|  |
| --- |
| **• Lab 3: Create a database called library\_db and a table books with columns: book\_id, title, author, publisher, year\_of\_publication, and price. Insert five records into the table.**  Ans:  CREATE DATABASE libray\_db;  CREATE TABLE book(  b\_id int,  tital text,  author varchar(20),  publisher varchar(20),  year\_pub INT,  price int  );  INSERT INTO book VALUES(101,'theanimal','kishan','meru',2025,2500),(102,'hassinaruba','k.k','m.k',2023,2300),(103,'thekingdom','rana','mahendra',2021,5000),(104,'meluha','d.sastri','m.shastri',1999,3000),(105,'mafia','r.s','g.s',2020,3500); |
| **• Lab 4: Create a table members in library\_db with columns: member\_id, member\_name, date\_of\_membership, and email. Insert five records into this table.**  Ans:  CREATE TABLE members(  m\_id INT,  m\_name text,  d\_mship date,  email text  );  INSERT INTO members VALUES(1,'suresh',’2000/04/12’,'suresh@gmail.com'),(2,'hardik',’2000/4/12’,'hardik@gmail.com'),(3,'dev',’2024/7/22’,'dev@gmail.com'),(4,'sauvrav',’2020/4/12’,'saurav@gmail.com'),(5,'gaurav',’2016/05/2024’,'gaurav@gmail.com'); |
| **• Lab 3: Retrieve all members who joined the library before 2022. Use appropriate SQL syntax with WHERE and ORDER BY.**  Ans:  [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM book WHERE year\_pub<='2022/01/01' ORDER BY year\_pub; |
| **• Lab 4: Write SQL queries to display the titles of books published by a specific author. Sort the results by year\_of\_publication in descending order.**  Ans:  SELECT \* FROM book WHERE year\_pub<='2022/01/01' ORDER BY year\_pub DESC; |
| **• Lab 3: Add a CHECK constraint to ensure that the price of books in the books table is greater than 0.**  Ans:  [CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) book1(  b\_id int,    tital text,  author varchar(20),  publisher varchar(20),  year\_pub INT,  price int CHECK(price>0) ); |
| **• Lab 4: Modify the members table to add a UNIQUE constraint on the email column, ensuring that each member has a unique email address.**  Ans:  ALTER TABLE student MODIFY COLUMN email text UNIQUE; |
| **• Lab 3: Create a table authors with the following columns: author\_id, first\_name, last\_name, and country. Set author\_id as the primary key.**  Ans:  CREATE TABLE authors(  a\_id int PRIMARY KEY,  a\_firstname text,  a\_lastname text,  a\_country text,  ); |
| **• Lab 4: Create a table publishers with columns: publisher\_id, publisher\_name, contact\_number, and address. Set publisher\_id as the primary key and contact\_number as unique.**  Ans:  CREATE TABLE publishers(  p\_id int PRIMARY KEY,  p\_name text,  contact\_no int UNIQUE,  address text  ); |

|  |
| --- |
| • Lab 3: Add a new column genre to the books table. Update the genre for all existing records.  Ans:  ALTER TABLE book ADD genre text;  [UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) book [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) genre=' wild discovry ' WHERE b\_id=101;  [UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) book [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) genre='thiller' WHERE b\_id=102;  UPDATE book SET genre='historical' WHERE b\_id=103; |
| UPDATE book SET genre=' fantasy fiction' WHERE b\_id=104;  UPDATE book SET genre='crime' WHERE b\_id=105; |
| • Lab 4: Modify the members table to increase the length of the email column to 100 characters.  Ans:  ALTER TABLE members MODIFY email varchar(100); |
| • Lab 3: Drop the publishers table from the database after verifying its structure.  DROP TABLE publishers; |
| • Lab 4: Create a backup of the members table and then drop the original members table.  Ans:  DROP TABLE members; |
| • Lab 4: Insert three new authors into the authors table, then update the last name of one of the authors.  Ans:  INSERT INTO authors VALUES(1,'suresh','rameshbhai','mumbai'),(2,'akash','jigarhbhai','mumbai'),(3,'gaurav','nitinbhai','delhi');    UPDATE authors SET a\_lastname='dineshbhai' WHERE a\_id=1;  UPDATE authors SET a\_lastname='hiteshbhai' WHERE a\_id=2;  UPDATE authors SET a\_lastname='sanjaybhai' WHERE a\_id=3; |
| • Lab 5: Delete a book from the books table where the price is higher than $100.  Ans:  DELETE FROM book WHERE price>=3000; |
| • Lab 3: Update the year\_of\_publication of a book with a specific book\_id.  Ans:    UPDATE book SET year\_pub=2018 WHERE b\_id=101;  UPDATE book SET year\_pub=2020 WHERE b\_id=102; |

|  |
| --- |
| • Lab 4: Increase the price of all books published before 2015 by 10%.  Ans:    UPDATE book SET price=500 WHERE b\_id=103;  UPDATE book SET price=300 WHERE b\_id=104; |
| • Lab 3: Remove all members who joined before 2020 from the members table.  Ans:  [DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) FROM member WHERE m\_ship<'2020/01/01'; |
| • Lab 4: Delete all books that have a NULL value in the author column.  Ans:    [DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) FROM book WHERE author [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23operator_is) null; |
| • Lab 4: Write a query to retrieve all books with price between $50 and $100  Ans:  SELECT \* FROM book WHERE price BETWEEN 2500 AND 3500; |
| • Lab 5: Retrieve the list of books sorted by author in ascending order and limit the results to the top 3 entries.  Ans:  SELECT \* FROM book ORDER BY author LIMIT 3; |
| • Lab 3: Perform an INNER JOIN between books and authors tables to display the title of books and their respective authors' names.  Ans:  SELECT author.a\_nane,book.b\_name,book.b\_id FROM book INNER JOIN author ON book.b\_id=author.b\_id; |
| • Lab 4: Use a FULL OUTER JOIN to retrieve all records from the books and authors tables, including those with no matching entries in the other table.  Ans:  SELECT author.a\_nane,book.b\_name,book.b\_id FROM book full JOIN author ON book.b\_id=author.b\_id; |

|  |
| --- |
| • Lab 3: Group books by genre and display the total number of books in each genre.  Ans:  SELECT gener,count(title) FROM book1 GROUP BY gener; |
| • Lab 4: Group members by the y  ear they joined and find the number of members who joined each year.  Ans:  SELECT year, COUNT(name) FROM member GROUP BY year; |
| • Lab 3: Write a stored procedure to retrieve all books by a particular author.  Ans:  DELIMITER $$  CREATE PROCEDURE ins( b\_id int, b\_name text, author text, price int )  BEGIN  INSERT INTO book VALUES(b\_id,b\_name,author,price);  END  CALL ins(101,'the lion','m.joshi',6000);  CALL ins(102,'the mom','m.gandhi',5000); |
| • Lab 4: Write a stored procedure that takes book\_id as an argument and returns the price of the book.  Ans:  CREATE PROCEDURE GetBookPrice(IN book\_id INT, OUT book\_price DECIMAL(10, 2))  BEGIN  SELECT price INTO book\_price  FROM books  WHERE b\_id = book\_id;  END;  CALL GetBookPrice(1, @price);  SELECT @price; |
| • Lab 3: Create a view to show only the title, author, and price of books from the books table.  Ans:  CREATE VIEW v1 AS SELECT title,author,price FROM book1;  SELECT \* FROM v1; |
| • Lab 4: Create a view to display members who joined before 2020  Ans:  CREATE VIEW v2 AS SELECT \* FROM member WHERE year<2020;  SELECT \* FROM v2; |

|  |
| --- |
| • Lab 3: Create a trigger to automatically update the last\_modified timestamp of the books table whenever a record is updated.  Ans:    [CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-trigger.html) [TRIGGER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-trigger.html) tri2 AFTER [UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) ON book2 FOR EACH ROW  BEGIN  [INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO alldata(id,name,proccess)[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html%23function_values)(new.b\_id,new.b\_name,'update record');  END  UPDATE book2 SET b\_name='the dad' WHERE b\_id=1; |
| • Lab 4: Create a trigger that inserts a log entry into a log\_changes table whenever a DELETE operation is performed on the books table.  Ans:      DELIMITER $$  CREATE TRIGGER tri3 AFTER DELETE ON book2 FOR EACH ROW  BEGIN  INSERT INTO alldata(id,name,proccess)VALUES(old.b\_id,old.b\_name,'delete record');  END  [DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) FROM book2 WHERE b\_id=2; |