

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
1.1 数据流程图.....	2
1.2 数据字典.....	4
2 概念结构设计	4
2.1 分 E-R 图建立	5
2.2 全局/整体 E-R 图	7
3 逻辑结构设计	8
3.1 建立关系模式.....	9
3.2 关系模式规范化处理	9
3.3 用户子模式设计	11
3.4 关系模式逻辑结构定义	12
4. 数据库物理设计	14
5. 数据库实施与测试	15
5.1 数据库及数据库对象建立	15
5.1.1 创建基本表.....	15
5.1.2 创建视图.....	22
5.1.3 创建索引	24
5.1.4 创建存储过程	24
5.1.5 创建触发器.....	38
5.2 数据入库.....	40
5.3 数据库测试.....	49
6. 总结	55
7. 附录	56

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：

- a. 档案管理员对学生档案进行登记和修改
- b. 档案管理员对班级档案进行登记和修改
- c. 当登记或修改班级档案时，对学生的班级信息进行修改
- d. 学生对自己的档案进行查询
- e. 班主任对自己的所带班级档案进行查询

（2）成绩管理：

- a. 老师对成绩进行登记修改
- b. 老师查询成绩时，生成仅有该老师所带科目的成绩表
- c. 班主任查询成绩时，生成全班的成绩表
- d. 学生查询成绩时，生成仅有自己成绩的成绩表

（3）入学管理：

- a. 学生提交报告单
- b. 根据学生报告单将学生分为，正常报道、退学、未报道三种
- c. 修改正常报道学生的学生档案
- d. 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

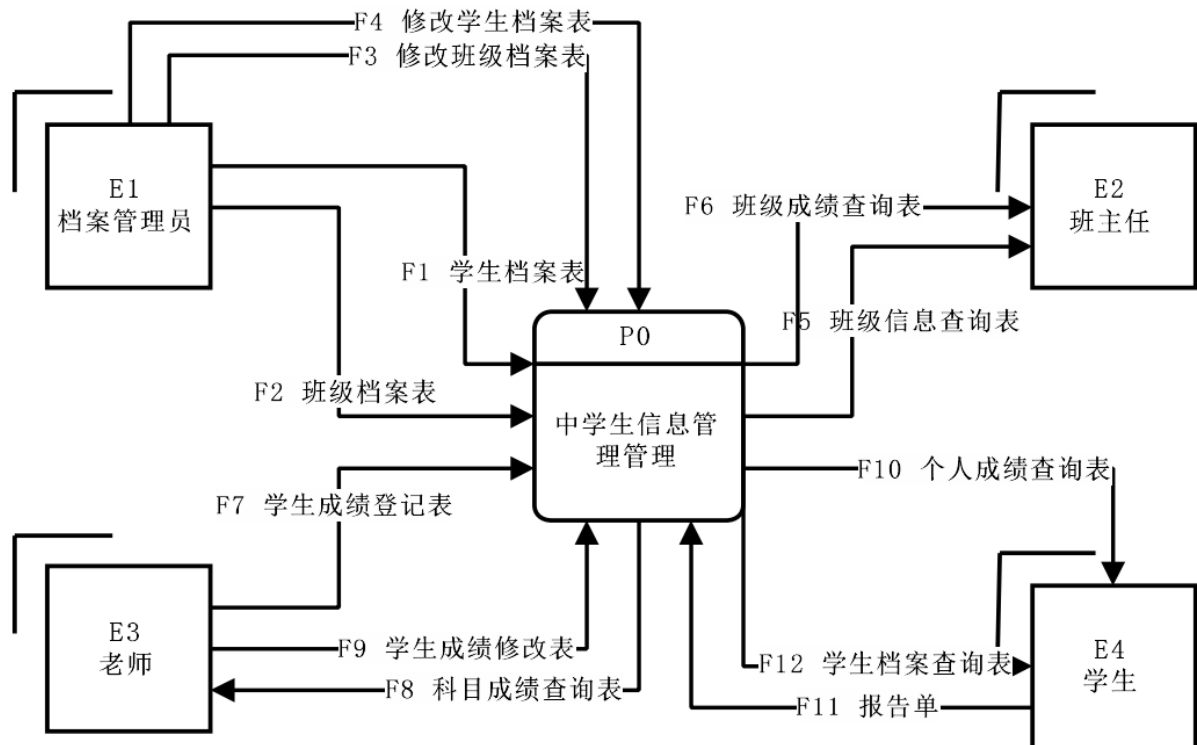


图 1-1 顶层数据流

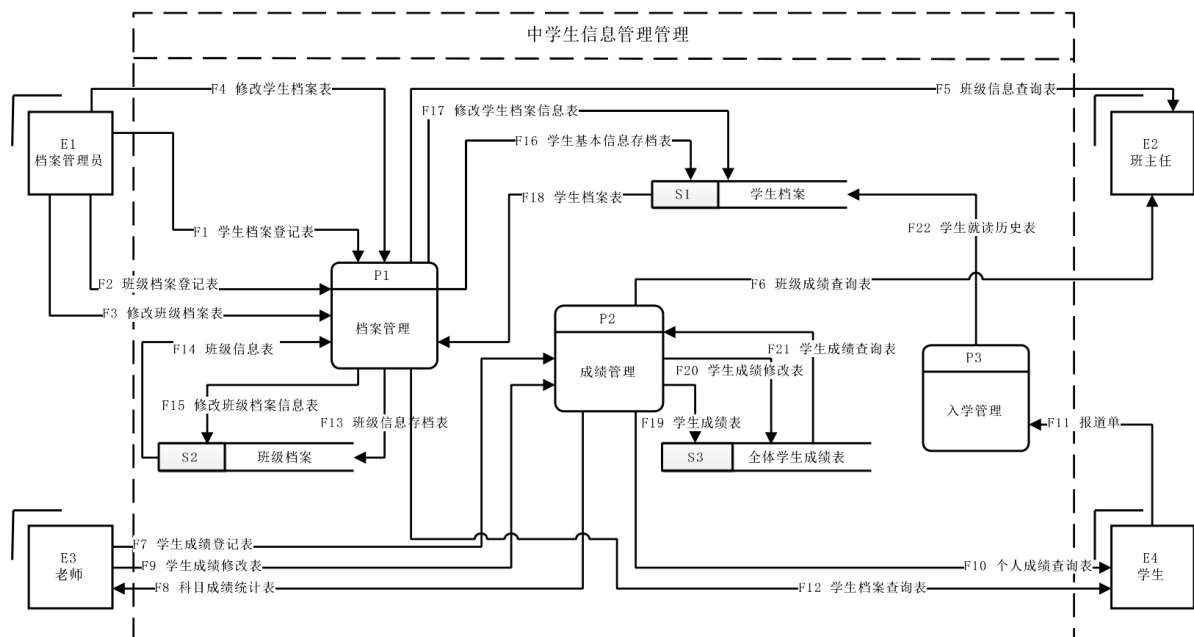


图 1-2 第一层数据流程图

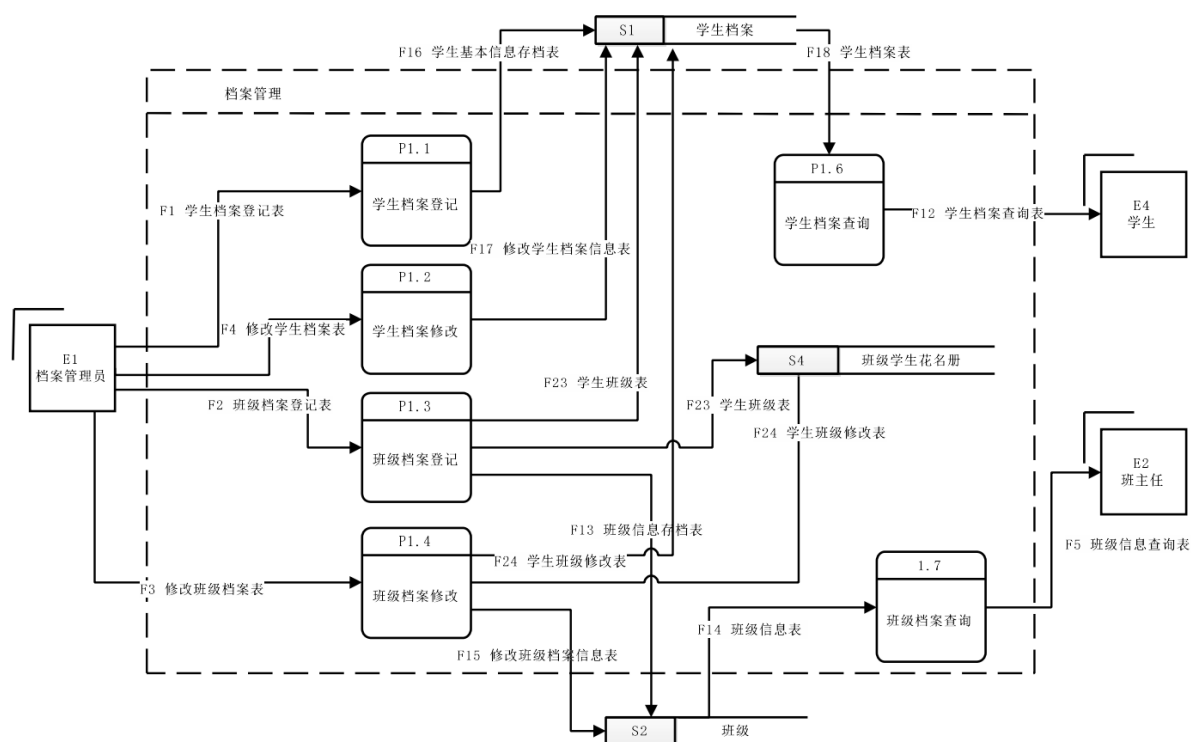


图 1-3 第二层数据流程图——档案管理

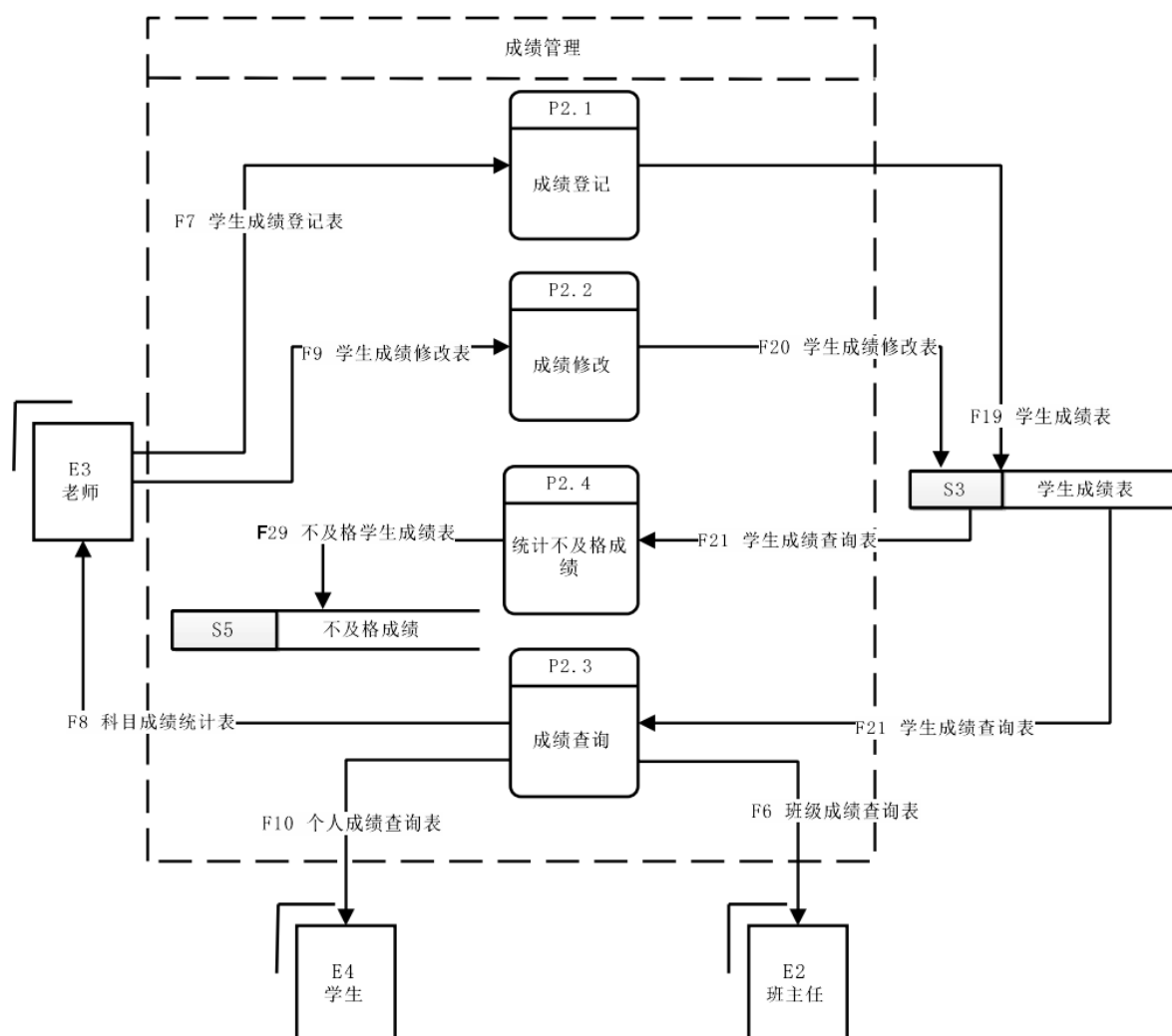


图 1-4 第二层数据流程图——成绩管理

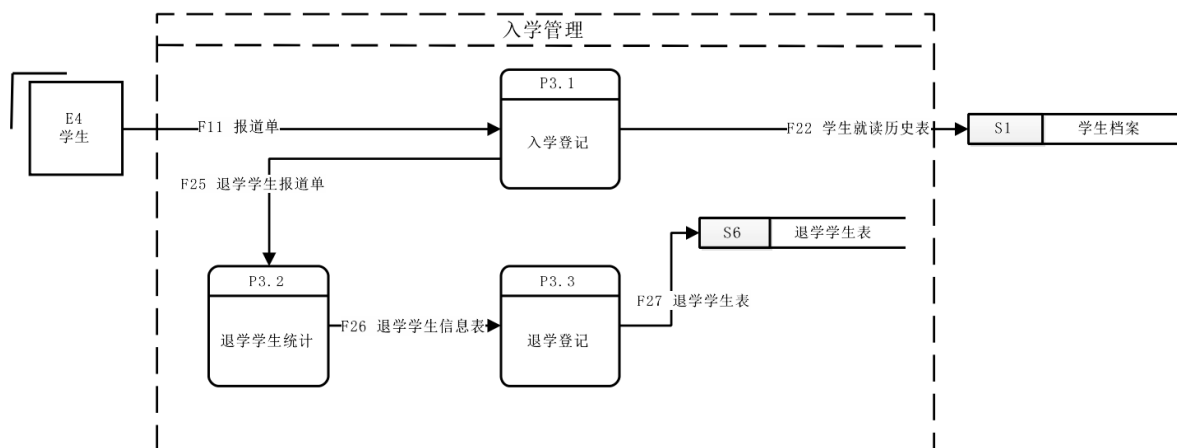


图 1-5 第二层数据流程图——入学管理

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

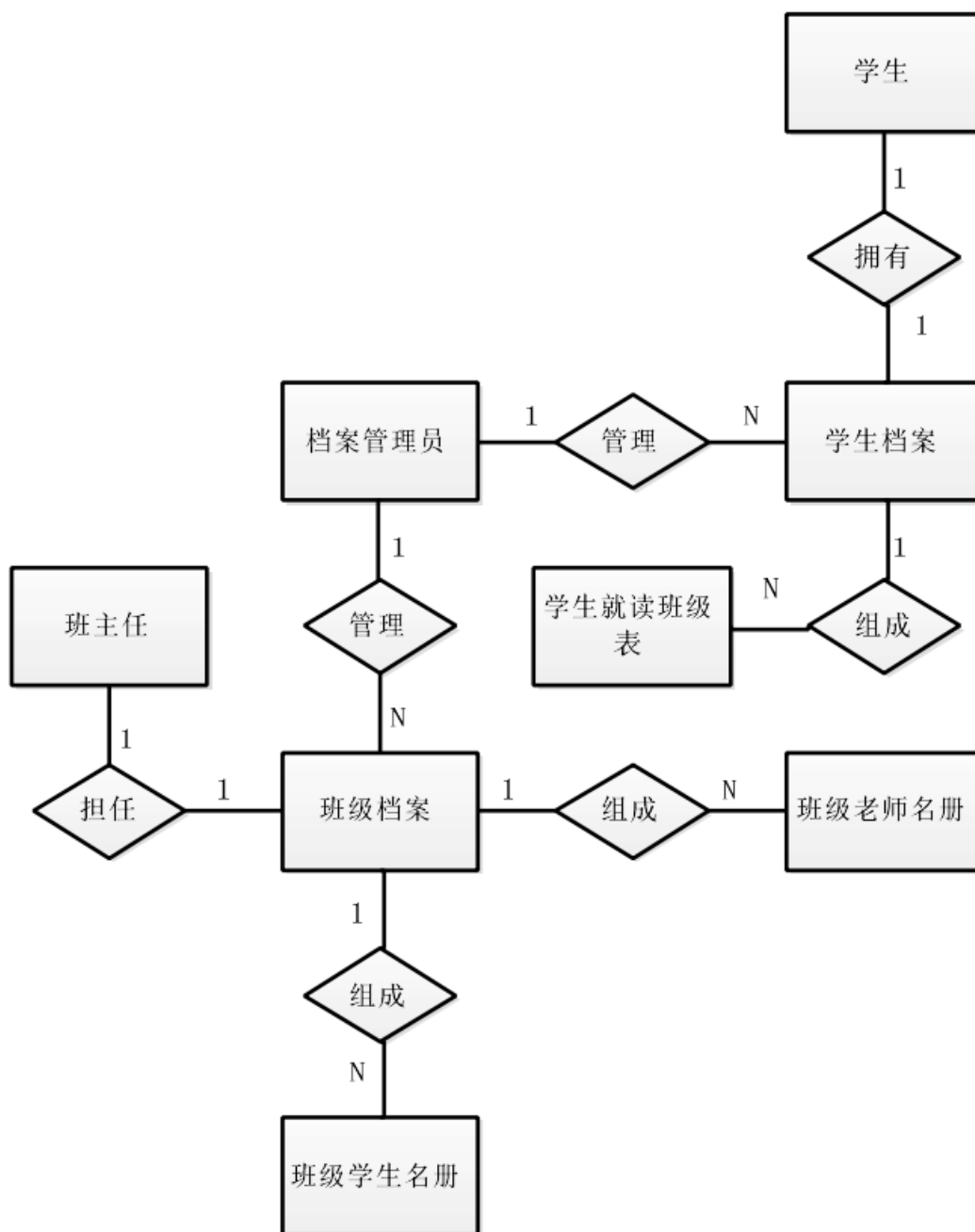


图 2-1 分 E-R 图—档案管理

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）

（2）成绩管理 E-R 图

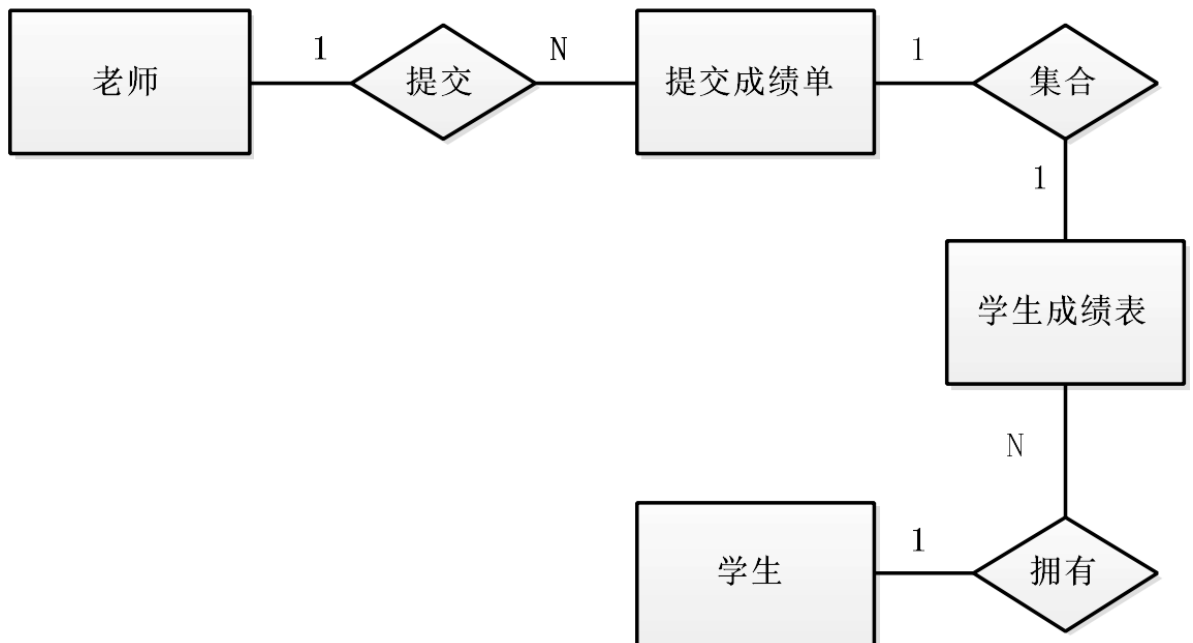


图 2-2 分 E-R 图—成绩管理

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

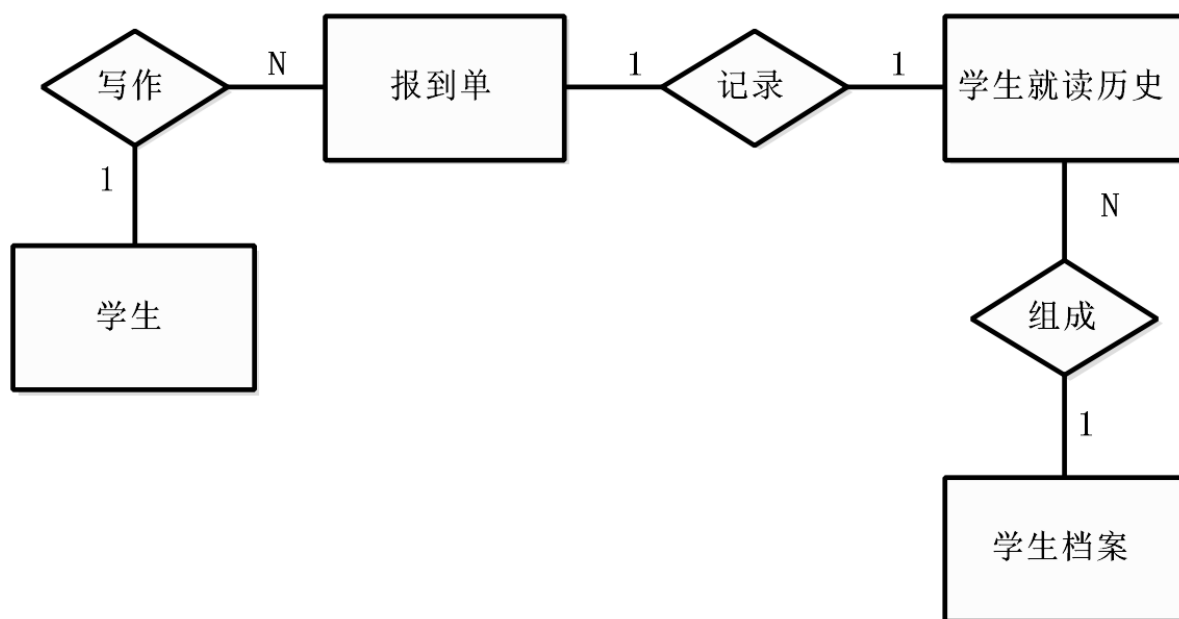


图 2-3 分 E-R 图——入学管理

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）

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。

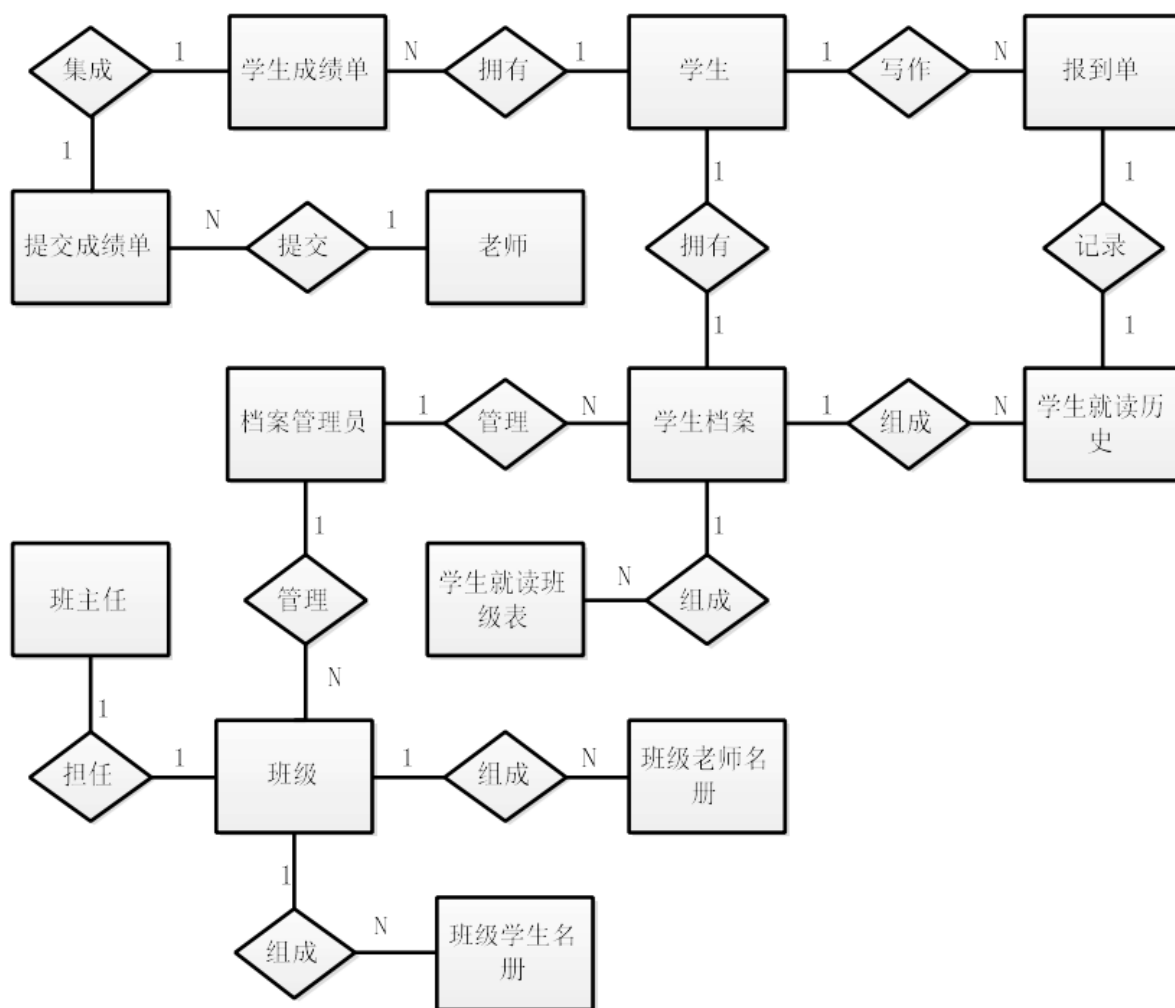


图 2-4 全局 E-R 图

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。

2. 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

3. 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

4. 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

5. 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

6. 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。

7. 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

8. 班级老师名册（班级 ID，老师 ID）

主属性为班级 ID，老师 ID。

9. 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

10. 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

11. 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

12. 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

13. 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

班级成绩单：查询班级所有人成绩

科目成绩单：查看单科科目的所有人成绩

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，历史老师姓名，政治老师姓名，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）

班级成绩单（班级 ID，学生 ID，学生姓名，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，成绩排名）

科目成绩单（学生 ID，科目，学期，成绩，成绩排名）

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
-----	----	------	----	----	----	------

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
StuBirthday	学生的出生日期	data				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		
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		
TeaID	老师 ID	varchar	10	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		
StuID	学生 ID	varchar	10	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	data				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	data				1900/1/1-2100/1/1

4. 数据库物理设计

建立索引：

IND_GOODSNAME：普通类型，用于顾客根据电脑名称查询相应的电脑编号进行购买。

IND_STOCK：唯一类型，用于销售时对相应的商品库存量进行查询。

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. 创建商品表

create table Goods

```
(  
GoodsNo          varchar2(10) not null,  
GoodsName        varchar2(50),  
GoodsEnterPrice  float,  
GoodsPrice       float,  
Type             varchar2(20),  
Screen           varchar2(20),  
CPU              varchar2(20),  
Memery           varchar2(10),  
HardDisk         varchar2(30),  
constraint PK_GoodsNo primary key (GoodsNo),  
constraint CK_GoodsEnterPrice check (GoodsEnterPrice>=0),  
constraint CK_GoodsPrice check (GoodsPrice>=0));
```

2. 创建客户表

create table Customer

```
(  
CusNo            varchar2(20),  
CusName          varchar2(20),  
Phone            varchar2(30),  
Points           int,  
CusLevel         varchar2(10),  
Birth            date,  
Gender           varchar2(5),
```

```
constraint PK_CusNo primary key (CusNo),
constraint CK_Points check (Points>=0),
constraint CK_Gender check (Gender='男' or Gender='女'),
constraint CK_CusLevel check
    (CusLevel='钻石' or CusLevel='金卡'
    or CusLevel='银卡' or CusLevel='普通')
);
```

3. 创建销售单

```
create table Sales
(
    SaleNo          varchar2(20),
    CusNo           varchar2(20),
    SaleDisTotal    float,
    SaleTotal       float,
    SaleDiscount     float,
    SaleTime        date,
    constraint PK_SaleNo primary key (SaleNo),
    constraint FK_SCusNo foreign key (CusNo)
        references Customer(CusNo),
    constraint CK_SaleDiscount check
        (SaleDiscount>=0 and SaleDiscount<=1),
    constraint CK_SaleTotal check (SaleTotal>=0),
    constraint CK_SaleDisTotal check (SaleDisTotal>=0)
);
```

4. 创建销售细则单

```
create table SaleRecord
(
    SaleRecordNo    varchar2(20),
    SaleNo          varchar2(20),
    GoodsNo         varchar2(20),
    SaleRecordPrice  float,
```

```
SaleRecordNum    int,  
constraint PK_SaleRecordNo primary key (SaleRecordNo),  
constraint FK_RSaleNo foreign key (SaleNo)  
    references Sales(SaleNo),  
constraint FK_RGoodsNo foreign key (GoodsNo)  
    references Goods(GoodsNo),  
constraint CK_SaleRecordPrice check (SaleRecordPrice>=0),  
constraint CK_SaleRecordNum check (SaleRecordNum>0)  
);
```

5. 创建售后单

```
create table Service
```

```
(  
    SerNo          varchar2(20),  
    SaleRecordNo   varchar2(20),  
    SerType        varchar2(6),  
    SerReason      varchar2(60),  
    SerDate        date,  
    SerProgress    varchar2(20),  
    SerMessage     varchar2(60),  
    constraint PK_SerNo primary key (SerNo),  
    constraint FK_SerSaleRecordNo foreign key (SaleRecordNo)  
        references SaleRecord(SaleRecordNo),  
    constraint CK_SerType check  
        (SerType='退货' or SerType='换货' or SerType='维修'));
```

6. 创建维修单

```
create table Repairs
```

```
(  
    RepairNo       varchar2(20),  
    SerNo          varchar2(20),  
    RepairDate     date,  
    RepairAmount   float,
```

```
RepairProgress  varchar2(20),
constraint PK_RepairNo primary key (RepairNo),
constraint FK_RepSerNo foreign key (SerNo)
    references Service(SerNo),
constraint CK_RepairAmount check (RepairAmount>=0)
);
```

7. 创建换货单

```
create table Changes
(
    ChangeNo      varchar2(20),
    SerNo         varchar2(20),
    ChangeDate    date,
    ChangeProgress varchar2(20),
    constraint PK_ChangeNo primary key (ChangeNo),
    constraint FK_ChaSerNo foreign key (SerNo)
        references Service(SerNo)
);
```

8. 创建退货单

```
create table Refunds
(
    RefundNo      varchar2(20),
    SerNo         varchar2(20),
    RefundDate    date,
    RefundAmount  float,
    RefundProgress varchar2(20),
    constraint PK_RefundNo primary key (RefundNo),
    constraint FK_RefSerNo foreign key (SerNo)
        references Service(SerNo),
    constraint CK_RefundAmount check (RefundAmount>=0)
);
```

9. 创建返厂单

create table Returns

```
(
    ReturnNo      varchar2(20),
    SerNo          varchar2(20),
    ReturnDate     date,
    constraint PK_ReturnNo primary key (ReturnNo),
    constraint FK_RetSerNo foreign key (SerNo)
        references Service(SerNo)
);
```

10. 创建缺货单

create table Shortage

```
(
    ShortageNo     varchar2(20),
    GoodsNo        varchar2(20),
    ShortageDate   date,
    constraint PK_ShortageNo primary key (ShortageNo),
    constraint FK_ShoGoodsNo foreign key (GoodsNo)
        references Goods(GoodsNo)
);
```

11. 创建采购单

create table Reserve

```
(
    ReserveNo      varchar2(20),
    GoodsNo        varchar2(20),
    ReserveNum     int,
    ReserveDate    date,
    constraint PK_ReserveNo primary key (ReserveNo),
    constraint FK_ResGoodsNo foreign key (GoodsNo)
        references Goods(GoodsNo),
    constraint CK_ReserveNum check (ReserveNum>0)
);
```

12. 创建库存单

```
create table Stock
(
    StockNo      varchar2(20),
    GoodsNo      varchar2(20),
    StockNumber  int,
    constraint PK_StockNo primary key (StockNo),
    constraint FK_StoGoodsNo foreign key (GoodsNo)
        references Goods(GoodsNo),
    constraint CK_StockNumber check (StockNumber>0)
);
```

13. 创建出库单

```
create table Delivery
(
    DeliveryNo   varchar2(20),
    GoodsNo      varchar2(20),
    DeliveryNum  int,
    DeliveryReason varchar2(20),
    DeliveryTime date,
    constraint PK_DeliveryNo primary key (DeliveryNo),
    constraint FK_DelGoodsNo foreign key (GoodsNo)
        references Goods(GoodsNo),
    constraint CK_DeliveryNum check (DeliveryNum>0)
);
```

14. 创建入库单

```
create table Warehousing
(
    WareNo      varchar2(20),
    GoodsNo      varchar2(20),
    WareNumber  int,
    WareOrigin  varchar2(20),
```

```
WareTime      date,
constraint PK_WareNo primary key (WareNo),
constraint FK_WarGoodsNo foreign key (GoodsNo)
      references Goods(GoodsNo),
constraint CK_WareNumber check (WareNumber>0)
);
```

15. 创建进货单

```
create table Purchase
(
  PurchaseNo      varchar2(20),
  PurchaseAmount  float,
  PurchaseNum     int,
  PurchaseDate    date,
  constraint PK_PurchaseNo primary key (PurchaseNo),
  constraint CK_PurchaseAmount check (PurchaseAmount>=0),
  constraint CK_PurchaseNum check (PurchaseNum>0)
);
```

16. 创建进货细则单

```
create table PurRecord
(
  PurRecordNo     varchar2(20),
  PurchaseNo      varchar2(20),
  GoodsNo         varchar2(20),
  PurRecordNum    int,
  constraint PK_PurRecordNo primary key (PurRecordNo),
  constraint FK_PurPurchaseNo foreign key (PurchaseNo)
      references Purchase(PurchaseNo),
  constraint FK_PurGoodsNo foreign key (GoodsNo)
      references Goods(GoodsNo),
  constraint CK_PurRecordNum check (PurRecordNum>0));
```


5. 1. 2 创建视图

1.商品销售情况

```
create or replace view V_SALES (商品编号, 总销售量)
as select Goods.GoodsNo, sum(SaleRecordNum)
from Goods, SaleRecord
where Goods.GoodsNo=SaleRecord.Goodsno
group by Goods.Goodsno
order by Goods.Goodsno;
```

2.在售商品

```
create or replace view V_GOODS
(电脑编号,电脑名称,售价,电脑种类,屏幕尺寸,
CPU 型号,内存容量,硬盘容量)
as select GoodsNo, GoodsName, GoodsPrice,
Type, Screen, CPU, Memery, HardDisk
from Goods;
```

3.售后情况

```
create or replace view V_PROGRESS
(售后单号,售后类型,审核进度,维修进度,退货进度,换货进度,反馈信息)
as select Service.SerNo, Service.Sertype, SerProgress,
RepairProgress, RefundProgress, ChangeProgress, SerMessage
from SERVICE left outer join REPAIRS
on SERVICE.SERNO=REPAIRS.SERNO
left outer join REFUNDS on SERVICE.SERNO=REFUNDS.SERNO
left outer join CHANGES on SERVICE.SERNO=CHANGES.SERNO
order by Service.SerNo;
```

4.账单

--销售额=销售记录单总额

--利润=销售额-进价-维修-退货

```
create or replace view V_BILL(日期,销售额,销售利润)--总利润
as select substr(to_char(Sales.Saletime,'YYYY/MM/DD'),1,10),
sum(salerecordnum*goodsprice*salediscount),
```

```
sum(salerecordnum*goodsprice*salediscount)
-sum(salerecordnum*goodsenterprice)
from salerecord,goods,sales
where salerecord.goodsno=goods.goodsno
      and salerecord.salenos=sales.salenos
group by substr(to_char(Sales.Saletime,'YYYY/MM/DD'),1,10)
order by substr(to_char(Sales.Saletime,'YYYY/MM/DD'),1,10);
```

5.商品售后统计

```
create or replace view V_SERVICE(
    商品编号,退货数量,换货数量,维修次数)
as
select goods.goodsno,
       sum(case when sertype='退货' and serprogress='审核通过'
                then 1 else 0 end),
       sum(case when sertype='换货' and serprogress='审核通过'
                then 1 else 0 end),
       sum(case when sertype='维修' and serprogress='审核通过'
                then 1 else 0 end)
from
       goods left join salerecord on goods.goodsno=salerecord.goodsno
       left join service on service.salerecordno=salerecord.salerecordno
group by goods.goodsno
order by goods.goodsno;
```

6.购物发票

```
create or replace view V_INVOICE(
    销售单编号,客户姓名,消费金额,销售日期)
as
select sales.salenos,customer.cusname,
       sales.saledistotal,sales.saletime
from sales left join customer
       on sales.cusno=customer.cusno
```

order by sales.salenno;

5.1.3 创建索引

1.建立商品表商品名称索引

```
create index IND_GOODSNAME on GOODS (GOODSNAME);
```

2.建立库存表商品编号索引

```
create unique index IND_STOCK on STOCK (GOODSNO);
```

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
销售	PRO_SALE	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
更新商品	PRO_UPGOODS	商品信息及功能	查询商品信息，进行相应的增改操作
更新客户	PRO_UPCUS	顾客信息及功能	查询顾客信息，进行相应的增改操作

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;
      elseif cuslevel='金卡' then discount:=0.98;
      elseif 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.salenos=salerecord.salenos
    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.changenno%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(changenno,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

	CUSNO	CUSNAME	PHONE	POINTS	CUSLEVEL	BIRTH	GENDER
1	CR0001	刘宝国	15891797387	3999	普通	1988/10/23	男
2	CR0002	刘畅	13289895959	16998	金卡	1995/2/16	女
3	CR0003	郭娜	18681818185	4399	普通	1985/4/3	女
4	CR0004	于洋	17310037734	31988	钻石	2000/6/12	女
5	CR0005	惠敏	15686093068	8198	银卡	1981/9/30	女
6	CR0006	彭茹	15353535861	16896	金卡	1999/5/9	女
7	CR0007	林一鹤	15877387083	3199	普通	1989/12/26	女
8	CR0008	华茜	17392950657	4899	普通	1997/2/5	女
9	CR0009	张霖锐	18829291876	10999	银卡	1986/3/24	男
10	CR0010	金天	15353535829	4398	普通	1995/9/29	男
11	CR0011	李雨欣	13289898181	16989	金卡	1993/10/26	女
12	CR0012	马宝宝	18729292381	7996	银卡	1982/11/20	男
13	CR0013	袁治军	18291912975	18999	金卡	1995/1/18	男
14	CR0014	杨勇	13289898800	5689	普通	1984/12/7	男
15	CR0015	米永海	15691859068	15889	金卡	1993/9/15	男
16	CR0016	马德顺	15353535870	26969	钻石	1986/5/18	男
17	CR0017	李桂兰	15829919086	9986	银卡	1999/12/4	女
18	CR0018	李敏	13119199898	3899	普通	2000/1/15	女
19	CR0019	郑仲冬	18729299221	19889	金卡	1989/12/6	女
20	CR0020	李朋伯	15686093068	8959	银卡	1997/2/18	男
21	CR0021	张婉婷	13991111876	19999	金卡	1986/4/25	女
22	CR0022	范红	13629283789	10298	银卡	1997/8/30	女
23	CR0023	冯雪松	13571933678	3599	普通	1995/5/16	男
24	CR0024	李世康	15029999232	23199	钻石	1994/11/24	男
25	CR0025	顾昊	18092929393	2599	普通	1999/11/5	男
26	CR0026	王琳	17691199919	16899	金卡	1989/5/19	女
27	CR0027	谭燕	17602900007	2389	普通	1991/12/7	女
28	CR0028	薛恩艺	13299002200	8999	银卡	2000/3/15	男

图 5-1 客户单

2. 商品单

	GOODSNO	GOODSNAME	GOODSENTERPRICE	GOODSPRICE	TYPE	SCREEN	CPU	MEMERY	HARDDISK
1	GD0001	华硕飞行堡垒7	7640	8499	游戏本	15.6英寸	Intel i7-9750H	16GB	512GB
2	GD0002	华硕飞行堡垒7	6280	6999	游戏本	15.6英寸	Intel i7-9750H	8GB	512GB
3	GD0003	华硕飞行堡垒6	6550	7299	游戏本	15.6英寸	Intel i7-8750H	8GB	256GB+1TB
4	GD0004	华硕飞行堡垒6	6280	6999	游戏本	15.6英寸	Intel i7-8750H	16GB	512GB+1TB
5	GD0005	华硕a豆	3590	3999	轻薄本	13.3英寸	Intel i3-8145U	4GB	512GB
6	GD0006	华硕猎鹰V4	4120	4599	一体机	23.8英寸	Intel i5-8265U	8GB	128GB
7	GD0007	华硕顽石六代FL8700F	3760	4199	常规本	15.6英寸	Intel i5-8265U	4GB	256GB
8	GD0008	华硕玩家国度 魔霸3	12010	13699	游戏本	15.6英寸	Intel i7-9750H	32GB	512GB+2TB
9	GD0009	华硕玩家国度 魔霸3	10520	11999	游戏本	15.6英寸	Intel i7-9750H	16GB	512GB
10	GD0010	华硕玩家国度 GX800	58180	67999	游戏本	18.4英寸	Intel i7-8720HK	64GB	512GB*3
11	GD0011	华硕灵耀S 2代	5230	5799	轻薄本	14.0英寸	Intel i7-8565U	8GB	256GB
12	GD0012	华硕灵耀Deluxe14	6480	7199	轻薄本	14.0英寸	Intel i5-8256U	8GB	512GB
13	GD0013	华硕灵耀Deluxe14s	7410	8199	轻薄本	14.0英寸	Intel i7-10510U	8GB	512GB
14	GD0014	华硕灵耀X2 Pro	27910	31999	轻薄本	15.6英寸	Intel i9-9980HK	32GB	1TB
15	GD0015	华硕傲世V4000	6090	6799	一体机	27.0英寸	Intel i7-8550U	8GB	128GB+1TB
16	GD0016	华硕破晓7	3890	4299	轻薄本	15.6英寸	Intel i5-8265U	8GB	256GB+1TB
17	GD0017	华硕Mars15	7080	7799	轻薄本	15.6英寸	Intel i7-9750H	8GB	512GB
18	GD0018	华硕VivoBook14	5010	5599	轻薄本	14.0英寸	Intel i5-8256U	8GB	512GB
19	GD0019	华硕VivoBook15s	4850	5399	轻薄本	15.6英寸	Intel i5-10210U	8GB	512GB
20	GD0020	华硕顽石Y4200D	3420	3799	轻薄本	14.0英寸	Ryzen 5 3500U	8GB	256GB
21	GD0021	华硕K505	3890	4299	常规本	15.6英寸	AMD A9-9420P	8GB	128GB+512GB
22	GD0022	华硕X505	2530	2799	常规本	15.6英寸	AMD A6-9225	4GB	512GB
23	GD0023	华硕E203	2680	2999	常规本	11.6英寸	Intel N3350	4GB	128GB+256GB
24	GD0024	华硕Y5200	4560	5099	轻薄本	15.6英寸	Intel i5-8265U	12GB	512GB+1TB
25	GD0025	华硕顽石FL8700	3490	3899	轻薄本	15.6英寸	Ryzen 5 3500U	8GB	512GB
26	GD0026	华硕V241	4410	4899	一体机	23.8英寸	Intel i5-8265U	8GB	256GB+1TB
27	GD0027	华硕V4000	3690	4099	一体机	21.5英寸	Intel i3-8130U	8GB	256GB
28	GD0028	华硕灵耀TP401	3760	4199	轻薄本	14.0英寸	Intel m3-7Y30	4GB	128GB+1TB

图 5-2 商品单

3. 销售单

	SALENO	CUSNO	SALEDISTOTAL	SALETOTAL	SALEDISCOUNT	SALETIME
1	SA0036	CR0020	17396	17396	1	2019/12/26
2	SA0037	CR0020	14597	14597	1	2019/12/26
3	SA0001	CR0001	36298	36298	1	2019/12/1
4	SA0002	CR0021	5291.02	5399	0.98	2019/12/3
5	SA0003	CR0013	8230.04	8398	0.98	2019/12/4
6	SA0004	CR0011	3821.02	3899	0.98	2019/12/5
7	SA0005	CR0004	5625.03	5799	0.97	2019/12/5
8	SA0006	CR0003	8199	8199	1	2019/12/6
9	SA0007	CR0012	16828.02	16998	0.99	2019/12/7
10	SA0008	CR0019	11759.02	11999	0.98	2019/12/7
11	SA0009	CR0024	5431.03	5599	0.97	2019/12/8
12	SA0010	CR0006	74282.04	75798	0.98	2019/12/9
13	SA0011	CR0008	4199	4199	1	2019/12/9
14	SA0012	CR0009	5741.01	5799	0.99	2019/12/12
15	SA0013	CR0018	5798	5798	1	2019/12/15
16	SA0014	CR0023	17897	17897	1	2019/12/15
17	SA0015	CR0016	6983.03	7199	0.97	2019/12/15
18	SA0016	CR0027	3799	3799	1	2019/12/15
19	SA0017	CR0014	13699	13699	1	2019/12/16
20	SA0018	CR0001	7998	7998	1	2019/12/16
21	SA0019	CR0018	13998	13998	1	2019/12/16
22	SA0020	CR0011	15579.06	15897	0.98	2019/12/16
23	SA0021	CR0002	7446.04	7598	0.98	2019/12/18
24	SA0022	CR0028	3959.01	3999	0.99	2019/12/18
25	SA0023	CR0024	4073.03	4199	0.97	2019/12/18
26	SA0024	CR0017	10393.02	10498	0.99	2019/12/19
27	SA0025	CR0026	17246.04	17598	0.98	2019/12/20
28	SA0026	CR0017	5048.01	5099	0.99	2019/12/23
29	SA0027	CR0022	11482.02	11598	0.99	2019/12/24
30	SA0028	CR0025	7299	7299	1	2019/12/24
31	SA0029	CR0010	8199	8199	1	2019/12/25
32	SA0030	CR0005	11879.01	11999	0.99	2019/12/25
33	SA0031	CR0003	8598	8598	1	2019/12/26
34	SA0032	CR0007	4199	4199	1	2019/12/26
35	SA0033	CR0009	4850.01	4899	0.99	2019/12/26

图 5-3 销售记录单

4. 销售细则

	SALERECORDNO	SALENO	GOODSNO	SALERECORDNUM
1	SR0045	SA0035	GD0006	1
2	SR0046	SA0035	GD0022	2
3	SR0047	SA0035	GD0012	1
4	SR0048	SA0036	GD0006	1
5	SR0049	SA0036	GD0022	1
6	SR0050	SA0036	GD0012	1
7	SR0001	SA0001	GD0016	1
8	SR0002	SA0001	GD0014	1
9	SR0003	SA0002	GD0019	1
10	SR0004	SA0003	GD0028	2
11	SR0005	SA0004	GD0025	1
12	SR0006	SA0005	GD0011	1
13	SR0007	SA0006	GD0013	1
14	SR0008	SA0007	GD0001	1
15	SR0009	SA0007	GD0001	1
16	SR0010	SA0008	GD0009	1
17	SR0011	SA0009	GD0018	1
18	SR0012	SA0010	GD0010	1
19	SR0013	SA0010	GD0017	1
20	SR0014	SA0011	GD0007	1
21	SR0015	SA0012	GD0011	1
22	SR0016	SA0013	GD0022	1
23	SR0017	SA0013	GD0023	1
24	SR0018	SA0014	GD0027	1
25	SR0019	SA0014	GD0015	1
26	SR0020	SA0014	GD0002	1
27	SR0021	SA0015	GD0012	1
28	SR0022	SA0016	GD0020	1
29	SR0023	SA0017	GD0008	1
30	SR0024	SA0018	GD0005	1
31	SR0025	SA0018	GD0005	1
32	SR0026	SA0019	GD0004	2
33	SR0027	SA0020	GD0004	1
34	SR0028	SA0020	GD0021	1

图 5 -4 销售细则单

5. 出库单

	DELIVERYNO	GOODSNO	DELIVERYNUM	DELIVERYREASON	DELIVERYTIME
1	DE0068	GD0027	1	换货	2019/12/27
2	DE0058	GD0006	1	销售	2019/12/26
3	DE0059	GD0022	2	销售	2019/12/26
4	DE0060	GD0012	1	销售	2019/12/26
5	DE0061	GD0006	1	销售	2019/12/26
6	DE0062	GD0022	1	销售	2019/12/26
7	DE0063	GD0012	1	销售	2019/12/26
8	DE0064	GD0006	1	销售	2019/12/27
9	DE0065	GD0012	1	销售	2019/12/27
10	DE0066	GD0011	1	销售	2019/12/27
11	DE0067	GD0015	1	销售	2019/12/27
12	DE0001	GD0016	1	销售	2019/12/1
13	DE0002	GD0014	1	销售	2019/12/1
14	DE0003	GD0019	1	销售	2019/12/3
15	DE0004	GD0028	2	销售	2019/12/4
16	DE0005	GD0025	1	销售	2019/12/5
17	DE0006	GD0011	1	销售	2019/12/5
18	DE0007	GD0013	1	销售	2019/12/6
19	DE0008	GD0001	1	换货	2019/12/7
20	DE0009	GD0001	1	销售	2019/12/7
21	DE0010	GD0009	1	销售	2019/12/7
22	DE0011	GD0018	1	销售	2019/12/8
23	DE0012	GD0010	1	销售	2019/12/9
24	DE0013	GD0017	1	销售	2019/12/9
25	DE0014	GD0007	1	销售	2019/12/9
26	DE0015	GD0011	1	销售	2019/12/12
27	DE0016	GD0020	1	换货	2019/12/15
28	DE0017	GD0015	1	换货	2019/12/15
29	DE0018	GD0022	1	销售	2019/12/15
30	DE0019	GD0023	1	销售	2019/12/15
31	DE0020	GD0027	1	销售	2019/12/15
32	DE0021	GD0015	1	销售	2019/12/15
33	DE0022	GD0002	1	销售	2019/12/15
34	DE0023	GD0012	1	销售	2019/12/15
35	DE0024	GD0020	1	销售	2019/12/15

图 5-5 出库单

6. 入库单

	WARENO	GOODSNO	WARENUMBER	WAREORIGIN	WARETIME
1	WH0001	GD0014	1	退货	2019/12/1
2	WH0002	GD0011	1	退货	2019/12/5
3	WH0003	GD0001	1	返厂	2019/12/7
4	WH0004	GD0003	8	进货	2019/12/7
5	WH0005	GD0022	4	进货	2019/12/8
6	WH0006	GD0006	5	进货	2019/12/9
7	WH0007	GD0007	1	退货	2019/12/9
8	WH0008	GD0007	10	进货	2019/12/11
9	WH0009	GD0009	6	进货	2019/12/11
10	WH0010	GD0011	9	进货	2019/12/12
11	WH0011	GD0009	4	进货	2019/12/12
12	WH0012	GD0016	5	进货	2019/12/12
13	WH0013	GD0019	8	进货	2019/12/14
14	WH0014	GD0026	10	进货	2019/12/14
15	WH0015	GD0002	3	进货	2019/12/14
16	WH0016	GD0020	1	返厂	2019/12/15
17	WH0017	GD0015	1	返厂	2019/12/15
18	WH0018	GD0014	5	进货	2019/12/15
19	WH0019	GD0004	1	返厂	2019/12/16
20	WH0020	GD0004	1	返厂	2019/12/16
21	WH0021	GD0008	1	返厂	2019/12/16
22	WH0022	GD0019	9	进货	2019/12/16
23	WH0023	GD0021	1	退货	2019/12/16
24	WH0024	GD0005	1	退货	2019/12/16
25	WH0025	GD0015	10	进货	2019/12/17
26	WH0026	GD0028	1	返厂	2019/12/18
27	WH0027	GD0020	1	返厂	2019/12/18
28	WH0028	GD0001	5	进货	2019/12/18
29	WH0029	GD0018	8	进货	2019/12/18
30	WH0030	GD0024	6	进货	2019/12/18
31	WH0031	GD0005	1	退货	2019/12/18
32	WH0032	GD0004	8	进货	2019/12/20
33	WH0033	GD0013	5	进货	2019/12/20
34	WH0034	GD0008	1	退货	2019/12/20

图 5-6 入库单

7. 库存单

	STOCKNO	GOODSNO	STOCKNUMBER
1	ST0001	GD0001	2
2	ST0002	GD0002	1
3	ST0003	GD0003	8
4	ST0004	GD0004	5
5	ST0005	GD0005	5
6	ST0006	GD0006	1
7	ST0007	GD0007	10
8	ST0008	GD0008	4
9	ST0009	GD0009	8
10	ST0010	GD0010	5
11	ST0011	GD0011	5
12	ST0012	GD0012	6
13	ST0013	GD0013	1
14	ST0014	GD0014	5
15	ST0015	GD0015	8
16	ST0016	GD0016	7
17	ST0017	GD0017	9
18	ST0018	GD0018	6
19	ST0019	GD0019	16
20	ST0020	GD0020	5
21	ST0021	GD0021	10
22	ST0022	GD0022	0
23	ST0023	GD0023	7
24	ST0024	GD0024	5
25	ST0025	GD0025	8
26	ST0026	GD0026	17
27	ST0027	GD0027	4
28	ST0028	GD0028	2

图 5-7 库存单

8. 缺货单

SHORTAGENO	GOODSNO	SHORTAGEDATE
S00001	GD0022	2019/12/8
S00002	GD0006	2019/12/9
S00003	GD0007	2019/12/12
S00004	GD0011	2019/12/15
S00005	GD0016	2019/12/15
S00006	GD0014	2019/12/16
S00007	GD0026	2019/12/18
S00008	GD0015	2019/12/20
S00009	GD0001	2019/12/23
S00010	GD0013	2019/12/25
S00011	GD0016	2019/12/26
S00012	GD0024	2019/12/26
S00013	GD0017	2019/12/29
S00014	GD0026	2019/12/30
S00015	GD0001	2019/12/26
S00016	GD0001	2019/12/26
S00017	GD0022	2019/12/26
S00018	GD0022	2019/12/27

图 5-8 缺货单

9. 采购单

	RESERVENO	GOODSNO	RESERVENUM	RESERVEDATE
1	RV0001	GD0003	8	2019/12/1
2	RV0002	GD0022	4	2019/12/2
3	RV0003	GD0006	5	2019/12/3
4	RV0004	GD0007	10	2019/12/7
5	RV0005	GD0009	6	2019/12/8
6	RV0006	GD0011	9	2019/12/8
7	RV0007	GD0009	4	2019/12/9
8	RV0008	GD0016	5	2019/12/11
9	RV0009	GD0019	8	2019/12/11
10	RV0010	GD0026	10	2019/12/12
11	RV0011	GD0002	3	2019/12/12
12	RV0012	GD0014	5	2019/12/14
13	RV0013	GD0019	9	2019/12/14
14	RV0014	GD0015	10	2019/12/15
15	RV0015	GD0001	5	2019/12/15
16	RV0016	GD0018	8	2019/12/16
17	RV0017	GD0024	6	2019/12/17
18	RV0018	GD0004	8	2019/12/17
19	RV0019	GD0013	5	2019/12/18
20	RV0020	GD0016	5	2019/12/20
21	RV0021	GD0017	10	2019/12/20
22	RV0022	GD0026	8	2019/12/22
23	RV0023	GD0027	6	2019/12/23
24	RV0024	GD0028	5	2019/12/25
25	RV0025	GD0005	6	2019/12/25
26	RV0026	GD0012	10	2019/12/26
27	RV0027	GD0023	8	2019/12/26
28	RV0028	GD0008	5	2019/12/29
29	RV0029	GD0021	10	2019/12/29

图 5-9 采购单

10. 进货单

	PURCHASENO	PURCHASEAMOUNT	PURCHASENUM	PURCHASEDATE
1	PU0001	131800	8	2019/12/7
2	PU0002	25300	4	2019/12/8
3	PU0003	73440	5	2019/12/9
4	PU0004	231040	16	2019/12/11
5	PU0005	199890	13	2019/12/12
6	PU0006	46680	5	2019/12/12
7	PU0007	256020	21	2019/12/14
8	PU0008	509220	5	2019/12/15
9	PU0009	72750	9	2019/12/16
10	PU0010	115710	10	2019/12/17
11	PU0011	238100	19	2019/12/18
12	PU0012	240580	13	2019/12/20
13	PU0013	58350	5	2019/12/22
14	PU0014	204280	18	2019/12/23
15	PU0015	62180	17	2019/12/25
16	PU0016	85520	18	2019/12/26
17	PU0017	61550	5	2019/12/29
18	PU0018	38900	10	2019/12/29
19	PU0019	411020	24	2019/12/30
20	PU0021	386130	11	2019/12/27
21	PU0020	10170	13	2019/12/27

图 5-10 进货单

11. 进货细则

	PURRECORDNO	PURCHASENO	GOODSNO	PURRECORDNUM
1	PR0001	PU0001	GD0003	8
2	PR0002	PU0002	GD0022	4
3	PR0003	PU0003	GD0006	5
4	PR0004	PU0004	GD0007	10
5	PR0005	PU0004	GD0009	6
6	PR0006	PU0005	GD0011	9
7	PR0007	PU0005	GD0009	4
8	PR0008	PU0006	GD0016	5
9	PR0009	PU0007	GD0019	8
10	PR0010	PU0007	GD0026	10
11	PR0011	PU0007	GD0002	3
12	PR0012	PU0008	GD0014	5
13	PR0013	PU0009	GD0019	9
14	PR0014	PU0010	GD0015	10
15	PR0015	PU0011	GD0001	5
16	PR0016	PU0011	GD0018	8
17	PR0017	PU0011	GD0024	6
18	PR0018	PU0012	GD0004	8
19	PR0019	PU0012	GD0013	5
20	PR0020	PU0013	GD0016	5
21	PR0021	PU0014	GD0017	10
22	PR0022	PU0014	GD0026	8
23	PR0023	PU0015	GD0027	6
24	PR0024	PU0015	GD0028	5
25	PR0025	PU0015	GD0005	6
26	PR0026	PU0016	GD0012	10
27	PR0027	PU0016	GD0023	8
28	PR0028	PU0017	GD0008	5
29	PR0029	PU0018	GD0021	10
30	PR0030	PU0019	GD0010	6

图 5-11 进货细则单

12. 售后单

	SERNO	SALEREORDNO	SERTYPE	SERREASON	SERDATE	SERPROGRESS	SERMESAGE
1	SE0055	SR0015	退货	对电脑性能不满意	2019/12/27	审核未通过	电脑超过质保期, 不享受退货服务
2	SE0056	SR0018	换货	电脑键盘受损	2019/12/27	审核通过	换货成功, 已更换新机
3	SE0001	SR0007	维修	屏幕故障	2019/12/1	审核通过	维修完毕, 电脑可正常使用
4	SE0002	SR0029	维修	开机时有异响	2019/12/3	审核通过	维修完毕, 电脑可正常使用
5	SE0003	SR0002	退货	电脑外观不满意	2019/12/4	审核通过	退货成功, 已退还购买费用
6	SE0004	SR0004	维修	电脑外壳开裂	2019/12/5	审核通过	维修完毕, 电脑可正常使用
7	SE0005	SR0022	换货	系统故障	2019/12/5	审核通过	换货成功, 已更换新机
8	SE0006	SR0025	换货	电脑系统蓝屏	2019/12/6	审核未通过	电脑超过质保期, 不享受换货服务
9	SE0007	SR0039	退货	买错型号	2019/12/7	审核通过	退货成功, 已退还购买费用
10	SE0008	SR0032	换货	开机时有异响	2019/12/7	审核通过	换货成功, 已更换新机
11	SE0009	SR0009	换货	散热器无法正常开启	2019/12/8	审核通过	换货成功, 已更换新机
12	SE0010	SR0040	维修	散热器无法正常开启	2019/12/9	审核通过	维修完毕, 电脑可正常使用
13	SE0011	SR0037	换货	电脑无法开机	2019/12/12	审核未通过	系统已激活, 不享受退货服务
14	SE0012	SR0036	退货	买错型号	2019/12/15	审核通过	退货成功, 已退还购买费用
15	SE0013	SR0026	换货	系统故障	2019/12/15	审核通过	换货成功, 已更换新机
16	SE0014	SR0027	换货	电脑无法开机	2019/12/15	审核通过	换货成功, 已更换新机
17	SE0015	SR0016	退货	电脑硬盘无法读取	2019/12/15	审核未通过	电脑硬盘受损, 不享受退货服务
18	SE0016	SR0028	退货	7天无理由退货	2019/12/16	审核通过	退货成功, 已退还购买费用
19	SE0017	SR0021	维修	散热器无法正常开启	2019/12/16	审核通过	维修完毕, 电脑可正常使用
20	SE0018	SR0005	维修	光驱无法使用	2019/12/16	审核通过	维修完毕, 电脑可正常使用
21	SE0019	SR0031	退货	系统故障	2019/12/16	审核通过	退货成功, 已退还购买费用
22	SE0020	SR0019	换货	电脑无法开机	2019/12/18	审核通过	换货成功, 已更换新机
23	SE0021	SR0017	维修	电脑无法开机	2019/12/18	审核未通过	电脑进水, 人为损坏不保修
24	SE0022	SR0024	退货	电脑性能不足	2019/12/18	审核通过	退货成功, 已退还购买费用
25	SE0023	SR0023	换货	15天无理由换货	2019/12/19	审核通过	换货成功, 已更换新机
26	SE0024	SR0018	维修	开机时有异响	2019/12/20	审核通过	维修完毕, 电脑可正常使用
27	SE0025	SR0012	维修	触摸板故障	2019/12/23	审核通过	维修完毕, 电脑可正常使用
28	SE0026	SR0006	退货	买错型号	2019/12/24	审核通过	退货成功, 已退还购买费用
29	SE0027	SR0030	换货	15天无理由换货	2019/12/24	审核通过	换货成功, 已更换新机
30	SE0028	SR0003	维修	系统故障	2019/12/25	审核通过	维修完毕, 电脑可正常使用

图 5-12 售后记录单

13. 维修单

	REPAIRNO	SERNO	REPAIRDATE	REPAIRAMOUNT	REPAIRPROGRESS
1	RP0001	SE0001	2019/12/1	60	维修完毕
2	RP0002	SE0002	2019/12/3	50	维修完毕
3	RP0003	SE0004	2019/12/5	100	维修完毕
4	RP0004	SE0010	2019/12/9	0	维修完毕
5	RP0005	SE0017	2019/12/16	0	维修完毕
6	RP0006	SE0018	2019/12/16	30	维修完毕
7	RP0007	SE0024	2019/12/20	40	维修完毕
8	RP0008	SE0025	2019/12/23	26	维修完毕
9	RP0009	SE0028	2019/12/25	60	维修完毕
10	RP0010	SE0033	2019/12/30	95	维修完毕
11	RP0011	SE0036	2019/12/27	64.3	维修完毕

图 5-13 维修单

14. 退货单

REFUNDNO	SERNO	REFUNDDATE	REFUNDAMOUNT	REFUNDPROGRESS
RD0001	SE0003	2019/12/1	31999	退货成功
RD0002	SE0007	2019/12/25	7226.01	退货成功
RD0003	SE0012	2019/12/20	13425.02	退货成功
RD0004	SE0016	2019/12/16	4213.02	退货成功
RD0005	SE0019	2019/12/18	3959.01	退货成功
RD0006	SE0022	2019/12/16	3999	退货成功
RD0007	SE0026	2019/12/5	5625.03	退货成功
RD0008	SE0030	2019/12/29	4157.01	退货成功
RD0009	SE0032	2019/12/9	4199	退货成功
RD0010	SE0035	2019/12/27	7199	退货成功

图 5-14 退货单

15. 换货单

	CHANGENO	SERNO	CHANGEDATE	CHANGEPROGRESS
1	CH0011	SE0056	2019/12/27	换货成功
2	CH0001	SE0009	2019/12/7	换货成功
3	CH0002	SE0005	2019/12/15	换货成功
4	CH0003	SE0020	2019/12/15	换货成功
5	CH0004	SE0013	2019/12/16	换货成功
6	CH0005	SE0014	2019/12/16	换货成功
7	CH0006	SE0023	2019/12/16	换货成功
8	CH0007	SE0008	2019/12/18	换货成功
9	CH0008	SE0027	2019/12/18	换货成功
10	CH0009	SE0029	2019/12/25	换货成功
11	CH0010	SE0031	2019/12/26	换货成功

图 5-15 换货单

14. 返厂单

	RETURNNO	SERNO	RETURNDATE
1	RN0011	SE0056	2019/12/27
2	RN0001	SE0005	2019/12/15
3	RN0002	SE0008	2019/12/18
4	RN0003	SE0009	2019/12/7
5	RN0004	SE0013	2019/12/16
6	RN0005	SE0014	2019/12/16
7	RN0006	SE0020	2019/12/15
8	RN0007	SE0023	2019/12/16
9	RN0008	SE0027	2019/12/18
10	RN0009	SE0029	2019/12/25
11	RN0010	SE0031	2019/12/26

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。

	CUSNO	CUSNAME	PHONE	POINTS	CUSLEVEL	BIRTH	GENDER
28	CR0028	薛恩艺	13299002200	8999	银卡	2000/3/15	男
27	CR0027	谭燕	17602900007	2389	普通	1991/12/7	女
26	CR0026	王琳	17691199919	16899	金卡	1989/5/19	女

图 5-17 客户表原始数据

```

declare
    birth date:=to_date('2000/06/12','yyyy/mm/dd');
begin
    pro_upcus('增加','','赵志新','13510010000',birth,'男');
    pro_upcus('修改','CR0028','','13500000000',null,'');

```

图 5-18 客户信息增改测试语句

	CUSNO	CUSNAME	PHONE	POINTS	CUSLEVEL	BIRTH	GENDER
27	CR0027	谭燕	17602900007	2389	普通	1991/12/7	女
28	CR0028	薛恩艺	13500000000	8999	银卡	2000/3/15	男
29	CR0029	赵志新	13510010000	0	普通	2000/6/12	男

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。

GOODSNO	GOODSNAME	GOODSENTERPRICE	GOODSPRICE	TYPE	SCREEN
GD0027	华硕V4000	3690	4099	一体机	21.5英寸
GD0028	华硕灵耀TP401	3760	4199	轻薄本	14.0英寸

图 5-20 商品表原始数据

```

begin
    pro_upgoods('增加', null, '华硕飞行堡垒5', 5299, 4750, '游戏本',
        '15.6英寸', 'Intel i5-8256U', '8GB', '512GB');
    pro_upgoods('修改', 'GD0028', '', 3999, 3510, '', '', '', '', '');
end;|

```

图 5-21 商品信息增改测试语句

	GOODSNO	GOODSNAME	GOODSENTERPRICE	GOODSPRICE	TYPE	SCREEN	CPU
1	GD0029	华硕飞行堡垒5	5299	4750	游戏本	15.6英寸	Intel i5-8256U
29	GD0028	华硕灵耀TP401	3510	3999	轻薄本	14.0英寸	Intel m3-7Y30
28	GD0027	华硕V4000	3600	4000	一体机	21.5英寸	Intel i3-8130U

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。

	SERNO	SALERECDNO	SERTYPE	SERREASON	SERDATE	SERPROGRESS	SERMESSAGE
10	SE0067	SR0026	维修	无法开机	2019/12/29	审核未通过	主板进水，人为损坏
9	SE0066	SR0025	维修	键盘松动	2019/12/29	审核通过	维修完毕，电脑可正常使用
8	SE0065	SR0026	维修	无法开机	2019/12/29	审核未通过	主板进水，人为损坏

图 5-23 售后记表原始数据

```

begin
    pro_service('SR0005', '退货', '性能不满意', '检查通过', '');
    --预期结果：超过7天，不能退货

    dbms_output.put_line(' ');
    pro_service('SR0025', '换货', '屏幕显示有问题', '检查通过', '');
    --预期结果：成功换货

    dbms_output.put_line(' ');
    pro_service('SR0030', '维修', '外壳开裂', '检查通过', '');
    --预期结果：成功维修

    dbms_output.put_line(' ');
    pro_service('SR0045', '维修', '电脑漏电', '检查未通过', '主板进水，人为损坏');
    --预期结果：不予维修
end;

```

图 5-24 售后测试语句

SQL 输出 统计

清除 缓冲区大小 10000 ☒ 启用

您的电脑超过质保期，不再享受退货服务，请谅解
本次售后结束，感谢光临！

您的电脑已成功更换
本次售后结束，感谢光临！

您的电脑已维修完毕
本次售后结束，感谢光临！

您的售后申请未通过审核，抱歉
本次售后结束，感谢光临！

图 5-25 售后测试输出

	SERNO	SALERECDNO	SERTYPE	SERREASON	SERDATE	SERPROGRESS	SERMESSAGE
14	SE0071	SR0045	维修	电脑漏电	2019/12/29	审核未通过	主板进水，人为损坏
13	SE0070	SR0030	维修	外壳开裂	2019/12/29	审核通过	维修完毕，电脑可正常使用
12	SE0069	SR0025	换货	屏幕显示有问题	2019/12/29	审核通过	换货成功，已更换新机
11	SE0068	SR0005	退货	性能不满意	2019/12/29	审核未通过	电脑超过质保期，不享受退货服务
10	SE0067	SR0026	维修	无法开机	2019/12/29	审核未通过	主板进水，人为损坏

图 5-26 售后记录表测试后数据

	CHANGENO	SERNO	CHANGEDATE	CHANGEPROGRESS
2	CH0016	SE0069	2019/12/29	换货成功
1	CH0015	SE0062	2019/12/29	换货成功

图 5-27 换货单测试后数据

	DELIVERYNO	GOODSNO	DELIVERYNUM	DELIVERYREASON	DELIVERYTIME
1	DE0068	GD0027	1	换货	2019/12/27
11	DE0067	GD0015	1	销售	2019/12/27

图 5-28 出库单测试后数据

	RETURNNO	SERNO	RETURNDATE
3	RN0016	SE0069	2019/12/29
2	RN0015	SE0062	2019/12/29

图 5-29 返厂单测试后数据

	REPAIRNO	SERNO	REPAIRDATE	REPAIRAMOUNT	REPAIRPROGRESS
4	RP0015	SE0070	2019/12/29	92.53	维修完毕
3	RP0014	SE0066	2019/12/29	1.81	维修完毕
2	RP0013	SE0064	2019/12/29	57.7	维修完毕

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：

	SALENO	CUSNO	SALEDISTOTAL	SALETOTAL	SALEDISCOUNT	SALETIME
38	SA0039	CR0015	12598	12598	1	2019/12/27
37	SA0038	CR0020	11798	11798	1	2019/12/27

图 5-31 销售单测试前数据

	SALERECORDNO	SALENO	GOODSNO	SALERECORDNUM
53	SR0054	SA0039	GD0015	1
52	SR0053	SA0039	GD0011	1
51	SR0052	SA0039	GD0010	1

图-32 销售细则测试前数据

	CUSNO	CUSNAME	PHONE	POINTS	CUSLEVEL	BI
2	CR0001	刘宝国	15891797387	3999	普通	19
3	CR0002	刘畅	13289895959	16998	金卡	19
4	CR0003	郭耀	13331313131	1000	普通	19

图 5-33 客户单测试前数据

```
select * from STOCK
where goodsno='GD0009' or goodsno='GD0010'
or goodsno='GD0014' or goodsno='GD0001';
```

	STOCKNO	GOODSNO	STOCKNUMBER
1	ST0001	GD0001	1
2	ST0009	GD0009	4
3	ST0010	GD0010	1
4	ST0014	GD0014	3

图 5-34 库存单测试前数据

```
begin
  pro_sale('CR0001', 3, 'GD0009:2, GD0010:2, GD0014:1, GD');
  --预期结果：销售失败（库存不足）
  --GD0010号商品添加单缺货单

  dbms_output.put_line('');
  pro_sale('CR0002', 1, 'GD0001:1');
  --预期结果：销售成功
  --GD0001号商品被添加到采购单（采购触发器）
  --客户CR0002等级被更新为钻石（积分等级触发器）
end;
```

图 5-35 销售测试语句

	SALENO	CUSNO	SALEDISTOTAL	SALETOTAL	SALEDISCOUNT	SALETIME
1	SA0043	CR0002	8329.02	8499	0.98	2019/12/29
39	SA0039	CR0015	12598	12598	1	2019/12/27

图 5-36 销售单测试后数据

	SALERECDNO	SALENO	GOODSNO	SALERECDNUM
1	SR0055	SA0043	GD0001	1
54	SR0054	SA0039	GD0015	1

图 5-37 销售细则单测试后数据

```
select * from STOCK
where goodsno='GD0009' or goodsno='GD0010'
or goodsno='GD0014' or goodsno='GD0001';
```

	STOCKNO	GOODSNO	STOCKNUMBER
1	ST0001	GD0001	0
2	ST0009	GD0009	4
3	ST0010	GD0010	1
4	ST0014	GD0014	3

图 5-38 库存单测试后数据

	RESERVENO	GOODSNO	RESERVENUM	RESERVEDATE
15	RV0015	GD0001	5	2019/12/15
14	RV0014	GD0015	10	2019/12/15

图 5-39 采购单测试后数据

5. 进货存储过程测试:

	STOCKNO	GOODSNO	STOCKNUMBER
3	ST0003	GD0003	8
4	ST0004	GD0004	5

图 5-40 库存单测试前数据

	PURCHASENO	PURCHASEAMOUNT	PURCHASENUM	PURCHASEDATE
1	PU0022	88440	13	2019/12/29
21	PU0021	386130	11	2019/12/27

图 5-41 进货单测试前数据

	WARENO	GOODSNO	WARENUMBER	WAREORIGIN	WARETIME
4	WH0073	GD0002	8	进货	2019/12/29
3	WH0072	GD0001	5	进货	2019/12/29
2	WH0069	GD0005	1	返厂	2019/12/29

图 5-42 入库单测试前数据

```

begin
    pro_purchase(2, 'GD0003:5, GD0004:8');
    --预期结果：添加进货记录和进货细则
    --添加入库单，修改库存信息
end;

```

图 5-43 进货测试语句

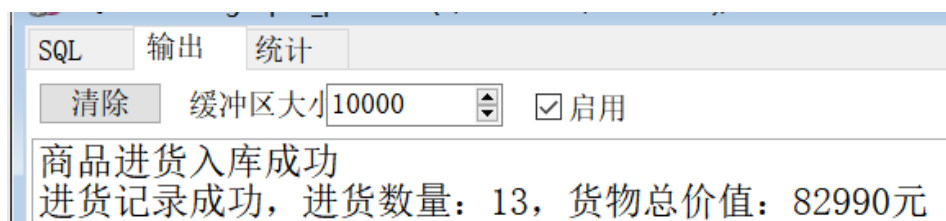


图 5-44 进货测试输出

	WARENO	GOODSNO	WARENUMBER	WAREORIGIN	WARETIME
4	WH0073	GD0002	8	进货	2019/12/29
3	WH0072	GD0001	5	进货	2019/12/29
2	WH0069	GD0005	1	返厂	2019/12/29

图 5-45 入库单测试后数据

	STOCKNO	GOODSNO	STOCKNUMBER
3	ST0003	GD0003	13
4	ST0004	GD0004	13
5	ST0005	GD0005	5

图 5-46 库存单测试后数据

	PURCHASENO	PURCHASEAMOUNT	PURCHASENUM	PURCHASEDATE
1	PU0023	82990	13	2019/12/29

图 5-47 进货单测试后数据

	PURRECORDNO	PURCHASENO	GOODSNO	PURRECORDNUM
2	PR0075	PU0023	GD0004	8
1	PR0074	PU0023	GD0003	5

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
D1-1	StuID	学生的学号		varchar	10	
D1-2	IDNumber	学生身份证号		varchar	20	
D1-3	StuName	学生的姓名		varchar	20	
D1-4	StuAge	学生的年龄		number	5	1-100
D1-5	StuBirthday	学生的出生日期		data		1900/1/1-2100/1/1
D1-6	StuImg	学生的照片		blob		
D1-7	StuPhone	学生的联系电话		varchar	20	
D1-8	StuAddress	学生的家庭住址		varchar	40	
D1-9	StuPeriod	学生届数		number	5	1900-2100
D1-10	StuClass	学生就读班级		varchar	10	
D1-11	StuSem	当前学期		number	1	{1, 2, 3, 4, 5, 6}
D1-12	FirstSem	第一学期是否就读		number	1	{0, 1}
D1-13	SecondSem	第二学期是否就读		number	1	{0, 1}
D1-14	ThirdSem	第三学期是否就读		number	1	{0, 1}
D1-15	FourSem	第四学期是否就读		number	1	{0, 1}
D1-16	FiveSem	第五学期是否就读		number	1	{0, 1}
D1-17	SixSem	第六学期是否就读		number	1	{0, 1}
D1-18	InSchool	学生是否在读		number	1	{0, 1}
D1-19	FamilyStatus	学生的家庭状况		varchar	10	{非贫困, 贫困, 特困}

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
D1-20	Career	学生毕业去向		varcahr	50	
D1-21	TeaID	老师的工号		varchar	10	
D1-22	TeaName	老师的姓名		varchar	20	
D1-23	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”}
D1-24	ClassTeacherID	班主任 ID		varchar	10	
D1-25	ClassTeacherName	班主任姓名		varchar	20	
D1-26	ClassID	班级号		varchar	10	
D1-27	ClassName	班名		varchar	20	
D1-28	ClassTeacher	班主任 ID		varchar	10	
D1-29	ClassNumber	班级人数		number	5	
D1-30	ClassGrade	班级的年级		number	1	{1, 2, 3}
D1-31	ClassPeriod	班级的届		number	5	1900-2100
D1-32	ChineseTea	班级语文老师的 ID		varchar	10	
D1-33	MathTea	班级数学老师的 ID		varchar	10	
D1-34	EnglishTea	班级英语老师的 ID		varchar	10	
D1-35	HistoryTea	班级历史老师的 ID		varchar	10	
D1-36	PliticsTea	班级政治老师的 ID		varchar	10	
D1-37	GeographyTea	班级地理老师的 ID		varchar	10	
D1-38	PhysicsTea	班级物理老师的 ID		varchar	10	
D1-39	ChemistryTea	班级化学老师的 ID		varchar	10	
D1-40	BiologyTea	班级生物老师的 ID		varchar	10	
D1-41	Subject	科目		varchar	20	{“语文”, “数学”, “英语”, “历史”, “政治”, “地理”, “物理”, “化学”, “生物”}

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
D1-42	Goal	成绩		number	5	0-100
D1-43	Semester	学期		number	1	{1, 2, 3, 4, 5, 6}
D1-44	ChineseGoal	语文成绩		number	5	0-100
D1-45	MathGoal	数学成绩		number	5	0-100
D1-46	EnglishGoal	英语成绩		number	5	0-100
D1-47	HistoryGoal	历史成绩		number	5	0-100
D1-48	PliticsGoal	政治成绩		number	5	0-100
D1-49	GeographyGoal	地理成绩		number	5	0-100
D1-50	PhysicsGoal	物理成绩		number	5	0-100
D1-51	ChemistryGoal	化学成绩		number	5	0-100
D1-52	BiologyGoal	生物成绩		number	5	0-100
			D1-40 到			
D1-53	AverageGoal	平均成绩	D1-48 求平均值	number	5	0-100
			D1-40 到			
D1-54	TotalGoal	总成绩	D1-48 求和	number	5	0-900
D1-55	GoalRank	成绩排名		number	5	
D1-56	FinCheck	完成报道		number	1	{0, 1}
D1-42	Goal	成绩		number	5	0-100
D1-43	Semester	学期		number	1	{1, 2, 3, 4, 5, 6}
D1-44	ChineseGoal	语文成绩		number	5	0-100
D1-45	MathGoal	数学成绩		number	5	0-100
D1-46	EnglishGoal	英语成绩		number	5	0-100
D1-47	HistoryGoal	历史成绩		number	5	0-100
D1-48	PliticsGoal	政治成绩		number	5	0-100
D1-49	GeographyGoal	地理成绩		number	5	0-100
D1-50	PhysicsGoal	物理成绩		number	5	0-100
D1-51	ChemistryGoal	化学成绩		number	5	0-100
D1-52	BiologyGoal	生物成绩		number	5	0-100
			D1-40 到			
D1-53	AverageGoal	平均成绩	D1-48 求平均值	number	5	0-100
			D1-40 到			
D1-54	TotalGoal	总成绩	D1-48 求和	number	5	0-900
D1-55	GoalRank	成绩排名		number	5	
D1-56	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, StuBirthday, Stulmg, StuPhone, StuAddress, StuPeriod, StuClass, Semester, InSchool, FirstSem, SecondSem, ThirdSem, FourSem, FiveSem, SixSem, Career
DS-2	Teacher	老师档案表	TealD, 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, ClassName, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TaotalGoal, GoalRank
DS-9	ClassGradeTable	班级全部学生的成绩	StuID, StuName, Semester, ChineseGoal, MathGoal, EnglishGoal, HistoryGoal, PliticsGoal, GeographyGoal, PhysicsGoal, ChemistryGoal, BiologyGoal, AverageGoal, TotalGoal
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	退货登	商品管	DS-8	5 次/月	10 次/月

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
F16	换货请求	请求 向仓库发出的换货请求	记表 换货登记表	理模块 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