

## 第3章 关系数据库标准语言 SQL

《概论》第3章详细介绍关系数据库语言 SQL。SQL 语言是关系数据库的标准语言。它内容十分丰富,是学习关系数据库概念和技术的重要组成部分。

### 3.1 基本知识点

关系模型和关系数据库是《概论》的重点,第3章又是重点中的重点,是全书篇幅最长的一章,因为关系数据库系统的主要功能是通过 SQL 语言来实现的。

① **需要了解的**:SQL 语言的发展过程,从而进一步了解关系数据库技术和关系数据库管理系统(RDBMS)产品的发展过程。

② **需要牢固掌握的**:掌握 SQL 语言的特点和优点,体会面向过程的语言和 SQL 语言的区别。体会关系数据库系统为数据库应用系统的开发提供良好环境,减轻了用户负担,提高用户生产率的原因。

③ **需要举一反三的**:熟练正确地使用 SQL 语言完成对数据库的查询、插入、删除、更新操作,特别是各种各样的查询,掌握 SQL 语言强大的查询功能。

在完成具体的 SQL 语句时,希望读者能有意识地 and 关系代数、关系演算等语言进行比较,了解它们各自的特点。

④ **难点**:本章的难点在于用 SQL 语言正确完成复杂查询。因此在学习的过程中一定要多练习,要在某一个 RDBMS 产品上进行实际运行,检查自己的答案,检测查询的结果是否正确。只有通过大量练习才能真正达到举一反三的熟练程度。

### 3.2 习题解答和解析

1. 试述 SQL 的特点。

答:

① **综合统一**。SQL 语言集数据定义语言(DDL)、数据操纵语言(DML)和数据控制语言

(DCL)的功能于一体。

② 高度非过程化。用 SQL 语言进行数据操作,只要提出“做什么”,而无须指明“怎么做”,因此无须了解存取路径。存取路径的选择以及 SQL 语句的操作过程由系统自动完成。

③ 面向集合的操作方式。SQL 语言采用集合操作方式,不仅操作对象、查找结果可以是元组的集合,而且一次插入、删除、更新操作的对象也可以是元组的集合。

④ 以同一种语法结构提供两种使用方式。SQL 语言既是自含式语言,又是嵌入式语言。作为自含式语言,它能够独立地用于联机交互的使用方式,也能够嵌入到高级语言程序中,供程序员设计程序时使用。

⑤ 语言简洁,易学易用。

解析:

详细内容可参考《概论》第 3.1.2 小节。注意不要仅仅背这些特点,关键是要通过具体的练习,通过使用 SQL 语句来理解这些特点。

2. 说明在 DROP TABLE 时,RESTRICT 和 CASCADE 的区别。

答:

RESTRICT 表示表的删除是有限制条件的。要删除的基本表不能被其他表的约束所引用,不能有视图,不能有触发器,不能有存储过程或函数等。如果存在这些依赖该表的对象,则表不能被删除。

CASCADE 表示表的删除没有限制条件,在删除基本表的同时,相关的依赖对象(如视图)都将被删除。

3. 有两个关系  $S(A,B,C,D)$  和  $T(C,D,E,F)$ ,写出与下列查询等价的 SQL 表达式:

(1)  $\sigma_{A=10}(S)$ ; (2)  $\Pi_{A,B}(S)$ ; (3)  $S \bowtie T$ ; (4)  $S \bowtie_{S.C=T.C} T$ ; (5)  $S \bowtie_{A<E} T$ ; (6)  $\Pi_{C,D}(S) \times T$ 。

答:

(1) SELECT \* FROM S WHERE A = 10

(2) SELECT DISTINCT A,B FROM S

(3) SELECT A,B,S.C,S.D,E,F FROM S,T WHERE S.C=T.C AND S.D=T.D

(4) SELECT A,B,S.C,S.D,T.C,T.D,E,F FROM S,T WHERE S.C=T.C

(5) SELECT A,B,S.C,S.D,T.C,T.D,E,F FROM S,T WHERE A < E

(6) SELECT S1.C,S1.D,T.C,T.D,E,F FROM T,(select DISTINCT C,D FROM S) AS S1

4. 用 SQL 语句建立第 2 章习题 6 中的 4 个表;针对建立的 4 个表用 SQL 语言完成第 2 章习题 6 中的查询。

答:

建 S 表:

S(SNO,SNAME,STATUS,CITY);

CREATE TABLE S

```
(SNO CHAR(3),  
  SNAME CHAR(10),  
  STATUS CHAR(2),  
  CITY CHAR(10) );
```

建 P 表:

```
P(PNO,PNAME,COLOR,WEIGHT);  
CREATE TABLE P  
(PNO CHAR(3),  
  PNAME CHAR(10),  
  COLOR CHAR(4),  
  WEIGHT INT);
```

建 J 表:

```
J(JNO,JNAME,CITY);  
CREATE TABLE J  
(JNO CHAR(3),  
  JNAME CHAR(10),  
  CITY CHAR(10) );
```

建 SPJ 表:

```
SPJ(SNO,PNO,JNO,QTY);  
CREATE TABLE SPJ  
(SNO CHAR(3),  
  PNO CHAR(3),  
  JNO CHAR(3),  
  QTY INT);
```

解析:

读者完成建表后首先插入若干数据,如第2章第6题。

① 求供应工程 J1 零件的供应商号码 SNO。

```
SELECT SNO  
FROM SPJ  
WHERE JNO='J1';
```

② 求供应工程 J1 零件 P1 的供应商号码 SNO。

```
SELECT SNO  
FROM SPJ  
WHERE JNO='J1' AND PNO='P1';
```

③ 求供应工程 J1 零件为红色的供应商号码 SNO。

```
SELECT SNO          /* 这是嵌套查询 */
```

```

FROM SPJ
WHERE JNO='J1'
      AND PNO IN          /* 找出红色零件的零件号码 PNO */
      (SELECT PNO
       FROM P              /* 从 P 表中找 */
       WHERE COLOR='红');

```

或

```

SELECT SNO
FROM SPJ,P              /* 这是两表连接 */
WHERE JNO='J1'          /* 这是复合条件连接查询 */
      AND SPJ.PNO=P.PNO AND COLOR='红';

```

④ 求没有使用天津供应商生产的红色零件的工程号 JNO。

解析:

读者可以对比第 2 章习题 5 中用 ALPHA 语言来完成该查询的解答:

```

GET W (J.JNO): ¬∃SPJX( SPJX .JNO=J.JNO ∧
                      ∃SX ( SX.SNO=SPJX .SNO ∧SX .CITY='天津' ∧
                      ∃PX ( PX .PNO=SPJX .PNO ∧PX .COLOR='红' ))

```

如果理解了有关该题的解析说明,那么本题的解答可以看成是把关系演算用 SQL 来表示的过程。

```

SELECT JNO              /* 这种解法是使用多重嵌套查询 */
FROM J                  /* 注意:从 J 表入手,以包含那些 */
WHERE NOT EXISTS        /* 尚未使用任何零件的工程号 */
  (SELECT *
   FROM SPJ
   WHERE SPJ.JNO=J.JNO
        AND SNO IN
        (SELECT SNO          /* 天津供应商的 SNO */
         FROM S
         WHERE CITY='天津')
        AND PNO IN          /* 红色零件的 PNO */
        (SELECT PNO
         FROM P
         WHERE COLOR='红'));

```

或

```

SELECT JNO
FROM J

```

**WHERE NOT EXISTS**

```
(SELECT *
FROM SPJ,S,P          /* 这里的子查询是一个多表连接 */
WHERE SPJ.JNO=J.JNO AND SPJ.SNO=S.SNO
      AND SPJ.PNO=P.PNO AND S.CITY='天津'
      AND P.COLOR='红');
```

⑤ 求至少用了供应商 S1 所供应的全部零件的工程号 JNO(类似于《概论》书 3.4.3 小节 [例 3.63])。

解析:

本查询的解析可以参考第 2 章第 5 题,用 ALPHA 语言的逻辑蕴涵来表达。

上述查询可以抽象为:要求这样的工程  $x$ ,使  $(\forall y) p \rightarrow q$  为真。即:

对于所有的零件  $y$ ,满足逻辑蕴涵  $p \rightarrow q$ :

$p$  表示谓词:“供应商 S1 供应了零件  $y$ ”

$q$  表示谓词:“工程  $x$  选用了零件  $y$ ”

即:只要“供应商 S1 供应了零件  $y$ ”为真,则“工程  $x$  选用了零件  $y$ ”为真。

逻辑蕴涵可以转换为等价形式:

$$(\forall y)p \rightarrow q \equiv \neg(\exists y(\neg(p \rightarrow q))) \equiv \neg(\exists y(\neg(\neg p \vee q))) \equiv \neg\exists y(p \wedge \neg q)$$

它所表达的语义为:不存在这样的零件  $y$ ,供应商 S1 供应了  $y$ ,而工程  $x$  没有选用  $y$ 。用 SQL 语言表示如下:

```
SELECT DISTINCT JNO
FROM SPJ SPJZ
WHERE NOT EXISTS      /* 这是一个相关子查询 */
  (SELECT *            /* 父查询和子查询均引用了 SPJ 表 */
   FROM SPJ SPJX      /* 用别名 SPJZ、SPJX 将父查询 */
   WHERE SNO='S1'     /* 与子查询中的 SPJ 表区分开 */
   AND NOT EXISTS
     (SELECT *         /* 用别名 SPJY 与父查询 */
      FROM SPJ SPJY    /* 中的 SPJ 表区分开 */
      WHERE SPJY.PNO=SPJX.PNO
            AND SPJY.JNO=SPJZ.JNO));
```

可以把 SQL 语言和关系代数、ALPHA 语言、QBE 语言进行比较,体会各种语言的优点。

5. 针对习题 3 中的 4 个表,试用 SQL 语言完成以下各项操作:

答:

① 找出所有供应商的姓名和所在城市。

```
SELECT SNAME,CITY
```

```
FROM S;
```

- ② 找出所有零件的名称、颜色、重量。

```
SELECT PNAME,COLOR,WEIGHT  
FROM P;
```

- ③ 找出使用供应商 S1 所供应零件的工程号码。

```
SELECT JNO  
FROM SPJ  
WHERE SNO='S1';
```

- ④ 找出工程项目 J2 使用的各种零件的名称及其数量。

```
SELECT P.PNAME,SPJ.QTY  
FROM P,SPJ  
WHERE P.PNO=SPJ.PNO  
AND SPJ.JNO='J2';
```

- ⑤ 找出上海厂商供应的所有零件号码。

```
SELECT DISTINCT PNO  
FROM SPJ  
WHERE SNO IN  
(SELECT SNO  
FROM S  
WHERE CITY='上海');
```

- ⑥ 找出使用上海产的零件的工程名称。

```
SELECT JNAME  
FROM J,SPJ,S  
WHERE J.JNO=SPJ.JNO  
AND SPJ.SNO=S.SNO  
AND S.CITY='上海';
```

或

```
SELECT JNAME  
FROM J  
WHERE JNO IN  
(SELECT JNO  
FROM SPJ,S  
WHERE SPJ.SNO=S.SNO  
AND S.CITY='上海');
```

- ⑦ 找出没有使用天津产的零件的工程号码。

```
SELECT JNO
```

```
FROM J
WHERE NOT EXISTS
  (SELECT *
   FROM SPJ
   WHERE SPJ.JNO=J.JNO
    AND SNO IN
      (SELECT SNO
       FROM S
       WHERE CITY='天津'));
```

或

```
SELECT JNO
FROM J
WHERE NOT EXISTS
  (SELECT *
   FROM SPJ,S
   WHERE SPJ.JNO=J.JNO
    AND SPJ.SNO=S.SNO
    AND S.CITY='天津');
```

- ⑧ 把全部红色零件的颜色改成蓝色。

```
UPDATE P
SET COLOR='蓝'
WHERE COLOR='红';
```

- ⑨ 由 S5 供给 J4 的零件 P6 改为由 S3 供应,请作必要的修改。

```
UPDATE SPJ
SET SNO='S3'
WHERE SNO='S5'
  AND JNO='J4'
  AND PNO='P6';
```

- ⑩ 从供应商关系中删除 S2 的记录,并从供应情况关系中删除相应的记录。

```
DELETE
FROM SPJ
WHERE SNO='S2';

DELETE
FROM S
WHERE SNO='S2';
```

解析:

注意删除顺序,应该先从 SPJ 表中删除供应商 S2 所供应零件的记录,然后从 S 表中删除 S2。

⑪ 请将 (S2,J6,P4,200) 插入供应情况关系。

```
INSERT INTO SPJ(SNO,JNO,PNO,QTY) /* INTO 子句中指明列名 */  
VALUES(S2,J6,P4,200); /* 插入的属性值与指明列要对应 */
```

或

```
INSERT INTO SPJ /* INTO 子句中没有指明列名 */  
VALUES(S2,P4,J6,200); /* 插入的记录在每个属性列上均有值 */  
/* 并且属性列要和表定义中的次序一致 */
```

6. 什么是基本表? 什么是视图? 两者的区别和联系是什么?

答:

基本表是本身独立存在的表,在 SQL 中一个关系就对应一个基本表。

视图是从一个或几个基本表导出的表。视图本身不独立存储在数据库中,是一个虚表。即数据库中只存放视图的定义而不存放视图对应的数据,这些数据仍存放在导出视图的基本表中。视图在概念上与基本表等同,用户可以如同基本表那样使用视图,可以在视图上再定义视图。

7. 试述视图的优点。

答:

- ① 视图能够简化用户的操作。
- ② 视图使用户能以多种角度看待同一数据。
- ③ 视图对重构数据库提供了一定程度的逻辑独立性。
- ④ 视图能够对机密数据提供安全保护。

详细解释参见《概论》第 3.7.4 小节。

8. 哪类视图是可以更新的,哪类视图是不可更新的? 各举一例说明。

答:

基本表的行列子集视图一般是可更新的。如《概论》第 3.7.1 小节中的[例 3.84]。

若视图的属性来自聚集函数、表达式,则该视图肯定是不可以更新的。如《概论》第 3.7.1 小节[例 3.89]中的 S\_G 视图。

9. 请为三建工程项目建立一个供应情况的视图,包括供应商代码(SNO)、零件代码(PNO)、供应数量(QTY)。针对该视图完成下列查询:

- ① 找出三建工程项目使用的各种零件代码及其数量。
- ② 找出供应商 S1 供应三建工程的情况。

答:

创建视图:

```
CREATE VIEW V_SPJ AS
```



```
SELECT SNO,PNO,QTY
FROM SPJ
WHERE JNO=
    (SELECT JNO
     FROM J
     WHERE JNAME='三建');
```

对该视图查询:

```
① SELECT PNO,QTY
   FROM V_SPJ;
② SELECT PNO,QTY           /* S1 供应三建工程的零件代码和对应的数量 */
   FROM V_SPJ
   WHERE SNO='S1';
```

### 3.3 补充习题

#### 1. 选择题

- (1) 关于 SQL 语言,下列说法正确的是( )。
- A. 数据控制功能不是 SQL 语言的功能之一  
B. SQL 采用的是面向记录的操作方式,以记录为单位进行操作  
C. SQL 是非过程化的语言,用户无须指定存取路径  
D. SQL 作为嵌入式语言语法与独立的语言有较大差别
- (2) 对表中数据进行删除的操作是( )。
- A. DELETE                      B. DROP                      C. ALTER                      D. UPDATE
- (3) 数据库中建立索引的目的是为了( )。
- A. 加快建表速度              B. 加快存取速度              C. 提高安全性                      D. 节省存储空间
- (4) 视图是数据库系统三级模式中的( )。
- A. 外模式                      B. 模式                      C. 内模式                      D. 模式映像
- (5) 下列说法不正确的是( )。
- A. 基本表和视图一样,都是关系  
B. 可以使用 SQL 对基本表和视图进行操作  
C. 可以从基本表或视图上定义视图  
D. 基本表和视图都存储数据

#### 2. 判断题

- (1) 视图不仅可以从单个基本表导出,还可以从多个基本表导出。 ( )
- (2) 不是所有的视图都可以进行更新,但视图都可以进行插入。 ( )

- (3) SELECT 子句中的目标列可以是表中的属性列,也可以是表达式。 ( )
- (4) 在 SQL 语句中表达某个属性 X 为空,可以使用 WHERE X=NULL。 ( )
- (5) SQL 语句中逻辑运算符 AND 和 OR 的优先级是一样的。 ( )
- (6) 使用 ANY 或 ALL 谓词时必须与比较运算符同时使用。 ( )

### 3. 填空题

- (1) SQL 语言具有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和数据控制的功能。
- (2) SQL 语句中用来消除重复的关键词是\_\_\_\_\_。
- (3) 若一个视图是从单个基本表导出的,并且只是去掉了基本表的某些行和某些列,但保留了主码,这类视图称为\_\_\_\_\_。
- (4) SQL 语言的数据定义功能包括\_\_\_\_\_、表定义、视图定义和\_\_\_\_\_等。

### 4. 问答题

- (1) 解释相关子查询和不相关子查询。
- (2) 写出 ANY 和 ALL 谓词与聚集函数或 IN 谓词可能存在的等价转换关系。

### 5. 综合题

关系 R 包含 A、B、C 三个属性,包含的数据如下:

R	A	B	C
	10	NULL	20
	20	30	NULL

写出对查询语句 SELECT \* FROM R WHERE X;当 X 为下列条件时的查询结果:

- ① A IS NULL
- ② A>8 AND B<20
- ③ A>10 OR B<20
- ④ C+10>25
- ⑤ EXISTS(SELECT B FROM R WHERE A=10)
- ⑥ C IN (SELECT B FROM R)

## 3.4 补充习题答案

### 1. 选择题

(1)	(2)	(3)	(4)	(5)
C	A	B	A	D

## 2. 判断题

(1)	(2)	(3)	(4)	(5)	(6)
√	×	√	×	×	√

## 3. 填空题

(1) 数据定义 数据查询 数据操纵

(2) DISTINCT

(3) 行列子集视图

(4) 模式定义 索引定义

## 4. 问答题

(1) 解释相关子查询和不相关子查询。

答:

在嵌套查询中,如果子查询的查询条件不依赖于父查询,称为不相关子查询;如果子查询的查询条件依赖于父查询,称为相关子查询。

(2) 写出 ANY 和 ALL 谓词与聚集函数或 IN 谓词可能存在的等价转换关系。

答:

	=	<>	<	<=	>	>=
ANY	IN	--	< MAX	<= MAX	> MIN	>= MIN
ALL	--	NOT IN	< MIN	<= MIN	> MAX	>= MAX

## 5. 综合题

① 空的结果集

② 空的结果集

③

A	B	C
20	30	NULL

④

A	B	C
10	NULL	20

⑤

<i>A</i>	<i>B</i>	<i>C</i>
10	NULL	20
20	30	NULL

解析:

SELECT \* FROM R WHERE EXISTS(SELECT B FROM R WHERE A=10)

这里的条件中(SELECT B FROM R WHERE A=10)返回一条记录,只不过这条记录只有一个属性 B,且其值是 NULL,所以 EXISTS 是返回真值。因此最后的答案是返回 R 的全部记录。

⑥ 空的结果集

解析:

SELECT \* FROM R WHERE C IN (SELECT B FROM R),

(SELECT B FROM R)执行的结果是(NULL,30),由于该集合中存在 NULL 值,无论 C 取何值,条件表达式 C IN (NULL,30) 总为 NULL 值,因此该 WHERE 条件为假,SELECT \* FROM R WHERE C IN (NULL,30) 执行结果应该是空的结果集。

## 第4章

## 数据库安全性

《概论》第4章详细介绍数据库安全性问题和实现技术。信息安全、计算机系统安全以及数据库系统安全是信息安全的重要内容。随着计算机特别是计算机网络的发展,数据的共享日益加强,数据的安全保密越来越重要。

### 4.1 基本知识点

数据库的安全性问题和计算机系统的安全性是紧密联系的,计算机系统的安全性问题可分技术安全类、管理安全类和政策法规类三大类安全性问题。我们讨论数据库的安全性,讨论数据库技术安全类问题,即从技术上如何保证数据库系统的安全性。

① 需要了解的:什么是计算机系统安全性问题,什么是数据库的安全性问题,威胁数据库安全性的因素有哪些。

② 需要牢固掌握的:TCSEC 和 CC 标准的主要内容。C2 级 DBMS、B1 级 DBMS 的主要特征。DBMS 提供的安全措施,包括用户身份鉴别、自主存取控制和强制存取控制技术、视图技术和审计技术、数据加密存储和加密传输等。

③ 需要举一反三的:使用 SQL 语言中的 GRANT 语句和 REVOKE 语句来实现自主存取控制。

④ 难点:强制存取控制机制中确定主体能否存取客体的存取规则,要理解并掌握存取规则为什么要这样规定,特别是规则(2)关于主体写客体的规则。

### 4.2 习题解答和解析

1. 什么是数据库的安全性?

答:数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

2. 举例说明对数据库安全性产生威胁的因素。

解析:

请读者列出自己生活和工作中遇到、听到的实际案例。

3. 试述信息安全标准的发展历史,试述 CC 评估保证级划分的基本内容。

答:

信息安全标准的发展历史详见《概论》书图 4.1。

简单地讲,TCSEC 是 1985 年美国国防部颁布的《DoD 可信计算机系统评估准则》。CC 通用准则 V2.1 版于 1999 年被 ISO 采用为国际标准,2001 年被我国采用为国家标准。目前 CC 已经基本取代了 TCSEC,成为评估信息产品安全性的主要标准。

CC 评估保证级划分的基本内容如下表所示。

评估保证级	定义
EAL1	功能测试
EAL2	结构测试
EAL3	系统的测试和检查
EAL4	系统的设计、测试和复查
EAL5	半形式化设计和测试
EAL6	半形式化验证的设计和测试
EAL7	形式化验证的设计和测试

详见《概论》表 4.2。

4. 试述实现数据库安全性控制的常用方法和技术。

答:

实现数据库安全性控制的常用方法和技术有以下几种。

① 用户身份鉴别:系统提供多种方式让用户标识自己的名字或身份。用户要使用数据库系统时,由系统进行核对,通过鉴定后才可以使数据库。

② 多层存取控制:系统提供用户权限定义和合法权限检查功能,用户只有获得某种权限才能访问数据库中的某些数据。

③ 视图机制:为不同的用户定义不同的视图,通过视图机制把要保密的数据对无权存取的用户隐藏起来,从而自动对数据提供一定程度的安全保护。

④ 审计:建立审计日志,把用户对数据库的所有操作自动记录下来放入审计日志中,审计员可以利用审计信息重现导致数据库现有状况的一系列事件,找出非法存取数据的人、时间和内容等。

⑤ 数据加密:对存储和传输的数据进行加密处理,从而使不知道解密算法的人无法获知

数据的内容。

5. 什么是数据库中的自主存取控制方法和强制存取控制方法?

答:

**自主存取控制方法**:定义各个用户对不同数据对象的存取权限。当用户对数据库访问时首先检查用户的存取权限。防止不合法用户对数据库的存取。

**强制存取控制方法**:每一个数据对象被(强制地)标以一定的密级,每一个用户也被(强制地)授予某一个级别的许可证。系统规定只有具有某一许可证级别的用户才能存取某一个密级的数据对象。

**自主存取控制中自主的含义是**:用户可以将自己拥有的存取权限“自主”地授予别人。即用户具有一定的“自主”权。

6. 对下列两个关系模式:

学生(学号,姓名,年龄,性别,家庭住址,班级号)

班级(班级号,班级名,班主任,班长)

使用 GRANT 完成下列授权功能:

- ① 授予用户 U1 拥有对两个表的所有权限,并可给其他用户授权;
- ② 授予用户 U2 对学生表具有查看权限,对家庭住址具有更新权限;
- ③ 将对班级表查看权限授予所有用户。
- ④ 将对学生表的查询、更新权限授予角色 R1。
- ⑤ 将角色 R1 授予用户 U1,并且 U1 可继续授权给其他角色。

答:

- ① GRANT ALL PRIVILEGES ON TABLE 学生,班级 TO U1 WITH GRANT OPTION;
- ② GRANT SELECT,UPDATE(家庭住址) ON TABLE 学生 TO U2;
- ③ GRANT SELECT ON TABLE 班级 TO PUBLIC;
- ④ GRANT SELECT,UPDATE ON TABLE 学生 TO R1;
- ⑤ GRANT R1 TO U1 WITH ADMIN OPTION;

7. 今有两个关系模式:

职工(职工号,姓名,年龄,职务,工资,部门号)

部门(部门号,名称,经理名,地址,电话号)

请用 SQL 的 GRANT 和 REVOKE 语句(加上视图机制)完成以下授权定义或存取控制功能:

- ① 用户王明对两个表有 SELECT 权限。

GRANT SELECT ON TABLE 职工,部门 TO 王明;

- ② 用户李勇对两个表有 INSERT 和 DELETE 权限。

GRANT INSERT,DELETE ON TABLE 职工,部门 TO 李勇;

- ③ \* 每个职工只对自己的记录有 SELECT 权限;

```
GRANT SELECT ON TABLE 职工 WHEN USER()=NAME TO ALL;
```

这里假定系统的 GRANT 语句支持 WHEN 子句和 USER() 的使用。用户将自己的名字作为 ID。注意,不同的系统这些扩展语句可能是不同的。读者应该了解所使用的 DBMS 产品的扩展语句。

- ④ 用户刘星对职工表有 SELECT 权限,对工资字段具有更新权限。

```
GRANT SELECT,UPDATE(工资) ON TABLE 职工 TO 刘星;
```

- ⑤ 用户张新具有修改这两个表的结构的权限。

```
GRANT ALTER TABLE ON TABLE 职工,部门 TO 张新;
```

- ⑥ 用户周平具有对两个表所有权限(读,插,改,删数据),并具有给其他用户授权的权限。

```
GRANT ALL PRIVILEGES ON TABLE 职工,部门 TO 周平 WITH GRANT OPTION;
```

- ⑦ 用户杨兰具有从每个部门职工中 SELECT 最高工资、最低工资、平均工资的权限,他不能查看每个人的工资。

首先建立一个视图。然后对这个视图定义杨兰的存取权限。

```
CREATE VIEW 部门工资 AS
SELECT 部门.名称,MAX(工资),MIN(工资),AVG(工资)
FROM 职工,部门
WHERE 职工.部门号=部门.部门号
GROUP BY 职工.部门号;
GRANT SELECT ON TABLE 部门工资 TO 杨兰;
```

8. 把习题 7 中①~⑦的每一种情况,撤销各用户所授予的权限。

答:

- ① **REVOKE SELECT ON TABLE 职工,部门 FROM 王明;**  
② **REVOKE INSERT,DELETE ON TABLE 职工,部门 FROM 李勇;**  
③ **REVOKE SELECT ON TABLE 职工 WHEN USER()=NAME FROM ALL;**

这里假定用户将自己的名字作为 ID,且系统的 REVOKE 语句支持 WHEN 子句,系统也支持 USER() 的使用。

- ④ **REVOKE SELECT,UPDATE ON TABLE 职工 FROM 刘星;**  
⑤ **REVOKE ALTER TABLE ON TABLE 职工,部门 FROM 张新;**  
⑥ **REVOKE ALL PRIVILEGES ON TABLE 职工,部门 FROM 周平;**  
⑦ **REVOKE SELECT ON TABLE 部门工资 FROM 杨兰;**

**DROP VIEW 部门工资;**

9. 理解并解释 MAC 机制中主体、客体、敏感度标记的含义。



答:

主体是系统中的活动实体,既包括 DBMS 所管理的实际用户,也包括代表用户的各进程。

客体是系统中的被动实体,是受主体操纵的,包括文件、基本表、索引、视图等。

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。敏感度标记被分成若干级别,如绝密、机密、可信、公开等。主体的敏感度标记称为许可证级别,客体的敏感度标记称为密级。

10. 举例说明强制存取控制机制是如何确定主体能否存取客体。

答:

假设要对关系变量 S 进行强制存取控制,为简化起见,假设要控制存取的数据单元是元组,则每个元组标以密级,如下表所示(4=绝密,3=机密,2=秘密)。

SNO	SNAME	STATUS	CITY	CLASS
S1	Smith	20	London	2
S2	Jones	10	Paris	3
S3	Clark	20	London	4

假设系统的存取规则是:

① 仅当主体的许可证级别大于或等于客体的密级时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才能写相应的客体。

假定用户 U1 和 U2 的许可证级别分别为 3 和 2,则根据规则用户 U1 能读元组 S1 和 S2,可修改元组 S2;用户 U2 只能读元组 S1,修改元组 S1。

11. 什么是数据库的审计功能,为什么要提供审计功能?

答:

审计功能是指 DBMS 的审计模块在用户对数据库执行操作的同时,把所有操作自动记录到系统的审计日志中。

因为任何系统的安全保护措施都不是完美无缺的,蓄意盗窃破坏数据的人总可能存在。利用数据库的审计功能,审计员可以根据审计日志中记录的信息,分析和重现导致数据库现有状况的一系列事件,找出非法存取数据的人、时间和内容等。

## 4.3 补充习题

1. 选择题

(1) 强制存取控制策略是 TCSEC/TDI 哪一级安全级别的特色( )。

A. C1

B. C2

C. B1

D. B2

(2) SQL 的 GRANT 和 REVOKE 语句可以用来实现( )。

- A. 自主存取控制                      B. 强制存取控制  
C. 数据库角色创建                    D. 数据库审计

(3) 在强制存取控制机制中,当主体的许可证级别等于客体的密级时,主体可以对客体进行如下操作( )。

- A. 读取                      B. 写入                      C. 不可操作                      D. 读取、写入

## 2. 填空题

(1) 数据库安全技术包括用户身份鉴别、\_\_\_\_、\_\_\_\_、\_\_\_\_和数据加密等。

(2) 在数据加密技术中,原始数据通过某种加密算法变换为不可直接识别的格式,称为\_\_\_\_。

(3) 数据库角色实际上是一组与数据库操作相关的各种\_\_\_\_。

(4) 在对用户授予列 INSERT 权限时,一定要包含对\_\_\_\_的 INSERT 权限,否则用户的插入会因为空值被拒绝。除了授权的列,其他列的值或者取\_\_\_\_,或者为\_\_\_\_。

## 4.4 补充习题答案

### 1. 选择题

(1)	(2)	(3)
C	A	D

### 2. 填空题

(1) 自主存取控制和强制存取控制    视图    审计

(2) 密文

(3) 权限

(4) 主码    空值    默认值

## 第5章

## 数据库完整性

《概论》第5章详细介绍数据库的完整性。数据库的完整性是指数据库中数据的正确性。由于数据库中的数据之间是相互联系的,因此数据库的完整性还包含数据的相容性。

数据库的完整性包括三个方面,完整性约束定义机制、完整性检查机制和违背完整性约束条件时应采取的预防措施。

### 5.1 基本知识点

① 需要了解的:什么是数据库的完整性约束条件,完整性约束条件的分类,数据库的完整性概念与数据库的安全性概念的区别和联系。

② 需要牢固掌握的:DBMS 完整性控制机制的三个方面,即完整性约束条件的定义、完整性约束条件的检查和违约处理;使用触发器实现数据库完整性的方法。

③ 需要举一反三的:用 SQL 语言定义关系模式的完整性约束条件,包括定义每个模式的主码;定义参照完整性;定义与应用有关的完整性。

④ 难点:RDBMS 如何实现参照完整性的策略,即当操作违反实体完整性、参照完整性和用户定义的完整性约束条件时,RDBMS 应该如何进行处理,以确保数据的正确与有效。其中比较复杂的是参照完整性的实现机制。

### 5.2 习题解答和解析

1. 什么是数据库的完整性?

答:数据库的完整性是指数据的正确性和相容性。

2. 数据库的完整性概念与数据库的安全性概念有什么区别和联系?

答:数据的完整性和安全性是两个不同的概念,但是有一定的联系。前者是为了防止数据库中存在不符合语义的数据,防止错误信息的输入和输出,即所谓垃圾进垃圾出(Garbage In Garbage Out)所造成的无效操作和错误结果;后者是保护数据库防止恶意的破坏和非法的

存取。也就是说,安全性措施的防范对象是非法用户和非法操作,完整性措施的防范对象是不合语义的数据。

3. 什么是数据库的完整性约束条件?

答:完整性约束条件是指数据库中的数据应该满足的语义约束条件。

4. DBMS 的完整性控制机制应具有哪三个方面的功能?

答:

① 定义功能,即提供定义完整性约束条件的机制。

② 检查功能,即检查用户发出的操作请求是否违背了完整性约束条件。

③ 违约处理功能:如果发现用户的操作请求使数据违背了完整性约束条件,则采取一定的动作来保证数据的完整性。

5. RDBMS 在实现参照完整性时需要考虑哪些方面?

答:

RDBMS 在实现参照完整性时需要考虑可能破坏参照完整性的各种情况,以及用户违约后的处理策略。

《概论》表 5.1 清楚地总结了可能破坏参照完整性的 4 种情况以及可以采取的不同的违约处理策略,详细讨论可以参见《概论》第 5.2.2 小节。

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 ←	修改外码值	拒绝
删除元组 →	可能破坏参照完整性	拒绝/级联删除/设置为空值
修改主码值 →	可能破坏参照完整性	拒绝/级联修改/设置为空值

6. 假设有下面两个关系模式:

职工(职工号,姓名,年龄,职务,工资,部门号),其中职工号为主码;

部门(部门号,名称,经理名,电话),其中部门号为主码;

用 SQL 语言定义这两个关系模式,要求在模式中完成以下完整性约束条件的定义:

(1) 定义每个模式的主码;(2) 定义参照完整性;(3) 定义职工年龄不得超过 60 岁。

答:

```
CREATE TABLE DEPT
```

```
(Deptno NUMBER(2) PRIMARY KEY,
```

```
Deptname VARCHAR(10),
```

```
Manager VARCHAR(10),
```

```

PhoneNumber Char(12)
);

```

```

CREATE TABLE EMP

```

```

(Empno NUMBER(4) PRIMARY KEY,
 Ename VARCHAR(10),
 Age NUMBER(2),
 Job VARCHAR(9),
 Sal NUMBER(7,2),
 Deptno NUMBER(2),
 CONSTRAINT C1 CHECK (Age <= 60),
 CONSTRAINT FK_DEPTNO FOREIGN KEY (Deptno) REFERENCES DEPT (Deptno));

```

7. 在关系系统中,当操作违反实体完整性、参照完整性和用户定义的完整性约束条件时,一般是如何分别进行处理的?

答:对于违反实体完整性和用户定义的完整性的操作,一般都采用拒绝执行的方式进行处理;而对于违反参照完整性的操作,并不都是简单地拒绝执行,有时要根据应用语义执行一些附加的操作,以保证数据库的正确性。具体的处理可以参见上面第5题或《概论》第5.2.2小节。

8. 某单位想举行一个小型的联谊会,关系 Male 记录注册的男宾信息,关系 Female 记录注册的女宾信息。建立一个断言,将来宾的人数限制在 50 人以内(提示,先创建了关系 Female 和关系 Male)。

答:

```

CREATE TABLE Male                                /* 创建关系 Male */
(
  SerialNumber SmallInt PRIMARY KEY,              /* 注册的序列号 */
  Name Char(8),
  Age SmallInt,
  Occupation Char(20)
);

CREATE TABLE Female                              /* 创建关系 Female */
(
  SerialNumber SmallInt PRIMARY KEY,              /* 注册的序列号 */,
  Name Char(8),
  Age SmallInt,
  Occupation Char(20)
);

CREATE ASSERTION Party                            /* 创建断言 PARTY */
CHECK((SELECT COUNT(*) FROM Male)+(SELECT COUNT(*) FROM Female) <= 50);

```

`<=50);`

注意,KingbaseES 目前还不支持断言。

## 5.3 补充习题

### 1. 选择题

(1) 定义关系的主码意味着主码属性( )。

- A. 必须唯一
- B. 不能为空
- C. 唯一且部分主码属性不为空
- D. 唯一且所有主码属性不为空

(2) 关于语句 `CREATE TABLE R(no int,sum int CHECK(sum > 0))` 和 `CREATE TABLE R(no int,sum int,CHECK(sum > 0))`,以下说法不正确的是( )。

- A. 两条语句都是合法的
- B. 前者定义了属性上的约束条件,后者定义了元组上的约束条件
- C. 两条语句的约束效果不一样
- D. 当 sum 属性改变时检查,上述两种 CHECK 约束都要被检查

(3) 下列说法正确的是( )。

- A. 使用 `ALTER TABLE ADD CONSTRAINT` 可以增加基于元组的约束
- B. 如果属性 A 上定义了 UNIQUE 约束,则 A 不可以为空
- C. 如果属性 A 上定义了外码约束,则 A 不可以为空
- D. 不能使用 `ALTER TABLE ADD CONSTRAINT` 增加主码约束

### 2. 填空题

(1) 在 `CREATE TABLE` 时,用户定义的完整性可以通过\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_等子句实现。

(2) 关系 R 的属性 A 参照引用关系 T 的属性 A,T 的某条元组对应的 A 属性值在 R 中出现,当要删除 T 的这条元组时,系统可以采用的策略包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

(3) 定义数据库完整性一般是由 SQL 的\_\_\_\_\_语句实现的。

### 3. 综合题

(1) 考虑下面的关系模式:

研究人员(人员编号,姓名,年龄,职称)

项目(项目编号,名称,负责人编号,类别)

参与(项目编号,人员编号,工作时间)/\*一个研究人员可以参加多个项目,一个项目有多个研究人员参加,工作时间给出某研究人员参加某项目的月数\*/

写出下面的完整性约束:

① 定义三个关系中的主码、外码、参照完整性;

- ② 每个研究人员的年龄不能超过 35 岁；
- ③ 每个研究人员的职称只能是“讲师”、“副教授”或“教授”；
- ④ 一个研究人员参加各种项目的总工作时间不能超过 12 个月；
- ⑤ 每个项目至少有 5 位研究人员；
- ⑥ 每个研究人员参加的项目数不能超过 3 个。

(2) 考虑题(1)中的关系模式,使用 ALTER TABLE ADD CONSTRAINT 声明如下完整性约束。

① 负责人编号参照研究人员的“人员编号”属性,当对“研究人员”更新时,若违反约束则拒绝操作；

- ② 同①,但当违反约束时将负责人编号置为 NULL；
- ③ 同①,但当违反约束时将项目中的相应元组删除或修改；
- ④ 工作时间在 1 到 12 之间；
- ⑤ 项目名称不能为空。

(3) 使用 CHECK 短语写出题(1)中项目关系的参照完整性约束。

(4) 考虑下面的关系模式：

Teacher(Tno, Tname, Tage, Tsex)

Department(Dno, Dname, Tno) /\* 其中 Tno 为系主任的职工号 \*/

Work(Tno, Dno, Year, Salary) /\* 某系某职工在某一年的工资 \*/

将下列要求写成触发器：

- ① 在插入新教师时,也将此教师信息插入到 Work 关系中,不确定的属性赋以 NULL 值；
- ② 在更新教师年龄时,如果新年龄比旧年龄低,则用旧年龄代替。

## 5.4 补充习题答案

### 1. 选择题

(1)	(2)	(3)
D	C	A

### 2. 填空题

- (1) NOT NULL    UNIQUE    CHECK
- (2) 拒绝执行    级联删除    设为空值
- (3) DDL

### 3. 综合题

- (1) CREATE TABLE 研究人员

```
(人员编号 int PRIMARY KEY,  
  姓名 Char(8),  
  年龄 SmallInt CHECK(年龄 <= 35),  
  职称 Char(8) CHECK(职称 in('讲师','副教授','教授'))  
);
```

CREATE TABLE 项目

```
(项目编号 int PRIMARY KEY,  
  名称 Char(20),  
  负责人编号 int,  
  类别 Char(8),  
  FOREIGN KEY(负责人编号) REFERENCES 研究人员(人员编号)  
);
```

CREATE TABLE 参与

```
(项目编号 int,  
  人员编号 int,  
  工作时间 SmallInt,  
  PRIMARY KEY(项目编号,人员编号),  
  FOREIGN KEY(项目编号) REFERENCES 项目(项目编号),  
  FOREIGN KEY(人员编号) REFERENCES 研究人员(人员编号),  
);  
  
CREATE ASSERTION 工作时间限制 /* 创建断言工作时间限制 */  
CHECK(12 >= ALL(SELECT SUM(工作时间) FROM 参与 GROUP BY 人员编号));  
  
CREATE ASSERTION 项目参加人数 /* 创建断言项目参加人数限制 */  
CHECK(5 <= ALL(SELECT COUNT(人员编号) FROM 参与 GROUP BY 项目编号));  
  
CREATE ASSERTION 研究员参加项目 /* 创建断言研究员参加项目数限制 */  
CHECK(3 >= ALL(SELECT COUNT(项目编号) FROM 参与 GROUP BY 人员编号));
```

(2)

- ① ALTER TABLE 项目 ADD CONSTRAINT C1 FOREIGN KEY(负责人编号) REFERENCES 研究人员(人员编号) ON UPDATE NO ACTION; /\* 违反约束时 KingbaseES 的默认值是 NO ACTION, 所以 NO ACTION 也可以缺省 \*/
- ② ALTER TABLE 项目 ADD CONSTRAINT C2 FOREIGN KEY(负责人编号) REFERENCES 研究人员(人员编号) ON DELETE SET NULL ON UPDATE SET NULL;
- ③ ALTER TABLE 项目 ADD CONSTRAINT c3 FOREIGN KEY(负责人编号) REFERENCES 研究人员(人员编号) ON DELETE CASCADE ON UPDATE CASCADE;
- ④ ALTER TABLE 参与 ADD CONSTRAINT c4 CHECK(工作时间 >= 1 AND 工作时间 <= 12);
- ⑤ ALTER TABLE 项目 ADD CONSTRAINT c5 CHECK(名称 IS NOT NULL);



(3) CHECK(负责人编号 IN(SELECT 人员编号 FROM 研究人员));

(4)

① CREATE TRIGGER NewTeacher

AFTER INSERT ON Teacher

FOR EACH ROW AS

BEGIN /\* 插入新教师后将此教师信息插入到 Work 中,不确定的属性赋以 NULL \*/

INSERT INTO work VALUES( new.tno, NULL, NULL, NULL);

END;

② CREATE TRIGGER UpdateAge

AFTER UPDATE of tage on Teacher

FOR EACH ROW AS

BEGIN

IF( new.tage < old.tage) /\* 更新教师年龄时,如果新年龄比旧年龄小 \*/

UPDATE Teacher SET tage=old.tage WHERE tno=new.tno; /\* 则用旧年龄代替 \*/

END;

注意:不同的 RDBMS 实现的触发器语法各不相同,互不兼容。请读者在上机实验时注意阅读所用系统的使用说明。

