



哈爾濱工業大學

Harbin Institute of Technology

数据库系统

万晓珑 博士
大数据计算研究中心

wxl@hit.edu.cn



哈爾濱工業大學

Harbin Institute of Technology

数据库系统

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

数据库安全性

❖ 安全性问题案例

- **Facebook**数据泄露，**6800**万用户个人信息。
- 某知名企业涉嫌侵犯数百亿条公民个人信息
- 新三板挂牌公司涉窃取**30**亿条个人信息 非法操控公众账号加粉或关注
- 圆通**10**亿快递信息泄露
- 某连锁酒店**5**亿条用户数据泄露

数据库安全性（续）

- 数据库的安全性是指保护数据库以**防止不合法使用**所造成的数据泄露、更改或破坏。
- 系统安全保护措施是否有效是数据库系统主要的性能指标之一。

第四章 数据库安全性

4.1 数据库安全性概述

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 其他安全性保护

4.7 小结

4.1 数据库安全性概述

4.1.1 数据库的不安全因素

4.1.2 安全标准简介

4.1.1 数据库的不安全因素

1. 非授权用户对数据库的恶意存取和破坏

- 一些黑客（**Hacker**）和犯罪分子在用户存取数据库时猎取用户名和用户口令，然后假冒合法用户偷取、修改甚至破坏用户数据。
- 数据库管理系统提供的安全措施主要包括**用户身份鉴别、存取控制和视图**等技术。

数据库的不安全因素（续）

2.数据库中重要或敏感的数据被泄露

- 黑客和敌对分子千方百计盗窃数据库中的重要数据，一些机密信息被暴露。
- 数据库管理系统提供的主要技术有强制存取控制、数据加密存储和加密传输等。
- 审计日志分析

数据库的不安全因素（续）

3.安全环境的脆弱性

- 数据库的安全性与计算机系统的安全性紧密联系
 - 计算机硬件、操作系统、网络系统等的安全性
- 建立一套可信（**Trusted**）计算机系统的概念和标准

4.1 数据库安全性概述

4.1.1 数据库的不安全因素

4.1.2 安全标准简介

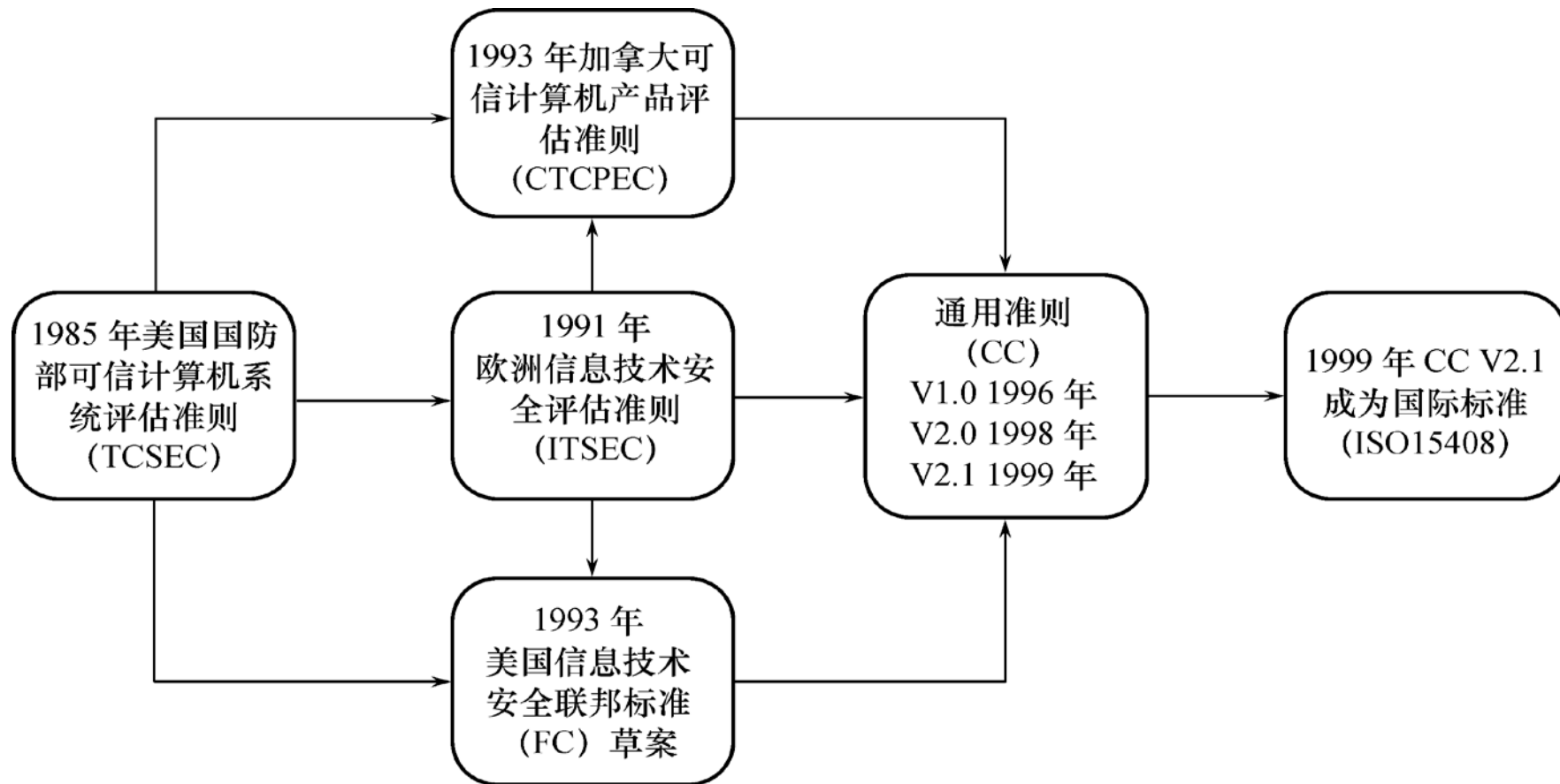
4.1.2 安全标准简介

- ❖ 1985年美国国防部（**DoD**）正式颁布《**DoD可信计算机系统评估准则**》（简称**TCSEC**或**DoD85**）
- ❖ 不同国家建立在**TCSEC**概念上的评估准则
 - 欧洲的信息技术安全评估准则（**ITSEC**）
 - 加拿大的可信计算机产品评估准则（**CTCPEC**）
 - 美国的信息技术安全联邦标准（**FC**）

安全标准简介（续）

- ❖ 1993年，**CTCPEC、FC、TCSEC**和**ITSEC**联合行动，解决原标准中概念和技术的差异，称为**CC（Common Criteria）**项目
- ❖ 1999年 **CC V2.1**版被**ISO**采用为国际标准
- ❖ 2001年 **CC V2.1**版被我国采用为国家标准
- ❖ 目前**CC**已基本取代了**TCSEC**，成为评估信息产品安全性的主要标准。

4.1.2 安全标准简介



信息安全标准的发展历史

第四章 数据库安全性

4.1 数据库安全性概述

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 其他安全性

4.7 小结

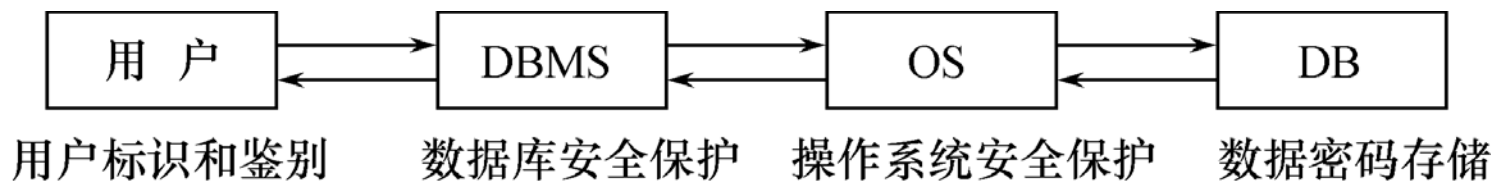
4.2 数据库安全性控制

❖ 非法使用数据库的情况

- 编写合法程序绕过数据库管理系统及其授权机制
- 直接或编写应用程序执行非授权操作
- 通过多次合法查询数据库并推导出一些保密数据

数据库安全性控制（续）

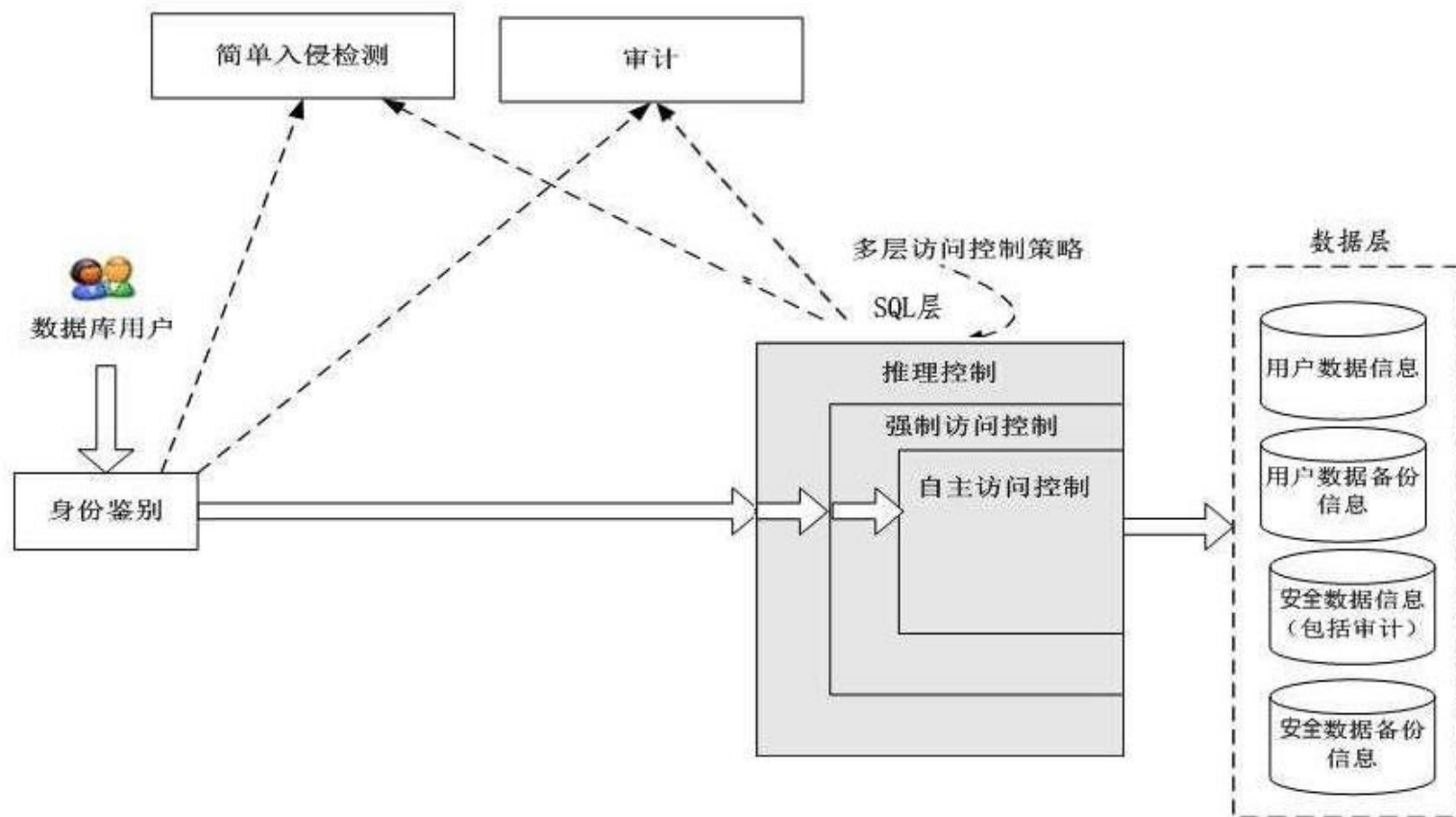
- 计算机系统中，安全措施是一级一级层层设置



计算机系统的安全模型

- 系统根据用户标识鉴定用户身份，合法用户准许进入计算机系统
- 数据库管理系统进行存取控制，只允许用户执行合法操作
- 操作系统有自己的保护措施
- 数据以密码形式存储到数据库中

数据库安全性控制（续）



数据库管理系统安全性控制模型

数据库安全性控制（续）

❖ 存取控制流程

- 首先，数据库管理系统对提出**SQL**访问请求的数据库用户进行**身份鉴别**，防止不可信用户使用系统。
- 然后，在**SQL**处理层进行**自主存取控制**和**强制存取控制**，进一步可以进行**推理控制**。
- 还可以对用户访问行为和系统关键操作进行**审计**，对异常用户行为进行简单入侵检测。

数据库安全性控制（续）

❖ 数据库安全性控制的常用方法

- 用户标识和鉴定
- 存取控制
- 视图
- 审计
- 数据加密

4.2 数据库安全性控制

4.2.1 用户身份鉴别

4.2.2 存取控制

4.2.3 自主存取控制方法

4.2.4 授权：授予与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法

4.2.1 用户身份鉴别

❖ 用户身份鉴别 (**Identification & Authentication**)

- 系统提供的最外层安全保护措施
- 用户标识：由用户名和用户标识号组成
(用户标识号在系统整个生命周期内唯一)

用户身份鉴别（续）

❖ 用户身份鉴别的方法

1. **静态口令**鉴别：一般由用户自己设定，静态不变的
2. **动态口令**鉴别：口令是动态变化的，每次鉴别时均需使用动态产生的新口令登录数据库管理系统，即采用一次一密的方法
3. **生物特征**鉴别：通过生物特征进行认证的技术，生物特征如指纹、虹膜和掌纹等
4. **智能卡**鉴别：智能卡是一种不可复制的硬件，内置集成电路的芯片，具有硬件加密功能

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：授予与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法

4.2.2 存取控制

❖ 存取控制机制组成

- 定义用户权限(即对某一数据对象的操作权力)，并登记到数据字典。
- 合法权限检查
 - 用户发出存取数据库操作请求
 - **DBMS**查找数据字典，进行合法权限检查

❖ 用户权限定义和合法权限检查机制一起组成数据库管理系统的存取控制子系统

存取控制（续）

❖ 常用存取控制方法

■ 自主存取控制（Discretionary Access Control）

- 用户对不同的数据对象有不同的存取权限
- 不同的用户对同一对象也有不同的权限
- 用户还可将其拥有的存取权限转授给其他用户

■ 强制存取控制（Mandatory Access Control）

- 每一个数据对象被标以一定的密级
- 每一个用户也被授予某一个级别的许可证
- 对于任意一个对象，只有具有合法许可证的用户才可以存取

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：授予与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法

4.2.3 自主存取控制方法

- ❖ 通过 SQL 的 **GRANT** 语句和 **REVOKE** 语句实现
- ❖ 用户权限组成
 - 数据对象
 - 操作类型
- ❖ 定义用户存取权限：定义用户可以在哪些数据库对象上进行哪些类型的操作
- ❖ 定义存取权限称为 **授权**

自主存取控制方法（续）

❖ 关系数据库系统中存取控制对象

| 对象类型 | 对象 | 操作类型 |
|---------------|--------|---|
| 数据库 模式 | 模式 | CREATE SCHEMA |
| | 基本表 | CREATE TABLE, ALTER TABLE |
| | 视图 | CREATE VIEW |
| | 索引 | CREATE INDEX |
| 数据 | 基本表和视图 | SELECT, INSERT, UPDATE, DELETE, REFERENCES, ALL PRIVILEGES |
| | 属性列 | SELECT, INSERT, UPDATE, REFERENCES, ALL PRIVILEGES |

关系数据库系统中的存取权限

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：授予与回收

4.2.5 数据库角色

4.2.6 强制存取控制方法

4.2.4 授权：授予与回收

1. GRANT

❖ GRANT语句的一般格式：

GRANT <权限>[,<权限>]...

ON <对象类型> <对象名>[,<对象类型> <对象名>]...

TO <用户>[,<用户>]...

[WITH GRANT OPTION];

❖ 语义：将对指定操作对象的指定操作权限授予指定的用户

GRANT (续)

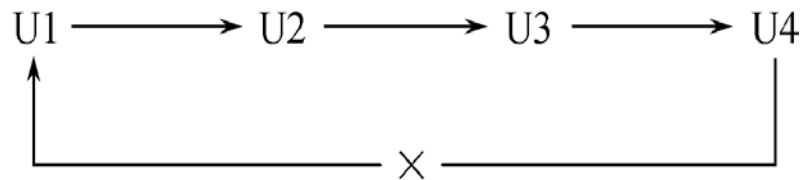
- 发出**GRANT**:
 - 数据库管理员
 - 数据库对象创建者 (即属主**Owner**)
 - 拥有该权限的用户
- 接受权限的用户
 - 一个或多个具体用户
 - **PUBLIC** (即全体用户)

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4.1] 把查询Student表权限授给用户U1

```
GRANT  SELECT  
ON TABLE  Student  
TO  U1;
```

**[例4.2] 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4.3] 把对表**SC**的查询权限授予所有用户

GRANT SELECT ON TABLE SC TO PUBLIC;

[例4.4] 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

例题（续）

[例4.5] 把对表**SC**的**INSERT**权限授予**U5**用户，并允许他再将此权限授予其他用户

GRANT INSERT ON TABLE SC TO U5
WITH GRANT OPTION;

U5拥有对表**SC**的**INSERT**权限，

还可以传播此权限：

[例4.6] **GRANT INSERT**
ON TABLE SC
TO U6

同样，**U6**还可以将此权限授予**U7**：

[例4.7] **GRANT INSERT**
ON TABLE SC TO U7;

但**U7**不能再传播此权限。

授权：授予与回收（续）

2.REVOKE

❖ 授予的权限可以由数据库管理员或其他授权者用 **REVOKE** 语句收回

❖ **REVOKE** 语句的一般格式为：

REVOKE <权限>[,<权限>]...

ON <对象类型> <对象名>[,<对象类型><对象名>]...

FROM <用户>[,<用户>]...[**CASCADE** | **RESTRICT**];

REVOKE (续)

[例4.8] 把用户U4修改学生学号的权限收回

```
REVOKE UPDATE(Sno)  
ON TABLE Student  
FROM U4;
```

REVOKE (续)

[例4.9] 收回所有用户对表**SC**的查询权限

```
REVOKE SELECT  
ON TABLE SC  
FROM PUBLIC;
```

REVOKE (续)

[例4.10] 把用户U5对SC表的INSERT权限收回

```
REVOKE INSERT  
ON TABLE SC  
FROM U5 CASCADE ;
```

- 将用户U5的INSERT权限收回的时候应该使用**CASCADE**，否则拒绝执行该语句
- 如果U6或U7还从其他用户处获得对SC表的INSERT权限，则他们仍具有此权限，系统只收回直接或间接从U5处获得的权限

小结: SQL灵活的授权机制

- ❖ 数据库管理员：拥有所有对象的所有权限，根据实际情况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，可以使用**GRANT**，把权限授予其他用户
- ❖ 被授权的用户：如果具有“继续授权”的许可，可以把获得的权限再授予其他用户
- ❖ 所有授予出去的权力在必要时又都可用**REVOKE**语句收回

授权：授予与回收（续）

3.创建数据库模式的权限

❖ 数据库管理员在**创建用户**时实现

❖ **CREATE USER**语句格式

CREATE USER <username>

[WITH][DBA|RESOURCE|CONNECT];

注： **CREATE USER**不是**SQL**标准，各个系统的实现相差甚远

授权：授予与回收（续）

❖ **CREATE USER <username>**

[WITH][DBA|RESOURCE|CONNECT];

❖ **CREATE USER**语句格式说明

- 只有系统的**超级用户**才有权创建一个新的数据库用户
- 新创建的数据库用户有三种权限：**CONNECT**、**RESOURCE**和**DBA**
- 如没有指定创建的新用户的权限，默认该用户拥有**CONNECT**权限。拥有**CONNECT**权限的用户不能创建新用户，不能创建模式，也不能创建基本表，只能

授权：授予与回收（续）

❖ CREATE USER语句格式说明（续）

- 拥有**RESOURCE**权限的用户能创建基本表和视图，成为所创建对象的属主。但不能创建模式，不能创建新的用户
- 拥有**DBA**权限的用户是系统中的超级用户，可以创建新的用户、创建模式、创建基本表和视图等；**DBA**拥有对所有数据库对象的存取权限，还可以把这些权限授予一般用户

授权：授予与回收（续）

| 拥有的权限 | 可否执行的操作 | | | |
|-----------------|------------------------|--------------------------|-------------------------|-------------------------|
| | CREATE USER | CREATE SCHEMA | CREATE TABLE | 登录数据库， 执行数据查询和 操纵 |
| DBA | 可以 | 可以 | 可以 | 可以 |
| RESOURCE | 不可以 | 不可以 | 可以 | 可以 |
| CONNECT | 不可以 | 不可以 | 不可以 | 可以，但必须拥有相应权限 |

权限与可执行的操作对照表

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：授予与回收

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（续）

1.角色的创建

CREATE ROLE <角色名>

2.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（续）

3. 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]

- 该语句把角色授予某用户，或授予另一个角色
- 授予者是角色创建者或拥有该角色的**ADMIN OPTION**
- 指定了**WITH ADMIN OPTION**则获得某种权限的角色或用户还可以把这种权限授予其他角色

一个角色的权限：直接授予这个角色的全部权限加上其他角色授予这个角色的全部权限

数据库角色（续）

4. 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>]...

- 用户可以回收角色的权限，从而修改角色拥有的权限
- **REVOKE**执行者是
 - 角色的创建者
 - 拥有在这个角色上的**ADMIN OPTION**

数据库角色（续）

[例4.11] 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) 然后使用**GRANT**语句，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;

数据库角色（续）

[例4.12] 角色的权限修改

GRANT DELETE

ON TABLE Student

TO R1;

使角色**R1**在原来的基础上增加了**Student**表的
DELETE 权限

数据库角色（续）

[例4.13]

REVOKE SELECT

ON TABLE Student

FROM R1;

使R1减少了**SELECT**权限

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

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）

- ❖ 在强制存取控制中，数据库管理系统所管理的全部实体被分为主体和客体两大类
- ❖ 主体是系统中的活动实体
 - 数据库管理系统所管理的实际用户
 - 代表用户的各进程
- ❖ 客体是系统中的被动实体，受主体操纵
 - 文件、基本表、索引、视图

强制存取控制方法（续）

❖ 敏感度标记（Label）

- 对于主体和客体，**DBMS**为它们每个实例指派一个敏感度标记（Label）
- 敏感度标记分若干级别：绝密（**Top Secret, TS**），机密（**Secret, S**），可信（**Confidential, C**），公开（**Public, P**）， $TS \geq S \geq C \geq P$

❖ 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

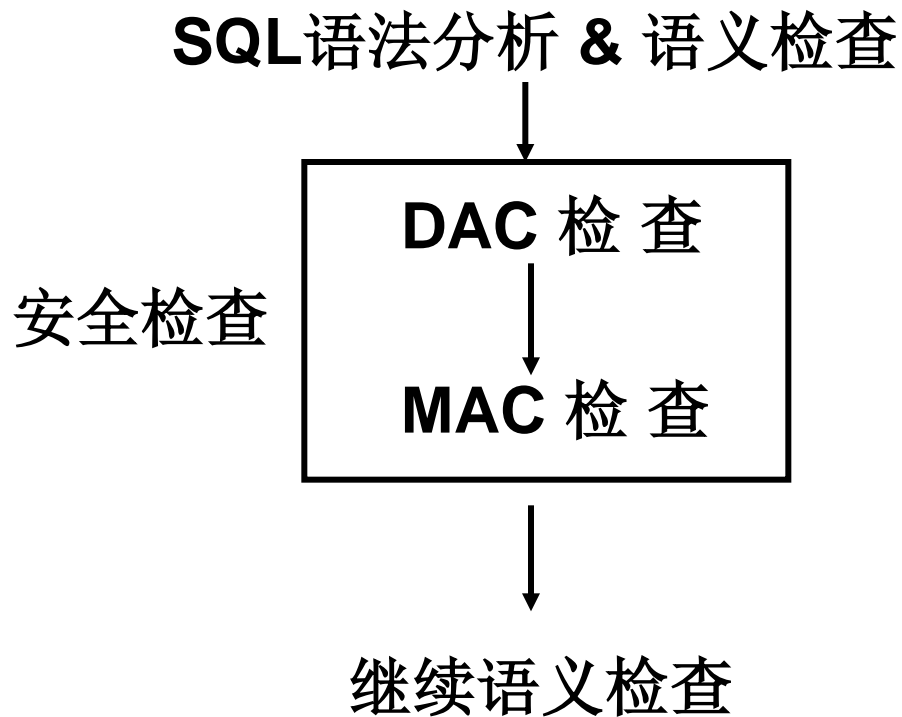
(2) 仅当主体的许可证级别**小于或等于**客体的密级时，该主体才能**写**相应的客体。

(**S**级别的用户，读取**S**级别数据，写为**P**级别数据，所有人都可以读取该数据，出现泄露问题。该**S**级用户，只能把数据写为**S**级别或者**TS**级别，不存在数据泄露的问题。)

强制存取控制方法（续）

- ❖ 强制存取控制（**MAC**）是对数据本身进行密级标记，无论数据如何复制，**标记与数据是一个不可分的整体**，只有符合密级标记要求的用户才可以操纵数据。
- ❖ 实现强制存取控制时要首先实现自主存取控制
 - 原因：较高安全性级别提供的安全保护要包含较低级别的所有保护
- ❖ 自主存取控制与强制存取控制共同构成数据库管理系统的安全机制

DAC + MAC安全检查



先进行自主存取控制检查，通过自主存取控制检查的数据对象再由系统进行强制存取控制检查，只有通过强制存取控制检查的数据对象方可存取。

第四章 数据库安全性

4.1 数据库安全性概述

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 其他安全性保护

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
- ❖ 间接地实现支持存取谓词的用户权限定义

视图机制（续）

[例4.14] 建立计算机系学生的视图，把对该视图**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;

第四章 数据库安全性

4.1 数据库安全性概述

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 其他安全性保护

4.7 小结

4.4 审计

❖ 什么是审计

- 启用一个专用的审计日志（**Audit Log**），将用户对数据库的所有操作记录在上面
- 审计员利用审计日志监控数据库中的各种行为，找出非法存取数据的人、时间和内容
- **C2**以上安全级别的**DBMS**必须具有审计功能

审计（续）

❖ 审计功能的可选性

- 审计很费时间和空间
- **DBA**可以根据应用对安全性的要求，灵活地打开或关闭审计功能
- 审计功能主要用于安全性要求较高的部门

审计（续）

1. 审计事件的类别

- 服务器事件：审计数据库服务器发生的事件
- 系统权限：对系统拥有的结构或模式对象进行操作的审计，要求该操作的权限是通过系统权限获得
- 语句事件：对**SQL**语句，如**DDL**、**DML**、**DQL**及**DCL**语句的审计
- 模式对象事件：对特定模式对象(表、视图、存储过程、函数等)上进行的**SELECT**或**DML**操作的审计

审计（续）

2. AUDIT语句和NOAUDIT语句

- **AUDIT**语句：设置审计功能
- **NOAUDIT**语句：取消审计功能

[例4.15] 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4.16] 取消对**SC**表的一切审计

NOAUDIT ALTER,UPDATE ON SC;

第四章 数据库安全性

4.1 数据库安全性概述

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 其他安全性保护

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

❖ 加密的基本思想

- 根据一定的算法将原始数据—明文(**Plain text**)变换为不可直接识别的格式—密文(**Cipher text**)

❖ 加密方法

- 存储加密：将数据在写到磁盘时对数据进行加密，授权用户读取数据时再对其进行解密
- 传输加密：链路加密和传输端加密。

第四章 数据库安全性

4.1 数据库安全性概述

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 其他安全性保护

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。
- ❖ 数据库管理系统是管理数据的核心，因而其自身必须具有一整套完整而有效的安全性机制。
- ❖ 实现数据库系统安全性的技术和方法
- ❖ 小测：请列举两项数据库系统安全性技术。

小结（续）

- 用户身份鉴别
- 存取控制技术：自主存取控制和强制存取控制
- 视图技术
- 审计技术
- 数据加密存储和加密传输