UTM

UNIVERSITI TEKNOLOGI MALAYSIA

DATABASE

SECD2523 (SECTION 01)

2024/2025 – SEMESTER 1

PHASE 3

DATABASE LOGICAL DESIGN & SQL

LECTURER: DR HASLINA BINTI HASHIM

COURSE: SECBH

GROUP NUMBER: 6

NAME	MATRIC NO.
MUHAMMAD HAZIQ BIN MOHD HAFIZAL	A23CS0132
ALDANISHA MUADZ BINTI MUZAFFAR	A23CS0039
NAZATUL NADHIRAH BINTI SABTU	A23CS0144
NURUL ATHIRAH SYAFIQAH BINTI MOHD RAZALI	A23CS0163

TABLE OF CONTENT

1.0 INTRODUCTION

In today's digital age, libraries are increasingly transitioning to online platforms to enhance efficiency and accessibility. An online library management system (OLMS) is a computer program that helps libraries run better. It lets people borrow books online, search for them, and return them. This makes it easier for people to use the library without having to go there.

Many OLMS programs already exist, but some have problems. They might not keep track of books very well, or they might be hard to use. This project will make a new OLMS that fixes these problems. It will be easy to use and will keep track of books very well.

2.0 OVERVIEW OF PROJECT

The Online Library Management System (OLMS) is designed to modernize and streamline library operations by enabling digital interactions between users and the library's resources. It provides seamless access to core functionalities such as user registration, book search, borrowing, returning, and inventory management, while automating key processes to ensure real-time updates and reduce staff intervention. Users can register and log in to manage their borrowing history, search for books with real-time availability updates, reserve available items, or request special orders for unavailable books, triggering supplier notifications. The system facilitates borrowing and returning by tracking due dates, calculating fines for overdue items, and updating inventory upon returns. It also automates stock updates, monitors inventory levels to prevent shortages, and maintains accurate records of book conditions. Additionally, OLMS processes fine payments and generates real-time reports and analytics, providing insights into borrowing patterns, inventory status, and user engagement. By addressing the shortcomings of traditional library systems, the OLMS enhances accessibility, efficiency, and user satisfaction, transforming the management of library resources into a seamless, intuitive, and efficient process.

3.0 DATABASE CONCEPTUAL DESIGN

3.1 UPDATED BUSINESS RULE

Library staff:

I. Add and update book information, including availability, through the system.
II. Manage user registrations and assign unique IDs to new users.
III. Monitor and review the reports generated by the system.
IV. Oversee the book inventory status.

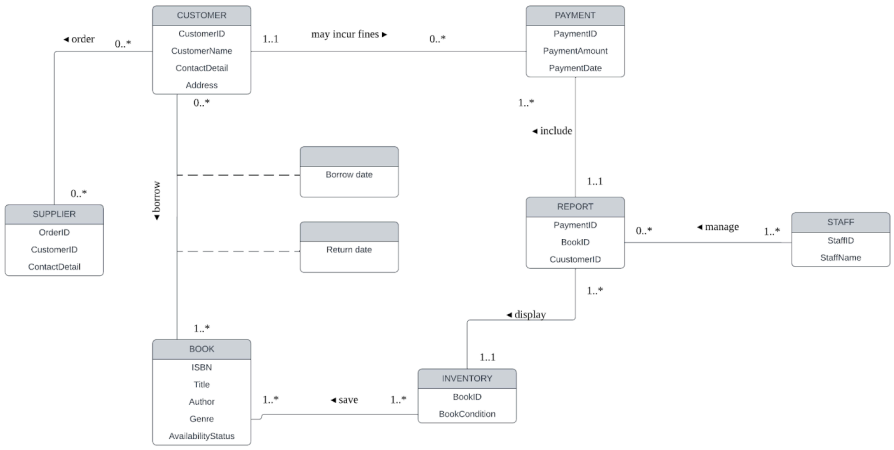
System:

I. Automatically verify user data using validation rules and securely store it as customer data.
II. Display real-time book availability status.
III. Notify the supplier and place orders for unavailable books.
IV. Assign due dates for borrowed books and track return dates.
V. Calculate fines for overdue returns and update user accounts accordingly.
VI. Automatically generate real-time monthly reports by drawing data from all data stores (Customer Data, Book Inventory, and Book Loan).

Customer:

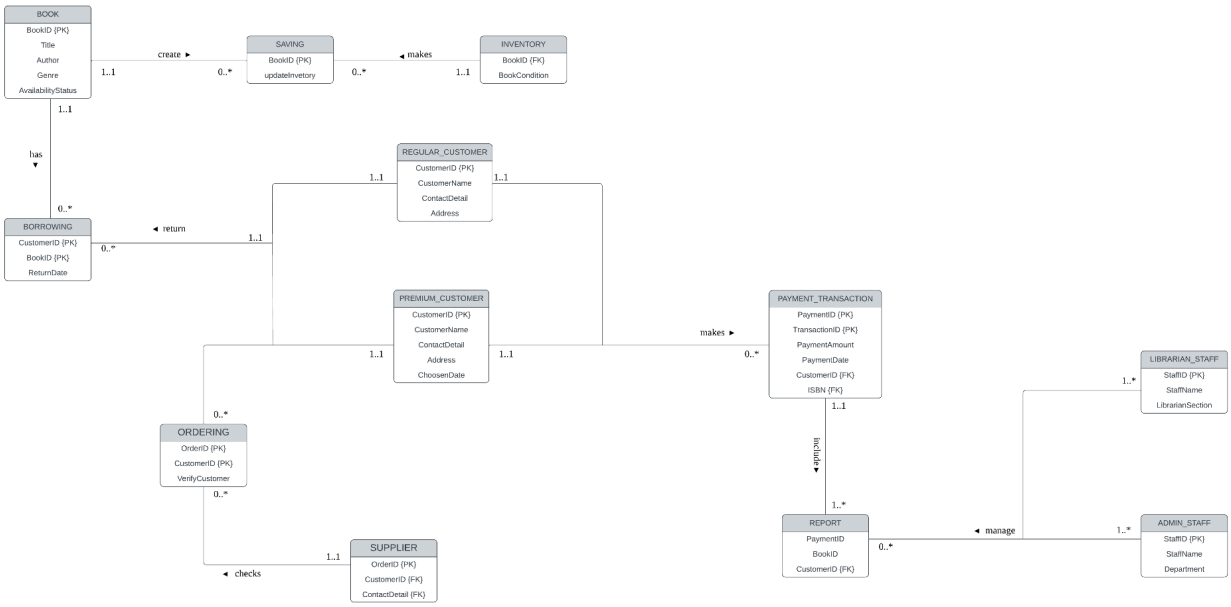
I. Register and log into the system to borrow books.
II. Check book availability and reserve books online.
III. Borrow books by filling in account information and borrowing details online.
IV. Update the return status of borrowed books online.
V. Clear outstanding fines before borrowing more items.

3.2 CONCEPTUAL ERD



4.0 DB LOGICAL DESIGN

4.1 LOGICAL ERD



4.2 UPDATED DATA DICTIONARY

Description of Entity

Entity	Description	Occurrence
Regular Customer	Regular customer's information.	Customers can borrow books, make payments, and may receive reports on their transactions.
Premium Customer	Premium customer's information.	Premium customers have additional benefits, such as exclusive offers and priority access to book borrowing.
Payment Transaction	Payment transaction's information.	Payments are processed when customers purchase books, borrow books, or pay fines.
Book	Details about books available in the library.	Books are added to the inventory, borrowed by customers, and managed by staff.
Supplier		Suppliers provide books or other materials to maintain the library/store inventory.
Borrowing	Book loan details by customers.	Tracks details of books borrowed by customers, including return dates and overdue records.
Ordering	Book order details by customers.	Handles book orders placed by customers and verifies customers for order eligibility.
Report	Report's information.	Reports are generated periodically for management and staff to review system activity and customer transactions.

Inventory	Inventory's information.	Inventory is updated when books are added, sold, borrowed, or returned.
Librarian Staff	Librarian staff's information.	Librarian staff manage book borrowing, inventory, and customer interactions.
Admin Staff	Administrative staff's information.	Admin staff oversee staff roles, system operations, and management reports.

Description of Relationship

Entity	**Multiplicity**	**Relationship**	**Multiplicity**	**Entity**
Regular Customer	1…1	Borrow	1…*	Borrowing
Premium Customer	1…1	Borrow	1…*	Borrowing
Customer	1…1	Make	1…*	Payment Transaction
Book	1…*	Part of	1…1	Inventory
Librarian Staff	1…1	Manage	1…*	Borrowing
Admin Staff	1…1	Oversee	1…*	Report
Borrowing	1…1	Generate	1…*	Report
Book	1…*	Supplied by	1…*	Supplier
Customer	1…1	Request	1…*	Ordering

Description Attributes

Entity	Attributes	Description	Data Type	Null	Multi-Valued
Regular Customer	Customer ID	Unique identifier for a regular customer.	INT	No	No
	CustomerName	Full name of the customer.	VARCHAR(100)	No	No
	ContactDetail	Contact number of email for communication.	VARCHAR(50)	Yes	No
	Address	Residential or mailing address of the customer	VARCHAR(255)	Yes	No
Premium Customer	CustomerID	Unique identifier for a premium customer.	INT	No	No
	MembershipBenefits	Description of additional services or offers provided to premium customers.	TEXT	Yes	No
Book	ISBN	Unique identifier for each book in the library.	INT	No	No

	Title	Title of the book	VARCHAR(150)	No	No
	Author	Name(s) of the book's author(s)	VARCHAR(100)	Yes	No
	Genre	Category or genre of the book	VARCHAR(50)	Yes	No
	AvailabilityStatus	Current availability status	VARCHAR(20)	No	No
Borrowing	CustomerID	Foreign key linking to the customer borrowing the book	INT	No	No
	ISBN	Foreign key linking to the book being borrowed	INT	No	No
	BorrowDate	Date when the book was borrowed	DATE	No	No
	DueDate	Date by which the book must be returned	DATE	No	No
	ReturnDate	Actual date when the book was returned	DATE	Yes	No
Payment Transaction	PaymentID	Unique identifier for each payment	INT	No	No

		transaction			
	TransactionID	Foreign key linking to the borrowing transaction associated with the payment	INT	No	No
	PaymentAmount	Amount paid by customer	DECIMAL(10,2)	No	No
	PaymentDate	Date the payment was made.	DATE	No	No
	PaymentMethod	Payment method used	VARCHAR(30)	Yes	No
Supplier	SupplierID	Unique identifier for the supplier providing books to the library	INT	No	No
	SupplierName	Name of the supplier	VARCHAR(100)	No	No
	ContactDetail	Communication details of the supplier	VARCHAR(50)	Yes	No
Inventory	ISBN	Foreign key linking to the book in inventory	INT	No	No

	StockQuantity	Number of copies of the book available in the library	INT	No	No
	BookCondition	Condition of the book	VARCHAR(50)	No	No
Report	ReportID	Unique identifier for each report generated by the system	INT	No	No
	GeneratedDate	Date when the report was created	DATE	No	No
	ReportType	Type of report	VARCHAR950)	No	No

4.3 NORMALIZATION

1. REGULAR_CUSTOMER (CustomerID, CustomerName, ContactDetail, Address)
   Fd1 : CustomerID → CustomerName, ContactDetail, Address
   1NF&2NF&3NF&BNCF:
   REGULAR_CUSTOMER(<u>CustomerID</u>, CustomerName, ContactDetail, Address)

2. PREMIUM_CUSTOMER (CustomerID, CustomerName, ContactDetail, Address)
   Fd1: CustomerID → CustomerName, ContactDetail, Address
   1NF&2NF&3NF&BNCF:
   REGULAR_CUSTOMER(<u>CustomerID</u>, CustomerName, ContactDetail, Address)

3. BORROWING (CustomerID,BookID, ReturnBook)
   Fd1: CustomerID, BookID → BorrowDate, DueDate, ReturnDate
   1NF&2NF&3NF&BNCF:
   BORROWING(<u>CustomerID, BookID</u>, BorrowDate, DueDate, ReturnDate)

4. BOOK(BookID, Title, Author, Genre, AvailabilityStatus)
   Fd1: BookID → Title, Author, Genre, AvailabilityStatus
   1NF&2NF&3NF&BNCF:
   BOOK(<u>BookID</u>, Title, Author, Genre, AvailabilityStatus)

5. SAVING(BookID, updateInventory)
   Fd1: BookID → UpdateInventory
   1NF&2NF&3NF&BNCF:
   SAVING(<u>BookID</u>, UpdateInventory)

6. INVENTORY(BookID, BookCondition)
   Fd1: BookID → BookCondition
   1NF&2NF&3NF&BNCF:
   INVENTORY(<u>BookID</u>, BookCondition)

7. ORDERING(OrderID, CustomerID, VerifyCustomer)
   Fd1: OrderID → CustomerID, BookID, OrderDate

1NF&2NF&3NF&BNCF:
ORDERING(OrderID, CustomerID, BookID, OrderDate)

8. SUPPLIER(OrderID, CustomerID, ContactDetail)
   Fd1: SupplierID → SupplierName, ContactDetail
   1NF&2NF&3NF&BNCF:
   SUPPLIER(SupplierID, SupplierName, ContactDetail)

9. PAYMENT_TRANSACTION(PaymentID, TransactionID, PaymentAmount, PaymentDate, CustomerID, ISBN)
   Fd1: PaymentID → TransactionID, PaymentAmount, PaymentDate, CustomerID, ISBN
   1NF&2NF&3NF&BNCF:
   PAYMENT_TRANSACTION(PaymentID, TransactionID, PaymentAmount, PaymentDate, CustomerID, ISBN)

10. LIBRARIAN_STAFF(StaffID, LibrarianSection)
    Fd1: StaffID → LibrarianSection
    1NF&2NF&3NF&BNCF:
    LIBRARIAN_STAFF(StaffID, LibrarianSection)

11. ADMIN_STAFF(StaffID, StaffName, Department)
    Fd1: StaffID → StaffName, Department
    1NF&2NF&3NF&BNCF:
    ADMIN_STAFF(StaffID, StaffName, Department)

12. REPORT(PaymentID, BookID, CustomerID)
    Fd1: ReportID → PaymentID, BookID, CustomerID
    1NF&2NF&3NF&BNCF:
    REPORT(ReportID, PaymentID, BookID, CustomerID)

5.0 RELATIONAL DB SCHEMAS (AFTER NORMALIZATION)

The relation database schema for Online Library Management System (OLMS) database is a set of relation schema which is consists of ;

REGULAR_CUSTOMER	(CustomerID, CustomerName, ContactDetail, Address)
PREMIUM_CUSTOMER	(CustomerID, CustomerName, ContactDetail, Address)
BORROWING	(CustomerID, BookID, BorrowDate, DueDate, ReturnDate)
BOOK	(BookID, Title, Author, Genre, AvailabilityStatus)
SAVING	(BookID, UpdateInventory)
INVENTORY	(BookID, BookCondition)
ORDERING	(OrderID, CustomerID, BookID, OrderDate)
SUPPLIER	(SupplierID, SupplierName, ContactDetail)
PAYMENT_TRANSACTION	(PaymentID, TransactionID, PaymentAmount, PaymentDate, CustomerID, ISBN)
LIBRARIAN_STAFF	(StaffID, LibrarianSection)
ADMIN_STAFF	(StaffID, StaffName, Department)
REPORT	(ReportID, PaymentID, BookID, CustomerID)

REGULAR_CUSTOMER

CustomerID	CustomerName	ContactDetail	Address

PREMIUM_CUSTOMER

CustomerID	CustomerName	ContactDetail	Address

BORROWING

CustomerID	BookID	BorrowDate	DueDate	ReturnDate

BOOK

BookID	Title	Author	Genre	AvailabilityStatus

SAVING

BookID	UpdateInventory

INVENTORY

BookID	BookCondition

ORDERING

OrderID	CustomerID	BookID	OrderDate

SUPPLIER

SupplierID	SupplierName	ContactDetail

PAYMENT_TRANSACTION

PaymentID	TransactionID	PaymentAmount	PaymentDate	CustomerID	ISBN

LIBRARIAN_STAFF

StaffID	LibrarianSection

ADMIN_STAFF

StaffID	StaffName	Department

REPORT

ReportID	PaymentID	BookID	CustomerID

6.0 SQL STATEMENT (DDL & DML)

6.1 DDL:

```
-- Create Customers Table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    ContactDetails VARCHAR(100),
    MembershipType VARCHAR(20),
    Address VARCHAR(255)
);

-- Create Books Table
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(100) NOT NULL,
    Author VARCHAR(50),
    Genre VARCHAR(30),
    ISBN VARCHAR(13) UNIQUE,
    AvailabilityStatus VARCHAR(20) NOT NULL
);

-- Create BorrowTransactions Table
CREATE TABLE BorrowTransactions (
    TransactionID INT PRIMARY KEY,
    CustomerID INT,
    BookID INT,
    BorrowDate DATE NOT NULL,
    DueDate DATE NOT NULL,
    ReturnDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
```

```
);

-- Create Payments Table
CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY,
    TransactionID INT,
    Amount DECIMAL(10, 2) NOT NULL,
    PaymentDate DATE NOT NULL,
    FOREIGN KEY (TransactionID) REFERENCES BorrowTransactions(TransactionID)
);

-- Create Staff Table
CREATE TABLE Staff (
    StaffID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Role VARCHAR(50),
    ContactDetails VARCHAR(100)
);

-- Create Inventory Table
CREATE TABLE Inventory (
    BookID INT PRIMARY KEY,
    StockQuantity INT NOT NULL,
    BookCondition VARCHAR(50),
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
);

-- Create Suppliers Table
CREATE TABLE Suppliers (
    SupplierID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
```

```
    ContactDetails VARCHAR(100)
);

-- Create Reservations Table
CREATE TABLE Reservations (
    ReservationID INT PRIMARY KEY,
    CustomerID INT,
    BookID INT,
    ReservationDate DATE NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
);
```

DML:

```
-- Insert Data into Customers Table
INSERT INTO Customers (CustomerID, Name, ContactDetails, MembershipType, Address)
VALUES (123, 'Syahmi', '0124958330', 'Regular', '47180 Puchong');
```

CustomerID	Name	ContactDetails	MembershipType	Address
123	Syahmi	0124958330	Regular	47180 Puchong

```
-- Insert Data into Books Table
INSERT INTO Books (BookID, Title, Author, Genre, ISBN, AvailabilityStatus)
VALUES (101, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', '9780743273565', 'Available');
```

BookID	Title	Author	Genre	ISBN	AvailabilityStatus
101	The Great Gatsby	F. Scott Fitzgerald	Fiction	9780743273565	Available

-- Insert Data into BorrowTransactions Table

INSERT INTO BorrowTransactions (TransactionID, CustomerID, BookID, BorrowDate, DueDate)

VALUES (1001, 123, 101, '2025-1-05', '2025-1-15');

TransactionID	CustomerID	BookID	BorrowDate	DueDate	ReturnDate
1001	123	101	2025-1-05	2025-1-15	NULL

-- Insert Data into Payments Table

INSERT INTO Payments (PaymentID, TransactionID, Amount, PaymentDate)

VALUES (2001, 1001, 5.00, '2025-1-16');

PaymentID	TransactionID	Amount	PaymentDate
2001	1001	5.00	2025-1-16

-- Insert Data into Inventory Table

INSERT INTO Inventory (BookID, StockQuantity, BookCondition)

VALUES (101, 5, 'New');

BookID	StockQuantity	BookCondition
101	5	New

7.0 SUMMARY

In this phase, we focused on improving and refining the database design for the Online Library Management System (OLMS). The first step was transforming the Conceptual Entity Relationship Diagram (ERD) into a Logical ERD. This involved carefully analyzing the relationships between different entities, simplifying complex connections, and removing any non-relational features. This process ensured that the database is well-organized, easy to use, and capable of meeting the needs of the system.

We also worked on creating the relational schema, which converted important entities like customers, books, borrowing, payments, and inventory into detailed tables. Each table was designed with clear attributes and primary keys to make sure the data is easy to identify and retrieve. To make the database as efficient and reliable as possible, we applied normalization techniques. This helped us remove any unnecessary duplicates, minimize data redundancy, and ensure the database followed best practices for maintaining accuracy and integrity. These changes are critical for ensuring user and book information is consistent and dependable.

The Logical ERD we finalized serves as a visual blueprint of how all the different parts of the database work together. It clearly shows the relationships between entities and provides a roadmap for organizing and managing data effectively. In addition, the data dictionary was updated to reflect these changes, ensuring that all attributes, relationships, and table structures remain consistent and aligned with the improved database design.

Overall, this phase taught us how to turn initial conceptual ideas into a practical, well-structured database design. The updated database will make the OLMS more efficient, helping to automate tasks like tracking book loans, managing inventory, and processing payments. By improving these processes, the system will save time for library staff and provide a better, more seamless experience for users. This refined database structure lays the foundation for a reliable, user-friendly library system that supports the daily needs of both staff and patrons.