

1.Qt SQL模块包含的主要类

类名	功能描述
QSqlDatabase	用于建立与数据库的连接
QSqlDriver	用于访问具体的SQL数据库的底层抽象类
QSqlDriverCreator	为某个具体的数据库驱动提供SQL驱动的模板类
QSqlDriverCreatorBase	所以SQL驱动器的基类
QSqlDriverPlugin	用于定制QSqlDeiver插件的抽象基类
QSqlError	SQL数据库错误学习，可以用于访问上一次出错的学习
QSqlField	操作数据表或视图的字段类
QSqlIndex	操作数据库的索引类
QSqlQuery	执行各种SQL语句的类
QSqlQueryModel	SQL查询结果数据的只读数据模型。用于SELECT查询结果数据基类的只读显示
QSqlRecord	封装了数据记录操作的类
QSqlRelation	用于存储SQL外键信息的类，用于QSqlRelationTableModel数据源中设置代码字段与关联数据表的关系
QSqlRelationDelegate	用于QSqlRelationTableModel的一个代码字段的显示和编辑代理组件，一般是一个QComboBox组件，下拉列表中自动填充代码表的代码字段对应的实际内容
QSqlRelationTableModel	用于一个数据表的可编辑的数据模型，支持代码字段的外键
QSqlResult	访问SQL数据库的抽象接口
QSqlTableModel	编辑一个单一数据表的数据模型类
QDataWidgetMapper	用于界面组件与字段之间实现映射，实现字段内容自动显示的类

2.QSqlTableModel使用

(1) QSqlTabelModel类的主要函数

函数	功能描述
QSqlDatabase database()	返回其数据库连接
void setTable(QString &tableName)	设置数据表名称，不立即读取记录，但
QString tableName()	返回设置的数据表名称
void setFilter(QString &filter)	设置记录的过滤条件

void setSort(int column,Qt::SortOrder order)	设置排序字段的排序规则，需要调用sort
void sort(int column,Qt::SortOrder order)	按列号column和排序规则立即进行排序
void setEditStrategy(EditStrategy strategy)	设置编辑策略
bool setHeaderData(int section,Qt::Orientation orientation,QVariant &value)	设置表头，一般用于设置字段的显示名称
bool isDirty()	若有未更新到数据库的修改，返回true
int fieldIndex(QString &fieldName)	根据字段名称返回其在模型中的字段序号，不存在则返回-1
QSqlIndex primaryKey()	返回数据表的主索引
int rowCount()	返回记录条数
bool select()	查询数据表的数据，并使用设置的排序规则
bool selectRow(int row)	刷新获取指定行数的数据
void clear()	清楚数据模型，释放所有获取的数据
QSqlRecord record()	返回一条空记录，只有字段名，可用于插入
QSqlRecord record(int row)	返回行号为row的一条记录，包含记录值
bool setRecord(int row,QSqlRecord &values)	更新一条记录的数据到数据模型上，通过字段名称匹配，而不是位置匹配
bool insertRecord(int row,QSqlRecord &record)	在行号row之前插入一条数据
bool insertRows(int row,int count)	再行号row之前插入count空行，编辑策略为OnFieldChange或OnRowChange时有效
bool removeRows(int row,int count)	从行号row开始，删除count行，若编辑策略为OnManualSubmit，需要调用submit()里删除
void revertRow(int row)	取消行号为row的记录的修改
void revert()	编辑策略为OnFieldChange或OnRowChange时，取消当前行的修改，对编辑策略OnManualSubmit无效
void revertAll()	取消所有未提交的修改
bool submit()	提交当前行的修改到数据库，对编辑策略OnManualSubmit无效
bool submitAll()	提交所有未更新的修改到数据库，若成功则返回true，否则返回false，错误的详细信息可以从lastError()获取

## (2) QSqlRecord类的主要函数

--	--

函数	功能描述
void clear()	清除记录的所有字段定义和数据
void clearValues()	清除所有字段的数据，将字段数据内容设置为null
bool contains(QString &name)	判断记录是否含有名称为name的字段
bool isEmpty()	若记录里没有字段返回true
int count()	返回记录的字段个数
QString fieldName(int index)	返回序号为index的字段的名称
int indexOf(QString &name)	返回字段名称为name的字段的序号，如果字段不存在返回-1
QSqlField field(QString &name)	返回字段名称为name的字段对象
QVariant value(QString &name)	返回字段名称为name的字段的值
void setValue(QString &name,QVariant &val)	设置字段名称为name的字段的值为val
bool isNull(const QString &name)	判断字段名称为name的字段数据是否为null
void setNull(const QString &name)	设置名称为name的字段的值为null

(3) QSqlField类的主要函数

函数	功能描述
void clear()	清除字段数据，设置为NULL；如果字段是只读的，则不清除
bool isNull()	判断字段值是否为NULL
bool setReadOnly(bool readOnly)	设置一个字段为只读，只读的字段不能使用setValue()设置值，或使用clear()清值
QVariant value()	返回字段的值
void setValue(QVariant &value)	设置字段的值

3.QSqlQueryModel使用

(1) QSqlQueryModel类的主要函数

函数	功能描述
void clear()	清除数据模型，是否所有获得的值
QSqlError lastError()	返回上次的错误，可获得错误的类型和
QSqlQuery query()	返回当前关联的QSqlQuery对象

void setQuery(QSqlQuery &query)	设置一个QSqlQuery对象，获取数据
void setQuery(QString &query)	设置一个SELECT语句创建查询，获取数据
QSqlRecord record()	返回一个空记录，包含当前查询的字段
QSqlRecord record(int row)	返回行号为row的记录
int rowCount()	返回查询到的记录条数
int columnCount()	返回查询的字段个数
void setHeaderData(int section,Qt::Orientation orientation,QVariant &value)	设置表头数据，一般用于设置字段的表头

#### 4.QSqlQuery使用

函数	功能描述
bool prepare(QString &query)	设置准备执行的SQL语句，一般用于带参数的SQL语句
void bindValue(QString &placeholder,QVariant &val)	设置SQL语句中参数的值，以占位符表示参数
void exec()	执行由prepare()和bindValue()设置的SQL语句
void exec(QString &query)	直接执行一个不带参数的SQL语句
bool isActive()	如果成功执行了exec()就返回true
bool isSelected()	如果执行的SQL语句是SELECT语句，返回true
QSqlRecord record()	返回当前记录
QVariant value(QString &name)	放回当前记录名称为name的字段
bool isNull(QString &name)	判断一个字段是否为空，当query非活动、未定位再有效记录、无此字段或字段为空时都返回true
int size()	对于SELECT语句，返回查询到的记录条数，其他语句返回-1
int numRowsAffected()	返回SQL语句影响的记录条数，对于SELECT语句无定义
bool first()	定位到第一条记录，isActive和isSelected都为true时才有效
bool previous()	定位到上一条记录，isActive和isSelected都为true时才有效
bool next()	定位到下一条记录，isActive和isSelected都为true时才有效
bool last()	定位到最后一条记录，isActive和isSelected都为true时才有效
bool seek(int index)	定位到指定序号的记录

int at()	返回当前记录的序号
----------	-----------