

KNIME opens the Doors to Big Data

*A Practical example of Integrating any
Big Data Platform into KNIME*

Tobias Koetter
Rosaria Silipo

Tobias.Koetter@knime.com
Rosaria.Silipo@knime.com

Table of Contents

KNIME opens the Doors to Big Data <i>A Practical example of Integrating any Big Data Platform into KNIME for Time Series Prediction</i>	1
Summary	3
Do We Really Need Big Data?.....	3
The Energy Prediction Project	4
The Speed Challenge	5
Connector Nodes to Big Data Platforms	6
1. Impala Connector	7
2. Apache HIVE	8
3. parStream	8
Writing SQL Queries	9
1. Database Query Node	9
2. Database Joiner Node	10
3. Database GroupBy Node	12
4. Other Database Helper Nodes	13
Running the Final SQL Query and Retrieving the Data.....	13
Conclusions.....	15

Summary

It is ever more the case that the amount of available raw data collected by a system increases at an exponential rate, quickly reaching a very large size and a very high number of features, sometimes qualifying for what is referred to as “big data”. In case of particularly large data sets, it can be helpful to take advantage of big data platform performances, especially to run ETL procedures. Indeed, by designing appropriate SQL queries, it is possible to run most of ETL procedures directly on big data platforms. Integrating the execution of SQL queries on big data platforms into KNIME would, of course, speed up the whole KNIME workflow execution by preparing the data quicker for further analytical processes.

Once established that it would be beneficial to integrate some big data processing into a KNIME workflow, the problem has just started. Connecting to a big data platform, designing the appropriate SQL query and retrieving the data accordingly, for example, can be quite complex. There are also so many different types of big data platforms that choosing a specific one to work with can turn into quite a long and tedious task. And this is where KNIME can help.

KNIME provides a number of connector nodes to connect to databases in general and to big data platforms in particular through KNIME Big Data Extension. Some connector nodes have been specifically designed for specific big data platforms. These dedicated connectors provide a very simple configuration window requiring only the basic access parameters, such as credentials, for example.

Writing a complex SQL query is not for everybody. For the less expert SQL users, KNIME provides a number of SQL transparent nodes, which enable users to set a function without ever touching the underlying SQL query. These SQL helper nodes and the existence of dedicated connector nodes make the implementation of ETL procedures on a big data platform extremely easy and fast. They also make it very easy to switch from one big data platform to another, preserving the agility feature of the KNIME Analytics Platform even after the integration of a big data platform into the workflow.

In this whitepaper we show step-by-step how to integrate a big data platform into a KNIME workflow.

The workflow used in this whitepaper is available on the EXAMPLES server under 004_Databases/004_005_Energy_Prepare_Data (Big Data). A sample of the original data can be retrieved from www.knime.com/files/reducedenergydata.zip. KNIME Analytics Platform can be downloaded free of charge from www.knime.org/download and KNIME Big Data Extension can be purchased at <http://www.knime.org/knime-big-data-extension>.

Do We Really Need Big Data?

There are a few different stages in the discovery process (see “[Open Innovation in the Age of Big Data](#)” by M. Berthold). Big data applies mainly to the first phase, in which not much is yet known about the underlying system; we are looking here for interesting connections in the data in order to gain initial insights. Establishing these connections in a huge amount of semi or fully unstructured and/or highly heterogeneous data – usually referred to as “big data” – might create an even higher value, since such data are derived from bigger and more complex data sources.

On the other hand, connection exploration on large data sets can end up consuming all our computational resources and time. Big data platforms can help us to store, quickly retrieve, and work on such huge amounts of data. Big data platforms come in particularly handy when aggregations are

required so as to change and compact the data structure, effectively reducing its dimensionality. Running data analytics algorithms on a data set with lower dimensionality then no longer prohibits the use of traditional analytical tools, designed to work on smaller data.

But not all projects need big data! KNIME Analytics Platform, for example, can easily handle a few millions rows for ETL or machine learning, providing the column number is limited. Performance starts decreasing with many millions of rows and a high number of input columns. Note that a decrease in performance means that execution on KNIME Analytics Platform becomes slower, even significantly slower, but does not crash.

Big data platforms definitely help boost workflow execution performance. In addition, the integration of big data platforms into KNIME is simple and completed in three steps: first drag & drop the appropriate connector node, second, configure its access parameters (including credentials), and third execute the node. Finally, if you are not satisfied with the big data platform of choice, switching between one big data platform and another is really as easy as connecting two nodes!

The Energy Prediction Project

To make the case for big data, we went back to an older project about energy time series prediction, “[Big Data, Smart Energy, and Predictive Analytics](#)”, which is available under Resources/whitepapers on the [KNIME website](#).

This project focused on smart energy data from the [Irish Smart Energy Trials](#), where the electricity usage of circa 6,000 households and businesses was monitored via meter IDs for a little over one year. The goal of the project was two-fold: create the ability to define custom contract offers and predict future electricity usage to shield the electricity companies from power shortage or power surplus.

The project implementation consisted of three phases, with one workflow for each phase: data preparation, smart meter clustering, and time series prediction.

The data preparation part involved mainly datetime format conversions and aggregations to calculate measures of energy usage for each meter ID. Such energy usage measures were then used to cluster the original 6,000 meter IDs into a maximum of 30 clusters. The average time series of electricity usage values was calculated for each cluster and adopted as the cluster time series prototype. Finally, a few time series analysis techniques were applied to predict future energy usage for each meter ID cluster.

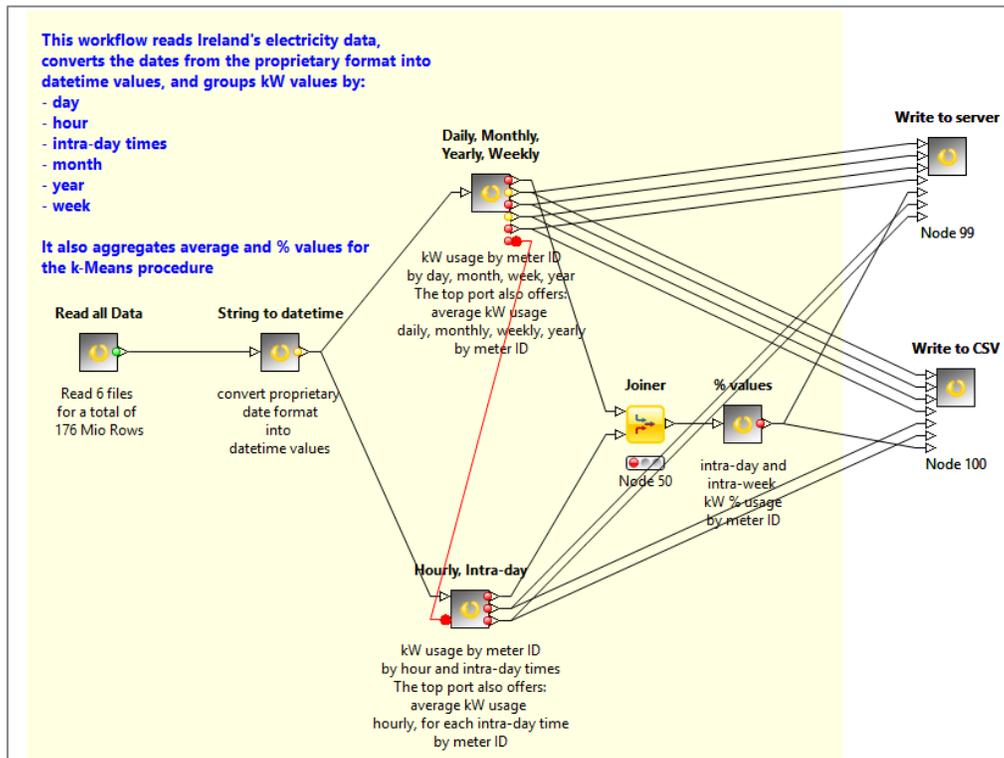
Sampling the energy usage every half an hour for a little over one year (July 2009 - December 2010) across 6000 smart meters produced around 176 million rows in the original data. The first workflow, named “Prepare Data”, imports all these data and processes them for datetime conversion and aggregations. This is the workflow with the longest execution time. While conversion operations take a reasonable time in terms of user waiting time, aggregations and sorting can take an extremely long time if performed on all 176 million rows. Put specifically, “Prepare Data” workflow running on a quad-processor laptop with 8GB RAM with a solid state hard disk takes around 2 days, give or take an hour.

The need to speed up this part of the analysis is evident. Waiting longer than 1 day to get results and maybe, after that, even realizing that the project specs needed changing, can be a big drawback in the whole project design. It was here that we thought a Big Data platform might help to accelerate execution and keep the project times within a reasonable frame.

The “Prepare Data” workflow used in this whitepaper and the pdf document describing the whole project are publicly available and can be downloaded from <http://www.knime.org/white-papers#timeseries>.

Note. A sample of the original data can be retrieved from www.knime.com/files/reducedenergydata.zip. However, the entire dataset must be requested directly from the Irish Social Science Data Archive http://www.ucd.ie/t4cms/CER_energy_electric_issda-data-request-form.docx

Figure 1. The original “Prepare Data” workflow. This workflow imports the data from the Iris Smart Energy Trials, converts them, and aggregates them to extract meaningful measures about energy usage for each meter ID.



The Speed Challenge

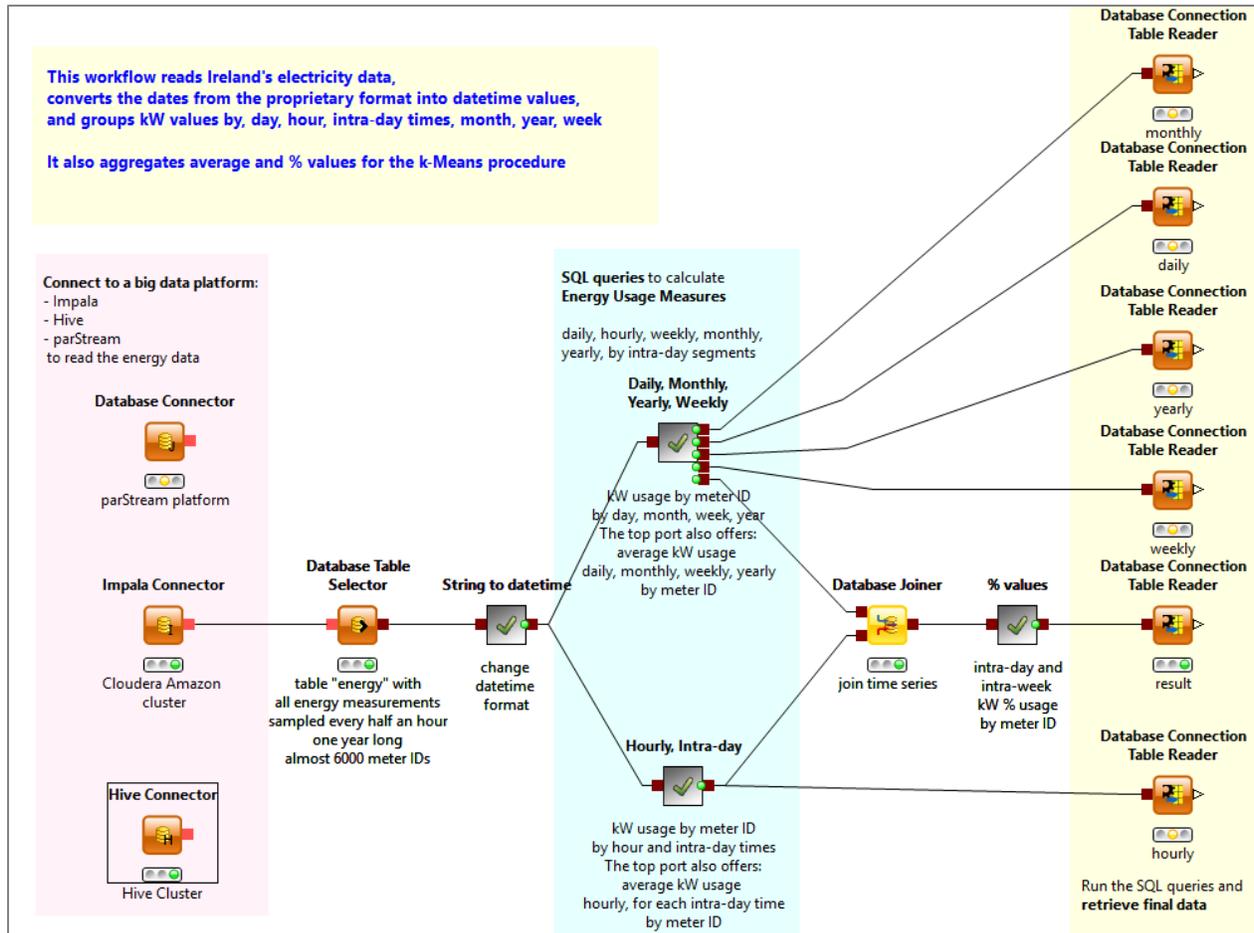
As mentioned above, in terms of computational performance, the bottle neck of this whole project lies in the “Prepare Data” workflow. Thus, we have turned to a series of in-database aggregations using different flavors of big data platforms.

A copy of the original data set has been dumped into HDFS Hive, Impala, and parStream databases. We then build a workflow to access any of these database tables and to perform the necessary in-database aggregations.

In the new workflow, first a Connector node establishes the connection with the database and then a Database Table Selector node selects the table with the data. At this point, we can build the appropriate SQL queries to perform datetime conversions, aggregate the energy values at the hour, day, week, month, and year level, calculate intra-day and intra-week percentage values, and finally retrieve the data.

The figure below shows the final workflow – “004005_Energy_Prepare_Data (Big Data)” – using big data in-database ETL operations to speed up the data manipulation part of the Irish Energy project.

Figure 2. The “004005_Energy_Prepare_Data (Big Data)” workflow: The first part connects to the database and selects the table to work on; the second part nests the aggregations/conversions/joins queries into a single query; the third part runs the final query and exports the data from the database platform into KNIME



Connector Nodes to Big Data Platforms

The new workflow starts with a Connector node. A Connector node creates a connection to a database via a JDBC driver. In order to connect, you generally need to provide the JDBC driver file, the server URL, the access port, the database name, and of course the credentials to access the database.

With regard to the dilemma on whether to use credentials or username and password in the authentication box, please remember that credentials are encrypted by default, while username and password need to be encrypted explicitly by setting a Master Key in the Preferences page. You can create a credential by right-clicking the workflow in the KNIME Explorer panel and selecting “Workflow Credentials”. The list of available credentials will then show in the Credentials menu of the connector node.

The Database category in the Node Repository panel contains a generic Database Connector node. This node connects to all possible databases once the above information has been provided, including the

path to the appropriate JDBC driver file. However, a number of dedicated easier-to-use connector nodes are also available, particularly for big data platforms.

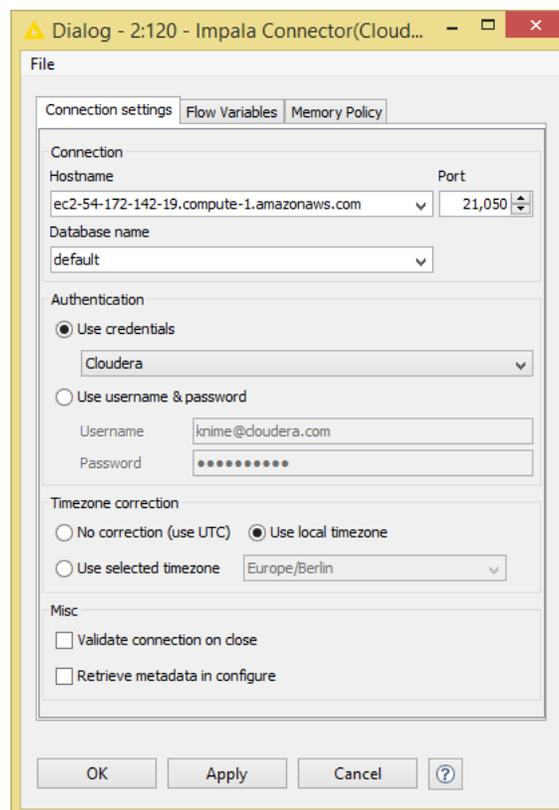
Indeed, connecting to a big data platform can be quite a complex task, if simply using the generic Database Connector node. A dedicated node is supposed to make the connection setting easier and faster, since some settings are already hard-coded in the node. Connector nodes are available from the commercial [KNIME Big Data Extension product](#) for the most commonly used big data platforms and can be found under the Database/Connector category in the Node Repository panel.

The red square output port in the Database Connector node as well as in the other dedicated Connector nodes provides a simple and generic connection to a database. No table has been selected, no query has been assembled yet.

1. Impala Connector

The Impala Connector node is the first connector node that was tried in this project. This node connects to a Cloudera Impala database (<http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html>). Impala is a fully integrated, state-of-the-art analytic database architected specifically to leverage the flexibility and scalability strengths of Hadoop – combining the familiar SQL support and multi-user performance of a traditional analytic database with the rock-solid foundation of open source Apache Hadoop and the production-grade security and management extensions of Cloudera Enterprise.

Figure 3. Configuration window of the Impala Connector node



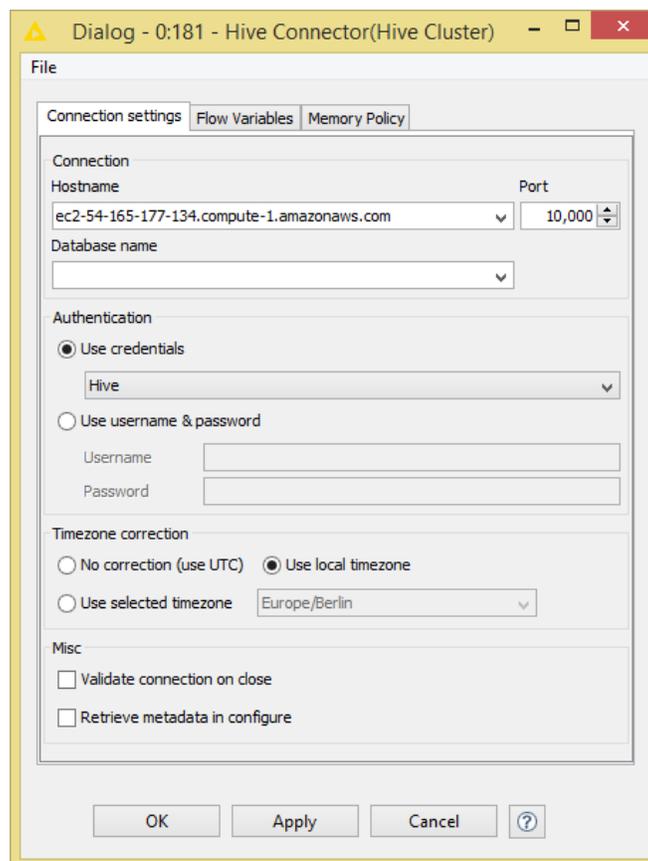
The Impala JDBC driver is already included in the node. So, the only information still required in the node configuration window is the server URL and port, the database name, and the authentication credentials. Executing the node establishes the connection between the KNIME platform and the selected Impala database.

2. Apache HIVE

Let's leave Impala and let's turn to Apache Hive (<https://hive.apache.org/>). Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.

To connect to Apache Hive, there is another dedicated node named Hive Connector. Like the Impala Connector, the Hive Connector node has already embedded the required JDBC driver to connect to an Apache Hive platform. What is still needed is the server URL and the credentials as for the Impala Connector. The database name is not needed, if the default database is used.

Figure 4. Configuration window of the Apache Hive Connector node



3. parStream

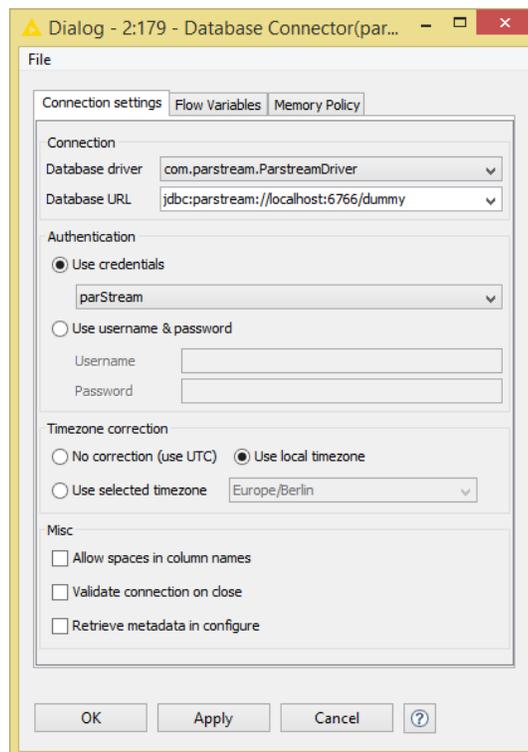
Hadoop Hive and Cloudera Impala are naturally not the only possible choices for a big data platform. If a dedicated Connector node is not available yet inside the Database/Connector category from KNIME Big Data Extension, a connection can always be established using a generic Database Connector node.

For this project, we also used the parStream Big Data platform. parStream DB (<https://www.parstream.com/product/parstream-db/>) is a distributed, massively parallel processing columnar database based on a shared architecture. It was specifically engineered to deliver both big data and fast data, enabled by a unique High Performance Compressed Index (HPCI). This removes the extra step and time required for data decompression.

No KNIME dedicated connector is yet available for the parStream big data platform. We therefore decided to use a Generic Database Connector node. The only difference in settings between a dedicated Impala/Hive Connector and a Generic Database Connector is the required database driver file, which must be provided by the platform vendor and imported into the Preferences page under KNIME/Databases. All other settings, such as Database URL and Credentials, are common to all dedicated and generic connector nodes.

Once we got the database driver from the parStream vendor, the database URL and the access credentials, the connection to the database was as effortless as using a dedicated connector node.

Figure 5. Configuration window of the Connector node to the parStream platform



Writing SQL Queries

There are many ways of building database queries within KNIME, depending on the level of SQL expertise. Most nodes to build database queries can be found in the categories Database/Manipulation and Database/Utility in the Node Repository panel of the KNIME workbench.

1. Database Query Node

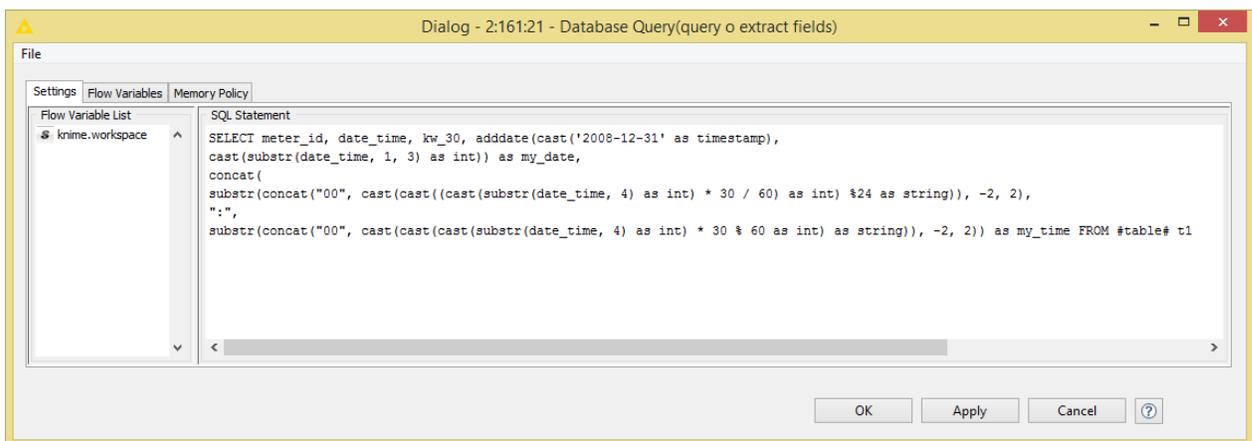
If you are an expert SQL user, you can write the appropriate SQL query in SQL editor of the configuration window of the Database Query node. We used the Database Query node in the metanode named “String

to datetime” to convert the datetime format from the original compact format to a more transparent one, using the following SQL query:

```
SELECT meter_id, date_time, kw_30, adddate(cast('2008-12-31' as timestamp),
      cast(substr(date_time, 1, 3) as int)) as my_date,
      concat(
        substr(concat("00", cast(cast((cast(substr(date_time, 4) as int) * 30 / 60) as int)
%24 as string)), -2, 2),
        ":",
        substr(concat("00", cast(cast(cast(substr(date_time, 4) as int) * 30 % 60 as int)
as string)), -2, 2))
as my_time FROM #table# t1
```

However, if you are less of an SQL expert, a few nodes in the Database/Manipulation category can break up a few operations for you. This includes a Database Row/Column Filter, a Database Sorter, a Database GroupBy, and a Database Joiner node. These nodes implement the corresponding SQL query and only expose the required parameters, saving you the time of building an SQL query from scratch.

Figure 6. Configuration window of the Database Query node

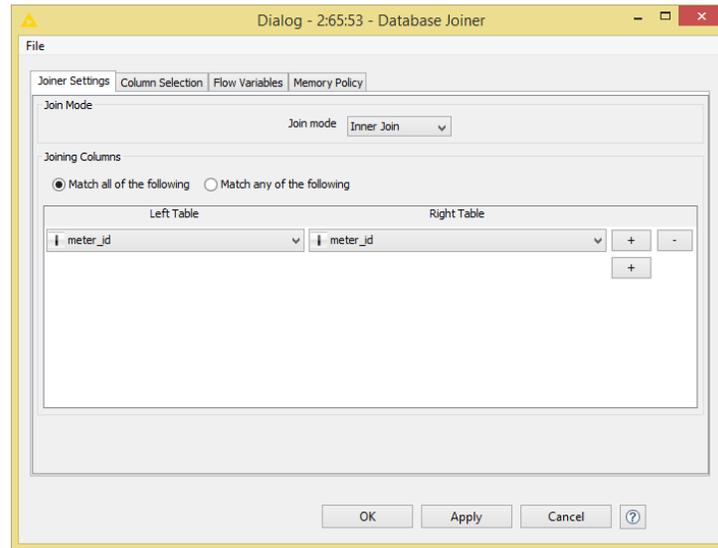


2. Database Joiner Node

For example, in the metanode named “Daily, Monthly, Yearly, Weekly” a series of Database Joiner nodes add additional SQL steps to join the results from the previously built SQL queries.

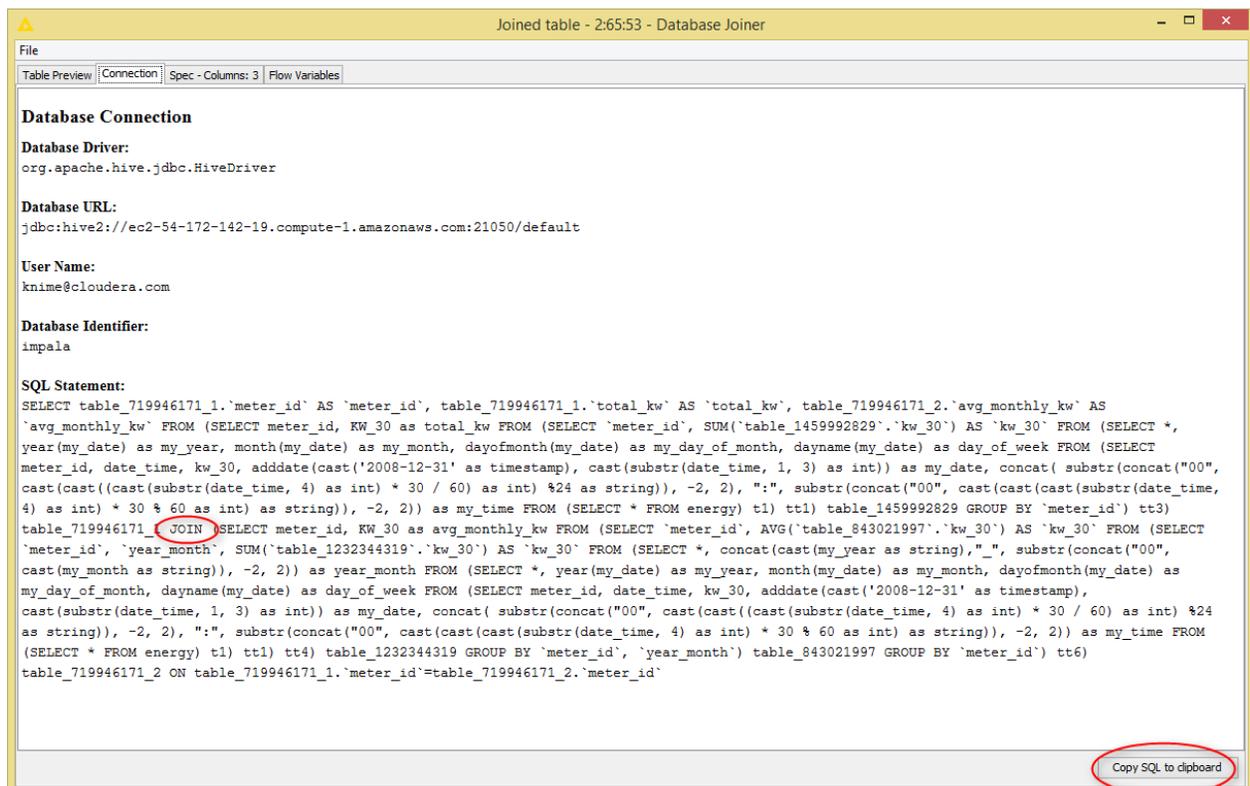
The configuration window of the Database Joiner node resembles the configuration window of the Joiner node, which makes the transition from data manipulation to database query manipulation very easy.

Figure 7. Configuration window of the Database Joiner node



The output port of the Database Joiner node is an SQL query (brown square port), resulting from the combination of the SQL query in the configuration window on top of the input SQL query/queries. Right-clicking the node and selecting the last option of the context menu – the one that usually shows the processed data table – takes you to a table preview and through the Connection tab to the final SQL query. Inside the SQL query you can find the latest add produced by the execution of the current node. In the case of a Database Joiner, this would be a JOIN query.

Figure 8. Output connection of the Database Joiner node

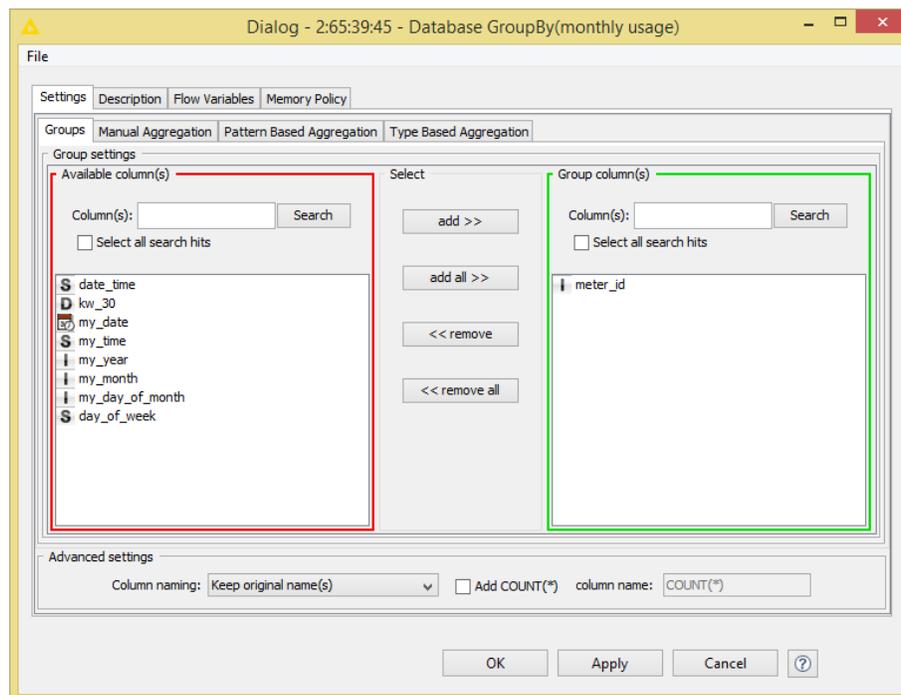


Also, a button named “Copy SQL to clipboard”, at the bottom of the Connection tab, allows the final complex SQL query to be exported to another platform for testing and/or execution.

3. Database GroupBy Node

Another useful node for the SQL non-savvy user is the Database GroupBy node. This node works similarly to the GroupBy node, grouping rows together according to values in selected columns. The only difference between the two GroupBy nodes is in the output results: one produces an aggregated data table (white triangle), the other one an SQL query (brown square), which, if executed, will produce the aggregated data table.

Figure 9. Configuration window of the Database GroupBy node



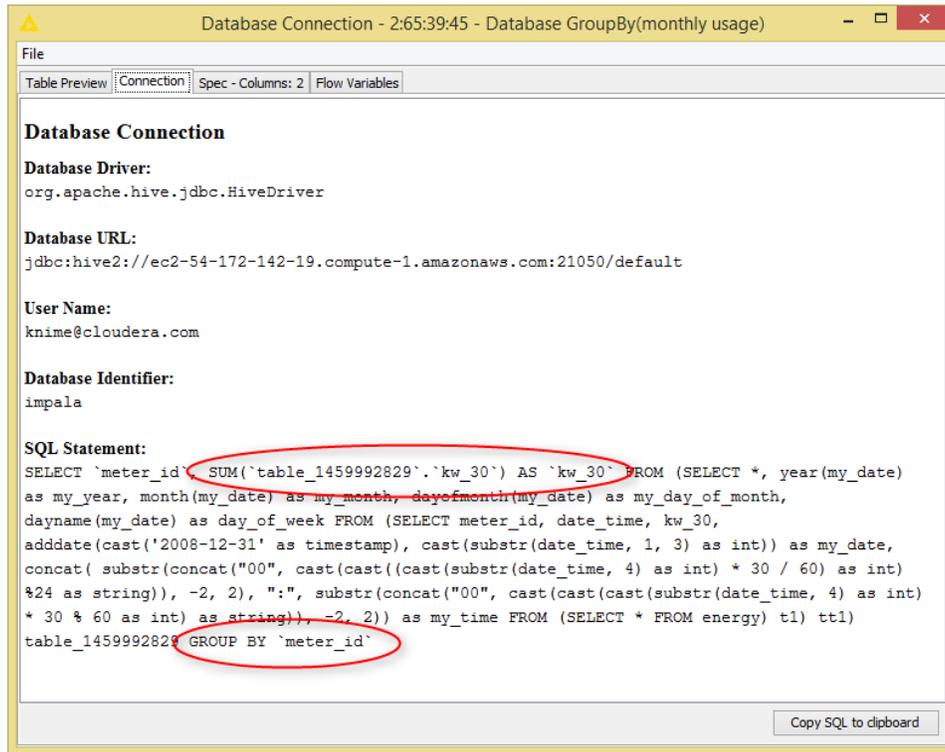
The configuration window of the Database GroupBy node is similar to the configuration window of the GroupBy node. The Groups tab allows the row groups to be defined based on values in the selected columns. The Manual Aggregation tab defines the aggregation method(s) on the aggregation column(s). Two additional tabs, Pattern Based Aggregation and Type Based Aggregation, extend the selection options for the aggregation column(s) and the aggregation method(s).

Notice that the Database GroupBy node offers aggregation methods specific for the big data platform / database it is connected to.

The resulting SQL query piles up on top of the input SQL query by inserting a GROUPBY statement and the corresponding aggregation method statement in the output SQL query.

An example of a Database GroupBy node can be found in the “Total Values” metanode inside the “Daily, Monthly, Yearly, Weekly” metanode in the “004005_Energy_Prepare_Data (Big Data)” workflow.

Figure 10. The SQL query output by a Database GroupBy node. Notice the GROUP BY and SUM statements.



4. Other Database Helper Nodes

Under the Database/Manipulation category, there are a few more nodes to help you build database queries, such as the Database Row/Column Filter node and the Database Sorter node. All three nodes work similarly to a Row/Column Filter or Sorter node, the only difference being that the output result is an SQL query rather than a data table.

A few more database helper nodes can be found under the Database/Utility category. In particular, the SQL Inject node injects a pre-built SQL query, available as a flow variable (red circle port), into the input database connection (red square port). The output is then an SQL query ready to be executed for that database connection.

The SQL Extract node mirrors the SQL Inject node. It reads an SQL query at the input port and copies it into a flow variable and a data table at the output ports.

The Database SQL Executor node finally executes an SQL query on the database connection at the input port without exporting any results to the output port. This can be used to run preliminary SQL statements.

Running the Final SQL Query and Retrieving the Data

All data manipulation nodes from the original “Prepare Data” workflow (Fig. 1) have been translated into SQL queries to execute on a big data platform and packaged into the “004005_Energy_Prepare_Data (Big Data)” workflow (Fig. 2).

The metanode named “Daily, Monthly, Yearly, Weekly” generate the daily, monthly, yearly, weekly time series values of consumed energy in one workflow and the SQL statements necessary to generate them in the other. Similarly, the metanode named “Hourly, Intra-day” generates the SQL statement for the time series of hourly and intra-day time segment consumed energy values. Finally, the metanode named “% Values” calculates or generates the SQL statements to calculate a number of statistical measures about energy usage over time.

However, in contrast to the “Prepare Data” workflow, when using the “004005_Energy_Prepare_Data (Big Data)” workflow we obtain a series of complex SQL statements at the end of all those metanodes. We still need to execute them and retrieve the resulting data from the big data platform.

The node that runs the input SQL statement and retrieves the resulting data is the Database Connection Table Reader. This node takes an SQL statement at its input port, runs it on the database/big data platform connection, and retrieves the resulting data table. Therefore it needs no configuration settings.

If the table to be retrieved is particularly big, the retrieval time might be too long and the connection to the database might timeout. You can increase the timeout for database operations from the Preferences page under KNIME/Database.

Running the new workflow – “004005_Energy_Prepare_Data (Big Data)” –, which creates the SQL queries and retrieves the final data accordingly, takes less than half an hour for all the big data platforms we tested for this project. This is a huge improvement in speed with respect to the almost two days needed for the execution of the original workflow!

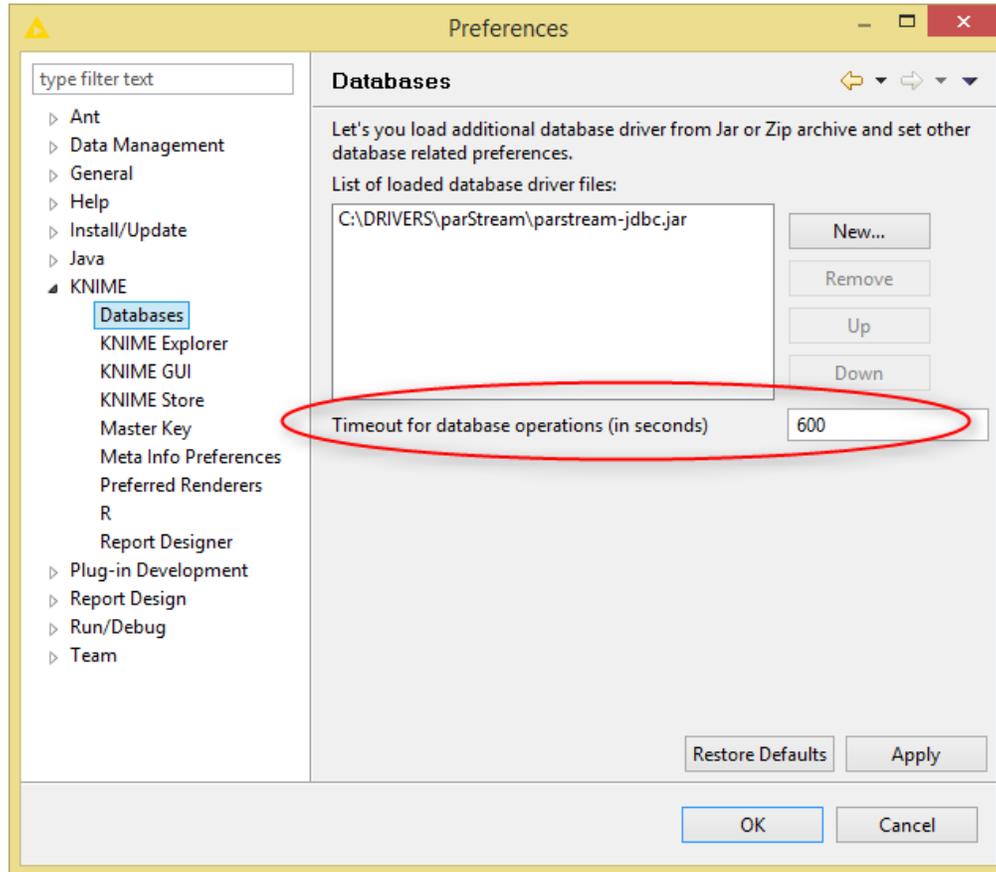
With this workflow, we have prepared the data for further analysis: reading, cleaning, transforming, aggregating, and calculating some descriptive measures. This is already a great step in understanding the underlying system. However, this workflow simply organizes and systematizes the data without really using any intelligence to extract more worthy insights. It is only after introducing data analytics algorithms, from statistics and machine learning, that the application can be made capable of learning and therefore more intelligent.

We tapped into the big data platforms to run the ETL procedures faster and with lower memory requirements. However, we also took advantage of the machine learning algorithms available in KNIME to inject intelligence into the application.

Indeed, after calculating the energy measures, a k-Means algorithm was applied to group together households – i.e. meter IDs – with similar behavior in energy usage. Indeed, one of the resulting clusters includes units using energy mainly during the night hours. Another cluster shows an energy usage time series closely following the business hours. One more cluster shows energy consumption at any time of the day, hinting at family units with components in different age ranges. And so on ...

All in all 30 clusters have been built showing more or less defined energy usage behavior. For each one of these clusters, the average time series of energy usage is calculated across all meter IDs included in the cluster. Finally, time series analysis is applied to the 30 resulting average time series in order to predict future peaks or shortage in energy production.

Figure 11. Parameter controlling Timeout Time for database operations in the KNIME Preferences page



Conclusions

In this short whitepaper, we have shown how to integrate a big data platform into a KNIME workflow in order to delegate the execution of ETL operations on a huge amount of data. The integration process is very straightforward.

1. Drag & drop the appropriate connector node into the workflow to connect to the big data platform of choice
2. Configure the connector node with the parameters required to access the data on the big data platform, i.e. credentials, server URL, and other platform specific settings
3. Define the SQL query to perform the ETL operations with the help of SQL manipulation nodes. The SQL manipulation nodes indeed help you build the correct SQL query without needing to be knowledgeable about SQL queries.
4. Finally, the execution of a data retrieval node (Database Connection – Table Reader node) allows the user to retrieve the data using the previously built SQL query.

Such an easy approach opens the door to the introduction of big data platforms into KNIME, without the headache of configuring each tiny platform detail. It also preserves the quick prototyping feature of a KNIME workflow. Indeed, the user can change the big data platform of choice, just by changing the database connector node in step 1 and reconnecting it to the subsequent SQL builder nodes.

It is really that easy to integrate big data platforms in KNIME and to considerably speed up the whole ETL part of data science discovery travel!

In this whitepaper we rescued an ETL workflow developed in 2013 for the analysis of energy usage time series (see KNIME whitepaper: “[Big Data, Smart Meters, and Predictive Analytics](#)”). Data about energy usage had been collected for more than a year through smart meters in an experimental project in Ireland between 2009 and 2011.

Introducing big data platforms into the workflow for the execution of the ETL operations reduced the execution time from almost two days to less than half an hour!

The workflow used in this whitepaper is available on the EXAMPLES server under 004_Databases/004005_Energy_Prepare_Data (Big Data). A sample of the original data can be retrieved from www.knime.com/files/reducedenergydata.zip.

Please remember that we are not allowed to distribute the original dataset. The entire dataset must be requested directly from the Irish Social Science Data Archive reported below: http://www.ucd.ie/t4cms/CER_energy_electric_issda-data-request-form.docx.

KNIME Analytics Platform can be downloaded for free from www.knime.org/download and KNIME Big Data Extensions can be purchased at <http://www.knime.org/knime-big-data-extensions>.