实验 5 数据库设计

一、实验目的

掌握数据库设计基本方法及数据库设计工具。

二、实验内容和要求

掌握数据库设计基本步骤，包括数据库概念结构设计、逻辑结构设计，物理结构设计，数据库模式 SQL 语句生成。能够使用数据库设计工具进行数据库设计。

三、实验重点和难点

实验重点：概念结构设计、逻辑结构设计。

实验难点：逻辑结构设计。逻辑结构设计虽然可以按照一定的规则从概念结构转换而来，但是由于概念结构通常比较抽象，较少考虑更多细节，因此转换而成的逻辑结构还需要进一步调整和优化。逻辑结构承接概念结构和物理结构，处于核心地位，因而是数据库设计的重点，也是难点。

四、实验过程

1、实验材料

数据库设计工具使用 PowerDisigner

2、概述

1）项目题材：人事管理系统；

2）项目背景： 广州某公司新成立需要设计一个数据库保存和管理部门及员工等人事管理信息的数据库；

3）需求分析：

a.系统功能要求：

员工各种信息的输入，包括员工的基本信息、子女保险信息、职称等。

员工各种信息的修改；

对于转出、辞职、辞退、退休员工信息的删除；

按照一定的条件，查询、统计符合条件的员工信息；至少应该包括每个员工详细信息的查询、按工作岗位或工作部门查询等;

对查询、统计的结果打印输出

b.数据库要求：在数据库中至少应该包含下列数据表：

员工基本信息表；

员工子女保险情况表，反映员工子女投保的信息；

企业工作岗位表；

企业部门信息表。

2、数据库设计思路

员工的基本信息有：员工 id，姓名，年龄，性别；（主键用下划线标出）

员工子女信息：姓名，年龄

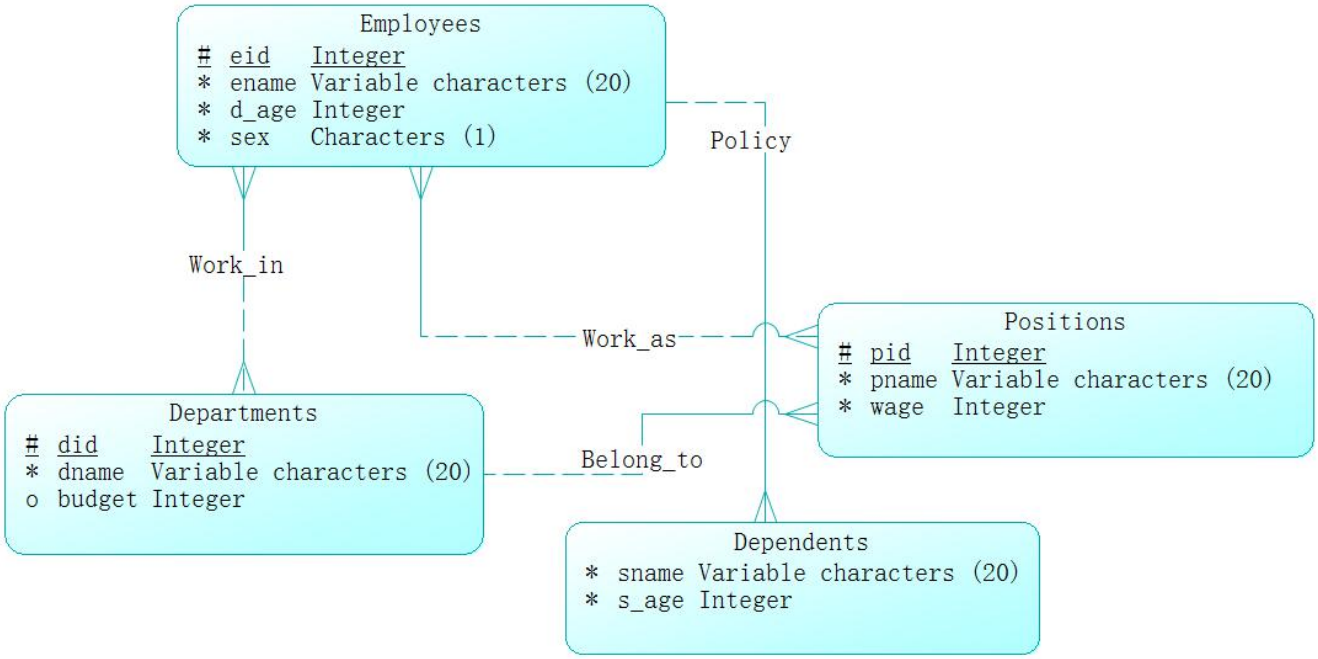
部门信息有：部门 id，部门名称，预算；

岗位信息有：岗位 id，岗位名称，工资；

各个实体关系：

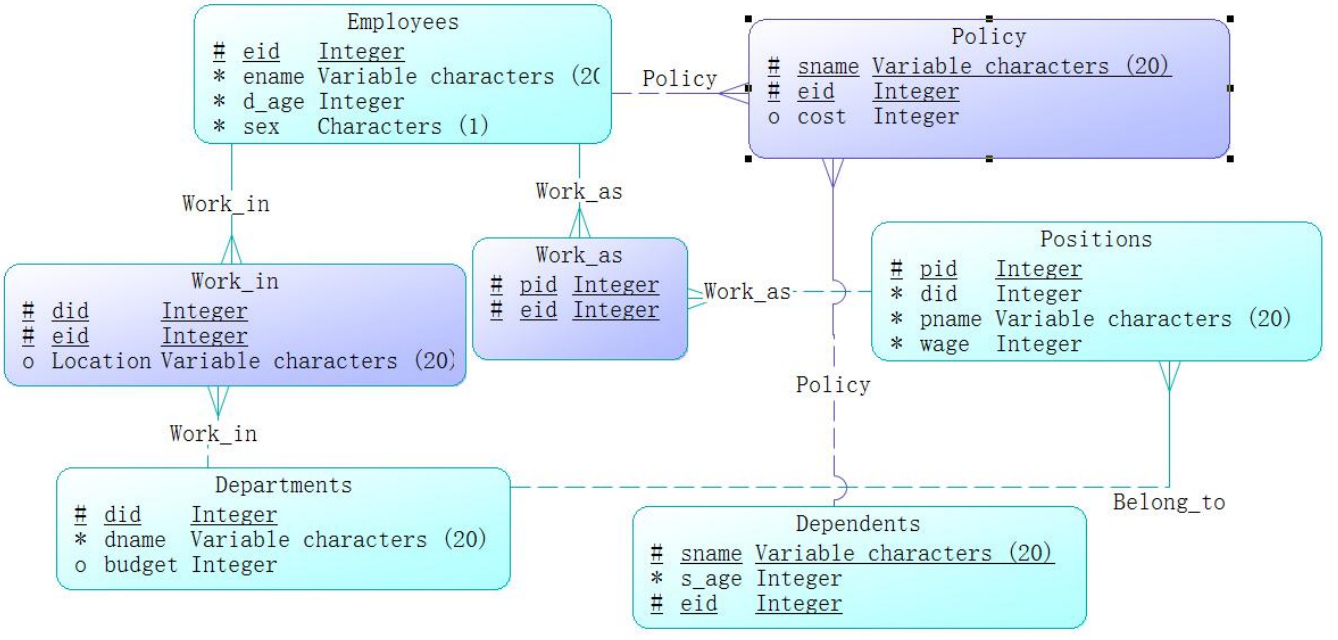
Work_in:员工在某部门工作
Work_as:员工担任工作职位
Belong:部门提供
Policy:员工子女信息是弱实体集，依赖于员工实体

3、概念结构设计（画出 E-R 模型图）



4、数据库逻辑结构设计

1.把 E-R 模型图转换为关系表



2.描述每一个基本表关系

存在某个员工进入多个部门和某个部门有多个员工的情况，Employees 和 Departments 关系是多对多；同理 Employees 和 Positions 关系也是多对多；Dependents 弱实体集，与 Employees 多对一；Positions 与 Departments 是多对一；

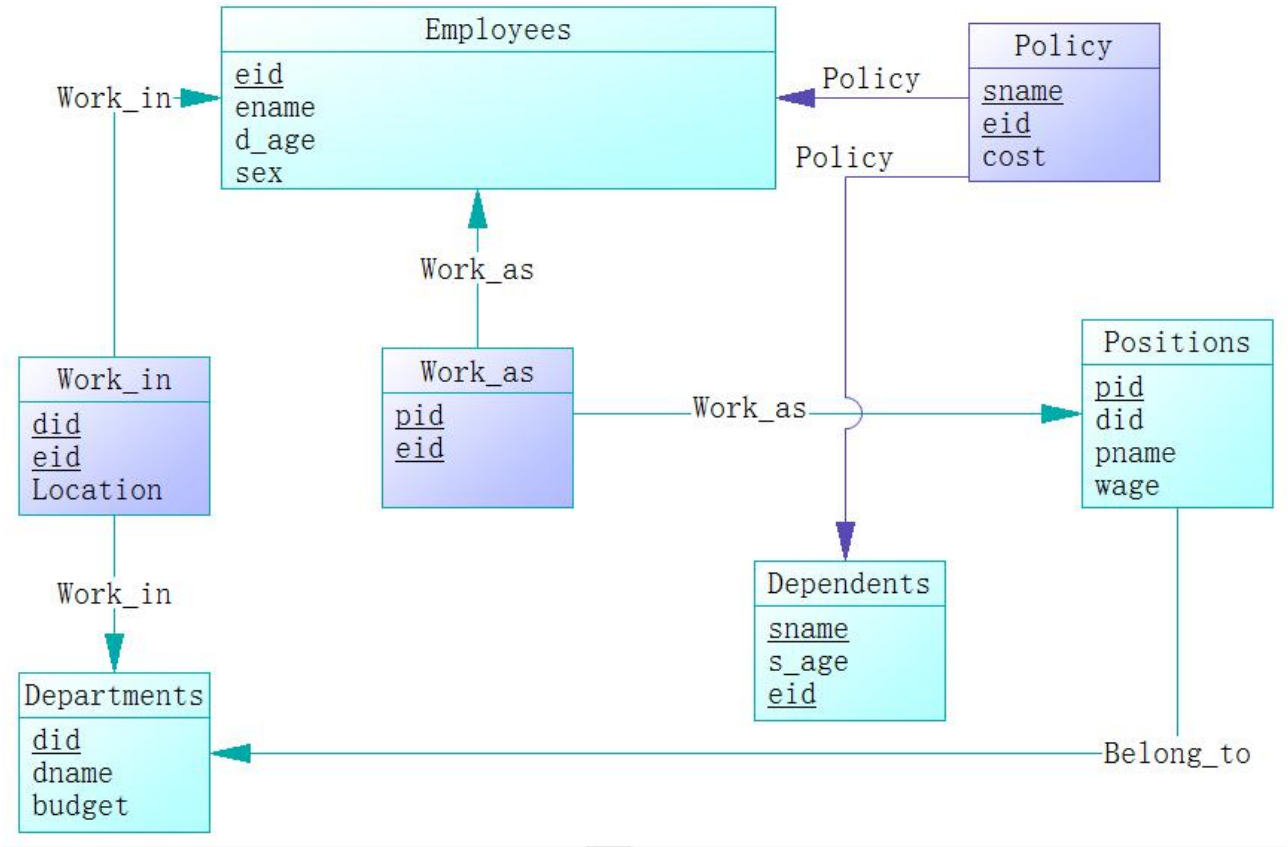
联系表：Work_in：外键 did 和 eid 作为主标识符

Work_as：外键 pid 和 eid 作为主标识符

Policy：外键 pid

使用 pd 转换出来的逻辑结构图没有 Policy 关系表，于是自行优化和添加 Policy 关系表。

5、数据库物理结构设计



1.定义索引

由于在公司中使用 id 来查询增删改各种信息的情况比较多，因此主要基于 B+树、以 id 为聚簇索引，加快查找速度

Employees：聚簇索引 eid，B+树索引

Dependents：聚簇索引 <eid, sname>，B+树索引

Policy：聚簇索引 <eid,sname>，B+树索引

Departments：聚簇索引 did，B+树索引

Positions：聚簇索引 pid，B+树索引

Work_in：聚簇索引 <eid,did>，B+树索引

Work_as：聚簇索引 <eid,pid>，B+树索引
Belong_to：聚簇索引 <did,pid>，B+树索引

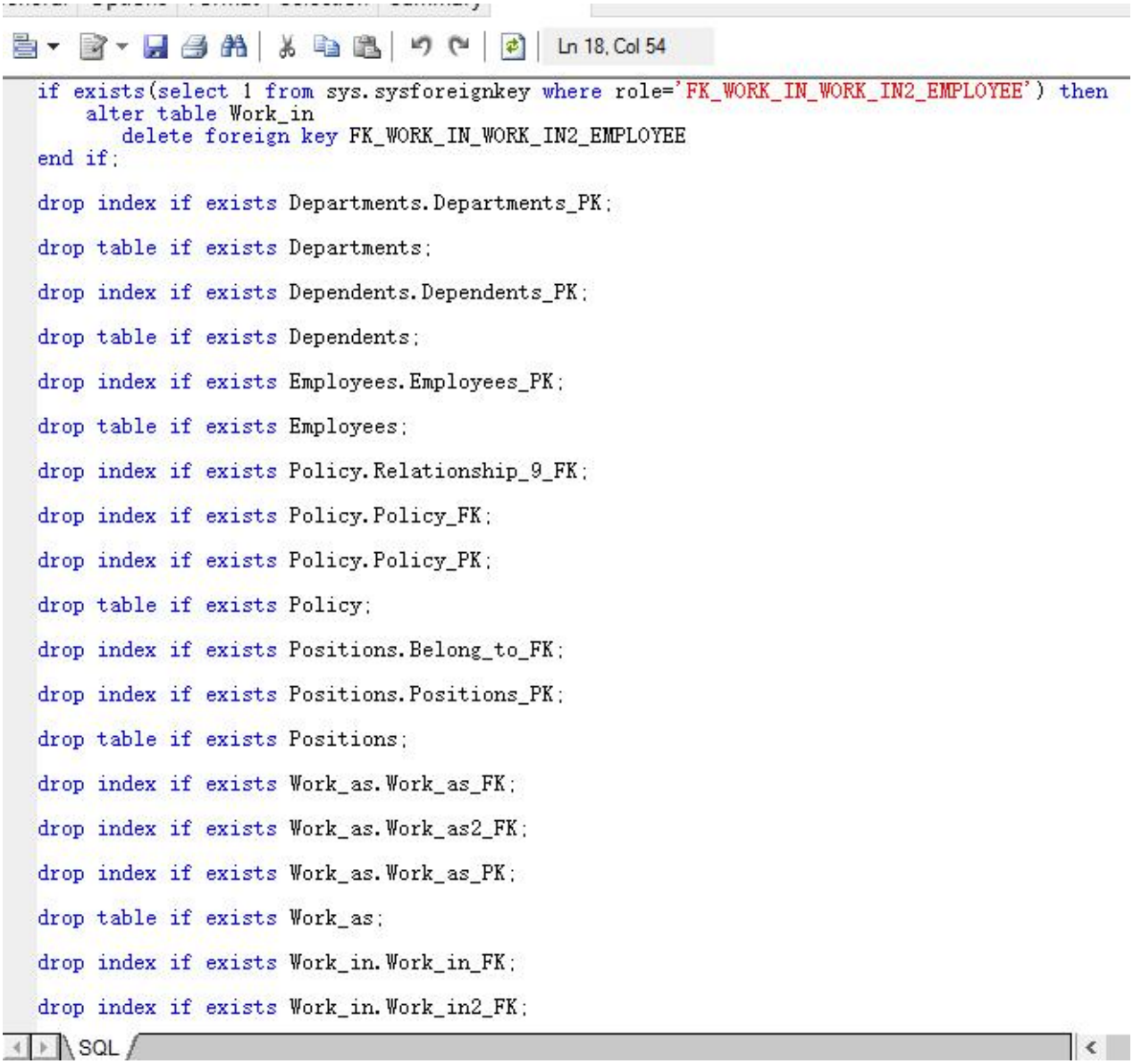
2 定义权限

仅公司高层管理人员有权限（密码）可以登陆数据库系统并修改人事管理信息，部门经理可以进行查询本部门员工的相关信息

6、数据库模式 SQL 语句生成

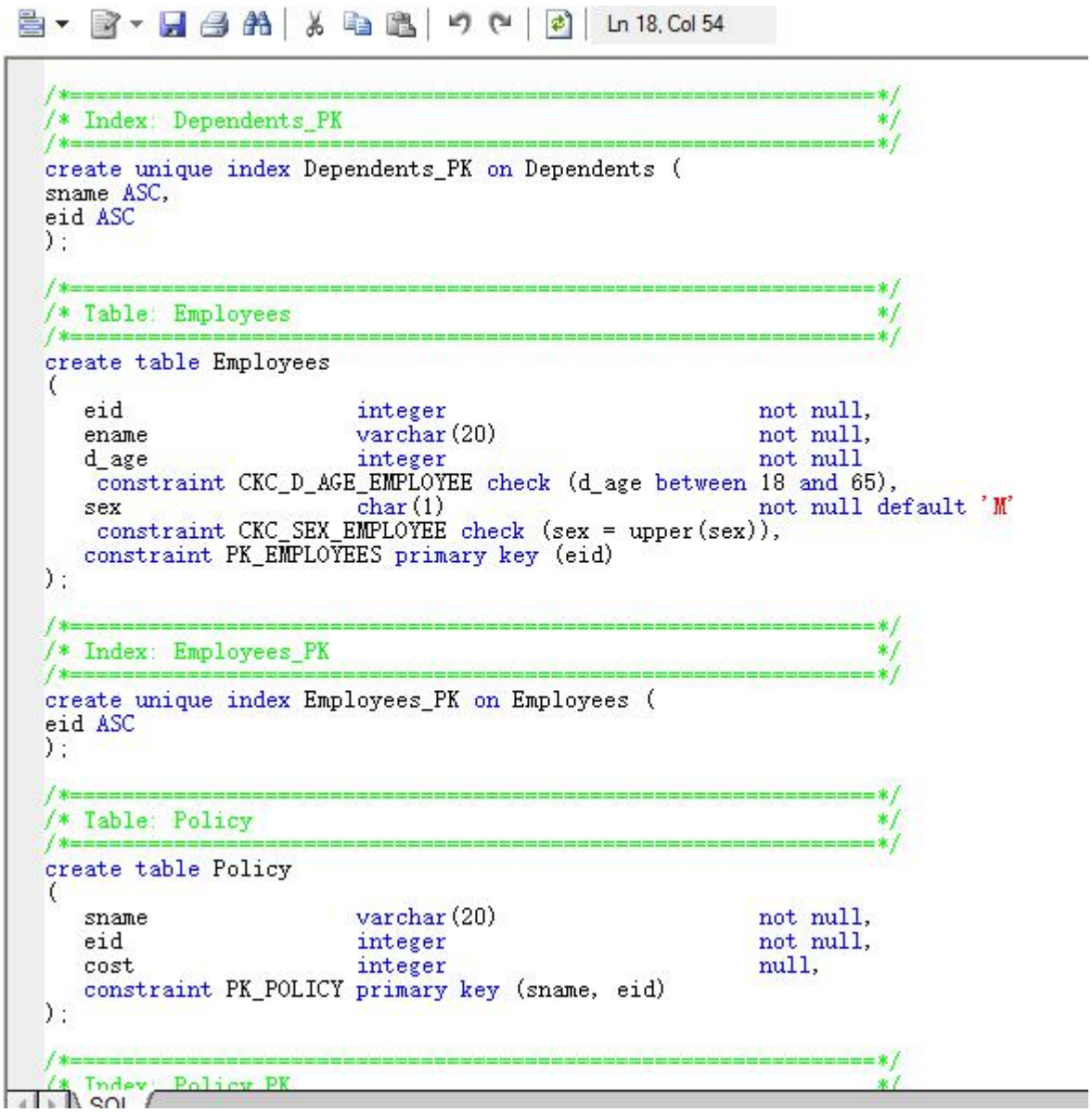
截图如下：

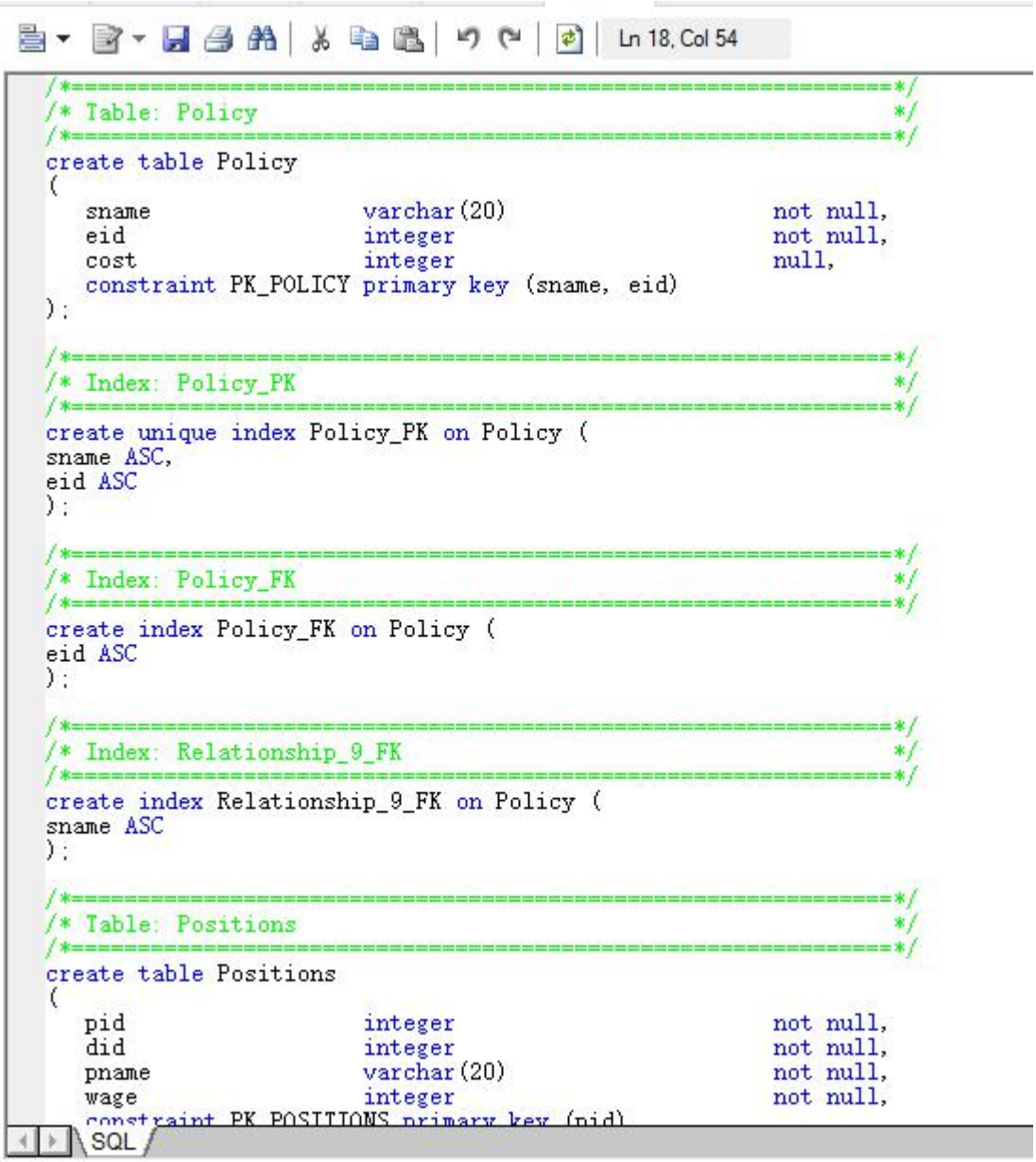
```
if exists(select 1 from sys.sysforeignkey where role='FK_POLICY_POLICY_EMPLOYEE') then
    alter table Policy
       delete foreign key FK_POLICY_POLICY_EMPLOYEE
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_POLICY_RELATIONS_DEPENDEN') then
    alter table Policy
       delete foreign key FK_POLICY_RELATIONS_DEPENDEN
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_POSITION_BELONG_TO_DEPARTME') then
    alter table Positions
       delete foreign key FK_POSITION_BELONG_TO_DEPARTME
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_WORK_AS_WORK_AS_POSITION') then
    alter table Work_as
       delete foreign key FK_WORK_AS_WORK_AS_POSITION
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_WORK_AS_WORK_AS2_EMPLOYEE') then
    alter table Work_as
       delete foreign key FK_WORK_AS_WORK_AS2_EMPLOYEE
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_WORK_IN_WORK_IN_DEPARTME') then
    alter table Work_in
       delete foreign key FK_WORK_IN_WORK_IN_DEPARTME
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_WORK_IN_WORK_IN2_EMPLOYEE') then
    alter table Work_in
       delete foreign key FK_WORK_IN_WORK_IN2_EMPLOYEE
end if;

drop index if exists Departments.Departments_PK;

drop table if exists Departments;

drop index if exists Dependents.Dependents_PK;
```

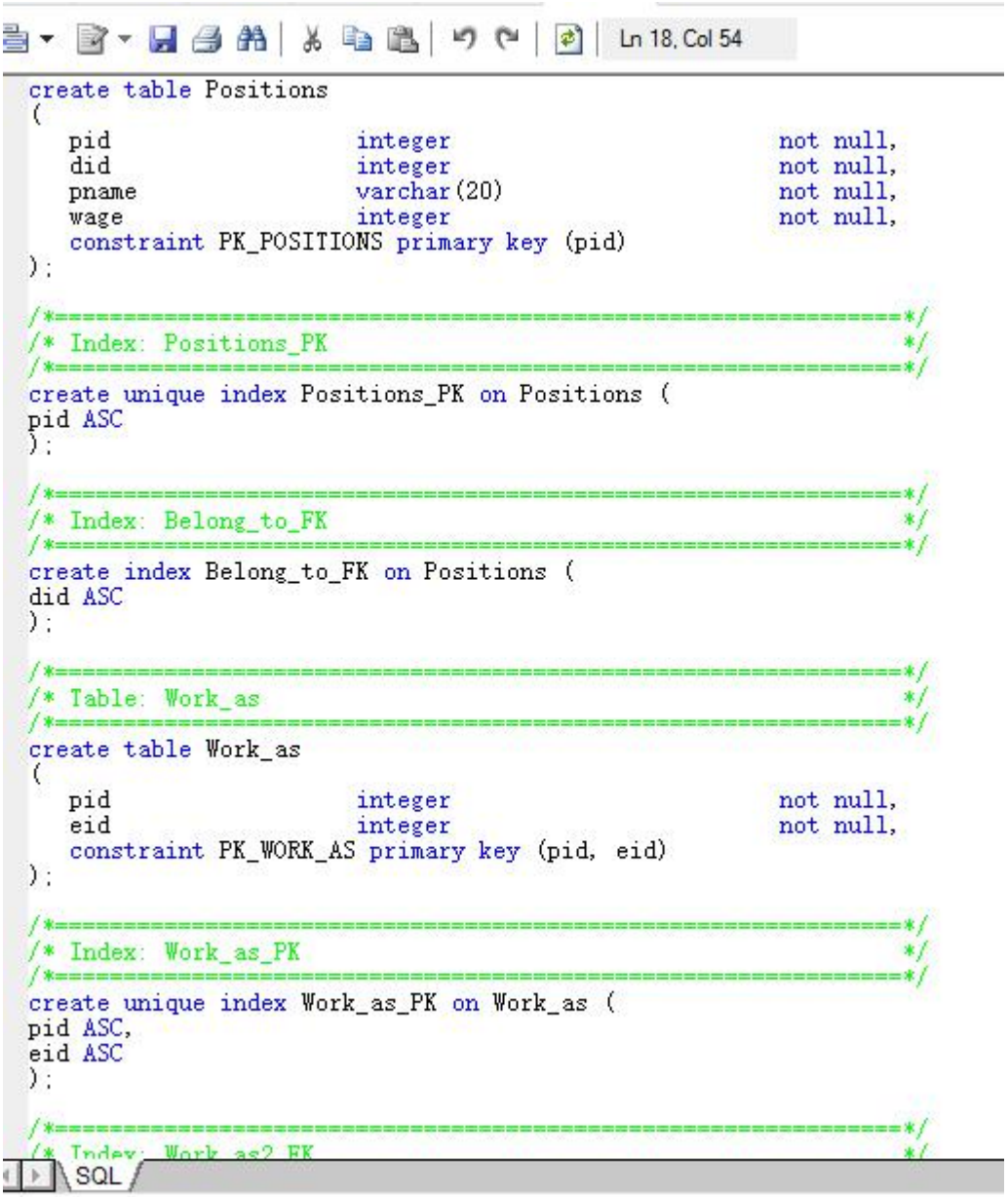

Ln 18, Col 54
if exists(select 1 from sys.sysforeignkey where role='FK_WORK_IN_WORK_IN2_EMPLOYEE') then
alter table Work_in
delete foreign key FK_WORK_IN_WORK_IN2_EMPLOYEE
end if;
drop index if exists Departments.Departments_PK;
drop table if exists Departments;
drop index if exists Dependents.Dependents_PK;
drop table if exists Dependents;
drop index if exists Employees.Employees_PK;
drop table if exists Employees;
drop index if exists Policy.Relationship_9_FK;
drop index if exists Policy.Policy_FK;
drop index if exists Policy.Policy_PK;
drop table if exists Policy;
drop index if exists Positions.Belong_to_FK;
drop index if exists Positions.Positions_PK;
drop table if exists Positions;
drop index if exists Work_as.Work_as_FK;
drop index if exists Work_as.Work_as2_FK;
drop index if exists Work_as.Work_as_PK;
drop table if exists Work_as;
drop index if exists Work_in.Work_in_FK;
drop index if exists Work_in.Work_in2_FK;
SQL

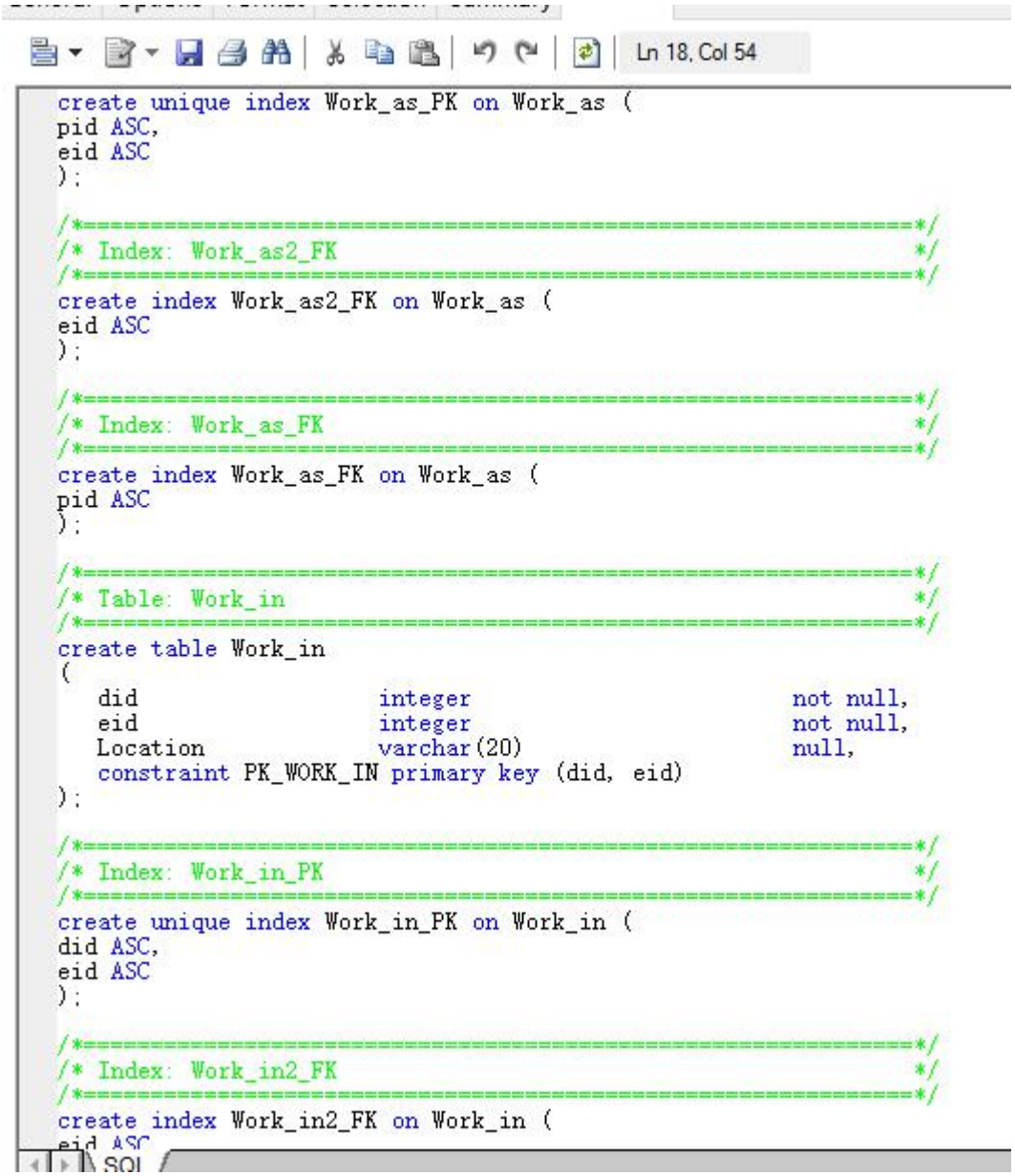
```sql
drop table if exists Work_as;

drop index if exists Work_in.Work_in_FK;

drop index if exists Work_in.Work_in2_FK;

drop index if exists Work_in.Work_in_PK;

drop table if exists Work_in;

/*==============================================================*/
/* Table: Departments                                           */
/*==============================================================*/
create table Departments
(
   did                  integer                        not null,
   dname                varchar(20)                    not null,
   budget               integer                        null,
   constraint PK_DEPARTMENTS primary key (did)
);

/*==============================================================*/
/* Index: Departments_PK                                        */
/*==============================================================*/
create unique index Departments_PK on Departments (
did ASC
);

/*==============================================================*/
/* Table: Dependents                                            */
/*==============================================================*/
create table Dependents
(
   sname                varchar(20)                    not null,
   s_age                integer                        not null,
   eid                  integer                        not null,
   constraint PK_DEPENDENTS primary key (sname, eid)
);

/*==============================================================*/
/* Index: Dependents_PK                                         */
/*==============================================================*/
```

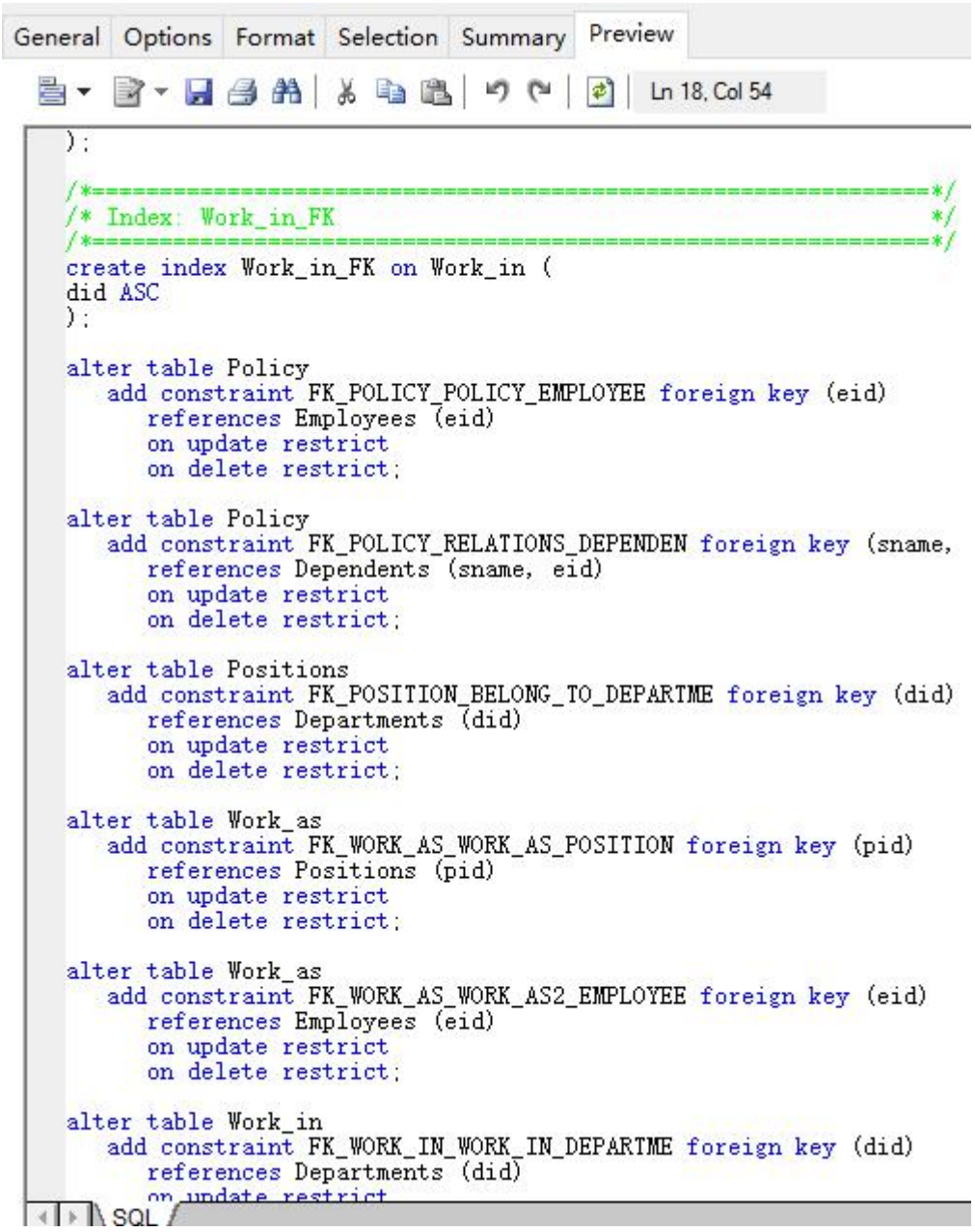
SQL


Ln 18, Col 54
/*==*/
/* Index: Dependents_PK   */
/*==*/
create unique index Dependents_PK on Dependents (
sname ASC,
eid ASC
);
/*==*/
/* Table: Employees   */
/*==*/
create table Employees
(
eid integer not null,
ename varchar(20) not null,
d_age integer not null
constraint CKC_D_AGE_EMPLOYEE check (d_age between 18 and 65),
sex char(1) not null default 'M'
constraint CKC_SEX_EMPLOYEE check (sex = upper(sex)),
constraint PK_EMPLOYEES primary key (eid)
);
/*==*/
/* Index: Employees_PK   */
/*==*/
create unique index Employees_PK on Employees (
eid ASC
);
/*==*/
/* Table: Policy   */
/*==*/
create table Policy
(
sname varchar(20) not null,
eid integer not null,
cost integer null,
constraint PK_POLICY primary key (sname, eid)
);
/*==*/
SQL


Ln 18, Col 54
/*==*/
/* Table: Policy */
/*==*/
create table Policy
(
sname varchar(20) not null,
eid integer not null,
cost integer null,
constraint PK_POLICY primary key (sname, eid)
);
/*==*/
/* Index: Policy_PK */
/*==*/
create unique index Policy_PK on Policy (
sname ASC,
eid ASC
);
/*==*/
/* Index: Policy_FK */
/*==*/
create index Policy_FK on Policy (
eid ASC
);
/*==*/
/* Index: Relationship_9_FK */
/*==*/
create index Relationship_9_FK on Policy (
sname ASC
);
/*==*/
/* Table: Positions */
/*==*/
create table Positions
(
pid integer not null,
did integer not null,
pname varchar(20) not null,
wage integer not null,
SQL


Ln 18, Col 54
create table Positions
(
pid integer not null,
did integer not null,
pname varchar(20) not null,
wage integer not null,
constraint PK_POSITIONS primary key (pid)
);
/* Index: Positions_PK */
create unique index Positions_PK on Positions (
pid ASC
);
/* Index: Belong_to_FK */
create index Belong_to_FK on Positions (
did ASC
);
/* Table: Work_as */
create table Work_as
(
pid integer not null,
eid integer not null,
constraint PK_WORK_AS primary key (pid, eid)
);
/* Index: Work_as_PK */
create unique index Work_as_PK on Work_as (
pid ASC,
eid ASC
);
SQL


Ln 18, Col 54
create unique index Work_as_PK on Work_as (
pid ASC,
eid ASC
);
/*==*/
/* Index: Work_as2_FK */
/*==*/
create index Work_as2_FK on Work_as (
eid ASC
);
/*==*/
/* Index: Work_as_FK */
/*==*/
create index Work_as_FK on Work_as (
pid ASC
);
/*==*/
/* Table: Work_in */
/*==*/
create table Work_in
(
did integer not null,
eid integer not null,
Location varchar(20) null,
constraint PK_WORK_IN primary key (did, eid)
);
/*==*/
/* Index: Work_in_PK */
/*==*/
create unique index Work_in_PK on Work_in (
did ASC,
eid ASC
);
/*==*/
/* Index: Work_in2_FK */
/*==*/
create index Work_in2_FK on Work_in (
eid ASC
SQL


General
Options
Format
Selection
Summary
Preview
Ln 18, Col 54
);
/*==*/
/* Index: Work_in_FK */
/*==*/
create index Work_in_FK on Work_in (
did ASC
);
alter table Policy
add constraint FK_POLICY_POLICY_EMPLOYEE foreign key (eid)
references Employees (eid)
on update restrict
on delete restrict;
alter table Policy
add constraint FK_POLICY_RELATIONS_DEPENDEN foreign key (sname,
references Dependents (sname, eid)
on update restrict
on delete restrict;
alter table Positions
add constraint FK_POSITION_BELONG_TO_DEPARTME foreign key (did)
references Departments (did)
on update restrict
on delete restrict;
alter table Work_as
add constraint FK_WORK_AS_WORK_AS_POSITION foreign key (pid)
references Positions (pid)
on update restrict
on delete restrict;
alter table Work_as
add constraint FK_WORK_AS_WORK_AS2_EMPLOYEE foreign key (eid)
references Employees (eid)
on update restrict
on delete restrict;
alter table Work_in
add constraint FK_WORK_IN_WORK_IN_DEPARTME foreign key (did)
references Departments (did)
SQL

```
   add constraint FK_WORK_AS_WORK_AS_POSITION foreign key (pid)
      references Positions (pid)
      on update restrict
      on delete restrict;

alter table Work_as
   add constraint FK_WORK_AS_WORK_AS2_EMPLOYEE foreign key (eid)
      references Employees (eid)
      on update restrict
      on delete restrict;

alter table Work_in
   add constraint FK_WORK_IN_WORK_IN_DEPARTME foreign key (did)
      references Departments (did)
      on update restrict
      on delete restrict;

alter table Work_in
   add constraint FK_WORK_IN_WORK_IN2_EMPLOYEE foreign key (eid)
      references Employees (eid)
      on update restrict
      on delete restrict;
```

SQL

五、实验反思和总结

在本次实验中，由于对之前学过的数据库知识遗忘以及对 pd 不够熟练，不得不查看教材和上网查找资料，导致实验进展很慢，比如，在设计概念结构中，遇到不能创建联系集的情况，后来查找资料后才了解到 pd 是在逻辑结构才能创建联系集。虽然本次实验是单人实验，和大神同学交流收获颇多，虽然和该同学实验所选题材不同，但他考虑到很多方面的事情，给出一些有用的建议，使得我可以完善我的实验。

通过这次实验，我的实验存在做的过于简单，没有做好足够的优化等缺点，可见在实际情况中，数据库的创建应该更加复杂，更需要一个方便可用的数据库设计工具例如 pd。也发现自己对数据库知识掌握不牢固，我还有很多知识不懂，更应该抓紧时间学习。