## QT连接数据库

用SQLite建立数据库，将得到的数据库文件放在工程目录的debug文件夹下即可通过C++程序访问并修改数据库。

以下是尝试连接数据库并取得数据的代码：

|  |
| --- |
| #include <QApplication>  #include <QMessageBox>  #include <iostream>  #include <QtDebug>  #include <QSqlDatabase>  #include <QtSql>  int main(int argc, char \*argv[]){  QApplication app(argc, argv);  MainWindow w;  w.show();  QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");  std::cout<<"START"<<std::endl;  db.setDatabaseName("try.db");  if ( !db.open()){  QMessageBox::critical(NULL, QObject::tr("Collection"), QObject::tr("failed to connect to database!"));  return 0;  }  QSqlQuery query;  query.exec("select \* from course");  //对数据库进行操作，insert、create等均可  while(query.next()){  int one=query.value(1).toInt();  QString two=query.value(0).toString();  qDebug()<<one<<","<<two;}//while  std::cout<<"END"<<std::endl;  db.close();  return app.exec();  } |

运行结果：

START

1 , "语文"

3 , "英语"

2 , "数学"

END

## 数据库有关培养方案的TABLES（粗略版）

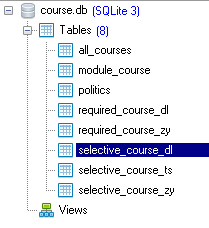


Figure 数据库结构

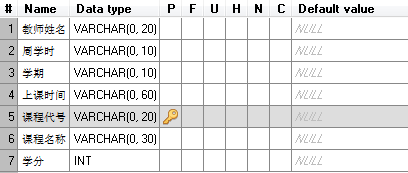


Figure 列出所有课程的表单结构

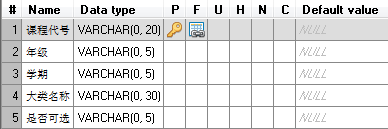


Figure 大类必修课

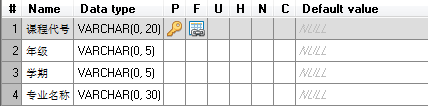


Figure 专业必修

其余表单分别表示通识课、大类选修课、模块选修、专业选修等。表格结构相似。