ExLab: - 1:

lab-1:

1. CREATE DATABASE library_db;

2. CREATE TABLE books (

book_id INT PRIMARY KEY,

title VARCHAR(200),author VARCHAR(100),

publisher VARCHAR(100),

year_of_publication INT,

price DECIMAL(8, 2)

);

3. INSERT INTO books (book_id, title, author, publisher,

year_of_publication, price) VALUES

(1, 'The Great Gatsby', 'F. Scott', 'Scribner', 1925, 10.99),

lab-2:

1. CREATE TABLE members (

member_id INT PRIMARY KEY, member_name,

VARCHAR(100), date_of_membership DATE, email

VARCHAR(100)

);

2. INSERT INTO members (member_id, member_name,

date_of_membership, email) VALUES

(1, 'Alice Johnson', '2021-01-15', 'alice.johnson@example.com'),

---

ExLab: - 2:

lab -1:

SELECT *

FROM members

WHERE date_of_membership < '2022-01-01'

ORDER BY date_of_membership;

lab -2: SELECT title

FROM books

```
WHERE author = 'George Orwell' ORDER BY
year_of_publication DESC;
```

ExLab: - 3:

lab -1:

```
ALTER TABLE books
ADD CONSTRAINT chk_price_positive CHECK (price > 0);
```

lab -2:

```
ALTER TABLE members
ADD CONSTRAINT uq_member_email UNIQUE (email);
```

ExLab: - 4:

lab -1:

```
CREATE TABLE authors (
author_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
country VARCHAR(50)
);
```

lab -2:

```
CREATE TABLE publishers (
publisher_id INT PRIMARY KEY,
publisher_name VARCHAR(100),
contact_number VARCHAR(20) UNIQUE,
address VARCHAR(150)
);
```

ExLab: - 5:

lab -1:

```
ALTER TABLE books
ADD genre VARCHAR(50);
```

UPDATE books SET genre = 'Classic';

lab -2:

ALTER TABLE members

MODIFY email VARCHAR(100);

ALTER TABLE members

ALTER COLUMN email TYPE VARCHAR(100);

ExLab: - 6:

lab -1:

DESC publishers;

DROP TABLE publishers;

lab -2:

CREATE TABLE members_backup AS SELECT * FROM

members;

DROP TABLE members;

ExLab: - 7:

lab -1:

INSERT INTO authors (author_id, first_name, last_name) VALUES

(101, 'John', 'Smith');UPDATE authors SET last_name = 'Williams' WHERE author_id =

103;

lab -2:

DELETE FROM books WHERE price > 100;

ExLab: - 8:

lab -1:

UPDATE books SET year_of_publication = 2022 WHERE book_id = 5;

lab -2:

UPDATE books SET price = price * 1.10 WHERE

year_of_publication < 2015;

ExLab: - 9:

lab -1:

```
DELETE FROM members WHERE join_date < '2020-01-01';
```

lab -2:

```
DELETE FROM books WHERE author IS NULL;
```

ExLab:- 10:

lab -1:

```
SELECT * FROM books WHERE price BETWEEN 50 AND 100;
```

lab -2:

```
SELECT * FROM books ORDER BY author ASC LIMIT 3;
```

ExLab: - 11:

lab -1:

```
GRANT SELECT ON books TO librarian;
```

lab -2:

```
GRANT INSERT, UPDATE ON members TO admin;
```

ExLab: - 12:

lab -1:

```
REVOKE INSERT ON books FROM librarian;
```

lab -2:

```
REVOKE ALL PRIVILEGES ON members FROM admin;
```

ExLab: - 13:

lab -1:

```
BEGIN;
INSERT INTO books (book_id, title, author, price) VALUES (201,
'SQL Basics', 'John Smith', 45);
INSERT INTO books (book_id, title, author, price) VALUES (202,
'Advanced SQL', 'Emily Johnson', 75);
```

```
COMMIT;
INSERT INTO books (book_id, title, author, price) VALUES (203,
'SQL Mastery', 'Michael Brown', 95);
ROLLBACK;
```

lab -2:

```
BEGIN;
SAVEPOINT before_update;
UPDATE members SET status = 'inactive' WHERE last_login <
'2022-01-01';
UPDATE members SET membership_type = 'basic' WHERE
membership_type = 'premium';
ROLLBACK TO SAVEPOINT before_update;
COMMIT;
```

ExLab: - 14:

lab -1:

```
SELECT books.title, authors.first_name, authors.last_name
FROM books
INNER JOIN authors ON books.author_id = authors.author_id;
```

lab -2:

```
SELECT books.title, authors.first_name, authors.last_name
FROM books
FULL OUTER JOIN authors ON books.author_id =
Authors.author_id;
```

ExLab: - 15:

lab -1:

```
SELECT genre, COUNT(*) AS total_books
FROM books
GROUP BY genre;
```

lab -2:

```
SELECT EXTRACT(YEAR FROM join_date) AS join_year,
COUNT(*) AS total_members
FROM members
GROUP BY EXTRACT(YEAR FROM join_date);
```

ExLab: - 16:

lab -1:

```
CREATE PROCEDURE GetBooksByAuthor(IN authorName
VARCHAR(100))
BEGIN
SELECT * FROM books WHERE author = authorName;
END;
```

lab -2:

```
CREATE PROCEDURE GetBookPrice(IN b_id INT)
BEGIN
SELECT price FROM books WHERE book_id = b_id;
END;
```

ExLab: - 17:

lab -1:

```
CREATE VIEW book_summary AS
SELECT title, author, price FROM books;
```

lab -2:

```
CREATE VIEW early_members AS
SELECT * FROM members WHERE join_date < '2020-01-01';
```

ExLab: - 18:

lab -1:

```
CREATE TRIGGER update_last_modified
BEFORE UPDATE ON books
FOR EACH ROW
```

```
SET NEW.last_modified = NOW();
```

lab -2:

```
CREATE TRIGGER log_book_deletion
AFTER DELETE ON books
FOR EACH ROW
INSERT INTO log_changes (action_type, book_id, action_time)
VALUES ('DELETE', OLD.book_id, NOW());
```

ExLab: - 19:

lab -1:

```
BEGIN
INSERT INTO books (book_id, title, author, price)
VALUES (301, 'PLSQL Guide', 'Anna Scott', 59.99);
DBMS_OUTPUT.PUT_LINE('Book inserted successfully.');
END;
```

lab -2:

```
DECLARE
total_books NUMBER;
BEGIN
SELECT COUNT(*) INTO total_books FROM books;
DBMS_OUTPUT.PUT_LINE('Total number of books: ' ||
total_books);
END;
```

ExLab: - 20:

lab -1:

```
DECLARE
book_id NUMBER := 101;
price NUMBER := 49.99;
BEGIN
DBMS_OUTPUT.PUT_LINE('Book ID: ' || book_id || ', Price: $' ||
```

```
price);
END;
lab -2:
DECLARE
CONSTANT discount_rate NUMBER := 0.10;
original_price NUMBER := 100;
final_price NUMBER;
BEGIN
final_price := original_price - (original_price * discount_rate);
DBMS_OUTPUT.PUT_LINE('Discounted price: $' || final_price);
END;
```

```
ExLab: - 21:
lab -1:
DECLARE
price NUMBER := 120;
BEGIN
IF price > 100 THEN
DBMS_OUTPUT.PUT_LINE('The book is expensive.');
ELSE
DBMS_OUTPUT.PUT_LINE('The book is affordable.');
END IF;
END;
lab -2:
DECLARE
CURSOR book_cursor IS SELECT title, author, price FROM
books;
v_title books.title%TYPE;
v_author books.author%TYPE;
v_price books.price%TYPE;
BEGIN
```

```
FOR book_record IN book_cursor LOOP
DBMS_OUTPUT.PUT_LINE('Title: ' || book_record.title ||
', Author: ' || book_record.author ||
', Price: $' || book_record.price);
END LOOP;
END;
```

ExLab: - 22:

lab -1:

```
DECLARE
CURSOR member_cursor IS SELECT * FROM members;
v_member members%ROWTYPE;
BEGIN
OPEN member_cursor;
LOOP
FETCH member_cursor INTO v_member;
EXIT WHEN member_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Member ID: ' ||
v_member.member_id ||
', Name: ' || v_member.name);
END LOOP;
CLOSE member_cursor;
END;
```

lab -2:

```
DECLARE
CURSOR author_books IS SELECT title FROM books WHERE
author = 'John Smith';
v_title books.title%TYPE;
BEGIN
OPEN author_books;
LOOP
```

```
FETCH author_books INTO v_title;
EXIT WHEN author_books%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
END LOOP;
CLOSE author_books;
END;
```

ExLab: - 23:

lab -1:

```
START TRANSACTION;
INSERT INTO members (member_id, name, join_date) VALUES
(401, 'David Green', '2025-07-01');
SAVEPOINT before_update;
UPDATE members SET name = 'David G.' WHERE member_id =
401;
ROLLBACK TO before_update;
COMMIT;
```

lab -2:

```
START TRANSACTION;
INSERT INTO books (book_id, title, author, price) VALUES (501,
'Database Systems', 'Alan Turing', 60);
INSERT INTO books (book_id, title, author, price) VALUES (502,
'AI and SQL', 'Ada Lovelace', 85);
COMMIT;
START TRANSACTION;
SAVEPOINT update_point;
UPDATE books SET price = price + 10 WHERE book_id = 501;
```