1. 概述

关系模型的完整性规则是对关系的某种约束条件，也就是关系的值需要满足的约束条件

实体完整性和参照完整性：关系模型必须满足的完整性约束条件，称为**关系的两个不变性**，应该由关系系统自动支持

用户定义的完整性：应用领域需要遵循的约束条件，体现了具体领域中的语义约

数据库的**完整性**：数据的正确性和相容性

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

为维护数据库的完整性，DBMS必须提供定义完整性约束条件的机制、提供完整性检查的方法、违约处理

1. 实体完整性
2. 定义

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

Primary key在一个CREATE TABLE只能写一次

1. 实体完整性检查和违约处理

增加、修改时

·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**（主属性）是否为空，只要有一个为空就拒绝插入或修改

检查记录中主码值是否唯一的一种方法是进行**全表扫描**，为了避免全表扫描，一般会在主码上建立索引

1. 参照完整性
2. 定义

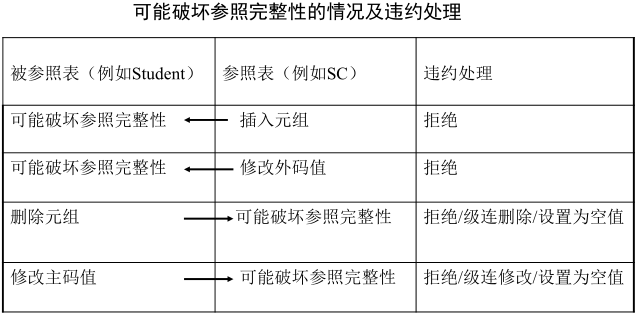
在CREATE TABLE中用FOREIGN KEY短语定义哪些列为**外码**，用REFERENCES短语指明这些外码参照哪些表的**主码**

FOREIGN KEY (Sno) REFERENCES Student(Sno)

一个参照完整性将两个表中的相应元组联系起来

1. 参照完整性检查和违约处理

增加、删除、修改时



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

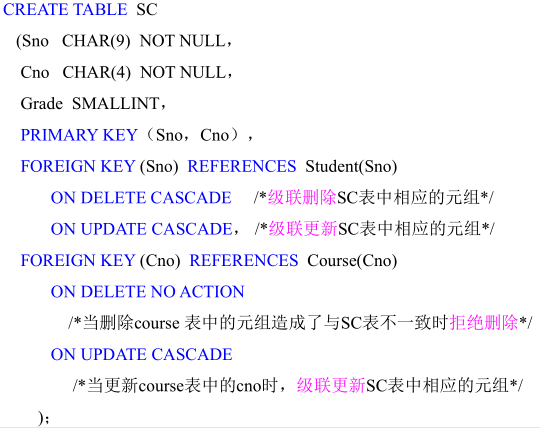
1. 级联（CASCADE）操作

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

1. 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

（此表不能设置NOT NULL，因为sno与cno是外码的同时也是主码，主属性不能为空）



1. 用户定义完整性
2. 定义

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

用户定义完整性包括：属性上的约束条件的定义、属性上的约束条件检查和违约处理、元组上的约束条件的定义、元组上的约束条件检查和违约处理

CREATE TABLE时定义：列值非空（NOT NULL）、列值唯一（UNIQUE）、检查列值是否满足一个布尔表达式（CHECK）

Sno CHAR(9) NOT NULL 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就不必写了

Dname CHAR(9) UNIQUE

Ssex CHAR(2) CHECK (Ssex IN ('男', '女') )

Grade SMALLINT CHECK(Grade >=0 AND Grade <=100)

1. 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，RDBMS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

1. 元组上的约束条件的定义

在CREATE TABLE时可以用**CHECK**短语定义元组上的约束条件，即**元组级的限制**

同属性值限制相比，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

示例：当学生的性别是男时，其名字不能以Ms.打头

CHECK (Ssex=' 女' OR Sname NOT LIKE 'Ms.%')

1. 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，RDBMS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

1. 完整性约束命名字句
2. 完整性约束命名子句

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

Sno NUMERIC(6)

CONSTRAINT C1 CHECK (Sno BETWEEN 90000 AND 99999)

1. 修改表中的完整性限制

可以先删除原来的约束条件，再增加新的约束条件

