信息工程学院

《数据库课程设计》论文

题 目：中学生档案管理系统数据库设计

|  |  |
| --- | --- |
| 学 号： |  |
| 专业班级： |  |
| 姓 名： |  |
| 指导老师： |  |
| 完成日期： |  |

中学生档案管理系统数据库设计

王泽玺

（信息工程学院软件工程2020级1班）

摘 要： 中学生档案管理系统是为了解决中学档案管理中，面临的复杂流程所开发的系统。它主要包含档案管理、成绩管理和入学管理三个模块。数据库设计中主要进行了分析中学生档案管理的需求，进行概念结构、逻辑结构和物理结构的设计，并通过建立视图、存储过程和触发器实现了满足不同用户的需求。如对学生、班主任和任课老师对成绩查询的结果各不相同。通过Oracle19c和PL/SQL 14进行数据库实施。实现了较为自动化、人性化的中学生档案管理系统。

关键字：数据库设计；中学生档案；Oracle 19c；

**目 录**

[1需求分析 1](#_Toc80864202)

[1.1数据流程图 2](#_Toc80864203)

[1.2数据字典 4](#_Toc80864204)

[2 概念结构设计 4](#_Toc80864205)

[2.1 分E-R图建立 5](#_Toc80864206)

[2.2 全局/整体E-R图 7](#_Toc80864207)

[3 逻辑结构设计 8](#_Toc80864208)

[3.1 建立关系模式 9](#_Toc80864209)

[3.2 关系模式规范化处理 9](#_Toc80864210)

[3.3 用户子模式设计 11](#_Toc80864211)

[3.4 关系模式逻辑结构定义 11](#_Toc80864212)

[4．数据库物理设计 14](#_Toc80864213)

[5．数据库实施与测试 14](#_Toc80864214)

[5.1 数据库及数据库对象建立 14](#_Toc80864215)

[5.1.1创建基本表 14](#_Toc80864216)

[5.1.2创建视图 18](#_Toc80864217)

[5.1.3创建索引 22](#_Toc80864218)

[5.1.4创建存储过程 23](#_Toc80864219)

[5.1.5创建触发器 37](#_Toc80864220)

[5.2 数据入库 39](#_Toc80864221)

[5.3 数据库测试 48](#_Toc80864222)

[6．总结 54](#_Toc80864223)

[7．附录 55](#_Toc80864224)

在如今，信息化已经相当的普及。尤其是数据库其在数据存储管理方案的优势是人工所无法比拟的，因此作为有许多信息需要存档和管理的中学来说，拥有一个功能齐全、使用简单的中学生档案管理系统，可以大大提升中学生档案管理的效率。

# 1需求分析

本系统主要任务是实现中学生档案管理系统。系统分为以下三个功能模块：档案管理、成绩管理和入学管理。具体功能可以从以下三个方面介绍：

（1）档案管理：

1. 档案管理员对学生档案进行登记和修改
2. 档案管理员对班级档案进行登记和修改
3. 当登记或修改班级档案时，对学生的班级信息进行修改
4. 学生对自己的档案进行查询
5. 成绩管理：
6. 老师对成绩进行登记
7. 老师查询成绩时，生成仅有该老师所带科目的成绩表
8. 班主任查询成绩时，生成全班的成绩表
9. 学生查询成绩时，生成仅有自己成绩的成绩表
10. 入学管理：
11. 学生提交报告单
12. 根据学生报告单将学生分为，退学、正常三种

c.统计退学学生名单，修改学生档案，并将其存入退学学生名单中

## 1.1数据流程图

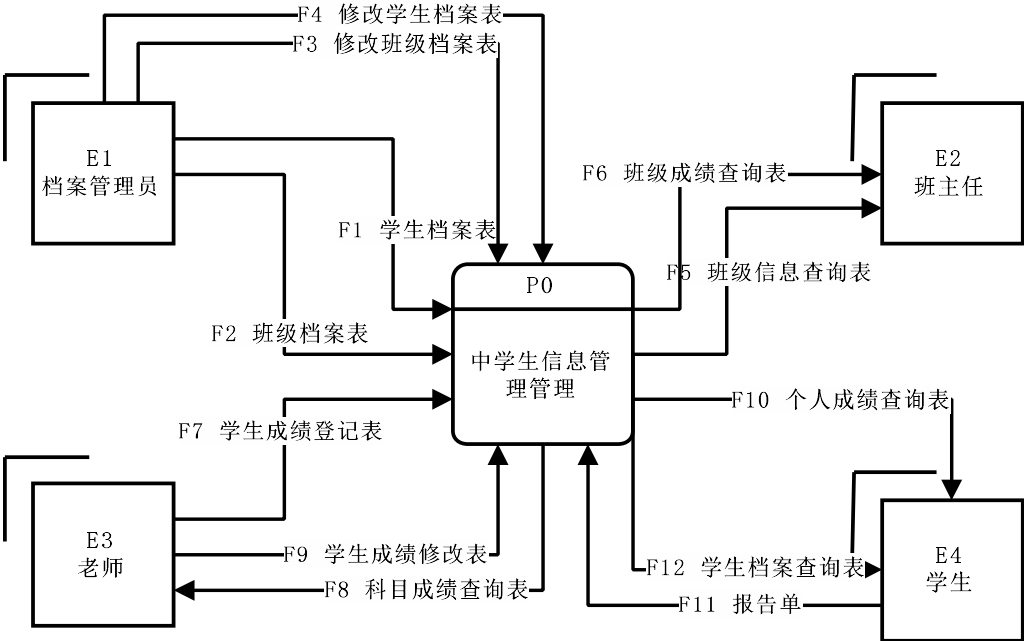


图 1-1 顶层数据流

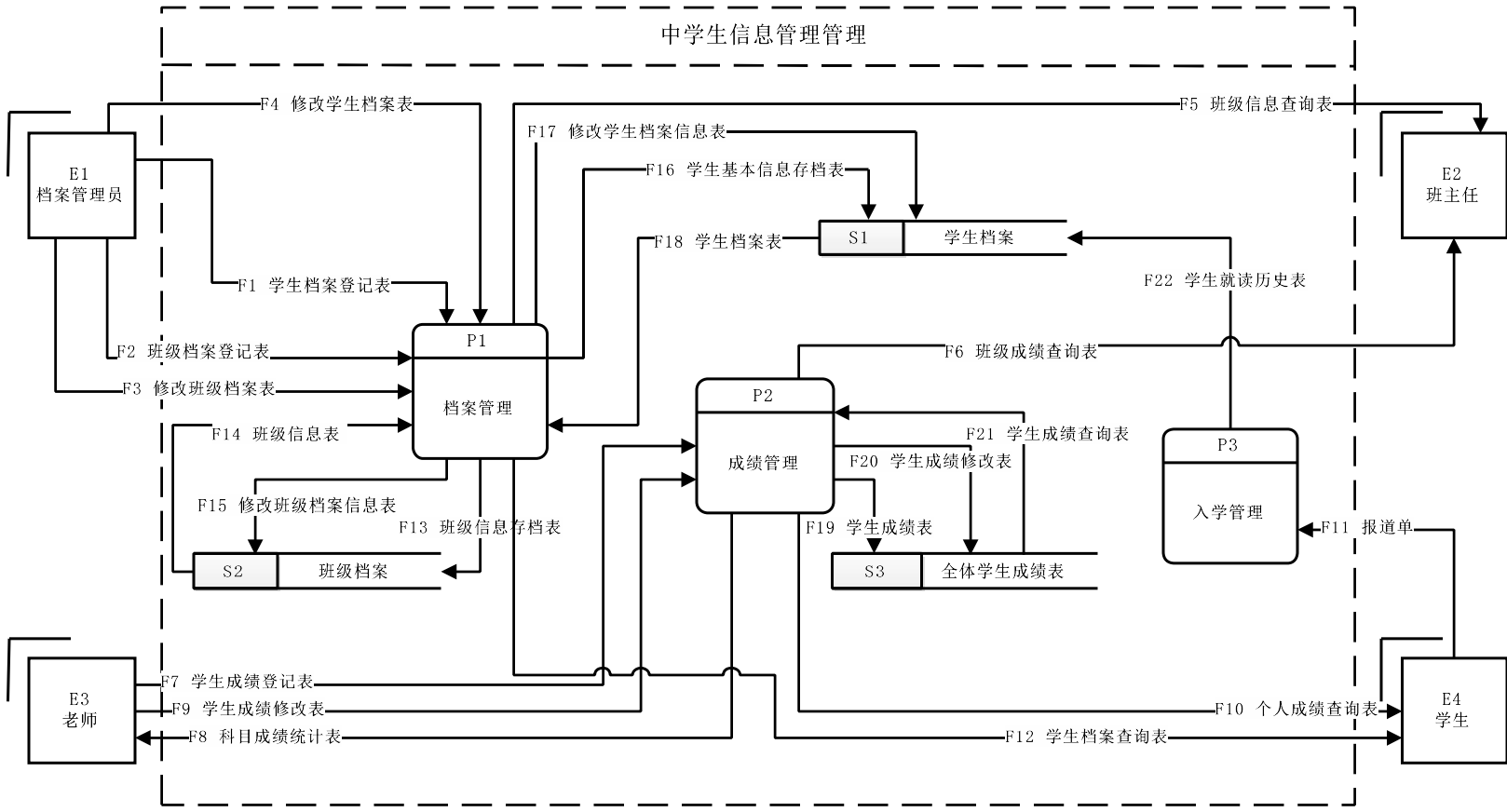


图 1-2 第一层数据流程图

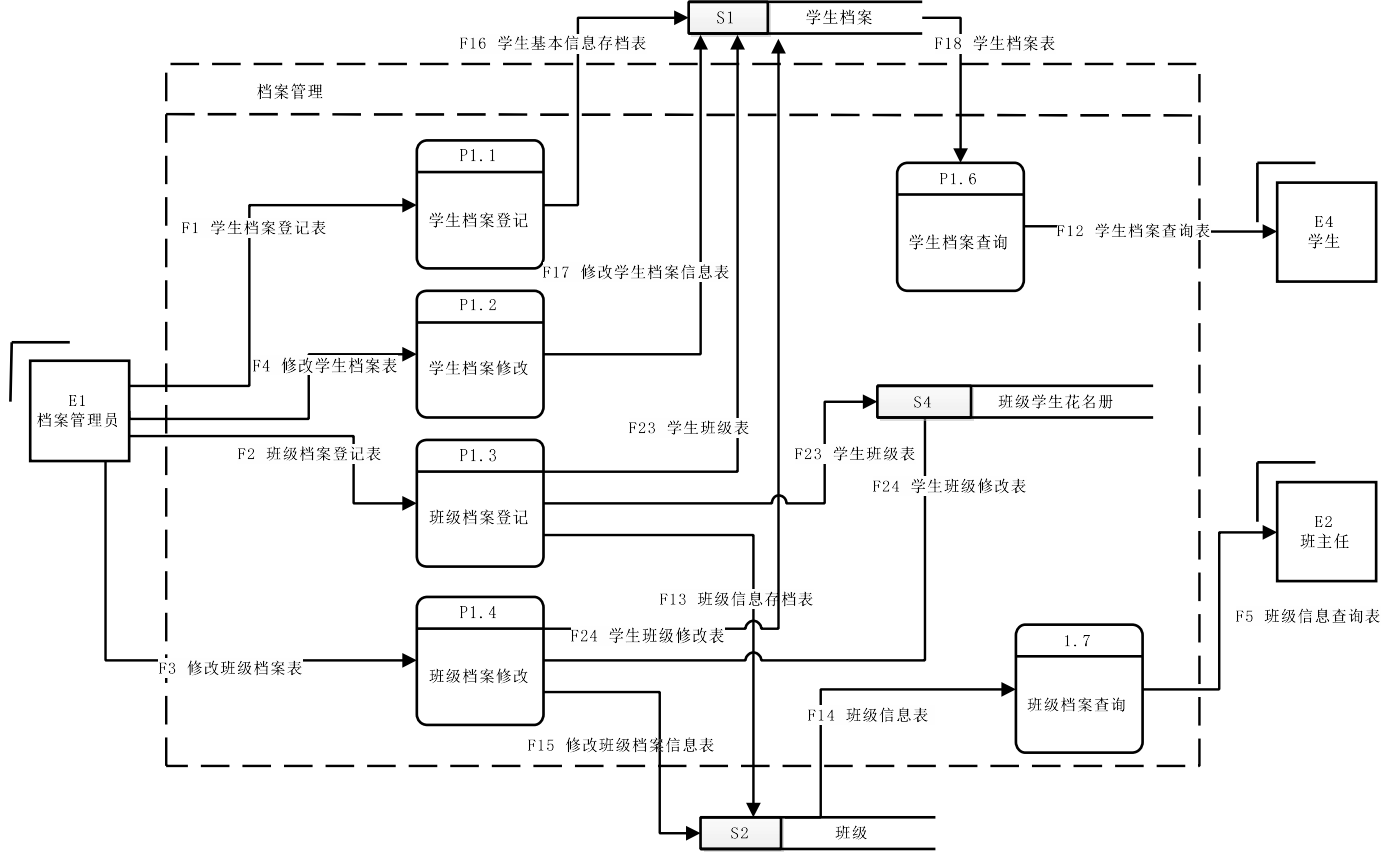


图 1-3 第二层数据流程图——档案管理

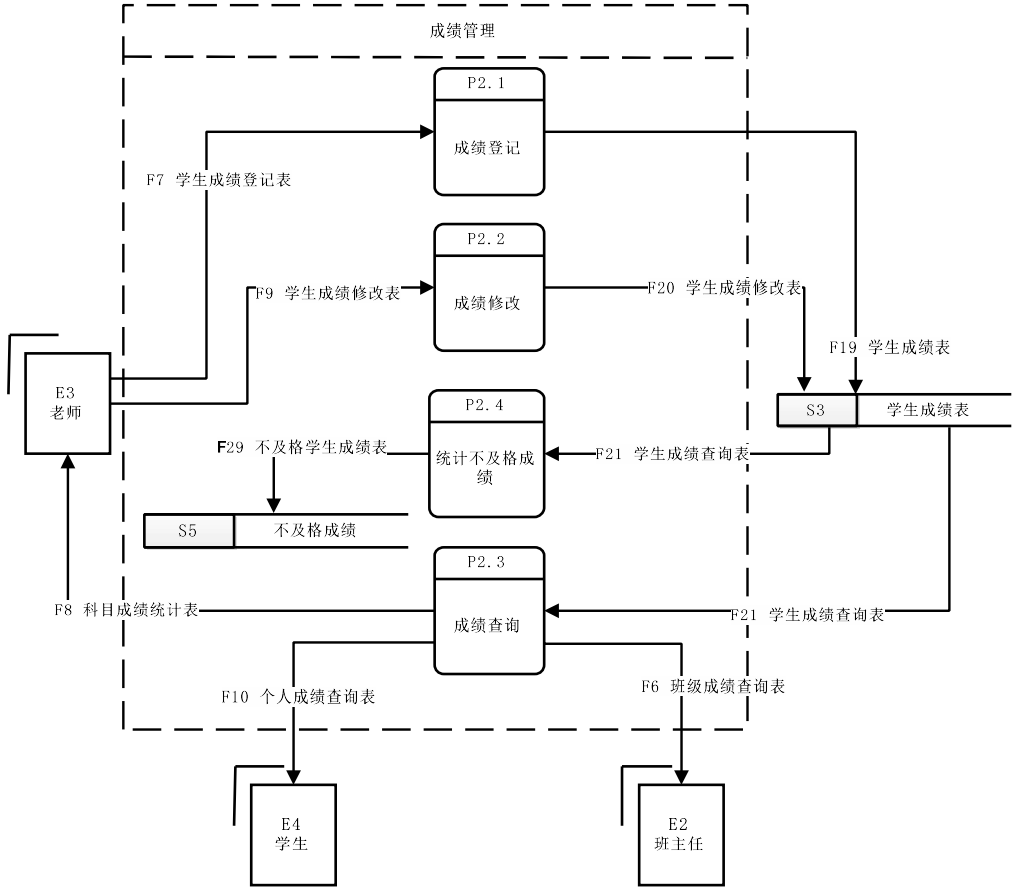


图 1-4 第二层数据流程图——成绩管理

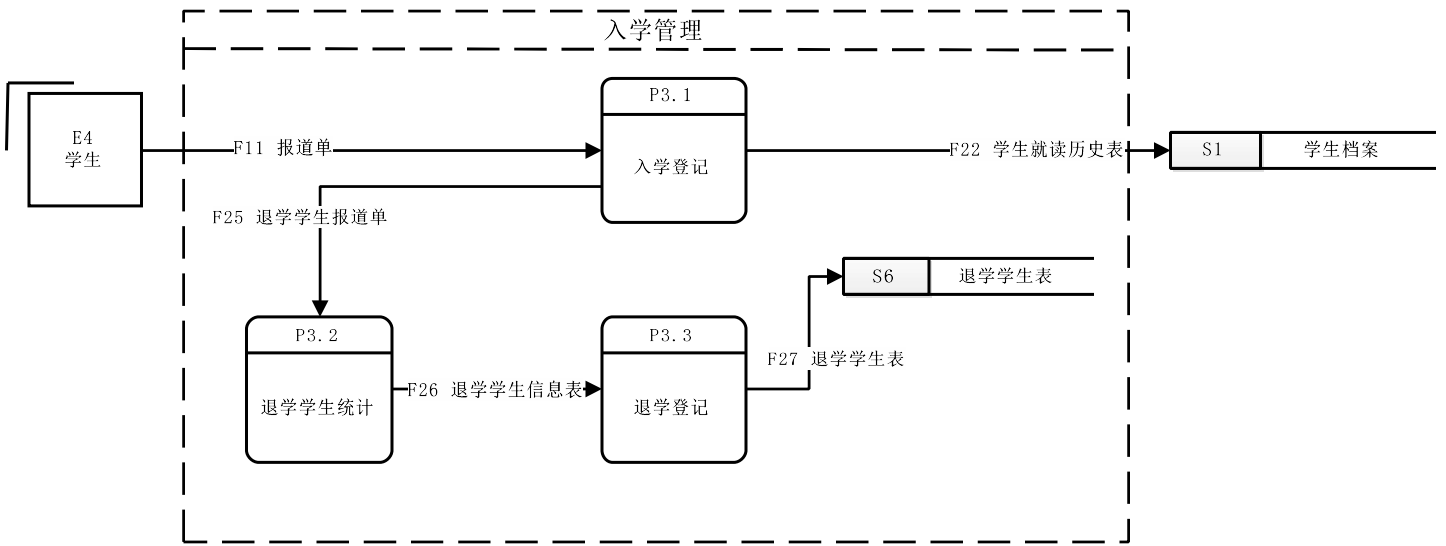


图 1-5 第二层数据流程图——入学管理

## 1.2数据字典

通过分析数据流程图及模块功能关系，确定出56个数据项。（见附录一）

根据数据流程图，确定13个数据结构。（见附录二）

根据流程图分析并删除冗余与错误，共包含29个数据流。（见附录三）

数据流程图中有13个处理逻辑。（见附录四）

分析数据流程图，确定6个数据存储。（见附录五）

# 2 概念结构设计

概念结构设计目标：通过对用户需求进行综合，归纳与抽象，形成一个独立于具体数据库管理系统的概念模型。

概念结构设计任务和方法：根据需求分析得到的数据流程图与数据字典，采用自底向上的方法，先抽象成分E-R图再消除冲突、冗余合成全局E-R图，通过数据项、数据结构等确定实体的属性及联系。

概念设计内容：确定实体与实体之间的联系，形成一个整体的超市管理系统的逻辑模型，然后让我们在数据项、数据结构、数据流、处理逻辑、存储结构这些数据字典的基础上，结合实体间的联系，抽象出关系模型。

## 2.1 分E-R图建立

在建立E-R图的时候，常采用的策略是自顶向下得进行需求分析，设计出各子系统的分E-R图。

1. 档案管理E-R图

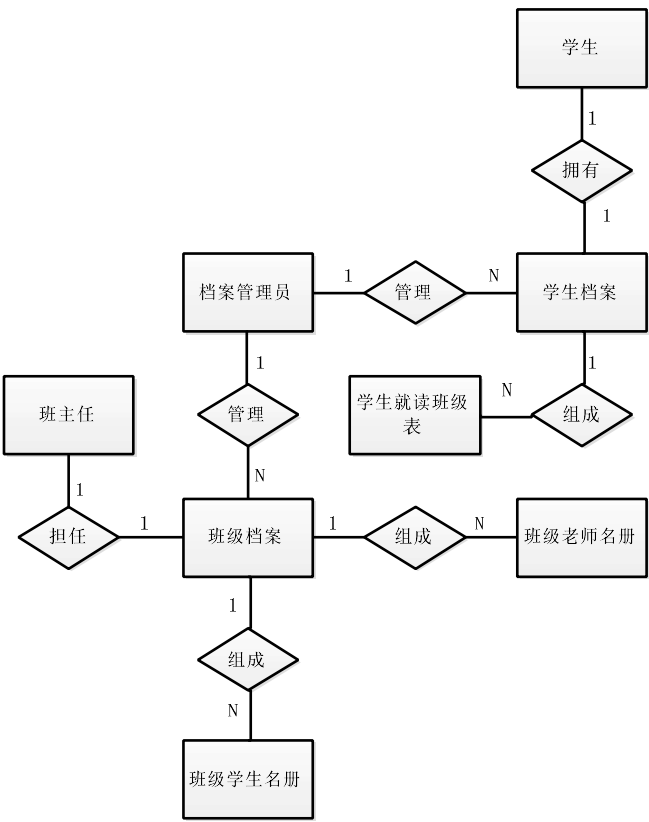


图 2-1 分E-R图—档案管理

属性：

学生（学生ID，姓名）

学生档案（学生ID，姓名，身份证号，姓名，年龄，出生日期，照片，联系电话，家庭住址，届，当前班级，是否在读，家庭状况，毕业去向）

学生就读班级表（学生ID，班级ID，学期）

档案管理员（管理员ID，管理员姓名）

班主任（班主任ID，班主任姓名，班级ID）

班级（班级ID，班名，班主任ID，班级人数，班级年级，班级届）

班级老师名册（班级ID，老师ID，科目）

班级学生名册（学生ID，班级ID）

(2)成绩管理E-R图

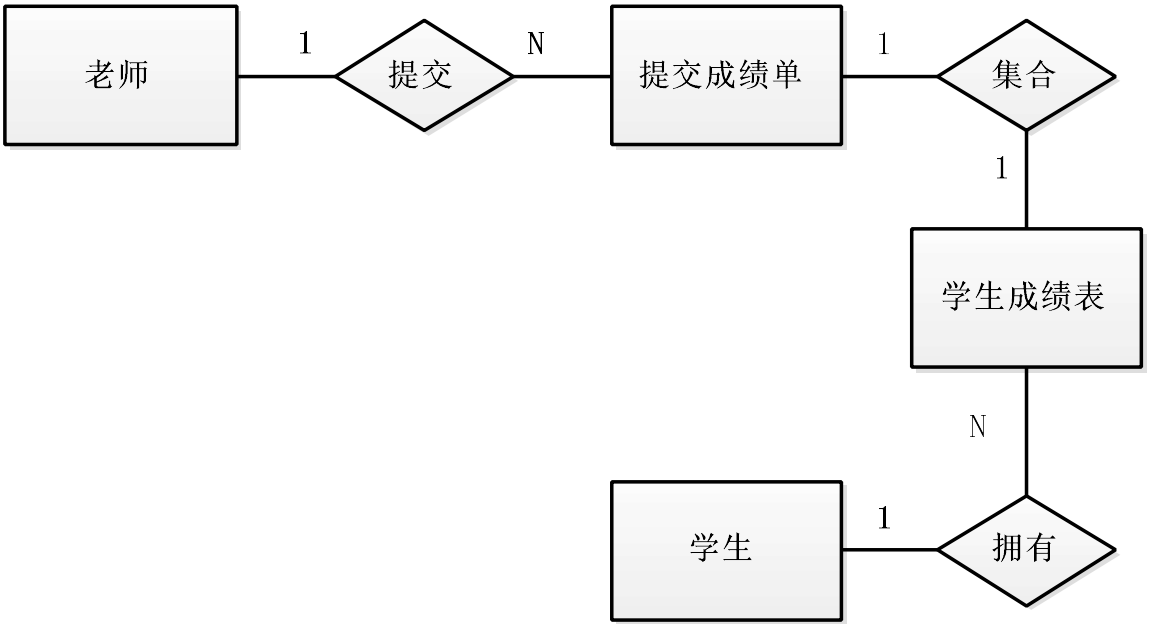


图 2-2 分E-R图—成绩管理

属性：

学生（学生ID，姓名）

老师（老师ID，老师姓名，科目）

提交成绩单（学生ID，成绩，科目，学期）

学生成绩表（学生ID，语文成绩，数学成绩，英语成绩，历史成绩，政治成绩，地理成绩，物理成绩，化学成绩，生物成绩，平均成绩，总成绩，成绩排名）

(3)入学管理E-R图

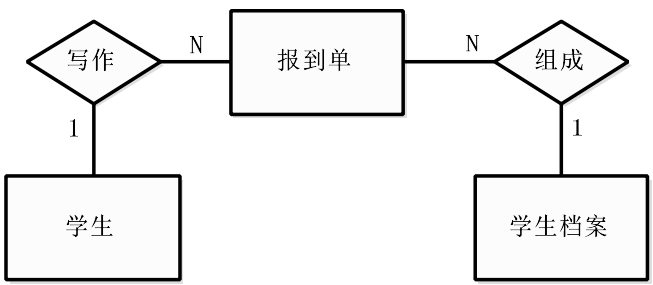


图 2-3 分E-R图——入学管理

属性：

学生（学生ID，姓名）

学生档案（学生ID，姓名，身份证号，姓名，年龄，出生日期，照片，联系电话，家庭住址，届，当前班级，是否在读，家庭状况，毕业去向）

报道单（学生ID，是否就读，学期，报道日期）

## 2.2 全局/整体E-R图

E-R图的合成主要包括两个部分：

1.合并：解决各分E-R图之间的冲突，将分E-R图合并起来生成初步E-R图；

2.修改和重构：消除不必要的冗余，生成基本E-R图。

各子系统的E-R图之间的冲突主要有三种，分别是：

属性冲突：属性域冲突——属性值的类型、取值范围或取值集合不同和属性取值单位冲突。

命名冲突：同名异义——不同意义的对象在不同的局部应用中具有相同的名字；异名同义——同一意义的对象在不同的局部应用中具有不同的名字。

结构冲突：同一对象在不同应用中具有不同的抽象；同一实体在不同分E-R图中所包含的属性个数和属性排列次序不完全相同；实体间的联系在不同的分E-R图中为不同类型。

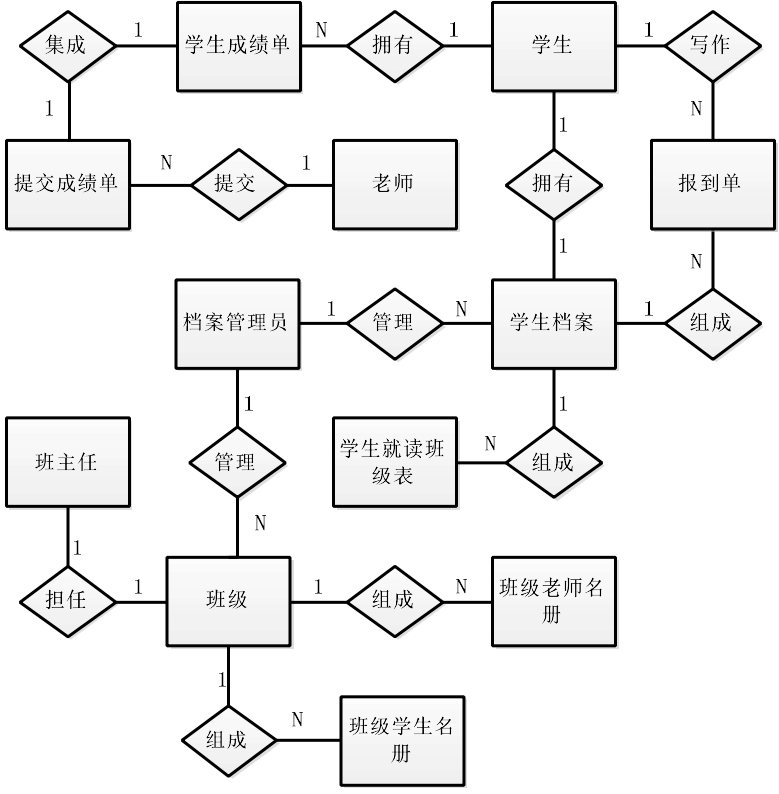


图2-4 全局E-R图

# 3 逻辑结构设计

逻辑结构是独立于任何一种数据模型的信息结构，逻辑结构设计的任务就是把概念结构设计阶段设计好的基本E-R图转换为与选用数据库管理系统产品所支持的数据模型相符合的逻辑结构。

也就是说进一步分析实体与实体间的联系转换成关系模式（通俗来说，就是表），重点是明确实体间到底是1:1还是n:m，实体与实体之间联系的纽带是什么。接着进行数据模型的优化，确定依赖，将关系进行分解，最后达到满足3NF的要求。最后建立用户子模式，建立若干视图。

具体任务包括将E-R图转换成关系模型、模型优化、数据库模式定义、用户模式设计。

## 3.1 建立关系模式

将E-R模型转换为关系模式的转换方法：

一个1:1联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。

一个1：n联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并。

一个m：n联系转换为一个关系模式。

三个或三个以上实体间的一个多元联系可以转换为一个关系模式。

具有相同码的关系模式可合并。

最终建立关系模式如下：

档案管理员（管理员ID，管理员姓名）

班主任（班主任ID，班主任姓名，班级ID）

老师（老师ID，老师姓名，科目）

学生（学生ID，姓名）

学生档案（学生ID，姓名，身份证号，姓名，年龄，出生日期，照片，联系电话，家庭住址，届，当前班级，是否在读，家庭状况，毕业去向）

学生就读班级表（学生ID，班级ID，学期）

班级（班级ID，班名，班主任ID，班级人数，班级年级，班级届）

班级老师名册（班级ID，老师ID，科目）

班级学生名册（学生ID，班级ID）

提交成绩单（学生ID，成绩，科目，学期）

学生成绩表（学生ID，语文成绩，数学成绩，英语成绩，历史成绩，政治成绩，地理成绩，物理成绩，化学成绩，生物成绩，平均成绩，总成绩，学期）

不及格成绩单（学生ID，成绩，科目，学期）

报道单（学生ID，是否就读，学期，报道日期）

## 3.2 关系模式规范化处理

根据转换规则得到的初步关系模式有可能会存在插入异常、更新异常、删除异常和数据冗余的现象，根据范式理论，分析了每个关系的主属性和非主属性，非主属性对主码的函数依赖，主要分析了转换得到的每个关系模式是否满足3NF，对不满足3NF的关系模式进行了模式分解，使各关系模式达到3NF的要求，从而有效消除插入异常、更新异常、删除异常和数据冗余等问题。

1. 档案管理员（管理员ID，管理员姓名）

主属性为管理员ID，非主属性为管理员姓名。对主属性时完全函数依赖关系，且没有传递函数依赖。

1. 班主任（班主任ID，班主任姓名）

主属性为班主任ID，非主属性为班主任姓名，班级ID。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 老师（老师ID，老师姓名，科目）

主属性为老师ID，非主属性为老师姓名，科目。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 学生（学生ID，姓名）

主属性为学生ID，非主属性为姓名。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 学生档案（学生ID，身份证号，性别，年龄，出生日期，照片，联系电话，家庭住址，届，当前学期，家庭状况，毕业去向）

主属性为学生ID，非主属性为身份证号，年龄，出生日期，照片，联系电话，家庭住址，届，当前学期，家庭状况，毕业去向。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 学生就读班级表（学生ID，学期，班级ID）

主属性为学生ID，学期, 非主属性为班级ID。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 班级（班级ID，班名，班主任ID，班级人数，班级年级，班级届）

主属性为班级ID，非主属性为班名，班主任ID，班级人数，班级年级，班级届。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 班级老师名册（班级ID，老师ID）

主属性为班级ID，老师ID。

1. 班级学生名册（学生ID，班级ID）

主属性为学生ID。非主属性为班级ID。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 提交成绩单（学生ID，老师ID，学期，成绩）

主属性为学生ID，科目，学期，非主属性为成绩。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 学生成绩表（学生ID，科目，学期，成绩）

主属性为学生ID，科目，学期，非主属性为成绩。对主属性时完全函数依赖关系，且没有传递函数依赖

1. 报道单（学生ID，学期，是否就读，报道日期）

主属性为学生ID，学期，非主属性为是否就读，报道日期。对主属性时完全函数依赖关系，且没有传递函数依赖

## 3.3 用户子模式设计

根据需求分析共设计了n个视图，分别为：

班级信息表：查询班级具体信息及老师信息

个人成绩表：学生查看自己的成绩

不及格成绩单：查看学生不及格成绩

退学学生信息表：查看所有退学学生的信息

贫困学生信息表：查看贫困学生的信息

视图的属性列表为：

表3-1 班级信息表

|  |  |
| --- | --- |
| **属性名** | **含义** |
| ClassID | 班级ID |
| ClassName | 班名 |
| ClassTeacherName | 班主任姓名 |
| ClassNumber | 班级人数 |
| ClassGrade | 班级年级 |
| ClassPeriod | 班级届 |
| ChineseTeacher | 语文老师姓名 |
| MathTeacher | 数学老师姓名 |
| EnglishTeacher | 英语老师姓名 |
| PoliticsTeacher | 政治老师姓名 |
| HistoryTeacher | 历史老师姓名 |
| GeographyTeacher | 地理老师姓名 |
| PhysicsTeacher | 物理老师姓名 |
| ChemistryTeacher | 化学老师姓名 |
| BiologyTeacher | 生物老师姓名 |

表3-2 个人成绩表

|  |  |
| --- | --- |
| **属性名** | **含义** |
| StuID | 学生ID |
| StuName | 学生姓名 |
| Semester | 学期 |
| ChineseGoal | 语文成绩 |
| MathGoal | 数学成绩 |
| EnglishGoal | 英语成绩 |
| PoliticsGoal | 历史成绩 |
| HistoryGoal | 政治成绩 |
| GeographyGoal | 地理成绩 |
| PhysicsGoal | 物理成绩 |
| ChemistryGoal | 化学成绩 |
| BiologyGoal | 生物成绩 |
| TotalGoal | 总成绩 |
| AvgGoal | 平均成绩 |

表3-3 不及格成绩单

|  |  |
| --- | --- |
| **属性名** | **含义** |
| StuID | 学生ID |
| StuName | 学生姓名 |
| Semester | 学期 |
| Goal | 成绩 |
| Subject | 科目 |

表3-4 退学学生信息表

| **属性名** | **含义** |
| --- | --- |
| StuID | 学生ID |
| StuName | 学生姓名 |
| IDNumber | 学生身份证号 |
| StuAge | 学生的年龄 |
| StuSex | 学生性别 |
| StuBirthday | 学生的出生日期 |
| StuImg | 学生的照片 |
| StuPhone | 学生的联系电话 |
| StuAddress | 学生的家庭住址 |
| StuPeriod | 学生届数 |
| ThisSem | 当前学期 |
| FamilyStatus | 学生的家庭状况 |
| Career | 学生毕业去向 |

表3-5 贫困学生信息表

| **属性名** | **含义** |
| --- | --- |
| StuID | 学生ID |
| StuName | 学生姓名 |
| IDNumber | 学生身份证号 |
| StuAge | 学生的年龄 |
| StuSex | 学生性别 |
| StuBirthday | 学生的出生日期 |
| StuImg | 学生的照片 |
| StuPhone | 学生的联系电话 |
| StuAddress | 学生的家庭住址 |
| StuPeriod | 学生届数 |
| ThisSem | 当前学期 |
| FamilyStatus | 学生的家庭状况 |
| Career | 学生毕业去向 |

## 3.4 关系模式逻辑结构定义

对每个关系模式要以表格形式描述其具体内容。例如，描述关系模式：

表3-1档案管理员

| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| --- | --- | --- | --- | --- | --- | --- |
| AdmID | 管理员ID | varchar | 10 | 是 |  |  |
| AdmName | 管理员姓名 | varchar | 20 |  |  |  |

表3-2 班主任

| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| --- | --- | --- | --- | --- | --- | --- |
| ClassTeacherID | 班主任ID | varchar | 10 | 是 |  |  |
| ClassTeacherName | 班主任姓名 | varchar | 20 |  |  |  |

表3-3老师

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| TeaID | 老师ID | varchar | 10 | 是 |  |  |
| TeaName | 老师姓名 | varchar | 20 |  |  |  |
| Subject | 老师所带科目 | varchar | 20 |  |  | {"语文","数学","英语","历史","政治","地理","物理","化学","生物"} |

表3-4 学生

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| StuID | 学生ID | varchar | 10 | 是 |  |  |
| StuName | 学生的姓名 | varchar | 20 |  |  |  |

表3-5 学生档案

| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| --- | --- | --- | --- | --- | --- | --- |
| StuID | 学生ID | varchar | 10 | 是 | 是 |  |
| IDNumber | 学生身份证号 | varchar | 20 |  |  |  |
| StuAge | 学生的年龄 | number | 5 |  |  | 1-100 |
| StuSex | 学生性别 | varchar | 5 |  |  | {男, 女} |
| StuBirthday | 学生的出生日期 | date |  |  |  | 1900/1/1-2100/1/1 |
| StuImg | 学生的照片 | blob |  |  |  |  |
| StuPhone | 学生的联系电话 | varchar | 20 |  |  |  |
| StuAddress | 学生的家庭住址 | varchar | 40 |  |  |  |
| StuPeriod | 学生届数 | number | 5 |  |  | 1900-2100 |
| ThisSem | 当前学期 | number | 1 |  |  | {1, 2, 3, 4, 5, 6} |
| FamilyStatus | 学生的家庭状况 | varchar | 10 |  |  | {非贫困, 贫困, 特困} |
| Career | 学生毕业去向 | varcahr | 50 |  |  |  |

表3-6 学生就读班级表

| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| --- | --- | --- | --- | --- | --- | --- |
| StuID | 学生ID | varchar | 10 | 是 | 是 |  |
| ClassID | 班级ID | varchar | 20 |  | 是 |  |
| Semester | 学期 | varchar | 1 | 是 |  | {1, 2, 3, 4, 5, 6} |

表3-7 班级

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| ClassID | 班级号 | varchar | 10 | 是 |  |  |
| ClassName | 班名 | varchar | 20 |  |  |  |
| ClassTeacherID | 班主任ID | varchar | 10 |  | 是 |  |
| ClassNumber | 班级人数 | number | 5 |  |  |  |
| ClassGrade | 班级的年级 | number | 1 |  |  | {1, 2, 3} |
| ClassPeriod | 班级的届 | number | 5 |  |  | 1900-2100 |

表3-8 班级老师名册

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| ClassID | 班级ID | varchar | 10 | 是 | 是 |  |
| TeaID | 老师ID | varchar | 10 | 是 | 是 |  |

表3-9 班级学生名册单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| ClassID | 班级ID | varchar | 10 | 是 | 是 |  |
| StuID | 学生ID | varchar | 10 | 是 | 是 |  |

表3-10 提交成绩单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| StuID | 学生ID | varchar | 10 | 是 | 是 |  |
| TeaID | 老师ID | varchar | 10 | 是 | 是 |  |
| Semester | 学期 | varchar | 1 | 是 |  | {1, 2, 3, 4, 5, 6} |
| Goal | 成绩 | number | 5 |  |  | [0,100] |

表3-11 学生成绩表

| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| --- | --- | --- | --- | --- | --- | --- |
| StuID | 学生ID | varchar | 10 | 是 | 是 |  |
| Subject | 科目 | varchar | 20 | 是 |  | {"语文","数学","英语","历史","政治","地理","物理","化学","生物"} |
| Semester | 学期 | varchar | 1 | 是 |  | {1, 2, 3, 4, 5, 6} |
| Goal | 成绩 | number | 5 |  |  | [0,100] |

表3-12 报道单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **属性名** | **含义** | **数据类型** | **长度** | **主键** | **外键** | **约束条件** |
| StuID | 学生ID | varchar | 10 | 是 | 是 |  |
| Semester | 学期 | varchar | 1 | 是 |  | {1, 2, 3, 4, 5, 6} |
| InSchool | 是否继续就读 | number | 1 |  |  | {0,1} |
| ReportDate | 报道日期 | date |  |  |  | 1900/1/1-2100/1/1 |

# 4．数据库物理设计

建立索引：

IND\_STUDENT：唯一类型，用于对学生的学生学号索引。

IND\_GOAL：聚簇类型，用于对成绩表中的学生学号添加索引。

IND\_GOALSUBJECT：聚簇类型，用于对成绩表中的科目添加索引

IND\_TEASUBJECT：聚簇类型，为老师表中的老师科目添加索引

# 5．数据库实施与测试

主要包括数据库实施和测试两个部分。

## 5.1 数据库及数据库对象建立

### 5.1.1创建基本表

1.创建档案管理员表

create table Administrator

(

AdmNo varchar2(10) primary key,

AdmName varchar2(20)

);

2.创建班主任表

create table

(

classteacherid varchar2(10) primary key,

classteachername varchar2(20)

);

3.创建老师表

create table Teacher

(

TeaID varchar2(10) primary key,

TeaName varchar2(20),

Subject varchar2(20),

check(Subject in('语文', '数学', '英语', '历史', '政治', '地理', '物理', '化学', '生物'))

);

4.创建学生表

create table Classteacher

(

ClassTeacherID varchar(10),

ClassTeacherName varchar(20),

primary key (ClassTeacherID)

)

5.创建学生档案

create table StudentFile

(

StuID varchar2(10) primary key,

IDNumber varchar2(20),

StuAge varchar2(5),

StuSex varchar2(5),

StuBirthday date,

StuImg blob,

StuPhone varchar2(20),

StuAddress varchar2(40),

StuPeriod number(5),

ThisSem number(1),

FamilyStatus varchar2(10),

Career varchar2(25),

foreign key (StuID) references Student(StuID),

constraint CK\_AGE check(StuAge >= 1 and StuAge <= 100),

constraint CK\_SEX check(StuSex in('男', '女'))

constraint CK\_BIRTHDAY check(StuBirthday > to\_date('1900-01-01','YYYY-MM-DD') and StuBirthday < to\_date('2100-01-01','YYYY-MM-DD')),

constraint CK\_PERIOD check(StuPeriod > 1900 and StuPeriod < 2100),

check(ThisSem in(1, 2, 3, 4, 5, 6)),

check(FamilyStatus in('非贫困', '贫困', '特困'))

)

6.学生就读班级表

create table StudentClass

(

StuID varchar2(10),

ClassID varchar2(10),

Semester varchar2(1),

primary key(StuID, Semester),

foreign key (StuID) references Student(StuID),

foreign key (ClassID) references Classes(ClassID)

)

7.创建班级表

create table Classes

(

ClassID varchar2(10),

ClassName varchar2(20),

ClassTeacherID varchar2(10),

ClassNumber number(5),

ClassGrade number(1),

ClassPeriod number(5),

primary key (ClassID),

foreign key (ClassTeacherID) references ClassTeacher(ClassTeacherID),

check(ClassGrade in(1, 2, 3)),

check(ClassPeriod > 1900 and ClassPeriod < 2100)

)

8.创建班级老师表

create table TeaClass

(

ClassID varchar2(10),

TeaID varchar2(10),

primary key (ClassID, TeaID),

foreign key (ClassID) references Classes(ClassId),

foreign key (TeaID) references Teacher(TeaID)

)

9.创建班级学生表

create table StuClass

(

ClassID varchar2(10),

StuID varchar2(10),

primary key (ClassID, StuID),

foreign key (ClassID) references Classes(ClassId),

foreign key (StuID) references Student(StuID)

)

10.创建提交成绩单

create table PutGoal

(

StuID varchar2(10),

TeaID varchar2(10),

Semester varchar2(1),

Goal number(5),

primary key (StuID, TeaID, Semester),

foreign key (StuID) references Student(StuID),

foreign key (TeaID) references Teacher(TeaID),

check(Semester in(1, 2, 3, 4, 5, 6)),

check(Goal >= 0 and Goal <= 100)

)

11.创建学生成绩单

create table StudentGoal

(

StuID varchar2(10),

Subject varchar2(20),

Semester varchar2(1),

Goal number(5),

primary key (StuID, Subject, Semester),

foreign key (StuID) references Student(StuID),

check(Subject in('语文', '数学', '英语', '历史', '政治', '地理', '物理', '化学', '生物')),

check(Semester in(1, 2, 3, 4, 5, 6)),

check(Goal >= 0 and Goal <= 100)

)

12.创建学生报道单

create table ReportTable

(

StuID varchar2(10),

Semester varchar2(1),

InSchool number(1),

ReportDate date,

primary key (StuID, Semester),

foreign key (StuID) references Student(StuID),

check(Semester in(1, 2, 3, 4, 5, 6)),

check(InSchool in(0, 1)),

check(ReportDate> to\_date('1900-01-01','YYYY-MM-DD') and ReportDate < to\_date('2100-01-01','YYYY-MM-DD'))

)

### 5.1.2创建视图

1.班级信息表

create or replace view classmessage

as

select

t1.ClassID, t1.Classname, t1.ClassTeacherName, t1.Classnumber, t1.Classgrade, t1.Classperiod,

t2.TeaName ChineseTeacher,

t3.TeaName MathTeacher,

t4.TeaName EnglishTeacher,

t5.TeaName PoliticsTeacher,

t6.TeaName HistoryTeacher,

t7.TeaName GeographyTeacher,

t8.TeaName PhysicsTeacher,

t9.TeaName ChemistryTeacher,

t10.TeaName BiologyTeacher

from

(

select \* from Classes, Classteacher

where Classes.Classteacherid = Classteacher.Classteacherid

)t1,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='语文' and TeaClass.Teaid = Teacher.Teaid

)t2,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='数学' and TeaClass.Teaid = Teacher.Teaid

)t3,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='英语' and TeaClass.Teaid = Teacher.Teaid

)t4,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='政治' and TeaClass.Teaid = Teacher.Teaid

)t5,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='历史' and TeaClass.Teaid = Teacher.Teaid

)t6,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='地理' and TeaClass.Teaid = Teacher.Teaid

)t7,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='物理' and TeaClass.Teaid = Teacher.Teaid

)t8,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='化学' and TeaClass.Teaid = Teacher.Teaid

)t9,

(

select \*

from TeaClass, Teacher

where Teacher.Subject='生物' and TeaClass.Teaid = Teacher.Teaid

)t10

where

t1.ClassID=t2.ClassID and

t1.ClassID=t3.ClassID and

t1.ClassID=t4.ClassID and

t1.ClassID=t5.ClassID and

t1.ClassID=t6.ClassID and

t1.ClassID=t7.ClassID and

t1.ClassID=t8.ClassID and

t1.ClassID=t9.ClassID and

t1.ClassID=t10.ClassID;

2.个人成绩单

create or replace view PersonalGoalTable

as

select t1.StuID, t1.StuName, T2.Semester,

T2.Goal ChineseGoal,

t3.Goal MathGoal,

t4.Goal EnglishGoal,

t5.Goal PoliticsGoal,

t6.Goal HistoryGoal,

t7.Goal GeographyGoal,

t8.Goal PhysicsGoal,

t9.Goal ChemistryGoal,

t10.Goal BiologyGoal，

t11.TotalGoal,

t11.AvgGoal

from

(

select \* from Student

)t1,

(

select \* from StudentGoal

where Subject='语文'

)t2,

(

select \* from StudentGoal

where Subject='数学'

)t3,

(

select \* from StudentGoal

where Subject='英语'

)t4,

(

select \* from StudentGoal

where Subject='政治'

)t5,

(

select \* from StudentGoal

where Subject='历史'

)t6,

(

select \* from StudentGoal

where Subject='地理'

)t7,

(

select \* from StudentGoal

where Subject='物理'

)t8,

(

select \* from StudentGoal

where Subject='化学'

)t9,

(

select \* from StudentGoal

where Subject='生物'

)t10,

(

select StuID, Semester, sum(Goal) TotalGoal, avg(Goal) AvgGoal

from StudentGoal

group by StuID, Semester

)t11

where

t1.StuID = t2.StuID and

t2.StuID = t3.StuID and t2.Semester = t3.Semester and

t2.StuID = t4.StuID and t2.Semester = t4.Semester and

t2.StuID = t5.StuID and t2.Semester = t5.Semester and

t2.StuID = t6.StuID and t2.Semester = t6.Semester and

t2.StuID = t7.StuID and t2.Semester = t7.Semester and

t2.StuID = t8.StuID and t2.Semester = t8.Semester and

t2.StuID = t9.StuID and t2.Semester = t9.Semester and

t2.StuID = t10.StuID and t2.Semester = t10.Semester and

t2.StuID = t11.StuID and t2.Semester = t11.Semester;

3.不及格成绩单

create or replace view UnPastGoalTable

as

select Student.StuId, Student.Stuname, StudentGoal.Subject, StudentGoal.Semester, StudentGoal.Goal

from Student, StudentGoal

where Student.StuID = StudentGoal.StuID and StudentGoal.Goal < 60

4.退学学生信息表

create or replace view DropoutStudentTable

as

select Student.Stuid, Student.StuName, Studentfile.Idnumber, Studentfile.Stuage, Studentfile.Stusex, Studentfile.Stubirthday, Studentfile.Stuimg, Studentfile.Stuphone, Studentfile.Stuaddress, Studentfile.Stuperiod, Studentfile.Familystatus, Studentfile.Career

from Student, Studentfile

where Student.Stuid in(

select Stuid

from ReportTable

where InSchool = 0

)

### 5.1.3创建索引

1.建立学生表的学号索引

create unique index IND\_STUDENT on Student(StuID);

2.建立成绩表中学号索引

create cluster index IND\_GOAL on StudentGoal(StuID);

3.建立成绩表中科目索引

create cluster index IND\_GOALSUBJECT on StudentGoal(Subject);

4.建立老师中科目索引

create cluster index IND\_TEASUBJECT on Teacher(Subject);

### 5.1.4创建存储过程

表5-1 存储过程汇总表

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **命名** | **输入** | **描述** |
| 提交成绩 | PRO\_PUTGOAL | 成绩信息 | 老师提交成绩，并插入学生成绩表 |
| 修改成绩 | PRO\_CHANGEGOAL | 成绩信息 | 老师修改成绩，并修改学产生成绩表 |
| 学生班级设置 | PRO\_STUCLASS | 学生、班级ID和学期 | 添加学生班级信息，修改班级信息，修改班级学生名册 |
| 报道 | PRO\_REPORT | 学生报道信息 | 填写学生报道信息，并修改学生档案 |
| 学生存档 | PRO\_INSTUFILE | 学生信息 | 添加学生信息 |
| 学生档案修改 | PRO\_INSTUFILE | 学生信息 | 修改学生信息 |
| 班级存档 | PRO\_ClASSES | 班级信息信息 | 添加班级信息 |
| 班级档案修改 | PRO\_ClASSES | 班级信息信息 | 修改班级信息 |
| 个人成绩查询 | PRO\_PERSONALGOAL | 学号 | 查询各个学期的成绩 |
| 班级成绩查询 | PRO\_CLASSGOAL | 班级号及学期 | 查询班级所有学生的成绩，并进行排名，统计分数段 |
| 科目成绩查询 | PRO\_SUBGOAL | 科目、学期及年级 | 查询单科成绩，并进行排名，统计分数段 |
| 年级成绩查询 | PRO\_GRADEGOAL | 年级及学期 | 查询年级成绩，并进行排名，统计分数段 |

1.创建提交成绩存储过程

\\*

提交成绩描述

1.传入参数（老师ID，学生ID，成绩，学期）

2.插入提交成绩单中的成绩

3.查找老师所带科目

4.提交学生成绩单

\*\create or replace procedure PRO\_PUTGOAL(iTeaID in Teacher.TeaID%type, iStuID in Student.StuID%type, iGoal in PutGoal.Goal%type, iSemester in PutGoal.Semester%type) is

iSubject Teacher.Subject%type;

begin

--插入到提交成绩单

insert into PutGoal values(iStuID, iTeaid, iSemester, iGoal);

--插入到学生成绩单

select Subject into iSubject from Teacher where TeaID = iTeaID;

insert into StudentGoal values(iStuID, iSubject, iSemester, iGoal);

end PRO\_PUTGOAL;

2.创建修改成绩存储过程

\\*

修改成绩成绩描述

1.传入参数（老师ID，学生ID，成绩，学期）

2.更新提交成绩单中的成绩

3.查找老师所带科目

4.更新学生成绩单

\*\

create or replace procedure PRO\_ChangeGOAL(iTeaID in Teacher.TeaID%type, iStuID in Student.StuID%type, iGoal in PutGoal.Goal%type, iSemester in PutGoal.Semester%type) is

iSubject Teacher.Subject%type;

begin

--插入到提交成绩单

update PutGoal set Goal = iGoal

where StuID = iStuID and Teaid = iTeaid and Semester = iSemester;

--插入到学生成绩单

select Subject into iSubject from Teacher where TeaID = iTeaID;

update StudentGoal set Goal = iGoal

where StuID = iStuID and Subject = iSubject and Semester = iSemester;

end PRO\_ChangeGOAL;

3.创建学生班级设置存储过程

\\*

创建学生班级描述

1.传入参数（学生ID，班级ID，学期）

2.插入到就读历史中

3.若学期为一，插入到班级学生表中，不为一更新班级学生表

\*\

create or replace procedure PRO\_STUCLASS(iStuID in StudentClass.Stuid%type, iClassID in Studentclass.Classid%type, iSemester in Studentclass.Semester%type) is

mclassid StuClass.Classid%type;

begin

if iSemester != 1 --学期不为1，更新

then

update StuClass set ClassID = iClassID where StuID = iStuID;

else --学期为1，插入

insert into StuClass values(iClassID, iStuID);

end if;

--插入到就读历史中

insert into StudentClass values(iStuID, iClassID, iSemester);

end PRO\_STUCLASS;

4.创建报道存储过程

/\*

报道描述

1. 传入参数（学生ID，是否继续就读，学期，日期）
2. 插入报道表中
3. 若学生退学，删除班级学生信息
4. 若不退学，更新学生档案表

\*/

create or replace procedure PRO\_REPORT(iStuID in Reporttable.Stuid%type, iInSchool in Reporttable.Inschool%type, iSemester in Reporttable.Semester%type, iDate in varchar2) is

begin

--插入到报道表中

insert into Reporttable values(iStuID, iSemester, iInSchool, to\_date(iDate, 'YYYY-MM-DD'));

if iInSchool = '0' --退学

then

delete from StuClass where StuID = iStuID;

else --继续学习

update StudentFile set Thissem = iSemester where StuID = iStuID;

if iSemester != 1 --不是第一学期

then

insert into StudentClass values(iStuID, (select ClassID from StuClass where iStuID = StuID), iSemester);

end if;

end if;

end PRO\_REPORT;

1. 创建学生存档存储过程

/\*

插入学生档案存储过程

1.传入参数（学生ID，学生身份证号，学生年龄···）

2.对传入参数进行越界检查

3.插入学生档案表中

\*/

create or replace procedure PRO\_INSTUFILE(iStuID in varchar2,

iIDNumber in varchar2,

iStuAge in varchar2,

iStuSex in varchar2,

iStuBirthday in varchar2,

iFileName in varchar2,

iStuPhone in varchar2,

iStuAddress in varchar2,

iStuPeriod in number,

iThisSem in number,

iFamilyStatus in varchar2) is

ju number := 1;

mStuID varchar2(10);

--插入图片要使用的数据

l\_bfile bfile;

l\_blob blob;

begin

if iStuAge not between 1 and 100

then

dbms\_output.put\_line('年龄错误');

ju := 0;

end if;

if iStuSex not in('男', '女')

then

dbms\_output.put\_line('性别错误');

ju := 0;

end if;

if iStuBirthday not between '1900-1-1' and '2100-1-1'

then

dbms\_output.put\_line('年龄错误');

ju := 0;

end if;

if iStuPeriod not between 1900 and 2100

then

dbms\_output.put\_line('届错误');

ju := 0;

end if;

if iThisSem not in(1, 2, 3, 4, 5, 6)

then

dbms\_output.put\_line('学期错误');

ju := 0;

end if;

if iFamilyStatus not in('非贫困', '贫困', '特困')

then

dbms\_output.put\_line('家庭状况错误');

ju := 0;

end if;

if ju = 0

then

dbms\_output.put\_line('有数据错误，请修改数据');

else

insert into Studentfile Values(

iStuID,

iIDNumber,

iStuAge,

iStuSex,

to\_date(iStuBirthday, 'YYYY-MM-DD'),

EMPTY\_BLOB (),

iStuPhone,

iStuAddress,

iStuPeriod,

iThisSem,

iFamilyStatus

) return StuImg into l\_blob;

l\_bfile:=bfilename('IMAGES',iFileName);

dbms\_output.put\_line(iFileName);

dbms\_lob.open(l\_bfile,dbms\_lob.file\_readonly);

dbms\_lob.loadfromfile(l\_blob,l\_bfile,dbms\_lob.getlength(l\_bfile));

dbms\_lob.close(l\_bfile);

commit;

end if;

end PRO\_INSTUFILE;

1. 创建学生档案修改存储过程

/\*

更新学生档案存储过程

1.传入参数（学生ID，学生身份证号，学生年龄···）

2.对传入参数进行越界检查

3.跟新学生档案表中

\*/

create or replace procedure PRO\_UPSTUFILE(iStuID in varchar2,

iIDNumber in varchar2,

iStuAge in varchar2,

iStuSex in varchar2,

iStuBirthday in varchar2,

iFileName in varchar2,

iStuPhone in varchar2,

iStuAddress in varchar2,

iStuPeriod in number,

iThisSem in number,

iFamilyStatus in varchar2) is

ju number := 1;

mStuID varchar2(10);

--插入图片要使用的数据

l\_bfile bfile;

l\_blob blob;

begin

if iStuAge not between 1 and 100

then

dbms\_output.put\_line('年龄错误');

ju := 0;

end if;

if iStuSex not in('男', '女')

then

dbms\_output.put\_line('性别错误');

ju := 0;

end if;

if iStuBirthday not between '1900-1-1' and '2100-1-1'

then

dbms\_output.put\_line('年龄错误');

ju := 0;

end if;

if iStuPeriod not between 1900 and 2100

then

dbms\_output.put\_line('届错误');

ju := 0;

end if;

if iThisSem not in(1, 2, 3, 4, 5, 6)

then

dbms\_output.put\_line('学期错误');

ju := 0;

end if;

if iFamilyStatus not in('非贫困', '贫困', '特困')

then

dbms\_output.put\_line('家庭状况错误');

ju := 0;

end if;

if ju = 0

then

dbms\_output.put\_line('有数据错误，请修改数据');

else

update Studentfile set

IDNumber = iIDNumber,

StuAge = iStuAge,

StuSex = iStuSex,

StuBirthday = to\_date(iStuBirthday, 'YYYY-MM-DD'),

StuImg = EMPTY\_BLOB (),

StuPhone = iStuPhone,

StuAddress = iStuAddress,

StuPeriod = iStuPeriod,

ThisSem = iThisSem,

FamilyStatus = iFamilyStatus

where StuID = iStuID

return StuImg into l\_blob;

l\_bfile:=bfilename('IMAGES',iFileName);

dbms\_output.put\_line(iFileName);

dbms\_lob.open(l\_bfile,dbms\_lob.file\_readonly);

dbms\_lob.loadfromfile(l\_blob,l\_bfile,dbms\_lob.getlength(l\_bfile));

dbms\_lob.close(l\_bfile);

commit;

end if;

end PRO\_UPSTUFILE;

1. 创建班级存档存储过程

/\*

插入班级档案存储过程

1.传入参数（班级ID，班级名，班主任ID，班级年级，班级届）

2.对传入参数进行越界检查

3.插入班级表中

\*/

create or replace procedure PRO\_INCLASSES(iClassId in Classes.Classid%type, iClassName in Classes.Classname%type, iClassTeacherID in Classes.Classteacherid%type, iClassGrade in Classes.ClassGrade%type,iClassPeriod Classes.Classperiod%type) is

ju number := 1;

begin

--判断年级是否正确

if iClassGrade not in (1, 2, 3)

then

dbms\_output.put\_line('年级错误');

ju := 0;

end if;

--判读届数是否正确

if iClassPeriod not between 1900 and 2100

then

dbms\_output.put\_line('届数错误');

ju := 0;

end if;

--插入数据

if ju = 1

then

insert into Classes values(iClassId, iClassName, iClassTeacherID, 0, iClassGrade, iClassPeriod);

end if;

end PRO\_INCLASSES;

1. 创建班级档案修改存储过程

/\*

班级档案修改存储过程

1.传入参数（班级ID，班级名，班主任ID，班级年级，班级届）

2.对传入参数进行越界检查

3.更新班级表中

\*/

create or replace procedure PRO\_UPCLASSES(iClassId in Classes.Classid%type, iClassName in Classes.Classname%type, iClassTeacherID in Classes.Classteacherid%type, iClassNumber in Classes.Classnumber%type, iClassGrade in Classes.ClassGrade%type,iClassPeriod Classes.Classperiod%type) is

ju number := 1;

begin

--判断年级是否正确

if iClassGrade not in (1, 2, 3)

then

dbms\_output.put\_line('年级错误');

ju := 0;

end if;

--判读届数是否正确

if iClassPeriod not between 1900 and 2100

then

dbms\_output.put\_line('届数错误');

ju := 0;

end if;

--插入数据

if ju = 1

then

update Classes set

ClassName = iClassName,

ClassTeacherID = iClassTeacherID,

ClassNumber = iClassNumber,

ClassGrade = iClassGrade,

ClassPeriod = iClassPeriod

where ClassID = iClassID;

end if;

end PRO\_UPCLASSES;

1. 创建个人成绩查询存储过程

/\*

查询个人成绩存储过程描述

1.传入参数（学生ID）

2.定义游标，查询学生的成绩

3.循环输出学生每个学期的成绩

\*/

create or replace procedure PRO\_PERSONALGOAL(iStuID in Student.StuID%type) is

id Student.StuID%type;

stuname Student.Stuname%type;

Semester StudentGoal.Semester%type;

ChineseGoal StudentGoal.Goal%type;

MathGoal StudentGoal.Goal%type;

EnglishGoal StudentGoal.Goal%type;

PoliticsGoal StudentGoal.Goal%type;

HistoryGoal StudentGoal.Goal%type;

GeographyGoal StudentGoal.Goal%type;

PhysicsGoal StudentGoal.Goal%type;

ChemistryGoal StudentGoal.Goal%type;

BiologyGoal StudentGoal.Goal%type;

TotalGoal StudentGoal.Goal%type;

AvgGoal StudentGoal.Goal%type;

cursor c1 is

select \* from Personalgoaltable

where StuID = iStuID;

begin

open c1;

loop

fetch c1 into

id, stuname, Semester,ChineseGoal,MathGoal,EnglishGoal,PoliticsGoal,HistoryGoal,GeographyGoal,PhysicsGoal,ChemistryGoal,BiologyGoal,TotalGoal,AvgGoal;

exit when c1%notfound;

dbms\_output.put\_line( ' 姓名：' || stuname || ' 学期：' || Semester || ' 语文成绩：' || ChineseGoal || ' 数学成绩：' || MathGoal || ' 英语成绩：' || EnglishGoal || ' 政治成绩：' || PoliticsGoal || ' 历史成绩：' ||HistoryGoal|| ' 地理成绩：' || GeographyGoal || ' 物理成绩：' || PhysicsGoal || ' 化学成绩：' || ChemistryGoal || ' 生物成绩：' ||BiologyGoal || ' 总成绩：' || TotalGoal || ' 平均成绩：');

end loop;

end PRO\_PERSONALGOAL;

1. 创建班级成绩查询存储过程

/\*

查询班级成绩存储过程描述

1.传入参数（班级ID，学期）

2.定义游标，查询班级学生的成绩并排序

3.循环输出班级成绩，并统计分数段

4.输出分数段

\*/

create or replace procedure PRO\_ClassGoal(iClassID in Classes.Classid%type, iSemester in Studentgoal.Semester%type) is

stuid Student.Stuid%type;

stuname Student.Stuname%type;

ChineseGoal StudentGoal.Goal%type;

MathGoal StudentGoal.Goal%type;

EnglishGoal StudentGoal.Goal%type;

PoliticsGoal StudentGoal.Goal%type;

HistoryGoal StudentGoal.Goal%type;

GeographyGoal StudentGoal.Goal%type;

PhysicsGoal StudentGoal.Goal%type;

ChemistryGoal StudentGoal.Goal%type;

BiologyGoal StudentGoal.Goal%type;

TotalGoal StudentGoal.Goal%type;

AvgGoal StudentGoal.Goal%type;

i number := 1;

yx number := 0;

lh number := 0;

jg number := 0;

bjg number := 0;

cursor c1 is

select t1.StuID, t1.StuName,

t2.Goal ChineseGoal,

t3.Goal MathGoal,

t4.Goal EnglishGoal,

t5.Goal PoliticsGoal,

t6.Goal HistoryGoal,

t7.Goal GeographyGoal,

t8.Goal PhysicsGoal,

t9.Goal ChemistryGoal,

t10.Goal BiologyGoal，

t11.TotalGoal,

t11.AvgGoal

from

(

select \* from Student

)t1,

(

select \* from StudentGoal

where Subject='语文'

)t2,

(

select \* from StudentGoal

where Subject='数学'

)t3,

(

select \* from StudentGoal

where Subject='英语'

)t4,

(

select \* from StudentGoal

where Subject='政治'

)t5,

(

select \* from StudentGoal

where Subject='历史'

)t6,

(

select \* from StudentGoal

where Subject='地理'

)t7,

(

select \* from StudentGoal

where Subject='物理'

)t8,

(

select \* from StudentGoal

where Subject='化学'

)t9,

(

select \* from StudentGoal

where Subject='生物'

)t10,

(

select StuID, Semester, sum(Goal) TotalGoal, avg(Goal) AvgGoal

from StudentGoal

group by StuID, Semester

)t11,

(

select \* from StuClass

)t12

where

t12.ClassID = iClassID and

t1.StuID = t12.StuID and

t1.StuID = t2.StuID and

t2.StuID = t3.StuID and t2.Semester = iSemester and

t2.StuID = t4.StuID and t2.Semester = iSemester and

t2.StuID = t5.StuID and t2.Semester = iSemester and

t2.StuID = t6.StuID and t2.Semester = iSemester and

t2.StuID = t7.StuID and t2.Semester = iSemester and

t2.StuID = t8.StuID and t2.Semester = iSemester and

t2.StuID = t9.StuID and t2.Semester = iSemester and

t2.StuID = t10.StuID and t2.Semester = iSemester and

t2.StuID = t11.StuID and t2.Semester = iSemester

order by AvgGoal desc;

begin

open c1;

loop

fetch c1 into

stuid,stuname,ChineseGoal,MathGoal,EnglishGoal,PoliticsGoal,HistoryGoal,GeographyGoal,PhysicsGoal,ChemistryGoal,BiologyGoal,TotalGoal,AvgGoal;

exit when c1%notfound;

dbms\_output.put\_line('学号：'|| stuid || ' 姓名：' || stuname || ' 语文成绩：' || ChineseGoal || ' 数学成绩：' || MathGoal || ' 英语成绩：' || EnglishGoal || ' 政治成绩：' || PoliticsGoal || ' 历史成绩：' ||HistoryGoal|| ' 地理成绩：' || GeographyGoal || ' 物理成绩：' || PhysicsGoal || ' 化学成绩：' || ChemistryGoal || ' 生物成绩：' ||BiologyGoal || ' 总成绩：' || TotalGoal || ' 平均成绩：' || AvgGoal|| ' 排名：'||i);

i := i + 1;

if AvgGoal >= 85

then

yx := yx + 1;

elsif AvgGoal >= 75

then

lh := lh + 1;

elsif AvgGoal >= 60

then

jg := jg + 1;

else

bjg := bjg + 1;

end if;

end loop;

dbms\_output.put\_line('平均成绩优秀（大于85）人数：' || yx);

dbms\_output.put\_line('平均成绩良好（75到84）人数：' || lh);

dbms\_output.put\_line('平均成绩及格（60到74）人数：' || jg);

dbms\_output.put\_line('平均成绩不及格(小于60)人数：' || bjg);

end ;

1. 创建科目成绩查询存储过程

/\*

查询科目成绩存储过程描述

1.传入参数（科目，学期，年级）

2.定义游标，查询学生的成绩并排序

3.循环输出科目成绩，并统计分数段

4.输出分数段

\*/

create or replace procedure PRO\_SubjectGoal(iSubject in StudentGoal.Subject%type, iSemester in Studentgoal.Semester%type, iGrade in Classes.Classgrade%type) is

StudentID Student.Stuid%type;

StudentName Student.Stuname%type;

Goal StudentGoal.Goal%type;

i number := 1;

yx number := 0;

lh number := 0;

jg number := 0;

bjg number := 0;

cursor c1 is

--查询对应科目，对应学期，对应年级的成绩

select Student.StuID, Student.StuName, StudentGoal.Goal

from Student, StudentGoal

where Student.StuID = StudentGoal.StuID and StudentGoal.Semester = iSemester and StudentGoal.Subject = iSubject and

Student.StuID in (

select StuClass.StuID

from StuClass, Classes

where StuClass.ClassID = Classes.ClassID and Classes.ClassGrade = iGrade

)

order by StudentGoal.Goal desc;

begin

--输出成绩

open c1;

loop

fetch c1 into

StudentID, StudentName, Goal;

exit when c1%notfound;

dbms\_output.put\_line('学号：'|| StudentID || ' 姓名：' || StudentName || ' 成绩：' || Goal || ' 排名：'||i);

if Goal >= 85

then

yx := yx + 1;

elsif Goal >= 75

then

lh := lh + 1;

elsif Goal >= 60

then

jg := jg + 1;

else

bjg := bjg + 1;

end if;

i := i+1;

end loop;

dbms\_output.put\_line('优秀（大于85）人数：' || yx);

dbms\_output.put\_line('良好（75到84）人数：' || lh);

dbms\_output.put\_line('及格（60到74）人数：' || jg);

dbms\_output.put\_line('不及格(小于60)人数：' || bjg);

end PRO\_SubjectGoal;

1. 创建年级成绩查询存储过程

/\*

查询年级成绩存储过程描述

1.传入参数（年级，学期）

2.定义游标，查询年级学生的成绩并排序

3.循环输出学生成绩，并统计分数段

4.输出分数段

\*/

create or replace procedure PRO\_GradeGoal(iGrade in Classes.ClassGrade%type, iSemester in Studentgoal.Semester%type) is

stuid Student.Stuid%type;

stuname Student.Stuname%type;

ChineseGoal StudentGoal.Goal%type;

MathGoal StudentGoal.Goal%type;

EnglishGoal StudentGoal.Goal%type;

PoliticsGoal StudentGoal.Goal%type;

HistoryGoal StudentGoal.Goal%type;

GeographyGoal StudentGoal.Goal%type;

PhysicsGoal StudentGoal.Goal%type;

ChemistryGoal StudentGoal.Goal%type;

BiologyGoal StudentGoal.Goal%type;

TotalGoal StudentGoal.Goal%type;

AvgGoal StudentGoal.Goal%type;

i number := 1;

yx number := 0;

lh number := 0;

jg number := 0;

bjg number := 0;

cursor c1 is

select t1.StuID, t1.StuName,

t2.Goal ChineseGoal,

t3.Goal MathGoal,

t4.Goal EnglishGoal,

t5.Goal PoliticsGoal,

t6.Goal HistoryGoal,

t7.Goal GeographyGoal,

t8.Goal PhysicsGoal,

t9.Goal ChemistryGoal,

t10.Goal BiologyGoal，

t11.TotalGoal,

t11.AvgGoal

from

(

select \* from Student

)t1,

(

select \* from StudentGoal

where Subject='语文'

)t2,

(

select \* from StudentGoal

where Subject='数学'

)t3,

(

select \* from StudentGoal

where Subject='英语'

)t4,

(

select \* from StudentGoal

where Subject='政治'

)t5,

(

select \* from StudentGoal

where Subject='历史'

)t6,

(

select \* from StudentGoal

where Subject='地理'

)t7,

(

select \* from StudentGoal

where Subject='物理'

)t8,

(

select \* from StudentGoal

where Subject='化学'

)t9,

(

select \* from StudentGoal

where Subject='生物'

)t10,

(

select StuID, Semester, sum(Goal) TotalGoal, avg(Goal) AvgGoal

from StudentGoal

group by StuID, Semester

)t11,

(

select \* from StuClass

)t12,

(

select \* from Classes

)t13

where

t13.ClassGrade = iGrade and

t12.ClassID = t13.ClassID and

t1.StuID = t12.StuID and

t1.StuID = t2.StuID and

t2.StuID = t3.StuID and t2.Semester = iSemester and

t2.StuID = t4.StuID and t2.Semester = iSemester and

t2.StuID = t5.StuID and t2.Semester = iSemester and

t2.StuID = t6.StuID and t2.Semester = iSemester and

t2.StuID = t7.StuID and t2.Semester = iSemester and

t2.StuID = t8.StuID and t2.Semester = iSemester and

t2.StuID = t9.StuID and t2.Semester = iSemester and

t2.StuID = t10.StuID and t2.Semester = iSemester and

t2.StuID = t11.StuID and t2.Semester = iSemester

order by AvgGoal desc;

begin

open c1;

loop

fetch c1 into

stuid,stuname,ChineseGoal,MathGoal,EnglishGoal,PoliticsGoal,HistoryGoal,GeographyGoal,PhysicsGoal,ChemistryGoal,BiologyGoal,TotalGoal,AvgGoal;

exit when c1%notfound;

dbms\_output.put\_line('学号：'|| stuid || ' 姓名：' || stuname || ' 语文成绩：' || ChineseGoal || ' 数学成绩：' || MathGoal || ' 英语成绩：' || EnglishGoal || ' 政治成绩：' || PoliticsGoal || ' 历史成绩：' ||HistoryGoal|| ' 地理成绩：' || GeographyGoal || ' 物理成绩：' || PhysicsGoal || ' 化学成绩：' || ChemistryGoal || ' 生物成绩：' ||BiologyGoal || ' 总成绩：' || TotalGoal || ' 平均成绩：' || AvgGoal|| ' 排名：'||i);

i := i + 1;

if AvgGoal >= 85

then

yx := yx + 1;

elsif AvgGoal >= 75

then

lh := lh + 1;

elsif AvgGoal >= 60

then

jg := jg + 1;

else

bjg := bjg + 1;

end if;

end loop;

dbms\_output.put\_line('优秀（平均成绩大于85）人数：' || yx);

dbms\_output.put\_line('良好（平均成绩75到84）人数：' || lh);

dbms\_output.put\_line('及格（平均成绩60到74）人数：' || jg);

dbms\_output.put\_line('不及格(平均成绩小于60)人数：' || bjg);

end ;

### 5.1.5创建触发器

5-2 触发器汇总表

|  |  |  |
| --- | --- | --- |
| **命名** | **作用表** | **描述** |
| T\_INSTUCLASS | StuClass | 班级学生名称插入时，修改班级人数信息 |
| T\_UPSTUCLASS | StuClass | 班级学生名称修改时，修改班级人数信息 |
| T\_DESTUCLASS | StuClass | 班级学生名称删除时，修改班级人数信息 |

1. T\_INSTUCLASS发器

create or replace trigger T\_INSTUCLASS

before insert

on stuclass

for each row

declare

-- local variables here

begin

--班级人数加1

update Classes set ClassNumber = ClassNumber + 1 where ClassID = :new.classid;

end T\_INSTUCLASS;

2. T\_UPSTUCLASS触发器

create or replace trigger T\_UPSTUCLASS

before update

on stuclass

for each row

declare

-- local variables here

begin

--旧班级人数减1

update Classes set ClassNumber = ClassNumber - 1 where ClassID = :old.classid;

--新班级人数加1

update Classes set ClassNumber = ClassNumber + 1 where ClassID = :new.classid;

end T\_INSTUCLASS;

3. T\_DESTUCLASS触发器

create or replace trigger T\_DESTUCLASS

before update

on stuclass

for each row

declare

-- local variables here

begin

--更新原班级人数

update Classes set ClassNumber = ClassNumber - 1 where ClassID = :old.classid;

end T\_INSTUCLASS;

## 5.2 数据入库

1.管理员信息表（10行数据）

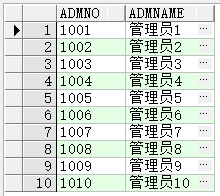


图5-1管理员信息表

2. 班主任信息表（10行数据）

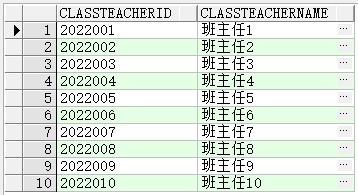


图5-2班主任信息表

3. 老师信息表（27行数据）



图5-3老师信息表

4. 学生表（100行数据）



图5 -4学生表

5. 学生档案表（100行数据）

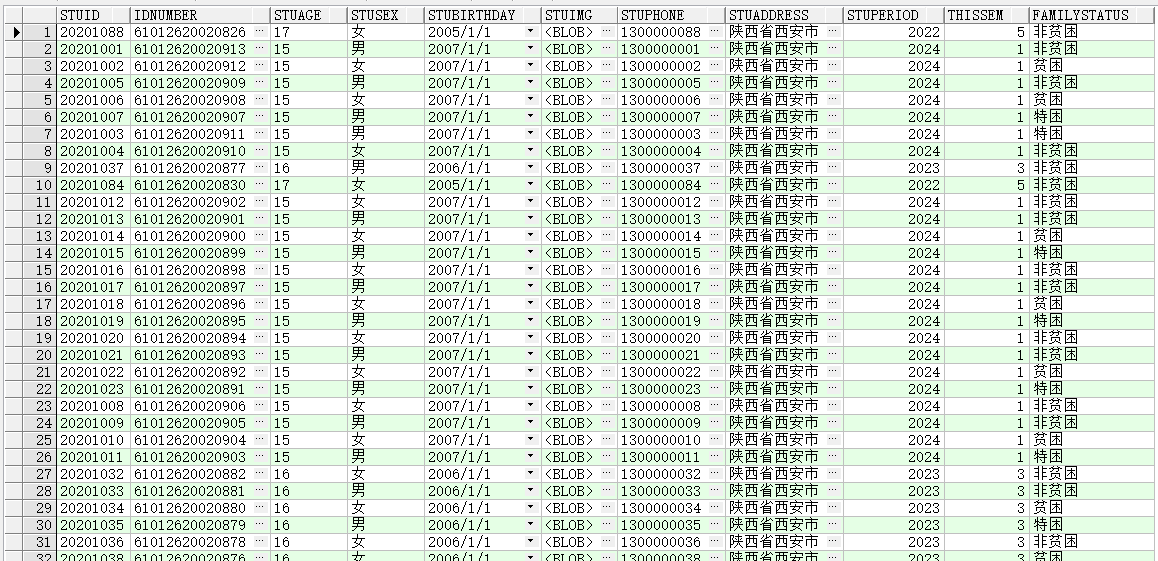


图5-5学生档案表

6.学生就读班级表（320行数据）

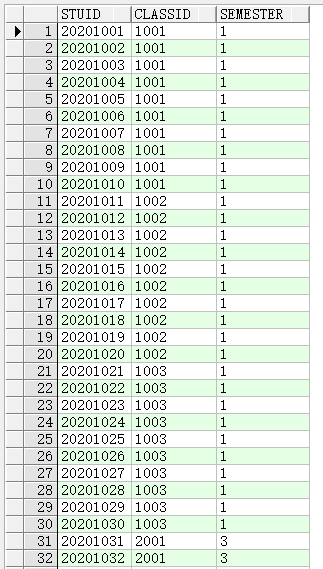


图5-6学生就读班级表

7．班级信息表（10行数据）

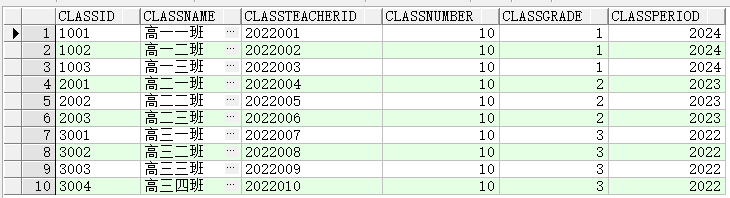


图5-7班级信息表

8.班级老师名册（90行数据）

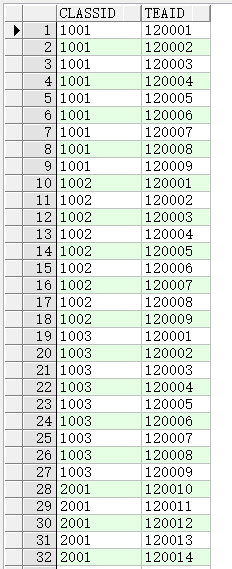


图5-8班级老师名册

9. 班级学生名册（100行数据）

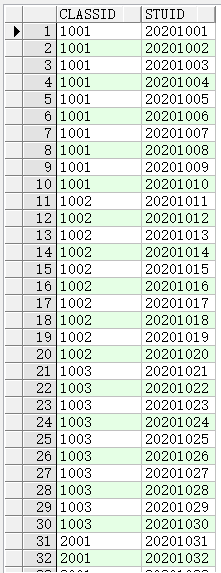


图5-9班级学生名册

10.提交成绩单（2880行数据）

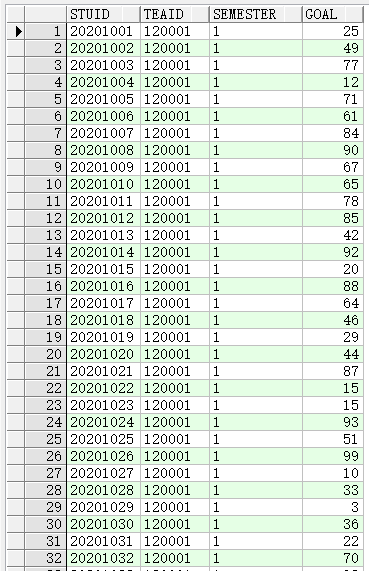


图5-10提交成绩单

11.学生成绩表（2880行数据）

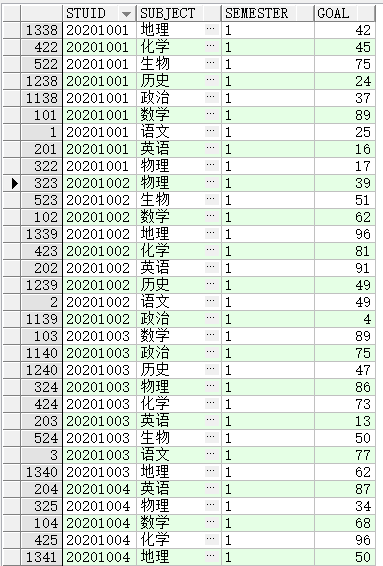


图5-11学生成绩表

12.报道单（290行数据）

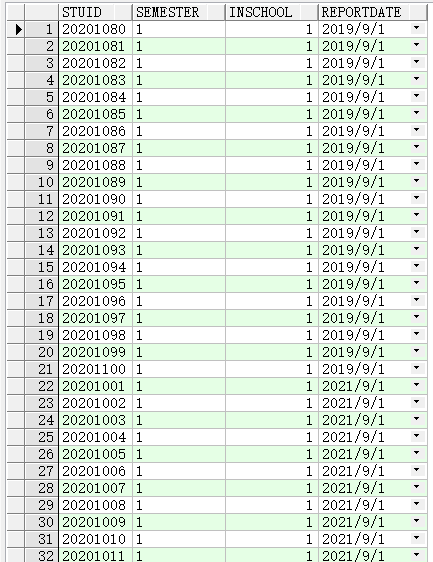


图5-12报道单

## 5.3 数据库测试

视图测试：

1.班级信息表：经测试，查询视图结果与预期相同。



图5-13班级信息表

2.个人成绩表：经测试，查询视图结果与预期相同。

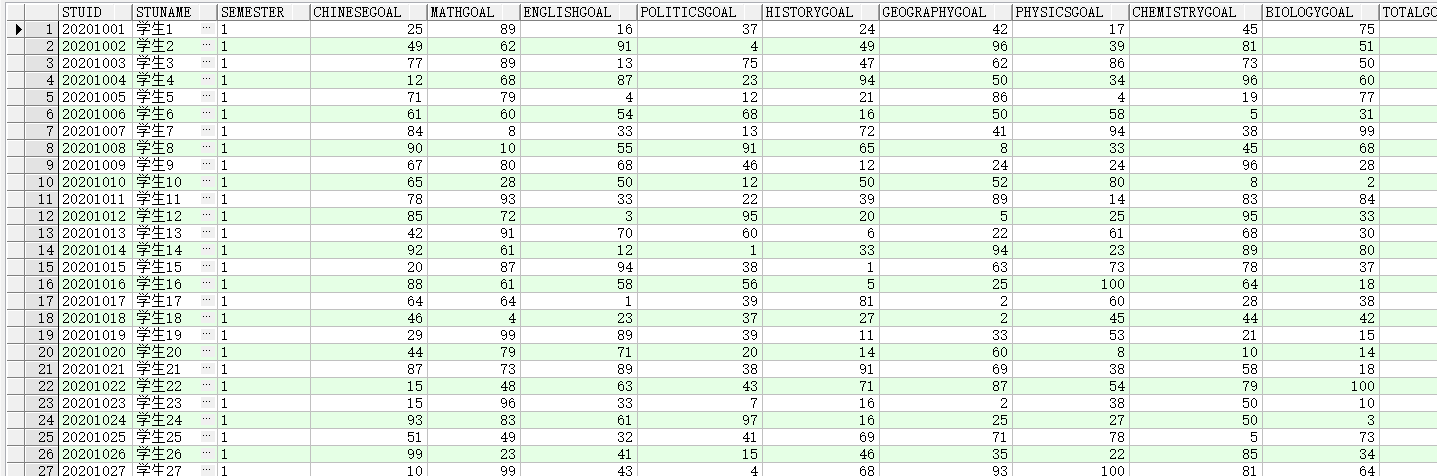


图5-14 个人成绩表

3.不及格成绩单：经测试，查询视图结果与预期相同。



图5-15 不及格成绩单

4.退学学生信息表：经测试，查询视图结果与预期相同。

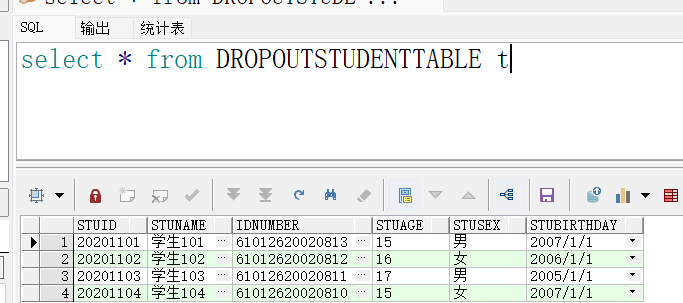


图5-16 退学学生信息表

5.贫困学生信息表：经测试，查询视图结果与预期相同。

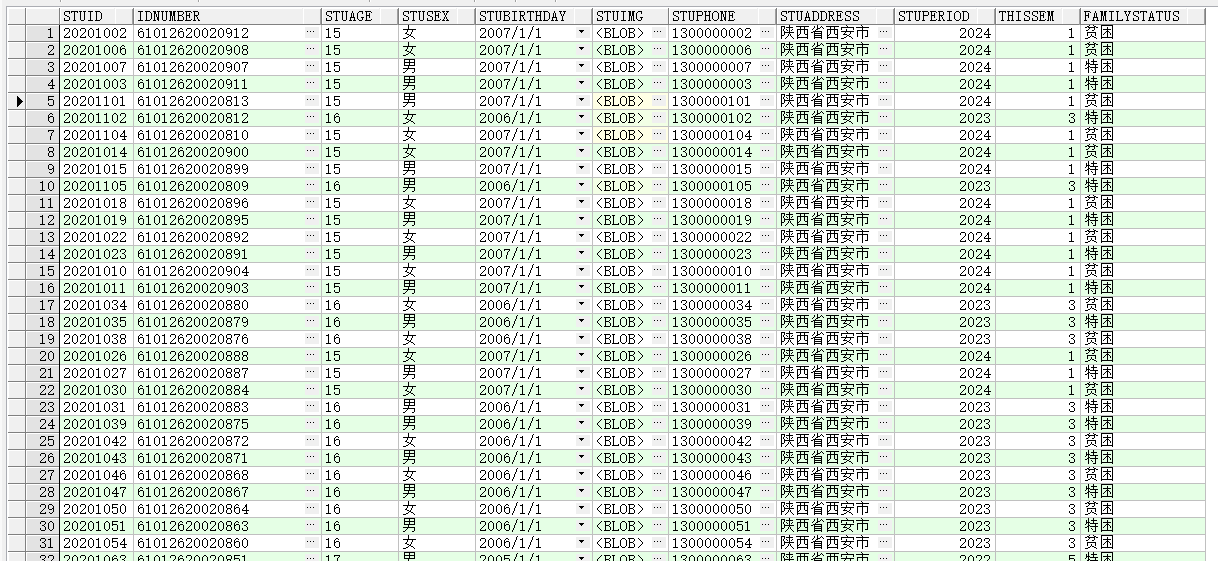


图5-17 贫困学生信息表

存储过程测试：

1.提交成绩：经测试，存储过程运行结果与预期相同。

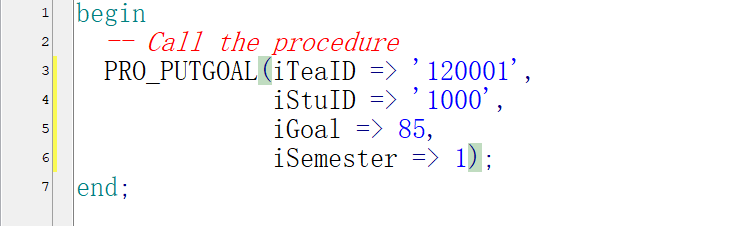


图5-18 提交成绩存储过程调用

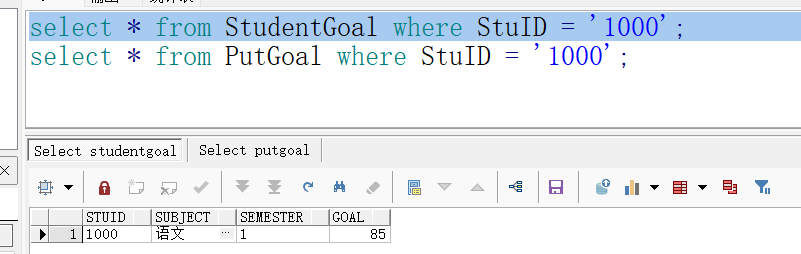


图5-19 提交成绩存储过程调用结果1

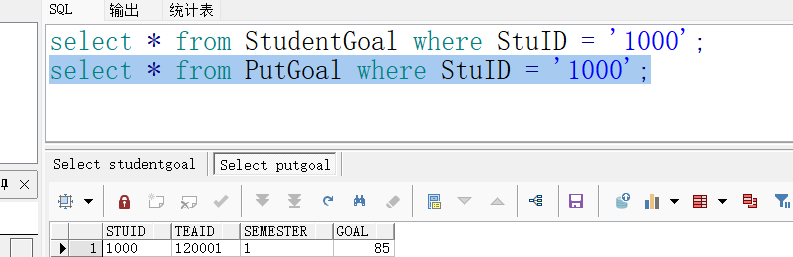


图5-20 提交成绩存储过程调用结果2

2.学生班级设置：经测试，存储过程运行结果与预期相同。

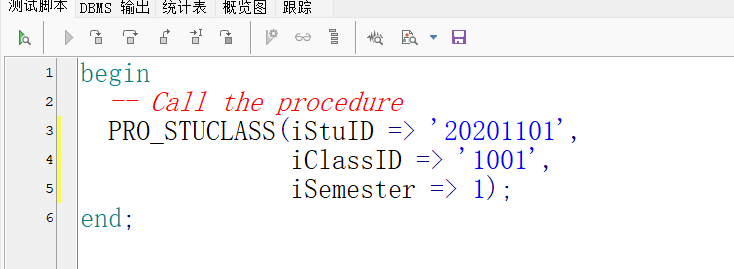


图5-21 学生班级设置存储过程调用

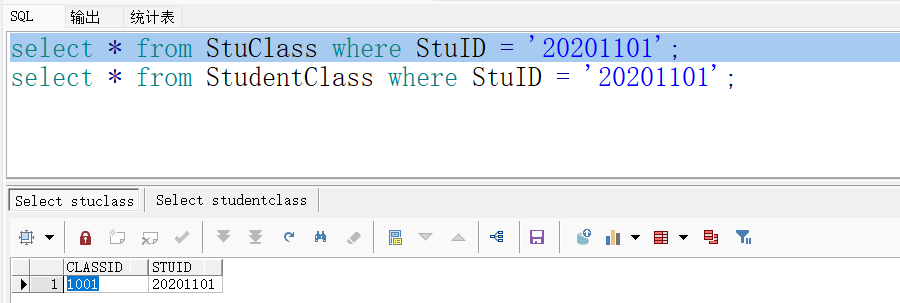


图5-22 学生班级设置存储过程调用结果1

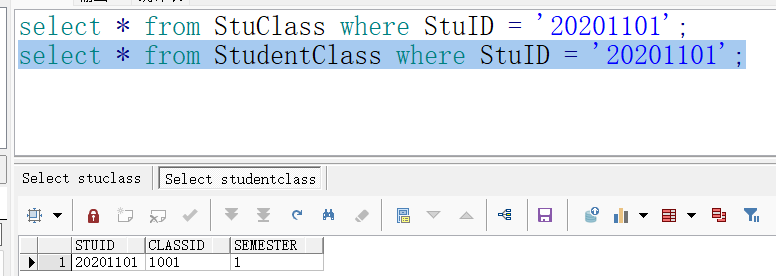


图5-23 学生班级设置存储过程调用结果2

3.报道：经测试，存储过程运行结果与预期相同。

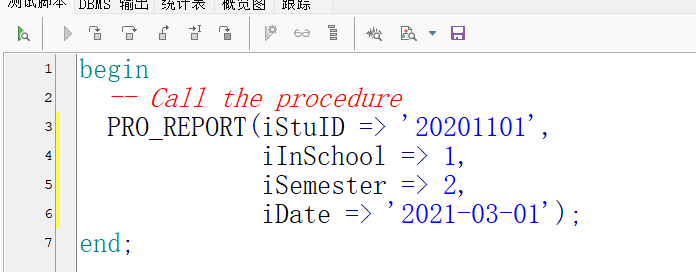


图5-24 报道存储过程调用

图5-25 报道存储过程调用结果1

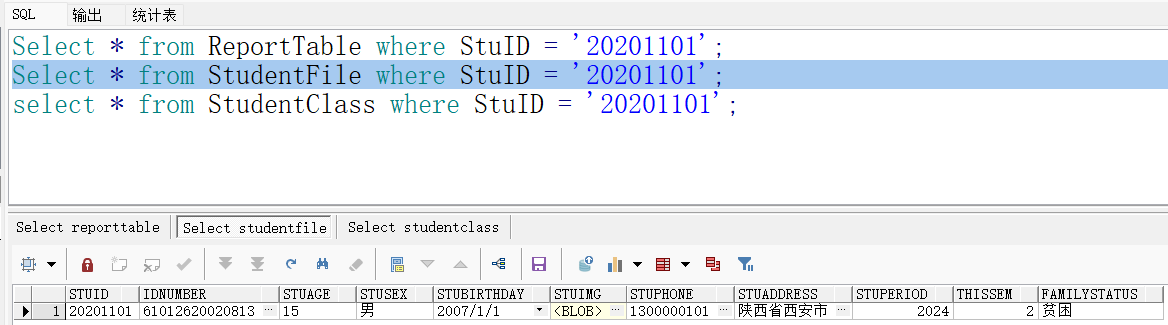
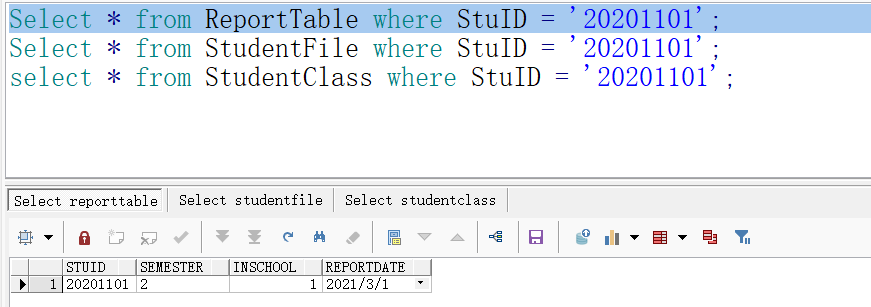


图5-26 报道存储过程调用结果2

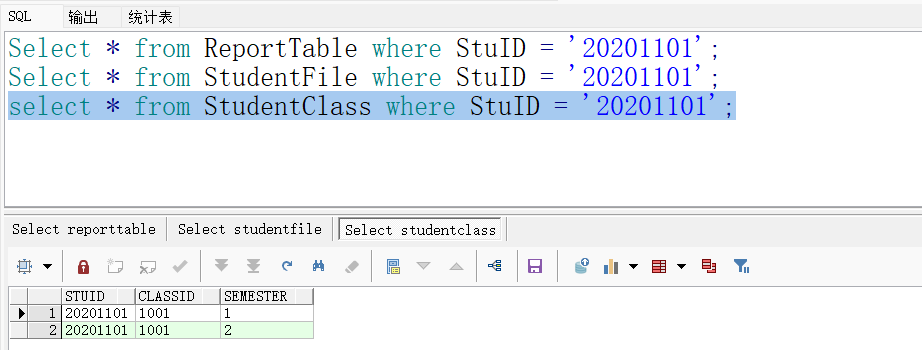


图5-27 报道存储过程调用结果3

4.学生存档：经测试，存储过程运行结果与预期相同。

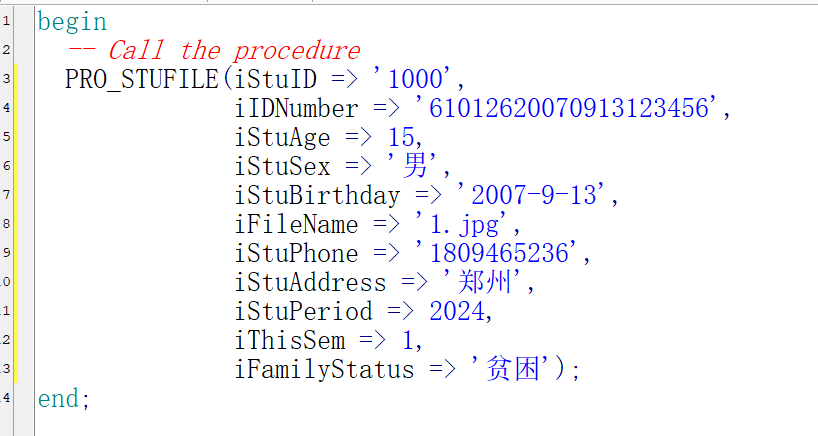


图5-28 学生存档存储过程调用

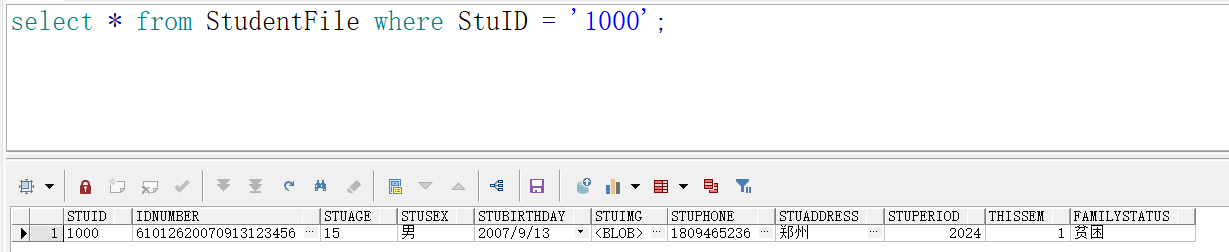


图5-29 学生存档存储过程调用结果

5.班级存档：经测试，存储过程运行结果与预期相同。

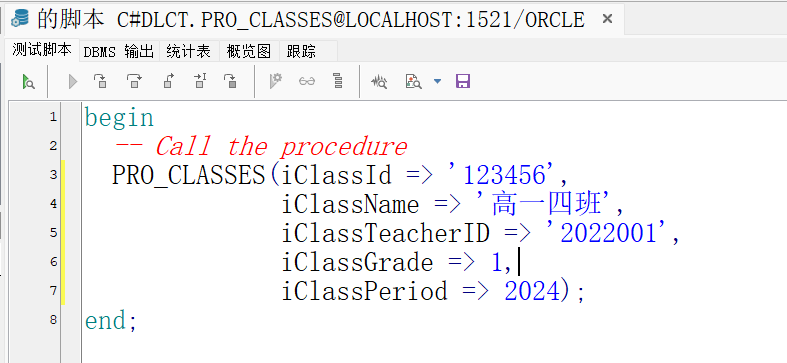


图5-30 班级存档存储过程调用

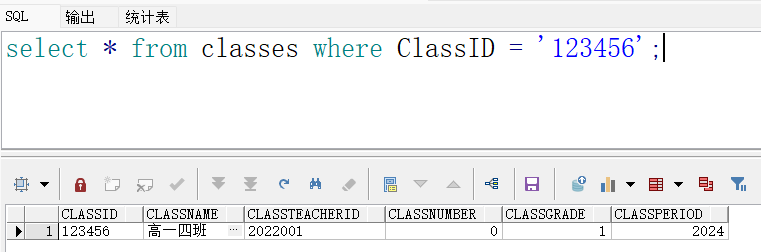


图5-31 班级存档存储过程调用结果

6.个人成绩查询：经测试，存储过程运行结果与预期相同。

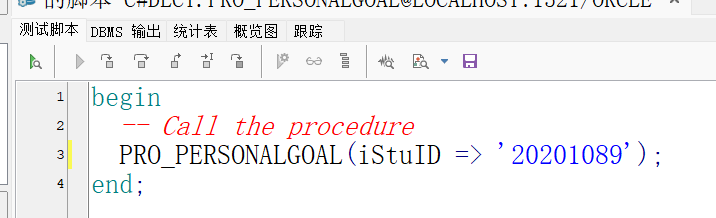


图5-32 个人成绩查询存储过程调用

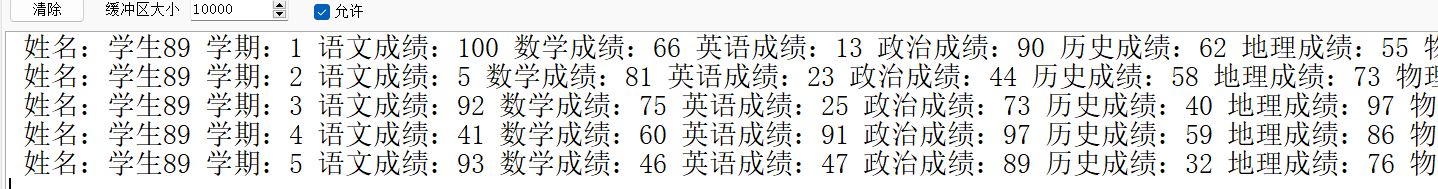


图5-33 个人成绩查询存储过程调用结果1

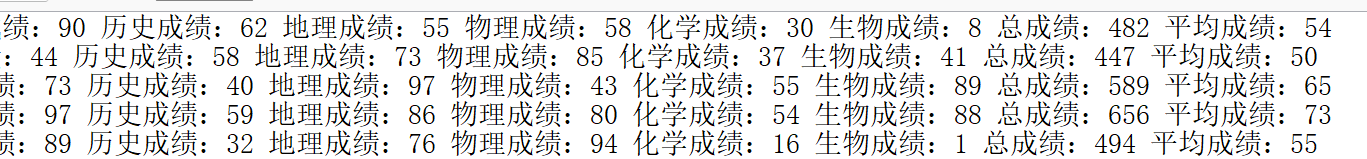


图5-34 个人成绩查询存储过程调用结果2

7.班级成绩查询：经测试，存储过程运行结果与预期相同。

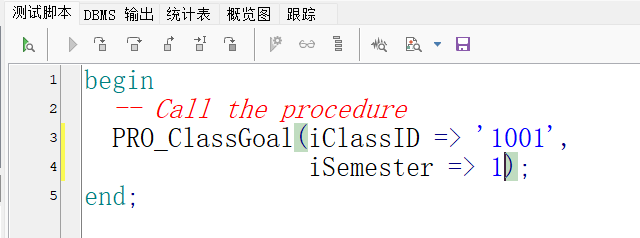


图5-35 班级成绩查询存储过程调用

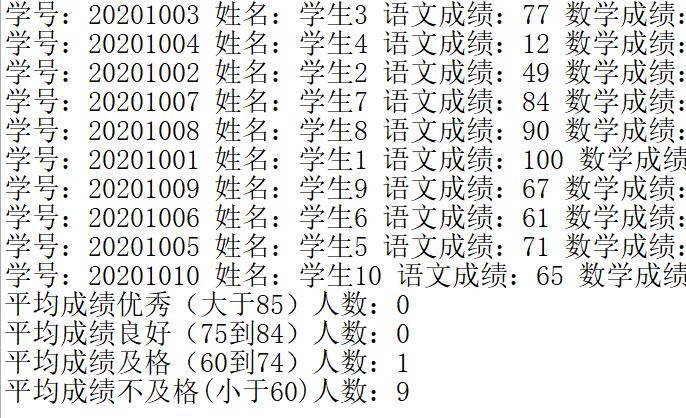


图5-36 班级成绩查询存储过程调用结果1

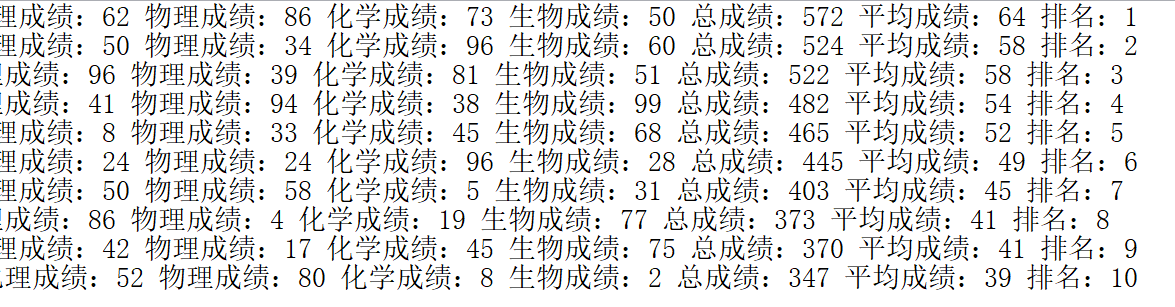


图5-37 班级成绩查询存储过程调用结果2

8.科目成绩查询：经测试，存储过程运行结果与预期相同。

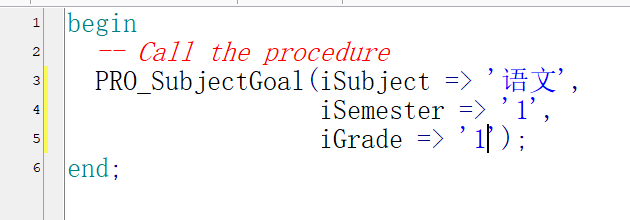


图5-38 科目成绩查询存储过程调用

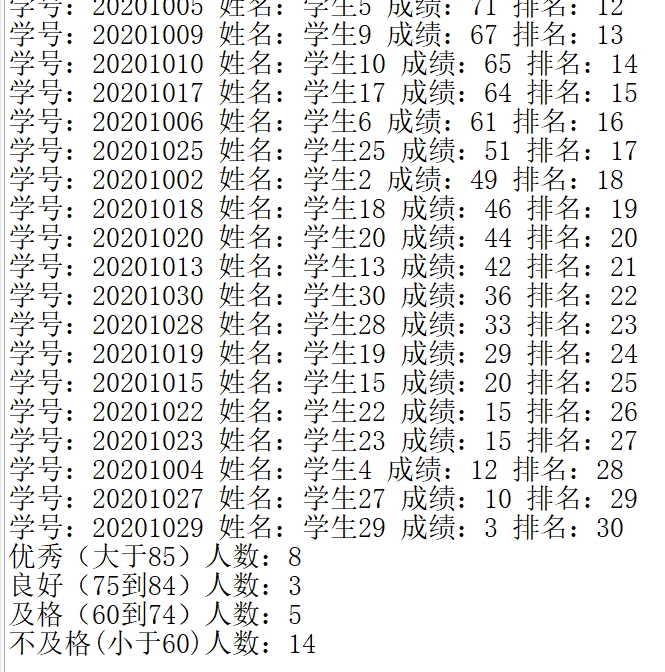


图5-39 科目成绩查询存储过程调用结果

9.修改成绩：经测试，存储过程运行结果与预期相同。

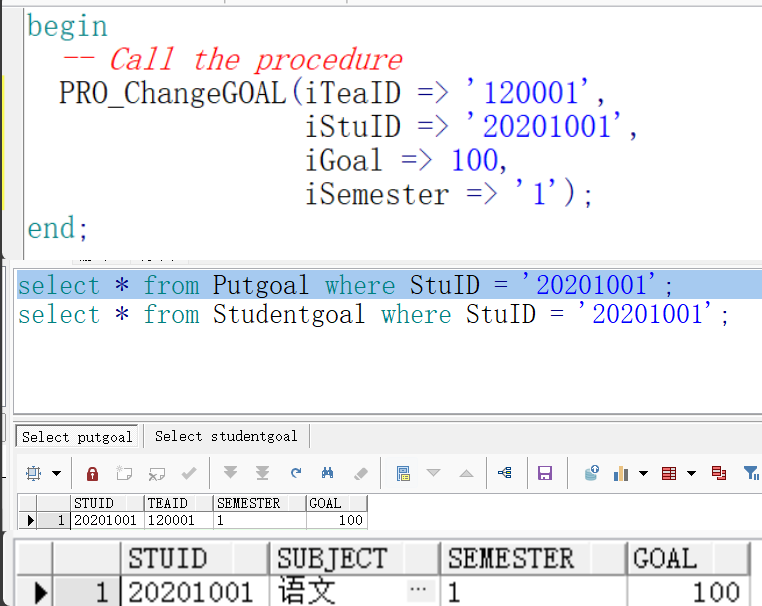


图5-40 成绩修改存储过程调用结果

10.修改学生档案：经测试，存储过程运行结果与预期相同。

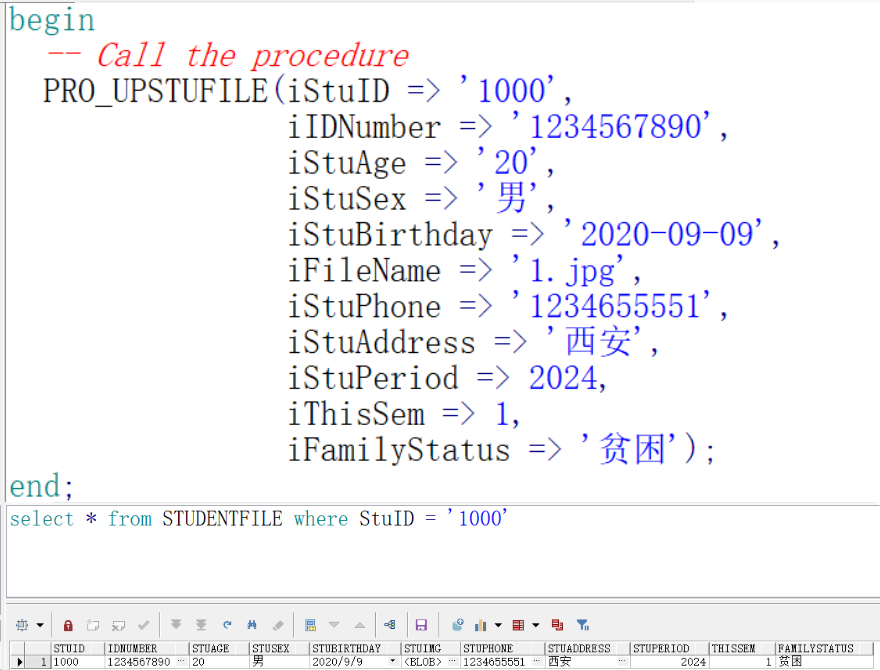


图5-41 学生档案修改存储过程调用结果

11.修改班级档案：经测试，存储过程运行结果与预期相同。

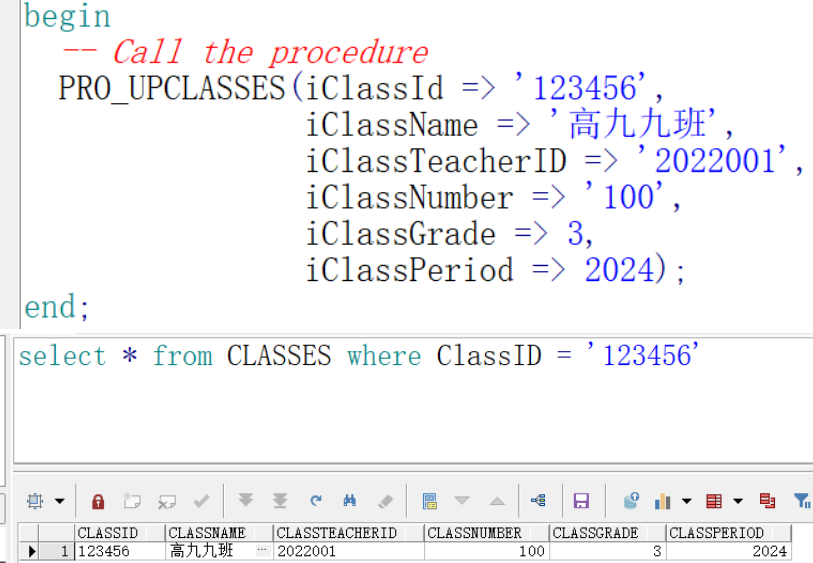


图5-42 班级档案修改存储过程调用结果

1. 年级成绩查询：经测试，存储过程运行结果与预期相同。

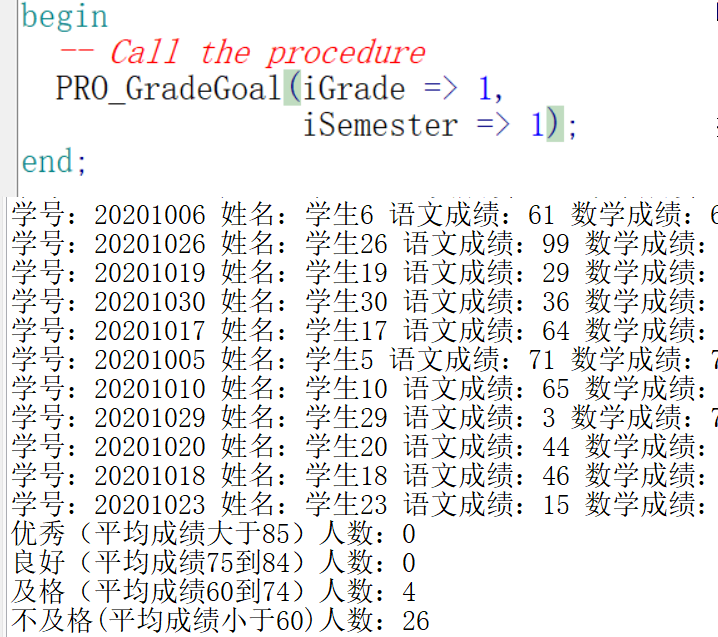


图5-43 班级档案修改存储过程调用结果

触发器测试：

1.T\_INSTUCLASS触发器：经测试，触发器运行结果与预期相同。

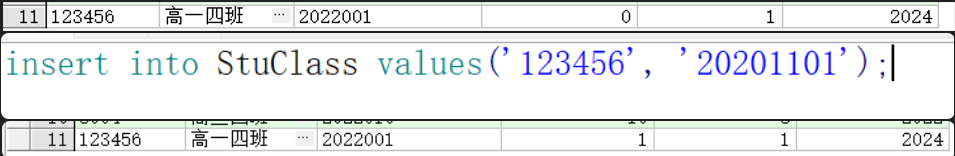


图5-44测试触发器T\_INSTUCLASS

2.T\_DESTUCLASS触发器：经测试，触发器运行结果与预期相同。

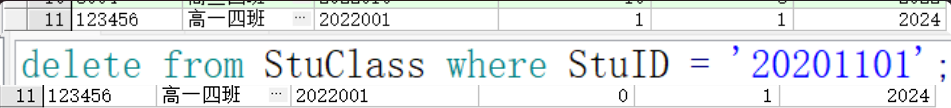


图5-45 测试触发器T\_DESTUCLASS

3. T\_UPSTUCLASS触发器：经测试，触发器运行结果与预期相同。

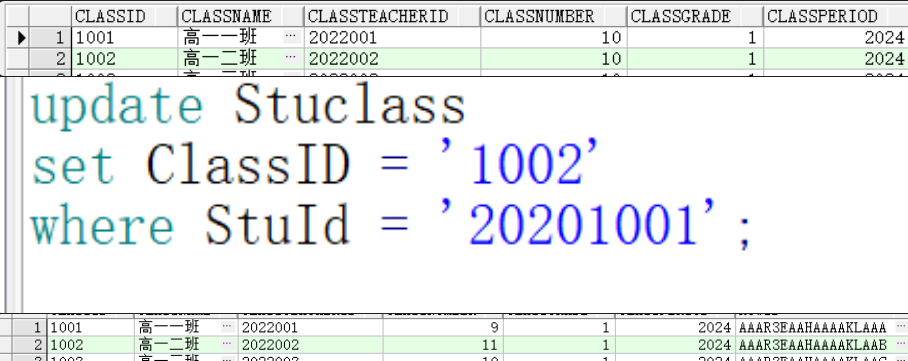


图5-46测试触发器T\_UPSTUCLASS

# 6．总结

通过这次实习，我收获了许多，认识到了不足的地方。感谢各位老师的指导，在这两周中给了我很多帮助。在这两周中我学到了许多知识。

首先是需求分析。在之前的数据库课程学习中，我所学习到的所有有关需求分析的知识都是理论上的，课程只教会我如何做需求分析，而没有教我怎么做需求分析。在这次实习中，我简单的体验了一次需求分析，体会到了需求分析是如此的困难。并且在进行需求分析时，我温习了数据流程图，并使用使用亿图图示绘制中学生档案管理系统的数据流程图，完成了数据字典设计。学会了如何分析业务，并将业务如何转换为数据流程。

其次是概念结构设计。我实践了如何从数据流程图中抽取ER图，如何将分ER图合成成全局ER图。

在逻辑结构设计阶段。如何将全局ER图转换成关系模式，如何将冗余的数据进行合并，并且使其满足3FN。

在数据库实施阶段，由于之前的工作做的比较到位，所以代码的编写还是比较简单的。

最后，感谢学院有这次机会让我们可以进行一个数据库完整设计，再一次感谢老师们在实习过程中给予的帮助。

# 7．附录

1.数据项

表6-1 数据项

| **编号** | **属性名** | **含义** | **与其他数据项的关系** | **数据类型** | **长度** | **约束条件** |
| --- | --- | --- | --- | --- | --- | --- |
| Dl-1 | StuID | 学生的学号 |  | varchar | 10 |  |
| Dl-2 | IDNumber | 学生身份证号 |  | varchar | 20 |  |
| Dl-3 | StutName | 学生的姓名 |  | varchar | 20 |  |
| Dl-4 | StuAge | 学生的年龄 |  | number | 5 | 1-100 |
| Dl-5 | StuBirthday | 学生的出生日期 |  | date |  | 1900/1/1-2100/1/1 |
| Dl-6 | StuImg | 学生的照片 |  | blob |  |  |
| Dl-7 | StuPhone | 学生的联系电话 |  | varchar | 20 |  |
| Dl-8 | StuAddress | 学生的家庭住址 |  | varchar | 40 |  |
| Dl-9 | StuPeriod | 学生届数 |  | number | 5 | 1900-2100 |
| Dl-10 | StuClass | 学生就读班级 |  | varchar | 10 |  |
| Dl-11 | StuSem | 当前学期 |  | number | 1 | {1, 2, 3, 4, 5, 6} |
| Dl-12 | FirstSem | 第一学期是否就读 |  | number | 1 | {0, 1} |
| Dl-13 | SecondSem | 第二学期是否就读 |  | number | 1 | {0, 1} |
| Dl-14 | ThirdSem | 第三学期是否就读 |  | number | 1 | {0, 1} |
| Dl-15 | FourSem | 第四学期是否就读 |  | number | 1 | {0, 1} |
| Dl-16 | FiveSem | 第五学期是否就读 |  | number | 1 | {0, 1} |
| Dl-17 | SixSem | 第六学期是否就读 |  | number | 1 | {0, 1} |
| Dl-18 | InSchool | 学生是否在读 |  | number | 1 | {0, 1} |
| Dl-19 | FamilyStatus | 学生的家庭状况 |  | varchar | 10 | {非贫困, 贫困, 特困} |
| Dl-20 | Career | 学生毕业去向 |  | varcahr | 50 |  |
| Dl-21 | TeaID | 老师的工号 |  | varchar | 10 |  |
| Dl-22 | TeaName | 老师的姓名 |  | varchar | 20 |  |
| Dl-23 | TeaSub | 老师所带科目 |  | varchar | 20 | {"语文","数学","英语","历史","政治","地理","物理","化学","生物"} |
| Dl-24 | ClassTeacherID | 班主任ID |  | varchar | 10 |  |
| Dl-25 | ClassTeacherName | 班主任姓名 |  | varchar | 20 |  |
| Dl-26 | ClassID | 班级号 |  | varchar | 10 |  |
| Dl-27 | ClassName | 班名 |  | varchar | 20 |  |
| Dl-28 | ClassTeacher | 班主任ID |  | varchar | 10 |  |
| Dl-29 | ClassNumber | 班级人数 |  | number | 5 |  |
| Dl-30 | ClassGrade | 班级的年级 |  | number | 1 | {1, 2, 3} |
| Dl-31 | ClassPeriod | 班级的届 |  | number | 5 | 1900-2100 |
| Dl-32 | ChineseTea | 班级语文老师的ID |  | varchar | 10 |  |
| Dl-33 | MathTea | 班级数学老师的ID |  | varchar | 10 |  |
| Dl-34 | EnglishTea | 班级英语老师的ID |  | varchar | 10 |  |
| Dl-35 | HistoryTea | 班级历史老师的ID |  | varchar | 10 |  |
| Dl-36 | PliticsTea | 班级政治老师的ID |  | varchar | 10 |  |
| Dl-37 | GeographyTea | 班级地理老师的ID |  | varchar | 10 |  |
| Dl-38 | PhysicsTea | 班级物理老师的ID |  | varchar | 10 |  |
| Dl-39 | ChemistryTea | 班级化学老师的ID |  | varchar | 10 |  |
| Dl-40 | BiologyTea | 班级生物老师的ID |  | varchar | 10 |  |
| Dl-41 | Subject | 科目 |  | varchar | 20 | {"语文","数学","英语","历史","政治","地理","物理","化学","生物"} |
| Dl-42 | Goal | 成绩 |  | number | 5 | 0-100 |
| Dl-43 | Semester | 学期 |  | number | 1 | {1, 2, 3, 4, 5, 6} |
| Dl-44 | ChineseGoal | 语文成绩 |  | number | 5 | 0-100 |
| Dl-45 | MathGoal | 数学成绩 |  | number | 5 | 0-100 |
| Dl-46 | EnglishGoal | 英语成绩 |  | number | 5 | 0-100 |
| Dl-47 | HistoryGoal | 历史成绩 |  | number | 5 | 0-100 |
| Dl-48 | PliticsGoal | 政治成绩 |  | number | 5 | 0-100 |
| Dl-49 | GeographyGoal | 地理成绩 |  | number | 5 | 0-100 |
| Dl-50 | PhysicsGoal | 物理成绩 |  | number | 5 | 0-100 |
| Dl-51 | ChemistryGoal | 化学成绩 |  | number | 5 | 0-100 |
| Dl-52 | BiologyGoal | 生物成绩 |  | number | 5 | 0-100 |
| Dl-53 | AverageGoal | 平均成绩 | Dl-40到Dl-48求平均值 | number | 5 | 0-100 |
| Dl-54 | TotalGoal | 总成绩 | Dl-40到Dl-48求和 | number | 5 | 0-900 |
| Dl-55 | GoalRank | 成绩排名 |  | number | 5 |  |
| Dl-56 | FinCheck | 完成报道 |  | number | 1 | {0, 1} |

2.数据结构

表6-2 数据结构

| **数据结构编号** | **数据结构名** | **数据结构含义** | **组成** |
| --- | --- | --- | --- |
| DS-1 | Student | 学生档案表 | StuID, IDNumber, StutName, StuAge, S tuBirthday, StuImg, StuPhone, StuAddress, StuPeriod, StuClass, Semester, InSchool, FirstSem, SecondSem, ThirdSem, FourSem, FiveSem, SixSem, Career |
| DS-2 | Teacher | 老师档案表 | TeaID, TeaName, TeaSub |
| DS-3 | ClassTeacher | 班主任档案表 | ClassTeacherID, ClassTeacherName, ClassID, ClassName |
| DS-4 | Class | 班级档案表 | ClassID, ClassName, ClassTeacher, ClassGrade, ClassPeriod, ChineseTea, MathTea, EnglishTea, HistoryTea, PliticsTea, GeographyTea, PhysicsTea, ChemistryTea, BiologyTea |
| DS-5 | ClassTeacher | 班级老师表 | ClassID, TeaID, Subject |
| DS-6 | AllGradeTabel | 全体学生成绩单 | StuID, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TotalGoal |
| DS-7 | PutGradeTabel | 提交成绩表 | StuID, Subject, Goal, Semster |
| DS-8 | StuClass | 学生班级表 | StuID, ClassID, Semster |
| DS-9 | ReportTable | 学生报道单 | StuID, InSchool, Semester |
| DS-10 | ClassGradeTable | 班级全部学生的成绩 | ClassName, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TaotalGoal, GoalRank |
| DS-11 | StuGradeTable | 学生单人的成绩 | StuID, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TotalGoal |
| DS-12 | SubGradeTable | 单科成绩 | StuName , Subject, Goal, GoalRank, Semester, TeaName |
| DS-13 | StuTable | 学生表 | StuID, StuName, InSchool, FinCheck, Semster |

3.数据流

表6-3 数据流

| **数据流编号** | **数据流名称** | **简述** | **数据流来源** | **数据流去向** | **数据流组成** | **数据流量** | **高峰流量** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| F1 | 学生档案登记表 | 档案管理员用于登记学生档案信息的表 | 档案管理员（E1） | 学生档案登记模块（P1.1） | Student | 200次/月 | 1000次/月 |
| F2 | 班级档案登记表 | 档案管理员用于登班级生档案信息的表 | 档案管理员（E1） | 班级档案登记模块（P1.3） | Class, ClassTea, ClassTeacher, StuClass | 10次/月 | 100次/月 |
| F3 | 修改班级登记表 | 档案管理员用于修改班级信息的表 | 档案管理员（E1） | 班级档案修改模块（P1.4） | Student | 10次/月 | 100次/月 |
| F4 | 修改学生档案表 | 档案管理员用于修改学生信息的表 | 档案管理员（E1） | 学生档案修改模块（P1.2） | Class, ClassTea, ClassTeacher, StuClass | 200次/月 | 1000次/月 |
| F5 | 班级信息查询表 | 班主任要求查询的班级信息的表 | 班级档案查询模块（P1.7） | 班主任（E2） | Class, StuClass | 10次/月 | 100次/月 |
| F6 | 班级成绩查询表 | 班主任用于查询全班成绩的表 | 成绩查询模块（P2.3） | 班主任（E2） | ClassGradeTable | 10次/月 | 200次/月 |
| F7 | 学生成绩登记表 | 老师用于登记成绩的表 | 老师（E3） | 成绩登记模块（P2.1） | PutGradeTable | 20次/月 | 50次/月 |
| F8 | 科目成绩统计表 | 老师用于查询自己课程的成绩的表 | 成绩查询模块（P2.3） | 老师（E3） | SubGradeTable | 20次/月 | 50次/月 |
| F9 | 学生成绩修改表 | 老师用于修改成绩的表 | 老师（E3） | 成绩修改模块（P2.2） | PutGradeTable | 20次/月 | 100次/月 |
| F10 | 个人成绩查询表 | 学生查询自己的成绩的表 | 成绩查询模块（P2.3） | 学生（E4） | StuGradeTable | 1000次/月 | 3000次/月 |
| F11 | 报道单 | 学生开学是上交的表单 | 学生（E4） | 入学登记（P3.1） | ReportTable | 200次/月 | 1000次/月 |
| F12 | 学生档案查询表 | 学生查询自己的档案 | 学生档案查询模块（P1.6） | 学生（E4） | Student | 20次/月 | 100次/月 |
| F13 | 班级信息存档表 | 将班级档案存入班级档案中的表 | 班级档案登记模块（P1.3） | 班级档案（S2） | Class, ClassTea, ClassTeacher, StuClass | 10次/月 | 100次/月 |
| F14 | 班级信息表 | 从班级档案中取出的表 | 班级档案（S2） | 班级档案查询模块（P1.7） | Class, ClassTea, ClassTeacher, StuClass | 10次/月 | 100次/月 |
| F15 | 修改班级档案信息表 | 用于修改班级档案的表 | 班级档案修改模块（P1.4） | 班级档案（S2） | Class, ClassTea, ClassTeacher, StuClass | 10次/月 | 100次/月 |
| F16 | 学生基本信息存档表 | 将档案管理员登记的学生档案信息写入学生档案的表 | 学生档案登记模块（P1.1） | 学生档案（S1） | Student | 200次/月 | 1000次/月 |
| F17 | 修改学生档案信息表 | 将档案管理员修改学生信息写入学生档案的表 | 学生档案修改模块（P1.2） | 学生档案（S1） | Student | 200次/月 | 1000次/月 |
| F18 | 学生档案表 | 从学生档案中取出的表 | 学生档案（S1） | 学生档案查询模块（P1.6） | Student | 20次/月 | 100次/月 |
| F19 | 学生成绩表 | 从学生成绩中取出的表 | 成绩登记模块（P2.1） | 全体学生成绩表（S3） | PutGradeTable | 20次/月 | 50次/月 |
| F20 | 学生成绩修改表 | 用于修改学生成绩的表 | 成绩修改模块（P2.2） | 全体学生成绩表（S3） | PutGradeTable | 20次/月 | 50次/月 |
| F21 | 学生成绩查询表 | 全体学生的表交予成绩查询模块进行分配 | 全体学生成绩表（S3） | 成绩查询模块（P2.3）统计不及格成绩（P2.4） | AllGradeTable | 20次/月 | 50次/月 |
| F22 | 学生就读历史表 | 将学生报道信息存入学生档案 | 入学登记模块（P3.1） | 学生档案（S1） | Student | 200次/月 | 1000次/月 |
| F23 | 学生班级表 | 学生班级对应的表 | 班级档案登记模块（P1.3） | 班级学生花名册（S4）学生档案（S1） | StuClass | 10次/月 | 100次/月 |
| F24 | 学生班级修改表 | 班级信息修改时学生班级进行修改 | 班级档案修改模块（P1.4） | 班级学生花名册（S4）学生档案（S1） | StuClass | 10次/月 | 100次/月 |
| F25 | 退学学生报道表 | 退学学生提交的报道单 | 入学登记模块（P3.1） | 退学学生统计模块（P3.2） | StuTable | 1次/月 | 10次/月 |
| F26 | 退学学生信息表 | 退学学生的信息表 | 退学学生统计模块（P3.2） | 退学登记模块（P3.3） | StuTable | 1次/月 | 10次/月 |
| F27 | 未报道学生表 | 未报道学生的学生信息表 | 入学登记模块（P3.1） | 退学登记模块（P3.2） | StuTable | 1次/月 | 10次/月 |
| F28 | 退学学生表 | 所有退学学生信息表 | 退学登记模块（P3.3） | 退学学生表（S4） | StuTable | 1次/月 | 10次/月 |
| F29 | 不及格学生成绩表 | 统计所有不及格学生的ID和不及格成绩 | 统计不及格成绩(P2.4) | 不及格成绩(S5) | PutGradeTabel | 50次/月 | 100次/月 |

4.处理逻辑

表6-4 处理逻辑

| 处理逻辑编号 | 处理逻辑名称 | 简述 | 输入的数据流 | 处理 | 输出的数据流 | 处理频率 |
| --- | --- | --- | --- | --- | --- | --- |
| P1.1 | 学生档案登记 | 登记学生档案 | F1学生档案登记表 F23学生班级表 | 将档案管理员登记的学生档案或者学生班级信息存入学生档案 | F16学生基本信息存档表 | 100次/月 |
| P1.2 | 学生档案修改 | 修改学生档案 | F4修改学生档案表 F24学生班级修改表 | 档案管理员修改学生档案，或档案管理员修改学生班级信息 | F17修改学生档案信息表 | 10次/月 |
| P1.3 | 班级档案登记 | 登记班级档案 | F2班级档案登记表 | 档案管理员将班级档案存入班级档案，并将学生班级表保存到S4 | F13班级信息存档表 F23学生班级表 | 20次/月 |
| P1.4 | 班级档案修改 | 修改班级档案 | F3 修改班级登记表 | 档案管理员修改学生档案，将班级信息存入班级档案，将学生班级信息保存到S4 | F15修改班级档案信息表 F24学生班级修改表 | 5次/月 |
| P1.5 | 班级档案查询 | 查询班级档案 | F14班级信息表 | 将选择的班级档案从班级档案从取出，供班主任查看 | F5班级信息查询表 | 20次/月 |
| P1.6 | 学生档案查询 | 查询学生档案 | F18学生档案表 | 将选择的学生档案从学生档案从取出，供班主任查看 | F12学生档案查询表 | 200次/月 |
| P2.1 | 成绩登记 | 老师登记成绩 | F7学生成绩登记表 | 将老师登记的成绩测存入全体学生成绩表 | F19学生成绩表 | 200次/月 |
| P2.2 | 成绩修改 | 老师修改成绩 | F9学生成绩修改表 | 将老师修改的成绩存入全体学生成绩表 | F20学生成绩修改表 | 20次/月 |
| P2.3 | 成绩查询 | 查询成绩 | F21学生成绩查询表 | 将学生成绩根据不同的查询人制作不同的视图 | F6班级成绩查询表 F8科目成绩统计表 F10个人成绩查询表 | 1000次/月 |
| P2.4 | 统计不及格成绩 | 统计不及格的成绩 | F21学生成绩查询表 | 将不及格的成绩从全体成绩中提取出来 | F29不及格成绩学生成绩表 | 300次/月 |
| P3.1 | 入学登记 | 学生入学登记 | F11报道单 | 将学生分为退学、未报到、正常三张表 | F22学生就读历史表 F25退学学生报道表 F27未报道学生表 | 300次/月 |
| P3.2 | 退学学生统计 | 统计退学学生 | F25退学学生报道表 F27未报道学生表 | 统计所有退学学生 | F26退学学生信息表 | 10次/月 |
| P3.3 | 退学登记 | 学生退学处理 | F26退学学生信息表 | 修改所有退学学生的档案 | F28退学学生表 | 10次/月 |

5.数据存储

表6-5 数据存储

| **编号** | **数据存储名称** | **简述** | **数据存储组成** | **相关联的处理** |
| --- | --- | --- | --- | --- |
| S1 | 学生档案 | 存储所有学生的档案信息 | Student, StuClass | P1.1 P1.2 1.3 1.4 P1.6 P3.1 |
| S2 | 班级档案 | 存储所有班级的档案信息 | Class | P1.3 P1.4 P1.5 |
| S3 | 全体学生成绩表 | 存放所有学生的成绩 | AllGradeTable | P2.1 P2.2 P2.3 |
| S4 | 班级学生花名册 | 存放班级学生名单 | StuClass | P1.3 P1.4 |
| S5 | 不及格成绩 | 存放所有不及格成绩 | PutGradeTable | P2.4 |
| S6 | 退学学生表 | 存放所有退学学生的信息 | StuTable | P3.3 |