Subject: Database portability: Sqlite3
Posted by fabio on Fri, 08 Dec 2006 22:02:49 GMT
View Forum Message <> Reply to Message

I like database portability, yet complete portability is never possible.....

In uppsrc/plugin/sqlite3/Sqlite3upp.cpp .....

```
Vector<String> Sqlite3Session::EnumDatabases() {
 // In theory, sqlite3 can "ATTACH" multiple databases (up to 10).
 // However, I don't know how to list them.
 Vector<String> out;
 out.Add(current_dbname);
 return out;
}

Vector<String> Sqlite3Session::EnumTables(String database) {
 // Ignores database
 Vector<String> out;
 Sql sql(*this);
 sql*Select(SqlId("tbl_name")).From(SqlId("sqlite_master")).Where(SqlId("type")=="table");
 while (sql.Fetch())
  out.Add(sql[0]);
 return out;
}
```

The solution (from file uppsrc/plugin/sqlite3/lib/shell.c) is:

for get list of databases:
PRAGMA database_list;

for get the list of all tables and views:
SELECT name FROM sqlite_master WHERE type IN ('table','view') AND name NOT LIKE
'sqlite_%'
ORDER BY 1;

for get list of column:
PRAGMA table_info(table);

and for date emulation problem .....

the functions:
const void *sqlite3_column_decltype16(sqlite3_stmt*,int); //UTF-16
or
const char *sqlite3_column_decltype(sqlite3_stmt *, int i); // UTF-8

return a c-string with type of column or null pointer if computed column.
For declared 'date'-column return the pseudo-type 'date' and not the real type 'text'.

Real type SQLITE_TEXT (3) is returned from function:
int sqlite3_column_type(sqlite3_stmt*, int iCol);
(1,2,3,4,5) or  (SQLITE_INTEGER,SQLITE_FLOAT,SQLITE_TEXT,SQLITE_BLOB,SQLITE_NULL)

one solution is test the two value SQLITE_TEXT && 'date' for detect colum of pseudo-type date

consequently ....

My first little (useless?) contribution is:
Sqlite3Session::EnumDatabases(), modified for database list
Sqlite3Session::EnumTables(),  modified for database name support
Sqlite3Session::EnumViews(),  created
Sqlite3Session::EnumColumns(),  created

```
Vector<String> Sqlite3Session::EnumDatabases() {
	Vector<String> out;
 Sql sql(*this);
 sql.Execute("PRAGMA database_list;");
 while (sql.Fetch())
		out.Add(sql[1]);  // sql[1] is database name, sql[2] is filename
 return out;
}

Vector<String> Sqlite3Session::EnumTables(String database) {
 Vector<String> out;
	String dbn=database;
	if(dbn.IsEmpty()) dbn=current_dbname; // for backward compatibility
 Sql sql(*this);
	sql.Execute("SELECT name FROM "+dbn+".sqlite_master WHERE type='table' AND name NOT LIKE 'sqlite_%' ORDER BY 1;");
 while (sql.Fetch())
  out.Add(sql[0]);
 return out;
}

Vector<String> Sqlite3Session::EnumViews(String database) {
 Vector<String> out;
	String dbn=database;
	if(dbn.IsEmpty()) dbn=current_dbname;
 Sql sql(*this);
	sql.Execute("SELECT name FROM "+dbn+".sqlite_master WHERE type='view' AND name NOT LIKE 'sqlite_%' ORDER BY 1;");
 while (sql.Fetch())
```

```
  out.Add(sql[0]);
 return out;
}
```

and a few changes of
void Sqlite3Connection::BindParam(int i, const Value& r)
bool Sqlite3Connection::Execute()
void Sqlite3Connection::GetColumn(int i, Ref f)

for date emulation.....

the result work with EditDate, SqlCtrl, SqlArray.

Remain the problem of aggregate function like max and min, but Sqlite3 permit creation of user function ..... and so .....

In the attached file (zip):
new uppsrc/plugin/sqlite3/sqlite3.h
new uppsrc/plugin/sqlite3/sqlite3upp.cpp
Sqlite3upp_changes.txt

Mirek can you modify the source of distribution from the attached file ? Thanks.

Fabio

## File Attachments

1) Sqlite3upp_Upgrade.zip, downloaded 1564 times