Create syntax

Create Streams

The create stream statement is used to tell Odysseus where the data comes from, this normally opens a connection to a source, e.g. a sensor or server.

The stream always consists of a name (here: "category") and a schema:

```
CREATE STREAM category (id INTEGER, name STRING, description STRING, parentid INTEGER) ....
```

Then, it is followed by a connection-property that tells how/where the stream can be accessed. Most used are the channel format and the generic access framework (which we recommend)

Odysseus Channel Format

Odysseus has a built-in byte-based format for transfering data. This is, for example, used by the nexmark example. This is called a "CHANNEL"-connection and looks like follows:

```
CREATE STREAM nexmark:person (timestamp STARTTIMESTAMP,id INTEGER,name STRING,email STRING,creditcard STRING,
city STRING,state STRING) CHANNEL localhost : 65440
```

Generic Access Framework

However, the recommended and new way is a generic access, which offers different protocols, wrappers etc. as described in Access framework. An example would be:

```
CREATE STREAM nexmark:person (timestamp STARTTIMESTAMP, id INTEGER, name STRING, email STRING, creditcard
STRING, city STRING, state STRING)
    WRAPPER 'GenericPush'
    PROTOCOL 'SizeByteBuffer'
    TRANSPORT 'NonBlockingTcp'
    DATAHANDLER 'Tuple'
    OPTIONS ( 'port' '65440', 'host' 'odysseus.offis.uni-oldenburg.de', 'ByteOrder' 'Little_Endian')
```

As you may see, there is a direct mapping between the needed parameters. So you can use each Protocol Handler and Data handler and Transport Handler in a CREATE STREAM statement. Thus, the wrapper must be also existing, which are e.g. GenericPush or GenericPull (see also Access framework). The Options-parameter is optional and is a comma separated list of key value pairs that are enclosed by quotation marks.

Create Views

You can also create a view, which is a logical view on a result of a continuous query.

```
CREATE VIEW nexQuery FROM (
    SELECT b.auction, DolToEur(b.price) AS euroPrice, b.bidder, b.datetime FROM nexmark:bid [UNBOUNDED] AS b
)
```

This allows you to resuse the query, e.g. as follows:

```
SELECT * FROM nexQuery
```

Create Sinks

Similar to creating sources for incoming data by "create stream", you can also create sinks for outgoing data. The notation is very similar to "create stream". Since it is also based on the Access Framework, you can also need different Protocol Handler and Data handler and Transport Handler. For example, the following creates a sink that writes a CSV file:

```
CREATE SINK writeout (timestamp STARTTIMESTAMP, auction INTEGER, bidder INTEGER, datetime LONG,    price DOUBLE)
    WRAPPER 'GenericPush'
    PROTOCOL 'CSV'
    TRANSPORT 'File'
    DATAHANDLER 'Tuple'
    OPTIONS ( 'filename' 'E:\test')
```