数据库安全

PART 01
数据库安全概述

PART 02
自主访问控制（DAC）

PART 03
基于角色的访问控制（RBAC）

PART 04
强制访问控制（MAC）

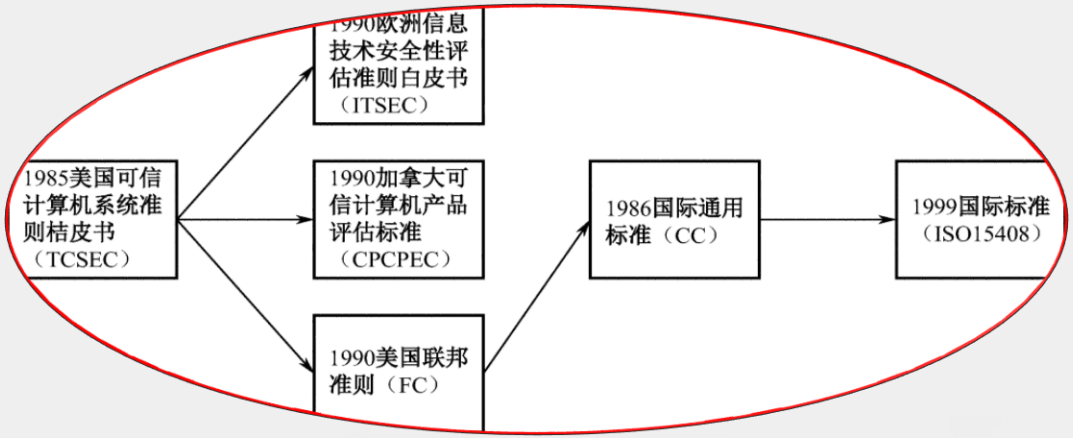
PART 05
审计和其他安全机制

- 数据库的**无意破坏**方式
  - 并发存取所引起的数据异常；
  - 数据的分布存储造成的不一致；
  - 逻辑错误造成更新事务未遵守保持数据一致的原则；
  - 事务处理过程中系统崩溃。

对策：
完整性约束控制技术
并发控制技术
数据库恢复技术

- 数据库的**恶意破坏**方式
  - 未经授权的读取数据；
  - 未经授权的修改数据；
  - 未经授权的破坏数据。

对策：数据库访问控制、身份鉴定、安全审计、数据加密等



- **TCSEC，也称为桔皮书**

  1985年美国国防部（DoD）颁布的《DoD可信计算机系统评估标准》

- **TDI，也称为紫皮书**

  1991年美国国家计算机安全中心（NCSC）颁布的《可信计算机系统评估标准关于可信数据库系统的解释》，TDI将TCSEC扩展到数据库管理系统。

- **ITSEC，欧洲白皮书**

  法、英、荷、德欧洲4国在20世纪90年代初联合发布信息技术安全评估标准。该标准将安全概念分为功能与评估两部分，首次提出了信息安全的机密性、完整性、可用性的概念。

- **CC—>ISO15408**

  信息技术安全评价的通用标准（CC）是由6个国家（美、加、英、法、德、荷）于1996年联合提出的，并逐渐形成国际标准ISO15408。CC标准是第一个信息技术安全评价国际标准，定义了评价信息技术产品和系统安全性的基本准则。

- **GB/T 18336－2001**

  我国将ISO/IEC 15408转化为国家标准——GB/T 18336－2001《信息技术安全性评估准则》，并直接应用于我国的信息安全测评认证工作。基础性等级划分标准GB17859－1999是信息系统安全等级保护实施指南。

- TCSEC/TDI安全级别划分
  - 四组（DCBA）七个等级。
  - 对用户登录、授权管理、访问控制、审计跟踪、隐蔽通道分析、可信通道建立、安全检测、生命周期保障、文档写作、用户指南等内容提出了规范性要求。
  - 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）

- **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专门机制来保障
- **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。
- **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 NT 3.5

    数据库：Oracle公司的Oracle 7，Sybase公司的 SQL Server 11.0.6

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）

- **B1级**
  - 标记安全保护。“安全”(Security)或“可信的”(Trusted)产品。
  - 对系统数据加以标记，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，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

- **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，在数据库方面暂时没有此级别的产品。
  - 典型例子

    操作系统：Trusted Information Systems公司的Trusted XENIX一种产品

    网络产品：Cryptek Secure Communications公司的LLC VSLAN一种产品

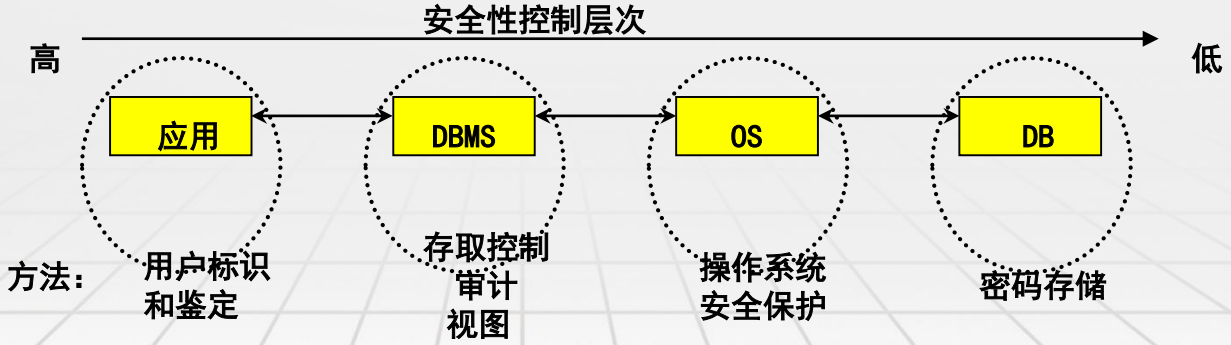
    数据库：没有符合B2标准的产品

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）

- **B3级**
  - 安全域保护
  - 该级的可信任运算基础（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。
- **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。
- **总结**
  - 支持自主存取控制（DAC）的DBMS属于C1级；
  - 支持审计功能的DBMS属于C2级；
  - 支持强制存取控制（MAC）的DBMS则可以达到B1级。
  - B2以上的系统标准更多地还处于理论研究阶段，产品化以至商品化的程度都不高，其应用也多限于一些特殊的部门如军队等。

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）

- 数据库系统的安全框架
  - 网络系统层次
    - 物理层面上避免入侵者进行物理破坏；
    - 网络层面上采用防火墙、入侵检测技术等手段阻止外部入侵。
  - 操作系统层次
    - 操作系统安全策略、安全管理策略和数据安全。
  - 数据库系统层次
    - 用户标识和鉴别、存取控制、数据分级、视图机制、审计、数据加密。



自主访问控制DAC是对用户访问数据库中各种资源（表、视图、程序等）的权限（创建、查询、更新等）的控制。

- **控制方式是自主的**
  - 由客体的属主对自己的客体进行管理
  - 由属主自己决定是否将自己的客体访问权或部分访问权授予其他主体
  - 在自主访问控制下，用户可以按自己的意愿，有选择地与其他用户共享他的文件。
- **可以在主体之间相互转让权限的访问控制**
  - 权限是指允许某个用户以某种方式访问一些数据对象；
  - 对用户访问数据库中各种资源（包括表、视图、程序等）的权利（包括创建、查询、更新、执行等）的控制；
  - 一个用户建立了一个数据对象（如表、视图）就自动具有了对这个数据对象的所有权利。
  - 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。
- **C2级，灵活**

- 具有**CONNECT特权**的用户
  - 可以与数据库连接，能根据授权进行数据库中数据的查询、更新，能创建视图。
- 具有**RESOURCE特权**的用户
  - 除具有CONNECT特权外，还能创建表、索引，修改表结构，能将自己创建的数据对象的访问权授予其他用户或从其他用户那儿收回，对自己创建的数据对象能进行跟踪审查。
- 具有**DBA特权**的用户
  - 能进行所有的数据库操作。
  - 特权不能任意扩散。

- **主体 S（Subject）**
  - 提出访问资源具体请求，是某一操作动作的发起者，但不一定是动作的执行者；
  - 可能是某一用户，也可以是用户启动的进程、服务和设备等。
- **客体 O（Object）**
  - 被访问资源的实体。所有可以被操作的信息、资源、对象都可以是客体；
  - 客体可以是信息、文件、记录等集合体，也可以是网络上硬件设施、无线通信中的终端，甚至可以包含另外一个客体。
- **控制策略 A（Attribution）**
  - 主体对客体的相关访问规则集合，即属性集合。
  - 访问策略体现了一种授权行为，也是客体对主体某些操作行为的默认。

- **访问数据的权限**
  - SELECT（读取权限）：允许读数据，但不能修改数据。

    例：SELECT（Pname，Paddr）表示只授予用户查询关系表中Pname，Paddr 两个属性中数据的权限，关系表中其他属性的数据对用户是屏蔽的。
  - INSERT（插入权限）：允许插入一条新的数据，但不能修改已有数据。
  - UPDATE（修改权限）：允许修改数据，但不能删除数据。
  - DELETE（删除权限）：允许删除数据。
- **修改数据库模式（SQL92标准）的权限**
  - Index（索引权限）：允许建立或删除索引。
  - Create（创建权限）：允许建立新的关系表。
  - Alter（修改权限）：允许对关系表中的属性进行增加、删除。
  - Drop（删除权限）：允许删除关系表。

- **其它权限**
  - REFERENCE权限：允许用户在建立关系的完整性约束中引用一个参照关系
  - USAGE权限：授权用户使用一个指定的域
  - TRIGGER权限：授权用户定义关系表中触发器的权利
  - EXECUTE权限：授予用户执行一个函数或过程的权利
  - UNDER权限：授权用户建立一给定类的子类

- 权限示例

```
INSERT INTO RecipeDetail(Mno)
SELECT Mno FROM  Medicine
WHERE Mname LIKE '%替硝唑%'
```

- 需要一个RecipeDetail 表上的INSERT权限，该权限可以仅仅是RecipeDetail表上的INSERT（Mno）权限
- 需要一个对Medicine关系表的SELECT授权，这个授权也可以只针对Medicine中的Mno、Mname属性

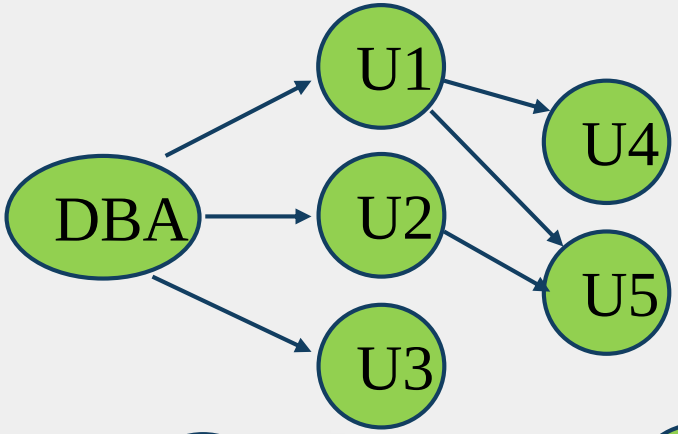
- 授权就是赋予用户一定的操作数据对象的权利。
- 授权可以由DBA授予，也可以由数据对象的创建者授予。
- 授权格式：

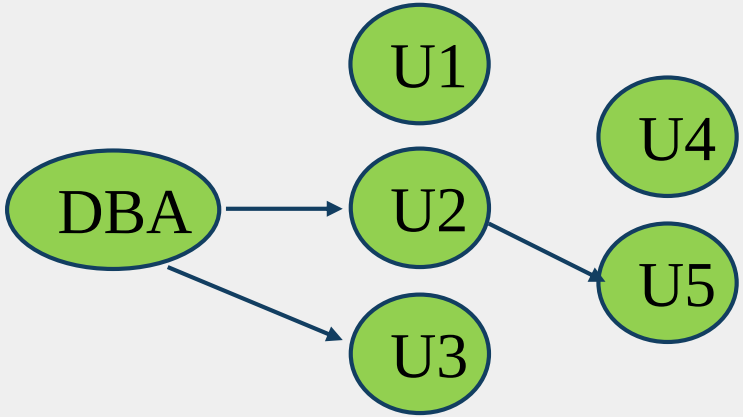
```
GRANT {all privileges|privilege{, privilege…,}}
ON [TABLE] tablename|viewname
TO [PUBLIC|user_name{,user_name…}]
[WITH GRANT OPTION]
```

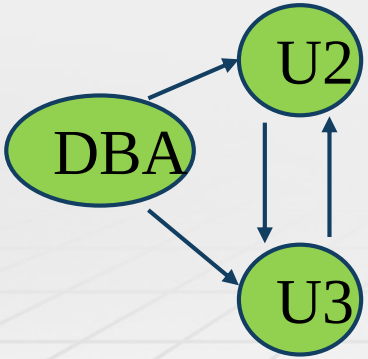
  - ALL PRIVILEGES是所有权限的总称
  - 数据对象可以是基本表，也可以是视图
  - 用户名可以代表单一用户也可以代表一组用户，当代表一组用户时我们称为角色。PUBLIC是所有数据库用户的总称；
  - WITH GRANT OPTION，受权者可以将此权限转授给其他用户；
  - 一个用户如果是表的创建者，他就自动拥有了对所创建表的所有权利以及将该表权利授予其他用户的权利，而且不能取消。

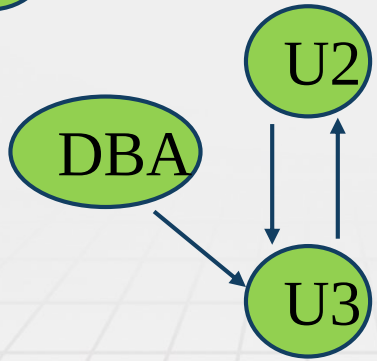
权限图

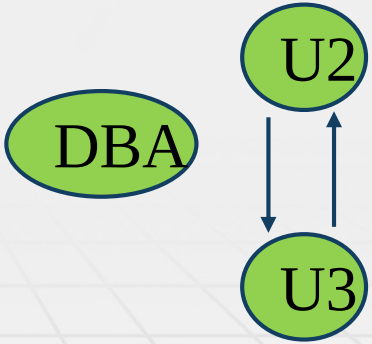
- 结点是用户，根结点是DBA，有向边Ui→Uj，表示用户Ui把某权限授给用户Uj。
- 一个用户拥有权限的充分必要条件是在权限图中有一条从根结点到该用户结点的路径。

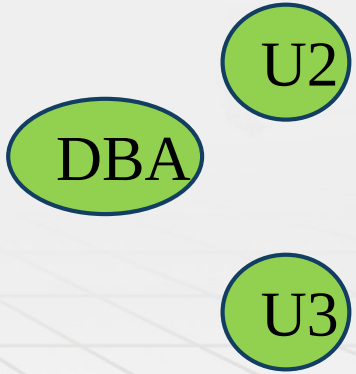

U1
U4
DBA
U2
U5
U3


U1
U4
DBA
U2
U5
U3


U2
DBA
U3


U2
DBA
U3


U2
DBA
U3


U2
DBA
U3

- 授权示例：假定用户WangPing创建了表RecipeDetail, Medicine, RecipeMaster,并且WangPing执行如下授权命令。

```
GRANT SELECT ON RecipeDetail TO LiXia;
GRANT SELECT ON RecipeMaster TO LiXia WITH GRANT OPTION;
GRANT UPDATE (Mprice) ON Medicine TO WangHao;
GRANT REFERENCE (Mno) ON Medicine TO ZhangYang;
GRANT INSERT, DELETE ON RecipeDetail TO MengFan WITH GRANT OPTION;
```

  - LiXia能够对RecipeDetail和RecipeMaster执行查询语句，并能将RecipeMaster的查询权限授予DengTian:

```
GRANT SELECT ON RecipeMaster TO DengTian;
```

  - WangHao只能修改Medicine表中的Mprice列的值。

- 收回权限格式

```
REVOKE [WITH GRANT OPTION FOR]{ALL PRIVILEGES|privilege{, Privilege…,}}
ON [TABLE] tablename|viewname
FROM [PUBLIC|user_name{,user_name…}]
[RESTRICT|CASCADE]
```

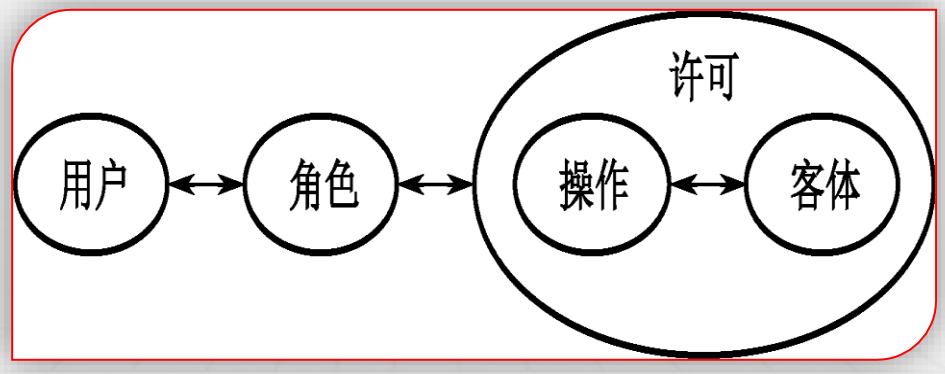
- 示例：若WangPing在授权后，发现用户的权限分配不恰当，就可以执行如下命令收回部分用户的操作权限：

```
REVOKE SELECT ON RecipeDetail FROM LiXia;
REVOKE UPDATE（Mprice）ON Medicine FROM WangHao;
REVOKE GRANT OPTION FOR SELECT ON RecipeMaster FROM LiXia;
```

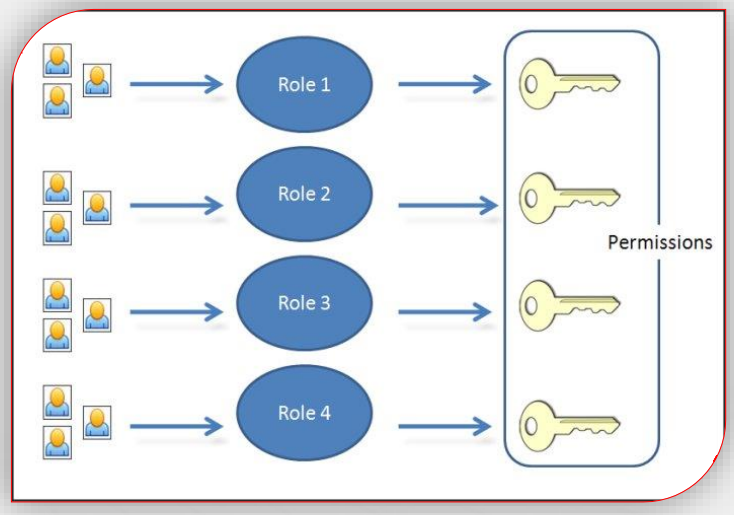
- RESTRICT与CASCADE
  - 从一个用户那里收回权限可能导致其他用户也失去该权限。这一行为称为级联回收CASCADE。在大多数数据库系统中，级联回收是默认行为。
  - 可以指定RESTRICT方式：REVOKE SELECT ON RecipeMaster FROM LiXia RESTRICT。

- 授权粒度：指可以定义的数据对象的范围
  - 它是衡量授权机制是否灵活的一个重要指标
  - 授权定义中数据对象的粒度越细，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会相应增大
  - 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
- 关系数据库中授权的数据对象粒度
  - 数据库
  - 表
  - 属性列
  - 行

- RBAC方法
  - 根据管理中相对稳定的职权和责任来划分角色
  - 将访问许可权分配给一定的角色
  - 用户通过饰演不同的角色获得访问许可权。



- 角色的作用
  - 角色可以看作是一组操作的集合，不同的角色具有不同的操作集。
  - 角色是访问控制中访问主体和受控对象之间的一座桥梁（**授权模板**）。
- 角色与用户关系
  - 一个用户可经授权而拥有多个角色，一个角色可有多个用户组成
- 角色与许可关系
  - 每个角色拥有多种许可，每个许可也可以授权给多个不同的角色，每个操作可施加与多个客体，每个客体可接受多个操作。



- 创建角色

```
CRETAE ROLE Admin;
```

- 角色授权

```
GRANT SELECT ON RecipeMaster TO Admin;
```

- 角色授予用户或其他角色

```
GRANT Admin TO LiXia;
CREATE ROLE Manager;
GRANT Admin to Manager;
GRANT Manager TO WangHao;
```

角色Manager除具有直接赋予它的权限外，还继承了角色Admin具有的权限。

- 强制访问控制：Mandatory Access Control
  - 目标是限制主体或发起者访问对象或目标执行某种操作的能力。
  - 主体通常是一个进程或线程
  - 对象可能是表、视图、索引、过程等
  - 主体的敏感标记称为许可证级别（Clearance Level），客体的敏感标记称为密级（Classification Level）
  - 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。
  - B1级，严格

- **保密性规则**　　例如：BLP模型
  - 仅当主体的许可证级别高于或者等于客体的密级时（Read, Lsubject≥Lobject），该主体才能读取相应的客体。（下读，RD）
  - 仅当主体的许可证级别低于或者等于客体的密级时（Write, Lsubject≤Lobject），该主体才能写相应的客体。（上写，WU）
- **完整性规则**　　例如：Biba模型
  - 仅当主体的许可证级别低于或者等于客体的密级时（Read, Lsubject≤Lobject），该主体才能读取相应的客体。（上读，RU）
  - 仅当主体的许可证级别高于或者等于客体的密级时（Write, Lsubject≥Lobject），该主体才能写相应的客体。（下写，WD）

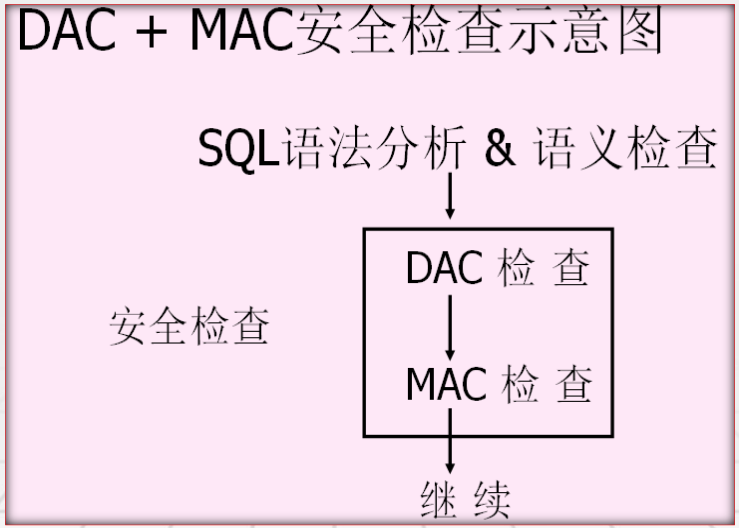
- MAC例子

假定系统设置有4个等级：top secret(TS), secret(S), confidential(C), unclassified(U)。

若用户LiXia和WangHao的许可证级别分别是C和S，则用户LiXia看不到表中的任何记录，而WangHao可以看到其中的一条记录。

Rno	Pno	Dno	Pgno	Rdatetime	Security class
1282317	481	140	1645	2007-07-21 13:12:01	S
1282872	201	21	2170	2007-07-22 10:10:03	TS

- MAC与DAC结合
  - DAC与MAC共同构成DBMS的安全机制
  - 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。



- **安全审计（Audit）作用——跑不了**
  - 跟踪审计是一种监视措施，记录了用户对数据库的所有操作。一旦发现问题，系统可自动报警，或根据数据进行事后的分析和调查。
- **C2以上安全级别的DBMS必须具有审计功能**
- **安全审计内容**
  - 安全审计数据产生、安全审计自动响应、安全审计分析、安全审计浏览、安全审计事件选择和安全审计事件存储。
- **跟踪审计的结果记录**
  - 操作类型、操作终端标识与操作者标识、操作日期和时间、涉及的数据、数据的前像后像。
- **审计负荷**
  - 审计通常是很费时间和空间的，所以DBMS往往都将其作为可选特征，允许DBA根据应用对安全性的要求，灵活地打开或关闭审计功能。

- **ORACLE审计操作**
  - 提供AUDIT语句设置审计功能，NOAUDIT语句取消审计功能。
  - 通过DBA_AUDIT_TRAIL视图可以查询审计结果。

- **审计操作类型**
  - 登录审计
  - 语句审计
  - 对象审计

- **审计示例：跟踪用户scott的表RecipeMaster上的所有更新操作**

```
SQL> AUDIT UPDATE on scott.RecipeMaster BY ACCESS;
SQL> NOAUDIT ALL ON RecipeMaster;
```

- 可以为不同的用户定义不同的视图，把数据对象限制在一定的范围之内。
- 通过视图机制把要保密的数据对无权存取的用户隐藏起来，从而自动地对数据提供一定程度的安全保护。
- 必须得到一个视图上的访问权限
- 示例：

```
CREATE VIEW Diagnosis-101 AS
  (SELECT * FROM Diagnosis WHERE Dno IN
     (SELECT Dno FROM Doctor WHERE Ddeptno=’101’))
```

通过将Diagnosis-101上的访问权限赋予某工作人员，他就能查询本科室医生的出诊记录了。

- 最外层安全保护措施，由系统提供一定的方式让用户标识自己的身份或名字，通过鉴定后才能提供机器使用权。
- 用户标识（User Identification）

  用一个用户名或者用户标识号（UID）来表明用户身份。系统内部记录着所有合法用户的标识，系统鉴别此用户是否是合法用户。
- 口令（Password）：系统核对口令以鉴别用户身份。
- 常用的鉴别用户身份的方法
  - 询问-应答系统：类似于地下工作者对暗语的办法
  - 物品鉴别：钥匙、磁卡都可以作为用户的身份凭证
  - 用户个人特征鉴别：签名、指纹、声音都是用户个人特征

- 加密基本思想
  - 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
  - 替换方法：使用密钥（Encryption Key）将明文中的每一个字符转换为密文中的一个字符
  - 置换方法：将明文的字符按不同的顺序重新排列
  - 混合方法：美国1977年制定的官方加密标准：数据加密标准（Data Encryption Standard，简称DES）
- 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；而有些没有，但提供了接口
  - 数据加密功能通常也作为可选特征，允许用户自由选择
    - 数据加密与解密是比较费时的操作
    - 数据加密与解密程序会占用大量系统资源
    - 应该只对高度机密的数据加密

- 数字签名用来验证数据的真实性
  - 用户使用私钥产生签名后的数据，数据公开后所有的人都可以用公钥来验证数据的创建者是谁。

- 数字签名主要过程
  - 信息发送者使用一单向散列函数（HASH函数）对信息生成信息摘要。
  - 信息发送者使用自己的私钥签名信息摘要。
  - 信息发送者把信息本身和已签名的信息摘要一起发送出去。
  - 信息接收者通过使用与信息发送者使用的同一个单向散列函数（HASH函数）对接收的信息本身生成新的信息摘要，再使用信息发送者的公钥对信息摘要进行验证，以确认信息发送者的身份和信息是否被修改过。

- 认证技术特征
  - 认证技术主要解决网络通信过程中通信双方的身份认可。
  - 认证的过程涉及加密和密钥交换。
  - 加密可使用对称加密、不对称加密及两种加密方法的混合方法。

- 公开密钥体系（Public Key Infrastructure，PKI）的构成
  - 认证机构（CA）
  - 数字证书库
  - 密钥备份及恢复系统
  - 证书作废系统
  - 应用接口（API）