**数据库原理与技术实验课设——长理教务管理系统**

一、数据库设计

1.概念结构设计。

确定表：

实现内容：学生表，课程表，选课表

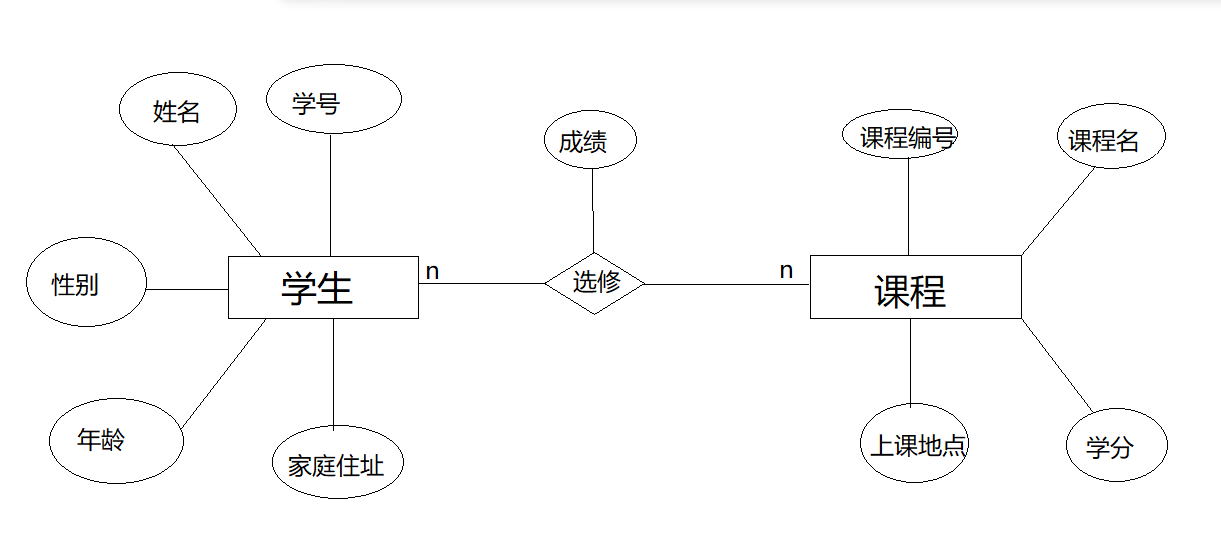
学生表的关键属性有：学号，姓名，性别，年龄，家庭住址。其中，学号（设为标识）为主键。

课程表的关键属性有：课程编号，课程名，上课地点（云塘，金盆岭），学分。其中，课程编号（设为标识）为主键。

选课表的关键属性有：学生学号，课程编号，成绩。其中（学生学号，课程编号）为主键，学生学号为参考 学生表（学号）的外键，课程编号为参考 课程表（课程编号）的外键。

E-R图：







2.逻辑结构设计。

关系模型：

学生：学号， 姓名，性别，年龄，家庭住址。

课程：课程编号，课程名，上课地点，学分。

选课：学生学号，课程编号，成绩。

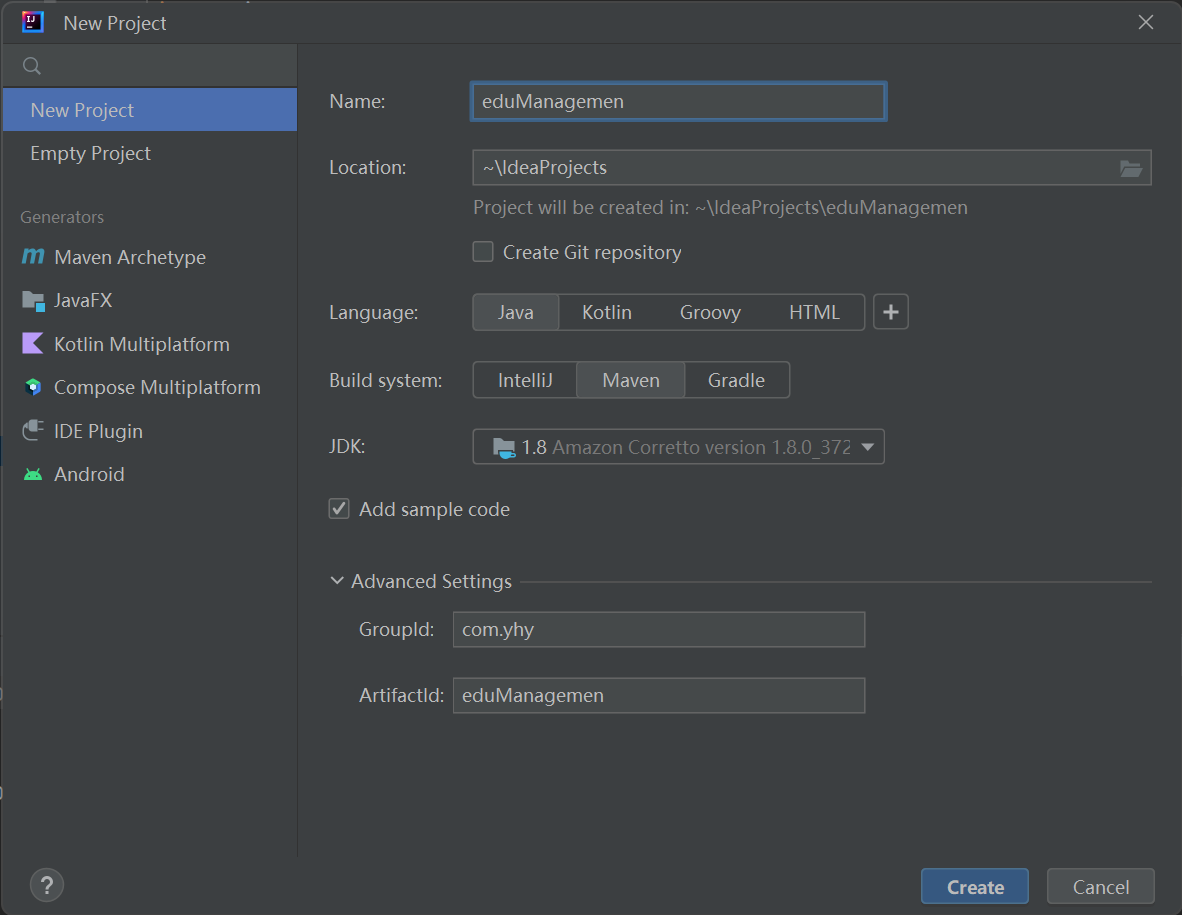


二、应用系统实现

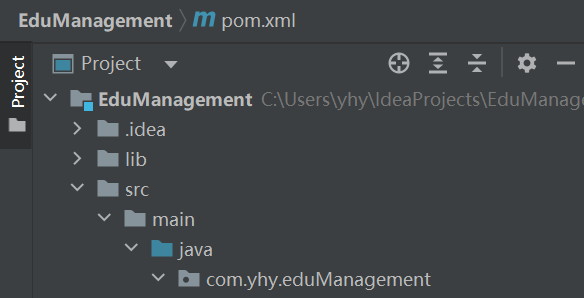


1.数据库连接

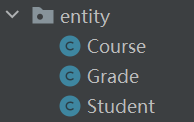
1.打开IDEA,新建项目（组id命名：com.yhy）

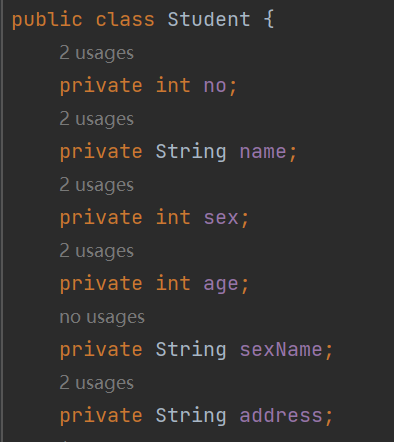


2.搭建基本框架



3.创建新软件包（package）entity，并实现对应实体类

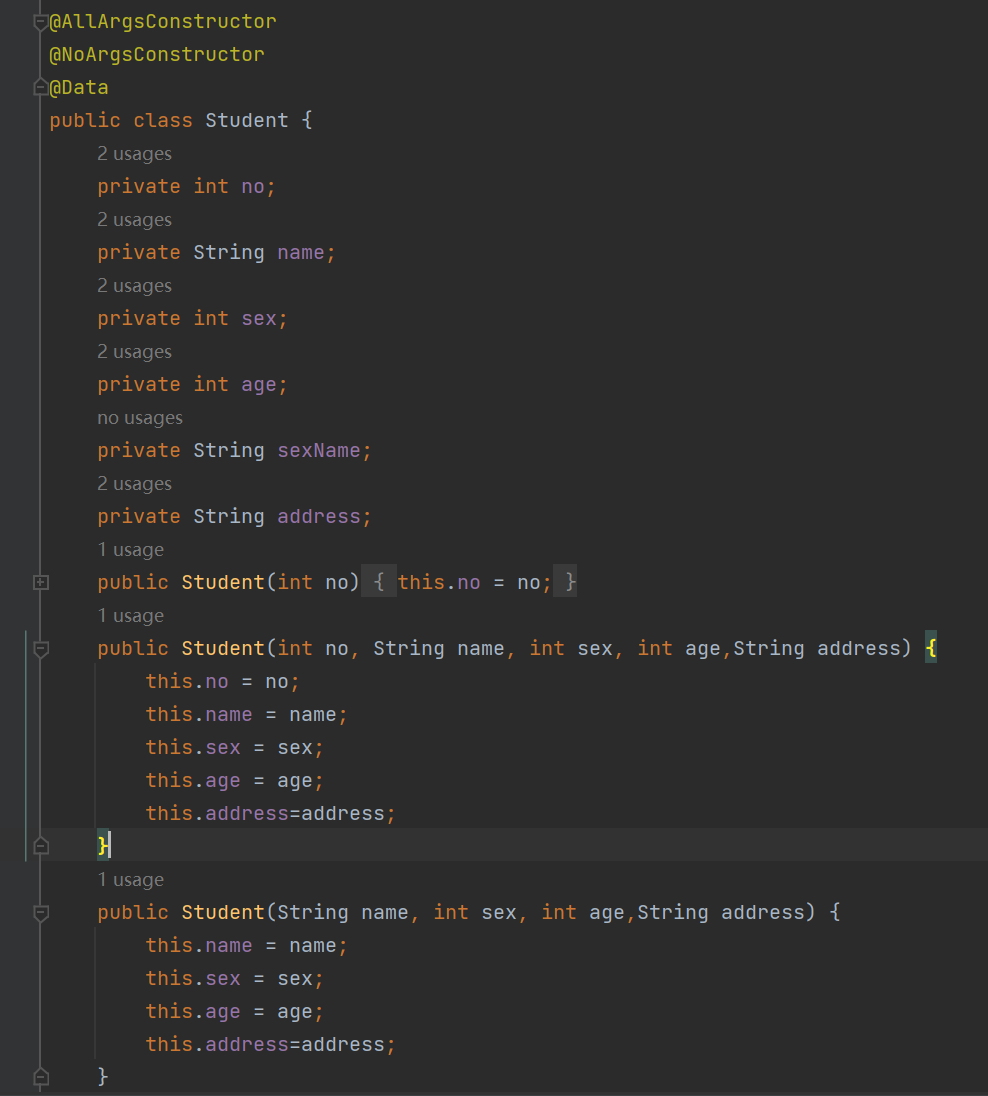


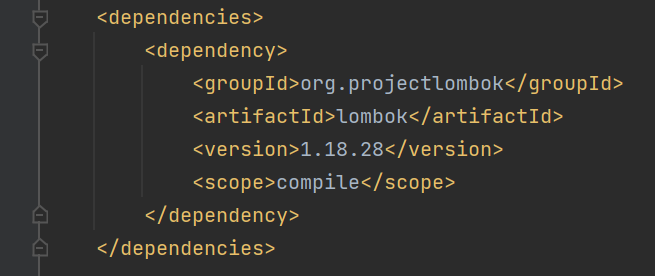


4.实现实体类的setter、getter方法，并实现构造器（不包含no）,

在类名上方敲上一个任意的lombok注解, 然后鼠标放上去，点击将lombok添加到类路径

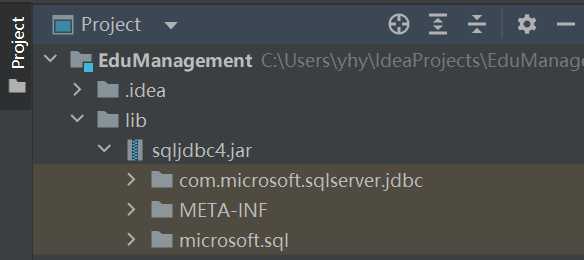
中



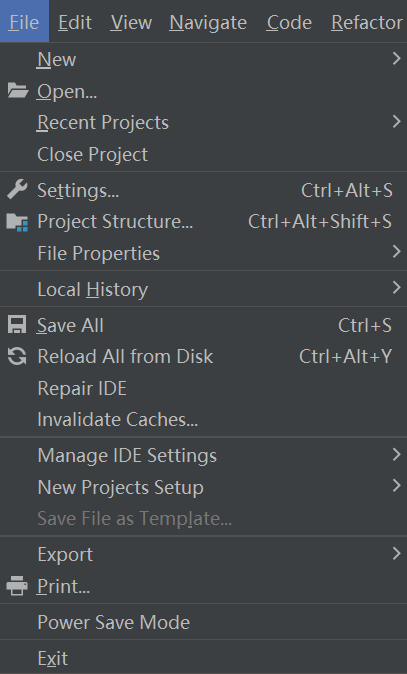


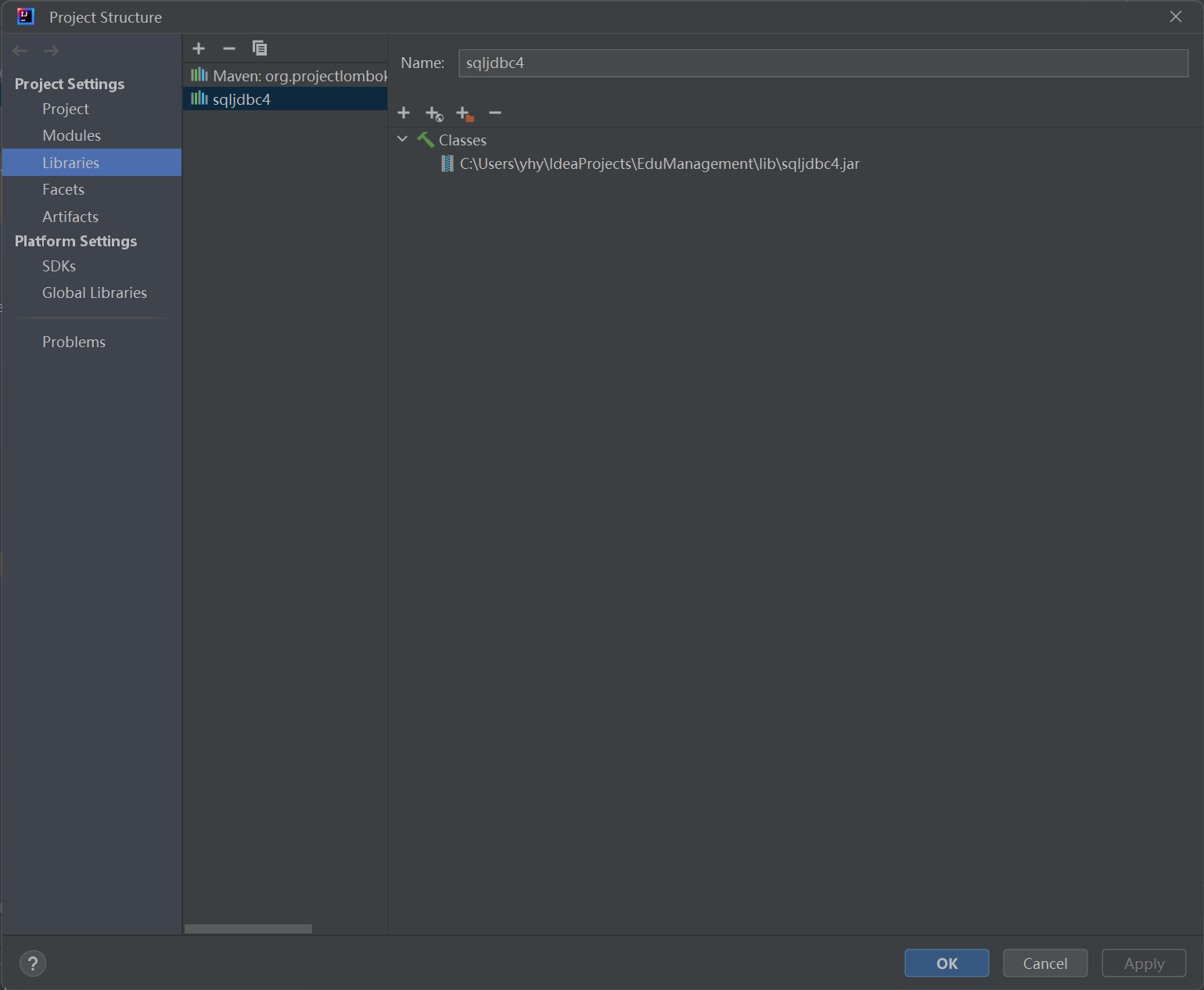
5.引入java数据库连接驱动Jdbc

在项目中创建一个名为lib的目录（Directory）,将事先下载好的sqljdbc放入其中。



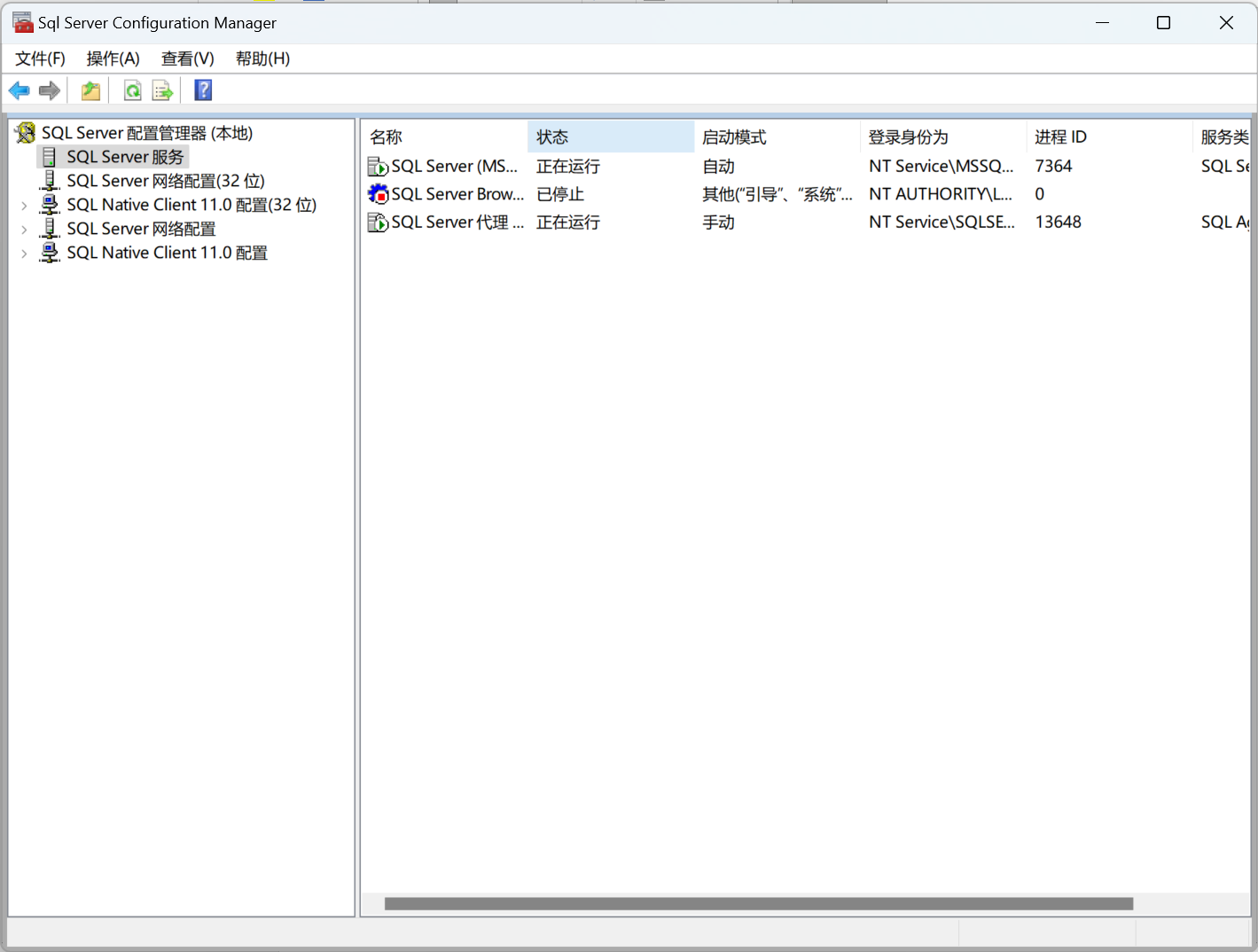
然后点击idea上方的File里的Project Structure，将sqljdbc添加到Libraries中。

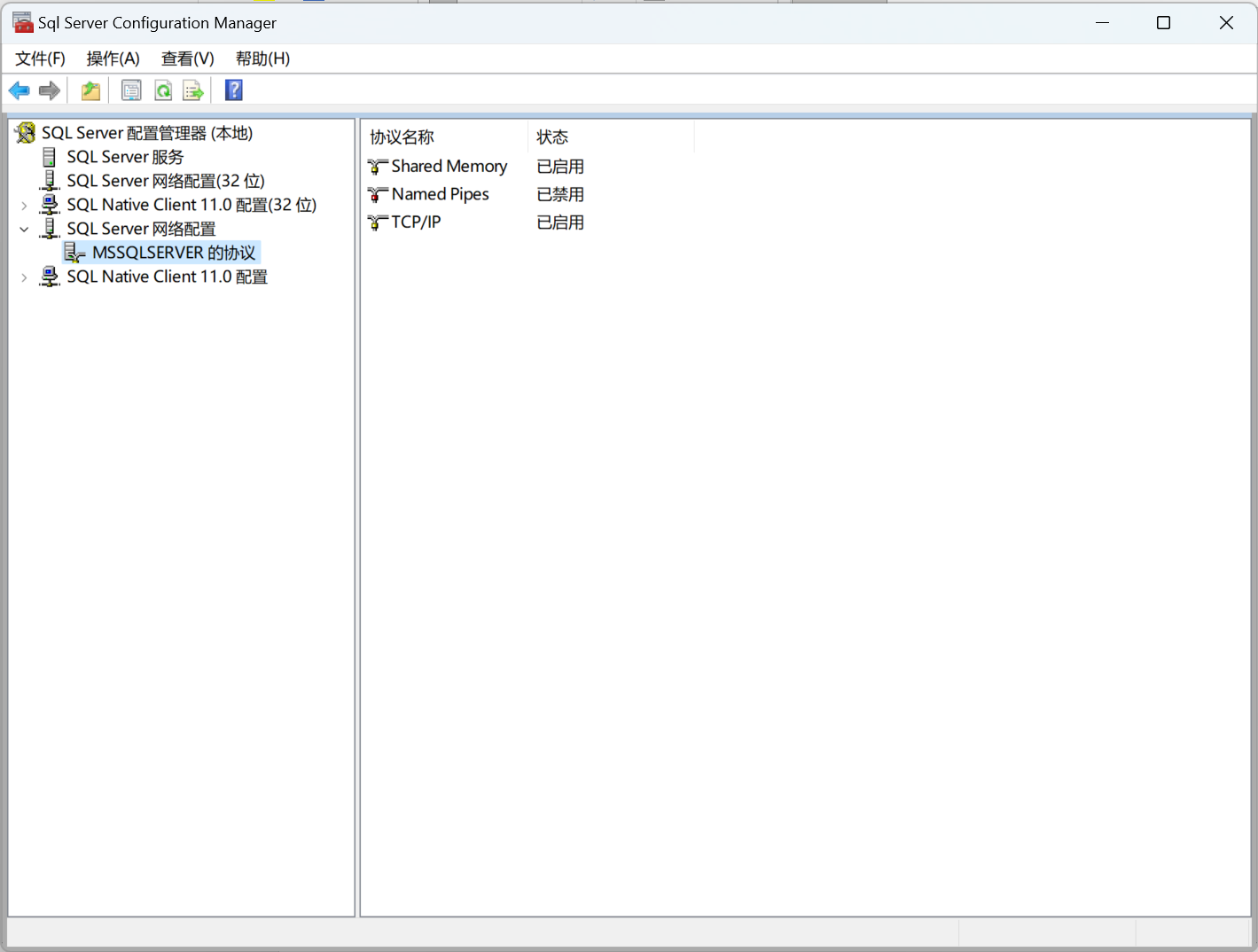




6.配置SQL Server

配置如图所示：





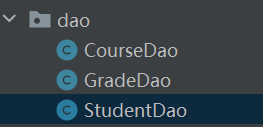


7.建立数据库连接工具类DBTool，如下面代码所示，之后可以通过conn对象来进行数据库的操作。

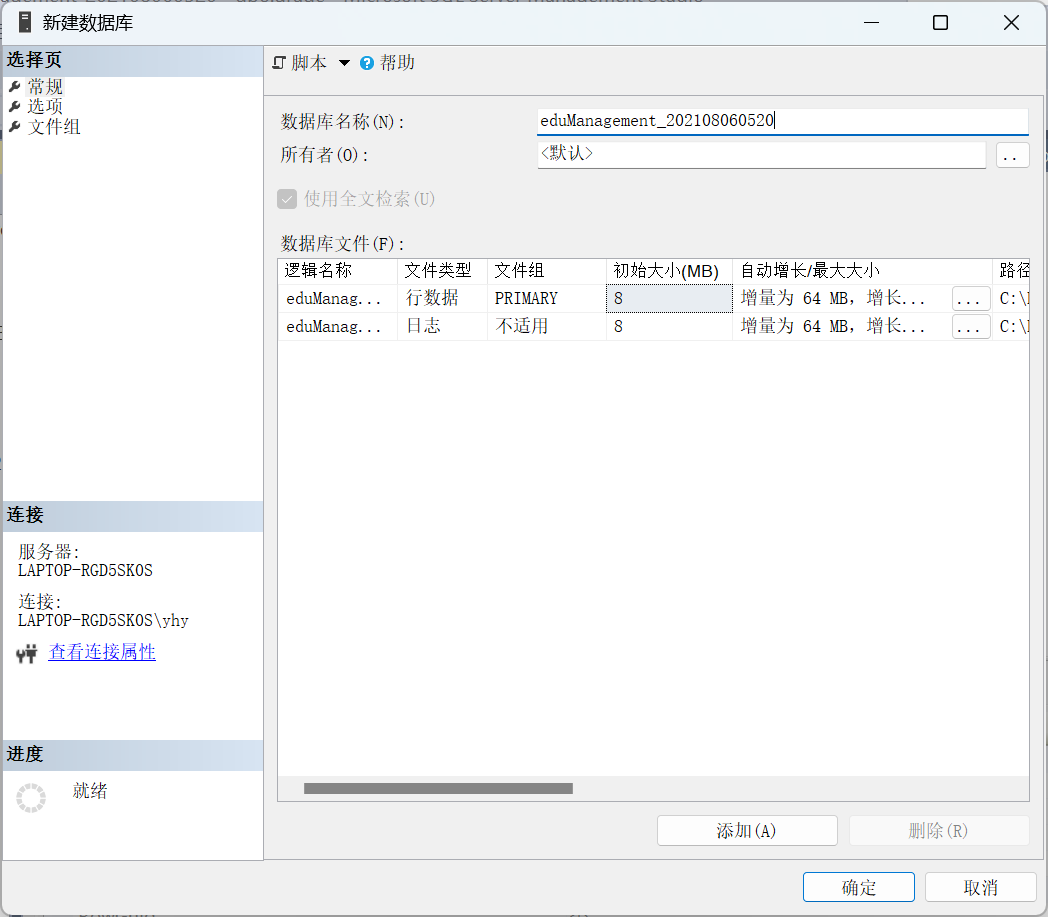
package com.yhy.eduManagement.util;  
  
import java.sql.\*;  
  
public class DBTool {  
 private static String *databaseName* = "eduManagement\_202108060520";//数据库名字  
 private static String *username* = "sa";//数据库超管账号，默认sa  
 private static String *password* = "123";//数据库超管密码，安装时自己设置的  
 private static Connection *conn*;//与数据库的连接  
 public static Connection getConn() {  
 return *conn*;  
 }  
 static {  
 *init*();  
 }  
 public static boolean init(){  
 try {  
 String dbURL = "jdbc:sqlserver://localhost:1433;trustServerCertificate=true;databaseName=" + *databaseName*;  
 *conn* = DriverManager.*getConnection*(dbURL, *username*, *password*);  
 System.*out*.println("数据库初始化成功");  
 return true;  
 } catch (SQLException e) {  
 System.*out*.println("数据库初始化失败");  
 e.printStackTrace();  
 return false;  
 }  
 }  
}

在项目中新建dao包，在其中编写对与数据库操作相关的函数，比如说查找学生、添加学生、删

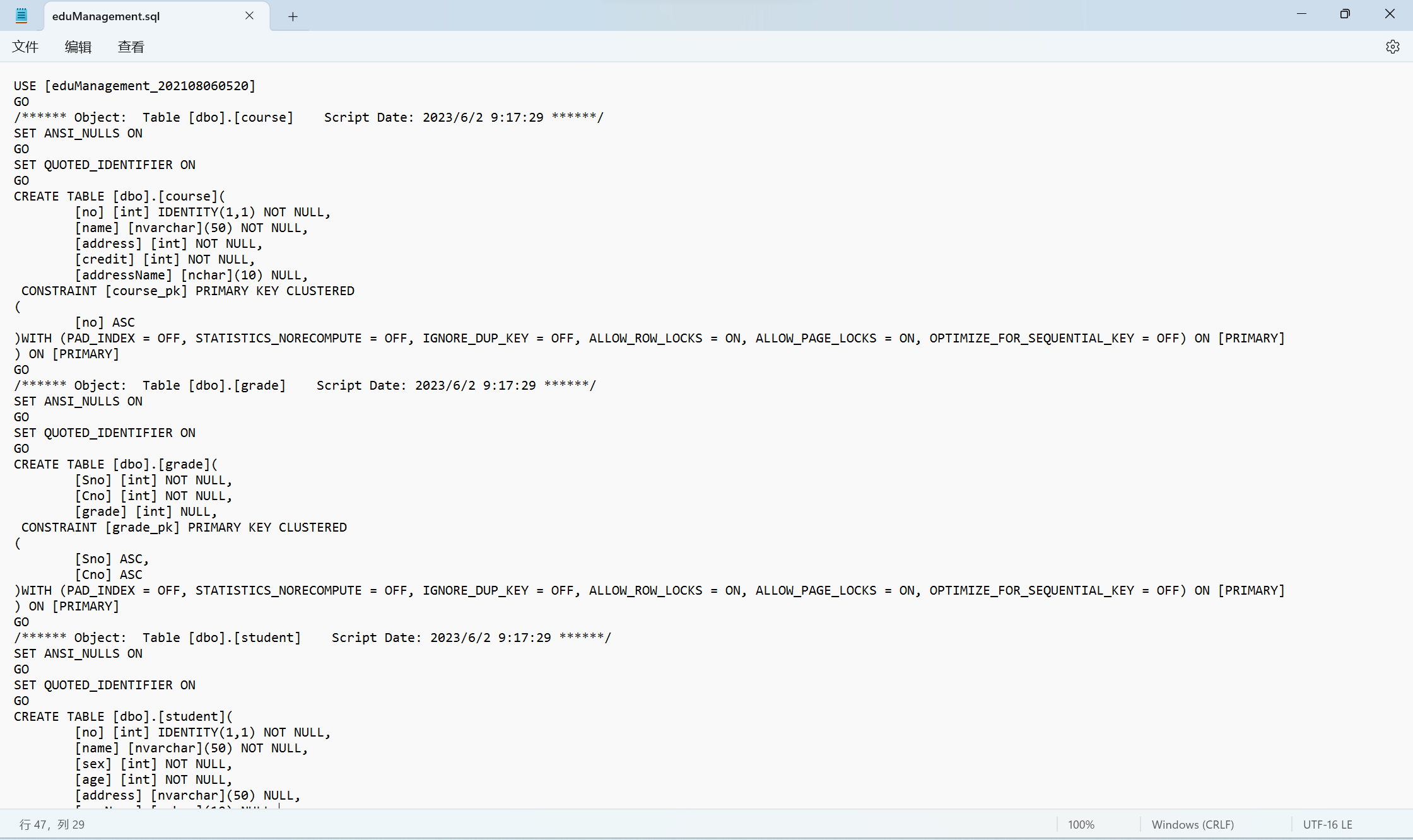
除学生、修改学生。

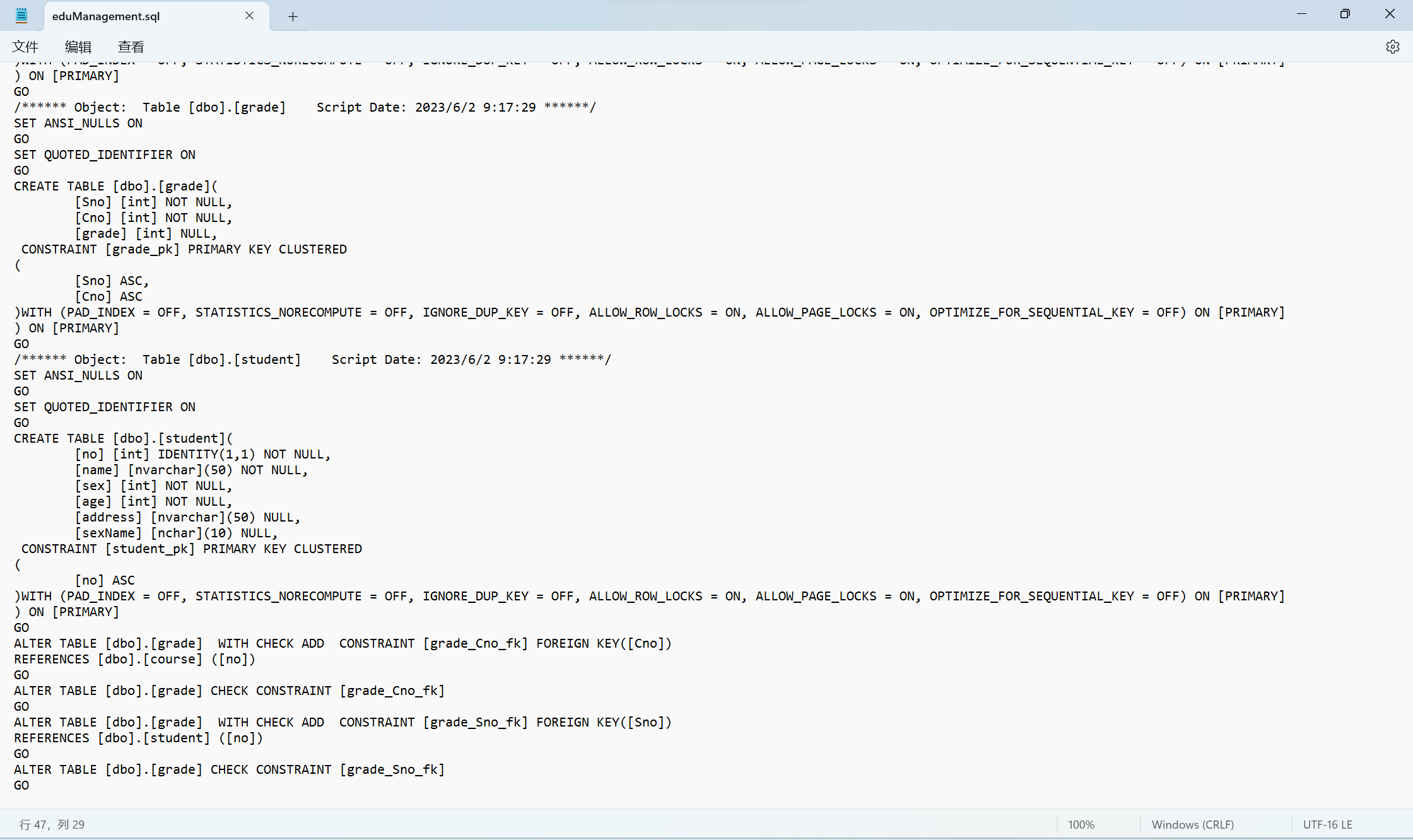


2.数据库的创建



3.数据表的创建与修改





4.数据查询

学生查询代码：

public static List<Student> selectStudents() throws SQLException {  
 String sql = "select \* from student";  
 Statement statement = DBTool.*getConn*().createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
 List<Student> students = new ArrayList<>();  
 while (resultSet.next()) {  
 Student student = new Student();  
 student.setNo(resultSet.getInt("no"));  
 student.setName(resultSet.getString("name"));  
 student.setSex(resultSet.getInt("sex"));  
 student.setAge(resultSet.getInt("age"));  
 student.setAddress(resultSet.getString("address"));  
 student.setSexName(*getStudentSexName*(student.getSex()));  
 students.add(student);  
 }  
 return students;  
}  
  
public static List<Student> selectStudents(String where) throws SQLException {  
 String sql = "select \* from student where " + where;  
 Statement statement = DBTool.*getConn*().createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
 List<Student> students = new ArrayList<>();  
 while (resultSet.next()) {  
 Student student = new Student();  
 student.setNo(resultSet.getInt("no"));  
 student.setName(resultSet.getString("name"));  
 student.setSex(resultSet.getInt("sex"));  
 student.setAge(resultSet.getInt("age"));  
 student.setAddress(resultSet.getString("address"));  
 student.setSexName(*getStudentSexName*(student.getSex()));  
 students.add(student);  
 }  
 return students;  
}

课程查询代码：

public static List<Course> selectCourses() throws SQLException {  
 String sql = "select \* from course";  
 Statement statement = DBTool.*getConn*().createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
 List<Course> courses = new ArrayList<>();  
 while (resultSet.next()) {  
 Course course = new Course();  
 course.setNo(resultSet.getInt("no"));  
 course.setName(resultSet.getString("name"));  
 course.setAddress(resultSet.getInt("address"));  
 course.setCredit(resultSet.getInt("credit"));  
 course.setAddressName(*getCourseAddressName*(course.getAddress()));  
 courses.add(course);  
 }  
 return courses;  
}  
  
public static List<Course> selectCourses(String where) throws SQLException {  
 String sql = "select \* from course where " + where;  
 Statement statement = DBTool.*getConn*().createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
 List<Course> courses = new ArrayList<>();  
 while (resultSet.next()) {  
 Course course = new Course();  
 course.setNo(resultSet.getInt("no"));  
 course.setName(resultSet.getString("name"));  
 course.setAddress(resultSet.getInt("address"));  
 course.setCredit(resultSet.getInt("credit"));  
 course.setAddressName(*getCourseAddressName*(course.getAddress()));  
 courses.add(course);  
 }  
 return courses;  
}

选课查询代码：

public static List<Grade> selectGrades() throws SQLException {  
 String sql = "select \* from grade";  
 Statement statement = DBTool.*getConn*().createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
 List<Grade> grades = new ArrayList<>();  
 while (resultSet.next()) {  
 Grade grade = new Grade();  
 grade.setSno(resultSet.getInt("Sno"));  
 grade.setCno(resultSet.getInt("Cno"));  
 grade.setGrade(resultSet.getInt("grade"));  
 grades.add(grade);  
 }  
 return grades;  
}  
  
public static List<Grade> selectGrades(String where) throws SQLException {  
 String sql = "select \* from grade where " + where;  
 Statement statement = DBTool.*getConn*().createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
 List<Grade> grades = new ArrayList<>();  
 while (resultSet.next()) {  
 Grade grade = new Grade();  
 grade.setSno(resultSet.getInt("Sno"));  
 grade.setCno(resultSet.getInt("Cno"));  
 grade.setGrade(resultSet.getInt("grade"));  
 grades.add(grade);  
 }  
 return grades;  
}

5.数据插入

添加学生代码：

public static int insertStudent(Student student) throws SQLException {  
 String sql = "insert into student(name,sex,age,address) values('" + student.getName() + "'," + student.getSex() + "," + student.getAge() + ",'" +student.getAddress() + "')";  
 System.*out*.println(sql);  
 Statement statement = DBTool.*getConn*().createStatement();  
 statement.executeUpdate(sql,Statement.*RETURN\_GENERATED\_KEYS*);  
 ResultSet rs = statement.getGeneratedKeys();  
 rs.next();  
 return rs.getInt(1);  
}

添加课程代码：

public static int insertCourse(Course course) throws SQLException {  
 String sql = "insert into course(name,address,credit) values('" + course.getName() + "'," + course.getAddress() + "," + course.getCredit() + ")";  
 System.*out*.println(sql);  
 Statement statement = DBTool.*getConn*().createStatement();  
 statement.executeUpdate(sql,Statement.*RETURN\_GENERATED\_KEYS*);  
 ResultSet rs = statement.getGeneratedKeys();  
 rs.next();  
 return rs.getInt(1);  
}

添加选课代码：

public static void insertGrade(Grade grade) throws SQLException {  
 String sql = "insert into grade(Sno,Cno,grade) values('" + grade.getSno() + "'," + grade.getCno() + "," + grade.getGrade() + ")";  
 System.*out*.println(sql);  
 Statement statement = DBTool.*getConn*().createStatement();  
 statement.executeUpdate(sql);  
}

6.数据更新

更新学生代码：

public static int updateStudent(Student student) throws SQLException {  
 String sql = "update student set name='" + student.getName() + "',sex=" +  
 student.getSex() + ",age=" + student.getAge() + ",address=" + student.getAddress() +  
 " where no=" + student.getNo();  
 Statement statement = DBTool.*getConn*().createStatement();  
 return statement.executeUpdate(sql);//返回所影响的行数  
}

更新课程代码：

public static int updateCourse(Course course) throws SQLException {  
 String sql = "update course set name='" + course.getName() + "',address=" +  
 course.getAddress() + ",credit=" + course.getCredit() +  
 " where no=" + course.getNo();  
 Statement statement = DBTool.*getConn*().createStatement();  
 return statement.executeUpdate(sql);//返回所影响的行数  
}

更新选课代码：

public static int updateGrade(Grade grade) throws SQLException {  
 String sql = "update grade set Sno='" + grade.getSno() + "',Cno=" +  
 grade.getCno() + ",grade=" + grade.getGrade() +  
 " where Sno=" + grade.getSno()+" and Cno=" + grade.getCno();  
 Statement statement = DBTool.*getConn*().createStatement();  
 return statement.executeUpdate(sql);//返回所影响的行数  
}

7.数据删除

删除学生代码：

public static int deleteStudent(Student student) throws SQLException {  
 String sql = "delete from student where no="+student.getNo();  
 Statement statement = DBTool.*getConn*().createStatement();  
 return statement.executeUpdate(sql);//返回所影响的行数  
}

删除课程代码：

public static int deleteCourse(Course course) throws SQLException {  
 String sql = "delete from course where no="+course.getNo();  
 Statement statement = DBTool.*getConn*().createStatement();  
 return statement.executeUpdate(sql);//返回所影响的行数  
}

删除选课代码：

public static int deleteGrade(Grade grade) throws SQLException {  
 String sql = "delete from grade where Sno="+grade.getSno()+" and Cno=" + grade.getCno();  
 Statement statement = DBTool.*getConn*().createStatement();  
 return statement.executeUpdate(sql);//返回所影响的行数  
}

8.运行展示：



