**Documentation: Inventory Adjustment System**

**Overview**

This Python script integrates with a MySQL database to manage coffee supply chain data, adjust inventory levels based on sentiment and risk analysis results, and update the database with the changes. It uses the mysql.connector library for database operations and pandas for handling data from a CSV file.

**Code Features**

**1. Database Connection**

The script connects to a MySQL database using the mysql.connector library. The credentials and database name are customizable.

**Connection Details**

* **Host**: 127.0.0.1
* **User**: root
* **Password**: Omkar@003
* **Database**: inventory\_management

**2. Table Creation**

Three tables are created in the database to manage inventory and track risks:

**Tables**

1. **inventory**
   * Stores stock levels and thresholds for different countries.
   * **Columns**:
     + country\_name: Name of the country (Primary Key).
     + stock\_level: Current stock level.
     + threshold: Minimum stock level to maintain.
2. **risk\_data**
   * Stores risk analysis results.
   * **Columns**:
     + id: Auto-incrementing primary key.
     + title: News article title.
     + risk\_level: Risk level (High, Medium, Low).
     + stock\_adjustment: Adjustment value based on risk.
     + Published At: Date of the risk publication.
3. **adjusted\_inventory**
   * Tracks adjusted stock levels based on risk analysis.
   * **Columns**:
     + country\_name: Name of the country (Primary Key).
     + stock\_level: Original stock level.
     + stock\_adjusted: Adjusted stock level.
     + adjustment: Adjustment value.
     + Published At: Date of the adjustment.

**3. Mock Data Insertion**

The script inserts mock data into the inventory table for demonstration purposes. This data includes countries, their stock levels, and thresholds.

**4. Risk Data Handling**

The script loads sentiment and risk analysis results from a CSV file (sentiment\_and\_risk\_analysis\_results.csv). The Published At column is parsed as a date without renaming it.

**CSV Columns**

* Title: Title of the news article.
* Risk Level: Risk level (High, Medium, Low).
* Published At: Date of publication.

**5. Stock Adjustment Logic**

The script adjusts stock levels based on the risk level and updates the adjusted\_inventory table:

* **High Risk**: Decrease stock by 20%.
* **Medium Risk**: No change.
* **Low Risk**: Increase stock by 5%.

**Adjustment Logic**

1. Retrieve the current stock level for the country.
2. Calculate the adjustment based on the risk level.
3. Update the adjusted\_inventory table if:
   * A record exists and the new Published At date is later.
   * A record does not exist for the country.

**6. Database Updates**

The script:

1. Stores adjusted inventory data in the adjusted\_inventory table.
2. Inserts risk data into the risk\_data table.

**7. Fetching Adjusted Inventory**

After processing all risk data, the script fetches and displays the updated adjusted\_inventory table as a DataFrame for review.

**8. Closing the Database Connection**

The script ensures the database connection is closed at the end.

**How to Use the Script**

**Prerequisites**

1. Install the required Python libraries:

bash

CopyEdit

pip install mysql-connector-python pandas

1. Set up a MySQL database and update the connection credentials (host, user, password, database) in the script.
2. Ensure the sentiment\_and\_risk\_analysis\_results.csv file exists in the same directory as the script.

**Execution Steps**

1. Run the script to:
   * Create the necessary tables.
   * Insert mock data into the inventory table.
   * Load risk data from the CSV file.
   * Adjust stock levels based on risk analysis.
   * Update the database with adjusted stock levels and risk data.
2. Review the adjusted inventory printed as a DataFrame.