

# 数据库原理

## The Theory of Database System

### 第六章 数据库保护



中国矿业大学计算机学院



中国矿业大学数据库原理精品课程

# 第六章 数据库保护

6.1 事务

6.2 数据库恢复

6.3 并发控制

6.4 数据库安全性

6.5 数据库完整性



## 6.4 数据库安全性

### 数据库的安全性

主要指防止非法用户使用数据库造成数据泄露、更改或破坏，以达到保护数据库的目的。

- 保护数据库以防止不合法使用



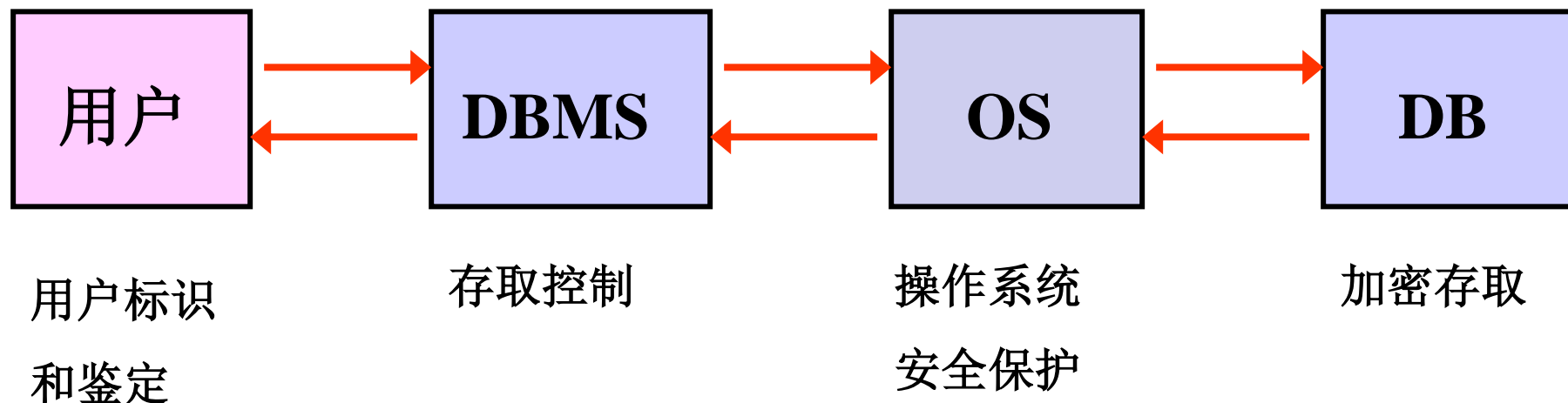
# 计算机系统安全性

为计算机系统建立和采取的各种安全保护措施，保护计算机系统（硬件、软件及数据）不会遭到破坏。

- 一般计算机系统中，安全措施是一级一级层层设置的



# 数据库安全性控制



(1)用户标识与鉴别  
(3)视图机制  
(5)数据加密

(2)存取控制  
(4)审计



# (1)用户标识与鉴别

- 利用用户名及口令的方法来实现最外层的安全保护措施。
- 特点：简单、易行  
易被窃取



no	name	pword	role
0001	xiaxh	7689	1
0002	gexin	9080	1
0003	yqy	1263	2



## (2)存取控制

- 是对用户存取数据库的权力的控制。
- 定义用户权限：
  - 1、一般数据库用户
  - 2、数据库的拥有者
  - 3、有DBA特权的用户
- 用户权限定义和**合法权检查机制**一起组成了DBMS的安全子系统





# DBMS实现数据安全性保护的过程

- 用户或DBA把授权决定告知系统

## SQL的GRANT和REVOKE

- DBMS把授权的结果存入数据字典
- 当用户提出操作请求时，DBMS根据授权定义进行检查，以决定是否执行操作请求



# 数据库安全性控制(续)

## (3)视图机制

为不同的用户定义不同的视图。

## (4)数据加密

## (5)审计

把用户对数据库的所有操作自动记录下来放入审计日志中



# 第六章 数据库保护

6.1 事务

6.2 数据库恢复

6.3 并发控制

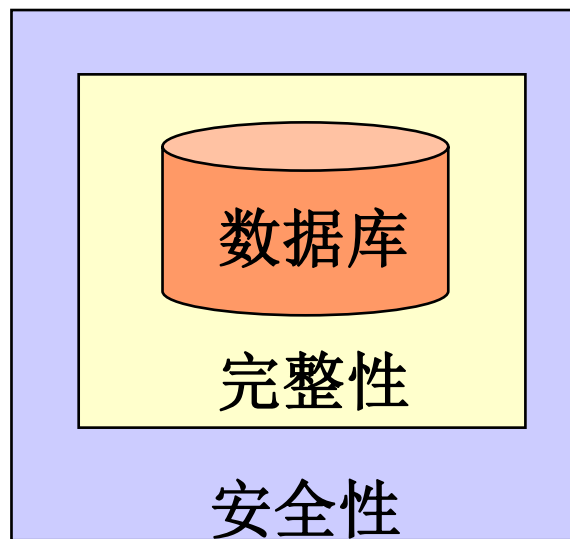
6.4 数据库安全性

6.5 数据库完整性



## 6.5 数据库完整性

- 定义：数据的正确性和相容性。
- 完整性和安全性的区别：
  - 完整性防止错误信息的输入与输出
  - 安全性是保护数据库防止恶意的破坏和非法的存取



# 一、完整性约束条件的类型

- 完整性约束条件：  
    加在数据库数据之上的语义约束条件。
- 作用对象：关系、元组、列
- 状态：静态、动态
- 静态约束：数据库每一确定状态时的数据对象所应满足的约束条件。
- 动态约束：数据库从一种状态转变为另一种状态时，新、旧值之间所应满足的约束条件。



状态			
动态	动态 列级约束 ④	动态 元组约束 ⑤	动态 关系约束 ⑥
静态	静态 列级约束 ①	静态 元组约束 ②	静态 关系约束 ③
对象 粒度	列	元组	关系



# 1、静态约束

- **静态列约束：** 是对一个列的取值域等的限制。
  - 对数据类型的约束；
  - 对数据格式的约束；
  - 对取值范围或取值集合的约束；
  - 对空值的约束；
  - 其它约束；



CREATE TABLE S

```
( sno      CHAR(10) NOT NULL UNIQUE,  
  sname    CHAR(20) UNIQUE,  
  gender   CHAR(1) ,  
  age      INT,  
  sdept    CHAR(30) );
```





```
CREATE TABLE S  
( sno CHAR(10) PRIMARY KEY,  
  sname CHAR(20),  
  gender CHAR(2) DEFAULT '男'  
    CHECK(SEX='男' OR SEX ='女'),  
  age TINYINT DEFAULT 20,  
  sdept CHAR(30) DEFAULT '计算机学院');
```



# 静态约束(续)

- **静态元组约束：**规定组成一个元组的各个列之间的约束关系。

例如：

工资（工资编号，基础工资，岗位津贴，……）

要求：岗位津贴不能大于基础工资



# 断言

- 一个断言就是一个谓词，用以说明数据库状态必须满足的一个条件。
- 语法格式：  
**create assertion <断言名> check <谓词>**



```
create assertion limitconstraint check  
  (not exists  
    (select * from 工资  
      where 岗位津贴>基础工资)  
  );
```



# 静态约束(续)

- **静态关系约束**：反映了一个关系中各个元组之间或者若干关系之间存在的联系或约束。
  - 实体完整性约束；
  - 参照完整性约束；
  - 函数依赖约束；
  - 统计约束；

即字段值与关系中多个元组的统计值之间的约束关系



```
CREATE TABLE SC  
( Sno CHAR(5) REFERENCES S(Sno) ,  
  Cno CHAR(3) REFERENCES C(Cno),  
  Grade INT,  
  PRIMARY KEY (Sno,Cno));
```



# 关系模式如下：

Employee(Eno, Ename, Salary, Dno)

Department(Dno, Dptname, Mrgno)

约束：任何部门经理的工资不超过其所在部门  
平均工资的10倍



```
CREATE ASSERTION SalaryConstraint
CHECK(
    NOT EXISTS
    (SELECT *
      FROM Department as D, Employee as E
     WHERE D.Mrgno = E.Enno AND
           Salary > 10 * (SELECT AVG(Salary)
                          FROM Employee
                          WHERE Dno = D.Dno)
    )
);
```





## 2、动态约束

- 动态列约束
- 动态元组约束
- 动态关系约束

数据库从一种状态转变为另一种状态时，新、旧值之间所应满足的约束条件。



# 动态约束

## ➤修改列定义时的约束

### 修改列值时的约束

- **修改列值时新旧值之间要满足的约束条件**

## ➤修改元组值

- 各个字段之间要满足的约束条件

## ➤关系变化前后状态



# 触发器

- 概念：触发器是一种特殊的存储过程。
- 作用对象：表、视图、模式、数据库。
- 分类：
  - DML触发器
    - after(for)——执行代码后，执行触发器；
    - instead of ——执行代码前，用触发器的内容代替设定的操作。
  - DDL触发器



# 触发器的组成:

- 触发事件: 引起触发器被触发的事件。
- 触发时间: **BEFORE or AFTER**
- 触发操作: **SQL块**
- 触发对象: 包括表、视图、模式、数据库。
- 触发频率: 说明触发器内定义的动作被执行的次数。
  - 语句级(**STATEMENT**)触发器
  - 行级(**ROW**)触发器



# MySQL触发器语法:

**CREATE TRIGGER trigger\_name**

**trigger\_time    #标识触发时机**

**trigger\_event   #标识触发事件**

**ON tbl\_name**

**FOR EACH ROW**

**trigger\_stmt;   #触发器程序体**



在“订单”表建立一个插入触发器，在添加订单时，减少“商品”表相应的货品记录中的库存。



```
delimiter $  
create trigger t1  
after insert on 订单  
for each row  
begin  
    update 商品  
    set 库存量 = 库存量 - new.购买数量  
    where 商品编号=new.商品编号;  
end$  
delimiter ;
```



两个内存临时表:

➤ **new**表: 插入的数据

➤ **old**表: 删除的数据





# 触发器的作用

- 完成比约束更复杂的数据约束，保证数据的完整性。
- 检查做执行的**SQL**语句是否被允许。
- 实现级联修改。
- 调用更多的存储过程(约束是不能调用存储过程的)。
- 更改原本要执行的**SQL**语句。
- 防止数据表结构更改或数据表被删除。



## 二、完整性控制机制的功能

### 1、定义功能

**能够定义完整性约束条件**

### 2、检查功能

**检查操作请求是否违背了完整性约束条件**

### 3、保证完整性约束条件

**若检查发现操作请求违背了完整性约束条件，则拒绝该操作。**



# 三、关系系统三类完整性的实现

- 关系数据库系统都提供了定义和检查实体完整性、参照完整性和用户定义的完整性的功能
- 违反实体完整性规则和用户定义的完整性规则的操作：

**—一般是拒绝执行**



- 违反参照完整性的操作：

- 拒绝执行

- 接受这个操作，同时执行一些附加的操作，以保证数据库的状态正确



# 违约反应：可有三种策略

- **级联删除 (CASCADES)**

将参照关系中外码值与被参照关系中要删除元组主码值相对应的元组一起删除。

- **受限删除 (RESTRICTED)**

当参照关系中没有任何元组的外码值与要删除的被参照关系的元组的主码值相对应时，系统才执行删除操作，否则拒绝此删除操作。



# 违约反应(续)

- **置空值删除 (NULLIFIES)**

删除被参照关系的元组，并将参照关系中与被参照关系中被删除元组主码值相等的外码值置为空值。

- 这三种处理方法，哪一种是正确的，要依应用环境的语义来定



# 四、完整性约束的表达方式

- 1、在创建和修改基表模式时说明约束
- 2、用断言说明约束
- 3、用触发器表示约束
- 4、用过程说明约束



# 本节重点

- 数据库的安全性
  - 常用策略
- 数据库的完整性
  - 完整性约束的种类
  - 保证完整性约束的技术
  - 对违反完整性约束的处理策略

