SQL安全性定义

主体 客体 权限 角色 审计

加密 SQL注入 统计数据库安全 隐私保护

制度决定论：制定时有罪推定 审判时无罪推定

准入：合法用户，身份鉴定，标识 + 口令

权限：存取范围，只能干什么 VS 不能干什么

配额：资源控制，粮票、阶梯水费、DOS、DDOS

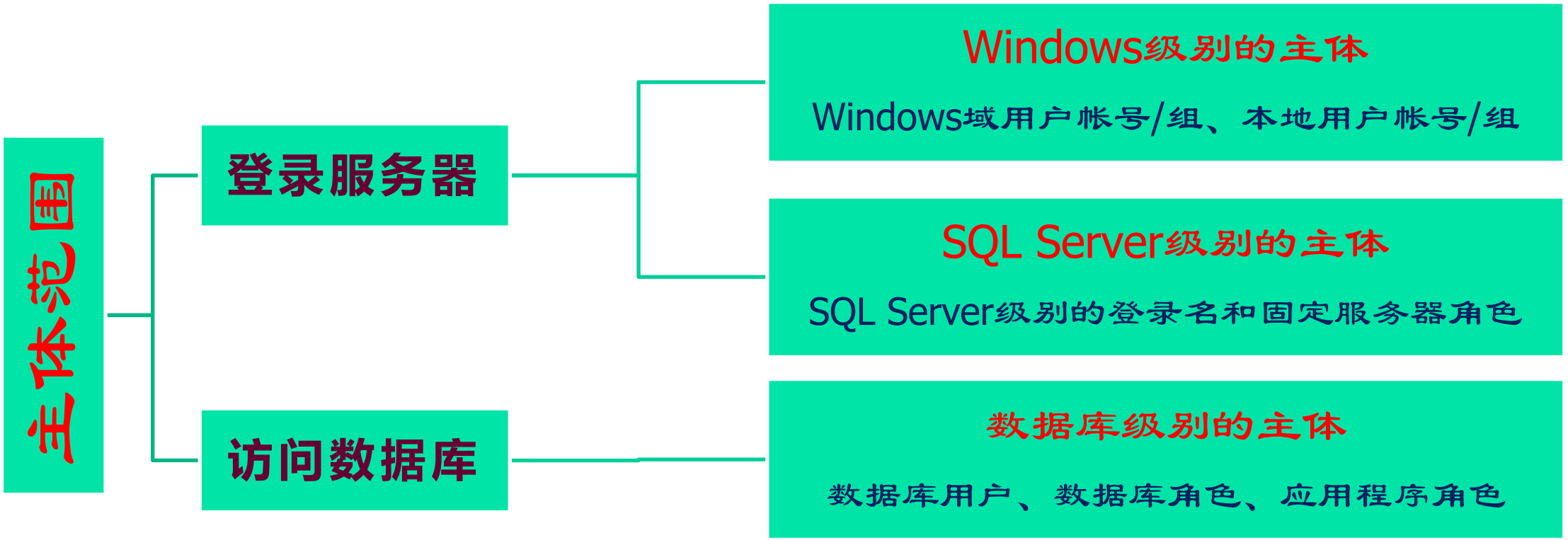
审计：无处不在的探头，一切皆有案底

加密：雾里看花花非花，我知道你不知道

主体

主体(principal)：是可以授予权限以访问特定数据库对象的对象

包括登录用户、角色、应用程序



Windows级别的主体

创建Windows登录名： **create login** login_name from windows

create login [ljchen-PC\SQLUser] from windows

删除Windows登录名： **drop login** login_name

拒绝和允许Windows用户访问SQL Server

deny connect SQL to login_name

grant connect SQL to login_name

Windows级别的主体

查看Windows登录名： **sys.server_principals**

```
select    name, type, sid, principal_id
from      sys.server_principals
where     type_desc = 'WINDOWS_LOGIN'
```

	name	type	sid	principal_id
1	ljchen-PC\ljchen	U	0x0105000000000005150000031A032523C632A344E5DE12...	259
2	NT SERVICE\SQLWriter	U	0x0106000000000005500000000732B9753646EF90356745CB...	260
3	NT SERVICE\Winmgmt	U	0x0106000000000005500000005A048DDFF9C7430AB450D4...	261
4	NT Service\MSSQLSERVER	U	0x0106000000000005500000000E20F4FE7B15874E48E19026...	262
5	NT AUTHORITY\SYSTEM	U	0x01010000000000512000000	263
6	NT SERVICE\SQLSERVERAGENT	U	0x0106000000000005500000000DCA88F14B79FD47A992A3D...	264
7	NT SERVICE\ReportServer	U	0x0106000000000005500000000214401ACF066EA342187301...	265
8	ljchen-PC\SQLUser	U	0x0105000000000005150000031A032523C632A344E5DE12...	268

SQL Server级别的主体

创建SQL Server登录名： **create login** login_name

```
create login  sweetheart with  password = '123456'
                                default_database = 'demoDB'
```

查看SQL Server登录名

```
select    name, type , sid, principal_id
from      sys.server_principals
where     type_desc = 'SQL_LOGIN'
```

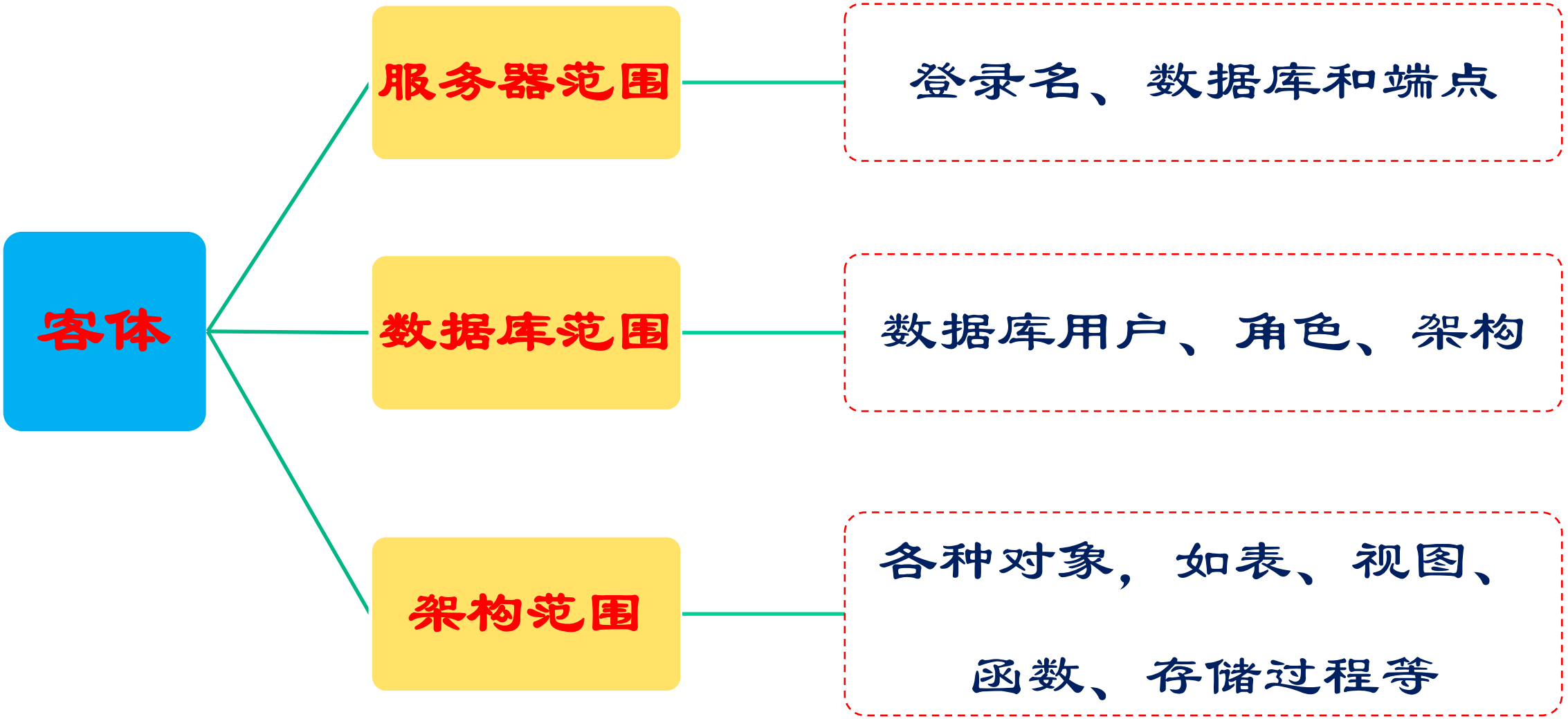
数据库级别的主体

创建SQL Server数据库用户： **create user** user_name

```
create user        Carefully
for login          [ljchen-PC\SQLUser]
with default_schema = Finance
```

	UserName	RoleName	LoginName	DefDBName	DefSchemaName	UserID	SID
1	Carefully	public	ljchen-PC\SQLUser	master	Finance	9	0x0105000000000005150000031A032523C632A344E5DE1

客体：安全对象(securable)



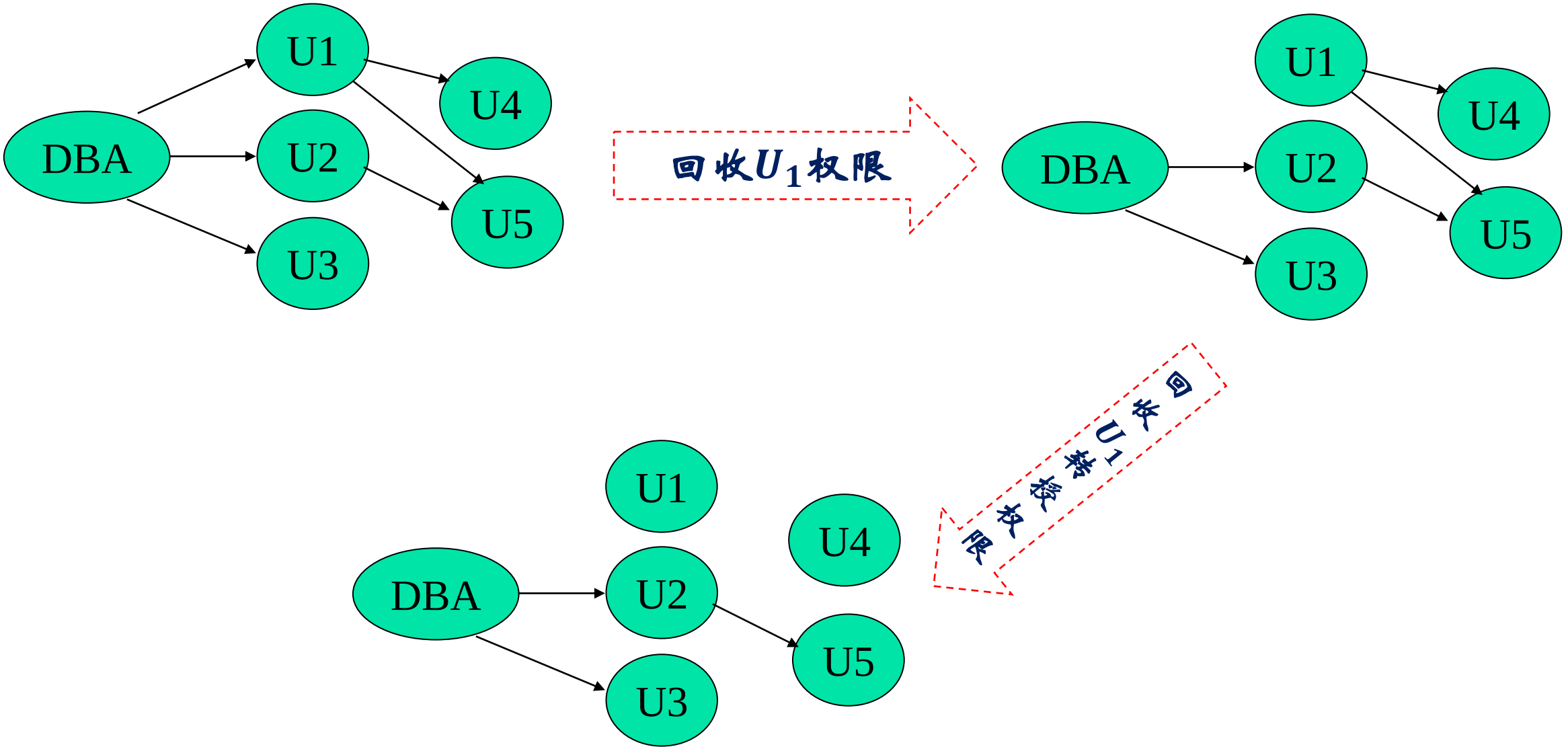
权限

- **权限**(permission)：允许主体在安全对象上执行操作
- **权限的转授和回收**：允许用户把已获得的权限转授给其他用户，或者把已授给其他用户的权限再回收上来

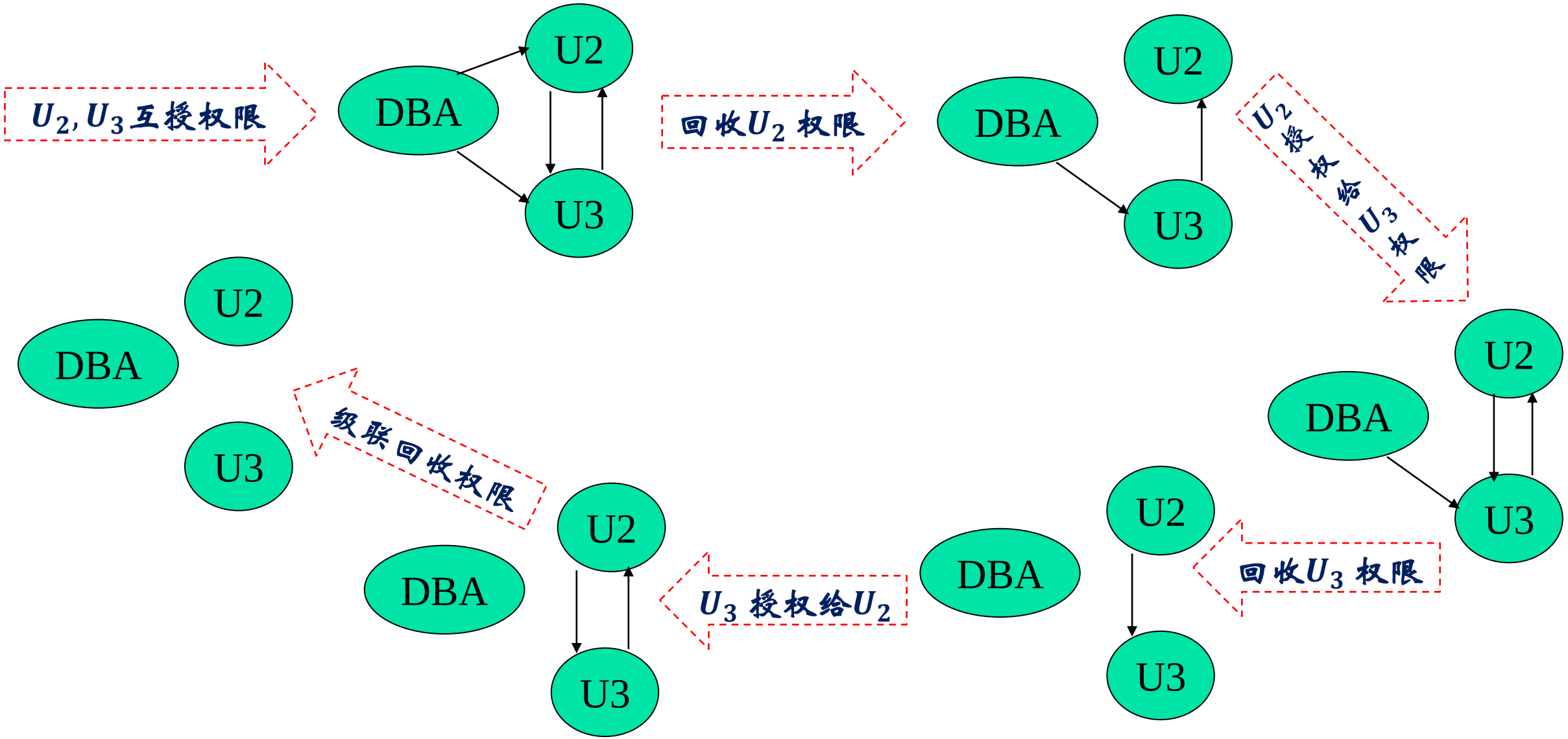
权限图：结点是用户，根结点是DBA

- 有向边$U_i \longrightarrow U_j$，表示用户U_i把某权限授给用户U_j
- 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

授权图


U1
U4
DBA
U2
U5
U3
回收U_1权限
U1
U4
DBA
U2
U5
U3
回收U_1转授权限
U1
U4
DBA
U2
U5
U3

授权图要始终保证授权路径起点是DBA



报告可用权限

sys.fn_builtin_permissions

```
select    class_desc, permission_name,
          covering_permission_name, parent_class_desc,
          parent_covering_permission_name
from      sys.fn_builtin_permissions('object')
```

	class_desc	permission_name	covering_permission_name	parent_class_desc	parent_covering_permission_name
1	OBJECT	SELECT	RECEIVE	SCHEMA	SELECT
2	OBJECT	UPDATE	CONTROL	SCHEMA	UPDATE
3	OBJECT	REFERENCES	CONTROL	SCHEMA	REFERENCES
4	OBJECT	INSERT	CONTROL	SCHEMA	INSERT
5	OBJECT	DELETE	CONTROL	SCHEMA	DELETE
6	OBJECT	EXECUTE	CONTROL	SCHEMA	EXECUTE
7	OBJECT	RECEIVE	CONTROL	SCHEMA	CONTROL
8	OBJECT	VIEW CHANGE TRACKING	CONTROL	SCHEMA	VIEW CHANGE TRACKING
9	OBJECT	VIEW DEFINITION	CONTROL	SCHEMA	VIEW DEFINITION
10	OBJECT	ALTER	CONTROL	SCHEMA	ALTER
11	OBJECT	TAKE OWNERSHIP	CONTROL	SCHEMA	CONTROL
12	OBJECT	CONTROL		SCHEMA	CONTROL

授权命令

```
grant     权限
on        对象名
to        {用户 [，用户]… | public}
          [with grant option]
```

with grant option

获得权限的用户可以把权限再授予其它用户

表级权限：**select, update, insert, delete, index, alter, drop, resource**等以及它们的总和**all**

为什么需要references权限？

回收权限

revoke	权限
on	对象
from	{用户 [，用户]… | **public**}

授权路径的起点一定是**DBA**

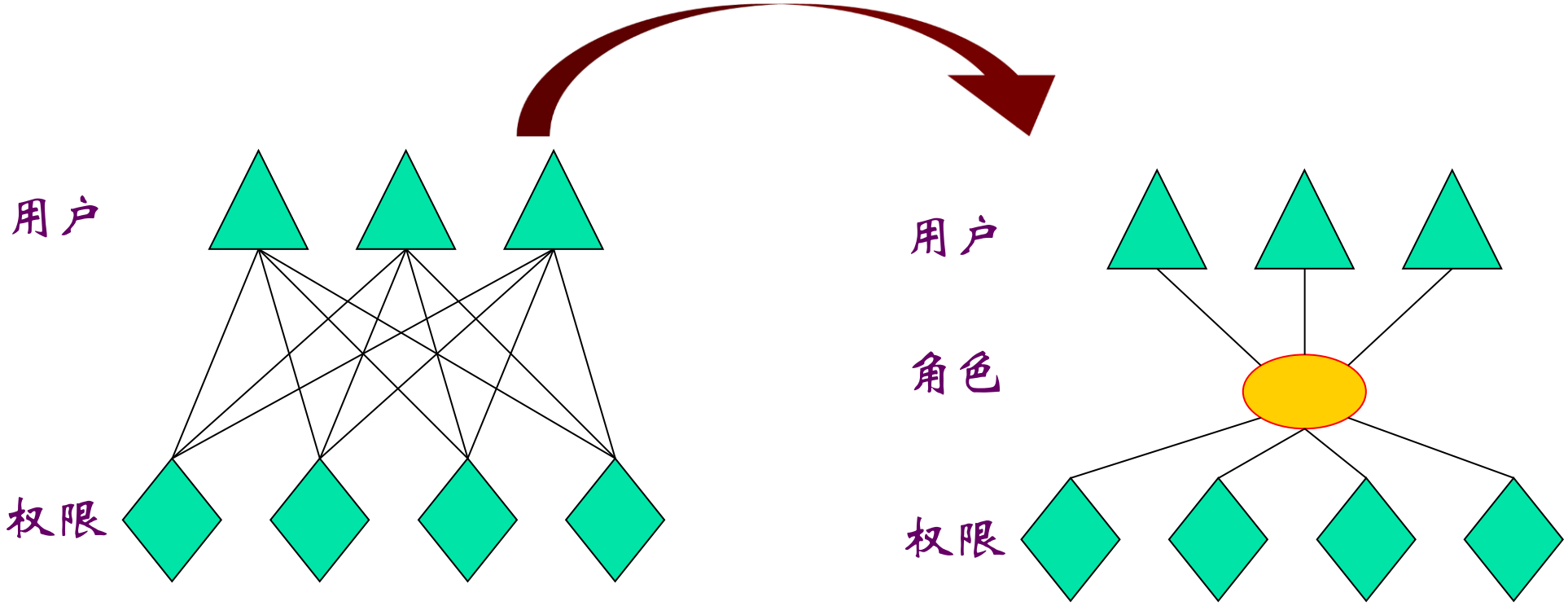
收回权限时，若该用户已将权限转授给其它用户，则也一并收回

grant select , insert on S **to** Liming **with grant option**

revoke insert on S **from** Liming

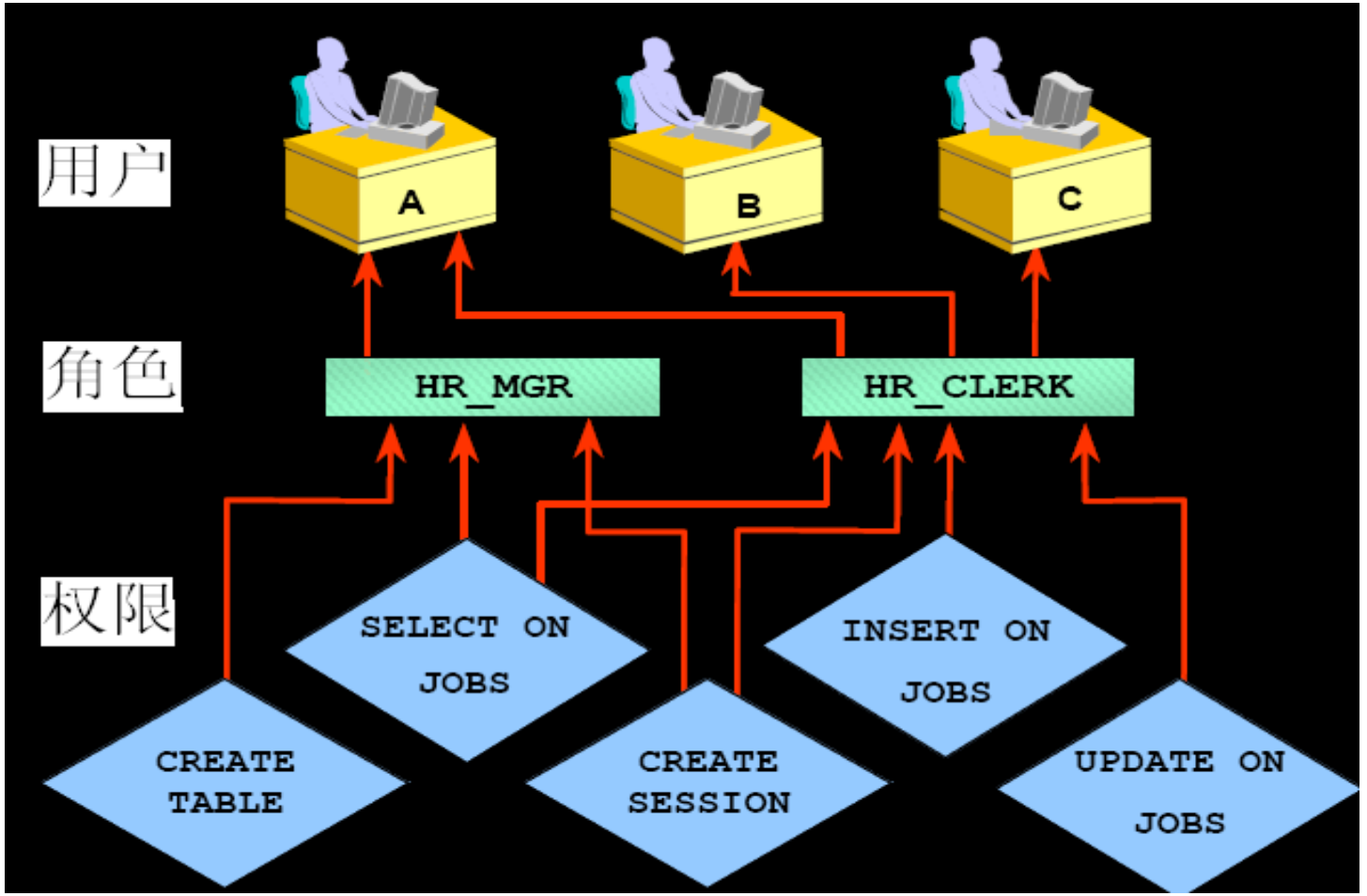
角色

- 角色是一组相关权限的结合，即将多个不同的权限集合在一起就形成了角色



角色

人事权



创建数据库角色：**create role** role_name

角色

查看SQL Server固定服务器角色

```
select      name
from        sys.server_principals
where       type_desc = 'SERVER_ROLE'
```

public sysadmin securityadmin
serveradmin setupadmin
processadmin diskadmin
dbcreator bulkadmin

添加登录名到固定服务器角色：sp_addsrvrolemember

```
sp_addsrvrolemember  'sweetHeart', 'sysadmin'
```

数据库角色

添加用户名到数据库角色：**sp_addrolemember**

sp_addrolemember ‘**db_datawriter**’, ‘Carefully’

查看固定数据库角色**sp_helpdbfixedrole**

	DbFixedRole	Description
1	db_owner	DB Owners
2	db_accessadmin	DB Access Administrators
3	db_securityadmin	DB Security Administrators
4	db_ddladmin	DB DDL Administrators
5	db_backupoperator	DB Backup Operator
6	db_datareader	DB Data Reader
7	db_datawriter	DB Data Writer
8	db_denydatareader	DB Deny Data Reader
9	db_denydatawriter	DB Deny Data Writer

查看角色成员

sp_helprolemember

	DbRole	MemberName
1	db_datawriter	Carefully
2	db_owner	dbo
3	RSExecRole	NT SERVICE\ReportServer

利用当前用户实现行级精细存取控制

```
declare     @usr char(30)
set         @usr = user
select      'The current user is: '+ @usr
```

- 普通员工只能查看自己的记录
- 部门经理可以查看他所管理的员工
- 人力资源代表可以查看所有员工

```
select      *
from        student
where       sname = user
```

基于视图的安全性控制

授权Tom只有察看职工平均工资的权限

```
create view  avg_sal
    as
        ( select    avg(sal)
          from      teacher )
```

```
grant   SELECT  on  avg_sal  to  'Tom'
```

访问控制类型

自主访问控制（DAC）

- 对客体拥有控制权的主体能够将对该客体的访问权自主地授予其它主体，并在随后任何时刻将这些权限回收

强制访问控制（MAC）

- 敏感度标记：绝密、机密、可信、公开
- 主体：许可证级别；客体：密级

保密性规则

- **下读**：仅当主体许可证级别高于或等于客体密级时才能读取相应客体
- **上写**：仅当主体许可证级别低于或等于客体密级时体才能写相应客体

审计

- 审计就是对指定用户在数据库中的操作情况进行监控和记录，用以审查用户的相关活动
- 如数据被非授权用户删除，用户越权管理，权限管理不正确，用户获得不应有的系统权限等

- 审计就是监视和收集指定数据库的活动数据
- 如哪些表经常被修改，用户共执行了多少次I/O操作等，为优化提供依据

审计： SQL Server

服务器审核： **create server audit** MyServerAudit **to file** ...

服务器审核规范

create server audit specification MyServerAuditSpe

for server audit MyServerAudit

alter server audit specification MyServerAuditSpe

add (SERVER_PRINCIPAL_CHANGE_GROUP)

审计： SQL Server

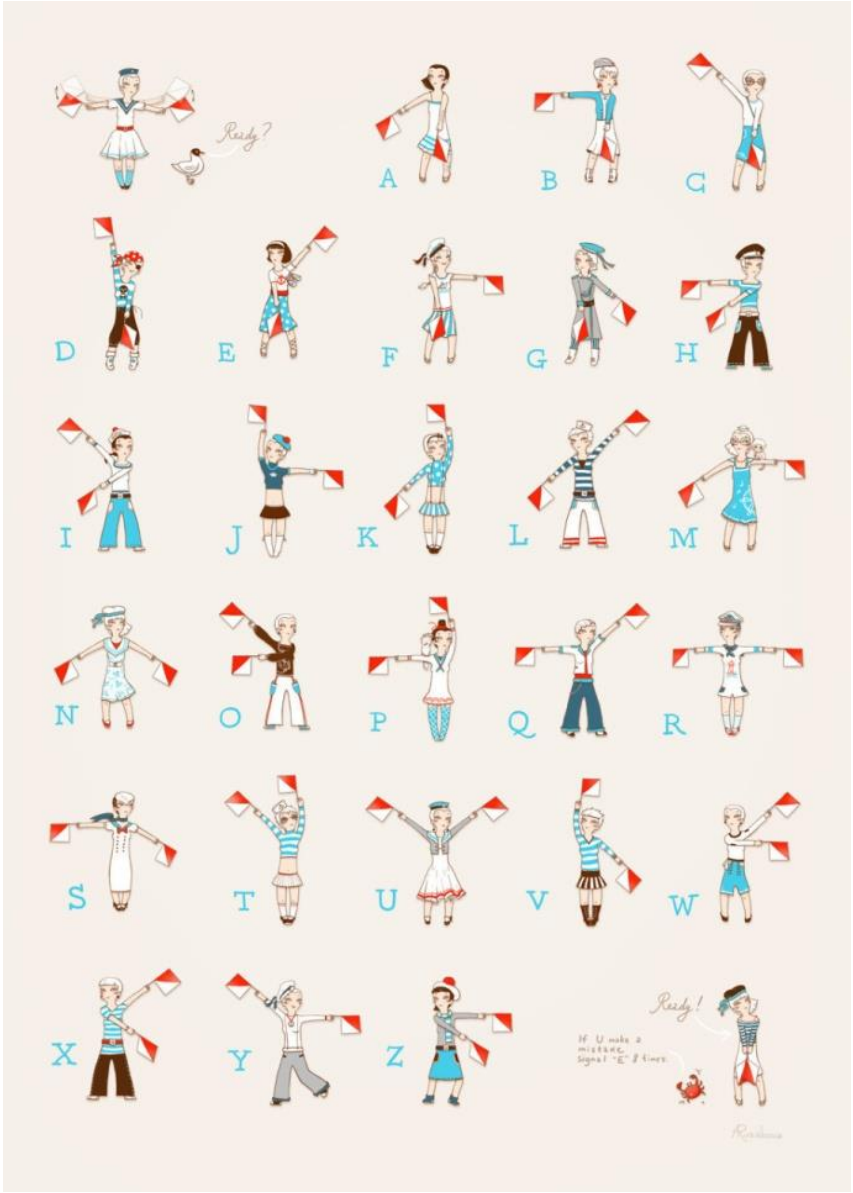
数据库审核规范

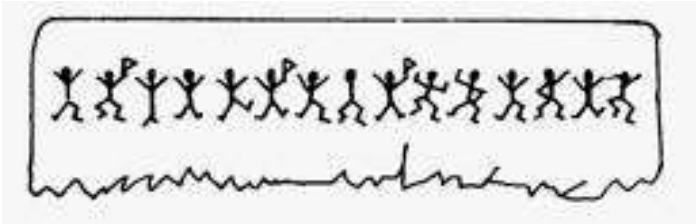
```
create database audit specification  MyDBAudit
        for server audit  MyServerAudit
alter database audit specification  MyDBAudit
        add ( SELECT ON student )
```

查看审核历史

```
select    event_time, succeeded, statement
from      sys.fn_get_audit_file(…)
```

加密


Ready?
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Ready!




Caesar
Cipher
"Three Letters Back"
RoyalOrderoftheHolyMackerel.com

短语加密

加密数据： **encryptByPassPhrase** ({ 'passphrase', 'cleartext'})

select **encryptByPassPhrase** ('hello', 'who am i')

0x0100000021D68E2E078E3EA6752239788B69D8B9BF1AD542A7C9774C9CAF66304F215F49

还原数据： **decryptByPassPhrase** ({ 'passphrase', 'ciphertext' }

select **decryptByPassPhrase** ('hello',

0x0100000021D68E2E078E3EA6752239788B69D8B9BF1AD542A7C9774C9CAF66304F215F49)

0x77686F20616D2069

非对称密钥加密

```
create asymmetric key  myAsym_key

insert into emp(ename, salary)  values ( 'tom',

    EncryptByAsymkey( Asymkey_ID('myAsym_key'), 100000000) )
```

```
select      DecryptByAsymkey( Asymkey_ID('myAsym_key'), salary )

from        emp

where       name = 'tom'
```

对称密钥加密

```
create symmetric key  mySym_key
insert into  emp(ename, salary) values ( 'tom',
        EncryptBykey( Key_GUID('mySym_key'), 100000000 ) )
```

```
select      DecryptByKey(Key_GUID('mySym_key'), salary )
from        emp
where       name = 'tom'
```

SQL注入

认证过程发出的查询语句

```
select  * from users
where  username = 'jake'
   and    PASSWORD = 'jakespasswd'
```

攻击者篡改这个SQL语句

```
select  * from users
where  username = 'jake'
   and    ( PASSWORD = 'jakespasswd'  or  'x' = 'x' )
```

资源控制： Oracle

PROFILE

➢CPU使用时间限制	CPU_PER_SESSION
➢逻辑读个数限制	LOGICAL_READS_PER_SESSION
➢用户会话限制	SESSION_PER_USER
➢会话空闲时间限制	IDLE_TIME
➢会话可持续时间限制	CONNECT_TIME
➢会话专用SGA空间限制	PRIVATE_SGA
➢口令限制	PASSWORD_LIFE_TIM
	PASSWORD_LOCK_TIME
	FAILED_LOGIN_ATTEMPTS

统计数据库安全性

要求：用户只能查询数据的聚集值，不能访问个体

漏洞一：个体太少

- 查询选修“古典哲学史”的学生的平均成绩

漏洞二：多次查询，太多交叠

- Q_1: 查询n个学生的总成绩为x
- Q_2: 查询n个学生+A的总成绩为y，推出A的总成绩为y-x

统计数据库安全性

防范措施

- 查询引用的数据不能少于n
- 两个查询的交不能多于m

推出个体信息至少需要$1+\frac{n-2}{m}$次查询

Student(ID, GPA)，ID从1到50

要求任何查询结果只能是一个聚集值，每次至少使用4条元组，任何两个查询的交不能大于2条元组。给出一个查询集合，使得能确定ID=9的GPA

隐私保护

文明，是向拥有隐私权的社会不断迈进的进程

文明，是将一个人从一群人当中解放出来的进程

- 一九八四，数据监控：Big Data is Big Brother
- 中央数据银行：数据脚印整合，1+1>2
- 爱国者法案，国土安全法，万维信息触角
- 统一身份识别：驾照、护照、社会安全号

隐私保护数据挖掘：数据发布

Name	Age	Sex	Zipcode	Disease
Andy	5	M	12000	gastric ulcer
Bill	9	M	14000	dyspepsia
Ken	6	M	18000	pneumonia
Nash	8	M	19000	bronchitis
Joe	12	M	22000	pneumonia
Sam	19	M	24000	pneumonia
Linda	21	F	58000	flu
Jane	26	F	36000	gastritis
Sarah	28	F	37000	pneumonia
Mary	56	F	33000	flu

Age	Sex	Zipcode	Disease
5	M	12000	gastric ulcer
9	M	14000	dyspepsia
6	M	18000	pneumonia
8	M	19000	bronchitis
12	M	22000	pneumonia
19	M	24000	pneumonia
21	F	58000	flu
26	F	36000	gastritis
28	F	37000	pneumonia
56	F	33000	flu

连接推理

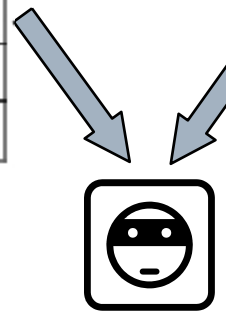
发布表

Age	Sex	Zipcode	Disease
5	M	12000	gastric ulcer
9	M	14000	dyspepsia
6	M	18000	pneumonia
8	M	19000	bronchitis
12	M	22000	pneumonia
19	M	24000	pneumonia
21	F	58000	flu
26	F	36000	gastritis
28	F	37000	pneumonia
56	F	33000	flu

Quasi-identifier (QI) attributes

选民登记表

Name	Age	Sex	Zipcode
Andy	5	M	12000
Bill	9	M	14000
Ken	6	M	18000
Nash	8	M	19000
Mike	*7*	*M*	*17000*
Joe	12	M	22000
Sam	19	M	24000
Linda	21	F	58000
Jane	26	F	36000
Sarah	28	F	37000
Mary	56	F	33000



An adversary

[Swe00]的研究表明，87%的美国人口信息可以通过性别、出生日期、5位邮政编码进行个人重建

k-anonymity

有相同的准标识属性组(QI)的元组至少有k个(2-anonymous)

QI 属性

ST属性

Age	Sex	Zipcode	Disease
[1, 10]	M	[10001, 15000]	gastric ulcer
[1, 10]	M	[10001, 15000]	dyspepsia
[1, 10]	M	[15001, 20000]	pneumonia
[1, 10]	M	[15001, 20000]	bronchitis
[11, 20]	M	[20001, 25000]	pneumonia
[11, 20]	M	[20001, 25000]	pneumonia
[21, 60]	F	[30000, 60000]	flu
[21, 60]	F	[30000, 60000]	gastritis
[21, 60]	F	[30000, 60000]	pneumonia
[21, 60]	F	[30000, 60000]	flu

4 QI 分组

我男朋友给他浇两次水我闺蜜有时候就回回浇一次或者两次

然后我就按照他俩浇水的日期查看我和我男朋友的聊天记录

发现我闺蜜回一次水的时候我男朋友不是加班就是公司集体活动

这种巧合也未免太明显了

我就猜到他浇水2次就是问约吗

闺蜜回一次就是约

回两次就是不约