Md Yaseen

DAY 9 Assignment

Assignment-1:

Analyse 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.

SOLUTION:-

Let us analyse a business scenario for an University Course Registration System:

Entities:

1. **Course (Course_ID, Course_Name, Dept_ID, Credits)**

   Attributes:

   - Course_ID (primary key, unique identifier for a course)
   - Course_Name (name of the course)
   - Dept_ID (foreign key referencing the department that offers the course)
   - Credits (number of credits associated with the course)

2. **Department (Dept_ID, Dept_Name)**

   Attributes:

   - Dept_ID (primary key, unique identifier for a department)
   - Dept_Name (name of the department)

3. **Student (Stud_ID, Stud_Name, Major)**

   Attributes:

   - Stud_ID (primary key, unique identifier for a student)

   - Stud_Name (name of the student)
   - Major (student's major field of study)

4. **Registration (Registration_ID, Stud_ID, Course_ID, Semester)**

   Attributes:

   - Registration_ID (primary key, unique identifier for a registration)
   - Stud_ID (foreign key referencing the student enrolled in the course)
   - Course_ID (foreign key referencing the course taken by the student)
   - Semester (semester in which the student is registered for the course)

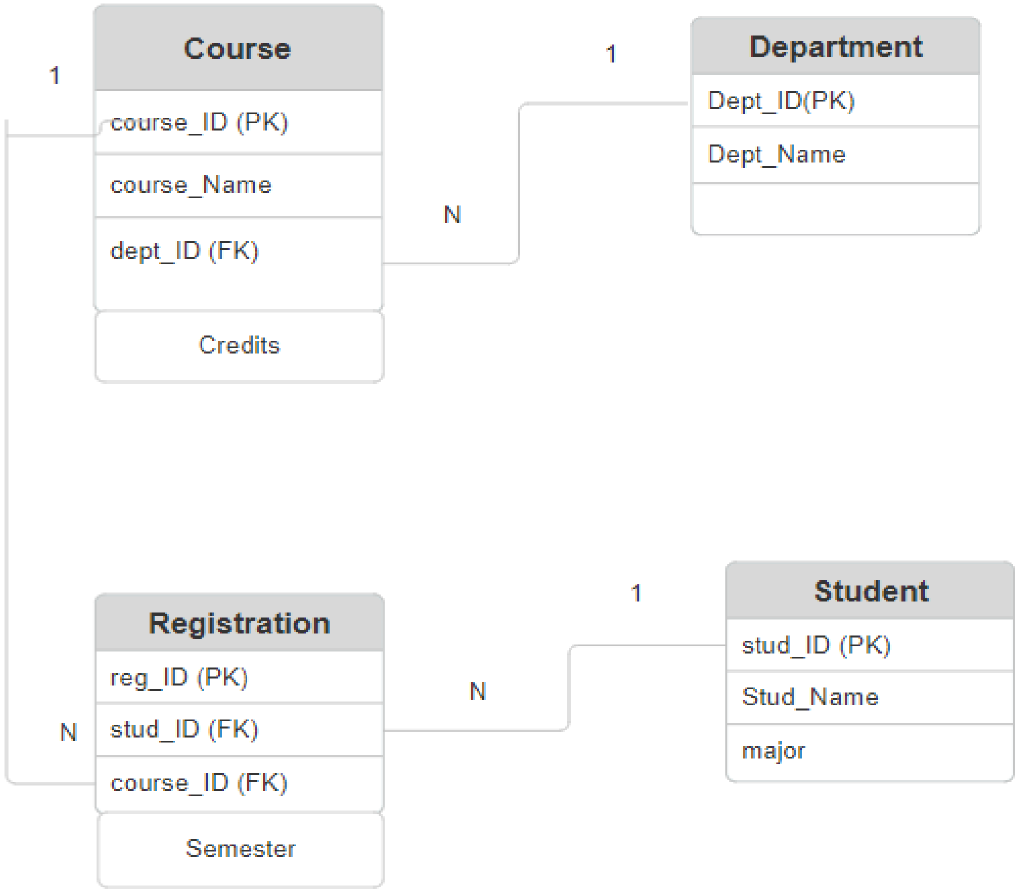
Relationships:

1. **A Department offers many Courses (one-to-many)**
2. **A Student can register for many Courses (one-to-many)**
3. **A Course can have many Students registered (one-to-many)**

Normalization (Third Normal Form):

- The Registration entity depends on both Stud_ID and Course_ID, which are both part of the primary key. This satisfies the definition of 3NF.
- There are no transitive dependencies (where the value of an attribute depends on another non-key attribute) in any of the entities.

ER-Diagram for above schema design:



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.

SOLUTION:-

Library System Database Schema:

Tables and Fields:

- **Books:** bookId (PK), title, author, publisher, yearPublished, ISBN (UNIQUE)
- **Members:** membID (PK), name, address, mobile, email (UNIQUE)
- **Borrow:** borrowID (PK), bookID (FK), membID (FK), borrowDate, returnDate
- **Authors:** authorID (PK), name, Bio
- **BookAuthors:** bookID (FK), authorID (FK)

Constraints:

- **NOT NULL :** Ensures necessary fields are not left empty.
- **Unique:** Ensure uniqueness where necessary.
- **CHECK:** Enforce rules (e.g., CHECK (YearPublished > 2000)).

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.

SOLUTION:

ACID Properties and Transaction Simulation

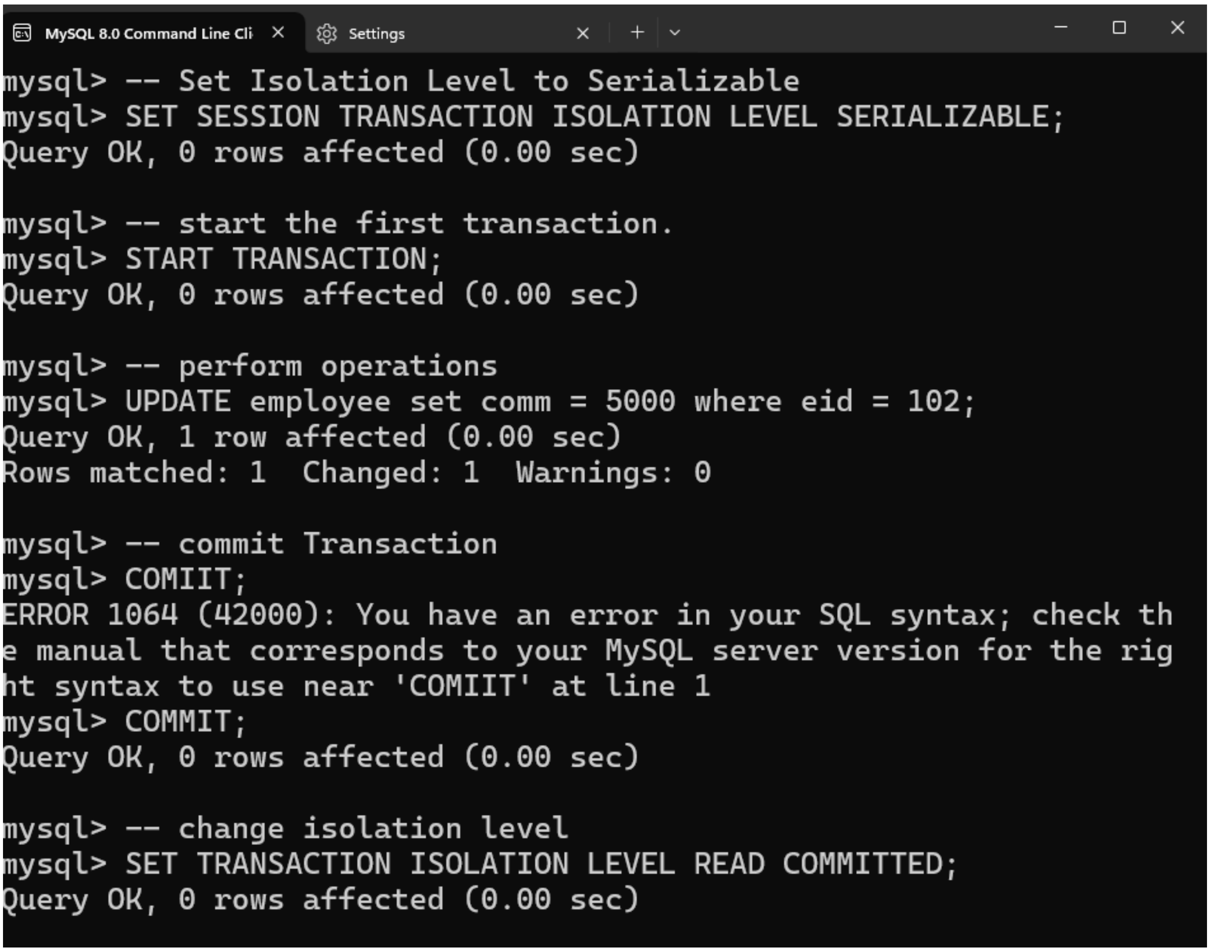
ACID Properties:

- **Atomicity:** Ensures that all operations in a transaction are completed; if not, the transaction is aborted.

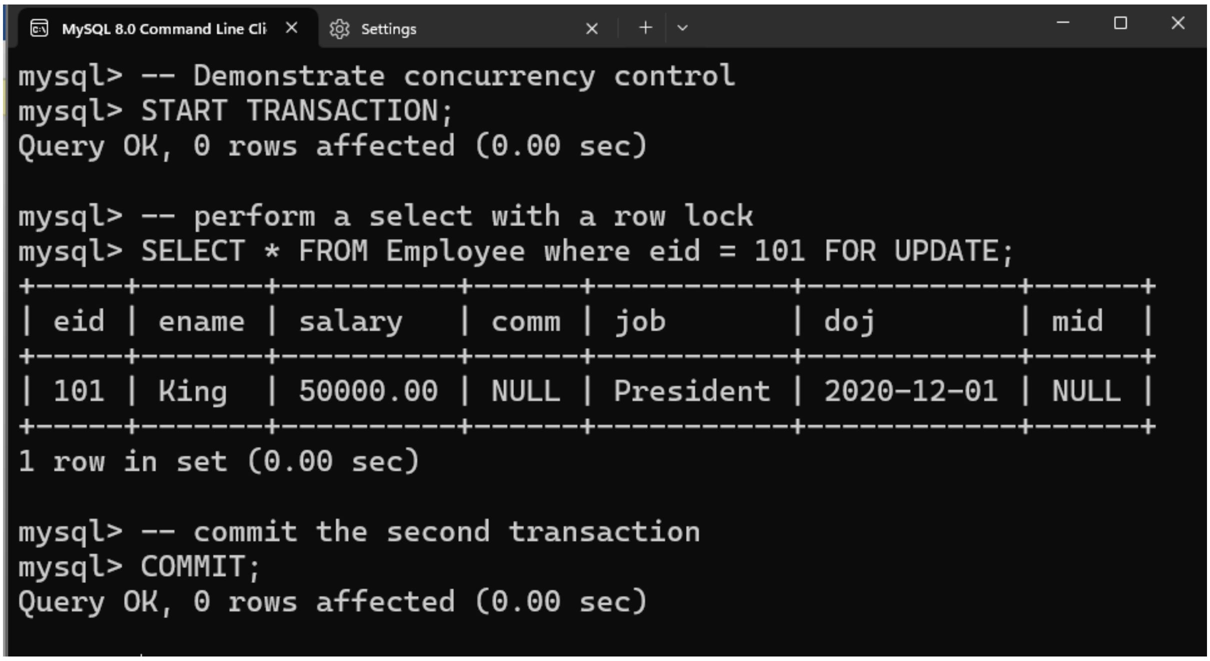
- **Consistency:** Ensures the database remains in a consistent state before and after the transaction.

- **Isolation:** Ensures that transactions are executed independently.

- **Durability:** Ensures that once a transaction is committed, it remains in the system even in case of a failure.



```
mysql> -- Set Isolation Level to Serializable
mysql> SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
Query OK, 0 rows affected (0.00 sec)

mysql> -- start the first transaction.
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> -- perform operations
mysql> UPDATE employee set comm = 5000 where eid = 102;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> -- commit Transaction
mysql> COMIIT;
ERROR 1064 (42000): You have an error in your SQL syntax; check th
e manual that corresponds to your MySQL server version for the rig
ht syntax to use near 'COMIIT' at line 1
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> -- change isolation level
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)
```



```
mysql> -- Demonstrate concurrency control
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> -- perform a select with a row lock
mysql> SELECT * FROM Employee where eid = 101 FOR UPDATE;
+-----+-------+----------+------+-----------+------------+------+
| eid | ename | salary   | comm | job       | doj        | mid  |
+-----+-------+----------+------+-----------+------------+------+
| 101 | King  | 50000.00 | NULL | President | 2020-12-01 | NULL |
+-----+-------+----------+------+-----------+------------+------+
1 row in set (0.00 sec)

mysql> -- commit the second transaction
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

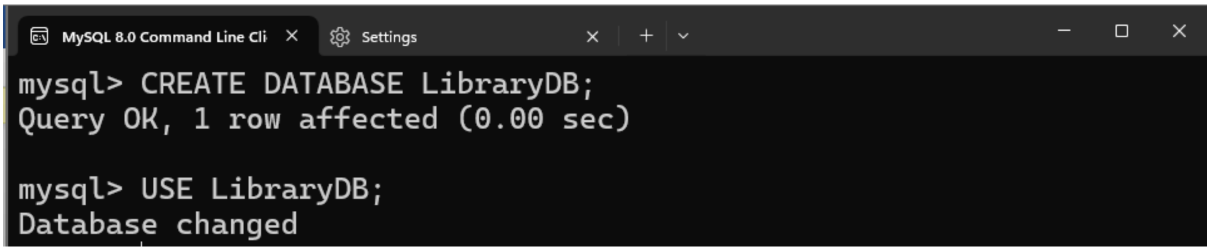
Assignment-4:

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.

SOLUTION:

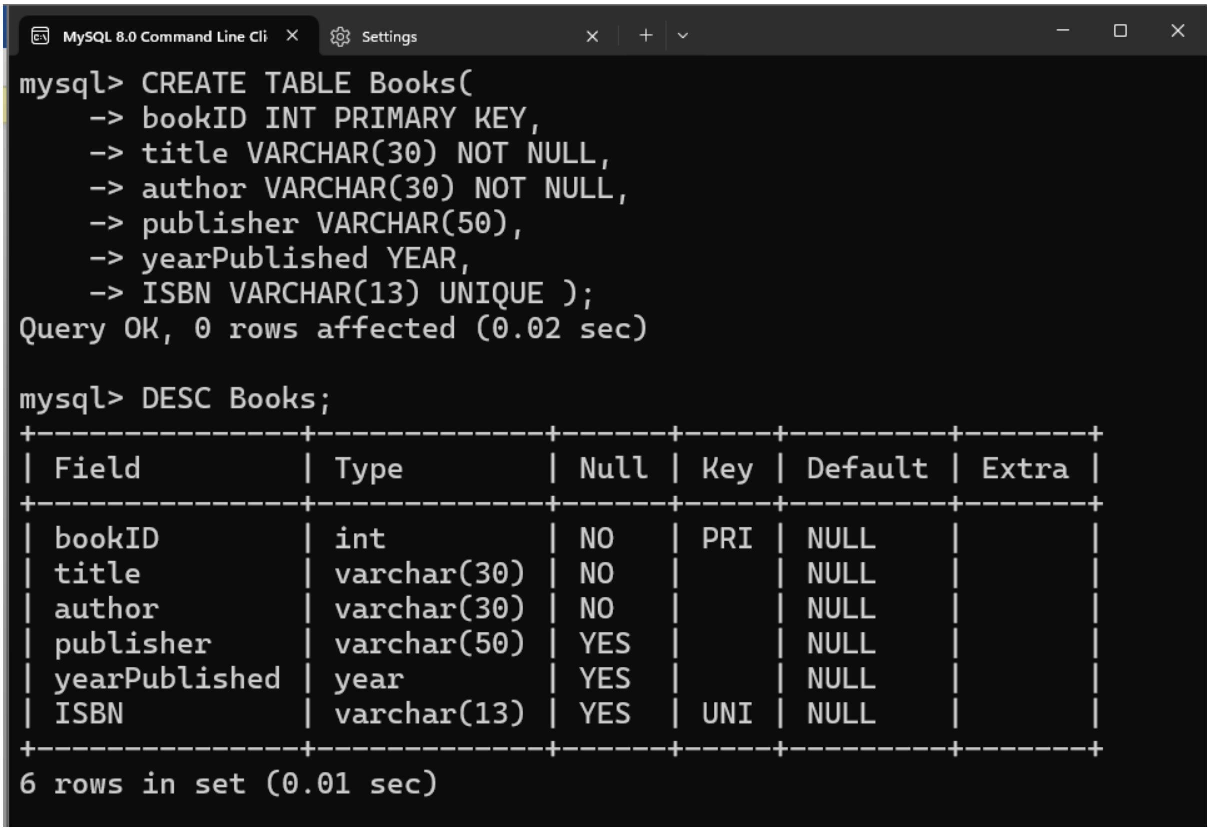
SQL Commands for Database and Table Creation.

- **Creating Database**



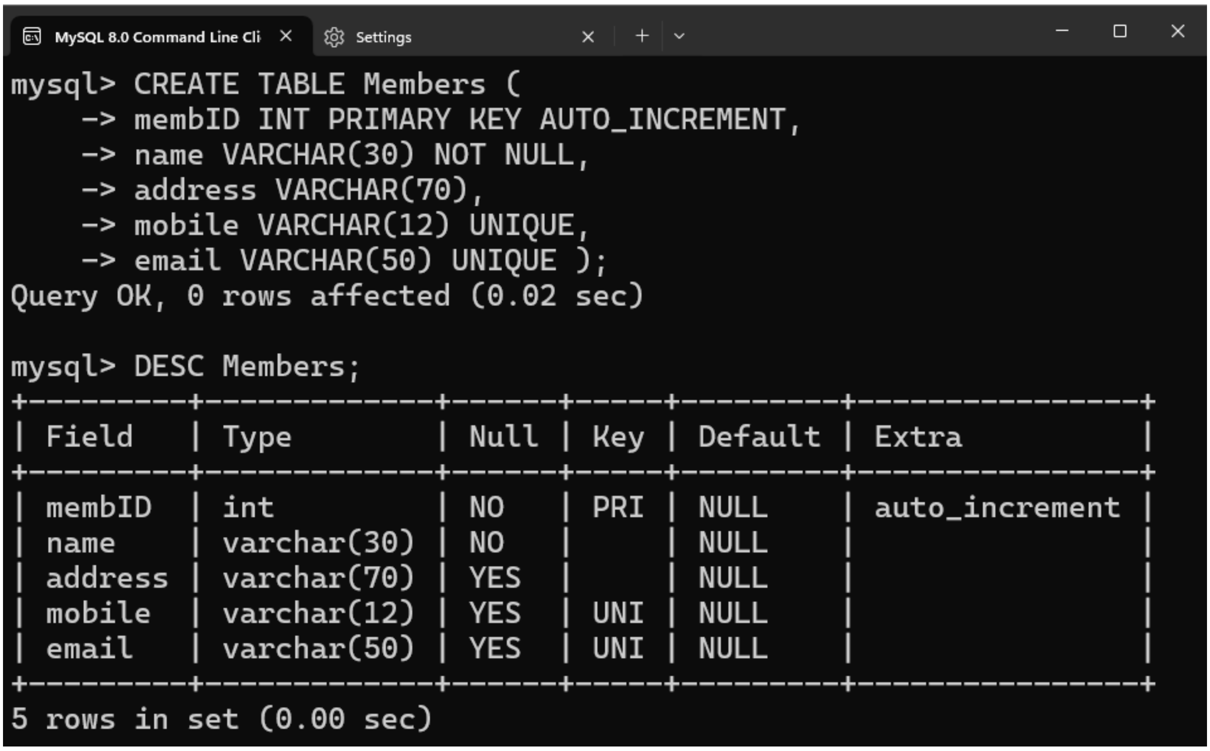
```
mysql> CREATE DATABASE LibraryDB;
Query OK, 1 row affected (0.00 sec)

mysql> USE LibraryDB;
Database changed
```

- **Creating Table Books**



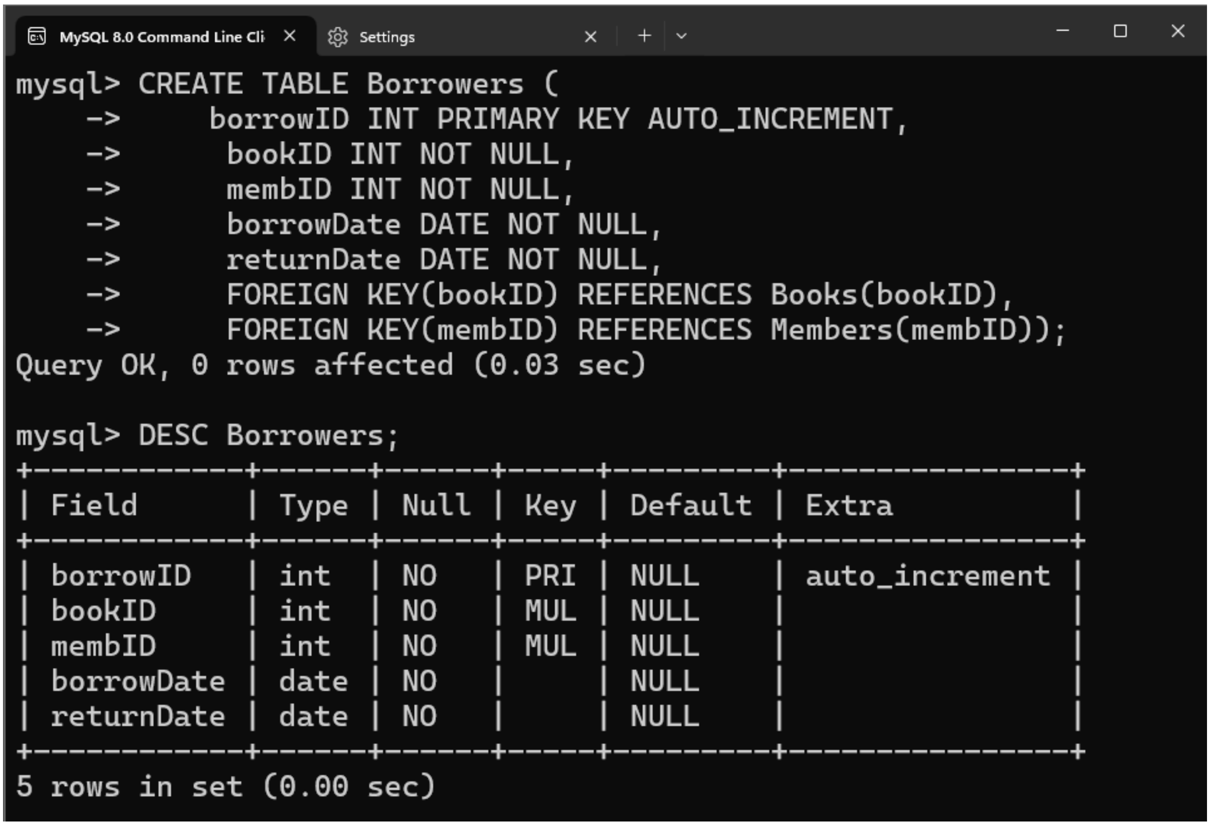
```
mysql> CREATE TABLE Books(
    -> bookID INT PRIMARY KEY,
    -> title VARCHAR(30) NOT NULL,
    -> author VARCHAR(30) NOT NULL,
    -> publisher VARCHAR(50),
    -> yearPublished YEAR,
    -> ISBN VARCHAR(13) UNIQUE );
Query OK, 0 rows affected (0.02 sec)

mysql> DESC Books;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| bookID        | int         | NO   | PRI | NULL    |       |
| title         | varchar(30) | NO   |     | NULL    |       |
| author        | varchar(30) | NO   |     | NULL    |       |
| publisher     | varchar(50) | YES  |     | NULL    |       |
| yearPublished | year        | YES  |     | NULL    |       |
| ISBN          | varchar(13) | YES  | UNI | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

- **Creating Table Members**



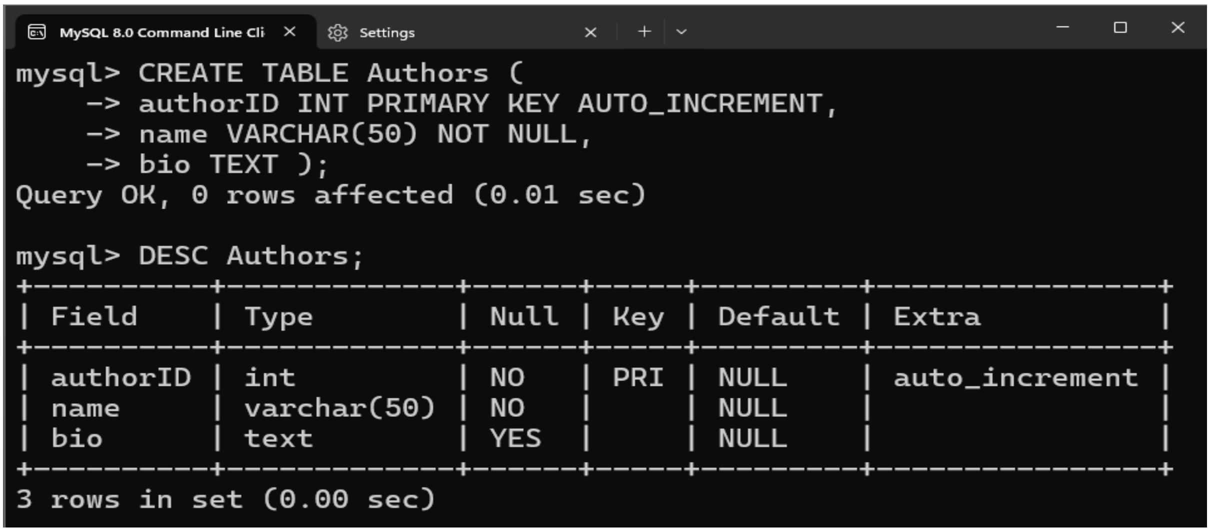
```
mysql> CREATE TABLE Members (
    -> membID INT PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(30) NOT NULL,
    -> address VARCHAR(70),
    -> mobile VARCHAR(12) UNIQUE,
    -> email VARCHAR(50) UNIQUE );
Query OK, 0 rows affected (0.02 sec)

mysql> DESC Members;
+---------+-------------+------+-----+---------+----------------+
| Field   | Type        | Null | Key | Default | Extra          |
+---------+-------------+------+-----+---------+----------------+
| membID  | int         | NO   | PRI | NULL    | auto_increment |
| name    | varchar(30) | NO   |     | NULL    |                |
| address | varchar(70) | YES  |     | NULL    |                |
| mobile  | varchar(12) | YES  | UNI | NULL    |                |
| email   | varchar(50) | YES  | UNI | NULL    |                |
+---------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

- **Creating Table Borrower:**



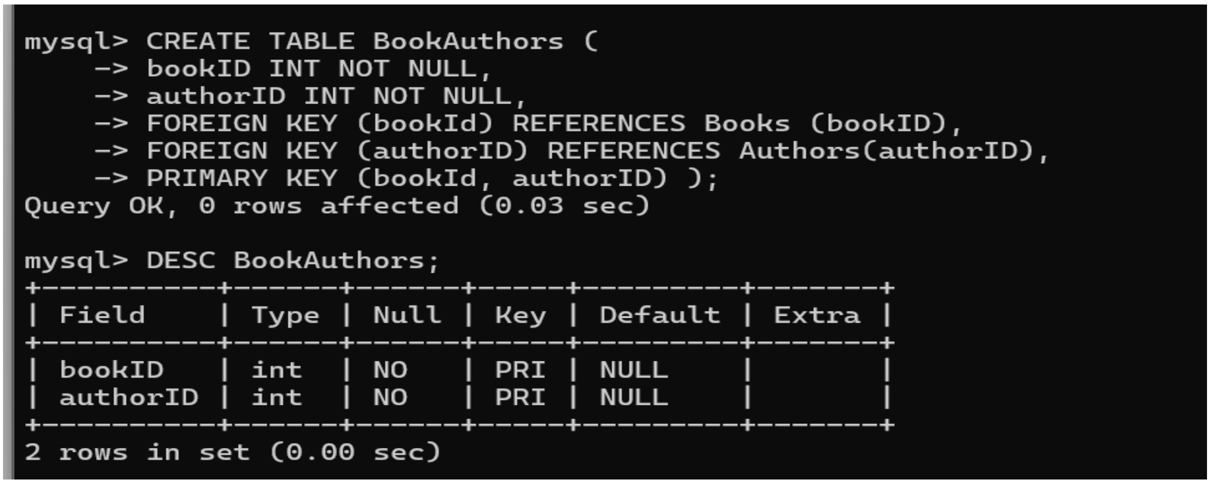
```
mysql> CREATE TABLE Borrowers (
    ->     borrowID INT PRIMARY KEY AUTO_INCREMENT,
    ->      bookID INT NOT NULL,
    ->      membID INT NOT NULL,
    ->      borrowDate DATE NOT NULL,
    ->      returnDate DATE NOT NULL,
    ->      FOREIGN KEY(bookID) REFERENCES Books(bookID),
    ->      FOREIGN KEY(membID) REFERENCES Members(membID));
Query OK, 0 rows affected (0.03 sec)

mysql> DESC Borrowers;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| borrowID   | int  | NO   | PRI | NULL    | auto_increment |
| bookID     | int  | NO   | MUL | NULL    |                |
| membID     | int  | NO   | MUL | NULL    |                |
| borrowDate | date | NO   |     | NULL    |                |
| returnDate | date | NO   |     | NULL    |                |
+------------+------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

- **Creating Table Authors:**



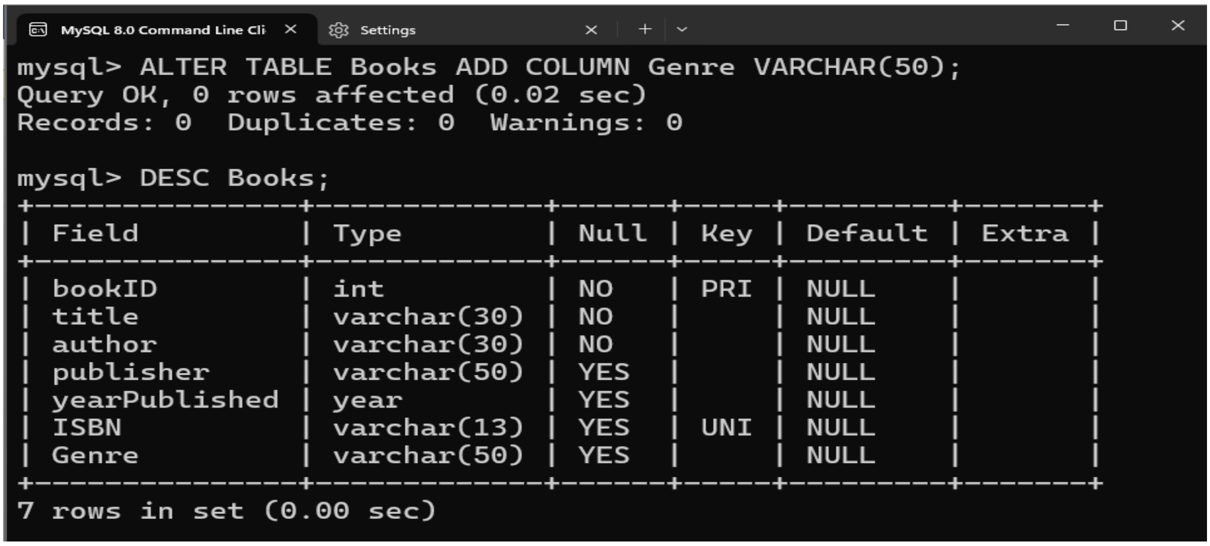
```
mysql> CREATE TABLE Authors (
    -> authorID INT PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(50) NOT NULL,
    -> bio TEXT );
Query OK, 0 rows affected (0.01 sec)

mysql> DESC Authors;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| authorID | int         | NO   | PRI | NULL    | auto_increment |
| name     | varchar(50) | NO   |     | NULL    |                |
| bio      | text        | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

- **Creating Table BookAuthors**



```
mysql> CREATE TABLE BookAuthors (
    -> bookID INT NOT NULL,
    -> authorID INT NOT NULL,
    -> FOREIGN KEY (bookId) REFERENCES Books (bookID),
    -> FOREIGN KEY (authorID) REFERENCES Authors(authorID),
    -> PRIMARY KEY (bookId, authorID) );
Query OK, 0 rows affected (0.03 sec)

mysql> DESC BookAuthors;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| bookID   | int  | NO   | PRI | NULL    |       |
| authorID | int  | NO   | PRI | NULL    |       |
+----------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

- **Alter Table**



```
mysql> ALTER TABLE Books ADD COLUMN Genre VARCHAR(50);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC Books;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| bookID        | int         | NO   | PRI | NULL    |       |
| title         | varchar(30) | NO   |     | NULL    |       |
| author        | varchar(30) | NO   |     | NULL    |       |
| publisher     | varchar(50) | YES  |     | NULL    |       |
| yearPublished | year        | YES  |     | NULL    |       |
| ISBN          | varchar(13) | YES  | UNI | NULL    |       |
| Genre         | varchar(50) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

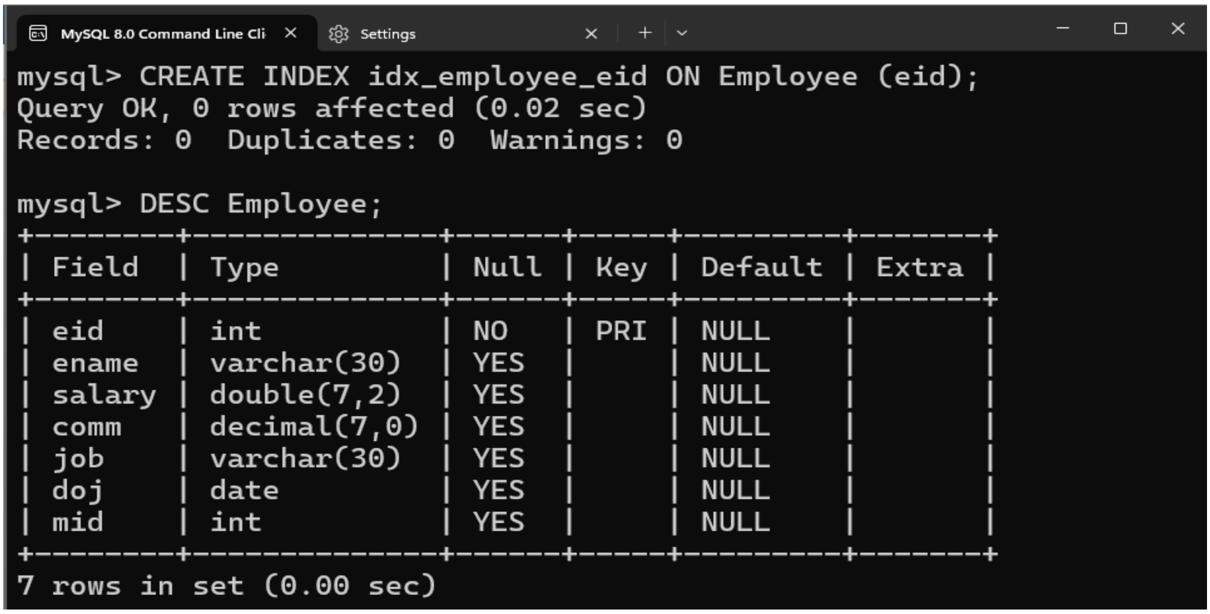
Assignment-5:

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 analyse the impact on query execution.

SOLUTION:

Creating and Dropping Index

- **Creating Index**



```
mysql> CREATE INDEX idx_employee_eid ON Employee (eid);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC Employee;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| eid    | int          | NO   | PRI | NULL    |       |
| ename  | varchar(30)  | YES  |     | NULL    |       |
| salary | double(7,2)  | YES  |     | NULL    |       |
| comm   | decimal(7,0) | YES  |     | NULL    |       |
| job    | varchar(30)  | YES  |     | NULL    |       |
| doj    | date         | YES  |     | NULL    |       |
| mid    | int          | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

--Query to search for employee by eid

SELECT * FROM Employee WHERE eid = 102;

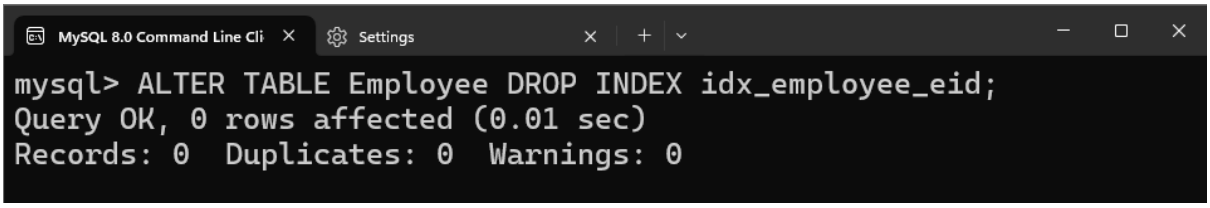
Query without Index:

Without an index, the database performs a full table scan to find rows where eid matches ‘eid'. This is inefficient, especially for large tables, because each row must be checked.

Query with Index:

With the index idx_employee_eid in place, the database uses the index to directly locate the rows where Title matches ‘edi'. This drastically reduces the number of rows that need to be examined and speeds up the query execution.

- **Drop Index:**



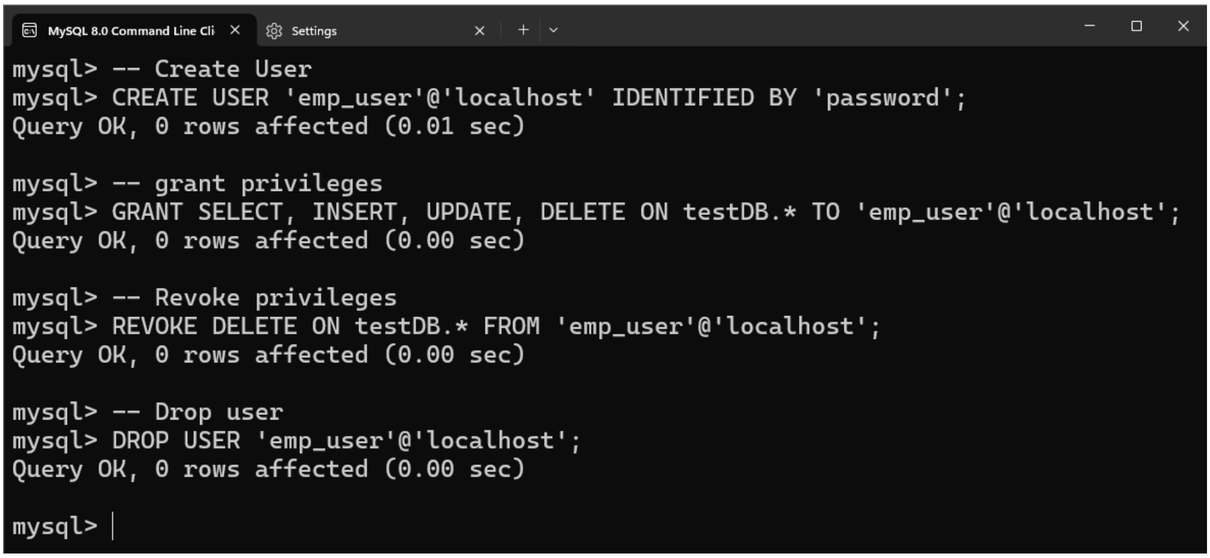
```
mysql> ALTER TABLE Employee DROP INDEX idx_employee_eid;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

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.

SOLUTION:

Creating and Managing Database Users



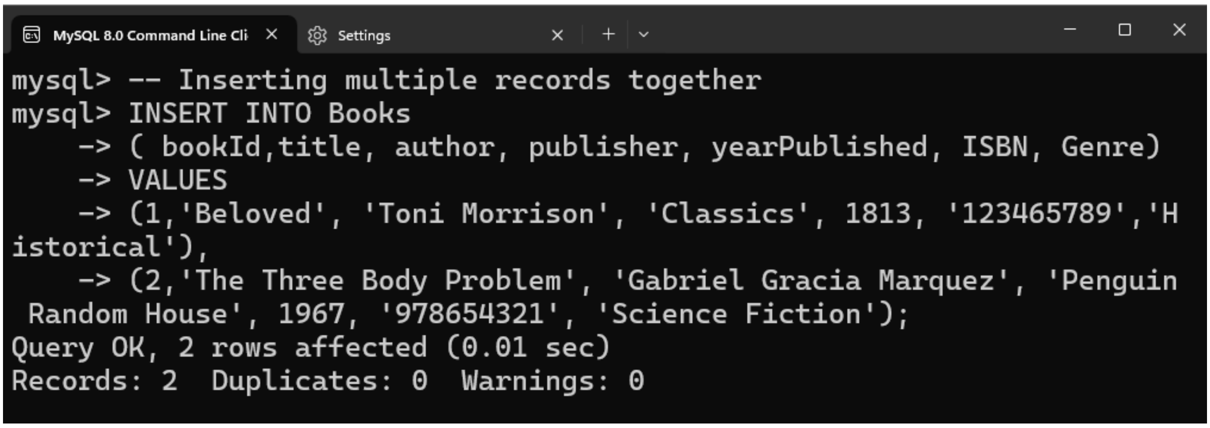
```
mysql> -- Create User
mysql> CREATE USER 'emp_user'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> -- grant privileges
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON testDB.* TO 'emp_user'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> -- Revoke privileges
mysql> REVOKE DELETE ON testDB.* FROM 'emp_user'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> -- Drop user
mysql> DROP USER 'emp_user'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

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.

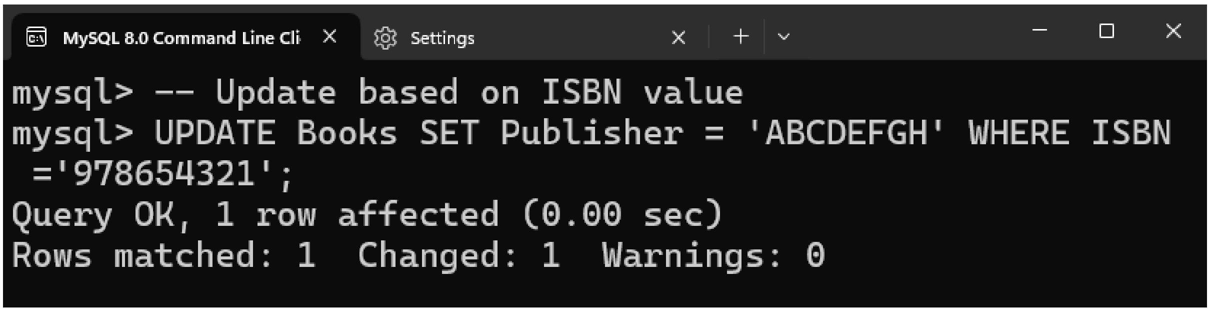
SOLUTION:

- **SQL Statements for Data Manipulation:**



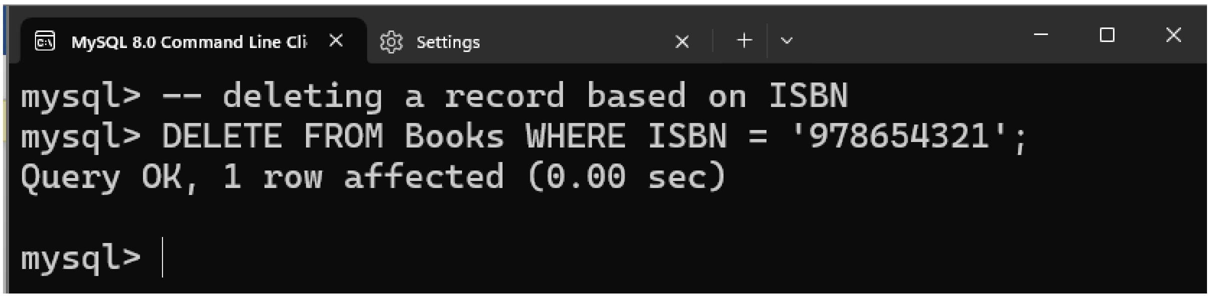
```
mysql> -- Inserting multiple records together
mysql> INSERT INTO Books
    -> ( bookId,title, author, publisher, yearPublished, ISBN, Genre)
    -> VALUES
    -> (1,'Beloved', 'Toni Morrison', 'Classics', 1813, '123465789','H
istorical'),
    -> (2,'The Three Body Problem', 'Gabriel Gracia Marquez', 'Penguin
 Random House', 1967, '978654321', 'Science Fiction');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

- **Update Existing Records**



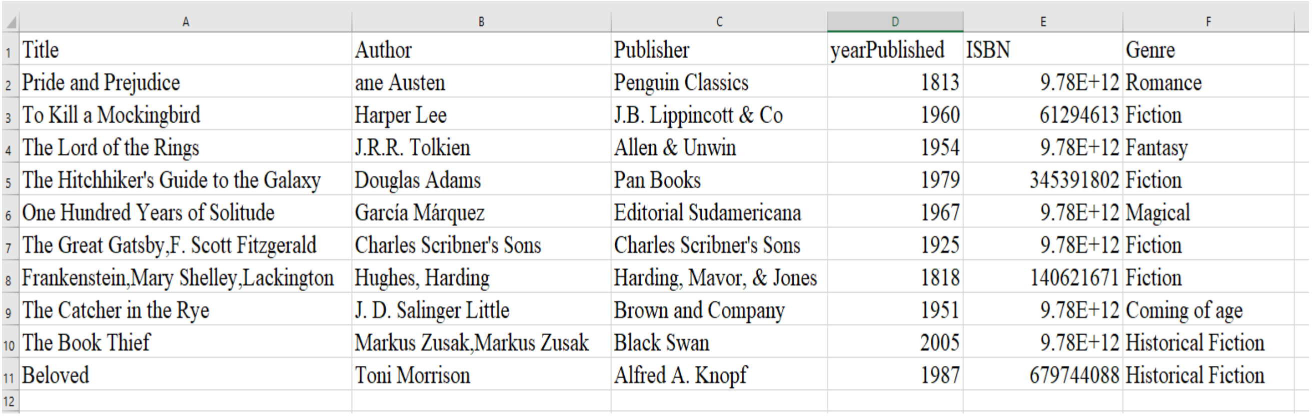
```
mysql> -- Update based on ISBN value
mysql> UPDATE Books SET Publisher = 'ABCDEFGH' WHERE ISBN
 ='978654321';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

- **Delete Records**



```
mysql> -- deleting a record based on ISBN
mysql> DELETE FROM Books WHERE ISBN = '978654321';
Query OK, 1 row affected (0.00 sec)

mysql>
```

■ Books.csv File



	A	B	C	D	E	F
1	Title	Author	Publisher	yearPublished	ISBN	Genre
2	Pride and Prejudice	ane Austen	Penguin Classics	1813	9.78E+12	Romance
3	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co	1960	61294613	Fiction
4	The Lord of the Rings	J.R.R. Tolkien	Allen & Unwin	1954	9.78E+12	Fantasy
5	The Hitchhiker's Guide to the Galaxy	Douglas Adams	Pan Books	1979	345391802	Fiction
6	One Hundred Years of Solitude	García Márquez	Editorial Sudamericana	1967	9.78E+12	Magical
7	The Great Gatsby,F. Scott Fitzgerald	Charles Scribner's Sons	Charles Scribner's Sons	1925	9.78E+12	Fiction
8	Frankenstein,Mary Shelley,Lackington	Hughes, Harding	Harding, Mavor, & Jones	1818	140621671	Fiction
9	The Catcher in the Rye	J. D. Salinger Little	Brown and Company	1951	9.78E+12	Coming of age
10	The Book Thief	Markus Zusak,Markus Zusak	Black Swan	2005	9.78E+12	Historical Fiction
11	Beloved	Toni Morrison	Alfred A. Knopf	1987	679744088	Historical Fiction
12						

■ Bulk Insert from a file

LOAD DATA LOCAL INFILE ‘C:\Users\yaseen\Desktop\Assignment\Book1.csv’ INTO TABLE Books Fields TERMINATED BY ‘,’ LINES TERMINATED BY ‘\n’ IGNORE 1 LINES;