DATA INGESTION INTO HADOOP USING APCHE NIFI

Activity 2

Apache NiFi is a dataflow system based on the concepts of flow-based programming. It supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. It has a web-based user interface for design, control, feedback, and monitoring of data flows. Put simply NiFi was built to automate the flow of data between systems. While the term 'dataflow' is used in a variety of contexts, we use it here to mean the automated and managed flow of information between systems.

In this exercise, students will learn how to move data from MySQL database into a hive table all while capturing new record inserts in real time.

MySQL Database: Linux
Hive: Hortonworks Sandbox

Creating the Database in MySQL

Log in to MySQL and create a source database from which records will be fetched.

```
CREATE database falconpro;

CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `age` int(11) DEFAULT NULL,
  `gender` char(1) DEFAULT NULL,
  `occupation_id` int(11) DEFAULT NULL,
  `zip_code` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

Insert the following records into the table user.

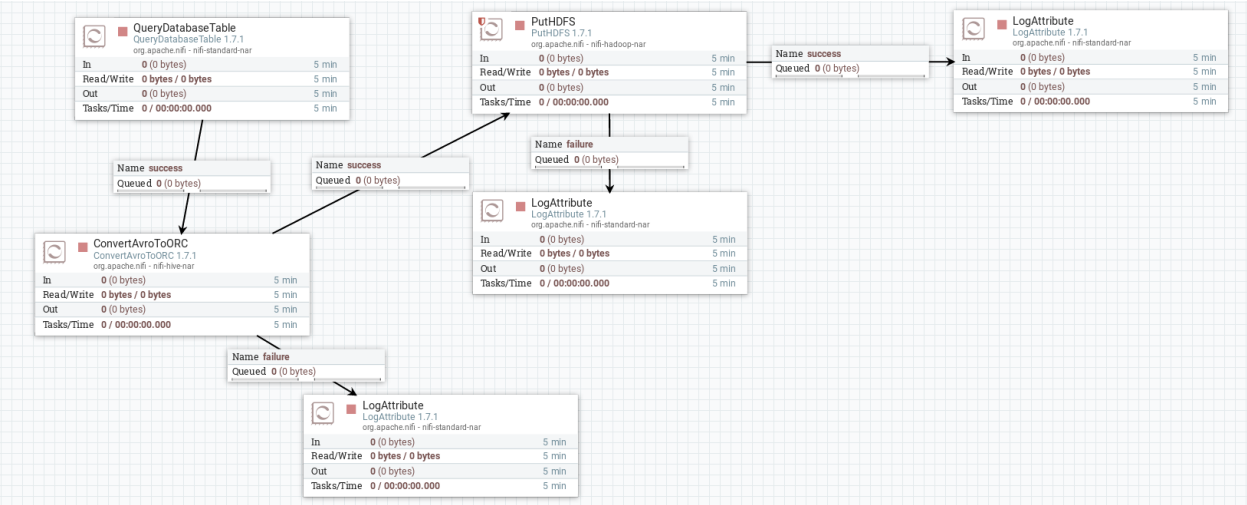
```
INSERT INTO `users` VALUES (1,24,'M',20,'85711'),(2,53,'F',14,'94043'),
(3,23,'M',21,'32067'),(4,24,'M',20,'43537'),(5,33,'F',14,'15213'),
(6,42,'M',7,'98101'),(7,57,'M',1,'91344'),(8,36,'M',1,'05201');
```

Creating the Table in Hive

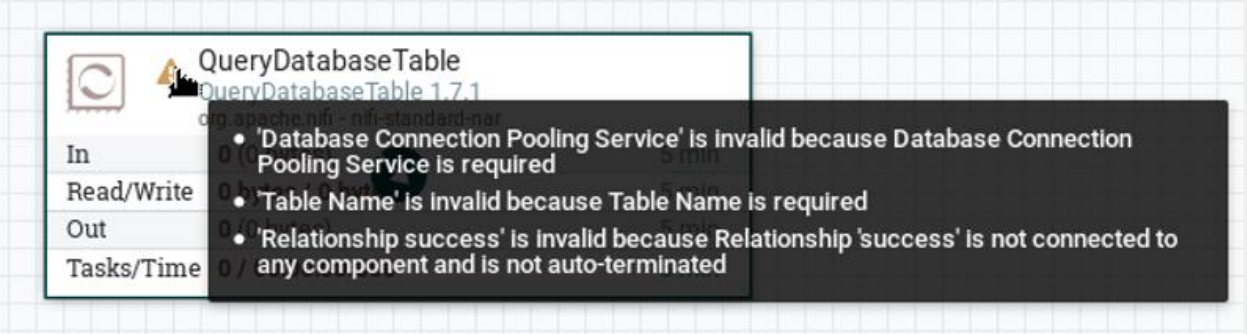
Create a table by running the following script below.

```
CREATE external TABLE falconUsers (
  id int,
  age int,
  gender string,
  occupation_id int,
  zip_code string)
stored as orc
location '/user/maria_dev/falcon_db_copy';
```

Creating the Flow in Nifi



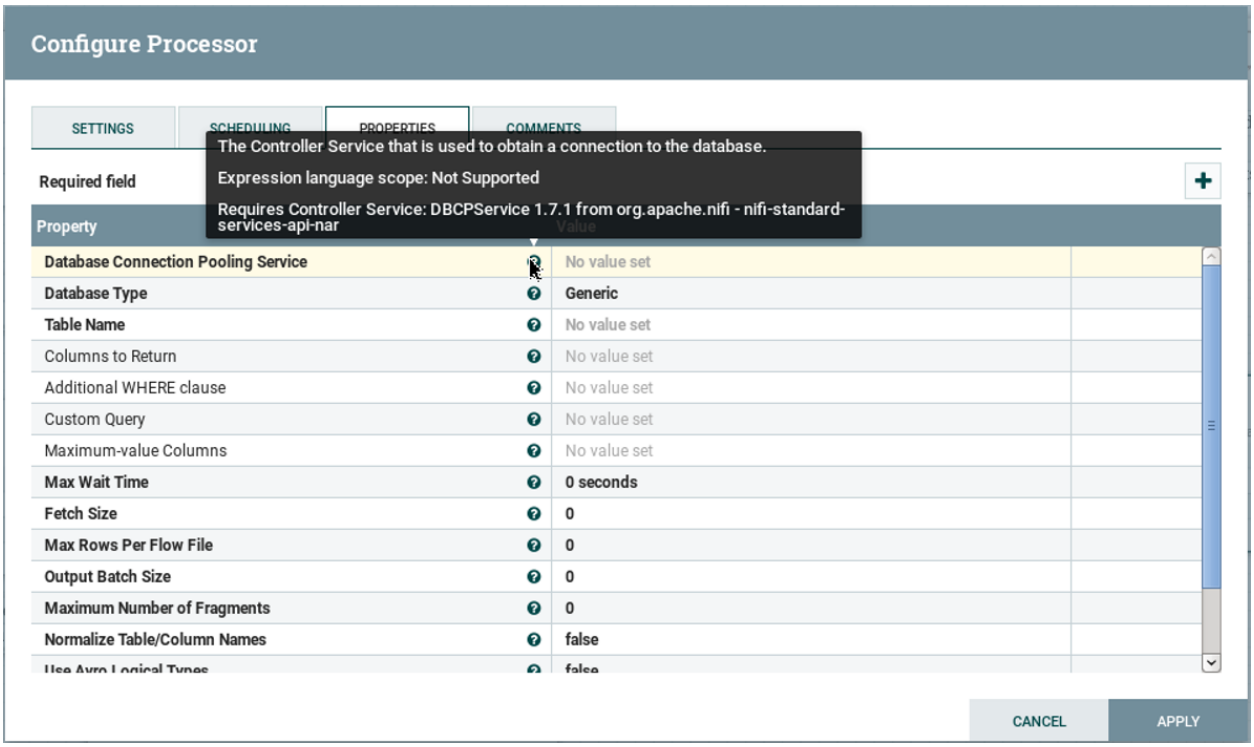
Since the purpose of the exercise is to fetch records from a database, specifically from a table in that database, the processor of choice would be the QueryDatabaseTable component.



Of the three warning messages listed, we will focus on the 'Database Connection Pooling Service' warning as the 'Table Name' warning is self-explanatory and the 'Relationship success' warning was addressed in Exercise 1.

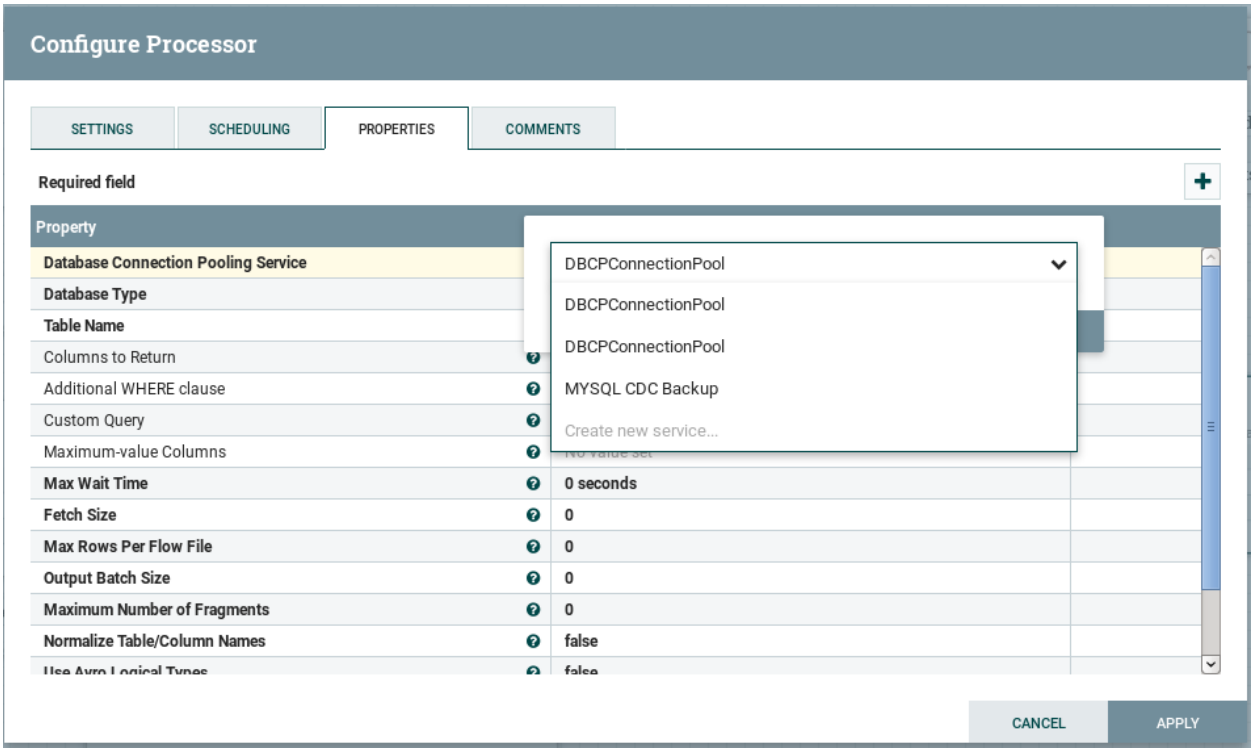
Inside QueryDatabaseTable

This processor generates and executes a SQL select query to fetch all rows whose values in the specified Maximum Value column(s) are larger than the previously-seen maxima. Query result will be converted to Avro format. As seen in the image below, the Database Connection Pooling Service requires values in its field. But in this case it requires a connection to a Controller Service. This component will be used to establish connection to our database.

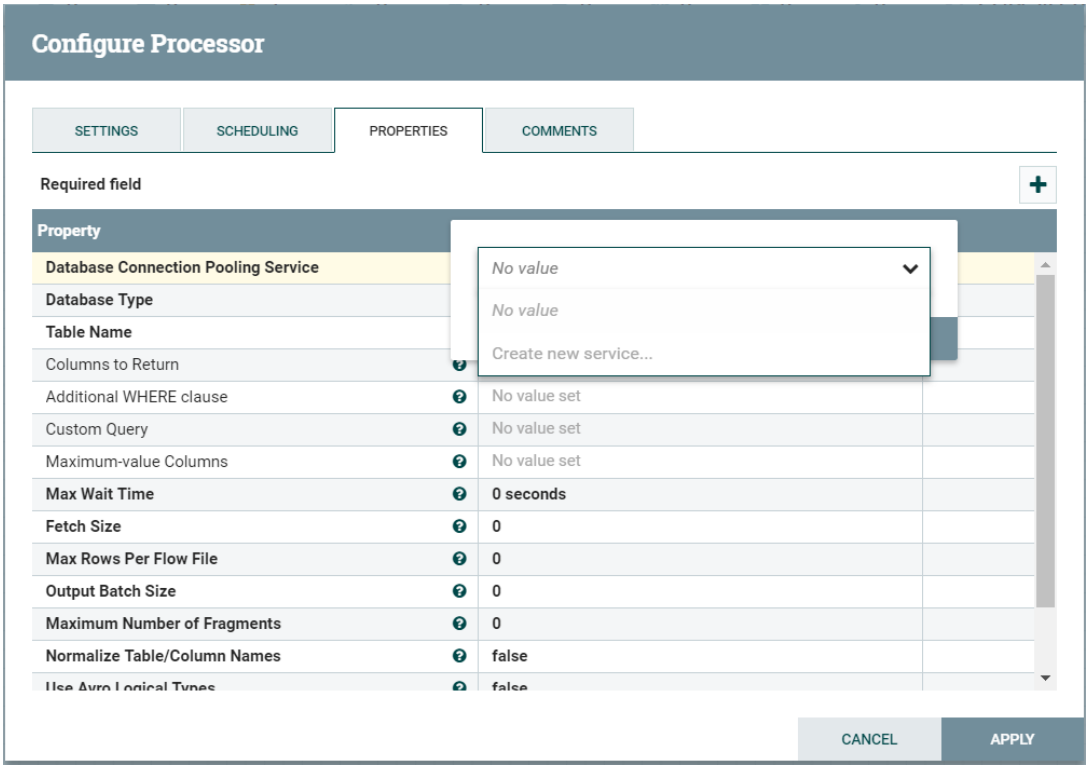


Possible connection are listed in a dropdown by simply inside the value field.

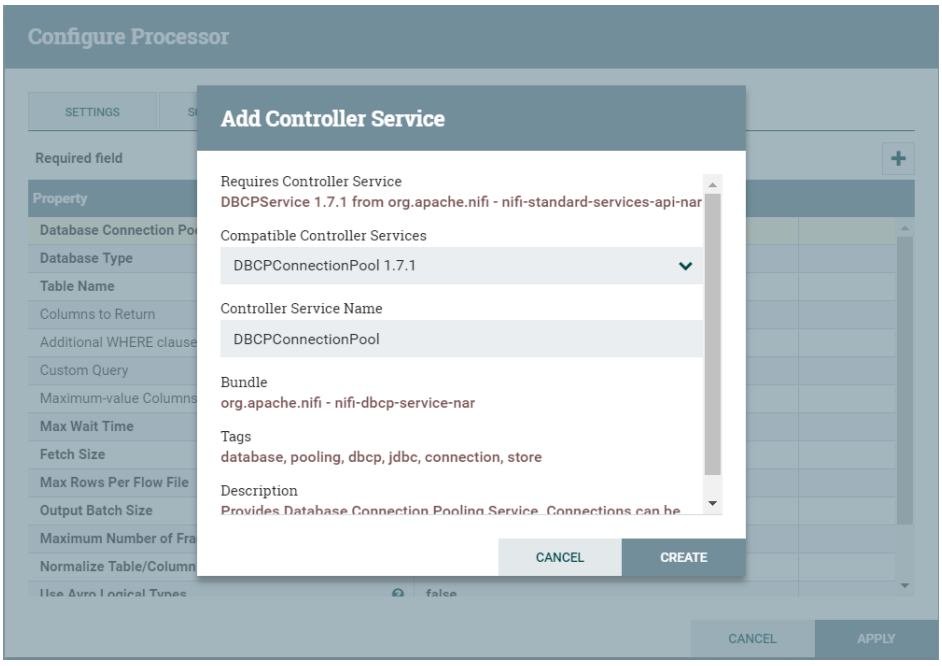
Select DBCPConnectionPool.

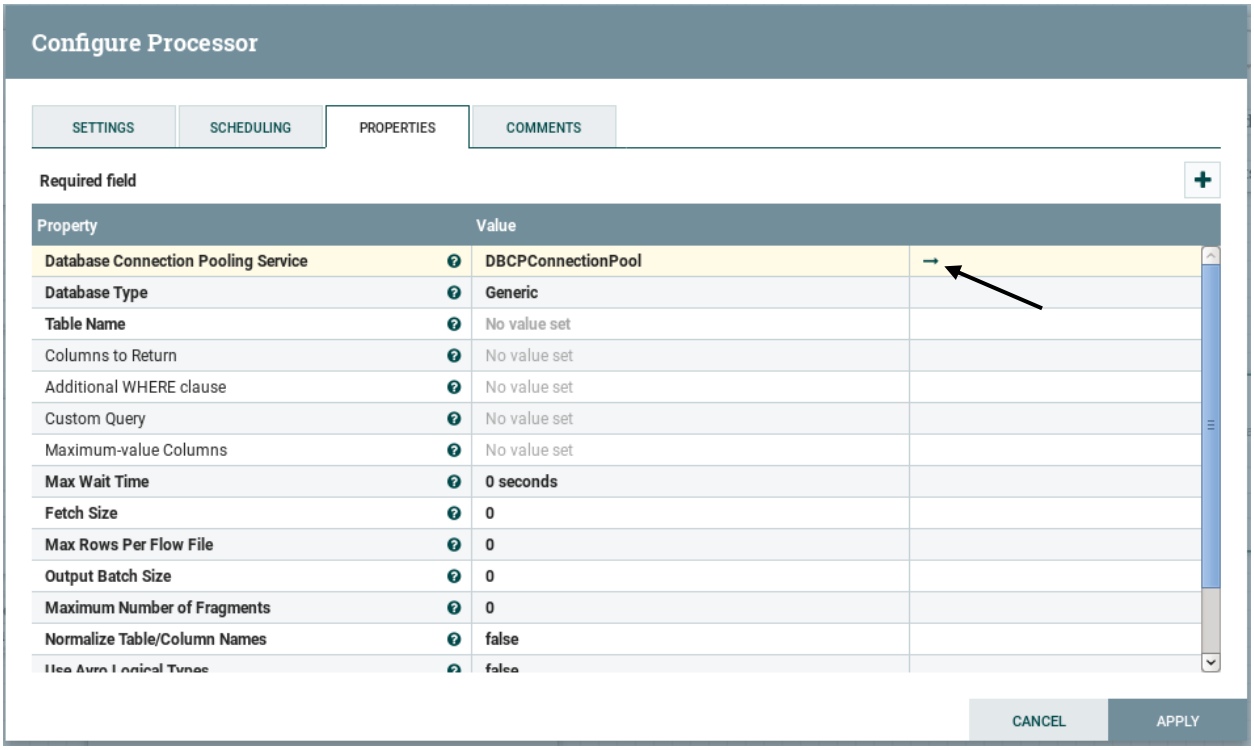


Note: For first time users the “Create new service” field would appear instead of the list of options in the dropdown in the image above.

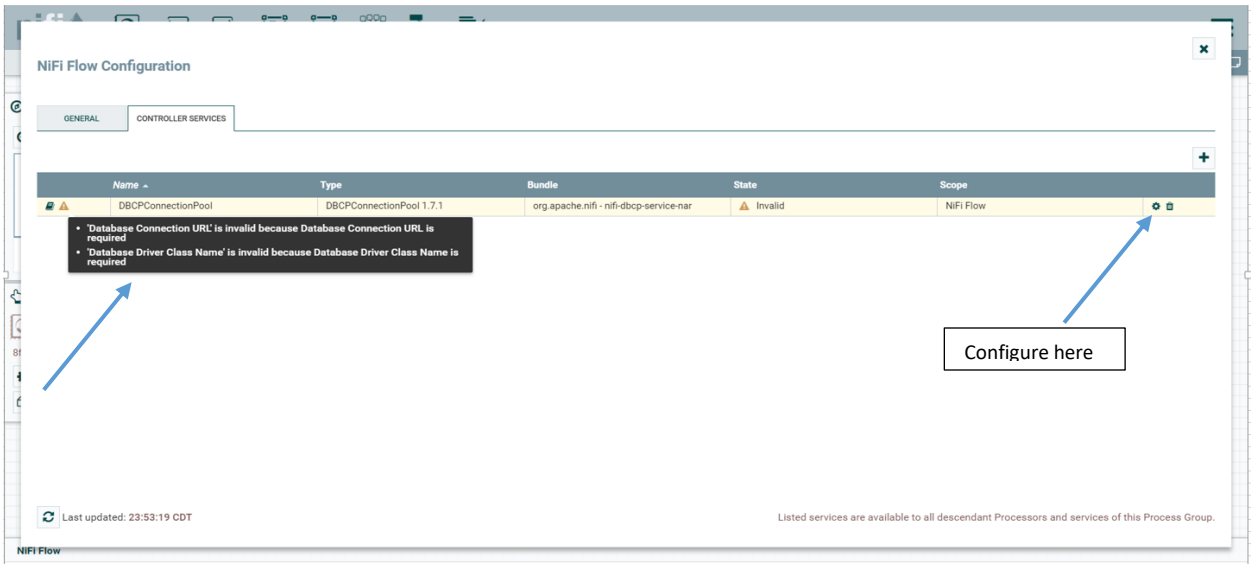


By clicking on Create new service… Nifi automatically makes a suggestion for the Controller Service to use.

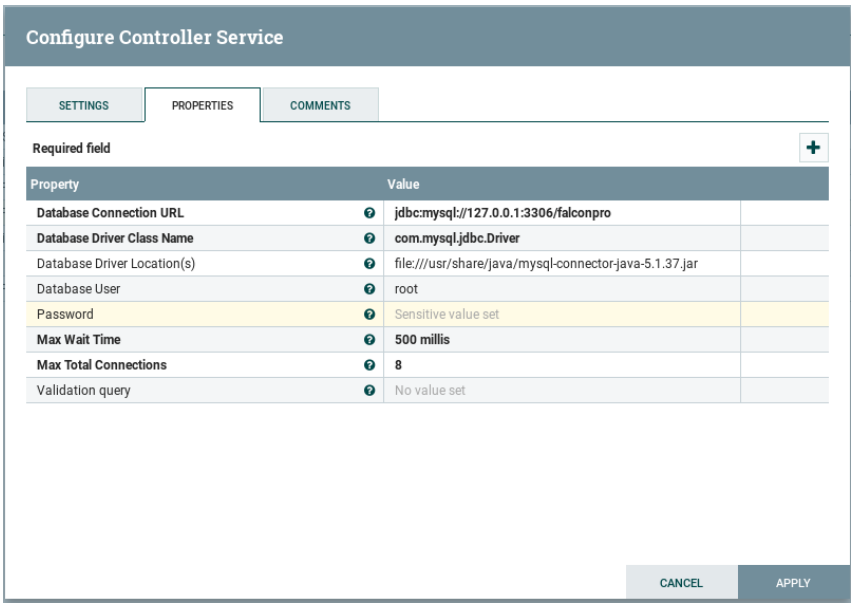




Upon selecting it an arrow appears. Click on the arrow to provide properties field values to establish connection to MySQL and the intended database.



Below are the details that go into this component. Make sure to use the proper path for your jar file, correct values for your Database user name and password.



After adding the details, the component must be enabled. Going back to the QueryDatabaseTable processor, add the following details:

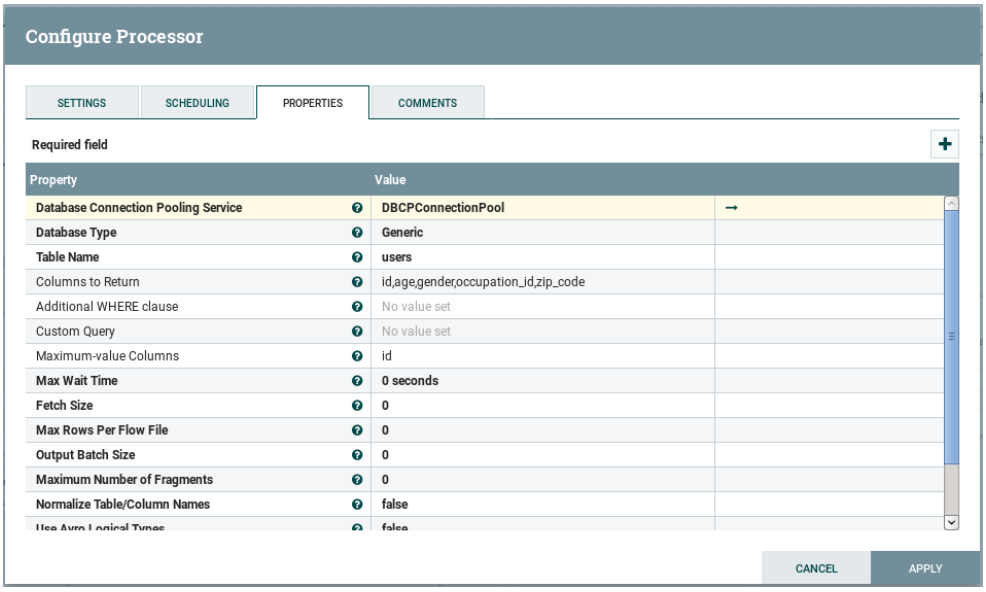
Table Name: users

Columns: users.columns

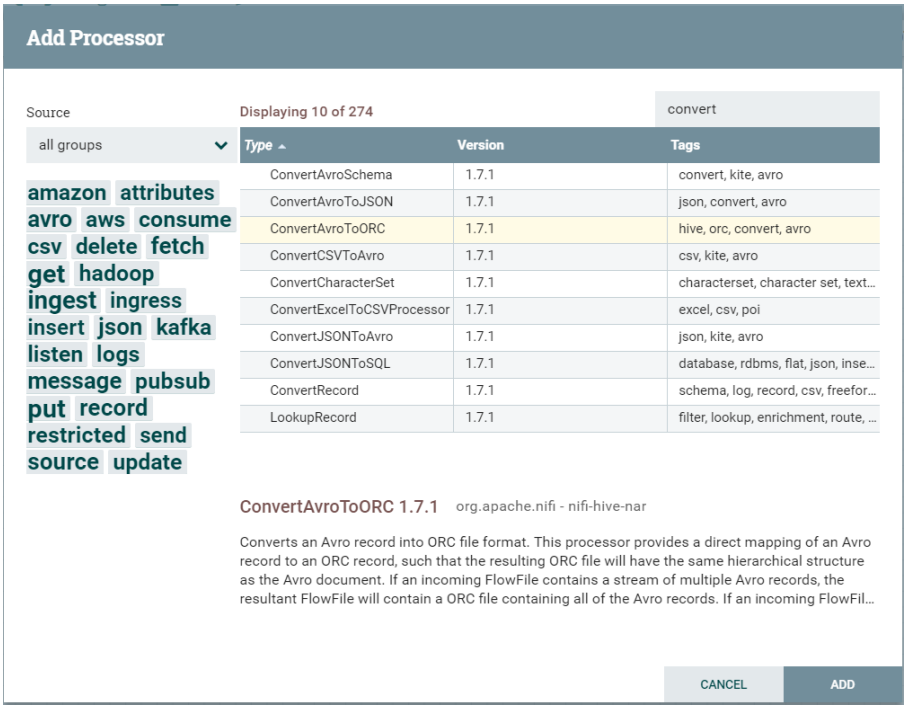
Maximum-value: id

As stated above, this processor generates and executes a SQL select query to fetch all rows whose values in the specified **Maximum Value column**(s) are larger than the previously-seen maxima.

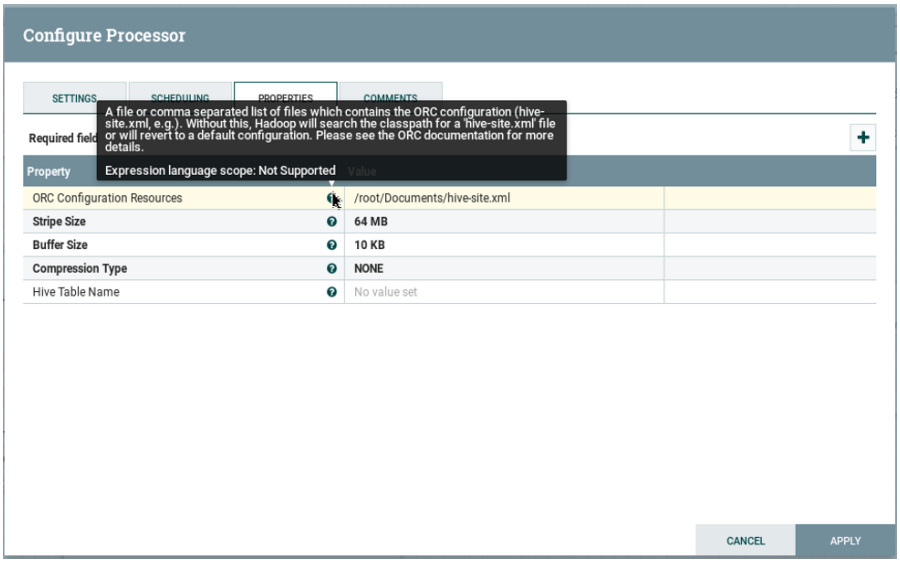
For this being the case, it's best to always use fields with auto-increment unique primary keys, current timestamp or other trigger events on your table for the Maximum Value in the processor.



Next up is the ConvertAvroToORC processor. In many cases, using avro type file to populate data in a hive table can result in null values in the columns. Therefore flowfiles in Avro format must be converted to ORC as the default file format of query results from the QueryDatabaseTable are converted to Avro format.

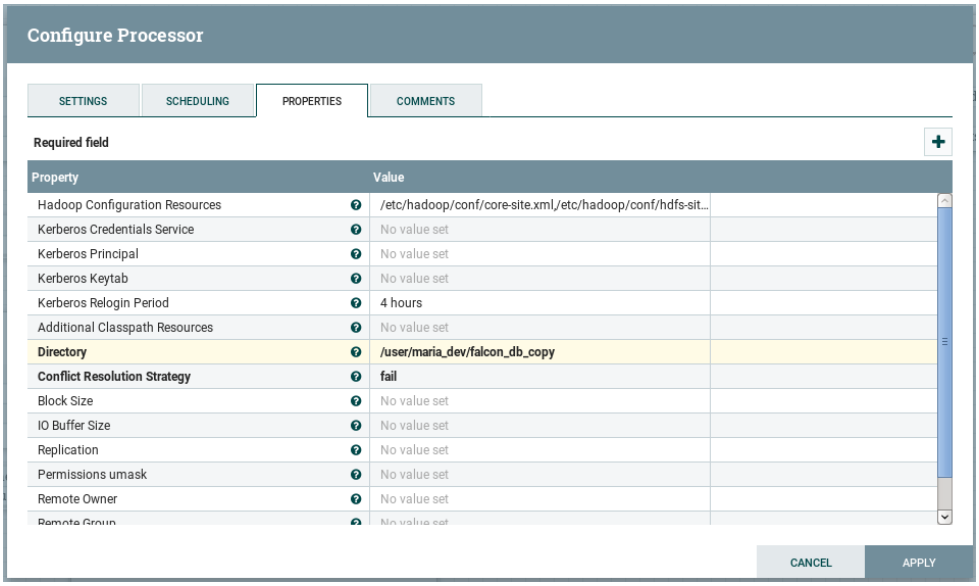


The processor is configured by adding the path to the hive-site.xml in the ORC Configuration Resources value field.



We will then create a LogAttribute processor to keep track of failed flow file events and send all the success flow files to our PutHDFS processor.

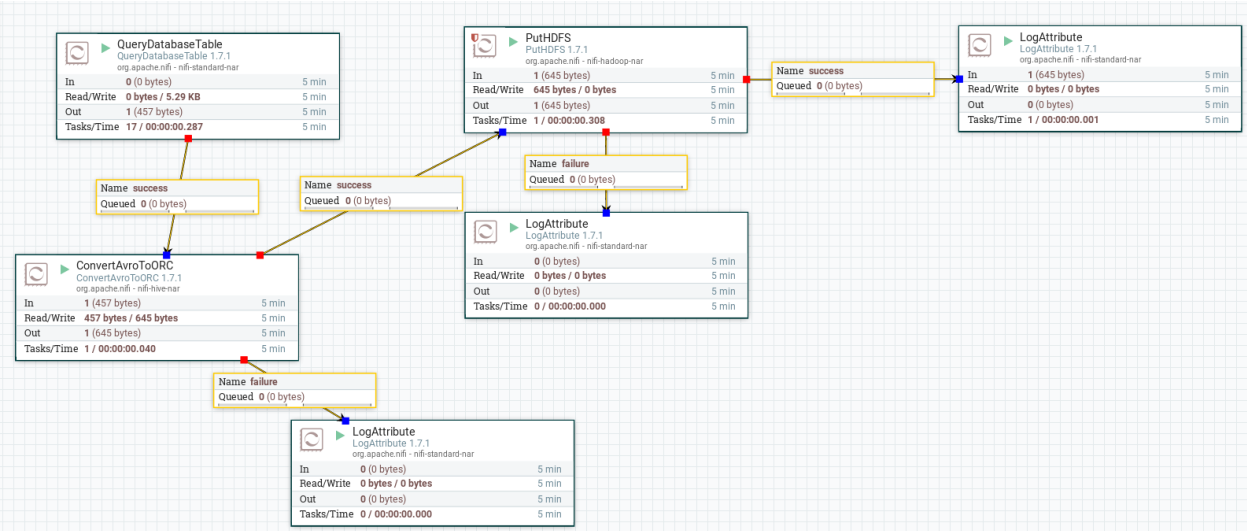
The PutHDFS should be configured as such:



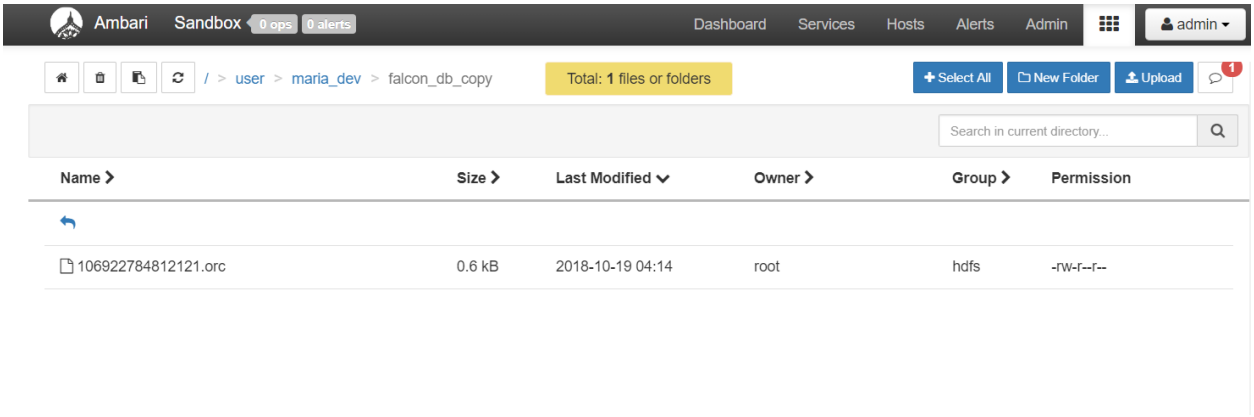
Lastly we will add two additional LogAttribute processors to track our success and failures.

Select all the processor and press PLAY.

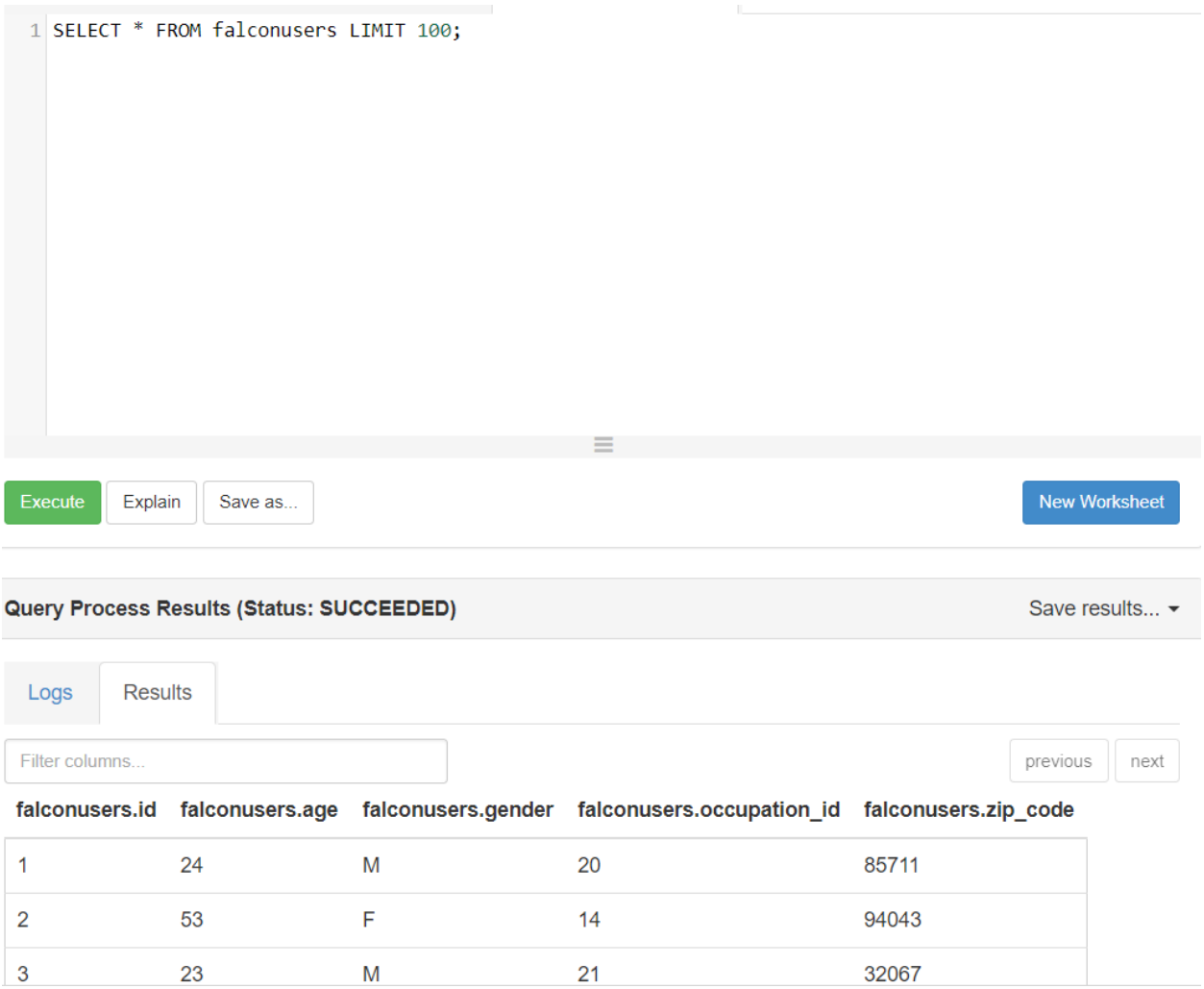
Nifi



HDFS



HIVE



falconusers.id	falconusers.age	falconusers.gender	falconusers.occupation_id	falconusers.zip_code
1	24	M	20	85711
2	53	F	14	94043
3	23	M	21	32067

For incremental imports, add new records in the users table in the falcon database in MySQL all while Nifi running. Nifi will automatically fetch the new records into Hadoop.