**Question 1:**

CREATE TABLE users (

id INT PRIMARY KEY,

firstname VARCHAR(30),

lastname VARCHAR(50),

email VARCHAR(30),

password VARCHAR(20)

);

**Problem Description**

You are required to build a **Java application using JDBC** that allows users to:

1. **Register** with their details.
2. **Login** using their email and password.

**Functional Requirements**

**1. User Registration**

* The application should accept the user's details:
  + ID (integer, unique)
  + First Name
  + Last Name
  + Email (unique)
  + Password
* It should insert the data into the users table.
* If the email or ID already exists, it should show an error message.

**2. User Login**

* The user should provide:
  + Email
  + Password
* The application should:
  + Verify the credentials against the users table.
  + Show a success message if matched, otherwise show an error.

Enter ID: 1

Enter First Name: Alice

Enter Last Name: Johnson

Enter Email: alice@example.com

Enter Password: alice123

User registered successfully!

Sample input 2:

Enter ID: 2

Enter First Name: Bob

Enter Last Name: Smith

Enter Email: alice@example.com

Enter Password: bob123

Output:

Error: Email already exists. Please use a different email.

Sample Input:

Enter Email: alice@example.com

Enter Password: alice123

Login successful! Welcome, Alice Johnson.

Enter Email: alice@example.com

Enter Password: wrongpass

Login failed: Invalid email or password.

**Question 2:**

CREATE TABLE events (

event\_id INT PRIMARY KEY,

event\_name VARCHAR(100),

location VARCHAR(100),

event\_date DATE,

organizer\_email VARCHAR(100)

);

Build a **Java application using JDBC** that manages events. The application should allow users to:

1. **Add new events**
2. **View all events scheduled after a certain date**.

**Functional Requirements**

**1. Add Event**

* Prompt the user to enter the following:
  + Event ID (unique)
  + Event Name
  + Location
  + Event Date (YYYY-MM-DD)
  + Organizer Email
* Store this data in the events table.

**2. View Events After a Specific Date**

* Prompt the user to enter a date (YYYY-MM-DD).
* Display all events **scheduled after** the given date, sorted by event\_date.

**3. View Events After a Specific Date**

* Enter a **start date** and an **end date**.
* Retrieve and display all events **scheduled between** those two dates (inclusive), sorted by date.

SELECT \* FROM events

WHERE event\_date BETWEEN ? AND ?

ORDER BY event\_date;

**Sample Input: Add Event**

Enter Event ID: 101

Enter Event Name: Tech Conference

Enter Location: New York

Enter Event Date (YYYY-MM-DD): 2025-12-01

Enter Organizer Email: organizer@tech.com

**Output:**

Event added successfully!

**Sample Input: View Events After Date**

Enter date (YYYY-MM-DD): 2025-11-01

**Output:**

Upcoming Events After 2025-11-01:

ID Name Location Date Organizer

-------------------------------------------------------------

101 Tech Conference New York 2025-12-01 organizer@tech.com

**Input:**

Enter start date (YYYY-MM-DD): 2025-11-01

Enter end date (YYYY-MM-DD): 2025-12-31

**Output:**

Events between 2025-11-01 and 2025-12-31:

ID Name Location Date Organizer

-------------------------------------------------------------

101 Tech Conference New York 2025-12-01 organizer@tech.com

102 AI Meetup San Jose 2025-11-20 contact@ai.org

If no events are found:

No events found between 2025-11-01 and 2025-12-31.

**Question 3:**

**JDBC Problem Statement: Insert Record Only If Not Exists**

**CREATE TABLE employees (**

emp\_id INT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

department VARCHAR(50)

**);**

**Problem Description**

Create a Java application using JDBC that allows inserting new employee records into the employees table.  
Before inserting, the application must:

* Check if the emp\_id or email already exists in the table.
* If either exists:
  + Do not insert.
  + Show an appropriate error message or throw a custom exception.
* If neither exists:
  + Proceed with the insertion and show a success message.

**Functional Requirements**

1. Insert New Employee

Prompt the user to enter:

* emp\_id (int)
* name (string)
* email (string)
* department (string)

Then:

* Check the table to see if the emp\_id or email already exists.
* If yes:
  + Print:
    - "Employee with this ID already exists." or
    - "Email is already registered."
* If no:
  + Insert the employee data into the table.
  + Show: "Employee added successfully!"

**SQL Queries**

**Check if emp\_id or email exists:**

**SELECT \* FROM employees WHERE emp\_id = ? OR email = ?;**

**Insert if not found:**

**INSERT INTO employees (emp\_id, name, email, department)**

**VALUES (?, ?, ?, ?);**

**Sample Input & Output**

**Input:**

Enter Employee ID: 101

Enter Name: John Doe

Enter Email: john.doe@example.com

Enter Department: IT

**Output:**

Employee added successfully!

**Input (Duplicate Email):**

Enter Employee ID: 102

Enter Name: Jane Smith

Enter Email: john.doe@example.com

Enter Department: HR

**Output:**

Error: Email is already registered.