Open for Innovation
KNIME

Welcome to
KNIME Big Data Workshop

Going live at:

Berlin 5:00 PM (CEST)

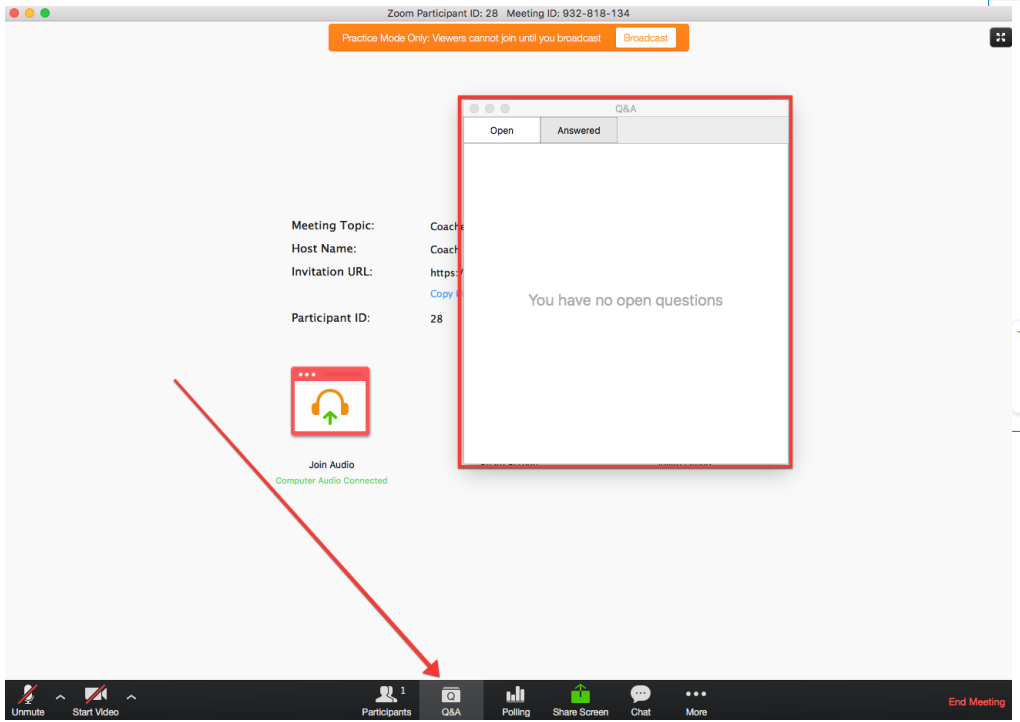
New York City 11:00 AM (EDT)

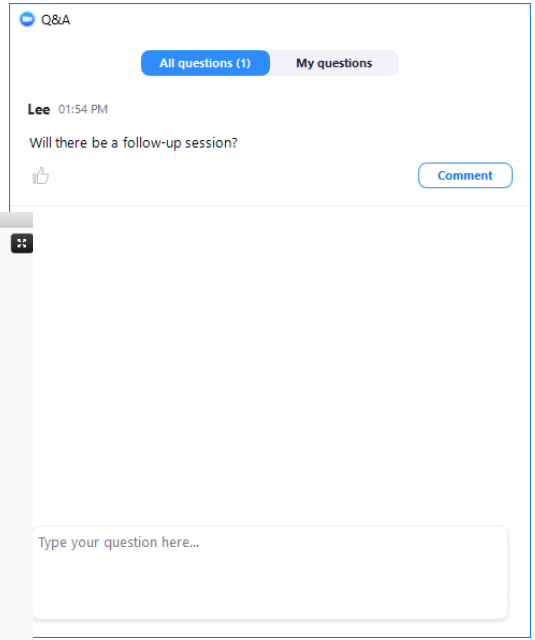
Austin 10:00 AM (CDT)

London 4:00 PM (GMT)

Before we start…

- Please use the Q&A section to post your questions.
- Upvote for your favorite questions.
- Session is recorded and will be available on YouTube.





Open for Innovation KNIME

Agenda

- What is "big data"?
- KNIME Big Data Extensions
  - Introduction to Hadoop and Spark
  - KNIME Big Data Connectors
  - KNIME Extension for Apache Spark
  - KNIME H2O Sparkling Water Integration
  - KNIME Workflow Executor for Apache Spark

What is "Big Data" about?

1

"...ways to analyze, systematically extract information from [...] data sets that are too large or complex to be dealt with by traditional data-processing application software." [1]

2

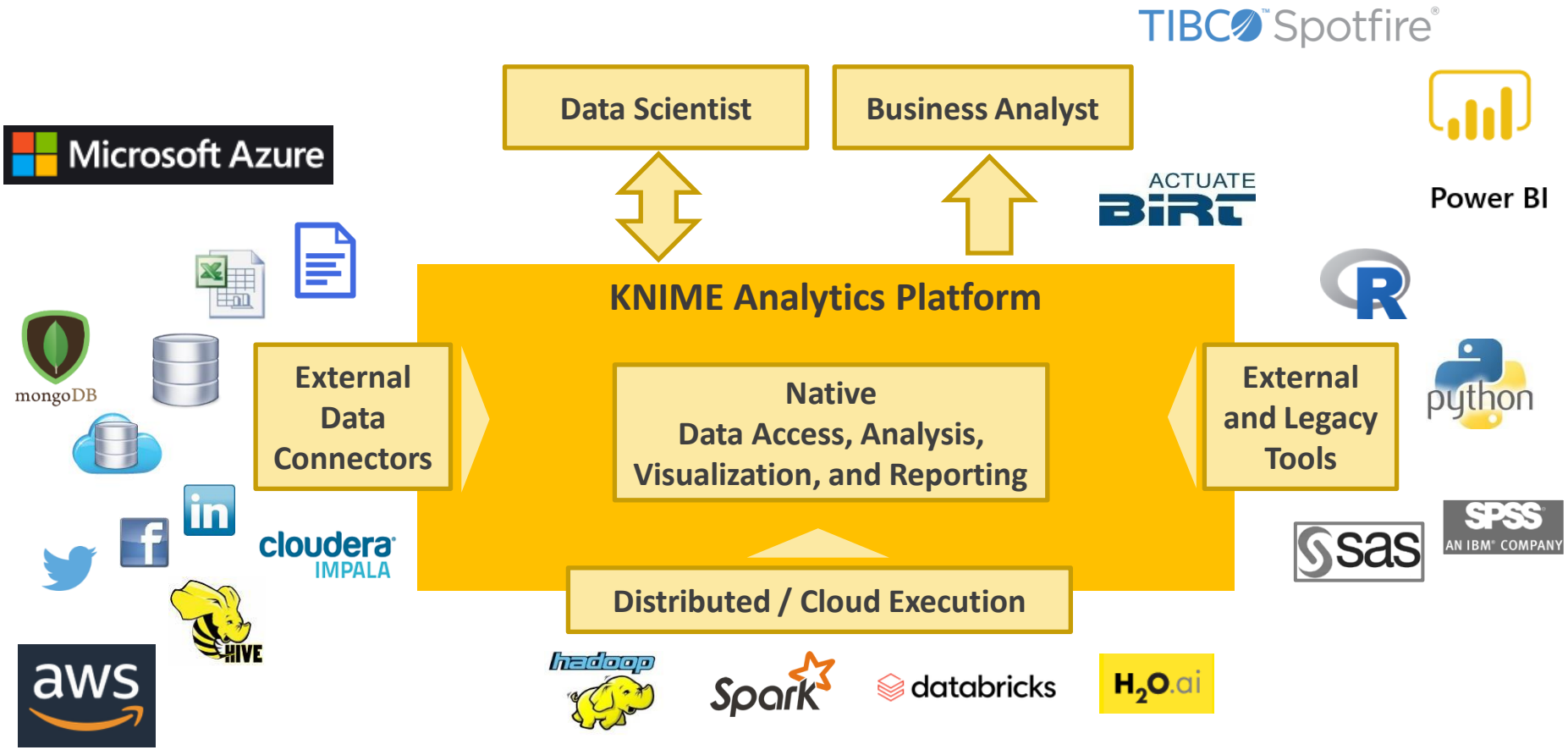
The three Vs of what makes data "big":

- Volume (size of data)
- Variety (tabular, text, images, video, audio, time series, ...)
- Velocity (produced fast and continuously)

Goal of big data technologies:
Enable predictive or other types of advanced analytics to extract **value** from big data.

[1] https://en.wikipedia.org/wiki/Big_data

KNIME Analytics Platform: Open for Every Data, Tool, and User



KNIME Big Data Extensions

- Introduction to Hadoop and Spark
- KNIME Big Data Connectors
- KNIME Extension for Apache Spark
- KNIME H2O Sparkling Water Integration
- KNIME Workflow Executor for Apache Spark

Apache Hadoop

- Open-source project for distributed storage and processing of large data sets
- Designed to scale up to thousands of machines

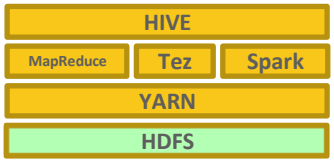
- First release in 2006
  - Rapid adoption, promoted to top level Apache project in 2008
  - Inspired by Google File System (2003) paper
- Spawned diverse ecosystem of products

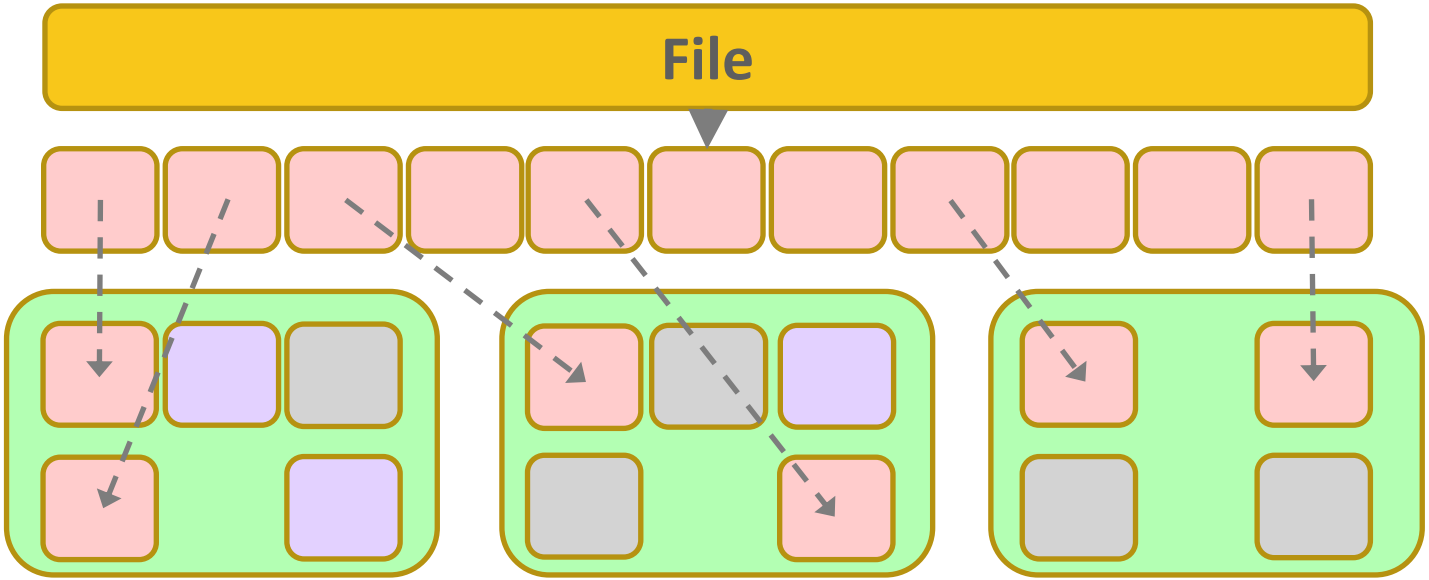
Hadoop Ecosystem

SQL	HIVE		
Processing	MapReduce	Tez	Spark
Resource Management	YARN		
Storage	HDFS		

HDFS

- Hadoop distributed file system
- Stores large files across multiple machines

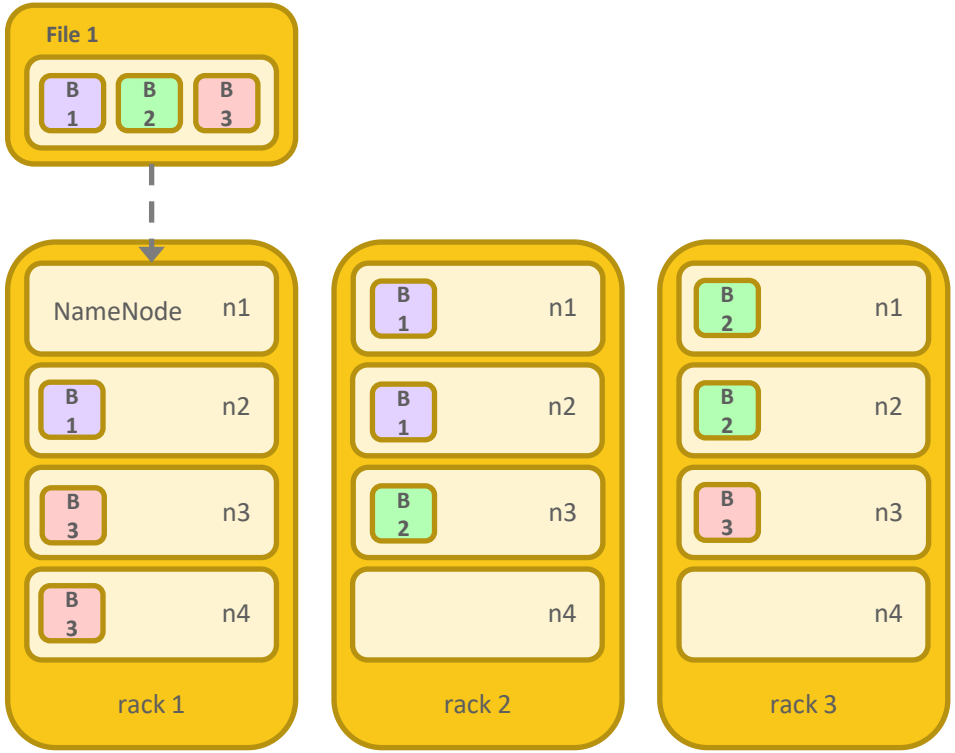




HDFS – Data Replication and File Size

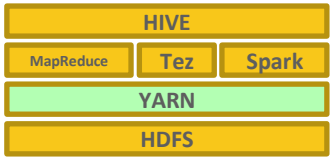
Data Replication

- All blocks of a file are stored as sequence of blocks
- Blocks of a file are replicated for fault tolerance (usually 3 replicas)
  - improves data reliability, availability, and network bandwidth utilization

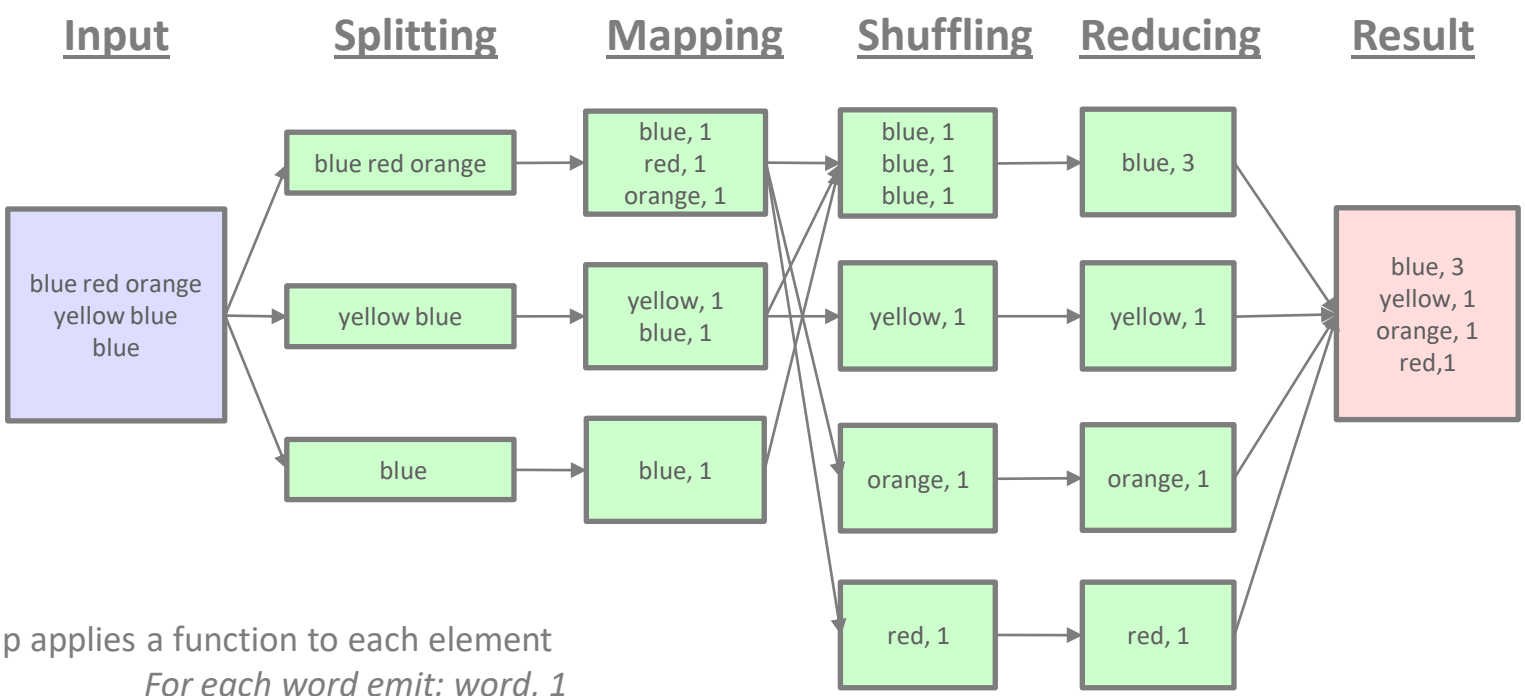


YARN

- Cluster resource management system
- Two elements
  - Resource manager (one per cluster):
    - Knows where worker nodes are located and how many resources they have
    - Scheduler: Decides how to allocate resources to applications
  - Node manager (many per cluster):
    - Launches application containers
    - Monitor resource usage and report to Resource Manager



MapReduce

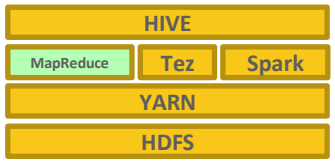


Map applies a function to each element

For each word emit: word, 1

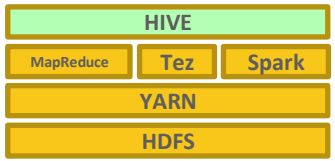
Reduce aggregates a list of values to one result

For all equal words sum up count

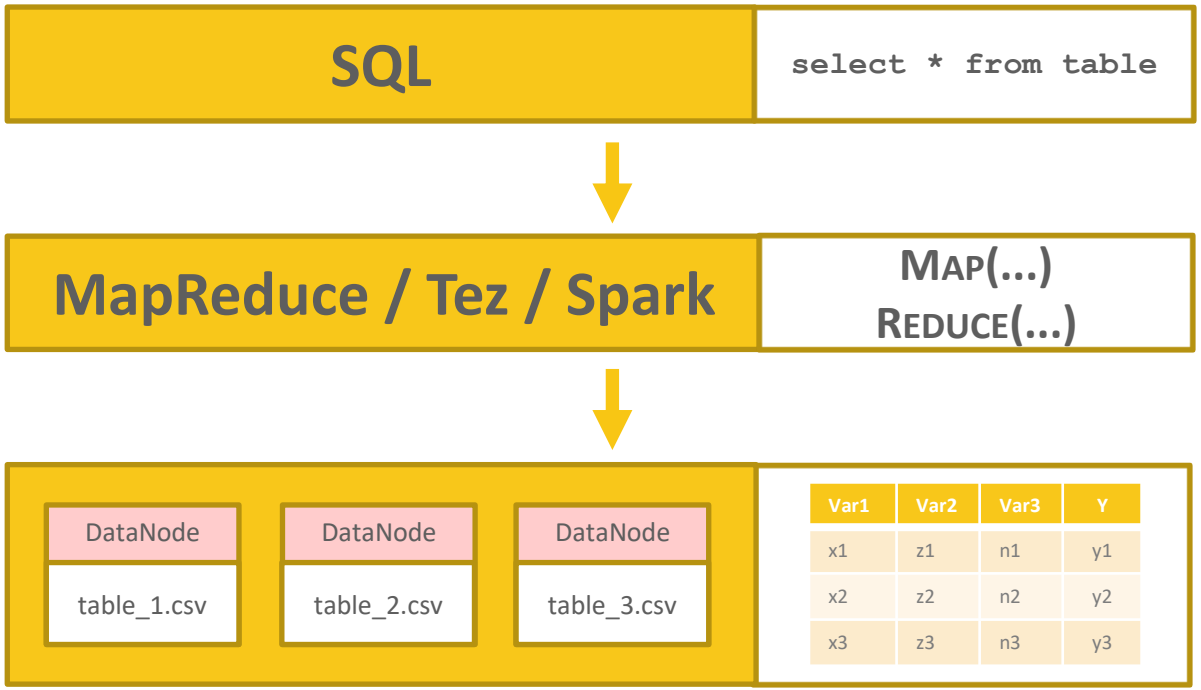


Hive

- **SQL database** on top of files in HDFS
- Provides data summarization, query, and analysis
- Interprets a set of files as a database table (schema information to be provided)
- Translates SQL queries to MapReduce, Tez, or Spark jobs
- Supports various file formats:
  - Text/CSV
  - SequenceFile
  - Avro
  - ORC
  - Parquet

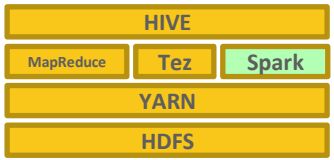


Hive



Spark

- Cluster computing framework for large-scale data processing
- In-memory computing
  - much (!) faster than MapReduce
- Programmatic interface (Scala, Java, Python, R)
- Great for:
  - Iterative algorithms
  - Interactive data analysis



Spark – Data Representation

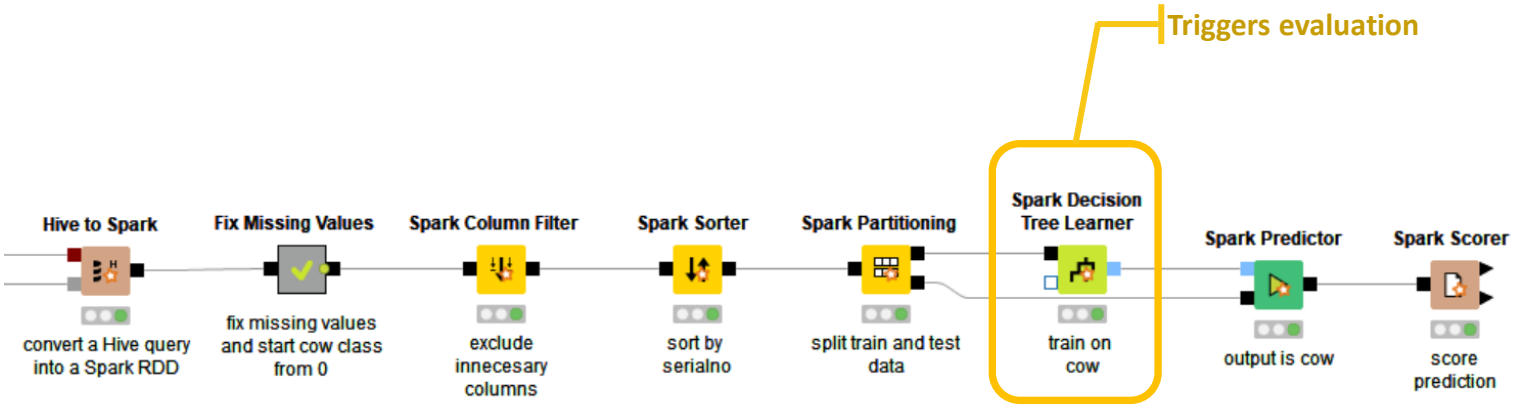
DataFrame:

Name	Surname	Age
John	Doe	35
Jane	Roe	29
…	…	…

- *Table-like*: Collection of rows, organized in columns with names and types
- *Immutable*:
  - Data manipulation = creating new DataFrame from an existing one by applying a *function* on it
- *Lazily evaluated*:
  - Functions are not executed until an *action* is triggered, that requests to actually see the row data
- *Distributed:*
  - Each row belongs to exactly one *partition*
  - Each partition is held by a Spark Executor

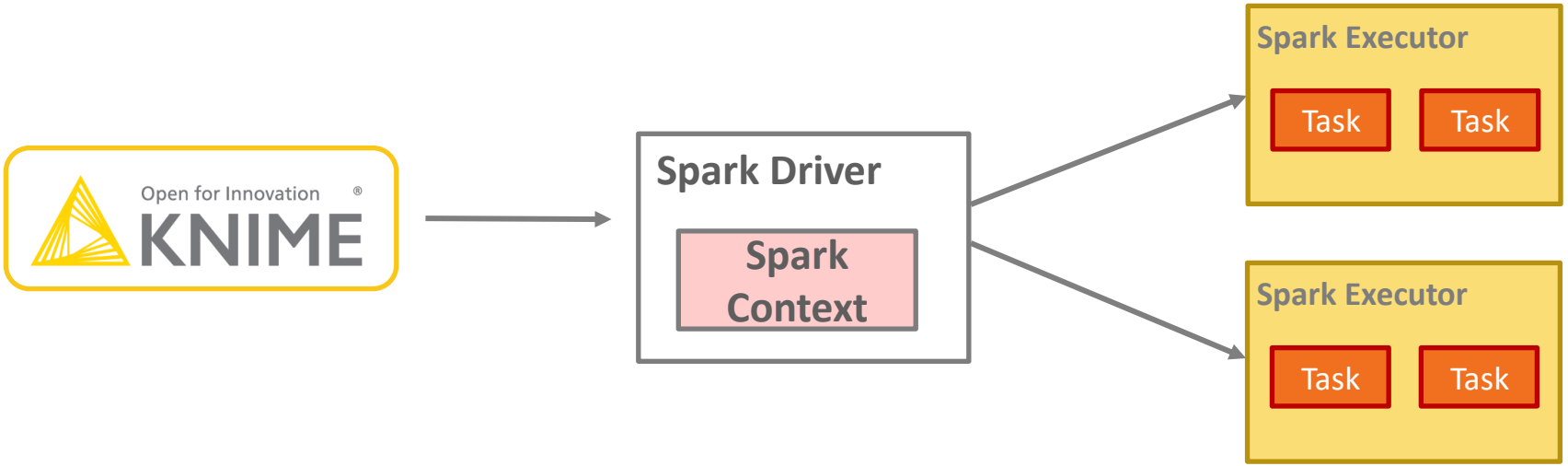
Spark – Lazy Evaluation

- Functions ("*transformations*") on DataFrames are not executed immediately
- Spark keeps record of the transformations for each DataFrame
- The actual execution is only triggered once the data is needed
- Offers the possibility to optimize the transformation steps



Spark Context

- Spark Context
  - Main entry point for Spark functionality
  - Represents connection to a Spark cluster
  - Allocates resources on the cluster

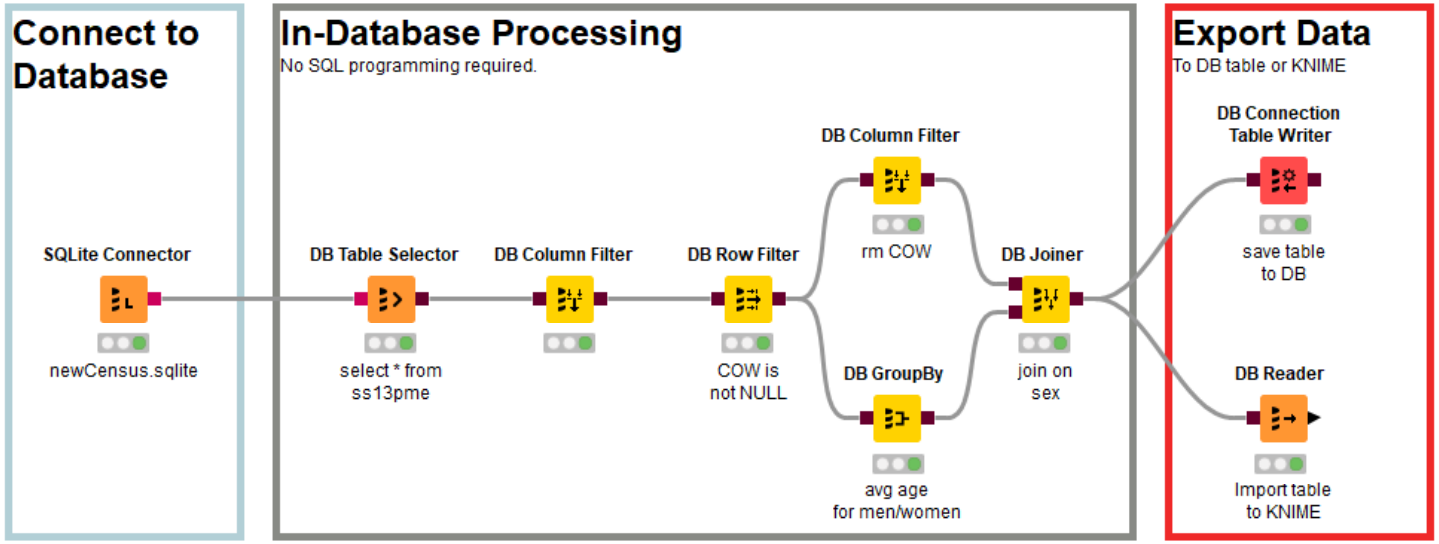


KNIME Big Data Extensions

- Introduction to Hadoop and Spark
- KNIME Big Data Connectors
- KNIME Extension for Apache Spark
- KNIME H2O Sparkling Water Integration
- KNIME Workflow Executor for Apache Spark

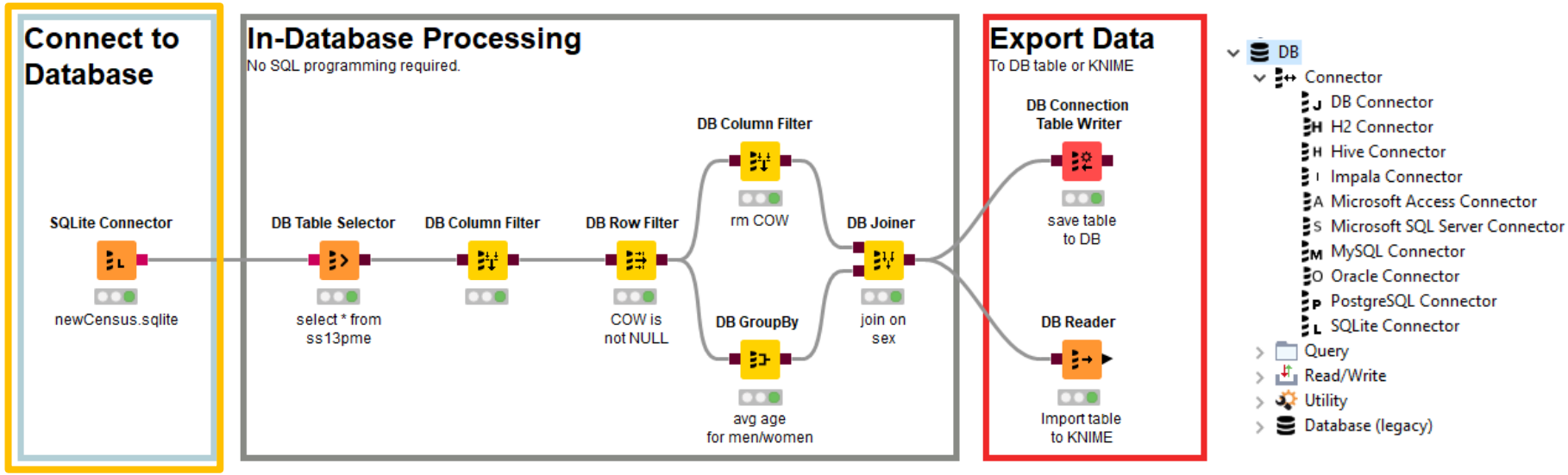
Database Extension

- Visually assemble complex SQL statements (no SQL coding needed)
- Connect to all JDBC-compliant databases
- Harness the power of your database within KNIME



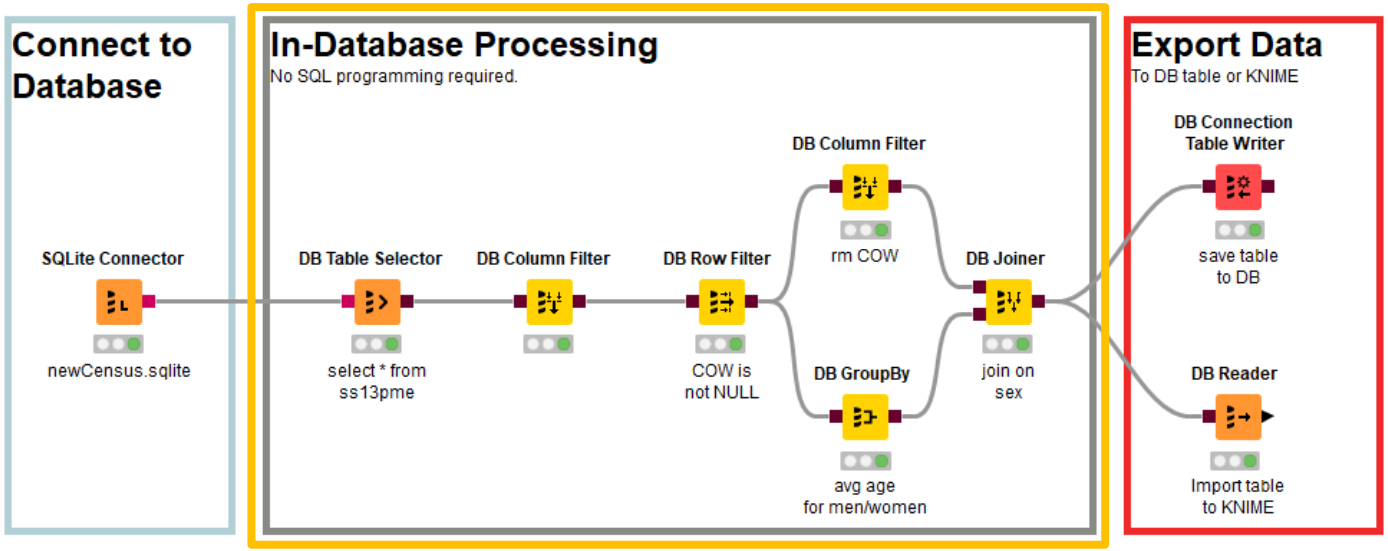
Database Connectors

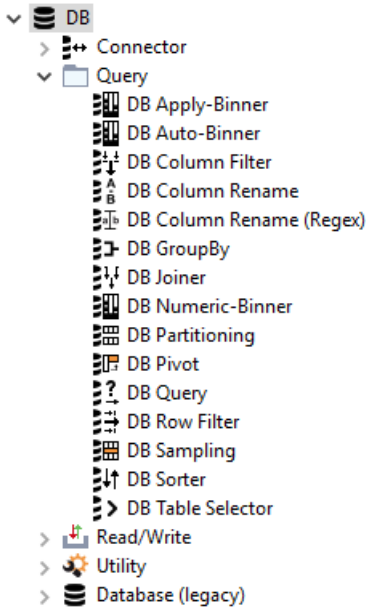
- Many dedicated DB Connector nodes available
- If connector node missing, use DB Connector node with JDBC driver



In-Database Processing

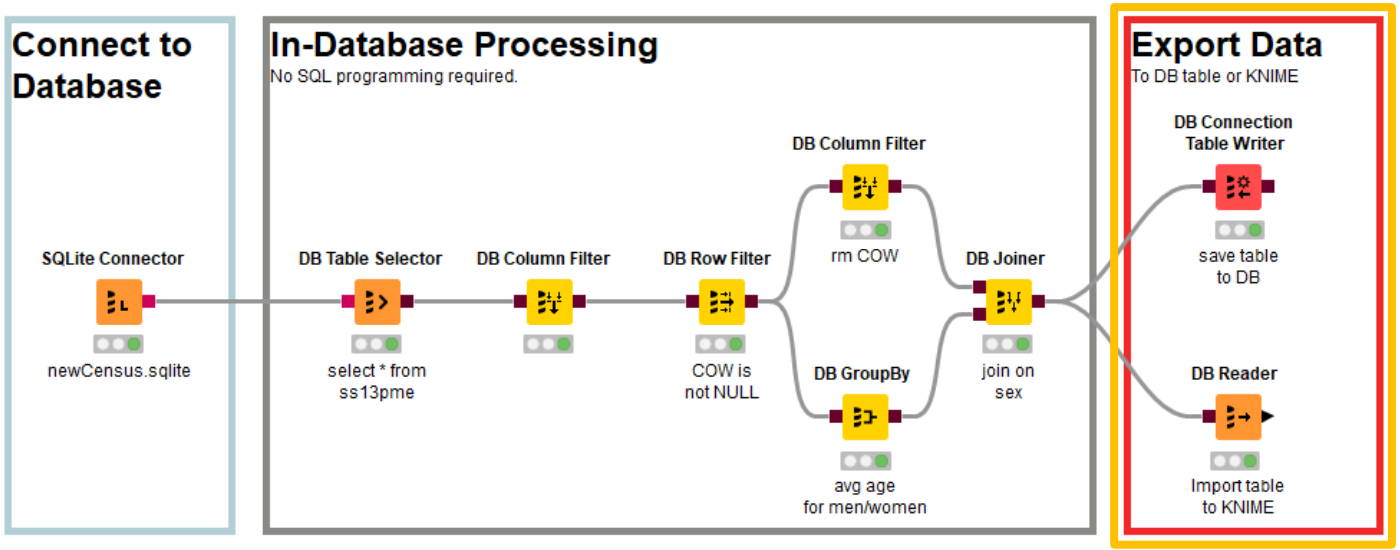
- Database Manipulation nodes generate SQL query on top of the input SQL query (brown square port)
- SQL operations are **executed on the database**!

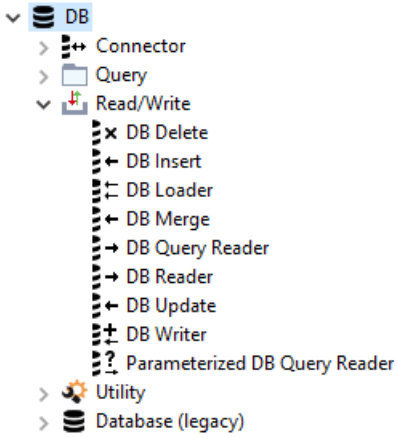




Export Data

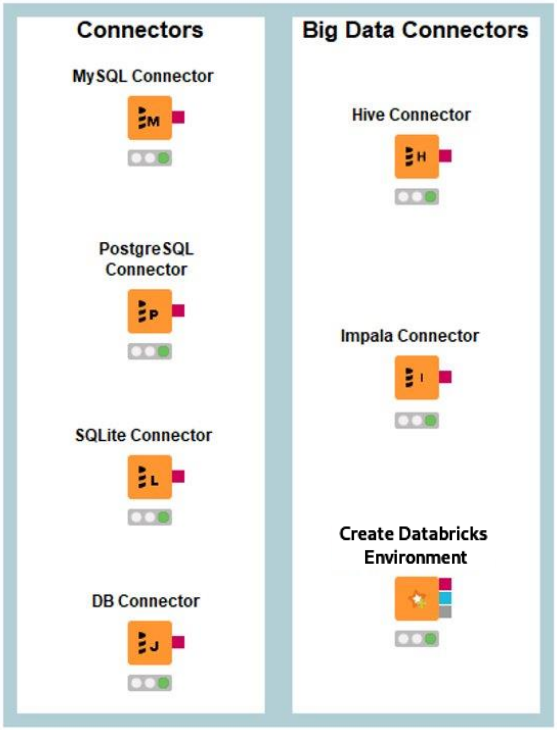
- Writing data back into database
- Exporting data into KNIME





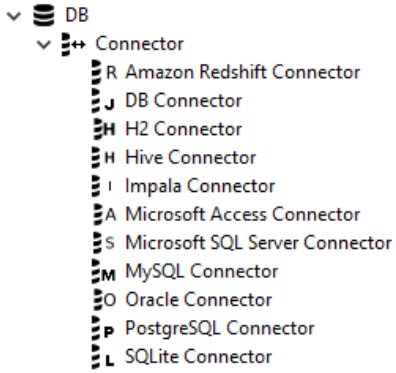
KNIME Big Data Connectors

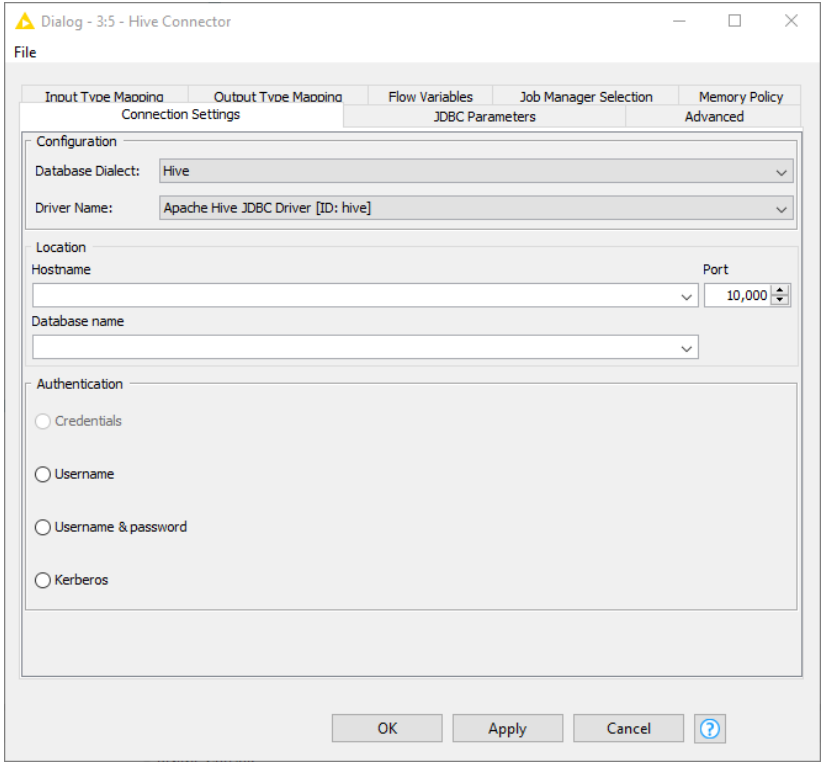
- Built upon Database extension
- Include drivers/libraries for HDFS, Hive, Impala and Databricks
- Preconfigured connectors
  - Hive
  - Impala
  - Databricks (Thriftserver)



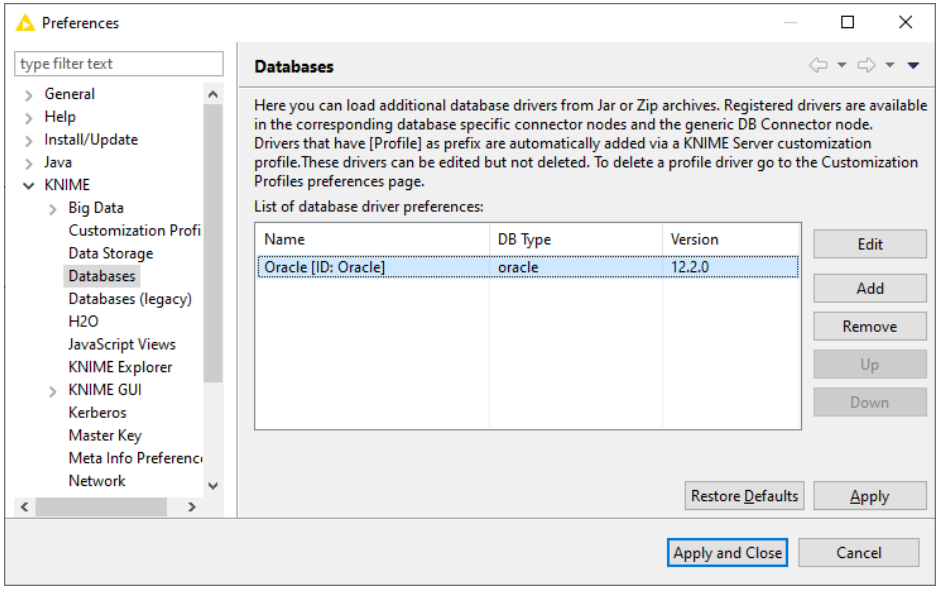
Hive Connector

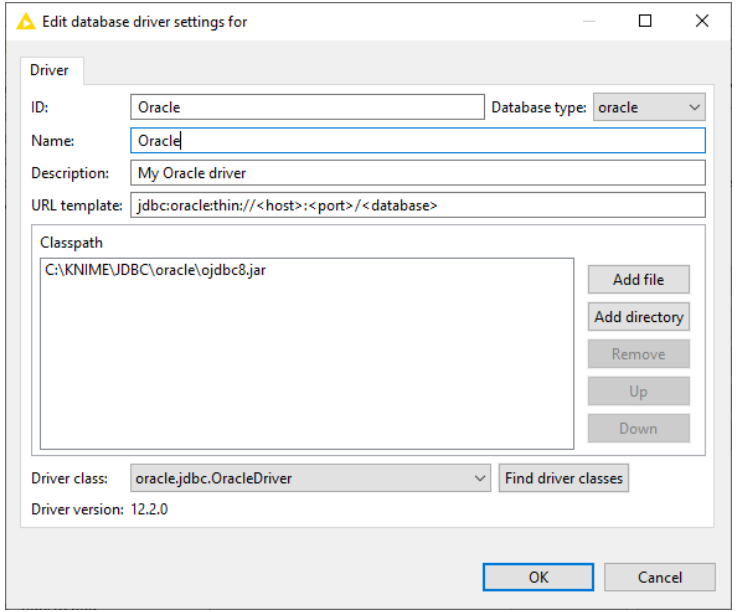
- Creates JDBC connection to Hive
- On unsecured clusters no password required





Preferences: Registering proprietary JDBC drivers



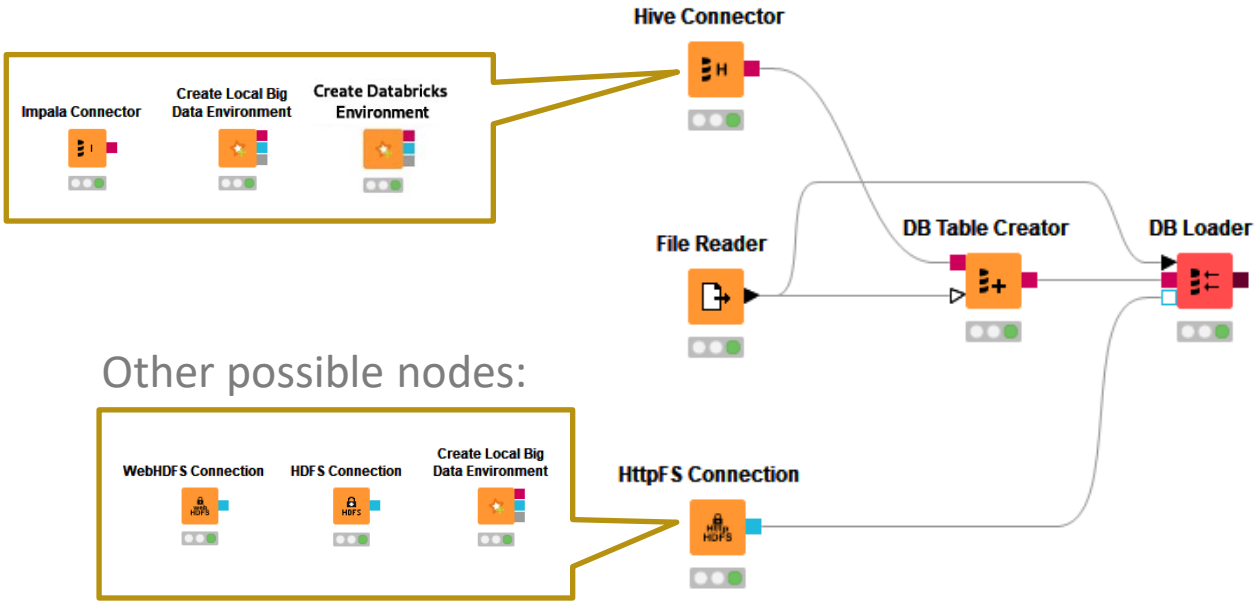


Useful for:

- Cloudera Hive/Impala JDBC drivers
- Databricks JDBC Driver

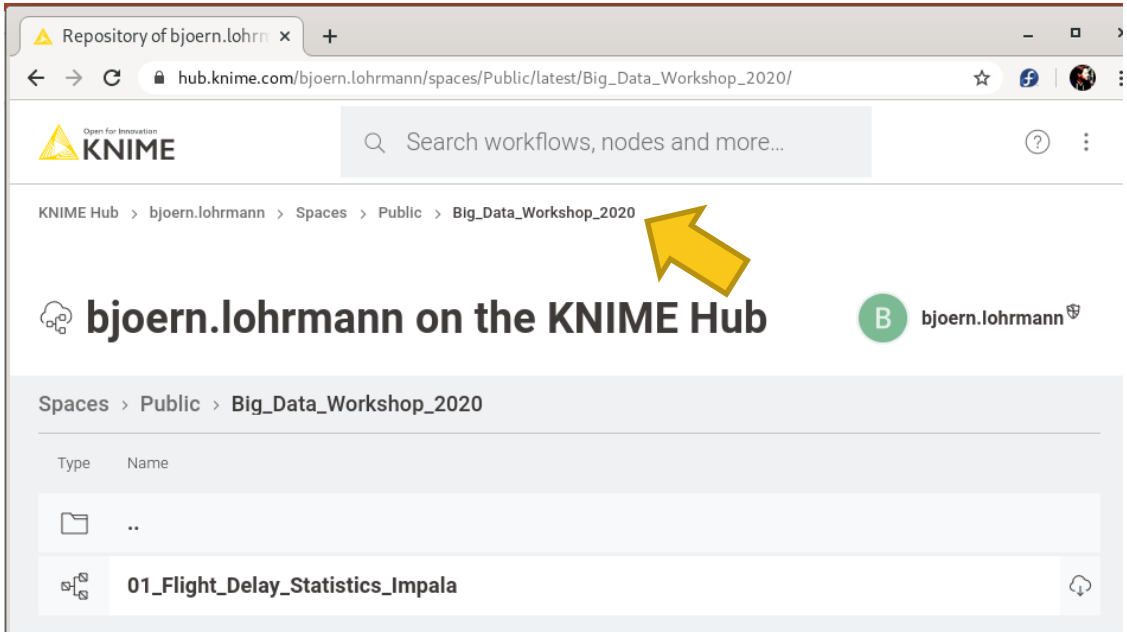
Loading Data into Hive/Impala

- Connectors are from KNIME Big Data Connectors Extension
- Use DB Table Creator and DB Loader from regular DB framework



Demo: 01_Flight_Delay_Statistics_Impala

Shortened URL to KNIME Hub folder: https://tinyurl.com/ycyh3kya



KNIME Big Data Extensions

- Introduction to Hadoop and Spark
- KNIME Big Data Connectors
- KNIME Extension for Apache Spark
- KNIME H2O Sparkling Water Integration
- KNIME Workflow Executor for Apache Spark

KNIME Extension for Apache Spark

- Based on Spark MLlib
- Scalable machine learning library
- Supports algorithms for
  - Classification (decision tree, naïve bayes, …)
  - Regression (logistic regression, linear regression, …)
  - Clustering (k-means)
  - Collaborative filtering (ALS)
  - Dimensionality reduction (SVD, PCA)

Spark Integration in KNIME

- Apache Spark
  - IO
  - Column
  - Mining
  - Misc
  - Row
  - Statistics
  - Create Databricks Environment
  - Create Local Big Data Environment
  - Create Local Big Data Environment (Legacy
  - Create Spark Context (Jobserver)
  - Create Spark Context (Livy)
  - Destroy Spark Context

- Apache Spark
  - IO
    - Database
      - DB to Spark
      - Database to Spark (legacy)
      - Hive to Spark
      - Hive to Spark (legacy)
      - Impala to Spark
      - Impala to Spark (legacy)
      - Spark to DB
      - Spark to Database (legacy)
      - Spark to Hive
      - Spark to Hive (legacy)
      - Spark to Impala
      - Spark to Impala (legacy)
    - Read
      - Avro to Spark
      - CSV to Spark
      - JSON to Spark
      - ORC to Spark
      - Parquet to Spark
      - Table to Spark
      - Text to Spark
    - Write
      - Spark to Avro
      - Spark to CSV
      - Spark to JSON
      - Spark to ORC
      - Spark to Parquet
      - Spark to Table
      - Spark to Text
    - Persist Spark DataFrame/RDD
    - Unpersist Spark DataFrame/RDD
  - Column
    - Convert & Replace
      - Spark Category To Number
      - Spark Column Rename
      - Spark Column Rename (Regex)
      - Spark Compiled Transformations Applier
      - Spark Normalizer
      - Spark Number To Category (Apply)
      - Spark Transformations Applier
    - Split & Combine
      - Spark Joiner
    - Transform
      - Spark Missing Value
      - Spark Missing Value (Apply)
    - Spark Column Filter

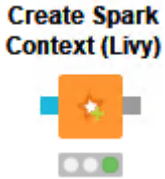
- Mining
  - Clustering
    - Spark Cluster Assigner
    - Spark k-Means
  - Dimensionality Reduction
    - Spark PCA
    - Spark SVD
  - Item Sets / Association Rules
    - Spark Association Rule (Apply)
    - Spark Association Rule Learner
    - Spark Frequent Item Sets
  - PMML
    - Spark Compiled Model Predictor
    - Spark MLlib to PMML
    - Spark PMML Model Predictor
  - Prediction
    - Spark Decision Tree Learner
    - Spark Decision Tree Learner (MLlib)
    - Spark Decision Tree Learner (Regression)
    - Spark Gradient Boosted Trees Learner
    - Spark Gradient Boosted Trees Learner (Regression)
    - Spark Gradient-Boosted Trees Learner (MLlib)
    - Spark Linear Regression Learner (MLlib)
    - Spark Linear SVM Learner (MLlib)
    - Spark Logistic Regression Learner (MLlib)
    - Spark Naive Bayes Learner (MLlib)
    - Spark Predictor (Classification)
    - Spark Predictor (MLlib)
    - Spark Predictor (Regression)
    - Spark Random Forest Learner
    - Spark Random Forest Learner (Regression)
    - Spark Random Forests Learner (MLlib)
  - Scoring
    - Spark Entropy Scorer
    - Spark Numeric Scorer
    - Spark Scorer
  - Spark Collaborative Filtering Learner (MLlib)

- Misc
  - Java Snippet
    - Spark DataFrame Java Snippet
    - Spark DataFrame Java Snippet (Sink)
    - Spark DataFrame Java Snippet (Source)
    - Spark RDD Java Snippet
    - Spark RDD Java Snippet (Sink)
    - Spark RDD Java Snippet (Source)
  - Management
    - List Spark DataFrames/RDDs
  - PySpark
    - PySpark Script (1 to 1)
    - PySpark Script (1 to 2)
    - PySpark Script (2 to 1)
    - PySpark Script (2 to 2)
    - PySpark Script Source
  - Spark Repartition
  - Spark SQL Query
- Row
  - Spark Concatenate
  - Spark GroupBy
  - Spark Partitioning
  - Spark Pivot
  - Spark Row Filter
  - Spark Row Sampling
  - Spark Sorter
- Statistics
  - Spark Correlation Filter
  - Spark Correlation Matrix
  - Spark Linear Correlation
  - Spark Statistics

Spark Contexts: Creating

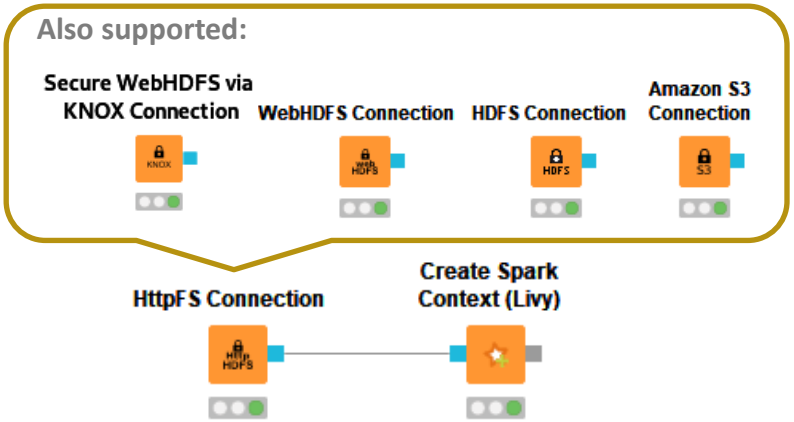
Three nodes to create a Spark context:

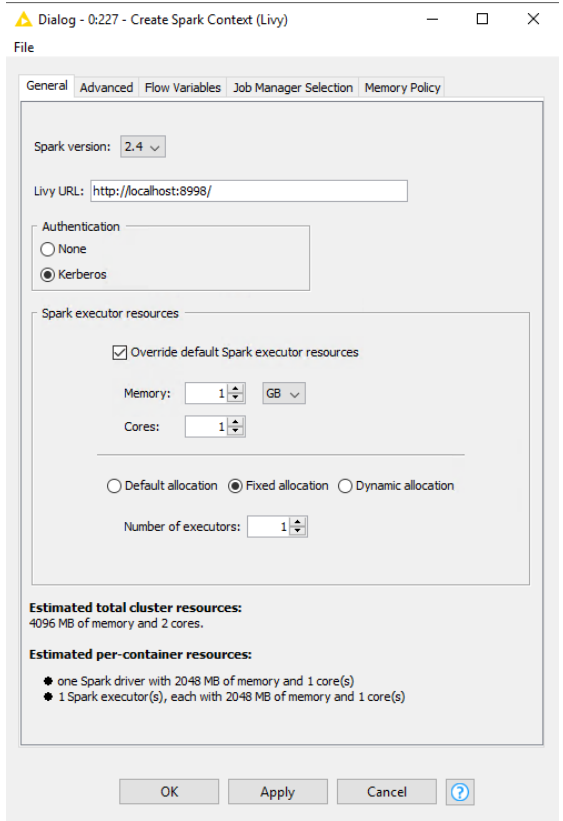
- Create Local Big Data Environment
  - Runs Spark locally on your machine (no cluster required)
  - Good for workflow prototyping
- Create Spark Context (Livy)
  - Requires a cluster that provides the Livy service
  - Good for production use
- Create Databricks Environment
  - Requires a Databricks cluster on AWS or Azure
  - Provides DB connection, DBFS and Spark



Create Spark Context (Livy)

- Allows to use Spark nodes on clusters with Apache Livy
- Compatible with CDH, HDP, HDInsight and EMR

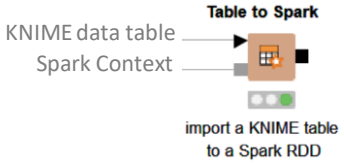




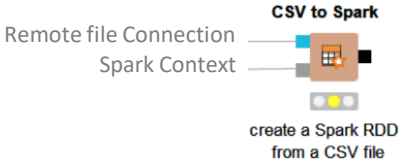
Import Data to Spark

- Tools & Services
  - Apache Spark
    - IO
      - Database
        - Database to Spark
        - Hive to Spark
        - Impala to Spark
        - Spark to Database
        - Spark to Hive
        - Spark to Impala
      - Read
        - Avro to Spark
        - CSV to Spark
        - JSON to Spark
        - ORC to Spark
        - Parquet to Spark
        - Table to Spark
        - Text to Spark
      - Write
        - Spark to Avro
        - Spark to CSV
        - Spark to JSON
        - Spark to ORC
        - Spark to Parquet
        - Spark to Table
        - Spark to Text
      - Persist Spark DataFrame/RDD
      - Unpersist Spark DataFrame/RDD

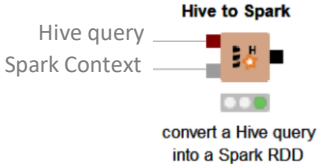
- From KNIME



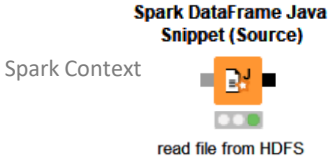
- From CSV file in HDFS



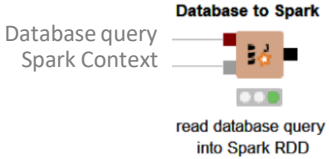
- From Hive



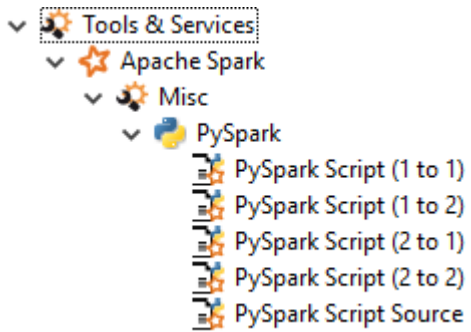
- From other sources

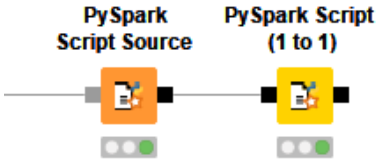


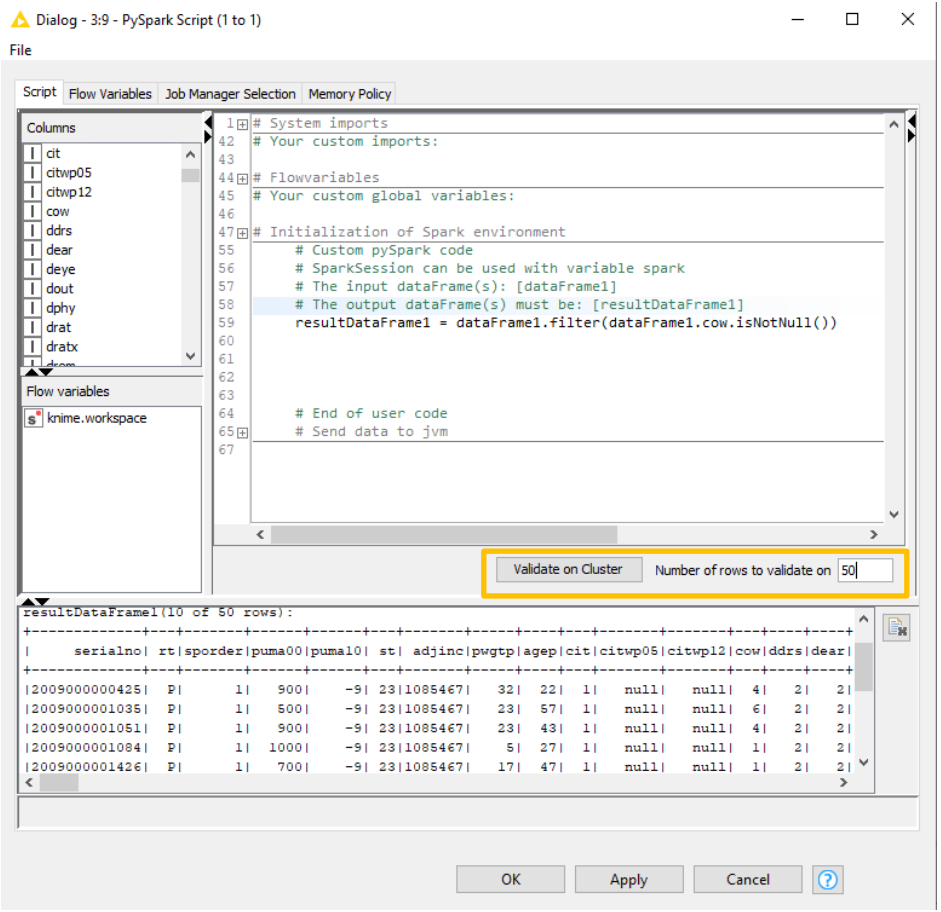
- From Database



Modularize and Execute Your Own Spark Code: PySpark Script

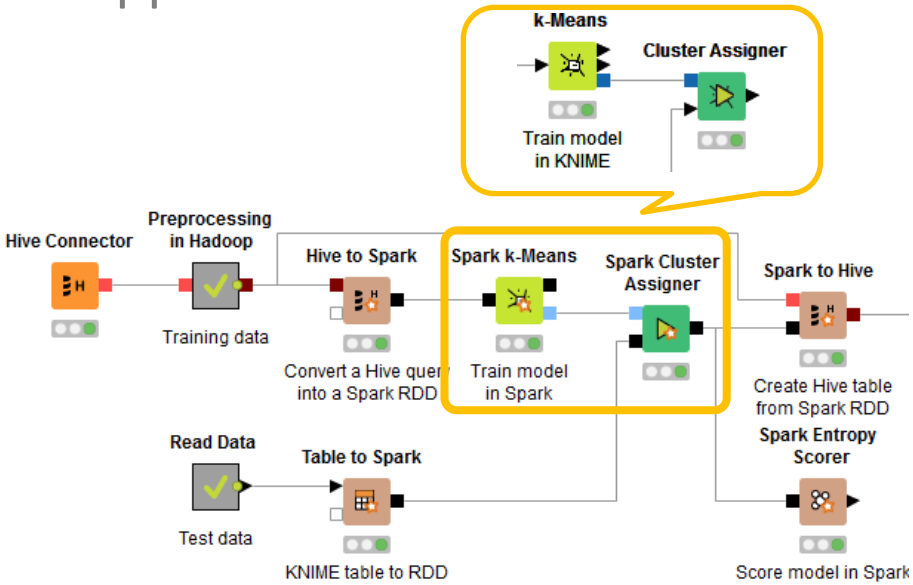


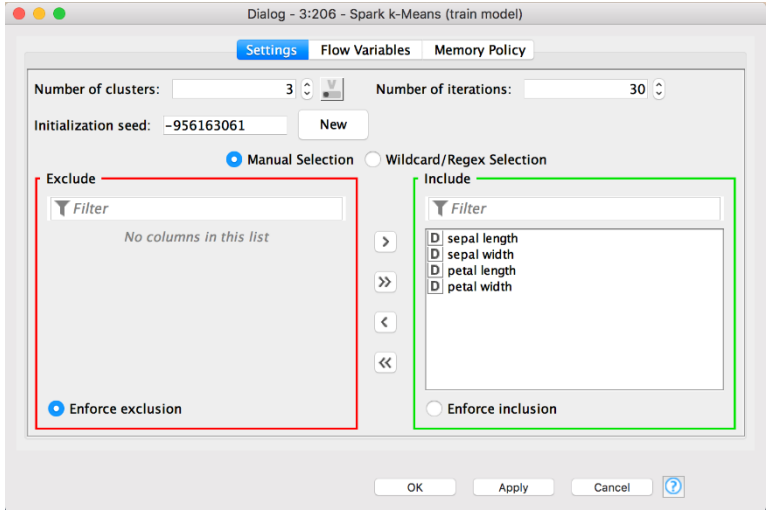




MLlib Integration: Familiar Usage Model

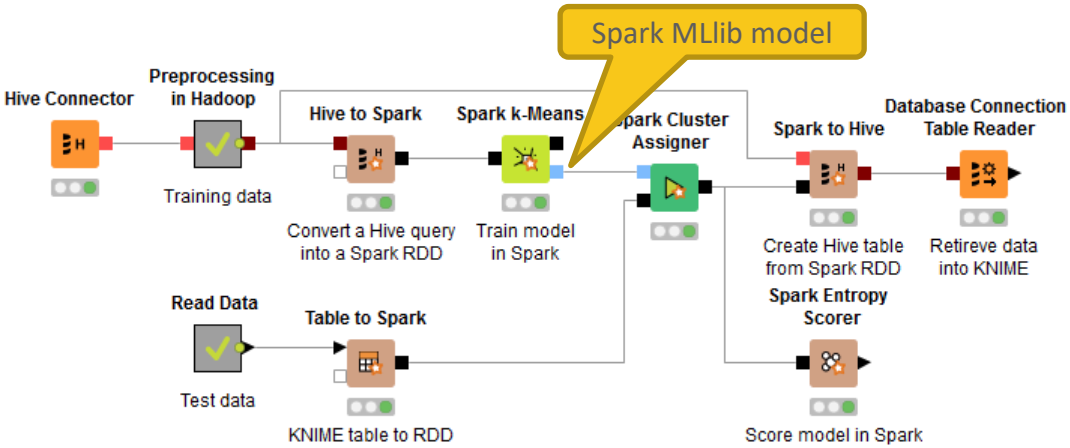
- Usage model and dialogs similar to existing nodes
- No coding required
- Various algorithms for classification, regression and clustering supported





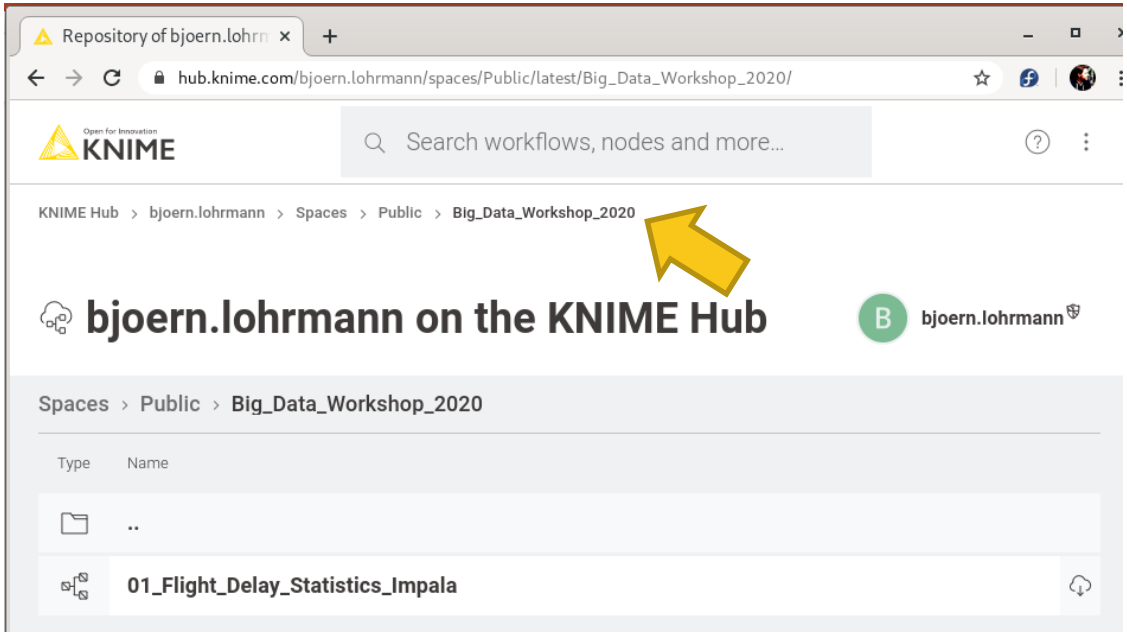
MLlib Integration: Spark MLlib Model Port

- MLlib model ports for model transfer
- Model ports provide more information about the model itself



Demo: 02_Taxi_Demand_Prediction_Training_workflow

Shortened URL to KNIME Hub folder: https://tinyurl.com/ycyh3kya

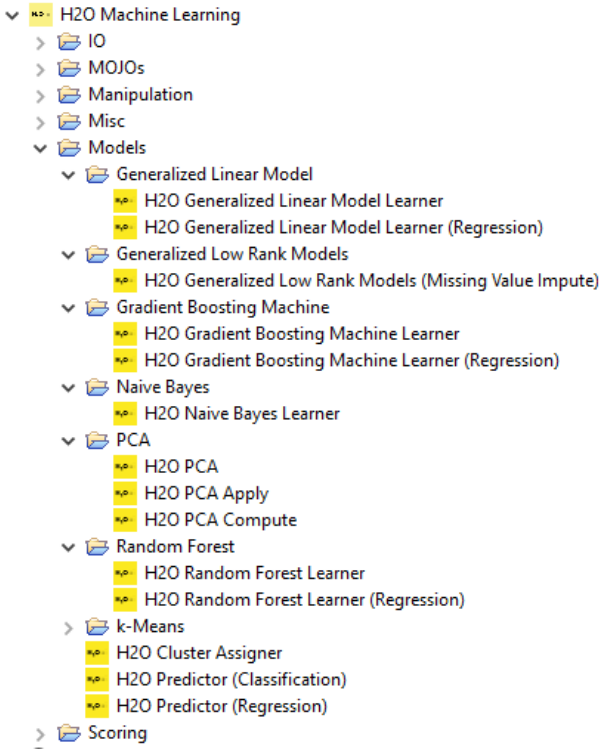


KNIME Big Data Extensions

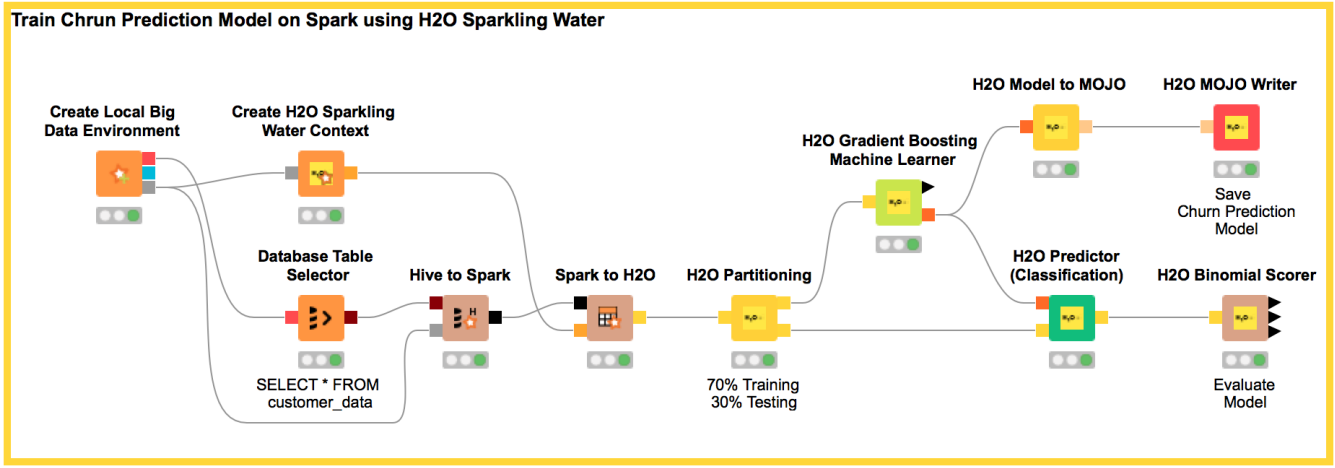
- Introduction to Hadoop and Spark
- KNIME Big Data Connectors
- KNIME Extension for Apache Spark
- KNIME H2O Sparkling Water Integration
- KNIME Workflow Executor for Apache Spark

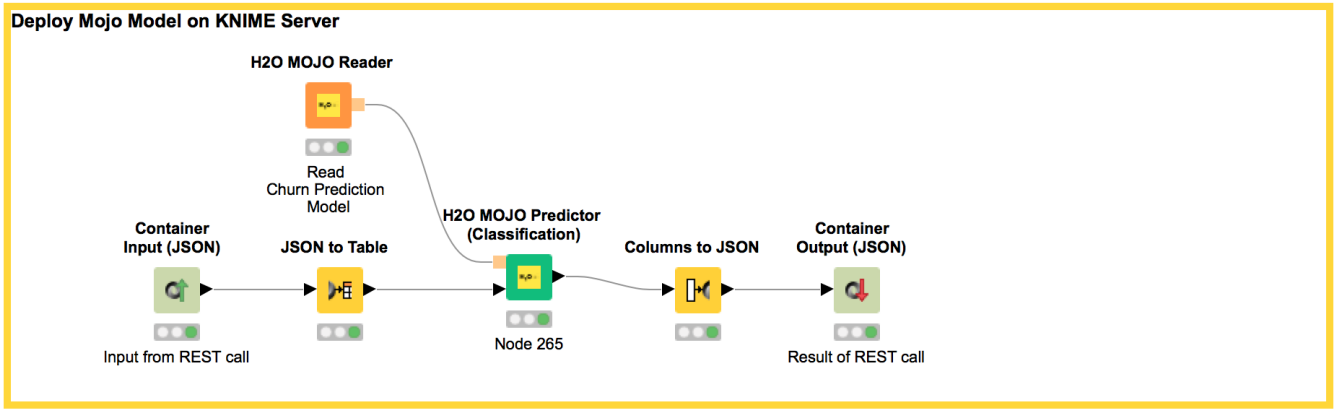
H2O Integration

- KNIME integrates the H2O machine learning library
- H2O: Open source, focus on scalability and performance
- Supports many different models
  - Generalized Linear Model
  - Gradient Boosting Machine
  - Random Forest
  - k-Means, PCA, Naive Bayes, etc.
- Includes support for MOJO model objects for deployment
- Sparkling water = H2O on Spark



The H2O Sparkling Water Integration

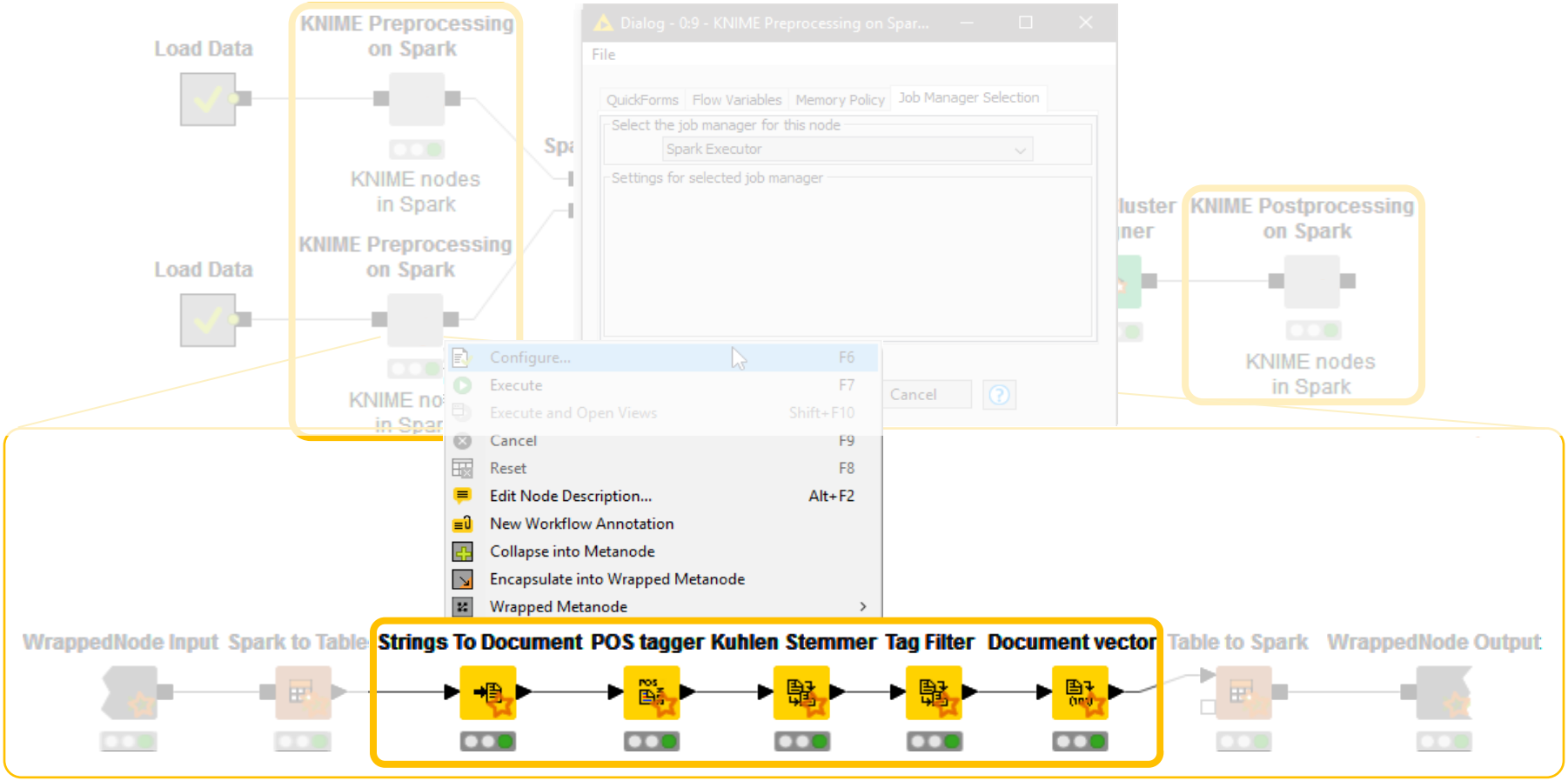




KNIME Big Data Extensions

- Introduction to Hadoop and Spark
- KNIME Big Data Connectors
- KNIME Extension for Apache Spark
- KNIME H2O Sparkling Water Integration
- KNIME Workflow Executor for Apache Spark

KNIME Workflow Executor for Apache Spark



Use Cases & Limitations

- Each workflow replica processes the rows of **one** partition!

- **Good match for**:
  - KNIME nodes that operate row-by-row
    - Many pre- and postprocessing nodes
    - Predictor nodes
    - Nodes that are streamable
  - Parallel execution of standard KNIME workflows on “small” data
    - Hyper-parameter optimization

- **Bad match for**: Any node that needs all rows, such as
  - GroupBy, Joiner, Pivoting, …
  - Model learner nodes