Query Analyzer and Processlist data

sniffer.data

```
CREATE TABLE IF NOT EXISTS [query_master](
    [id] INTEGER PRIMARY KEY AUTOINCREMENT,
    [query] TEXT,
    UNIQUE([query]))

CREATE TABLE IF NOT EXISTS [query_snapshot] (
    [pkeyid] INTEGER PRIMARY KEY AUTOINCREMENT,
    [id] INTEGER,
    [threadid] INTEGER,
    [user] TEXT,
    [querytime] INTEGER,
    [uptime] INTEGER,
    [host] TEXT DEFAULT)
```

Here, you see the same pattern as above: the [id] column in the [query_snapshot] table identifies a row in the [query_master] where the actual/textual query is saved. Also note, that a UNIQUE KEY is defined on the [query] column so that we can use an INSERT ... ON DUPLICATE KEY construction and thus ensure that the [query_master] table only has the same query stored once. But in [query_snapshot] table there is one row for every instance of the query.

Actually, with general/slow query log analyzers we use identical tables. The log CHUNK retrieved from the server isparsed and the tables populated like - you see in your sniffer.data database. The tables used for log analysis, however, are MEMORY tables and they are only available in SQL DM for MySQL and only for as long as they are needed.

Processlist

Also, SQL DM for MySQL processlist feature uses a SQLite MEMORY table (for every server). The table structure is as follows:

```
CREATE TEMPORARY TABLE IF NOT EXISTS [processlist](
  [Id] INTEGER NOT NULL PRIMARY KEY,
  [User] TEXT,
  [Host] TEXT,
  [Db] TEXT,
  [Command] TEXT,
  [Time] INTEGER,
  [State] TEXT,
  [Info] TEXT,
  [Action] TEXT)
```

So, that is how the MySQL processlist displays in SQL DM for MySQL - unlike when connected to MySQL directly - it can be filtered, sorted etc. by using WHERE, ORDER BY, GROUP BY etc. with a SELECT query against the SQL DM for MySQL [processlist] table. But as it is a MEMORY table you can only query it from inside the SQL DM for MySQL processlist interface.

Information about the SQL DM for MySQL database schema itself

There is a schema_version table in all databases created by SQL DM for MySQL. Every time SQL DM for MySQL starts it checks here if the database is up to date with the current program version. If it is not, SQL DM for MySQL will perform the necessary schema upgrades at start-up. Schema definition reads as follows:

```
CREATE TABLE IF NOT EXISTS [schema_version] (
    [schema_desc] TEXT,
    [schema_major_version] TEXT,
    [schema_minor_version] TEXT,
    PRIMARY KEY ([schema_major_version], [schema_minor_version]))
```