第五章　数据库完整性

一、什么是数据库的完整性

- 数据库的完整性是指数据的正确性和相容性，防止不合语义的数据进入数据库。

例如:

学生的年龄必须是整数，取值范围为14--29；
学生的性别只能是男或女；
学生的学号一定是唯一的；
学生所在的系必须是学校开设的系；

二、完整性和安全性的关系

- 数据的安全性是保护数据库防止恶意破坏和非法存取。
- 数据的完整性是为了防止数据库中存在不符合语义的数据，也就是防止数据库中存在不正确的数据。

三、完整性控制机制

为维护数据库的完整性，DBMS必须能够

1.提供定义完整性约束条件的机制

2.提供完整性检查的方法

3.违约处理

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

5.1.1 定义实体完整性

- 实体完整性在CREATE TABLE中用PRIMARY KEY定义。
- 单属性构成的码的定义方法
  - 列级约束条件
  - 表级约束条件
- 多个属性构成的码只能定义为表级约束条件。

5.1.1 定义实体完整性

- [例1] 将Student表中的Sno属性定义为码。

```
CREATE TABLE Student
    (Sno CHAR(9) PRIMARY KEY,
    Sname CHAR(8) NOT NULL,
    Ssex CHAR(2),
    Sage SMALLINT,
    Sdept CHAR(20) );
```

5.1.1 定义实体完整性

- [例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));
```

5.1.2 实体完整性检查和违约处理

- 当用户对基本表插入数据或者对主码列进行更改操作时，RDBMS将自动进行检查：
  - 主码值是否唯一
  - 主码的各个属性是否为空
- 检查方法
  - RDBMS对主码自动建立一个索引。
  - 通过索引检查基本表中是否已经存在新的主码值。

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

5.2.1 定义参照完整性

- 参照完整性在CREATE TABLE中用FOREIGN KEY 短语定义哪些列为外码列，用REFERENCES短语指明这些外码参照哪些表的主码。

5.2.1 定义参照完整性

- [例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),
    FOREIGN  KEY  (Sno)  REFERENCES Student,
    FOREIGN  KEY  (Cno)  REFERENCES Course(Cno)
    );
```

5.2.2 参照完整性检查和违约处理

- 在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。
- 当发生不一致时，处理策略
  - 拒绝执行（NO ACTION）
  - 级联操作（CASCADE）
  - 设置为空值

5.2.2 参照完整性检查和违约处理

- 一般情况，当违反参照完整性时，系统选用默认策略，拒绝执行。如果要采用其他策略则必须在创建时显式地加以说明。

5.2.2 参照完整性检查和违约处理

- [例4] 显式定义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),
    FOREIGN  KEY  (Sno)  REFERENCES Student(Sno)
          ON DELETE CASCADE  ON UPDATE CASCADE,
    FOREIGN  KEY  (Cno)  REFERENCES Course(Cno)
          ON DELETE NO ACTION  ON UPDATE CASCADE
    );
```

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

5.3.1 属性上的约束条件

- 在CREATE TABLE 中定义属性上的约束条件
  - 列值非空
  - 列值唯一
  - 检查列值是否满足一个条件表达式

1. 不允许取空值

- [例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,
    …
    );
```

2.列值唯一

- [例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,
    Location CHAR(10) ,
    PRIMARY  KEY (Deptno)
    );
```

3. 用CHECK短语指定列值应该满足的条件

- [例7] Student表的Ssex只允许取“男”和“女”。

```
CREATE  TABLE  Student1
    (Sno  CHAR(9) PRIMARY  KEY,
    Sname  CHAR(8)  NOT  NULL,
    Ssex CHAR(2)  CHECK (Ssex  IN ('男','女')),
    Sage  SMALLINT,
    Sdept CHAR(20)
    );
```

3. 用CHECK短语指定列值应该满足的条件

- [例8] SC表的Grade值应该在0－100之间。

```
CREATE  TABLE  SC
        (Sno  CHAR(9)  NOT NULL,
        Cno  CHAR(4)  NOT NULL,
        Grade SMALLINT  CHECK(Grade >=0 AND
                                        Grade <= 100),
        PRIMARY  KEY (Sno,Cno),
        FOREIGN  KEY  (Sno)  REFERENCES Student(Sno),
        FOREIGN  KEY  (Cno)  REFERENCES Course(Cno)
        );
```

5.3.2 属性上的约束条件检查和违约处理

- 当往表中插入元组或修改属性的值时，RDBMS就检查属性上的约束条件是否被满足，如果不满足则操作被拒绝执行。

5.3.3 元组上的约束条件

- 与属性上约束条件的定义类似，在CREATE TABLE 语句中可以用CHECK短语定义元组上的约束条件。

5.3.3 元组上的约束条件

- [例9] 当学生的性别是男时，其名字不能以Ms.打头

```
CREATE  TABLE  Student
        (Sno  CHAR(9),
        Sname  CHAR(8)  NOT  NULL,
        Ssex CHAR(2) ,
        Sage  SMALLINT,
        Sdept CHAR(20),
        PRIMARY  KEY (Sno),
          CHECK (Ssex = '女' OR Sname NOT LIKE 'Ms.%')
        );
```

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

1. 完整性约束命名子句

- 子句格式

CONSTRAINT <完整性约束条件名>
[PRIMARY KEY 短语 |
FOREIGN KEY 短语 |
CHECK 短语]

1. 完整性约束命名子句

- [例10] 建立学生登记表Student，要求学号在90000~99999之间，姓名不能取空值，年龄小于30，性别只能是“男”或“女”。

```
CREATE  TABLE  Student
        (Sno  NUMERIC(6) CONSTRAINT  C1
                    CHECK (Sno BETWEEN 90000  AND  99999),
        Sname  CHAR(20)  CONSTRAINT  C2  NOT  NULL,
        Sage  NUMERIC(3)  CONSTRAINT  C3
                    CHECK (Sage < 30),
        Ssex CHAR(2) CONSTRAINT  C4
                    CHECK (Ssex  IN ('男', '女')),
                    CONSTRAINT StudentKey  PRIMARY  KEY (Sno)
        );
```

1. 完整性约束命名子句

- [例11] 建立教师表TEACHER，要求每个教师的应发工资不低于3000元。

```
CREATE  TABLE TEACHER
        (Eno  NUMERIC(4) PRIMARY  KEY ,
        Ename CHAR(10),
        Job  CHAR(8),
        Sal  NUMERIC (7,2),
        Deduct NUMERIC (7,2),
        Deptno NUMERIC (2)
        CONSTRAINT  EMPFKey FOREIGN KEY (Deptno)
                    REFERENCES DEPT(Deptno),
        CONSTRAINT C1 CHECK (Sal + Deduct >=3000)
        );
```

2. 修改表中的完整性约束

- [例12] 去掉[例10] Student表中对性别的约束。

```
ALTER  TABLE  Student
DROP  CONSTRAINT  C4;
```

2. 修改表中的完整性约束

- [例13] 修改表 Student 中的约束条件，要求学号改为在900000~999999之间，年龄由小于30改为小于40。

```
ALTER  TABLE  Student
    DROP  CONSTRAINT  C1,C3;
ALTER  TABLE  Student
    ADD CONSTRAINT  C1 CHECK (Sno
            BETWEEN 90000  AND  99999),
    ADD CONSTRAINT  C3 CHECK (Sage < 40);
```

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

IDENTITY列

- IDENTITY列

  当用户向表中插入新的数据行时，系统自动为该行赋值，并保证其值在表中的唯一性。每个表中只能有一个IDENTITY列，其值不能由用户更新，不允许空值。

  IDENTITY列的数据类型只能为 int，smallint，tinyint，numeric，decimal。

  例：CREATE TABLE T_name
  (sno INT IDENTITY (1,10)
  ……)

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

5.7 触发器

- 触发器（Trigger）是用户定义在关系表上的一类由事件驱动的特殊过程。一旦定义，任何用户对表的增、删、改操作均由服务器自动激活相应的触发器，在DBMS核心层进行集中的完整性控制。触发器类似于约束，但比约束更加灵活，可以实施比FOREIGN KEY 约束、CHECK约束更为复杂的检查和操作，具有更精细和更强大的数据控制能力。

第五章 数据库完整性

5.1 实体完整性

5.2 参照完整性

5.3 用户定义的完整性

5.4 完整性约束命名子句

5.5 域中的完整性约束

5.6 断言

5.7 触发器

5.8 小结

5.8 小结

- 三类完整性
  - 实体完整性
  - 参照完整性
  - 自定义完整性
- 完整性约束命名子句
- 断言
- 触发器