

# 71 QSqlQueryModel

## 71.1 基于MV结构的数据库应用

一个典型的数据库应用往往需要将数据库中的数据显示在界面上，同时将用户通过界面对数据所做的修改反映到数据库中。Qt借助MV结构维系数据库与用户界面间的数据同步。关系型数据库中的数据保存在数据表中，而数据表的逻辑结构就是一张二维表格，这与QTableView视图组件的数据模型恰好是一致的。在Qt中，用于数据库访问的模型类包括QSqlQueryModel、QSqlTableModel和QSqlRelationalTableModel三个。其中：

- QSqlQueryModel类表示通过查询语句获得的结果集，其中的数据是只读的，不可编辑修改；
- QSqlTableModel类表示一张完整的数据表，其中的数据可以编辑修改；
- QSqlRelationalTableModel类在QSqlTableModel类的基础上，增加了对基于外键的表连接的支持。

## 71.2 常用方法

QSqlQueryModel类的常用方法如下：

```
1 void QSqlQueryModel::setQuery(QString& query); // 设置一条查询语句
2 void QSqlQueryModel::setQuery(QSqlQuery& query); // 设置一个查询对象
3 QSqlQuery QSqlQueryModel::query(); // 获取当前查询对象
4 int QSqlQueryModel::rowCount(); // 获取结果集记录数
5 int QSqlQueryModel::columnCount(); // 获取结果集字段数
6 QSqlRecord QSqlQueryModel::record(); // 获取包含字段信息的空
   记录
7 QSqlRecord QSqlQueryModel::record(int row); // 获取结果集中第row行
   记录
8 void QSqlQueryModel::clear(); // 清除所有查询到的数据
9 void QSqlQueryModel::setHeaderData(int section,
10 Qt::Orientation orientation, QVariant& value); // 设置表头数据
11 QSqlError QSqlQueryModel::lastError(); // 获取最近错误
```

调用QSqlQueryModel对象的setQuery方法，为其设置一条查询语句，与该对象关联的视图组件即可显示出执行该查询语句所得到的结果集，只可浏览，不可编辑。