



## 第三章 关系数据库标准语言 SQL (续3)



# 第三章 关系数据库标准语言SQL

---

**3.1 SQL概述**

**3.2 数据定义**

**3.3 查询**

**3.4 数据更新**

**3.5 视图**

**3.6 数据控制**

**3.7 嵌入式SQL**

**3.8 小结**



## 3.6 数据控制

---

- 概述
- 3.6.1 授权
- 3.6.2 收回权限
- 小结



# 概述

---

数据控制亦称为数据保护，包括数据的：

- 安全性控制
- 完整性控制
- 并发控制
- 恢复



# SQL语言的数据控制功能

---

SQL语言提供了数据控制功能，能够在一定程度上保证数据库中数据的完全性、完整性，并提供了一定的并发控制及恢复能力。



# 1. 完整性

---

- 数据库的完整性是指数据库中数据的正确性与相容性。
- SQL语言定义完整性约束条件
  - CREATE TABLE语句
  - ALTER TABLE语句
    - 码
    - 取值唯一的列
    - 参照完整性
    - 其他约束条件



## 2. 并发控制

---

- 并发控制: 当多个用户并发地对数据库进行操作时, 对他们加以控制、协调, 以保证并发操作正确执行, 保持数据库的一致性。
- SQL语言并发控制能力:  
提供事务、事务开始、事务结束、提交等概念



### 3. 恢复

---

- 恢复: 当发生各种类型的故障导致数据库处于不一致状态时, 将数据库恢复到一致状态的功能。
- SQL语言恢复功能:  
提供事务回滚、重做等概念  
(UNDO、REDO)





## 4. 安全性

---

- 安全性：保护数据库，防止不合法的使用所造成的数据泄露和破坏。

- 保证数据安全性的主要措施

**存取控制**：控制用户只能存取他有权存取的数据，规定不同用户对于不同数据对象所允许执行的操作。



# DBMS实现数据安全性保护的过程

---

- ❑ 用户或DBA把授权决定告知系统
- ❑ SQL的GRANT和REVOKE
- ❑ DBMS把授权的结果存入数据字典
- ❑ 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



## 安全性(续)

---

### □ 谁定义？

DBA和表的建立者（即表的属主）

### □ 如何定义？

**SQL语句：**

**GRANT**

**REVOKE**



- GRANT语句的一般格式：

GRANT <权限>[, <权限>]...

[ON <对象类型> <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）
- 功能：将对指定操作对象的指定操作权限授予指定的用户。



## (1) 操作权限

对 象	对象类型	操 作 权 限
属性列	TABLE	SELECT, INSERT, UPDATE DELETE, ALL PRIVIEGES
视 图	TABLE	SELECT, INSERT, UPDATE DELETE, ALL PRIVIEGES
基本表	TABLE	SELECT, INSERT, UPDATE DELETE ALTER, INDEX, ALL PRIVIEGES
数据库	DATABASE	CREATETAB



## (2) 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）



### (3) 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



## 例 题

---

例1 把查询Student表权限授给用户U1

**GRANT SELECT**

**ON TABLE Student**

**TO U1;**





## 例 题（续）

---

例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 把查询Student表和修改学生学号的权限授  
给用户U4

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



## 例 题（续）

---

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;



## 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：

```
GRANT INSERT ON TABLE SC TO U6  
WITH GRANT OPTION;
```

同样，U6还可以将此权限授予U7：

```
GRANT INSERT ON TABLE SC TO U7;
```

但U7不能再传播此权限。

U5--> U6--> U7



## 例 题（续）

---

例6 DBA把在数据库S\_C中建立表的权限授予用户  
U8

```
GRANT CREATE TABLE  
TO U8;
```



## 3.6 数据控制

---

- 概述
- 3.6.1 授权
- 3.6.2 收回权限
- 小结



# SQL收回权限的功能

---

- 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





# 例 题

---

例7 把用户U4修改学生学号的权限收回。

REVOKE UPDATE(Sno)

ON TABLE Student

FROM U4;



## 例 题（续）

---

例8 收回所有用户对表SC的查询权限。

REVOKE **SELECT**

ON TABLE SC

FROM **PUBLIC**;



## 例 题（续）

---

例9 把用户U5对SC表的INSERT权限收回

REVOKE INSERT

ON TABLE SC

FROM U5;



## 权限的级联回收

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**



- ❑ 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。
- ❑ 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。
- ❑ 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。
- ❑ 所有授予出去的权力在必要时又都可以用REVOKE语句收回。