RDBMS and SQL

Assignment 1: Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form

Scenario: Restaurant Management System

The restaurant has menu items, each identified by a unique item ID, with attributes such as name, description, and price. Customers can place orders for menu items. Each order can contain multiple menu items, and each menu item can be included in multiple orders.

Entities:

1. Item

ItemID (Primary Key)

Name

Description

Price

2.Order

OrderID (Primary Key)

Date

TableNumber

3.Order Item

OrderItemID (Primary Key)

OrderID (Foreign Key)

ItemID (Foreign Key)

Quantity

Relationships:

1.Order includes Order Item

One order can include many order items.

Each order item belongs to exactly one order.

2.Menu Item is listed in Order Item

menu item can be listed in many order items.

Each order item corresponds to exactly one menu item.

Attributes:

Menu Item: ItemID, Name, Description, Price

Order: OrderID, Date, TableNumber

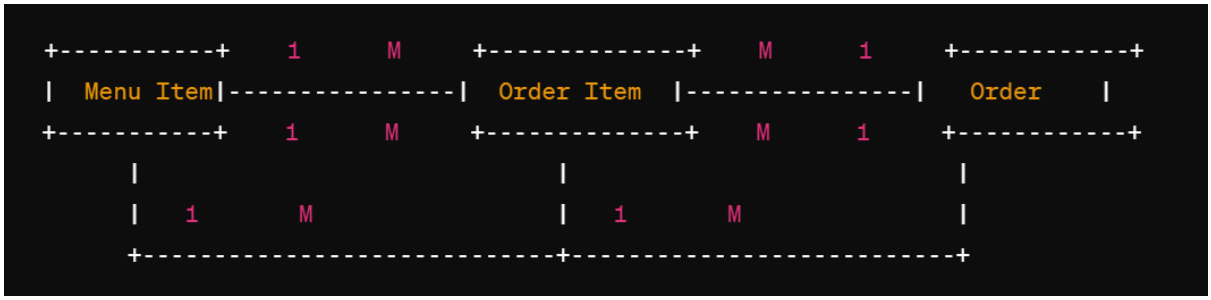
Order Item: OrderItemID, OrderID, ItemID, Quantity

Cardinality:

Menu Item - Order Item: One-to-Many (A menu item can be listed in multiple order items, but each order item corresponds to only one menu item)

Order - Order Item: One-to-Many (An order can include multiple order items, but each order item belongs to only one order)

ER diagram



Assignment 2: Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

Database schema for a library system

Tables:

Books

book_id (Primary Key)

title

author

ISBN (UNIQUE)

Authors

author_id (Primary Key)

author_name

Members

member_id (Primary Key)

name

Transactions

transaction_id (Primary Key)

book_id (Foreign Key)

member_id (Foreign Key)

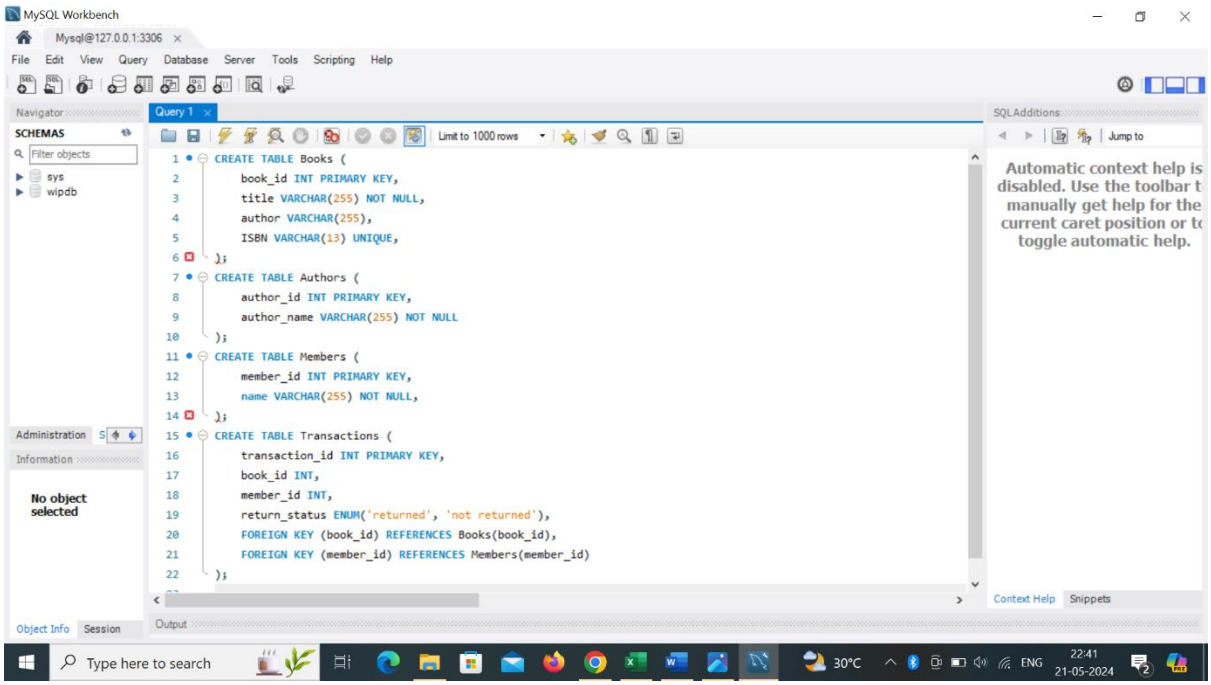
return_status (CHECK: "returned" or "not returned")

Primary and Foreign Keys:

Primary Keys: Unique identifiers for each record.

book_id for Books, author_id for Authors, member_id for Members, and transaction_id for Transactions.

Foreign Keys: Establish relationships between tables



Assignment 3: Explain the ACID properties of a transaction in your own words. Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.

ACID Properties:

Atomicity: Transactions are all or nothing. They either complete successfully, and all changes are saved (committed), or if there's an error, they're rolled back to their original state (aborted). It's like a single unit of work that can't be divided.

Consistency: Transactions ensure that the database remains in a consistent state before and after their execution. This means that data integrity rules, constraints, and relationships are preserved. If a transaction violates any of these rules, it's rolled back to maintain consistency.

Isolation: Transactions run independently of each other, even if they're executed concurrently. Isolation ensures that the intermediate states of transactions are not visible to other transactions until they're committed. This prevents interference and ensures each transaction sees a consistent view of the database.

Durability: Once a transaction is committed, its changes are permanent and will survive system failures. This means that even if the system crashes, the changes made by committed transactions will be preserved and can't be undone.

SQL Statements to Simulate a Transaction with Locking:

START TRANSACTION;

UPDATE accounts SET balance = balance - 100 WHERE account_id = 'sender_account_id';

UPDATE accounts SET balance = balance + 100 WHERE account_id = 'receiver_account_id';

COMMIT;

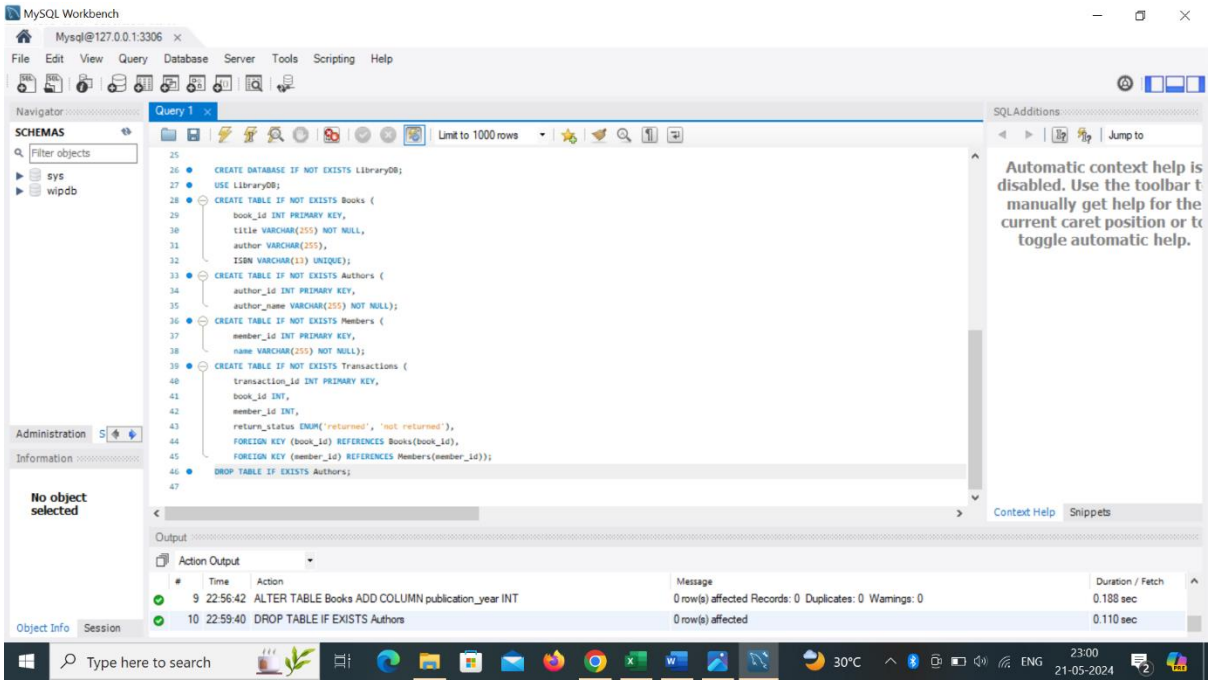
Demonstrating Different Isolation Levels:

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

Assignment 4: Write SQL statements to CREATE a new database and tables that reflect the library schema you designed earlier. Use ALTER statements to modify the table structures and DROP statements to remove a redundant table.



Assignment 2: Demonstrate the creation of an index on a table and discuss how it improves query performance. Use a DROP INDEX statement to remove the index and analyze the impact on query execution

Creating an Index:

Indexes in databases are used to speed up the retrieval of rows from a table by creating a quick lookup structure

CREATE INDEX idx_title ON Books (title);

Improve Query Performance:

When you execute a query that involves searching, sorting, or filtering based on the indexed column (in this case, title), the database engine can use the index to quickly locate the relevant rows without scanning the entire table sequentially.

SELECT * FROM Books WHERE title = 'xyz';

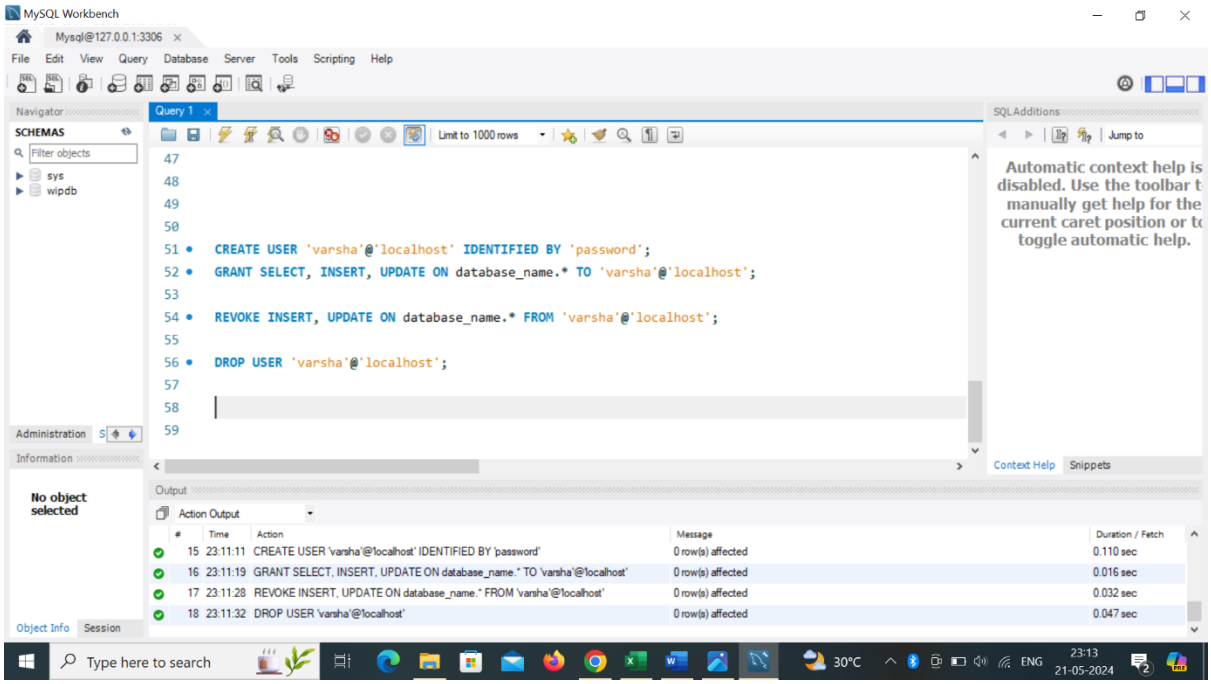
Dropping the Index:

DROP INDEX IF EXISTS idx_title;

Impact on Query Execution:

Without the index on the column, the database engine will have to perform a full table scan when executing queries that involve searching, sorting, or filtering based on the column. This can result in slower query performance, especially for large tables, as the database has to examine every row in the table to find the relevant ones. dropping an index can lead to slower query execution, particularly for queries that rely on the indexed column.

Assignment 6: Create a new database user with specific privileges using the CREATE USER and GRANT commands. Then, write a script to REVOKE certain privileges and DROP the user.



Assignment 7: Prepare a series of SQL statements to INSERT new records into the library tables, UPDATE existing records with new information, and DELETE records based on specific criteria. Include BULK INSERT operations to load data from an external source