

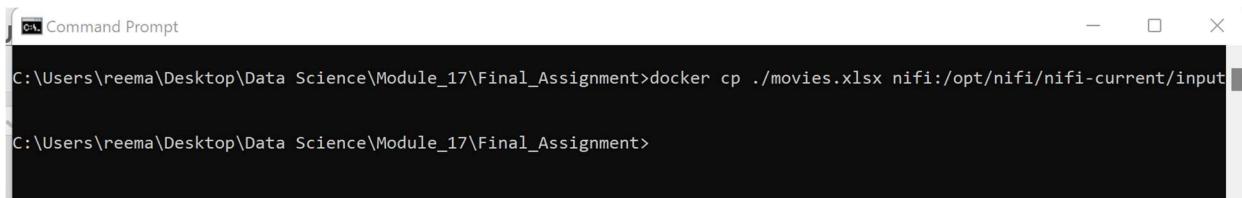
Part 1: Writing Data to an Excel File

1. Deployed NiFi in Docker. Created two folders: input and output in NiFi CLI via Docker GUI



```
$ docker exec -it 9de85627a1b72cc0d3a8bce90b9fb338fc66f63db1716fa0822d4a63ba195a16 /bin/sh
$ pwd
/opt/nifi/nifi-current
$ mkdir input
$ mkdir output
$ pwd
/opt/nifi/nifi-current
$ ls
LICENSE  bin           database_repository  flowfile_repository  logs          run
NOTICE   conf          docs            input             output         state
README   content_repository extensions      lib              provenance_repository work
$
```

2. Copied the movies.xlsx file in the input folder.

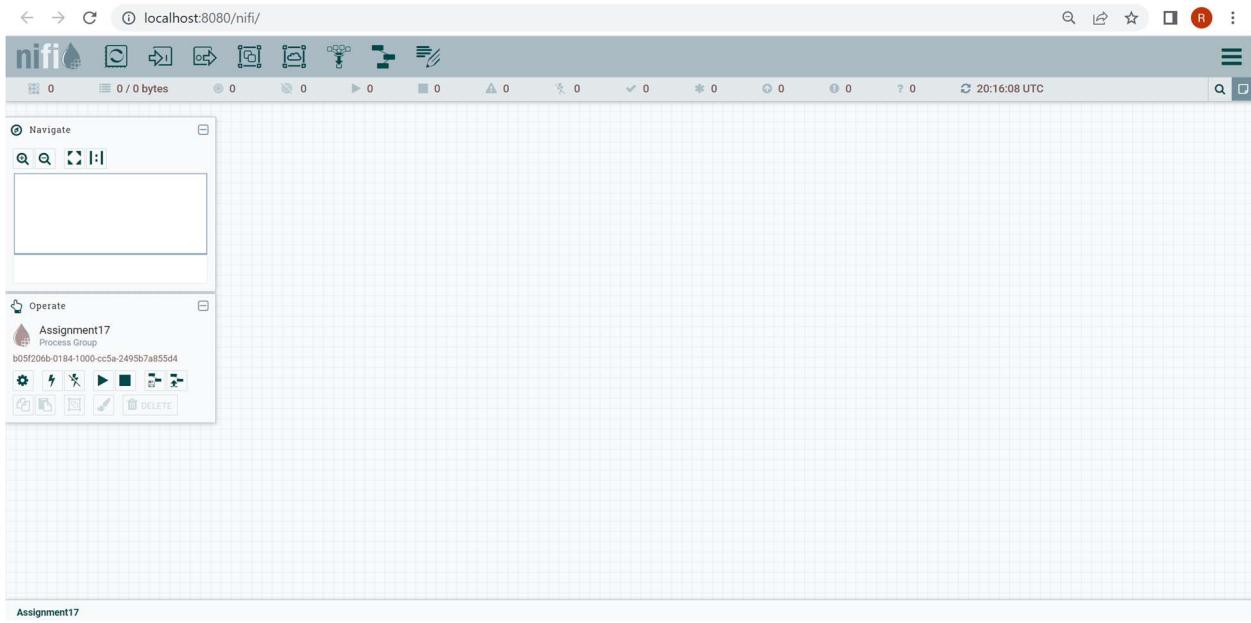


```
C:\Users\reema\Desktop\Data Science\Module_17\Final_Assignment>docker cp ./movies.xlsx nifi:/opt/nifi/nifi-current/input
C:\Users\reema\Desktop\Data Science\Module_17\Final_Assignment>
```

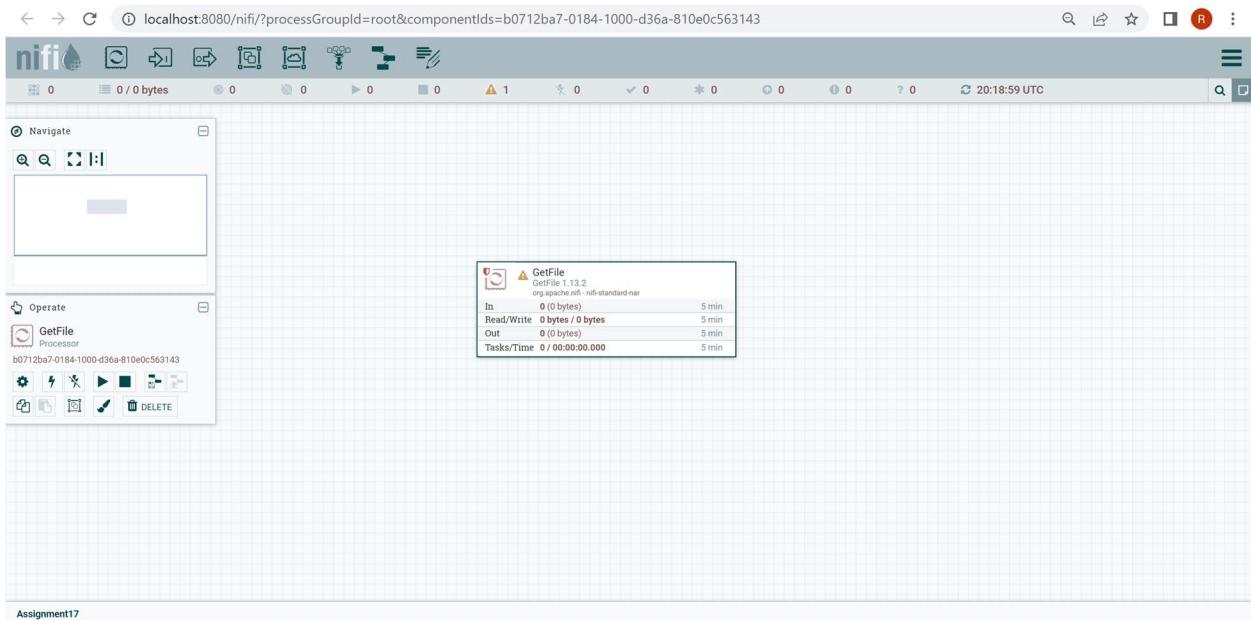


```
$ docker exec -it 9de85627a1b72cc0d3a8bce90b9fb338fc66f63db1716fa0822d4a63ba195a16 /bin/sh
/opt/nifi/nifi-current/input
$ ls
movies.xlsx
$
```

3. Navigated to <http://localhost:8080/nifi/> & created the Assignment17 process group.



4. Configured the *properties* for the GetFile processor.



Configure Processor

⚠ Invalid

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Scheduling Strategy ?	Timer driven		
Concurrent Tasks ?	1	Run Schedule ?	15 sec
Execution ?	All nodes		
CANCEL APPLY			

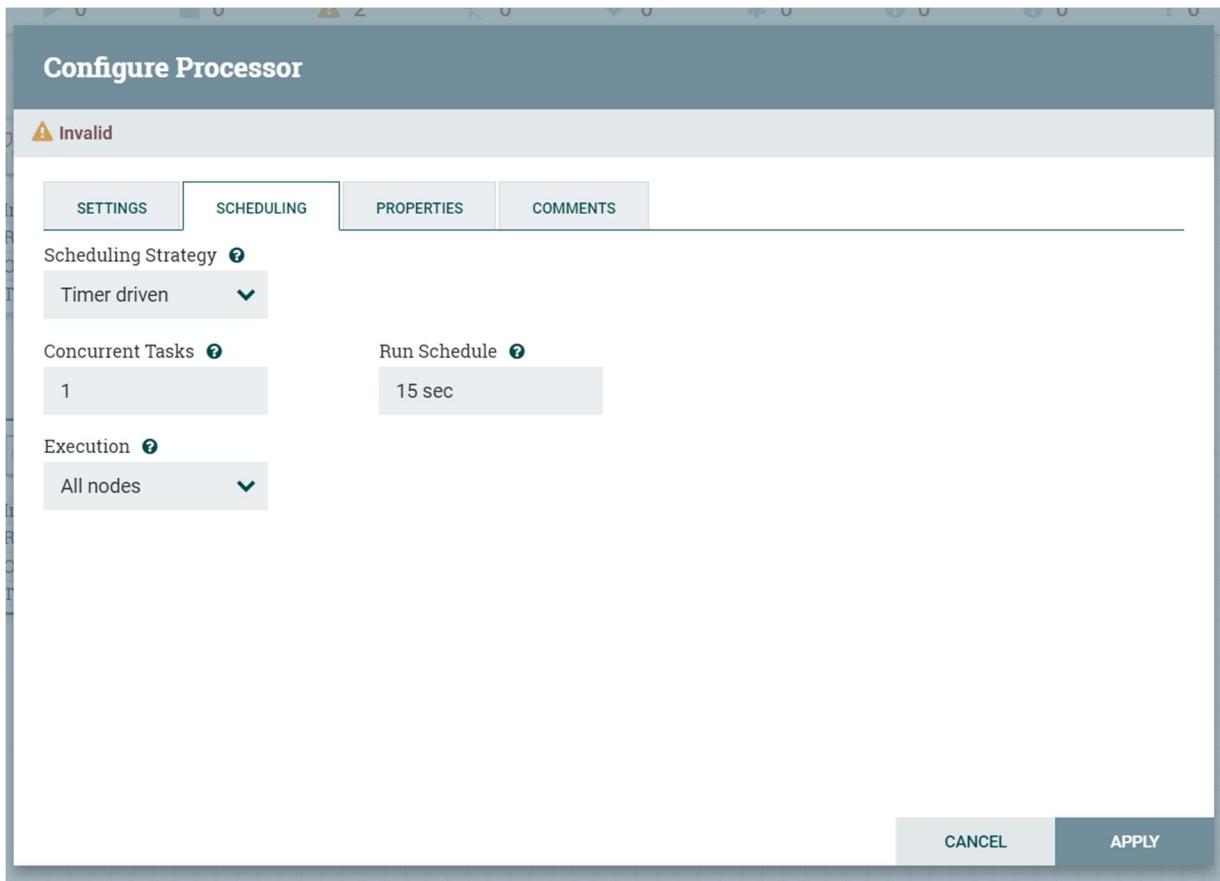
localhost:8080/nifi/processGroupId=root&componentIds=b0712ba7-0184-1000-d36a-810e0c563143

The screenshot shows the Apache NiFi user interface. At the top, there's a navigation bar with icons for back, forward, search, and other system functions. Below it is the main dashboard showing various metrics like flow count, byte count, and processor status. On the left, there's a sidebar with options like 'Operate' and 'GetFile Processor'. The central part of the screen is occupied by the 'Configure Processor' dialog box, which is identical to the one shown above but includes a properties section with the following configuration:

Property	Value
Input Directory	/opt/nifi/nifi-current/input
File Filter	movies.xlsx
Path Filter	No value set
Batch Size	10
Keep Source File	false
Recurse Subdirectories	true
Polling Interval	0 sec
Ignore Hidden Files	true
Minimum File Age	0 sec
Maximum File Age	No value set
Minimum File Size	0 B
Maximum File Size	No value set

At the bottom of the dialog are 'CANCEL' and 'APPLY' buttons.

5. Configured the *properties* for the ConvertExcelToCSVProcessor processor.



Configure Processor

! Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field



Property	Value
Sheets to Extract	Assignment
Number of Rows to Skip	0
Columns To Skip	No value set
Format Cell Values	false
CSV Format	Custom Format
Value Separator	,
Include Header Line	true
Quote Character	"
Escape Character	\
Comment Marker	No value set
Null String	No value set
Trim Fields	true
Output Mode	Do Not Output Value

CANCEL

APPLY

6. Configured the *properties* for the PutFile processor.

Configure Processor

⚠ Invalid

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Name PutFile	<input checked="" type="checkbox"/> Enabled	Automatically Terminate Relationships ?	
Id b1090219-0184-1000-f4e1-5b0b0b901649	<input checked="" type="checkbox"/> failure Files that could not be written to the output directory for some reason are transferred to this relationship		
Type PutFile 1.13.2	<input checked="" type="checkbox"/> success Files that have been successfully written to the output directory are transferred to this relationship		
Bundle org.apache.nifi - nifi-standard-nar			
Penalty Duration ? 30 sec	Yield Duration ? 1 sec		
Bulletin Level ? WARN			
CANCEL APPLY			

Configure Processor

⚠ Invalid

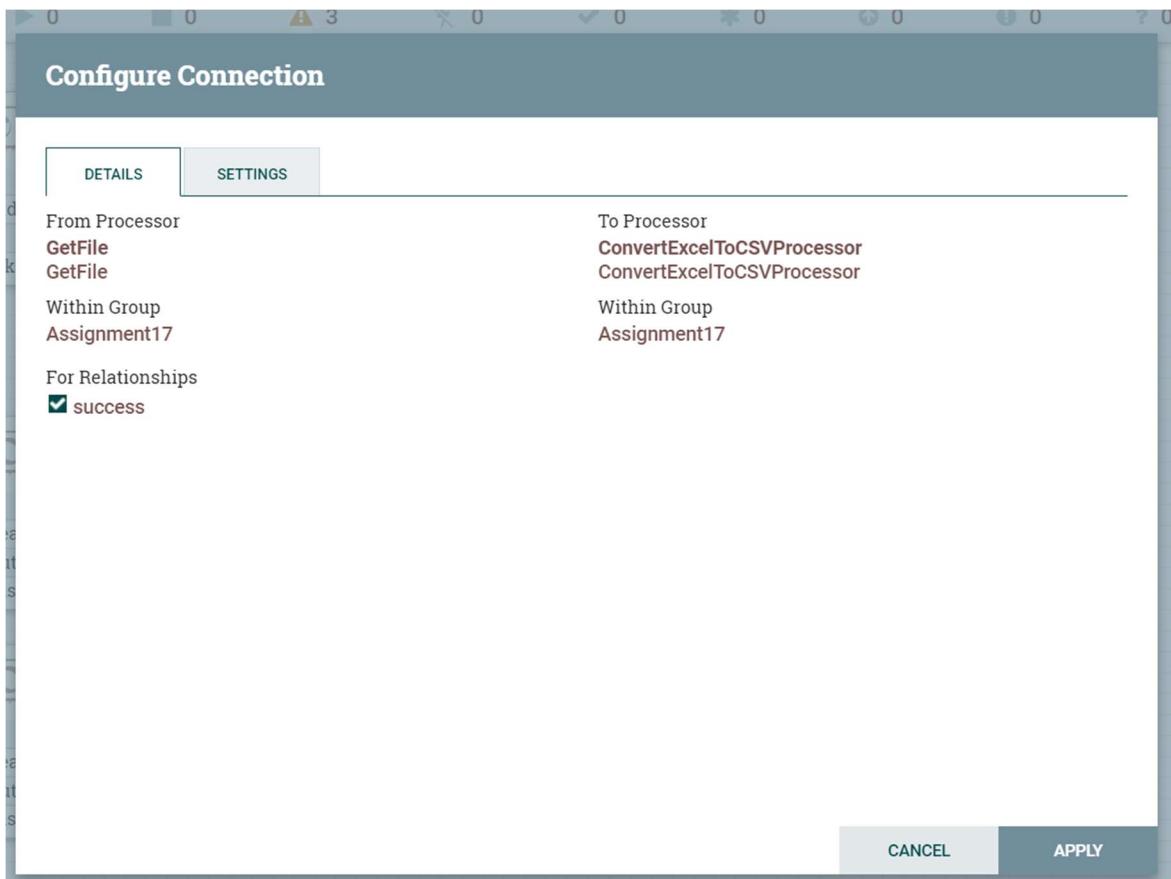
SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Scheduling Strategy ? Timer driven	Run Duration ? 0ms 25ms 50ms 100ms 250ms 500ms 1s 2s Lower latency Higher throughput		
Concurrent Tasks ? 1	Run Schedule ? 15 sec		
Execution ? All nodes			
CANCEL APPLY			

localhost:8080/nifi/?processGroupId=root&componentId=b0efa7b9-0184-1000-cddc-9b5d69fe48e8

Required field	
Property	Value
Directory	/opt/nifi/nifi-current/output
Conflict Resolution Strategy	fail
Create Missing Directories	true
Maximum File Count	No value set
Last Modified Time	No value set
Permissions	No value set
Owner	No value set
Group	No value set

CANCEL
APPLY

7. Successfully connected all the *processors* with the correct relationships.

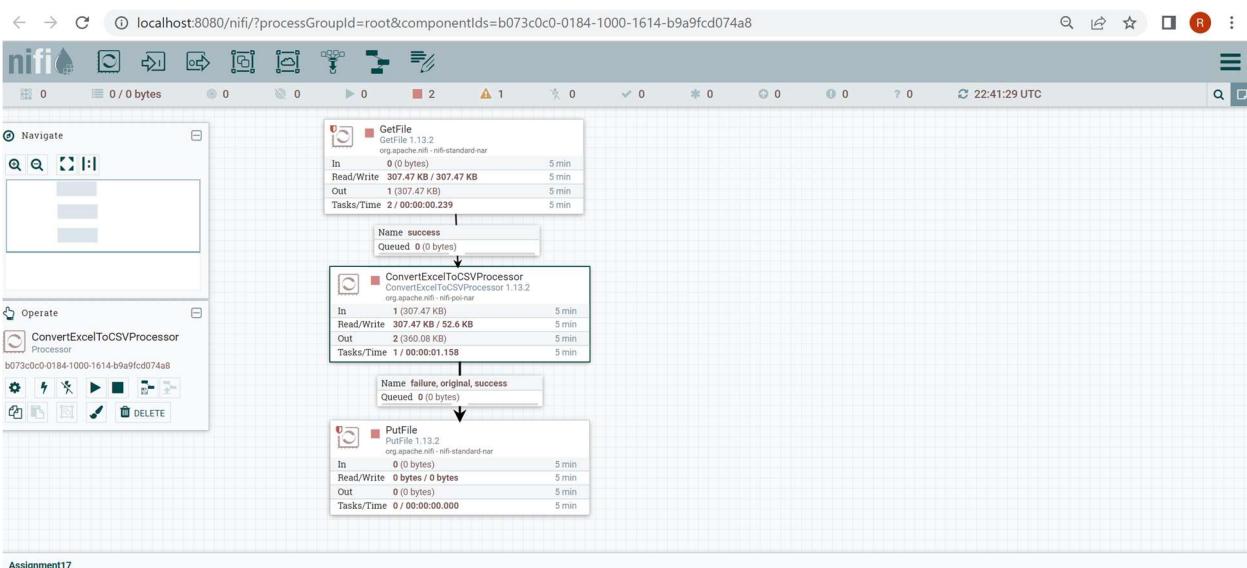


Create Connection

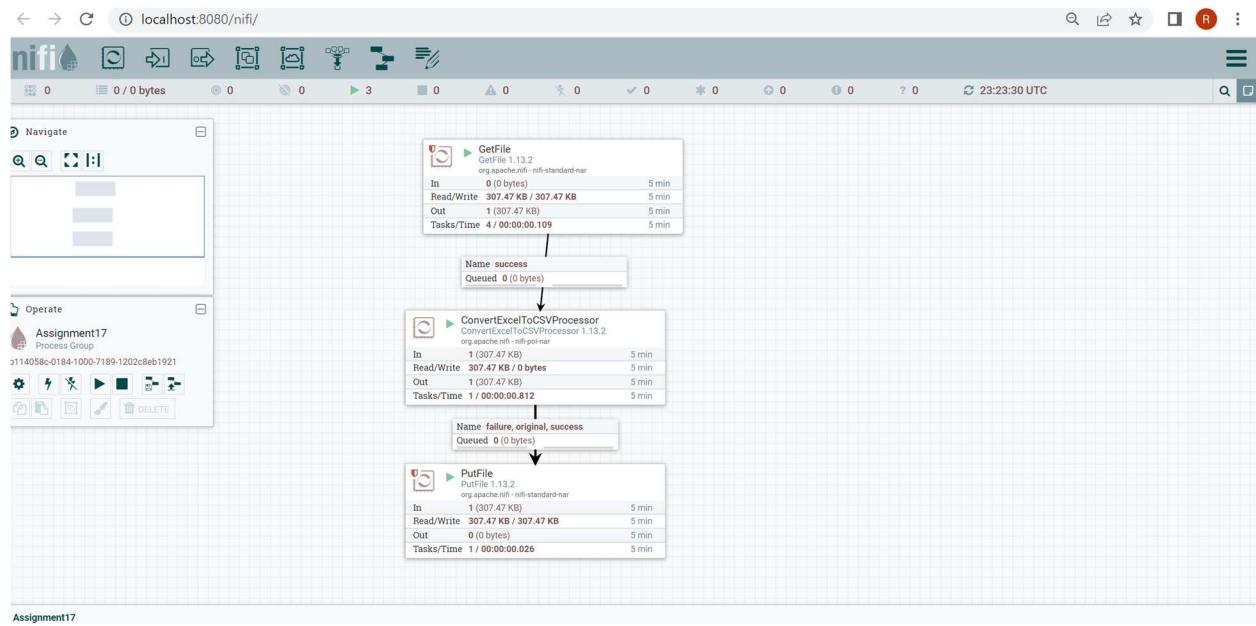
DETAILS **SETTINGS**

From Processor ConvertExcelToCSVProcessor ConvertExcelToCSVProcessor	To Processor PutFile PutFile
Within Group Assignment17	Within Group Assignment17
For Relationships <input checked="" type="checkbox"/> failure <input checked="" type="checkbox"/> original <input checked="" type="checkbox"/> success	

CANCEL **ADD**



8. Below screenshot shows that all the *processors* are running (as indicated by a green arrow).



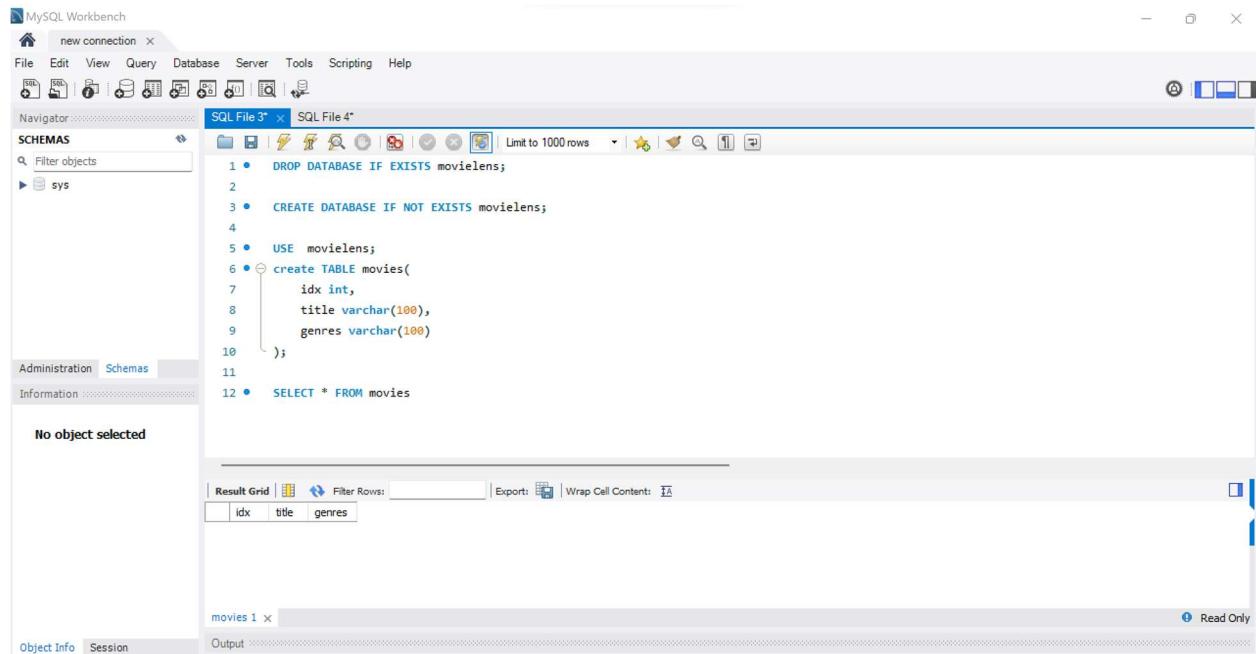
9. Below screenshot shows that the movies_Assignment.csv file has been created.

The screenshot shows a terminal window with the command `docker exec -it b42b0a2d3fc435a31abb0f75855c32921a0aa0a495232be223c5c9411038a28 /bin/sh` running. The user then runs `$ pwd` to show the directory `/opt/nifi/nifi-current/output`, followed by `$ ls` which lists the files `movies.xlsx` and `movies_Assignment.csv`.

```
$ pwd
/opt/nifi/nifi-current/output
$ ls
movies.xlsx  movies_Assignment.csv
$
```

Part 2: Writing Data to an SQL Database

1. Initialized an empty movies table in the movielens database on MySQL Workbench.

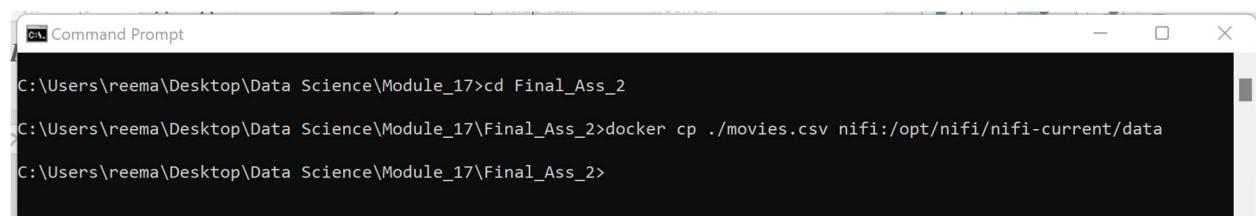


The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. Below the menu is a toolbar with various icons. The main area has two tabs: 'SQL File 3' and 'SQL File 4'. The code in 'SQL File 4' is as follows:

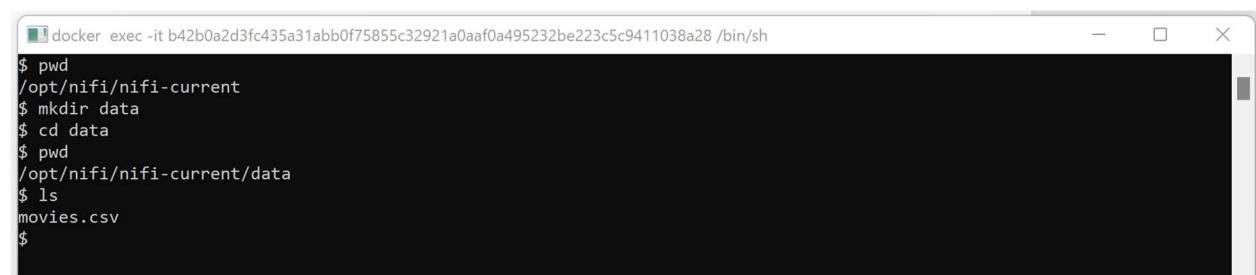
```
1 •  DROP DATABASE IF EXISTS movielens;
2
3 •  CREATE DATABASE IF NOT EXISTS movielens;
4
5 •  USE movielens;
6 •  create TABLE movies(
7     idx int,
8     title varchar(100),
9     genres varchar(100)
10   );
11
12 •  SELECT * FROM movies
```

The 'Result Grid' pane below shows a table structure with columns 'idx', 'title', and 'genres'. At the bottom, there are buttons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The status bar at the bottom indicates 'movies 1 x' and 'Read Only'.

2. Copied movies.csv from local machine to NiFi CLI & verified that the movies.csv file is on the NiFi server under data folder.

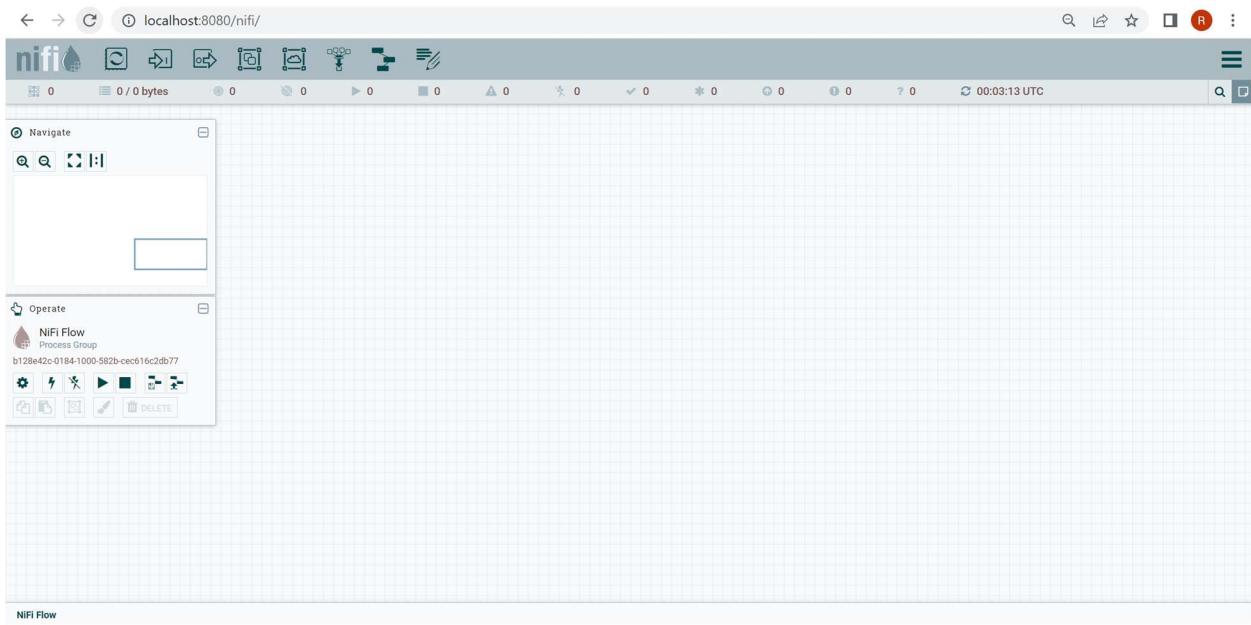


The screenshot shows a Command Prompt window. The user is in the directory C:\Users\reema\Desktop\Data Science\Module_17\Final_Ass_2. They run the command: docker cp ./movies.csv nifi:/opt/nifi/nifi-current/data. The prompt then returns to the previous directory: C:\Users\reema\Desktop\Data Science\Module_17\Final_Ass_2>

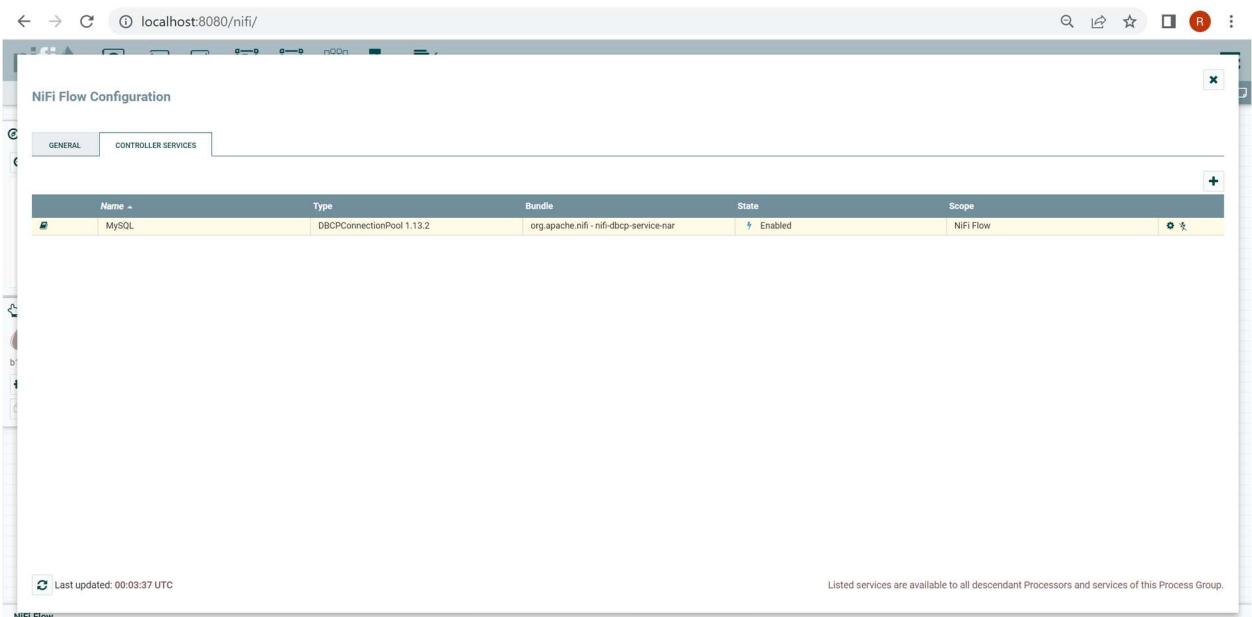


The screenshot shows a terminal window with the command: docker exec -it b42b0a2d3fc435a31abb0f75855c32921a0aa0495232be223c5c9411038a28 /bin/sh. Inside the container, the user runs: \$ pwd, which shows /opt/nifi/nifi-current. Then they run: \$ mkdir data, \$ cd data, \$ pwd, which shows /opt/nifi/nifi-current/data. Finally, they run: \$ ls, which lists 'movies.csv'.

3. Navigated to <http://localhost:8080/nifi/> & opened the NiFi UI.



4. Successfully created and enabled the MySQL controller service.



5. Screenshot of the *controller* screen showing that the three *controller* services (*reader*, *writer*, and MySQL) are enabled.

Controller Service Details

SETTINGS PROPERTIES COMMENTS

Required field

Property	Value
Database Connection URL	jdbc:mysql://myfinalsql:3306
Database Driver Class Name	com.mysql.jdbc.Driver
Database Driver Location(s)	/opt/nifi/drivers/mysql-connector-j-8.0.31.jar
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Password	No value set
Database User	root
Password	Sensitive value set
Max Wait Time	500 millis
Max Total Connections	8
Validation query	No value set
Minimum Idle Connections	0
Max Idle Connections	8
Max Connection Lifetime	-1

OK

Configure Controller Service

SETTINGS

PROPERTIES

COMMENTS

Required field



Property	Value
Schema Access Strategy	Use String Fields From Header
CSV Parser	Apache Commons CSV
Date Format	No value set
Time Format	No value set
Timestamp Format	No value set
CSV Format	Custom Format
Value Separator	,
Record Separator	\n
Treat First Line as Header	true
Ignore CSV Header Column Names	false
Quote Character	"
Escape Character	\
Comment Marker	No value set
Null String	No value set

CANCEL

APPLY

Configure Controller Service

Required field

Property	Value
Schema Write Strategy	Do Not Write Schema
Schema Cache	No value set
Schema Access Strategy	Inherit Record Schema
Date Format	No value set
Time Format	No value set
Timestamp Format	No value set
Pretty Print JSON	false
Suppress Null Values	Never Suppress
Output Grouping	Array
Compression Format	none

CANCEL **APPLY**

localhost:8080/nifi/

NiFi Flow Configuration

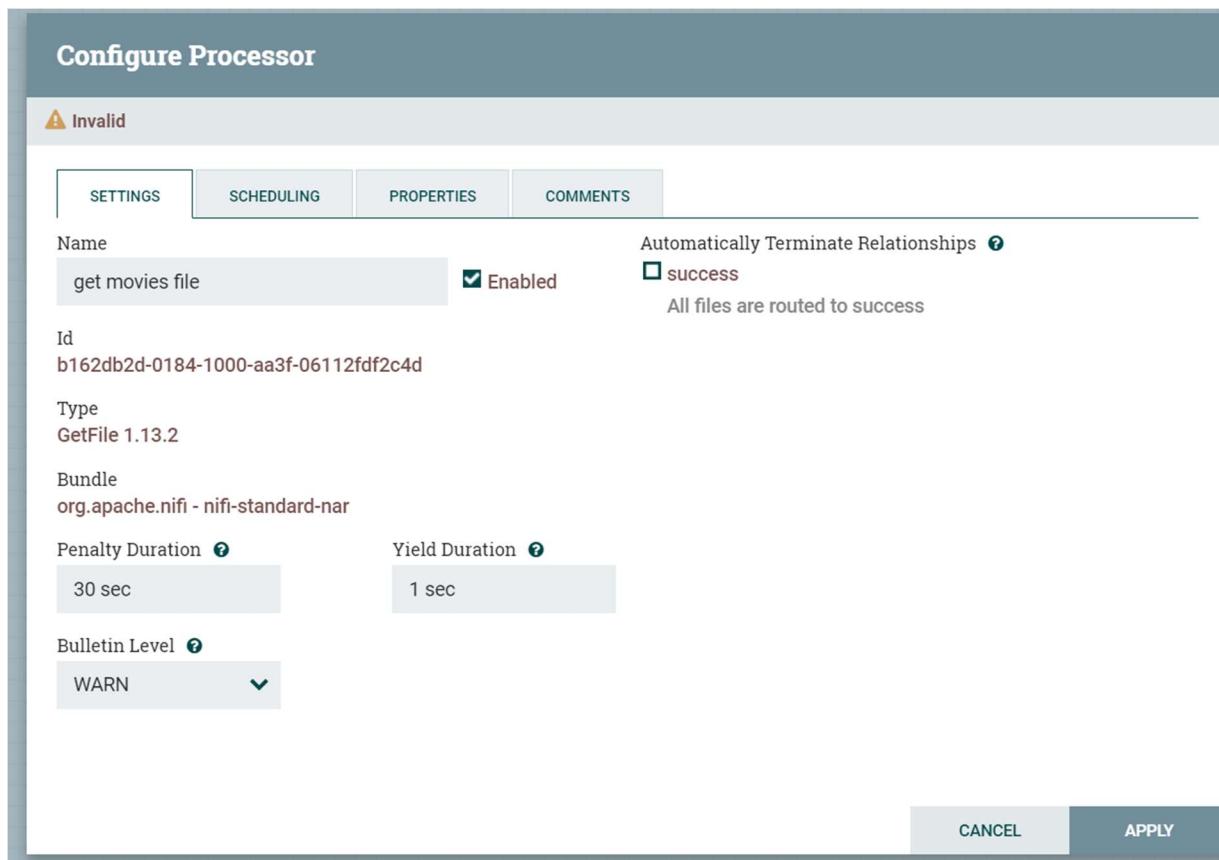
GENERAL **CONTROLLER SERVICES**

Name	Type	Bundle	State	Scope
CSVReader	CSVReader 1.13.2	org.apache.nifi - nifi-record-serialization-services-n...	Enabled	NiFi Flow
JsonRecordSetWriter	JsonRecordSetWriter 1.13.2	org.apache.nifi - nifi-record-serialization-services-n...	Enabled	NiFi Flow
MySQL	DBCPConnectionPool 1.13.2	org.apache.nifi - nifi-dbc-service-nar	Enabled	NiFi Flow

Last updated: 00:03:37 UTC

Listed services are available to all descendant Processors and services of this Process Group.

6. Screenshot showing complete data pipeline, including all five *processors*: GetFile, SplitText, ConvertRecord, ConvertJSONToSQL, and PutSQL.



Configure Processor

! Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field



Property	Value	
Input Directory	? /opt/nifi/nifi-current/data	
File Filter	? [^\.]*	
Path Filter	? No value set	
Batch Size	? 10	
Keep Source File	? false	
Recurse Subdirectories	? true	
Polling Interval	? 0 sec	
Ignore Hidden Files	? true	
Minimum File Age	? 0 sec	
Maximum File Age	? No value set	
Minimum File Size	? 0 B	
Maximum File Size	? No value set	

CANCEL

APPLY

Configure Processor

! Invalid

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Name SplitText	<input checked="" type="checkbox"/> Enabled	Automatically Terminate Relationships ?	
Id b1679300-0184-1000-7cf8-44ead8f7f2a9	<input checked="" type="checkbox"/> failure If a file cannot be split for some reason, the original file will be routed to this destination and nothing will be routed elsewhere		
Type SplitText 1.13.2	<input checked="" type="checkbox"/> original The original input file will be routed to this destination when it has been successfully split into 1 or more files		
Bundle org.apache.nifi - nifi-standard-nar	<input type="checkbox"/> splits The split files will be routed to this destination when an input file is successfully split into 1 or more split files		
Penalty Duration ? 30 sec	Yield Duration ? 1 sec		
Bulletin Level ? WARN			

CANCEL

APPLY

Configure Processor

 Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field



Property	Value
Line Split Count	1
Maximum Fragment Size	No value set
Header Line Count	1
Header Line Marker Characters	No value set
Remove Trailing Newlines	true

CANCEL

APPLY

Configure Processor

! Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field



Property	Value	
Record Reader	? CSVReader	→
Record Writer	? JsonRecordSetWriter	→
Include Zero Record FlowFiles	? true	

CANCEL

APPLY

Configure Processor

Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Name

ConvertJSONToSQL

Enabled

Automatically Terminate Relationships

failure

A FlowFile is routed to this relationship if it cannot be converted into a SQL statement. Common causes include invalid JSON content or the JSON content missing a required field (if using an INSERT statement type).

original

When a FlowFile is converted to SQL, the original JSON FlowFile is routed to this relationship

sql

A FlowFile is routed to this relationship when its contents have successfully been converted into a SQL statement

Id

b16a85dd-0184-1000-c713-eb7d1b2135da

Type

ConvertJSONToSQL 1.13.2

Bundle

org.apache.nifi - nifi-standard-nar

Penalty Duration

30 sec

Yield Duration

1 sec

Bulletin Level

WARN

CANCEL

APPLY

Configure Processor

! Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field



Property	Value	→	▲
JDBC Connection Pool	MySQL	→	▲
Statement Type	INSERT		
Table Name	movies		
Catalog Name	movielens		
Schema Name	No value set		
Translate Field Names	true		
Unmatched Field Behavior	Ignore Unmatched Fields		
Unmatched Column Behavior	Fail on Unmatched Columns		
Update Keys	No value set		
Quote Column Identifiers	false		
Quote Table Identifiers	false		
SQL Parameter Attribute Prefix	sql		▼

Read/Write: 0 bytes / 0 bytes

0 min

CANCEL

APPLY

Configure Processor

⚠ Invalid

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Name PutSQL	<input checked="" type="checkbox"/> Enabled	Automatically Terminate Relationships ?	
Id b16cea17-0184-1000-b57f-44d4643164c4	<input checked="" type="checkbox"/> failure <p>A FlowFile is routed to this relationship if the database cannot be updated and retrying the operation will also fail, such as an invalid query or an integrity constraint violation</p> <input type="checkbox"/> retry <p>A FlowFile is routed to this relationship if the database cannot be updated but attempting the operation again may succeed</p> <input checked="" type="checkbox"/> success <p>A FlowFile is routed to this relationship after the database is successfully updated</p>		
Type PutSQL 1.13.2			
Bundle org.apache.nifi - nifi-standard-nar			
Penalty Duration ? 30 sec	Yield Duration ? 1 sec		
Bulletin Level ? WARN			

CANCEL

APPLY

Configure Processor

! Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

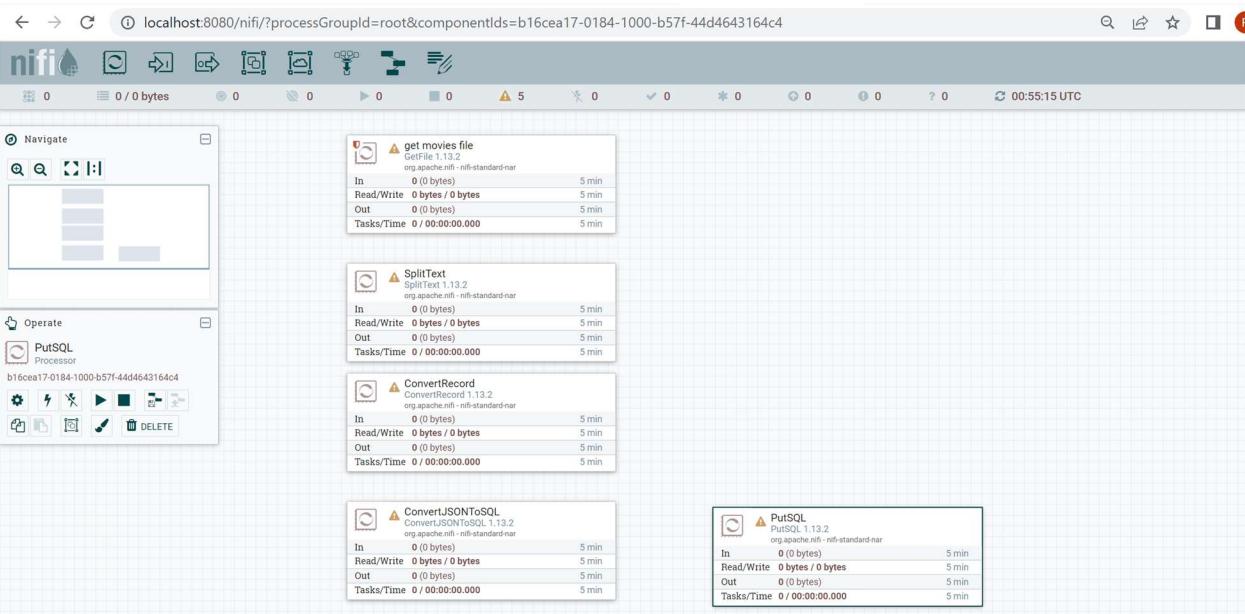
Required field



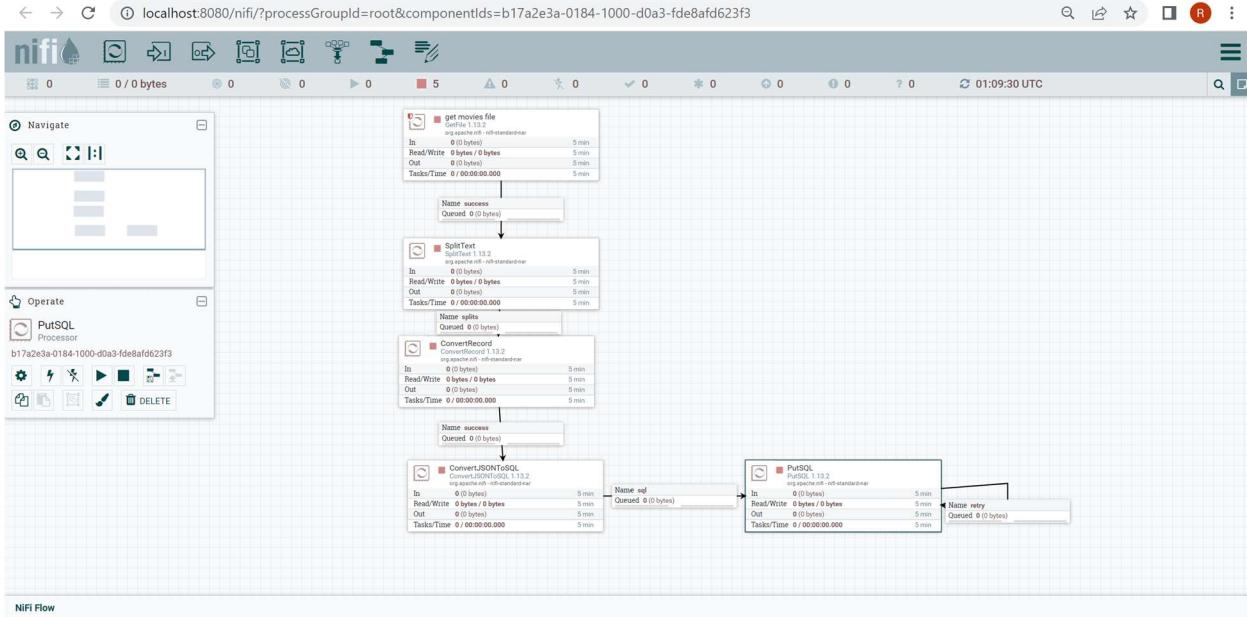
Property	Value
JDBC Connection Pool	MySQL
SQL Statement	No value set
Support Fragmented Transactions	true
Database Session AutoCommit	false
Transaction Timeout	No value set
Batch Size	100
Obtain Generated Keys	false
Rollback On Failure	false

CANCEL

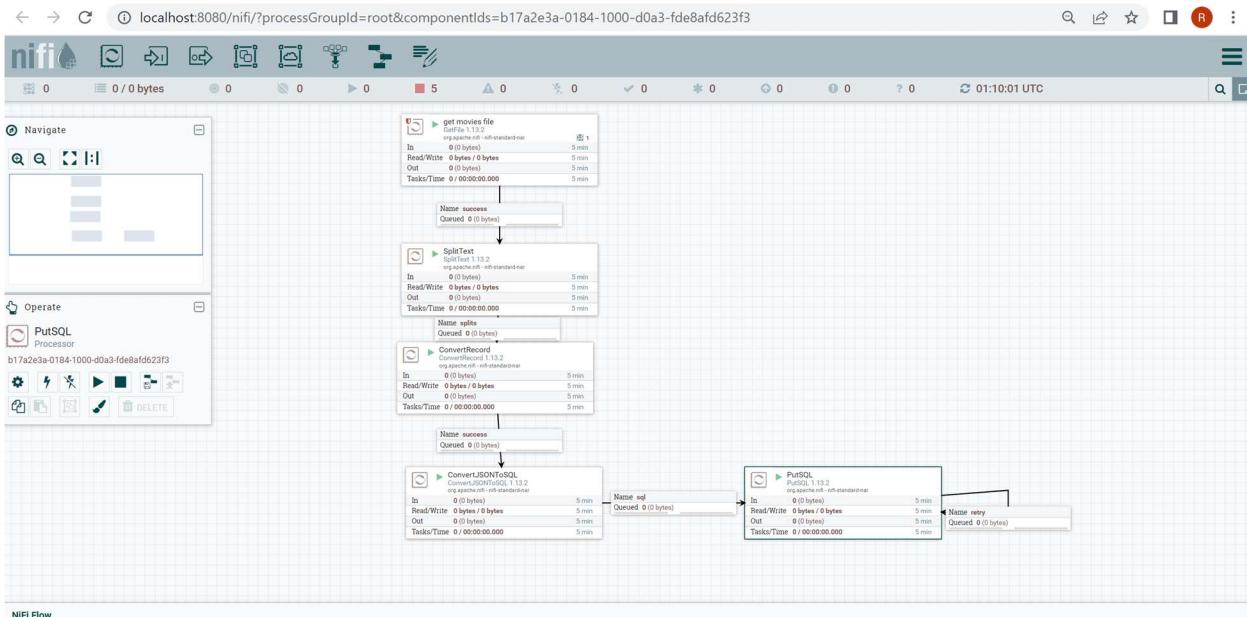
APPLY

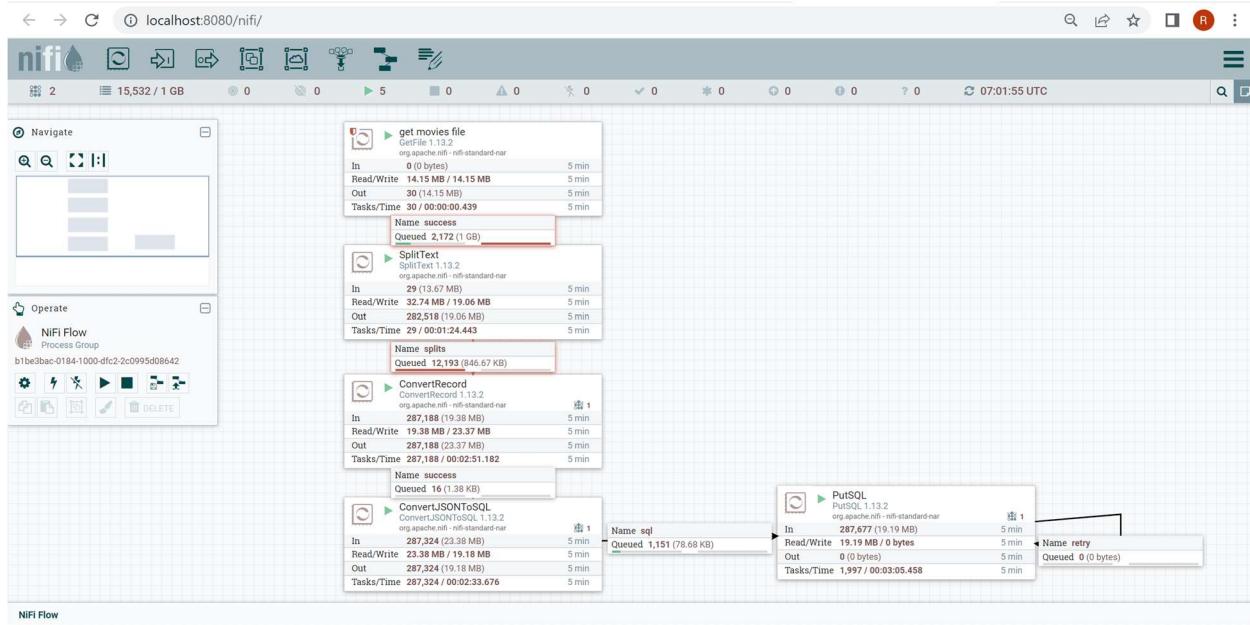


7. Screenshot of all five *processors* showing that the correct *connectors* have been added between the *processors*.



8. Screenshot showing all five *processors* are connected and running.





9. Screenshot showing the result of below query which shows that the movies table in the movielens database is now saturated with data.

The screenshot shows the MySQL Workbench interface with the following details:

- Connection:** new connection
- Schemas:** movielens
- Query:** SQL File 4*

```
1 USE movielens;
2 SELECT * FROM movies;
```

- Result Grid:**

movieId	title	genres
1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
2	Jumanji (1995)	Adventure Children Fantasy
3	Grumpier Old Men (1995)	Comedy Romance
6	Heat (1995)	Action Crime Thriller
4	Waiting to Exhale (1995)	Comedy Drama Romance
5	Father of the Bride Part II (1995)	Comedy
7	Sabrina (1995)	Comedy Romance
8	Tom and Huck (1995)	Adventure Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action Adventure Thriller
11	American President, The (1995)	Comedy Drama Romance