**本科学生综合性实验报告**

**课程名称：数据库系统原理**

项目名称 超市仓库管理系统数据库设计

开课学期 2022 至 2023 学年 第二学期

完成时间 2023 年 06 月 23 日

**目录**

[1 需求分析 1](#_Toc138446590)

[1.1 背景 1](#_Toc138446591)

[1.1.1 面向的服务对象 2](#_Toc138446592)

[1.2 系统目标与系统边界 2](#_Toc138446593)

[1.2.1 系统目标 2](#_Toc138446594)

[1.2.2 系统边界 3](#_Toc138446595)

[1.3 功能需求分析 3](#_Toc138446596)

[1.3.1 系统的功能描述 3](#_Toc138446597)

[1.3.2 系统总体功能图 4](#_Toc138446598)

[1.3.3 数据流图 5](#_Toc138446599)

[1.4 数据需求及业务规则分析 6](#_Toc138446600)

[1.4.1 数据需求描述 6](#_Toc138446601)

[1.4.2 数据字典 7](#_Toc138446602)

[1.4.3 业务规则描述 7](#_Toc138446603)

[1.4.4 完整性约束规则描述 8](#_Toc138446604)

[2 概念设计 9](#_Toc138446605)

[2.1 确定基本实体集 9](#_Toc138446606)

[2.1.1 发现基本实体集 9](#_Toc138446607)

[2.1.2 定义基本实体集(数据字典) 9](#_Toc138446608)

[2.2 局部概念建模 12](#_Toc138446609)

[2.2.1 入库业务的局部概念建模 12](#_Toc138446610)

[2.2.2 出库业务的局部概念建模 13](#_Toc138446611)

[2.3 定义联系集 13](#_Toc138446612)

[2.3.1 联系集分析 13](#_Toc138446613)

[2.3.2 联系集属性数据字典 15](#_Toc138446614)

[2.4 概念模型(E-R图) 16](#_Toc138446615)

[3 逻辑设计 16](#_Toc138446616)

[3.1 关系数据库模式设计 16](#_Toc138446617)

[3.2 完整性约束的说明 19](#_Toc138446618)

[4 数据库应用与安全性设计 20](#_Toc138446619)

[4.1 数据库脚本 20](#_Toc138446620)

[4.2 数据库完整性设计 43](#_Toc138446621)

[4.3 数据库安全性设计 44](#_Toc138446622)

[4.3.1 角色、用户和权限设置 44](#_Toc138446623)

[4.3.2 使用视图保证数据安全 44](#_Toc138446624)

[4.4 存储过程设计 44](#_Toc138446625)

[5 个人工作 46](#_Toc138446626)

[5.1 各成员工作占比 46](#_Toc138446627)

[5.2 个人工作（组长） 46](#_Toc138446628)

[5.3 组员工作 46](#_Toc138446629)

# 1 需求分析

## 1.1 背景

超市作为零售行业的一种常见形态，是社会经济中的重要组成部分之一。近年来，超市行业的规模稳中有增。超市、大卖场是一个重要的商品流通渠道。据Euromonitor数据，2021年超市在我国商品流通渠道中占比约为12.7%，是继电商之后最大的现代零售业态。2010年之后超市的增速有所放慢，但仍保持正增长。2021年我国超市及大卖场行业规模为3.09万亿元，2016-2021年复合增长率为2.1%。

在超市营业过程中，货物的进出库管理是非常重要的环节，直接关系到超市的经营流程和效益。超市仓库管理的重要性主要表现在以下几个方面：

1. 保证库存的符合需求：在超市仓库管理中，采取合理的进货/出货管理方式，可以保证货品始终满足市场的需求，从而防止货品积压，提高货品周转率，进而实现超市的盈利。
2. 确保商品质量：超市仓库管理可以确保货品处于最佳的品质状态，从而保证顾客的权益。充分了解货品信息和品质信息，进而建立合理的检查机制，可以让货品跟进合理的流转平台，在每个节点进行合理筛选，保证了商品质量。
3. 提高管理效率：通过超市仓库管理可以有效的协调各个部门之间的工作，使得超市整个工作流程更为高效和有序。对于货品的出入库能够及时快速的反馈到仓库管理系统中，做到库存实时掌控。
4. 优化商业模式：通过仓库管理手段，超市能够更充分的了解市场需求，并进行相应的决策调整。

然而，管理仓库具有很大的挑战性，尤其是当它涉及范围广泛的产品、大量和复杂的库存管理流程时。出现缺货、积压和不准确的库存跟踪等错误的可能性很高，这可能导致收入损失、客户满意度下降和商业声誉受损。

这些问题的关键在于开发有效的仓库管理系统。该系统必须能够管理所有仓库操作，包括库存跟踪、订单履行、拣货等，确保空间和资源的有效利用。因此，利用SQL Server开发仓库管理系统具有重要的现实意义。通过自动化和简化仓库操作，可以显著减少错误并优化资源使用，从而提高效率、最大限度地降低成本并提高客户满意度。此外，此类系统的实施为仓库运营提供了宝贵的数据洞察力，使管理层能够做出明智的决策，从而提高整体业务绩效。

### 1.1.1 面向的服务对象

本项目的数据库需求分析的对象，主要面向国内的大润发、永辉、物美等中等规模连锁超市的单个门店。它们通常具有以下特征：

1. 单个门店的总面积在1000~2000平方米左右。
2. 仓库通常设置在超市购物区的隔壁，与超市的购物区域仅一墙之隔。顾客在商场角落看到的一些铁门拉开后就能看到仓库。
3. 仓库的面积占总面积的10%左右。
4. 仓库内有不同的功能分区，不同的库存分区通常以隔断等形式隔开。根据不同类型的商品类型存放在不同的库房中。
5. 库房内有货架，可以根据货架号存放货物。

本仓库管理系统仅分析这些门店的超市部分的商品，即需要在超市收银台结账的商品；不分析设立在超市内部的家电、数码产品、品牌美妆等产品的柜台。因为这些柜台通常只是缴纳了柜台的租金，并不使用超市的仓库。以海尔家电为例，门店是由海尔的经销商自行租赁的，经销商有自己的仓库，也可以从海尔的云仓调货给顾客。它们的库存管理与超市无关。

## 1.2 系统目标与系统边界

### 1.2.1 系统目标

本系统旨在实现以下目标：

1. 仓库信息初始化：导入现有的货物信息、仓库容量。

2. 货物进出库管理：填写出入库单号，确保货物出入库无误。

3. 货物信息统计：按货物类型、仓库编号等属性统计库存信息。

4. 数据安全性：仓库管理员信息管理，设置禁止删除根管理员、根管理员管理其他管理员等功能，防止数据的泄露和违规操作。

### 1.2.2 系统边界

本项目的出库功能不考虑出库后的货物摆在购物区的哪个位置。因为超市的货架摆放涉及到“陈列费”、生产日期等问题。一般情况下，某个新品牌想要开拓市场，会付费给零售商，将自己的产品摆放在显眼的位置，这就是陈列费。同时，超市的员工在码货时会把生产日期靠后的产品放在更显眼的位置。码货的位置是需要超市员工灵活落实的，不是仓库管理员应该考虑的问题。

本项目的入库功能不考虑货物在装卸状态的情况。在实际的超市仓库中，往往有地磅、装卸区等区域。在装卸区可能会发生一些问题，超市员工发现货物在运输过程中损坏了，超市经理要求供应商A运10箱醋，但是有2箱在路上碎了。在这种情况下，本项目的入库系统只关心入库了8箱供应商A的醋，不关心货损掉的那2箱该怎么记录和处理。

## 1.3 功能需求分析

### 1.3.1 系统的功能描述

**1) 仓库管理员人员管理**

①以编号和密码验证登入权限，成功登入后才能进行后续操作，保证信息安全性。

②设置根管理员，只有根管理员有权限管理查询其他管理员在数据库中的信息。

1. 管理员信息初始化：对接入本系统前的货物信息进行导入。
2. 信息查询：实时查询管理员的编号、姓名、密码等。
3. 信息修改：增加、删除、修改现有的管理员信息。

**2) 出/入库管理**

①入库管理：入库信息管理入库货物的入库编号、入库来源(供应商)、入库时间、管理员编号和入库明细。

②出库管理：出库信息管理出库货物的出库编号、出库来源(货位) 、入库时间、管理员编号和出库明细。

**3) 货物信息管理**

①货物信息初始化：对接入本系统前的货物信息进行导入。

②信息查询：实时查询货物的编号、名称、类型等。

③信息修改：增加、删除、修改现有的货物信息。

1. 增加：用于管理超市新进的某品牌产品、某品种蔬果等情况。
2. 删除：用于管理超市不再考虑进货的货物，可能是商品爆出产品安全问题，也有可能是供应商决定撤出本地市场等情况。
3. 修改：非出库操作引起的库存信息变更，例如超过保质期等货损情况。

**4) 供应商管理**

①供应商信息初始化：对在接入本系统前的供应商信息进行导入。

②信息查询：实时查询供应商信息，如供应商编号、供应商姓名、手机号、地址等。

③信息修改：增加、删除、修改现有的供应商信息。

1. 增加：如市场新成立的品牌与本超市建立了供应关系。
2. 删除：如市场现有的某供应商退出本地市场。
3. 修改：如供应商搬离了现在的地址、修改了电话号码等。

### 1.3.2 系统总体功能图

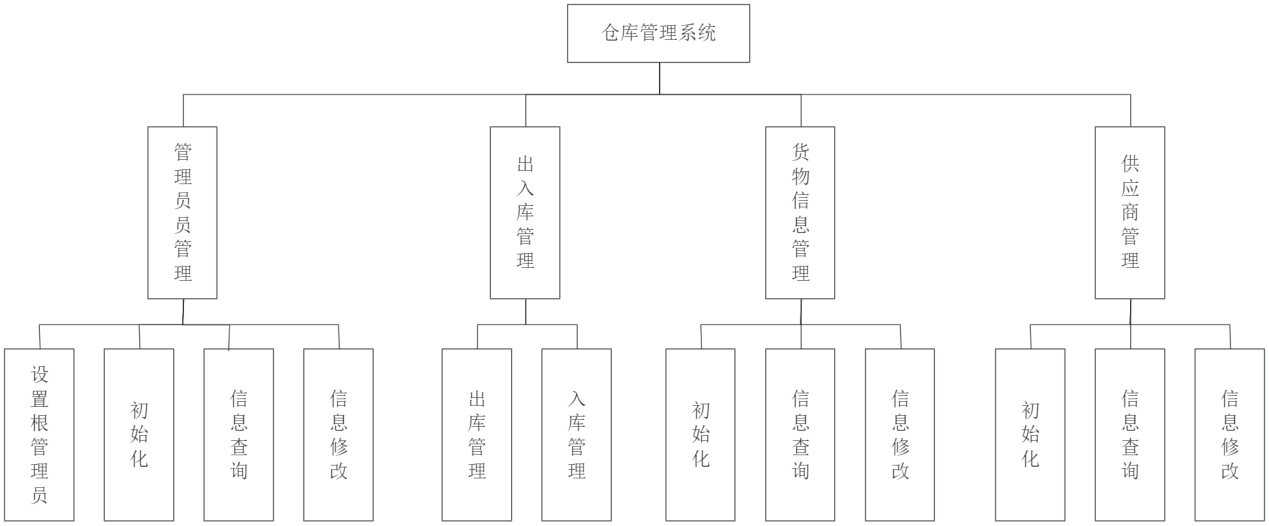


图1 系统总体功能图

### 1.3.3 数据流图

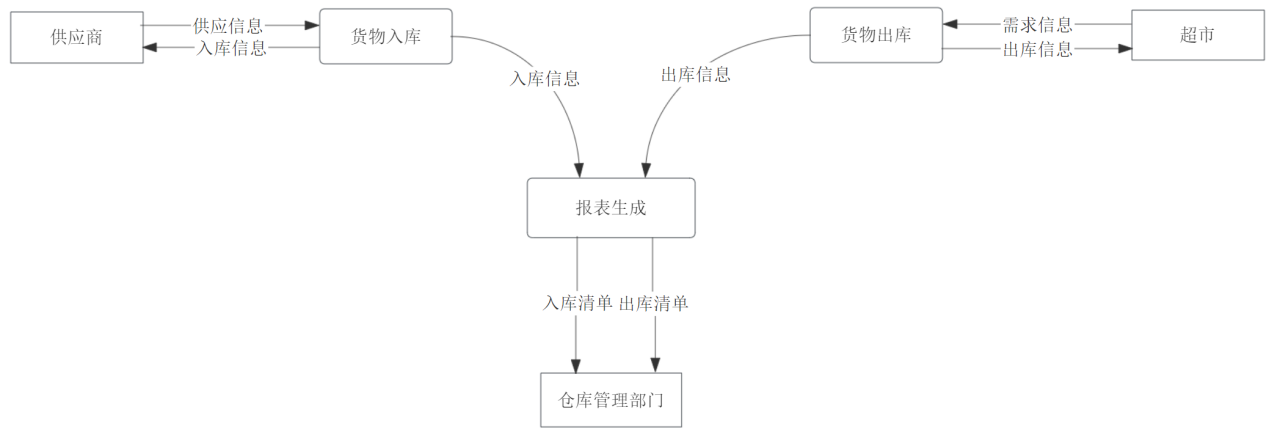


图2 顶层数据流图

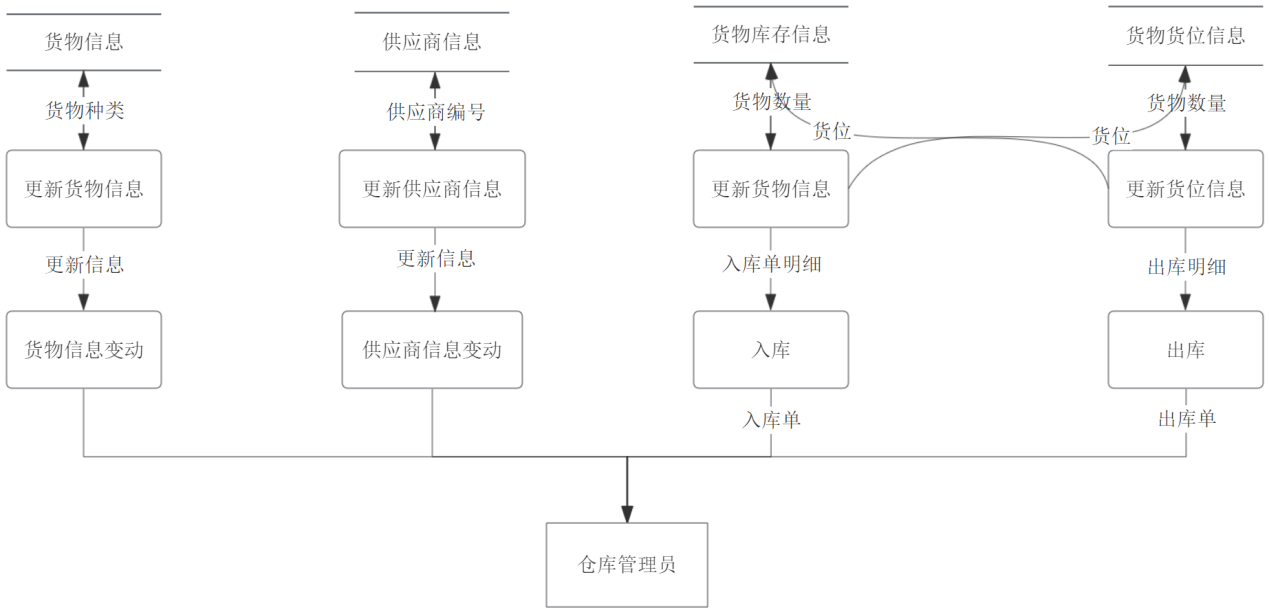


图3 二层数据流图

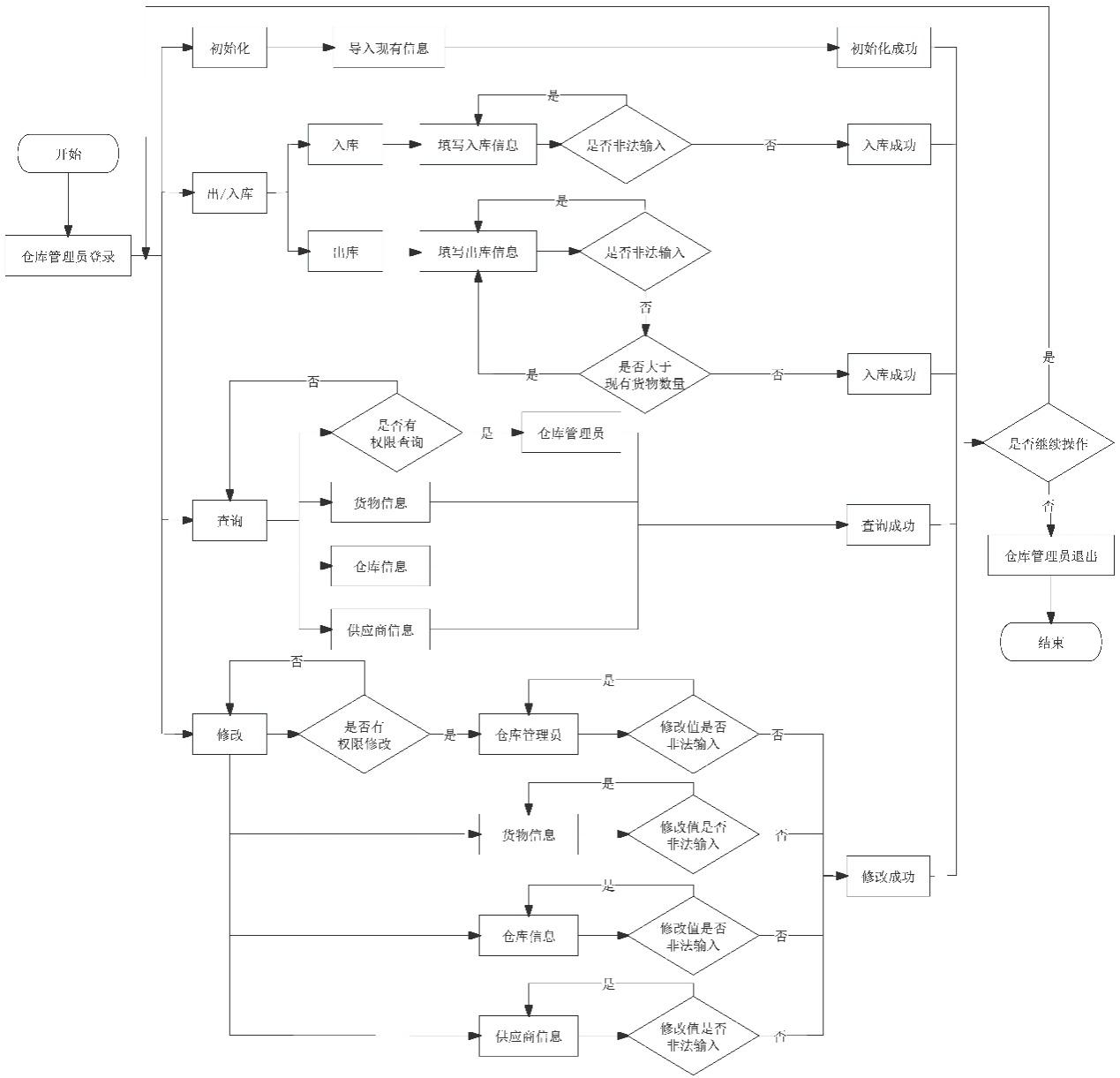


图4 超市仓库管理员流程图

## 1.4 数据需求及业务规则分析

### 1.4.1 数据需求描述

（1）仓库管理员：包括管理员编号、管理员姓名、密码、性别、身份证号、手机号、根管理员等信息，由管理员编号唯一标识。其中根管理员记录该管理员的管理权限，包括“是”、“否”等状态。

（2）货物类别：包括分类号、分类名称，由分类号唯一标识。

（3）货物：包括货物编号、货物名称、货物数量、分类号、货位、生产日期、保质期等信息，由货物编号唯一标识。其中到期时间是由生产日期和保质期计算出来的派生属性。

（4）供应商：包括供应商编号、供应商姓名、手机号、地址等信息，由供应商编号唯一标识。其中住址是复合属性，由省份、城市、街道组成。

（5）入库单：包括入库编号、入库时间、管理员编号、供应商编号、入库明细（包括货物编号、货物名称、入库数量）等信息，由入库编号唯一标识。

（6）出库单：包括出库编号、出库时间、出库明细（包括货物编号、货物名称、出库数量、入库金额）等信息，由出库编号唯一标识。

### 1.4.2 数据字典

参见“2.1.2 定义基本实体集(数据字典)”。

### 1.4.3 业务规则描述

（1）所有通过编号-密码验证后登录进入系统的仓库管理员都能对仓库货物进行出入库管理，对货物进行增删改查操作。只有一位根管理员能对仓库管理员实体集中的信息进行增删改查操作。

（2）每次货物出入库由仓库管理员进行出入库操作。一张入库单对应一位仓库管理员，一张出库单对应一位仓库管理员，表示一次出库/入库操作由一位仓库管理员负责。一位仓库管理员可以对应多张出库单/入库单，表示一位仓库管理员可以负责多张出库单/入库单。

（3）一张入库单对应一个供应商编号，一个供应商编号可以对应多张入库单。

（4）一张入库单对应一个入库编号、一张入库单对应多条入库明细，一条入库明细对应一个货物编号，一个货物编号对应一条入库明细和一张入库单。

（5）一张出库单对应一个出库编号、一张出库单对应多条出库明细，一条出库明细对应一个货物编号，一个货物编号对应一条出库明细。

（6）入库单的入库时间必须在当前时间或之前，不得提前入库。

（7）出库单的出库时间必须在货物对应的保质期之前，对于过期货物需要集中处理，不进行出库操作。

（8）仓库管理员执行入库操作时，系统根据入库单中入库明细的货物编号和入库数量对货物中对应货物编号的货物数量进行增加，出库时进行减少。规定出库时的出库数量不能大于货物对应货物编号当前的货物数量

（9）一个货物编号对应一个分类号，一个分类号可以对应多个货物编号。

（10）一个货物编号对应一个货位，一个货位对应一个货物编号。

（11）一个货物编号对应一个生产日期和保质期。

（12）一个货物编号对应一个货物名称，一个货物名称可以对应多个货物编号。

### 1.4.4 完整性约束规则描述

（1）管理员编号、供应商编号、货物编号、分类号、货位、入库编号、出库编号必须唯一，且不能取空值。

（2）系统需记录每种货物的货物数量，在入库时增加货物数量，出库时减少货物数量。

（3）一个入库单的入库数量、出库单的出库数量不能超过该货物的货物数量。

（4）入库单的入库时间、出库单的出库时间不能晚于当前时间。

（5）出库明细的货物编号必须存在于货物的货物编号。

（6）各种编号的编码规则：

① 管理员编号的编码规则：由10位数字组成。

② 供应商编号的编码规则：由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。

③ 货物编号的编码规则：由10位数字组成，其中前6位数字代表上架日期，后4位数字为流水号。

④ 分类号的编码规则：由3位数字组成。

⑤ 货位的编码规则：由8位数字组成，其中前2位数字代表库房，接下来2位代表货架，最后4位代表货位。

⑥入库编号的编码规则：由1位字母加9位数字组成，1位字母I代表入库，前6位数字代表入库日期，后3位数字为流水号。

⑦ 出库编号的编码规则：由1位字母加9位数字组成，其中1位字母O代表入库，前6位数字代表出库日期，最后3位数字为流水号。

# 2 概念设计

## 2.1 确定基本实体集

### 2.1.1 发现基本实体集

仓库管理员Administrator实体集。其属性有：管理员编号administratorNo、管理员姓名administratorName、管理员性别sex、身份证ID号identitycard、手机号administratorPhoneNo、根管理员rootAdministrator、密码adPassWord。

入库单CheckIn依赖实体集。其属性有：入库编号checkInNo、入库时间checkInTime、管理员编号administratorNo。

入库明细CheckInDetails弱实体集。其属性有：入库明细编号checkInDetailsNo、货物编号wareNo、入库数量checkInQuantity、供应商编号supplierNo。

出库单CheckOut依赖实体集。其属性有：出库编号checkOutNo、出库时间checkOutTime、管理员编号administratorNo。

出库明细CheckOutDetails弱实体集。其属性有：出库明细编号checkOutDetailNo、货物编号wareNo、出库数量outQuantity。

货物Ware实体集。其属性有：货物编号wareNo、货物名称wareName、货物数量wareQuantity、货物类别号classNo、货位wareLocation、保质期shelfLife、生产日期productionDate、到期时间dueDate、供应商编号supplierNo。

货物类别WareClass实体集。其属性有：货物类别号classNo、货物类别名称className。

供应商Supplier实体集。其属性有：供应商编号supplierNo、供应商名称supplierName、手机号supplierPhoneNo、地址supplierAddress。

### 2.1.2 定义基本实体集(数据字典)

(1) 仓库管理员Administrator实体集。其属性有：管理员编号administratorNo、管理员姓名administratorName、管理员性别sex、身份证ID号identitycard、手机号administratorPhoneNo、根管理员rootAdministrator、密码adPassWord。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| administratorNo | 管理员编号 | char(10) | 否 |
| administratorName | 管理员姓名 | varchar(20) | 否 |
| sex | 管理员性别 | char(1),取值范围： {‘男‘，‘女’} | 是 |
| identitycard | 身份证ID号 | char(18) | 否 |
| administratorPhoneNo | 手机号 | char(11) | 否 |
| rootAdministrator | 根管理员 | char(2),取值范围： {‘是’，‘否’}， 根管理员有权限增删改管理员 | 否 |
| adPassWord | 密码 | char(10)，不能少于6位 | 否 |

(2) 入库单CheckIn依赖实体集。其属性有：入库编号checkInNo、入库时间checkInTime、管理员编号administratorNo。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| checkInNo | 入库编号 | char(10)，由1位字母加9位数字组成，其中1位字母I代表入库，前6位数字代表出库日期，最后3位数字为流水号。 | 否 |
| checkInTime | 入库时间 | smalldatetime，取值范围：1900-01-01~当前 | 否 |
| administratorNo | 管理员编号 | char(10) | 否 |

(3) 入库明细CheckInDetails弱实体集。其属性有：入库明细编号checkInDetailsNo、货物编号wareNo、入库数量checkInQuantity、供应商编号supplierNo。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| checkInDetailsNo | 入库明细编号 | char(3)，3位流水号。 | 否 |
| wareNo | 货物编号 | char(10)，由10位数字组成， 其中前6位数字代表上架日期，后4位数字为流水号。 | 否 |
| checkInQuantity | 入库数量 | int | 否 |
| supplierNo | 供应商编号 | char(8)，由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。 | 否 |

(4) 出库单CheckOut依赖实体集。其属性有：出库编号checkOutNo、出库时间checkOutTime、管理员编号administratorNo。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| checkOutNo | 出库编号 | char(10)，由1位字母加9位数字组成，其中1位字母O代表出库，前6位数字代表出库日期，最后3位数字为流水号。 | 否 |
| checkOutTime | 出库时间 | smalldatetime，取值范围：1900-01-01~当前 | 否 |
| administratorNo | 管理员编号 | char(10) | 否 |

(5) 出库明细CheckOutDetails弱实体集。其属性有：出库明细编号checkOutDetailNo、货物编号wareNo、出库数量checkOutQuantity。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| checkOutDetailsNo | 出库明细编号 | char(3)，3位流水号。 | 否 |
| wareNo | 货物编号 | char(10)，由10位数字组成，其中前6位数字代表上架日期，后4位数字为流水号。 | 否 |
| checkOutQuantity | 出库数量 | int | 否 |

(6) 货物Ware实体集。其属性有：货物编号wareNo、货物名称wareName、货物数量wareQuantity、货物类别号classNo、货位wareLocation、保质期shelfLife、生产日期productionDate、到期时间dueDate、供应商编号supplierNo。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| wareNo | 货物编号 | char(10)，由10位数字组成，其中前6位数字代表上架日期，后4位数字为流水号。 | 否 |
| wareName | 货物名称 | varchar(20) | 否 |
| wareQuantity | 货物数量 | int | 否 |
| classNo | 货物类别号 | char(3) | 否 |
| wareLocation | 货位 | char(8)，其中 前2位数字代表库房， 接下来2位代表货架， 最后4位代表货位。 | 否 |
| shelfLife | 保质期 | char(2)，单位为月 | 否 |
| productionDate | 生产日期 | datetime，取值范围：1900-01-01~当前 | 否 |
| dueDate | 到期时间 | datetime，取值范围：1900-01-01~当前 | 否 |
| supplierNo | 供应商编号 | char(8)，由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。 | 否 |

(7) 货物类别WareClass实体集。其属性有：货物类别号classNo、货物类别名称className。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| classNo | 货物类别号 | char(3) | 否 |
| className | 货物类别名称 | varchar(20) | 否 |

(8) 供应商Supplier实体集。其属性有：供应商编号supplierNo、供应商名称supplierName、手机号supplierPhoneNo、地址supplierAddress。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 域及约束 | 允许NULL |
| supplierNo | 供应商编号 | char(8)，由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。 | 否 |
| supplierName | 供应商名称 | varchar(10) | 否 |
| supplierPhoneNo | 手机号 | char(11) | 否 |
| supplierAddress | 地址 | varchar(60) | 是 |

## 2.2 局部概念建模

### 2.2.1 入库业务的局部概念建模

入库业务涉及仓库管理员、供应商、货物等基本实体集，并伴随着生成入库单和入库明细。伴随着“入库”业务而形成的入库单需要单独建模为依赖实体集，包含入库编号、入库时间、管理员编号等基本属性。

由于入库单实体集与货物实体集之间存在多对多的入库明细联系集，将入库明细建模为入库单的弱实体集，标识联系集为填写入库明细，包括入库明细编号、货物编号、货物类别号、入库数量、供应商编号等基本属性。入库明细编号为部分码，入库明细由入库编号与入库明细编号共同标识。

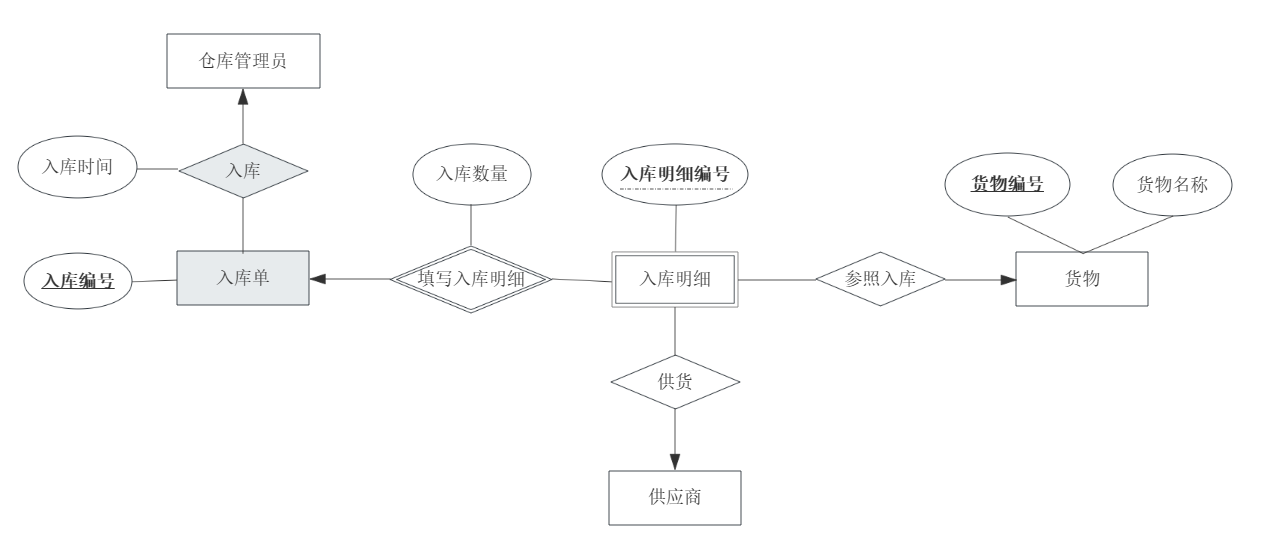


图5 入库业务的建模

### 2.2.2 出库业务的局部概念建模

出库业务涉及仓库管理员、货物等基本实体集，并伴随着生成出库单和出库明细。伴随着“出库”业务而形成的出库单需要单独建模为依赖实体集，包含出库编号、出库时间等基本属性。

由于出库单实体集与货物实体集之间存在多对多的出库明细联系集，将出库明细建模为出库单的弱实体集，标识联系集为填写出库明细，包括出库明细编号、货物编号、货物类别号、出库数量等基本属性。出库明细编号为部分码。出库明细由出库编号与出库明细编号共同标识。

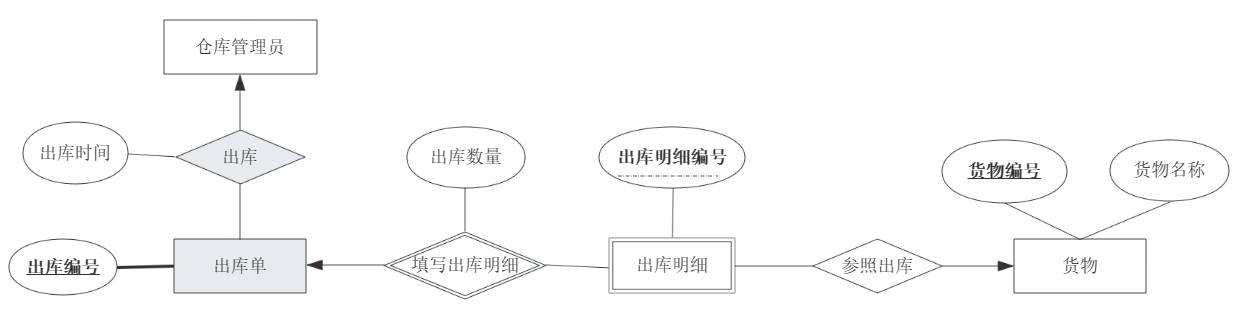


图6 出库业务的建模

## 2.3 定义联系集

### 2.3.1 联系集分析

**入库（CheckIn）联系集**

仓库管理员（Administrator）与入库单（CheckIn）依赖实体集之间是多对多关系联系集。该联系集没有联系属性。其描述属性有：入库时间（checkInTime）等。已建模为入库单依赖实体集的属性。

多对多是因为可能存在大于等于一个仓库管理员填写了大于等于一个入库单。

**出库（CheckOut）联系集**

仓库管理员（Administrator）与出库单（CheckOut）依赖实体集之间是多对多关系联系集。其描述属性有：出库时间（checkOutTime）等。已建模为出库单依赖实体集的属性。

多对多是因为可能存在大于等于一个仓库管理员填写了大于等于一个出库单。

**填写入库明细（FillInCheckIn）标识联系集**

入库单（CheckIn）依赖实体集与入库明细（CheckInDetails）弱实体集之间是一对多联系集。其描述属性有：入库数量（checkInQuantity）等。已建模为入库明细弱实体集的属性。

一对多是因为，一个入库明细只填写一个货物编号的货物，一个入库单会一次入库大于等于一种货物（以货物编号标识“种”）。

**填写出库明细（FillInCheckOut）标识联系集**

出库单（CheckOut）依赖实体集与出库明细（CheckOutDetails）弱实体集之间是一对多联系集。其描述属性有：出库数量（checkOutQuantity）等。已建模为出库明细弱实体集的属性。

一对多是因为，一个出库明细只填写一个货物编号的货物，一个出库单会一次出库大于等于一种货物（以货物编号标识“种”）。

**参照入库（ReferCheckIn）联系集**

货物（Ware）与入库明细（CheckInDetails）之间是一对多联系集。该联系集没有联系属性。

一对多是因为，一种货物（以货物编号标识“种”）受多个入库单的影响。

**参照出库（ReferCheckOut）联系集**

货物（Ware）与出库明细（CheckOutDetails）之间是一对多联系集。该联系集没有联系属性。

一对多是因为，一种货物（以货物编号标识“种”）受多个出库单的影响。

**供货（Supply）联系集**

供货商（Supplier）与入库明细（CheckInDetails）之间是一对多联系集。该联系集没有联系属性。

一对多是因为，一位供货商会重复出现在多个入库明细上。

**分类（Sort）联系集**

货物类别（WareClass）与货物（Ware）之间是一对多联系集。该联系集没有联系属性。

一对多是因为，一种货物类别包含多种货物（以货物编号标识“种”）。

### 2.3.2 联系集属性数据字典

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 |
| checkInTime | 入库时间 | 联系 | smalldatetime，取值范围：1900-01-01~当前，NOT NULL |
| checkOutTime | 出库时间 | 联系 | smalldatetime，取值范围：1900-01-01~当前，NOT NULL |
| checkInQuantity | 入库数量 | 联系 | int NOT NULL |
| checkOutQuantity | 出库数量 | 联系 | int NOT NULL |

## **2.4 概念模型(E-R图)**

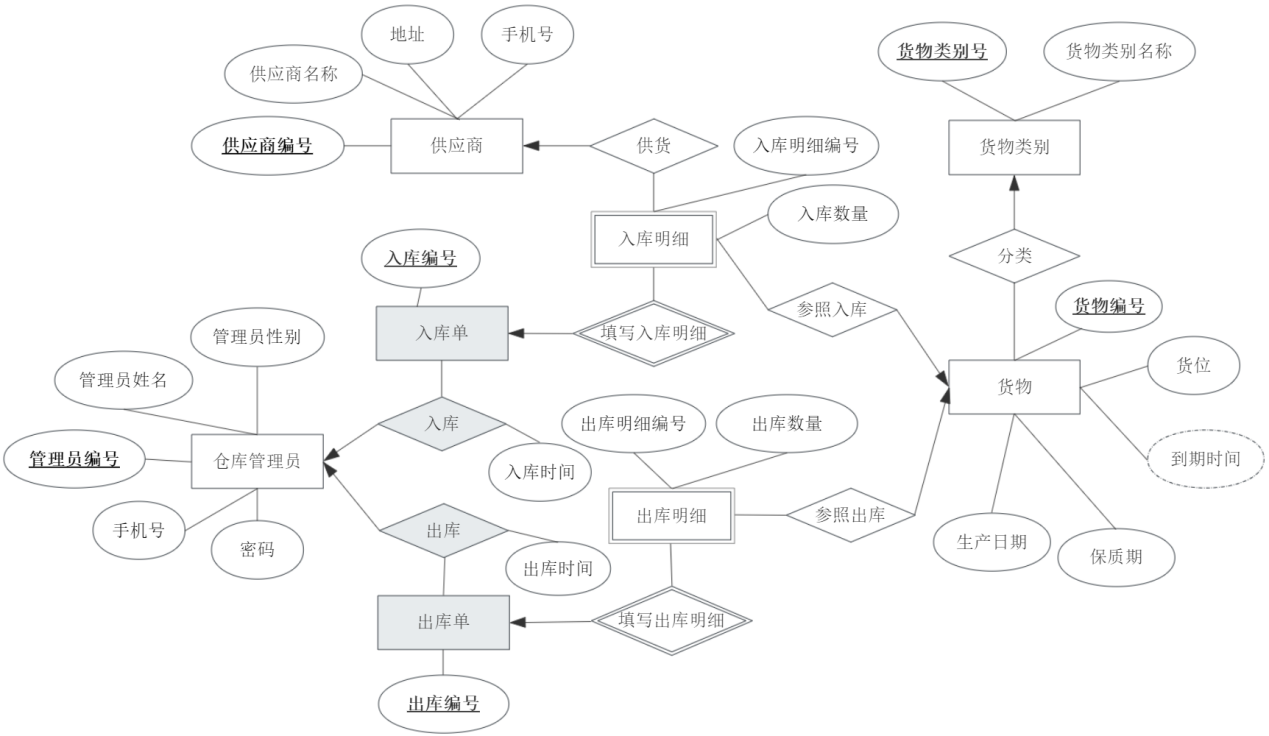


图7 总E-R图

# 3 逻辑设计

## 3.1 关系数据库模式设计

(1) 仓库管理员Administrator实体集。其属性有：管理员编号administratorNo、管理员姓名administratorName、管理员性别sex、身份证ID号identitycard、手机号administratorPhoneNo、根管理员rootAdministrator、密码adPassWord。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| **administratorNo** | 管理员编号 | 主码 | char(10) | 否 | 2202291166 |
| administratorName | 管理员姓名 |  | varchar(20) | 否 | 张三.尼格买提 |
| sex | 管理员性别 |  | char(2),取值范围：{‘男‘，‘女’} | 是 | 女 |
| identitycard | 身份证ID号 |  | char(18) | 否 | 413026200001050000 |
| administratorPhoneNo | 手机号 |  | char(11) | 否 | 17630903974 |
| rootAdministrator | 根管理员 |  | char(2),取值范围：{‘是’，‘否’}，根管理员有权限增删改管理员 | 否 | 是 |
| adPassWord | 密码 |  | char(10)，不能少于6位 | 否 | 111111 |

(2) 入库单CheckIn依赖实体集。其属性有：入库编号checkInNo、入库时间checkInTime、管理员编号administratorNo。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| **checkInNo** | 入库编号 | 主码 | char(10)，由1位字母加9位数字组成，其中1位字母I代表入库，前6位数字代表出库日期，最后3位数字为流水号。 | 否 | I230101001 |
| checkInTime | 入库时间 |  | smalldatetime，取值范围：1900-01-01~当前 | 否 | 2023-02-03 |
| ***administratorNo*** | 管理员编号 | 外码 | char(10) | 否 | 2202291166 |

(3) 入库明细CheckInDetails弱实体集。其属性有：入库明细编号checkInDetailsNo、货物编号wareNo、入库数量checkInQuantity、供应商编号supplierNo。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| checkInDetailsNo | 入库明细编号 | 部分码 | char(3)，3位流水号。 | 否 | 101 |
| wareNo | 货物编号 |  | char(10)，由10位数字组成，其中前6位数字代表上架日期,后4位数字为流水号。 | 否 | AB100001 |
| checkInQuantity | 入库数量 |  | int | 否 | 50 |
| supplierNo | 供应商编号 | 外码 | char(8)，由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。 | 否 | 20230001 |

(4) 出库单CheckOut依赖实体集。其属性有：出库编号checkOutNo、出库时间checkOutTime、管理员编号administratorNo。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| **checkOutNo** | 出库编号 | 主码 | char(10)，由1位字母加9位数字组成，其中1位字母O代表出库，前6位数字代表出库日期，最后3位数字为流水号。 | 否 | O20230  2020001 |
| checkOutTime | 出库时间 |  | smalldatetime，取值范围：1900-01-01~当前 | 否 | 2023-02-03 |
| ***administratorNo*** | 管理员编号 | 外码 | char(10) | 否 | 2202291166 |

(5) 出库明细CheckOutDetails弱实体集。其属性有：出库明细编号checkOutDetailNo、货物编号wareNo、出库数量checkOutQuantity。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| checkOutDetailsNo | 出库明细编号 | 部分码 | char(3)，3位流水号。 | 否 | 101 |
| wareNo | 货物编号 |  | char(10)，由10位数字组成，其中前6位数字代表上架日期,后4位数字为流水号。 | 否 | AB10  0001 |
| checkOutQuantity | 出库数量 |  | int | 否 | 50 |

(6) 货物Ware实体集。其属性有：货物编号wareNo、货物名称wareName、货物数量wareQuantity、货物类别号classNo、货位wareLocation、保质期shelfLife、生产日期productionDate、到期时间dueDate、供应商编号supplierNo。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| **wareNo** | 货物编号 | 主码 | char(10)，由10位数字组成，其中前6位数字代表上架日期，后4位数字为流水号。 | 否 | 2001010001 |
| wareName | 货物名称 |  | varchar(20) | 否 | 康师傅绿茶 |
| wareQuantity | 货物数量 |  | int | 否 | 50 |
| ***classNo*** | 货物类别号 | 外码 | char(3) | 否 | 010 |
| wareLocation | 货位 |  | char(8)，其中前2位数字代表库房，接下来2位代表货架，最后4位代表货位。 | 否 | 1010970 |
| shelfLife | 保质期 |  | char(2)，单位为月 | 否 | 18 |
| productionDate | 生产日期 |  | datetime，取值范围：1900-01-01~当前 | 否 | 2023/3/20 |
| dueDate | 到期时间 | 派生 | datetime，取值范围：1900-01-01~当前 | 否 | 2024/3/20 |
| ***supplierNo*** | 供应商编号 | 外码 | char(8)，由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。 | 否 | 20230001 |

(7) 货物类别WareClass实体集。其属性有：货物类别号classNo、货物类别名称className。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| classNo | 货物类别号 | 主码 | char(3) | 否 | 010 |
| className | 货物类别名称 |  | varchar(20) | 否 | 饮料 |

(8) 供应商Supplier实体集。其属性有：供应商编号supplierNo、供应商名称supplierName、手机号supplierPhoneNo、地址supplierAddress。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 允许NULL | 实例 |
| **supplierNo** | 供应商编号 | 主码 | char(8)，由8位数字组成，其中前4位数字为建立日期的年份，后4位为该年度的流水号。 | 否 | 20230001 |
| supplierName | 供应商名称 |  | varchar(10) | 否 | 张三 |
| supplierPhoneNo | 手机号 |  | char(11) | 否 | 17630903974 |
| supplierAddress | 地址 | 复合 | varchar(60) | 是 | 江西省南昌市玉屏西大街665号 |

## 3.2 完整性约束的说明

除去1.4.4中已经描述的完整性约束外，还有一些完整性约束是值得说明的。

（1）在仓库管理员实体集中，设置的性别属性仅能取‘男’和‘女’。

（2）根管理员属性取‘是’或‘否’，是代表有权限增删改管理员，通过定义触发器来实现。

（3）在货物实体集中，到期时间属性由保质期和生产日期属性共同决定。

（4）在供货商实体集中，地址是符合属性，由省份、城市、街道组成，若缺省则为0。

# 4 数据库应用与安全性设计

## 4.1 数据库脚本

USE master

-----------------↓创建数据库↓-----------------

CREATE DATABASE WareDB

on(

name='WareDB',

filename='D:\Document\_user\SQL\Document\WareDB\WareDB.mdf',

size=3,

maxsize=20,

filegrowth=1)

log on

(name='WareDB',

filename='D:\Document\_user\SQL\Document\WareDB\WareDB\_Log.ldf',

size=1,

maxsize=5,

filegrowth=1)

GO

-----------------↓创建表及主键、外键、CHECK↓-----------------

USE WareDB

CREATE TABLE Administrator (

administratorNo char(10) NOT NULL,

administratorName varchar(20) NOT NULL,

sex char(2) NULL,

identitycard char(18) UNIQUE NOT NULL,

administratorPhoneNo char(11) NOT NULL,

rootAdministrator char(2) NOT NULL,

adPassWord char(10) NOT NULL,

CONSTRAINT PK\_Administrator PRIMARY KEY (administratorNo),

CONSTRAINT CK\_rootAdministrator CHECK (rootAdministrator IN ('是', '否')),

CONSTRAINT CK\_sex CHECK (sex IN ('男', '女')),

CONSTRAINT CK\_adPassWord CHECK (LEN(AdPassWord) >= 6)

)

PRINT 'The Administrator table has been created.'

GO

CREATE TABLE Supplier (

supplierNo char(8) NOT NULL,

supplierName varchar(10) NOT NULL,

supplierPhoneNo char(11) NOT NULL,

supplierAddress varchar(60) NULL,

CONSTRAINT PK\_Supplier PRIMARY KEY (supplierNo)

)

PRINT 'The Supplier table has been created.'

GO

CREATE TABLE CheckIn (

checkInNo char(10) NOT NULL,

checkInTime smalldatetime DEFAULT GETDATE() NOT NULL,

administratorNo char(10) NOT NULL,

CONSTRAINT PK\_CheckIn PRIMARY KEY (checkInNo),

CONSTRAINT FK\_CheckIn FOREIGN KEY (administratorNo) REFERENCES Administrator(administratorNo),

CONSTRAINT CK\_checkInNo CHECK

(checkInNo LIKE 'I' + RIGHT(YEAR(GETDATE()), 2) + RIGHT('0' + CAST(MONTH(GETDATE()) AS VARCHAR(2)), 2) + RIGHT('0' + CAST(DAY(GETDATE()) AS VARCHAR(2)), 2) + '[0-9][0-9][0-9]')

)

PRINT 'The CheckIn table has been created.'

GO

CREATE TABLE CheckInDetails (

checkInDetailsNo char(3) NOT NULL,

checkInNo char(10) NOT NULL,

wareNo char(10) NOT NULL,

checkInQuantity int NOT NULL,

supplierNo char(8) NOT NULL,

CONSTRAINT PK\_CheckInDetails PRIMARY KEY CLUSTERED (checkInDetailsNo,checkInNo),

CONSTRAINT FK1\_CheckInDetails FOREIGN KEY (checkInNo) REFERENCES CheckIn(checkInNo),

CONSTRAINT FK2\_CheckInDetails FOREIGN KEY (supplierNo) REFERENCES Supplier(supplierNo),

CONSTRAINT CK\_checkInDetailsNo CHECK (checkInDetailsNo LIKE '[0-9][0-9][0-9]')

)

PRINT 'The CheckInDetails table has been created.'

GO

CREATE TABLE CheckOut (

checkOutNo char(10) NOT NULL,

checkOutTime smalldatetime DEFAULT GETDATE() NOT NULL,

administratorNo char(10) NOT NULL,

CONSTRAINT PK\_CheckOut PRIMARY KEY (checkOutNo),

CONSTRAINT FK\_CheckOut FOREIGN KEY (administratorNo) REFERENCES Administrator(administratorNo),

CONSTRAINT CK\_checkOutNo CHECK

(checkOutNo LIKE 'O' + RIGHT(YEAR(GETDATE()), 2) + RIGHT('0' + CAST(MONTH(GETDATE()) AS VARCHAR(2)), 2) + RIGHT('0' + CAST(DAY(GETDATE()) AS VARCHAR(2)), 2) + '[0-9][0-9][0-9]')

)

PRINT 'The CheckOut table has been created.'

GO

CREATE TABLE CheckOutDetails (

checkOutDetailsNo char(3) NOT NULL,

checkOutNo char(10) NOT NULL,

wareNo char(10) NOT NULL,

checkOutQuantity int NOT NULL,

CONSTRAINT PK\_CheckOutDetails PRIMARY KEY CLUSTERED (checkOutDetailsNo,checkOutNo),

CONSTRAINT FK\_CheckOutDetails FOREIGN KEY (checkOutNo) REFERENCES CheckOut(checkOutNo),

CONSTRAINT CK\_checkOutDetailsNo CHECK (checkOutDetailsNo LIKE '[0-9][0-9][0-9]'),

)

PRINT 'The CheckOutDetails table has been created.'

GO

CREATE TABLE WareClass (

classNo char(3) NOT NULL,

className varchar(20) NOT NULL,

CONSTRAINT PK\_WareClass PRIMARY KEY (classNo),

CONSTRAINT CK\_ClassNo CHECK (classNo LIKE '[0-9][0-9][0-9]'),

)

PRINT 'The WareClass table has been created.'

GO

CREATE TABLE Ware (

wareNo char(10) NOT NULL,

wareName varchar(20) NOT NULL,

wareQuantity int NOT NULL,

classNo char(3) NOT NULL,

wareLocation char(8) NOT NULL,

shelfLife char(2) NOT NULL,

productionDate datetime NOT NULL,

dueDate datetime NOT NULL,

supplierNo char(8) NOT NULL,

CONSTRAINT PK\_Ware PRIMARY KEY (wareNo),

CONSTRAINT FK1\_Ware FOREIGN KEY (classNo) REFERENCES WareClass(classNo),

CONSTRAINT FK2\_Ware FOREIGN KEY (supplierNo) REFERENCES Supplier(supplierNo),

CONSTRAINT CK\_WareNo CHECK (WareNo LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

)

PRINT 'The Ware table has been created.'

GO

-----------------↓完整性设计↓-----------------

--入库数量更新触发器

USE WareDB

GO

CREATE TRIGGER checkIn\_ware\_quantity

ON CheckInDetails

AFTER INSERT

AS

BEGIN

DECLARE @checkInQuantity INT;

DECLARE @wareNo INT;

SELECT @checkInQuantity = inserted.checkInQuantity, @wareNo = inserted.wareNo

FROM inserted;

UPDATE Ware

SET wareQuantity = wareQuantity + @checkInQuantity

WHERE wareNo = @wareNo;

PRINT '入库数量为' + CAST(@checkInQuantity AS VARCHAR(MAX)) + '，wareQuantity的数值已更新';

END;

GO

--出库数量更新触发器

USE WareDB

GO

CREATE TRIGGER checkOut\_ware\_quantity

ON CheckOutDetails

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @checkOutQuantity int, @wareNo int, @wareQuantity int;

SELECT @checkOutQuantity = checkOutQuantity, @wareNo = wareNo

FROM inserted;

SELECT @wareQuantity = wareQuantity

FROM Ware

WHERE wareNo = @wareNo;

IF @wareQuantity < @checkOutQuantity

BEGIN

ROLLBACK TRANSACTION;

PRINT '库存不足，出库失败';

END

ELSE

BEGIN

UPDATE Ware

SET wareQuantity = wareQuantity - @checkOutQuantity

WHERE wareNo = @wareNo;

PRINT '出库数量为' + CAST(@checkOutQuantity AS VARCHAR(10)) + '，wareQuantity的数值已更新';

END

END;

GO

USE WareDB

GO

--低库存触发器

CREATE TRIGGER WareQuantityCheckTrigger

ON Ware

AFTER UPDATE

AS

BEGIN

IF EXISTS (SELECT \* FROM inserted WHERE wareQuantity < 30)

BEGIN

DECLARE @WareName VARCHAR(50)

SELECT @WareName = wareName FROM inserted

PRINT '库存警报：商品[' + @WareName + ']库存数量低于30'

END

END

GO

-- 创建定时任务检查保质期

USE WareDB;

IF OBJECT\_ID('dbo.CheckDueDate') IS NOT NULL

DROP PROCEDURE dbo.CheckDueDate;

GO

CREATE PROCEDURE dbo.CheckDueDate

AS

BEGIN

SET NOCOUNT ON;

DECLARE @today DATE = GETDATE();

SELECT '商品编号：' + wareNo + '，到期日期：' + CONVERT(NVARCHAR(10), dueDate, 120)

+ '，距离到期还有 ' + CAST(DATEDIFF(day, @today, dueDate) AS NVARCHAR(10)) + ' 天。'

FROM Ware

WHERE DATEDIFF(day, @today, dueDate) < 30;

END;

GO

IF EXISTS (SELECT \* FROM msdb.dbo.sysjobs WHERE name = N'CheckDueDate')

BEGIN

EXEC msdb.dbo.sp\_delete\_job @job\_name = N'CheckDueDate';

END

DECLARE @jobId binary(16);

EXEC msdb.dbo.sp\_add\_job @job\_name=N'CheckDueDate', @enabled=1, @job\_id=@jobId OUTPUT;

EXEC msdb.dbo.sp\_add\_jobstep @job\_id=@jobId, @step\_name=N'Check Due Date', @subsystem=N'TSQL',

@command=N'EXEC dbo.CheckDueDate;', @database\_name='WareDB', @retry\_attempts=0, @retry\_interval=0, @output\_file\_name=N'NUL', @flags=0;

EXEC msdb.dbo.sp\_add\_schedule @schedule\_name=N'Daily at 10am', @freq\_type=4, @freq\_interval=1, @freq\_subday\_type=1, @freq\_subday\_interval=0, @freq\_relative\_interval=0, @freq\_recurrence\_factor=1, @active\_start\_time=100000, @active\_end\_time=235959;

EXEC msdb.dbo.sp\_attach\_schedule @job\_name=N'CheckDueDate', @schedule\_name=N'Daily at 10am';

EXEC msdb.dbo.sp\_add\_jobserver @job\_name=N'CheckDueDate', @server\_name = N'(local)';

GO

--根管理员权限触发器

USE WareDB

GO

CREATE TRIGGER CheckRootAdministrator

ON Administrator

FOR INSERT, UPDATE, DELETE

AS

BEGIN

IF EXISTS (

SELECT \* FROM inserted WHERE rootAdministrator <> '是'

)

BEGIN

ROLLBACK TRANSACTION;

RAISERROR('无权限', 16, 1);

END

END

GO

--雇员年龄非法触发器

USE WareDB

GO

CREATE TRIGGER CheckIllegalEmployee

ON Administrator

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT \* FROM inserted)

BEGIN

DECLARE @BirthDate DATE, @Age INT

SELECT @BirthDate = CONVERT(DATE, SUBSTRING(identitycard, 7, 8), 112) FROM inserted

SELECT @Age = DATEDIFF(YEAR, @BirthDate, GETDATE())

IF @Age < 16

BEGIN

RAISERROR('雇员年龄非法！', 16, 1)

ROLLBACK TRANSACTION

END

END

END

-----------------↓安全性设计↓-----------------

/\*创建角色r1和r2\*/

CREATE ROLE r1;

CREATE ROLE r2;

/\*创建用户root作为仓库的根管理员，并赋予r1角色\*/

CREATE LOGIN root WITH PASSWORD = '111111';

CREATE USER root FOR LOGIN root;

ALTER ROLE r1 ADD MEMBER root;

/\*授予r1角色对仓库数据库中所有表的查询、插入、删除、更新操作的权限,创建表和视图的权限，删除视图的权限，管理所有管理员的权限\*/

GRANT SELECT, INSERT, DELETE, UPDATE ON ALL TABLES IN DATABASE TO r1;

GRANT CREATE TABLE, CREATE VIEW TO r1;

GRANT DELETE ON ALL VIEWS IN DATABASE TO r1;

GRANT SELECT, INSERT, DELETE, UPDATE ON Administrator TO r1;

/\*创建普通仓库管理员用户并分配给r2角色\*/

DECLARE @adminNo NVARCHAR(100);

DECLARE @adPassWord NVARCHAR(100);

DECLARE admin\_cursor CURSOR FOR

SELECT administratorNo, adPassWord FROM Administrator WHERE rootAdministrator = ‘否’;

OPEN admin\_cursor;

FETCH NEXT FROM admin\_cursor INTO @adminNo, @adPassWord;

WHILE @@FETCH\_STATUS = 0

BEGIN

DECLARE @username NVARCHAR(100) = @adminNo;

DECLARE @password NVARCHAR(100) = @adPassWord;

EXEC sp\_addlogin @username, @password;

EXEC sp\_adduser @username, @username;

EXEC sp\_addrolemember 'r2', @username;

FETCH NEXT FROM admin\_cursor INTO @adminNo, @adPassWord;

END;

CLOSE admin\_cursor;

DEALLOCATE admin\_cursor;

/\*授予r2角色对仓库数据库中除Administrator表以外的表的查询、插入、删除、更新操作的权限：\*/

GRANT SELECT, INSERT, DELETE, UPDATE ON ALL TABLES EXCEPT Administrator TO r2;

GRANT CREATE VIEW TO r2;

/\*只有根管理员能查看所有管理员信息的视图，仅允许角色r1的成员进行查看操作\*/

CREATE VIEW RootAdminView AS

SELECT \*

FROM Administrator;

GRANT SELECT ON dbo.RootAdminView TO r1;

REVOKE SELECT ON dbo.RootAdminView FROM PUBLIC; -- 撤销对所有用户的权限

/\*只有根管理员能查看所有管理员工作内容（负责的入库单出库单编号）的视图，仅允许角色r1的成员进行查看操作\*/

CREATE VIEW AdminWorkView AS

SELECT a.administratorNo, b.checkInNo,b.checkInTime, c.checkOutNo, c.checkOutTime

FROM Administrator a,CheckIn b,CheckOut c

LEFT JOIN CheckIn b ON a.administratorNo = b.administratorNo

LEFT JOIN CheckOut c ON a.administratorNo = c.administratorNo;

GRANT SELECT ON dbo.AdminWorkView TO r1;

REVOKE SELECT ON dbo.AdminWorkView FROM PUBLIC; -- 撤销对所有用户的权限

/\*存在出入库记录的货物视图。仅允许角色r1和r2的成员进行查看操作。视图的筛选条件会确保只列出在入库明细或出库明细中存在记录的货物信息。\*/

CREATE VIEW WaresView AS

SELECT \*

FROM Ware

WHERE wareNo IN (

SELECT wareNo

FROM Ware

WHERE wareNo IN (

SELECT DISTINCT wareNo

FROM CheckInDetails

)

OR wareNo IN (

SELECT DISTINCT wareNo

FROM CheckOutDetails

)

)

GRANT SELECT ON dbo.WaresView TO r1,r2;

-----------------↓存储过程设计↓-----------------

-- 按月统计入库工作量最大的前3位仓库管理员的入库信息。

CREATE PROCEDURE admin\_tot(@month char(2))

AS

BEGIN

DECLARE @admin\_no char(10), @admin\_name varchar(20), @work\_num int

DECLARE @text varchar(100), @count int

SET @count=0

SELECT @text='=======入库工作量最大的前3位仓库管理员的入库信息=========='

PRINT @text

SELECT @text='管理员编号 管理员名称 工作量'

PRINT @text

DECLARE admin\_cur SCROLL CURSOR FOR

SELECT a.administratorNo, administratorName, count(checkInNo)

FROM Administrator a, CheckIn b

WHERE a.administratorNo=b.administratorNo AND substring(checkInNo,4,2)=@month

GROUP BY a.administratorNo

ORDER BY count(checkInNo) DESC

OPEN admin\_cur

FETCH admin\_cur INTO @admin\_no, @admin\_name, @work\_num int

WHILE(@@FETCH\_STATUS=0)

BEGIN

SELECT @text=@admin\_no+' '+ @admin\_name+' '+CONVERT(char(10),@work\_num)

PRINT @text

SET @count=@count+1

IF @count<3

FETCH get\_cur INTO @admin\_no, @admin\_name,@work\_num

ELSE

BREAK

END

CLOSE admin\_cur

DEALLOCATE admin\_cur

END

EXEC admin\_tot '10'

GO

-- 输出供应货物位于前5名的供应商中的热门的前3种货物的货物编号、货物名称、总入库数量

CREATE PROCEDURE supplier\_tot

AS

GEGIN

DECLARE @ware\_no char(10),@ware\_name varchar(20),@ware\_num int

DECLARE @text varchar(100),@count int

SET @count=0

SELECT @text='=======货物供应种类前5位的供应商的前3种货物信息============'

PRINT @text

SELECT @text='货物编号 货物名称 总入库数量'

PRINT @text

DECLARE get\_cur SCROLL CURSOR FOR

SELECT a.wareNo,wareName,sum(checkInQuantity)

FROM Ware a,CheckInDetails b, (

SELECT TOP 5 a.suppilerNo,suppilername,sum(checkInQuantity)

FROM Supplier a,CheckInDetails b,

WHERE a.supplierNo=b.supplierNo

GROUP BY a.supplierNo,suppilername

ORDER BY sum(checkInQuantity) DESC) c

WHERE a.wareNo=b.wareNo and a.suppilerNo=c.suppilerNo

GROUP BY a.wareNo,wareName

ORDER BY sum(checkInQuantity) DESC

OPEN get\_cur

FETCH get\_cur INTO @ware\_no, @ware\_name,@ware\_num

WHILE(@@FETCH\_STATUS=0)

BEGIN

SELECT @text=@ware\_no+' '+ @ware\_name+' '+CONVERT(char(10),@ware\_num)

PRINT @text

SET @count=@count+1

IF @count<3

FETCH get\_cur INTO @ware\_no, @ware\_name,@ware\_num

ELSE

BREAK

END

CLOSE get\_cur

DEALLOCATE get\_cur

END

EXEC supplier\_tot

GO

-- 根据供应商编号按年统计该供应商的供应信息，输出该供应商的名称、地址、手机号，以及所有与该供应商有关的货物入库信息。

CREATE PROCEDURE sup\_tot(@supplier\_no char(8), @year char(2))

AS

BEGIN

DECLARE @supplier\_name varchar(10), @supplier\_addr varchar(60), supplier\_phone char(11)

DECLARE @ware\_no char(10),@ware\_num int

DECLARE @text varchar(100)

SELECT @text='====================供应商供应表===================='

PRINT @text

SELECT @text='-----------------------------------------------------------------'

PRINT @text

SELECT supplierName=@supplier\_name, supplierAddress=@supplier\_addr, supplierPhoneNo=@supplier\_phone

FROM Supplier

WHERE supplierNo=@supplier\_no

SELECT @text='供应商名称：' + @supplier\_name

PRINT @text

SELECT @text='供应商地址：' + @supplier\_addr

PRINT @text

SELECT @text='供应商手机号：' + @supplier\_phone

PRINT @text

SELECT @text='-----------------------------------------------------------------'

PRINT @text

SELECT @text='货物编号 总数量'

PRINT @text

DECLARE ware\_cur SCROLL CURSOR FOR

SELECT wareNo, sum(checkInQuantity)

FROM Supplier a, CheckInDetails b

WHERE a.supplierNo=b.supplierNo AND substring(checkInNo,2,2)=@year

GROUP BY a.supplierNo

OPEN ware\_cur

FETCH ware\_cur INTO @ware\_no, @ware\_num

WHILE(@@FETCH\_STATUS=0)

BEGIN

SELECT @text=@ware\_no+' '+ CONVERT(char(10),@ware\_num)

PRINT @text

FETCH get\_cur INTO @admin\_no, @admin\_name,@work\_num

END

CLOSE ware\_cur

DEALLOCATE ware\_cur

END

EXEC sup\_tot '20230001', '23'

GO

-- 仓库管理员查询入库信息，输入货物的货物编号，使用游标返回所有入库信息。逐行显示该货物的货物编号、货物名称、入库时间、入库数量、供应商编号，并统计该货物的入库总数量。

CREATE PROCEDURE GetCheckInInfo @ware\_no char(10)

AS

BEGIN

DECLARE @ware\_name varchar(20), @checkIn\_time datetime, @checkIn\_quantity int, @supplier\_no char(8), @total int

SET @total = 0

DECLARE checkIn\_cur CURSOR FOR

SELECT a.wareNo, wareName, checkInTime, chickInQuantity, a.supplierNo

FROM Ware a, CheckIn b, CheckInDetails c

WHERE a.wareNo=c.wareNo AND b.checkInNo=c.checkInNo AND a.wareNo=@ware\_no

OPEN checkIn\_cur

FETCH checkIn\_cur INTO @ware\_name, @checkIn\_time, @checkIn\_quantity, @supplier\_no

WHILE(@@FETCH\_STATUS=0)

BEGIN

PRINT '货物编号: ' + @ware\_no

PRINT '货物名称: ' + @ware\_name

PRINT '入库时间: ' + CONVERT(varchar(20), @checkIn\_time, 120)

PRINT '入库数量: ' + CONVERT(varchar(10), @checkIn\_quantity)

PRINT '供应商编号: ' + @supplier\_no

PRINT ''

SET @total = @total + @checkIn\_quantity

FETCH checkIn\_cur INTO @ware\_name, @checkIn\_time, @checkIn\_quantity, @supplier\_no

END

PRINT '该货物的入库总数量: ' + CONVERT(varchar, @total)

CLOSE checkIn\_cur

DEALLOCATE checkIn\_cur

END

EXEC GetCheckInInfo @ware\_no = '货物编号'

GO

-- 仓库管理员查询出库信息，输入货物的货物编号，使用游标返回所有出库信息。逐行显示该货物的货物编号、货物名称、出库时间、出库数量，并统计该货物的出库总数量。

CREATE PROCEDURE GetCheckOutInfo @ware\_no char(10)

AS

BEGIN

DECLARE @ware\_name varchar(20), @checkOut\_time datetime, @checkOut\_quantity int, @total int

SET @total = 0

DECLARE checkOut\_cur CURSOR FOR

SELECT a.wareNo, wareName, checkOutTime, chickOutQuantity

FROM Ware a, CheckOut b, CheckOutDetails c

WHERE a.wareNo=c.wareNo AND b.checkOutNo=c.checkOutNo AND a.wareNo=@ware\_no

OPEN checkOut\_cur

FETCH checkOut\_cur INTO @ware\_name, @checkOut\_time, @checkOut\_quantity

WHILE(@@FETCH\_STATUS=0)

BEGIN

PRINT '货物编号: ' + @ware\_no

PRINT '货物名称: ' + @ware\_name

PRINT '出库时间: ' + CONVERT(varchar(20), @checkOut\_time, 120)

PRINT '出库数量: ' + CONVERT(varchar(10), @checkOut\_quantity)

PRINT '供应商编号: ' + @supplier\_no

PRINT ''

SET @total = @total + @checkOut\_quantity

FETCH checkOut\_cur INTO @ware\_name, @checkOut\_time, @checkOut\_quantity

END

PRINT '该货物的出库总数量: ' + CONVERT(varchar, @total)

CLOSE checkOut\_cur

DEALLOCATE checkOut\_cur

END

EXEC GetCheckOutInfo @ware\_no = '货物编号'

GO

-- 统计货物的入库和出库数量之比来分析货物周转率。

CREATE PROCEDURE wareRate(@ware\_no char(10))

AS

BEGIN

DECLARE @checkIn\_num int, @checkOut\_num int, @rate numeric(3,2)

SELECT sum(checkInQuantity)

FROM CheckInDetails

WHERE wareNo=@ware\_no

SELECT sum(checkOutQuantity)

FROM CheckOutDetails

WHERE wareNo=@ware\_no

IF @checkIn\_num > 0

SELECT @rate=@checkOut\_num/@checkIn\_num

ELSE

BREAK

END

EXEC wareRate 'AB100001'

GO



图8 数据关系模式导航图

## 4.2 数据库完整性设计

库存变化的记录：在货物出入库等操作时，需要通过触发器记录下该货物的库存变化记录，并将记录保存相应的库存表中。例如，CheckInDetails里的checkInQuantity、CheckOutDetails里的checkOutQuantity应通过触发器更新Ware里的wareQuantity。

库存警报的生成与处理：当某一货物的库存量低于阈值时，需要通过触发器生成一条库存警报信息，同时发送通知给仓库管理员或相关人员。例如，Ware里的wareQuantity小于30时向仓库管理员发出库存警报信息。

库存检查的执行：在货物出库操作时，需要通过触发器进行库存检查，确保所有操作都是合法的，如果存在问题，则回滚事务并提示错误信息。例如，CheckOutDetails里的outQuantity不得大于Ware里的wareQuantity。

货物临期检查：对于仓库中的物品，要定期检查保质期是否新鲜。例如创建定时任务，每天固定时间执行以下检测：Ware表中的dueDate小于30时用触发器生成一条临期提醒。

仓库管理员信息管理：仅根仓库管理员可以增删查改其他仓库管理员信息，否则用触发器生成一条“无权限！”提醒。根仓库管理员录入或修改其他仓库管理员身份证号时，自动检测其出生日期字段距今是否满16年，否则用触发器生成一条警告信息。例如，在某仓库管理员希望对其他仓库管理员的信息进行增删查改时，检查Administrator的rootAdministrator属性是否为‘是’，如果不是就回滚事务并生成一条“无权限”警告；在根仓库管理员录入或修改其他仓库管理员时，用触发器判断Administrator中identitycard的7至14位距今是否满16年，如果不是就回滚事务并生成一条“雇员年龄非法！”警告。

## 4.3 数据库安全性设计

### 4.3.1 角色、用户和权限设置

（1）角色1：r1。根管理员。创建用户root为仓库的根管理员，并赋予r1角色。具有对仓库数据库中所有表的查询、插入、删除、更新的操作权限，授予创建表和视图以及删除视图的权限。具有管理普通仓库管理员在数据库中的信息的权限。

（2）角色2：r2。普通仓库管理员。为所有在仓库管理员Administrator实体集中的非根管理员创建用户，并赋予r2角色。具有对仓库数据库中除Administrator表以外的表的查询、插入、删除、更新的操作权限，授予创建视图的权限。

### 4.3.2 使用视图保证数据安全

（1）根管理员查看所有管理员的视图：只有根管理员能够查看所有管理员信息。该视图列出管理员编号和姓名等基本信息，并加入过滤器以限制只有根管理员才能够查看所有管理员的信息。

（2）根管理员查看所有管理员的工作内容的视图：列出管理员编号、入库编号、出库编号、入库时间、出库时间。只允许根管理员查询。

（3）对货物的视图：列出货物名称、货物类型、库存数量等货物信息，只允许管理员进行操作，方便仓库管理员及时了解查询仓库内货物的库存情况。

## 4.4 存储过程设计

（1）按月统计入库工作量最大的前3位仓库管理员的入库信息。

===========入库工作量最大的前3位仓库管理员的入库信息 ============

管理员编号 管理员名称 工作量

2202291101 张三 12

2202291111 李四 10

2202291121 王五 9

（2）输出供应货物位于前5名的供应商中的热门的前3种货物的货物编号、货物名称、总入库数量。

============货物供应种类前5位的供应商的前3种货物信息 ============

货物编号 货物名称 总入库数量

AC100001 康师傅矿泉水 103

AD100001 康师傅绿茶 102

AB100001 康师傅茉莉花茶 101

（3）按年统计每位供应商的供应信息，输出该供应商的名称、地址、手机号，以及所有与该供应商有关的货物入库信息。

====================供应商供应表====================

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

供应商名称： 责任有限有限责任公司

供应商地址： 江西省南昌市玉屏西大街665号

供应商手机号：13000000000

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

货物编号 总数量

AB100001 10

AA100001 40

（4）仓库管理员查询入库信息，输入货物的货物号，使用游标返回所有入库信息。同时逐行显示该货物的货物编号、货物名称、入库时间、入库数量、供应商编号，并统计该货物的入库总数量。

（5）仓库管理员查询出库信息，输入货物的货物号，使用游标返回所有出库信息。同时逐行显示该货物的货物编号、货物名称、出库时间、出库数量，并统计该货物的出库总数量。

（6）统计货物的入库和出库数量之比来分析货物周转率。