

第四章 数据库安全性

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。
- 但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。

第四章 数据库安全性

- 什么是数据库的安全性

- 数据库的安全性是指保护数据库，防止因用户非法使用数据库造成数据泄露、更改或破坏。

- 什么是数据的保密

- 数据保密是指用户合法地访问到机密数据后能否对这些数据保密。
- 通过制订法律道德准则和政策法规来保证。

第四章 数据库安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 小结

第四章 安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 小结

4.1.1 数据库的不安全因素

- 对数据库安全性产生威胁的因素主要有以下几个方面
 - 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

4.1.2 安全标准简介

- 为降低进而消除对系统的安全攻击，各国引用或制定了一系列安全标准
 - TCSEC、TDI
 - CC

4.1.2 安全标准简介

- TCSEC标准的基本内容

- 安全策略
- 责任
- 保证
- 文档

TCSEC/TDI安全级别划分

- 四组(division)七个等级

- 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。

TCSEC/TDI安全级别划分

[illegible]

TCSEC/TDI安全级别划分

-  表示该级不提供对该指标的支持；
-  表示该级新增的对该指标的支持；
-  表示该级对该指标的支持与相邻低一级的等级一样；
-  表示该级对该指标的支持较下一级有所增加或改动。

TCSEC/TDI安全级别划分

- 目前许多大型DBMS 达到了C2级，其安全版本达到了B1
- 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

第四章 数据库安全性

4.1 计算机安全性概论

4.2 数据库安全性控制

4.3 视图机制

4.4 审计

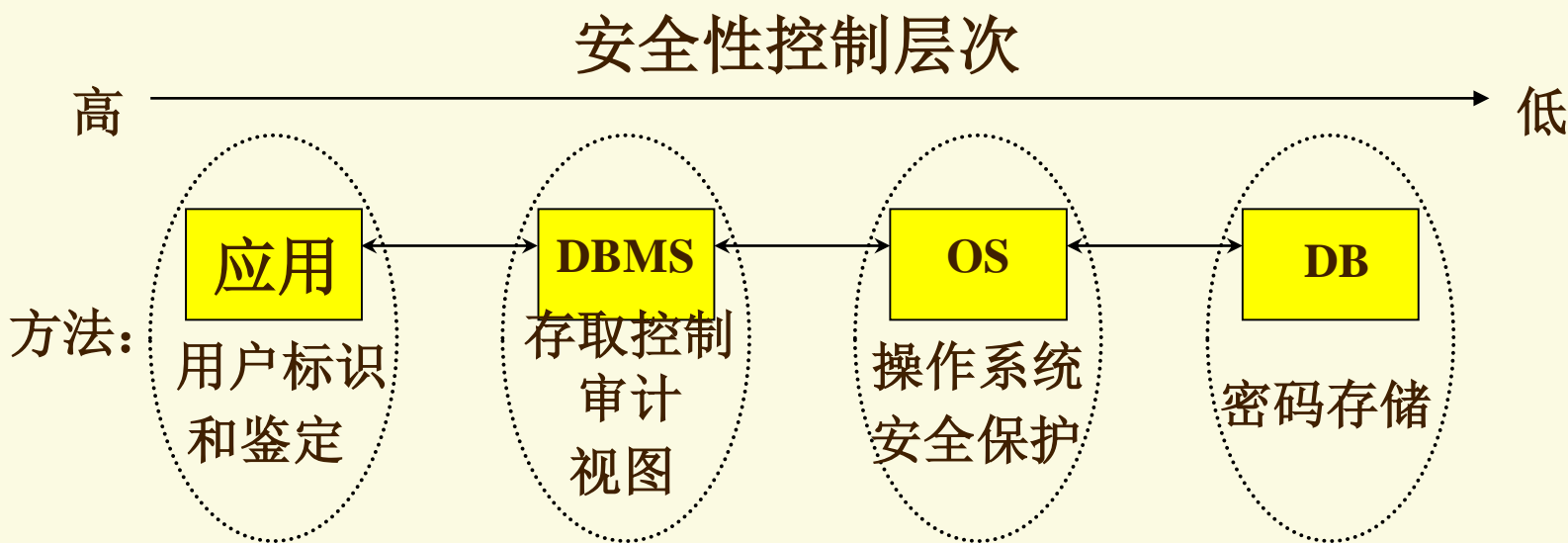
4.5 数据加密

4.6 其他安全性保护

4.7 小结

4.2 数据库安全性控制

● 计算机系统安全模型



4.2 数据库安全性控制

- 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 用户标识与鉴别

- 用户标识与鉴别
 - 系统提供的最外层安全保护措施
 - 每个用户在系统中都有一个用户标识。每个用户标识由用户名和用户标识号两部分组成。

4.2.1 用户标识与鉴别

● 基本方法

- 系统提供一定的方式让用户标识自己的名字或身份；
- 系统内部记录着所有合法用户的标识；
- 每次用户要求进入系统时，由系统核对用户提供的身份标识；
- 通过鉴定后才提供机器使用权。

4.2.1 用户标识与鉴别

- 用户身份鉴别的方法有很多种
 - 静态口令鉴别
 - 动态口令鉴别
 - 生物特征鉴别
 - 智能卡鉴别

4.2.2 存取控制

- 存取控制机制的组成

- 定义用户权限
- 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DBMS的安全子系统

4.2.2 存取控制

定义存取权限

- 在数据库系统中，为了保证用户只能访问他有权存取的数据，必须预先对每个用户定义存取权限。

检查存取权限

- 对于通过鉴定获得上机权的用户（即合法用户），系统根据他的存取权限定义对他的各种操作请求进行控制，确保他只执行合法操作。

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
- 强制存取控制（Mandatory Access Control，简称MAC）
 - B1级
 - 严格

4.2.2 存取控制

● 自主存取控制方法

- 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

4.2.2 存取控制

● 强制存取控制方法

- 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（DAC）方法

- 大型数据库管理系统几乎都支持自主存取控制，目前SQL标准也对自主存取控制提供支持，这主要通过SQL的GRANT语句和REVOKE语句来实现。
- 定义一个用户的存取权限就是定义这个用户可以在哪些数据库对象上进行哪些类型的操作。
- 在数据库系统中，定义存取权限称为授权。

4.2.3 自主存取控制（DAC）方法

- 用户权限由两个要素组成
 - 数据对象
 - 操作类型
- 存取控制的对象不仅有数据本身，还有数据库模式

表4.3 关系数据库系统中的存取权限

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, DELETE, REFERENCES, ALL PRIVILEGES

4.2.4 授权：授予与回收

- 授权 **GRANT**
- 回收 **REVOKE**
- 创建数据库模式的权限

一、GRANT

- 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];

- 功能：将对指定操作对象的指定操作权限授予指定的用户。

(1) 用户的权限

- 数据库的建立表（**CREATETAB**）的权限属于**DBA**，可由**DBA**授予普通用户，普通用户拥有此权限后可以建立基本表。
- 基本表或视图的属主拥有对该表或视图的一切操作权限。

(2) 接受权限的用户

- 一个或多个具体用户
- **PUBLIC**（全体用户）

(3) WITH GRANT OPTION子句

- 如果指定了**WITH GRANT OPTION**子句，则获得某种权限的用户还可以把这种权限再授予别的用户。
- 如果没有指定**WITH GRANT OPTION**子句，则获得某种权限的用户只能使用该权限，但不能传播该权限。

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 PRIVILEGES

ON TABLE Student

TO U2, U3;

GRANT ALL PRIVILEGES

ON TABLE 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;
```

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。

GRANT语句小结

- 一次向一个用户授权（例1）这是最简单的一种授权操作；
- 一次向多个用户授权（例2、例3）；
- 一次传播多个同类对象的权限(例2)；
- 一次可以完成对基本表、视图和属性列这些不同对象的授权（例4）；
- 授予关于DATABASE的权限必须与授予关于TABLE的权限分开。

二、REVOKE

- 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>]...;

- 功能：从指定用户那里收回对指定对象的指定权限。

例题

- 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;
```


例题（续）

- 例9 收回所有用户对表SC的查询权限。

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

例题（续）

- 例10 把用户U5对SC表的INSERT权限收回。

```
REVOKE INSERT  
ON TABLE SC  
FROM U5 CASCADE;
```

在 sql server中，收回使用 with grant option 授出的权限，必须使用 CASCADE 子句。

权限的级联回收

- 系统将收回直接或间接从U5处获得的对SC表的INSERT权限。

三、创建数据库模式的权限

- **GRANT**和**REVOKE**语句向用户授予或收回对数据的操作权限。
- 对数据库模式的授权则由**DBA**在创建用户时实现。
- **CREATE USER**语句一般格式如下：
CREATE USER <username>
[WITH] [DBA | RESOURCE | CONNECT]

4.2.5 数据库角色

- 数据库角色是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。
- 使用角色来管理数据库可以简化授权的过程。
- 在SQL语言中首先用**CREATE ROLE**语句创建角色，然后用**GRANT**语句给角色授权。

4.2.5 数据库角色

1. 角色的创建
2. 给角色授权
3. 将一个角色授予其他角色或用户
4. 角色权限的收回

1. 角色的创建

- **CREATE ROLE** <角色名>

- 刚刚创建的角色是空的，没有任何内容。可以用**GRANT**为角色授权。

2. 给角色授权

GRANT <权限>[,<权限>]...

ON <对象类型> 对象名

TO <角色> [,<角色>]...

3. 将一个角色授予其他角色或用户

GRANT <角色1> [<角色2>]...

TO <角色3> [<用户1>]...

[WITH ADMIN OPTION]

- 把角色授予某用户，或授予另一角色。
- 授予者或者是角色的创建者，或者拥有在这个角色上的ADMIN OPTION。

3. 将一个角色授予其他角色或用户

- 如果指定了 **WITH ADMIN OPTION**子句，则获得某种权限的角色还可以把这种权限再授予其他角色。
- 一个角色包含的权限包括直接授予这个角色的全部权限加上其他角色授予这个角色的全部权限。

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. 角色权限的收回

- [例11] 通过角色来实现将一组权限授予一个用户。

(1) 创建一个角色R1

```
CREATE ROLE R1;
```

(2) 给R1授权

```
GRANT SELECT,UPDATE,INSERT  
ON TABLE Student  
TO R1;
```

4. 角色权限的收回

(3) 将这个角色授予王平，张明，赵玲

GRANT R1

TO 王平，张明，赵玲;

exec sp_addrolemember 'r1','user_ebook'

(4) 通过R1回收王平的权限

REVOKE R1

FROM 王平

4. 角色权限的收回

- [例12] 角色的权限修改。

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

- [例13] 使R1减少对Student表的SELECT权限。

**REVOKE SELECT
ON TABLE Student
FROM R1**

4.2.6 强制存取控制（MAC）方法

- 自主存取控制能够通过授权机制有效地控制对敏感数据的存取。但是由于用户对数据的存取权限是“自主”的，用户可以自由地决定将数据的存取权限授予何人、决定是否也将“授权”的权限授予别人。这样，仍可能存在数据的“无意泄漏”。
- 造成这一问题的根本原因在于，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。

4.2.6 强制存取控制（MAC）方法

- 所谓MAC是指系统为保证更高层次的安全性，按照TDI/TCSEC标准中安全策略的要求，所采取的强制存取检查手段。它不是用户能直接感知或进行控制的。
- MAC适用于那些对数据有严格而固定密级分类的部门。

4.2.6 强制存取控制（MAC）方法

- 在MAC中，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，既包括实际用户，也包括代表用户的各进程。
 - 客体：系统中的被动实体，是受主体操纵的，包括文件、基本表、索引、视图等。

4.2.6 强制存取控制（MAC）方法

- DBMS对于主体和客体的每个实例（值）指派一个 **敏感度标记（Label）**。
- 敏感度标记被分成若干级别，例如
 - 绝密（Top Secret）
 - 机密（Secret）
 - 可信（Confidential）
 - 公开（Public）
- 主体的敏感度标记称为 **许可证级别**
- 客体的敏感度标记称为 **密级**

4.2.6 强制存取控制（MAC）方法

- 当某一用户以标记Label注册进入系统时，系统要求他对任何客体的存取必须遵循如下规则：
 - 仅当主体的许可证级别大于或等于客体的级别时，该主体才能读取相应的客体。
 - 仅当主体的许可证级别***等于客体的级别时，该主体才能写相应的客体。

4.2.6 强制存取控制（MAC）方法

- 强制存取控制（MAC）是对数据进行密级标记，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，从而更安全。
- 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。DAC和MAC共同构成DBMS的安全机制。

第四章 数据库安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 小结

4.3 视图机制

- 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，从而自动地对数据提供一定程度的安全保护。
- 视图机制更主要的功能在于提供数据独立性，其安全保护功能太不精细，往往远不能达到应用系统的要求。

4.3 视图机制

- 在实际应用中通常是视图机制与授权机制配合使用，首先用视图机制屏蔽掉一部分保密数据，然后在视图上面再进一步定义存取权限。
 - 这时视图机制实际上间接实现了支持存取谓词的用户权限定义

4.3 视图机制

- [例14] 建立视图，定义权限，授权

```
CREATE VIEW CS_Student  
AS
```

```
SELECT *
```

```
FROM Student
```

```
WHERE Sdept = 'CS';
```

```
GRANT SELECT  
ON CS_Student  
TO 王平;
```

```
GRANT INSERT, DELETE  
ON CS_Student  
TO 张明;
```


第四章 数据库安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 小结

4.4 审计 (Audit)

- 按照TDI/TCSEC标准中安全策略的要求，“审计”功能就是DBMS达到C2以上安全级别必不可少的一项指标。
- 审计功能把用户对数据库的所有操作自动记录下来放入审计日志 (Audit Log) 中。
- DBA可以利用审计跟踪的信息，重现导致数据库现有状况的一系列事件，找出非法存取数据的人、时间和内容等。

4.4 审计 (Audit)

- 审计功能的可选性

- 审计很费时间和空间，所以DBMS往往都将其作为可选特征
- 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。

4.4 审计 (Audit)

- 用户识别和鉴定、存取控制、视图等安全性措施均为强制性机制，将用户操作限制在规定的安全范围内。审计技术是**预防**手段，监测可能的不合法行为。
- 由于任何系统的安全性措施都不可能是完美无缺的，蓄意盗窃、破坏数据的人总是想方设法打破控制。所以，当数据相当敏感，或者对数据的处理极为重要时，就必须使用审计技术。

第四章 安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 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（明文）变换为不可直接识别的格式（密文）
- 不知道解密算法的人无法获知数据的内容

- 数据加密主要包括

- 存储加密
- 传输加密

第四章 安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 小结

4.6 其他安全性保护

- 推理控制
- 屏蔽信道
- 数据隐私保护

第四章 安全性

4.1 计算机安全性概论

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 其他安全性保护

4.7 小结

4.7 小结

- 技术和方法
 - 存取控制技术
 - 视图技术
 - 审计技术
 - 数据加密