**利用DB2，创建一个新的数据库，在该数据库下创建如下表格：**

1. 建立数据库

create database Company;

connect to company;

1. 建表

Employee: 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）；

工号为主键

分别对应ENAME,ENO,EBIRTH,EADDR,ESALARY,MNO,DNO

create table Employee

(ENAME char(20) not null,

ENO smallint not null primary key,

EBIRTH char(20) ,

EADDR char(30),

ESALARY int,

MNO smallint,

DNO smallint);

Department: 部门（部门名称，部门编号，部门负责人的工号）；

部门编号为主键

分别对应DNAME,DNO,ENO

create table Department

(DNAME char(20)not null ,

DNO smallint not null primary key,

ENO smallint);

Project: 项目（项目名称，项目编号，所在城市，主管部门编号）；

项目编号为主键

分别对应PNAME,PNO,PCITY,DNO

create table Project

(PNAME char(30) not null,

PNO smallint not null primary key,

PCITY char(20),

DNO smallint);

Work 工作（职工工号，项目编号，工作时间）；

//职工工号和项目编号为主键

//简单起见先不设置主键

分别对应ENO,PNO,WTIME

create table Work

(ENO smallint,

PNO smallint,

WTIME char(20));

实际做法是：

create table Work

(ENO smallint not null,

PNO smallint not null,

WTIME char(20),

constraint ENO\_PNO primary key(ENO,PNO));

Relative: 家属（职工工号，家属的姓名，家属的性别）。

//职工工号和家属姓名为主键

//简单起见先不设置主键

分别对应ENO,RNAME,RSEX

create table Relative

(ENO smallint,

RNAME char(20),

RSEX char(1));

//设置主键

实际做法是

create table Relative

(ENO smallint not null,

RNAME char(20) not null,

RSEX char(1),

constraint ENO\_RNAME primary key(ENO,RNAME));

至此，建表结束。

1. 插入数据

职工：

insert into Employee values

('A',1001,'90.1.1','东',100,null,101),

('B',1002,'90.2.1','西',100,1001,102),

('C',1003,'90.3.1','南',100,1002,102),

('D',1004,'90.4.1','北',100,1002,103),

('E',1005,'90.5.1','中',100,1004,103);

部门：

insert into Department values

('市场部',101,1001),

('营销部',102,1002),

('业务部',103,1004);

项目：

insert into Project values

('mission impossible',1,'东',101),

('lol',2,'西',102),

('hello world',3,'南',103);

工作：

insert into Work values

(1001,1,'11.11.6-11.11.28'),

(1002,2,'11.12.6-11.12.28'),

(1003,2,'11.11.6-11.11.28'),

(1004,3,'11.12.1-11.12.15'),

(1005,3,'11.11.6-11.11.28');

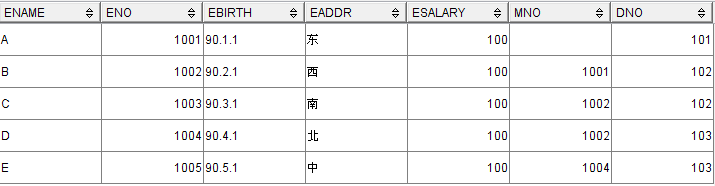
家属：

insert into relative values

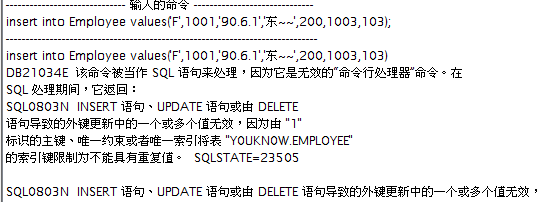
(1001,'Alice','f'),

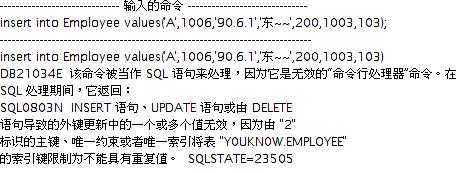
(1002,'Bob','m');

1. 执行违反数据约束的操作，并观察结果
   1. 验证主键约束
      1. 原先的数据



* + 1. 输入insert into Employee values('F',1001,'90.6.1','东~~',200,1003,103);



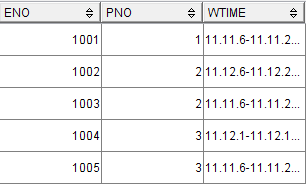
* 1. 验证唯一性约束
     1. 添加alter table Employee add constraint U\_ENAME unique(ENAME);
     2. 输入insert into Employee values('A',1006,'90.6.1','东~~',200,1003,103);
     3. 
  2. 验证外键约束
     1. 添加外键约束

alter table Work add

constraint FK\_PNO foreign key(PNO)

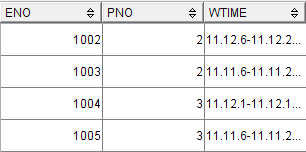
references Project(PNO) on delete cascade;

* + 1. 原先的Work表和Project表





* + 1. 输入Delete from Project where PNO = 1;
    2. 现在的Work表和Project表

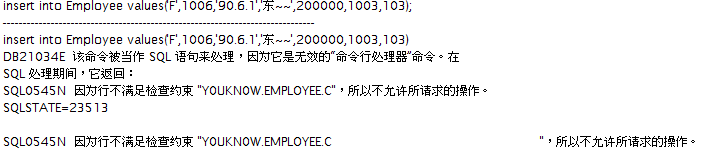




* 1. 验证check约束
     1. alter table Employee add

constraint c CHECK(ESALARY < 100000);

* + 1. 输入 insert into Employee values('F',1006,'90.6.1','东~~',200000,1003,103);



* 1. 验证允许空值约束
     1. 事实上，在建立Employee表时，1001号没有上司，MNO为null

1. 创建一个触发器
   1. create trigger inc\_salary

after insert

on Work

Referencing NEW as N

for each row

update Employee set ESALARY = 1.02 \* ESALARY where

ENO = N.ENO;

* 1. insert into Work values(1002,3,'11.11.11-11.11.31');
  2. 
  3. 工资涨了两元

1. 创建一个用户，赋予工作表查询权限
   1. grant select

on Work

to Test;

* 1. 

/××××

**要求：**

1.表名和属性名用英文字符表示；

2.表中各属性类型自行设计；

3.分析该关系模式中存在的各种数据约束，并在创建表时体现出来；

4.加入样本数据（自行设计），执行违反数据约束的操作，并观察结果；

5.创建一个触发器，当职工参加一个新的项目时，年薪增加2%；

6.创建一个用户，将工作表的查询权限赋予该用户。

×××/