Workflows in Petroleum Exploration and Production

DAVID WONG

NOVEMBER 2017
About Me
A Lot of Raw Data
In Many Forms

FILES


https://en.wikipedia.org/wiki/Well_logging

STRUCTURED DATABASE

WEB SERVICES

UNSTRUCTURED DATABASE
Interpretation Applications

RESERVOIR SIMULATION

GEOLOGICAL INTERPRETATION

JOA OIL & GAS BV, FROM JEWEL SUITE

PETRAL
Very large data model that covers many subject areas such as:

- Wells
- Production
- Seismic
- Land mineral rights
- Lithology
- Facility
- GIS

1700 tables
53 subject areas
KNIME Advantages
KNIME Advantages

- See **RESULTS** of each node.

- DIY’er could **BUILD ELABORATED WORKFLOWS** without big software engineering background.

- **VERSATILE PROGRAMMING ENVIRONMENT** – connecting nodes, SQL, Java, R, Python.

- **CONNECT** with all kind of Java technologies through SDK.
Generate Scout Ticket with KNIME

Retrieve wellbore information from database and send it to report:
- Database Reader
- Unpivoting
- Column Filter
- Data to Report

Retrieve directional survey information from database and send it to report:
- Database Reader
- Java Snippet
- Java Snippet
- Data to Report

Call custom KNIME node to reduce decline curve
- Database Reader
- DeclineCurve
- Data to Report
Generate Scout Ticket with KNIME

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Well Number</td>
<td>177154087800</td>
</tr>
<tr>
<td>Well Name</td>
<td>ST008800</td>
</tr>
<tr>
<td>Operator Number</td>
<td>2375</td>
</tr>
<tr>
<td>Bottom Field Name</td>
<td>KTPA</td>
</tr>
<tr>
<td>Scout Date</td>
<td>19951204</td>
</tr>
<tr>
<td>AKB Elevation</td>
<td>97</td>
</tr>
<tr>
<td>BH Total MD</td>
<td>6179</td>
</tr>
<tr>
<td>True Vertical Depth</td>
<td>4693</td>
</tr>
<tr>
<td>Surface N/S Distance</td>
<td>03228</td>
</tr>
<tr>
<td>Surface E/W Code</td>
<td>W</td>
</tr>
<tr>
<td>Surface Area</td>
<td>ST</td>
</tr>
<tr>
<td>Surface Block</td>
<td>54</td>
</tr>
<tr>
<td>Bottom N/S Distance</td>
<td>533</td>
</tr>
<tr>
<td>Bottom E/W Distance</td>
<td>1349</td>
</tr>
<tr>
<td>Bottom E/W Code</td>
<td>W</td>
</tr>
<tr>
<td>Bottom Area</td>
<td>ST</td>
</tr>
<tr>
<td>Bottom Depth</td>
<td>64</td>
</tr>
<tr>
<td>Total Depth Date</td>
<td>199510104</td>
</tr>
<tr>
<td>Status Code</td>
<td>12500124</td>
</tr>
<tr>
<td>Type Code</td>
<td>D</td>
</tr>
<tr>
<td>District Code</td>
<td>2</td>
</tr>
<tr>
<td>Site Code</td>
<td>COM</td>
</tr>
<tr>
<td>Water Depth</td>
<td>65</td>
</tr>
<tr>
<td>Surface Longitude</td>
<td>-90.416616</td>
</tr>
<tr>
<td>Surface Latitude</td>
<td>28.833671</td>
</tr>
<tr>
<td>Bottom Longitude</td>
<td>-90.423093</td>
</tr>
<tr>
<td>Bottom Latitude</td>
<td>28.83147</td>
</tr>
<tr>
<td>Surface Lease Number</td>
<td>19</td>
</tr>
<tr>
<td>Complex IR Num</td>
<td>220235</td>
</tr>
<tr>
<td>Structure Number</td>
<td>1</td>
</tr>
</tbody>
</table>

Scout Ticket For 177154087800

Directions Survey

Decline Curve Analysis
Download Well Data
Download GIS Data

AREA: SOUTH TIMBLIER

BLOCK: 24445
Java Snippet

DISTANCE FROM SURFACE LOCATION

```
1m// system imports
3 // Your custom imports:
5 // system variables
7 // Your custom variables:
9 static double first_lat = -10000;
11 static double first_long = -10000;
13 public double deg2rad(double deg) {
15 return deg * (Math.PI/180);
17 }
19 // expression start
21 // Enter your code here:
23 if (first_lat == -10000.) first_lat = c_latitude;
25 if (first_long == -10000.) first_long = c_longitude;
27 double R = 6371; // Radius of the earth in km
29 double dLat = deg2rad(c_Latitude-first_lat); // deg2rad below
31 double dLon = deg2rad(c_Longitude-first_long); // deg2rad below
33 double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
35 Math.cos(deg2rad(first_lat)) * Math.cos(deg2rad(c_Longitude)) *
37 Math.sin(dLon/2) * Math.sin(dLon/2);
39 double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
41 out_distance = R * c = 3280.84; // Distance in foot
43 out_debug1 = a;
```
DECLINE CURVE CALCULATION

```java
@override
protected void execute() {
    final void execute(ExecutionContext ecc) throws Exception {
        // TODO: do something here
        logger.info("Decline Curve Calculation: This is not yet implemented");
        // the data table spec of the single output table,
        // the table will have three columns:
        DataTableSpec allColSpecs = new DataTableSpec();
        // ----------------- array size
        allColSpecs[0] = new DataColumnSpec("date", IntCell.TYPE);
        allColSpecs[1] = new DataColumnSpec("fct", DoubleCell.TYPE);
        allColSpecs[2] = new DataColumnSpec("rate", DoubleCell.TYPE);
        // the execution context will provide us with storage capacity, in this
        // case a data container to which we will add rows sequentially
        // Note: This container can also handle arbitrary big data tables, it
        // will buffer to disk if necessary.
        BufferedDataContainer outputSpec = new BufferedDataContainer();
        // let's add n_count rows to it
        // calculate summation stuff
        double sumMonthly = 0;
        double sum = 0;
        for (finalDataRow row : inTable) {
            double oMonthly = ((DoubleCell) row.getCell(3)).getDoubleValue();
            if (oMonthly == 0.) {
                continue;
            } else {
                sum += oMonthly;
                sumMonthly += oMonthly;
            }
        }
        // calculate rate
        finalRow = outputSpec.addRow();
        finalRow.setDouble("rate", sumMonthly / sum);
    }
}
```
Thank you.

QUESTIONS? DAVID.WONG@MMIAGENCY.COM