```
-- ============================================================================== -
- Lab Experiment 02 – Implementation of DML Commands in SQL --
============================================================================== --
Objective: -- To implement DML (Data Manipulation Language) commands such as -- INSERT,
UPDATE, and DELETE using suitable examples. -- Scenario: -- You are managing a database
for a library with two tables: Books and Members. --
============================================================================== --
Step 1: Create Database DROP DATABASE IF EXISTS LibraryDB; -- output -- 22:49:18 DROP
DATABASE IF EXISTS LibraryDB 0 row(s) affected 0.016 sec CREATE DATABASE LibraryDB; -
- output -- 22:49:49 CREATE DATABASE LibraryDB 1 row(s) affected 0.000 sec USE LibraryDB;
-- output -- 22:50:43 USE LibraryDB 0 row(s) affected 0.000 sec --
============================================================================== --
Task 01: Create Tables -- Create Books and Members tables as per given structure. CREATE
TABLE Books ( book_id INT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(100), author
VARCHAR(50), published_year YEAR, available_copies INT, total_copies INT, genre
VARCHAR(50), ISBN VARCHAR(20) ); -- output -- 22:51:25 CREATE TABLE Books ( book_id
INT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(100), author VARCHAR(50),
published_year YEAR, available_copies INT, total_copies INT, genre VARCHAR(50), ISBN
VARCHAR(20) ) 0 row(s) affected 0.047 sec CREATE TABLE Members ( member_id INT
PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50),
membership_date DATE, membership_type VARCHAR(20), email VARCHAR(100),
phone_number VARCHAR(20), address VARCHAR(100) ); -- output -- 22:51:58 CREATE
TABLE Members ( member_id INT PRIMARY KEY AUTO_INCREMENT, first_name
VARCHAR(50), last_name VARCHAR(50), membership_date DATE, membership_type
VARCHAR(20), email VARCHAR(100), phone_number VARCHAR(20), address
VARCHAR(100) ) 0 row(s) affected 0.031 sec -- write the query here to verify the creation --
WRITE YOUR OUTPUT BELOW --
============================================================================== --
Task 02: Integrity Constraints -- Ensure that available_copies is always less than or equal to
total_copies. -- WRITE YOUR QUERY BELOW -- OUTPUT: -- (Display confirmation message or
describe constraint added.) -- WRITE YOUR OUTPUT BELOW --
============================================================================== --
Task 03: Insert a New Book -- Scenario: Insert a record for the new book titled -- '1984_The
Black Swan' by George Orwell (published in 1949, 4 copies available). -- WRITE YOUR QUERY
BELOW -- OUTPUT: -- (Run SELECT * FROM Books; to verify insertion) -- WRITE YOUR
OUTPUT BELOW --
============================================================================== --
Task 04: Add New Members -- Scenario: -- Member 1: David Lee, Membership Date – 2024-04-
15, Membership Type – Platinum -- Member 2: Emma Wilson, Membership Date – 2024-05-22,
Membership Type – Silver -- WRITE YOUR QUERIES BELOW -- OUTPUT: -- (Run SELECT *
FROM Members; to verify insertion) -- WRITE YOUR OUTPUT BELOW --
============================================================================== --
Task 05: Update Book Details -- Scenario: The library has acquired 2 additional copies of the
book '1984'. -- Update the record accordingly. -- WRITE YOUR QUERY BELOW -- OUTPUT: --
(Run SELECT * FROM Books; to check updated copies) -- WRITE YOUR OUTPUT BELOW --
============================================================================== --
Task 06: Modify a Member's Information -- Scenario: Emma Wilson has upgraded her
membership from 'Silver' to 'Gold'. -- WRITE YOUR QUERY BELOW -- OUTPUT: -- (Run
SELECT * FROM Members; to check updated membership) -- WRITE YOUR OUTPUT BELOW
-- ============================================================================== -
- Task 07: Delete a Book Record -- Scenario: The book '1984_The Black Swan' is no longer
available in the library. -- Remove it from the database. -- WRITE YOUR QUERY BELOW --
OUTPUT: -- (Run SELECT * FROM Books; to verify deletion) -- WRITE YOUR OUTPUT
BELOW --
============================================================================== --
Task 08: Remove a Member -- Scenario: David Lee has canceled his membership. Remove his
```

record. -- WRITE YOUR QUERY BELOW -- OUTPUT: -- (Run SELECT * FROM Members; to verify deletion) -- WRITE YOUR OUTPUT BELOW -- ======================================================================== -- Extension (Optional) -- Task: Create a Borrowing table to track which members have borrowed which books. -- Include foreign keys referencing Books(book_id) and Members(member_id). -- WRITE YOUR TABLE CREATION QUERY BELOW -- OUTPUT: -- (Run DESC Borrowing; to verify structure) -- WRITE YOUR OUTPUT BELOW -- ======================================================================== -- END OF EXPERIMENT 02 -- ========================================================================