第四章 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
- 数据库系统中的数据共享不能是无条件共享

  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

- 数据共享必然带来数据库的**安全性问题**

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
- 4.5 数据加密
- 4.6 小结

4.1 数据库安全性概述

4.1.1 计算机系统的三类安全性问题

4.1.2 安全标准简介

4.1.3 数据库的不安全因素

4.1.1 计算机系统的三类安全性问题

- 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。

计算机系统的三类安全性问题

➢ 技术安全

指计算机系统中采用具有一定安全性的硬件、软件来实现对计算机系统及所存数据的安全保护，当计算机系统受到无意或恶意攻击时仍能保持系统正常运行，保证系统内的数据不增加、不丢失、不泄露

➢ 管理安全

防止由于管理不善导致的计算机设备和数据介质的物理破坏、丢失等软硬件意外故障以及场地的意外事故等安全问题

➢ 政策法律

指政府部门建立的有关计算机犯罪、数据安全保密的法律道德准则和政策法规、法令

4.1.2 安全标准简介

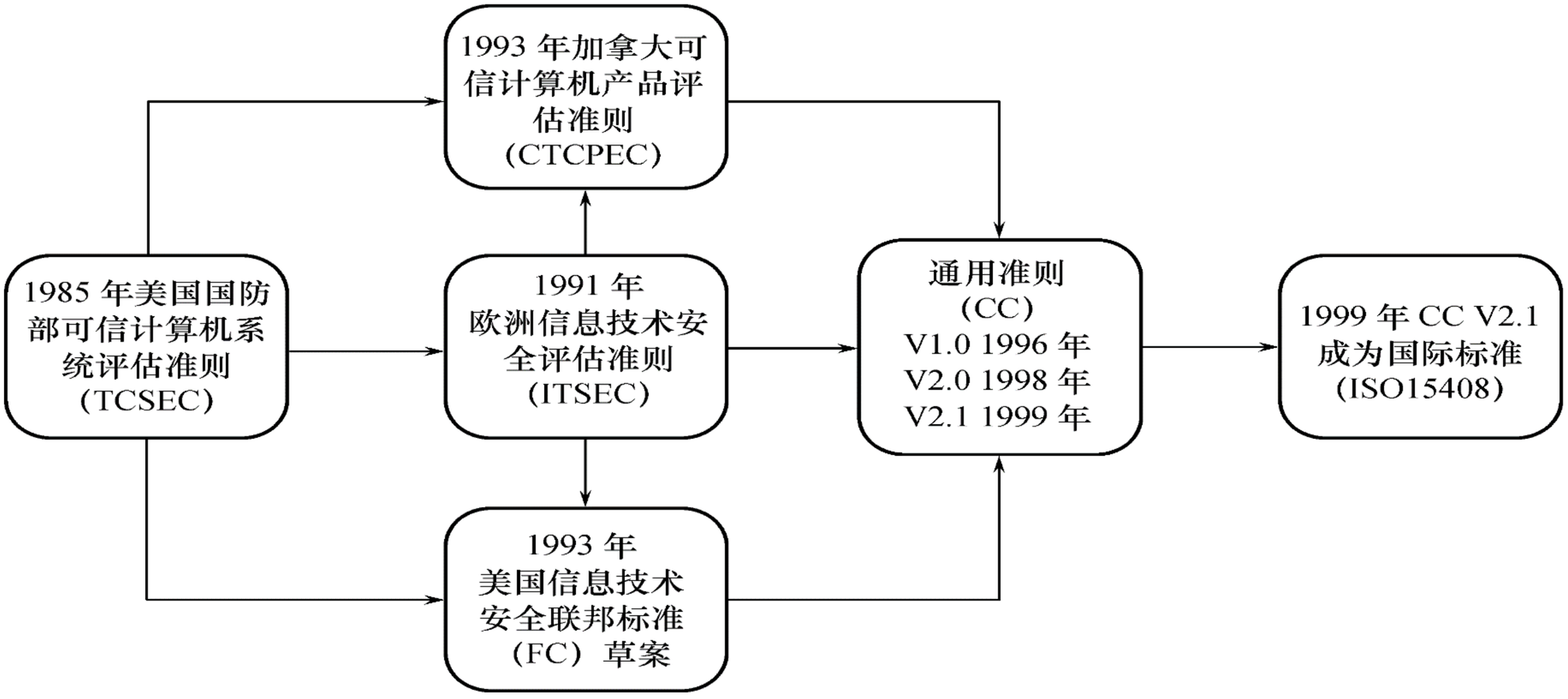
- **TCSEC标准 (Trusted Computer System Evaluation Criteria, TCSEC)**

  1985年美国国防部(DoD)正式颁布的《DoD可信计算机系统评估准则》

- **CC标准 (Common Criteria)**

  提出国际公认的表述信息技术安全性的结构，和早期的评估准则比，具有结构开放、表达方式通用的特点

安全标准简介（续）



信息安全标准的发展历史

安全标准简介（续）

1、TCSEC/TDI标准的基本内容

- 1991年4月美国国家计算机安全中心颁布了《可信计算机系统评估准则关于可信数据库系统的解释》（Trusted Database Interpretation, TDI），将TCSEC扩展到数据库管理系统

- TDI中定义了数据库管理系统的设计与实现中需满足和用以进行安全性级别评估的标准

- 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

2、TCSEC/TDI安全级别划分

根据计算机系统对各项指标的支持情况，将系统划分成4组7个等级，D、C(C1,C2)、B(B1,B2,B3)、A(A1)，系统可信程度逐渐增高

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）

TCSEC/TDI安全级别划分（续）

–D级，最低级别，例如DOS

–C1级，非常初级的自主安全保护。实现对用户和数据的分离，进行自主存取控制(DAC)，保护或限制用户权限的传播。现有商业系统稍作改进即可满足要求。

–C2级，安全产品的最低档次，提供受控的存取保护。将C1级的DAC进一步细化，以个人身份注册负责，并实施审计和资源隔离，例如Windows 2000、Oracle7

TCSEC/TDI安全级别划分（续）

–B1级，标记安全保护。对系统的数据加以标记，并对标记的主体和客体实施强制存取控制（MAC）以及审计等安全机制。B1级别的产品才是真正意义上的安全产品，例如Trusted Oracle7

–B2级，结构化保护。建立形式化的安全策略模型并对系统内所有主体和客体实施DAC和MAC

–B3级，安全域。审计跟踪能力更强，并提供系统恢复过程。

–A1级，验证设计。提供B3级保护的同时给出系统的形式化设计说明和验证以确信各安全保护真正实现。

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来，并逐步成为新的商业标准

CC

1、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

用以规范产品和系统的安全行为

- 安全保证要求

解决如何正确有效地实施这些功能

CC文本组成

- **简介和一般模型**

  介绍CC中的有关术语、基本概念和一般模型以及与评估有关的一些框架

- **安全功能要求**

  列出一系列功能组件、子类和类。分11类(66个子类，135个组件），分别是安全审计(FAU)、通信(FCO)、密码支持(FCS)、用户数据保护(FDP)、标识和鉴别(FIA)、安全管理(FMT)、隐私(FPR)、TSF保护(FPT)、资源利用(FRU)、TOE访问(FTA)、可信路径和信道(FTP)。

- **安全保证要求**

  列出一系列保证组件、子类和类。分7类（26个子类，74个组件），分别是配置管理(ACM)、交付和运行(ADO)、开发(ADV)、指导性文档(AGD)、生命周期支持(ALC)、测试(ATE)、脆弱性评定(AVA)。

CC

- CC的具体应用通过PP和ST两种结构实现的
- PP用于表达一类产品或系统的用户需求，是CC在某一领域的具体化，PP的编制有助于提高安全保护的针对性和有效性。CC没有专门针对DBMS的解释， DBMS PP是CC在DBMS领域的具体化，相当于对DBMS的解释。
- ST是对特定的一种产品的描述，将安全要求具体化，解决安全要求的具体实现。参加CC评估的产品必须提供自己的ST

CC评估保证级划分

评估保证级（EAL）是在CC第三部分中预先定义的由保证组件组成的保证包，每个保证包描述一组特定的保证要求，对应一种评估保证级别。

从EAL1到 EAL7分7级，保证程度逐渐增高。

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**

4.1.3 数据库的不安全因素

对数据库安全性产生威胁的因素包括：

- 非授权用户对数据库的恶意存取和破坏
- 数据库中重要或敏感的数据被泄露
- 安全环境的脆弱性

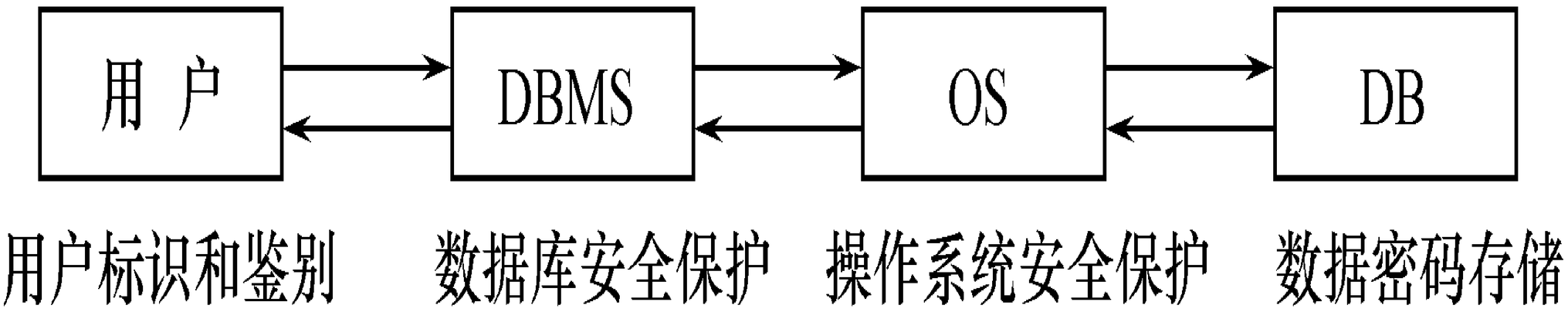
维护数据库的安全性即防止数据库被不合法使用，从而所造成的数据泄露、更改或破坏！

4.2 数据库安全性控制概述

- 非法使用数据库的情况
  - 编写合法程序绕过DBMS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从中推导出一些保密数据

数据库安全性控制概述（续）

计算机系统中，安全措施是一级级层层设置的



计算机系统的安全模型

数据库安全性控制概述（续）

- 数据库安全性控制的常用方法
  - 用户标识和鉴定
  - 存取控制
  - 视图
  - 审计
  - 密码存储

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

4.2.1 用户标识与鉴别

系统提供的最外层安全保护措施

- 用户标识
- 口令
  - 系统核对口令以鉴别用户身份

4.2.2 存取控制

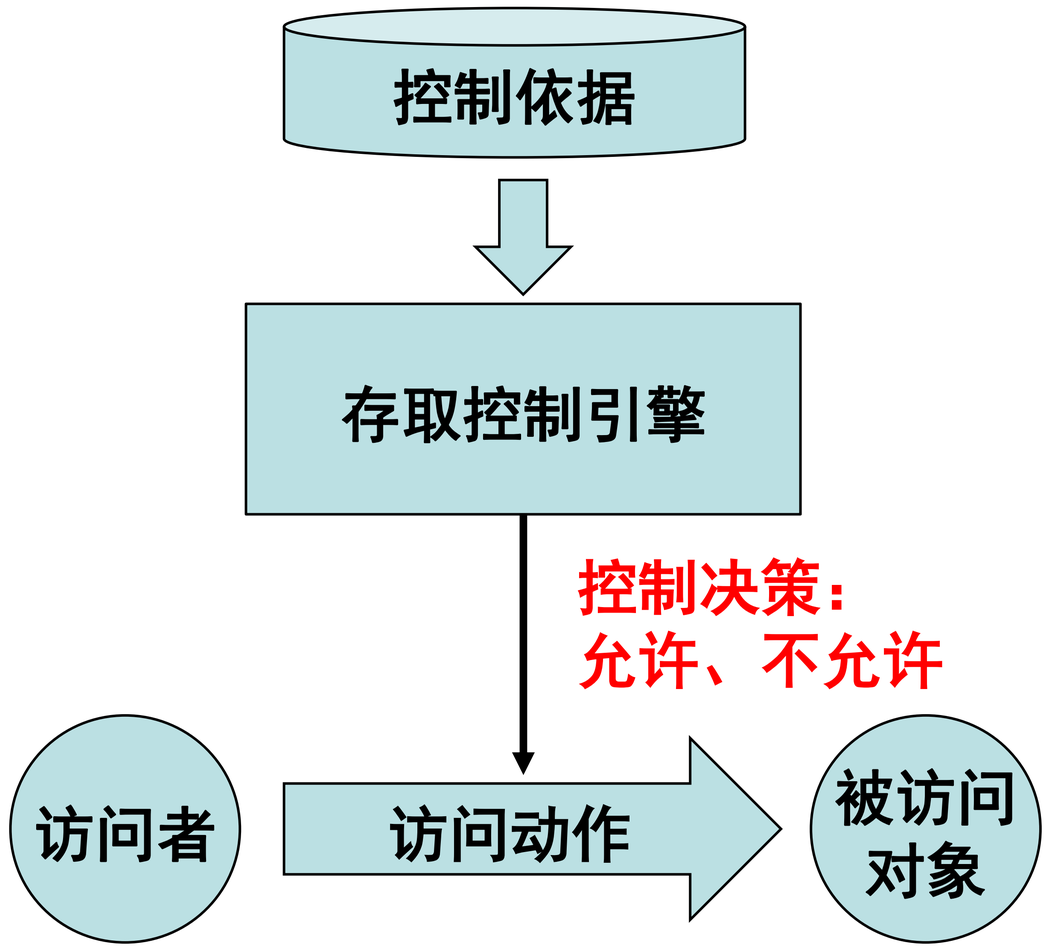
- 存取控制机制组成
  - 定义用户权限
  - 合法权限检查

- 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

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

存取控制（续）



4.2.3 自主存取控制方法

- 通过SQL的**GRANT**语句和**REVOKE**语句实现
- 用户权限由两个要素组成
  - 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**

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

4.2.4 授权与回收

一、授权GRANT

- GRANT语句的一般格式：

  GRANT <权限>[,<权限>]...

  [ON <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：

- **DBA**
- **数据库对象创建者（即属主Owner）**
- **拥有该权限的用户**

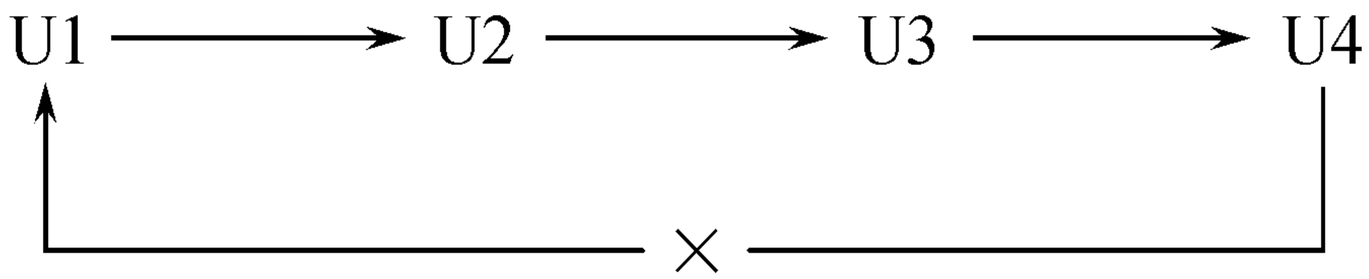
接受权限的用户

- **一个或多个具体用户**
- **PUBLIC（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3] 把对表SC的查询权限授予所有用户

```
GRANT SELECT
ON TABLE SC
TO PUBLIC;
```

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**

例题（续）

[例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;
```

传播权限

执行例5后，U5不仅拥有了对表SC的INSERT权限，还可以传播此权限：

[例6] GRANT INSERT ON TABLE SC TO U6
WITH GRANT OPTION;

同样，U6还可以将此权限授予U7：

[例7] GRANT INSERT ON TABLE SC TO U7;

但U7不能再传播此权限。

传播权限（续）

下表是执行了［例1］到［例7］的语句后，学生-课程数据库中的用户权限定义表

授权用户名	被授权用户名	数据库对象名	允许的操作类型	能否转授权
DBA	U1	关系Student	SELECT	不能
DBA	U2	关系Student	ALL	不能
DBA	U2	关系Course	ALL	不能
DBA	U3	关系Student	ALL	不能
DBA	U3	关系Course	ALL	不能
DBA	PUBLIC	关系SC	SELECT	不能
DBA	U4	关系Student	SELECT	不能
DBA	U4	属性列Student.Sno	UPDATE	不能
DBA	U5	关系SC	INSERT	能
U5	U6	关系SC	INSERT	能
U6	U7	关系SC	INSERT	不能

回收权限

二、回收REVOKE

- 授予的权限可以由DBA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>]
FROM <用户>[,<用户>]...
[CASCADE|RESTRICT];
```

REVOKE（续）

[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;
```

REVOKE（续）

[例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权限收回的时候必须级联(CASCADE)收回**
- **系统只收回直接或间接从U5处获得的权限**

REVOKE（续）

执行［例8］到［例10］的语句后，学生-课程数据库中的用户权限定义表

授权用户名	被授权用户名	数据库对象名	允许的操作类型	能否转授权
DBA	U1	关系Student	SELECT	不能
DBA	U2	关系Student	ALL	不能
DBA	U2	关系Course	ALL	不能
DBA	U3	关系Student	ALL	不能
DBA	U3	关系Course	ALL	不能
DBA	U4	关系Student	SELECT	不能

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
- 所有授予出去的权限在必要时又都可用REVOKE语句收回

授权与回收（续）

三、创建数据库模式的权限

DBA在创建用户时实现

CREATE USER语句格式：

```
CREATE USER <username>
［WITH］［DBA | RESOURCE | CONNECT］
```

授权与回收（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

4.2.5 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. 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12]　角色的权限修改

```
GRANT DELETE
ON TABLE Student
TO R1
```

[例13]　角色的权限收回

```
REVOKE SELECT
ON TABLE Student
FROM  R1;
```

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

- 主要功能是提供数据独立性
- 间接实现了支持存取谓词的用户权限定义

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权限授于王平，把该视图上的所有操作权限授于张明

1、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）

2、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;
```

4.4 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

［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;
```

- NOAUDIT语句：取消审计功能

［例16］取消对SC表的一切审计

```
NOAUDIT ALTER，UPDATE ON SC;
```

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）变换为不可直接识别的格式（密文），从而使不知道解密算法的人无法获知数据的内容。

数据加密

- 加密方法
  - 替换方法：使用密钥将明文中的每个字符转换为密文中的一个字符
  - 置换方法：将明文中的字符按不同的顺序重新排列
  - 混合方法
- DBMS中的数据加密
  - 有些数据库产品提供了数据加密例行程序，可根据用户的要求自动对存储和传输的数据进行加密处理
  - 另一些数据库产品本身未提供加密程序，但提供了接口，允许用户用其他厂商的加密程序对数据加密

4.6 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
- TCSEC和CC

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

本章作业

第六版：

P.149

6((1),(4),(5),(6),(7))、7

第五版：

P.155

7((1),(4),(5),(6),(7))、8