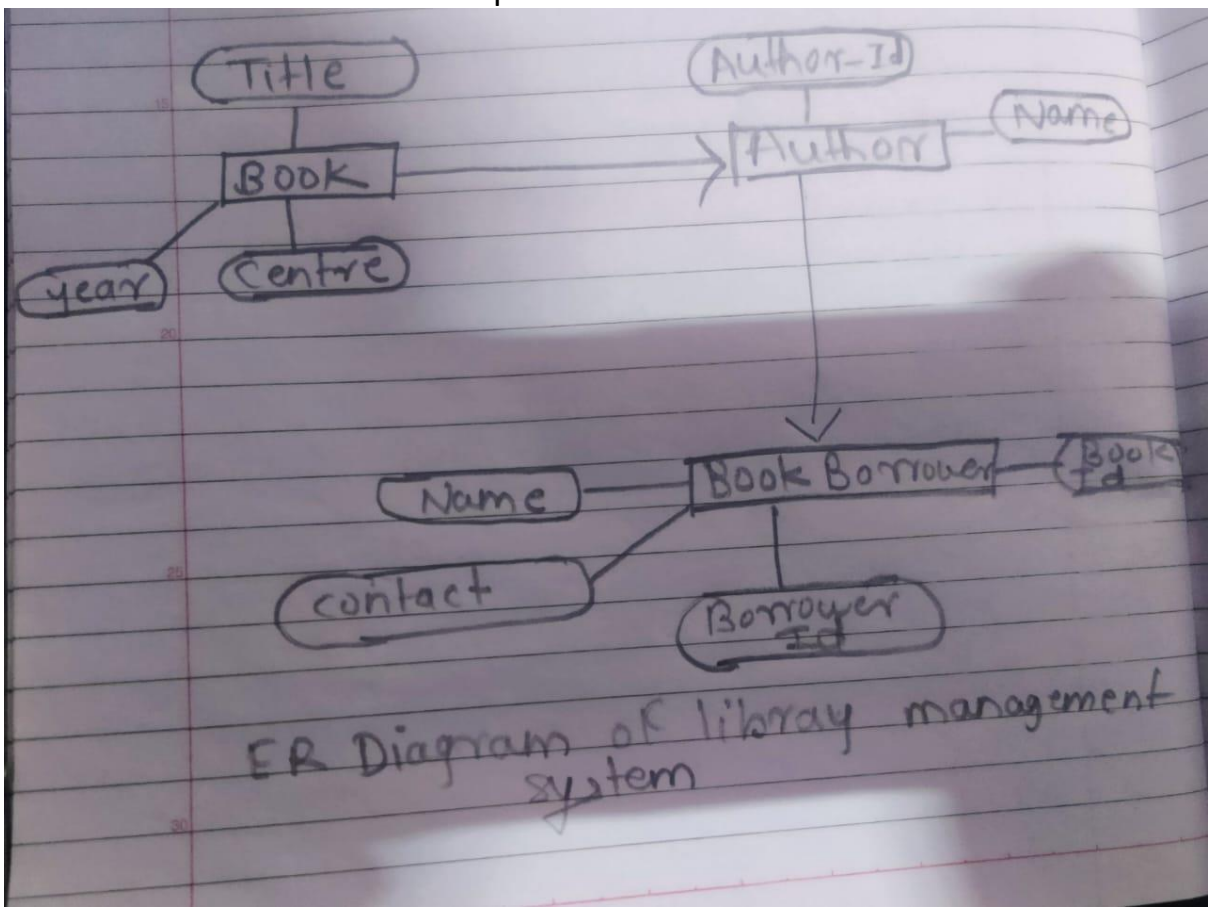
Assignment 1: Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.

Identify Entities and Attributes:
Start by brainstorming the main objects or concepts that hold relevant information
for your business. These become your entities.
For each entity, list the descriptive characteristics or properties you want to store.
These are the attributes.
Example Scenario (Library Management System):
Entities:
Book
Author
Borrower
Attributes:
Book: ISBN, Title, Publication Year, Genre
Author: Author ID (primary key), Name, Nationality
Borrower: Borrower ID (primary key), Name, Contact Information
2. Define Relationships:
Consider how entities interact with each other. A relationship represents an association between two or more entities.
Relationships can be one-to-one (1:1), one-to-many (1:M), or many-to-many (M:N).
Example Scenario Relationships:
A Book can be written by one Author (1:M).
An Author can write many Books (M:1).
A Borrower can borrow many Books (M:N).
A Book can be borrowed by many Borrowers (M:N).
3. Normalize the ER Diagram:
Normalization is a process to minimize data redundancy and improve data integrity
in a database. There are three main normal forms (1NF, 2NF, and 3NF) with increasing levels of normalization.
1NF (First Normal Form): Eliminates repeating groups within an entity.
2NF (Second Normal Form): Ensures no partial dependencies on the primary key.
3NF (Third Normal Form): Eliminates transitive dependencies on the primary key.
Normalization Steps for the Library Example:

1NF: We already have 1NF as there are no repeating groups.
2NF: No partial dependencies exist based on primary keys (Author ID and Borrower ID).
3NF: The Borrower entity might have a transitive dependency on Book through the
Author entity. To address this, we can create a separate entity Book_Borrower to
link Book and Borrower with their own primary key and eliminate the dependency.
4. Create the ER Diagram:
Use standard ERD symbols: Rectangles for entities, diamonds for relationships, ovals for attributes.
Label entities, attributes, and cardinalities (1:1, 1:M, M:N) on the connecting lines
between entities and relationships.

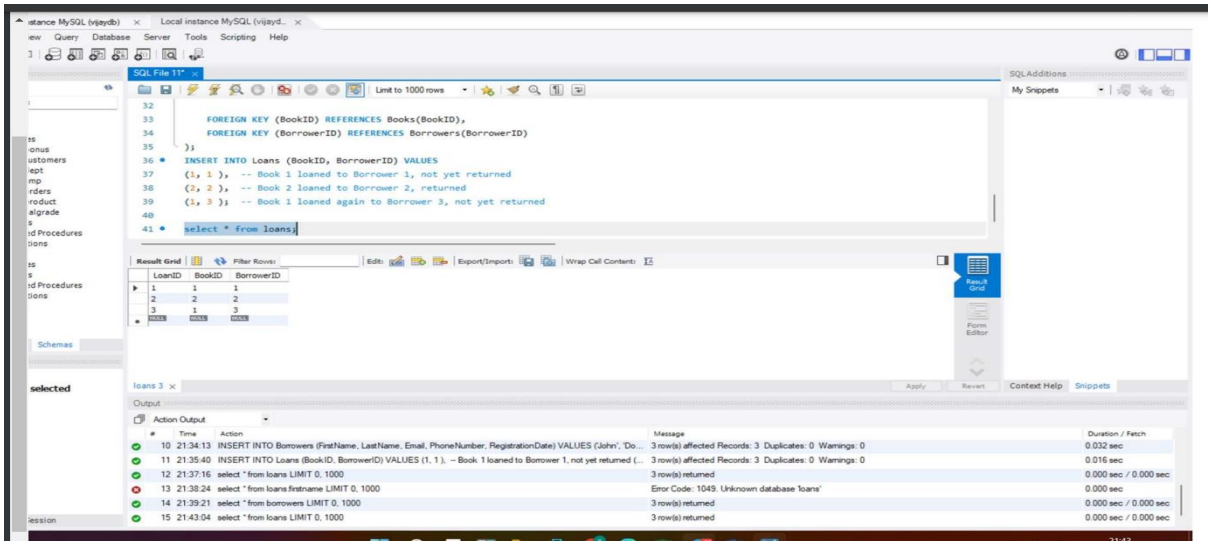


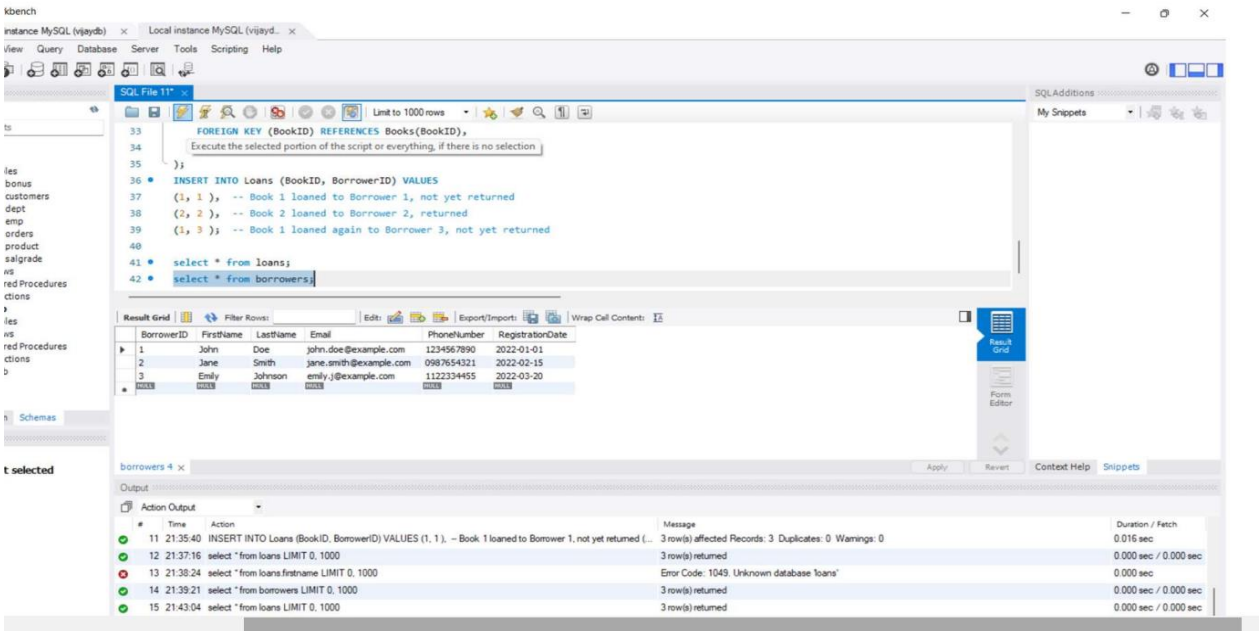
Assignment 2: Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

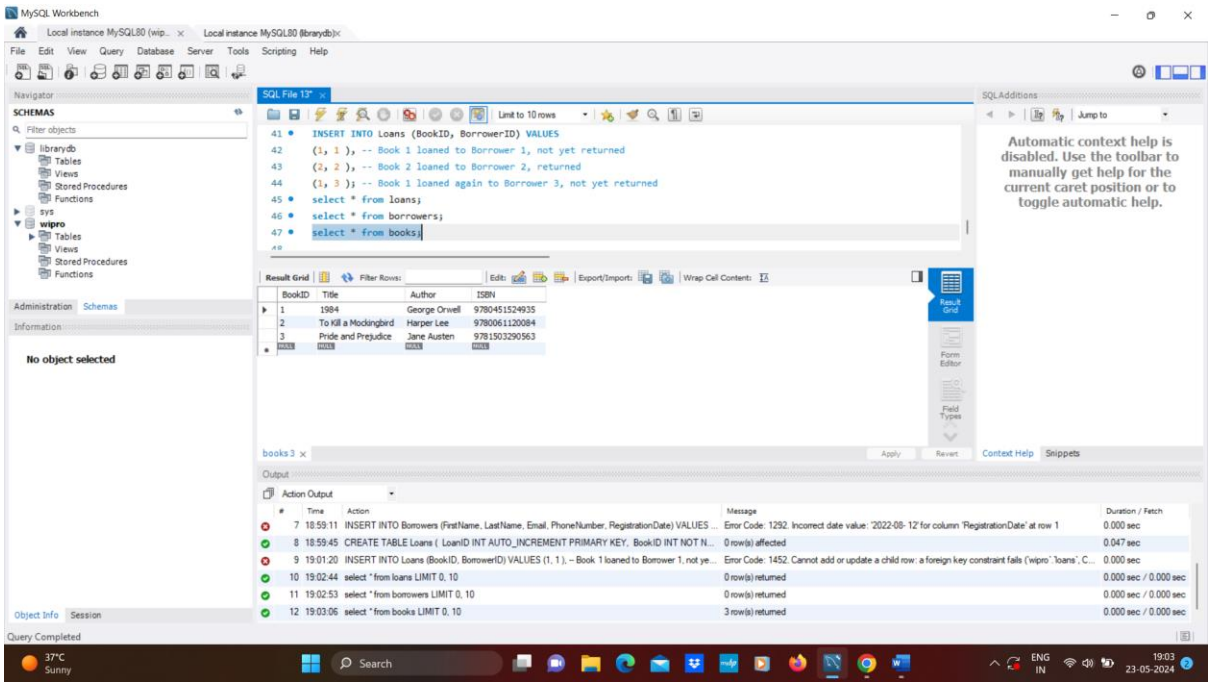
CREATE TABLE Books (

```
 BookID INT AUTO_INCREMENT PRIMARY KEY,
 Title VARCHAR(255) NOT NULL,
 Author VARCHAR(255) NOT NULL,
 ISBN VARCHAR(20) NOT NULL UNIQUE
 );
 INSERT INTO Books (Title, Author, ISBN) VALUES
('1984', 'George Orwell', '9780451524935'),
('To Kill a Mockingbird', 'Harper Lee', '9780061120084'),
('Pride and Prejudice', 'Jane Austen', '9781503290563');
 drop table books;
 CREATE TABLE Borrowers (
 BorrowerID INT AUTO_INCREMENT PRIMARY KEY,
 FirstName VARCHAR(255) NOT NULL,
 LastName VARCHAR(255) NOT NULL,
Email VARCHAR(255) UNIQUE,
 PhoneNumber VARCHAR(15),
 RegistrationDate DATE NOT NULL
);
INSERT INTO Borrowers (FirstName, LastName, Email,
PhoneNumber, RegistrationDate) VALUES
('John', 'Doe', 'john.doe@example.com', '1234567890', '2022-01-
01'),
('Jane', 'Smith', 'jane.smith@example.com', '0987654321', '2022-02-
15'),
('Emily', 'Johnson', 'emily.j@example.com', '1122334455', '2022-03-
20');
```

```
CREATE TABLE Loans (
 LoanID INT AUTO_INCREMENT PRIMARY KEY,
 BookID INT NOT NULL,
 BorrowerID INT NOT NULL,

 FOREIGN KEY (BookID) REFERENCES Books(BookID),
 FOREIGN KEY (BorrowerID) REFERENCES Borrowers(BorrowerID)
);
INSERT INTO Loans (BookID, BorrowerID) VALUES
(1, 1 ), -- Book 1 loaned to Borrower 1, not yet returned
(2, 2 ), -- Book 2 loaned to Borrower 2, returned
(1, 3 ); -- Book 1 loaned again to Borrower 3, not yet returned
select * from loans;
select * from borrowers;
select * from books;
```







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.

Atomicity:

Ensures that a transaction is all or nothing. If any part of the transaction fails, the entire transaction is rolled back, leaving the database in its original state.
Consistency:

Ensures that a transaction brings the database from one valid state to another, maintaining the integrity constraints of the database.
Isolation:

Ensures that transactions are executed in isolation from one another. Intermediate states of a transaction are not visible to other transactions until the transaction is committed.
Durability:

Ensures that once a transaction has been committed, it will remain so, even in the event of a system failure. The changes made by the transaction are permanently stored in the database.
Transaction Simulation with Explicit Locking

```
START TRANSACTION;
-- Lock the rows in the Accounts table for Alice and Bob
SELECT * FROM Accounts WHERE AccountID IN (1, 2) FOR UPDATE;
-- Deduct from Alice's account
UPDATE Accounts SET Balance = Balance - 200 WHERE AccountID = 1;
-- Add to Bob's account
UPDATE Accounts SET Balance = Balance + 200 WHERE AccountID =
2;
COMMIT;
```

N n ,n,mnkmml/

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.

```
CREATE TABLE Books (
 BookID INT AUTO_INCREMENT PRIMARY KEY,
 Title VARCHAR(255) NOT NULL,
 Author VARCHAR(255) NOT NULL,
 ISBN VARCHAR(20) NOT NULL UNIQUE
 );
 INSERT INTO Books (Title, Author, ISBN) VALUES
('1984', 'George Orwell', '9780451524935'),
('To Kill a Mockingbird', 'Harper Lee', '9780061120084'),
('Pride and Prejudice', 'Jane Austen', '9781503290563');
```

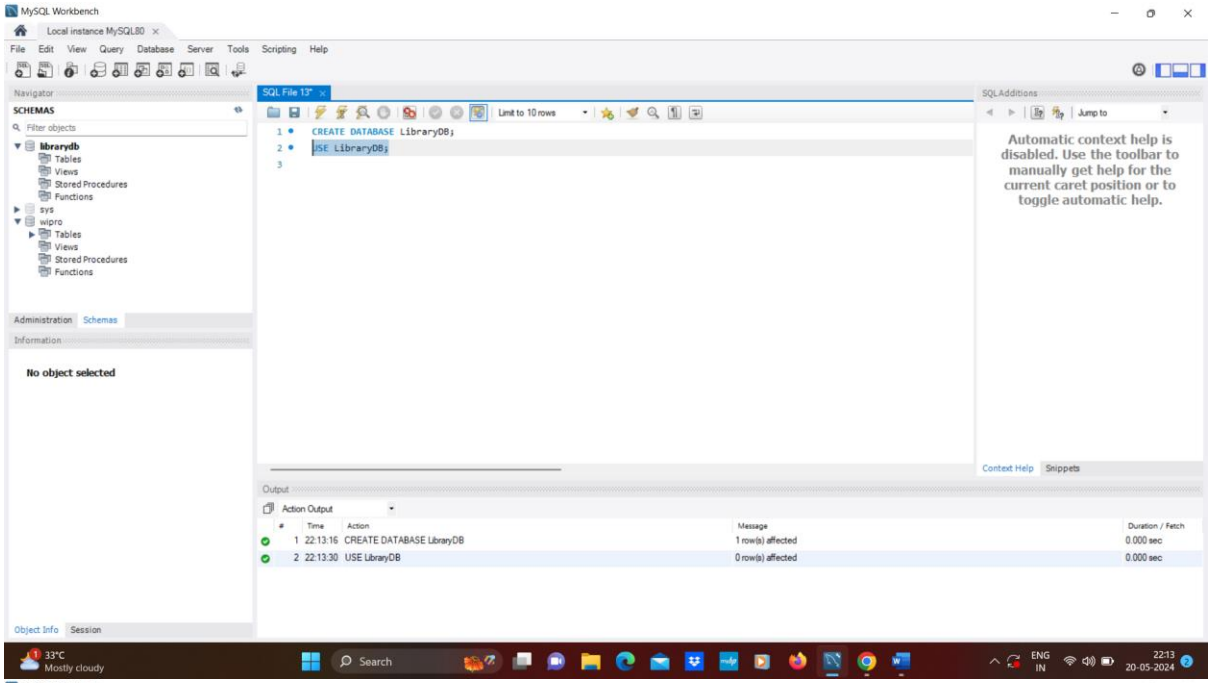
```
 drop table books;
 CREATE TABLE Borrowers (
 BorrowerID INT AUTO_INCREMENT PRIMARY KEY,
 FirstName VARCHAR(255) NOT NULL,
 LastName VARCHAR(255) NOT NULL,
 Email VARCHAR(255) UNIQUE,
 PhoneNumber VARCHAR(15),
 RegistrationDate DATE NOT NULL
);
INSERT INTO Borrowers (FirstName, LastName, Email,
PhoneNumber, RegistrationDate) VALUES
('John', 'Doe', 'john.doe@example.com', '1234567890', '2022-01-
01'),
('Jane', 'Smith', 'jane.smith@example.com', '0987654321', '2022-02-
15'),
('Emily', 'Johnson', 'emily.j@example.com', '1122334455', '2022-03-
20');
CREATE TABLE Loans (
 LoanID INT AUTO_INCREMENT PRIMARY KEY,
 BookID INT NOT NULL,
 BorrowerID INT NOT NULL,

 FOREIGN KEY (BookID) REFERENCES Books(BookID),
 FOREIGN KEY (BorrowerID) REFERENCES Borrowers(BorrowerID)
);
INSERT INTO Loans (BookID, BorrowerID) VALUES
(1, 1 ),
(2, 2 ),
(1, 3 );
```

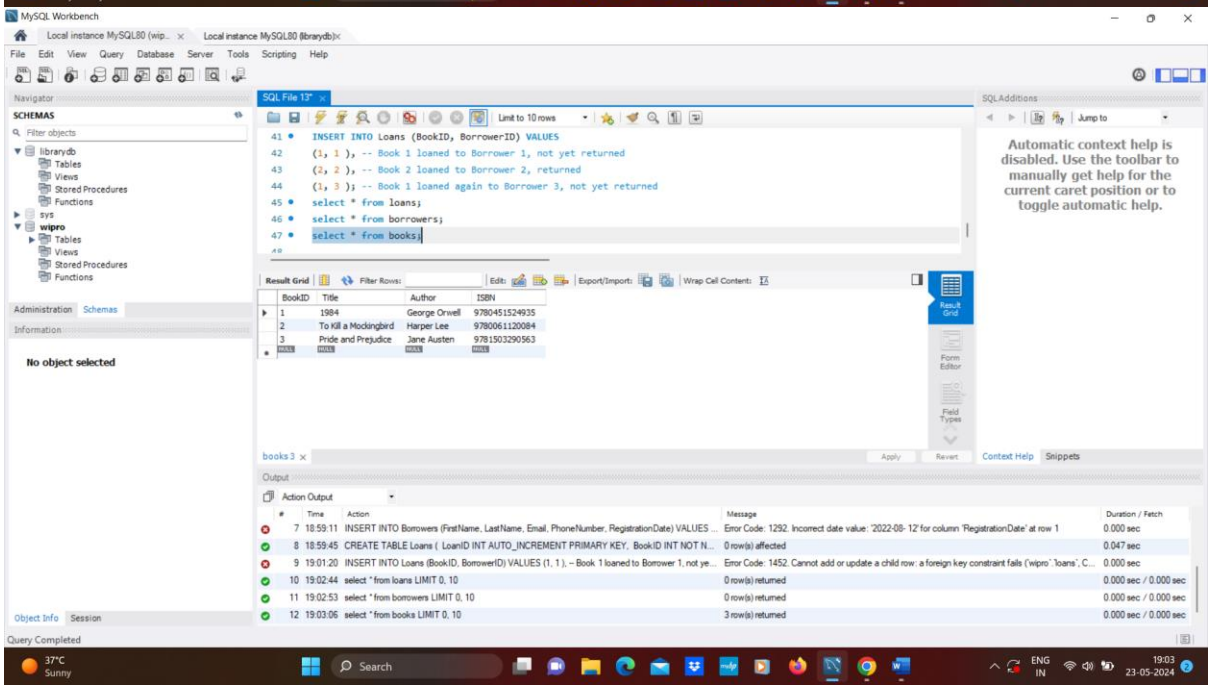
Step2:

Alter statement use

For add one extra column in table

For rename the column name





Step4 :

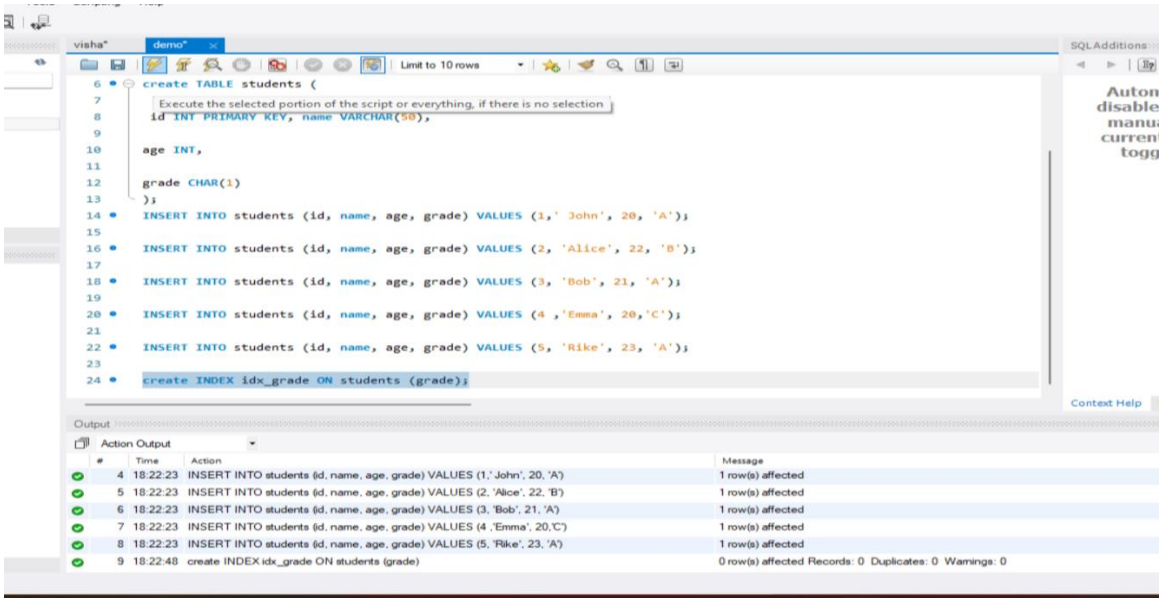
Use of drop command

Drop table xyz;

Assignment 5: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.

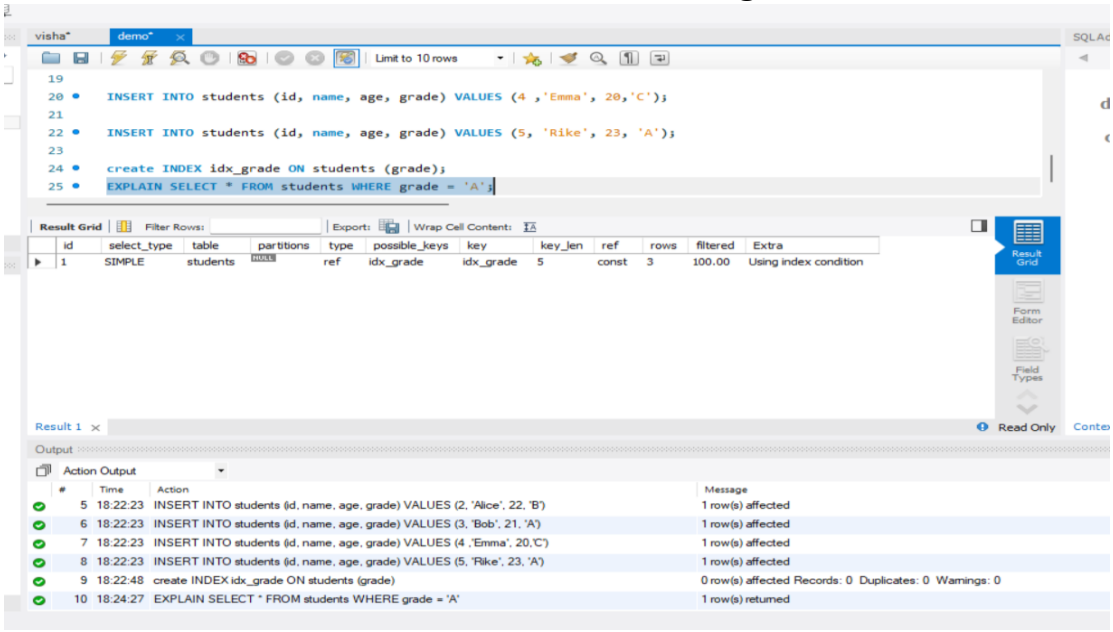
Create an index on the 'grade' column

CREATE INDEX idx_grade ON students (grade);

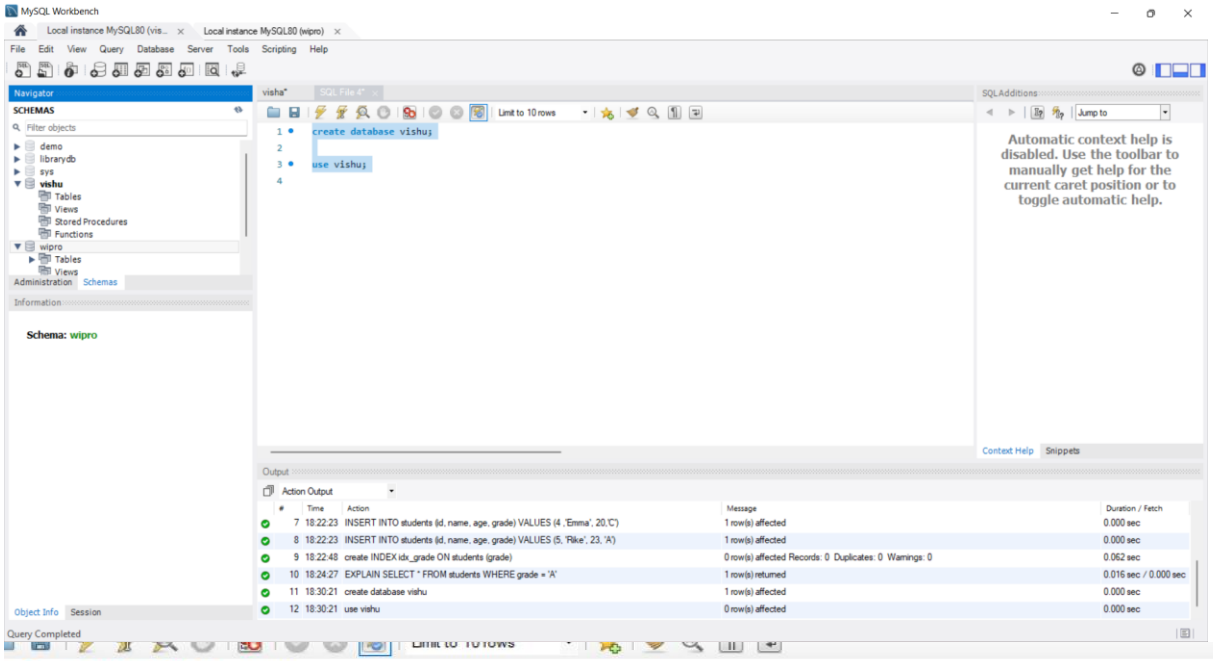


Query without index

EXPLAIN SELECT * FROM students WHERE grade = 'A';



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.

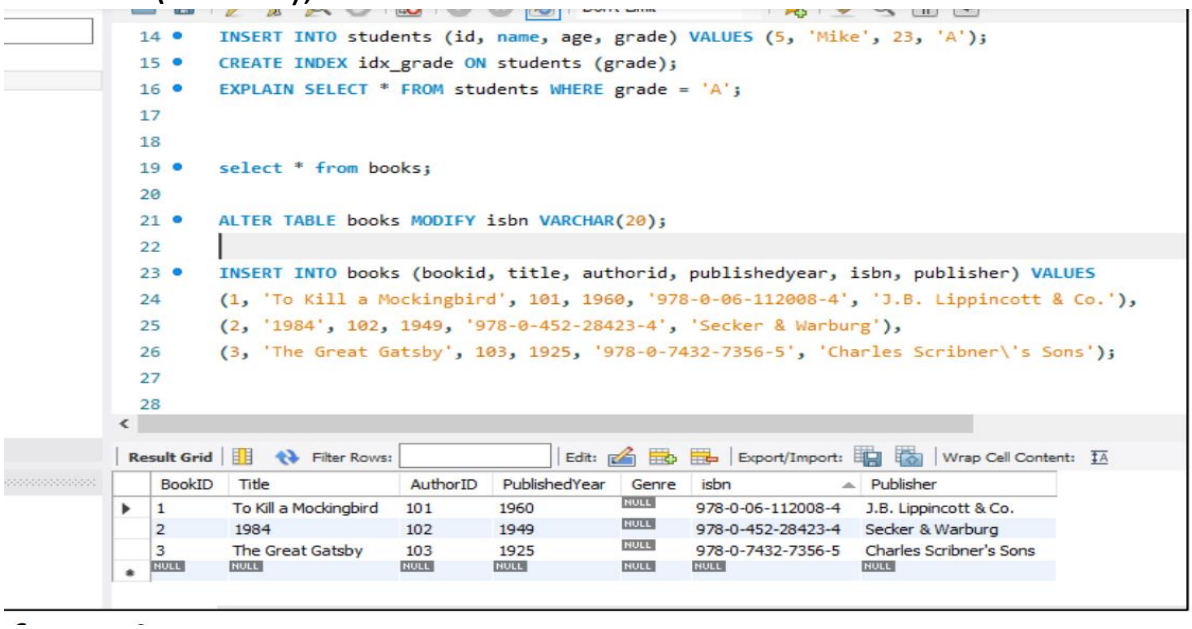



```
1  create database vishu;
2
3  use vishu;
4  create user 'vishu'@'localhost' identified by 'password';
5
6
7
8
9  GRANT ALL PRIVILEGES ON wipro. To 'vishu'@'localhost';
10
11 revoke ALL PRIVILEGES ON wipro.* FROM 'vishu'@'localhost";
12
13
14 drop user 'vishu'@'localhost';
```

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.

INSERT INTO books (bookid, title, authorid, publishedyear, isbn, publisher)
VALUES
(1, 'To Kill a Mockingbird', 101, 1960, '978-0-06-112008-4', 'J.B. Lippincott &
Co.'),

(2, '1984', 102, 1949, '978-0-452-28423-4', 'Secker & Warburg'),
(3, 'The Great Gatsby', 103, 1925, '978-0-7432-7356-5', 'Charles Scribner\'s Sons');

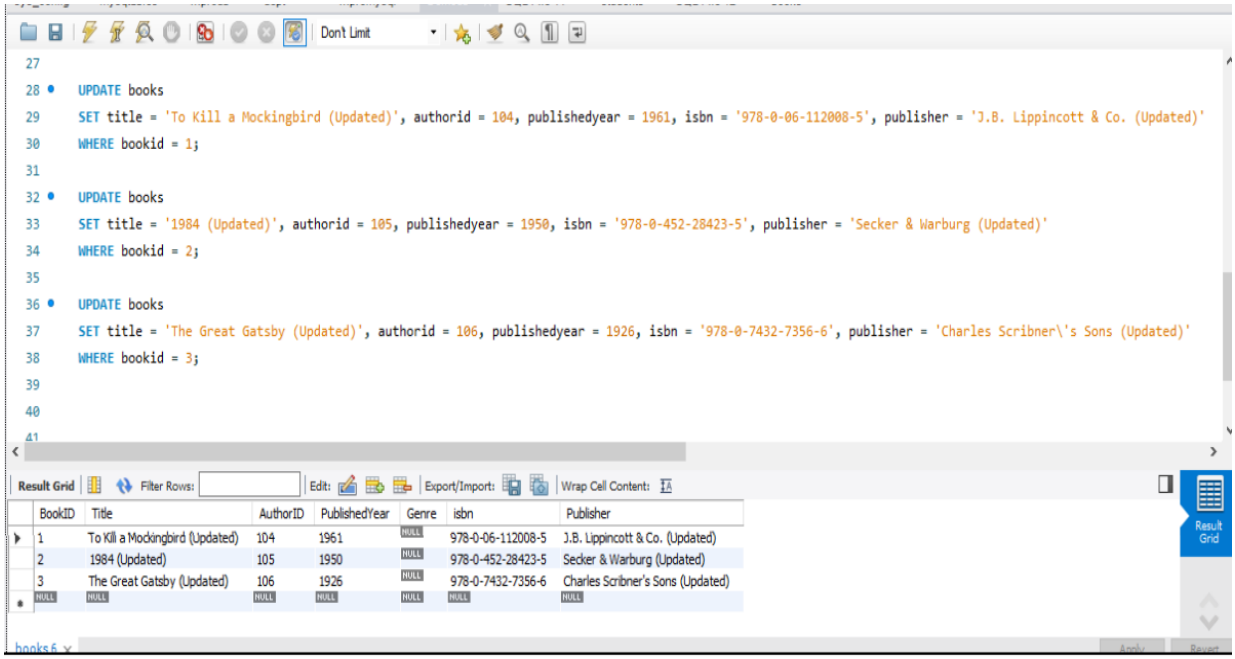


Update book information

UPDATE books
SET title = 'To Kill a Mockingbird (Updated)', authorid = 104, publishedyear =
1961, isbn = '978-0-06-112008-5', publisher = 'J.B. Lippincott & Co. (Updated)'
WHERE bookid = 1;

UPDATE books
SET title = '1984 (Updated)', authorid = 105, publishedyear = 1950, isbn = '978-0-
452-28423-5', publisher = 'Secker & Warburg (Updated)'
WHERE bookid = 2;

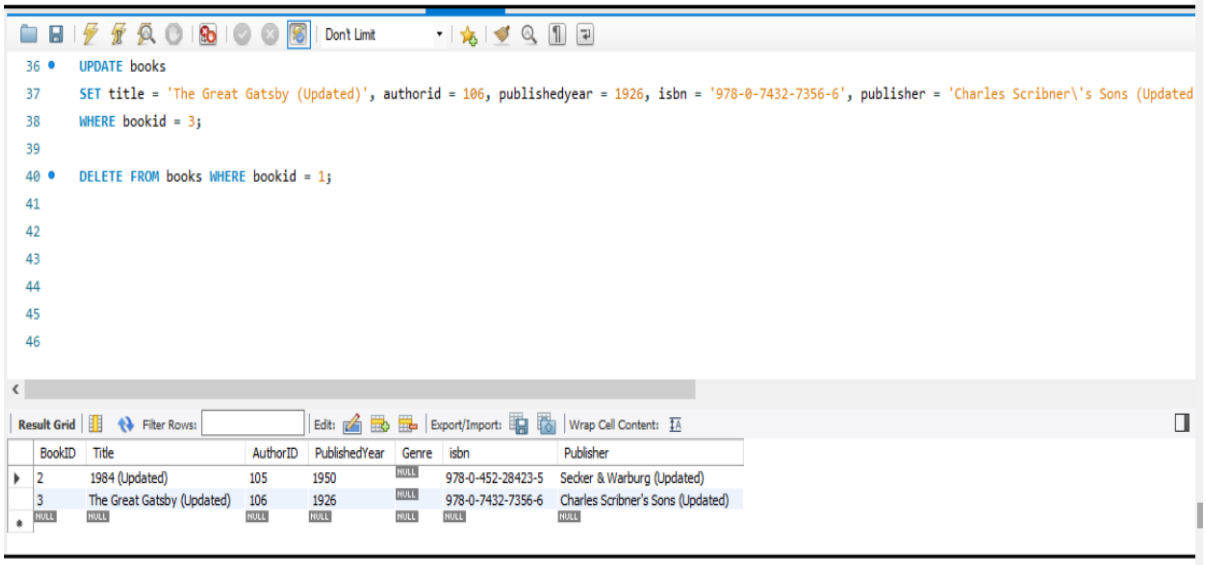
UPDATE books
SET title = 'The Great Gatsby (Updated)', authorid = 106, publishedyear = 1926,
isbn = '978-0-7432-7356-6', publisher = 'Charles Scribner\'s Sons (Updated)'
WHERE bookid = 3;



```
UPDATE books
SET title = 'To Kill a Mockingbird (Updated)', authorid = 104, publishedyear = 1961, isbn = '978-0-06-112008-5', publisher = 'J.B. Lippincott & Co. (Updated)'
WHERE bookid = 1;

UPDATE books
SET title = '1984 (Updated)', authorid = 105, publishedyear = 1950, isbn = '978-0-452-28423-5', publisher = 'Secker & Warburg (Updated)'
WHERE bookid = 2;

UPDATE books
SET title = 'The Great Gatsby (Updated)', authorid = 106, publishedyear = 1926, isbn = '978-0-7432-7356-6', publisher = 'Charles Scribner\'s Sons (Updated)'
WHERE bookid = 3;
```

BookID	Title	AuthorID	PublishedYear	Genre	isbn	Publisher
1	To Kill a Mockingbird (Updated)	104	1961	NULL	978-0-06-112008-5	J.B. Lippincott & Co. (Updated)
2	1984 (Updated)	105	1950	NULL	978-0-452-28423-5	Secker & Warburg (Updated)
3	The Great Gatsby (Updated)	106	1926	NULL	978-0-7432-7356-6	Charles Scribner's Sons (Updated)

Delete a book record



```
UPDATE books
SET title = 'The Great Gatsby (Updated)', authorid = 106, publishedyear = 1926, isbn = '978-0-7432-7356-6', publisher = 'Charles Scribner\'s Sons (Updated
WHERE bookid = 3;

DELETE FROM books WHERE bookid = 1;
```

BookID	Title	AuthorID	PublishedYear	Genre	isbn	Publisher
2	1984 (Updated)	105	1950	NULL	978-0-452-28423-5	Secker & Warburg (Updated)
3	The Great Gatsby (Updated)	106	1926	NULL	978-0-7432-7356-6	Charles Scribner's Sons (Updated)

BULK INSERT Books
FROM 'C:\path\to\books.csv'
WITH (
 FIELDTERMINATOR = ',',
 ROWTERMINATOR = '\n',
 FIRSTROW = 2 -- If the first row contains headers
);
Bulk insert data into the Members table from a CSV file
BULK INSERT Members
FROM 'C:\path\to\members.csv'

```
WITH (
 FIELDTERMINATOR = ',',
 ROWTERMINATOR = '\n',
 FIRSTROW = 2 -- If the first row contains headers
);
Bulk insert data into the Loans table from a CSV file
BULK INSERT Loans
FROM 'C:\path\to\loans.csv'
WITH (
 FIELDTERMINATOR = ',',
 ROWTERMINATOR = '\n',
 FIRSTROW = 2 -- If the first row contains headers
);
```