数据库系统概论

An Introduction to Database System

第五章　数据库完整性

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

    例如:
      - 学生的学号必须唯一
      - 性别只能是男或女
      - 成绩的取值范围为0~100
  - 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

数据库完整性（续）

- 数据的完整性和安全性是两个不同概念
  - 数据的完整性
    - 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
    - 防范对象：不合语义的、不正确的数据
  - 数据的安全性
    - 保护数据库 防止恶意的破坏和非法的存取
    - 防范对象：非法用户和非法操作

完整性是阻止合法用户通过合法操作向数据库中加入不正确的数据
安全性防范的是非法用户和非法操作 存取数据库中的正确数据

数据库完整性（续）

❖ 为维护数据库的完整性，数据库管理系统必须：

1.提供定义完整性约束条件的机制

- 完整性约束条件也称为完整性规则，是数据库中的数据必须满足的语义约束条件
- SQL标准使用了一系列概念来描述完整性，包括关系模型的实体完整性、参照完整性和用户定义完整性
- 这些完整性一般由SQL的数据定义语言语句来实现

数据库完整性（续）

2.提供完整性检查机制

- 数据库管理系统中检查数据是否满足完整性约束条件的机制称为完整性检查。
- 一般在**INSERT**、**UPDATE**、**DELETE**语句执行后开始检查，也可以在事务提交时检查

数据库完整性（续）

3.违约处理

- 数据库管理系统若发现用户的操作违背了完整性约束条件，就采取一定的动作
  - 拒绝（NO ACTION）执行该操作
  - 级连（CASCADE）执行其他操作

数据库完整性（续）

- 由DBMS进行完整性检查的好处
  - 不必由应用程序来完成，从而减轻了应用程序员的负担。
  - 能够为所有的用户和所有的应用提供一致的数据库完整性，避免出现漏洞。

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 完整性约束命名字句

*5.5 域中的完整性限制

5.6 断言

5.7 触发器

5.8 小结

5.1 实体完整性

5.1.1 实体完整性定义

5.1.2 实体完整性检查和违约处理

5.1.1 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*/
    );
```

5.1 实体完整性

5.1.1 实体完整性定义

5.1.2 实体完整性检查和违约处理

5.1.2 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。
  - 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

实体完整性检查和违约处理（续）

- 检查记录中主码值是否唯一的一种方法是进行全表扫描
  - 依次判断表中每一条记录的主码值与将插入记录上的主码值（或者修改的新主码值）是否相同

待插入记录

Keyi	F2i	F3i	F4i	F5i

基本表

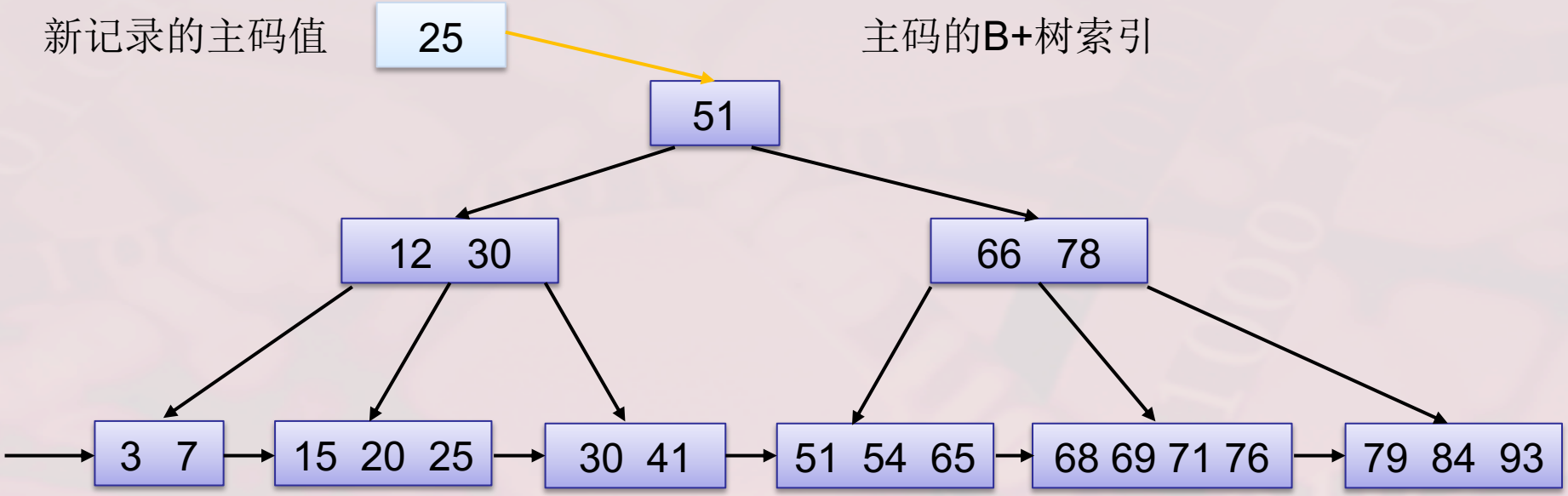
Key1	F21	F31	F41	F51
Key2	F22	F32	F42	F52
Key3	F23	F33	F43	F53
⋮				

实体完整性检查和违约处理（续）

- 表扫描缺点
  - 十分耗时
- 为避免对基本表进行全表扫描，RDBMS核心一般都在主码上自动建立一个索引

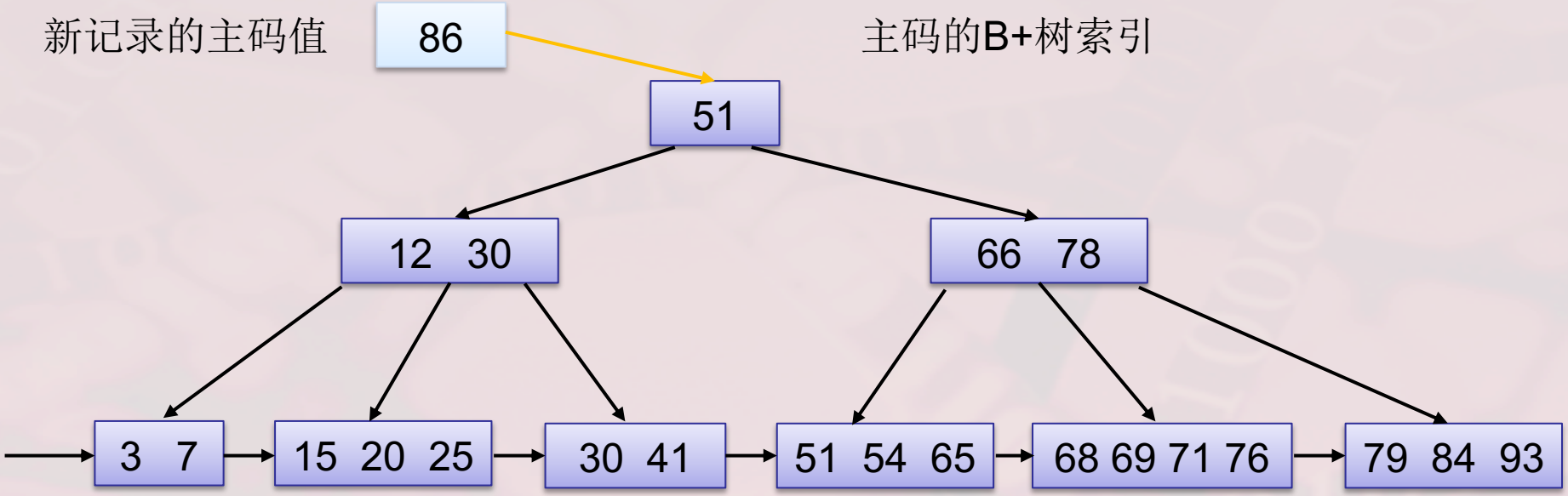
实体完整性检查和违约处理（续）

例：新插入记录的主码值是25



实体完整性检查和违约处理（续）

例：新插入记录的主码值是86



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 完整性约束命名字句

*5.5 域中的完整性限制

5.6 断言

5.7 触发器

5.8 小结

5.2 参照完整性

5.2.1 参照完整性定义

5.2.2 参照完整性检查和违约处理

参照完整性规则

若属性（或属性组）***F***是基本关系***R***的外码它与基本关系***S***的主码K_s相对应（基本关系***R***和***S***可以是相同的关系），则对于***R***中每个元组在***F***上的值必须为：

- 或者取空值（***F***的每个属性值均为空值）
- 或者等于***S***中某个元组的主码值

参照完整性规则（续）

例：学生关系的“专业号”是外码，它参照专业关系的主码“专业号”

学生关系 $\xrightarrow{\text{专业号}}$ 专业关系

学生关系中每个元组的“专业号”属性只取两类值：

（1）空值，表示该学生尚未确定专业

（2）非空值，这时该值必须是专业关系中某个元组的“专业号”值，表示该学生不可能属于一个不存在的专业

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：Student表的Sdept属性是外码，参照DEPT表的主码Deptno

```
CREATE TABLE Student
 (Sno   CHAR(9) PRIMARY KEY,      /* 列级完整性约束条件,Sno是主码*/
  Sname CHAR(20) UNIQUE,
  Ssex    CHAR(2),
  Sage   SMALLINT,
  Sdept  CHAR(20) FOREIGN KEY REFERENCES DEPT(Deptno)
                                          /*在列级定义参照完整性*/
  );
```

参照完整性定义（续）

例：Student表的Sdept属性是外码，参照DEPT表的主码Deptno

```
CREATE TABLE Student
 (Sno   CHAR(9) PRIMARY KEY,      /* 列级完整性约束条件,Sno是主码*/
  Sname CHAR(20) UNIQUE,
  Ssex    CHAR(2),
  Sage   SMALLINT,
  Sdept  CHAR(20),
  FOREIGN KEY(Sdept) REFERENCES DEPT(Deptno)
                                        /*在表级定义参照完整性*/
  );
```

参照完整性定义（续）

例：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

```
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*/
  );
```

5.2 参照完整性

5.2.1 参照完整性定义

5.2.2 参照完整性检查和违约处理

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

DBMS什么时候要进行参照完整性的检查？

例:表SC和Student有四种可能破坏参照完整性的情况(1)

❖ SC表中增加一个元组，该元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等

Student

Sno	Sname	Ssex	Sage	Sdept
121	李文	女	19	12
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
121	101	95
121	103	88
121	201	79
122	201	87
111	101	90

?

例:表SC和Student有四种可能破坏参照完整性的情况(2)

❖ 修改SC表中的一个元组，修改后该元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等

Student

Sno	Sname	Ssex	Sage	Sdept
121	李文	女	19	12
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
121	101	95
121	103	88
121	201	79
20	101	82
122	102	90
122	201	87

?

例:表SC和Student有四种可能破坏参照完整性的情况(3)

❖ 从Student表中删除一个元组，造成SC表中某些元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等。

Student

Sno	Sname	Ssex	Sage	Sdept
~~121~~	~~李文~~	~~女~~	~~19~~	~~12~~
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
121	101	95
121	103	88
121	201	79
122	101	82
122	102	90
122	201	87

例:表SC和Student有四种可能破坏参照完整性的情况(4)

❖ 修改Student表中一个元组的Sno属性，造成SC表中某些元组的Sno属性的值在表Student中找不到一个元组，其Sno属性的值与之相等 。

Student

Sno	Sname	Ssex	Sage	Sdept
211	李文	女	19	12
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
121	101	95
121	103	88
121	201	79
122	101	82
122	102	90
122	201	87

参照完整性检查和违约处理

表5.1 可能破坏参照完整性的情况及违约处理

被参照表（例如Student）	参照表（例如SC）	违约处理

参照完整性检查和违约处理（续）

❖ 参照完整性违约处理

（1） 拒绝（NO ACTION）执行

- 不允许该操作执行。该策略一般设置为默认策略

（2） 级联（CASCADE）操作

- 当删除或修改被参照表（Student）的一个元组造成了与参照表（SC）的不一致，则删除或修改参照表中的所有造成不一致的元组

参照完整性检查和违约处理（续）

Student

Sno	Sname	Ssex	Sage	Sdept
~~121~~	~~李文~~	~~女~~	~~19~~	~~12~~
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
~~121~~	~~101~~	~~95~~
~~121~~	~~103~~	~~88~~
~~121~~	~~201~~	~~79~~
122	101	82
122	102	90
122	201	87

如果某个学生退学了，该学生先前的所有选课记录也就都一并删除了。

参照完整性检查和违约处理（续）

❖ 参照完整性违约处理

（1） 拒绝（NO ACTION）执行

- 不允许该操作执行。该策略一般设置为默认策略

（2） 级联（CASCADE）操作

- 当删除或修改被参照表（Student）的一个元组造成了与参照表（SC）的不一致，则删除或修改参照表中的所有造成不一致的元组

（3）设置为空值（SET-NULL）

- 当删除或修改被参照表的一个元组时造成了不一致，则将参照表中的所有造成不一致的元组的对应属性设置为空值。

参照完整性检查和违约处理（续）

Student

外码

Sno	Sname	Ssex	Sage	Sdept
121	李文	女	19	
122	刘力	男	18	
133	王强	男	19	13

Dept

主码

Deptno	Dname
11	计算机网络
~~12~~	~~计算机软件~~
13	计算机应用

某个专业删除了，该专业的所有学生专业未定，等待重新分配专业。

参照完整性检查和违约处理（续）

用户想要删除Student表中的学号为121的学生，而这个学生在SC表中有选课记录，能否将学号“设置为空值”

Student

Sno	Sname	Ssex	Sage	Sdept
121	李文	女	19	12
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
	101	95
	103	88
	201	79
122	101	82
122	102	90
122	201	87

不存在的学生，或者不知学号的学生，选修了某些课程，其成绩记录在Grade列中

参照完整性检查和违约处理（续）

Student

Sno	Sname	Ssex	Sage	Sdept
~~121~~	~~李文~~	~~女~~	~~19~~	~~12~~
122	刘力	男	18	12
133	王强	男	19	13

SC

Sno	Cno	Grade
~~121~~	~~101~~	~~95~~
~~121~~	~~103~~	~~88~~
~~121~~	~~201~~	~~79~~
122	101	82
122	102	90
122	201	87

删除Student表中的学号为121的学生

1. **级联删除： DBMS会删除121号学生的学生记录及其选课记录**
2. **拒绝执行： DBMS拒绝执行删除语句**

参照完整性检查和违约处理（续）

Student

Sno	Sname	Ssex	Sage	Sdept
~~121~~	~~李文~~	~~女~~	~~19~~	
~~122~~	~~刘力~~	~~男~~	~~18~~	
133	王强	男	19	13

Dept

Deptno	Dname
11	计算机网络
~~12~~	~~计算机软件~~
13	计算机应用

删除DEPT表中的12号专业

1. **级联删除：** **DBMS删除DEPT表的12号专业以及学生表中所有在12号专业学习的学生。**
2. **拒绝执行：** **DBMS拒绝执行删除语句。**
3. **设置为空值：** **DBMS将学生表中12号专业的学生的专业号设置为空值，并删除DEPT表的12号专业。**

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*/
);
```

小结

- 参照完整性的定义方法
  - CREATE TABLE … FOREIGN KEY … REFERENCE
- 参照完整性的检查时机
  - 对被参照表和参照表进行增删改操作时
- 参照完整性的违约处理
  - 拒绝执行
  - 级联操作
  - 设置为空值

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 完整性约束命名字句

*5.5 域中的完整性限制

5.6 断言

5.7 触发器

5.8 小结

5.3 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

5.3 用户定义的完整性

5.3.1 属性上的约束条件

5.3.2 元组上的约束条件

1. 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),
   …
);
```

属性上约束条件的定义（续）

（2）列值唯一

[例5.6]建立专业表DEPT，要求专业名称Dname列取值唯一且不能取空值，专业编号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*/
    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)
     );
```

2. 属性上的约束条件检查和违约处理

- 属性上的约束条件检查和违约处理
  - 插入元组或修改属性的值时，关系数据库管理系统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 用户定义的完整性

5.3.1 属性上的约束条件

5.3.2 元组上的约束条件

1. 元组上约束条件的定义

- 属性上的约束条件：只涉及单个属性

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

- 元组级的限制：可以设置不同属性之间的取值的相互约束条件
- 在CREATE TABLE时可以用CHECK子句定义元组上的约束条件

元组上约束条件的定义（续）

[例5.9]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.%')
          /*定义了元组中Sname和 Ssex两个属性值之间的约束条件*/
    );
```

- ✓ 性别是女性的元组都能通过该项检查，因为Ssex=‘女’成立;
- ✓ 当性别是男性时，要通过检查则名字一定不能以Ms.打头

2. 元组上约束条件检查和违约处理

- 元组上的约束条件检查和违约处理
  - 插入元组或修改属性的值时，关系数据库管理系统检查元组上的约束条件是否被满足
  - 如果不满足则操作被拒绝执行

小结

- 属性上的用户定义完整性
  - 定义方法
  - 检查时机
  - 违约处理
- 元组上的用户定义完整性
  - 定义方法
  - 检查时机
  - 违约处理

小结

	实体完整性	参照完整性	用户定义的完整性
定义方法	CREATE TABLE	CREATE TABLE	CREATE TABLE
检查时机	执行插入 修改操作	参照表：插入/修改 被参照表：删除/修改	执行插入 修改操作
违约处理	拒绝执行	拒绝执行/级联操作/设置为空值	拒绝执行

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 域中的完整性限制

5.6 断言

5.7 触发器

5.8 小结

5.4 完整性约束命名子句

1.完整性约束命名子句

CONSTRAINT <完整性约束条件名><完整性约束条件>

- <完整性约束条件>包括NOT NULL、UNIQUE、PRIMARY KEY短语、FOREIGN KEY短语、CHECK短语等

完整性约束命名子句（续）

[例5.10]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)
  );
```

完整性约束命名子句（续）

[例5.11]建立教师表TEACHER，要求每个教师的应发工资不低于3000元。应发工资是工资列Sal与扣除项Deduct之和。

```
CREATE TABLE TEACHER
    (   Eno     NUMERIC(4)  PRIMARY KEY    /*在列级定义主码*/
        Ename   CHAR(10),
        Job     CHAR(8),
        Sal     NUMERIC(7,2),
        Deduct  NUMERIC(7,2),
        Deptno  NUMERIC(2),
        CONSTRAINT TEACHERFKey FOREIGN KEY (Deptno)
                REFERENCES DEPT(Deptno),
        CONSTRAINT C1 CHECK (Sal + Deduct >= 3000)
    );
```

完整性约束命名子句（续）

2. 修改表中的完整性限制

- 使用ALTER TABLE语句修改表中的完整性限制

[例5.12]去掉例5.10 Student表中对性别的限制。

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

. 修改表中的完整性限制（续）

[例5.13] 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

- 可以先删除原来的约束条件，再增加新的约束条件
  - ALTER TABLE Student
    DROP CONSTRAINT C1;
  - ALTER TABLE Student
    ADD CONSTRAINT C1 CHECK (Sno BETWEEN 900000
    AND 999999),
  - ALTER TABLE Student
    DROP CONSTRAINT C3;
  - ALTER TABLE Student
    ADD CONSTRAINT C3 CHECK(Sage < 40);

域约束：CREATE DOMAIN语句(含 CHECK子句)

定义：

```
CREATE DOMAIN COLOR CHAR(6) DEFAULT '???'
        CHECK(VALUE IN
                ('Red','Yell
```

格式同SELECT的WHERE

使用：

```
CREATE TABLE PART ( pno char(5),
                    pname varchar(20),
                    partcolor COLOR,
                    ……          );
```

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 完整性约束命名字句

*5.5 域中的完整性限制

5.6 断言

5.7 触发器

5.8 小结

断言

- SQL中，可以使用 CREATE ASSERTION语句，通过声明性断言来指定更具一般性的约束。
- 可以定义涉及多个表的或聚集操作的比较复杂的完整性约束。
- 断言创建以后，任何对断言中所涉及的关系的操作都会触发关系数据库管理系统对断言的检查，任何使断言不为真值的操作都会被拒绝执行

断言（续）

1. 创建断言的语句格式

CREATE ASSERTION<断言名><CHECK 子句>

- 每个断言都被赋予一个名字
- **<CHECK 子句>**中的约束条件与**WHERE**子句的条件表达式类似。

断言（续）

[例5.18] 限制数据库课程最多60名学生选修

```
CREATE ASSERTION ASSE_SC_DB_NUM
CHECK (60 >= (select count(*)
              From Course,SC
              Where SC.Cno=Course.Cno and
                    Course.Cname ='数据库')
      );
```

触发时机：向SC表中插入元组
违约反应：拒绝执行

断言（续）

[例5.19]限制每一门课程最多60名学生选修

```
CREATE ASSERTION ASSE_SC_CNUM1
    CHECK(60 >= ALL (SELECT count(*)
                     FROM  SC
                     GROUP by cno)
          );
```

断言（续）

[例5.20]限制每个学期每一门课程最多60名学生选修

首先需要修改SC表的模式，增加一个“学期（TERM）”属性

如果断言很复杂，则系统检测和维护断言的开销较高，这是在使用断言时应该注意的

```
CREATE ASSERTION ASSE_SC_CNUM2
CHECK(60 >= ALL (SELECT count(*)
                 FROM SC
                 GROUP by cno,TERM)
                 );
```

断言（续）

2. 删除断言的语句格式为

- DROP ASSERTION <断言名>;

5.8 小结

- 数据库的完整性是为了保证数据库中存储的数据是正确的

- 关系数据库管理系统完整性实现的机制
  - 完整性约束定义机制
  - 完整性检查机制
  - 违背完整性约束条件时关系数据库管理系统应采取的动作