第四章 数据库安全性

Principles of Database Systems

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

第四章 数据库安全性

4.1 安全性概述

- 1定义

  数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

2.重要性

数据库系统中大量数据集中存放，许多用户直接共享。

系统的安全保护措施是否有效是数据库系统的主要性能指标之一。

3.计算机系统三类安全性问题

技术安全

管理安全

政策法律类

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

安全措施：用户身份鉴别、存取控制和视图等

2）数据库中重要或敏感数据被泄露

安全措施：审计、日志、入侵检测等

3）安全环境的脆弱性，包括：硬件、OS、网络等

可信计算机系统

4.1.2 安全标准简介

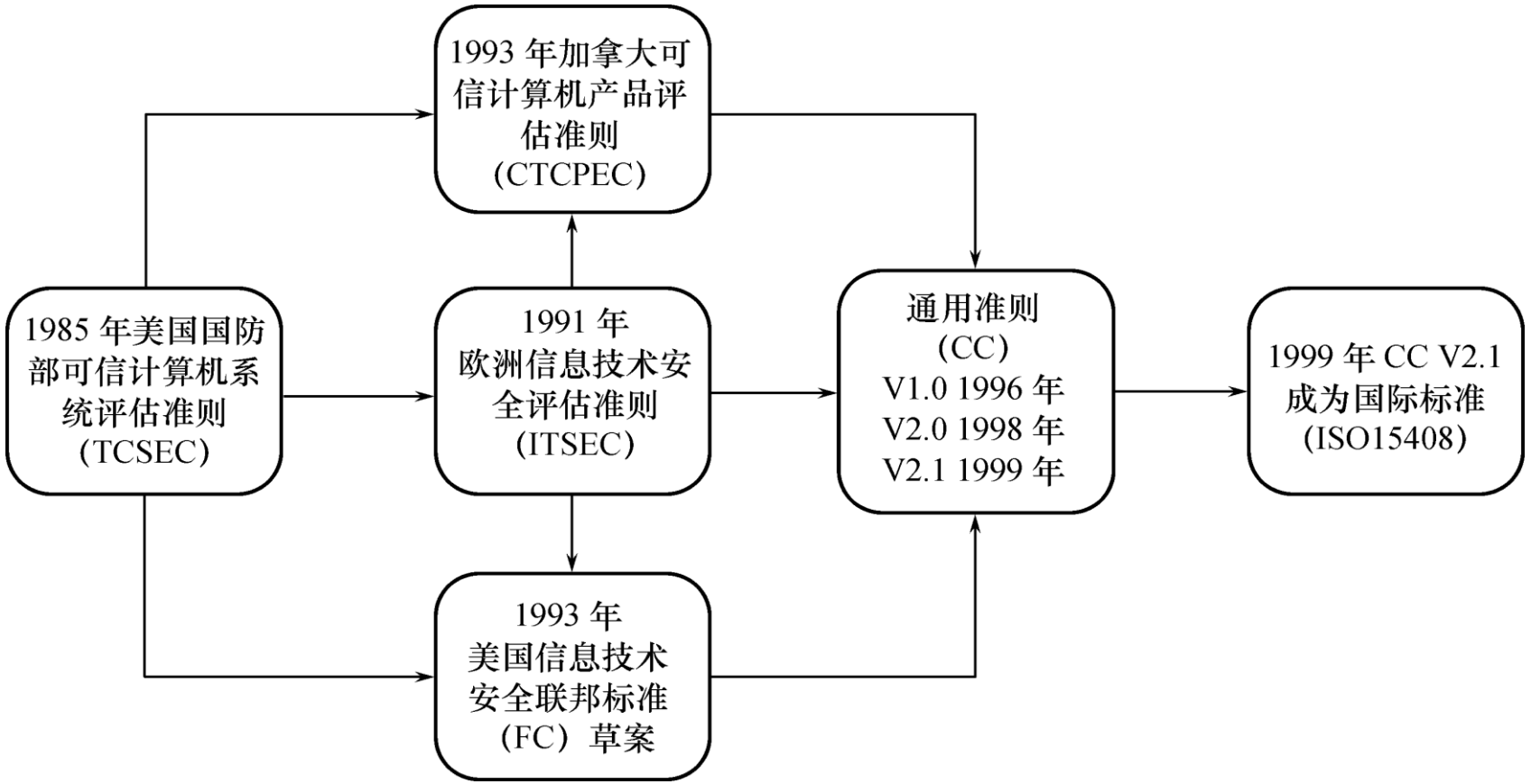
-TCSEC：20世纪60年代后期，美国国防部（DOD）开始对计算机安全评估标准进行研究，并在1985年发布了《DoD可信计算机系统评估标准》（TCSEC，即桔皮书）。后来又颁布了《可信计算机系统评估标准关于可信数据库系统的解释》（TDI，即紫皮书）。

-ITSEC：法、英、荷、德欧洲四国90年代初联合发布信息技术安全评估标准（ITSEC，欧洲白皮书），它提出了信息安全的机密性、完整性、可用性的安全属性。

-CC：1996年，由六个国家（美、加、英、法、德、荷）联合提出了信息技术安全评价的通用标准（CC）。该标准提出了目前国际上公认的表述信息技术安全性的结构。

-ISO 15408：1999年，CC 2.1版被ISO采纳为国际标准ISO 15408。

4.1.2 安全标准简介



信息安全标准的发展历史

4.1.2 安全标准简介

- **TCSEC/TDI安全级别划分**
  - **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。**

安全级别	定　　义
A1	验证设计（Verified Design）
B3	安全域（Security Domains），安全域TCB
B2	结构化保护（Structural Protection），安全策略模型
B1	标记安全保护（Labeled Security Protection），MAC
C2	受控的存取保护（Controlled Access Protection），个人注册
C1	自主安全保护（Discretionary Security Protection），DAC
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

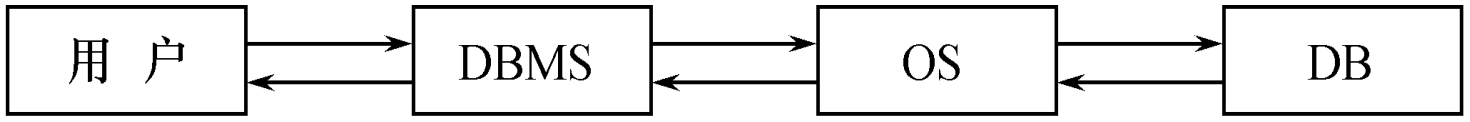
4.1.2 安全标准简介

- **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

4.2 数据库安全控制技术

- 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
- 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

- 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

4.2.1 用户身份鉴别

- **用户标识与鉴别（Identification & Authentication）：**
  - **系统提供的最外层安全保护措施；**
  - **系统提供一定的方式让用户标识自己的名字和身份，系统进行核实，通过鉴定后才提供系统使用权。**

常用鉴别方法：

- **口令：静态（易被窃取）、动态（一次一密）**
- **生物特征识别：指纹、声音、照片等**
- **智能卡**
- **回答问题**

4.2.2 存取控制

1. 什么是存取控制?

对于获得上机权的用户还要根据系统预先定义好的外模式（视图）或用户权限进行存取控制，保证用户只能存取他有权存取的数据。

2. 方法

- 定义用户权限
- 合法权限检查

3.常用存取控制方法

- 自主存取控制（Discretionary Access Control，DAC），C2级，灵活
- 强制存取控制（Mandatory Access Control，MAC），B1级，严格

4.2.2 存取控制

- 自主存取控制（Discretionary Access Control，简称DAC），C2级。
- 是由用户或DBA定义存取权限的一种控制策略。通过SQL的GRANT语句和REVOKE语句实现。
- 访问权限由两个要素组成：数据对象和操作。类型系统通过控制数据对象的访问权限防止非授权访问。

对象类型	对象	操 作 类 型
数据库	模式	CREATE SCHEMA
	基本表	CREATE TABLE，ALTER TABLE
模式	视图	CREATE VIEW
	索引	CREATE INDEX
数据	基本表和视图	SELECT，INSERT，UPDATE，DELETE，REFERENCES，ALL PRIVILEGES
数据	属性列	SELECT，INSERT，UPDATE，REFERENCES
		ALL PRIVILEGES

4.2.4 授权与回收

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

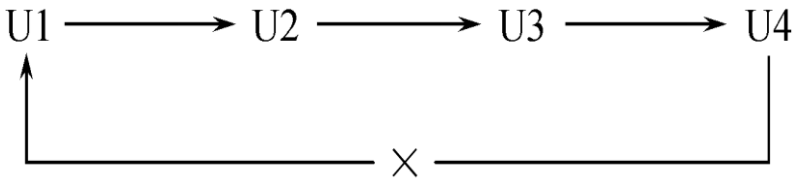
- 一个或多个具体用户
- PUBLIC（全体用户）

1.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**
- **WITH GRANT OPTION子句:**
  - **若指定：可以再授予；否则，没有指定：不能传播**
- **不允许循环授权**



4.2.4 授权与回收

例1 把查询Student表权限授给用户U1。

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

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 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。
```

4.2.4 授权与回收

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;
```

// 执行此SQL语句后，U5不仅拥有了对表SC的INSERT权限，还可以传播此权限，即由U5用户发上述GRANT命令给其他用户。

如：U5可以将此权限授予U6：

```
GRANT INSERT ON TABLE SC TO U6 WITH GRANT OPTION;
```

同样，U6还可以将此权限授予U7：

```
GRANT INSERT ON TABLE SC TO U7;
```

//因为U6未给U7传播的权限，因此U7不能再传播此权限。

4.2.4 授权与回收

2. 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];
```

[例] 把用户U4修改学生学号的权限收回。

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

4.2.4 授权与回收

[例] 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

DAC：主体对它所属的对象和运行的程序拥有全部的俄控制权。提供一种“need-to-know” 的访问授权方法，默认拒绝任何人访问。访问必须被显式地赋予访问者。

缺陷——用户可以自由地决定将数据地存取权限授予任何人、决定是否将“授权”的权限授予别人，而系统对此无法控制。

4.2.4 授权与回收

3.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] ;

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。

- 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>] …
```

4.2.5 数据库角色

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]
```

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>] …
```

4.2.5 数据库角色

[例] 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;
```

4.2.5 数据库角色

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;
```

5. 角色的权限修改

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

练习

- 下列对于基于角色的访问控制模型的说法错误的是?

A. 它将若干特定的用户集合与权限联系在一起

B. 角色一般可以按照部门、岗位、工种等与实际业务紧密相关的类别来划分

C. 因为角色的变动往往远远低于个体的变动，所以基于角色的访问控制维护起来比较便利

D. 对于数据库系统的适应性不强，是其在实际使用中的主要弱点

自主存取控制缺点

- 可能存在数据的“无意泄露”。
- DAC容易受到特洛伊木马攻击。
- 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

- 问题：DAC可能存在的数据“无意泄露”
- 解决：对系统控制下的所有主客体赋予安全性标记，系统根据标记安全策略，实施强制存取控制。
- 强制存取控制（MAC)：
  - 保证更高程度的安全性，B1级
  - 管理员管理访问控制、制定策略，用户能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

4.2.6强制存取控制

1) 实体类别

主体——系统中的活动实体，包括用户和用户的进程。

客体——系统中的被动实体，受主体操纵的基表、索引、视图等。

2) 敏感度标记（label）：

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

主体敏感度级别——许可证级别（Clearance Level）

客体敏感度标记——密级（Classification Level）

4.2.6强制存取控制

3. 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**等于或小于**客体的密级时，该主体才能**写**相应的客体

- 优点：

❖禁止拥有高许可证级别的主体更新低密级的数据对象，从而保证了敏感数据的可靠性；

❖禁止低许可证级别的主体浏览高密级的数据，避免了敏感数据的泄漏；

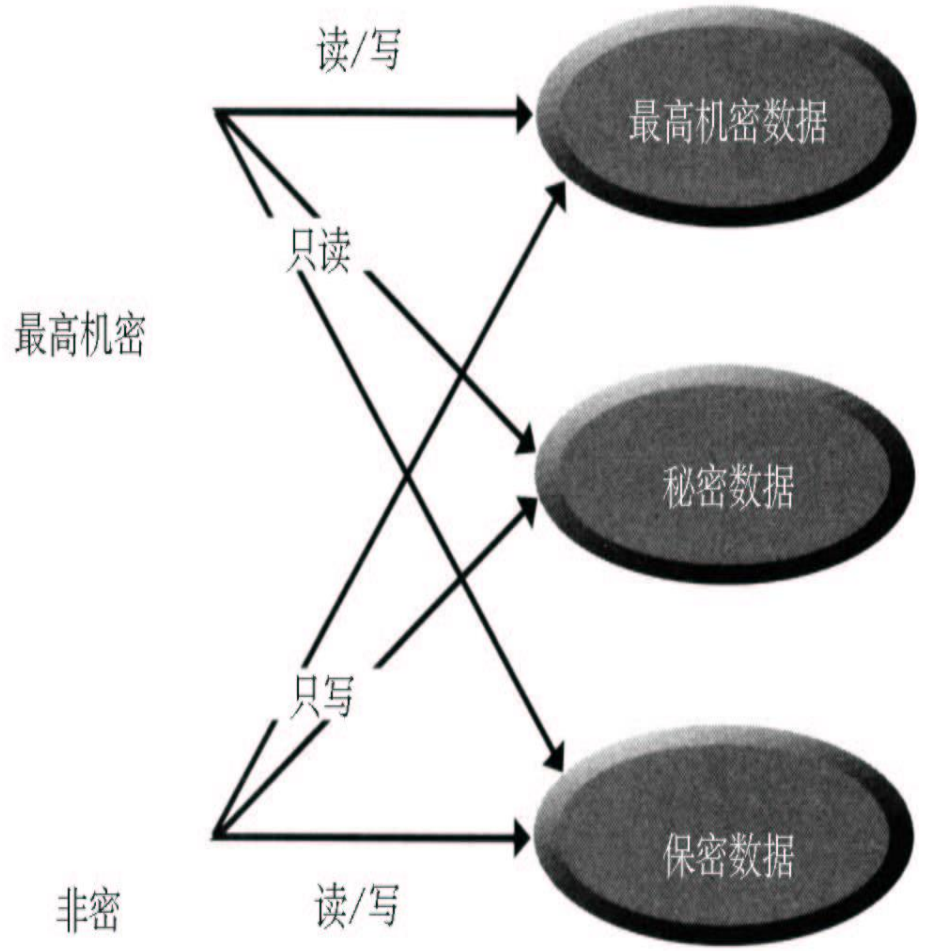
❖MAC对数据本身进行密级标记，无论数据如何复制，**标记与数据是不可分割的整体**。只有符合密级标记要求的用户才可以操作相应数据，提高了安全性级别。

4.2.6强制存取控制

最高密级主体

向下读
向上写

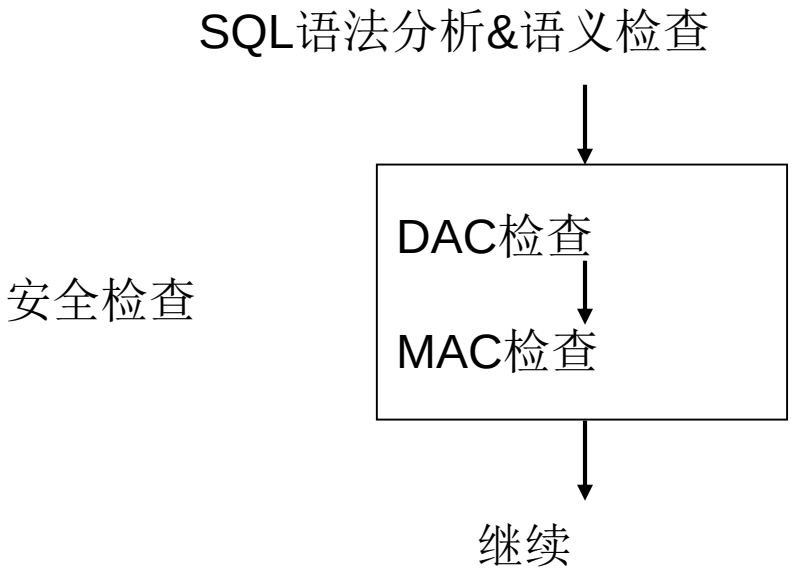
最低密级主体



4.2.6强制存取控制

4. 兼容关系

安全级别之间具有偏序向下兼容关系，较高安全性级别的安全保护措施要包含较低级别的所有保护措施，同时应提供更多更完善的保护能力。因此实现MAC必须先实现DAC，DAC和MAC共同构成DBMS的安全机制。



4.3 视图机制

- 视图就象架设在用户与基表之间的一道桥梁，用户可以通过视图访问基表，也可以直接访问基表，但对安全级别要求较高的数据一般通过视图进行访问，从而避免直接访问基表中其他数据。
- 对于终端用户，虽然数据库中的数据是面向全局的，但通过视图隔离，他只能看到专门为他定义的视图中与自己相关的数据。其他与他无关的数据被子模式即视图隔离或屏蔽了。
- 例：

```
CREATE  VIEW  VIEW_1  AS
   SELECT COLUMN_1  FROM TABLE_1;
GRANT ALL ON VIEW_1 TO USER_2;
```

//此例中，用户USER_2通过视图VIEW_1可以访问表TABLE_1中的列COLUMN_1而无法访问COLUMN_2，这就是VIEW_1的屏蔽作用。

练习

- 从安全的角度来看，数据库视图（view）的主要用途是：
  - A. 确保相关完整性
  - B. 方便访问数据
  - C. 限制用户对数据的访问
  - D. 提供审计跟踪

4.4 审计

- 审计：对用户使用系统资源情况的登记和审查。
  - 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。
- 审计分类：
  - 用户级审计
    - 用户对自己拥有的数据库对象上发生的操作进行审计；
    - 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。

4.4 审计

- AUDIT语句：设置审计功能
- NOAUDIT语句：取消审计功能

[例] 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;
```

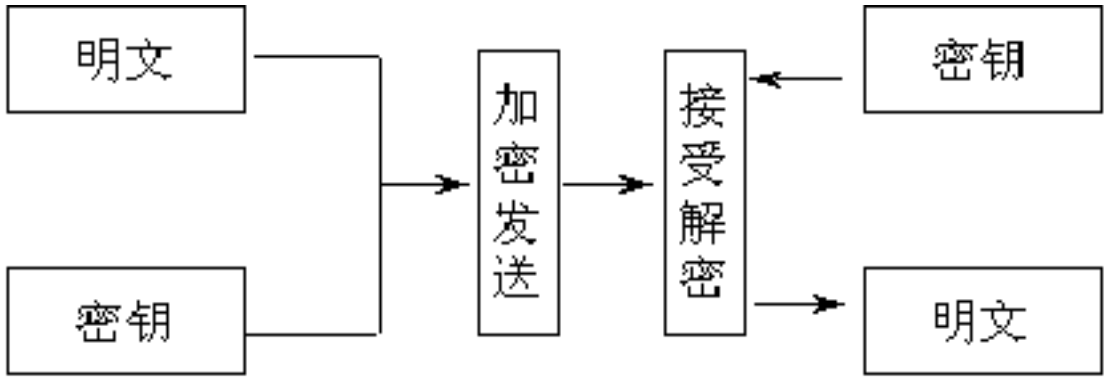
[例] 取消对SC表的一切审计

```
NOAUDIT ALTER, UPDATE ON SC;
```

4.5 数据加密

1. 思想

数据库中的数据以密码形式存放，使用时由用户设计的解码程序将其转换成用户可读的数据。这样，数据库中的数据即使被窃取，也只能是一些无法辨认的代码。



2. 加密方法：信息编码，信息换位，信息替换等

3. 数据库加密：存储加密、传输加密

4.6 其他安全性保护

- 统计数据库
  - 允许用户查询聚集类型的信息（如合计、平均值等）
  - 不允许查询单个记录信息
- 数据库中特殊的安全性问题
  - 隐蔽的信息通道
  - 能从合法的查询中推导出不合法的信息
- 措施：

  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
- **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

练习

- 以下关于数据库安全的说法**错误**的是?

A. 数据库系统的安全性很大程度上依赖于 DBMS 的安全机制

B. 许多数据库系统在操作系统下以文件形式进行管理，因此利用操作系统漏洞可以窃取数据库文件

C. 为了防止数据库中的信息被盗取，在操作系统层次对文件进行加密是唯一从根本上解决问题的手段

D. 数据库的安全需要在网络系统、操作系统和数据库管理系统三个方面进行保护

练习

- 下面关于访问控制模型的说法不正确的是：

  A. DAC 模型中主体对它所属的对象和运行的程序拥有全部的控制权。

  B. DAC 实现提供了一个基于“need-to-know”的访问授权的方法，默认拒绝任何人的访问。访问许可必须被显式地赋予访问者。

  C．在 MAC 这种模型里，管理员管理访问控制。管理员制定策略，策略定义了哪个主体能访问哪个对象。但用户可以改变它。

  D．RBAC 模型中管理员定义一系列角色（roles）并把它们赋予主体。系统进程和普通用户可能有不同的角色。设置对象为某个类型，主体具有相应的角色就可以访问它。

拓展：RBAC

Level	Name	RBAC Functional Capabilities
1	**Flat RBAC** (see Figure 1)	• users acquire permissions through roles • must support many-to-many user-role assignment • must support many-to-many permission-role assignment • must support user-role assignment review • users can use permissions of multiple roles simultaneously
2	**Hierarchical RBAC** (see Figure 2)	Flat RBAC + • must support role hierarchy (partial order) • **level 2a** requires support for arbitrary hierarchies • **level 2b** denotes support for limited hierarchies
3	**Constrained RBAC** (see Figures 6 and 7)	Hierarchical RBAC + • must enforce separation of duties (SOD) • **level 3a** requires support for arbitrary hierarchies • **level 3b** denotes support for limited hierarchies
4	**Symmetric RBAC** (see Figures 9 and 10)	Constrained RBAC + • must support permission-role review with performance effectively comparable to user-role review • **level 4a** requires support for arbitrary hierarchies • **level 4b** denotes support for limited hierarchies

RBAC (Role-Based Access Control) ≠ DAC or MAC

在RBAC中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。

小结

- 介绍了数据库安全性定义和安全标准TCSEC和CC。
- 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
- 数据库安全控制技术：
  - 用户标识和鉴别
  - 自主存取控制DAC：GRANT和REVOKE语句，　　C2级
  - 数据库角色CREATE ROLE
  - 强制存取控制MAC　　B1级
  - 视图机制
  - 审计
  - 加密
- 本章作业：P155　　6，7　(请在超星平台提交，deadline 5.7)