

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

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 library_db;
```

```
CREATE TABLE books (  
    book_id int ,  
    title text ,  
    AUTHOR text ,  
    publisher varchar(60),  
    YEAR_of_publication int,  
    price INT  
);
```

```
INSERT INTO books VALUES(101, 'To Kill a Mockingbird', 'Harper Lee', 'J.B. Lippincott & Co.', 1960, 1800),  
(102, '1984', 'George Orwell', 'Secker & Warburg', 1949, 1599),  
(103, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Charles Scribner\'s Sons', 1925, 1099),  
(104, 'Pride and Prejudice', 'Jane Austen', 'T.Egerton', 1813, 1299),  
(105, 'Moby-Dick', 'Herman Melville', 'Harper & Brothers', 1851, 1799);
```

book_id	title	AUTHOR	publisher	YEAR_of_publication	price
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	1800
102	1984	George Orwell	Secker & Warburg	1949	1599
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1925	1099
104	Pride and Prejudice	Jane Austen	T.Egerton	1813	1299
105	Moby-Dick	Herman Melville	Harper & Brothers	1851	1799

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 (  
    member_id int ,  
    member_name varchar(30),  
    date_of_membership date,  
    email text  
);
```

```
INSERT INTO members VALUES  
(1, 'John Doe', '2022-05-10', 'john.doe@example.com'),  
(2, 'Jane Smith', '2023-01-15', 'jane.smith@example.com'),  
(3, 'Alice Johnson', '2021-11-25', 'alice.johnson@example.com'),  
(4, 'Bob Brown', '2023-03-30', 'bob.brown@example.com'),  
(5, 'Charlie Davis', '2022-07-20', 'charlie.davis@example.com');
```

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

member_id	member_name	date_of_membership	email
1	John Doe	2022-05-10	john.doe@example.com
2	Jane Smith	2023-01-15	jane.smith@example.com
3	Alice Johnson	2021-11-25	alice.johnson@example.com
4	Bob Brown	2023-03-30	bob.brown@example.com
5	Charlie Davis	2022-07-20	charlie.davis@example.com

Lab 3: Retrieve all members who joined the library before 2022. Use appropriate SQL syntax with WHERE and ORDER BY.

ANS :-

```
SELECT member_name , date_of_membership FROM members WHERE date_of_membership <='2022-01-01' ORDER BY date_of_membership;
```

member_name	date_of_membership
Alice Johnson	2021-11-25

```
SELECT * FROM members ORDER BY member_name;
```

member_id	member_name	1	date_of_membership	email
3	Alice Johnson		2021-11-25	alice.johnson@example.com
4	Bob Brown		2023-03-30	bob.brown@example.com
5	Charlie Davis		2022-07-20	charlie.davis@example.com
2	Jane Smith		2023-01-15	jane.smith@example.com
1	John Doe		2022-05-10	john.doe@example.com

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 AUTHOR, title FROM books;
```

AUTHOR	title
Harper Lee	To Kill a Mockingbird
George Orwell	1984
F. Scott Fitzgerald	The Great Gatsby
Jane Austen	Pride and Prejudice
Herman Melville	Moby-Dick

Lab 3: Add a CHECK constraint to ensure that the price of books in the books table is greater than 0.

ANS :-

```
CREATE TABLE books(  
    book_id int PRIMARY KEY ,
```

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

title text ,
AUTHOR text ,
publisher varchar(50),
YEAR_of_publication int ,
price int CHECK(price>0));

book_id	title	AUTHOR	publisher	YEAR_of_publication	price
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	1800
102	1984	George Orwell	Secker & Warburg	1949	1599
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1925	1099
104	Pride and Prejudice	Jane Austen	T.Egerton	1813	1299
105	Moby-Dick	Herman Melville	Harper & Brothers	1851	1799

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 members MODIFY COLUMN email text UNIQUE;

date_of_membership	email
2022-05-10	john.doe@example.com
2023-01-15	jane.smith@example.com
2021-11-25	alice.johnson@example.com
2023-03-30	bob.brown@example.com
2022-07-20	charlie.davis@example.com

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 AUTHOR(  
  AUTHOR_id int ,  
  FIRST_name text ,  
  LAST_name varchar(50),  
  country text,  
  PRIMARY KEY(AUTHOR_id));
```

AUTHOR_id	FIRST_name	LAST_name	country
-----------	------------	-----------	---------

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(  
  publisher_id int ,  
  publisher_name text ,  
  contact_number int ,  
  address text,  
  PRIMARY KEY(publisher_id),  
  UNIQUE(contact_number));
```

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

```
publisher_id int PRIMARY KEY ,  
publisher_name varchar(50) ,  
concat_number int UNIQUE ,  
address text  
);
```

publisher_id	publisher_name	concat_number	address
--------------	----------------	---------------	---------

Lab 3: Add a new column genre to the books table. Update the genre for all existing records.

ANS :-

```
ALTER TABLE books ADD COLUMN genre text;  
UPDATE books SET genre='Modern Indian Social Drama' WHERE book_id=101;  
UPDATE books SET genre='Political Fiction, Dystopian' WHERE book_id=102;  
UPDATE books SET genre='Modernist Indian Fiction' WHERE book_id=103;  
UPDATE books SET genre='Indian Romantic Literature' WHERE book_id=104;  
UPDATE books SET genre='Adventure, Mythological' WHERE book_id=105;
```

book_id	title	AUTHOR	publisher	YEAR_of_publication	price	genre
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	1800	Modern Indian Social Drama
102	1984	George Orwell	Secker & Warburg	1949	1599	Political Fiction, Dystopian
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1925	1099	Modernist Indian Fiction
104	Pride and Prejudice	Jane Austen	T.Egerton	1813	1299	Indian Romantic Literature
105	Moby-Dick	Herman Melville	Harper & Brothers	1851	1799	Adventure, Mythological

Lab 4: Modify the members table to increase the length of the email column to 100 characters.

ANS :-

```
ALTER TABLE members MODIFY COLUMN email varchar(100) ;
```

Lab 3: Drop the publishers table from the database after verifying its structure.

ANS :-

```
DROP TABLE publishers;
```

Lab 4: Create a backup of the members table and then drop the original members table.

ANS :-

```
CREATE TABLE members_backup SELECT * FROM members;
```

```
DROP TABLE members;
```

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

Lab 4: Insert three new authors into the authors table, then update the last name of one of the authors.

ANS :-

```
INSERT INTO author VALUES
(11,'narsinh','mehta','talaja'),
(12,'premannad','bhatt','vadodra'),
(13,'javerchand','meghani','botad');
```

AUTHOR_id	FIRST_name	LAST_name	country
11	narsinh	mehta	talaja
12	premannad	bhatt	vadodra
13	javerchand	meghani	botad

```
UPDATE author SET LAST_name='jayant meghani' WHERE AUTHOR_id=13;
```

AUTHOR_id	FIRST_name	LAST_name	country
11	narsinh	mehta	talaja
12	premannad	bhatt	vadodra
13	javerchand	jayant meghani	botad

Lab 5: Delete a book from the books table where the price is higher than \$100.

ANS :-

```
DELETE FROM books WHERE price>8500;
```

book_id	title	AUTHOR	publisher	YEAR_of_publication	price	genre
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	5000	Modern Indian Social Drama
102	1984	George Orwell	Secker & Warburg	1949	8000	Political Fiction, Dystopian
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1925	4000	Modernist Indian Fiction
105	Moby-Dick	Herman Melville	Harper & Brothers	1851	5400	Adventure, Mythological

Lab 3: Update the year_of_publication of a book with a specific book_id.

ANS :-

```
UPDATE books SET YEAR_of_publication=1960 WHERE book_id=101;
UPDATE books SET YEAR_of_publication=1956 WHERE book_id=102;
UPDATE books SET YEAR_of_publication=1872 WHERE book_id=103;
UPDATE books SET YEAR_of_publication=2001 WHERE book_id=105;
```

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

book_id	title	AUTHOR	publisher	YEAR_of_publication	price	genre
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	5000	Modern Indian Social Drama
102	1984	George Orwell	Secker & Warburg	1956	8000	Political Fiction, Dystopian
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1872	4000	Modernist Indian Fiction
105	Moby-Dick	Herman Melville	Harper & Brothers	2001	5400	Adventure, Mythological

Lab 4: Increase the price of all books published before 2015 by 10%.

ANS :-

UPDATE books SET price=price*0.10+price WHERE YEAR_of_publication<2015;

UPDATE books SET price=price*0.10+price WHERE YEAR_of_publication<2015;

UPDATE books SET price=price*0.10+price WHERE YEAR_of_publication<2015;

UPDATE books SET price=price*0.10+price WHERE YEAR_of_publication<2015;

book_id	title	AUTHOR	publisher	YEAR_of_publication	price	genre
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	5500	Modern Indian Social Drama
102	1984	George Orwell	Secker & Warburg	1956	8800	Political Fiction, Dystopian
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1872	4400	Modernist Indian Fiction
105	Moby-Dick	Herman Melville	Harper & Brothers	2001	5940	Adventure, Mythological

Lab 3: Remove all members who joined before 2020 from the members table.

ANS :-

DELETE FROM members_backup WHERE date_of_membership<2020-01-01;

member_name	date_of_membership
John Doe	2022-05-10
Jane Smith	2023-01-15
Alice Johnson	2021-11-25
Bob Brown	2023-03-30
Charlie Davis	2022-07-20

Lab 4: Delete all books that have a NULL value in the author column.

ANS :-

DELETE FROM books WHERE AUTHOR is null;

book_id	title	AUTHOR	publisher	YEAR_of_publication	price	genre
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	8053	Modern Indian Social Drama
102	1984	George Orwell	Secker & Warburg	1956	12884	Political Fiction, Dystopian
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1872	6442	Modernist Indian Fiction
105	Moby-Dick	Herman Melville	Harper & Brothers	2001	8697	Adventure, Mythological

Lab 4: Write a query to retrieve all books with price between \$50 and \$100.

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

ANS :-

```
SELECT * FROM books WHERE price BETWEEN 4200 AND 8500;
```

book_id	title	AUTHOR	publisher	YEAR_of_publication	price	genre
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	8053	Modern Indian Social Drama
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1872	6442	Modernist Indian Fiction

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 books ORDER BY AUTHOR LIMIT 3 ;
```

book_id	title	AUTHOR ▲ 1	publisher	YEAR_of_publication	price	genre
103	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1872	6442	Modernist Indian Fiction
102	1984	George Orwell	Secker & Warburg	1956	12884	Political Fiction, Dystopian
101	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	8053	Modern Indian Social Drama

Lab 3: Perform an INNER JOIN between books and authors tables to display the title of books and their respective authors' names.

ANS :-

```
SELECT book.b_id , book.b_title , author1.A_name FROM book INNER JOIN author1 ON book.b_id = author1.b_id;
```

b_id	b_title	A_name
111	Bhakti Kavya	Narsinh Mehta
112	Bhakti Geet	Mirabai
113	Gujarat Na Kavi	Suresh Joshi
114	Sambhavna	Jayant Khatri
115	Gyanpith	Trivedi Gopaldas
116	Bhumika	Kavi Kant
117	Jeevansutra	Bhavesh Parekh

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 book.b_id , book.b_title , author1.A_name FROM book FULL JOIN author1 ON book.b_id = author1.b_id;
```

Lab 3: Group books by genre and display the total number of books in each genre.

ANS :-

```
SELECT genre , COUNT(title) FROM books GROUP BY genre;
```

genre	COUNT(title)
Adventure, Mythological	1
Modern Indian Social Drama	1
Modernist Indian Fiction	2
Political Fiction, Dystopian	1

Lab 4: Group members by the year they joined and find the number of members who joined each year.

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

ANS :-

```
SELECT year , COUNT(member_name) FROM members_backup GROUP BY year;
```

year	COUNT(member_name)
2001	1
2002	2
2005	2

Lab 3: Write a stored procedure to retrieve all books by a particular author.

ANS :-

DELIMITER \$\$

```
CREATE PROCEDURE pro_1 (b_id int , book_name varchar(60) , author_name varchar(60) , price int )  
BEGIN
```

```
    INSERT INTO books VALUES(b_id , book_name , author_name , price);
```

```
END;
```

```
CALL pro_1(1, 'Godan', 'Premchand', 3000);
```

```
CALL pro_1(2, 'Rangbhoomi', 'Premchand', 3500);
```

```
CALL pro_1(3, 'Madhushala', 'Harivansh Rai Bachchan', 2500);
```

b_id	book_name	author_name	price
1	Godan	Premchand	3000
2	Rangbhoomi	Premchand	3500
3	Madhushala	Harivansh Rai Bachchan	2500

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;
```

```
@price
```

```
3000.00
```

Lab 3: Create a view to show only the title, author, and price of books from the books table.

ANS :-

```
CREATE VIEW v_1 AS SELECT title , AUTHOR , price FROM books;
```

```
SELECT * FROM v_1;
```


EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

title	AUTHOR	price
To Kill a Mockingbird	Harper Lee	8053
1984	George Orwell	12884
The Great Gatsby	F. Scott Fitzgerald	6442
Moby-Dick	Herman Melville	8697
The Great Gatsby	F. Scott Fitzgerald Charles Scribner	6442

Lab 4: Create a view to display members who joined before 2020.

ANS :-

```
CREATE VIEW v_3 AS SELECT member_name FROM members_backup WHERE year<2020;
SELECT * FROM v_3;
```

member_name

John Doe

Alice Johnson

Charlie Davis

Lab 3: Create a trigger to automatically update the last_modified timestamp of the books table whenever a record is updated.

ANS :-

DELIMITER \$\$

```
CREATE TRIGGER t_22 AFTER UPDATE ON book_1 FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO author(id , b_title , pub_years , records)VALUES(new.book_id ,new.book_name , new.pub_year ,
'Record is Successfully update.');
```

```
END
```

```
UPDATE book_1 SET pub_year=1944 WHERE book_id=101;
```

id	b_title	pub_years	DOT	records
0 1		1950	2025-01-18 23:43:30	Record is Successfully inserted.
0 2		2000	2025-01-18 23:43:30	Record is Successfully inserted.
0 3		1920	2025-01-18 23:43:30	Record is Successfully inserted.
101	Satyanveshi	1950	2025-01-18 23:44:23	Record is Successfully inserted.
102	Shreemad Bhagavad Gita	2000	2025-01-18 23:44:23	Record is Successfully inserted.
103	English Society	1920	2025-01-18 23:44:23	Record is Successfully inserted.
101	Satyanveshi	1944	2025-01-18 23:46:21	Record is Successfully update.

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 \$\$

EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

```
CREATE TRIGGER t_22 AFTER INSERT ON book_1 FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO author(id , b_title , pub_years , records)VALUES(new.book_id ,new.book_name , new.pub_year ,  
'Record is Successfully inserted.');
```

```
END
```

id	b_title	pub_years	DOT	records
0	1	1950	2025-01-18 23:43:30	Record is Successfully inserted.
0	2	2000	2025-01-18 23:43:30	Record is Successfully inserted.
0	3	1920	2025-01-18 23:43:30	Record is Successfully inserted.
101	Satyanveshi	1950	2025-01-18 23:44:23	Record is Successfully inserted.
102	Shreemad Bhagavad Gita	2000	2025-01-18 23:44:23	Record is Successfully inserted.
103	English Society	1920	2025-01-18 23:44:23	Record is Successfully inserted.
101	Satyanveshi	1944	2025-01-18 23:46:21	Record is Successfully update.

```
DELIMITER $$
```

```
CREATE TRIGGER t_24 AFTER DELETE ON book_1 FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO author(id , b_title , pub_years , records)VALUES(old.book_id ,old.book_name , old.pub_year ,  
'Record is Successfully deleted.');
```

```
END
```

id	b_title	pub_years	DOT	records
0	1	1950	2025-01-18 23:43:30	Record is Successfully inserted.
0	2	2000	2025-01-18 23:43:30	Record is Successfully inserted.
0	3	1920	2025-01-18 23:43:30	Record is Successfully inserted.
101	Satyanveshi	1950	2025-01-18 23:44:23	Record is Successfully inserted.
102	Shreemad Bhagavad Gita	2000	2025-01-18 23:44:23	Record is Successfully inserted.
103	English Society	1920	2025-01-18 23:44:23	Record is Successfully inserted.
101	Satyanveshi	1944	2025-01-18 23:46:21	Record is Successfully update.
101	Satyanveshi	1944	2025-01-18 23:51:38	Record is Successfully deleted.