第6章 数据库的安全性

北京理工大学 计算机学院
张文耀
zhwenyao@bit.edu.cn

数据库的安全问题

- 数据库中的数据是可以共享的
- 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
  - 数据库中数据的共享是在**DBMS**统一的严格的控制之下的共享，即：只允许有合法使用权限的用户访问允许他存取的数据
- 数据库安全性是保护数据库不被非法使用和防止非法用户恶意造成的破坏。
- 安全性措施的防范对象是非法用户的进入和合法用户的非法操作。
- 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

主要内容

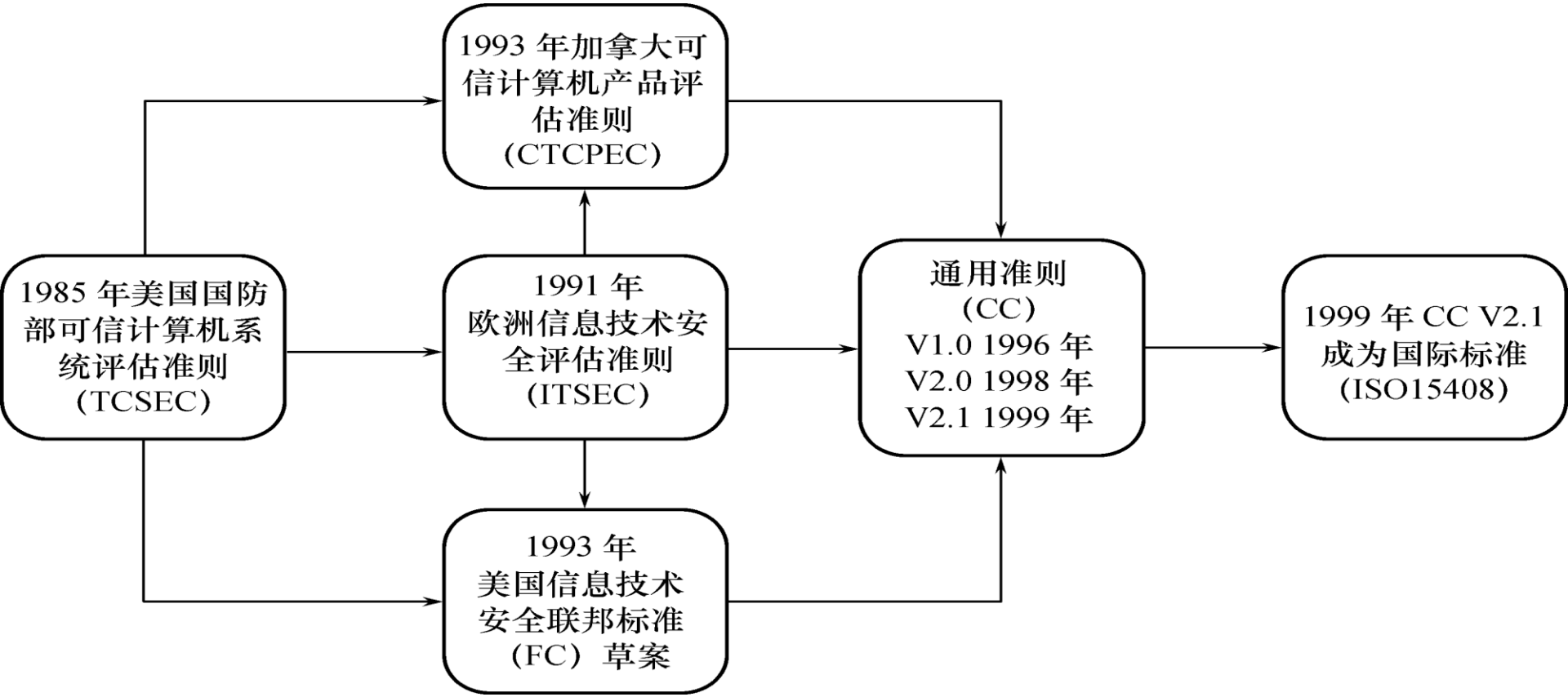
6.1 计算机安全性概述
6.2 数据库安全性概述
6.3 用户标识与鉴别
6.4 存取控制
6.5 视图机制
6.6 数据加密
6.7 数据库审计
6.8 统计数据库的安全性
6.9 SQL Server的安全控制

6.1 计算机安全性概述

- 什么是计算机系统安全性
  - 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。
- 三类计算机系统安全性问题
  - 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其所存数据的安全保护，当计算机系统受到无意或恶意的攻击时仍能保证系统正常运行，保证系统内的数据不增加、不丢失、不泄露。
  - 管理安全
  - 政策法律

- 可信计算机系统安全标准
  - **TCSEC**
    1985年美国国防部**(DoD)**颁布的《**DoD**可信计算机系统评估标准》（**Trusted Computer System Evaluation Criteria**），又称桔皮书
  - **TDI**
    1991年美国国家计算机安全中心（**NCSC**）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**），又称紫皮书。
  - **CC**
    1993年将各自独立的准则集合成单一的、能被广泛使用的**IT**安全准则（**Common Criteria**）**CC**通用标准

- 信息安全标准的发展历史



2001年我国采用CC V2.1作为国家标准

TCSEC标准

- **TCSEC标准的目的**
  - 提供一种标准，使用户可以对其计算机系统内敏感信息安全操作的可信程度做评估。
  - 给计算机行业的制造商提供一种可循的指导规则，使其产品能够更好地满足敏感应用的安全需求。
- **TDI将TCSEC扩展到数据库管理系统**
  - 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安全级别划分ABCD四组7个等级**

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）

- 四组7个等级：
  - 从低到高的顺序为D，C（C1，C2）， B（B1，B2，B3）， A（A1）
  - 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，即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要求，同时提供更多或更完善的保护能力

- **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安全保障机制
- **C1级**
  - 提供非常初级的自主安全保护
  - 能够实现用户和数据的分离，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**C1**。

- **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，并不突出标注“安全”特性；
  - 典型例子
    - 操作系统
      - **Windows NT 3.5**，
      - **Open VMS VAX 6.0**和**6.1**
    - 数据库
      - **Oracle 7**、**SQL Server 2000**、**Sybase 11**

- **B1级**
  - 标记安全保护。
  - 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
  - 真正意义上的安全产品，能满足大型企业或一般政府部门的需求，多冠以“安全”或“可信”字样。
  - 典型例子
    - 数据库
      - **Trusted Oracle 7**
      - **Sybase Secure SQL Server 11**

- **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**

- **B3级**
  - 安全域保护
  - 该级的可信运算基(TCB)必须满足访问监控器的要求
  - 审计跟踪能力更强
  - 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。

CC标准

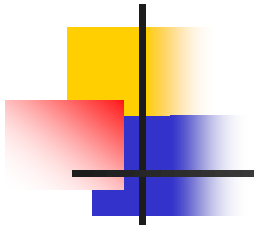
- CC（Common Criteria）的目的
  - 解决各个标准中概念和技术上的差异，将各自独立的准则集合成一组单一的、能被广泛使用的IT安全准则。
- CC的贡献
  - 提出了国际公认的表述信息技术安全性的结构
  - 把信息产品的安全要求分为：安全功能要求和安全保证要求。
- CC文本的组成
  - 简介和一般模型
  - 安全功能要求
  - 安全保证要求

- **CC的安全评估保证级划分**
  - 根据系统对安全保证要求的支持情况提出了评估保证级（**Evaluation Assurance Level，EAL**）
  - 从**EAL1**至**EAL7**共分为七级，按保证程度逐渐增高

CC级别	定　　义	TCSEC级别
EAL1	功能测试（functionally tested）	
EAL2	结构测试（structurally tested）	C1
EAL3	系统地测试和检查	C2
EAL4	系统地设计、测试和复查	B1
EAL5	半形式化设计和测试	B2
EAL6	半形式化验证的设计和测试	B3
EAL7	形式化验证的设计和测试	A1

6.2 数据库安全性概述

- 数据的安全性是指在数据库应用系统的不同层面提供安全防范措施，保护数据库不受恶意访问。
- 非法使用数据库的情况：
  - 用户编写一段合法的程序绕过**DBMS**及其授权机制，通过操作系统直接存取、修改或备份数据库中的数据；
  - 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；
  - .....
- 破坏安全性的行为可能是无意的、故意的、恶意的。
- 杜绝对数据库的恶意访问几乎是不可能的。



- 数据库安全性的主要目标是保证数据的
  - 完整性
  - 可用性
  - 保密性
  - 可审计性

- **数据库的安全性不是独立的**

  不仅涉及数据库的安全机制，也涉及到硬件系统、操作系统、网络系统的安全机制：

  - 数据库系统层
    - 数据库系统需要保证只允许那些获得授权的用户访问权限范围内的数据，权限范围外的数据不允许访问和修改。
  - 操作系统层
    - 不管数据库系统多安全，操作系统安全性方面的弱点也会造成数据的不安全隐患。
  - 网络层
    - 由于几乎所有的数据库系统都允许通过终端或网络进行远程访问，网络层的安全同样需要考虑。

- 数据库安全性控制的常用技术和方法
  - 访问控制技术（用户标识与鉴别）
  - 存取控制技术
  - 视图
  - 数据加密
  - 数据库审计

- 访问控制技术：防止未授权的人访问系统本身，这种安全问题对所有计算机系统都存在。访问控制技术主要通过创建用户帐户和口令、由**DBMS**控制登录过程来实现。
- 存取控制技术：**DBMS**必须提供相应的技术保证用户只能访问他的权限范围内的数据，而不能访问数据库的其他内容。
- 数据加密技术：用于保护敏感数据的传输和存储，可以对数据库的敏感数据提供额外的保护。
- 数据库审计：审计是在数据库系统运行期间，记录数据库的访问情况，以利用审计数据分析数据库是否受到非法存取。

6.3 用户标识与鉴别

- 用户标识和鉴别
  - 保证数据库安全性最简单、最基本的措施；
  - 系统提供的最外层安全保护措施；
  - 任何对数据库系统的访问都需要通过用户标识来获得授权，拥有数据库登录权限的用户才能进入数据库管理系统；
  - 属于访问控制技术
  - 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）

- 基本方法
  - 静态口令鉴别：用户自己设定用户名（user name）以及口令（password）来标识用户身份，口令是静态不变。
  - 动态口令鉴别：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：基于生物特征识别的身份认证技术，例如指纹识别、人脸识别、虹膜识别…
  - 智能卡的鉴别：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
  - …

6.4 存取控制

- 存取控制是数据库系统内部对已经进入系统的合法用户的访问控制，目的是只允许用户进行权限范围内的数据存取操作。
- 存取控制技术是数据库安全系统中的核心技术，也是最有效的安全手段。

- 存取控制机制包括两部分内容
  - 定义用户权限
    - 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，并将用户权限登记到数据字典中，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；
    - 系统查找数据字典，进行合法性检查，确保只执行合法操作。
- 用户权限定义和合法权限检查机制一起组成了**DBMS**的安全子系统。

- 存取控制机制可分为：
  - 自主存取控制（**Discretionary Access Control ,DAC**）
    - 同一用户对于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
    - 比较灵活，**C2**级
  - 强制存取控制（**Mandatory Access Control, MAC**）
    - 每个数据对象都被标以一定的安全类别或安全级别
    - 每个用户也被标以一定的安全类别或安全级别
    - 对于任意一个对象，只有具有合法分类级别的用户才可以存取
    - 比较严格，**B1**级

6.4.1 自主存取控制

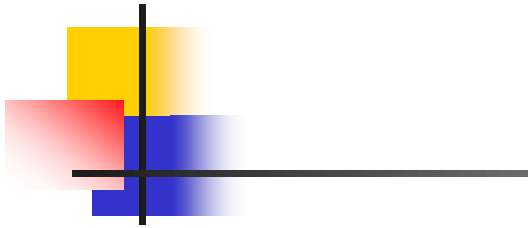
- 一种基于存取矩阵的模型，包括三要素：
  - 主体（Subject）
    是指一个提出请求或要求的实体，主体可以是DBMS所管理的实际用户，或其它任何代表用户行为的进程、作业和程序。
  - 客体（Object）
    是接受其他实体访问的被动实体，受主体操纵；客体可以是文件、记录、视图等。
  - 控制策略
    是主体对客体的操作行为集和约束条件集，即主体对客体的访问规则集。

- 在自主存取控制模型中，主体、客体和控制策略构成了访问控制矩阵
  - 矩阵的列标识主体
  - 矩阵的行表示客体
  - 矩阵中的元素是控制策略（如读、写、删除和修改等）

主体 / 客体	主体 1	主体 2	…	主体 n
客体 1	写	读	…	读/写
客体 2	写	读	…	读
…	…	…	…	…
客体 m	读	读/写	…	写

- 自主访问控制
  - 主体按访问控制矩阵中的权限要求访问客体
  - 每个用户对每个数据对象都要给定某个级别的存取权限，例如读、写等。
  - 当用户申请以某种方式存取某个数据对象时，系统根据存取矩阵判断用户是否具备此项操作权限，以此决定是否许可用户执行该操作。
  - 在自主访问控制中，访问控制的实施由系统完成。
  - 访问控制矩阵的元素是可以更改的。

- 访问控制矩阵需要定义存取权限。
- 在数据库系统中，定义存取权限称为授权。
- 用户权限包括两个要素
  - 数据库对象
  - 操作类型
- 定义用户存取权限就是定义用户可以在哪些数据库对象上进行哪些类型的操作。



- 关系数据库中授权的数据对象
  - 数据库模式
  - 基本表
  - 视图
  - 元组（记录）
  - 属性列
  - 索引

- 各个数据对象所允许的操作

数据对象	操 作 类 型
模式	**CREATE SCHEMA，DROP**
基本表	**CREATE TABLE，ALTER TABLE， DROP**
视图	**CREATE VIEW， DROP**
索引	**CREATE INDEX， DROP**
表和视图的元组	**SELECT，INSERT，UPDATE，DELETE，REFERENCES，ALL PRIVILEGES**
属性列	**SELECT，INSERT，UPDATE， DELETE，REFERENCES，ALL PRIVILEGES**

- 授权粒度
  - 授权粒度是指可以定义权限的数据对象的范围；
  - 关系数据库中授权的数据对象粒度从大到小依次为：数据库、 表、 属性列、 行（记录）；
  - 授权的粒度越细，即可以定义的数据对象的范围越小，授权子系统就越灵活，但系统定义与检查权限的开销也会相应地增大；
  - 授权粒度是衡量授权机制是否灵活的一个重要指标；
  - 另外，能否提供与数据值有关的授权反映了授权子系统精巧程度。

SQL自主存取控制的实现

- 大型数据库管理系统几乎都支持自主控制存取
- **SQL**标准通过**GRANT**和**REVOKE**实现自主存取控制
  - **4.6**节介绍
  - **DBA**拥有最高权限
  - 数据对象的所有者（**Owner**）拥有该对象的所有权限
  - 有一项特殊的权限：授予权限的权限
  - 授权实例**P120－121**

【例】GRANT INSERT
ON TABLE SC
TO U6
WITH GRANT OPTION;

REVOKE UPDATE(Sno)
ON TABLE Student
FROM U4;

数据库角色

- 在用户数量比较大的情况下，为了便于权限管理，需要引入角色的概念。
- 数据库角色（**Role**）
  - 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。
- 创建角色
  - **CREATE ROLE <角色名>**

- 给角色授权

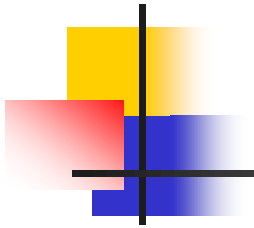
  GRANT <权限1>[，<权限2>]…
  ON <对象类型>对象名
  TO <角色1>[，<角色2>]…

- 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 GRANT OPTION]

- 角色权限的收回

  REVOKE <权限1>[，<权限2>]…
  ON <对象类型> <对象名>
  FROM <角色1>[，<角色2>]…



- 角色和用户的关系
  - 多对多
  - 一个用户可以拥有多个角色
  - 一个角色可以授予给多个用户
- 角色的使用

  1）把权限授予角色；

  2）把角色授予用户。

【例 】用户Wang通过角色实现将一组授权授予用户Li和Zhao。

1）Wang创建一个角色R1：

```
CREATE ROLE R1;
```

2）Wang授予角色R1拥有三个关系的查询权限：

```
GRANT SELECT
    ON TABLE  STUDENT, SC, COURSE
    TO R1;
```

3）Wang将角色R1授予用户Li和Zhao，使他们具有角色R1所包含的全部权限：

```
GRANT R1
    TO Li, Zhao;
```

权限的传播

- **SQL**提供非常灵活的授权机制
  - **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都可用**REVOKE**语句收回
- 这样的存取控制就是自主存取控制。

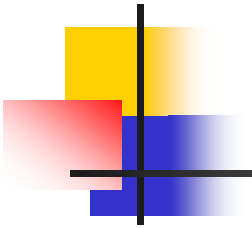
自主存取控制的优缺点

- 优点：
  - 权限控制灵活；
  - 能够通过授权机制有效控制用户对数据的存取。
- 缺点：
  - 不能提供一个确实可靠的安全保证；
  - 权限的“自主”传播可能导致故意或者是无意的数据泄漏；
  - 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。

6.4.2 强制存取控制

- **MAC－Mandatory Access Control**
- 一种安全性高的访问控制策略
- 是为了保证系统具有更高程度的安全性而采取的强制检查手段
- 不是用户能直接感知或进行控制的
- 需要在安全级别的基础上对数据或用户进行分类
- 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

- **MAC的实体分类**
  - **DBMS**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
  - **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，是受主体操纵的对象
  - 文件
  - 数据表、视图
  - 记录
  - 属性列
  - ……



- 主体和客体被标记成不同的安全分类级别（敏感度标记）
  - 安全分类级别是事先定义的
  - 典型的级别是：
    - 绝密（**Top Secret**）
    - 机密（**Secret**）
    - 可信（**Confidential**）
    - 公开（**Public**）
  - 主体的安全分类级别称为许可证级别**(Clearance Level)**
  - 客体的安全分类级别称为密级**(Classification Level)**

- 强制存取控制规则

  MAC通过对比主体的安全级别和客体的安全级别，最终确定主体能否存取客体，具体规则为：

  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；（上读）

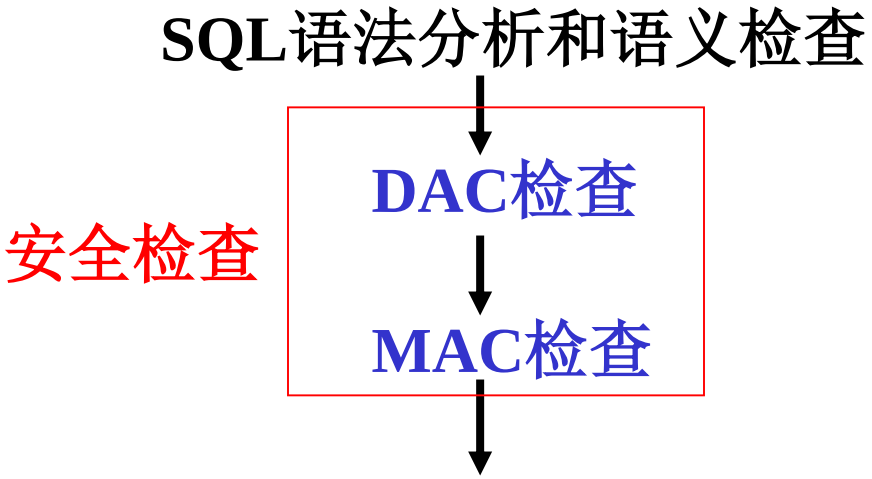
  2）仅当主体的许可证级别等于或小于客体的密级时，该主体才能写相应的客体。（下写）

  - 许可证级别低的主体不能读取安全级别比他高的客体；
  - 禁止拥有高许可证级别的主体更新低密级的数据对象，从而禁止敏感数据的泄漏。

- **MAC的特点**
  - 对数据进行密级标记，无论数据如何复制和更新，密级与数据是不可分的。
  - 只有符合密级标记要求的用户才可以操纵数据。
  - 系统给所有主体和客体分配不同级别的安全属性（安全分类级别），形成完整的系统授权状态。
  - 可以提供更高级别的安全性，但是也带来诸多的不便。

- 一般用户或程序不能修改系统安全授权状态，只有特定的系统权限管理员才可以这么做。

- **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，因为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；**
  - **进行MAC检查时，需先进行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。**



6.5 视图机制

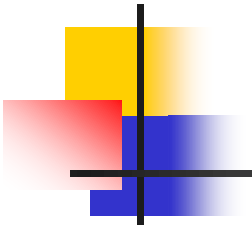
- 视图的定义和操作见**4.5**节。
- 视图是一种多角度观察数据的机制，主要功能是提供数据独立性；但同时也是一种重要的自主授权机制。
- 视图可以把数据对无权存取的用户隐藏起来，从而自动对数据提供一定程度的安全保护。
- 视图的安全保护功能并不精细，往往不能达到应用系统的要求。
- 在实际应用中，视图机制通常配合授权机制使用。

- 使用视图可实现的安全保护
  - 将用户限定在数据表中特定的数据行上。
    例如，只允许员工查看与自己有关的业务记录。
  - 将用户限定在数据表中特定的数据列上。
    例如，只允许员工查看其他人员的公共信息，如部门、办公电话等，不允许查看任何个人信息，例如工资等。
  - 将多个表中的列连接起来，使它们看起来像一个数据表。
  - 提供聚合信息而非提供详细信息。

- 视图机制与授权机制配合
  - 首先用视图机制屏蔽掉一部分保密数据，在视图上面再进一步定义存取权限；
  - 间接实现了支持存取谓词的用户权限定义；
  - 通过定义不同的视图及有选择地授予视图上的权限，可以将用户、组或角色限制在不同的数据子集内。

6.6 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数据（密文）。
  - 未授权的用户即使获得加密后的数据，也很难获得真正的数据。
- 加密方法
  - 替换方法：将明文中的每一个字符转换为另一个字符
  - 置换方法：将明文的字符按不同的顺序重新排列
  - 混合方法：将两种或多种加密方法结合起来，提高加密算法的强度



- **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
  - 有些数据库产品本身未提供加密程序，但提供了接口
  - 数据加密功能通常作为可选选项
  - 数据加密与解密操作比较费时，会占用大量系统资源
  - 一般只对高度机密的数据加密
  - 加密的方式
    - 字段级的加密
    - 文件级的加密

6.7 数据库审计

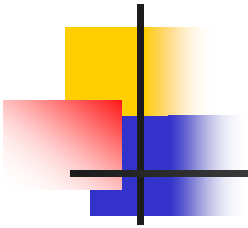
- 任何系统的安全措施都不是绝对可靠的。
- 数据库审计

  把任何人对数据库所作的任何操作都记录在审计数据库中；通过阅读审计数据库，可以发现非法访问数据库的人、时间、地点以及所有访问数据库的对象和所执行的动作。
  - 启用一个专用的审计日志（Audit Log）

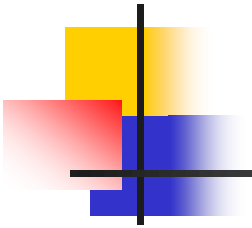
    将用户对数据库的所有操作记录在上面
  - 审计员利用审计日志

    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

- 审计数据库（审计日志）一般包含：
  - 操作类型（查询、修改等）
  - 操作终端标识与操作者标识
  - 操作日期和时间
  - 所涉及到的数据（表、视图、记录、属性等）
  - 数据的前像和后像
  - 成功或失败的注册、授权



- 数据库审计
  - 一种预防手段
  - 随时记录数据库的访问情况，作出分析以便参考
  - 在发现非法访问后提供初始记录以便进一步处理
  - 审计是在数据库系统运行期间进行的
  - 数据库审计主要应用于安全性要求较高的部门
  - 审计很费时间和空间，一般作为**DBMS**的可选特征
  - 达到**C2**以上安全级别的**DBMS**必须具备审计功能



- 审计分为
  - 用户级审计
    - 任何用户可设置的审计
    - 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操作
  - 系统级审计
    - 只能由**DBA**设置
    - 主要监测成功或失败的登录要求，以及所有数据库级权限下的操作

- 审计设置示例

  - 对修改**SC**表结构或更新**SC**表数据的操作进行审计

    AUDIT ALTER, UPDATE ON SC;

  - 取消对**SC**表的修改和更新审计

    NOAUDIT ALTER, UPDATE ON SC;

6.8 统计数据库安全性

- 统计数据库
  - 主要用于产生各类统计数据
  - 允许用户查询各种统计的信息，如平均值、汇总值等
  - 不允许查询单个记录信息，单记录信息在存取过程中应得到保护。
    例：允许查询“程序员的平均工资是多少？”
    不允许查询“程序员张勇的工资？”
- 统计数据库的安全问题
  - 在统计数据库中存在着特殊的安全性问题，即可能存在着隐蔽的信息通道，使得可以从合法的查询中推导出不合法的信息。

【例】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。

【例】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，那么用户B的工资=Y-(X-Z)。

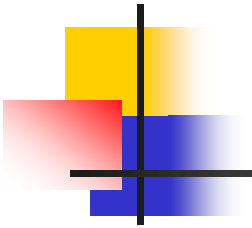
- 统计数据库的安全措施
  - 制定一些查询规则
    - 任何查询至少要涉及**N(N**足够大**)**个以上的记录
    - 任意两个查询的相交数据项不能超过**M**个
    - 任意用户的查询次数不超过**1+(N-2)/M**
  - 采取某些技术手段进行数据污染
  - 加强安全管理
- 无论采取什么样的安全机制，都仍然会存在绕过这些机制的途径
- 有效的安全措施应该使：试图破坏安全的人所花费的代价 **>>** 得到的利益

6.9 SQL Server的安全控制

- **SQL Server的安全体系结构(四个等级)**
  - 客户机操作系统的安全认证
    - 获得客户机操作系统的使用权
    - 操作系统管理员的任务
  - 登录**SQL Server**的安全认证
    - 通过登录帐户来标识用户，检验用户是否具有连接到**SQL Server**服务器的资格，决定用户能否获得**SQL Server**的访问权
  - 使用数据库的安全认证
    - 验证用户是否为具体数据库的合法用户，
    - 获取具体数据库的访问权

- 使用数据库对象的安全认证
  - 检查用户权限的最后一个阶段
  - 判断用户是否具有相应数据对象的操作权限
- **SQL Server**的安全性建立在验证和访问许可的机制上。用户访问数据需要经过三个步骤：
  - 登录验证
    连接数据库服务器
  - 用户验证
    访问数据库
  - 许可确认（权限验证）
    操作数据库对象

- SQL Server的用户
  - Windows授权用户（来自Windows的用户或组）
  - SQL Server授权用户
- SQL Server的登录验证模式
  - Windows 身份验证模式
    登录者只需要通过Windows的验证，就可以连接到SQL Server上
  - SQL Server验证模式
    登录者连接SQL Server时，必须提供SQL Server登录帐户和密码
  - 混合模式
    使用 Windows 身份验证或 SQL Server 身份验证



- **SQL Server的企业管理器可以对SQL Server登录进行管理，**
  - 选择身份验证模式
  - 设定登录成功后的当前数据库及默认的数据库语言等
- **SQL Server提供了一系列系统存储过程管理SQL Server登录功能，主要包括：**
  - **sp_grantlogin、sp_revokelogin、sp_denylogin、sp_addlogin、sp_droplogin、sp_helplogins等**

- 在SQL Server中，帐户有两类，
  - 登录帐户（login name）
  - 使用数据库的用户（user name）。
- 登录帐户的一次合法的登录只表明它通过了Windows的验证或SQL Server的验证，但不表明它可以对数据库数据进行某种操作，他只能连接到SQL Server上，并不能访问任何数据库数据。
- 如果想进一步访问SQL Server数据库中的数据，一个登录必须与一个或多个数据库的用户相关联后，才能访问数据库。

- 登录帐户必须与每一个需要访问的数据库中的用户帐户建立映射关系，每个登录帐户在一个数据库中只能有一个用户帐户，一个登录帐户可以映射为多个数据库中的用户。
- 管理数据库用户的过程实际上就是建立登录与数据库用户之间的映射关系的过程。
- 如果在新建登录过程中，指定对某个数据库具有存取权限，则在这个数据库中自动创建一个同名的用户。
- **SQL Server**提供存储过程管理数据库用户。

- **SQL Server**安装后，默认数据库中自动创建两个用户：**dbo**和**guest**。
  - **dbo**
    数据库拥有者用户，隶属于**sa**登录，拥有**public**和**db_owner**角色，具有数据库的所有特权。
  - **guest**
    客户访问用户，没有隶属的登录，拥有**public**数据库角色。
  - 任何一个登录都可以通过**guest**用户来存取相应数据。
  - 默认情况下，新建立的数据库只有一个**dbo**用户。

SQL Server的权限管理

- 在SQL Server中，权限分为三种
  - 对象权限
    - 对象权限主要针对数据库中的表、视图和存储过程，决定对这些对象能执行哪些操作 。
  - 语句权限
    - 语句权限主要指用户是否具有权限来执行某条语句 。
  - 系统权限（隐含权限、内置权限）
    - 系统权限控制那些只能由SQL Server预定义的系统角色的成员或数据库对象所有者执行的活动。
    - 预定义的系统角色的成员拥有特定的权限。
    - 数据库对象所有者可以对所拥有对象执行一切活动。

语句	对象	预定义
CREATE DATABASE	SELECT INSERT UPDATE DELETE REFERENCES — TABLE / VIEW	固定角色
CREATE TABLE		对象所有者
CREATE VIEW		
CREATE PROCEDURE		
CREATE RULE	SELECT UPDATE REFERENCES — COLUMN	
CREATE DEFAULT		
CREATE FUNCTION		
BACKUP DATABASE		
BACKUP LOG	EXEC — STORED PROCEDURE	

- 隐含权限由系统预先定义好的，不需要、也不能进行设置。
- 可以对“对象权限”和“语句权限”进行权限设置
- 在**SQL Server**中，使用
  - **GRANT**语句把权限授予某一用户以允许该用户执行某些语句或操作某些对象
  - 使用**REVOKE**语句取消用户对某一对象或语句的权限
  - 使用**Deny** 语句拒绝权限，用来禁止用户对某一对象或语句的权限

SQL Server的角色管理

- **SQL Server**管理者可以将某些用户设置为某一角色，这样只对角色进行权限设置便可实现对所有用户权限的设置，以便减少权限管理的工作量。
- 在用户成为角色成员时，用户自动拥有角色的所有权限。
- 数据库角色可以看作是对某个数据库具有相同访问权限的用户帐户和组的集合。
- 对角色的权限修改适用于该角色的所有成员。
- 数据库角色应用于单个数据库。

- **SQL Server的角色包括：**
  - 服务器角色
    角色及其权限都是预先定义的，适应于服务器范围，其权限不能被修改。
  - 数据库角色
    - 预定义的数据库角色
      其管理访问数据库的权限是预先定义的，不能进行任何修改。
    - 用户定义的数据库角色
      - 标准角色
      - 应用程序角色
        只能通过特定的应用程序间接存取数据库的数据

- **SQL Server的服务器角色**

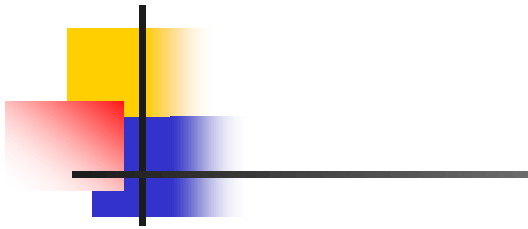
角色	*权限*
sysadmin	执行任何活动
dbcreator	创建和更改数据库
diskadmin	管理磁盘文件
processadmin	管理 SQL Server 过程
serveradmin	配置服务器设置
setupadmin	安装复制
securityadmin	管理和检查服务器账户
bulkadmin	执行 BULK INSERT 语句

角色	权限
public	维护所有默认权限
db_owner	执行所有数据库角色活动
db_accessadmin	添加和删除数据库用户, 组及角色
db_ddladmin	添加、更改或删除数据库对象
db_security admin	分配语句执行和对象权限
db_backupoperator	备份数据库
db_datareader	读取任何表中的数据
db_datawriter	添加、更改或删除所有表中的数据
db_denydatareader	不能读取任何表中的数据
db_denydatawriter	不能更改任何表中的数据

- 预定义(固定)的数据库角色

本章小结

- 数据的安全性越来越重要
- **DBMS**必须具备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
  - 用户标识和鉴别
  - 存取控制技术
    - 自主存取控制
    - 强制存取控制
  - 视图技术
  - 数据加密
  - 审计技术
- 实际数据库系统**SQL Server**的安全控制



- 许多大型**DBMS**达到了**C2**级，其安全版本达到了**B1**级
  - **C2**级的**DBMS**必须具有自主存取控制功能和初步的审计功能
  - **B1**级的**DBMS**必须具有强制存取控制和增强的审计功能

思考题

- 教材P138
  习题1—9