

[管理]



Qing



博客等级: 18

博客积分: 78篇

博客访问: 407,824

关注人气: **378** 

获赠金笔: 12

赠出金笔: 0









### 相关博文

更多>>

## 推荐博文

- 2550mAh电
- ■日本雾霾之战: 民众与政府的博弈
- 南非猎豹冒死捕杀豪猪被扎满嘴刺
- ■别光顾着把枪口对准柴静
- 马来西亚男子与眼镜王蛇接吻(图
- 【早评】降息利好将会在后市中逐
- ■哪些人能从柴静的"穹顶之下"赚
- "深圳机场撞人事件"不是一个人
- ■政采剔除国外品牌: 早该说不了!
- ■【DIY】做一枚宫灯迎元宵





寻找撒尿小孩儿 38年

标签: qt sqlite qsqldatabase qsqlquery qsqlerror 分类: Qt

+ 转 载 ▼

Qt之操作数据库(SQLite) ■ (2013-12-09 16:58:05) [编辑][删除]

QtSq1模块提供了与平台以及数据库种类无关的访问SQL数据库的接口,这个接口由利用Qt的模型视图结构将 数据库与用户界面集成的一套类来支持。

QSqlDatabase对象象征了数据库的关联。Qt使用驱动程序与各种数据库的应用编程接口进行通信。Qt的桌面 版 (Desktop Edition)包括如下一些驱动程序:

驱动程序	数据库
QDB2	IBM DB2 7.1版以及更新的版本
QIBASE	Borland InterBase
QMYSQL	MySqI
QOCI	甲骨文公司(Oracle Call Interface)
QODBC	ODBC (包括微软公司的QSL服务)
QPSQL	PostgreSQL的7.3版以及更高版本
QSQLITE	QSLite第3版
QSQLITE2	QSLite第2版
QTDS	Qybase自适应服务器

由于授权的许可限制,Qt的开源版本无法提供所有的驱动程序,当配置Qt时,即可以选择Qt本身包含的SQL 驱动程序,也可以以查件的形式建立驱动程序,公共领域中不断发展的SQLite数据库将向Qt提供支持。 如下讨论关于Qt进行SQLite的基本操作。

# 代码如下:

} else

```
//添加数据库驱动、设置数据库名称、数据库登录用户名、密码
QSqlDatabase database = QSqlDatabase::addDatabase("QSQLITE");
database.setDatabaseName("database.db");
database.setUserName("root");
database.setPassword("123456");
//打开数据库
if(!database.open())
qDebug()<<database.lastError();</pre>
qFatal("failed to connect.");
```

//QSqlQuery类提供执行和操作的SQL语句的方法。

//可以用来执行DML(数据操作语言)语句,如SELECT、INSERT、UPDATE、DELETE,

//以及DDL (数据定义语言) 语句,例如CREATE TABLE。

//也可以用来执行那些不是标准的SQL的数据库特定的命令。

QSqlQuery sql query;

QString create\_sql = "create table student (id int primary key, name varchar(30), age int)";





鲨鱼爱攻击男性 冲浪者

非洲雄狮撕咬河 马尸体





揭秘世界最毒死 亡之蛙

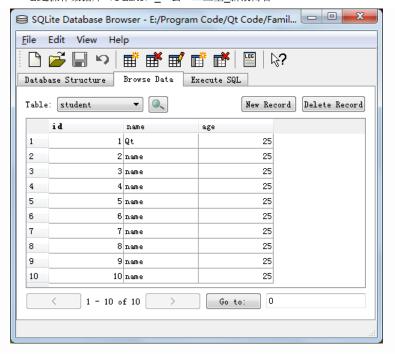
40岁的灵魂歌 顺子

查看更多>>

#### 谁看过这篇博文 todaysky 14分钟前 🙎 风中精灵 3月2日 🙎 一方通行 3月2日 🙎 飞翔的70哥 3月2日 booooooo... 2月28日 🙎 han. wei. g… 2月27日 🤦 学电子 2月26日 🙎 flyingbear 2月24日 🤦 众乐乐才… 2月20日 🙎 蓝色鼓点 2月17日 **2** 625747 2月17日 🙎 影无痕 2月15日

```
QString select_max_sql = "select max(id) from student";
QString insert_sql = "insert into student values (?, ?, ?)";
QString update_sql = "update student set name = :name where id = :id";
QString select_sql = "select id, name from student";
QString select_all_sql = "select * from student";
QString delete_sql = "delete from student where id = ?";
QString clear_sql = "delete from student";
sql_query. prepare(create_sql);
if(!sql_query.exec())
{
qDebug()<<sql_query.lastError();</pre>
}
else
{
qDebug()<<"table created!";</pre>
//查询最大id
int \max_{i} = 0;
sql_query.prepare(select_max_sql);
if(!sql_query.exec())
qDebug()<<sql_query.lastError();</pre>
}
else
{
while(sql_query.next())
max_id = sql_query.value(0).toInt();
qDebug()<<QString("max id:%1").arg(max_id);</pre>
//插入数据
sql_query.prepare(insert_sql);
sql_query.addBindValue(max_id+1);
sql_query.addBindValue("name");
sql_query.addBindValue(25);
if(!sql_query.exec())
\verb|qDebug|() << \verb|sql_query.lastError|()|;
}
else
{
qDebug()<<"inserted!";</pre>
}
//更新数据
sql_query.prepare(update_sql);
sql_query.bindValue(":name", "Qt");
sql_query.bindValue(":id", 1);
if(!sql_query.exec())
qDebug()<<sql_query.lastError();</pre>
}
else
qDebug()<<"updated!";</pre>
//查询部分数据
if(!sql_query.exec(select_sql))
qDebug()<<sql_query.lastError();</pre>
}
else
```

```
\  \, \text{while} \, (\text{sql\_query.next}\, ()\, )
int id = sql_query.value("id").toInt();
QString name = sql_query.value("name").toString();
qDebug()<<QString("id:%1 name:%2").arg(id).arg(name);</pre>
//查询所有数据
sql_query.prepare(select_all_sql);
if(!sql_query.exec())
qDebug()<<sql_query.lastError();</pre>
else
while(sql_query.next())
int id = sql_query.value(0).toInt();
QString name = sql_query.value(1).toString();
int age = sql_query.value(2).toInt();
qDebug()<<QString("id:%1 name:%2 age:%3").arg(id).arg(name).arg(age);</pre>
//删除数据
sql_query.prepare(delete_sql);
sql\_query.addBindValue(max\_id);
if(!sql_query.exec())
{
qDebug()<<sql_query.lastError();</pre>
}
else
qDebug()<<"deleted!";</pre>
//清空表
sql_query.prepare(clear_sql);
if(!sql_query.exec())
qDebug()<<sql_query.lastError();</pre>
}
else
qDebug()<<"cleared";</pre>
//关闭数据库
database.close();
//删除数据库
QFile::remove("database.db");
可以通过一些工具对SQLite进行管理,如下:
```



### SQLite的管理

管理工具挺多的,这里简单介绍几款:

- SQLite Manager: 开放源代码的SQLite管理工具,用来管理本地电脑上的SQLite数据库,可以独立运行(以 XULRunner方式),也可以作为Firefox、Thunderbird、Seamonkey、Songbird、Komodo、Gecko等的插件。
- SQLite Administrator: 一个用来管理SQLite数据库文件的图形化工具,可进行创建、设计和管理操作。提供代码编辑器具有自动完成和语法着色,支持中文,适合初学者。
- SQLite Database browser: 一个SQLite数据库的轻量级GUI客户端,基于Qt库开发,界面清洁,操作简单,主要是为非技术用户创建、修改和编辑SQLite数据库的工具,使用向导方式实现。

更多关于SQLite的介绍与使用,请参考:

- Connect to Sqlite and do insert, delete, update and select.
- 全方位认识SQLite(一).
- 全方位认识SQLite(二).

### 注:

技术在于交流、沟通, 转载请注明出处并保持作品的完整性。

作者: └☆奋斗ing♥孩子` 原文: http://blog. sina. com. cn/s/blog\_a6fb6cc90101gx30. html.





做第一个评论者吧! 圖抢沙发>>



新浪BLOG意见反馈留言板 不良信息反馈 电话: 4006900000 提示音后按1键(按当地市话标准计费) 欢迎批评指正 新浪简介 | About Sina | 广告服务 | 联系我们 | 招聘信息 | 网站律师 | SINA English | 会员注册 | 产品答疑