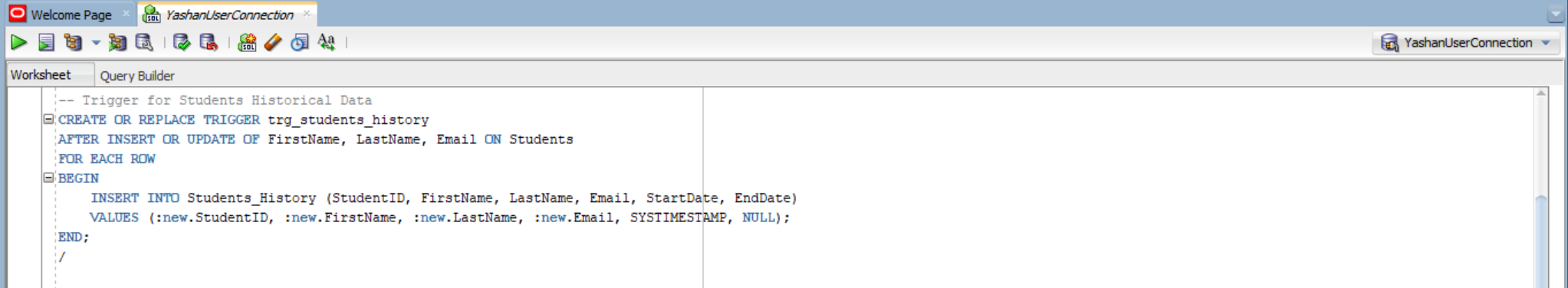
Description automatically generated

Creating the Table history (students) :

A close-up of a computer screen

Description automatically generated

Creating the Trigger for student history table :



And the Second History table (issues):

A screenshot of a computer

Description automatically generated

Creating the Trigger for issue history table :

A screenshot of a computer

Description automatically generated

and the third History table (Reservations) :

A white background with a black and white flag

Description automatically generated

Creating the Trigger for Reservation history table:

A close up of a screen

Description automatically generated

**Triggers:**

**trg\_students\_history :** The trigger trg\_students\_history captures and records historical data for insertions or updates of FirstName, LastName, or Email in the Students table, storing the changes in the Students\_History table with a timestamp.

**trg\_issues\_history**: The trigger trg\_issues\_history captures and records historical data for insertions or updates of IssueDate or DueDate in the Issues table, maintaining a historical log in the Issues\_History table with timestamps.

**trg\_reservations\_history**: The trigger trg\_reservations\_history captures and records historical data for insertions or updates of ReservationDate or Status in the Reservations table, storing the changes in the Reservations\_History table with timestamps.

***The result from the table Reservation\_history:***

A screenshot of a computer

Description automatically generated

***The result from the table students\_history:***

A screenshot of a computer

Description automatically generated

***The result from the table Problem \_history:***

A screenshot of a computer

Description automatically generated

***Views:***

*Issue view:*

A screenshot of a computer

Description automatically generated

*Reservation view:*

A screenshot of a computer

Description automatically generated

*Student view:*

A screenshot of a computer

Description automatically generated

*A screenshot of a computer

Description automatically generated*

***Now create new Access and open it   
  
A screenshot of a computer

Description automatically generated***

***Go to External Data***

***A screenshot of a computer

Description automatically generated***

***Navigate to” New Data Source”  
  
A screenshot of a computer

Description automatically generated***

***Click on ”From Other Sources” => And then “ODBC Database”***

A screenshot of a computer

Description automatically generated

***Click Ok***

***It is going to show you this page :  
  
A screenshot of a computer

Description automatically generated***

***Go to the second bar “Machine Data Source”***A screenshot of a computer

Description automatically generated

***Click on “Assignment2” and then ok.***

***Choose the tables that you made in oracle.***

A screenshot of a computer

Description automatically generated

***Finally press OK. You have imported successfully  
  
A screenshot of a computer

Description automatically generated***

***Access:   
Relationship:***A screenshot of a computer program

Description automatically generated

***FORMS***

***The forms employed for inserting, updating, and deleting data from the primary tables act as user-friendly interfaces, streamlining interactions with the database. Designed for accuracy and efficiency, these forms facilitate the addition of new records, modification of existing data, and removal of entries. Their purpose is to enhance user experience, ensuring structured and secure data management processes while upholding quality and integrity standards within the database.***

* + 1. ***ASSIGNMENT2\_ISSUES***

A screenshot of a computer

Description automatically generated

* + 1. ***ASSIGNMENT2\_RESERVATIONS***

A screenshot of a computer

Description automatically generated

* + 1. ***ASSIGNMENT2\_STUDENTS***

A screenshot of a computer

Description automatically generated

***REPORTS***

***After utilizing the forms in Microsoft Access for inserting, updating, or deleting data within the Cars, Sales, or Maintenance tables, users can seamlessly track the changes by employing the designated report views. These report views serve as a comprehensive interface, allowing users to review and analyze the historical modifications made to the data in the tables. By accessing the report views, users gain valuable insights into the evolution of the database, ensuring transparency and facilitating informed decision-making. The reports provide a detailed overview of the alterations, including timestamps and relevant details, enabling users to effectively monitor and comprehend the chronological history of changes within the database tables.***

A screenshot of a computer

Description automatically generated

***MENU:***

***A library with many books on shelves

Description automatically generated***