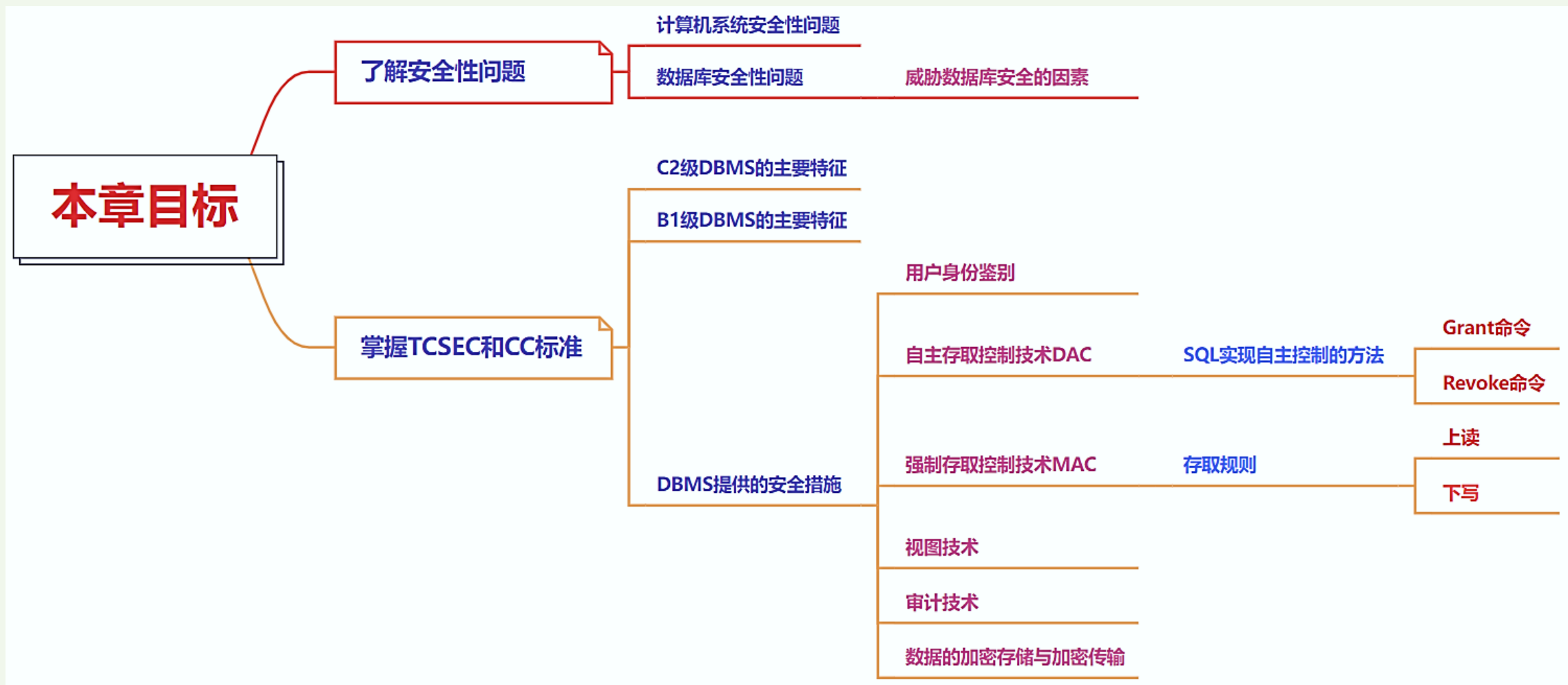


PRIDE AND PREJUDICE.

第4章 数据库的安全性



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






Blackmail



Cyberattack real-time map

安全问题无时无处不在!



- 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（如何定义安全性问题？）
- 数据库系统中的数据共享不是无条件的共享，数据库中数据的共享是在**DBMS**统一严格控制之下的共享，即只允许具有合法使用权限的用户访问允许他存取的数据
 - 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**
- 数据库的安全性和计算机系统的安全性紧密联系



- 所谓**计算机系统的安全性**，是指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**，**数据遭到破坏或泄露**等
- 计算机系统的安全性问题可分**技术安全类**、**管理安全类**和**政策法律类**三大类安全性问题

安全性类别	定义
技术安全性	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所存储数据的安全保护。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、不被破坏等。
管理安全性	指技术安全之外的，诸如软硬件意外故障（如自然灾害导致的故障）、场地的意外事故、管理不善导致的计算机设备和数据介质的物理破坏、丢失等安全问题。
政策法律类	指政府部门建立的有关计算机犯罪、数据安全保密的法律道德准则和政策法规、法令等。

- 本章讨论**数据库技术安全类问题**，即**从技术上如何保证数据库系统的安全**



- 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
- 数据成为一个部门、企业、乃至一个国家的重要资源，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
- 安全问题不是数据库系统所独有的，所有计算机系统都有这个问题
 - OS，网络系统，应用程序，硬件，系统架构，移动安全
- 本节主要内容：
 - 数据库的不安全因素
 - 安全标准简介



■ 1.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
- 有的黑客还故意锁定并修改数据，进行勒索和破坏等犯罪活动
- 必须阻止有损数据库安全的非法操作
- **DBMS**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



<https://www.cert.org.cn/>



<http://www.cverc.org.cn/>



■ 2.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**重要敏感的机密数据**，造成数据泄露
- **SQL注入**导致数据库被破坏的相关例子
 - 攻击者利用**SQL注入**技术，在入侵检测不严的情况下欺骗数据库服务器执行非授权的查询和操作，使得机密数据被泄露、篡改或锁定
 - **SQL注入技术**：在应用程序中事先定义好的查询语句的**结尾**添加额外的**SQL语句**

```
SELECT * FROM users WHERE username = '${username}' AND password = '${password}';
```



攻击者可在username或password字段输入以下内容

```
OR '1'='1'
```



原SQL查询变成以下语句：

```
SELECT * FROM users WHERE username = " OR '1'='1' AND password = '${password}';
```

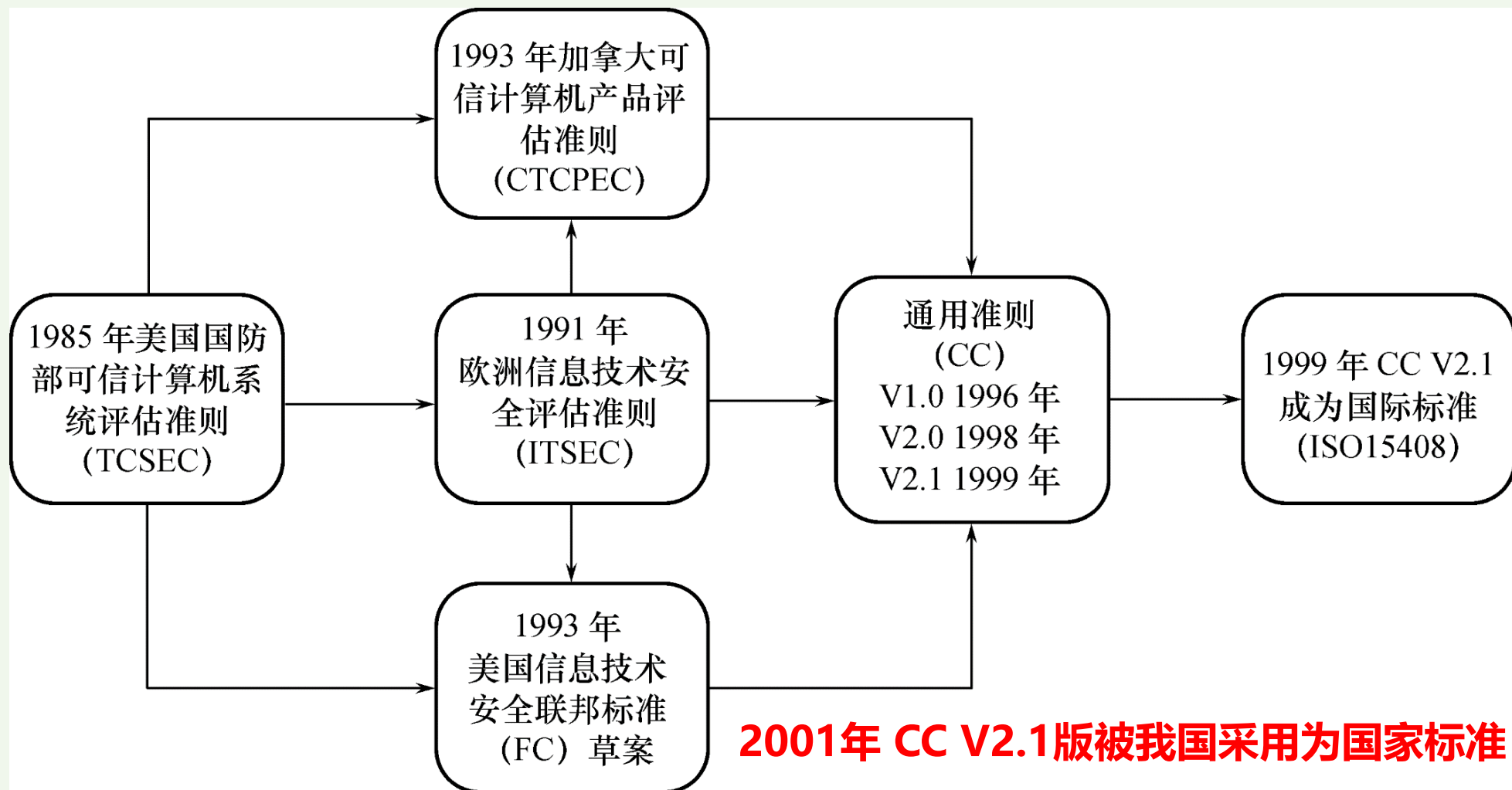
因'1'='1'总为真，所以攻击者无需知道用户名和密码即可登录



■ 3. 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





- 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》(TDI)
 - TDI又称紫皮书。它将TCSEC扩展到DBMS
 - TDI中定义了DBMS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

- 4组(division)7个等级
 - D
 - C (C1, C2)
 - B (B1, B2, B3)
 - A (A1)
- 系统可靠或可信程度逐渐增高

安全级别	安全指标
A1	验证设计(verified design)
B3	安全域(security domain)
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)



等级		特征
D		<ul style="list-style-type: none"> • 最低级别，将一切不符合更高标准的系统都归于D组，如DOS为D级的操作系统
C	C1	<ul style="list-style-type: none"> • 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
	C2	<ul style="list-style-type: none"> • 安全产品的最低档 • 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
B	B1	<ul style="list-style-type: none"> • 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 • 被认为是真正意义上的安全产品，多冠以“安全”或“可信的”产品
	B2	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 安全域 • 该级的可信计算基(trusted computing base, 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
A	A1	<ul style="list-style-type: none"> • 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, 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

表 4.2 不同安全级别对安全指标的支持情况

安全指标	安全策略							责任			保证								文档				
	自主存取控制	客体重用	标记完整性	标记信息的扩散	主体敏感度标记	设备标记	强制存取控制	标识与鉴别	可信路径	审计	系统体系结构	系统完整性	屏蔽信道分析	可信设施管理	可信恢复	可安全测试	设计规范和验证	配置管理	可信分配	安全特性用户指南	可信设施手册	测试文档	设计文档
C1																							
C2																							
B1																							
B2																							
B3																							
A1																							



- CC V2.1于1999年被ISO采用为国际标准
- 根据系统对安全保证要求的支持情况CC提出了评估保证级(Evaluation Assurance Level, EAL)
 - 从EAL1至EAL7共分为7级
 - 按保证程度逐渐增高

表4.3 CC评估保证级的划分及与TCSEC/TDI安全级别的对应

评估保证级	定义	TCSEC安全级别
EAL1	功能测试	
EAL2	结构测试	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



- 计算机系统中，安全措施是一级一级层层设置

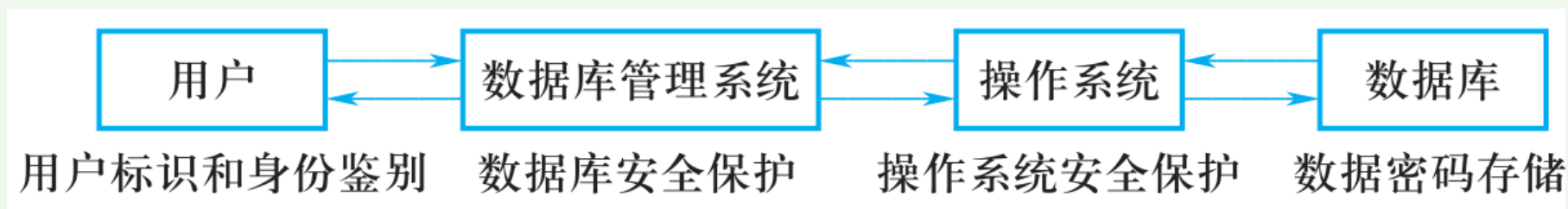


图4.1 计算机系统的安全模型

- 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
- DBMS要执行安全保护，包括：多种存取机制，目的是只允许用户执行合法操作
- 用户退出系统后根据情况进行安全审计
- 操作系统有自己的保护措施
- 对敏感数据在传输过程中要进行加密保护，最后还应以密码形式存储到数据库中



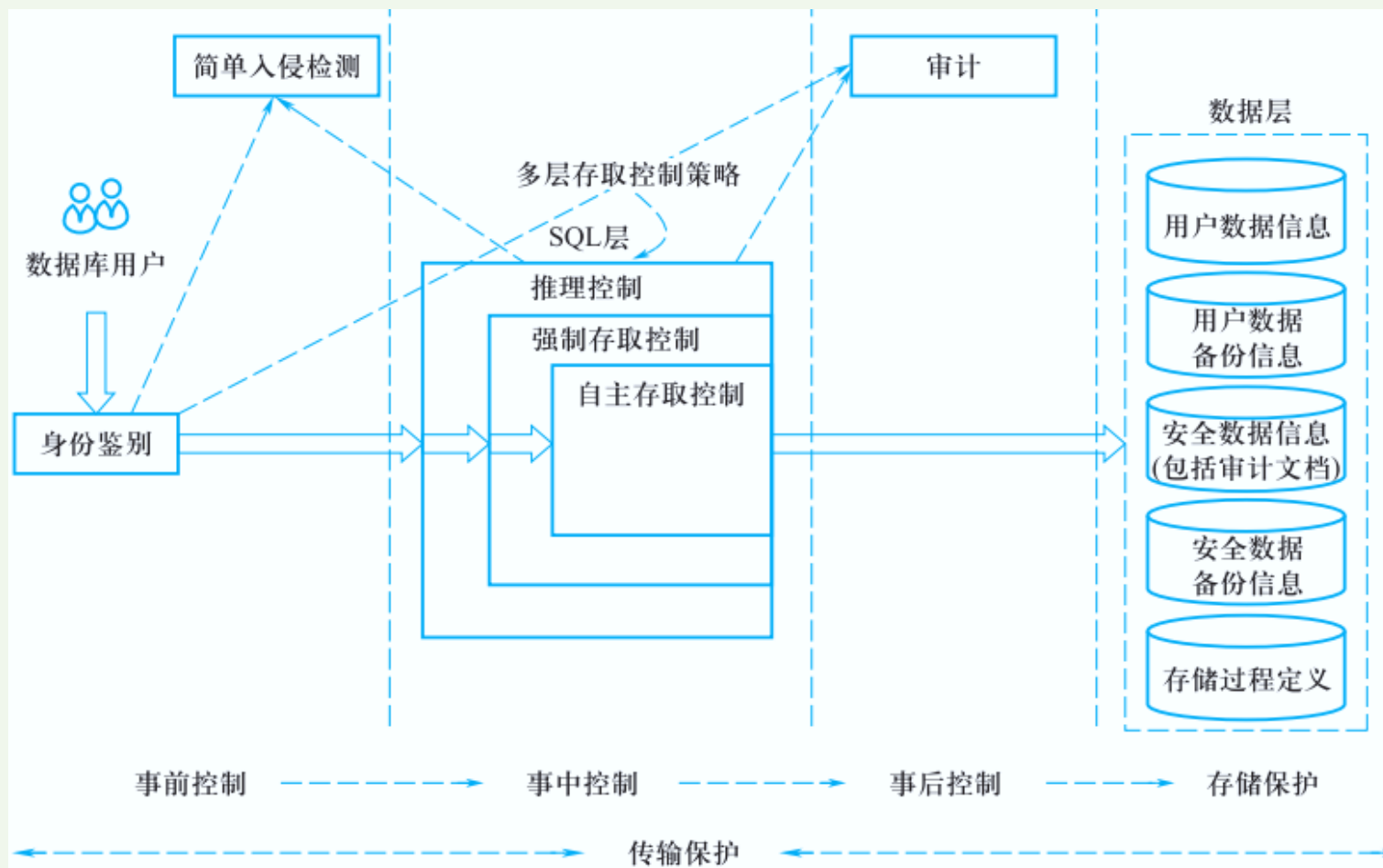


图4.2 DBMS的安全性控制模型

- ①事前控制：
通过身份鉴别和入侵检测共同实现
- ②事中控制：
在SQL处理层提供多层访问控制
- ③事后控制：
配置审计规则，对用户访问行为和系统关键操作进行审计
- ④存储保护：
在数据存储层，对敏感数据、重要的存储过程定义加密存储
- ⑤传输保护：
提供了传输加密功能

■ 数据库安全性控制技术

- 用户身份鉴别
- 存取控制
- 自主存取控制方法
- 授权与权限回收
- 数据库角色
- 强制存取控制方法



- **用户身份鉴别**是DBMS提供的最外层安全保护措施
- 每个用户在DBMS中都有一个**用户标识**
 - **用户标识**由**用户名+用户标识号(user identification number)**组成
 - **用户标识号**在系统整个生命周期内**唯一**
- 常用的用户身份鉴别方法：
 - **静态口令鉴别**
 - **动态口令鉴别**
 - **生物特征鉴别**
 - **智能卡鉴别**
 - **入侵检测**

- 数据库系统的存取控制机制用于确保只授权给有资格的用户访问数据库的权限，令所有未被授权的人员无法接近数据
- 存取控制机制主要包括定义用户权限和合法权限检查两部分
 - 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(privilege)
 - 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，根据安全规则/授权规则进行合法权限检查



- 用户权限定义和合法权限检查机制一起组成了DBMS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
 - 用户还可以将其拥有的存取权限转授给其他用户 (自主的含义)
 - 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

- 通过 SQL 的**GRANT**语句和**REVOKE**语句实现
- **用户权限**组成：数据库对象及其操纵类型
- **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**

表4.4 关系数据库系统中不同对象具有的主要存取权限

对象类型	对象	操 作 类 型
数据库和模式	数据库和模式	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



- 4.4.1授权
- 4.4.2权限回收
- 4.4.3SQL授权机制的特点



- SQL中使用**GRANT**和**REVOKE**语句向用户授予或收回对数据的操纵权限

- 授权语句格式:

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

谁能够发出GRANT语句?

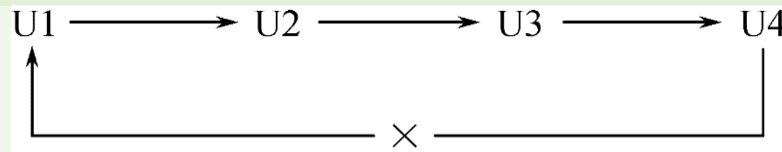
- DBA
- 数据库对象创建者(即属主owner);
- 拥有该权限的用户

接收权限的用户:

- 一个或多个具体用户;
- PUBLIC(即全体用户)

WITH GRANT OPTION子句

- 指定了该子句, 则获得权限的用户可以把这种权限再授予其他用户
- 没有指定, 则不能传播已获得的权限
- SQL标准规定**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表的全部操作权限授予用户U2和U3

GRANT ALL PRIVILIGES ON TABLE Student,Course TO U2,U3;

[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;

- 对属性列的授权时必须明确指出相应属性列名



[例4.5] 把对表SC的INSERT权限授予用户U5，并允许将此权限再授予其他用户

GRANT INSERT ON TABLE SC TO U5 WITH GRANT OPTION;

[例4.6] 用户U5将获得的操作权限授予用户U6

GRANT INSERT ON TABLE SC TO U6 WITH GRANT OPTION;

[例4.7] 用户U6再将该权限授予用户U7

GRANT INSERT ON TABLE SC TO U7 ;

- 但U7不能再传播此权限，即最大传播次数为3次



表4.5 执行了例4.1~4.7语句后学生选课数据库的用户权限定义

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



- 权限回收语句格式:

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**];

- 语义：由数据库管理员或其他授权者收回已授予的权限
- 指定了**CASCADE**，则级联收回已授予的权限；指定了**RESTRICT**，如果转授了权限，则不能收回
- 默认值为**RESTRICT**

[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权限收回时应使用**CASCADE**，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

表4.6 执行了例4.8~4.10语句后学生选课数据库的用户权限定义

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



■ SQL灵活的授权机制

– 数据库管理员

- 拥有所有对象的所有权限
- 根据实际情况将不同权限授予不同的用户

– 用户

- 拥有自己建立对象的全部操作权限
- 可以使用**GRANT**，把权限授予其他用户

– 被授权的用户

- 如果具有**WITH GRANT OPTION**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



■ 创建数据库的权限*

- SQL标准没有创建数据库的标准定义，故各具体RDBMS的实现可能各不相同
- 基本流程：**DBA创建新用户 > DBA授予用户登录RDBMS权限和用户创建数据库的权限 > 新用户登录并创建新数据库**
 - 可以参见openGauss用户管理的内容 ----实验中将会用到

权限与可执行的操作对照表

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





- 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
- 所有授予出去的权限在必要时又都可使用REVOKE语句收回



- 数据库角色(role)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
- 可以为一组具有相同权限的用户创建一个角色
- 可以简化授权的过程

- 角色创建的语句格式:

CREATE ROLE <角色名>; --新建角色是空的, 没有任何内容, 需要使用**grant**授权

- 给角色授权:

GRANT <权限>[,<权限>]... **ON** <对象类型>对象名 **TO** <角色>[,<角色>]...;



- 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];

- 角色权限的授权语句：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
- 一个角色的权限：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
(注意：若角色包含相同权限，这些权限不会因相同而被删除)



- 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**的执行者是

- 角色的创建者

- 拥有在这个（些）角色上的**WITH ADMIN OPTION**权限拥有者



[例4.14] 通过角色来实现将一组权限授予一个用户

– 实现步骤:

1.CREATE ROLE R1; --创建角色R1

2.GRANT SELECT, UPDATE, INSERT ON TABLE Student TO R1; --给R1授权

3.GRANT R1 TO 王平, 张明, 赵玲; --将R1授权给用户

4.REVOKE R1 FROM 王平; --回收用户从角色中获得的权限

[例4.15] 给角色增加新的权限

GRANT DELETE ON TABLE Student TO R1; --给角色R1增加权限DELETE

[例4.16] 减少角色的权限

REVOKE SELECT ON TABLE Student FROM R1; --给删除角色R1的权限SELECT

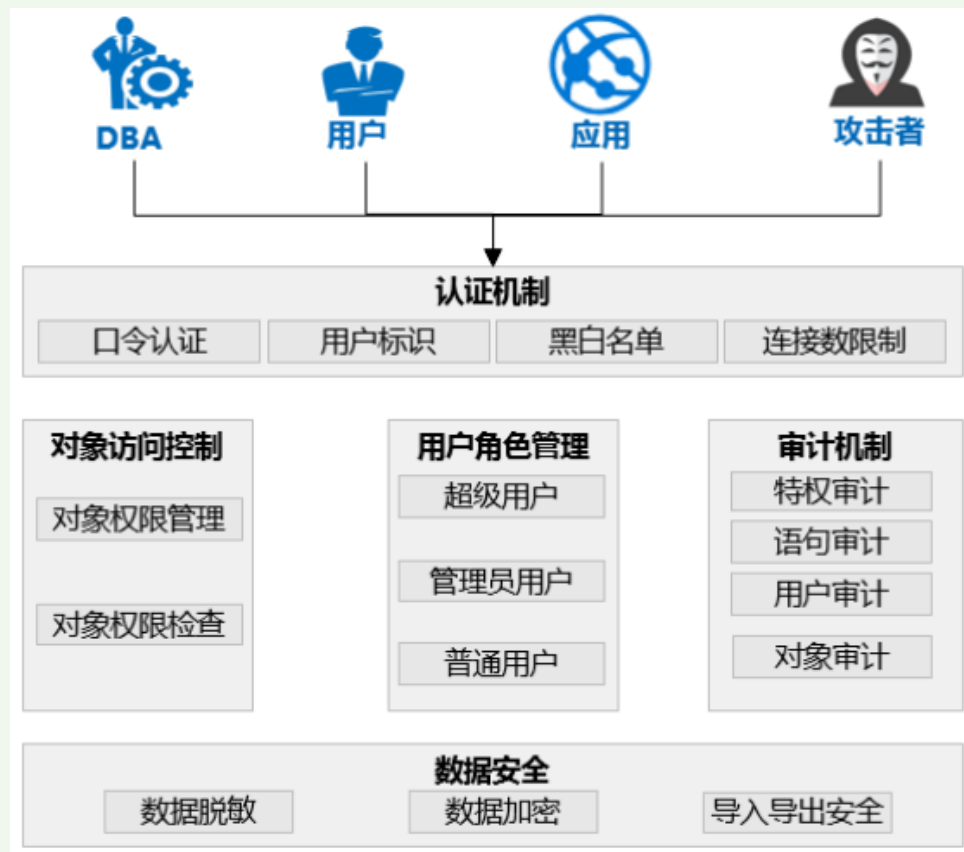


- 官网:

<https://www.opengauss.org/zh/docs/3.1.0/docs/CharacteristicDescription/%E6%95%B0%E6%8D%AE%E5%BA%93%E5%AE%89%E5%85%A8.html>

- 网络:

<https://baijiahao.baidu.com/s?id=1722802793849324928>



- openGauss支持Grant和Revoke命令实现权限的授予和回收
- 具体用法详见《openGauss开发者指南》16.13和16.14节

- **SELECT**

允许对指定的表、视图、序列执行SELECT命令，update或delete时也需要对应字段上的select权限。

- **INSERT**

允许对指定的表执行INSERT命令。

- **UPDATE**

允许对声明的表中任意字段执行UPDATE命令。通常，update命令也需要select权限来查询出哪些行需要更新。SELECT... FOR UPDATE和SELECT... FOR SHARE除了需要SELECT权限外，还需要UPDATE权限。

- **DELETE**

允许执行DELETE命令删除指定表中的数据。通常，delete命令也需要select权限来查询出哪些行需要删除。



■ 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**
- **解决思路**：对系统控制下的所有主客体实施强制存取控制策略

■ 强制存取控制(MAC)

- 指系统为保证更高层次的安全性，按照**TCSEC/TDI**标准中安全策略的要求所采取的强制存取检查手段
 - 保证更高层次的安全性
 - 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



- 在MAC中，DBMS所管理的全部实体被分为**主体**和**客体**两大类：

- **主体**：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**：系统中的被动实体，受主体操纵
 - 文件、基本表、索引、视图等

- **敏感度标记(Label)**

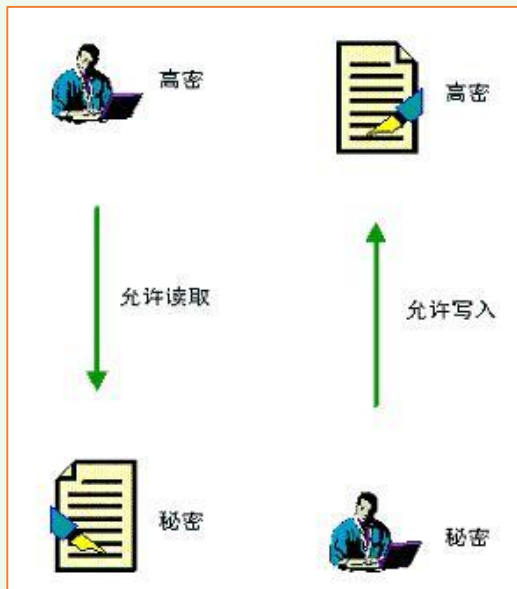
- 对于主体和客体，DBMS为它们每个实例(值)指派一个敏感度标记
- 敏感度标记分成若干级别：

- **绝密**(Top Secret, TS)
- **机密**(Secret, S)
- **可信**(Confidential, C)
- **公开**(Public, P)
- $TS \geq S \geq C \geq P$

- 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)



- **Bell-LaPadula(BLP)**保密性模型是第一个能够提供分级别数据机密性保障的安全策略模型，一般应用于军事用途
- **强制存取控制规则**(课本上介绍的即为**BLP**保密模型)
 - **上读**：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
 - **下写**：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

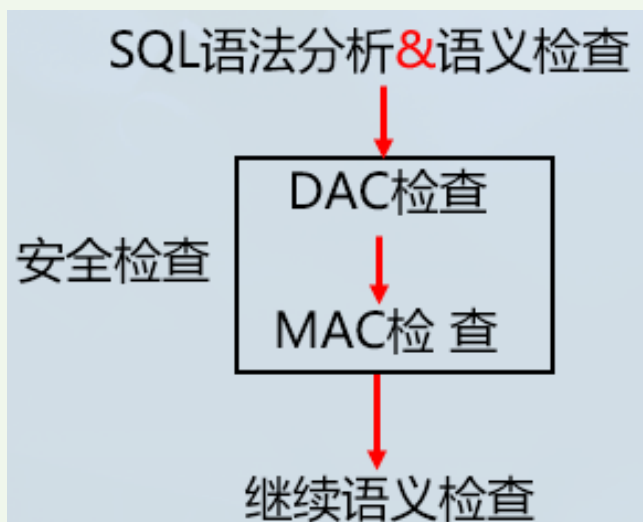


- 假如一个用户，他的安全级别为“高密”，想要访问安全级别为“秘密”的文档，他将能够成功读取该文件，但不能写入；而安全级别为“秘密”的用户访问安全级别为“高密”的文档，则会读取失败，但他能够写入。这样，文档的**保密性**就得到了保障。
- **两种规则的共同点**：它们均**禁止**了拥有高许可证级别的主体**更新**低密级的数据对象，从而防止了敏感数据的泄露。

参考：<https://blog.csdn.net/ajian005/article/details/8490082>



- **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
- **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
- **DAC与MAC共同构成DBMS的安全机制**



1. 先进行DAC检查
2. 通过DAC检查的允许存取的数据对象，再由系统自动进行MAC检查
3. 只有通过MAC检查的数据对象方可存取



- 通过视图机制，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
- 视图机制配合授权机制，间接地实现支持存取谓词的用户权限定义

[例4.17] 建立计算机科学与技术专业学生的视图，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

```
CREATE VIEW CS_Student    /*先建立视图CS_Student*/  
AS  
SELECT * FROM Student WHERE Smajor='计算机科学与技术'  
WITH CHECK OPTION      /*对该视图进行增删改时，必须满足Smajor='计算机科学与技术'的条件*/
```



GRANT SELECT ON CS_Student TO 王平; --王平只能检索计算机科学与技术学生的信息

GRANT ALL PRIVILEGES ON CS_Student TO 张明; --张明具有检索和增删改该视图的所有权限



■ 审计

- 启用一个专用的审计日志 (**Audit Log**)
- 把用户对数据库的所有操作自动记录放入审计日志
- 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



■ 1. 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



■ 2.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



■ 3.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

■ 用户级审计

- 任何用户可设置的审计，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

[例4.18] 对修改SC表结构或修改SC表数据的操作进行审计

SHOW AUDIT_TRAIL; --先显示当前审计开关状态

SET AUDIT_TRAIL TO ON; --打开审计开关

AUDIT ALTER, UPDATE ON SC BY ACCESS; --对SC表设置审计

[例4.19] 取消对SC表的ALTER和UPDATE操作审计

NOAUDIT ALTER,UPDATE ON SC;

- 审计设置以及审计日志一般都存储在数据字典中。必须把审计开关打开，才可以在系统表SYS_AUDITTRAIL中查看到审计信息



■ 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**) 经过一系列复杂计算, 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



■ 1.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
- 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，安全性较高

– 非透明存储加密

- 通过多个加密函数实现

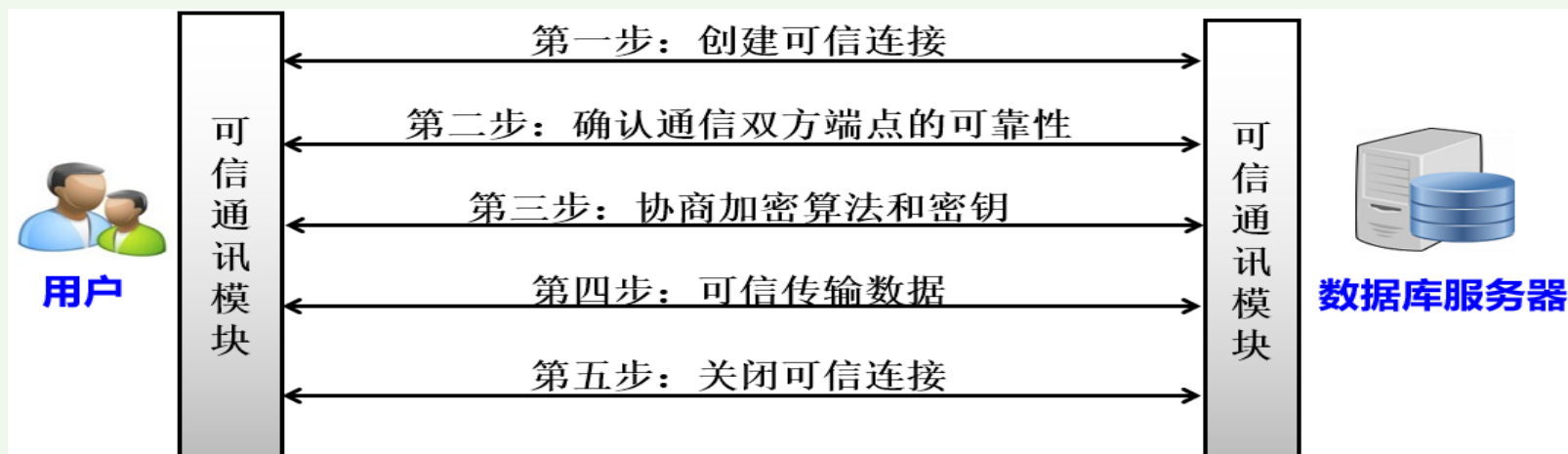
■ 2.传输加密

— 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
- 发送端和接收端需要密码设备，中间节点不需要密码设备
- 所需密码设备相对较少，容易被非法监听者获取敏感信息



基于安全套接层协议SSL传输方案的实现思路：

1. 确认通信双方端点的可靠性

- 采用基于数字证书的服务器和客户端认证方式
- 通信时均首先向对方提供己方证书，然后使用本地的CA 信任列表和证书撤销列表对接收到的对方证书进行验证

2. 协商加密算法和密钥

- 确认双方端点的可靠性后，通信双方协商本次会话的加密算法与密钥
- 利用公钥基础设施方式

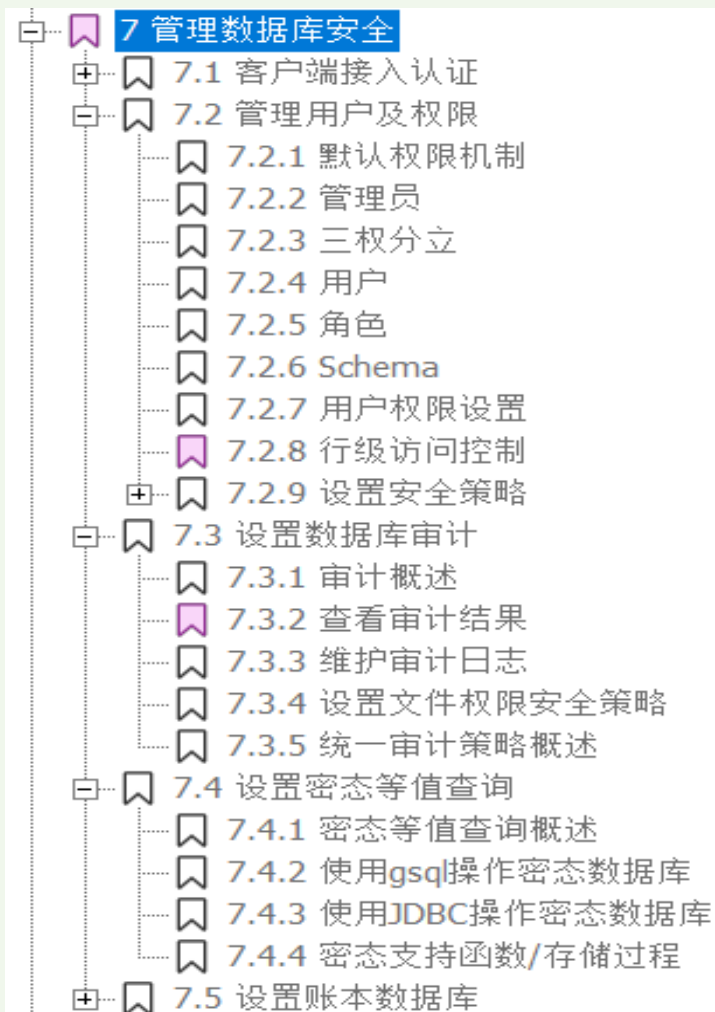
- 对称加密算法：AES
- 非对称加密算法：RSA

3. 可信数据传输

- 业务数据在被发送之前将被用某一组特定的密钥进行加密和消息摘要计算，以密文形式在网络上传输
- 当业务数据被接收的时候，需用相同一组特定的密钥进行解密和摘要计算
 - 特定的密钥是由通信双方磋商决定的，为且仅为双方共享
 - 会话密钥的生命周期仅限于本次通信



- openGauss管理数据库的安全管理详见《openGauss开发者指南》



■ 1.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

■ 2.隐蔽信道

- 处理强制存取控制未解决的问题

■ 3.数据隐私

- 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.“三权分立”的安全管理机制

- 解决数据库管理员权限过于集中的问题，遵照GB/T20273-2019，引进“三权分立”的安全管理机制
- 三权分立堵住了以前滥用数据库超级用户特权的安全漏洞，进一步提高了数据库的整体安全性



- **三权分立**是指将**原DBA**的管理权限分成三部分：**数据库管理员**，**安全管理员**，**审计管理员**

角色	职责
数据库管理员	<ul style="list-style-type: none">主要负责执行数据库日常管理的各种操作和自主访问控制
安全管理员	<ul style="list-style-type: none">主要负责强制访问规则的指定和管理，监督审计管理员和普通用户的操作，安全管理员自己不能操作和创建普通对象可以授权用户查看某些敏感数据（强制存取控制授权），但并不意味着这个用户可以看到这些敏感数据，他还需要得到数据库管理员的授权（自主存取控制授权）。同理，如果只有数据库管理员的自主存取控制授权而没有安全管理员的强制存取控制授权，用户还是不能看到这些敏感数据
审计管理员	<ul style="list-style-type: none">主要负责数据库的审计，监督数据库管理员和安全管理员的操作，自己不能创建和操作普通对象。审计管理员拥有一套机制，可以保护审计记录数据不会被数据库管理员或安全管理员删除或篡改

- 这三类用户彼此隔离，互不包容，各自维护自己权限许可范围内的对象，不能跨范围操作，也不能相互授权
- DBA不能对安全、审计相关的用户及数据库对象进行操作，不能将任何用户修改为安全员或审计员，不能授予、回收安全员、审计员的权限，不能切换到安全员、审计员的许可认证
- 安全员只能管理安全员和安全相关的系统对象
- 审计员只能管理审计员和审计相关的系统对象
- 这三类用户相互制约又相互协作共同完成数据库的安全管理工作



- 数据库应用的日益广泛，计算机网络的发展，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
 - 入侵检测
 - 存取控制技术：自主存取控制和强制存取控制
 - 视图技术
 - 审计技术
 - 数据存储加密和传输加密
- 大数据、云计算、移动互联、物联网、区块链等推动了数据库新技术的发展和应用
- 数据库安全性面临新的挑战
 - 如云数据存储服务安全可信技术
 - 分布数据访问控制技术
 - 隐私保护统计数据发布技术等



- 强制存取控制策略是TCSEC/TDI哪一级安全级别的特色()
A.C1 B. C2 C. B1 D. B2
- SQL的GRANT和REVOKE语句可以用来实现()
A.自主存取控制 B. 强制存取控制 C. 数据库角色创建 D. 数据库审计
- 在MAC机制中，当主体的许可证级别等于客体的密级时，主体可以对客体进行如下操作()
A.读取 B.写入 C.不可操作 D.读取、写入
- 在数据库的安全性控制中，授权的数据对象的____，授权子系统就越灵活
A. 范围越小 B. 约束越细致 C.范围越大 D.约束范围大

- 关于**DBMS**的安全机制，下列说法不正确的是_____。
 - A. 强制安全性机制是通过对数据和用户强制分类，从而使得不同类别用户能够访问不同级别的数据
 - B. 当有对**DB**访问操作时，任何人都被允许访问
 - C. 自主安全性是通过授权机制来实现的
 - D. 推断控制机制是防止通过历史信息或统计信息，推断出不该被其知道的信息，防止通过公开信息推断出私密信息
- 在对用户授予列**INSERT**权限时，一定要包含对_____的**INSERT**权限，否则用户的插入会因为空值而被拒绝。除了授权的列，其他列的值或者取_____, 或者为_____。
 - 主码; 空值, 默认值



- 教材第四章全部习题.
- 要求：作业在布置后一周内完成并提交到课程网站

