数据库管理1

授课教师：秦建斌

邮箱：qinjianbin@szu.edu.cn

深圳大学 计算机与软件学院

目录

- 数据库安全（对应第4章）
- 数据库完整性（对应第5章）
- 并发控制（对应第11章）

选择

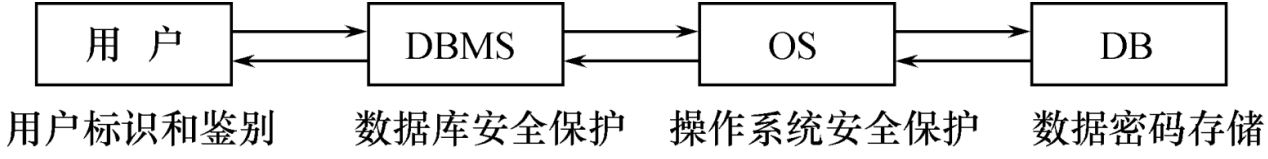
大题

数据库管理

- Database System = Database + DBMS
- DBMS （ **database management system** ）数据库管理系统
- 数据库安全（对应第4章）
- 数据库完整性 （对应第5章）

数据库安全性控制

- 数据库系统中的数据共享不能是无条件的共享
- 计算机系统中，安全措施是一级一级层层设置



数据库安全性控制的常用方法

- 用户标识和鉴定
- 存取控制
- 视图
- 审计
- 数据加密

数据库安全性控制

- 用户身份鉴别 （Identification & Authentication）
  - 系统提供的最外层安全保护措施
  - 用户标识：由用户名和用户标识号组成

    （用户标识号在系统整个生命周期内唯一）

数据库安全性控制

- 存取控制
  - 定义用户权限，并将用户权限登记到数据字典中
- 通过 SQL 的GRANT 语句和REVOKE 语句实现
- 用户权限组成
  - 数据对象
  - 操作类型
- 定义存取权限称为授权

数据库安全性控制

◘关系数据库系统中存取控制对象

对象类型	对象	操作类型
数据库模式	模式	CREATE SCHEMA 创建数据库.
	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

关系数据库系统中的存取权限

授权：授予与回收

1.GRANT

- GRANT语句的一般格式：

  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

例题

[例4.1] 把查询Student表权限授给用户U1

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

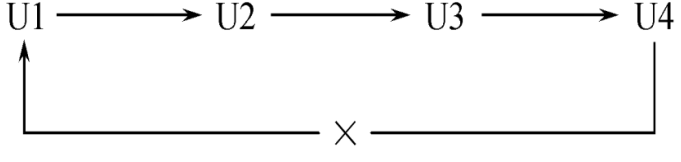
GRANT（续）

- 发出GRANT：
  - 数据库管理员
  - 数据库对象创建者（即属主Owner）
  - 拥有该权限的用户
- 按受权限的用户
  - 一个或多个具体用户
  - PUBLIC（即全体用户）

WITH GRANT OPTION子句

- WITH GRANT OPTION子句:
  - 指定：可以再授予
  - 没有指定：不能传播

- 不允许循环授权



例题（续）

[例4.2] 把对Student表和Course表的全部权限授予用户U2和U3

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

例题（续）

[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;
```

例题（续）

[例4.5] 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;
```

授权：授予与回收（续）

2.REVOKE

- 授予的权限可以由数据库管理员或其他授权者用REVOKE语句收回
- REVOKE语句的一般格式为：

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>];

REVOKE（续）

[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;
```

REVOKE（续）

[例4.9] 收回所有用户对表SC的查询权限

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

小结: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
  - 根据实际情况不同的权限授予不同的用户
- 用户：
  - 拥有自己建立的对象的全部的操作权限
  - 可以使用GRANT，把权限授予其他用户
- 被授权的用户
  - 如果具有“继续授权”的许可，可以把获得的权限再授予其他用户
- 所有授予出去的权力在必要时又都可用REVOKE语句收回

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
- 间接地实现支持存取谓词的用户权限定义

视图机制（续）

[例4.14] 建立计算机系学生的视图，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

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';
```

视图机制（续）

[例4.14] 建立计算机系学生的视图，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

在视图上进一步定义存取权限

```
GRANT  SELECT
ON  CS_Student
TO 王平;

GRANT ALL PRIVILIGES
ON  CS_Student
TO  张明;
```

视图机制（续）

视图可以是一个具体信息定义:

```
CREATE VIEW zwu (name, address, salary)
AS
    SELECT Name, Address, Salary
    FROM personnel
    WHERE personnel.Name = "Zimin Wu";

GRANT select ON zwu TO Zimin;
```

也可以是一个统计信息

```
CREATE VIEW salary_summary (total_salary)
AS SELECT sum(Salary)
FROM personnel;

GRANT select ON salary_summary TO bank_manager;
```

A actual scenario

- Suppose a teacher asks the teaching assistant (TA)to key in the scores of those students who have taken the re-sit exam of Database(2701) for him in a database system. What solution should the teacher have in order to protect the privacy of each student? (Hint: show only necessary information and grant proper privileges to the TA)
- Student information Table
- Score{SID,CID,score,resit,sname} 视图

数据库完整性

- 数据库的完整性
  - 数据的正确性
    - 是指数据是符合现实世界语义，反映了当前实际状况的
  - 数据的相容性
    - 是指数据库同一对象在不同关系表中的数据是符合逻辑的

例如，

- 学生的学号必须唯一
- 性别只能是男或女
- 本科学生年龄的取值范围为14~50的整数
- 学生所选的课程必须是学校开设的课程，学生所在的院系必须是学校已成立的院系
- 等

数据库完整性

1 实体完整性

2 参照完整性

3 用户定义的完整性

1 实体完整性定义

- 关系模型的实体完整性
  - CREATE TABLE中用PRIMARY KEY定义
- 单属性构成的码有两种说明方法
  - 定义为列级约束条件
  - 定义为表级约束条件
- 对多个属性构成的码只有一种说明方法
  - 定义为表级约束条件

实体完整性定义（续）

[例5.1] 将Student表中的Sno属性定义为码

（1）在列级定义主码

```
CREATE TABLE Student
 ( Sno  CHAR(9)  PRIMARY KEY,
   Sname  CHAR(20) NOT NULL,
   Ssex  CHAR(2),
   Sage  SMALLINT,
   Sdept  CHAR(20)
 );
```

实体完整性定义（续）

（2）在表级定义主码

```
CREATE TABLE Student
    ( Sno  CHAR(9),
       Sname  CHAR(20) NOT NULL,
      Ssex  CHAR(2),
      Sage  SMALLINT,
      Sdept  CHAR(20),
      PRIMARY KEY (Sno)
     );
```

实体完整性定义（续）

[例5.2] 将SC表中的Sno，Cno属性组定义为码

```
CREATE TABLE SC
    ( Sno   CHAR(9)  NOT NULL,
      Cno  CHAR(4)  NOT NULL,
      Grade    SMALLINT,
      PRIMARY KEY (Sno,Cno)    /*只能在表级定义主码*/
    );
```

实体完整性检查和违约处理

- 插入或对主码列进行更新操作时，关系数据库管理系统按照实体完整性规则自动进行检查。包括：
  - 检查主码值是否唯一，如果不唯一则拒绝插入或修改
  - 检查主码的各个属性是否为空，只要有一个为空就拒绝插入或修改

2 参照完整性定义

- 关系模型的参照完整性定义
  - 在CREATE TABLE中用FOREIGN KEY短语定义哪些列为外码
  - 用REFERENCES短语指明这些外码参照哪些表的主码

参照完整性定义（续）

例如，关系SC中（Sno，Cno）是主码。Sno，Cno分别参照Student表的主码和Course表的主码

[例5.3]定义SC中的参照完整性

```
CREATE TABLE SC
( Sno    CHAR(9)  NOT NULL,
  Cno    CHAR(4)  NOT NULL,
  Grade  SMALLINT,
  PRIMARY KEY (Sno, Cno),  /*在表级定义实体完整性*/
  FOREIGN KEY (Sno) REFERENCES Student(Sno),
  /*在表级定义参照完整性*/
  FOREIGN KEY (Cno) REFERENCES Course(Cno)
  /*在表级定义参照完整性*/
);
```

参照完整性检查和违约处理

- 一个参照完整性将两个表中的相应元组联系起来
- 对被参照表和参照表进行增删改操作时有可能破坏参照完整性，必须进行检查

被参照表（例如 student）	参照表（例如 SC）	违约处理
可能破坏参照完整性	← 插入元组	拒绝
可能破坏参照完整性	← 修改主码值	拒绝
删除元组 →	可能破坏参照完整性	拒绝/级连删除/设置为空值
修改主码值 →	可能破坏参照完整性	拒绝/级连删除/设置为空值

参照完整性检查和违约处理（续）

[例5.4] 显式说明参照完整性的违约处理示例

```
CREATE TABLE SC
( Sno   CHAR(9)  NOT NULL，
  Cno   CHAR(4)  NOT NULL，
  Grade  SMALLINT,
  PRIMARY KEY(Sno,Cno)，
  FOREIGN KEY (Sno) REFERENCES Student(Sno)
      ON DELETE CASCADE       /*级联删除SC表中相应的元组*/
   ON UPDATE CASCADE,     /*级联更新SC表中相应的元组*/
  FOREIGN KEY (Cno) REFERENCES Course(Cno)
   ON DELETE NO ACTION
     /*当删除course 表中的元组造成了与SC表不一致时拒绝删除*/
   ON UPDATE CASCADE
        /*当更新course表中的cno时，级联更新SC表中相应的元组*/
);
```

3 用户定义的完整性

- 用户定义的完整性是：针对某一具体应用的数据必须满足的语义要求
- 关系数据库管理系统提供了定义和检验用户定义完整性的机制，不必由应用程序承担

属性上约束条件的定义

- CREATE TABLE时定义属性上的约束条件
  - 列值非空（NOT NULL）
  - 列值唯一（UNIQUE）
  - 检查列值是否满足一个条件表达式（CHECK）

属性上约束条件的定义（续）

（1）不允许取空值

[例5.5] 在定义SC表时，说明Sno、Cno、Grade属性不允许取空值。

```
CREATE TABLE SC
( Sno CHAR(9)  NOT NULL,
  Cno CHAR(4)  NOT NULL,
  Grade  SMALLINT NOT NULL,
  PRIMARY KEY (Sno, Cno),
   …
  /* 如果在表级定义实体完整性，隐含了Sno，Cno不允许取空值，则在
  列级不允许取空值的定义 可以不写 * /
);
```

属性上约束条件的定义（续）

（2）列值唯一

[例5.6]建立部门表DEPT，要求部门名称Dname列取值唯一，部门编号Deptno列为主码

```
CREATE TABLE DEPT
   (   Deptno  NUMERIC(2),
       Dname  CHAR(9)  UNIQUE NOT NULL,
                          /*要求Dname列值唯一, 并且不能取空值*/
       Location  CHAR(10),
       PRIMARY KEY (Deptno)
    );
```

属性上约束条件的定义（续）

（3）用CHECK短语指定列值应该满足的条件

[例5.7] Student表的Ssex只允许取“男”或“女”。

```
CREATE TABLE Student
    ( Sno  CHAR(9) PRIMARY KEY,
      Sname CHAR(8) NOT NULL,
      Ssex  CHAR(2)  CHECK （Ssex IN （‘男’,’女’））,
                              /*性别属性Ssex只允许取'男'或'女' */
      Sage  SMALLINT,
      Sdept  CHAR(20)
    );
```

属性上约束条件的定义（续）

[例5.8] SC表的Grade的值应该在0和100之间。

```
CREATE TABLE  SC
    (  Sno     CHAR(9) ,
       Cno     CHAR(4),
       Grade   SMALLINT CHECK (Grade>=0 AND Grade <=100),
                                /*Grade取值范围是0到100*/
       PRIMARY KEY (Sno,Cno),
       FOREIGN KEY (Sno) REFERENCES Student(Sno),
       FOREIGN KEY (Cno) REFERENCES Course(Cno)
      );
```