Question 1: Data Modeling: I choose to use MySql

DDL = CREATE TABLE IF NOT EXISTS weather\_data (

id INT NOT NULL AUTO\_INCREMENT,

station\_id VARCHAR(255) NOT NULL,

date DATE NOT NULL,

Max\_Temp FLOAT,

Min\_Temp FLOAT,

Precipitation FLOAT,

PRIMARY KEY (id)

)

Extra Credit: Deployment:

To deploy an API, database, and scheduled data ingestion code on AWS, here is an approach that could be used:

1. Choose an appropriate AWS service for each component:

* API: Amazon API Gateway
* Database: Amazon Relational Database Service (RDS)
* Scheduled data ingestion code: AWS Lambda

1. Set up the API Gateway:

* Create an API Gateway REST API.
* Define the endpoints and methods for the API.
* Set up authentication and authorization for the API using AWS Identity and Access Management (IAM) and AWS Cognito, if necessary.

1. Set up the RDS database:

* Choose an appropriate database engine (e.g., MySQL, PostgreSQL, etc.) and create a new RDS instance.
* Configure the database instance settings and create the necessary database schema and tables.
* Set up the necessary security groups and IAM roles to control access to the database.

1. Set up the AWS Lambda function:

* Create a new Lambda function to handle the scheduled data ingestion code.
* Define the function's input and output parameters and specify the required runtime environment (e.g., Python, Node.js, etc.).
* Add any necessary dependencies or libraries to the function code.
* Set up the necessary IAM roles and security groups to control access to the Lambda function.

1. Connect the components:

* Configure the API Gateway to route requests to the appropriate Lambda function and return the response to the client.
* Modify the Lambda function code to interact with the RDS database as necessary.
* Use AWS CloudWatch to monitor the API Gateway and Lambda function for errors or performance issues.

By following this approach, I can deploy an API, database, and scheduled data ingestion code on AWS using a combination of API Gateway, RDS, and Lambda. This allows for scalability and easy management of the infrastructure.