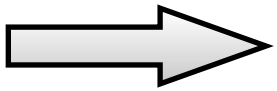
数 据 库 系 统 概 论
An Introduction to Database Systems

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



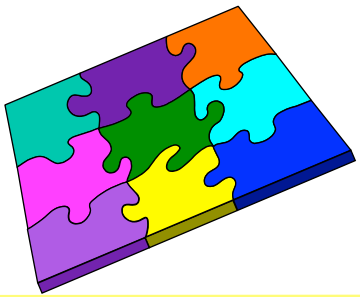
数据库安全性 —— 保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

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

本章主要内容

- 数据库安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其他安全性保护
- 小结



4.1 数据库安全性概述

数据库的不安全因素

安全标准简介

4.1.1 数据库的不安全因素

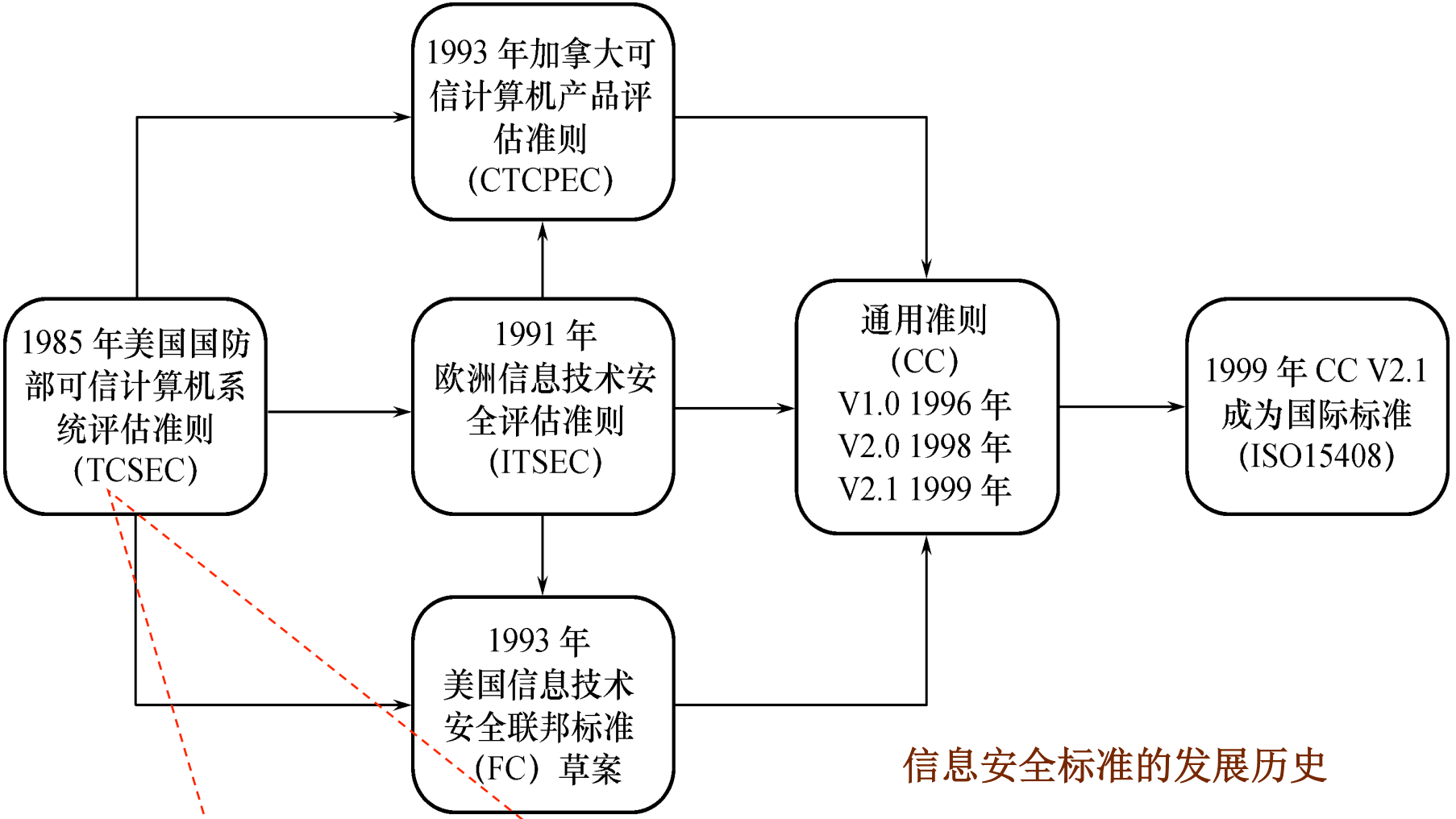
- 非授权用户对数据库的恶意存取和破坏
  - 一些黑客（**Hacker**）和犯罪分子在用户存取数据库时猎取用户名和用户口令，然后假冒合法用户偷取、修改甚至破坏用户数据。
  - 数据库管理系统提供的安全措施主要包括用户身份鉴别、存取控制和视图等技术。
- 数据库中重要或敏感的数据被泄露
  - 数据库管理系统提供的主要技术有强制存取控制、数据加密存储和加密传输、审计日志分析等。
- 安全环境的脆弱性
  - 数据库的安全性与计算机系统的安全性紧密联系
    - 计算机硬件、操作系统、网络系统等的安全性
  - 建立一套可信（**Trusted**）计算机系统的概念和标准

4.1 数据库安全性概述

数据库的不安全因素

安全标准简介

4.1.2 安全标准简介


1993 年加拿大可信计算机产品评估准则 (CTCPEC)
1985 年美国国防部可信计算机系统评估准则 (TCSEC)
1991 年欧洲信息技术安全评估准则 (ITSEC)
通用准则 (CC) V1.0 1996 年 V2.0 1998 年 V2.1 1999 年
1999 年 CC V2.1 成为国际标准 (ISO15408)
1993 年美国信息技术安全联邦标准 (FC) 草案
信息安全标准的发展历史
trusted computer system evaluation criteria （Orange Book）

4.1.2 安全标准简介

- TCSEC/TDI标准的基本内容
  - TCSEC(trusted computer system evaluation criteria, 可信计算机系统评估标准) （桔皮书）
  - 1991年4月美国NCSC（国家计算机安全中心）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》(Trusted Database Interpretation, 简称TDI)（紫皮书）
  - 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
  - 目前国际上广泛采用的美国标准,此标准将数据安全划分为四组七级

4.1.2 安全标准简介

➲ 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

偏序向下兼容

B2以上的系统还处于理论研究阶段

大力发展安全产品，试图将目前仅限于少数领域（如军队等）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

4.1.2 安全标准简介

- D级
  - 将一切不符合更高标准的系统均归于D组
  - 典型例子：DOS是安全标准为D的操作系统
    - DOS在安全性方面几乎没有什么专门的机制来保障
- 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

4.1.2 安全标准简介

- 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
  - 典型例子
    - 操作系统
      - Microsoft的Windows 2000，
      - 数字设备公司的Open VMS VAX 6.0和6.1
    - 数据库
      - Oracle公司的Oracle 7
      - Sybase公司的 SQL Server 11.0.6

4.1.2 安全标准简介

- **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
  - 典型例子
    - 操作系统
      - 数字设备公司的SEVMS VAX Version 6.0
      - 惠普公司的HP-UX BLS release 9.0.9+
    - 数据库
      - Oracle公司的Trusted Oracle 7
      - Sybase公司的Secure SQL Server version 11.0.6
      - Informix公司的Incorporated INFORMIX-OnLine / Secure 5.0

4.1.2 安全标准简介

- **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**。
  - 经过认证的**B2**级以上的安全系统非常稀少
  - 典型例子
    - 操作系统
      - **Trusted Information Systems**公司的**Trusted XENIX**产品
    - 标准的网络产品
      - **Cryptek Secure Communications**公司的**LLC VSLAN**产品
    - 数据库
      - 很少

4.1.2 安全标准简介

- **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。

4.1.2 安全标准简介

- 国际通用准则（CC）
  - ISO统一现有多种准则的结果，是目前最全面的评估准则。
  - 提出国际公认的表述信息技术安全性的结构,CC认为安全的实现应构建在如下的层次框架之上（自下而上）。

    1)安全环境：使用评估对象时必须遵照的法律和组织安全政策以及存在的安全威胁。

    2)安全目的：对防范威胁、满足所需的组织安全政策和假设声明。

    3)评估对象安全需求：对安全目的的细化，主要是一组对安全功能和保证的技术需求。

    4)评估对象安全规范：对评估对象实际实现或计划实现的定义。

    5)评估对象安全实现：与规范一致的评估对象实际实现。

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
      - 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安全功能
  - 结构开放、表达方式通用

4.1.2 安全标准简介

- CC文本组成
  - 简介和一般模型
    - 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

4.1.2 安全标准简介

CC评估保证级（Evaluation Assurance Level, 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

4.1.2 安全标准简介

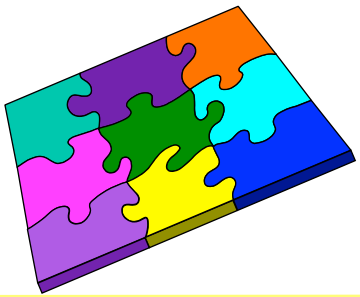
- 国家标准**GB17859-99**是我国计算机信息系统安全等级保护系列标准的核心，是实行计算机信息系统安全等级保护制度建设的重要基础。此标准将信息系统分成**5**个级别,如下表中所示：
- TCSEC标准与国标比较

TCSEC标准	我国标准
D级标准	无
C1级标准	第1级：用户自主保护级
C2级标准	第2级：系统审计保护级
B1级标准	第3级：安全标记保护级
B2级标准	第4级：结构化保护级
B3级标准	第5级：访问验证保护级
A级标准	

第四章　数据库安全性

本章主要内容

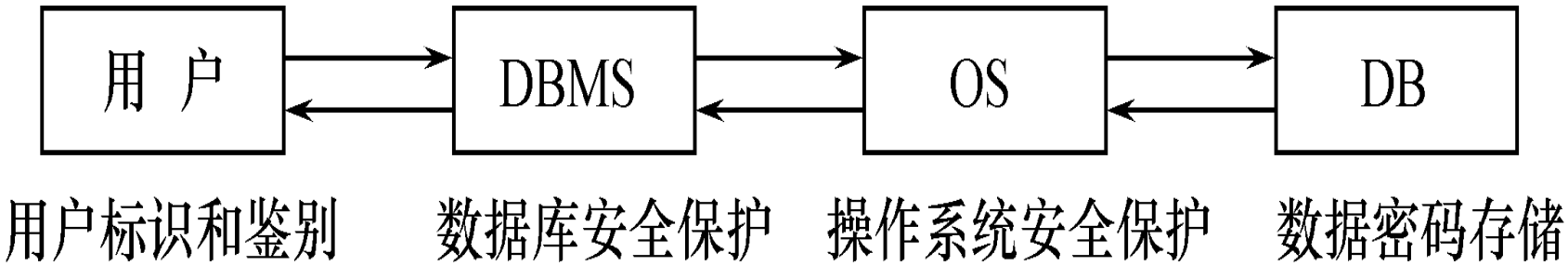
- 数据库安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其他安全性保护
- 小结



概述

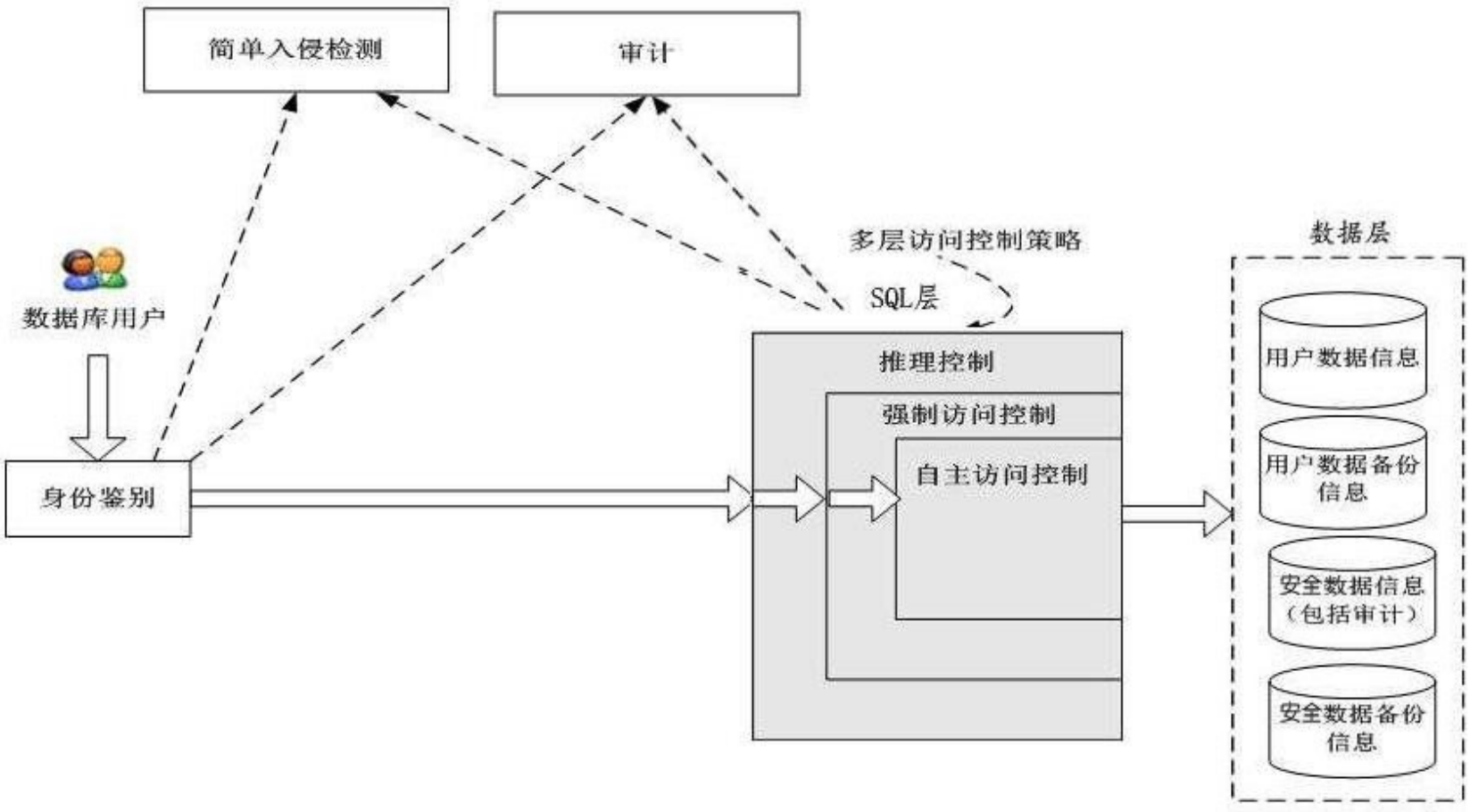
- 非法使用数据库的情况
  - 编写合法程序绕过DBMS及其授权机制
  - 直接或编写应用程序执行非授权操作
  - 通过多次合法查询数据库从中推导出一些保密数据

- 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

4.2 数据库安全性控制



数据库管理系统安全性控制模型

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

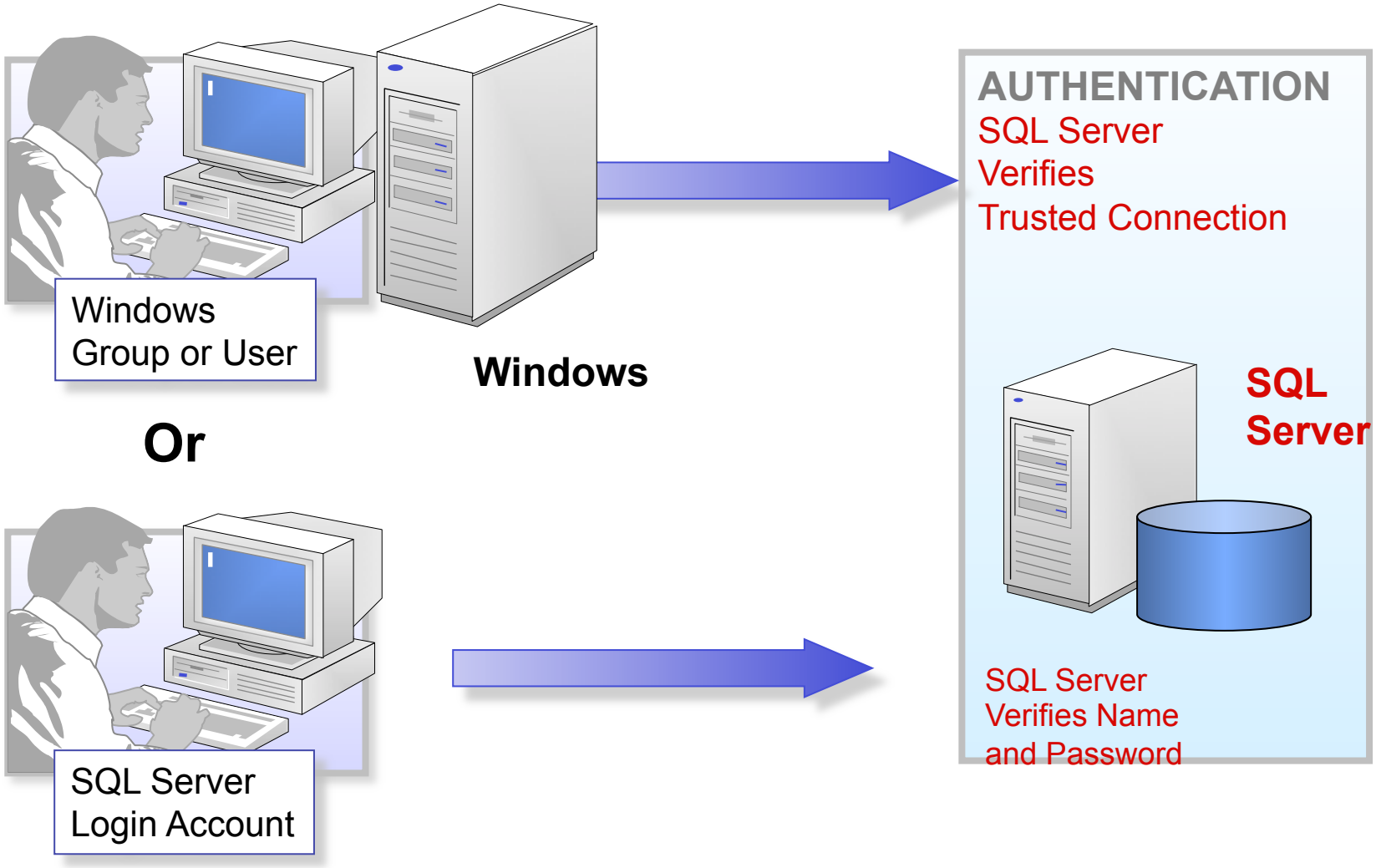
4.2.1 用户标识与鉴别

- 用户标识与鉴别（Identification & Authentication）
  - 系统提供的最外层安全保护措施
- 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）
- 用户身份鉴别方法
  - 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
  - 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
  - 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
  - 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

4.2.1 用户标识与鉴别

- SQL Server的安全机制一般主要包括三个方面:
  - 服务器级别的安全机制:这个级别的安全性主要通过登录帐户进行控制,要想访问一个数据库服务器,必须拥有一个登录帐户。登录帐户可以是Windows账户或组，也可以是SQL Server的登录账户。登录账户可以属于相应的服务器角色。至于角色，可以理解为权限的组合。
  - 数据库级别的安全机制:这个级别的安全性主要通过用户帐户进行控制,要想访问一个数据库，必须拥有该数据库的一个用户账户身份。用户账户是通过登录账户进行映射的，可以属于固定的数据库角色或自定义数据库角色。
  - 数据对象级别的安全机制:这个级别的安全性通过设置数据对象的访问权限进行控制。

- 用户标识与鉴别（**Identification & Authentication**）



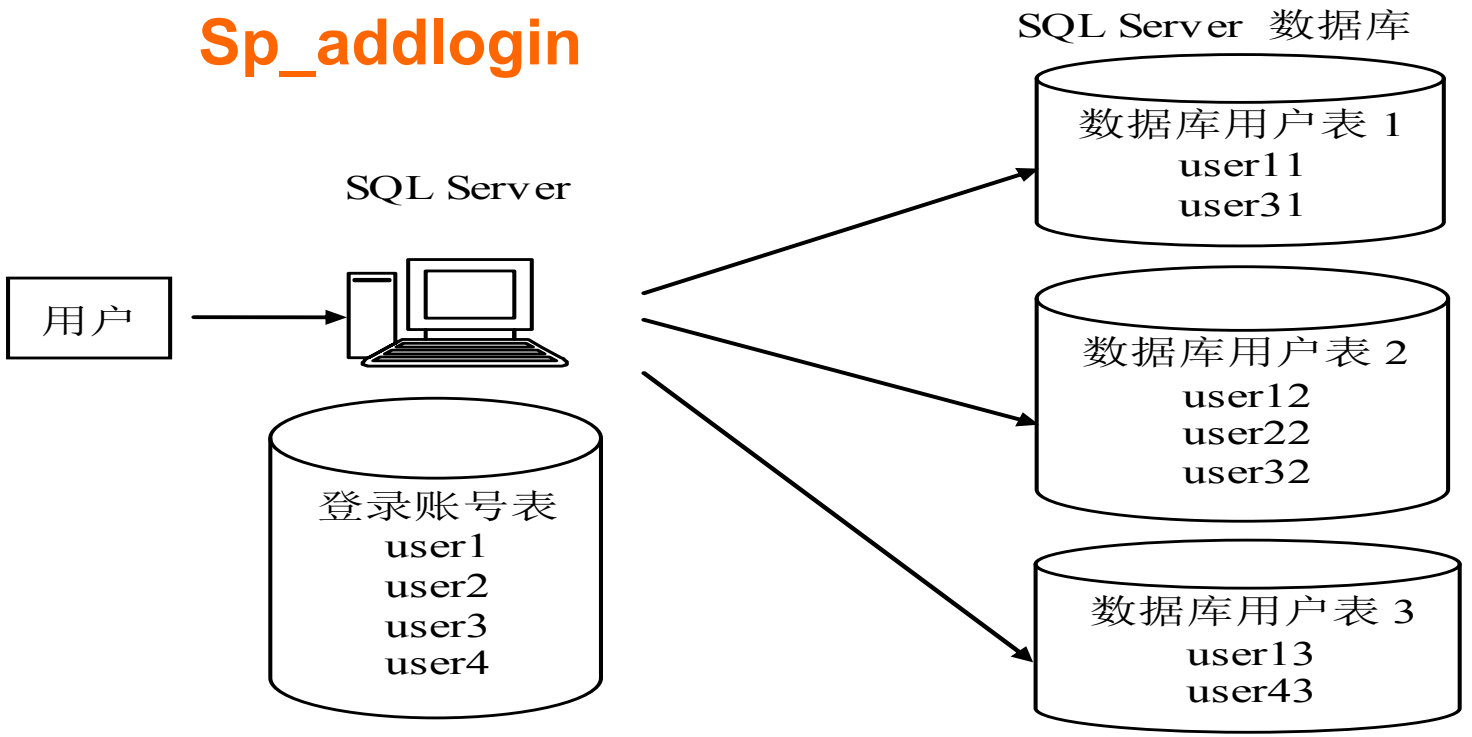
4.2.1 用户标识与鉴别

服务器登录账号：用来和SQL Server连接，有了登录账号您才能连接上SQL Server，才有使用SQL Server的权利。

数据库用户：登录账号有对应的用户账号，才有访问数据库对象的权利。

Sp_addlogin

Sp_adduser

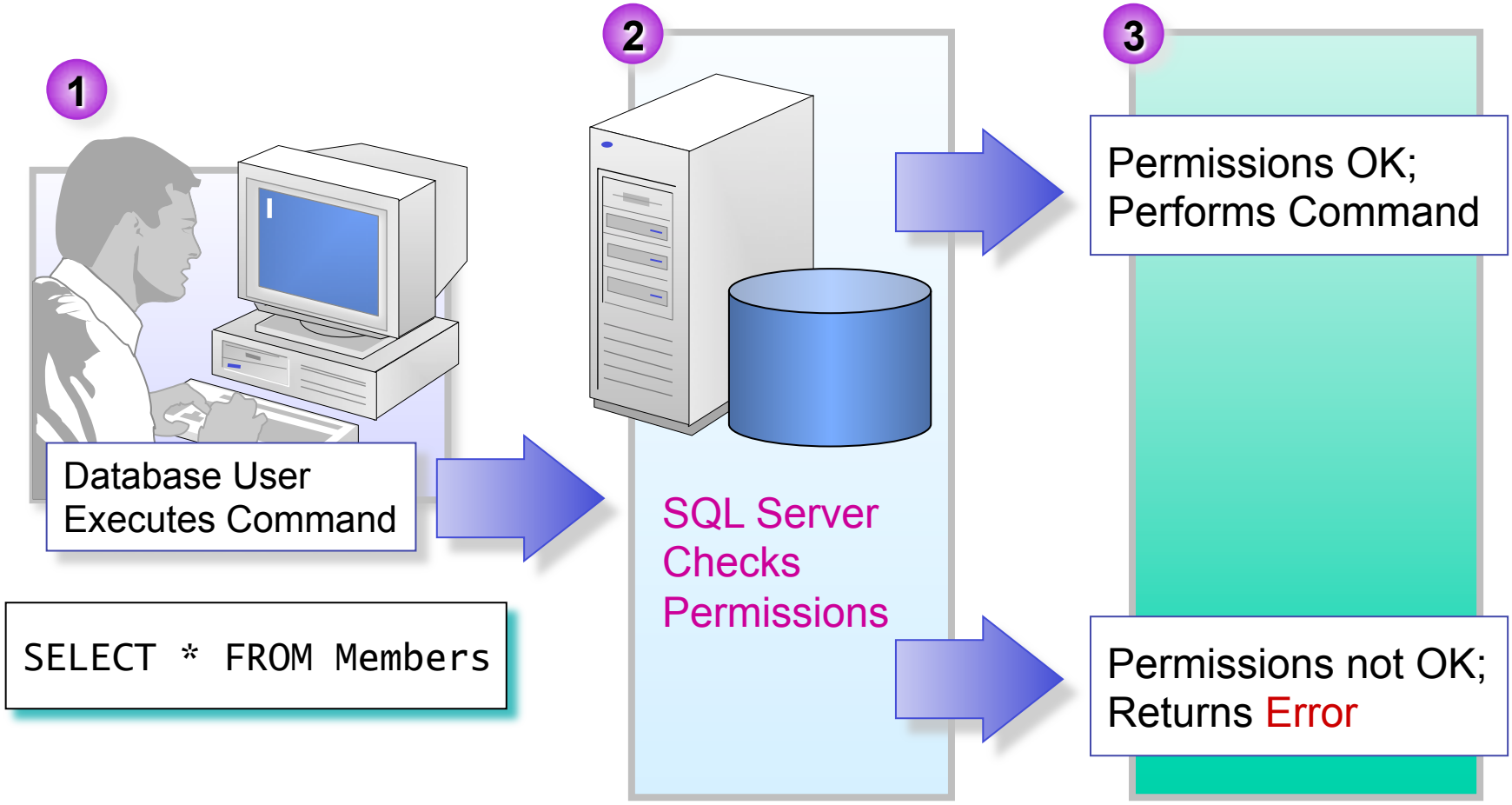


4.2 数据库安全性控制

- 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

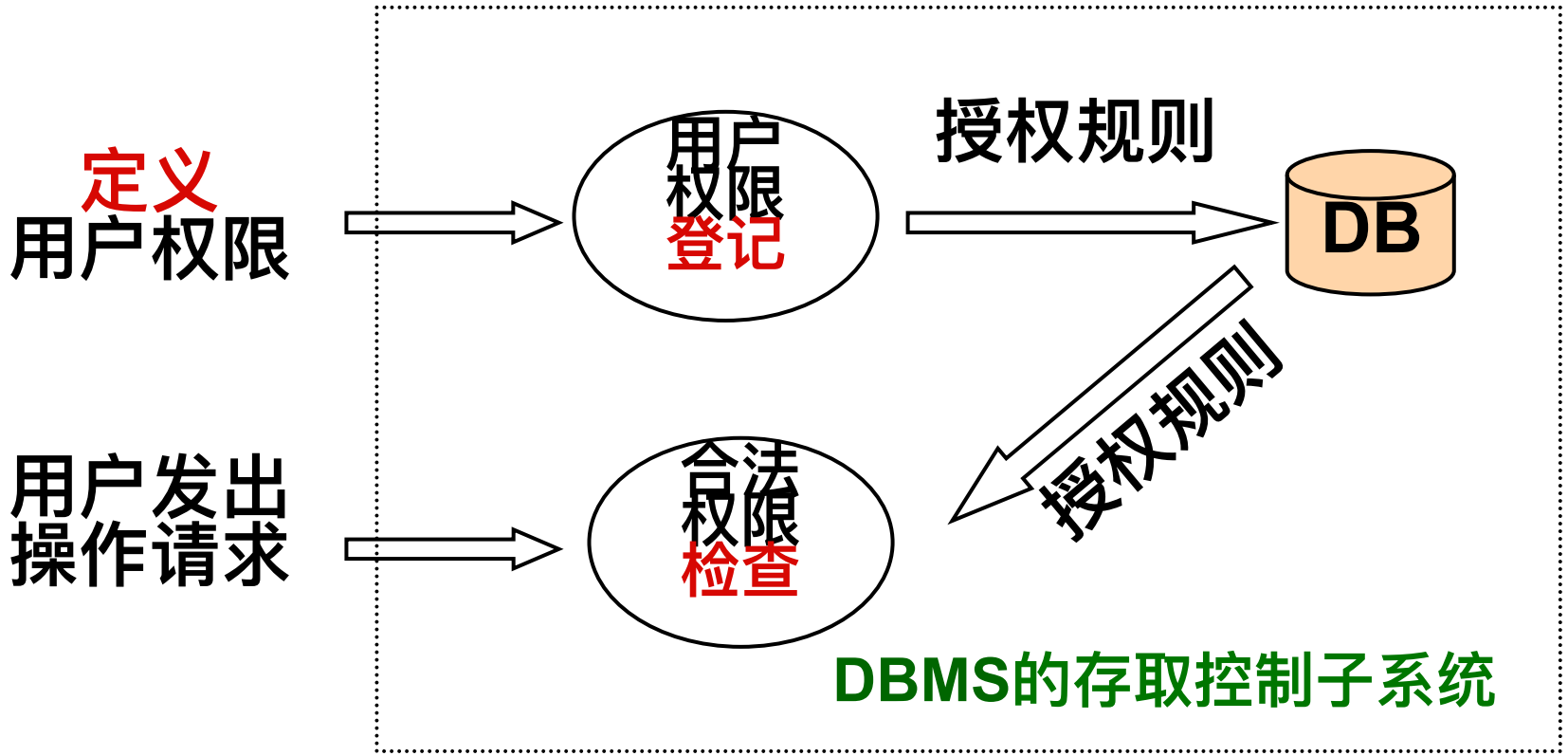
4.2.3 自主存取控制方法

操作权限



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**的存取控制子系统



4.2.2 存取控制

- 常用存取控制方法
  - 自主存取控制（Discretionary Access Control，简称DAC）
    - C2级
    - 灵活

    用户对于不同的数据对象有相应的存取权限，而且用户还可以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
    - 严格

    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（数据库、表、字段、元组）
  - 操作类型（create、alter、delete、update、insert、select）
- 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

授权粒度

授权粒度越细，授权子系统越灵活，
但系统定义与检查权限的开销越大

4.2.3 自主存取控制方法

➲ 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 授权与回收

一、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];

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）

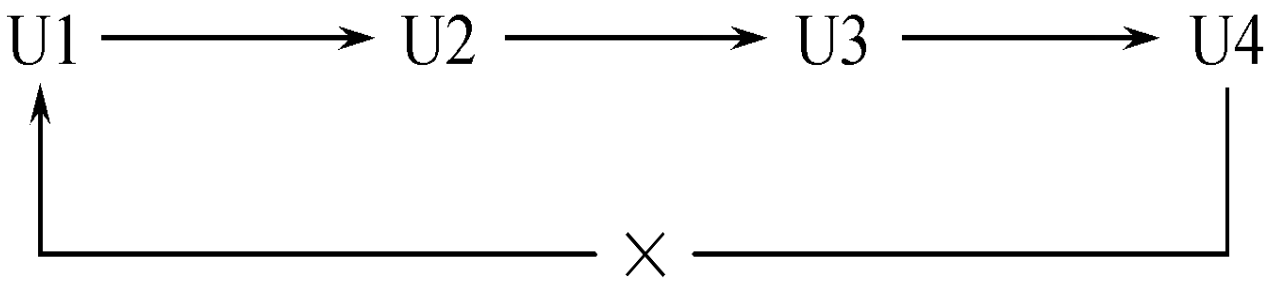
- 语义：将对指定操作对象的指定操作权限授予指定的用户

4.2.4 授权与回收

一、GRANT

- **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



4.2.4 授权与回收

一、GRANT

［例1］把查询Student表权限授给用户U1

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

4.2.4 授权与回收

一、GRANT

[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;
```

4.2.4 授权与回收

一、GRANT

[例3] 把对表SC的查询权限授予所有用户

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

4.2.4 授权与回收

一、GRANT

[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

一、GRANT

[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;
```

4.2.4 授权与回收

一、GRANT

- 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。

4.2.4 授权与回收

一、GRANT

➲ 传播权限

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

4.2.4 授权与回收

二、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>]...;

二、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;
```

二、REVOKE

[例9] 收回所有用户对表SC的查询权限

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

二、REVOKE

[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

4.2.4 授权与回收

二、REVOKE

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

4.2.4 授权与回收

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

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］

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

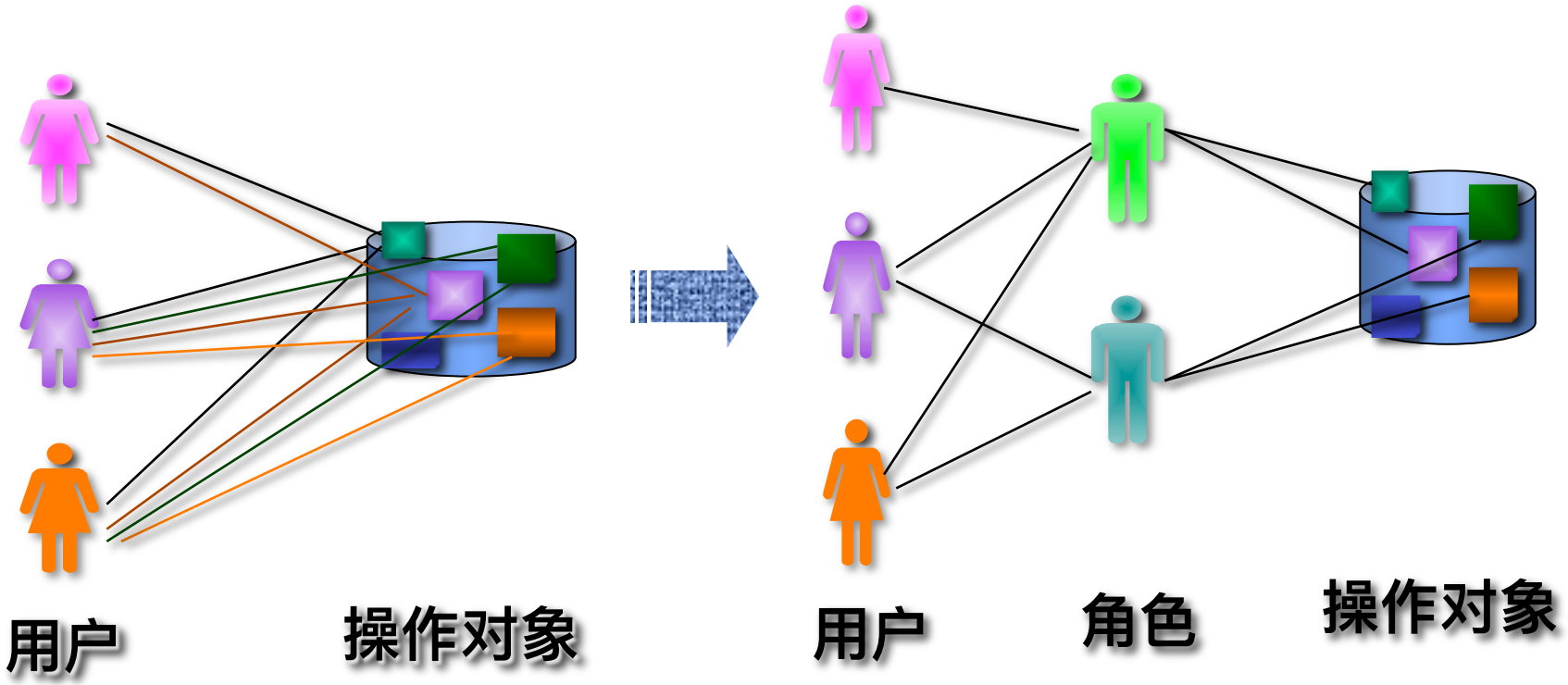
权限与可执行的操作对照表

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



4.2.5 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>]…

- 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>]…

4.2.5 数据库角色

[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;
```

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；
```

4.2.5 数据库角色

[例12] 角色的权限修改

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

[例13]

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

4.2.5 数据库角色

- 一、角色的创建

  CREATE ROLE <角色名>

T-SQL:
Sp_addrole

- 二、给角色授权

GRANT <权限>[，<权限>]…ON <对象类型>对象名

TO <角色>[，<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]

T-SQL:
Sp_addrolemenmber

- 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>]…

T-SQL:
Sp_droprole

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*

4.2.6 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)
  - 保证更高程度的安全性
  - 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

4.2.6 强制存取控制方法

主体、客体与敏感度标记

主体

- 是系统中的活动实体，包括：
  - DBMS所管理的实际用户、 代表用户的各进程

客体

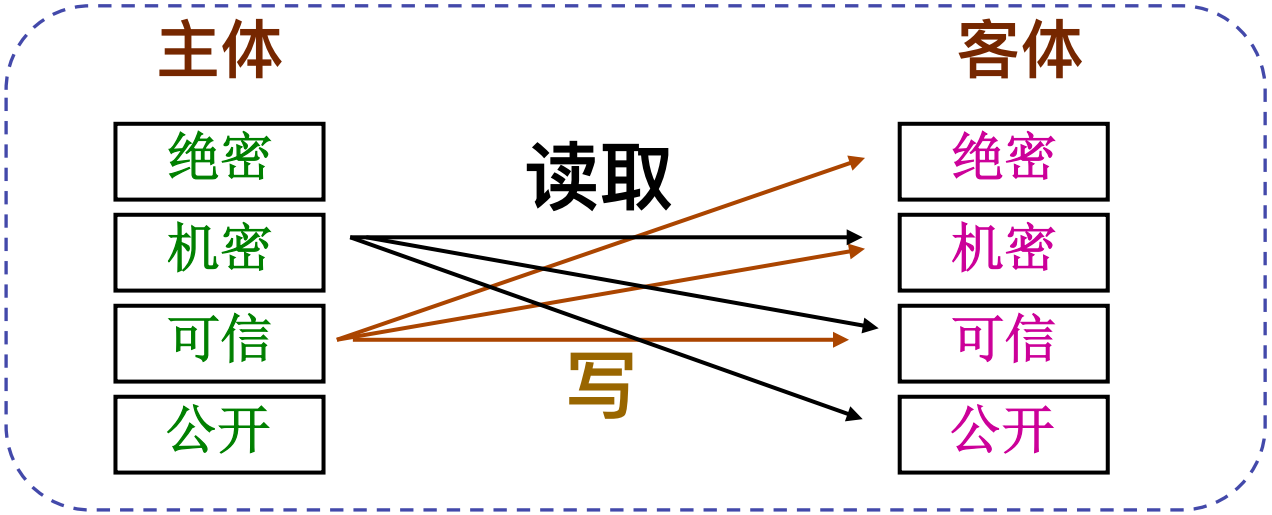
- 是系统中的被动实体，是受主体操纵的，包括：
  - 文件、 基表、 索引、 视图等

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。
  - 绝密（Top Secret，TS）、机密（Secret，S）、可信（Confidential，C）、公开（Public ,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）。

4.2.6 强制存取控制方法

- 强制存取控制规则
  - 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
  - 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 <= 客体的密级 → 主体能写客体

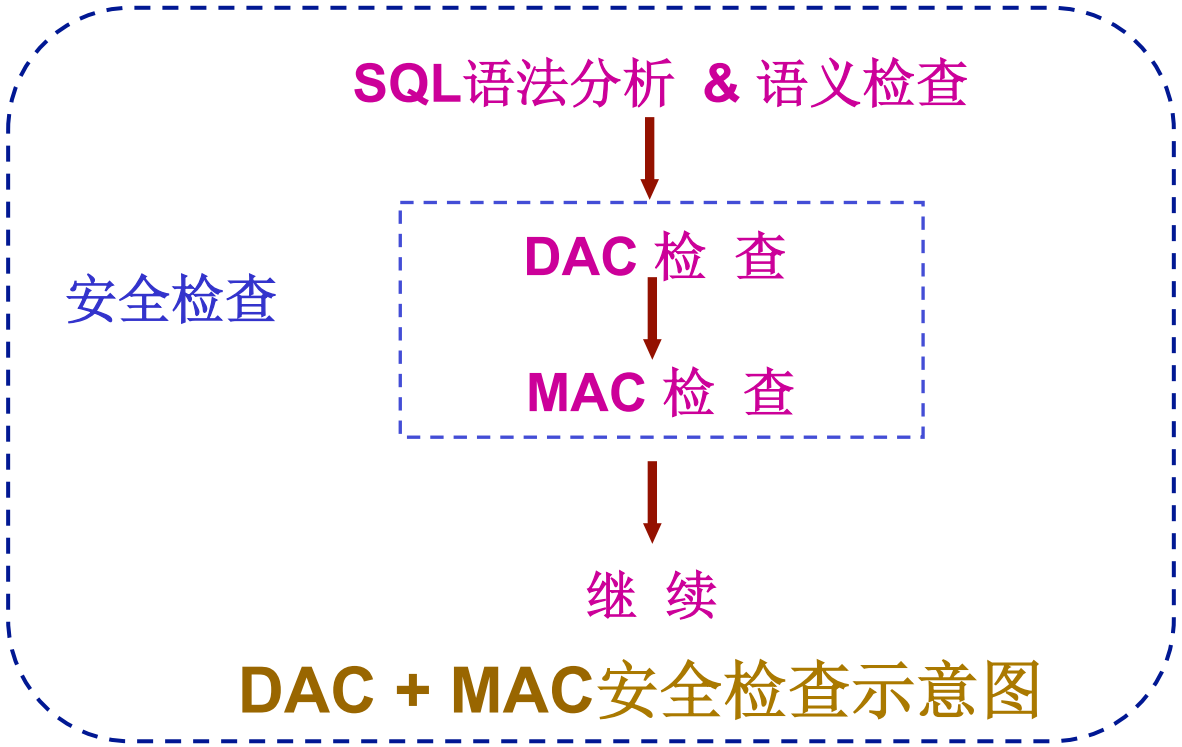


- 规则的共同点
  - *禁止了拥有高许可证级别的主体更新低密级的数据对象*

4.2.6 强制存取控制方法

- **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
    - 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

※ 先进行DAC检查，通过DAC检查的数据对象再由系统进行MAC检查，只有通过MAC检查的数据对象方可存取。

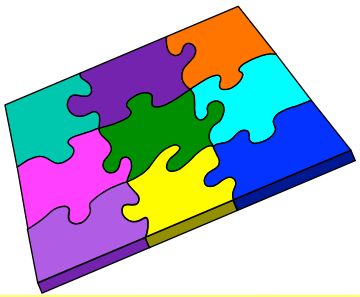


DAC + MAC安全检查示意图

第四章　数据库安全性

本章主要内容

- 数据库安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其他安全性保护
- 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
  - 主要功能是提供数据独立性，无法完全满足要求
  - 间接实现了支持存取谓词的用户权限定义

[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';
```

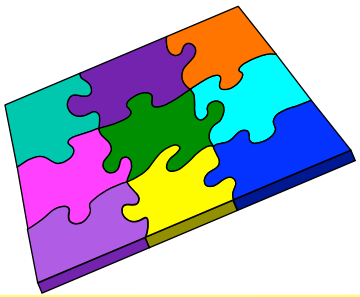
(2) 在视图上进一步定义存取权限

```
GRANT  SELECT
ON  CS_Student
TO 王平 ;
  GRANT ALL
PRIVILIGES
 ON  CS_Student
 TO  张明;
```

第四章　数据库安全性

本章主要内容

- 数据库安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其他安全性保护
- 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

4.4 审计

- 审计事件
  - 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的
  - 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
    - 对特定模式对象上进行的SELECT或DML操作的审计

4.4 审计

- 审计功能
  - 基本功能
    - 提供多种审计查阅方式
  - 提供多套审计规则：一般在初始化设定
  - 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

4.4 审计

- 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

- 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；
```

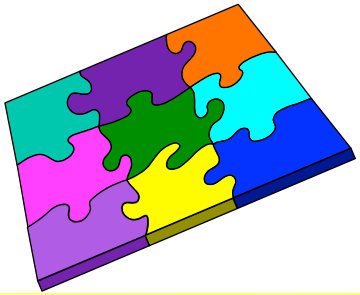
4.4 审计

- 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

第四章　数据库安全性

本章主要内容

- 数据库安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其他安全性保护
- 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（术语为明文，Plain text）变换为不可直接识别的格式（术语为密文，Cipher text）
  - 不知道解密算法的人无法获知数据的内容

- 山西票号创造了一套用汉字做符号的保密办法，用来作为汇票签发时间和银两数目的密押。每个票号的符号不同，而且又是不断变更的。
  - 如用“谨防假票冒取，勿忘细视书章”12个字，作为1年12个月每个月的代号。
  - “堪笑川情薄，天道最公平，昧心图自利，阴谋害他人，善恶终有报，到头必分明”30个字，作为1个月30天的每天的代号。
  - 汇票上的银两数字和单位，用“生客多察看，斟酌而后行”或“赵氏连城璧，由来天下传”10个字，代表“壹贰叁肆伍陆柒捌玖拾”10个数字，
  - 用“国宝流通”4个字，代表“万千百十”单位，

- 比如十月五日存银元两万两
  - “视薄客国”。

4.5 数据加密

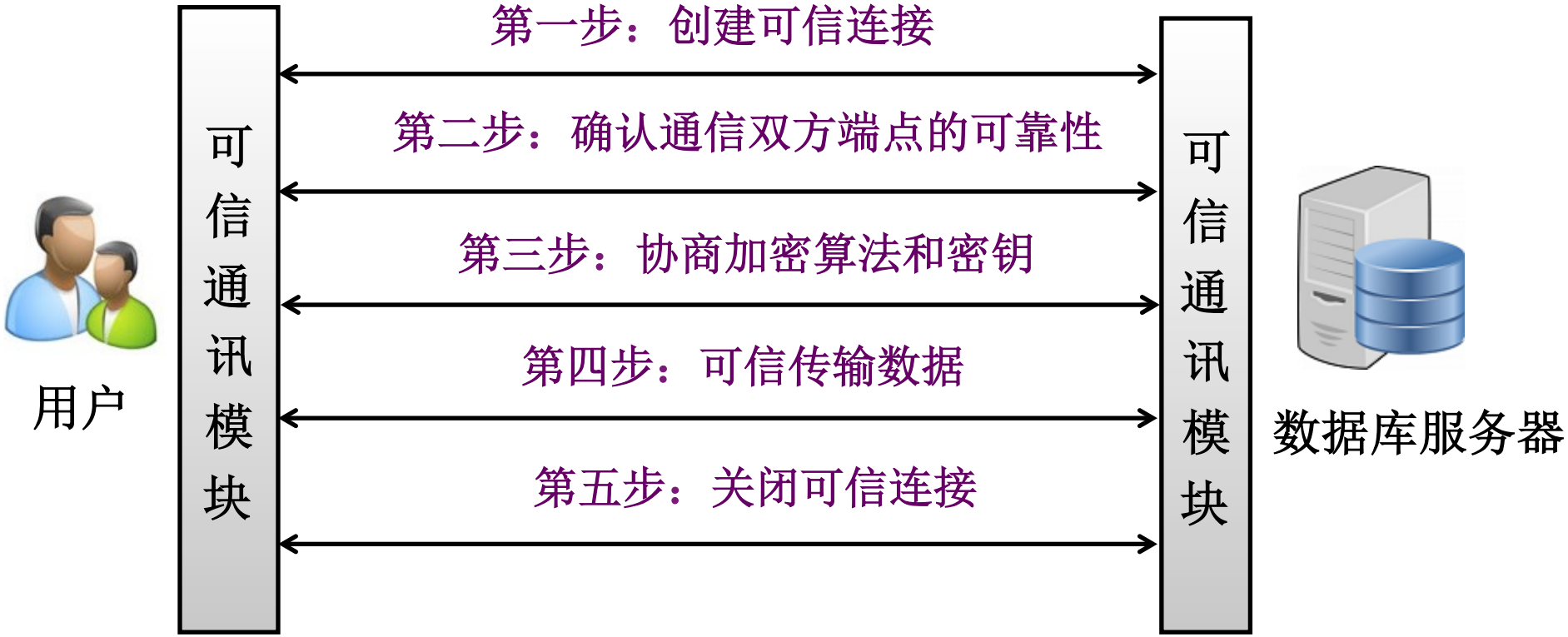
- 加密方法
  - 存储加密
    - 透明存储加密
      - 内核级加密保护方式，对用户完全透明
      - 将数据在写到磁盘时对数据进行加密，授权用户读取数据时再对其进行解密
      - 数据库的应用程序不需要做任何修改，只需在创建表语句中说明需加密的字段即可
    - 非透明存储加密
      - 通过多个加密函数实现

内核级加密方法：性能较好，安全完备性较高

4.5 数据加密

- 加密方法
  - 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

4.5 数据加密

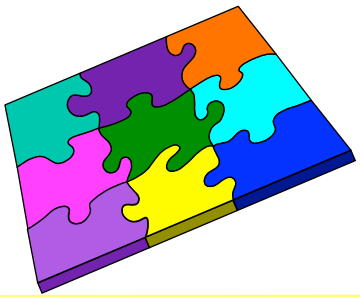


基于安全套接层协议(Security Socket Layer，SSL）
数据库管理系统可信传输方案示意图

第四章　数据库安全性

本章主要内容

- 数据库安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其他安全性保护
- 小结



4.6 其他安全性保护

- 推理控制
  - 处理强制存取控制未解决的问题
  - 避免用户利用能够访问的数据推知更高密级的数据
  - 常用方法
    - 基于函数依赖的推理控制
    - 基于敏感关联的推理控制
- 隐蔽信道
  - 处理强制存取控制未解决的问题
- 数据隐私保护
  - 控制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

4.6 统计数据库安全性

- 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

> ●如，公民健康状况数据库：公民个人的健康状况属于个人隐私，应是保密的，但是其统计信息又是应当公开的，如某疾病的发病率等。

- 统计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

例1：下面两个查询都是合法的：

1. 本公司共有多少女高级程序员？
2. 本公司女高级程序员的工资总额是多少？

如果第一个查询的结果是“1”，

那么第二个查询的结果显然就是这个程序员的工资数。

- 规则1：任何查询至少要涉及N(N足够大)个以上的记录

例2：用户A发出下面两个合法查询：

1．用户A和其他N个程序员的工资总额是多少？

2．用户B和其他N个程序员的工资总额是多少？

若第一个查询的结果是X，第二个查询的结果是Y，

由于用户A知道自己的工资是Z，

那么他可以计算出用户B的工资=Y-(X-Z)。

- 规则2：任意两个查询的相交数据项不能超过M个

可以证明，在上述两条规定下，如果想获知用户B的工资额

A至少需要进行1+(N-2)/M次查询

- 规则3：任一用户的查询次数不能超过1+(N-2)/M

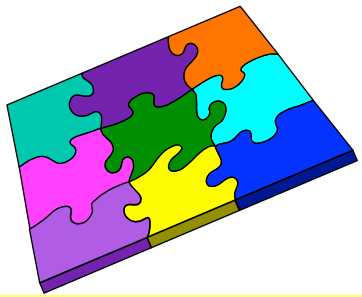
4.6 统计数据库安全性

- 安全保护策略很多，任何安全保护措施都是相对的，要付出一定代价的。
- 安全保护的原则：根据具体的应用要求权衡安全要求和成本代价后再选择合适的安全策略。
- 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

本章主要内容

- 计算机安全性概述
- 数据库安全性控制
- 视图机制
- 审计（Audit）
- 数据加密
- 其它安全性保护
- 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
- TCSEC和CC

4.7 小结

- 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

SQL Server的安全性

- SQL Server的安全性机制主要包括三个层次
  - SQL Server的登录安全性
  - SQL Server数据库的使用安全性
  - SQL Server数据库对象的使用安全性

SQL Server的安全认证模式(登录方式)

- 集成Windows OS登录方式
  - OS的用户和工作组映射为SQL Server的登录帐户；
  - 只要在OS上登录成功就承认其为SQL Server的合法用户，允许连接上SQL Server服务器。
- 标准的SQL Server登录方式
  - 用户需要用合法的帐户和正确的密码登录；
  - 先创建用户的登录帐户和设置密码，确定连接到哪个数据库；
  - 安装SQL Server时可创建管理员帐户 **sa**, 连接到**master**数据库
- 混合登录方式

SQL Server标准登录模式

- 1. 创建登录帐户
  - 使用SSMS
  - 使用存储过程 Sp_addlogin
- 2. 修改登录帐户的属性
  - 使用SSMS、使用存储过程
- 3. 删除登录帐户
  - 使用SSMS、使用存储过程 Sp_revokelogin

SQL Server用户管理

- 1. 添加数据库用户
  - SSMS或使用存储过程 Sp_adduser
- 2. 删除数据库用户
  - 使用SSMS或使用存储过程 Sp_dropuser
- 3. 特殊数据库用户
  - dbo 即数据库拥有者或数据库属主，sa可以作为他所管理系统的任何数据库的dbo用户
  - guest 代表对样板数据库拥有最基本查询的用户

SQL Server服务器角色

- 1. 角色：集中一组权限的管理单元（官位）。
- 2. SQL Server的固定服务器角色
  - 查看固定服务器角色 **Sp_helpsrvrole**
  - 添加服务器角色成员 **Sp_addsrvrolemember**
  - 删除服务器角色成员 **Sp_dropsrvrolemember**
- 3. 了解几个固定服务器角色
  - **Sysadmin** 系统管理员
  - **Serveradmin** 服务器管理员
  - **Securityadmin** 安全管理员
  - **dbcreate** 数据库创建者

SQL Server数据库角色

- 1. SQL Server的固定数据库角色（9种）
  - 查看固定数据库角色：Sp_helpdbfixedrole
  - 特殊的数据库固定角色 Public，所有数据库用户都属于public角色
- 2. 了解几个固定数据库角色
  - Db_ower 数据库属主，可进行所有DB角色的活动
  - Db_datawrite 可更新数据库用户表中的数据
  - Db_Securityadmin 数据库安全管理员
  - Db_backupoperater 数据库备份员

SQL Server数据库角色管理

- 1. 创建数据库角色及成员
  - 使用SSMS创建
  - 添加数据库角色： Sp_addrole
  - 添加数据库角色成员： Sp_addrolemember
- 2. 查看数据库角色： Sp_helprole
- 3. 删除数据库角色
  - 使用SSMS或使用存储过程Sp_droprole
- 4. 删除数据库角色成员： Sp_droprolemember

SQL Server的权限

- 1. SQL Server 权限种类(3种)
  - 对象权限：用户对数据库中的表、视图、存储过程等对象的操作权限
  - 语句权限：DDL语句权限，是否允许执行创建各种数据库对象、备份数据库和日志的语句
  - 隐含权限：指预定义的服务器角色、dbo等所拥有的权限，内置的
- 2. 对象权限的管理
  - (1) 授予权限：grant
  - (2) 撤消权限：revoke
  - (3) 拒绝访问：deny

本章习题