ETL 2 part 3: loading to mysql

Table of Contents

[Overview 2](#_Toc183193473)

[Diagram 2](#_Toc183193474)

[1. Create the database in MySQL where the data will be stored 2](#_Toc183193475)

[2. Create the table in the database where the data will be loaded 3](#_Toc183193476)

[3. Create the access for the user to load data 4](#_Toc183193477)

[4. Establish the connection between Pyspark and the SQL database 5](#_Toc183193478)

[5. Load the data from Pyspark to MySQL 8](#_Toc183193479)

# Overview

The CRM data has been transformed to meet business requirements. Now it is time to load the data to where the business can access it for visualizations and reporting. The transformed data will be loaded into a MySQL database where the business has access.

# Diagram

A computer and a cloud of information

Description automatically generated with medium confidence

# 1. Create the database in MySQL where the data will be stored

|  |
| --- |
| Description: The data needs a place to be loaded, so the database needs to be built. For this case, a database will be built called ‘crm’. \*Setup of MySQL won’t be demonstrated here. If that is of interest, then refer to the ETL 1 Guide. |
| 1. Open up MySQL Workbench.    2. Create the database and call it ‘crm’ by running the following SQL command. Confirm by refreshing the Schema pane. |

# 2. Create the table in the database where the data will be loaded

|  |
| --- |
| Description: With the database built, a table needs to be established for the data to be loaded into. This table will be called ‘hardware’. |
| 1. Create the table by running the following SQL command. \*Note: Make sure the data types match up to the ones of the data being loaded. Ex. The regional office values in the data are string or characters, so therefore the data type for the column in SQL must also be set to ‘char’ or character. Confirm by refreshing the Schema pane again. |

# 3. Create the access for the user to load data

|  |
| --- |
| Description: The data analyst who transformed the data will need to be registered to access to the ‘crm’ database, so the data loading can go through. |
| 1. Log into MySQL in the command prompt. Run the following command:    2. Create the user by running the following commands:    \*Note: After the ‘@’ the IP where the data is being loaded should be used here. After IDENTIFIED BY within the ‘’, a password of choice needs to be set. |

# 4. Establish the connection between Pyspark and the SQL database

|  |
| --- |
| Description: With the Pyspark user having access to the database, the endpoint connection needs to be made. This can be done |
| 1. Download MySQL Connector for Java on the machine where Pyspark is being ran. \*Note: Check compatibility of the download and the Spark version. Considering Spark 3.5.3 is being used, MySQL Connector 8.0.x will be used for compatibility.  Open up a web browser and search for the download version. Click the Archives tab and select an 8.0.x version with the OS being Platform Independent.    2. Once downloaded and extracted, move the MySQL Connector jar file to the jars file within the Spark folder.    3. It is important to confirm the driver can be referenced by Spark. Verify by performing the following:  Open a prompt in Windows run: spark-shell    Copy/paste the link for the Spark Web UI into a web browser      Click the Environment tab, scroll down to Classpath Entries and search for the mysql-connector jar file to confirm    Open another Jupyter Notebook and run the following code to test the driver |

# 5. Load the data from Pyspark to MySQL

|  |
| --- |
| Description: After verifying the driver works, the data can be loaded into the target database on MySQL. |
| 1. Run the following code to load the data to the table within the database:    2. Verify the data made it into the database    The data has made it to the database where business users can now consume it for visualizations! |