

# FairDB dynamical SQL processing interface

Denis Bertini<sup>1</sup>

<sup>1</sup>Scientific Computing ,GSI, Darmstadt, Germany

## Introduction

The FairRoot Virtual Database FairDB [1] uses internally the services of the ROOT classes TSQLServer and TSQLStatement to send and execute SQL commands with any of the supported database engine i.e MySQL, Oracle, PostGreSQL and SQLite.

Unfortunately the ROOT framework does only provide low-level drivers to the different SQL database engine. In order to properly interface the different SQL servers, FairDB adds additionnal layer on top of the ROOT framework for multiple database connection handling and SQL automatic translation.

## Database multiple connection

FairDB is designed to give the user the possibility to connect to and access data from more that one database. At initialization time, a pool of opened connections to different database engines is created using URL based syntax supported by the TSQLServer class.

The user can simply grap one when a transaction needs to be executed and then hand it back eliminating much of the overhead for the creation of connections.

If one connection fails, the next database in the list is used until the complete set of data can be retrieved. This gives the user the flexibility to create its own database from a subset of the official one and to put it ahead in the list.

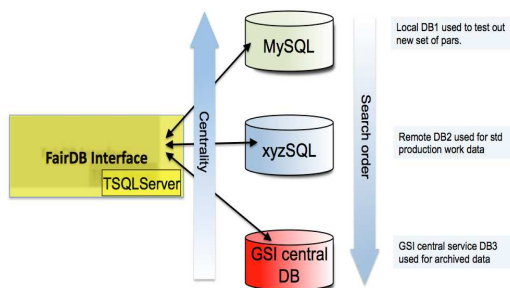


Figure 1: Database connection pool

## SQL automatic translation

Since FairDB supports different database engine, it uses a special pre-processing of a given SQL statement in order to translate the statement if needed (Fig [2]). If a SQL translation is needed, a fast parsing is processed and a new adapted SQL statement is dynamically created with

all translated tokens. The special SQL parsing and translation is done by the main SQL statement service class FairDbSQLStatement.

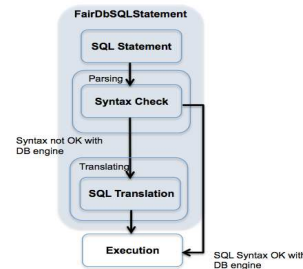


Figure 2: FairDbSQLStatement SQL automatic translation mechanism

## New ROOT PostGreSQL driver

Required by FairDB, the ROOT low level driver TPgSQLServer has been recently corrected and partially rewritten using the libPq interface to allow for dynamical table information query. The patch has been accepted and is part of the new ROOT release 5-34-15 [2]. The new driver together with the automatic translation give the user the possibility to store and retrieve parameter with PostGreSQL database centrally supported by the GSI Core-IT team (Fig [3]).

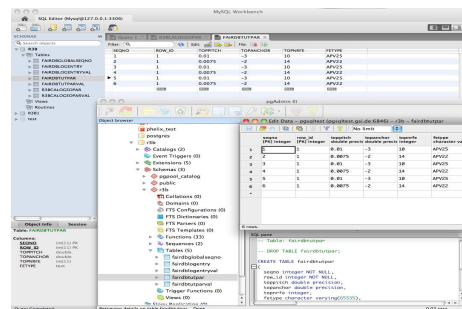


Figure 3: Same Parameter Table created with automatic SQL translation for MySQL and for PostGreSQL

## References

- [1] FairRoot Virtual Database (User Manual).
- [2] R. Brun, F. Rademakers, P. Canal, I. Antcheva, D. Buskulic, O. Couet, A. and M. Gheata *ROOT User Guide* CERN, Geneva 2005.

