**A logo with a lightning bolt

Description automatically generatedQuality Control Project – Mystic Manuscripts**

Welcome to Data Dynamo Consultants, where we turn data chaos into data clarity! We're excited to have you on board for a challenging and rewarding project that will hone your skills in data management and quality control. This project is more than just data cleaning; it's about understanding the importance of accuracy and reliability in data. Quality data is the cornerstone of trustworthy analysis, essential for our clients to make informed decisions.

## **Project Overview**

You've been assigned to a crucial project for one of our esteemed clients, "**Mystic Manuscripts**". They have entrusted us with a significant dataset, crucial for their upcoming analysis. Your mission is to ensure the integrity and quality of this data before it's used for any further analysis.

## **Your Mission**

**Dataset Upload**  
Import the dataset from Mystic Manuscripts into our SQL database. Treat this step with utmost care, as it sets the foundation for your project.

**Error Identification and Quality Control**  
Delve into the dataset and identify the errors. These could range from incorrect entries, missing information, duplicates, to more subtle data inconsistencies.

**Client Interaction**  
You can meet with the client – Mystic Manuscripts. During this meeting, you can verify anomalies, gather additional data insights, and clarify any challenges you’re having.

**Documentation and Quality Control Form:**   
Document each step of your error identification and correction process. Conclude your project by completing a Quality Control Form based on the provided template, which certifies the dataset's readiness for further use.

## **Deliverables**

**The following deliverables are expected:**

1. SQL code showcasing table creation, as well as any updated/alterations. (As a .sql or .txt file)
2. SQL code showcasing quality control checks (As a .sql or .txt file) - this excludes queries 1 - 7 that are required separately below
3. Quality control form referencing the checks that have been carried out. (As a .docx or .pdf file)
4. The completed SQL file with your queries - renamed to your team's name, downloadable from below with the comments intact (see the instructions below)
5. The completed Environment Variables file, downloadable from below with the values for TABLE\_NAME, SCHEMA\_NAME and TEAM\_NAME included and - .your\_team\_name appended to the file's name.

### **Requirements for 4 and 5**

1. Make sure that you ***GRANT*** the ***SELECT*** permission to the ***github\_classroom*** role on your table from the ***mystic\_manuscript*** schema
2. **Do not remove ANY** of the comments in the queries.sql file and ensure that you place your queries in the correct location.
   1. Additional comments can be added to the file, but the existing comments should not be removed or modified.
3. You should ONLY edit the queries.sql file and the .env file
   1. .env needs the appropriate values set for SCHEMA\_NAME, TABLE\_NAME and TEAM\_NAME

#### **Queries for the Mystic Manuscript Database**

There are many queries that will return the correct output - it is the output that is tested, not the way in which you arrive at it.

#### **Best Practices**

Queries WILL be linted as part of the grading process. The auto-grading process uses [SQLFluff](https://sqlfluff.com/).

A list of the rules can be found [here](https://docs.sqlfluff.com/en/stable/reference/rules.html)

If you fail the linting test (see below for how to run linting), you should look at the output and work out why.

The following rules have been disabled:

[AM04](https://docs.sqlfluff.com/en/stable/reference/rules.html#rule-AM04) - Query produces an unknown number of result columns

[LT09](https://docs.sqlfluff.com/en/stable/reference/rules.html#rule-LT09) - Select targets should be on a new line unless there is only one select target

All SQL keywords should be in ***UPPERCASE***, contrary to the default SQLFluff settings for postgres. Binary operators that are used on more than 1 line should be the last character of the line, contrary to the default SQLFluff settings for postgres.

#### **Query 1**

* This should be placed in the queries.sql file under the comment -- query1
* The output of the query should be a single ***integer*** value (1 column, 1 row)

#### **Query 2**

* This should be placed in the queries.sql file under the comment -- query2
* The output of the query should be a single ***integer*** value (1 column, 1 row)

#### **Query 3**

* This should be placed in the queries.sql file under the comment -- query3
* The output of the query should be a single ***integer*** value (1 column, 1 row)

***HINT:*** You will need to look into the INFORMATION\_SCHEMA to find the columns in the table

#### **Query 4**

* This should be placed in the queries.sql file under the comment -- query4
* The output of the query should be a single ***float*** value rounded to 2 decimal places (1 column, 1 row)

#### **Query 5**

* This should be placed in the queries.sql file under the comment -- query5
  + The output of the query should be a single ***float*** value rounded to 2 decimal places (1 column, 1 row)

***HINT:*** The result of this query should be the same as that obtained from Query 4; the method to get there should be different!

#### **Query 6**

* This should be placed in the queries.sql file under the comment -- query6
* The output of the query should be *5 random dates* extracted from your table that match the *required date format* of your source data set (1 column, 5 rows)

#### **Query 7**

* This should be placed in the queries.sql file under the comment -- query7
* The output of the query should be *5 random records* from your table (all columns, 5 rows)

### **Submission**

When ready submit the SQL file, renamed to your\_team\_name.sql and the ENV file renamed to .env.your\_team\_name to the assignment on Noodle with the other required files.

Make sure that you have the correct values in the .env file for the SCHEMA\_NAME, TABLE\_NAME and TEAM\_NAME variables using snake\_case with ***NO QUOTES***.

SCHEMA\_NAME=your\_schema\_name  
TABLE\_NAME=your\_table\_name  
TEAM\_NAME=your\_team\_name

### **Checking the Coding Standings**

You can run and install SQLFluff locally to check your queries - see the documentation [here](https://docs.sqlfluff.com/en/stable/gettingstarted.html)

You should create a file called .sqlfluff in the same folder as your SQL file:

[sqlfluff]  
exclude\_rules = AM04, LT09  
  
[sqlfluff:layout:type:binary\_operator]  
line\_position = trailing  
  
[sqlfluff:rules:capitalisation.keywords]  
capitalisation\_policy = upper

Following this, in the terminal, run the command:

sqlfluff lint your\_file\_name.sql --dialect postgres --config .sqlfluff

Fix any issues that are raised by the linting process before submitting the .sql file.