厦门大学计算机科学系本科生课程

《数据库系统原理》

第4章 数据库安全性
（2016版）

林子雨

厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

主页：http://www.cs.xmu.edu.cn/linziyu

第4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(Audit)
- 4.5 数据加密
- 4.6 统计数据库安全性

4.1 计算机安全性概论

- 4.1.1 计算机系统的三类安全性问题
- 4.1.2 安全标准简介

4.1.1 计算机系统的三类安全性问题

- 什么是计算机系统安全性

  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，数据遭到更改或泄露等。

- 计算机安全涉及问题

  计算机系统本身的技术问题
  - 计算机安全理论与策略
  - 计算机安全技术

  管理问题
  - 安全管理
  - 安全评价
  - 安全产品

  法学
  - 计算机安全法律

  犯罪学
  - 计算机犯罪与侦察
  - 安全监察

  心理学

4.1.1 计算机系统的三类安全性问题

- 三类计算机系统安全性问题
  - 技术安全类
  - 管理安全类
  - 政策法律类

4.1.1 计算机系统的三类安全性问题

- 技术安全类

  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其所存数据的安全保护，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。

- 管理安全类

  软硬件意外故障、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

  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

4.1.1 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 (桔皮书)

  TDI (紫皮书)

4.1.1 安全标准简介

- 1985年美国国防部（DoD）正式颁布《 DoD可信计算机系统评估标准》（简称TCSEC或DoD85）

  TCSEC又称桔皮书

  TCSEC标准的目的

  提供一种标准，使用户可以对其计算机系统内敏感信息安全操作的可信程度做评估。

  给计算机行业的制造商提供一种可循的指导规则，使其产品能够更好地满足敏感应用的安全需求。

4.1.1 安全标准简介

- **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》（ Trusted Database Interpretation 简称TDI）**

  TDI又称紫皮书。它将TCSEC扩展到数据库管理系统。

  TDI中定义了数据库管理系统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。

4.1.1 安全标准简介

- **TDI/TCSEC标准的基本内容**

  TDI与TCSEC一样，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

4.1.1 安全标准简介

- **R1 安全策略（Security Policy）**

  R1.1 自主存取控制 （Discretionary Access Control，简记为DAC）

  R1.2 客体重用（Object Reuse）

  R1.3 标记（Labels）

  R1.4 强制存取控制（Mandatory Access Control，简记为MAC）

- **R2 责任（Accountability）**

  R2.1 标识与鉴别（Identification & Authentication）

  R2.2 审计（Audit）

4.1.1 安全标准简介

- **R3 保证（Assurance）**
  - **R3.1 操作保证（Operational Assurance）**
  - **R3.2 生命周期保证（Life Cycle Assurance）**
- **R4 文档（Documentation）**
  - **R4.1 安全特性用户指南（Security Features User's Guide）**
  - **R4.2 可信设施手册（Trusted Facility Manual）**
  - **R4.3 测试文档（Test Documentation）**
  - **R4.4 设计文档（Design Documentation）**

4.1.1 安全标准简介

- **TCSEC/TDI安全级别划分**

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）

4.1.1 安全标准简介

- 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。

4.1.1 安全标准简介

- **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什么专门的机制来保障

- **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。

4.1.1 安全标准简介

- **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 Server2003，
  数字设备公司的Open VMS VAX 6.0和6.1

  数据库

  Oracle公司的Oracle
  Sybase公司的 SQL Server

4.1.1 安全标准简介

- **B1级**

标记安全保护。“安全”(Security)或“可信的”(Trusted)产品。

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。

典型例子：

操作系统

数字设备公司的SEVMS VAX Version 6.0

惠普公司的HP-UX BLS release 4.0.9+

数据库

Oracle公司的Trusted Oracle Sybase公司的Secure SQL Server version 11.0.6

Informix公司的Incorporated INFORMIX-OnLine / Secure 5.0

4.1.1 安全标准简介

- **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

  典型例子：

  操作系统

  只有**Trusted Information Systems**公司的**Trusted XENIX**一种产品

  标准的网络产品

  只有**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

4.1.1 安全标准简介

- **B3级**

  安全域。

  该级的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。

- **B2以上的系统**

  还处于理论研究阶段。

  应用多限于一些特殊的部门如军队等。

  美国正在大力发展安全产品，试图将目前仅限于少数领域应用的 B2 安全级别下放到商业应用中来，并逐步成为新的商业标准。

4.2 数据库安全性控制

- 4.2.1 用户标识与鉴别(Identification & Authentication)
- 4.2.2 存取控制
- 4.2.3 自主存取控制(DAC)方法
- 4.2.4 授权(Authorization)与回收
- 4.2.5 数据库角色
- 4.2.6 强制存取控制(MAC)方法

4.2.1 用户标识与鉴别(Identification & Authentication)

- 用户标识与鉴别（**Identification & Authentication**）是系统提供的最外层安全保护措施

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；
- 用户标识和鉴定可以重复多次。

4.2.2 存取控制

- 存取控制机制的功能

存取控制机制的组成：定义存取权限和检查存取权限。用户权限定义和合法权检查机制一起组成了DBMS的安全子系统。

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（Discretionary Access Control，简称DAC）**
  - 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
- **强制存取控制（Mandatory Access Control，简称 MAC）**
  - 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

- 定义存取权限

  存取权限由两个要素组成

  数据对象

  操作类型

4.2.3 自主存取控制(DAC)方法

关系数据库系统中的存取权限

- **类型**

对象数据	对象	操作类型
数据库	模式	CREATE SCHEMA
	基本表	CREATE TABLE,ALTER TABLE
模式	视图	CREATE VIWE
	索引	CREATE INDEX
数据	基本表和视图	SELECT,INSERT,UPDATE,DALETE,REFERENCES,ALL PRIVILEGES
	属性列	SELECT,INSERT,UPDATE,REFERENCES,ALL PRIVILEGES

4.2.3 自主存取控制(DAC)方法

关系系统中的存取权限

- 定义方法

  GRANT/REVOKE

DBMS实现数据安全性保护的过程

1. 用户或**DBA**把授权决定告知系统
2. **SQL**的**GRANT**和**REVOKE**
3. **DBMS**把授权的结果存入数据字典
4. 当用户提出操作请求时，**DBMS**根据授权定义进行检查，以决定是否执行操作请求

4.2.4 授权(Authorization)与回收

一. 授权权限GRANT

GRANT语句的一般格式：

GRANT <权限>[,<权限>]...

[ON <对象名>]

TO <用户>[,<用户>]...

[WITH GRANT OPTION];

谁定义？DBA和表的建立者（即表的属主）

GRANT功能：将对指定操作对象的指定操作权限授予指定的用户。

4.2.4 授权(Authorization)与回收

WITH GRANT OPTION子句

- 指定了WITH GRANT OPTION子句:

  获得某种权限的用户还可以把这种权限再授予别的用户。

- 没有指定WITH GRANT OPTION子句:

  获得某种权限的用户只能使用该权限，不能传播该权限。

4.2.4 授权(Authorization)与回收

- 用户的权限

- 建表（CREATETAB）的权限:属于DBA
- DBA授予-->普通用户
- 基本表或视图的属主拥有对该表或视图的一切操作权限
- 接受权限的用户:

  一个或多个具体用户

  PUBLIC（全体用户）

4.2.4 授权(Authorization)与回收

例题

例1 把查询Student表权限授给用户U1

```
GRANT SELECT
    ON TABLE Student
    TO U1;
```

4.2.4 授权(Authorization)与回收

例题

例2 把对Student表的全部权限授予用户U2和U3

```
GRANT ALL PRIVILEGES
      ON TABLE Student
      TO U2,U3;
```

4.2.4 授权(Authorization)与回收

例题

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

例4 把查询Student表和修改学生姓名的权限授给用户U4

```
GRANT UPDATE(Sname), SELECT
ON TABLE Student
TO U4;
```

4.2.4 授权(Authorization)与回收

例题

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;
```

4.2.4 授权(Authorization)与回收

- **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。

U5--> U6--> U7

二. 收回权限REVOKE

REVOKE语句的一般格式为：

REVOKE <权限>[,<权限>]...
[ON <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

4.2.4 授权(Authorization)与回收

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;
```

4.2.4 授权(Authorization)与回收

例题

例9 收回所有用户对表SC的查询权限

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

4.2.4 授权(Authorization)与回收

例题

例10 把用户U5对SC表的INSERT权限收回

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

4.2.4 授权(Authorization)与回收

- **权限的级联收回**

系统将收回直接或间接从U5处获得的对SC表的INSERT权限:

-->U5--> U6--> U7

收回U5、U6、U7获得的对SC表的INSERT权限:

<--U5<-- U6<-- U7

三. 创建数据库模式的权限

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 | RESOURSE | CONNECT];
```

4.2.4 授权(Authorization)与回收

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

4.2.4 授权(Authorization)与回收

小结：SQL灵活的授权机制

1. DBA拥有对数据库中所有对象的所有权限，并可以根据应用的需要将不同的权限授予不同的用户。
2. 用户对自己建立的基本表和视图拥有全部的操作权限，并且可以用GRANT语句把其中某些权限授予其他用户。
3. 被授权的用户如果有“继续授权”的许可，还可以把获得的权限再授予其他用户。
4. 所有授予出去的权力在必要时又都可以用REVOKE语句收回。

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。

1. 角色的创建

```
CREATE ROLE <角色名>
```

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的角色或用户

```
GRANT <角色1> [,<角色2>]...
TO <角色3> [,<用户1>]...
[WITH ADMIN OPTION]
```

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>]...
```

4.2.6 强制存取控制(MAC)方法

- 什么是强制存取控制
  - 强制存取控制 （MAC）是指系统为保证更高程度的安全性，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。
  - MAC 不是用户能直接感知或进行控制的。
  - MAC 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
- 主体与客体
  - 在MAC中，DBMS所管理的全部实体被分为主体和客体两大类
  - 主体是系统中的活动实体
    - DBMS所管理的实际用户
    - 代表用户的各进程
  - 客体是系统中的被动实体，是受主体操纵的
    - 文件
    - 基表
    - 索引
    - 视图

4.2.6 强制存取控制(MAC)方法

- 敏感度标记
  - 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（**Label**）
  - 敏感度标记分成若干级别
    - 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（**Clearance Level**）
  - 客体的敏感度标记称为密级（**Classification Level**）
  - **MAC**机制就是通过对比主体的**Label**和客体的**Label**，最终确定主体是否能够存取客体

4.2.6 强制存取控制(MAC)方法

- 强制存取控制规则
  - 当某一用户（或某一主体）以标记label注册入系统时，系统要求他对任何客体的存取必须遵循下面两条规则：

（1）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；

（2）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。

  - 用户可为写入的数据对象赋予高于自己的许可证级别的密级
  - 一旦数据被写入，该用户自己也不能再读该数据对象了。
- 修正规则：

仅当主体的许可证级别等于客体的密级时，该主体才能写相应的客体。

- 规则的共同点：
  - 禁止了拥有高许可证级别的主体更新低密级的数据对象

4.2.6 强制存取控制(MAC)方法

- 强制存取控制的特点
  - 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
  - 从而提供了更高级别的安全性

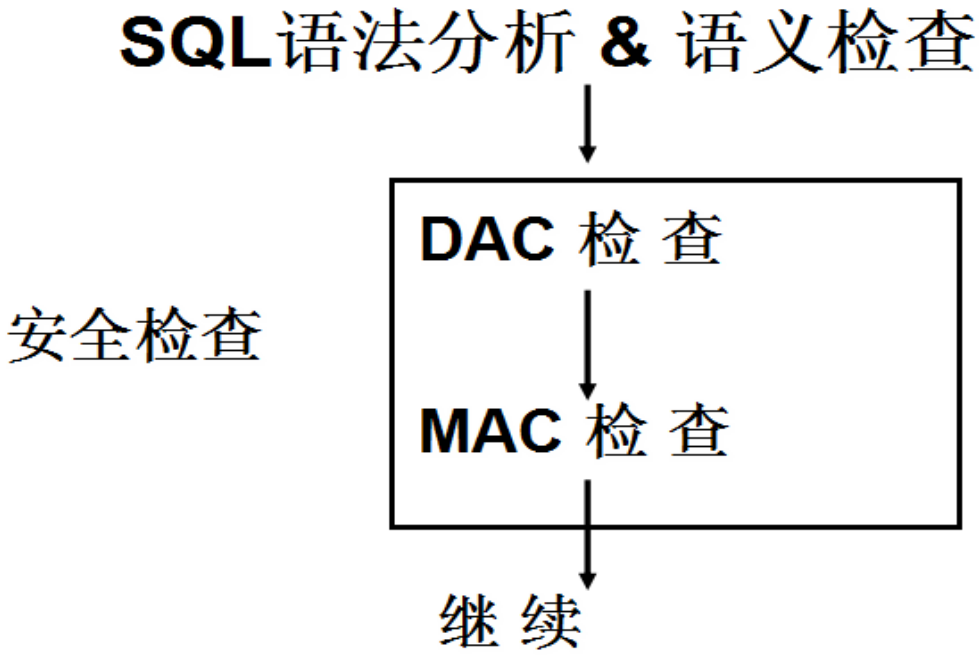
4.2.6 强制存取控制(MAC)方法

DAC与MAC

- DAC与MAC共同构成DBMS的安全机制
  - 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
- 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。

4.2.6 强制存取控制(MAC)方法

DAC + MAC安全检查示意图



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，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。

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
- 视图上面再进一步定义存取权限
- 间接实现了支持存取谓词的用户权限定义

例：王平只能检索计算机系学生的信息

```
//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';
```

在视图上进一步定义存取权限

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

课堂作业

Student

学 号 Sno	姓 名 Sname	性 别 Ssex	年 龄 Sage	所在系 Sdept	管理员 Sadmin
95001	李勇	男	20	CS	U2
95002	刘晨	女	19	IS	U3
95003	王敏	女	18	MA	U2
95004	张立	男	19	IS	U1

题目：请用视图和DAC（自主存取控制）实现以下功能：
只允许某个管理员查询属于自己管理的学生的信息

4.4 审计(Audit)

- 什么是审计

  启用一个专用的审计日志（**Audit Log**）将用户对数据库的所有操作记录在上面。

  DBA可以利用审计日志中的追踪信息找出非法存取数据的人

  C2以上安全级别的**DBMS**必须具有审计功能。

4.4 审计(Audit)

- 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。

- 强制性机制:

  用户识别和鉴定、存取控制、视图

- 预防监测手段:

  审计技术

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（术语为明文，**Plain text**）变换为不可直接识别的格式（术语为密文，**Cipher text**），不知道解密算法的人无法获知数据的内容

- 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**）

4.5 数据加密

- **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

  有些数据库产品本身未提供加密程序，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

4.6 统计数据库安全性

- 统计数据库的特点

  允许用户查询聚集类型的信息（例如合计、平均值等）

  不允许查询单个记录信息

  例：允许查询“程序员的平均工资是多少？”

  不允许查询“程序员张勇的工资？”

- 统计数据库中特殊的安全性问题

  隐蔽的信息通道

  从合法的查询中推导出不合法的信息

4.6 统计数据库安全性

例1：下面两个查询都是合法的：

1. 本公司共有多少女高级程序员？
2. 本公司女高级程序员的工资总额是多少？

如果第一个查询的结果是“1”，

那么第二个查询的结果显然就是这个程序员的工资数。

规则1：任何查询至少要涉及N(N足够大)个以上的记录

4.6 统计数据库安全性

例2：用户A发出下面两个合法查询：

1．用户A和其他N个程序员的工资总额是多少？

2．用户B和其他N个程序员的工资总额是多少？

若第一个查询的结果是X，第二个查询的结果是Y，

由于用户A知道自己的工资是Z，

那么他可以计算出用户B的工资=Y-(X-Z)。

原因：两个查询之间有很多重复的数据项

规则2：任意两个查询的相交数据项不能超过M个

4.6 统计数据库安全性

可以证明，在上述两条规定下，如果想获知用户B的工资额

A至少需要进行1+(N-2)/M次查询

规则3：任一用户的查询次数不能超过1+(N-2)/M

如果两个用户合作查询就可以使这一规定失效

4.6 统计数据库安全性

数据库安全机制的设计目标：

试图破坏安全的人所花费的代价 >> 得到的利益

附录：本章常用SQL语句

创建登录名
exec sp_addlogin 'L2','123456','test2'
exec sp_addlogin 'L2','123456'
exec sp_addlogin 'L2',
create login L4 with password='123456’
为某个数据库创建用户名
use test2
exec sp_adduser 'L2','U2’
删除用户
exec sp_dropuser 'U2’
删除登录名
exec sp_droplogin 'L2’
授权
grant select on course to U2
grant select on student to U2 with grant option
grant select on student to public

附录：本章常用SQL语句

回收权限

```
revoke select on course from U2
select * from test2.dbo.student
execute as user='U2'
select * from test2.dbo.course
revert
select SYSTEM_USER
grant select on course to U2
revoke select on student from U2 CASCADE
```

角色

```
create role R1
exec sp_droprole R2
grant select on course to R1
exec sp_addrolemember 'R1', 'U2'
exec sp_droprolemember 'R1', 'U2'
exec sp_addrolemember 'R1', 'R2'
exec sp_droprolemember ‘R1’, ‘R2’
```

附录：主讲教师

主讲教师：林子雨

单位：厦门大学计算机科学系
E-mail: ziyulin@xmu.edu.cn
个人网页：http://www.cs.xmu.edu.cn/linziyu
数据库实验室网站：http://dblab.xmu.edu.cn

扫一扫访问个人主页

林子雨，男，1978年出生，博士（毕业于北京大学），现为厦门大学计算机科学系助理教授（讲师），曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革局副局长。中国高校首个“数字教师”提出者和建设者，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度厦门大学奖教金获得者。主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文。作为项目负责人主持的科研项目包括1项国家自然科学青年基金项目(No.61303004)、1项福建省自然科学青年基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，同时，作为课题负责人完成了国家发改委城市信息化重大课题、国家物联网重大应用示范工程区域试点泉州市工作方案、2015泉州市互联网经济调研等课题。编著出版中国高校第一本系统介绍大数据知识的专业教材《大数据技术原理与应用》并成为畅销书籍，编著并免费网络发布40余万字中国高校第一本闪存数据库研究专著《闪存数据库概念与技术》；主讲厦门大学计算机系本科生课程《数据库系统原理》和研究生课程《分布式数据库》《大数据技术基础》。具有丰富的政府和企业信息化培训经验，曾先后给中国移动通信集团公司、福州马尾区政府、福建省物联网科学研究院、石狮市物流协会、厦门市物流协会、福建龙岩卷烟厂等多家单位和企业开展信息化培训，累计培训人数达2000人以上。

附录：课程助教

助教：谢荣东

单位：厦门大学计算机科学系数据库实验室2014级硕士研究生
E-mail: xrdxmu@sina.com

助教：薛倩

单位：厦门大学计算机科学系数据库实验室2015级硕士研究生
E-mail: xueqian_victoria@163.com

附录：班级网站

林子雨主讲《数据库系统原理》2016班级主页

http://dblab.xmu.edu.cn/post/5550/

扫一扫访问班级网站
支持手机浏览


Thank You!