

# 数据库系统

Database System

主讲：张仲楠 教授

Email: [zhongnan\\_zhang@xmu.edu.cn](mailto:zhongnan_zhang@xmu.edu.cn)

Office: 海韵A416



# 实验四

## 数据库安全性、完整性

## 实验目的

- 通过实验进一步理解数据库安全性和完整性的控制机制
- 掌握并能够较灵活地使用这些机制

# 实验内容

- 数据库完整性
  - 实体完整性
  - 参照完整性
  - 用户自定义的完整性
  - 触发器
- 数据库安全性
  - 授权机制
  - 取消授权机制

## 数据库完整性(1)

- 完全删除已经创建的四张表books, customers, purchases, pricing以及相关视图; **要注意删除的顺序!**
- 重新定义四张表(**属性参照lab2-createdb.sql**):
  - 定义新的books表, 其所拥有的属性与属性类型与原有的books表一致
  - 以**列级**完整性的方式定义属性bookId为主码, 同时在定义的时候必须提供约束名(books\_pk)
  - 定义属性title为“非空”, 同时在定义的时候必须提供约束名(books\_title)

## 数据库完整性(2)

### ○ 重新定义四张表(续):

- 定义新的customers表，其所拥有的属性与属性类型与原有的customers表一致
- 以**表级**完整性的方式定义属性cid为主码
- 以**列级**完整性的方式约束age不能小于18并且不能超过80，同时在定义的时候必须提供约束名(customers\_age)

## 数据库完整性(3)

### ○ 重新定义四张表(续):

- 定义新的purchases表，其所拥有的属性与属性类型与原有的purchases表一致
- 定义属性cid与bookId为主码，同时在定义的时候必须提供约束名(purchases\_pk)
- 定义属性bookId为外码，它参照Books表的主码。当删除Books表中的某一本书时，级联删除与它相关的购买记录
- 定义属性cid为外码，它参照customers表，同时在定义的时候必须提供约束名(purchases\_fk)。

## 数据库完整性(4)

- 按照如下方式重新定义四张表(续):
  - 定义新的pricing表，其所拥有的属性与属性类型与原有的pricing表一致
  - 以**列级**完整性的方式定义属性booId为主码，同时在定义的时候必须提供约束名(pricing\_pk)
  - 定义约束(price\_check)，要求当书的format是'paperback'时，当前价格不能超过50
- 使用lab2-createdb.sql中的插入语句为四张表添加数据(**实验报告中不用包括这些语句和结果截图**)



## 数据库完整性(5)

- 增加或修改完整性约束：
  - 针对books表，要求category的取值只能属于以下集合('textbook','politics','romance','cooking')
  - 将customers表中age的约束放宽，上限放宽到85。约束名不变。
  - 针对表pricing，将约束名price\_check修改为price\_check\_paper

## 数据库完整性(6)

- 定义如下的一张表：
- Score(Sno, Chinese, Math, English, Total)
  - Sno是主码，类型是长度为9的定长字符串
  - Chinese, Math, English分别表示三门课成绩，类型为整数，并且三门课的成绩取值范围是0-100，也可以无成绩(null)
  - 如果三门课均无成绩(null)，则无总成绩(null)

## 数据库完整性(7)

### ○ 创建触发器函数:

```
CREATE OR REPLACE FUNCTION tri_insert_func()  
RETURNS TRIGGER AS  
$$  
DECLARE  
BEGIN  
    INSERT INTO test_trigger_des_tbl  
VALUES (NEW.id1, NEW.id2, NEW.id3);  
    RETURN NEW;  
END  
$$ LANGUAGE PLPGSQL;
```

## 数据库完整性(8)

- 创建INSERT触发器:

```
CREATE TRIGGER insert_trigger  
BEFORE INSERT ON test_trigger_src_tbl  
FOR EACH ROW  
EXECUTE PROCEDURE tri_insert_func();
```

## 数据库完整性(9)

- 定义一个 **Before** 行级触发器，当向Score表插入新元组或修改Score表中某门课的成绩后，确保Total的取值正好是三门课成绩总和
- 向score表插入如下元组：  
(‘201515121’, 90,80,70,250),  
(‘201515122’, 90,null,70,150),  
(‘201515123’, null,80,null,null),  
(‘201515124’,null,null,null,100)  
(‘201515125’,110,90,80,280)  
结果如何？
- 将score表中sno为‘201515124’,的english属性值改为90，结果如何？

## 数据库安全性(1)

完成课本上第四章的习题：

- 6
- 7 除(3)
- 8 针对题7中的两个表
  - (1) 取消王明对部门表的SELECT权限
  - (2) 取消李勇对两个表的INSERT权限
  - (3) 取消刘星对职工表的工资字段的更新权限
  - (4) 撤销周平对两个表的所有权限以及他转授给其他用户的权限
- 只需要提交SQL语句，不用执行和截图

# 作业提交须知

- 需提交的文件包括：  
实验报告.doc（包含sql代码文本和运行结果截图）

```
1  
select sname, cno, grade  
from student, sc  
where student.sno=sc.sno;
```

← 文本

|   | SNAME | CNO | GRADE |
|---|-------|-----|-------|
| 1 | 李勇    | 1   | 92    |
| 2 | 李勇    | 2   | 85    |
| 3 | 李勇    | 3   | 88    |
| 4 | 刘晨    | 2   | 90    |
| 5 | 刘晨    | 3   | 80    |

← 截图

- 在课程中心(course.xmu.edu.cn)提交实验报告
- 截止时间：2023-05-12 23:59:59