H2 Database is a lightweight, fast, open-source relational database management system written in Java. It is often used for development, testing, and learning purposes because it can run in memory or as a standalone database with minimal setup. H2 supports standard SQL, provides a web-based console for easy management, and is commonly embedded in Java applications, especially with frameworks like Spring Boot. Its simplicity and zero-configuration nature make it ideal for prototyping and unit testing.

Folder Steps:

- Copy the H2Database folder to golden versions directory and extract it.
  -
- Create folder h2db inside C:\Users\<username>
- Purpose of creating h2db folder inside C:\Users\<username> is to store db data inside this directory.

How to run H2 DB

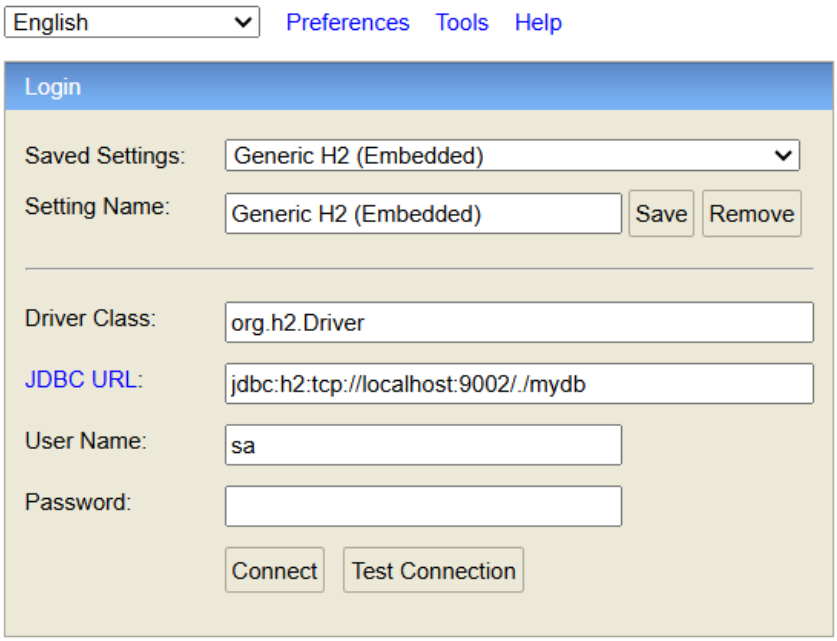
- Open command prompt and navigate to directory where H2Database > 2.1.214 is located

- Inside the command prompt, execute the command:
  - **java -cp h2-2.1.214.jar org.h2.tools.Server -web -webPort 9001 -tcp -tcpPort 9002 -ifNotExists -baseDir C:/Users/<username>/h2db** (remember this folder name can be anything which you had created in the previous step)
  - **cp** stands for classpath
  - **9001** is the port which will be running to see on web browser. You can use any other port of your choice.
  - **9002** is the port which will be running on tcp and helps to connect from other application like java program.
  - **-baseDir C:/Users/<username/h2db -** This specifies the base directory where you want to store the data. Remember: you have to create folder h2db before executing the jar command else it will show db not found.

- You will see the message if the DB started successfully:

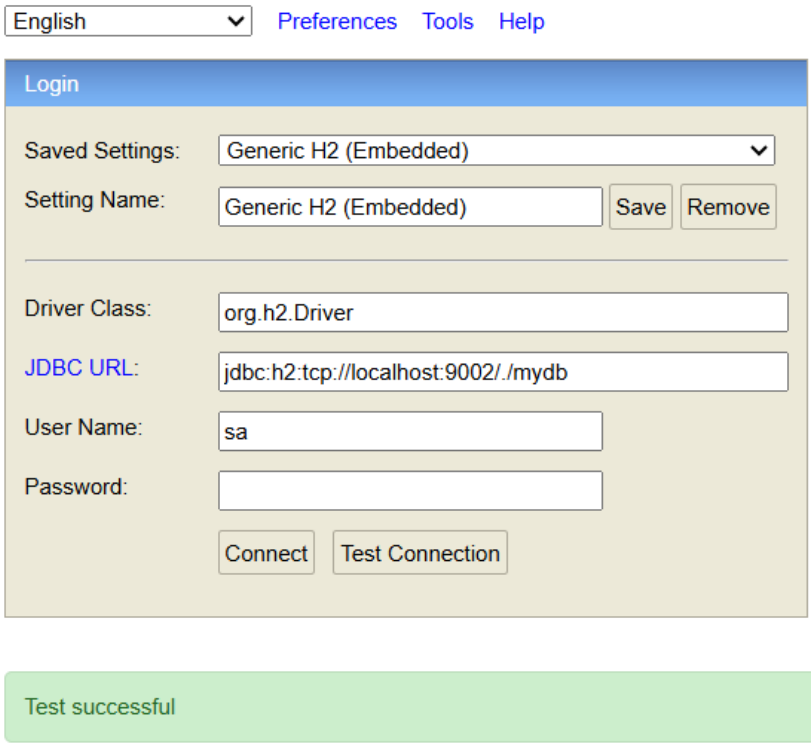
```
TCP server running at tcp://192.168.1.3:9002 (only local connections)
Web Console server running at http://192.168.1.3:9001 (others can connect)
```

- Open browser and enter **http://localhost:9001** and you will see the H2 DB console will be displayed.

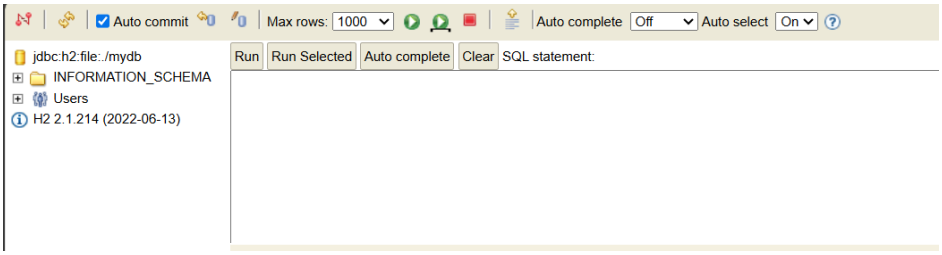


- Inside JDBC URL textbox, enter- **jdbc:h2:tcp://localhost:9002/./<your-db-name>**(refer the above image. For eg: I have given **mydb** you can give any other name. It will create db if does not exists inside the base directory which was mentioned in the previous step)
  - username: sa
  - password: leave blank

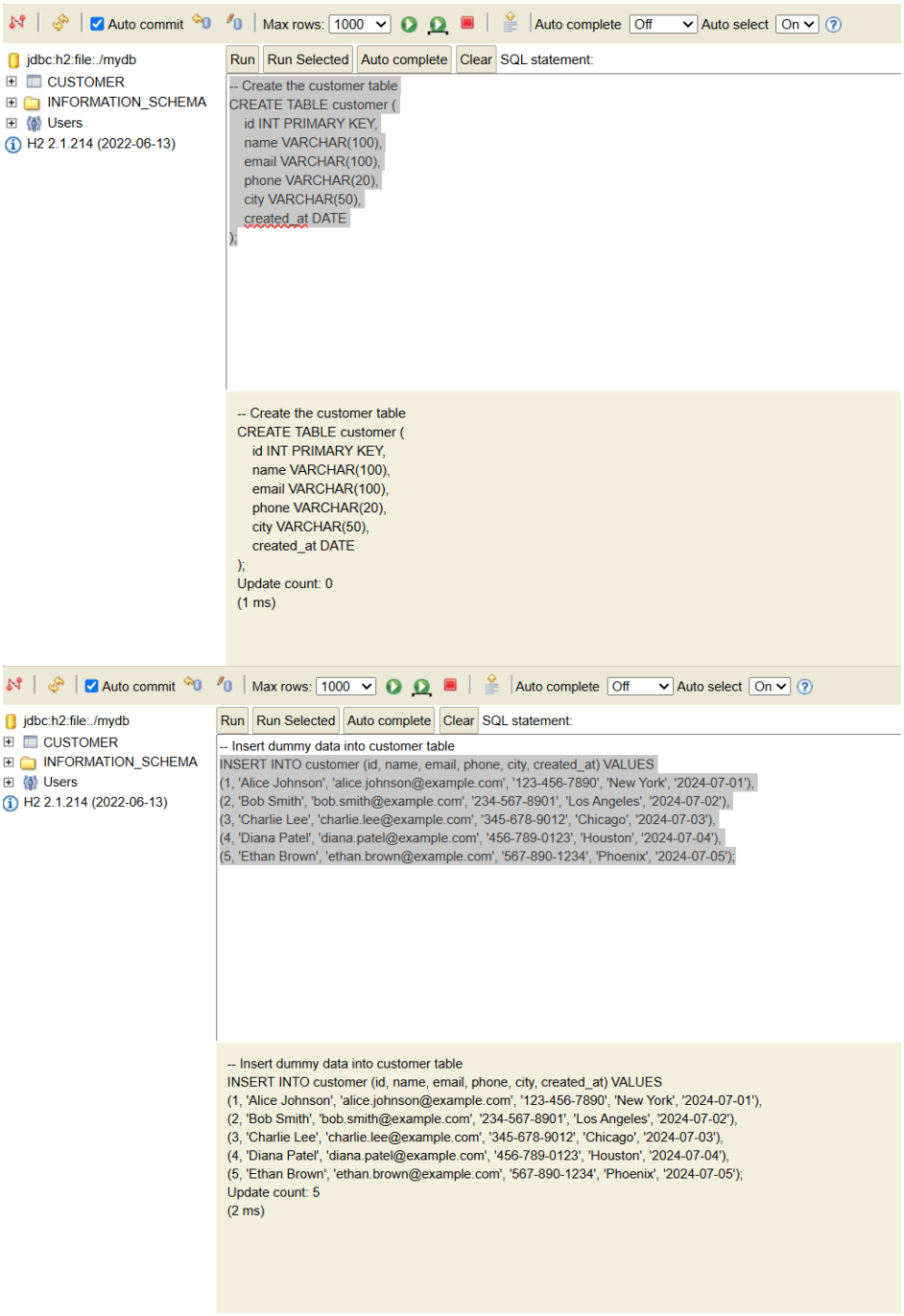
- **Click Test Connection**. If all configuration is proper, you will see message: Test Successful



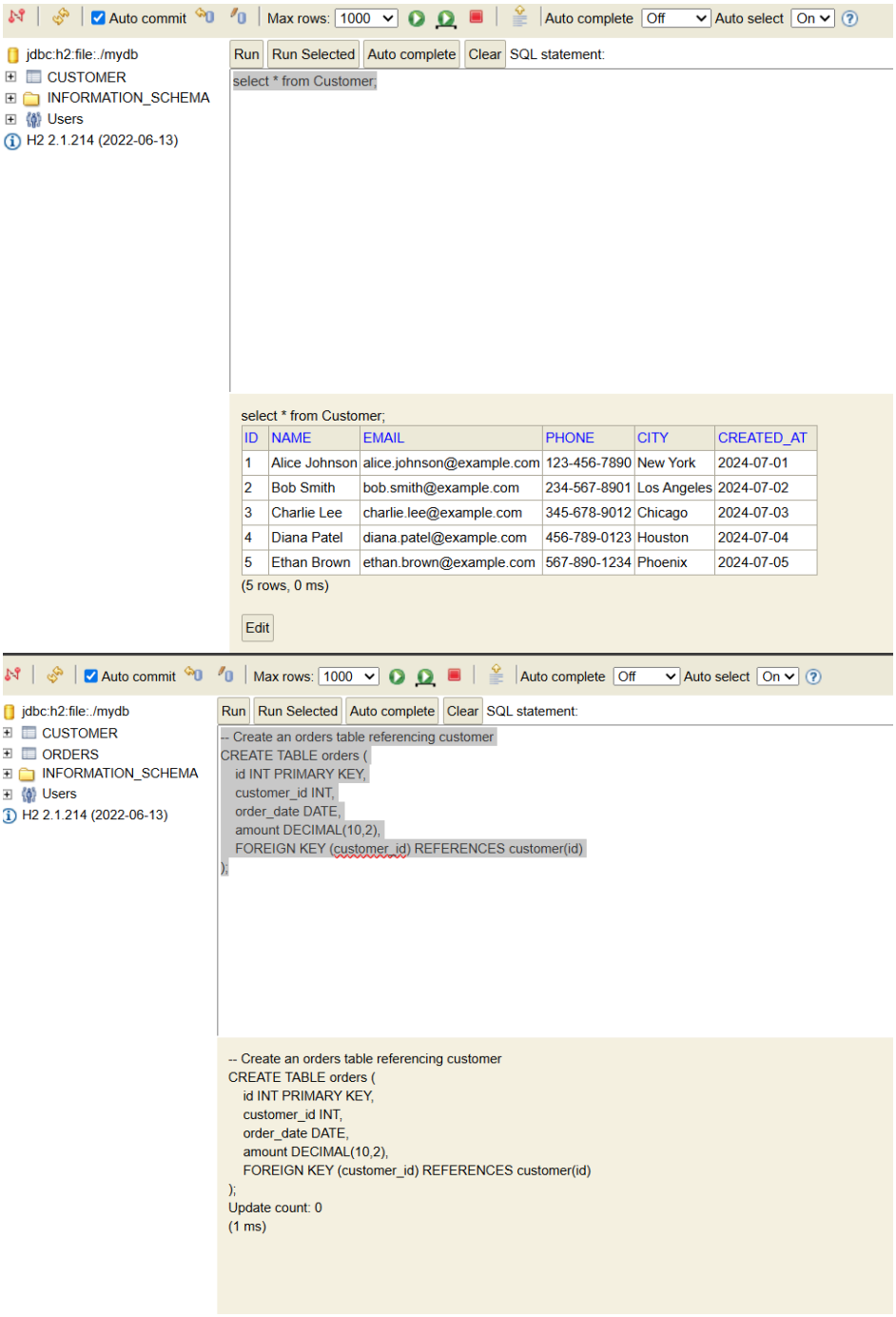
- If you see Test Successful Message, click **Connect** button to enter H2 DB console.

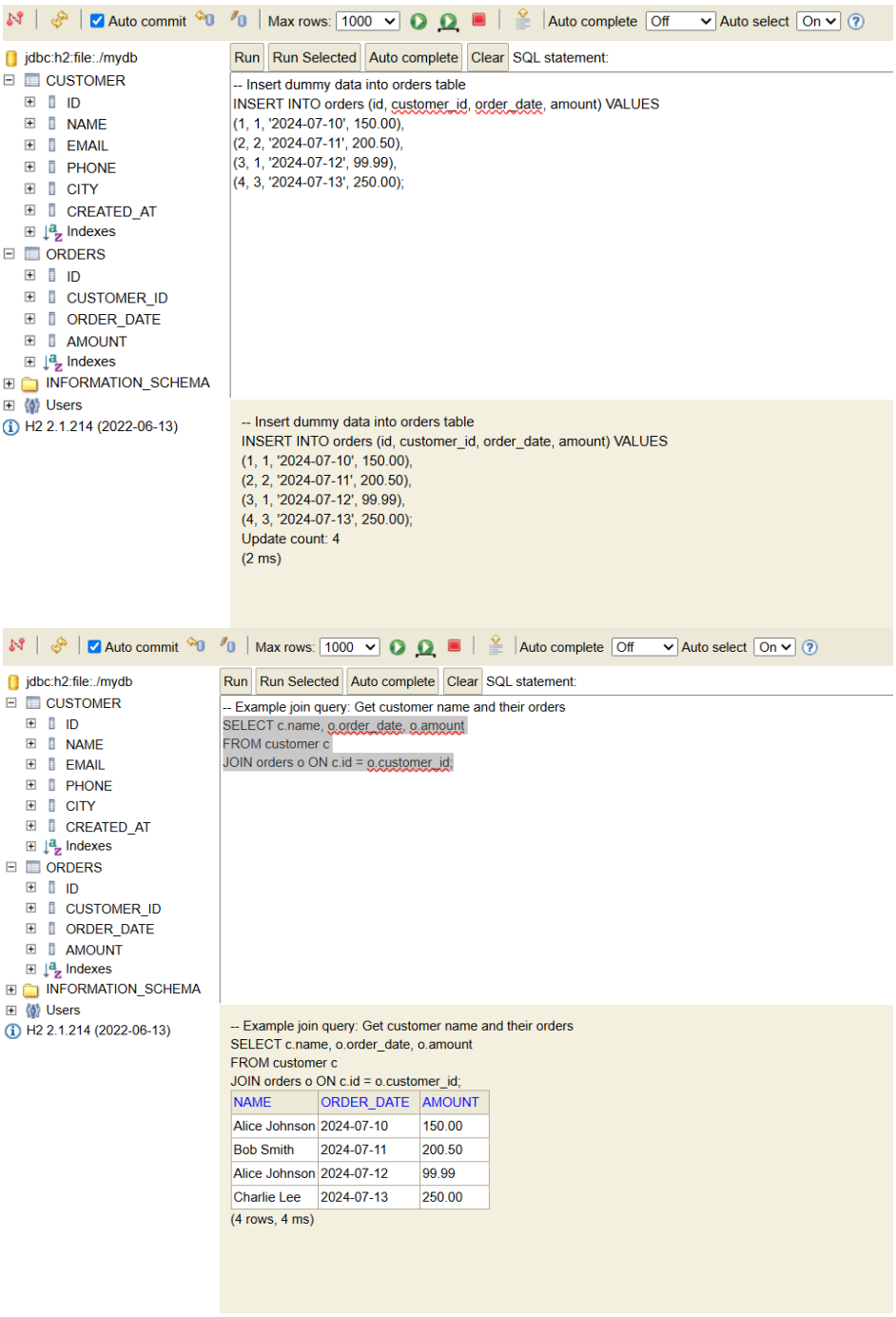


- You are now **successfully connected to H2DB**. Execute SQL query command in the query box and **click Run** Button, you will be able to see query getting executed successfully.


Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:./mydb
CUSTOMER
INFORMATION_SCHEMA
Users
H2 2.1.214 (2022-06-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
-- Create the customer table
CREATE TABLE customer (
id INT PRIMARY KEY,
name VARCHAR(100),
email VARCHAR(100),
phone VARCHAR(20),
city VARCHAR(50),
created_at DATE
);
-- Create the customer table
CREATE TABLE customer (
id INT PRIMARY KEY,
name VARCHAR(100),
email VARCHAR(100),
phone VARCHAR(20),
city VARCHAR(50),
created_at DATE
);
Update count: 0
(1 ms)
Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:./mydb
CUSTOMER
INFORMATION_SCHEMA
Users
H2 2.1.214 (2022-06-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
-- Insert dummy data into customer table
INSERT INTO customer (id, name, email, phone, city, created_at) VALUES
(1, 'Alice Johnson', 'alice.johnson@example.com', '123-456-7890', 'New York', '2024-07-01'),
(2, 'Bob Smith', 'bob.smith@example.com', '234-567-8901', 'Los Angeles', '2024-07-02'),
(3, 'Charlie Lee', 'charlie.lee@example.com', '345-678-9012', 'Chicago', '2024-07-03'),
(4, 'Diana Patel', 'diana.patel@example.com', '456-789-0123', 'Houston', '2024-07-04'),
(5, 'Ethan Brown', 'ethan.brown@example.com', '567-890-1234', 'Phoenix', '2024-07-05');
-- Insert dummy data into customer table
INSERT INTO customer (id, name, email, phone, city, created_at) VALUES
(1, 'Alice Johnson', 'alice.johnson@example.com', '123-456-7890', 'New York', '2024-07-01'),
(2, 'Bob Smith', 'bob.smith@example.com', '234-567-8901', 'Los Angeles', '2024-07-02'),
(3, 'Charlie Lee', 'charlie.lee@example.com', '345-678-9012', 'Chicago', '2024-07-03'),
(4, 'Diana Patel', 'diana.patel@example.com', '456-789-0123', 'Houston', '2024-07-04'),
(5, 'Ethan Brown', 'ethan.brown@example.com', '567-890-1234', 'Phoenix', '2024-07-05');
Update count: 5
(2 ms)

PUBLIC


Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:./mydb
CUSTOMER
INFORMATION_SCHEMA
Users
H2 2.1.214 (2022-06-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
select * from Customer;
select * from Customer;
ID NAME EMAIL PHONE CITY CREATED_AT
1 Alice Johnson alice.johnson@example.com 123-456-7890 New York 2024-07-01
2 Bob Smith bob.smith@example.com 234-567-8901 Los Angeles 2024-07-02
3 Charlie Lee charlie.lee@example.com 345-678-9012 Chicago 2024-07-03
4 Diana Patel diana.patel@example.com 456-789-0123 Houston 2024-07-04
5 Ethan Brown ethan.brown@example.com 567-890-1234 Phoenix 2024-07-05
(5 rows, 0 ms)
Edit
Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:./mydb
CUSTOMER
ORDERS
INFORMATION_SCHEMA
Users
H2 2.1.214 (2022-06-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
-- Create an orders table referencing customer
CREATE TABLE orders (
id INT PRIMARY KEY,
customer_id INT,
order_date DATE,
amount DECIMAL(10,2),
FOREIGN KEY (customer_id) REFERENCES customer(id)
);
-- Create an orders table referencing customer
CREATE TABLE orders (
id INT PRIMARY KEY,
customer_id INT,
order_date DATE,
amount DECIMAL(10,2),
FOREIGN KEY (customer_id) REFERENCES customer(id)
);
Update count: 0
(1 ms)


Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:./mydb
CUSTOMER
ID
NAME
EMAIL
PHONE
CITY
CREATED_AT
Indexes
ORDERS
ID
CUSTOMER_ID
ORDER_DATE
AMOUNT
Indexes
INFORMATION_SCHEMA
Users
H2 2.1.214 (2022-06-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
-- Insert dummy data into orders table
INSERT INTO orders (id, customer_id, order_date, amount) VALUES
(1, 1, '2024-07-10', 150.00),
(2, 2, '2024-07-11', 200.50),
(3, 1, '2024-07-12', 99.99),
(4, 3, '2024-07-13', 250.00);
-- Insert dummy data into orders table
INSERT INTO orders (id, customer_id, order_date, amount) VALUES
(1, 1, '2024-07-10', 150.00),
(2, 2, '2024-07-11', 200.50),
(3, 1, '2024-07-12', 99.99),
(4, 3, '2024-07-13', 250.00);
Update count: 4
(2 ms)

Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:./mydb
CUSTOMER
ID
NAME
EMAIL
PHONE
CITY
CREATED_AT
Indexes
ORDERS
ID
CUSTOMER_ID
ORDER_DATE
AMOUNT
Indexes
INFORMATION_SCHEMA
Users
H2 2.1.214 (2022-06-13)
Run
Run Selected
Auto complete
Clear
SQL statement:
-- Example join query: Get customer name and their orders
SELECT c.name, o.order_date, o.amount
FROM customer c
JOIN orders o ON c.id = o.customer_id;
-- Example join query: Get customer name and their orders
SELECT c.name, o.order_date, o.amount
FROM customer c
JOIN orders o ON c.id = o.customer_id;
NAME ORDER_DATE AMOUNT
Alice Johnson 2024-07-10 150.00
Bob Smith 2024-07-11 200.50
Alice Johnson 2024-07-12 99.99
Charlie Lee 2024-07-13 250.00
(4 rows, 4 ms)

Java-H2 DB connectivity

(Ensure the db is connected from the browser as shown in the above step)

```
import java.sql.*;

public class Main {
    public static void main(String[] args) {

        System.out.println("Hello world!");
        // JDBC URL for H2 database
        String jdbcURL = "jdbc:h2:tcp://localhost:9002/./mydb"; // Update
path if needed
        String username = "sa"; // Default username for H2
        String password = ""; // Default password for H2
        String query = "SELECT * FROM CUSTOMER"; // SQL query to fetch all
customers

        try (Connection connection = DriverManager.getConnection(jdbcURL,
username, password)) {
            if (connection != null) {
                System.out.println("Database connected successfully.");
            } else {
                System.out.println("Failed to connect to the database.");
                return;
            }

            try (Statement statement = connection.createStatement();
                 ResultSet resultSet = statement.executeQuery(query)) {

                System.out.println("Customer List:");
                while (resultSet.next()) {
                    int id = resultSet.getInt("id"); // Assuming 'id' is a
column
                    String name = resultSet.getString("name"); // Assuming
'name' is a column
                    String email = resultSet.getString("email"); // Assuming
'email' is a column
                    System.out.println("ID: " + id + ", Name: " + name + ",
Email: " + email);
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Spring Boot - Spring Data JPA - H2 Database

(Ensure the db is connected from the browser as shown in the above step)

application.properties

```
# H2 Database Configuration
spring.datasource.url=jdbc:h2:tcp://localhost:9002/./mydb
spring.datasource.username=sa
spring.datasource.password=
spring.datasource.driver-class-name=org.h2.Driver

# JPA Configuration
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=update
```