Subject: code to more than one database and your threads Posted by BetoValle on Mon, 08 Feb 2021 01:18:08 GMT View Forum Message <> Reply to Message

Hi,

how do I adapt the server structure, to load more than one database and in different situations?

at the beginning there is only 1 database. In this situation the posted examples include. But it is assumed that in the same access create 2 other databases (all mysql) and make access etc.. along with the 1st database originally created.

how to build access with the 2 threads? how would it be?

```
SKYLARK(q, "qualqueres:POST")
{
//..code to create or open others databases
//..and thread(s)
}
```

```
could you give a simple example?
```

```
//first database when load server...
void OpenSQL(MySqlSession& session)
{
    if(!session.Connect("user", "password", "database")) {
        SKYLARKLOG("Can't connect to database");
        Exit(1);
    }
    SQL = session;
    }
    void demoTeste::WorkThread()
    {
        MySqlSession session;
        OpenSQL(session);
        RunThread();
    }
```

Thanks

BetoValle wrote on Mon, 08 February 2021 02:18Hi,

how do I adapt the server structure, to load more than one database and in different situations?

at the beginning there is only 1 database. In this situation the posted examples include. But it is assumed that in the same access create 2 other databases (all mysql) and make access etc.. along with the 1st database originally created.

how to build access with the 2 threads? how would it be?

```
SKYLARK(q, "qualqueres:POST")
{
//..code to create or open others databases
//..and thread(s)
}
```

```
could you give a simple example?
```

```
//first database when load server...
void OpenSQL(MySqlSession& session)
{
```

```
if(!session.Connect("user", "password", "database")) {
   SKYLARKLOG("Can't connect to database");
   Exit(1);
  }
  SQL = session;
}
```

```
void demoTeste::WorkThread()
{
    MySqlSession session;
    OpenSQL(session);
    RunThread();
}
```

Thanks

Well, obviously with 2 databases connected at the same time, you cannot use global database SQL for at least one of these.

That said, w.r.t. threading, there is a mode that keeps SQL thread-local: Check SqlSession::PerThread. In that case one thread can have different default global SqlSession than another.

Page 3 of 3 ---- Generated from U++ Forum