

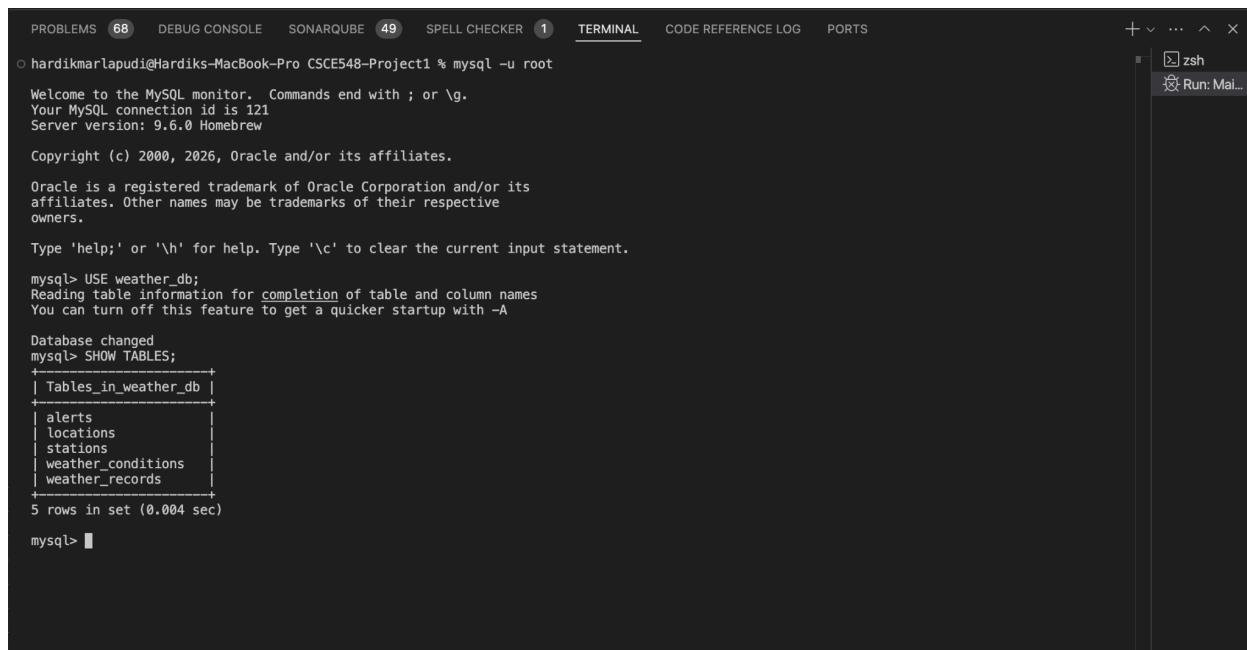
# **Weather Management System**

**CSCE 548**

**Hardik Marlapudi**

**Spring 2026**

# Database Schema Diagram



The screenshot shows a terminal window with the following content:

```
PROBLEMS 68 DEBUG CONSOLE SONARQUBE 49 SPELL CHECKER 1 TERMINAL CODE REFERENCE LOG PORTS
hardikmarlapudi@Hardiks-MacBook-Pro CSCE548-Project1 % mysql -u root

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 121
Server version: 9.6.0 Homebrew

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

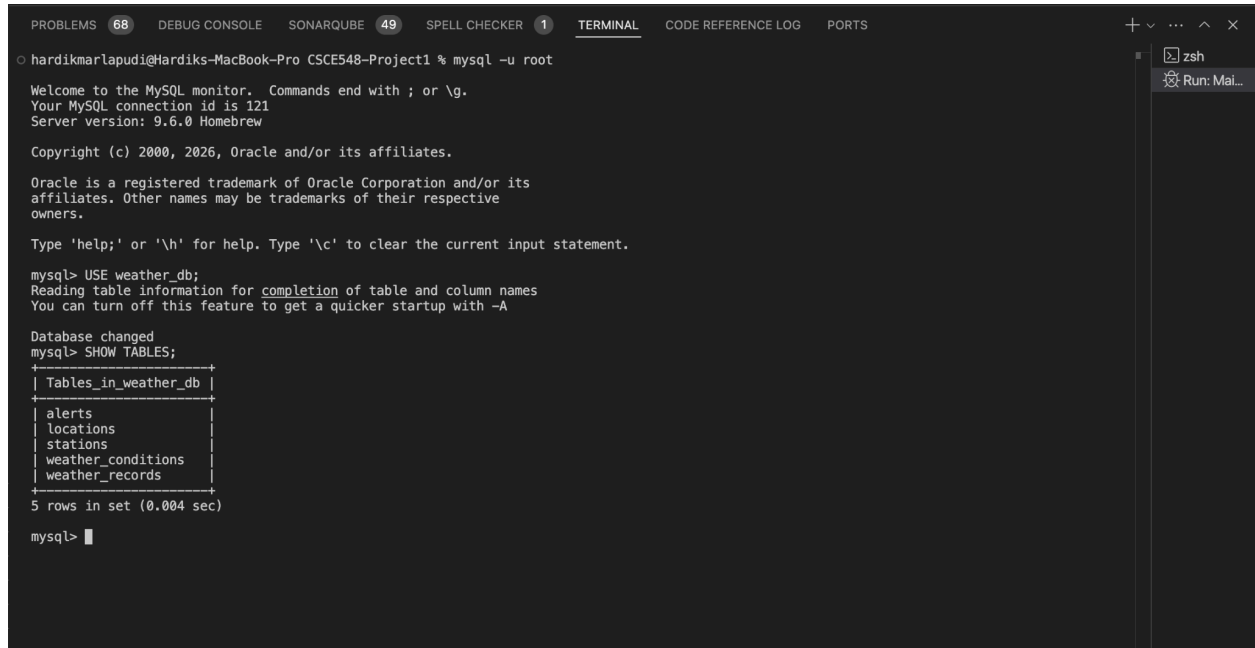
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE weather_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_weather_db |
+-----+
| alerts                |
| locations             |
| stations              |
| weather_conditions    |
| weather_records       |
+-----+
5 rows in set (0.004 sec)

mysql> 
```

# Tables Created



The screenshot shows a terminal window with a dark background. At the top, there is a toolbar with icons for PROBLEMS (68), DEBUG CONSOLE, SONARQUBE (49), SPELL CHECKER (1), TERMINAL, CODE REFERENCE LOG, and PORTS. The terminal content shows a user logging into MySQL as root. After the MySQL welcome message, the user enters 'USE weather\_db;'. Then, they enter 'SHOW TABLES;', which results in a table listing five tables: alerts, locations, stations, weather\_conditions, and weather\_records. The output indicates 5 rows in the set.

```
hardikmarlapudi@Hardiks-MacBook-Pro CSCE548-Project1 % mysql -u root

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 121
Server version: 9.6.0 Homebrew

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

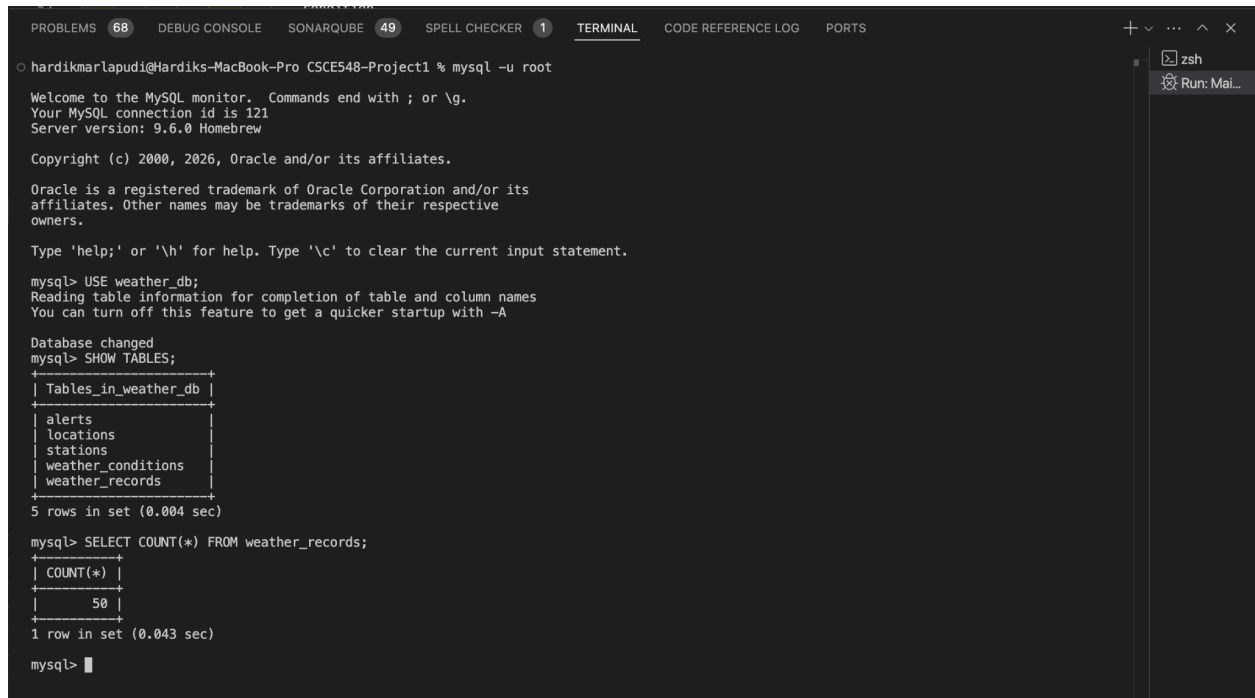
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE weather_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_weather_db |
+-----+
| alerts                |
| locations              |
| stations              |
| weather_conditions    |
| weather_records       |
+-----+
5 rows in set (0.004 sec)

mysql>
```

# 50+ Rows Verification



The screenshot shows a VS Code editor with a terminal window open. The terminal displays the output of a MySQL command prompt session. The user has connected to a MySQL server and is in the 'weather\_db' database. They have run 'SHOW TABLES;' which lists five tables: alerts, locations, stations, weather\_conditions, and weather\_records. Then, they have run 'SELECT COUNT(\*) FROM weather\_records;' which returns a single row with the value 50.

```
hardikmarlapudi@Hardiks-MacBook-Pro CSCE548-Project1 % mysql -u root

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 121
Server version: 9.6.0 Homebrew

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

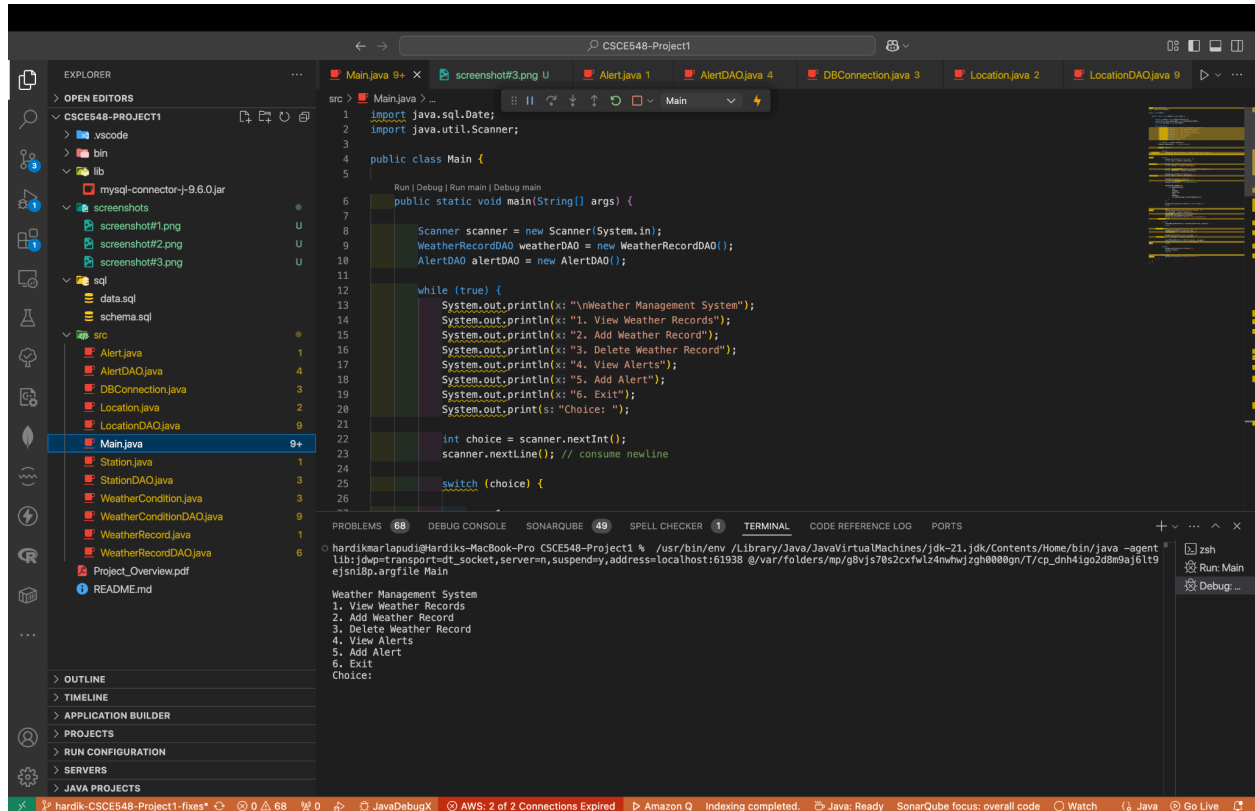
mysql> USE weather_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_weather_db |
+-----+
| alerts                |
| locations              |
| stations              |
| weather_conditions    |
| weather_records       |
+-----+
5 rows in set (0.004 sec)

mysql> SELECT COUNT(*) FROM weather_records;
+-----+
| COUNT(*) |
+-----+
| 50       |
+-----+
1 row in set (0.043 sec)

mysql>
```

# Console Application Running



# Data Retrieved from Database

The screenshot displays an IDE interface for a project named "CSCE548-Project1". The Explorer panel on the left shows the project structure, including a "src" directory with files like "Alert.java", "AlertDAO.java", "DBConnection.java", "Location.java", "LocationDAO.java", "Main.java", "Station.java", "StationDAO.java", "WeatherCondition.java", "WeatherConditionDAO.java", "WeatherRecord.java", and "WeatherRecordDAO.java". The main editor area shows the "data.sql" file, which contains SQL queries for inserting weather data and alerts. The Terminal panel at the bottom displays the output of the application, showing a list of weather records and alerts retrieved from the database.

```
sql > data.sql
81 ('New York','Central Park Station','Rainy',39.1,84,'2025-01-21'),
82 ('New York','Central Park Station','Cloudy',42.3,63,'2025-01-22'),
83 ('New York','Central Park Station','Sunny',45.0,51,'2025-01-23'),
84 ('New York','Times Square Station','Rainy',40.2,85,'2025-01-24'),
85 ('New York','Times Square Station','Cloudy',43.1,62,'2025-01-25');
86
87 -- =====
88 -- ALERTS
89 -- =====
90
91 INSERT INTO alerts (location_name, message) VALUES
92 ('Columbia', 'Heavy rain expected'),
93 ('New York', 'Cold weather advisory');
```

PROBLEMS (27) DEBUG CONSOLE SONARQUBE (8) SPELL CHECKER (1) TERMINAL CODE REFERENCE LOG PORTS

Line	City	Station	Condition	Temp	Humidity	Date
45	New York	Times Square Station	Sunny	44.2	52	2025-01-20
46	New York	Central Park Station	Rainy	39.1	84	2025-01-21
47	New York	Central Park Station	Cloudy	42.3	63	2025-01-22
48	New York	Central Park Station	Sunny	45.0	51	2025-01-23
49	New York	Times Square Station	Rainy	40.2	85	2025-01-24
50	New York	Times Square Station	Cloudy	43.1	62	2025-01-25

Weather Management System

1. View Weather Records
2. Add Weather Record
3. Delete Weather Record
4. View Alerts
5. Add Alert
6. Exit

Choice: 4

Location	Message
Columbia	Heavy rain expected
New York	Cold weather advisory

Weather Management System

1. View Weather Records
2. Add Weather Record
3. Delete Weather Record
4. View Alerts
5. Add Alert
6. Exit

Choice: []

## **Write the Overview Document Section**

This project is based on modeling a Weather Management System, which stores and manages weather details based on location and station. The system monitors daily weather conditions such as temperature and humidity levels, and it is capable of generating location-based weather alerts. The application is based on the integration of a database with a Java-based console interface.

## **AI Prompt Used**

ChatGPT was employed for creating the initial SQL schema, test data, and Java Data Access Object (DAO) code. The prompts were centered on creating a relational database with various tables, establishing foreign key relationships between them, and creating CRUD operations for the project. The result obtained from using ChatGPT was a starting point, which was later verified and corrected using manual debugging and testing.



## Changes I Made Manually

Example:

- Removed unnecessary join tables generated by AI
- Replaced users table with stations to better fit the problem
- Fixed SQL column mismatches causing runtime errors
- Corrected DAO queries to match final schema
- Ensured 50 sequential weather records
- Debugged MySQL connection and missing-table errors

## **AI Effectiveness Analysis**

ChatGPT helped hasten the development process, but it created overly complex schemas and used columns inconsistently. Debugging and refactoring were necessary to make it correct and adhere to foreign keys.