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： | 2020014322 |
| 专业班级： | 软件2001 |
| 姓 名： | 王泽玺 |
| 指导老师： | 杨靖 |
| 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：

关键字：数据库设计；中学生档案；Oracle 11g；

**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 12](#_Toc80864212)

[4．数据库物理设计 14](#_Toc80864213)

[5．数据库实施与测试 15](#_Toc80864214)

[5.1 数据库及数据库对象建立 15](#_Toc80864215)

[5.1.1创建基本表 15](#_Toc80864216)

[5.1.2创建视图 22](#_Toc80864217)

[5.1.3创建索引 24](#_Toc80864218)

[5.1.4创建存储过程 24](#_Toc80864219)

[5.1.5创建触发器 38](#_Toc80864220)

[5.2 数据入库 40](#_Toc80864221)

[5.3 数据库测试 49](#_Toc80864222)

[6．总结 55](#_Toc80864223)

[7．附录 56](#_Toc80864224)

在如今，个人电脑几乎是人手必备的电子产品，其销量也呈逐年上升的趋势。在这种背景下，传统的销售管理模式很难满足大型个人电脑销售商城的管理业务。因此，对于每个商城来说，一个功能齐全、使用简单的数据库系统几乎是必不可少的。这样的个人电脑销售管理系统能够大大便利商城的管理和工作效率，降低运营成本，提高经济效益。另外，该系统应当全面的涵盖个人电脑销售的方方面面，能够让用户进行分级、分类管理，使商城的物资管理层次分明，为采购和销售提供依据，库存检查功能可自动生成缺货信息，全面的售后流程能够满足客户的不同需求并进行处理。这些业务涉及电脑销售的各个环节，从而对商城销售管理进行全面的控制和管理，降低运行成本，增强商城的市场竞争力。

# 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. 班主任对自己的所带班级档案进行查询
6. 成绩管理：
7. 老师对成绩进行登记修改
8. 老师查询成绩时，生成仅有该老师所带科目的成绩表
9. 班主任查询成绩时，生成全班的成绩表
10. 学生查询成绩时，生成仅有自己成绩的成绩表
11. 入学管理：
12. 学生提交报告单
13. 根据学生报告单将学生分为，正常报道、退学、未报道三种
14. 修改正常报道学生的学生档案
15. 统计未报道学生和退学学生名单，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

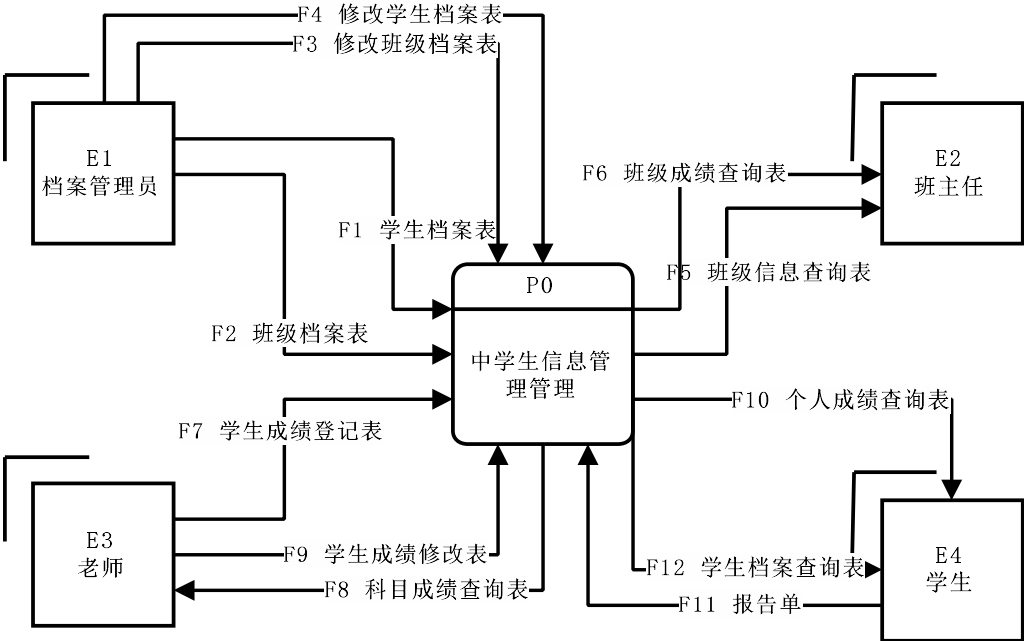


图 1-1 顶层数据流

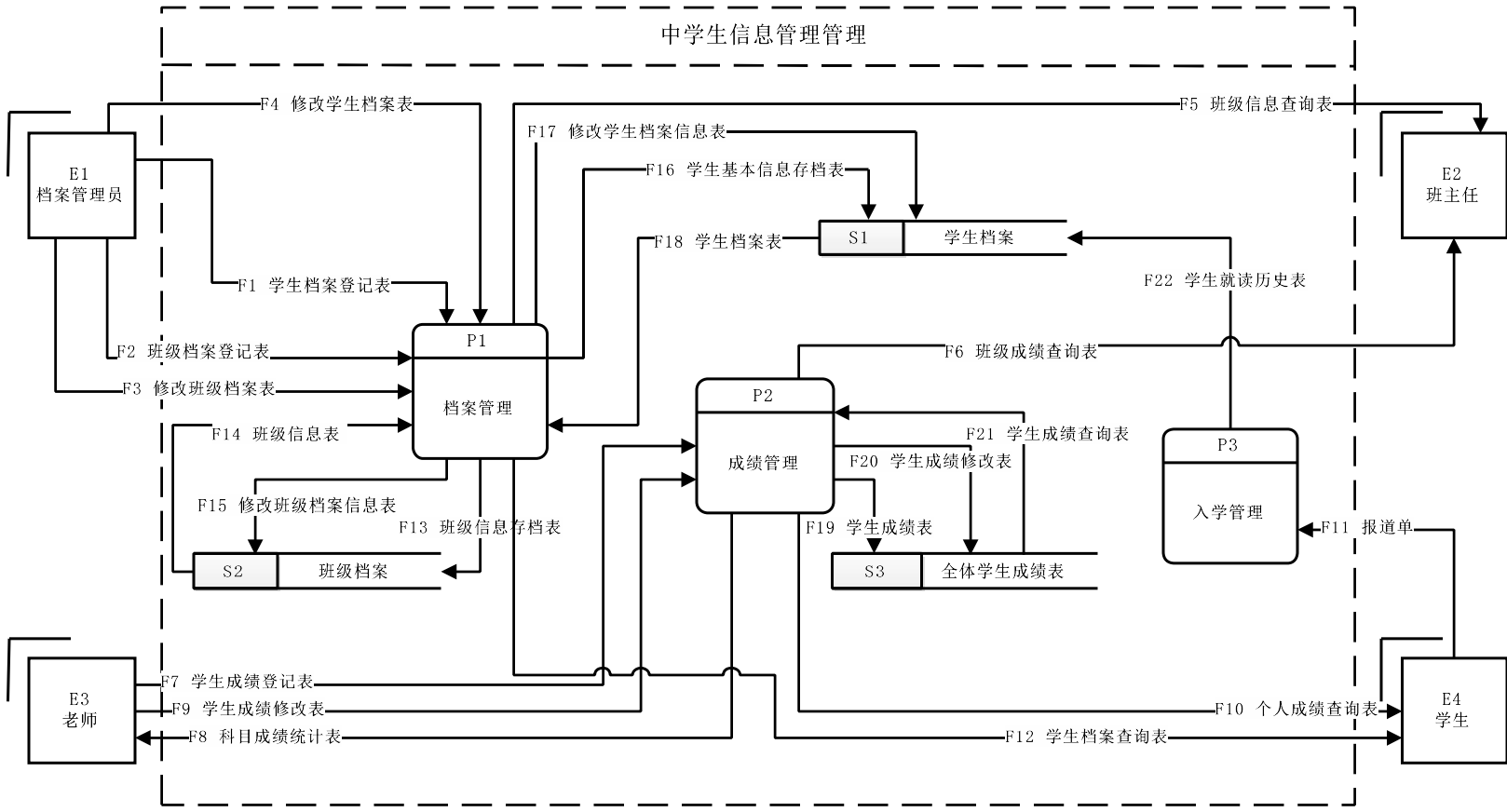


图 1-2 第一层数据流程图

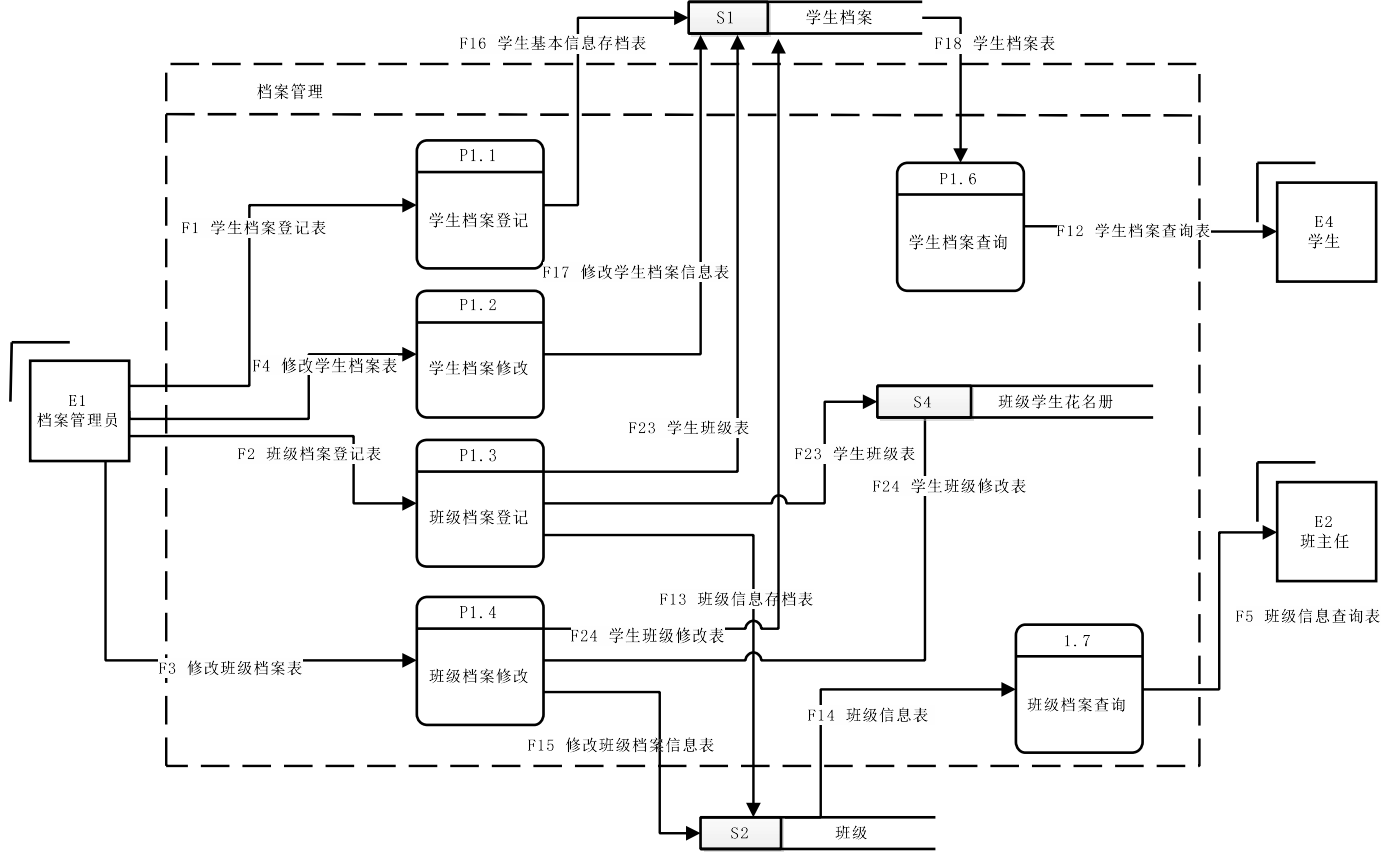


图 1-3 第二层数据流程图——档案管理

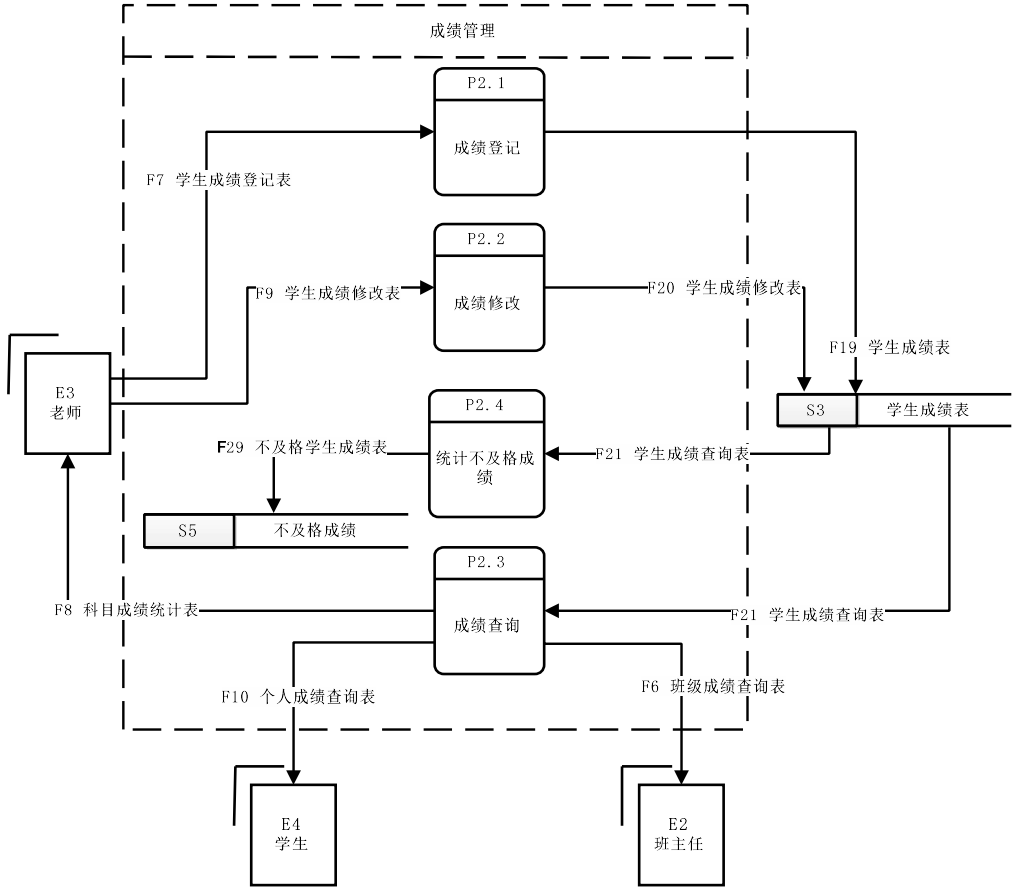


图 1-4 第二层数据流程图——成绩管理

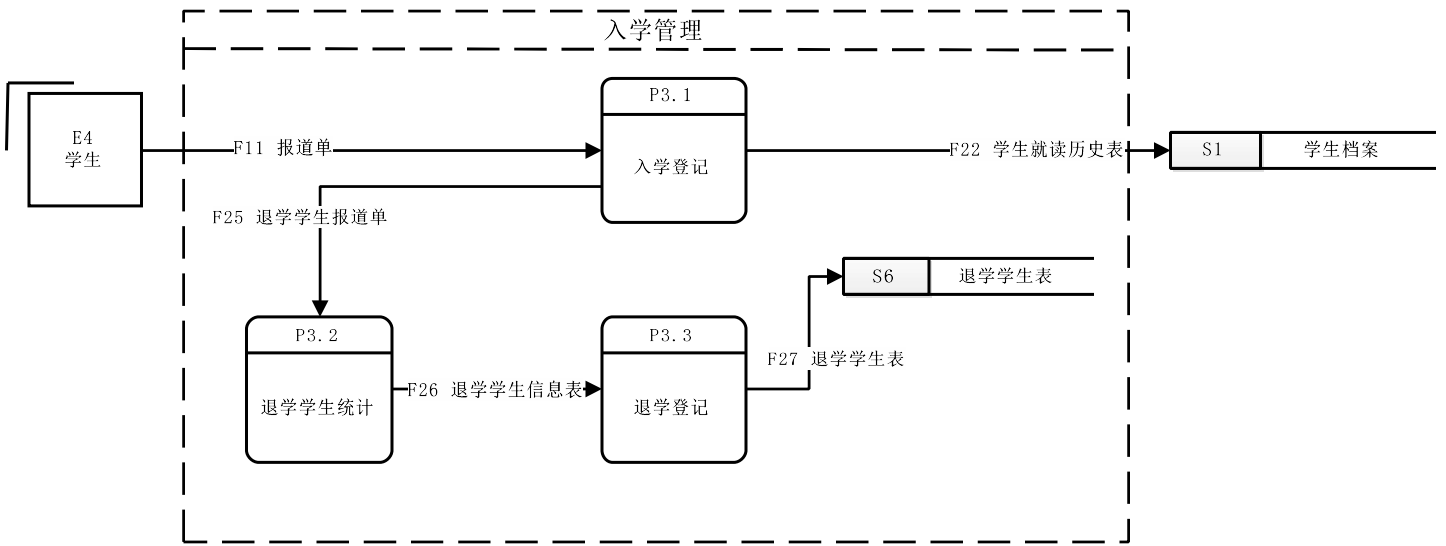


图 1-5 第二层数据流程图——入学管理

## 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41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

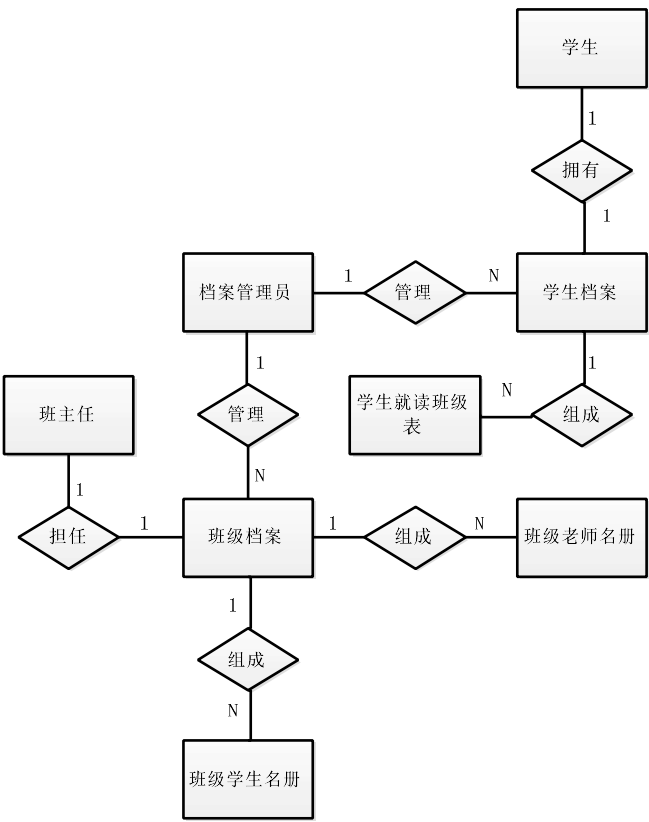


图 2-1 分E-R图—档案管理

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）

1. 成绩管理E-R图

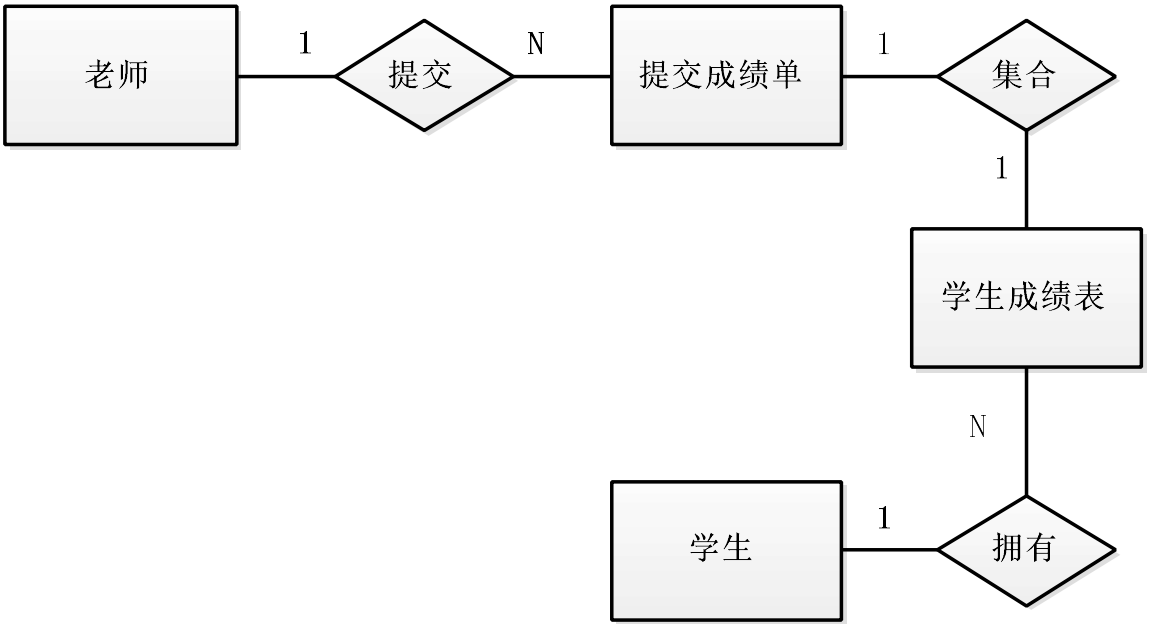


图 2-2 分E-R图—成绩管理

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）

1. 入学管理E-R图

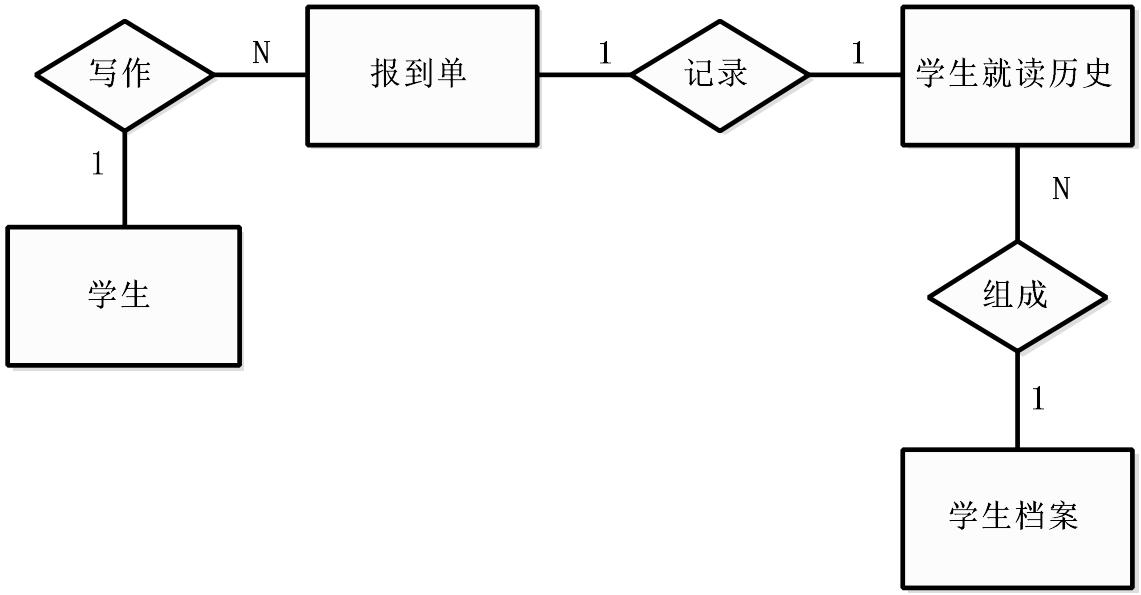


图 2-3 分E-R图——入学管理

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）

退学学生表（学生ID）

学生就读历史（学生ID，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。

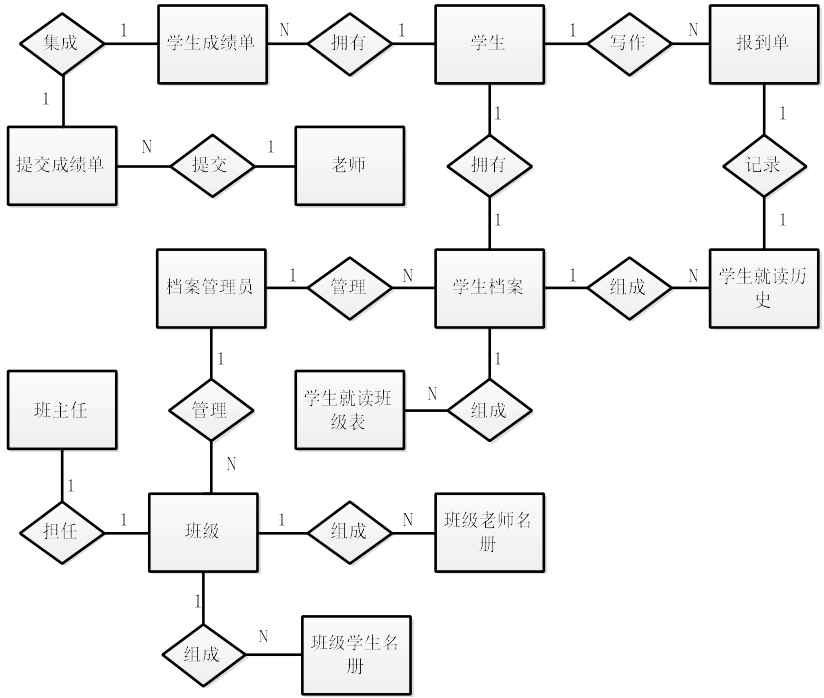


图2-4 全局E-R图

# 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）

退学学生表（学生ID）

学生就读历史（学生ID，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，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，班级ID，学期）

主属性为学生ID，班级ID，学期。

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

1. 学生就读历史（学生ID，学期，报道日期）

主属性为学生ID，学期，非主属性为报道日期。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

视图的属性列表为：

班级信息表（班级ID，班名，班主任姓名，班级人数，班级年级，班级届，语文老师姓名，数学老师姓名，英语老师姓名，政治老师姓名，历史老师姓名，地理老师姓名，物理老师姓名，化学老师姓名，生物老师姓名）

个人成绩单（学生ID，学期，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）

不及格成绩单（学生ID，学生姓名，科目，学期，成绩）

退学学生信息表（学生ID，学生姓名，身份证号，年龄，出生日期，照片，联系电话，家庭住址，届，家庭状况，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 |  |  |
| 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 |  |  |
| TeaID | 老师ID | varchar | 10 | 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 |  |  |
| 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 |  |  |
| 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 |

表3-13 学生就读历史

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |  |  |
| Semester | 学期 | varchar | 1 | 是 |  | {1, 2, 3, 4, 5, 6} |
| 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'))

)

13.创建学生就读历史

create table StudentHisory

(

StuID varchar2(10),

Semester varchar2(1),

ReportDate date,

primary key (StuID, Semester),

foreign key (StuID) references Student(StuID),

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

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 | 商品列表  客户编号 | 根据客户需求进行出库，做销售记录 |
| 出库 | PRO\_DELIVERY | 出库信息 | 查询库存，填写出库表，修改库存 |
| 入库 | PRO\_WARE | 入库信息 | 添加入库信息，修改库存信息 |
| 售后处理 | PRO\_SERVICE | 售后申请信息 | 查询销售记录，进行售后处理 |
| 退货 | PRO\_REFUND | 退货信息 | 查询售后信息，作退货记录并入库 |
| 换货 | PRO\_CHANGE | 换货信息 | 查询售后信息，作换货记录，进行返厂出库 |
| 维修 | PRO\_REPAIR | 维修信息 | 查询售后信息，作维修记录，记录修理费 |
| 采购 | PRO\_RESERVE | 采购商品列表 | 查询商品信息，添加采购记录 |
| 进货 | PRO\_PURCHASE | 进货商品列表 | 填写进货记录和进货细则，进行入库 |
| 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 |
| 科目成绩查询 | PRO\_SUBGOAL | 科目及学期 | 查询单科成绩，并进行排名 |

1.创建销售存储过程

create or replace procedure PRO\_SALE(

cno in customer.cusno%type,

goodsnum in number,

goodsno varchar2) is

isenough boolean:=true;

gno varchar2(20);

gnum number;

cuslevel customer.cuslevel%type;

total sales.saletotal%type:=0;

discount sales.salediscount%type;

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

begin

PRO\_DELIVERY(goodsnum,goodsno,'销售',isenough,total);

--进行出库操作，返回库存是否足够

if isenough=true then--库存充足，添加销售记录及销售细则信息

--获取客户等级以及折扣

select cuslevel into cuslevel from customer

where cno=customer.cusno;

if cuslevel='钻石' then discount:=0.97;

elsif cuslevel='金卡' then discount:=0.98;

elsif cuslevel='银卡' then discount:=0.99;

else discount:=1;

end if;

--添加销售单信息

dbms\_output.put\_line(

'商品总价：'||to\_char(total\*discount));

insert into sales

values(concat('SA',lpad(to\_char(NO\_SALES.NEXTVAL),4,'0')),

cno,total\*discount,total,discount,pdate);

commit;--\*\*\*重要

--添加销售细则

for i in 1..goodsnum loop

gno:=substr(goodsno,9\*i-8,6);

gnum:=to\_number(substr(goodsno,9\*i-1,1));

insert into salerecord

values(concat('SR',lpad(to\_char(NO\_SALERECORD.NEXTVAL),

4,'0')), concat('SA',lpad(to\_char(NO\_SALES.CURRVAL),

4,'0')),gno,gnum);

end loop;

dbms\_output.put\_line('您的商品已出库，谢谢惠顾');

end if;

commit;

end PRO\_SALE;

2.创建售后服务存储过程

create or replace procedure PRO\_SERVICE(

srno in salerecord.salerecordno%type,--销售细则号

sertype in service.sertype%type,--售后类型

reason in service.serreason%type,--售后申请原因

progress in service.serprogress%type,--'检查通过'或'检查未通过'

message in service.sermessage%type) is

money float;

serno service.serno%type:=concat(

'SE',lpad(to\_char(NO\_SERVICE.NEXTVAL),4,'0'));

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),

'YYYY/MM/DD');

bdate date;

dates number;

begin

--获取销售的天数，用于判断是否是否在售后时间范围内：

--7天无理由退货，15天无理由换货，2年售后保修

select sales.saletime into bdate from

salerecord left join sales on sales.saleno=salerecord.saleno

where salerecord.salerecordno=srno;

dates:=pdate-bdate;

--商品检查未通过

if progress='检查未通过' then

insert into service values(

serno,srno,sertype,reason,pdate,'审核未通过',message);

dbms\_output.put\_line('您的售后申请未通过审核，抱歉');

--退货处理

elsif sertype='退货' then

--超过7天

if dates>7 then

insert into service values(

serno,srno,sertype,reason,pdate,

'审核未未通过','电脑超过质保期，不享受退货服务');

dbms\_output.put\_line('您的电脑超过质保期，

不再享受退货服务，请谅解');

--可以退货

else

insert into service values(

serno,srno,sertype,reason,pdate,

'审核通过','退货成功，已退还购买费用');

commit;--\*\*

--进行退货

PRO\_REFUND(serno,srno);

--获取退款金额

select sales.salediscount\*goods.goodsprice into money

from salerecord left join sales

on sales.saleno=salerecord.saleno

left join goods

on goods.goodsno=salerecord.goodsno

where salerecord.salerecordno=srno;

dbms\_output.put\_line('您的电脑已退货，退款金额：'||

to\_char(money)||'元');

end if;

--换货处理

elsif sertype='换货' then

if dates>15 then

insert into service values(serno,srno,sertype,reason,pdate,

'审核未未通过','电脑超过质保期，不享受换货服务');

dbms\_output.put\_line(

'您的电脑超过质保期，不再享受换货服务，请谅解');

--可以换货

else

insert into service values(

serno,srno,sertype,reason,pdate,

'审核通过','换货成功，已更换新机');

commit;--\*\*

--进行换货

PRO\_CHANGE(serno,srno);

dbms\_output.put\_line('您的电脑已成功更换');

end if;

--维修处理

else

if dates>730 then

insert into service values(

serno,srno,sertype,reason,pdate,

'审核未未通过','电脑超过质保期，不享受免费维修服务');

dbms\_output.put\_line(

'您的电脑超过质保期，不再享受免费维修服务，请谅解');

--可以维修

else

insert into service values(

serno,srno,sertype,reason,pdate,

'审核通过','维修完毕，电脑可正常使用');

commit;--\*\*

--进行维修

PRO\_REPAIR(serno);

dbms\_output.put\_line('您的电脑已维修完毕');

end if;

end if;

dbms\_output.put\_line('本次售后结束，感谢光临！');

commit;

end PRO\_SERVICE;

3.创建退货存储过程

create or replace procedure PRO\_REFUND(

serno in service.serno%type,--售后单号

srno in salerecord.salerecordno%type) is--销售细则号

money float;

refundno refunds.refundno%type:=concat(

'RD',lpad(to\_char(NO\_REFUNDS.NEXTVAL),4,'0'));

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),

'YYYY/MM/DD');

begin

--获取退款金额

select sales.salediscount\*goods.goodsprice into money

from salerecord left join sales

on sales.saleno=salerecord.saleno

left join goods on goods.goodsno=salerecord.goodsno

where salerecord.salerecordno=srno;

insert into refunds values(refundno,serno,pdate,money,'退货成功');

commit;

PRO\_WAREHOUSING(refundno,'退货');

end PRO\_REFUND;

4.创建换货存储过程

create or replace procedure PRO\_CHANGE(

serno in service.serno%type,--售后单号

srno in salerecord.salerecordno%type) is--销售细则号

gno varchar2(20);

changeno changes.changeno%type:=concat(

'CH',lpad(to\_char(NO\_CHANGES.NEXTVAL),4,'0'));

returnno returns.returnno%type:=concat(

'RN',lpad(to\_char(NO\_RETURNS.NEXTVAL),4,'0'));

wareno warehousing.wareno%type:=concat(

'WH',lpad(to\_char(NO\_WAREHOUSING.NEXTVAL),4,'0'));

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),

'YYYY/MM/DD');

begin

select goodsno into gno from salerecord

where salerecord.salerecordno=srno;

insert into changes values(changeno,serno,pdate,'换货成功');

insert into returns values(returnno,serno,pdate);

insert into warehousing values(wareno,gno,1,'返厂',pdate);

commit;

end PRO\_CHANGE;

5.创建售后维修存储过程

create or replace procedure PRO\_REPAIR(

serno in service.serno%type) is--售后单号

repairno repairs.repairno%type:=concat(

'RP',lpad(to\_char(NO\_REPAIR.NEXTVAL),4,'0'));

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

begin

--模拟真实情况，随即生成维修费用

insert into repairs values(

repairno,serno,pdate,DBMS\_RANDOM.VALUE()\*100,'维修完毕');

commit;

end PRO\_REPAIR;

6.创建添加采购单存储过程

create or replace procedure PRO\_RESERVE(

goodsnum in number,

goodsno varchar2) is

gno varchar2(20);

gnum number;

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

begin

for i in 1..goodsnum loop

gno:=substr(goodsno,9\*i-8,6);

gnum:=to\_number(substr(goodsno,9\*i-1,1));

insert into reserve--添加采购记录

values(concat('RV',lpad(to\_char(NO\_RESERVE.NEXTVAL),4,'0')),gno,gnum,pdate);

end loop;

commit;

end PRO\_RESERVE;

7.创建进货存储过程

create or replace procedure PRO\_PURCHASE(

goodsnum in number,

goodsno varchar2) is

gno varchar2(20);

gnum number;

purno purchase.purchaseno%type;

sumnum number:=0;

total float:=0;

temp float;

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

begin

--增加进货记录

purno:='PU'||lpad(to\_char(NO\_PURCHASE.NEXTVAL),4,'0');

insert into purchase values(purno,0,0,pdate);

commit;--\*\*\*重要

--添加进货细则并统计

for i in 1..goodsnum loop

gno:=substr(goodsno,9\*i-8,6);

gnum:=to\_number(substr(goodsno,9\*i-1,1));

select goodsenterprice into temp from goods

where goods.goodsno=gno;

sumnum:=sumnum+gnum;

total:=total+temp\*gnum;

insert into purrecord values(

'PR'||lpad(to\_char(NO\_WAREHOUSING.NEXTVAL),4,'0'),

purno,gno,gnum);

end loop;

update purchase set purchase.purchaseamount=total,

purchasenum=sumnum

where purchase.purchaseno=purno;

PRO\_WAREHOUSING(purno,'进货');

dbms\_output.put\_line('进货记录成功，进货数量：'

||to\_char(sumnum)||'，货物总价值：'||to\_char(total)||'元');

commit;

end PRO\_PURCHASE;

8.创建入库存储过程

create or replace procedure PRO\_WAREHOUSING(

recordno varchar2,--进货单号或退货单号

origin in warehousing.wareorigin%type) is

gno goods.goodsno%type;

gnum number;

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

--尝试查询进货入库细则，储存到游标中

cursor cur is select purrecord.goodsno,purrecord.purrecordnum

from purrecord where purrecord.purchaseno=recordno;

begin

--进货入库，使用游标

if origin='进货' then

open cur;

loop

fetch cur into gno,gnum;

exit when cur%notfound;

insert into warehousing values(

'WH'||lpad(to\_char(NO\_WAREHOUSING.NEXTVAL),4,'0'),

gno,gnum,origin,pdate);

end loop;

close cur;

dbms\_output.put\_line('商品进货入库成功');

--退货入库，使用连接查询

else

select goodsno into gno from

refunds left join service

on refunds.serno=service.serno

left join salerecord

on service.salerecordno=salerecord.salerecordno

where refunds.refundno=recordno;

--添加入库信息

insert into warehousing values(

'WH'||lpad(to\_char(NO\_WAREHOUSING.NEXTVAL),4,'0'),

gno,1,origin,pdate);

dbms\_output.put\_line('商品退货入库成功');

end if;

commit;

end PRO\_WAREHOUSING;

9.创建出库存储过程

create or replace procedure PRO\_DELIVERY(

tmpnum in number,

tmpno varchar2,

reason in delivery.deliveryreason%type,

isenough out boolean,

total out float) is

gno varchar2(20);

gnum number;

stocknum number;

temptotal float;

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

begin

isenough:=true;

total:=0;

for i in 1..tmpnum loop

--检查每一件商品的库存，结果储存在isenough变量中

gno:=substr(tmpno,9\*i-8,6);

gnum:=to\_number(substr(tmpno,9\*i-1,1));

select stocknumber into stocknum from stock

where stock.goodsno=gno;

if stocknum<gnum then--库存不足

dbms\_output.put\_line(gno||'号商品库存不足');

isenough:=false;

insert into shortage--添加缺货记录

values(concat('SO',lpad(to\_char(NO\_SHORTAGE.NEXTVAL),

4,'0')),gno,pdate);

end if;

end loop;

if isenough=true then--库存足够，添加出库记录，更新库存信息

dbms\_output.put\_line('库存充足，正在出库');

for i in 1..tmpnum loop

gno:=substr(tmpno,9\*i-8,6);

gnum:=to\_number(substr(tmpno,9\*i-1,1));

select goodsprice into temptotal from goods where goodsno=gno;

total:=total+temptotal\*gnum;

insert into delivery

values(concat('DE',lpad(to\_char(NO\_DELIVERY.NEXTVAL),4,'0')),

gno,gnum,reason,pdate);

update stock set stocknumber=stocknumber-gnum

where stock.goodsno=gno;

end loop;

end if;

commit;

end PRO\_DELIVERY;

10.创建商品信息增删改存储过程

create or replace procedure PRO\_UPGOODS(

opt varchar2,--操作：增加、修改（进价，售价）

gno goods.goodsno%type,

gname goods.goodsname%type,

gprice float,

genterprice float,

gtype goods.type%type,

gscreen goods.screen%type,

gcpu goods.cpu%type,

gmemery goods.memery%type,

gharddisk goods.harddisk%type) is

goodsno varchar2(20);

begin

if opt='增加' then

goodsno:=concat('GD',lpad(to\_char(NO\_GOODS.NEXTVAL),4,'0'));

insert into goods values (goodsno,gname,gprice,genterprice,gtype,gscreen,gcpu,gmemery,gharddisk);

else update goods set

goods.goodsprice=gprice,goods.goodsenterprice=genterprice

where goodsno=gno;

end if;

commit;

end PRO\_UPGOODS;

11.创建客户信息增改存储过程

create or replace procedure PRO\_UPCUS(

opt varchar2,--操作类型：增加、修改(电话)

curno customer.cusno%type,

cusname customer.cusname%type,

cusphone customer.phone%type,

birth date,

gender customer.gender%type) is

cno varchar2(20);

begin

if opt='增加' then

cno:=concat('CR',lpad(to\_char(NO\_CUSTOMER.NEXTVAL),4,'0'));

insert into customer values (cno,cusname,cusphone,0,'普通',birth,gender);

else update customer set

customer.phone=cusphone where customer.cusno=curno;

end if;

commit;

end PRO\_UPCUS;

### 5.1.5创建触发器

1.更新客户积分等级触发器

CREATE OR REPLACE TRIGGER TRI\_POINTS

after insert

on sales

for each row

declare

pragma autonomous\_transaction;--oracle自定义事务处理

cuslevel customer.cuslevel%type;

points customer.points%type;

begin

--获取客户积分，等级

select cuslevel,points into cuslevel,points from

customer

where customer.cusno=:new.cusno;

--计算购买商品后客户积分和等级

points:=points+:new.saledistotal;

if points>6000 and points<14000 then cuslevel:='银卡';

elsif points>14000 and points<22000 then cuslevel:='金卡';

elsif points>22000 then cuslevel:='钻石';

else cuslevel:='普通';

end if;

--更新客户积分和等级

update customer

set customer.points=points,customer.cuslevel=cuslevel

where cusno=:new.cusno;

commit;

end TRI\_POINTS;

2.自动添加缺货记录触发器

--商品库存数量少于5时自动添加缺货记录

CREATE OR REPLACE TRIGGER TRI\_SHORTAGE

after update

on stock

for each row

declare

sno shortage.shortageno%type;

pdate date:=to\_date(to\_char(sysdate(),'YYYY/MM/DD'),'YYYY/MM/DD');

begin

if :new.stocknumber<5 then

sno:=concat('ST',lpad(to\_char(NO\_SHORTAGE.NEXTVAL),4,'0'));

insert into shortage values(sno,:new.goodsno,pdate);

end if;

end TRI\_SHORTAGE;

## 5.2 数据入库

1.客户单

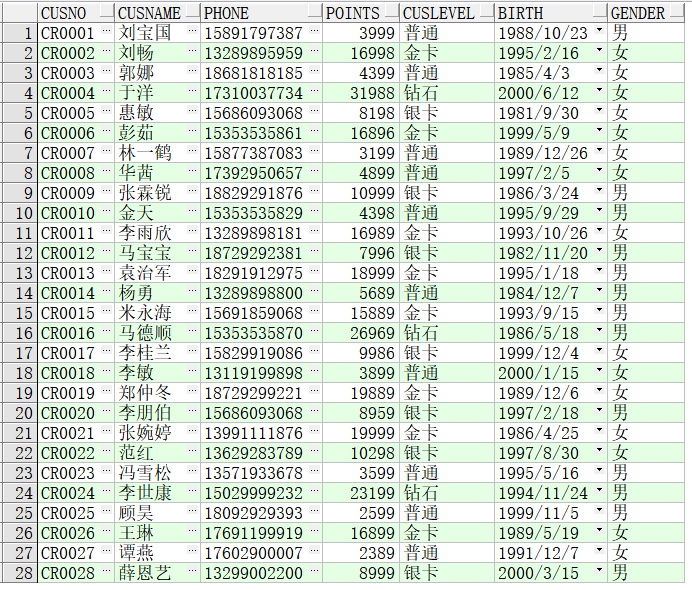


图5-1客户单

2. 商品单

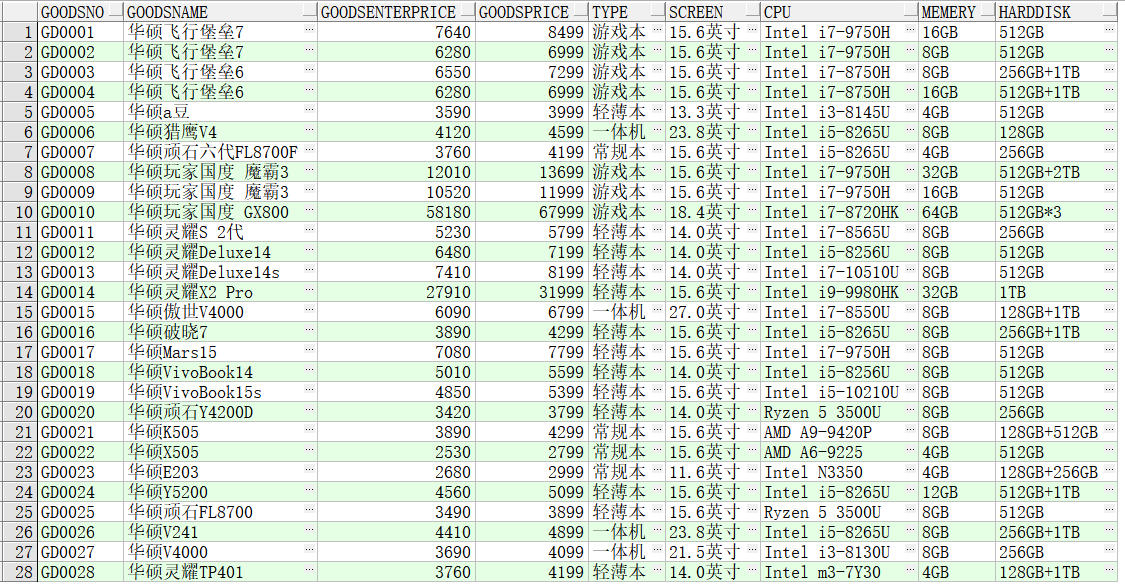


图5-2商品单

3. 销售单

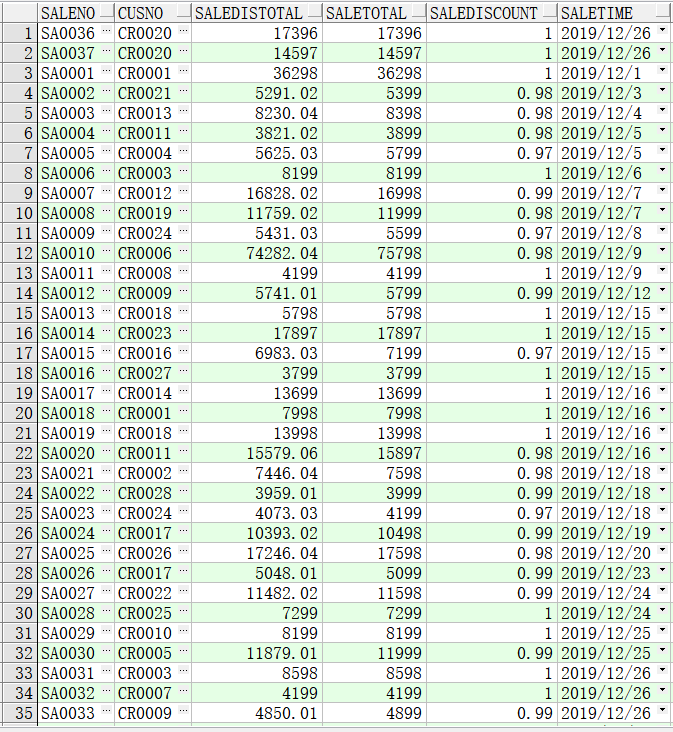


图5-3销售记录单

4. 销售细则

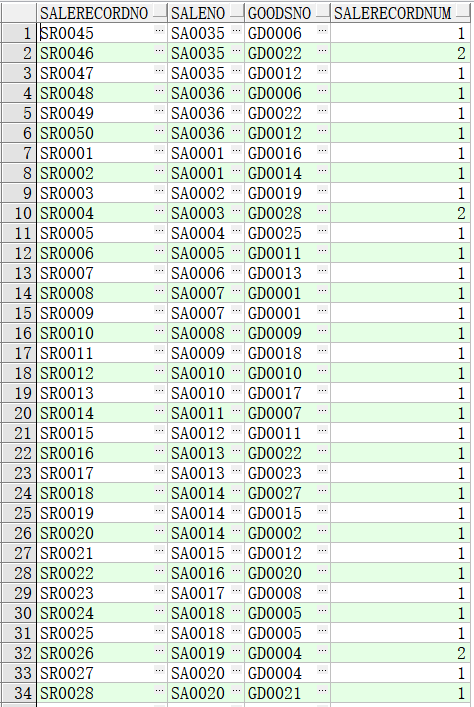


图5 -4销售细则单

5. 出库单

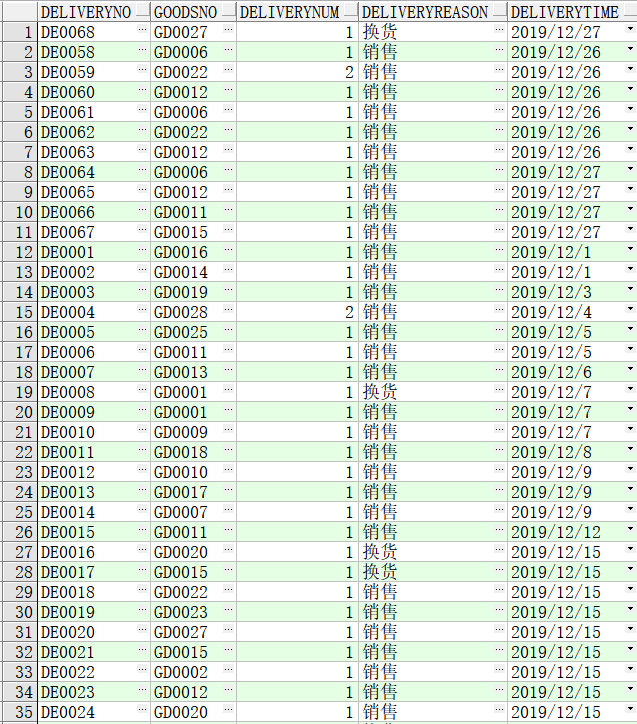


图5-5出库单

6. 入库单

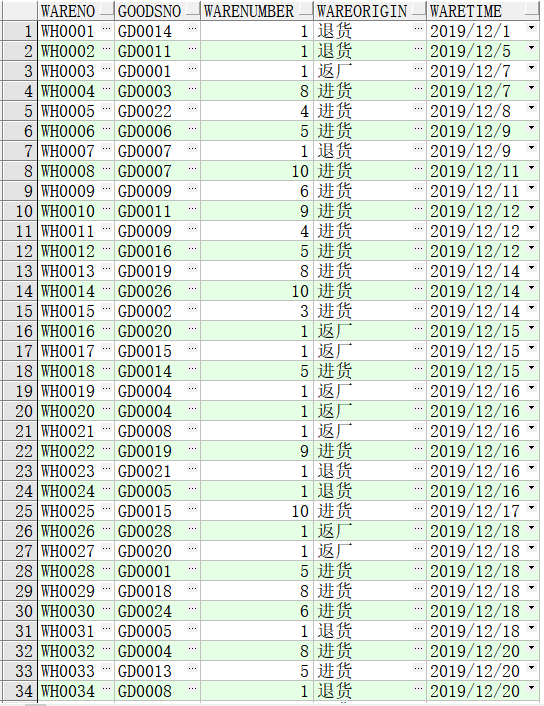


图5-6入库单

7. 库存单

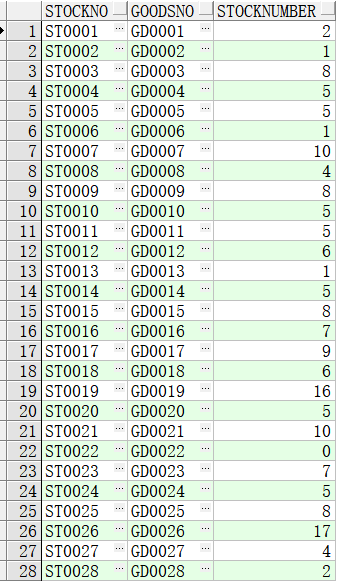


图5-7库存单

8. 缺货单

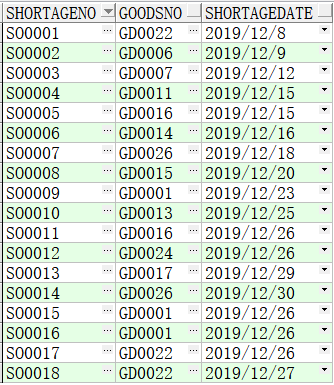


图5-8缺货单

9. 采购单

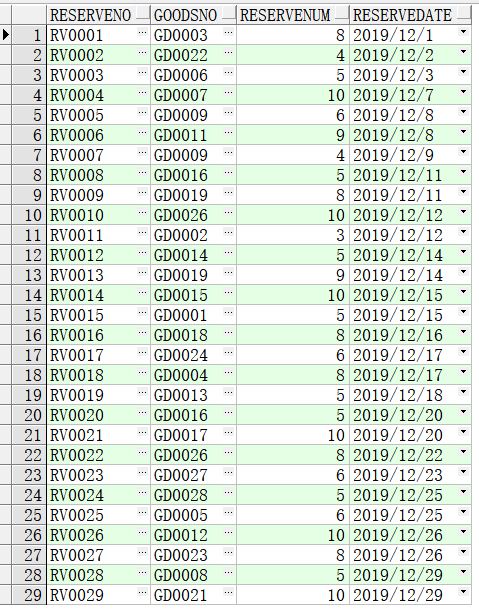


图5-9采购单

10.进货单

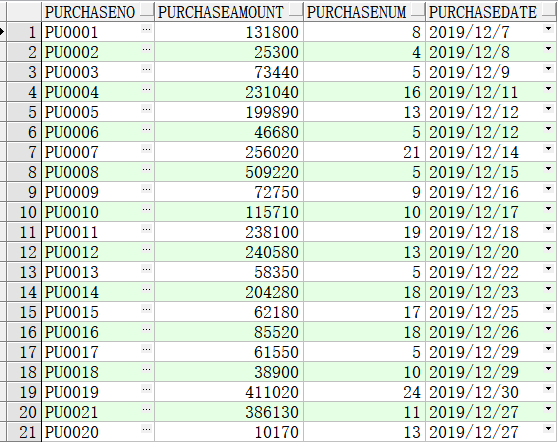


图5-10进货单

11.进货细则

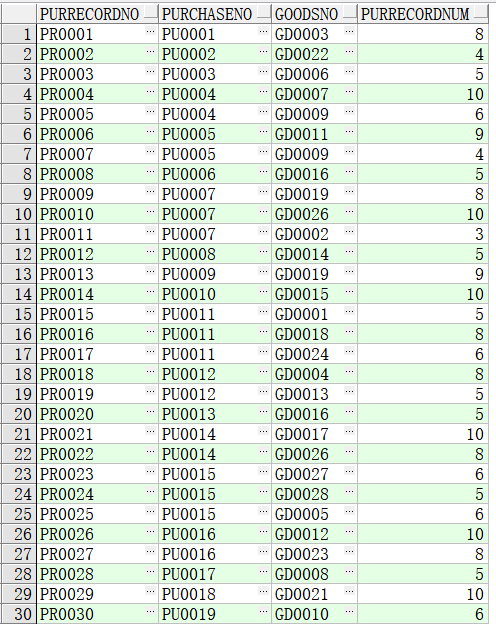


图5-11进货细则单

12.售后单

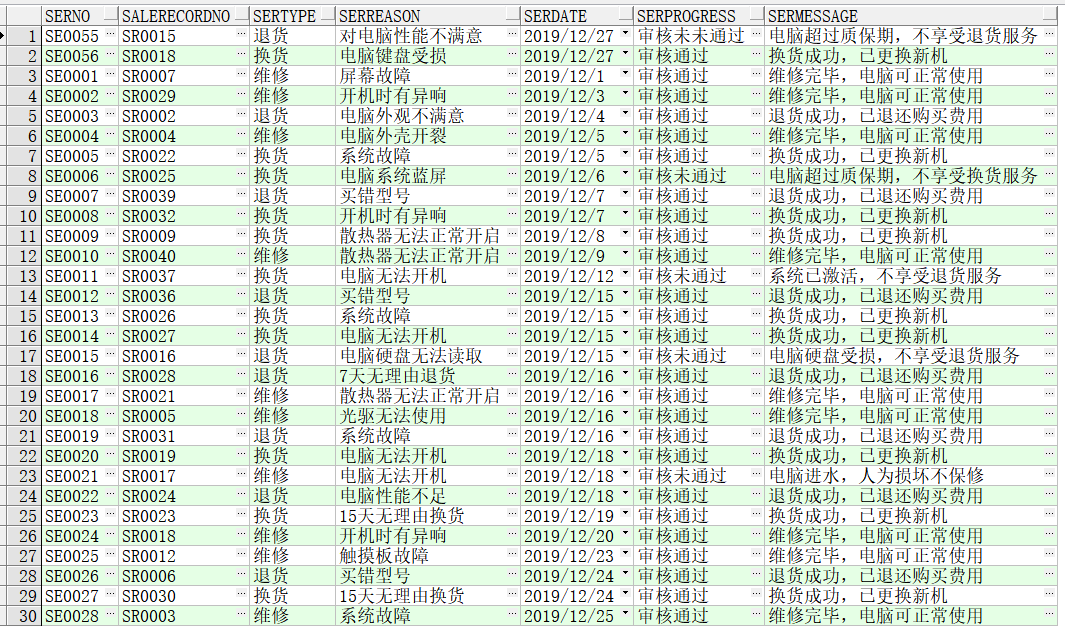


图5-12售后记录单

13.维修单

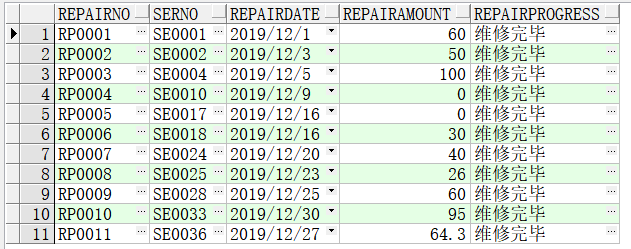


图5-13维修单

14.退货单

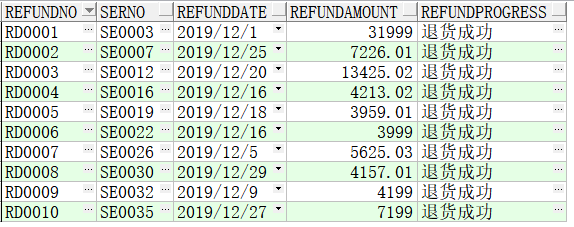


图5-14退货单

15.换货单

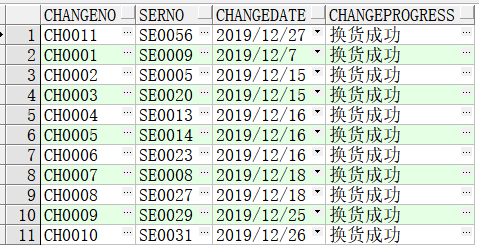


图5-15换货单

1. 返厂单

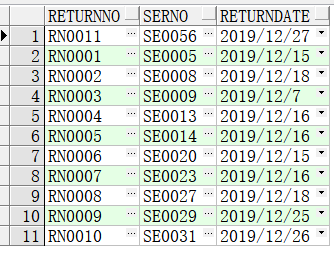


图5-16

## 5.3 数据库测试

存储过程与触发器测试：

1.客户信息增改测试：经测试，存储过程运行结果与预期相同。

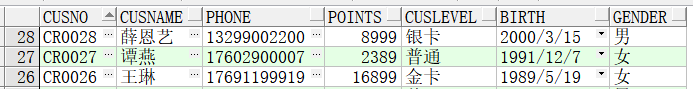


图5-17客户表原始数据

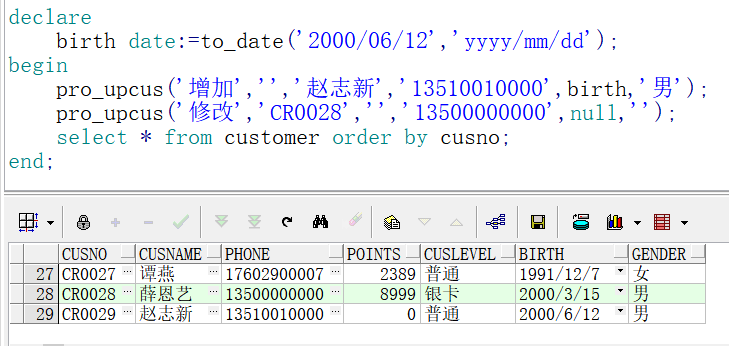


图5-18 客户信息增改测试语句

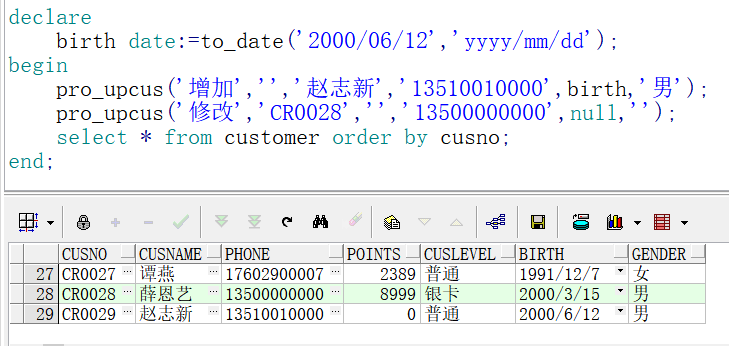


图5-19客户表测试后数据

2.商品信息增改测试：经测试，存储过程运行结果与预期相同。

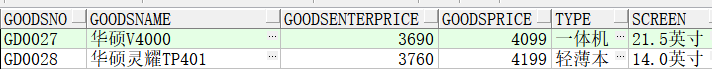


图5-20商品表原始数据

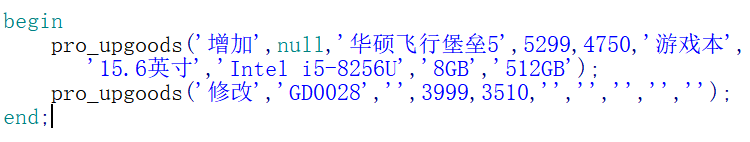


图5-21商品信息增改测试语句

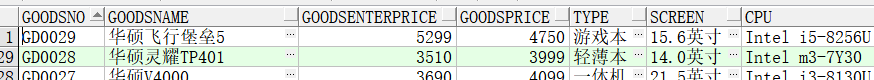


图5-22 商品表测试后数据

3.售后服务测试：经测试，存储过程运行结果与预期相同。

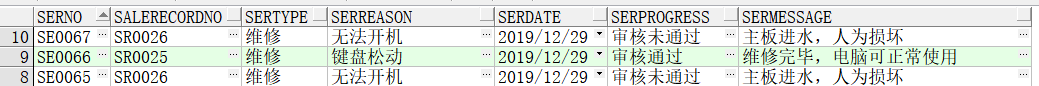


图5-23售后记表原始数据

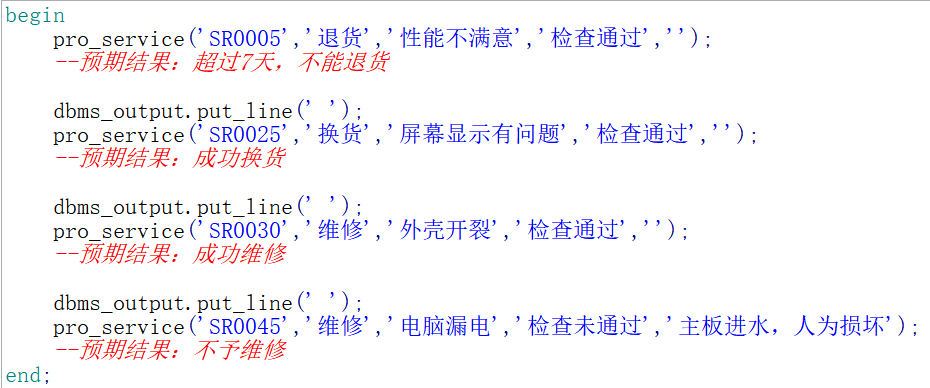


图5-24售后测试语句

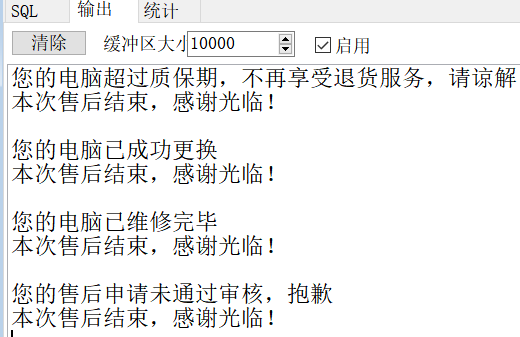


图5-25售后测试输出

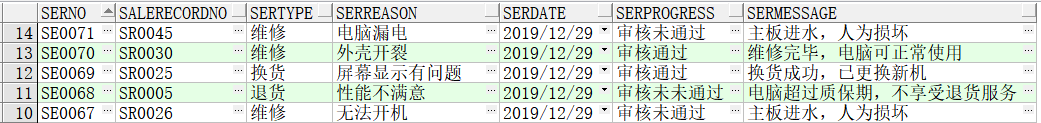


图5-26售后记录表测试后数据



图5-27换货单测试后数据

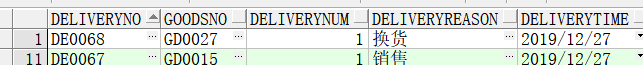


图5-28出库单测试后数据



图5-29返厂单测试后数据

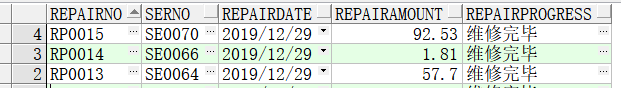


图5-30维修单测试后数据

4.销售存储过程及更新客户积分触发器测试：

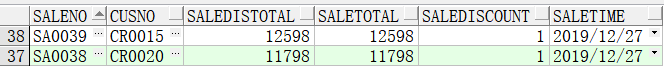


图5-31销售单测试前数据

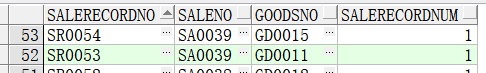


图-32销售细则测试前数据

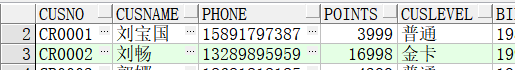


图5-33客户单测试前数据

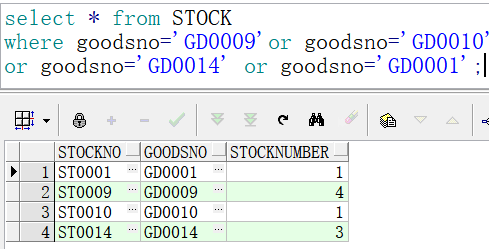


图5-34库存单测试前数据

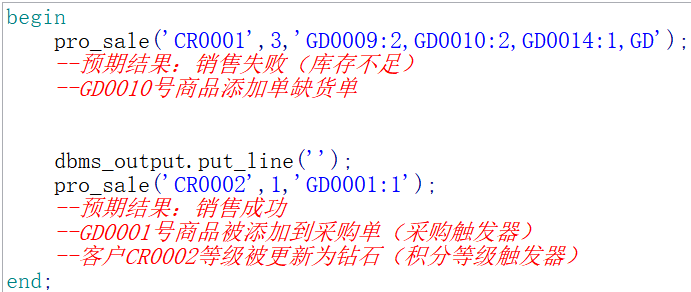


图5-35销售测试语句

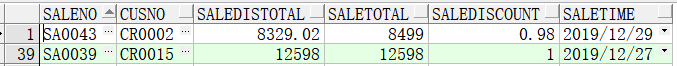


图5-36销售单测试后数据

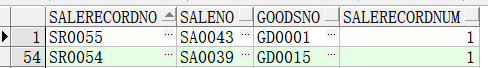


图5-37销售细则单测试后数据

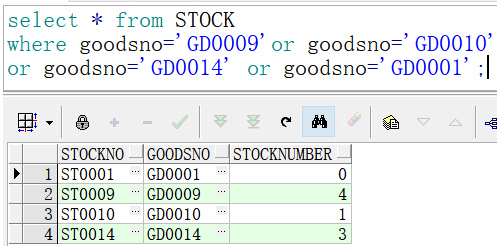


图5-38库存单测试后数据



图5-39采购单测试后数据

5.进货存储过程测试：

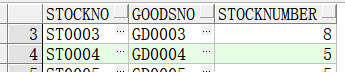


图5-40库存单测试前数据

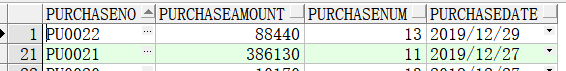


图5-41进货单测试前数据

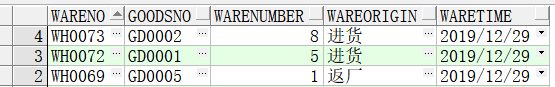


图5-42入库单测试前数据

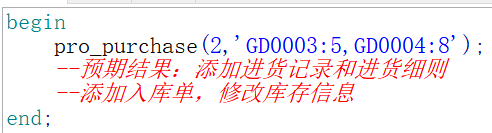


图5-43进货测试语句

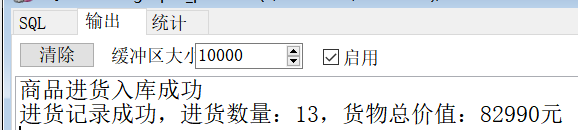


图5-44进货测试输出

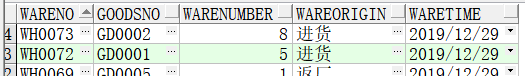


图5-45入库单测试后数据

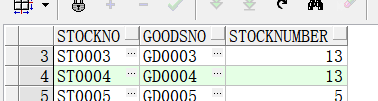


图5-46库存单测试后数据



图5-47进货单测试后数据

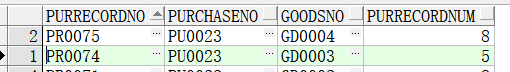


图5-48进货细则单测试后数据

# 6．总结

通过本次实习，我收获了很多，也发现了自己的许多不足。在为期两周的实训生活中，在老师的指导下按着制定的实训计划，从一开始的迷茫、忙乱，到现在对数据库系统的建立有了一个较为清晰的认识。

熟悉数据库系统开发流程：在最开始的需求分析阶段，通过上网查找资料，涉猎了许多电脑商城运营方面的知识。起初面对庞大的信息量不知所措。又花了大量时间仔细研究老师的模板后才渐渐有了思路，一步一步完成了需求分析的工作。这也是我认为数据库系统建立最费时费力费脑的环节。在之后的阶段中，也越来越发现做扎实需求分析工作的重要性。由于需求分析的不甚完善、全面，在之后的概念结构设计、逻辑结构设计中，我无数次转向修改需求分析，也让我再次认识到在最开始统筹全局的重要性。

巩固了数据库编程能力：在编写存储过程、触发器、视图代码的过程中，把想法转换成机器能识别的语言实在是下了很大功夫，多次的编译、多次的修改终于让系统能够满足用户的基本需求，实现了数据库系统方便用户操作数据的建立意义。

数据库源于生活：如果没有这次实训，我或许只会记得需求分析的方法是“自顶向下”这短短的四个字，但通过实训，用实践证明这短短四个字的精妙之处。通过实训也让我接触到了数据库真正的美，它是结合多方面学科知识，对于开发者的学习能力、规划能力、谨慎态度都有十分高的要求。也促使我更留心生活，因为数据库就源于生活需求的方方面面。

锻炼课程设计成果总结能力：这次实训也锻炼了我编辑课程论文、编辑以及汇报PPT的能力，为今后还会无数次面临的课程设计打下基础。对于课程论文的排版要求、汇报PPT的版面以及内容要求有了初步的认识。

# 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} |
| 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 | AllGradeTabel | 全体学生成绩单 | StuID, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TotalGoal |
| DS-6 | PutGradeTabel | 提交成绩表 | StuID, Subject, Goal, Semster |
| DS-7 | StuClass | 学生班级表 | StuID, ClassID, Semster |
| DS-8 | ReportTable | 学生报道单 | StuID, InSchool, Semester |
| DS-9 | 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-10 | StuGradeTable | 学生单人的成绩 | StuID, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TotalGoal |
| DS-11 | SubGradeTable | 单科成绩 | StuName , Subject, Goal, GoalRank, Semester, TeaName |
| DS-12 | StuTable | 学生表 | StuID, StuName, InSchool, FinCheck, Semster |

3.数据流

表6-3 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 | 顾客订单 | 顾客的购买商品的订单 | 顾客 | 销售管理模块 | DS-3 | 300次/月 | 500次/月 |
| F2 | 销售发票 | 顾客购买商品的消费凭证 | 销售管理模块 | 顾客 | DS-20 | 300次/月 | 500次/月 |
| F3 | 缺货反馈单 | 库存不足以满足销售时的反馈 | 销售管理模块 | 顾客 | DS-10 | 3次/月 | 6次/月 |
| F4 | 售后申请单 | 顾客的退货、换货、维修申请 | 顾客 | 售后服务管理模块 | DS-17 | 30次/月 | 60次/月 |
| F5 | 售后反馈单 | 周后管理对顾客售后申请的反馈 | 售后服务管理模块 | 顾客 | DS-18 | 30次/月 | 60次/月 |
| F6 | 采购单 | 商品管理部门做出的商品采购计划单 | 商品管理模块 | 采购部门 | DS-11 | 5次/月 | 10次/月 |
| F7 | 进货单 | 采购部门的进货记录 | 采购部门 | 商品管理模块 | DS-15 | 5次/月 | 10次/月 |
| F8 | 库存查询 | 仓库中的商品余量 | 库存表 | 销售管理模块 | DS-12 | 300次/月 | 500次/月 |
| F9 | 缺货信息 | 库存不足时产生的缺货信息 | 销售管理模块 | 缺货表 | DS-10 | 3次/月 | 6次/月 |
| F10 | 销售信息 | 销售产生的记录信息 | 销售管理模块 | 销售记录表 | DS-3 | 300次/月 | 500次/月 |
| F11 | 发货单 | 根据订单产生的发货请求 | 销售记录表 | 商品管理模块 | DS-13 | 300次/月 | 500次/月 |
| F12 | 订单信息 | 售后查询的订单信息 | 销售记录表 | 售后服务管理模块 | DS-3 | 300次/月 | 500次/月 |
| F13 | 退货单 | 售后产生的退货记录 | 售后服务管理模块 | 退货登记表 | DS-8 | 5次/月 | 10次/月 |
| F14 | 换货单 | 售后产生的换货记录 | 售后服务管理模块 | 换货登记表 | DS-7 | 5次/月 | 10次/月 |
| F15 | 退货请求 | 向仓库发出的退货请求 | 退货登记表 | 商品管理模块 | DS-8 | 5次/月 | 10次/月 |
| F16 | 换货请求 | 向仓库发出的换货请求 | 换货登记表 | 商品管理模块 | DS-7 | 5次/月 | 10次/月 |
| F17 | 库存信息 | 采购规划所需的库存信息 | 库存表 | 商品管理模块 | DS-12 | 30次/月 | 30次/月 |
| F18 | 出入库单 | 向库存表填写的出入库信息 | 商品管理模块 | 库存表 | DS-13, DS-14 | 30次/月 | 50次/月 |
| F19 | 缺货单 | 商品管理模块所需的缺货信息 | 缺货表 | 商品管理模块 | DS-10 | 30次/月 | 50次/月 |
| F20 | 订单信息 | 根据订单产生的出库所需的信息 | 订单管理 | 出库检查 | DS-3 | 300次/月 | 500次/月 |
| F21 | 顾客信息 | 购买商品的顾客的信息 | 订单管理 | 客户信息表 | DS-2 | 300次/月 | 500次/月 |
| F22 | 客户信息 | 订单管理所需的客户信息 | 客户信息表 | 订单管理 | DS-2 | 300次/月 | 500次/月 |
| F23 | 返厂单 | 退换货产生的返厂需求 | 退换处理 | 返厂记录表 | DS-9 | 3次/月 | 6次/月 |
| F24 | 出库单 | 换货所需的出库信息 | 退换处理 | 出入库管理 | DS-13 | 3次/月 | 6次/月 |
| F25 | 入库单 | 退货所需的入库信息 | 退换处理 | 出入库管理 | DS-14 | 3次/月 | 6次/月 |
| F26 | 出库记录 | 商品出库时的记录信息 | 出入库管理 | 出库表 | DS-13 | 300次/月 | 500次/月 |
| F27 | 入库记录 | 商品入库时的记录信息 | 出入库管理 | 入库表 | DS-14 | 20次/月 | 10次/月 |
| F28 | 审核进度 | 售后申请的审核进度 | 售后审核 | 售后记录表 | DS-18 | 30次/月 | 60次/月 |
| F29 | 售后信息 | 售后审核发出的售后信息 | 售后审核 | 故障检测 | DS-5 | 30次/月 | 60次/月 |
| F30 | 检测进度 | 故障检测的进度 | 故障检测 | 售后记录表 | DS-18 | 25次/月 | 50次/月 |
| F31 | 维修申请 | 售后维修的申请 | 故障检测 | 售后维修 | DS-5 | 25次/月 | 50次/月 |
| F32 | 维修进度 | 售后维修的进度 | 售后维修 | 售后记录表 | DS-18 | 20次/月 | 40次/月 |
| F33 | 维修信息 | 售后维修的记录 | 售后维修 | 维修记录表 | DS-6 | 20次/月 | 40次/月 |

4.处理逻辑

表6-4 处理逻辑

| **编号** | **处理逻辑名称** | **简述** | **输入数据流** | **输出数据流** | **解释** | **处理频率** |
| --- | --- | --- | --- | --- | --- | --- |
| P1.1 | 订单管理 | 开具商品订单 | F1顾客订单 F20客户信息 | F2销售发票 F10销售信息 F18订单信息 F19顾客信息 | 根据顾客订单和客户信息填写订单 | 300次/月 |
| P1.2 | 出库检查 | 检查库存余量 | F8库存信息 F18订单信息 | F3缺货反馈单 F9缺货信息 | 根据订单检查库存是否足够 | 300次/月 |
| P2.1 | 出入库管理 | 管理出入库的商品 | F6进货单 F11销售单 F24出库单 F25入库单 | F16出入库单 F26出库记录 F27入库记录 | 根据出库和入库申请进行出入库管理 | 500次/月 |
| P2.2 | 采购规划 | 制定采购计划 | F15库存信息 F17缺货单 | F5采购表 | 根据缺货单及销售情况制定采购计划 | 5次/月 |
| P2.3 | 退换处理 | 处理售后退换货 | F13退换货单 | F21返厂记录表 F22退货单 F23换货单 | 根据退换表进行退换处理 | 3次/月 |
| P3.1 | 售后审核 | 审核售后申请 | F4售后申请单 F12订单信息 | F28审核进度 F29售后信息 | 根据售后申请和订单信息检查是否可以进行售后服务 | 30次/月 |
| P3.2 | 故障检测 | 检测售后商品故障 | F29售后信息 | F13退换处理单 F30检测进度 F31检测结果 | 根据商品信息和售后商品检查是否符合售后条件 | 25次/月 |
| P3.3 | 售后维修 | 维修故障商品 | F31检测结果 | F32维修进度，F33维修信息 | 对符合规定的售后维修商品进行维修登记 | 20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 | 销售记录表 | 顾客订单的详细记录 | DS-3 | P1.1, P2.1, P3.1 |
| S2 | 退货记录表 | 售后退货的详细记录 | DS-8 | P2.3, P3.2 |
| S3 | 换货记录表 | 售后换货的详细记录 | DS-7 | P2.3, P3.2 |
| S4 | 缺货表 | 库存不足时的缺货登记表 | DS-10 | P1.3, P2.2 |
| S5 | 库存表 | 仓库中商品的库存量 | DS-12 | P1.3, P2.1, P2.2 |
| S6 | 客户信息 | 每个库户的详细信息表 | DS-2 | P1.1 |
| S7 | 返厂记录表 | 需要返厂的退换商品记录表 | DS-9 | P2.3 |
| S8 | 入库表 | 采购或退货产生的入库记录表 | DS-14 | P2.1 |
| S9 | 出库表 | 销售或换货产生的出库记录表 | DS-15 | P2.1 |
| S10 | 售后记录表 | 所有售后服务记录表 | DS-5 | P3.1, P3.2, P3.3 |
| S11 | 维修登记表 | 售后维修服务记录表 | DS-6 | P3.3 |