

## 实验结果：

注：在 **Linux DB2** 内 **reference** 命令应该为 **references**

另外，由于故意使其出现一场现象，所以定义的删除修改限制较多

利用 **DB2**，创建一个新的数据库，在该数据库下创建如下表格：

(一) 职工（姓名，工号，出生日期，家庭地址，年薪，管理员工号，所在部门编号）  
我给出的完整性约束如下：

1. 工号为主键，使用 int 型表示，并且范围在 11111-99999 之间
2. 管理员工号为外键，引用表职工 中的工号字段并且更新受限
3. 所在部门编号为外键，引用部门中的部门编号字段，并且更新受限
4. 出生日期必须满足在 1950-1-1 和 1990-12-31 日之间
5. 工人薪水必须满足在 3000 到 30000 之间

建立表如下：

```
create table WORKER (  
    wname varchar(20) not null,  
    wid int not null,  
    wbirthday date not null,  
    waddr varchar(30) not null,  
    wsalary float not null,  
    wadmin int,  
    wdept int not null,  
    constraint W1 primary key (wid),  
    constraint W2 check (wid between 11111 and 99999),  
    constraint W3 check (wbirthday between '1950-1-1' and '1990-12-31'),  
    constraint W4 check (wsalary between 3000 and 30000),  
    constraint W5 foreign key(wadmin) references WORKER(wid) on update restrict,  
    constraint W6 foreign key(wdept) references DEPT(did) on update restrict  
);
```

(二) 项目（项目名称，项目编号，所在城市，主管部门编号）；  
完整性约束如下：

1. 项目编号为主键，并且项目编号必须在 111-999 之间
2. 管理部门编号为外键，应用部门表中的部门编号
3. 一旦删除一个部门，其管理的相应项目也将删除
4. 新建项目还未实施的所在城市和主管部门可以为空

建立表如下：

```
create table Project (  
    pname varchar(20) not null,  
    pid int not null,  
    pcity varchar(20),  
    padmindept int,  
    constraint P1 primary key (pid),  
    constraint P2 foreign key (padmindept) references DEPT(did) on delete cascade,  
    constraint P3 check (pid between 111 and 999)  
);
```

(三) 部门（部门名称，部门编号，部门负责人工号）；  
完整性约束如下：

1. 新部门的部门负责人可以为空
2. 部门编号为主键，并且编号在 1111-9999 之间
3. 部门负责人的工号为外键，引用职工表中的工号

建表如下：

```
create table DEPT (
    dname varchar(20) not null,
    did int not null,
    dadmin int,
    constraint D1 primary key(did) ,
    constraint D2 foreign key(dadmin) references WORKER(wid) on update restrict,
    constraint D3 check (did between 1111 and 9999)
);
```

(四) 工作（职工工号，项目编号，工作时间）；

完整性约束如下：

1. 职工工号为外键，并且引用职工表中的工号
2. 项目编号为外键，并且引用项目表中的项目编号,并且更新受限
3. （职工工号，项目编号）为主键
4. 工作时间为天数，限定为 30 天至 10 年

建表如下：

```
create table Job (
    workerid int not null,
    pid int not null,
    worktime int not null,
    constraint J1 primary key (workerid,pid),
    constraint J2 foreign key (workerid) references WORKER(wid),
    constraint J3 foreign key(pid) references PROJECT(pid) on delete cascade on update restrict,
    constraint J4 check (worktime between 30 and 3650)
);
```

(五) 家属（职工工号，家属的姓名，家属的性别）

完整性约束如下：

1. 职工工号为外键，引用职工中的工号，并且删除职工的同时将删除家属，员工工号更新将会受限
2. 性别只能用 1 表示男性，2 表示女性
3. （职工工号，家属姓名）为主键

建表如下：

```
create table Dependents (
    workerid int not null,
    dname char(20) not null,
    dgender int not null,
    constraint D1 primary key (workerid,dname),
    constraint D2 check (dgender IN (1,2)),
    constraint D3 foreign key(workerid) references WORKER(wid) on delete cascade on update restrict
);
```

(六) 自行插入数据

1. 在部门中插入以下几个部门

- ( '人事部', 1111, null ) ( 这里没有违反任何原则 )
- ( '资源部', 11119, null ) ( 这里编号违反了约束原则 )
- ( '管理部', null, null ) ( 这里违反了主键不能为空的原则 )

插入代码以及执行结果如下：

`insert into DEPT values('人事部',1111,null);`

执行结果如下：

```
----- 输入的命令 -----
insert into DEPT values('人事部',1111,null);
-----
insert into DEPT values('人事部',1111,null)
DB20000I  SQL 命令成功完成。
```

`insert into DEPT values('资源部',11119,null);`

执行结果如下：

```
----- 输入的命令 -----
insert into DEPT values('资源部',11119,null);
-----
insert into DEPT values('资源部',11119,null)
DB21034E  该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0545N  因为行不满足检查约束
"DB2INST2.DEPT.D3"，所以不允许所请求的操作。  SQLSTATE=23513

SQL0545N  因为行不满足检查约束 "DB2INST2.DEPT.D3

说明：

检查约束违反可能在 INSERT 或 UPDATE 操作中发生。结果行违反了该表上的检
查约束定义。

无法处理该语句。
```

可以看到不满足约束条件 D3，所以无法执行

`insert into DEPT values('管理部', null, null)`

执行结果如下：

```

----- 输入的命令 -----
insert into DEPT values('管理部',null,null);
-----

insert into DEPT values('管理部',null,null)
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0407N 不允许对 NOT NULL 列 "TBSPACEID=2, TABLEID=6, COLNO=1"
赋予空值。 SQLSTATE=23502

SQL0407N 不允许对 NOT NULL 列 "TBSPACEID=2, TABLEID=6, COLNO=1"
说明：

发生了下列情况之一：
* 更新或插入值为 NULL，但是对象列在表定义中已声明为 NOT NULL。因此：
  * 空值不可插入到该列中。
  * 更新不能将该列中的值设置为 NULL。
  * 触发器中的 SET 转换变量语句不能将该列中的值设置为 NULL。

* 更新或者插入值是 DEFAULT，但是对象列被声明为 NOT NULL，而表定义中又
  没有 WITH DEFAULT。因此：
  * 缺省值 NULL 不能插入该列。
  * 更新不能在该列设置缺省值 NULL。

```

可以看到特定插入位置上不能为空，故不能正确执行

修正上述编号分别变为 1111, 1112, 1113，然后进行正确插入。正确插入的数据为：

`insert into DEPT values('人事部',1111,null),('资源部',1112,null),('管理部', 1113, null);`

2. 在工人表中插入三个源组分别如下：

(wky, 11111, 1979-6-15, Nanjing, 5000, null, 1111)

(正确的插入)

`insert into WORKER`

`values('wky',11111,'1979-6-15','Nanjing',5000,null,1111);`

执行结果：

```

----- 输入的命令 -----
insert into WORKER values('wky',11111,'1979-6-15','Nanjing',5000,null,1111);
-----

insert into WORKER values('wky',11111,'1979-6-15','Nanjing',5000,null,1111)
DB20000I SQL 命令成功完成。

```

(wzr, 11113, 1992-5-20, Nanjing, 4500, null, 1111)

(违反了生日定义，这里的日期超出了定义)

`insert into WORKER`

`values('wzr',11113,'1992-5-20','Nanjing',4500,null,1111);`

执行结果：

```

----- 输入的命令 -----
insert into WORKER values('wzr',11112,'1992-5-20','Nanjing',4500,null,1111)
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0545N 因为行不满足检查约束
"DB2INST2.WORKER.W3"，所以不允许所请求的操作。 SQLSTATE=23513

SQL0545N 因为行不满足检查约束 "DB2INST2.WORKER.W3

```

可以看到不满组行为检查 W3，故插入不可正常执行。

( 'wz', 11112, 1990-6-30, Nanjing, 6000, 11111, 1114 )  
 ( 这里违反了外键的定义，编号为 1114 的部门不存在 )

insert into WORKER  
 values('wz',11112,'1990-6-30','Nanjing',6000,11111,1114);  
 执行结果：

```

----- 输入的命令 -----
insert into WORKER values('wz',11112,'1990-6-30','Nanjing',6000,11111,1114);
-----
insert into WORKER values('wz',11112,'1990-6-30','Nanjing',6000,11111,1114)
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0530N FOREIGN KEY "DB2INST2.WORKER.W6"
的插入或更新值不等于父表的任何父键值。 SQLSTATE=23503

SQL0530N FOREIGN KEY "DB2INST2.WORKER.W6

```

可以看到显示结果为不满组 W6 的外键定义，即 1114 号部门根本未定义，所以插入不成功。

同样修改使得上述三个元组插入成功,正确的插入数据为：

insert into WORKER  
 values('wky',11111,'1979-6-15','Nanjing',5000,null,1111),  
 ('wzr',11113,'1990-5-20','Nanjing',4500,null,1113),  
 ('wz',11112,'1990-6-30','Nanjing',6000,11111,1112);

插入中可能出现的错误上面均出现了，现在将采取正确的插入方式进行插入操作：

3.在项目表中插入以下三个项目

( '人事调度', 111, 'Nanjing', 1111 )  
 ( '资源调度', 112, 'Shanghai', 1112 )  
 ( '项目管理', 113, 'Beijing', 1113 )

代码如下：

insert into project values('人事调度',111,'Nanjing',1111);  
 insert into project values('资源调度',112,'Shanghai',1112);  
 insert into project values('项目管理',113,'Beijing',1113);  
 执行正确。

4.在工作表中插入两个元组数据

( 11111, 111, 40 )

( 11112, 112, 50 )

( 11113, 113, 60 )

insert into JOB values ( 11111,111,40),( 11112,112,50),( 11113,113,60);

执行正确

5.在家属表里面插入三个元组

( 11111, 'wjz', 1 )

( 11112, 'jdd', 2 )

( 11113, 'wyb', 1 )

insert into Dependents values(11111,'wjz',1),(11112,'jdd',2),(11113,'wyb',1)

执行正确

上述插入数据完成之后每个表如下：

WORKER

WNAME	WID	WBIRTHDAY	WADDR	WSALARY	WADMIN	WDEPT
wky	11111	1979-6-15	Nanjing	5000		1111
wz	11112	1990-6-30	Nanjing	6000	11111	1113
wzr	11113	1990-5-20	Nanjing	4500		1112

DEPT

DNAME	DID	DADMIN
人事部	1111	
资源部	1112	
管理部	1113	

JOB

WORKERID	PID	WORKTIME
11111	111	40
11112	112	50
11113	113	60



## PROJECT

PNAME	PID	PCITY	PADMINDEPT
人事调度	111	Nanjing	1111
资源调度	112	Shanghai	1112
项目管理	113	Beijing	1113

## DEPENDENTS

WORKERID	DNAME	DGENDER
11111	wjz	1
11112	jdd	2
11113	wyb	1

(七) 下面观察更新删除修改现象

- 将 wky 的个人编号改为 11114 (这是不符合更新约束的更改)

`update worker set wid=11114 where wname='wky';`

执行结果如下:

```
----- 输入的命令 -----
update worker set wid=11111 where wname='wky';
-----
update worker set wid=11111 where wname='wky'
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0531N 不能更新关系 "DB2INST2.DEPENDENTS.D3" 的父行中的父键。
SQLSTATE=23001

SQL0531N 不能更新关系 "DB2INST2.DEPENDENTS.D3"
```

从上图中可以看到不满足 dependents 中的外键更新约束

- 将 wky 的出生日期改为 1992-6-15 (这是不满足出生日期范围内的更改)

`update worker set wbirthday='1992-6-15' where wname='wky';`

执行结果如下:

```
----- 输入的命令 -----
update worker set wbirthday='1992-6-15' where wname='wky';
-----
update worker set wbirthday='1992-6-15' where wname='wky'
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0545N 因为行不满足检查约束
"DB2INST2.WORKER.W3"，所以不允许所请求的操作。 SQLSTATE=23513

SQL0545N 因为行不满足检查约束 "DB2INST2.WORKER.W3"
```

由上述结果克制更新时同样必须满足再创建表时定义的约束

- 删除 wky 这个元组

`delete from worker where wname='wky';`

执行效果如下：

```
----- 输入的命令 -----
delete from worker where wname='wky';
-----
delete from worker where wname='wky'
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0532N 因为关系 "DB2INST2.JOB.J2"
限制删除，所以不能删除父行。 SQLSTATE=23504

SQL0532N 因为关系 "DB2INST2.JOB.J2
```

上图显示由于 JOB 中定义的外键限制，导致不可删除。

#### (八) 自定义触发器

触发器代码如下：

`create trigger t after insert on JOB`

`referencing new as Np`

`for each row`

`when(Np.pid <> any(select pid from job where workerid=Np.workerid))`

`update worker set wsalary=1.02*wsalary`

`where wid=Np.workerid;`

执行结果如下：

```
----- 输入的命令 -----
create trigger t after insert on JOB
referencing new as Np
for each row
when(Np.pid <> any(select pid from job where workerid=Np.workerid))
update worker set wsalary=1.02*wsalary
where wid=Np.workerid;
-----
create trigger t after insert on JOB
referencing new as Np
for each row
when(Np.pid <> any(select pid from job where workerid=Np.workerid))
update worker set wsalary=1.02*wsalary
where wid=Np.workerid
DB20000I SQL 命令成功完成。
```

插入触发器之前，观察薪水如下：

WNAME	WID	WBIRTHDAY	WADDR	WSALARY	WAD
wky	11111	1979-6-15	Nanjing	5,000	
wzr	11113	1990-5-20	Nanjing	4,500	
wz	11112	1990-6-30	Nanjing	6,000	



接下来执行插入命令，即让 wky 参加一个新的项目 113  
insert into job values(11111,113,80);  
执行后效果如下：

WNAME	WID	WBIRTHDAY	WADDR	WSALARY	WAI
wky	11111	1979-6-15	Nanjing	5,100	
wzr	11113	1990-5-20	Nanjing	4,500	
wz	11112	1990-6-30	Nanjing	6,000	

可以看到薪水被正确修改了。

#### (九) 自定义用户

由于 DB2 本身并不具备用户定义功能，而是直接使用 linux 系统的用户，所以在 linux 新建一个用户，我将其目录设定为我已经使用过的实例目录 /home/db2inst2 下。然后使用 db2inst2 用户赋予 test 权限。

命令如下：

useradd -d /home/db2inst2 test;

sudo passwd test;

键入密码之后创建完成。

然后使用 db2inst2 用户永兴 db2 执行下述命令：

grant connect on database to user test;

grant select on job to user test;

先要将连接权限给这个用户，否则的话此用户将不能与数据库进行连接  
之后查看 test 权限：如下图：



表	SELECT	INSERT	UPDATE	DELETE	CONTROL	ALTER
DB2INST2.JOB	✓	✗	✗	✗	✗	✗

之后我是用 test 用户登录进行验证：

试图进行更改操作：

update db2inst2.job set pid=112 where workerid=11113;

执行效果如下：

```
update db2inst2.job set pid=112 where workerid=11113
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0551N "TEST" 不具有对对象 "DB2INST2.JOB" 执行操作 "UPDATE"
的必需权限或特权。 SQLSTATE=42501
```

SQL0551N "TEST" 不具有对对象 "DB2INST2.JOB"

" 执行操作 "UPDATE" 的必需权限或特权。

试图进行删除操作：

`delete from db2inst2.job where workerid=11111;`

执行效果如下：

```
-----
delete from db2inst2.job where workerid=11111
DB21034E 该命令被当作 SQL
语句来处理，因为它是无效的“命令行处理器”命令。在 SQL
处理期间，它返回：
SQL0551N  "TEST" 不具有对对象 "DB2INST2.JOB" 执行操作 "DELETE"
的必需权限或特权。  SQLSTATE=42501
```

SQL0551N "TEST" 不具有对对象 "DB2INST2.JOB"

" 执行操作 "DELETE" 的必需权限或特权。

试图进行 select 操作：

`select * from db2inst2.job;`

执行效果如下：

WORKERID	PID	WORKTIME
11111	111	40
11112	112	50
11113	113	60
11111	113	80

可以看到查询操作是可以成功的。