数据库实验二

吴文涛

091220120

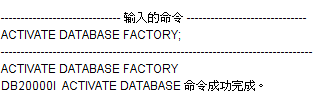
wuwentao137@gmail.com

# 建立数据库

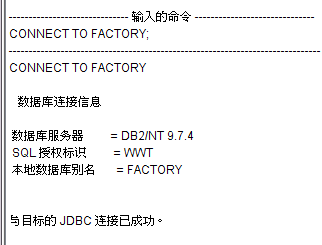
1.创建数据库，命令为：CREATE DATABASE Factory



2.激活数据库，命令为：ACTIVATE DATABASE FACTORY



3.连接数据库，命令为：CONNECT TO FACTORY



# 创建表

分析各个表之间的约束条件如下：

1. Employee中的员工的deptno必须在department中存在。
2. Employee中的管理员工号ADMINEMPTNO在employee中删除后应值为null。
3. 当删除department中的某个部门后，project的中属于该部门的项目也要被删除。
4. 当删除employee中的某个员工后，job和family中相对应的该员工的信息也要被删除。
5. 当project的某个项目被删除后，job中相对应的也要被删除。

根据约束条件创建表，命令如下：

CREATE TABLE EMPLOYEE

(EMPNAME CHAR(10) NOT NULL,

EMPNO INT NOT NULL PRIMARY KEY,

BIRTHDATE CHAR(20) NOT NULL,

ADDRESS CHAR(20),

SALARY INT,

ADMINEMPNO INT,

DEPTNO INT,

FOREIGN KEY(ADMINEMPNO) REFERENCES EMPLOYEE(EMPNO) ON DELETE SET NULL);

CREATE TABLE DEPARTMENT

(DEPNAME CHAR(10) NOT NULL,

DEPNO INT NOT NULL PRIMARY KEY,

ADMINEMPNO INT,

FOREIGN KEY(ADMINEMPNO) REFERENCES EMPLOYEE(EMPNO) ON DELETE RESTRICT);

CREATE TABLE PROJECT

(PRONAME CHAR(10) NOT NULL,

PRONO INT NOT NULL PRIMARY KEY,

CITY CHAR(20) NOT NULL,

ADMINDEPTNO INT NOT NULL,

FOREIGN KEY(ADMINDEPTNO) REFERENCES DEPARTMENT(DEPNO) ON DELETE CASCADE);

CREATE TABLE JOB

(EMPNO INT NOT NULL,

PRONO INT NOT NULL,

WORKTIME CHAR(20) NOT NULL,

PRIMARY KEY(EMPNO,PRONO),

FOREIGN KEY(PRONO) REFERENCES PROJECT(PRONO) ON DELETE CASCADE,

FOREIGN KEY(EMPNO) REFERENCES EMPLOYEE(EMPNO) ON DELETE CASCADE);

CREATE TABLE FAMILY

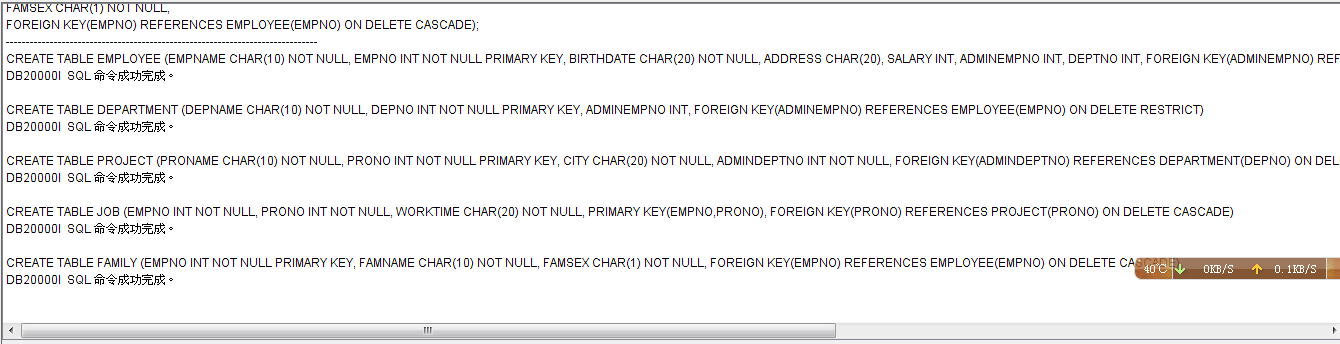
(EMPNO INT NOT NULL PRIMARY KEY,

FAMNAME CHAR(10) NOT NULL,

FAMSEX CHAR(1) NOT NULL,

FOREIGN KEY(EMPNO) REFERENCES EMPLOYEE(EMPNO) ON DELETE CASCADE);

结果如下：



在创建表的时候只能完成后四个约束条件，因为创建employee的时候department并未创建，因此在所有表都创建完时添加employee与department的约束1：

ALTER TABLE EMPLOYEE

ADD CONSTRAINT EMP\_DEPT

FOREIGN KEY (DEPTNO)

REFERENCES DEPARTMENT(DEPNO)

ON DELETE NO ACTION;

# 输入数据：

各个表的数据如下：

Employee：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 张三 | 001 | 1989.3 | Nanjing | 10000 | 002 | 101 |
| 李四 | 002 | 1988.4 | Shanghai | 15000 | Null | 101 |
| 王五 | 003 | 1990.5 | Beijing | 15000 | null | 102 |

Department：

|  |  |  |
| --- | --- | --- |
| RED | 101 | 002 |
| GREEN | 102 | 003 |

Project：

|  |  |  |  |
| --- | --- | --- | --- |
| SKY | 201 | Nanjing | 101 |
| SEA | 202 | Shanghai | 101 |
| GROUND | 203 | Beijing | 102 |

Job：

|  |  |  |
| --- | --- | --- |
| 001 | 201 | 2006.6 |
| 001 | 202 | 2007.6 |
| 002 | 201 | 2008.8 |
| 003 | 203 | 2009.1 |

Family：

|  |  |  |
| --- | --- | --- |
| 001 | 赵八 | F |
| 002 | 胡十 | F |

插入数据的命令如下：

INSERT INTO EMPLOYEE VALUES

('张三', 001, '1989.3', 'Nanjing', 10000, 002, 101),

('李四',002,'1988.4','Shanghai',15000,NULL,101),

('王五',003,'1990.5','Beijing',15000,NULL,102);

INSERT INTO DEPARTMENT VALUES

('RED',101,002),

('GREEN',102,003);

INSERT INTO PROJECT VALUES

('SKY',201,'Nanjing',101),

('SEA',202,'Shanghai',101),

('GROUND',203,'Beijing',102);

INSERT INTO JOB VALUES

(001,201,'2006.6'),

(001,202,'2007.6'),

(002,201,'2008.8'),

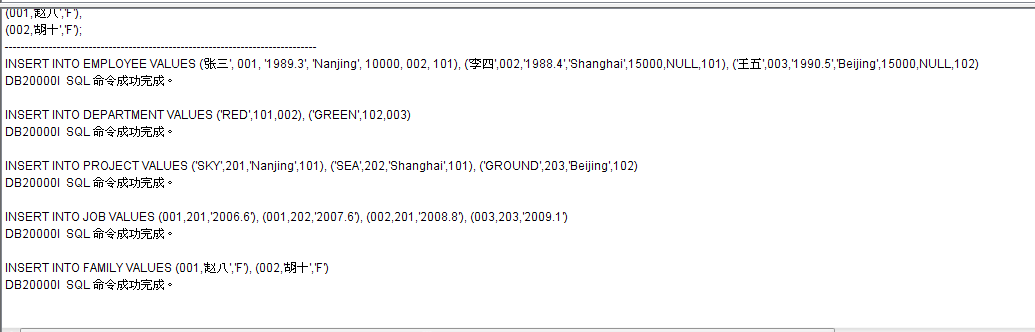
(003,203,'2009.1');

INSERT INTO FAMILY VALUES

(001,'赵八','F'),

(002,'胡十','F');

执行结果如下：



# 各个约束条件的检测

1.职员表与部门之间的约束检测

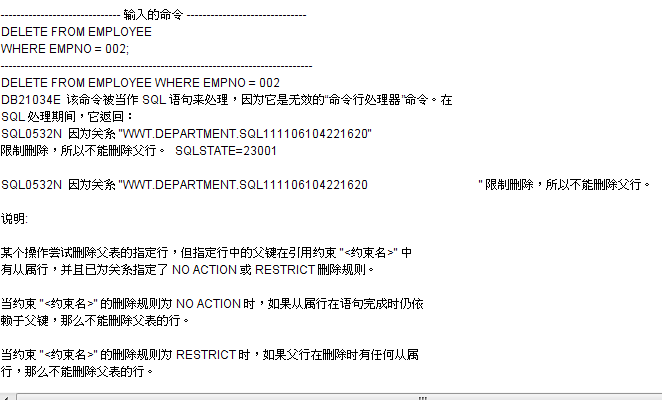
这两个表之间的约束条件为Delete restrict

若删除职员表中的李四

DELETE FROM EMPLOYEE

WHERE EMPNO = 002;

执行结果

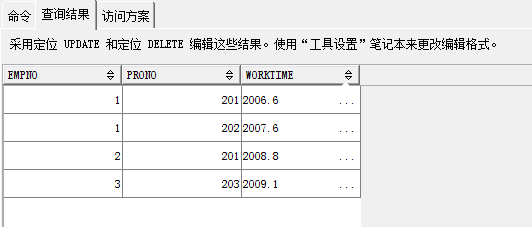


2.职员表与工作表和家属表的约束

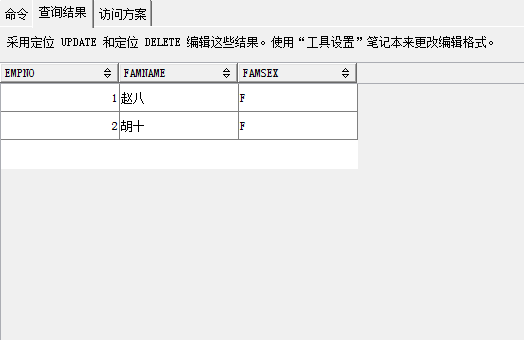
这两个表的约束条件为：Delete cascade

在操作之前

查看工作表的内容如下：



家属表的内容如下：



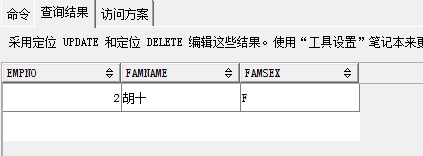
执行下面操作：

DELETE FROM EMPLOYEE

WHERE EMPNO = 001;

再次查看job和family表的内容分别如下：





3.项目与部门之间的约束

这两个表的约束条件为：Delete cascade

在操作之前project的内容如下：



执行以下操作

DELETE FROM DEPARTMENT

WHERE DEPNO = 101

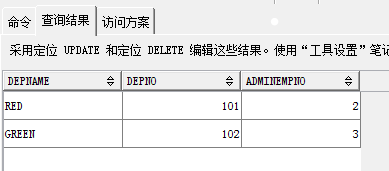
操作之后的project的内容如下：



4.职员表与部门表之间的约束

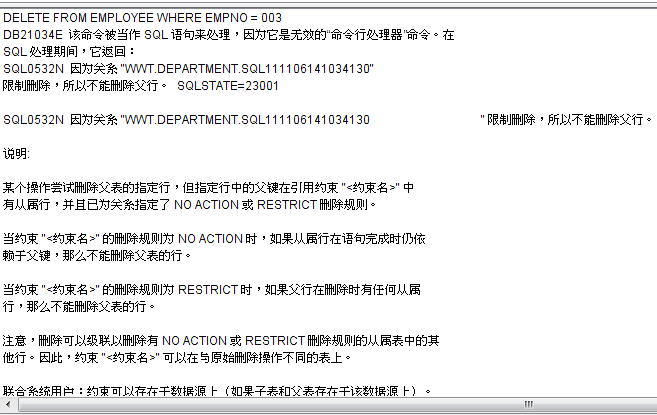
这两个表的约束条件为：Delate set null

在操作之前的部门表的内容如下：



执行以下操作：

如果直接执行删除会出现下面的结果：



因此首先解除约束

ALTER TABLE DEPARTMENT

DROP CONSTRAINT SQL111106141034130

SQL111106105653600为创建表时的约束自动生成的名字

然后才可以删除记录

DELETE FROM EMPLOYEE

WHERE EMPNO = 003;

5.职员表与部门表的约束条件

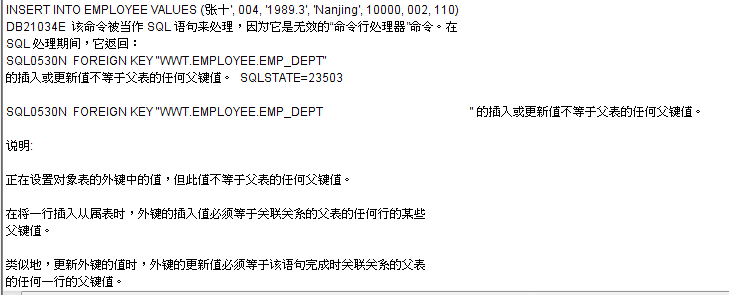
这两个表的约束条件为：Deletion no action

插入如下数据

INSERT INTO EMPLOYEE VALUES

('张十', 004, '1989.3', 'Nanjing', 10000, 002, 110);

部门号不在部门表中，因此会产生错误。



约束1必须在插入数据之后再执行，因为职员表与部门表是相互约束的。

# 触发器

触发器完成的任务是在job中添加一个元组后，在employee中相应的元组的薪水增加2%。

代码如下：

CREATE TRIGGER update\_sal

AFTER INSERT ON JOB

REFERENCING NEW AS N

FOR EACH ROW

UPDATE EMPLOYEE

SET SALARY = SALARY \* 1.02

WHERE EMPNO = N.EMPNO;

END;

执行之前employee的内容如下：



执行下面语句：

INSERT INTO JOB VALUES

(002,202,'2009,10');

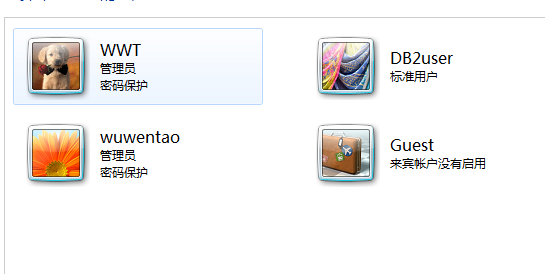
之后查看employee的内容如下：



可见002号员工的工资增加了300元

# 创建新用户并赋予select权限

1. 首先在windows下创建一个新用户DB2user



1. 然后将select权限给该新用户

GRANT SELECT ON EMPLOYEE TO DB2user;

GRANT SELECT ON PROJECT TO DB2user;

GRANT SELECT ON DEPARTMENT TO DB2user;

GRANT SELECT ON JOB TO DB2user;

GRANT SELECT ON FAMILY TO DB2user;

# 六、遇到的问题：

1. 表与表之间若有相互的约束条件，则应先解除约束条件再进行数据插入等操作。否则，不能插入数据。
2. 触发器若选择for each statement，则相应的语句中不能使用new、old等信息。
3. 赋予权限时，没找到如何将所有表一次全部赋予给新用户，现在只能分开赋予。
4. 在创建表的时候，设定的约束条件，系统会自动根据时间戳来取一个名字，根据这个名字可以来取消该约束条件。