



数据库系统概论

An Introduction to Database System

第四章 数据库安全性



数据库安全性



■ 问题的提出

- 数据库的一大特点是数据可以共享
- 数据共享必然带来数据库的安全性问题
- 数据库系统中的数据共享不能是无条件的共享

例： 军事秘密、国家机密、新产品实验数据、
市场需求分析、市场营销策略、销售计划、
客户档案、医疗档案、银行储蓄数据



数据库安全性



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，以保护计算机系统中的硬件、软件及数据，防止其因偶然或恶意的原因使系统遭到破坏，数据遭到更改或泄露等。

➤ 1. 数据库层次

➤ 2. 操作系统层次

➤ 3. 网络层次

➤ 4. 人员层次

➤ 5. 物理层次

物理层次的安全性



- 保护设备免于水灾和电源故障;
- 保护硬盘免于被偷、数据被删除和物理损坏;
- 保护网络和电缆免于被窃听 (wiretaps), 偷听 (non-invasive electronic eavesdropping) 和物理破坏

措施:

- 冗余设备
- 物理安全: 加锁等
- 检测物理安全的软件技术



人员层次的安全性



- 避免密码被偷、被破坏;

- 主要是管理问题

- 措施:

- 经常变更密码

- 用不可猜测密码

- 记录所有无效访问企图

- 数据审计



网络层次的安全性



- 每个站点都与可信站点通信;
- 站点链接不能够被盗用或消息修改

- 措施:



➤ 鉴别协议（基于口令的）



➤ 密码



操作系统层次的安全性

- 1. 防止非法登陆
- 2. 文件级的访问保护
- 3. 防止不恰当的超级用户授权
- 4. 防止不恰当的特权指令的使用



数据库层次的安全性

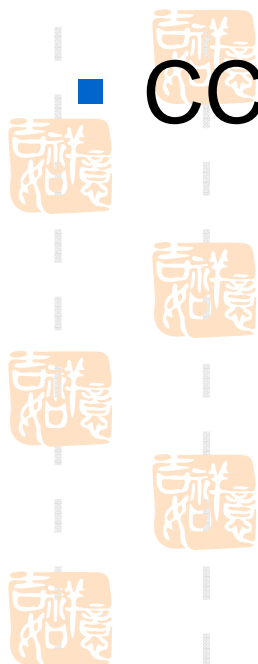
- 在物理层次，人员层次，网络层次，操作系统层次安全的基础上，才能保证数据库层次的安全性。
- 数据库的安全性保证数据库不被非法访问，它关注：
 - 每个用户可以授权读/写 部分数据(关系或文件)；
 - 站点层次的授权控制出现在分布式数据库系统。

4.1.2 安全标准简介



- TCSEC标准

- CC标准



安全标准简介（续）



■ TCSEC/TDI标准的基本内容

➤ TCSEC/TDI，从四个方面来描述安全性级别划分的指标

 ➤ 安全策略

 ➤ 责任

 ➤ 保证

 ➤ 文档



TCSEC/TDI安全级别划分

■ TCSEC/TDI安全级别划分

安全级别	定义
A1	验证设计（ Verified Design ）
B3	安全域（ Security Domains ）
B2	结构化保护（ Structural Protection ）
B1	标记安全保护（ Labeled Security Protection ）
C2	受控的存取保护（ Controlled Access Protection ）
C1	自主安全保护（ Discretionary Security Protection ）
D	最小保护（ Minimal Protection ）

按系统可靠或可信程度逐渐增高
各安全级别之间：偏序向下兼容

TCSEC/TDI安全级别划分（续）

■ B2以上的系统

➤ 还处于理论研究阶段

➤ 应用多限于一些特殊的部门，如军队等

➤ 美国正在大力发展安全产品，试图将目前仅限于少数领域应用的**B2**安全级别下放到商业应用中，并逐步成为新的商业标准

CC



■ CC

➤ 提出国际公认的表述信息技术安全性的结构

➤ 把信息产品的安全要求分为

➤ 安全功能要求

➤ 安全保证要求

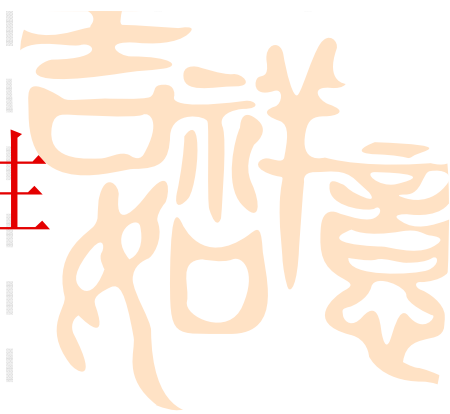


CC (续)

■ CC评估保证级划分

评估保证级	定 义	TCSEC安全级别（近似相当）
EAL1	功能测试（functionally tested）	
EAL2	结构测试（structurally tested）	C1
EAL3	系统地测试和检查（methodically tested and checked）	C2
EAL4	系统地设计、测试和复查（methodically designed, tested, and reviewed）	B1
EAL5	半形式化设计和测试（semiformally designed and tested）	B2
EAL6	半形式化验证的设计和测试（semiformally verified design and tested）	B3
EAL7	形式化验证的设计和测试（formally verified design and tested）	A1

第四章 数据库安全性



4.1 计算机安全性概述

4.2 数据库安全性控制

4.3 视图机制

4.4 审计 (**Audit**)

4.5 数据加密

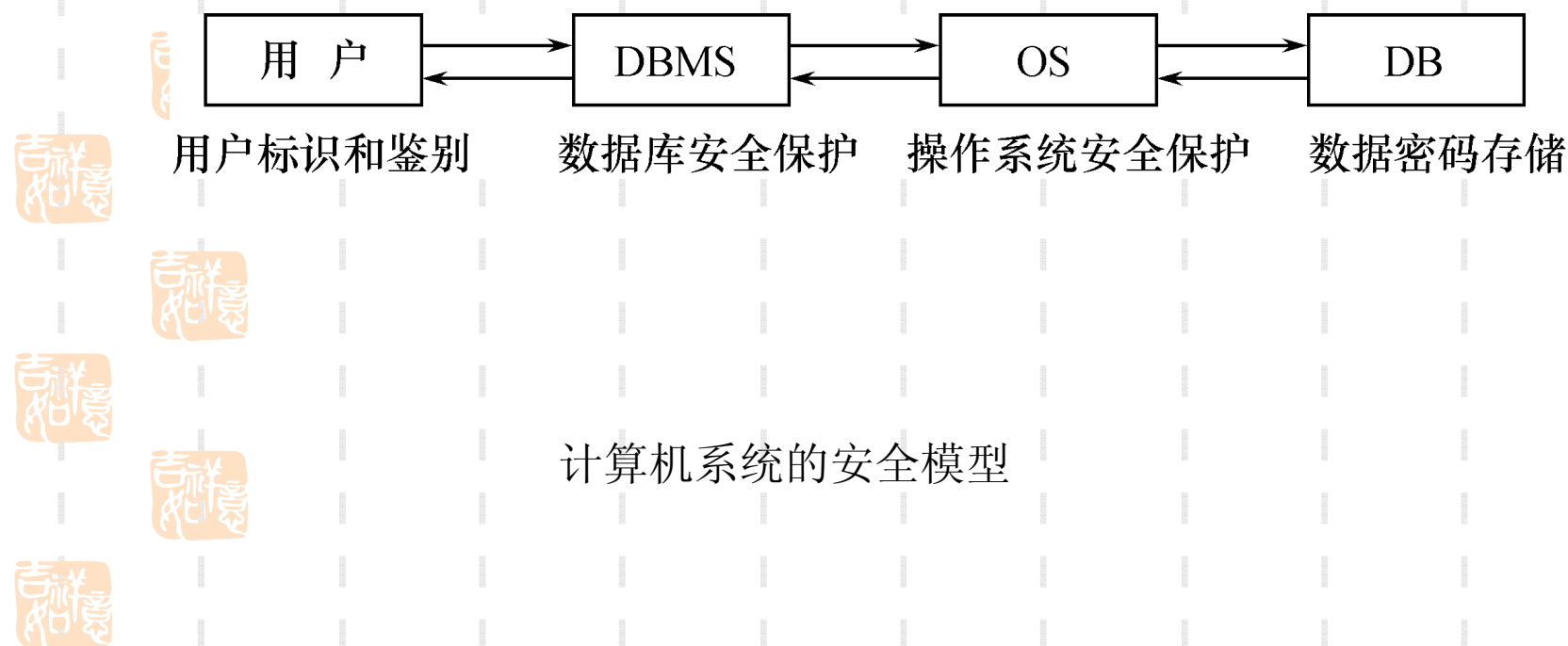
4.6 统计数据库安全性

4.7 小结



数据库安全性控制概述（续）

- 计算机系统中，安全措施是一级一级层层设置



数据库安全性控制概述（续）

- 数据库安全性控制的常用方法

- 用户标识和鉴别

- 存取控制

- 视图

- 审计

- 加密存储

4.2.1 用户标识与鉴别

- 用户标识与鉴别

(Identification & Authentication)



➤ 系统提供的最外层安全保护措施



■ 口令



➤ 系统核对口令以鉴别用户身份



4.2.2 存取控制



- 存取控制机制组成

- 定义用户权限

- 合法权限检查



存取控制（续）



■ 常用存取控制方法

➤ 自主存取控制（Discretionary Access Control，简称DAC）

➤ C2级



➤ 灵活



➤ 强制存取控制（Mandatory Access Control，简称 MAC）



➤ B1级



➤ 严格



4.2.3 自主存取控制方法

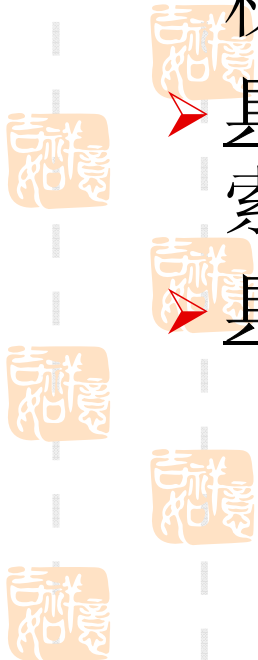
- 通过 SQL 的 **GRANT** 语句和 **REVOKE** 语句实现
- 定义存取权限称为**授权**



数据库用户类型



- 具有**CONNECT**权限的用户，只能连接数据库，成为一个数据库用户；这个用户最初除了连接，无其它任何权限，必须通过其它授权操作来增加权限。
- 具有**RESOURCE**权限的用户，可以创建表和索引，并可以控制这些资源的授权；
- 具有**DBA**特权的用户，有任何权限。



授权



- 授权：给予用户一定的访问特权
- 在SQL中，有两种授权
 - 授予某类数据库用户的特权，只能由DBA授予
 - 授予对某些数据对象进行某些操作的特权，可由DBA授予，也可由数据对象的创建者授予

■ 权限种类

➤ 对象权限

是指用户对数据库中的表、视图等对象中数据的操作权。

➤ 语句权限(SQL Server)/系统权限(Oracle)

相当于数据定义语言（DDL）的语句权限，这种权限专指是否允许执行：**CREATE TABLE**、**CREATE VIEW**等与创建数据库对象有关的操作。



第1种授权

授权:

GRANT <特权> **ON** 〈表名〉 **TO** <受权者>[, <受权者>]
[**WITH GRANT OPTION**]

特权: **ALL PRIVILEGES** |操作

操作: **SELECT** |**INSERT**|**DELETE**|**UPDATE** [属性表]

受权者: **PUBLIC** |用户标识符

收回特权:

REVOKE <特权> **ON** 〈表名〉 **FROM** <受权者>[, <受权者>]
>]

GRANT



➤ 发出GRANT:

➤ DBA

➤ 数据库对象创建者（即属主Owner）

➤ 拥有该权限的用户



➤ 按受权限的用户



➤ 一个或多个具体用户



➤ PUBLIC（全体用户）

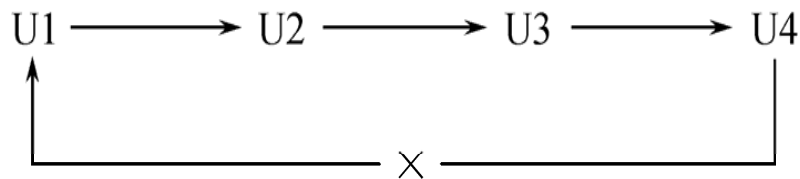


WITH GRANT OPTION子句

- WITH GRANT OPTION子句:

- 指定: 可以再授予
- 没有指定: 不能传播

- 不允许循环授权



例题

[例1] 把查询Student表权限授给用户U1

GRANT SELECT

ON TABLE Student

TO U1;

例题（续）

[例2] 把对Student表和Course表的全部权限授予用户U2和U3

GRANT ALL PRIVILIGES

ON TABLE Student, Course

TO U2, U3;

例题（续）

[例4] 把查询Student表和修改学生学号的权限授给用户U4

```
GRANT UPDATE(Sno), SELECT
```

```
ON TABLE Student
```

```
TO U4;
```

- 对属性列的授权时必须明确指出相应属性列名

授权的例子



■ DBA:

GRANT select, insert ON students TO CS3
WITH GRANT OPTION



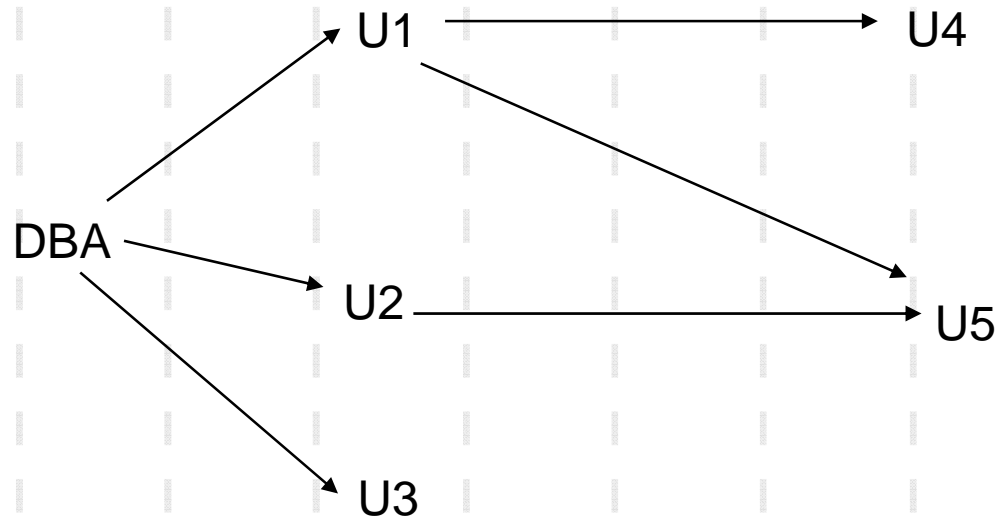
■ CS3:

GRANT select ON students TO li4



权限转授

- 采用授权图表示权限的传递。
- 使用[with grant option]



- DBA:
Grant update on branch to U1,U2,U3 with grant option
- U1:
Grant update on branch to U4,U5
- U2:
Grant update on branch to U5

REVOKE

[例8] 把用户U4修改学生学号的权限收回

REVOKE UPDATE(Sno)



ON TABLE Student



FROM U4;



第2种授权



GRANT 权限名 [, ...]

TO {数据库用户名 | 用户角色名} [, ...]

收权语句

REVOKE 权限名 [, ...]

FROM { 数据库用户名 | 用户角色名 }
[, ...]



定义用户权限和授权的例子

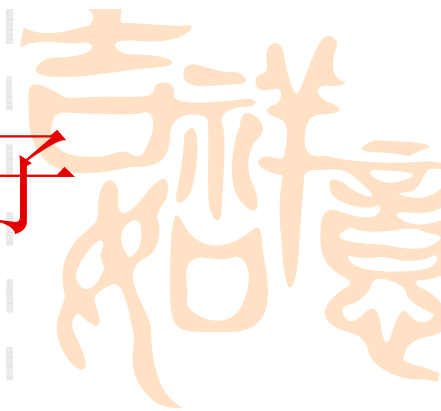
- 授予**user1**具有创建数据库表的权限。
 - **GRANT CREATE TABLE TO user1**
- 授予**user1**和**user2**具有创建数据库表和视图的权限。
 - **GRANT CREATE TABLE, CREATE VIEW TO user1, user2**



收回授权的命令和例子

例子:

REVOKE CREATE TABLE FROM user1



小结:SQL灵活的授权机制

- **DBA:** 拥有所有对象的所有权限

- 不同的权限授予不同的用户

- **用户:** 拥有自己建立的对象的全部的操作权限

- **GRANT:** 授予其他用户

- **被授权的用户**

- “继续授权” 许可: 再授予

- 所有授予出去的权力在必要时又都可用**REVOKE**语句

- 收回

4.2 数据库安全性控制



4.2.1 用户标识与鉴别

4.2.2 存取控制

4.2.3 自主存取控制方法

4.2.4 授权与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法



4.2.5 数据库角色



- 数据库角色：被命名的一组与数据库操作相关的权限

➤ 角色是权限的集合

➤ 可以为一组具有相同权限的用户创建一个角色

➤ 简化授权的过程



数据库角色




- 一、角色的创建

CREATE ROLE <角色名>

- 二、给角色授权

 GRANT <权限> [, <权限>] ...

 ON <对象类型>对象名

 TO <角色> [, <角色>] ...





数据库角色



- 三、将一个角色授予其他的角色或用户

GRANT <角色1> [, <角色2>] ...

TO <角色3> [, <用户1>] ...

[WITH ADMIN OPTION]



- 四、角色权限的收回

REVOKE <权限> [, <权限>] ...



ON <对象类型> <对象名>



FROM <角色> [, <角色>] ...



数据库角色（续）

[例11] 通过角色来实现将一组权限授予一个用户。

步骤如下：

1. 首先创建一个角色 R1

```
CREATE ROLE R1;
```

2. 使角色R1拥有Student表的SELECT、UPDATE、INSERT权限

```
GRANT SELECT, UPDATE, INSERT  
ON TABLE Student  
TO R1;
```

数据库角色（续）

3. 将这个角色授予王平，张明，赵玲。使他们具有角色R1所包含的全部权限

GRANT R1 TO 王平，张明，赵玲；

4. 可以一次性通过R1来回收王平的这3个权限

REVOKE R1 FROM 王平；

4.2 数据库安全性控制

4.2.1 用户标识与鉴别

4.2.2 存取控制

4.2.3 自主存取控制方法

4.2.4 授权与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法

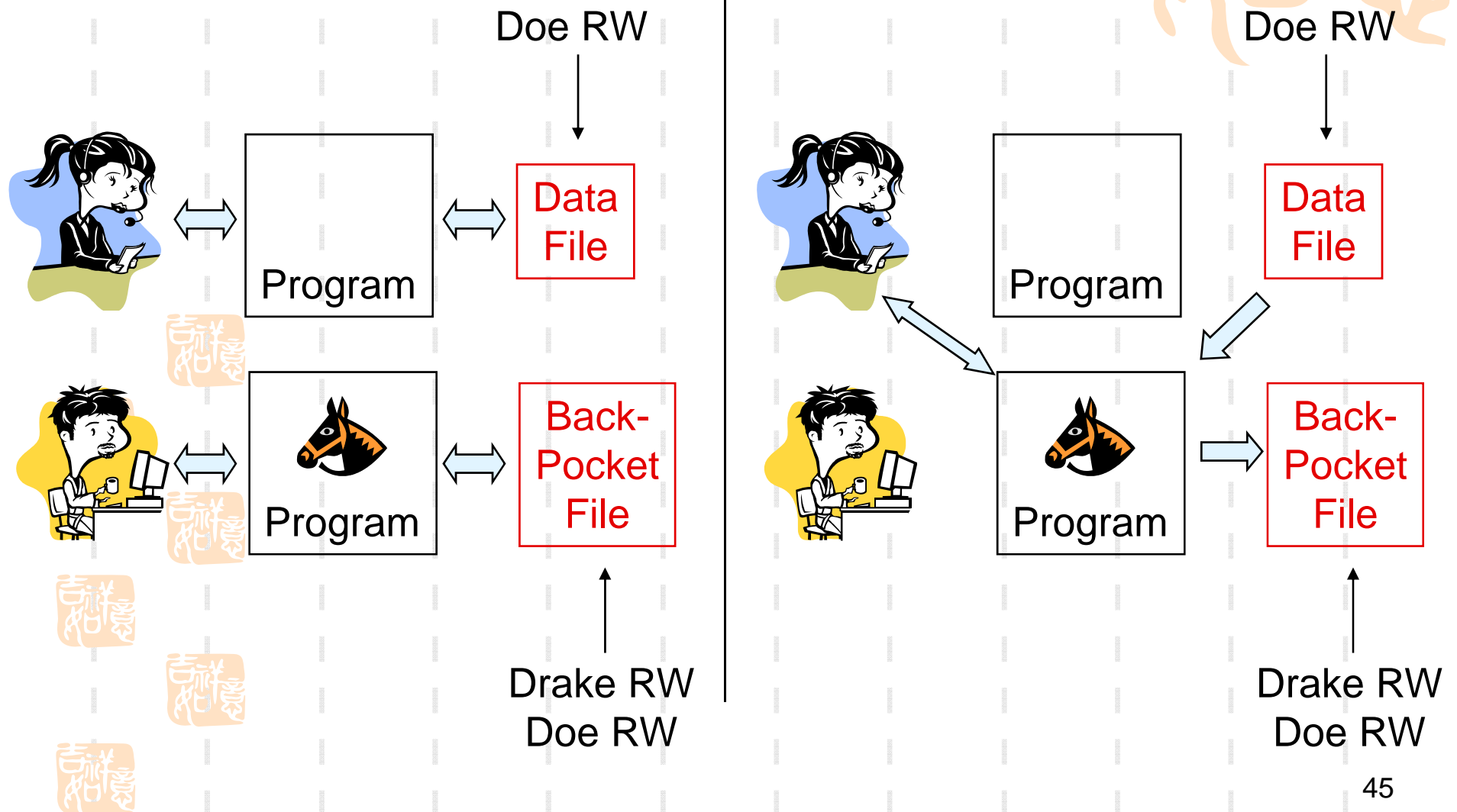
自主存取控制缺点



- 可能存在数据的“无意泄露”
- 原因：这种机制仅仅通过对数据的存取权限来进行安全控制，而数据本身并无安全性标记
- 解决：对系统控制下的所有主客体实施强制存取控制策略



Trojan Horse – Access Control



4.2.6 强制存取控制方法

- 强制存取控制（MAC）

- 保证更高层次的安全性

- 用户不能直接感知或进行控制

- 适用于对数据有严格而固定密级分类的部门

- 军事部门

- 政府部门

强制存取控制方法（续）

- **主体**是系统中的活动实体

- DBMS所管理的实际用户
- 代表用户的各进程

- **客体**是系统中的被动实体，是受主体操纵的

- 文件
- 基表
- 索引
- 视图

强制存取控制方法（续）

- 敏感度标记（Label）

- 绝密（Top Secret）
- 机密（Secret）
- 可信（Confidential）
- 公开（Public）

- 主体的敏感度标记称为许可证级别（Clearance Level）

- 客体的敏感度标记称为密级（Classification Level）

强制存取控制方法（续）

■ 强制存取控制规则

(1) 仅当主体的许可证级别 **大于或等于** 客体的密级时，该主体才能 **读** 取相应的客体

(2) 仅当主体的许可证级别 **等于** 客体的密级时，该主体才能 **写** 相应的客体

■ 修正规则

➤ 主体的许可证级别 \leq 客体的密级 \rightarrow 主体能写客体

强制存取控制方法（续）



- 规则的共同点

禁止了拥有高许可证级别的主体更新低密
级的数据对象



Control

Doe RW



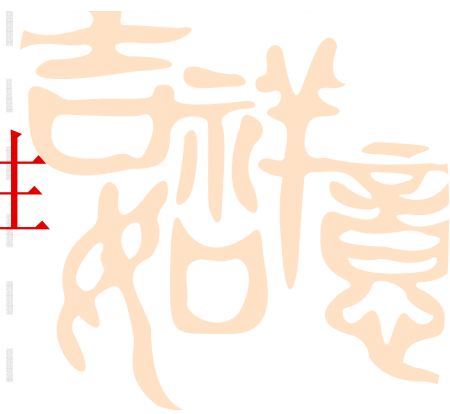
MAC与DAC



- DAC与MAC共同构成DBMS的安全机制
- 实现MAC时要首先实现DAC



第四章 数据库安全性



4.1 计算机安全性概述

4.2 数据库安全性控制

4.3 视图机制

4.4 审计 (**Audit**)

4.5 数据加密

4.6 统计数据库安全性

4.7 小结

4.3 视图机制

- 把要保密的数据对无权存取这些数据的用户隐藏起来，对数据提供一定程度的安全保护



视图机制（续）

[例14]建立计算机系学生的视图，把对该视图的
SELECT权限授予王平，把该视图上的所有操作
权限授予张明

先建立计算机系学生的视图CS_Student

```
CREATE VIEW CS_Student
```

```
AS
```

```
SELECT *
```

```
FROM Student
```

```
WHERE Sdept='CS';
```

视图机制（续）



在视图上进一步定义存取权限

GRANT SELECT
ON CS_Student
TO 王平；

GRANT ALL PRIVILIGES
ON CS_Student
TO 张明；



4.2 数据库安全性控制

4.1 计算机安全性概述

4.2 数据库安全性控制

4.3 视图机制

4.4 审计 (Audit)

4.5 数据加密

4.6 统计数据库安全性

4.7 小结

4.4 审计



■ 什么是审计

➤ 审计日志 (Audit Log)

将用户对数据库的所有操作记录在上面

➤ DBA利用审计日志

找出非法存取数据的人、时间和内容

➤ C2以上安全级别的DBMS必须具有



审计（续）



■ 审计分为

➤ 用户级审计

- 针对自己创建的数据库表或视图进行审计
- 记录所有用户对这些表或视图的一切成功和（或）不成功
的访问要求以及各种类型的**SQL**操作

➤ 系统级审计

➤ DBA设置

- 监测成功或失败的登录要求
- 监测**GRANT**和**REVOKE**操作以及其他数据库级权限下的
操作

审计（续）



- AUDIT语句：设置审计功能

- NOAUDIT语句：取消审计功能



审计（续）



[例15] 对修改SC表结构或修改SC表数据的操作
进行审计

AUDIT ALTER, UPDATE

ON SC;

[例16] 取消对SC表的一切审计

NOAUDIT ALTER, UPDATE

ON SC;

4.2 数据库安全性控制

4.1 计算机安全性概述

4.2 数据库安全性控制

4.3 视图机制

4.4 审计 (**Audit**)

4.5 数据加密

4.6 统计数据库安全性

4.7 小结

4.5 数据加密



- 数据加密

- 防止数据库中数据在存储和传输中失密的有效手段

- 加密方法

- 对称加密



- 不对称加密

- DBMS中的数据加密



- 加密和解密占用大量系统资源，应只对高度机敏的数据加密



第四章 数据库安全性



4.1 计算机安全性概述

4.2 数据库安全性控制

4.3 视图机制

4.4 审计 (**Audit**)

4.5 数据加密

4.6 统计数据库安全性

4.7 小结

4.6 统计数据库安全性

- 有些数据库中单个记录是保密的，但综合性数据是公开的（如一些相关的聚集数据，如总数、求和、平均值等允许外人访问）
- 提供用户综合性数据为主的数据库称为统计数据库
- 统计数据库中特殊的安全性问题
 - 从综合性数据推断单个记录的值
- 例子：Q1：本公司有多少女高级程序员？
Q2：本公司女高级程序员的工资总额是多少？

统计数据库安全性（续）



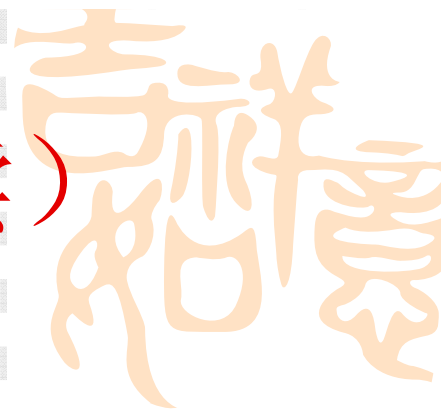
规则1：任何查询至少要涉及 N (N 足够大)个以上的记录

规则2：任意两个查询的相交数据项不能超过 M 个

规则3：任一用户的查询次数不能超过 $1 + (N-2)/M$



统计数据库安全性（续）



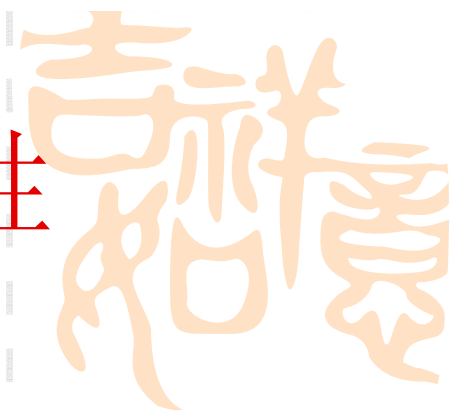
- 数据库安全机制的设计目标：

试图破坏安全的人所花费的代价 >> 得到的利

益



第四章 数据库安全性



4.1 计算机安全性概述

4.2 数据库安全性控制

4.3 视图机制

4.4 审计 (**Audit**)

4.5 数据加密

4.6 统计数据库安全性

4.7 小结

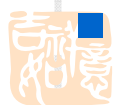


4.7 小结



- 数据的共享日益加强，数据的安全保密越来越重要

- DBMS是管理数据的核心，因而其自身必须具有一整套完整而有效的安全性机制



小结（续）



- 实现数据库系统安全性的技术和方法
 - 存取控制技术
 - 视图技术
 - 审计技术
- 自主存取控制功能
 - 通过SQL 的GRANT语句和REVOKE语句实现
- 角色
 - 使用角色来管理数据库权限可以简化授权过程
 - CREATE ROLE语句创建角色
 - GRANT 语句给角色授权



作业

- 6
- 8(a),(b),(f),(g)
- 9(对应于8(a),(b),(f),(g))

