1. display dept wise total employee in ascending order.  
   2. display dept wise sum(salary).  
   3. display dept wise sum(salary) and sum(salary) should be greater than 1lakh

1 : select dept , count(id)  from employee group by dept

2 : select dept , sum(salary)  from employee group by dept

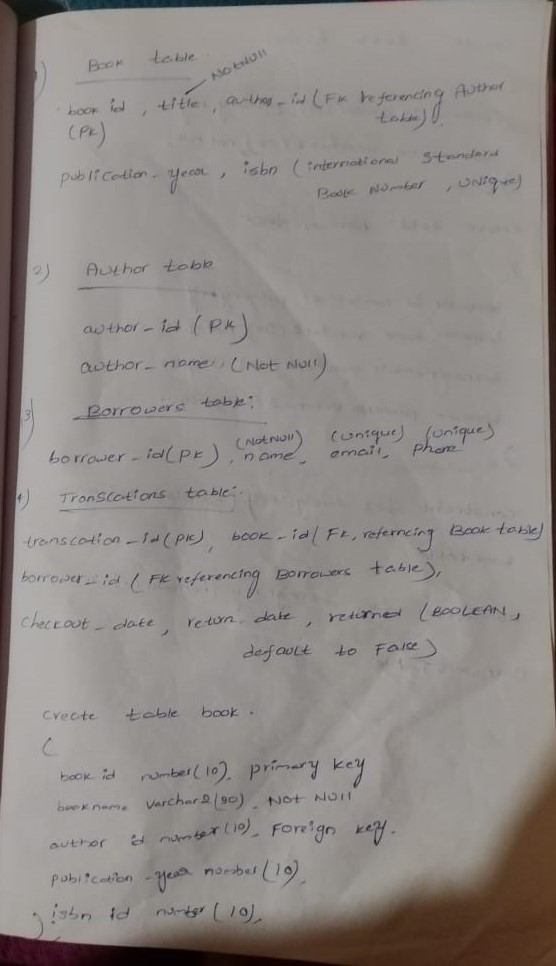
3 : select dept , sum(salary)  from employee group by dept having 100000

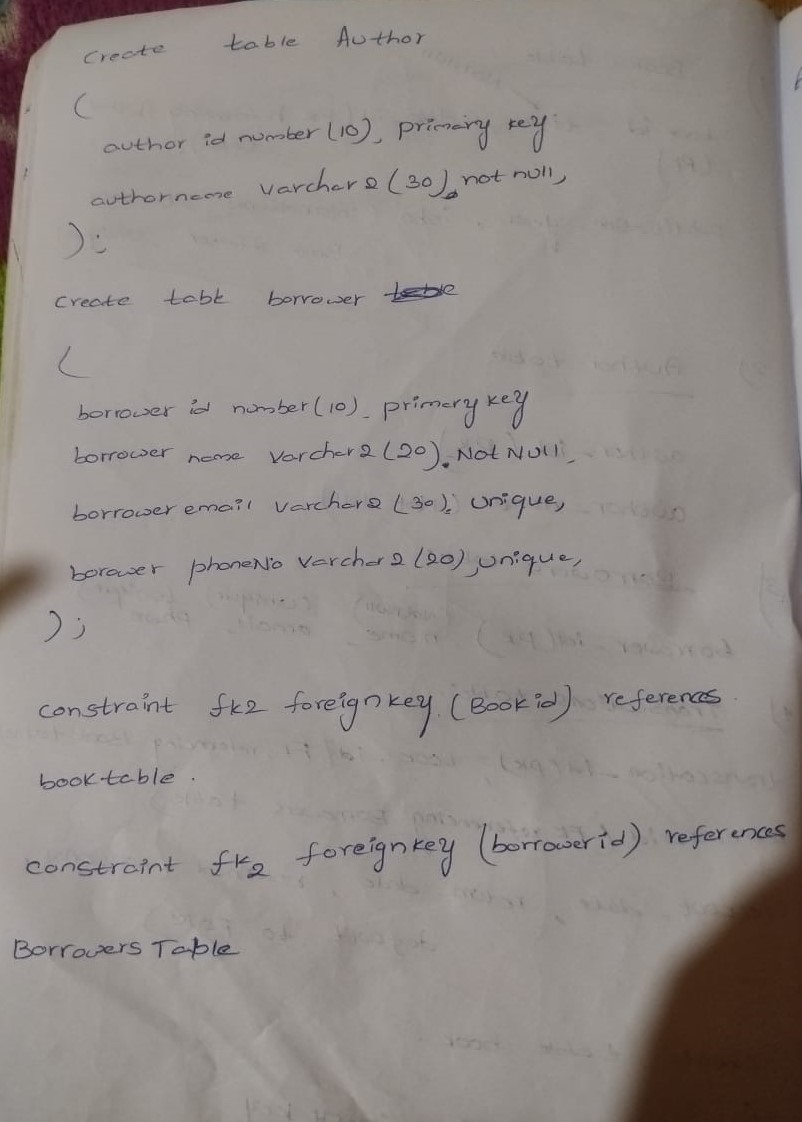
Assignment 1: Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

Assignment 2: 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.

Assignment 3: Write a SELECT query to retrieve all columns from a 'customers' table, and modify it to return only the customer name and email address for customers in a specific city.

Assignment 4: Craft a query using an INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region, and a LEFT JOIN to display all customers including those without orders.

Asssignment1:



Assignment2:

1. **Creating Tables:** Assuming you have tables like books, authors, members, and transactions

-- Create the books table

CREATE TABLE books (

book\_id INT PRIMARY KEY,

title VARCHAR(255),

author\_id INT,

genre VARCHAR(100),

publication\_year INT,

ISBN VARCHAR(20),

available BOOLEAN

);

-- Create the authors table

CREATE TABLE authors (

author\_id INT PRIMARY KEY,

author\_name VARCHAR(255),

nationality VARCHAR(100),

birth\_year INT

);

-- Create the members table

CREATE TABLE members (

member\_id INT PRIMARY KEY,

member\_name VARCHAR(255),

member\_email VARCHAR(255),

membership\_date DATE

);

-- Create the transactions table

CREATE TABLE transactions (

transaction\_id INT PRIMARY KEY,

book\_id INT,

member\_id INT,

issue\_date DATE,

return\_date DATE,

FOREIGN KEY (book\_id) REFERENCES books(book\_id),

FOREIGN KEY (member\_id) REFERENCES members(member\_id)

);

1. **Modifying Table Structure:** If you need to alter any table structures, you can use the ALTER TABLE statement. For example, if you want to add a new column to the books table:

ALTER TABLE books

ADD COLUMN language VARCHAR(50);

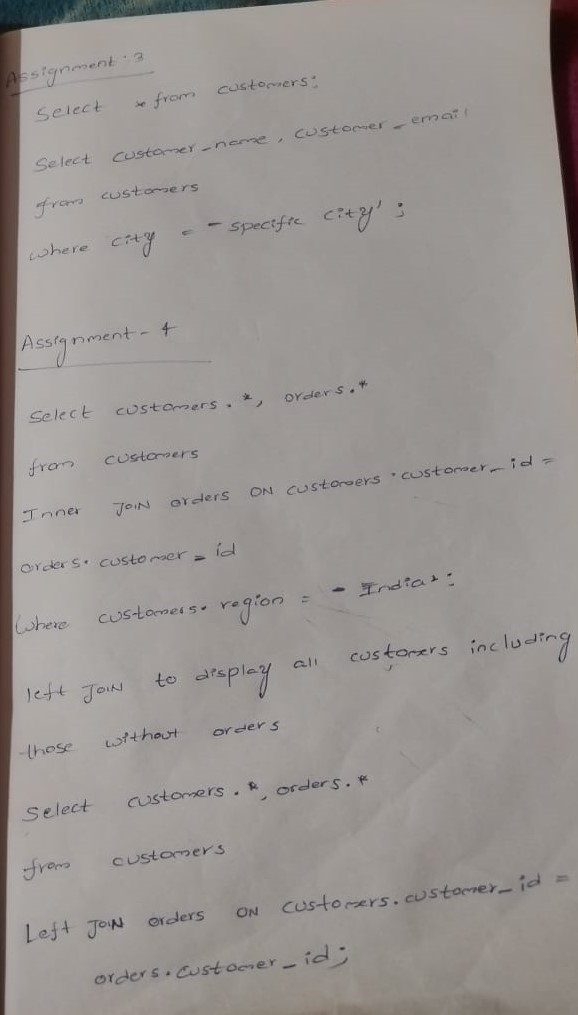
1. **Dropping a Redundant Table:** If you have a redundant table that you want to remove, you can use the DROP TABLE statement. For example, if you want to drop a table named redundant\_table

DROP TABLE redundant\_table;

Assignment3: SELECT \* FROM customers;

SELECT customer\_name, customer\_email FROM customers

WHERE city = 'New York';

Assignment4:

Top of Form

Bottom of Form