

**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.**

**(Note: For Better Understanding View the “SS” from the File I had Created)**

**Ans:**

For this I have performed this assignment on Linux terminal here I have listed the Commands and queries I used:

☐ **Step 1: Access MySQL**

- Open the terminal and Connect to the MySQL server with credentials:
  - Sudo -i
  - mysql
  - mysql --version

☐ **Step 2: Create a New Database**

- Create the new database:
  - CREATE DATABASE Library\_DB\_Asg;

☐ **Step 3: Use the New Database**

- Switch to the new database:
  - USE Library\_DB\_Asg;

☐ **Step 4: Create Tables**

- Here is a basic library schema, I have created tables for Books, Authors, and Borrowers.

→ Table: Authors ::

```
CREATE TABLE Author (  
    AuthorID INT AUTO_INCREMENT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    BirthDate DATE );
```

→ Table: Books ::

```
CREATE TABLE Books (  
    BookID INT AUTO_INCREMENT PRIMARY KEY,  
    Title VARCHAR(100),  
    AuthorID INT,  
    Publish_Year INT,  
    Genre VARCHAR(50),  
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID) );
```

→ Table: Borrowers

```
CREATE TABLE Borrowers (  
    BorrowerID INT AUTO_INCREMENT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Membership_Date DATE  
);
```

→ Table: Checkout ::

```
CREATE TABLE Checkout (  
    CheckoutID INT AUTO_INCREMENT PRIMARY KEY,  
    BookID INT,  
    BorrowerID INT,  
    Checkout_Date DATE,  
    ReturnDate DATE,
```

```
FOREIGN KEY (BookID) REFERENCES Books(BookID),  
  
FOREIGN KEY (BorrowerID) REFERENCES  
Borrowers(BorrowerID)  
  
);
```

→ Table: Fines ::

```
CREATE TABLE Fines (  
  
    FineID INT AUTO_INCREMENT PRIMARY KEY,  
  
    BorrowerID INT,  
  
    Amount DECIMAL(5, 2),  
  
    Paid BOOLEAN,  
  
    FOREIGN KEY (BorrowerID) REFERENCES Borrowers(BorrowerID)  
  
);
```

☐ **Step 5: Alter Tables (Add, Update and Delete) :**

- **Add a Column to the Authors Table**

- Add a column Nationality to the Authors table:
- Code:

```
ALTER TABLE Authors  
ADD COLUMN Nationality VARCHAR(50);
```

- **Update Column Properties in the Books Table**

- Update the Genre column to be NOT NULL:
- Code:

```
ALTER TABLE Books  
MODIFY COLUMN Genre VARCHAR(50) NOT NULL;
```

- **Drop a Column from the Author Table**
  - Drop the Nationality column from the Author table:
  - Code:

```
ALTER TABLE Author
DROP COLUMN Nationality;
```

**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.**

**(Note: For Better Understanding View the “SS” from the File I had Created)**

**Ans:**

Here I'm Using the Same Schema and Problem Statement as above i.e. Library Management:

For this Problem Solution Follow The Same Steps from Above Problem i.e. Assignment 4 up to step number 4.

☐ **Step 1:** Access MySQL through Terminal: (Same as Assignment 4 for this step 1).

☐ **Step 2:** Create a New Database (Same as Assignment 4 for this step 2).

☐ **Step 3:** Use the New Database (Same as Assignment 4 for this step 3).

☐ **Step 4:** Create Tables for the Database (Same as Assignment 4 for this step 4).

☐ **Step 5:** Insert the Data into the Created Table.

(Here I'm taking specifically Books Table to perform operations)

```
INSERT INTO Books (Title, AuthorID, PublishedYear, Genre) VALUES
    ('Book A', 1, 2020, 'Fiction'),
    ('Book B', 2, 2019, 'Non-Fiction'),
    ('Book C', 3, 2021, 'Fiction'),
    ('Book D', 1, 2018, 'Science'),
    ('Book E', 2, 2017, 'History'),
    ('Book F', 3, 2022, 'Fiction');
```

☐ **Step 6:** Create an Index:

```
CREATE INDEX title_idx ON Books (Title);
```

☐ **Step 7:** Analyse the Impact of the Index:

- Before Creating the Index:

```
EXPLAIN SELECT * FROM Books WHERE Title = 'Book A';
```

- **After Creating the Index:**

```
EXPLAIN SELECT * FROM Books WHERE Title = 'Book A';
```

- ☐ **Step 8:** Drop the Index:

```
DROP INDEX idx_title ON Books;
```

- ☐ **Step 9:** Impact of Dropping the Index:

```
EXPLAIN SELECT * FROM Books WHERE Title = 'Book A';
```

- ☐ **Why do we need an Index and what have we achieved?**

- Indexes improve query performance by allowing the database to locate rows with specific column values quickly.
- When an index is present, the database can use the index to find the rows efficiently rather than scanning the entire table.
- Before Index: The query might perform a full table scan, resulting in higher rows and potentially higher type values (e.g., ALL).
- After Index: The query should use the index, resulting in lower rows and a more efficient type value (e.g., ref or range).
- After Dropping Index: The query performance should revert to the initial state, with the database performing a full table scan again.

**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.**

**(Note: For Better Understanding View the “SS” from the File I had Created)**

**Ans:**

For this assignment, I have listed out commands to achieve this follow the steps:

### **Step 1: Access MySQL:**

Commands:

```
sudo -i  
mysql -u root -p
```

### **Step 2: Create a New Database User:**

Commands:

```
-- Create a new user  
CREATE USER '<new_user>'@'<localhost>' IDENTIFIED BY  
'<password>';  
  
-- Grant specific privileges to the new user  
GRANT SELECT, INSERT, UPDATE, DELETE ON Library_DB_Asg.* TO  
'<new_user>'@'<localhost>';  
  
-- Apply the changes  
FLUSH PRIVILEGES;
```

### Step 3: Revoke Certain Privileges:

Commands:

```
-- Revoke DELETE privilege from the new user
REVOKE DELETE ON Library_DB_Asg.* FROM 'new_user'@'localhost';

-- Apply the changes
FLUSH PRIVILEGES;
```

### Step 4: Drop the User:

Commands:

```
-- Drop the user
DROP USER 'new_user'@'localhost';
```

Explanation of the Commands

#### 1. Create a New User:

- `CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'password';` This command creates a new user with the specified username and password.
- `GRANT SELECT, INSERT, UPDATE, DELETE ON Library_DB_Asg.* TO 'new_user'@'localhost';` This command grants the user specific privileges on the Library\_DB\_Asg database.

#### 2. Apply Privilege Changes:

- `FLUSH PRIVILEGES;` This command reloads the privilege tables in MySQL, ensuring that the changes take effect immediately.



### 3. Revoke Specific Privileges:

- REVOKE DELETE ON Library\_DB\_Asg.\* FROM 'new\_user'@'localhost'; This command revokes the DELETE privilege from the user on the specified database.

### 4. Drop the User:

- DROP USER 'new\_user'@'localhost'; This command removes the user from the MySQL server.

**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.**

**(Note: For Better Understanding View the “SS” from the File I had Created)**

Ans:

Here I'm Using the Same Schema and Problem Statement as above i.e. Library Management:

For this Problem Solution Follow The Same Steps from Above Problem i.e. Assignment 4 up to step number 4.

- ☐ **Step 1:** Access MySQL through Terminal: (Same as Assignment 4 for this step 1).
- ☐ **Step 2:** Create a New Database (Same as Assignment 4 for this step 2).
- ☐ **Step 3:** Use the New Database (Same as Assignment 4 for this step 3).
- ☐ **Step 4:** Create Tables for the Database (Same as Assignment 4 for this step 4).
- ☐ **Step 5:** Insert the Data into the Created Table.

Table Author:

```
INSERT INTO Authors (FirstName, LastName, BirthDate) VALUES
('ABC', 'CBA', '1928-09-09'),
('PQR', 'RQP', '1841-18-11'),
('XYZ', 'ZYX', '1934-04-28');
```

Table Books:

```
INSERT INTO Books (Title, AuthorID, Publish_Year, Genre) VALUES
('War and Peace', 10, 1869, 'Historical Fiction'),
('Gods Of Space', 11, 1866, 'Psychological Fiction'),
('Journey To the West', 12, 1960, 'Southern Gothic');
```

Table Borrowers:

```
INSERT INTO Borrowers (FirstName, LastName, Membership_Date,  
Email) VALUES
```

```
('Tommy', 'Rides', '2024-01-15', 'TommyRides@example.com'),
```

```
('Jerry', 'Smith', '2023-02-20', 'JerrySmith@example.com');
```

☐ Step 6: Update Existing Records

- Update an Author's BirthDate
- Commands:

```
UPDATE Authors
```

```
SET BirthDate = '1931-07-11'
```

```
WHERE FirstName = ' XYZ ' AND LastName = ' ZYX ';
```

- Update a Book's Publish\_Year
- Commands:

```
UPDATE Books
```

```
SET PublishedYear = 1981
```

```
WHERE Title = ' Journey To the West ';
```

- Update a Borrower's Email
- Commands:

```
UPDATE Borrowers
```

```
SET Email = 'JerrySmith_updated@example.com'
```

```
WHERE FirstName = 'Jerry' AND LastName = 'Smith';
```

☐ Step 7: Delete Records Based on Specific Criteria:

- Delete a Book by Title:
- Commands:

```
DELETE FROM Books
```

WHERE Title = 'War and Peace';

- Delete an Author by LastName
- Commands:

DELETE FROM Authors

WHERE LastName = ' ZYX ';

☐ Step 8: Bulk Insert Operations

- Create a Sample CSV File
- Content:

Title,AuthorID,Publish\_Year,Genre

'Pride and Prejudice',4,1813,'Romantic Fiction'

'1984',5,1949,'Dystopian'

'The Great Gatsby',6,1925,'Tragedy'

- Load Data from CSV File:
- Commands:

LOAD DATA INFILE '/path/to/books.csv'

INTO TABLE Books

FIELDS TERMINATED BY ','

ENCLOSED BY ''

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(Title, AuthorID, PublishedYear, Genre);