

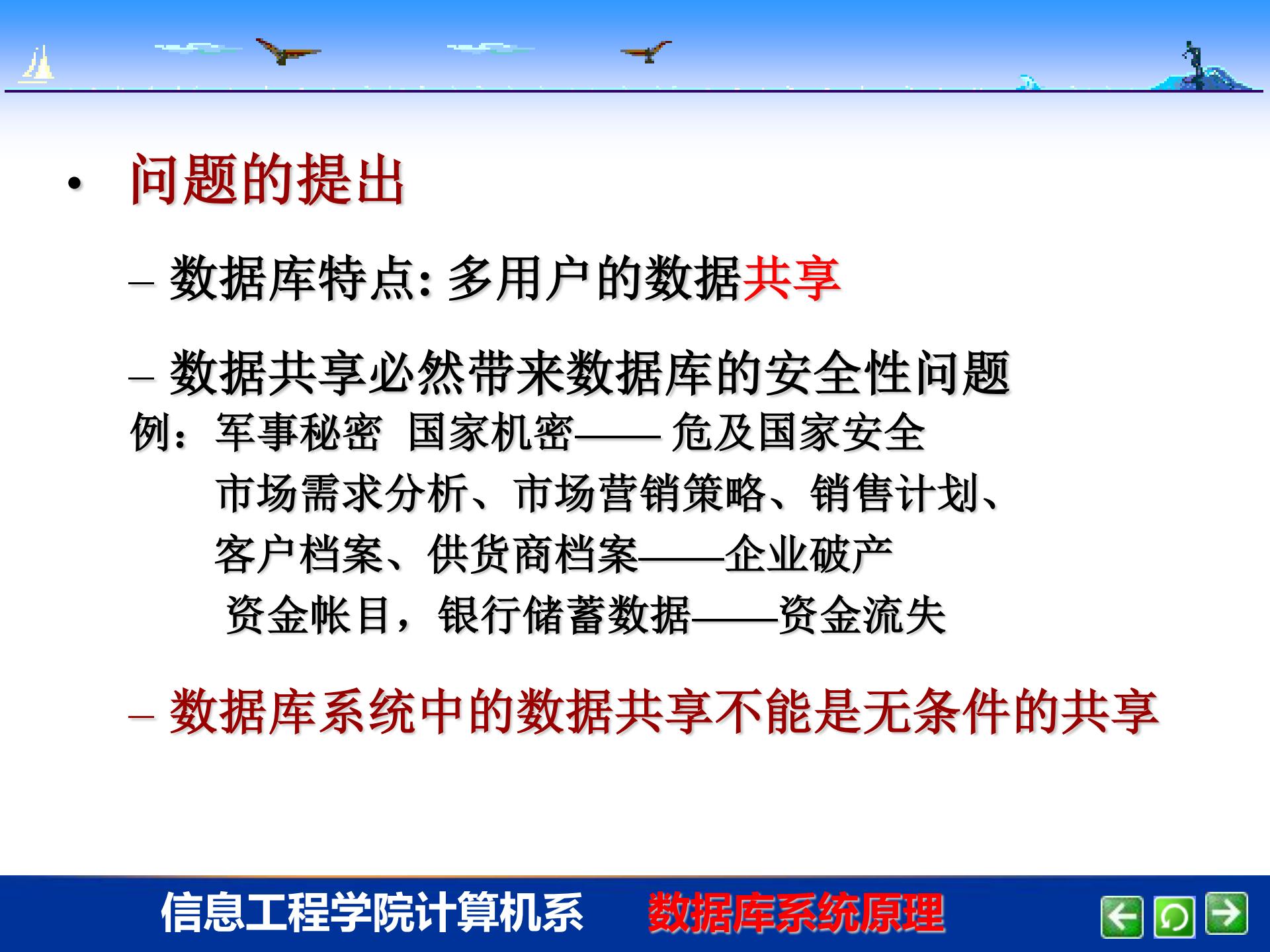


第四章 数据库安全性

4.1 计算机安全性问题

4.2 数据库安全性控制

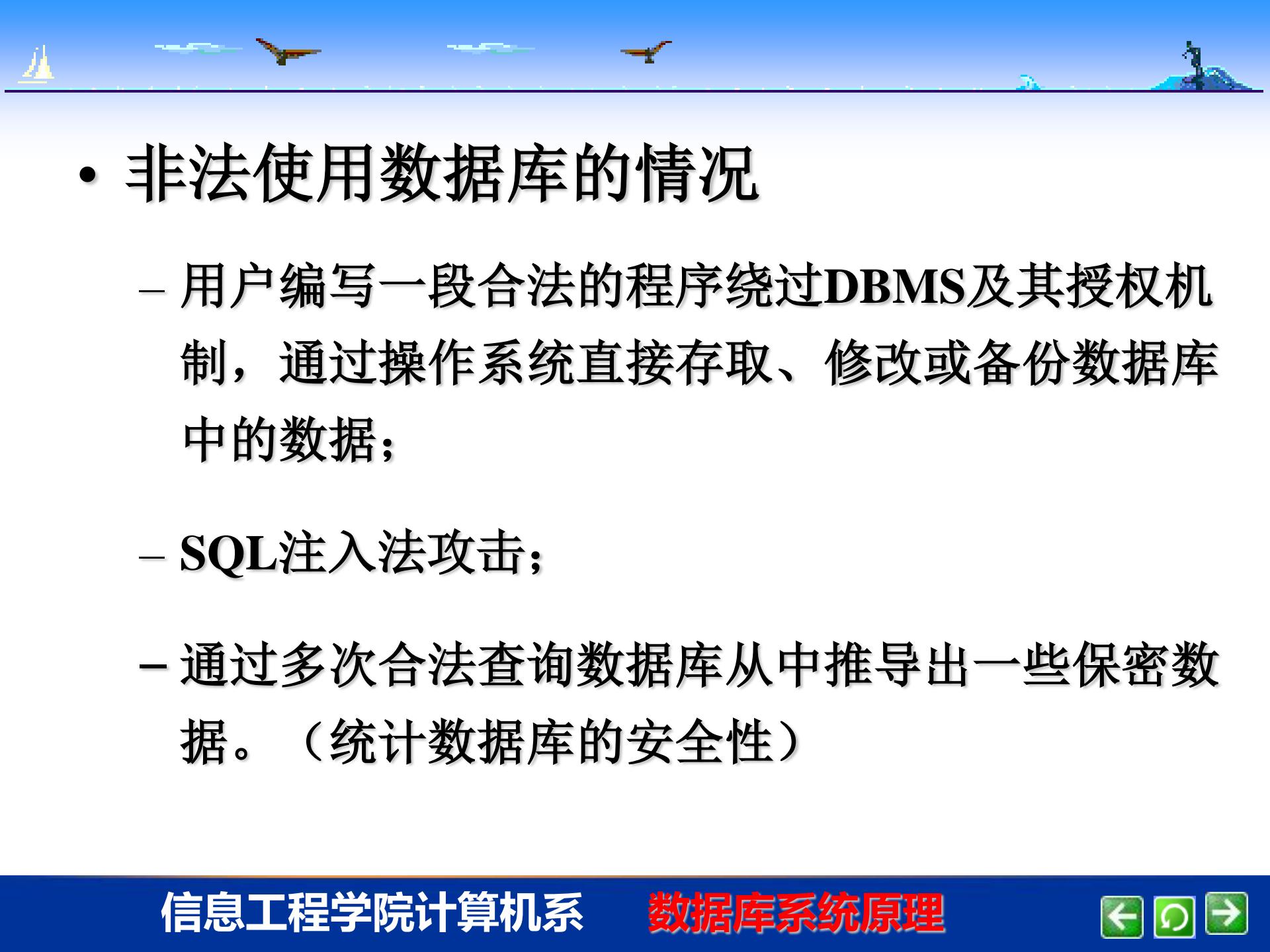
4.3 小结



- 问题的提出
 - 数据库特点: 多用户的数据**共享**
 - 数据共享必然带来数据库的安全性问题
例: 军事秘密 国家机密——危及国家安全
市场需求分析、市场营销策略、销售计划、
客户档案、供货商档案——企业破产
资金帐目, 银行储蓄数据——资金流失
 - 数据库系统中的数据共享不能是无条件的共享

• 什么是数据库的安全性

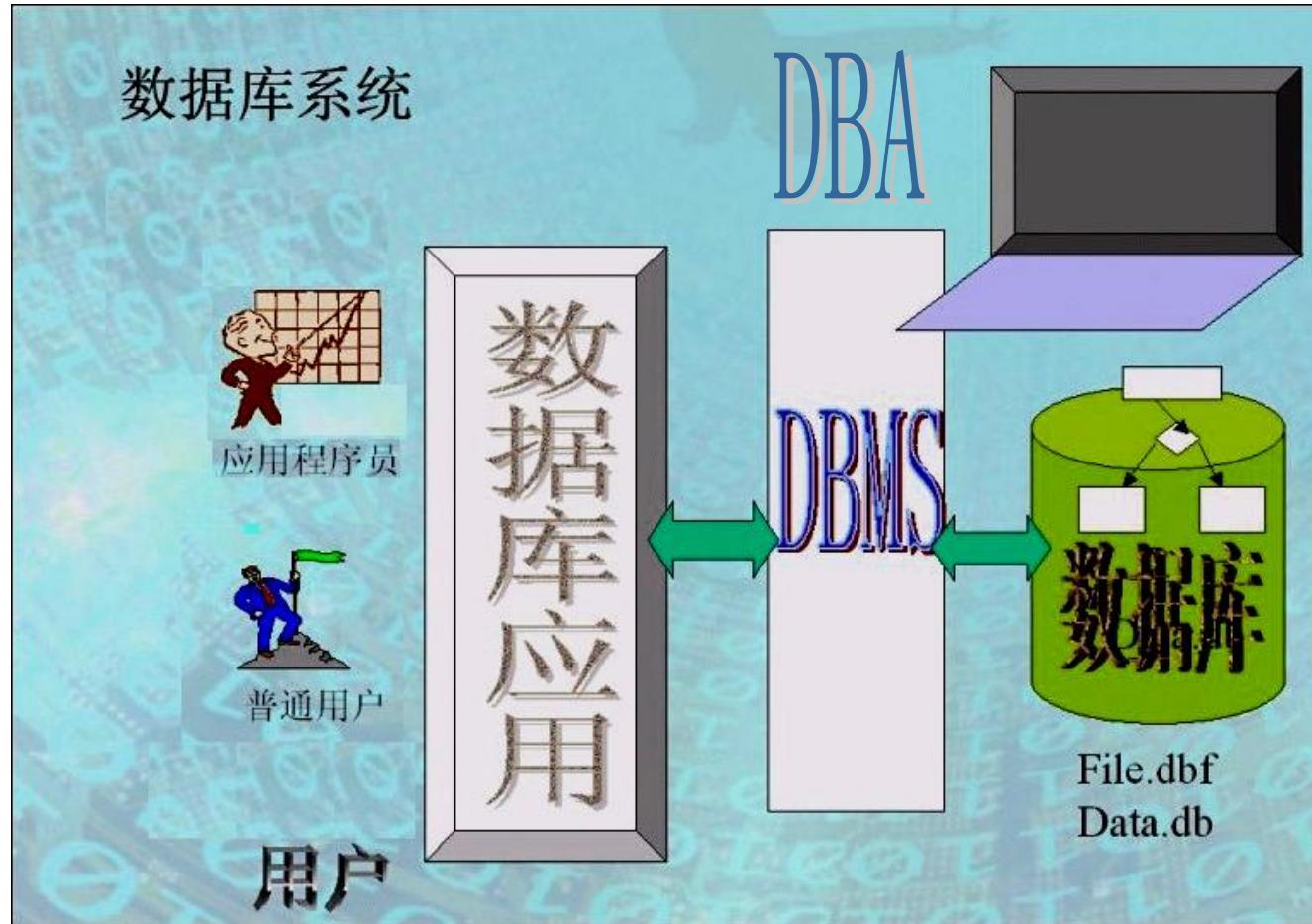
- 数据库的安全性是指保护数据库，防止因用户**非法**使用数据库造成数据泄露、更改或破坏。
- 技术层面；道德层面；法律层面



- 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；
 - SQL注入法攻击；
 - 通过多次合法查询数据库从中推导出一些保密数据。（统计数据的安全性）



数据库系统 (DBS)



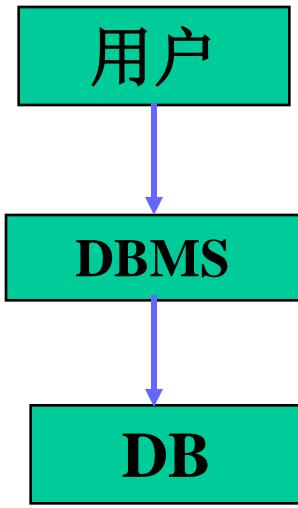


第四章 数据库安全性

4.1 计算机安全性概论

4.2 数据库安全性控制

4.3 小结



用户识别与鉴定

存取控制、审计、视图

密码存储 数据加密



- 数据库安全性控制的常用方法
 - 用户身份鉴定（身份认证）
 - 存取控制
 - 视图机制
 - 审计
 - 数据加密



1 用户身份鉴定

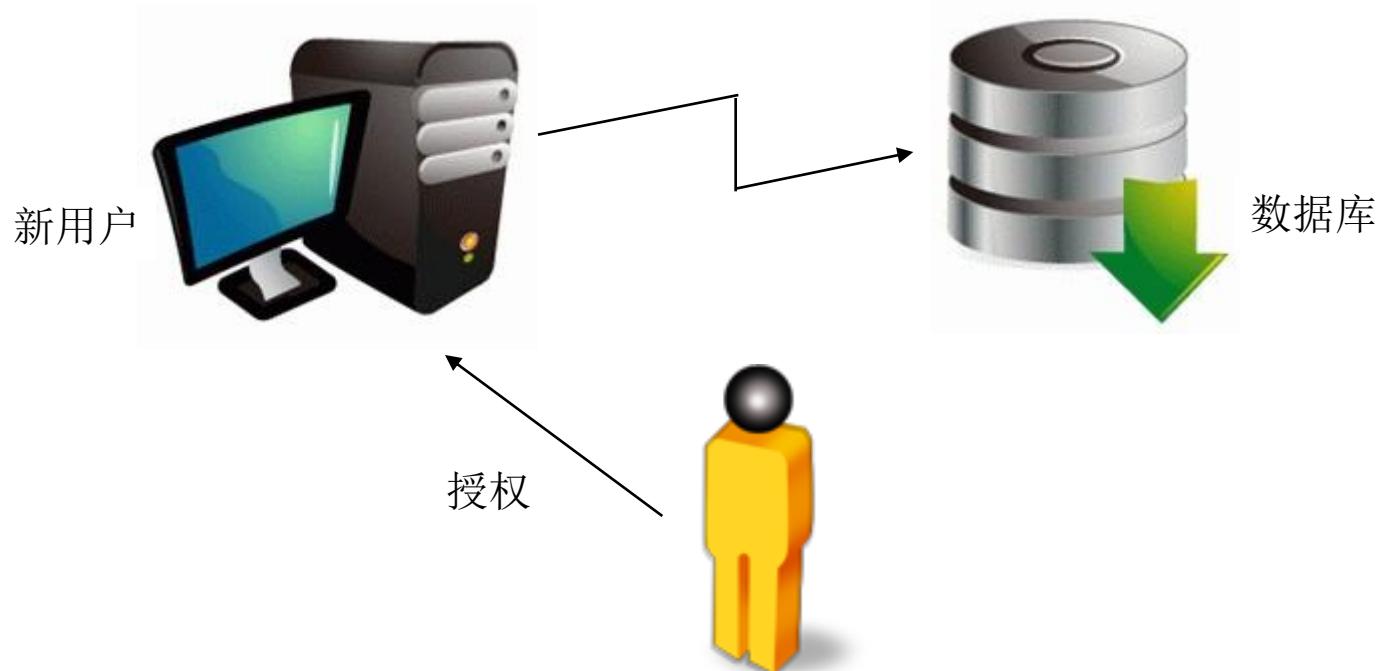
系统提供一定的方式让用户标识自己的名字或身份。

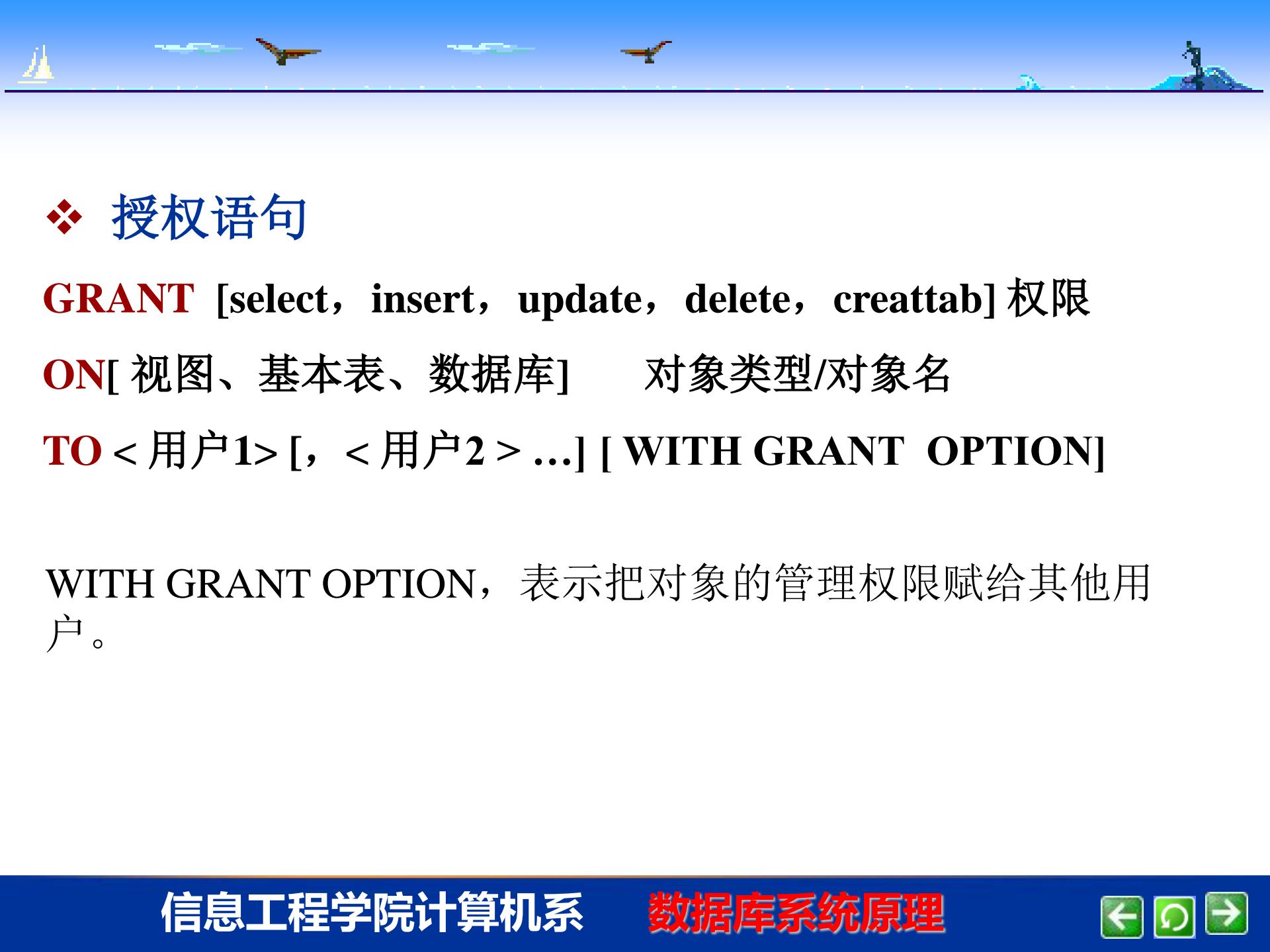
- ❖ 静态口令鉴定：用户名和记忆式密码
- ❖ 动态口令鉴定：短信密码、动态令牌
- ❖ 生物识别技术：指纹、视网膜图案、手相
- ❖ USB KEY认证
- ❖ 智能卡鉴定



2 用户存取权限控制

❖ 权限机制的基本思想：给用户授予不同类型的权限，对用户能够进行的数据库操作以及所操作的数据对象进行限定。





❖ 授权语句

GRANT [select, insert, update, delete, creattab] 权限

ON[视图、基本表、数据库] 对象类型/对象名

TO <用户1> [, <用户2> ...] [WITH GRANT OPTION]

WITH GRANT OPTION， 表示把对象的管理权限赋给其他用户。



权限：查询、插入、修改、删除、建表

对象：基本表、属性列、视图、数据库

例1 USER1可以对C表进行查询。

```
GRANT SELECT ON C
```

```
TO USER1
```



例2 将查询T表和修改教师职称的权限授予USER3，并允许将此权限授予其他用户。

```
GRANT SELECT, UPDATE (PROF)  
ON T  
TO USER3  
WITH GRANT OPTION
```

2. 回收权限:

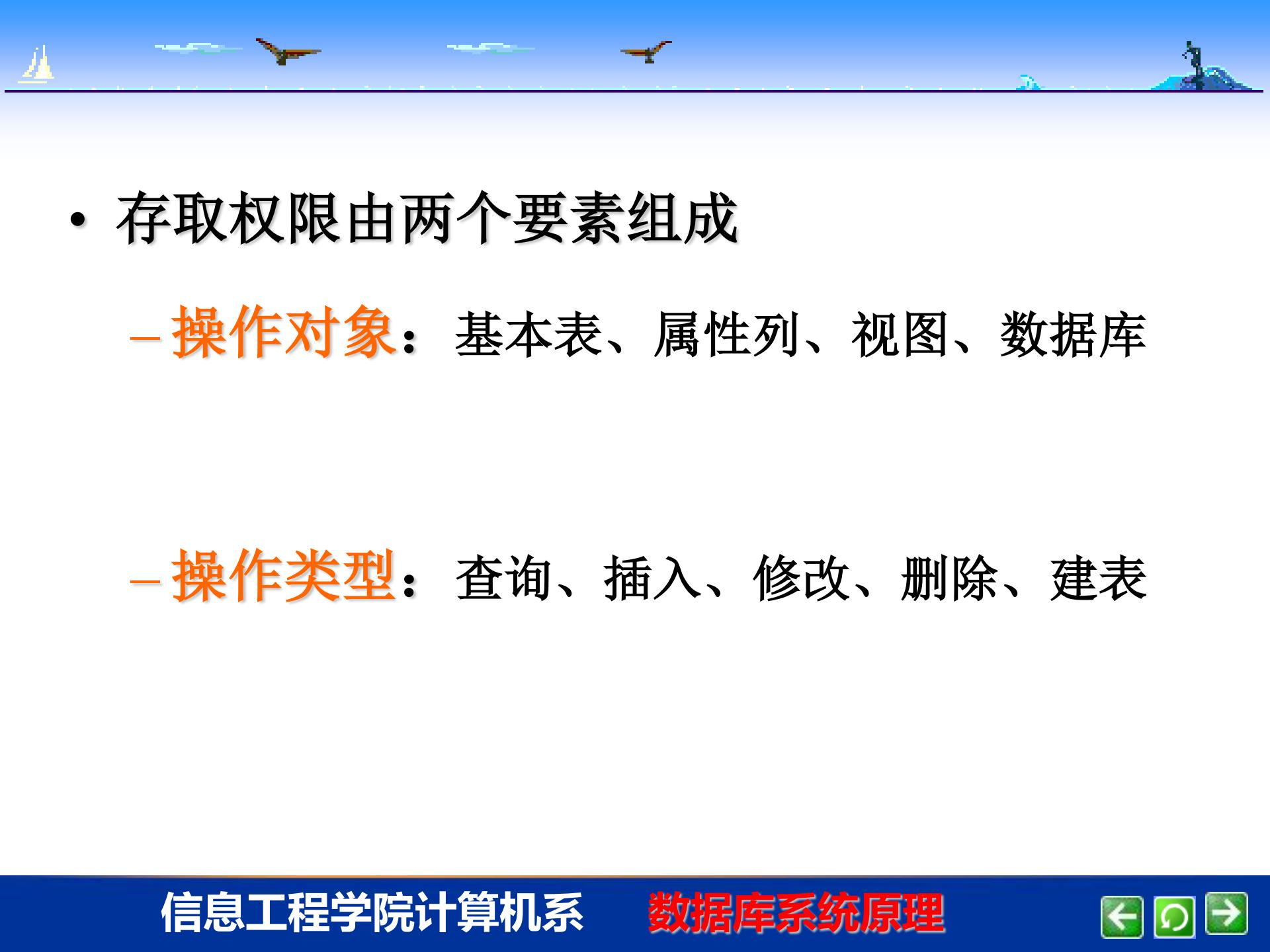
REVOKE <权限>[, <权限>]...

[ON <对象类型> <对象名>]

FROM <用户> [, <用户>]...;

例3 收回用户USER1对C表的查询权限。

```
REVOKE SELECT  
ON C  
FROM USER1
```



- 存取权限由两个要素组成
 - 操作对象：基本表、属性列、视图、数据库
 - 操作类型：查询、插入、修改、删除、建表



❖ 数据字典中授权表

SUBJECT 主体	OBJECT 数据对象	ACTION 操作类型	AUTHORIZATION CONSTRAINT 授权约束
用户名 用户组 机构名	属性 记录 关系 数据库 索引文件	查询 插入 删除 修改 建立表	明确主体、对象和操作的许可范围，如： 部门会计只能查看本部门职工的工资。



❖ **授权粒度：** 授权粒度是指可以定义的数据对象的范围 关系表、元组、属性列

用户标识	数据对象	操作类型
USER1	关系S	ALL
USER2	关系C	SELECT
USER2	关系SC	UPDATE
USER3	关系SC	INSERT
...

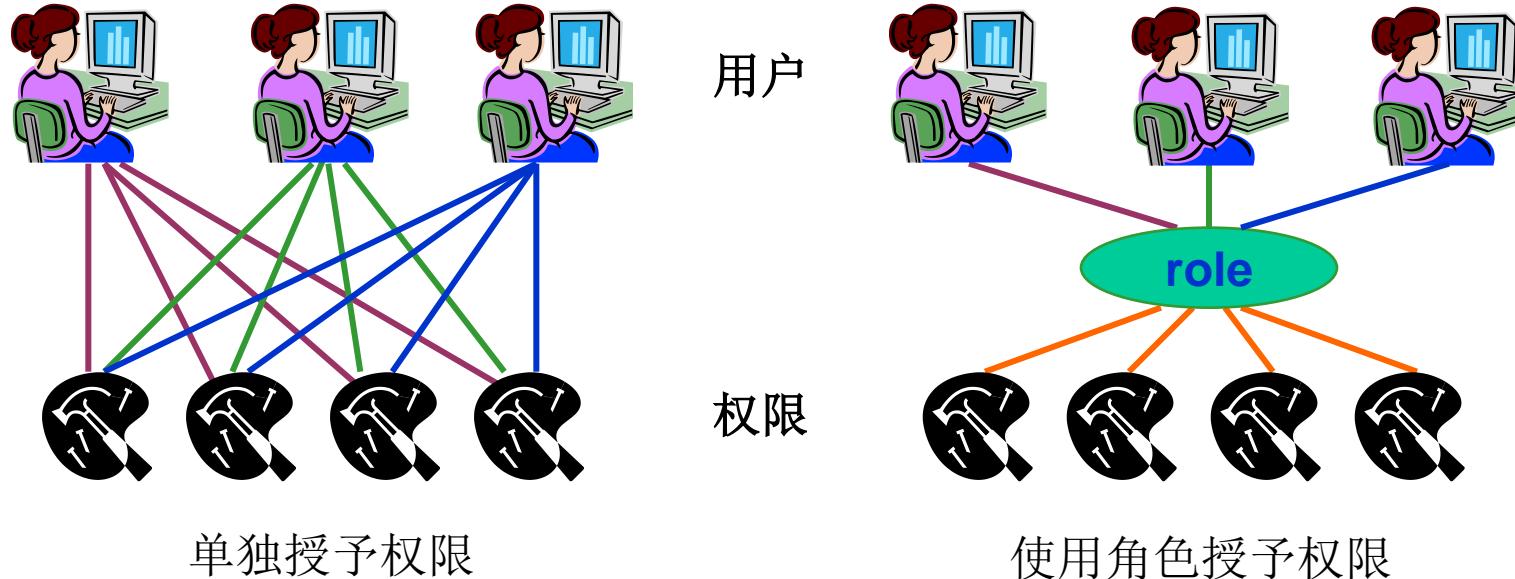
授权粒度粗

用户标识	数据对象	操作类型
USER1	关系S	ALL
USER2	列C. CNO	SELECT
USER2	列SC. SCORE	UPDATE
USER3	关系SC	INSERT
...

授权粒度细

角色

- 角色（ROLE）的目的就是为了简化权限的管理。



- 角色——是一组权限的集合，将角色赋给一个用户，这个用户就拥有了这个角色中的所有权限。



- 角色的创建

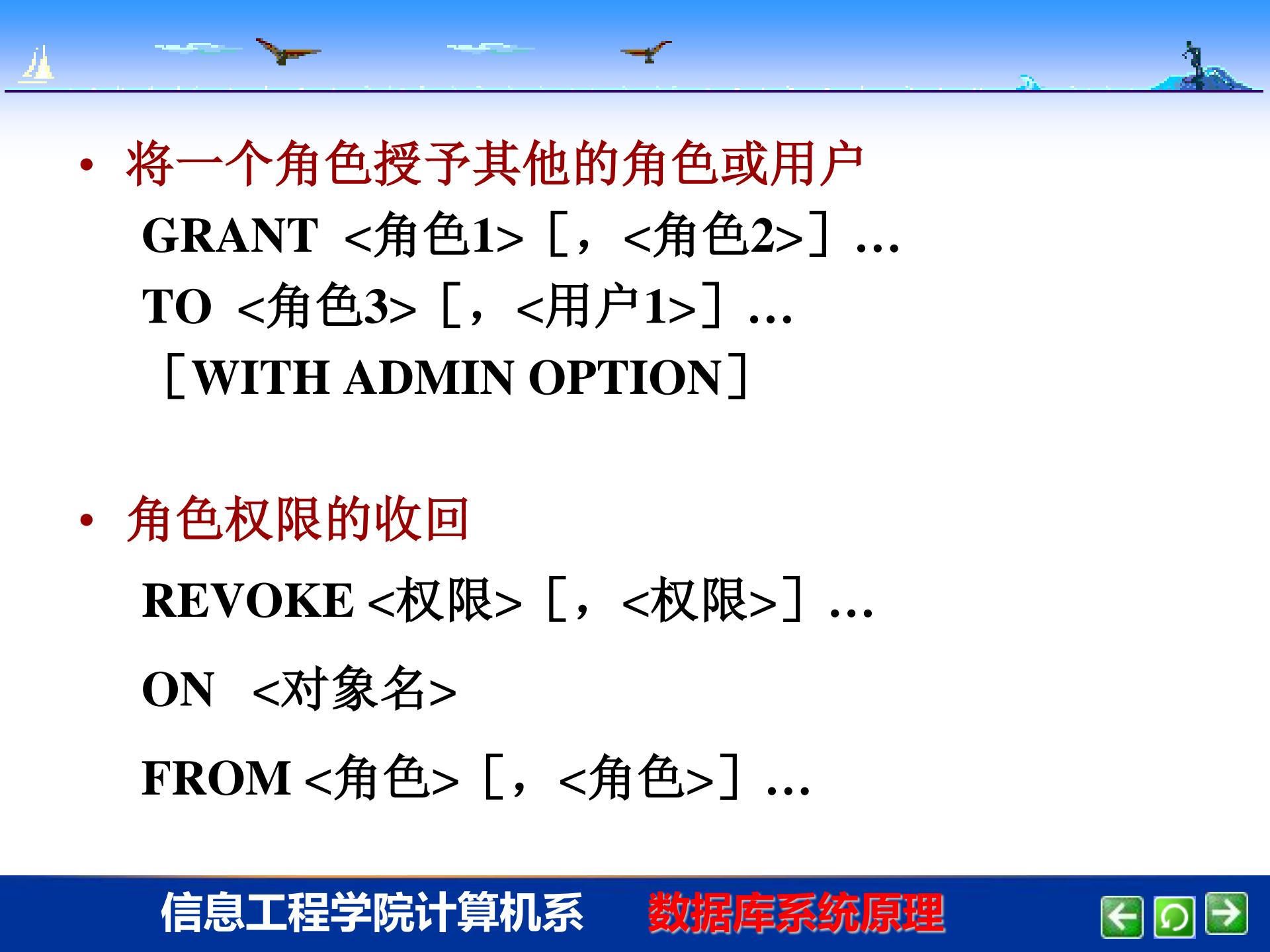
CREATE ROLE <角色名>

- 给角色定义权限集

GRANT <权限> [, <权限>] ...

ON 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>] ...

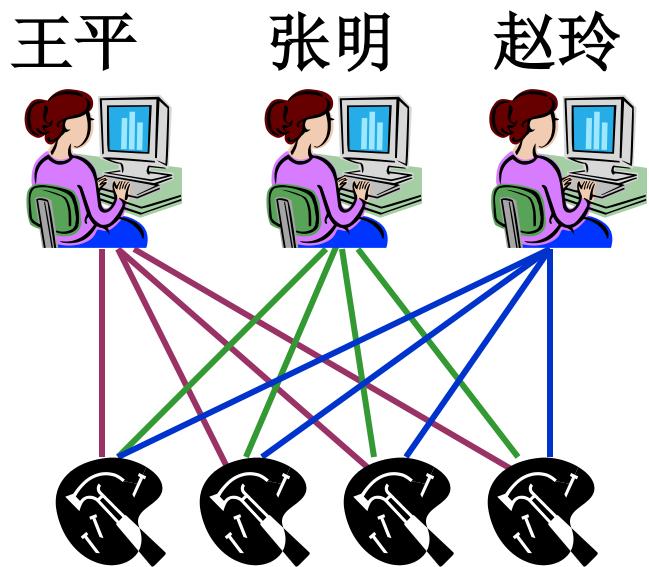
[WITH ADMIN OPTION]

- 角色权限的收回

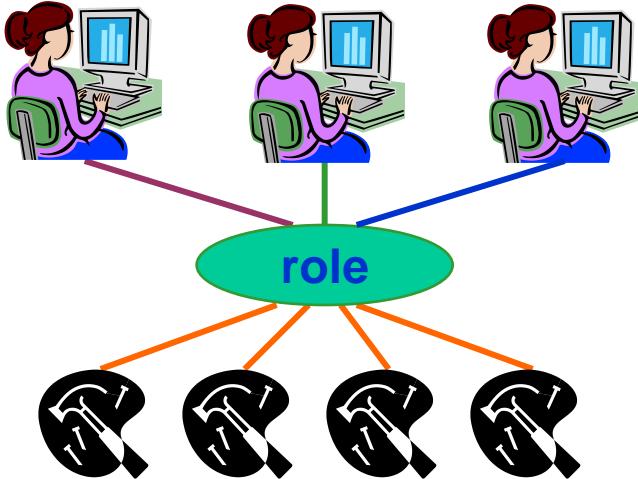
REVOKE <权限> [, <权限>] ...

ON <对象名>

FROM <角色> [, <角色>] ...



用户



Student表的SELECT、UPDATE、INSERT权限

[例] 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 Student
TO R1;**



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;



3 视图机制

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，从而自动地对数据提供一定程度的安全保护。

- 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
- 视图之上再进一步定义存取权限
- 间接实现了支持存取控制的用户权限定义



例：王平只能检索计算机系学生的信息

(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；
```



4 审计 (Audit)

- ❖ 对于某些高度敏感的保密数据，必须以审计功作为预防手段。
- ❖ 审计功能是一种监视措施，跟踪记录用户对数据库的所有操作。
- ❖ 审计日志 (Audit Log)

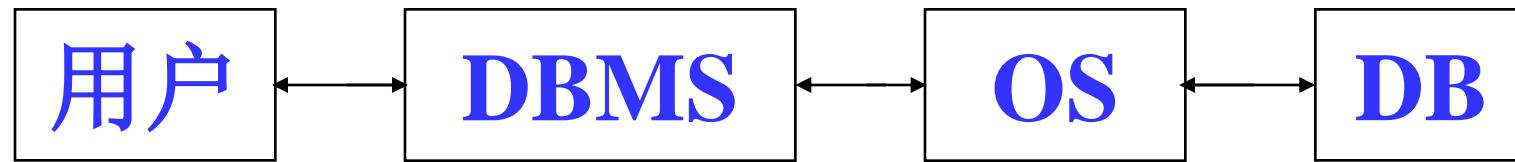


- 对修改 SC表结构或修改SC表数据的操作进行审计。

**AUDIT ALTER, UPDATE
ON SC;**

- 取消对SC表的一切审计。

**NOAUDIT ALTER, UPDATE
ON SC;**



用户身份
鉴别

存取控制
视图 审计

操作系统
安全保护

数据密码
存储



5 数据加密 (Data Encryption)

基本思想：根据一定的算法将原始数据（术语为明文）加密成为不可直接识别的格式（术语为密文），数据以密文的形式存储和传输。

73 20 65 77 20 65 20 83 84 85 68 69 78 84

I a m a s t u d e n t

74 20 66 78 20 66 20 84 85 86 69 70 79 85

J b n b t u v e f o u

加密算法：每个码加1



矩阵加密

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{18} \\ a_{21} & a_{22} & \dots & a_{28} \\ \dots & \dots & \dots & \dots \\ a_{81} & a_{82} & \dots & a_{88} \end{bmatrix}$$

A^{-1} 是A的逆矩阵

$A * A^{-1}$ 等于单位矩阵

$$\begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}$$

加密: $C = A * D$ (明文) = 密文

解密: $C * A^{-1} =$ 明文