南京大学 NANJING UNIVERSITY

奋进行动 团结奋斗 争先进位 坐言起行 应势而动

数 据 管 理 基 础

第4章 数据库安全性

智能软件与工程学院

数据库安全性 1

- 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
    - 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。

- 一般来说，对数据库的破坏来自以下四个方面：非法用户、非法数据、各种故障或误操作、多用户的并发访问。针对以上四种可能情况，数据库管理系统分别采取以下措施对数据库实施保护：安全性保护（权限管理）、完整性保护、故障恢复、并发控制。

数据库安全性 2

- 著名咨询机构Verizon发布的《2023年数据泄露报告》(DBIR)，对2022年一年发生的16312起安全事件和5199起数据泄漏事件进行分析，其中报告显示**人是数据泄露事件的关键因素！**

- **数据泄露事件**
  - 丰田200余万车主车辆数据遭泄露.....
  - 45亿国内快递信息遭泄露，数据来自快递平台及购物网站，包含用户真实姓名、电话与住址等。
  - 某科技公司在为政府部门开发运维信息管理系统的过程中，将建设单位采集的敏感数据擅自上传至租用的公有云服务器上。
  - 某信息科技公司网络数据泄露隐患：在没有依法建立数据安全管理制度和操作规程等数据保护措施的前提下，对存储的公民敏感信息数据未采取去标识化和加密保护；服务器存在未授权访问的漏洞。
  - 某著名消费信贷公司XXX金服的大规模用户个人信息泄露事件

数据库安全性 3

❑黑客/非法用户访问问题

- 2013年，国外某医疗保健改革计划的数据泄露事件，几乎所有用户数据被非法获取，许多人因此遭受身份盗用和财产损失。
- 2015年，国外某国家保险公司，黑客通过攻击员工工作电脑，成功获取了大量敏感数据，包括客户信息、保单、合同、电子邮件等。
- 2018年Oracle数据库、2020年MongoDB数据库、2022年SQL Server数据库和Elasticsearch数据库，黑客入侵勒索赎金。
- 2019年，欧洲某国政府电子交换系统遭受黑客攻击，导致许多政府机构信息被泄露，给国家安全和政府信任度带来了极大挑战。
- 2023年，黑客入侵了俄罗斯CyrenTrevel防御系统的数据库，得到了16年内6.64亿条飞行的信息，包括乘客姓名、电话、证件号、所乘航班、线路等。
- 黑客入侵宏碁公司内部服务器，160GB敏感数据遭泄露。
- 利用公司内部员工合法账号，从公司招聘系统内下载包含个人信息的简历非法出售给他人获利。
- 某校在校研究生，利用专业技术盗取全校学生个人信息，用于搭建学生颜值打分网站。

第4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
- 4.1.3 中国计算机安全标准制订历史

4.1.1 数据库的不安全因素

- **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

- **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

4.1.2 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）
- 1993年，CTCPEC、FC、TCSEC和ITSEC联合行动，解决原标准中概念和技术上的差异，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

图4.1 信息安全标准的发展历史

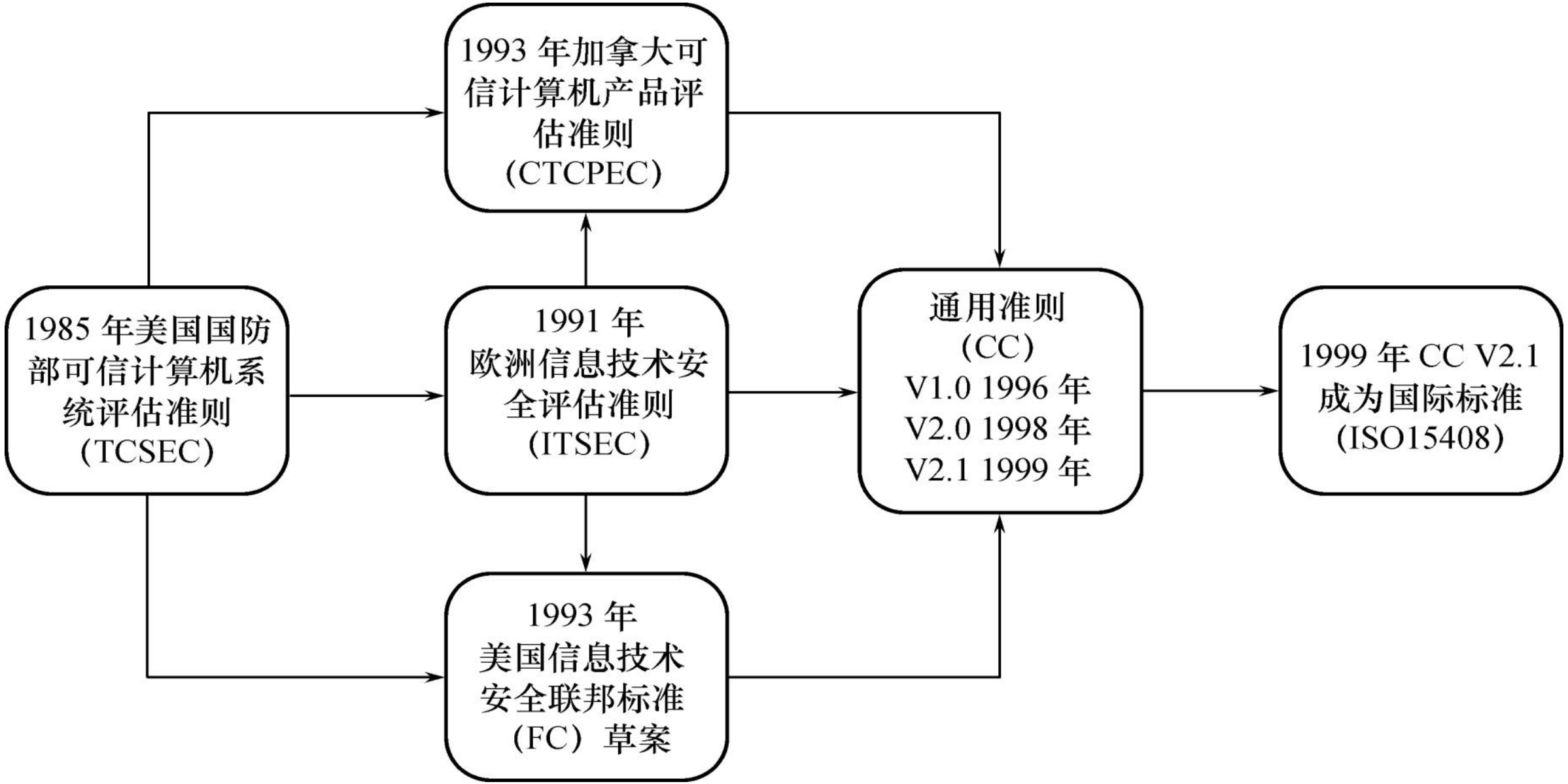


表4.1 TCSEC/TDI安全级别划分 1

❑TCSEC/TDI安全级别划分

- ➢按系统可靠或可信程度逐渐增高
- ➢各安全级别之间具有一种偏序向下兼容的关系

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 2

- D级
  - 将一切不符合更高标准的系统均归于D组
- C1级
  - 非常初级的自主安全保护
  - 能够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。
  - 现有的商业系统稍作改进即可满足
- 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
  - 达到C2级的产品在其名称中往往不突出“安全”（Security）这一特色
- B1级
  - 标记安全保护。“安全”（Security）或“可信的”（Trusted）产品。
  - 对系统的数据加以标记，对标记的主体和客体实施强制存取控制（MAC）、审计等安全机制
- B2级
  - 结构化保护
  - 建立形式化的安全策略模型并对系统内的所有主体和客体实施DAC和MAC
- B3级
  - 安全域
  - 该级的TCB必须满足访问监控器的要求，审计跟踪能力更强，并提供系统恢复过程
- A1级
  - 验证设计，即提供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 3

- 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文本组成（三个组成部分相互依存，缺一不可）
  - 简介和一般模型
    - 介绍有关术语、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
    - 列出了一系列类、子类和组件
  - 安全保证要求
    - 列出了一系列保证类、子类和组件
    - 提出了评估保证级（Evaluation Assurance Level，EAL），从EAL1至EAL7共分为七级

表4.2 CC评估保证级（EAL）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

❑安全标准的制订历史（中国）

➢GB 17859-1999：计算机信息系统安全保护等级划分准则

- 参照美国的TCSEC标准来制订等级划分标准

GB 17859-1999分级	TCSEC 分级
-	D
第 1 级：自主安全保护级	C1
第 2 级：系统审计保护级	C2
第 3 级：安全标记保护级	B1
第 4 级：　结构化保护级	B2
第 5 级：访问验证保护级	B3
-	A1

4.1.3 中国计算机安全标准制订历史

❑安全标准的制订历史（中国）

- 中国: 2002年(公安部)
  - 计算机信息系统安全等级保护系列标准，内容涉及：技术要求、管理要求、工程实施要求、实施管理办法、评测标准。
  - 技术要求有:
    - 网络技术要求(GA/T 387 － 2002)
    - 操作系统技术要求(GA/T 388 － 2002)
    - 数据库管理系统技术要求(GA/T 389 － 2002)
    - 通用技术要求(GA/T 390 － 2002)
    - 管理技术要求(GA/T 391 － 2002)

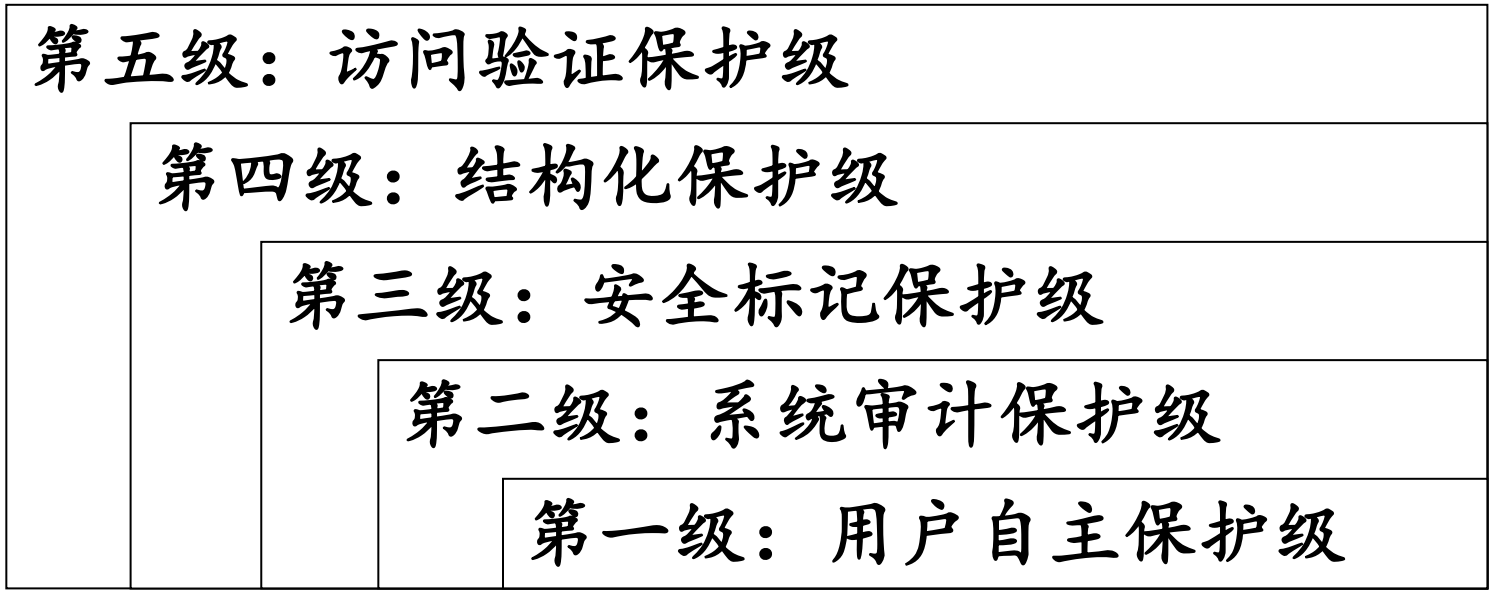
4.1.3 中国计算机安全标准制订历史

- 安全标准的制订历史（中国）

1. GB/T18336-2001：信息技术安全性评估准则
   - 参照CC标准
2. GJB2646-96：军用计算机安全评估标准
   - 参照美国的TCSEC标准
3. GJB5023-2001：军用数据库安全评估准则
4. GB/T20009-2005：信息安全技术--数据库管理系统安全评估准则
5. GB/T20273-2006：信息安全技术--数据库管理系统安全技术要求

4.1.3 中国计算机安全标准制订历史

- 计算机信息系统安全保护等级划分准则（GB 17859-1999）
  - 第一级：用户自主保护级
  - 第二级：系统审计保护级
  - 第三级：安全标记保护级
  - 第四级：结构化保护级
  - 第五级：访问验证保护级



数据库安全标准 – 安全功能要求

	用户自主保护级	系统审计保护级	安全标记保护级	结构化保护级	访问验证保护级
身份鉴别	+	+	+++	+++	++++
自主访问控制	+	++	++	+++	++++
强制访问控制			+	++	+++
标记			+	++	++
客体重用		+	+	++	++
审计		+	++	+++	++++
数据完整性	+	+	++	++	+++
隐蔽信道分析				+	++
可信路径				+	++
可信恢复					+
推理控制				+	+

第4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

- 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从中推导出一些保密数据

图4.2 计算机系统的安全模型

- 计算机系统中，安全措施是一级一级层层设置
  - 系统根据用户标识鉴定用户身份，合法用户才准许进入计算机系统
  - 数据库管理系统还要进行存取控制，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

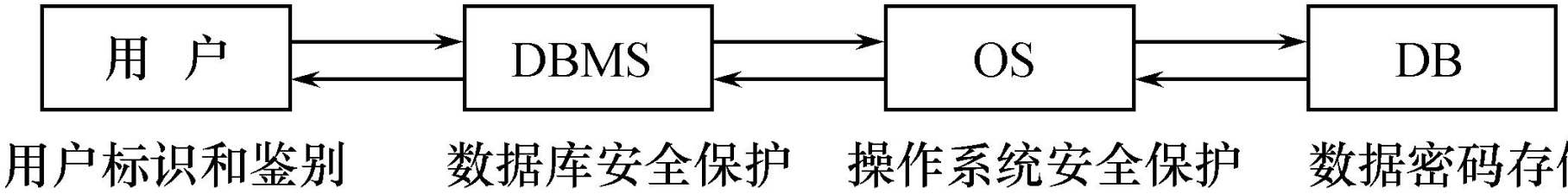
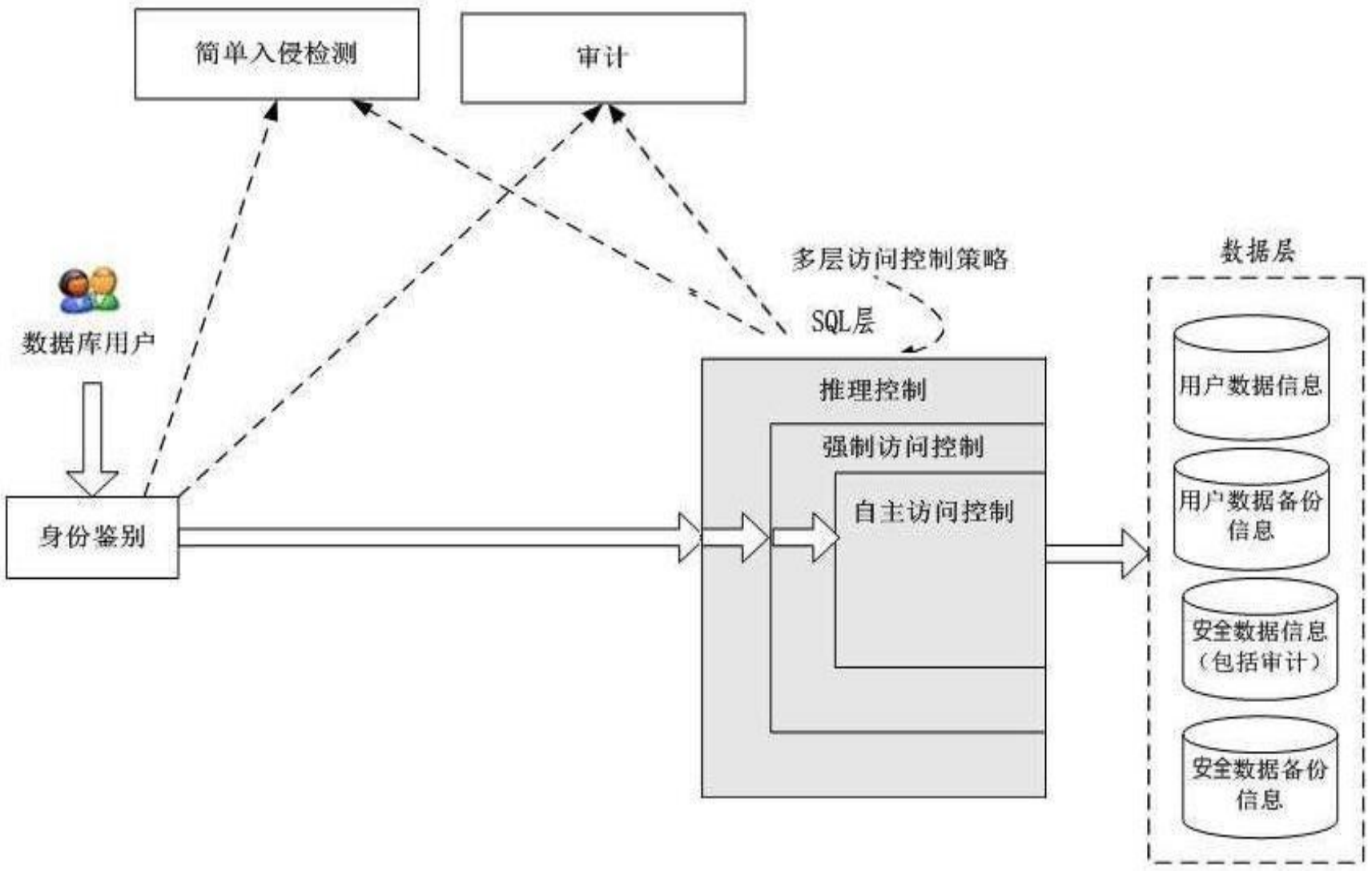


图4.3 数据库管理系统安全性控制模型



4.2 数据库安全性控制

❑存取控制的流程

- ➢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

4.2.1 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）

❑用户身份鉴别的方法

1. 静态口令鉴别：静态口令一般由用户自己设定，这些口令是静态不变的
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

4.2.2 存取控制

- 存取控制机制组成
  - 定义用户权限，并将用户权限登记到数据字典中
    - 用户对某一数据对象的操作权力称为权限
    - DBMS提供适当的语言来定义用户权限，存放在数据字典中，称做安全规则或授权规则
  - 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

- 定义用户权限和合法权限检查机制一起组成了数据库管理系统的存取控制子系统

常用存取控制方法

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

- C2级
- 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，简称 MAC）

- B1级
- 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

4.2.3 自主存取控制方法 1

- 通过 SQL 的 GRANT语句 和 REVOKE语句 实现

- 用户权限的组成要素
  - 数据库对象
  - 操作类型

- 定义用户的存取权限
  - 定义一个用户可以在哪些数据库对象上进行哪些类型的操作
  - 定义用户的存取权限称为‘授权管理’，包括向用户‘授予’或‘收回’对数据的操作权限。

4.2.3 自主存取控制方法 2

表4.3 关系数据库系统中的存取控制对象与权限

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

4.2.3 自主存取控制方法 3

- 用户存取权限的定义

1) 一个客体的‘属主’自动拥有该客体上的所有操作权限；
   - 客体的创建者即客体的属主 (Owner)
   - 系统应该控制客体所有权的转移

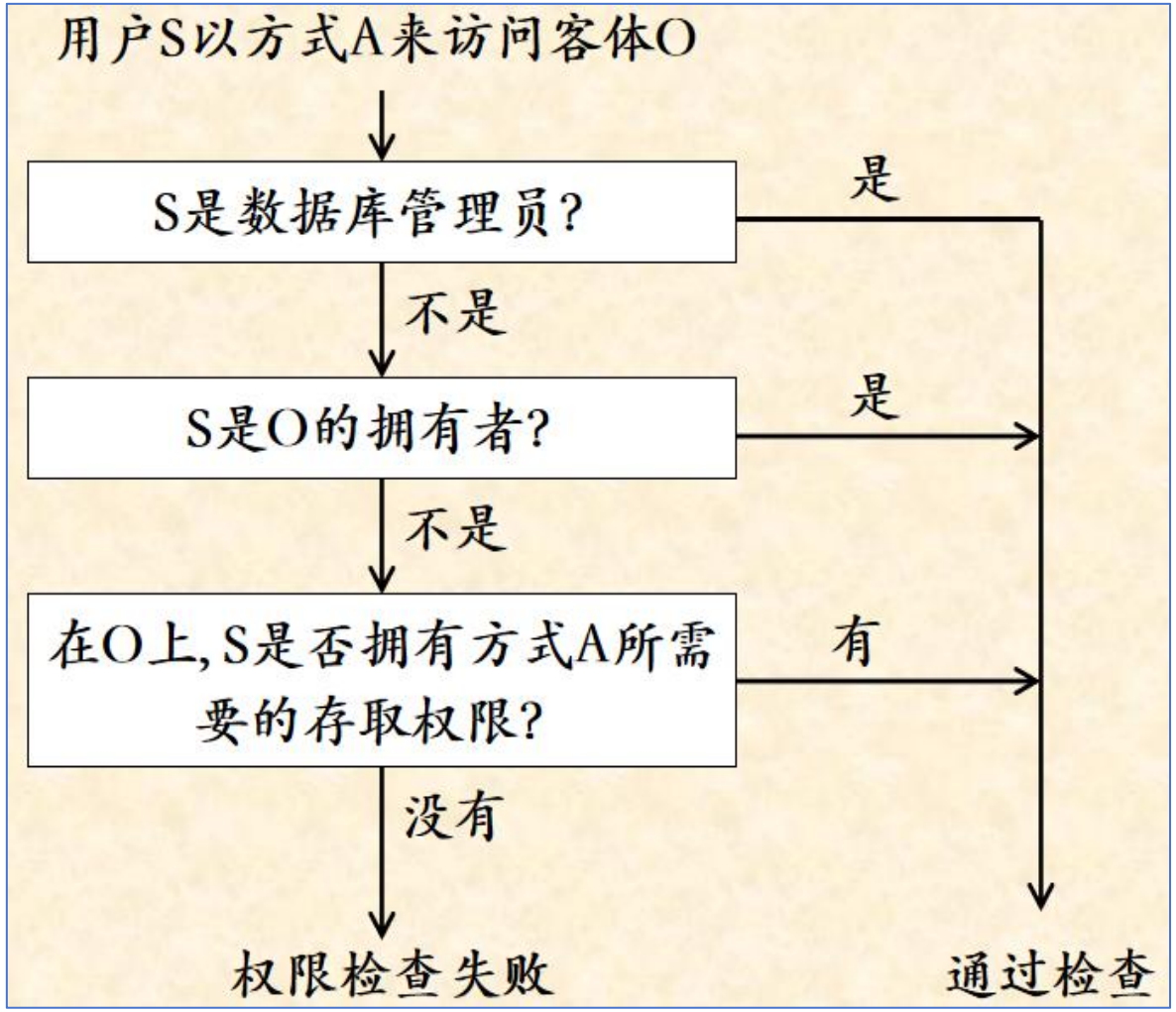
2) 拥有权限的用户可以自主地将他所拥有的权限传授给其它任意在数据库系统中注册的用户；

3) 用户只能通过上述两种方式（或之一）来获得在一个客体上的存取权限。

4.2.3 自主存取控制方法 4

- 访问控制
  1. 在用户登录时进行用户的身份鉴别
  2. 在用户访问数据库时执行访问控制检查

- 对于用户提交的每一条数据访问命令，系统将按照如右图所示的流程进行权限检查：



4.2.4 授权：授予与收回

- SQL中的授权机制
  - 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

GRANT 1

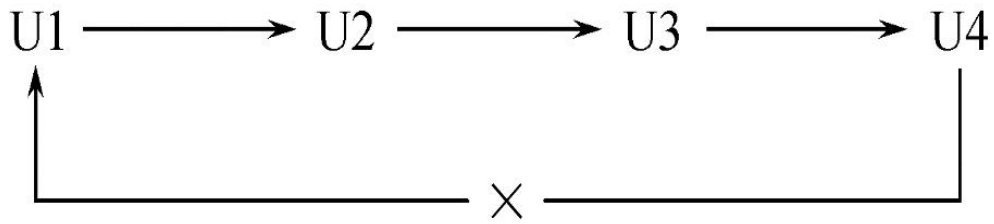
- GRANT语句的一般格式：

  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
- 授权用户（发出或执行GRANT语句的用户）
  - 数据库管理员
  - 数据库对象的创建者（即数据库对象的属主Owner）
  - 拥有该权限的用户
- 被授权用户（接受权限的用户）
  - 一个或多个具体用户
  - 一个或多个用户组（角色ROLE）
  - PUBLIC（即全体用户）

❑WITH GRANT OPTION子句：

- ➢在GRANT命令中，如果使用了WITH GRANT OPTION选项，那么被授权用户可以将接受到的权限再授予其他用户，否则，被授权用户不能把接受到的权限再转授给其他用户。
- ➢不允许循环授权



❑[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;
```

GRANT 3

- [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，必须明确指出相应属性列名

GRANT 4

- [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;
```

- 执行例4.5后，U5不仅拥有了对表SC的INSERT权限，还可以传播此权限：

- [例4.6]用户U5可以执行下述命令，将之前获得的表SC的INSERT权限及传播权限 授予用户U6。

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

- [例4.7]同样，U6还可以将此权限授予U7，但U7不能再传播此权限。

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

传播权限

- 执行了例4.1~例4.7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

REVOKE 1

- **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 2

- [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;
```

- [例4.9]收回所有用户对表SC的查询权限。

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

- [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权限收回的时候使用CASCADE，则同时收回U6或U7的INSERT权限，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。

❑执行例4.8~4.10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

创建数据库模式的权限 1

- 由数据库管理员在创建用户时，进行有关创建数据库模式的授权

- 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
  - 只有系统的超级用户才有权创建一个新的数据库用户

创建数据库模式的权限 2

- 新创建的数据库用户，可以被授予以下三种权限之一
  - **CONNECT**
    - 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，只能登录数据库。
  - **RESOURCE**
    - 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。
  - **DBA**
    - 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。

表4.6 权限与可执行操作的对照表

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

4.2.5 数据库角色 1

- **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

- 角色的创建

```
CREATE ROLE <角色名>;
```

- 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 2

- 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
  - 授予者是角色的创建者或拥有在这个角色上的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或用户

- WITH GRANT OPTION 与 WITH ADMIN OPTION 的区别
  - ADMIN OPTION是关于系统权限的授予权（系统权限：特定的数据库访问操作，通常是数据定义或数据控制命令，如数据库对象的创建等）
  - GRANT OPTION是关于对象权限的授予权（对象权限：指数据库对象的访问权限，如表中的数据访问、存储过程的调用等）
- 一个角色的权限：被直接授予这个角色的全部权限，加上通过其他角色授予这个角色的全部权限

4.2.5 数据库角色 3

- 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，或者是
  - 拥有在这个（些）角色上的 ADMIN OPTION

- 如果使用CASCADE执行权限的回收操作时，系统权限不会被级联回收，而对象权限则需要级联回收。

4.2.5 数据库角色 4

- [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：
  - 首先创建一个角色 R1

```
CREATE ROLE R1;
```

  - 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;
```

  - 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;
```

  - 可以一次性通过R1来回收王平的这3个权限

```
REVOKE R1 FROM 王平;
```

4.2.5 数据库角色 5

- [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

- [例4.13] 使R1减少了SELECT权限

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

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

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

- ➢B1级，保证更高程度的安全性
- ➢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
- ➢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，如：
  - 军事部门
  - 政府部门

强制存取控制方法 - 实体

- 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 - 敏感度标记

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

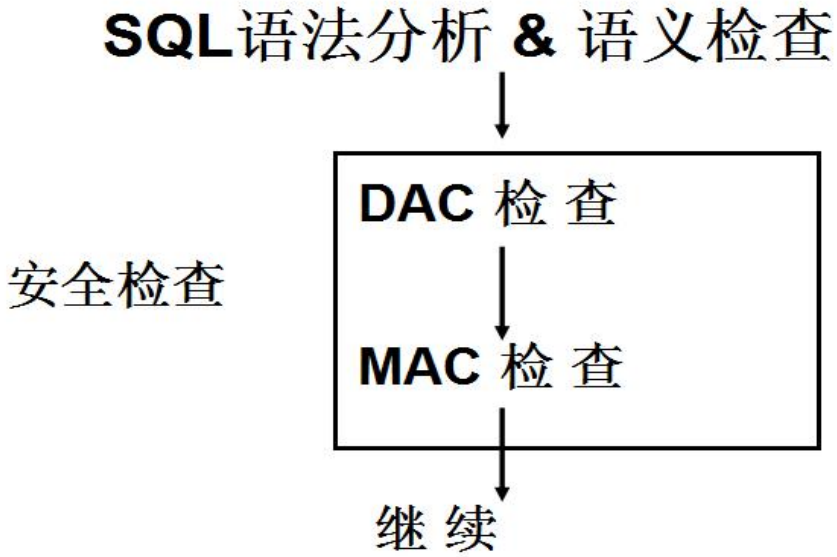
强制存取控制方法 - 强制存取控制规则

- 强制存取控制规则
  - ①（下读）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（上写）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

- 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。

DAC + MAC

- 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，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4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 1

- 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

- [例4.14]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。

- 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';
```

- 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

- 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

审计事件

- **服务器事件**
  - 审计数据库服务器发生的事件，包括数据库服务器的启停、配置文件的重新加载等
- **系统权限**
  - 对系统拥有的结构或模式对象进行操作的审计
  - 执行相关操作需要的权限是通过系统权限获得的
- **语句事件**
  - 对SQL语句的审计，如对DDL、DML、DQL及DCL语句的审计
- **模式对象事件**
  - 对特定模式对象上进行的SELECT或DML操作的审计
  - 模式对象主要包括表、视图、存储过程/函数，但不包括表上的索引、约束、触发器等。

审计功能

- 基本功能：提供多种审计查阅方式
- 多套审计规则：一般在数据库初始化时设定
- 提供审计分析和报表功能
- 审计日志管理功能
  - 防止审计员误删审计记录，审计日志必须先转储后删除；
  - 对转储的审计记录文件提供完整性和保密性保护；
  - 只允许审计员查阅和转储审计记录，不允许任何用户新增和修改审计记录等。
- 提供查询审计设置及审计记录信息的专门视图

审计级别

- **用户级审计**
  - 任何用户可设置的审计
  - 主要是用户针对自己创建的数据库表和视图进行审计
- **系统级审计**
  - 只能由数据库管理员设置
  - 监测成功或失败的登录要求、监测授权和收回操作以及其他数据库级权限下的操作

审计语句

- 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表上的ALTER和UPDATE操作的审计

```
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（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
- ➢传输加密

存储加密

❑透明存储加密

- ➢内核级加密保护方式，对用户完全透明
- ➢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
- ➢数据库的应用程序不需要做任何修改，只需在创建表语句中说明需加密的字段即可
- ➢内核级加密方法：性能较好，安全完备性较高

❑非透明存储加密

- ➢通过多个加密函数实现

传输加密

❑链路加密

- ➢在链路层进行加密
- ➢传输信息由报头和报文两部分组成
- ➢报文和报头均加密

❑端到端加密

- ➢在发送端加密，接收端解密
- ➢只加密报文不加密报头
- ➢所需密码设备数量相对较少，容易被非法监听者发现并从中获取敏感信息

4.6 其他安全性保护

❑推理控制

- ➢避免用户利用能够访问的数据推知更高密级的数据

❑隐蔽信道

- ➢间接数据传递

❑数据隐私保护

- ➢描述个人控制其不愿他人知道或他人不便知道的个人数据的能力
- ➢范围很广：数据收集、数据存储、数据处理和数据发布等各个阶段

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
  - 用户身份鉴别
  - 存取控制技术：自主存取控制 & 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