

第五章 数据库完整性

- 完整性的概念
 - 指数据的正确性与相容性
- 与安全性的比较
 - 安全性
 - 防止认为恶意破坏数据库和非法存取
 - 完整性
 - 防止不合语义的、不正确的数据进入数据库
- DBMS维护完整性的机制
 - 定义机制
 - 允许用户定义**实体完整性**、**参照完整性**和**用户定义的完整性**
 - 检查机制
 - 在用户的更新语句（事务）开始执行**后**检查这些操作执行后**是否会违背完整性约束条件**
 - 违约处理机制
 - 如果DBMS检查到用户操作违背了完整性约束条件，就采取一定动作以保证完整性
 - 拒绝 (No Action)
 - 级联 (Cascade)
- 实体完整性
 - 定义实体完整性
 - 实体完整性检查和违约处理
 - 检查主码值（和Unique码值）是否唯一
 - 不唯一，拒绝操作
 - 检查主码的各个属性值是否为空
 - 为空，拒绝操作
 - 主码（和Unique码）上自动建立B+树索引
 - 快速检查主码值是否唯一
- 参照完整性
 - 定义参照完整性(定义外码)
 - 参照完整性检查和违约处理
 - 违约处理策略
 - 事务与约束
 - 例子：两个相互参照问题的解决

示例：两个关系相互参照问题的解决。

- 员工(编号, 姓名, 工资, 所在部门)
- 部门(编号, 部门名, 部门领导)
- 员工关系的“所在部门”是参照部门关系的一个外码；部门关系的“部门领导”是参照员工关系的一个外码。
- 如何解决？

- 解决

```
create table employee
(
    ssn char(5),
    name varchar(20),
    salary numeric(8,2),
    dno char(3),

    primary key (ssn)
);
```

```
create table department
(
    dno char(3),
    dept_name varchar(20),
    mgr_ssn char(5),

    primary key(dno)
);
```

```

alter table employee
add constraint employee_fk_dno foreign
key (dno) references department initially
deferred;

alter table department
add constraint department_fk_mgr_ssn
foreign key (mgr_ssn) references
employee(ssn) initially deferred;

begin;
insert into department (dno, dept_name, mgr_ssn)
values('101', 'HR', '10001');
insert into department (dno, dept_name, mgr_ssn)
values('102', 'FI', '20001');

insert into employee(ssn, name, salary, dno)
values('10001','Zhao',20000, '101');
insert into employee(ssn, name, salary, dno)
values('20001','Singh',19000, '102');

commit;

```

- 用户定义的完整性

- **列级完整性**：涉及**单个属性**的约束

- **列值非空 NOT NULL**
 - **列值唯一 UNIQUE**
 - **列值满足表达式 CHECK**
 - 说明Student表的Sgender只能取“男”、“女”
 - 说明SC表的Grade在0到100分之间
 - 涉及多个列值之间的比较

```

CREATE TABLE T1
(ID char(4),
oldvalue int,
newvalue int,
CONSTRAINT T1_PKEY PRIMARY KEY(ID),
CONSTRAINT T1_CHECK CHECK(newvalue>oldvalue));

```

- **表级完整性**：涉及**单个或多个属性**的约束

- **列值唯一 UNIQUE**
 - **列值满足表达式 CHECK**

- 完整性约束命名子句

- **列级完整性约束**一般**不能**显示命名

- 表级完整性约束可显示命名
 - **CONSTRAINT**子句
 - 方便用**ALTER TABLE**语句修改约束
 - ALTER COLUMN *** **DROP/SET NOT NULL**
- 域中的完整性限制
 - create domain
 - 允许用户创建带有完整性约束的自定义域
 - create domain *person_name* char(20) not null;
- 触发器
 - 用户定义在关系表上的一类由事件驱动的特殊过程
 - 例子

```

■ 例18 为教师表Teacher定义一个完整性规则触发器，
规定“教授的工资不得低于4000元，如果低于
4000元，自动改为4000元”。
■ create trigger I U Sal after insert or update on
teacher
referencing new row as nrow //nrow引用新值
for each row //行级
when (nrow.Job='教授' and nrow.Sal < 4000 //条
件
begin
  update Teacher set sal = 4000 where
  Tno=nrow.Tno;
end

```

以上内容整理于 [幕布文档](#)