

数据库工程作业

要求:

- 1. 完成一个小型的数据库信息管理系统（或部分功能），并填写工程作业报告；程序和报告请在规定时间之内上传。
- 2. 开发模式（B/S 或 C/S）、开发高级语言任选，后台数据库使用大型数据库管理系统（SQL Server、Oracle、MySQL 等），不要使用桌面数据库。
- 3. 报告中所列举的四种操作，每种操作举一个例子即可。
- 4. 作业成绩按照报告中的标准评分，程序只实现报告中涉及的部分即可。
- 5. 作业完成后，请将工程作业报告和程序打包提交给助教老师，并联系助教老师进行系统说明和演示，回答相关问题。

工程作业报告

1. 项目信息（10 分）

学号	2213254	姓名	聂嘉欣	专业	计算机科学与技术
项目名称	龙神电子教务管理系统				
必备环境	MySql 环境、易语言可视化开发环境（运行时可不必要）				
系统主要功能简介（4 分）	<p>实现了一个功能较为齐全的电子教务管理系统：</p> <p>具体功能包括：</p> <p>1）登录，分为学生和管理员登录，在登录的时候还实现了密码可视化和隐藏。</p> <p>2）学生界面，具有多个选择夹来实现不同功能：学生信息展示，选课系统（可以查看所有能选的课程信息），退课系统（可以查看自己所有选择的课程），信息更新界面（根据需要更新邮箱和电话信息），密码更改界面。</p> <p>3）管理员界面，具有多个选择夹来实现不同功能：学生管理（可以查询所有学生信息或者单个学生信息，可以对单个学生进行开除操作，可以新增学生），教师管理（可以查询所有教师信息或者单个教师信息，可以对单个教师进行授予职位操作），学院管理（可以查询所有学院以及单个学院信息，可以对学院进行更名操作，可以新建学院），课程管理（可以查询所有以及单个课程信息，可以新增课程），成绩综合管理（可以进行成绩录入，可以对单个学生进行各种学分统计），大数据统计（实现对整个系统的各种数据进行统计，比如学生人数，教师人数等等共 6 项数据指标），特别权限管理（对学生密码进行强制更改），退出系统（实现随时退出系统并回滚到管理员界面）</p> <p>4）对人机交互的优化操作：在各种查询与输入框，我设置了很多的合法性检查以及对操作的提示信息，增强了交互时候的友好性，提高了用户使用时候的效率。</p>				

系统
主要
页面
截图
(6
分)

登录界面：（学生）



登录界面：（管理员）



学生界面：

学生教务系统

学籍信息 | 选课系统 | 退课系统 | 信息更新 | 更改密码

学籍信息展示页

学生id2002001

学生姓名刘桂兰

学生性别女

学生年龄18

所属学院外语学院

所属专业德语

邮箱obqrpqgmfn@yahoo.com

电话18953141211

显示信息

退出登录

管理员界面:

管理员教务系统

学生管理 | 教师管理 | 院系管理 | 课程管理 | 成绩综合管理 | 大数据统计 | 特别权限管理 | 退出系统

查询所有学生信息

单学生操作框

输入学生id

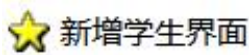
新增学生

查询

开除学籍

学生id	姓名	性别	年龄	所属院系id	所属专业id	邮
2002001	刘桂兰	女	18	1006	52012	fs
2002002	敖文	男	18	1010	52020	li
2002003	告知非	男	20	1010	52019	ps
2002004	马英	男	18	1009	52018	rr
2002005	崔志宣	男	23	1009	52017	xc
2002006	张晶	女	23	1008	52016	ut
2002007	李玉兰	男	20	1008	52015	dt
2002008	刘子奇	女	21	1007	52014	wj
2002009	张桂英	男	28	1007	52013	sk
2002010	侯阳	女	21	1006	52012	qv
2002011	赵萌萌	男	20	1006	52011	yf
2002012	朱洁	女	18	1005	52010	dr
2002013	袁丽娟	女	21	1005	52009	ka

新增学生界面:

	 <div> <div>输入</div> <div> <div>姓名 <input type="text"/></div> <div>性别 <input type="text"/></div> </div> <div> <div>年龄 <input type="text"/></div> <div>院系id <input type="text"/></div> </div> <div> <div>邮箱 <input type="text"/></div> <div>专业id <input type="text"/></div> </div> <div>电话 <input type="text"/></div> <div>添加</div> </div>
备注	一共有六个界面，分别为登录界面、管理员界面、学生界面、新增学生界面、新增学院界面、新增课程界面。

2. 系统配置（10分）

说明		(2分) 请说明系统配置情况(后台数据库, 高级语言); (8分) 请使用连接串连接高级语言和数据库, 并分析字符串的各个部分。			
配置 步骤 2分	DBMS	1. 安装并配置好 mysql 环境, mysql 用来提供数据库服务的, 版本是 Ver 5.7.26for Win64 on x86_64 (MySQL Community Server - GPL)。			
		2. 安装并破解好 navicat 数据库管理软件, 用于可视化管理提供便利条件。			
	高级 语言	1. 安装并配置好易语言环境进行数据库链接和可视化操作。			
		2. SQL 语句进行增删改查等操作。			
连接串 分析 (6分)		序号	名称	功能说明	取值
		1	连接 MySql ()	这是易语言中 MySQL 支持库中的函数, 功能是根据输入的参数进行 MySQL 数据库的连接操作, 并会返回一个 mysql 句柄来表示连接数据库是否成功。0 代表连接失败。	这个参数一共有五个参数, 在后面的表中会介绍每个参数的作用。
		2	参数 1	声明 MySQL 数据库服务器地址	127.0.0.1
		3	参数 2	用户名	root
		4	参数 3	密码	root
		5	参数 4	数据库名字	dragongod
	6	参数 5	端口	3306	

连接串代码 (截屏) (2分)	<table><tr><th>子程序名</th><th>返回值类型</th><th>公开</th><th>易包</th><th>备 注</th></tr><tr><td>_启动子程序</td><td>整数型</td><td></td><td></td><td>本子程序在程序启动后最先执行</td></tr></table> <pre>mysql句柄 = 连接MySQL ("127.0.0.1", "root", "root", "dragongod", 3306)</pre> <div><div>如果真 (mysql句柄 = 0)</div><div>信息框 ("数据库连接失败!", 0, "龙神提示您",)</div><div>返回 (0)</div><div>信息框 ("数据库连接成功", 0, "龙神欢迎您",)</div><div>执行SQL语句 (mysql句柄, "SET NAMES 'gbk' ")</div><div>载入 (登录窗口, , 真)</div></div>	子程序名	返回值类型	公开	易包	备 注	_启动子程序	整数型			本子程序在程序启动后最先执行					
子程序名	返回值类型	公开	易包	备 注												
_启动子程序	整数型			本子程序在程序启动后最先执行												
备注	<p>这里由于 mysql 句柄会在全局中不断的被调用，所以这里要将其声明成全局变量，如下：</p> <table><tr><th>全局变量名</th><th>类 型</th><th>数组</th><th>公开</th><th>备 注</th></tr><tr><td>mysql句柄</td><td></td><td></td><td></td><td></td></tr><tr><td>登录账号</td><td>文本型</td><td></td><td></td><td></td></tr></table>	全局变量名	类 型	数组	公开	备 注	mysql句柄					登录账号	文本型			
全局变量名	类 型	数组	公开	备 注												
mysql句柄																
登录账号	文本型															

3. 数据库设计 (14分)

说明	(10分) 按照数据表的创建顺序，依次给出所涉及数据表的信息，其中参照字段以“(字段1, 字段2, ……，字段n)”的形式给出，被参照字段以“表名(字段1, 字段2, ……，字段n)”的形式给出； (4分) 一般 DBMS 都可以为数据库生成关系图，请将该图片截屏并粘贴到表格中。				
数据表 (10)	创建顺序	数据表名称	主键	参照属性	被参照表及属性
	1	department	d_id	无	无
	2	major	m_id	(d_id)	department(d_id)
	3	teacher	t_id	(d_id)	department(d_id)
	4	course	c_id	(d_id, t_id)	department(d_id), teacher(t_id)
	5	student	s_id	(d_id, m_id)	department(d_id), major(m_id)
	6	student_course	s_id, c_id	(s_id, c_id)	student(s_id), course(c_id)
	7	user_admin	admin_ID	无	无
	8	user_student	s_id	(s_id)	student(s_id)

关系图 (4)	<pre> erDiagram user_student --o{ student : "has" user_admin --o{ user_admin : "has" student --o{ student_course : "has" student --o{ department : "has" student --o{ major : "has" student --o{ teacher : "has" student_course --o{ course : "has" department --o{ major : "has" department --o{ teacher : "has" major --o{ teacher : "has" user_student { int s_id PK char(52) s_pwd } user_admin { int admin_id PK varchar(15) admin_name char(52) admin_pwd } student { int s_id PK varchar(10) name char(1) sex int(0) age int(0) d_id FK int(0) m_id FK char(30) email varchar(11) tel } student_course { int s_id FK int(0) c_id FK int(0) score char(2) status } course { int(0) c_id PK varchar(15) c_name decimal(2, 1) credit int(0) d_id FK int(0) t_id FK } department { int(0) d_id PK varchar(15) d_name varchar(10) d_mng } major { int(0) m_id PK int(0) d_id FK varchar(20) m_name } teacher { int(0) t_id PK int(0) d_id FK varchar(15) t_name char(1) t_sex varchar(6) t_position } </pre>
备注	<p>这里 user_admin 表是独立的一张表用来对管理员的账号和密码进行存储。这里对课程的开课学院和老师并没有做限定性的要求，因为考虑到虽然大部分的专业课会是本学院开设，但是会有不少的 e 类课也会是其他学院开设，如果要具体实现会大幅度加大表的复杂度所以这里进行了一定程度上的简化处理。另一个需要说明的地方是不对学生的选课范围进行一定是本学院开设课程的限定，这样的原因是出于对可能会有不少同学选择 e 类课或者其他专业课进行辅修或者兴趣爱好。这里还有一个小限定就是一个课程只能被一个老师教授。</p>

4. 含有事务应用的删除操作（13 分）

说明	<p>(1 分) 简要说明该操作所要完成的功能；</p> <p>(2 分) 该操作会涉及的表（必须含有两张或两张以上的关系表，同时以“表名”的形式给出）</p> <p>(1 分) 表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”）</p> <p>(1 分) 删除条件涉及的字段描述（以“表名. 属性=? ”形式给出）</p> <p>(4 分) 实现该操作的关键代码（高级语言、SQL），截图即可；（其中如果删除语句中不包含任何形式的事务应用将扣除 3 分）</p> <p>(4 分) 如何执行该操作，按所述方法能够正常演示程序则给分。</p>
功	<p>当管理员对一个学生进行开除操作的时候，程序会自动删除掉在 student 表, student_course</p>

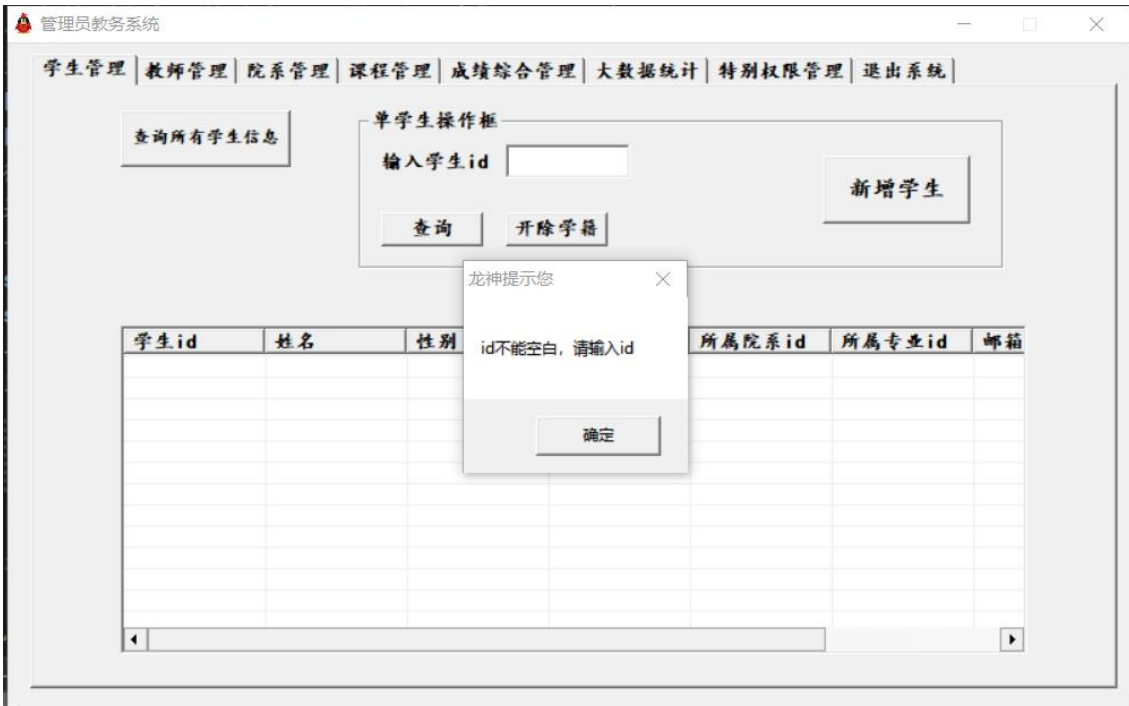
能描述 (1分)	<p>表和 user_student 表中级联相关的记录。将整个过程封装在一个 transaction 中。保证了删除时候的准确性和正确性确保不会发生删除错误。其中我会对是否完全成功删除掉所有目标记录进行检查，如果成功全部删除则会 commit，否则会 rollback。</p> <p>补充说明：这里我将 transaction 封装在了一个 PROCEDURE 中，唯一的输入参数是 studentid。</p>	
涉及的表 (2分)	student 表， student_course 表和 user_student 表	
表连接涉及字段 (1分)	<p>这个 delete_student_cascade 事务并没有在代码中声明明确的表连接字段来进行操作，而且统一通过输入的 studentid 来进行匹配，即 s_id 就是上述的三个表中本来用来连接的字段。</p> <p>代码如下：</p> <pre>DELETE FROM student_course WHERE s_id = studentid; DELETE FROM user_student WHERE s_id = studentid; DELETE FROM student WHERE s_id = studentid;</pre>	
删除条件字段描述 (1分)	字段	规则
	student.s_id=studentid	根据输入的学生 id 在 student 表中的匹配，然后将匹配成功的记录元组删除。
	user_student.s_id=studentid	根据输入的学生 id 在 user_student 表中的匹配，然后将匹配成功的记录元组删除。
	student_course.s_id=studentid	根据输入的学生 id 在 student_course 表中的匹配，然后将匹配成功的记录元组删除。
	补充： 合法性检查：sumcount=0	sumcount 是对删除操作后的表进行检查，将在上述三张表中的目标记录元组分别查询计数，并将计数加和赋给 sumcount，如果 sumcount 等于 0 则说明在三个表中已经没有要删除的目标元组，则 commit。否则则 rollback
代码 (4分)	<p>存储过程的事务定义代码：</p> <div><div>义</div><div>SQL 预览</div><pre>1 CREATE DEFINER=`root`@`localhost` PROCEDURE `delete_student_cascade`(IN studentid INT) 2 BEGIN 3 DECLARE countone INT; 4 DECLARE counttwo INT; 5 DECLARE countthree INT; 6 7 START TRANSACTION; 8 DELETE FROM student_course WHERE s_id = studentid; 9 DELETE FROM user_student WHERE s_id = studentid; 10 DELETE FROM student WHERE s_id = studentid; 11 SET countone = (SELECT COUNT(*) FROM student WHERE s_id = studentid), 12 counttwo = (SELECT COUNT(*) FROM student_course WHERE s_id = studentid), 13 countthree = (SELECT COUNT(*) FROM user_student WHERE s_id = studentid); 14 IF countone+counttwo+countthree = 0 THEN 15 COMMIT; 16 ELSE 17 ROLLBACK; 18 END IF; 19 END</pre></div>	

开除学生点击操作代码:

子程序名		返回值类型	公开	易包	备注
_单学生开除学籍_被单击					
变量名	类型	静态	数组	备注	
sql	文本型				
错误信息	文本型				
sql1	文本型				
结果集	整数型				

```
--如果真 (单学生操作学生id输入框.内容 = "")
--信息框 ("id不能空白, 请输入id", 0, "龙神提示您", )
--返回 ()
sql1 = "select * from student where s_id='%s'"
sql1 = 格式化文本 (sql1, 单学生操作学生id输入框.内容)
执行SQL语句 (mysql句柄, sql1)
结果集 = 取记录集 (mysql句柄)
--如果真 (取记录集行数 (结果集) ≠ 0)
--sql = "call delete_student_cascade('%s')"
--sql = 格式化文本 (sql, 单学生操作学生id输入框.内容)
--执行SQL语句 (mysql句柄, sql)
--取错误文本 (mysql句柄, 错误信息)
--如果 (错误信息 = "")
--信息框 ("成功开除该学生, 已经级联删除其相关信息", 0, "龙神恭喜您", )
--信息框 ("开除失败", 0, "龙神警告", )
--返回 ()
--信息框 ("输入的id不正确, 请重新输入", 0, "龙神提示您", )
```

首先登录进管理员界面, 然后找到在学生管理中的开除学生按钮, 如果不输入 id, 直接点击开除学生, 如下:



程序演示 (4分)

然后输入 id，如果 id 是非法的（即不在 student 中），如下：



如果输入了一个正确的学生 id: ，成功执行，如下：



补充：我这里删除的是 id 为 2002029 的学生，我们可以取 navicat 中查找一下，看是否该学生的信息还在系统中，发现所有表中都是到 id 为 2002028 就截止了，再在界面中查询一下如下：（说明目标删除学生的所有信息已经被成功的删除干净了。）

	
备注	这里的删除事务，我们也可以采用外键定义时候的 cascade 关键字定义，在删除的时候会自动级联删除，这样也不失为一种优秀的方法。

5. 触发器控制下的添加操作（20 分）

说明	（1 分）简要说明该操作所要完成的功能； （2 分）简要说明该触发器所要完成的功能 （1 分）该操作会涉及的表（以“表名”的形式给出）。 （2 分）该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空； （6 分）实现该操作的关键代码（高级语言、SQL），截图即可； （8 分）如何执行该操作，按所述方法能够正常演示程序则给分。	
功能描述 （1 分）	在新增学生操作的时候，会对输入的学生的年龄和电话进行检查，如果不符合合法性输入的规则的则会报错非法输入，如果在正常范围区间内，则会正确的插入数据。	
触发器描述 （2 分）	先进行对触发器 ageandtel_input_check 进行定义，然后用 if 语句进行判定，筛选输入学生年龄小于 10 岁或者大于 40 岁或者电话不是 1 开头长度为 11 位的输入，然后如果上面这三个条件规则出现一个就会进行报错信息输出，而且插入不会执行；如果一切正常，则会正常插入新学生信息数据。这样可以防止插入数据的时候失误输错数据，提高了程序的鲁棒性。	
涉及的表 （1 分）	student 表	
输入数	字段	规则
	age（年龄）	如果年龄小于 10 岁或者大于 40 岁会被判定为非法，否则为合法

据 (2 分)	tel (电话)	如果电话不是1 开头长度为 11 位的电话号会被判定成非法,否则合法。																																																
插入 操作 源码 (3 分)	<div><table><tr><td>窗口程序集名</td><td>保留</td><td>保留</td><td>备注</td></tr><tr><td>窗口程序集_新增学生窗口</td><td></td><td></td><td></td></tr></table><table><tr><td>子程序名</td><td>返回值类型</td><td>公开</td><td>易包</td><td>备注</td></tr><tr><td>_按钮1_被单击</td><td></td><td></td><td></td><td></td></tr></table><table><tr><td>变量名</td><td>类型</td><td>静态</td><td>数组</td><td>备注</td></tr><tr><td>sql</td><td>文本型</td><td></td><td></td><td></td></tr><tr><td>结果集</td><td>整数型</td><td></td><td></td><td></td></tr><tr><td>sqlextra</td><td>文本型</td><td></td><td></td><td></td></tr><tr><td>临时文本</td><td>文本型</td><td></td><td></td><td></td></tr><tr><td>错误信息</td><td>文本型</td><td></td><td></td><td></td></tr></table><pre>如果真 (姓名输入.内容 = "") 信息框 ("姓名不能空白, 请重新编辑", 0, ,) 返回 () 如果真 (性别输入.内容 = "") 信息框 ("性别不能空白, 请重新编辑", 0, ,) 返回 () 如果真 (年龄输入.内容 = "") 信息框 ("年龄不能空白, 请重新编辑", 0, ,) 返回 () 如果真 (院系id输入.内容 = "") 信息框 ("院系id不能空白, 请重新编辑", 0, ,) 返回 () 如果真 (邮箱输入.内容 = "") 信息框 ("邮箱不能空白, 请重新编辑", 0, ,) 返回 () 如果真 (专业id输入.内容 = "") 信息框 ("专业id不能空白, 请重新编辑", 0, ,) 返回 () 如果真 (电话输入.内容 = "") 信息框 ("电话不能空白, 请重新编辑", 0, ,) 返回 () sqlextra = "select * from major where m_id='%s' and d_id='%s'" sqlextra = 格式化文本 (sqlextra, 专业id输入.内容, 院系id输入.内容) 执行SQL语句 (mysql句柄, sqlextra) 结果集 = 取记录集 (mysql句柄) 如果真 (取记录集行数 (结果集) ≠ 0) sql = "insert into student(name,sex,age,d_id,m_id,email,tel) values('%s','%s','%s','%s','%s','%s','%s')" sql = 格式化文本 (sql, 姓名输入.内容, 性别输入.内容, 年龄输入.内容, 院系id输入.内容, 专业id输入.内容, 邮箱输入.内容, 电话输入.内容) 执行SQL语句 (mysql句柄, sql) 取错误文本 (mysql句柄, 错误信息) 如果 (错误信息 = "Error: illegal input,please exit again.") 信息框 ("输入年龄或电话非法, 插入失败", 0, "龙神提示您",) 信息框 ("新增成功", 0, "龙神恭喜你",) 返回 () 返回 () 信息框 ("没有找到匹配的学院与专业, 请重新输入", 0, "龙神提示您",) 返回 ()</pre></div>		窗口程序集名	保留	保留	备注	窗口程序集_新增学生窗口				子程序名	返回值类型	公开	易包	备注	_按钮1_被单击					变量名	类型	静态	数组	备注	sql	文本型				结果集	整数型				sqlextra	文本型				临时文本	文本型				错误信息	文本型			
窗口程序集名	保留	保留	备注																																															
窗口程序集_新增学生窗口																																																		
子程序名	返回值类型	公开	易包	备注																																														
_按钮1_被单击																																																		
变量名	类型	静态	数组	备注																																														
sql	文本型																																																	
结果集	整数型																																																	
sqlextra	文本型																																																	
临时文本	文本型																																																	
错误信息	文本型																																																	
触发 器源 码 (3 分)	定义	<pre>1 BEGIN 2 IF NEW.age<10 OR NEW.age>40 OR NEW.tel not like '1_____' THEN 3 SIGNAL SQLSTATE '45000' 4 SET MESSAGE_TEXT = 'Error: illegal input,please exit again.'; 5 END IF; 6 END</pre>																																																
程序 演示 (4 分)	插入的数据展示:																																																	

★ 新增学生界面

输入

姓名刘一手

性别男

年龄18

院系id1001

邮箱s@163.com

专业id52001

电话322623404

添加

结果展示:

管理员教务系统

学生管理 | 教师管理 | 院系管理 | 课程管理 | 成绩综合管理 | 大数据统计 | 特别权限管理 | 退出系统

查询所有学生信息

单学生操作框

★ 新增学生界面

输入

姓名刘一手

性别男

年龄18

院系id1001

邮箱s@163.com

专业id52001

电话322623404

添加

龙神恭喜你

新增成功

确定

新增学生

学生id	姓	性别	年龄	院系id	专业id	邮箱
2002017	龙					br
2002018	王					ul
2002019	刘					ic
2002020	李					r2
2002021	苏					ic
2002022	刘					2001
2002023	刘					2002
2002024	朱					2003
2002025	刘					2004
2002026	张					2005
2002027	李佳蓓	女	18	1003	52006	ec
2002028	李佳慧	女	24	1004	52007	in

程序
演示
(4
分)

插入的数据展示：（年龄非法）

★ 新增学生界面

输入

姓名刘一手性别男

年龄1院系id1001

邮箱163.com专业id52001

电话322623404

添加

结果展示：

管理员教务系统

学生管理 | 教师管理 | 院系管理 | 课程管理 | 成绩综合管理 | 大数据统计 | 特别权限管理 | 退出系统

查询所有学生信息 | 单学生操作框

★ 新增学生界面

输入

姓名刘一手性别男

年龄1院系id1001

邮箱163.com专业id52001

电话322623404

添加

新增学生

龙神提示您

输入年龄或电话非法，插入失败

确定

学生id	姓名	性别	年龄	院系id	专业id
2002017	龙				
2002018	王				
2002019	刘				
2002020	李				
2002021	苏				
2002022	刘				
2002023	孙				
2002024	朱				
2002025	孙				
2002026	张				
2002027	李佳蓓	女	18	1003	52006
2002028	李佳慧	女	24	1004	52007

插入的数据展示：（电话非法）

★

新增学生界面

— □ ×

输入

姓名

刘一手

性别

男

年龄

11

院系id

1001

邮箱

c@163.com

专业id

52001

电话

123

添加

结果展示:

管理员教务系统

— □ ×

学生管理

教师管理

院系管理

课程管理

成绩综合管理

大数据统计

特别权限管理

退出系统

查询所有学生信息

单学生操作框

★ 新增学生界面

— □ ×

新增学生

龙神提示您

输入年龄或电话非法, 插入失败

确定

输入

姓名

刘一手

性别

男

年龄

11

院系id

1001

邮箱

c@163.com

专业id

52001

电话

123

添加

学生id	姓	性	年	院	专
2002017	龙				
2002018	王				
2002019	刘				
2002020	李				
2002021	苏				
2002022	刘				
2002023	羽				
2002024	朱				
2002025	羽				
2002026	张				
2002027	李佳蓓	女	18	1003	52006
2002028	李佳慧	女	24	1004	52007

备注

可以注意到在我插入操作的源码中, 还有对学院 id 和专业 id 的同步审核, 这里的目的是为了学生可以正确的加入到存在的专业中, 专业属于学院, 这里如果没有匹配到存在的学院和专业也同样会报错, 导致插入失败。但是这一点我在操作源码中进行了体现没有加入到触发器中。而且这个地方同样有空白检查。

6. 存储过程控制下的更新操作（18 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明该存储过程所要完成的功能；</p> <p>（2 分）说明该操作涉及操作的表（必须包含两张或两张以上的关系表，以“表名形式”描述）</p> <p>（1 分）表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”）</p> <p>（2 分）该操作会修改字段（以“表名. 字段名”的形式给出），以及修改规则，如新数值的计算方法、在何种条件下予以修改等；</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（5 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>	
功能描述 （1 分）	对教师进行职位授予操作，但是要对教师是否有资格被授予新职称进行检查，这里需要对教师教授的所有课程的通过率进行检查，通过率高于 0.4 才能有资格被授予新职称。	
存储过程功能描述 （1 分）	这里更新的字段是对 teacher 表中的 t_position 属性进行更新操作，根据输入的教师 id 匹配到目标教师，然后对其进行上述的资格判定，如果有资格则执行职称授予操作，否则则会报错该教师无资格被授予新职称。	
涉及的关系表 （2 分）	teacher 表，course 表和 student_course 表	
表连接涉及字段 （1 分）	<p>course.c_id=student_course.c_id</p> <p>还有一个没有在代码中声明明确的表连接字段的连接，通过输入的 teacherID 来进行匹配，其实等于将 teacher 表中的 t_id 进行连接，代码如下：</p> <p>course.t_id=teacherID</p> <p>补充：这个存储过程也被我封装进一个 PROCEDURE，输入的参数之一是 teacherID。</p>	
更改字段 （2 分）	字段	规则
	t_position	更新前必须保证该教师教授的所有课程的通过率大于等于 40%，否则会更新失败。
更新代码 （3 分）	<pre>UPDATE teacher SET t_position=teacherposition WHERE t_id =teacherID;</pre>	

创建
存储
过程
源码
(3
分)

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `updateposition`(  
    IN teacherID INT,  
    IN teacherposition VARCHAR(6)  
)  
BEGIN  
  
    DECLARE countpass INT;  
    DECLARE countfail INT;  
  
    SELECT count(*) INTO countfail  
    FROM student_course  
    JOIN course ON course.c_id=student_course.c_id  
    WHERE course.t_id=teacherID and student_course.status='挂科';  
  
    SELECT count(*) INTO countpass  
    FROM student_course  
    JOIN course ON course.c_id=student_course.c_id  
    WHERE course.t_id=teacherID and student_course.status='修完';  
  
    IF countpass < countfail THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Error: the teacher has no right to get new position.';  
    ELSE  
        UPDATE teacher  
        SET t_position=teacherposition  
        WHERE t_id =teacherID;  
    END IF;  
END
```

存储
过程
执行
源码
(1
分)

子程序名	返回值类型	公开	易包	备注
单教师授予职位_被单击				

变量名	类型	静态	数组	备注
sql	文本型			
结果集	整数型			
sql1	文本型			
结果集1	整数型			
错误信息	文本型			
sql2	文本型			

```
如果真 (单教师操作教师id输入框.内容 = "")  
    信息框 ("id不能空白, 请输入id", 0, "龙神提示您", )  
    返回 ()  
如果真 (单教师操作职位输入框.内容 = "")  
    信息框 ("职位不能空白, 请输入职位", 0, "龙神提示您", )  
    返回 ()  
sql1 = "select * from teacher where t_id='%s'"  
sql1 = 格式化文本 (sql1, 单教师操作教师id输入框.内容)  
执行SQL语句 (mysql句柄, sql1)  
结果集1 = 取记录集 (mysql句柄)  
如果真 (取记录集行数 (结果集1) ≠ 0)  
    sql2 = "call updateposition('%s','%s')"  
    sql2 = 格式化文本 (sql2, 单教师操作教师id输入框.内容, 单教师操作职位输入框.内容)  
    执行SQL语句 (mysql句柄, sql2)  
    取错误文本 (mysql句柄, 错误信息)  
    如果 (错误信息 = "Error: the teacher has no right to get new position.")  
        信息框 ("该教师没有资格被授予职位, 请检查后重新输入", 0, "龙神警告您", )  
        信息框 ("授予职位成功!", 0, "龙神恭喜您", )  
        返回 ()  
    返回 ()  
信息框 ("没有匹配的教师, 请检查教师id后重新输入", 0, "龙神警告您", )  
返回 ()
```

程序
演示
(2
分)

管理员教务系统

学生管理 教师管理 院系管理 课程管理 成绩综合管理 大数据统计 特别权限管理 退出系统

查询所有教师信息

单教师操作框

输入教师id 521011 输入职位 高级教授

查询 授予职位

教师id 所属院系id 性别 职位

521001 1001 男 讲师

521002 1001 女 讲师

521003 1001 男 讲师

521004 1001 男 讲师

521005 1002 女 副教授

521006 1002 男 讲师

521007 1002 女 讲师

521008 1002 男 讲师

521009 1003 男 讲师

521010 1003 女 讲师

521011 1003 男 副教授

521012 1003 男 讲师

521013 1004 女 讲师

521014 1004 男 讲师

陈美华

方一强

王俊民

吴美玉

周逸珮

林纹书

黄阳

黎宏儒

龙神恭喜您

授予职位成功!

确定

简单查询验证一下，如下：

管理员教务系统

学生管理 教师管理 院系管理 课程管理 成绩综合管理 大数据统计 特别权限管理 退出系统

查询所有教师信息

单教师操作框

输入教师id 521011 输入职位 高级教授

查询 授予职位

教师id 所属院系id 姓名 性别 职位

521011 1003 周逸珮 男 高级教授

程序
演示
(2
分)

 管理员教务系统

学生管理

教师管理

院系管理

课程管理

成绩综合管理

大数据统计

特别权限管理

退出系统

查询所有教师信息

单教师操作框

输入教师id

521001

输入职位

高级教授

查询

授予职位

龙神警告您

该教师没有资格被授予职位，请检查后重新输入

确定

教师id	所属院系	姓名	性别	职位
521001	1001			讲师
521002	1001			讲师
521003	1001			讲师
521004	1001			讲师
521005	1002			副教授
521006	1002			讲师
521007	1002	陈天平	女	讲师
521008	1002	方一强	男	讲师
521009	1003	王俊民	男	讲师
521010	1003	吴美玉	女	讲师
521011	1003	周逸珮	男	副教授
521012	1003	林纹书	男	讲师
521013	1004	黄阳	女	讲师
521014	1004	黎宏儒	男	讲师

备注 无

7. 含有视图的查询操作（15 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明建立的该视图的功能；</p> <p>（2 分）简要说明该操作涉及的关系数据表（以“表名”的形式给出）</p> <p>（1 分）简要说明表连接涉及的字段（以“表 1. 属性=表 2. 属性”）</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>
操作功能描述（1 分）	<p>创建出一个视图，为了在学生选课和退课的界面通过在一张表上查询就可以给出课程的 id，名字，学分，授课教师和所属学院。这样我们就节省了复杂的连接查询等复杂查询操作的时间和空间。</p>
视图功能描述（1 分）	<p>上述的几个属性分别在三个表中（course 表，department 表和 teacher 表），这里需要重建视图去将他们通过正确的匹配合并到一张表上去，从而实现快速查询，无需在多张表上进行复杂的连接查询等操作。</p>
涉及的关系表（2 分）	<p>course 表，department 表和 teacher 表</p>
表连接字段（1 分）	<p>course.t_id=teacher.t_id</p> <p>course.d_id=department.d_id</p>
创建视图代码	<pre>CREATE VIEW student_choose_course AS SELECT c.c_id,c.c_name,c.credit,t.t_name,d.d_name FROM course c JOIN teacher t ON c.t_id=t.t_id JOIN department d ON c.d_id=d.d_id;</pre>

码
(3
分)

查
询
代
码
(3
分)

```
变量名 类 型 静态 数组 备 注
sql 文本型
sql = "select * from student_choose_course where c_id not in (select c_id from student_course where s_id='%s') or
der by c_id desc"
sql = 格式化文本 (sql, 登录账号)
更新超级列表框 (超级列表框选课, sql)
返回 ()
```

```
子程序名 返回值类型 公开 易包 备 注
_查询所有可退的课程信息_被单击
变量名 类 型 静态 数组 备 注
sql 文本型
sql = "select * from student_choose_course where c_id in (select c_id from student_course where s_id='%s' and sta
tus='在修') order by c_id desc"
sql = 格式化文本 (sql, 登录账号)
更新超级列表框 (超级列表框退课, sql)
返回 ()
```

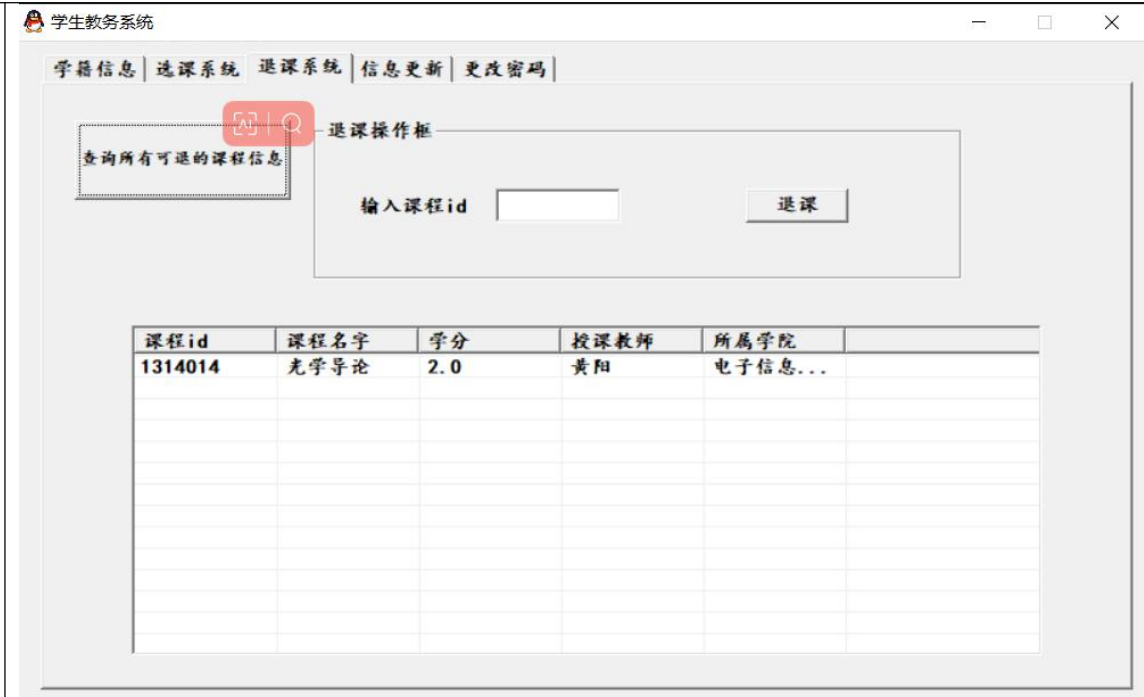
补充：上面两张查询代码截图分别是在查询可选课程和查询可退课程，其中 sql 语句中的 student_choose_course 就是我上述建立的视图名称，这两个功能都利用了新视图来进行快速查询。

程
序
演
示
(4
分)

学生界面查询所有可选课程（按 id 排序）：



学生界面查询所有可退课程（按 id 排序）：

	
备注	<p>上述的程序演示下面的超级列表框的数据查询逻辑就是通过 select 语句在我建立的新视图上进行单一查询，视图加快了查询速度，节省了的时间和设计逻辑时的复杂程度。</p>

8. 额外功能操作

操作功能描述（1分）	<p>在管理员界面中我实现了一个成绩录入功能，输入学生 id 和课程 id 和要输入的分，程序会对学生与课程自动匹配，然后如果匹配到了正确的学生选课，会对成绩合法性进行检查，如果也合格，就对学生选课表进行更新，这里我也利用了存储过程进行更新，录入成绩的同时对选课状态进行更新，小于 60 的成绩自动状态更新为挂科；大于等于 60 的状态自动更新成修完。</p>
涉及的关系表（2分）	<p>student_course 表</p>

存储过程定义代码

```
1 CREATE DEFINER=`root`@`localhost` PROCEDURE `updatescore`(  
2     IN studentID INT,  
3     IN courseID INT,  
4     IN courseGrade INT  
5 )  
6 BEGIN  
7     DECLARE courseStatus VARCHAR(10);  
8     -- 检查课程成绩是否合法  
9     IF courseGrade < 0 OR courseGrade > 100 THEN  
10         SIGNAL SQLSTATE '45000'  
11         SET MESSAGE_TEXT = 'Error: Score must be between 0 and 100.';  
12     ELSE  
13         -- 合法成绩, 根据及格标准更新课程状态  
14  
15         IF courseGrade < 60 THEN  
16             SET courseStatus = '挂科';  
17         ELSE  
18             SET courseStatus = '修完';  
19         END IF;  
20  
21     -- 更新学生选课表的课程状态和成绩  
22     UPDATE student_course  
23     SET status = courseStatus, score = courseGrade  
24     WHERE s_id = studentID AND c_id = courseID;  
25 END IF;  
26 END
```

操作源代码
(3分)

子程序名	返回值类型	公开	易包	备注
_录入按钮_被单击				

变量名	类型	静态	数组	备注
sql	文本型			
临时文本	文本型			
错误信息	文本型			
sql1	文本型			
结果集	整数型			

```
-- 如果真 (录入成绩学生id.内容 = "")  
信息框 ("id不能空白, 请输入id", 0, "龙神提示您", )  
返回 ()  
-- 如果真 (录入成绩学生课程id.内容 = "")  
信息框 ("课程id不能空白, 请输入课程id", 0, "龙神提示您", )  
返回 ()  
-- 如果真 (录入成绩.内容 = "")  
信息框 ("成绩不能空白, 请输入成绩", 0, "龙神提示您", )  
返回 ()  
sql1 = "select * from student_course where s_id='%s' and c_id='%s' and status='在修'"  
sql1 = 格式化文本 (sql1, 录入成绩学生id.内容, 录入成绩学生课程id.内容)  
执行SQL语句 (mysql句柄, sql1)  
结果集 = 取记录集 (mysql句柄)  
-- 如果真 (取记录集行数 (结果集) ≠ 0)  
sql = "call updatescore('%s', '%s', '%s')"  
sql = 格式化文本 (sql, 录入成绩学生id.内容, 录入成绩学生课程id.内容, 录入成绩.内容)  
执行SQL语句 (mysql句柄, sql)  
取错误文本 (mysql句柄, 错误信息)  
-- 如果 (错误信息 = "Error: Score must be between 0 and 100.")  
信息框 ("输入成绩非法, 录入成绩失败", 0, "龙神提示您", )  
信息框 ("成绩录入成功", 0, "龙神恭喜您", )  
返回 ()  
返回 ()  
信息框 ("没有匹配到可以录入成绩的学生选课, 请检查后重新输入", 0, "龙神提示您", )  
返回 ()
```

程序演示
(4分)

如果输入框有空白:



如果没有匹配的学生选课:



如果有匹配的学生选课, 但是成绩非法:

有匹配的学生选课而且学生成绩合法:

管理员教务系统

— □ ×

学生管理 | 教师管理 | 院系管理 | 课程管理 | 成绩综合管理 | 大数据统计 | 特别权限管理 | 退出系统

欢迎使用成绩录入系统

学生id 2002001

学生课程id 1314014

录入成绩 70

录入(自动检测)

学生学分情况统计

学生id

分: 标签

分: 标签

分: 标签

统计学分

龙神恭喜您

×

成绩录入成功

确定

查看 navicat 中记录，发现成功更新了成绩和学生的选课状态。

备注	无
----	---

无