In this assignment, you will practice performing ETL on a database that contains information about some movies by constructing NiFi pipelines that act first on an Excel file and then on a MySQL database. In the first part of the assignment, you will be challenged to construct a NiFi pipeline to move your data from an Excel file to a .csv file. In the second part of the assignment, you will be required to move the data from a .csv file to a MySQL database using a NiFi pipeline.

Before starting this assignment, review the submission instructions below to ensure that you collect the required screenshots as you progress through the assignment.

Note that this assignment has been tested using a Windows OS and the Catalina version of a Mac OS. If you use the Big Sur OS, you are recommended to use the myPhpAdmin *container* as demonstrated in this article: [Run MySQL & phpMyAdmin Locally Using Docker.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

[Links to an external site.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

**References**

[Maxwell Harper, F., and Joseph A. Konstan. “The MovieLens Datasets: History and Context.” *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, no. 4 (2015): 19:1–19:19. https://doi.org/10.1145/2827872.](https://doi.org/10.1145/2827872)

[Links to an external site.](https://doi.org/10.1145/2827872)

[MovieLens. “Summary.” *GroupLens*. 2021. https://files.grouplens.org/datasets/movielens/ml-latest-README.html.](https://files.grouplens.org/datasets/movielens/ml-latest-README.html)

[Links to an external site.](https://files.grouplens.org/datasets/movielens/ml-latest-README.html)

[Yuste, Miguel. “Run MySQL & PhpMyAdmin Locally in 3 Steps Using Docker.” *Medium*. 2019. https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

[Links to an external site.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

**To complete this assignment, follow these steps:**

Before you begin the steps of the assignment below, please be sure you have your two *containers* running within Docker and that they are connected to the same network titled netassignment; one *container* is for the NiFi *server* and one is for MySQL. Name the NiFi *container* nifi and the MySQL *container* mysql.

The *driver* must also be configured as demonstrated in [Video 17.6](https://classroom.emeritus.org/courses/10605/pages/creating-an-etl-pipeline-19-10).

**Part 1: Writing Data to an Excel File**

For the first part of this assignment, you will use the [movies.xlsx](https://classroom.emeritus.org/courses/10605/files/3007349/download) file.

1. Open the CLI for the NiFi *container.* In the *container* bash window, navigate inside the /opt/nifi/nifi-current folder and create two folders within it: input and output. Provide a screenshot to show that you successfully created the two folders.
2. Use the Docker copy command below to copy the [movies.xlsx](https://classroom.emeritus.org/courses/10605/files/3007349/download) file from your local machine in the input folder inside of your NiFi Docker *container*:  
   docker cp ./movies.xlsx nifi:/opt/nifi/nifi-current/input  
   Provide a screenshot to show that you successfully copied the movies.xlsx file in the input folder.
3. Open the NiFi UI in your browser. Create a new process group and name it Assignment17. Provide a screenshot to show that you successfully created the Assignment17 process group.
4. Add a GetFile *processor*. Provide a screenshot to show that you successfully added the GetFile *processor* to the NiFi canvas. In the SCHEDULING tab, set the Run Schedule field equal to 15 seconds. In the PROPERTIES tab, set the Input Directory field equal to /opt/nifi/nifi-current/input, and set the File Filter field equal to the Excel file name to be processed, in this case, movies.xlsx. Provide a screenshot to show that you correctly configured the *properties* for the GetFile *processor*.
5. Add a *processor* titled ConvertExcelToCSVProcessor. In the SCHEDULING tab, set the Run Schedule field equal to 15 seconds. In the PROPERTIES tab, set the Sheets to Extract field equal to Sheet 1 - movies. Provide a screenshot to show that you configured the *properties* for the ConvertExcelToCSVProcessor *processor* correctly.
6. Add a PutFile *processor*. In the SETTINGS tab, select success to Automatically Terminate Relationships. In the SCHEDULING tab, set the Run Schedule field equal to 15 seconds. In the PROPERTIES tab, set the Directory field equal to /opt/nifi/nifi-current/output. Provide a screenshot to show that you correctly configured the *properties* for the PutFile *processor*.
7. Connect the GetFile *processor* to the ConvertExcelToCSVProcessor *processor*. Select success for the relationship. Connect the ConvertExcelToCSVProcessor *processor* to the PutFile *processor*. Select failure, original, and success for the relationships. Provide a screenshot to show that you successfully connected all the *processors* with the correct relationships.
8. Start the GetFile, ConvertExcelToCSVProcessor*,* and PutFile *processors*. Provide a screenshot to show that all the *processors* are running (as indicated by a green arrow).
9. Navigate to the /opt/nifi/nifi-current/output folder in the NiFi CLI, and list the files. Verify that the *processor* has created a CSV file for processing. Provide a screenshot to show that the movies\_Assignment.csv file has been created.

**Part 2: Writing Data to an SQL Database**

For the second part of this assignment, you will use the [movies.csv](https://classroom.emeritus.org/courses/10605/files/3007104/download) file.

1. Using MySQL Workbench, create a new database called movielens and generate the movies table with the following three column fields:

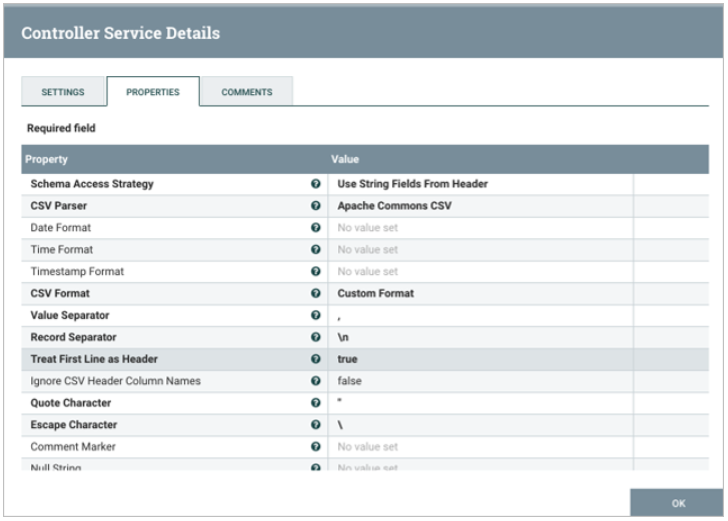
| create TABLE movies(  idx int,  title varchar(100),  genres varchar(100) ); |
| --- |

Confirm that you have an empty table. Perform the following *function*:

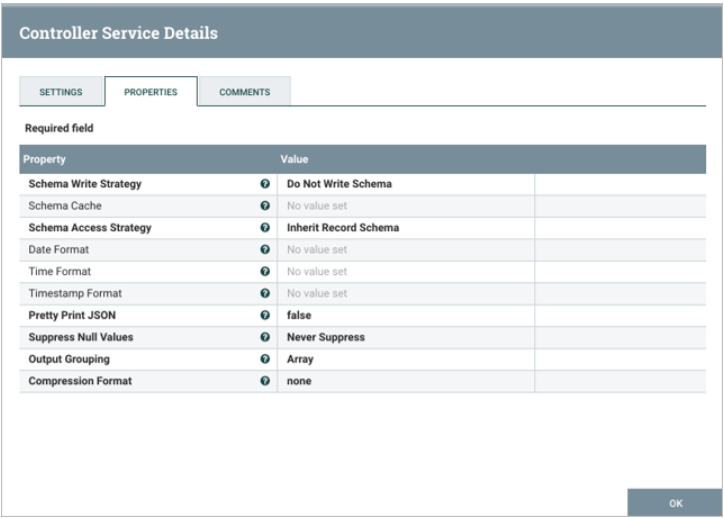
SELECT \* FROM movies

Provide a screenshot of your MySQL Workbench to show that you have successfully initialized an empty movies table in the movielens database.

1. Download the [movies.csv](https://classroom.emeritus.org/courses/10605/files/3007104/download) CSV file and place it on your NiFi *server* in a newly created *directory* with the path /opt/nifi/nifi-current/data.  
   To do this, open the NiFi CLI in the Terminal through your Docker window and navigate to the following path: opt/nifi/*.*Create a nifi-current *directory* and create a data *subdirectory* within it. Open up a second Terminal window on your local machine and navigate to the folder where you downloaded the movies.csv file. Perform the following command:  
   docker cp ./movies.csv nifi:/opt/nifi/nifi-current/data  
   Navigate back to the NiFi CLI window. Provide a screenshot to show that the movies.csv file is now on the NiFi *server*.
2. Open NiFi in your browser. The MySQL *controller* service will be set up in the exact same way as shown in [Video 17.7](https://classroom.emeritus.org/courses/10605/pages/creating-an-etl-pipeline-19-10). Additionally, the *driver* must be configured as demonstrated in [Video 17.6](https://classroom.emeritus.org/courses/10605/pages/creating-an-etl-pipeline-19-10). Provide a screenshot to show that you successfully opened the NiFi UI.
3. Create a *controller* service called MySQL. The MySQL *controller* service will be set up in the exact same way as in Video 17.7. Provide a screenshot to show that you successfully created and enabled the MySQL *controller* service.
4. Create *reader* and *writer processors*. The *reader* will read the data from your FlowFiles and the *writer* will write the data to MySQL commands.
   1. To create a *reader*, go to the *controller* services, select the “+” option, and search for “CSVReader”. Open up the configurations and confirm that your screen looks similar to the following:



b. To create a *writer*, select the “+” option, and search for “JsonRecordSetWriter”. Open up the configurations and confirm that your screen looks similar to the following:



Select the lightning bolt symbol to start up each *controller* and select “Service Only” under the service and reference components menu.  
Provide a screenshot of the *controller* screen to show that the three *controller* services (*reader*, *writer*, and MySQL) are enabled.

6. Now it is time to set up the data pipeline. This will consist of five *processors* flowing in the order specified below:

1. A GetFile *processor* (titled get movies file): In the PROPERTIES tab, set the input *directory* equal to /opt/nifi/nifi-current/data.
2. A SplitText *processor*: In the SETTINGS tab, select failure and original to Automatically Terminate Relationships. In the PROPERTIES tab, set the Line Split Count and Header Line Count fields both equal to 1.
3. A ConvertRecord *processor*: In the PROPERTIES tab, set the Record Reader field equal to CSVRead and the Record Writer field equal to JsonRecordWriter.
4. A ConvertJSONToSQL *processor*: In the SETTINGS tab, select failure and original to Automatically Terminate Relationships. In the PROPERTIES tab, set the JDBC Connection Pool field equal to MySQL, the Statement Type field equal to INSERT, the Table Name field equal to movies, and the Catalog Name field equal to movielens.
5. A PutSQL *processor*: In the SETTINGS tab, select failure and success to Automatically Terminate Relationships. In the PROPERTIES tab, set the JDBC Connection Pool field equal to MySQL.

Provide a screenshot of your complete data pipeline, including all five *processors*: GetFile, SplitText, ConvertRecord, ConvertJSONToSQL, and PutSQL.

7. Connect the *processors*.

1. The *connector* between the GetFile and SplitText *processors* will be a success relationship.
2. The *connector* between the SplitText and ConvertRecord *processors* will be a splits relationship.
3. The *connector* between the ConvertRecord and ConvertJSONToSQL *processors* will be a success relationship.
4. The *connector* between the ConvertJSONToSQL and PutSQL *processors* will be a sql relationship.
5. The final *connector* will stem from the PutSQL *processor* and *loop* back to itself. This will be a retry relationship.

Provide a screenshot of all five *processors* to show that the correct *connectors* have been added between the *processors*.

8. Start each *processor*, beginning with the GetFile *processor*. Watch the data propagate down toward the PutSQL *processor*. Provide a screenshot of your NiFi browser screen to show that all five *processors* are connected and running.

9. Navigate back to your MySQL Workbench and perform the following *query*:  
SELECT \* FROM movies;  
There should now be rows of data which you loaded into the table. Provide a screenshot of the result of this *query* to show that the movies table in the movieslens database is now saturated with data.

**Submission Instructions:**

Your submission for this assignment should be a Word document that includes the following screenshots, each labeled for the step that the screenshot represents:

**Part 1: Writing Data to an Excel File**

1. Provide a screenshot to show that you successfully created the two folders: input and output.
2. Provide a screenshot to show that you successfully copied the movies.xlsx file in the input folder.
3. Provide a screenshot to show that you successfully created the Assignment17 process group.
4. Provide a screenshot to show that you correctly configured the *properties* for the GetFile *processor*.
5. Provide a screenshot to show that you correctly configured the *properties* for the ConvertExcelToCSVProcessor *processor*.
6. Provide a screenshot to show that you correctly configured the *properties* for the PutFile *processor*.
7. Provide a screenshot to show that you successfully connected all the *processors* with the correct relationships.
8. Provide a screenshot to show that all the *processors* are running (as indicated by a green arrow).
9. Provide a screenshot to show that the movies\_Assignment.csv file has been created.

**Part 2: Writing Data to an SQL Database**

1. Provide a screenshot of your MySQL Workbench to show that you have successfully initialized an empty movies table in the movielens database.
2. Provide a screenshot to show that the movies.csv file is now on the NiFi *server*.
3. Provide a screenshot to show that you successfully opened the NiFi UI.
4. Provide a screenshot to show that you successfully created and enabled the MySQL *controller* service.
5. Provide a screenshot of the *controller* screen to show that the three *controller* services (*reader*, *writer*, and MySQL) are enabled.
6. Provide a screenshot of your complete data pipeline, including all five *processors*: GetFile, SplitText, ConvertRecord, ConvertJSONToSQL, and PutSQL.
7. Provide a screenshot of all five *processors* to show that the correct *connectors* have been added between the *processors*.
8. Provide a screenshot of your NiFi browser screen to show all five *processors* connected and running.
9. Provide a screenshot of the result of this *query* to show that the movies table in the movieslens database is now saturated with data.