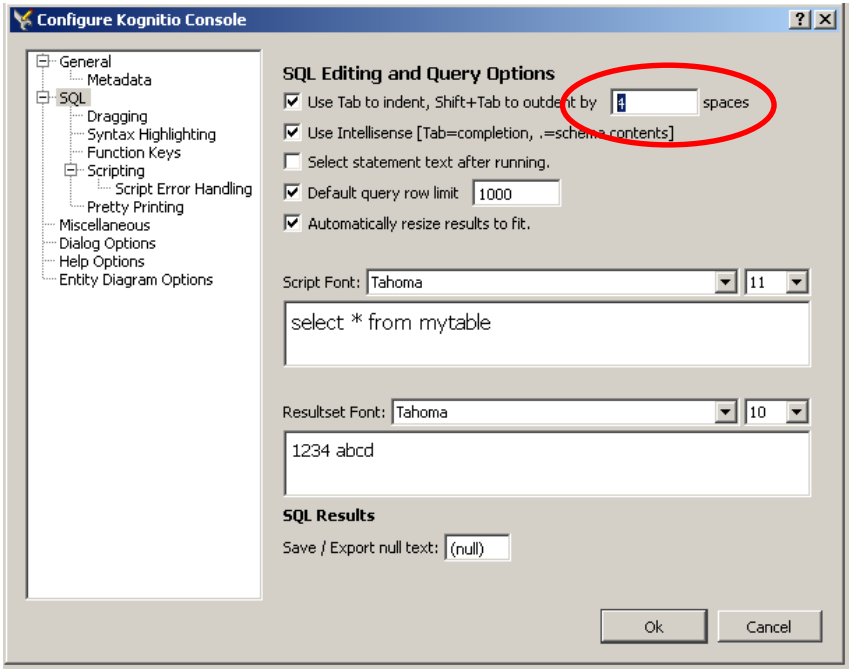
Passing data through to scripts using environment variables and SQL

Click here to download the supporting SQL script

External Script Training Overview

This is the Python version

1. Introduction to external script interface
2. Controlling Script Processes – threads and nodes
3. Controlling Data Processing – partitions
4. Passing parameters through to scripts



Python is strict about code indentation as it is used to define code within “for”, “if/else” statements etc.

When using Kognitio console set the tab indentation to 4 to match this requirement.

Tools->Configure->SQL

Kognitio

Passing data through to scripts:

Using environment variables in python

```
create external script Kog_P_AvgOverPartID_Parm environment python
receives(Part_ID int, value float)
partition by Part_ID
sends(Part_ID int, avg1 float, count1 float, perc1 float)
script S'EOF(
import csv,sys,os
input = csv.reader(sys.stdin)
result = csv.writer(sys.stdout)
prevpartid=''           #Holds the previous partition id
sum1=0                  #Initialise sum
count1=0                   #Initialise count
totalsales=float(os.environ['WX2_TOTALSALES'])

for row in input:
    if len(row)>0:
        if row[0]==prevpartid:
            sum1=sum1+float(row[1])
            count1=count1+1
        else:
            if prevpartid!='':
                avg1=sum1/count1
                perc1=sum1/totalsales
                result.writerow([prevpartid,avg1,count1,perc1])
            prevpartid=row[0]
            sum1=float(row[1])
            count1=1
if count1!=0:
    avg1=sum1/count1
    perc1=sum1/totalsales
    result.writerow([prevpartid,avg1,count1,perc1])
)EOF';
```

Parameters can be passed into Kognitio external scripts via environment variables

Add new output column for percentage of total sales

In python need to import the standard os package

os.environ accesses the environment variables passed through to script. The naming convention for Kognitio variables is WX2_*varname*

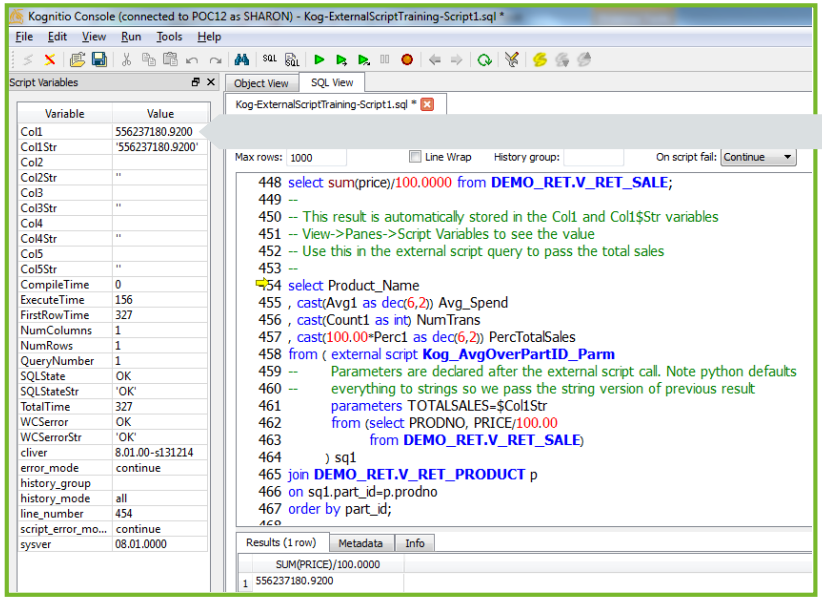
In this example the value passed is used to calculate the percentage of total sales for each product

Kognitio

Passing data through to scripts:
Assigning environment variables in SQL

```
select sum(price)/100.0000 from DEMO_RET.V_RET_SALE;
```

In your Console script run a query to get the total sales value



When any query is run in a script (via console or wxsubmit) the first row of output is stored in the “Col” environment variables
To view these in Console select View > Panes > Script Variables

```
select Product_Name
, cast(Avg1 as dec(6,2)) Avg_Spend
, cast(Count1 as int) NumTrans
, cast(100.00*Perc1 as dec(6,2)) PercTotalSales
from ( external script Kog_AvgOverPartID_Parm
       parameters TOTALSALES=$Col1Str
       from (select PRODNO, PRICE/100.00
                  from DEMO_RET.V_RET_SALE)
        ) sq1
join DEMO_RET.V_RET_PRODUCT p
on sq1.part_id=p.prodno
order by part_id;
```

Parameters are declared directly after the external script call.
Here TOTALSALES is assigned the result from the previous query. This will be stored in the environment variables (available to external script) as WX2_TOTALSALES

Kognitio

Passing data through to scripts:

Passing data sets via script interface

```
create external script Kog_AvgOverPartID_Pass environment python
receives(Part_ID int, Flag int, value float)
partition by Part_ID order by Flag
sends(Part_ID int, avg1 float, count1 float, perc1 float)
script S'EOF(
import csv,sys
input = csv.reader(sys.stdin)
result = csv.writer(sys.stdout)
prevpartid=''      #Holds the previous partition id
sum1=0             #Initialise sum
count1=0           #Initialise count
totalsales=0       #Initialise total sales
for row in input:
    if len(row)>0:
        if row[1]=='-1':
            totalsales=float(row[2])
        else:
            if row[0]==prevpartid:
                sum1=sum1+float(row[2])
                count1=count1+1
            else:
                 if prevpartid!='':
                    avg1=sum1/count1
                    perc1=sum1/totalsales
                    result.writerow([prevpartid,avg1,count1,perc1])
                prevpartid=row[0]
                sum1=float(row[1])
                count1=1
if count1!=0:
    avg1=sum1/count1
    perc1=sum1/totalsales
    result.writerow([prevpartid,avg1,count1,perc1])
)EOF';
```

If there is a lot of data to pass through to the script then it is better to pass it with the data via interface and SQL

Add a new input variable to flag the different types of row. In this case total sales and standard input

Use flag (row[1]) in code to differentiate input. Here if flag = -1 the row is totalsales not standard input (flag=0)

totalsales is used in same way as in the environment variable script

Kognitio

Passing data through to scripts:

SQL to pass through data

```
select Product_Name
, cast(Avg1 as dec(6,2)) Avg_Spend
, cast(Count1 as int) NumTrans
, cast(100.00*Perc1 as dec(8,4)) PercTotalSales
from ( external script Kog_AvgOverPartID_Pass
         from (select PRODNO
                    , 0 Flag
                    , PRICE/100.00
                from DEMO_RET.V_RET_SALE
               union all
                select PN.ProdNo
                    , -1 Flag
                    , TS.TotalSales
                  from(select sum(price)/100.0000 TotalSales
                       from DEMO_RET.V_RET_SALE) TS
                       cross join
                      (select distinct PRODNO
                       from DEMO_RET.V_RET_SALE) PN
                )
       ) sq1
join DEMO_RET.V_RET_PRODUCT p
on sq1.part_id=p.prodno
order by part_id;
```

The SQL used to feed data into the external script is more complex than previous examples

Sales Data is passed as before but flagged with a 0 to indicate standard input row

“union all” is used so that the total sales data can be passed with standard input

This “select” uses a cross (or cartesian) join[1] between total sales (TS) and Product No (PN) to send one row flagged -1 to each partition

[1] A cartesian or cross join simply joins every row in one table, view or sub-query to every row in another. They should be used with caution as can lead to large explosions in data volumes

Kognitio