数据库系统 week7 作业 2211290 姚知言

1.设计一个员工管理系统

该公司有很多子公司（branchoffice），每个子公司以子公司 ID（officeID）标识，除此之外还要记录公司名（officename），公司地址（officelocation）。

公司下设很多部门（department），以部门 ID（departmentID）标识，除此之外还要记录部门名（departmentname）。

部门下雇佣雇员（employee），雇员分为技术员（technicist）和管理者（management），通过雇员 ID（employeeID）进行标识。

对于技术员，除了雇员 ID 以外，还需要存储雇员名（employeename），入职时间（enrollmentdate），工资（salary），工作岗位（job），性别（sex），雇员类型（category），技术员等级（techlevel）。

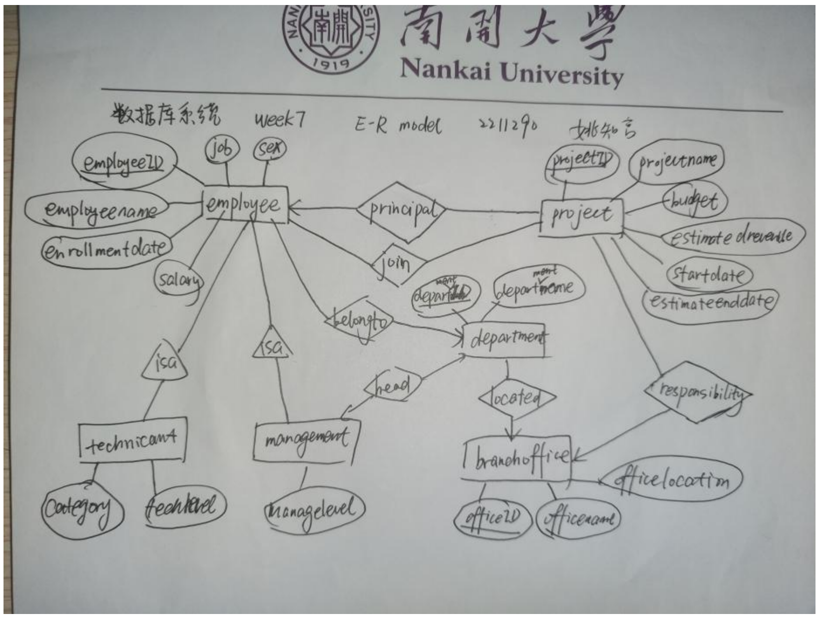
对于管理者，除了雇员 ID 以外，还需要存储雇员名（employeename），入职时间（enrollmentdate），工资（salary），工作岗位（job），性别（sex），管理者等级（managelevel）。

每个部门需要有一个部长，由一名管理者担任，一个管理者仅能担任一个部门的部长。

公司中由很多项目，以项目 ID(projectID)标识，还要存储项目名(projectname)，预算(budget)，预计收入（estimatedrevenue），开始时间（startdate），预计结束时间（estimatedenddate）。

每个项目需要很多雇员参加（joinn），也需要一个负责人（principal），一个负责人可以同时负责多个项目，还需要一个责任分公司（responsibility），每个分公司可以同时负责多个项目。

2.a）该领域的 ER 图如下



b）关系模式如下

主键加下划线，外键字体为黄色。

technicant(employeeID,category,techlevel)

management(employeeID,managelevel,departmentID)

employee(employeeID,employeename,enrollmentdate,salary,job,sex,departmentID)

joinn(employeeID,projectID)

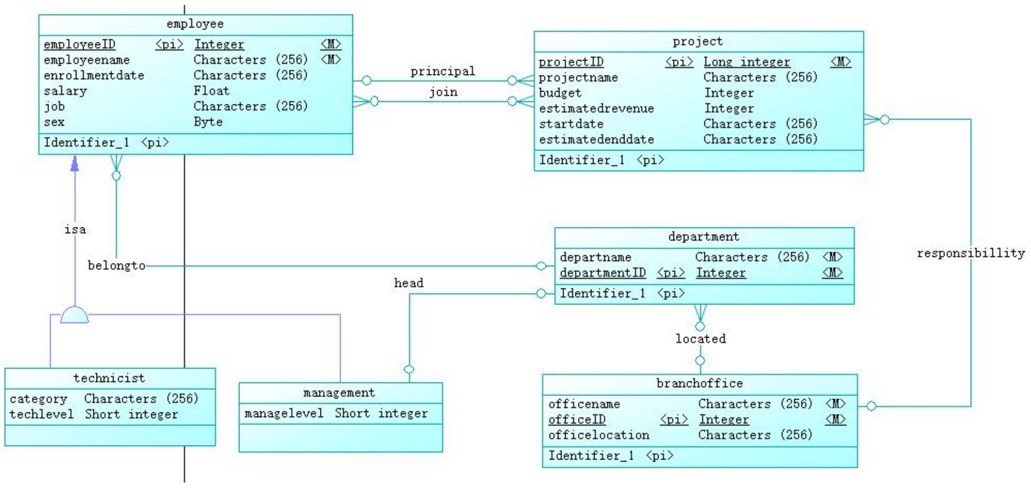
project(projectID,projectname,budget,estimatedrevenue,startdate,estimatedenddate,officeID,employeeID)

department(departmentID,departmentname,officeID,employeeID)

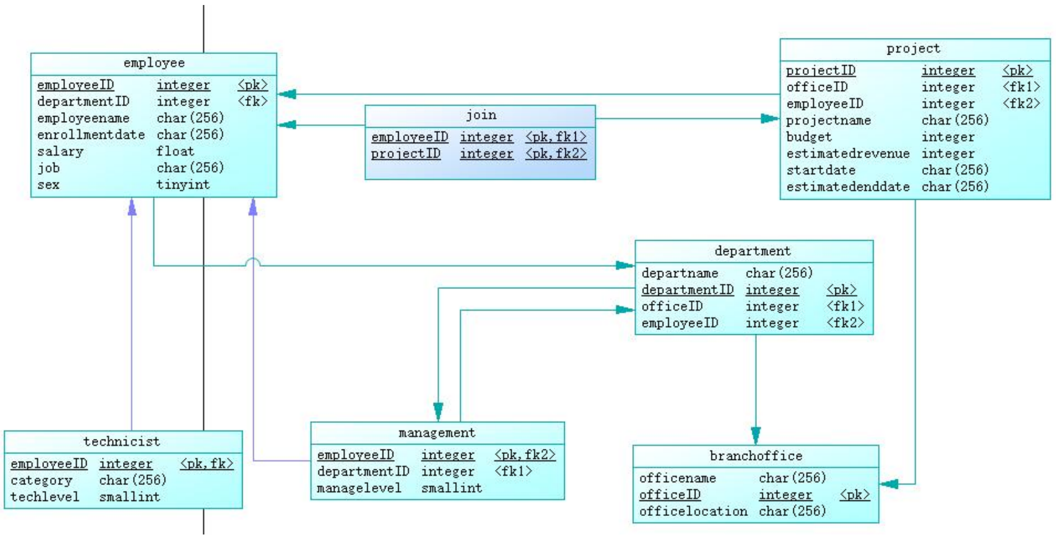
branchoffice(officeID,officename,officelocation) c）用 SQL 语句创建关系模式

```
create table branchoffice(
   officeID int primary key,
   officename char(20),
   officelocation char(256) );
create table employee(
   employeeID int primary key,
   employname char(20) not null,
   enrollmentdate char(20),
   salary float,
   job char(40),
   sex byte,
   departmentID int);
create table department(
   departmentID int primary key,
   departname char(40),
   officeID int,
   employeeID int,
   foreign key (employeeID) references employee(employeeID) on update cascade on delete
cascade);
create table joinn(
   employeeID int,
   projectID int,
   primary key (employeeID,projectID),
   foreign key (employeeID) references employee(employeeID) on update cascade on delete
cascade));
create table technicist(
   techlevel smallint,
   category char(20),
   employeeID int primary key,
   foreign key (employeeID) references employee(employeeID) on update cascade on delete
cascade);
create table management(
   employeeID int primary key,
   managelevel smallint,
   boss integer,
   foreign key (boss) references department(departmentID) on update cascade on delete
cascade,
   foreign key (employeeID) references employee(employeeID) on update cascade on delete
cascade);
create table project(
   projectID int primary key,
   resoffice int,
   principle int,
    projectname char(40),
    budget int,
    estimatedrevenue int,
    startdate char(20),
    enddate char(20),
    foreign key (resoffice) references branchoffice(officeID) on update cascade on delete cascade,
    foreign key(principle) references employee(employeeID) on update cascade on delete cascade);
alter table employee add foreign key(departmentID) references department(departmentID) on update cascade on delete cascade;
alter table department add foreign key(officeID) reference branchoffice(officeID) on update cascade on delete cascade;
alter table joinn add foreign key(projectID) reference project(projectID) on update cascade on delete cascade;
```
d）查询语句样例

单表查询：查询 sex=0 的雇员
```
select employeeID
from employee
where sex=0;
```
多表连接查询：查询参加项目的雇员，参加多个项目的雇员可能返回多次
```
select employeename
from employee,joinn
where joinn.employeeID=employee.enployeeID;
```
多表嵌套查询：同上，一个雇员返回一次
```
select employeename
from employee
Where employeeID in(select employeeID from joinn);
```
EXISTS 查询：同上
```
select employeename
from employee
where exists (select * from joinn where employee.employeeID=joinn.employeeID);
```
聚合操作查询：查找 ID111 的雇员加入了几个项目
```
select count(distinct projectID)
from joinn
Where employeeID=111;
```
3.a）该领域的 ER 图如下


employee
employeeID <pi> Integer <M>
employeename Characters (256) <M>
enrollmentdate Characters (256)
salary Float
job Characters (256)
sex Byte
Identifier_1 <pi>
principal
join
project
projectID <pi> Long integer <M>
projectname Characters (256)
budget Integer
estimatedrevenue Integer
startdate Characters (256)
estimatedenddate Characters (256)
Identifier_1 <pi>
responsibillity
isa
belongto
head
department
departname Characters (256) <M>
departmentID <pi> Integer <M>
Identifier_1 <pi>
located
technicist
category Characters (256)
techlevel Short integer
management
managelevel Short integer
branchoffice
officename Characters (256) <M>
officeID <pi> Integer <M>
officelocation Characters (256)
Identifier_1 <pi>

b）该领域的关系模型图如下



C）生成创建数据库语句如下

```
if exists(select 1 from sys.sysforeignkey where role='FK_DEPARTME_HEAD_MANAGEME') then
    alter table department
        delete foreign key FK_DEPARTME_HEAD_MANAGEME
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_DEPARTME_LOCATED_BRANCHOF') then
    alter table department
        delete foreign key FK_DEPARTME_LOCATED_BRANCHOF
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_EMPLOYEE_BELONGTO_DEPARTME') then
    alter table employee
        delete foreign key FK_EMPLOYEE_BELONGTO_DEPARTME
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_joinn_joinn_EMPLOYEE') then
    alter table "joinn"
        delete foreign key FK_joinn_joinn_EMPLOYEE
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_joinn_joinn2_PROJECT') then
```

```
    alter table "joinn"
        delete foreign key FK_joinn_joinn2_PROJECT
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_MANAGEME_HEAD2_DEPARTME') then
    alter table management
        delete foreign key FK_MANAGEME_HEAD2_DEPARTME
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_MANAGEME_ISA2_EMPLOYEE') then
    alter table management
        delete foreign key FK_MANAGEME_ISA2_EMPLOYEE
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_PROJECT_PRINCIPAL_EMPLOYEE') then
    alter table project
        delete foreign key FK_PROJECT_PRINCIPAL_EMPLOYEE
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_PROJECT_RESPONSIB_BRANCHOF') then
    alter table project
        delete foreign key FK_PROJECT_RESPONSIB_BRANCHOF
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_TECHNICI_ISA_EMPLOYEE') then
    alter table technicist
        delete foreign key FK_TECHNICI_ISA_EMPLOYEE
end if;
drop index if exists branchoffice.branchoffice_PK;
drop table if exists branchoffice;
drop index if exists department.head_FK;
drop index if exists department.located_FK;
drop index if exists department.department_PK;
drop table if exists department;
drop index if exists employee.belongto_FK;
drop index if exists employee.employee_PK;
drop table if exists employee;
drop index if exists "joinn".joinn2_FK;
drop index if exists "joinn".joinn_FK;
drop index if exists "joinn".joinn_PK;
drop table if exists "joinn";
drop index if exists management.head2_FK;
drop index if exists management.management_PK;
drop table if exists management;
drop index if exists project.principal_FK;
drop index if exists project.responsibillity_FK;
drop index if exists project.project_PK;
drop table if exists project;
drop index if exists technicist.technicist_PK;
drop table if exists technicist;
/*==============================================================*/
/* Table: branchoffice                                          */
/*==============================================================*/
create table branchoffice
(
    officename           char(256)                      not null,
    officeID             integer                        not null,
    officelocation       char(256)                      null,
    constraint PK_BRANCHOFFICE primary key (officeID)
);
/*==============================================================*/
/* Index: branchoffice_PK                                       */
/*==============================================================*/
create unique index branchoffice_PK on branchoffice (
officeID ASC
);
/*==============================================================*/
/* Table: department                                            */
/*==============================================================*/
create table department
(
    departname           char(256)                      not null,
    departmentID         integer                        not null,
    officeID             integer                        null,
    employeeID           integer                        null,
    constraint PK_DEPARTMENT primary key (departmentID)
);
/*==============================================================*/
/* Index: department_PK                                         */
/*==============================================================*/
create unique index department_PK on department (
departmentID ASC
);
/*==============================================================*/
/* Index: located_FK                                            */
/*==============================================================*/
create index located_FK on department (
officeID ASC
);
/*==============================================================*/
/* Index: head_FK                                               */
/*==============================================================*/
create index head_FK on department (
employeeID ASC
);
/*==============================================================*/
/* Table: employee                                              */
/*==============================================================*/
create table employee
(
   employeeID           integer                        not null,
   departmentID         integer                        null,
   employeename         char(256)                      not null,
   enrollmentdate       char(256)                      null,
   salary               float                          null,
   job                  char(256)                      null,
   sex                  tinyint                        null,
   constraint PK_EMPLOYEE primary key (employeeID)
);
/*==============================================================*/
/* Index: employee_PK                                           */
/*==============================================================*/
create unique index employee_PK on employee (
employeeID ASC
);
/*==============================================================*/
/* Index: belongto_FK                                           */
/*==============================================================*/
create index belongto_FK on employee (
departmentID ASC
);
/*==============================================================*/
/* Table: "joinn"                                               */
/*==============================================================*/
create table "joinn"
(
   employeeID           integer                        not null,
   projectID            integer                        not null,
   constraint PK_joinn primary key clustered (employeeID, projectID)
);
/*==============================================================*/
/* Index: joinn_PK                                              */
/*==============================================================*/
create unique clustered index joinn_PK on "joinn" (
employeeID ASC,
projectID ASC
);
```

```
/*==============================================================*/
/* Index: joinn_FK                                              */
/*==============================================================*/
create index joinn_FK on "joinn" (
employeeID ASC
);
/*==============================================================*/
/* Index: joinn2_FK                                             */
/*==============================================================*/
create index joinn2_FK on "joinn" (
projectID ASC
);
/*==============================================================*/
/* Table: management                                            */
/*==============================================================*/
create table management
(
   employeeID           integer                        not null,
   departmentID         integer                        null,
   managelevel          smallint                       null,
   constraint PK_MANAGEMENT primary key clustered (employeeID)
);
/*==============================================================*/
/* Index: management_PK                                         */
/*==============================================================*/
create unique clustered index management_PK on management (
employeeID ASC
);
/*==============================================================*/
/* Index: head2_FK                                              */
/*==============================================================*/
create index head2_FK on management (
departmentID ASC
);
/*==============================================================*/
/* Table: project                                               */
/*==============================================================*/
create table project
(
   projectID            integer                        not null,
   officeID             integer                        null,
   employeeID           integer                        null,
   projectname          char(256)                      null,
   budget               integer                        null,
    estimatedrevenue        integer                              null,
    startdate               char(256)                            null,
    estimatedenddate        char(256)                            null,
    constraint PK_PROJECT primary key (projectID)
);
/*==============================================================*/
/* Index: project_PK                                            */
/*==============================================================*/
create unique index project_PK on project (
projectID ASC
);
/*==============================================================*/
/* Index: responsibillity_FK                                    */
/*==============================================================*/
create index responsibillity_FK on project (
officeID ASC
);
/*==============================================================*/
/* Index: principal_FK                                          */
/*==============================================================*/
create index principal_FK on project (
employeeID ASC
);
/*==============================================================*/
/* Table: technicist                                            */
/*==============================================================*/
create table technicist
(
    employeeID              integer                              not null,
    category                char(256)                            null,
    techlevel               smallint                             null,
    constraint PK_TECHNICIST primary key clustered (employeeID)
);
/*==============================================================*/
/* Index: technicist_PK                                         */
/*==============================================================*/
create unique clustered index technicist_PK on technicist (
employeeID ASC
);
alter table department
    add constraint FK_DEPARTME_HEAD_MANAGEME foreign key (employeeID)
        references management (employeeID)
        on update restrict
        on delete restrict;
```

```
alter table department
   add constraint FK_DEPARTME_LOCATED_BRANCHOF foreign key (officeID)
      references branchoffice (officeID)
      on update restrict
      on delete restrict;
alter table employee
   add constraint FK_EMPLOYEE_BELONGTO_DEPARTME foreign key (departmentID)
      references department (departmentID)
      on update restrict
      on delete restrict;
alter table "joinn"
   add constraint FK_joinn_joinn_EMPLOYEE foreign key (employeeID)
      references employee (employeeID)
      on update restrict
      on delete restrict;
alter table "joinn"
   add constraint FK_joinn_joinn2_PROJECT foreign key (projectID)
      references project (projectID)
      on update restrict
      on delete restrict;
alter table management
   add constraint FK_MANAGEME_HEAD2_DEPARTME foreign key (departmentID)
      references department (departmentID)
      on update restrict
      on delete restrict;
alter table management
   add constraint FK_MANAGEME_ISA2_EMPLOYEE foreign key (employeeID)
      references employee (employeeID)
      on update restrict
      on delete restrict;
alter table project
   add constraint FK_PROJECT_PRINCIPAL_EMPLOYEE foreign key (employeeID)
      references employee (employeeID)
      on update restrict
      on delete restrict;
alter table project
   add constraint FK_PROJECT_RESPONSIB_BRANCHOF foreign key (officeID)
      references branchoffice (officeID)
      on update restrict
      on delete restrict;
alter table technicist
   add constraint FK_TECHNICI_ISA_EMPLOYEE foreign key (employeeID)
      references employee (employeeID)
      on update restrict
```

```
        on delete restrict;
```

4.a）两种关系模式差异不大，但是一对一一对多的关系，我设计的表可以给外键约束的变量和被约束的变量起不同的变量名，更贴近实际使用需求，PowerDesigner 只能起相同的变量名

b）powerdesigner 生成的语句注释较多，逻辑清晰，便于使用者阅读和修改。同时，powerdesigner 生成外键约束大多在类外实现，以防在被约束表未定义的时候发生异常。powerdisigner 还会为表创建索引，加快查找效率。powerdesigner 会为每个语句说明是否可以为空值。同时，powerdesigner 在创建表之前会将可能存在的已有表删除，保证程序正确执行。