Traditional ETL: not where we want to be ! Highly Uncool

# Data virtualization

From Wikipedia, the free encyclopedia

**Data virtualization** is any approach to data management that allows an application to retrieve and manipulate data without requiring technical details about the data, such as how it is formatted or where it is physically located.[1]

Unlike the traditional extract, transform, load ("ETL") process, the data remains in place, and real-time access is given to the source system for the data, thus reducing the risk of data errors and reducing the workload of moving data around that may never be used.

Unlike Data Federation it does not attempt to impose a single data model on the data (heterogeneous data). The technology also supports the writing of transaction data updates back to the source systems.[2]

To resolve differences in source and consumer formats and semantics, various abstraction and transformation techniques are used.

This concept and software is a subset of data integration and is commonly used within business intelligence, service-oriented architecture data services, cloud computing, enterprise search, and master data management.

## Commercial Tools [ edit ]

Commercially available data virtualization tools include:

- Enterprise Enabler Data Virtualization Platform
- Red Hat JBoss Data Virtualization
- Cisco Data Virtualization
- Denodo Data Virtualization Platform
- HiperFabric Data Virtualization and Integration

## Reference Books [ edit ]

- **Data Virtualization: Going Beyond Traditional Data Integration to Achieve Business Agility**, Judith R. Davis and Robert Eve
- **Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses** Rick van der Lans
- **Data Integration Blueprint and Modeling: Techniques for a Scalable and Sustainable Architecture** Anthony Giordano

## History [ edit ]

Enterprise Information Integration (EII), first coined by Metamatrix, now known as Red Hat JBoss Data Virtualization, and data federation have been used by some vendors to describe a core element of data virtualization: the capability to create relational JOINs in a federated VIEW.

# Federated database system

From Wikipedia, the free encyclopedia

A **federated database system** is a type of meta-database management system (DBMS), which transparently maps multiple autonomous database systems into a single **federated database**. The constituent databases are interconnected via a computer network and may be geographically decentralized. Since the constituent database systems remain autonomous, a federated database system is a contrastable alternative to the (sometimes daunting) task of merging several disparate databases. A federated database, or **virtual database**, is a composite of all constituent databases in a federated database system. There is no actual data integration in the constituent disparate databases as a result of data federation.

Through data abstraction, federated database systems can provide a uniform user interface, enabling users and clients to store and retrieve data from multiple noncontiguous databases with a single query—even if the constituent databases are heterogeneous. To this end, a federated database system must be able to decompose the query into subqueries for submission to the relevant constituent DBMS's, after which the system must composite the result sets of the subqueries. Because various database management systems employ different query languages, federated database systems can apply wrappers to the subqueries to translate them into the appropriate query languages.

http://statrgy.com/2015/05/20/best-sql-on-hadoop-tool/

# Best SQL-on-hadoop tool?

Posted on May 20, 2015 by Jean-Baptiste Poullet

Well, SQL seems to be cool again!

## SQL on Hadoop: Drill, Impala or Spark SQL

## Other SQL-on-Hadoop alternatives

There are other SQL-on-Hadoop alternatives out there:

- Presto: developed by Facebook, it is open-source but not supported by third-party vendors as long as I know), it can query data from multiple sources (Hive, Cassandra, RDBMS, etc);
- Pivotal HAWQ: developed by Pivotal, it has been recently open-sourced and is now available in the Hortonworks Data Platform (HDP), it can query data from multiple sources (Hive, HBase, etc);
- Big SQL: developed by IBM and part of its Big Insights platform, it is closed-source code, and can query multiple data sources (Hive, HBase, etc); it is probably only useful if you use IBM tools;
- Apache Phoenix: top-level Apache project, open-source, it can only query HBase since Phoenix is nothing else than a relational database layer over HBase, allowing low latency queries over HBase data; Phoenix shows much better performance than Hive and Impala over HBase on some benchmarks;
- Apache Tajo: Apache top-level project, Apache Tajo is a robust big data relational and distributed data warehouse system for Apache Hadoop. Tajo is designed for low-latency and scalable ad-hoc queries, online aggregation, and ETL (extract-transform-load process) on large-data sets stored on HDFS (Hadoop Distributed File System) and other data sources;
- Teradata SQL-H: developed by Teradata, "with Aster SQL-H™, users of the Teradata Aster Discovery Platform got the ability to issue SQL and SQL-MapReduce® queries directly on Hadoop data as if that data had been in Aster all along".
- Blink DB: developed by people from Universities of MIT, California (Berkeley) and Michigan. BlinkDB is a massively parallel, approximate query engine for running interactive SQL queries on large volumes of data. "It allows users to trade-off query accuracy for response time, enabling interactive queries over massive data by running queries on data samples and presenting results annotated with meaningful error bars."

## Kiss the overhead goodbye and enjoy data agility

Traditional query engines demand significant IT intervention before data can be queried. Drill gets rid of all that overhead so that users can just query the raw data in-situ. There's no need to load the data, create and maintain schemas or transform the data before it can be processed. Instead, simply include the path to a Hadoop directory, MongoDB collection or S3 bucket in the SQL query.

Drill leverages advanced query compilation and re-compilation techniques to maximize performance without requiring up-front schema knowledge.

```
SELECT * FROM dfs.root.`/web/logs`;

SELECT country, count(*)
  FROM mongodb.web.users
  GROUP BY country;

SELECT timestamp
  FROM s3.root.`clicks.json`
  WHERE user_id = 'jdoe';
```

Oh, mamma, interpretation problems ahead!


## Treat your data like a table even when it's not

Drill features a JSON data model that enables queries on complex/nested data as well as rapidly evolving structures commonly seen in modern applications and non-relational datastores. Drill also provides intuitive extensions to SQL so that you can easily query complex data.

Drill is the only columnar query engine that supports complex data. It features an in-memory shredded columnar representation for complex data which allows Drill to achieve columnar speed with the flexibility of an internal JSON document model.

## Keep using the BI tools you love

Drill supports standard SQL. Business users, analysts and data scientists can use standard BI/analytics tools such as Tableau, Qlik, MicroStrategy, Spotfire, SAS and Excel to interact with non-relational datastores by leveraging Drill's JDBC and ODBC drivers. Developers can leverage Drill's simple REST API in their custom applications to create beautiful visualizations.

Drill's virtual datasets allow even the most complex, non-relational data to be mapped into BI-friendly structures which users can explore and visualize using their

http://pedroalves-bi.blogspot.nl/2015/10/pentaho-60-ee-and-ce-is-available.html

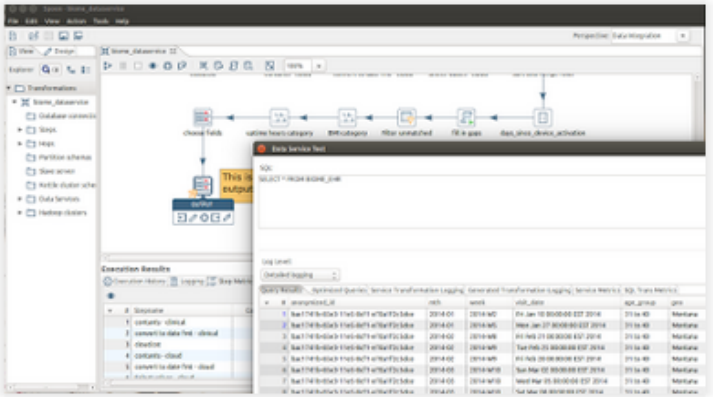
# Pentaho 6.0 (EE and CE) is available

You can get it from the usual places:

- Pentaho 6.0 EE from the main website
- Pentaho 6.0 CE from the community website

Lemme do a very quick overview of some of the great features. No real order here. And only of the stuff that comes to mind....

## Data Services (CE/EE)



*Data Services*

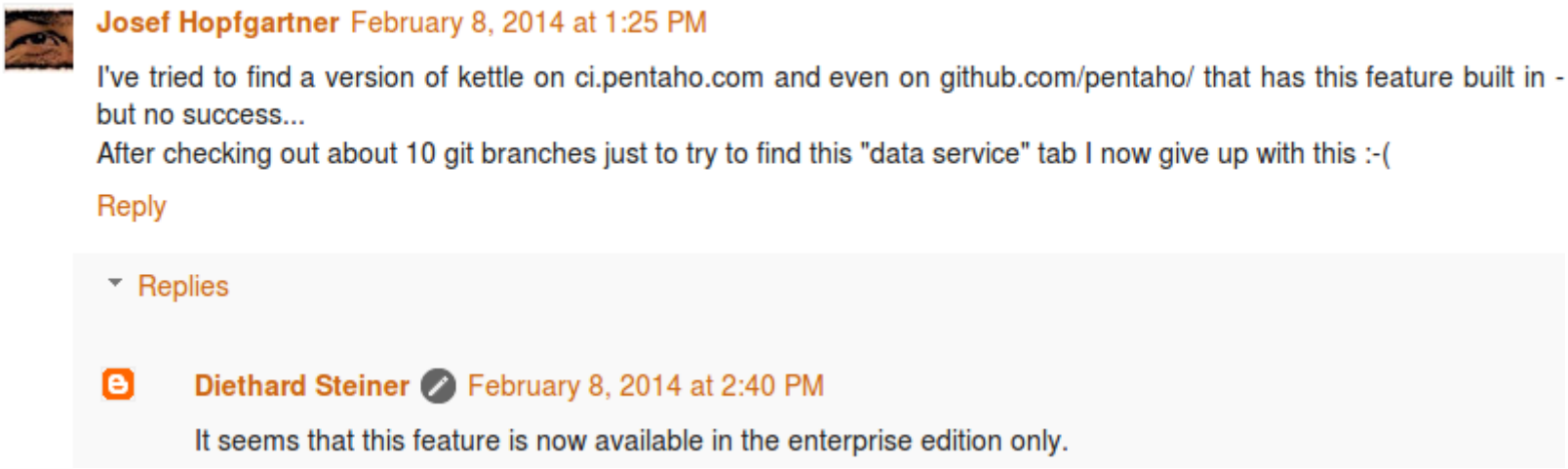
This is huge. Huge. Data Services have been my baby for the past few months and it's going to play a huge part of everything that we'll be doing in the future. It's a very simple concept that Matt Casters came up with a couple years back; but like happens most of the time, takes the rest of the world lots of time to understand what he actually means :p

Simply put - we can expose any point of our transformation as a virtual / umaterialized table that can be accessed from the outside through a jdbc interface. And this usage can be not only external but also internal.

Some of you may remember that the concept of dataservices appear in 5.0 as the codename jdbc thin driver, an EE only feature. But we huge improvements on the concept, not only at UI level but also on global *Caching* abilities and *Push Down Optimizations*.

# Data Federation with PDI CE

http://diethardsteiner.blogspot.nl/2013/01/creating-federated-data-service-with.html



**Josef Hopfgartner** February 8, 2014 at 1:25 PM

I've tried to find a version of kettle on ci.pentaho.com and even on github.com/pentaho/ that has this feature built in - but no success...
After checking out about 10 git branches just to try to find this "data service" tab I now give up with this :-(

Reply

▾ Replies

**Diethard Steiner** February 8, 2014 at 2:40 PM

It seems that this feature is now available in the enterprise edition only.

This was in a 5.0.4 RC I believe

## Thread: Pointless Data Services feature in Community Edition

10-27-2015, 04:38 AM

liorabel
**Junior Member**

### Pointless Data Services feature in Community Edition

With PDI 6, Pentaho have given us the amazing tool of easily usable Data Services.
Being able to expose reusable transformations as virtual JDBC tables is extremely useful. So much so that it is potentially going to change my whole ETL architecture.

However, I have not yet found any way to take advantage of Data Services in the Community Edition. We have no DI Server, Carte won't expose them, even Spoon can do nothing with them without them being exposed by a server somewhere.

So with this I have to ask, what's the point of having Data Services in the CE Spoon at all? They are as useless as ice to Eskimos.

Lior

10-28-2015, 06:31 AM

liorabel
**Junior Member**

> Originally Posted by **saidlp13**
>
> *In CE, carte is nothing but the data integration server. I think we need to take the help of this for data services what you mentioned.*

If this features exists in the CE Carte and just not documented, are you able to share a few details here?

# Thread: PDI 6.0 - dataservice ok in spoon, not visible as table with Carte

10-21-2015, 04:32 AM

**peter.kloosterhof**
**Junior Member**

## PDI 6.0 - dataservice ok in spoon, not visible as table with Carte

With pdi-ce-6.0.0.0-353 and Carte I can create and test dataservices succesfully.
But they are not listed as tables when connected via JDBC.

What does work:
http://localhost:9080/kettle/status/ works fine, and my file based repository is shown
I can start a transformation with
http://localhost:9080/kettle/execute...sformatie1.ktr, this returns the JSON output

I can start a job with
http://localhost:9080/kettle/runJob/...testje%2FMJob1, this returns the 'job started' message in xml
After this the transformation and job are listed in http://localhost:9080/kettle/status/

What doesn't:
I can create and test a dataservice in spoon from the above mentioned transformation but when connected via JDBC I can't explore the database, there are no tables visible. I'm using Squirrel-SQL as a client for testing.
http://localhost:9080/kettle/listServices returns a blank page
My .pentaho/metastore folder remains empty

What's the next step in troubleshooting?

Regards,

Peter

Hi,

I am facing a similar issue. Were you able to properly resolve this? If so, could you elaborate please?

Regards,
Lior

10-27-2015, 03:52 AM

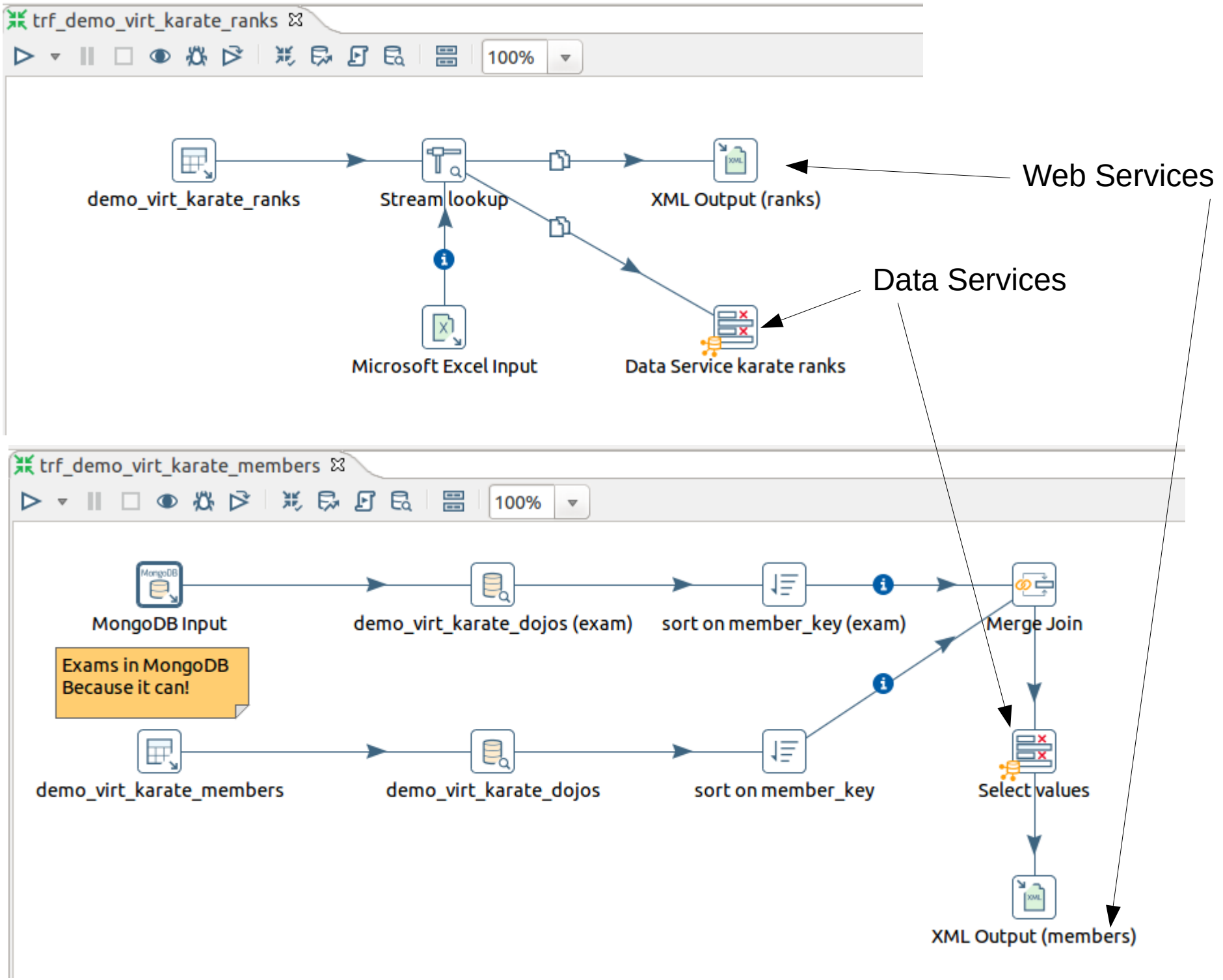
**peter.kloosterhof**
**Junior Member**

Nope, still have my hopes on one of the gurus on this forum

# Goal: get this stuff to work, and even join Data Services with SQL
# Example: karate ranks in Shotokan

松濤館
SHOTOKAN

When not figuring out data, I do this.
So what was your objection to my solution again ..?


trf_demo_virt_karate_ranks
100%
demo_virt_karate_ranks
Stream lookup
XML Output (ranks)
Microsoft Excel Input
Data Service karate ranks
Web Services
Data Services
trf_demo_virt_karate_members
100%
MongoDB Input
demo_virt_karate_dojos (exam)
sort on member_key (exam)
Merge Join
Exams in MongoDB
Because it can!
demo_virt_karate_members
demo_virt_karate_dojos
sort on member_key
Select values
XML Output (members)

# /home/edwin/.pentaho/metastore/pentaho/Data Service Transformation

demo_virt_karate_ranks.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<element>
  <id>demo_virt_karate_ranks</id>
  <value></value>
  <type>String</type>
  <children>
    <child>
      <id>trans_references</id>
      <value>org.pentaho.di.trans.dataservice.serialization.ServiceTrans$Reference</value>
      <type>String</type>
      <children>
        <child>
          <id>0</id>
          <value></value>
          <type>String</type>
          <children>
            <child>
              <id>transformation_location</id>
              <value>/media/data1/kff/projects/websol/datavault/code/pentaho_meetup/trf_demo_virt_karate_ranks.ktr</value>
              <type>String</type>
            </child>
            <child>
              <id>transformation_storage_method</id>
              <value>FILE</value>
              <type>String</type>
            </child>
          </children>
        </child>
      </children>
    </child>
  </children>
  <name>demo_virt_karate_ranks</name>
  <security>
    <owner/>
    <owner-permissions-list/>
  </security>
</element>
```

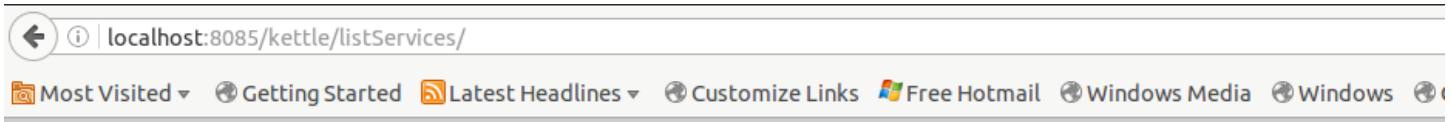
Does not get created when working
In File Repository!
Can do it easily by hand though.

.type.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<data-type>
  <name>Data Service Transformation</name>
  <description>Pointer to a saved transformation that supplies a data service</description>
</data-type>
```

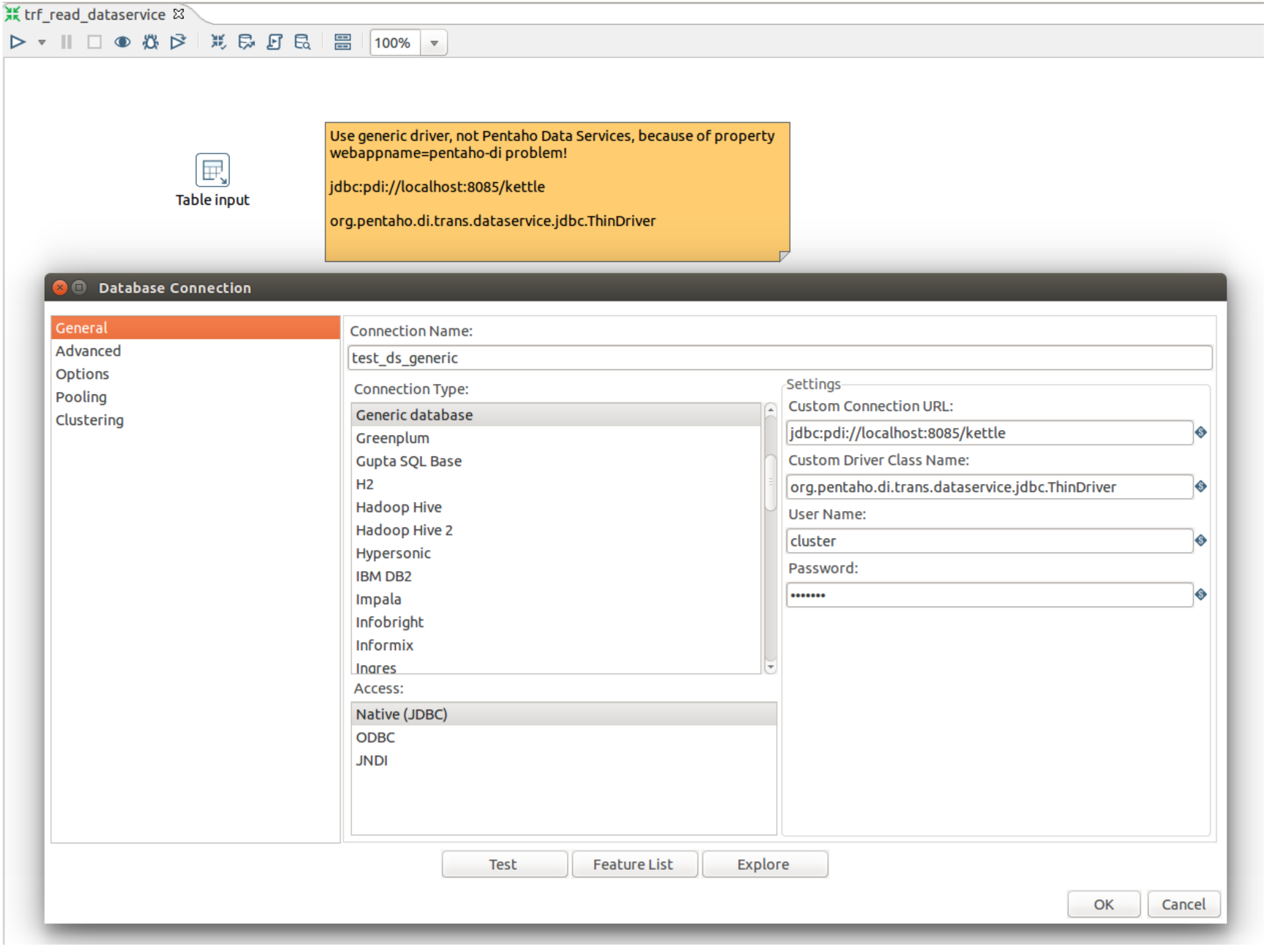
# Carte configuration: nothing special

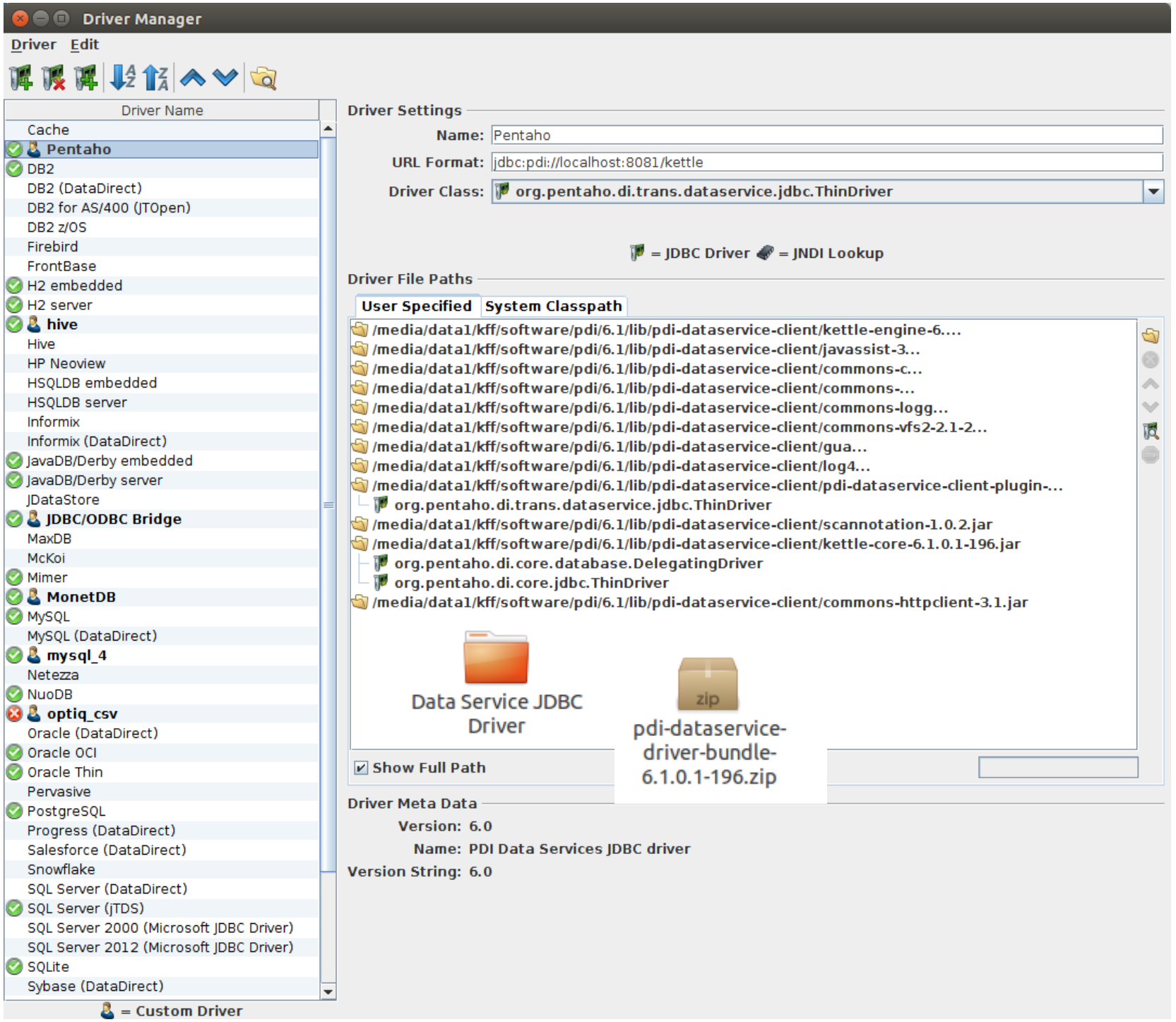
```
<slave_config>
  <!--
    Document description...

    - masters: You can list the slave servers to which this slave has to report back to.
               If this is a master, we will contact the other masters to get a list of all the slaves in the cluster.

    - report_to_masters : send a message to the defined masters to let them know we exist (Y/N)

    - slaveserver : specify the slave server details of this carte instance.
                    IMPORTANT : the username and password specified here are used by the master instances to connect to this slave.

  -->

  <masters>
  </masters>

  <slaveserver>
    <name>slave1-8085</name>
    <hostname>localhost</hostname>
    <port>8085</port>
    <username>cluster</username>
    <password>cluster</password>
    <master>N</master>
  </slaveserver>

</slave_config>
```




This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
−<services>
  −<service>
     <name>demo_virt_karate_ranks</name>
    −<row-meta>
      −<value-meta>
          <type>String</type>
          <storagetype>normal</storagetype>
          <name>rank_key</name>
          <length>32</length>
          <precision>-1</precision>
          <origin>demo_virt_karate_ranks</origin>
          <comments>rank_key</comments>
          <conversion_Mask/>
          <decimal_symbol>.</decimal_symbol>
          <grouping_symbol>,</grouping_symbol>
          <currency_symbol/>
          <trim_type>none</trim_type>
          <case_insensitive>N</case_insensitive>
          <sort_descending>N</sort_descending>
          <output_padding>N</output_padding>
          <date_format_lenient>N</date_format_lenient>
          <date_format_locale>en_US</date_format_locale>
          <date_format_timezone>Europe/Amsterdam</date_format_timezone>
          <lenient_string_to_number>N</lenient_string_to_number>
       </value-meta>
      −<value-meta>
          <type>String</type>
          <storagetype>normal</storagetype>
          <name>rank</name>
          <length>32</length>
          <precision>-1</precision>
          <origin>demo_virt_karate_ranks</origin>
          <comments>rank</comments>
          <conversion_Mask/>
          <decimal_symbol>.</decimal_symbol>
          <grouping_symbol>,</grouping_symbol>
          <currency_symbol/>
          <trim_type>none</trim_type>
          <case_insensitive>N</case_insensitive>
          <sort_descending>N</sort_descending>
```


trf_read_dataservice
100%
Table input
Use generic driver, not Pentaho Data Services, because of property webappname=pentaho-di problem!
jdbc:pdi://localhost:8085/kettle
org.pentaho.di.trans.dataservice.jdbc.ThinDriver
Database Connection
General
Advanced
Options
Pooling
Clustering
Connection Name:
test_ds_generic
Connection Type:
Generic database
Greenplum
Gupta SQL Base
H2
Hadoop Hive
Hadoop Hive 2
Hypersonic
IBM DB2
Impala
Infobright
Informix
Ingres
Access:
Native (JDBC)
ODBC
JNDI
Settings
Custom Connection URL:
jdbc:pdi://localhost:8085/kettle
Custom Driver Class Name:
org.pentaho.di.trans.dataservice.jdbc.ThinDriver
User Name:
cluster
Password:
Test
Feature List
Explore
OK
Cancel


Driver Manager
Driver Edit
Driver Name
Cache
Pentaho
DB2
DB2 (DataDirect)
DB2 for AS/400 (JTOpen)
DB2 z/OS
Firebird
FrontBase
H2 embedded
H2 server
hive
Hive
HP Neoview
HSQLDB embedded
HSQLDB server
Informix
Informix (DataDirect)
JavaDB/Derby embedded
JavaDB/Derby server
JDataStore
JDBC/ODBC Bridge
MaxDB
McKoi
Mimer
MonetDB
MySQL
MySQL (DataDirect)
mysql_4
Netezza
NuoDB
optiq_csv
Oracle (DataDirect)
Oracle OCI
Oracle Thin
Pervasive
PostgreSQL
Progress (DataDirect)
Salesforce (DataDirect)
Snowflake
SQL Server (DataDirect)
SQL Server (jTDS)
SQL Server 2000 (Microsoft JDBC Driver)
SQL Server 2012 (Microsoft JDBC Driver)
SQLite
Sybase (DataDirect)
= Custom Driver
Driver Settings
Name: Pentaho
URL Format: jdbc:pdi://localhost:8081/kettle
Driver Class: org.pentaho.di.trans.dataservice.jdbc.ThinDriver
= JDBC Driver = JNDI Lookup
Driver File Paths
User Specified
System Classpath
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/kettle-engine-6....
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/javassist-3...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/commons-c...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/commons-...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/commons-logg...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/commons-vfs2-2.1-2...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/gua...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/log4...
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/pdi-dataservice-client-plugin-...
org.pentaho.di.trans.dataservice.jdbc.ThinDriver
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/scannotation-1.0.2.jar
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/kettle-core-6.1.0.1-196.jar
org.pentaho.di.core.database.DelegatingDriver
org.pentaho.di.core.jdbc.ThinDriver
/media/data1/kff/software/pdi/6.1/lib/pdi-dataservice-client/commons-httpclient-3.1.jar
Data Service JDBC Driver
zip
pdi-dataservice-driver-bundle-6.1.0.1-196.zip
Show Full Path
Driver Meta Data
Version: 6.0
Name: PDI Data Services JDBC driver
Version String: 6.0

# Queries in DBVisualizer

Database Connection — Sticky — Database — Schema

Pentaho

```
1 select * from Kettle.demo_virt_karate_ranks;
2 select * from Kettle.demo_virt_karate_members;
3
4
5 --Error as PDI data services
6 select r.remark,m.*
7  from Kettle.demo_virt_karate_members m
8  left outer join Kettle.demo_virt_karate_ranks r
9  on m.rank = r.rank_key
10
11  --Now as Teiid views
12 select * from demo_virt_karate_ranks;
13 select * from demo_virt_karate_members;
14
15 --OK as Teiid views
16 select r.remark,m.*
17  from demo_virt_karate_members m
18  left outer join demo_virt_karate_ranks r
19  on m.rank = r.rank_key
20  order by m.date_exam,rank desc
21
```

7:20 [163] INS — Transactions not supported

Log

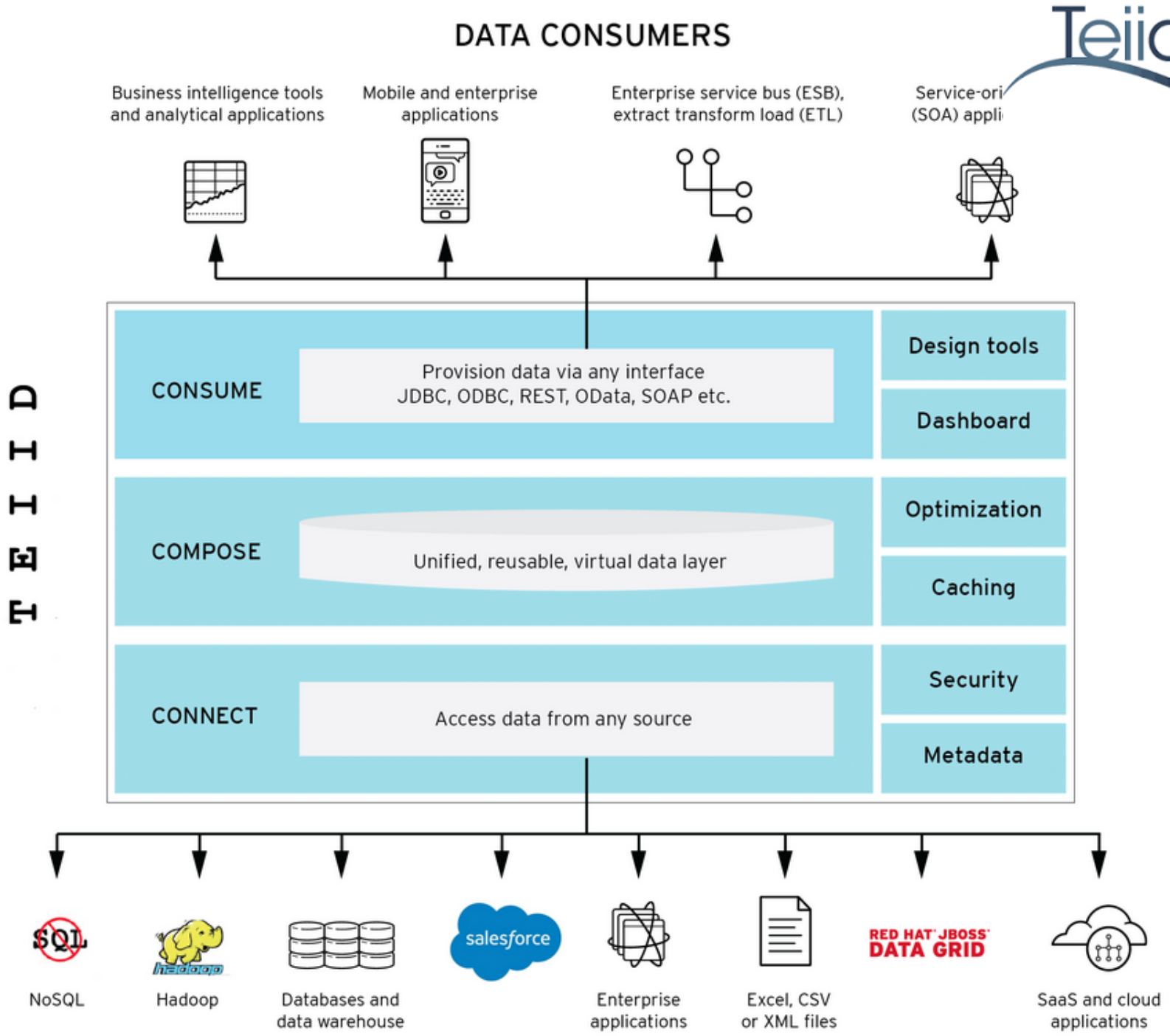
Preprocess script — Log to GUI — Log to File

```
... Physical database connection acquired for: Pentaho
00:14:48 [SELECT - 0 row(s), 0.017 secs] Found 11 parts for the FROM clause when only a table name and optionally an alias is supported: Kettle.demo_virt_karate_members m  left outer join Kettle.demo_virt_karate_ranks r  on m.rank = r.rank_key
... 1 statement(s) executed, 0 row(s) affected, exec/fetch time: 0.017/0.000 sec [0 successful, 0 warnings, 1 errors]
```

Table can only occur once in a query in Teiid in this setup

```
01:56:20 [SELECT - 0 row(s), 0.241 secs] [Error Code: 30443, SQL State: 50000] TEIID30443 java.sql.SQLException: java.sql.SQLException: TEIID60019 Streaming result has already been read once.  Ensure that on one read operation needs to be performed, for example XMLPARSE without the WELLFORMED operation must read the entire stream to validate its contents.  Or you may choose to use a non-streaming result.
... 1 statement(s) executed, 0 row(s) affected, exec/fetch time: 0.241/0.000 sec [0 successful, 0 warnings, 1 errors]
```

**Teiid is a data virtualization system that allows applications to use data from multiple, heterogeneous data stores.**



/media/data1/software/teiid-8.13.3/standalone/configuration

Standalone-teiid.xml

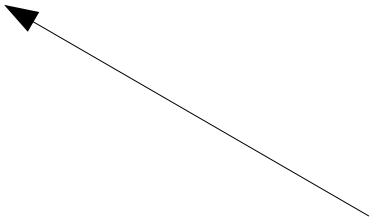
```
<resource-adapter id="webservice">
    <module slot="main" id="org.jboss.teiid.resource-adapter.webservice"/>
    <transaction-support>NoTransaction</transaction-support>
    <connection-definitions>
        <connection-definition class-name="org.teiid.resource.adapter.ws.WSManagedConnectionFactory" jndi-name="java:/ws_karate_ranks" enabled="true" use-java-context="true" pool-name="teiid-ws-ds">
            <config-property name="SecurityType">
                HTTPBasic
            </config-property>
            <config-property name="AuthPassword">
                cluster
            </config-property>
            <config-property name="EndPoint">
                http://cluster:cluster@localhost:8085/kettle/executeTrans/?trans=/media/data1/kff/projects/websol/datavault/code/pentaho_meetup/trf_demo_virt_karate_ranks.ktr
            </config-property>
            <config-property name="AuthUserName">
                cluster
            </config-property>
        </connection-definition>
        <connection-definition class-name="org.teiid.resource.adapter.ws.WSManagedConnectionFactory" jndi-name="java:/ws_karate_members" enabled="true" use-java-context="true" pool-name="teiid-ws-ds2">
            <config-property name="SecurityType">
                HTTPBasic
            </config-property>
            <config-property name="AuthPassword">
                cluster
            </config-property>
            <config-property name="EndPoint">
                http://cluster:cluster@localhost:8085/kettle/executeTrans/?trans=/media/data1/kff/projects/websol/datavault/code/pentaho_meetup/trf_demo_virt_karate_members.ktr
            </config-property>
            <config-property name="AuthUserName">
                cluster
            </config-property>
        </connection-definition>
    </connection-definitions>
</resource-adapter>
```

cd /media/data1/software/teiid-8.13.3/bin
./standalone.sh -c standalone-teiid.xml

/media/data1/software/teiid-8.13.3/standalone/deployments

test_webservice-vdb.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vdb name="test_webservice-vdb" version="1">

    <description>Shows how to call Web Services</description>

    <property name="UseConnectorMetadata" value="true" />
    <!--property name="{http://teiid.org/rest}auto-generate" value="true"/-->

    <model name="ws_karate_ranks" visible="false">
        <source name="ws_karate_ranks" translator-name="ws" connection-jndi-name="java:/ws_karate_ranks"/>
    </model>
    <model name="ws_karate_members" visible="false">
        <source name="ws_karate_members" translator-name="ws" connection-jndi-name="java:/ws_karate_members"/>
    </model>
    <model name="ws_combined" type="VIRTUAL">
        <metadata type="DDL"><![CDATA[
        CREATE VIEW demo_virt_karate_ranks AS
 SELECT
A.rank_key, A.rank, A.rank_type, A.belt_color, A.remark
FROM
(EXEC ws_karate_ranks.invokeHttp('GET',null,'http://cluster:cluster@localhost:8085/kettle/executeTrans/?trans=/media/data1/kff/projects/websol/datavault/code/pentaho_meetup/trf_demo_virt_karate_ranks.ktr', 'TRUE'))
AS f, XMLTABLE('/Ranks/Rank' PASSING XMLPARSE(DOCUMENT f.result)
COLUMNS rank_key string PATH 'rank_key', rank string PATH 'rank', rank_type string PATH 'rank_type', belt_color string PATH 'belt_color',remark string PATH 'remark') AS A;
        ]]>
        </metadata>
                <metadata type="DDL"><![CDATA[
        CREATE VIEW demo_virt_karate_members AS
 SELECT
A.karate_member_key, A.karate_member, A.date_of_birth, A.date_start, A.date_end,A.dojo_key, A.dojo, A.dojo_web_address,A.style,A.city, A.date_exam,A.rank,A.dojo_exam,A.city_exam
FROM
(EXEC ws_karate_members.invokeHttp('GET',null,'http://cluster:cluster@localhost:8085/kettle/executeTrans/?trans=/media/data1/kff/projects/websol/datavault/code/pentaho_meetup/trf_demo_virt_karate_members.ktr', 'TRUE'))
AS f, XMLTABLE('/KarateMembers/KarateMember' PASSING XMLPARSE(DOCUMENT f.result)
COLUMNS karate_member_key string PATH 'karate_member_key', karate_member string PATH 'karate_member', date_of_birth string PATH 'date_of_birth', date_start string PATH 'date_start',date_end string PATH 'date_end'
, dojo_key string PATH 'dojo_key', dojo string PATH 'dojo', dojo_web_address string PATH 'dojo_web_address',style string PATH 'style'
, city string PATH 'city', date_exam string PATH 'date_exam', rank string PATH 'rank',dojo_exam string PATH 'dojo_exam',city_exam string PATH 'city_exam'
) AS A;
        ]]>
        </metadata>
    </model>
</vdb>
```



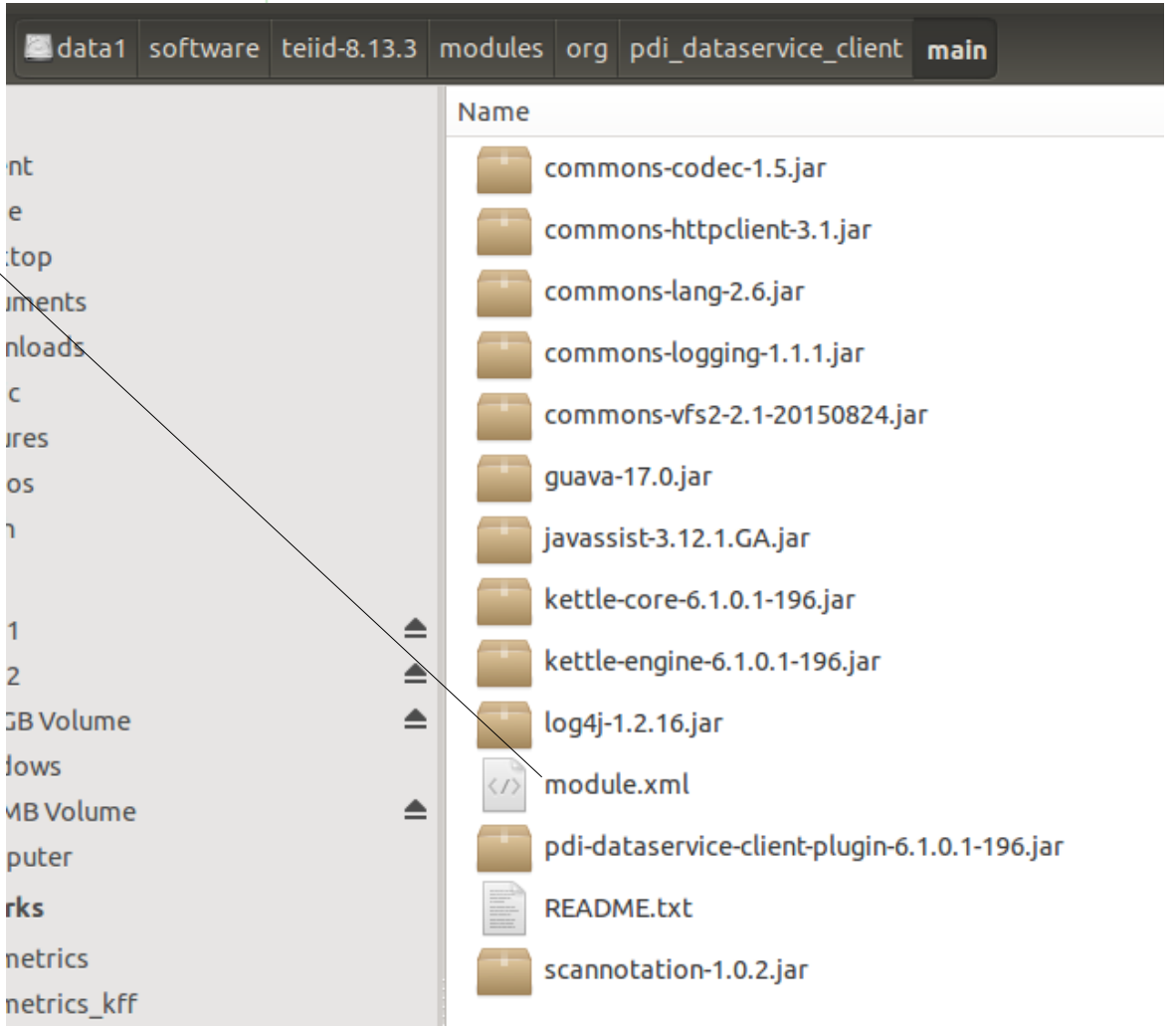
Virtual database definition

# PDI Data Service as JDBC source in Teiid ?

Standalone-teiid.xml

```
<datasource jta="true" jndi-name="java:/pdi_datasource" pool-name="pdi_datasource" enabled="true" use-ccm="true">
    <connection-url>jdbc:pdi://localhost:8085/kettle</connection-url>
    <driver-class>org.pentaho.di.trans.dataservice.jdbc.ThinDriver</driver-class>
    <driver>pdi_dataservice_client-driver</driver>
    <security>
        <user-name>cluster</user-name>
        <password>cluster</password>
    </security>
</datasource>
<datasource jta="true" jndi-name="java:/pdi_datasource7" pool-name="pdi_datasource7" enabled="true" use-ccm="true">
    <connection-url>jdbc:pdi://localhost:8085/kettle</connection-url>
    <driver-class>org.pentaho.di.trans.dataservice.jdbc.ThinDriver</driver-class>
    <driver>pdi_driver7</driver>
    <security>
        <user-name>cluster</user-name>
        <password>cluster</password>
    </security>
</datasource>
<drivers>
    <driver name="h2" module="com.h2database.h2">
        <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
    </driver>
    <driver name="teiid-local" module="org.jboss.teiid">
        <driver-class>org.teiid.jdbc.TeiidDriver</driver-class>
        <xa-datasource-class>org.teiid.jdbc.TeiidDataSource</xa-datasource-class>
    </driver>
    <driver name="teiid" module="org.jboss.teiid.client">
        <driver-class>org.teiid.jdbc.TeiidDriver</driver-class>
        <xa-datasource-class>org.teiid.jdbc.TeiidDataSource</xa-datasource-class>
    </driver>
    <driver name="pdi_dataservice_client-driver" module="org.pdi_dataservice_client">
        <driver-class>org.pentaho.di.trans.dataservice.jdbc.ThinDriver</driver-class>
    </driver>
    <driver name="pdi_driver7" module="org.pdi_dataservice_client7">
        <driver-class>org.pentaho.di.trans.dataservice.jdbc.ThinDriver</driver-class>
    </driver>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="org.pdi_dataservice_client">
  <resources>
    <resource-root path="pdi-dataservice-client-plugin-6.1.0.1-196.jar"/>
        <resource-root path="kettle-core-6.1.0.1-196.jar"/>
        <resource-root path="kettle-engine-6.1.0.1-196.jar"/>
        <resource-root path="commons-codec-1.5.jar"/>
        <resource-root path="commons-httpclient-3.1.jar"/>
        <resource-root path="commons-lang-2.6.jar"/>
        <resource-root path="commons-logging-1.1.1.jar"/>
        <resource-root path="commons-vfs2-2.1-20150824.jar"/>
        <resource-root path="guava-17.0.jar"/>
        <resource-root path="javassist-3.12.1.GA.jar"/>
        <resource-root path="log4j-1.2.16.jar"/>
        <resource-root path="scannotation-1.0.2.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```



test_pdi7-vdb.xml

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vdb name="test_pdi7-vdb" version="1">

    <description>Shows how to call Web Services</description>

    <property name="UseConnectorMetadata" value="true" />
    <!--property name="{http://teiid.org/rest}auto-generate" value="true"/-->

    <model name="pdi_datasource7">
        <source name="pdi_datasource7" translator-name="jdbc-ansi" connection-jndi-name="java:/pdi_datasource7"/>
    </model>
</vdb>
```

# That goes boom!
# Something with getColumns() to get JDBC info. (Roland, help!)

```
00:43:30,143 INFO  [org.teiid.CONNECTOR] (Worker0_async-teiid-threads0)  JDBCExecutionFactory Commit=true;DatabaseProductName=PDI;DatabaseProductVersion=7.0-SNAPSHOT;DriverMajorVersion=6;DriverMajorVersion
=0;DriverName=PDI Data Services JDBC driver;DriverVersion=6.0;IsolationLevel=0
00:43:30,151 INFO  [PDI Data Services JDBC driver] (Worker0_async-teiid-threads0)  -------------> Listing all tables!
00:43:30,183 INFO  [org.hibernate.validator.internal.util.Version] (ServerService Thread Pool -- 64)  HV000001: Hibernate Validator 5.1.3.Final
00:43:30,292 INFO  [org.wildfly.extension.undertow] (ServerService Thread Pool -- 64)  WFLYUT0021: Registered web context: /odata
00:43:30,295 INFO  [PDI Data Services JDBC driver] (Worker0_async-teiid-threads0)  -------------> Found 2 tables for the rows resultset.
00:43:30,296 INFO  [PDI Data Services JDBC driver] (Worker0_async-teiid-threads0)  getColumns(null, null, null, null)
00:43:30,319 INFO  [org.jboss.as.server] (ServerService Thread Pool -- 35)  WFLYSRV0010: Deployed "test_webservice-vdb.xml" (runtime-name : "test_webservice-vdb.xml")
00:43:30,319 INFO  [org.jboss.as.server] (ServerService Thread Pool -- 35)  WFLYSRV0010: Deployed "test_pdi7-vdb.xml" (runtime-name : "test_pdi7-vdb.xml")
00:43:30,319 INFO  [org.jboss.as.server] (ServerService Thread Pool -- 55)  WFLYSRV0010: Deployed "teiid-olingo-8.13.3-odata4.war" (runtime-name : "teiid-olingo-8.13.3-odata4.war")
00:43:30,319 INFO  [org.jboss.as.server] (ServerService Thread Pool -- 55)  WFLYSRV0010: Deployed "teiid-odata-8.13.3-odata2.war" (runtime-name : "teiid-odata-8.13.3-odata2.war")
00:43:30,428 WARN  [org.teiid.RUNTIME] (Worker0_async-teiid-threads0)  TEIID50036 VDB test_pdi7-vdb.1 model "pdi_datasource7" metadata failed to load. Reason:TEIID11010 java.sql.SQLException: org.pentaho.d
i.core.exception.KettleValueException:
Unexpected conversion error while converting value [NULLABLE Integer] to an Integer
java.lang.Integer cannot be cast to java.lang.Long
: org.teiid.translator.TranslatorException: TEIID11010 java.sql.SQLException: org.pentaho.di.core.exception.KettleValueException:
Unexpected conversion error while converting value [NULLABLE Integer] to an Integer
java.lang.Integer cannot be cast to java.lang.Long

        at org.teiid.translator.jdbc.JDBCExecutionFactory.getMetadata(JDBCExecutionFactory.java:297)
        at org.teiid.translator.jdbc.JDBCExecutionFactory.getMetadata(JDBCExecutionFactory.java:68)
        at org.teiid.query.metadata.NativeMetadataRepository.getMetadata(NativeMetadataRepository.java:92)
        at org.teiid.query.metadata.NativeMetadataRepository.loadMetadata(NativeMetadataRepository.java:60)
        at org.teiid.query.metadata.ChainingMetadataRepository.loadMetadata(ChainingMetadataRepository.java:55)
        at org.teiid.jboss.VDBService$6.run(VDBService.java:395)
        at org.teiid.jboss.VDBService$7.run(VDBService.java:446)
        at org.teiid.dqp.internal.process.DQPWorkContext.runInContext(DQPWorkContext.java:276)
        at org.teiid.dqp.internal.process.ThreadReuseExecutor$RunnableWrapper.run(ThreadReuseExecutor.java:119)
        at org.teiid.dqp.internal.process.ThreadReuseExecutor$3.run(ThreadReuseExecutor.java:210)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.sql.SQLException: org.pentaho.di.core.exception.KettleValueException:
Unexpected conversion error while converting value [NULLABLE Integer] to an Integer
java.lang.Integer cannot be cast to java.lang.Long

        at org.pentaho.di.trans.dataservice.jdbc.BaseResultSet.getValue(BaseResultSet.java:916)
        at org.pentaho.di.trans.dataservice.jdbc.BaseResultSet.getNonNullableValue(BaseResultSet.java:930)
        at org.pentaho.di.trans.dataservice.jdbc.BaseResultSet.getLong(BaseResultSet.java:287)
        at org.pentaho.di.trans.dataservice.jdbc.BaseResultSet.getInt(BaseResultSet.java:277)
        at org.teiid.translator.jdbc.JDBCMetdataProcessor.addColumn(JDBCMetdataProcessor.java:432)
        at org.teiid.translator.jdbc.JDBCMetdataProcessor.processColumns(JDBCMetdataProcessor.java:386)
        at org.teiid.translator.jdbc.JDBCMetdataProcessor.getColumns(JDBCMetdataProcessor.java:361)
        at org.teiid.translator.jdbc.JDBCMetdataProcessor.getTables(JDBCMetdataProcessor.java:312)
        at org.teiid.translator.jdbc.JDBCMetdataProcessor.getConnectorMetadata(JDBCMetdataProcessor.java:159)
        at org.teiid.translator.jdbc.JDBCExecutionFactory.getMetadata(JDBCExecutionFactory.java:295)
        ... 12 more
Caused by: org.pentaho.di.core.exception.KettleValueException:
Unexpected conversion error while converting value [NULLABLE Integer] to an Integer
java.lang.Integer cannot be cast to java.lang.Long
```

# DEMO

| | |
|---|---|
| Dockers to start: | docker_percona<br>docker_mongodb |
| Servers to start: | carte<br>Teiid |
| Queries: | DbVisualizer |