**SQL Project Title:** **Library Management System**

**Name – Kalpesh Parikh**

**Objective:**Build a basic database for a library system to manage books, members, and transactions like issuing and returning books.

**Tasks:**

1. **Create Tables:**
   * Books: Contains details of all books in the library.
     + Columns: BookID, Title, Author, Genre, YearPublished, CopiesAvailable
   * Members: Contains details of library members.
     + Columns: MemberID, FirstName, LastName, DateOfBirth, MembershipDate
   * Transactions: Logs book issuance and returns.
     + Columns: TransactionID, MemberID, BookID, IssueDate, ReturnDate

**Solution:**

**create table books(**

**book\_id int Primary key,**

**title varchar(50),**

**author varchar(50),**

**genere varchar(50),**

**yearpublished year,**

**copiesavailable int**

**);**

**create table members(**

**member\_id int Primary key,**

**firstname varchar(50),**

**lastname varchar(50),**

**dob date,**

**membershipdate date**

**);**

**create table transactions(**

**transaction\_id int Primary key,**

**member\_id int,**

**book\_id int,**

**issuedate date,**

**returndate date,**

**FOREIGN KEY (book\_id) REFERENCES books(book\_id),**

**FOREIGN KEY (member\_id) REFERENCES members(member\_id)**

**);**

1. **Insert Data:**Add sample data for books, members, and transactions.

**Solution:**

**Insert into Enterprise.books values(8, 'title8', 'author8', 'drama', 2004, 50);**

**Insert into Enterprise.members values(8, 'aplesh', 'mistry', '1975-08-02', '2002-08-30');**

**Insert into Enterprise.transactions values(14, 2, 5, '2002-08-10', '2002-08-30');**

1. **Queries to Practice:**
   * Retrieve all books by a specific author.

**Solution:**

**SELECT \* FROM enterprise.books**

**where author = 'author2'**

* + List books available in a particular genre.

**Solution:**

**SELECT \* FROM enterprise.books**

**where genere = 'drama'**

* + Find members who joined in a specific year.

**Solution:**

**select \* from enterprise.members**

**where membershipdate = '2005-01-01'**

* + Check which books are currently issued and their due dates.

**Solution:**

**select \* from enterprise.transactions**

**where issuedate >= '2005-01-01'**

* + Count the total number of books available in the library.

**Solution:**

**With new\_transactions as**

**(SELECT book\_id, count(book\_id) as cbook\_cnt**

**FROM enterprise.transactions**

**group by book\_id**

**)**

**select b.book\_id, (b.copiesavailable - nt.cbook\_cnt)**

**from enterprise.books b**

**join new\_transactions nt on b.book\_id = nt.book\_id;**

1. **Advanced Queries (Optional):**
   * Identify the member who has issued the most books.

**Solution:**

**SELECT member\_id, count(book\_id) as cnt**

**FROM enterprise.transactions**

**group by member\_id**

**order by cnt desc**

* + Generate a report of overdue books.

**Solution:**

**SELECT \***

**FROM enterprise.transactions**

**where returndate is NULL**