数据库原理课程实验

实验2 视图、安全性、触发器、数据库设计实验

1. 实验目的

（1）熟悉SQL语言有关视图的操作；

（2）掌握自主存取权限的定义和维护方法；

（3）掌握数据库触发器的设计和使用方法；

1. 实验内容和要求

（1）针对给定的数据库模式，以及相应的应用需求，创建视图；

（2）定义用户、角色，分配权限给用户、角色，回收权限，以相应的用户名登录数据库验证权限分配是否正确；

（3）定义BEFORE触发器和AFTER触发器。

（4）按照要求完成数据库的设计

1. 实验重点
2. 创建视图；分组统计查询、单表自身连接查询、多表连接查询
3. 定义角色、分配权限和回收权限；
4. 触发器的定义；
5. 数据库设计
6. 实验难点
7. 创建基本表时，为不同的列选择合适的数据类型，正确创建表级和列级完整性约束。
8. 利用触发器实现较为复杂的用户自定义完整性；
9. 按照数据库设计的流程，建立需求分析、概念结构和逻辑结构
10. 视图、安全性及触发器实验数据及问题

　　数据库名称：S\_T

各数据表字段如下所示：

　 学生student：

|  |  |  |  |
| --- | --- | --- | --- |
| Sno | Char(9) | Primary key | 学号 |
| Sname | Char(20) | Unique | 姓名 |
| Ssex | Char(2) |  | 性别 |
| Sage | SMALLINT |  | 年龄 |
| Sdept | Char（20） |  | 所在系 |

课程course：

|  |  |  |  |
| --- | --- | --- | --- |
| Cno | Char(4) | Primary key | 课程号 |
| Cname | Char(40) | Not null | 课程名称 |
| Cpno | Char(4) |  | 先行课 |
| Ccredit | SMALLINT |  | 学分 |

　　学生选课SC：

|  |  |  |  |
| --- | --- | --- | --- |
| Sno | Char(9) | Not null | 学号 |
| Cno | Char(4) | Not null | 课程号 |
| Grade | SMALlINT |  | 成绩 |

各个表中的示例：Student：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号（Sno） | 姓名（Sname） | 性别（Ssex） | 年龄（Sage） | 所在系（Sdept） |
| 201215121 | 李勇 | 男 | 20 | CS |
| 201215122 | 刘晨 | 女 | 19 | CS |
| 201215123 | 王敏 | 女 | 18 | MA |
| 201215124 | 张立 | 男 | 19 | IS |

Course：

|  |  |  |  |
| --- | --- | --- | --- |
| 课程号（Cno） | 课程名（Cname） | 先行课（Cpno） | 学分（Ccredit） |
| 1 | 数据库 | 5 | 4 |
| 2 | 数学 |  | 2 |
| 3 | 信息系统 | 1 | 4 |
| 4 | 操作系统 | 6 | 3 |
| 5 | 数据结构 | 7 | 4 |
| 6 | 数据处理 |  | 2 |
| 7 | PASCAL语言 | 6 | 4 |

SC表：

|  |  |  |
| --- | --- | --- |
| Sno（学号） | Cno（课程号） | 成绩（Grade） |
| 201215121 | 1 | 92 |
| 201215121 | 2 | 85 |
| 201215121 | 3 | 88 |
| 201215122 | 2 | 90 |
| 201215122 | 3 | 86 |
| 201215123 | 3 | 87 |
| 201215123 | 1 | 56 |
| 201215124 | 5 | 88 |
| 201215124 | 6 | 67 |

基于以上数据表，完成以下问题。

（1）建立信息系学生的视图（IS\_Student1），包括学号、姓名、年龄；

（2）建立信息系学生视图(IS\_Student2)，并要求进行修改和插入操作时仍需保证该视图只有信息系的学生；

（3）建立信息系选修了1号课程的学生的视图（学号、姓名、成绩）(IS\_S1)

（4）建立信息系选修了1号课程且成绩在90分以上的学生的视图（学号、姓名、成绩）(IS\_S2)；

（5）定义一个反映学生出生年份的视图(BT\_S),包括学号、姓名，出生年份；

（6）将学生的学号及平均成绩定义为一个视图（S\_G）；

（7）在信息系学生的视图中找出年龄小于20岁的学生；

（8）查询选修了1号课程的信息系学生；

（9）创建用户U1、U2；

create user 'u1'@'localhost' identified by '123456';

（9）把查询Student表的权限授给用户U1；

grant select on s\_t.student to 'u1'@'localhost';

（10）把对表SC的查询权限授予所有用户U1和U2；

（11）把查询Student表和修改学生姓名的权限授给用户U2；

（12）把用户U2修改学生姓名的权限收回；

权限授予后，可以通过select \* from mysql.tables\_priv;语句，查看授权情况

（13）创建触发器，当对表SC的Grade的属性进行修改时，若分数增加了10%，则将此次操作记录到另一个表SC\_U（Sno,Cno,Oldgrade,Newgrade）中，其中Oldgradesh是修改前的分数，Newgrade是修改后的分数。

6、数据库设计

要求学生完成如下题目中的一个，题目分配原则为：学号末两位除以5的余数加1。

（1）网上书店系统

电子商务的出现对人民的生活产生了巨大的影响，伴随着电子商务的发展，越来越多的用户选择在网上购买图书，由此网上书店系统应运而生。

网上书店系统中包括普通用户、会员和管理员。普通用户可以浏览图书、按类别查询图书、查看图书的详细信息，注册成为会员；会员可以浏览图书、按类别查询图书、查看图书的详细信息；加入购物车、修改购物车、订购图书、生成订单、查看订单、查看修改个人信息；管理员可以浏览图书、按类别查询图书、查看图书的信息、查看订单、处理订单进行发货处理、查看修改个人信息、更新图书信息等。

    网上书店系统中的图书销售基本规定如下：每个会员都可以通过订购图书生成多个订单，每个订单仅属于一个会员；每个订单中可以包含多种图书商品，每种图书商品可以出现在多个订单中；在订单信息中，每种图书都有对应的订购数量。对于实际网店的网上支付、商品评论等信息的处理，暂时略去。假设会员确认生成订单以后，就是已经付款了，管理员就可以对其进行发货处理等。

请对该系统进行需求分析（提供数据字典）、概念结构设计（提供E-R图）、逻辑结构设计（提供关系模式）、物理结构设计（建立数据库、数据表等）。

（2）图书借阅系统

图书馆中的图书信息、读者信息及图书借阅管理等是一项分成烦琐但重要的工资。图书借阅管理系统可以有效地管理图书资料信息，控制图书资料的借阅流程，对提高图书馆或阅览室的管理效率有很大的帮助。

    图书管理系统主要有以下三种用户：1)系统管理员：拥有系统的最高权限，可设置图书管理员等；2）图书管理员：维护图书的基本数据，包括图书种类处理、更新图书信息，进行读者的图书借阅和归还处理等；3）读者:可以查阅图书信息、借阅图书。

图书馆的图书情况和管理规定如下：每种图书类型都包括很多本不同的图书，同样的图书可以购买多本；每本图书可以被多次借阅；每位读者可以借阅多本图书；每本图书的借阅期限是一个月。

请对该图书借阅管理系统进行需求分析（提供数据字典）概念结构设计（提供E-R图）、逻辑结构设计（提供关系模式）、物理结构设计（建立数据库、数据表等）。

（3）教学管理系统

高等院校本着以学生为中心，提高教学质量的原则，教学管理越来越科学、规范，很多高校从基础教育和素质教育的角度出发，将课程分为必修课和选修课，其中必修课有公共基础课和专业必修课等，使得学生在完成本专业方向要求的必修课成学习之外，还可以根据自己的兴趣爱好，选修感兴趣的课程，以拓展知识面。

教学管理系统主要满足三类用户的需求：分别是教学管理人员（设置为系统管理员）、教师及学生，三类用户所具有的操作权限及操作内容是有区别的。教学管理系统中系统管理员可以对学生信息、教师信息和课程信息等进行有效的管理和维护，包括增加、删除和修改等基本的操作和维护功能以灵活的查询功能。教师和学生能够对个人的基本信息、授课和选课等所涉及的相关信息进行查询、更新等操作。

请对该教学管理系统进行需求分析（提供数据字典）概念结构设计（提供E-R图）、逻辑结构设计（提供关系模式）、物理结构设计（建立数据库、数据库等）。

（4）电子商务平台的设计与实现

电子商务平台分为两个子系统，一是后台管理子系统，一是电子商务子系统。后台管理子系统要求管理员登陆成功后，才能对商品进行管理，包括添加商品、查询商品、修改商品以及删除商品。除商品管理外，管理员还需要对商品类型、注册用户、用户的订单以及网站公告等进行管理。

电子商务子系统，包括非注册用户和用户。非注册用户或未登陆用户具有的功能如下：浏览首页、查看商品详情和查看公告。成功登陆的用户除具有未登陆用户具有的功能外，还具有购买商品、查看购物车、关注商品以及查看用户中心的功能。

请对该电子商务平台进行需求分析（提供数据字典）、概念结构设计（提供E-R 图）、逻辑结构设计（提供关系模式）、物理结构设计（建立数据库、数据表等）。

（5）仓库管理系统

仓库管理系统可以对库房信息、职工信息、商品信息等进行有效的管理和维护，包括数据记录的增加、删除、修改等基本的维护功能和灵活的查询功能。

    1）维护商品信息。实习对企业生产的商品信息的增加、删除、修改等。商品信息包括商品号、品名、类别、规格、单价和计量单位等。

    2）维护库房的基本信息。仓储中心有多个库房，需要实现对库房信息的增加、删除和更新等。库房信息包括库房的库房号、库名、地点和面积等。

    3）维护职工的个人基本信息（本系统中的职工仅限于在仓储中心工作的企业员工）。实现对职工个人信息的增加、删除和更新等。职工信息包括职工的工号、姓名、性别和电话等。

    仓库管理的基本规定如下：每种商品可以存放在不同的库房中，每间库房可以存放多种商品；一间库房可以由多位职工管理，每位职工可以在多个库房工作；每个职工可以出入库多种商品，每种商品也可以由多个职工出入库操作；每次出入库均记录出入库操作类型、商品的品名和数量、存放的库房、经手的职工和操作的日期等。

    请对该仓库管理平台进行需求分析（提供数据字典）、概念结构设计（提供E-R 图）、逻辑结构设计（提供关系模式）、物理结构设计（建立数据库、数据表等）。

7、实验报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告 | | | | |
| 实验名称：视图、安全性、触发器、数据库设计实验 | 姓名 | 龙正 | 日期 |  |
| 1. 建立信息系学生的视图（IS\_Student1），包括学号、姓名、年龄；       （2）建立信息系学生视图(IS\_Student2)，并要求进行修改和插入操作时仍需保证该视图只有信息系的学生；    （3）建立信息系选修了1号课程的学生的视图（学号、姓名、成绩）(IS\_S1)  （4）建立信息系选修了1号课程且成绩在90分以上的学生的视图（学号、姓名、成绩）(IS\_S2)；  （5）定义一个反映学生出生年份的视图(BT\_S),包括学号、姓名，出生年份；  （6）将学生的学号及平均成绩定义为一个视图（S\_G）；  （7）在信息系学生的视图中找出年龄小于20岁的学生；  （8）查询选修了1号课程的信息系学生；  （9）创建用户U1、U2；  create user 'u1'@'localhost' identified by '123456';  （9）把查询Student表的权限授给用户U1；  grant select on s\_t.student to 'u1'@'localhost';  （10）把对表SC的查询权限授予所有用户U1和U2；  （11）把查询Student表和修改学生姓名的权限授给用户U2；  （12）把用户U2修改学生姓名的权限收回；  权限授予后，可以通过select \* from mysql.tables\_priv;语句，查看授权情况  （13）创建触发器，当对表SC的Grade的属性进行修改时，若分数增加了10%，则将此次操作记录到另一个表SC\_U（Sno,Cno,Oldgrade,Newgrade）中，其中Oldgradesh是修改前的分数，Newgrade是修改后的分数。  仓库管理系统：  **需求分析**  在这个仓库管理系统中，我们需要对以下三个主要方面进行管理：   1. 商品信息 2. 库房信息 3. 职工信息   对于这些方面的管理，我们需要实现以下功能：   1. 商品信息维护：增加、删除、修改商品信息，包括商品号、品名、类别、规格、单价和计量单位等。 2. 库房信息维护：增加、删除、修改库房信息，包括库房号、库名、地点和面积等。 3. 职工信息维护：增加、删除、修改职工信息，包括工号、姓名、性别和电话等。 4. 出入库记录维护：记录每次出入库操作的类型、商品品名和数量、存放的库房、经手的职工和操作的日期等。 5. 灵活的查询功能：根据商品、库房、职工等信息进行多条件查询，并能够生成相应的报表。   **数据字典**  基于以上需求分析，我们可以得到以下数据字典：  **商品表（commodity）**   | **列名** | **类型** | **描述** | | --- | --- | --- | | commodity\_id | INT | 商品编号，主键 | | commodity\_name | VARCHAR | 商品名称 | | Category | VARCHAR | 商品类别 | | Specification | VARCHAR | 商品规格 | | Price | DECIMAL | 商品单价 | | Unit | VARCHAR | 商品计量单位 |   **库房表（warehouse）**   | **列名** | **类型** | **描述** | | --- | --- | --- | | warehouse\_id | INT | 库房编号，主键 | | warehouse\_name | VARCHAR | 库房名称 | | Location | VARCHAR | 库房地点 | | Area | DECIMAL | 库房面积 |   **职工表（employee）**   | **列名** | **类型** | **描述** | | --- | --- | --- | | employee\_id | INT | 职工编号，主键 | | employee\_name | VARCHAR | 职工姓名 | | Gender | VARCHAR | 职工性别 | | Phone | VARCHAR | 职工电话 |   **出入库记录表（stock\_record）**   | **列名** | **类型** | **描述** | | --- | --- | --- | | record\_id | INT | 记录编号，主键 | | operation\_type | VARCHAR | 出入库操作类型 | | commodity\_id | INT | 商品编号，外键 | | warehouse\_id | INT | 库房编号，外键 | | employee\_id | INT | 职工编号，外键 | | Quantity | INT | 商品数量 | | date | DATETIME | 操作日期 |   **概念结构设计**  基于以上数据字典，我们可以得到以下ER图：    **逻辑结构设计：**  根据上述的概念结构设计，我们可以得到以下的关系模式：  商品信息表（Commodity）：Commodity\_ID, Commodity\_Name, Commodity\_Type, Commodity\_Specification, Commodity\_Price, Commodity\_Unit  库房信息表（Warehouse）：Warehouse\_ID, Warehouse\_Name, Warehouse\_Location, Warehouse\_Area  职工信息表（Employee）：Employee\_ID, Employee\_Name, Employee\_Gender, Employee\_Tel  出入库记录表（InOutRecord）：Record\_ID, Commodity\_ID, Warehouse\_ID, Employee\_ID, Record\_Type, Record\_Date, Record\_Quantity  其中，每个表的列对应上述的概念结构图中的实体或属性，列之间的关系对应概念结构图中的关系。  **物理结构设计**：  基于上述的逻辑结构设计，我们可以创建相应的数据库和数据表。  首先，我们创建一个名为“WarehouseManagement”的数据库，然后在其中创建以上所述的四个数据表，每个表中包含对应的列以及主键、外键等约束条件。具体的建表语句可以如下：  -- 创建商品信息表  CREATE TABLE Commodity (  Commodity\_ID INT PRIMARY KEY,  Commodity\_Name VARCHAR(50) NOT NULL,  Commodity\_Type VARCHAR(50),  Commodity\_Specification VARCHAR(50),  Commodity\_Price DECIMAL(10, 2),  Commodity\_Unit VARCHAR(20)  );  -- 创建库房信息表  CREATE TABLE Warehouse (  Warehouse\_ID INT PRIMARY KEY,  Warehouse\_Name VARCHAR(50) NOT NULL,  Warehouse\_Location VARCHAR(100),  Warehouse\_Area DECIMAL(10, 2)  );  -- 创建职工信息表  CREATE TABLE Employee (  Employee\_ID INT PRIMARY KEY,  Employee\_Name VARCHAR(50) NOT NULL,  Employee\_Gender CHAR(1),  Employee\_Tel VARCHAR(20)  );  -- 创建出入库记录表  CREATE TABLE InOutRecord (  Record\_ID INT PRIMARY KEY,  Commodity\_ID INT NOT NULL,  Warehouse\_ID INT NOT NULL,  Employee\_ID INT NOT NULL,  Record\_Type CHAR(1) NOT NULL,  Record\_Date DATE NOT NULL,  Record\_Quantity INT NOT NULL,  CONSTRAINT fk\_Commodity\_ID FOREIGN KEY (Commodity\_ID) REFERENCES Commodity (Commodity\_ID),  CONSTRAINT fk\_Warehouse\_ID FOREIGN KEY (Warehouse\_ID) REFERENCES Warehouse (Warehouse\_ID),  CONSTRAINT fk\_Employee\_ID FOREIGN KEY (Employee\_ID) REFERENCES Employee (Employee\_ID)  );  这些表的定义包括了数据类型、主键、外键等重要信息，可以确保数据的完整性和一致性。接下来，我们可以向这些表中插入数据，以供系统使用。 | | | | |