

以拜仁慕尼黑为例的德甲联赛球队的数据库设计

一、领域介绍

建立德甲联赛球队数据库的目的是为了收集和整理德甲联赛球队的相关信息，方便球迷、媒体和研究人员快速获取和分析数据，从而更好地了解球队的情况和趋势。具体来说，德甲联赛球队数据库可以带来以下几个方面的好处：

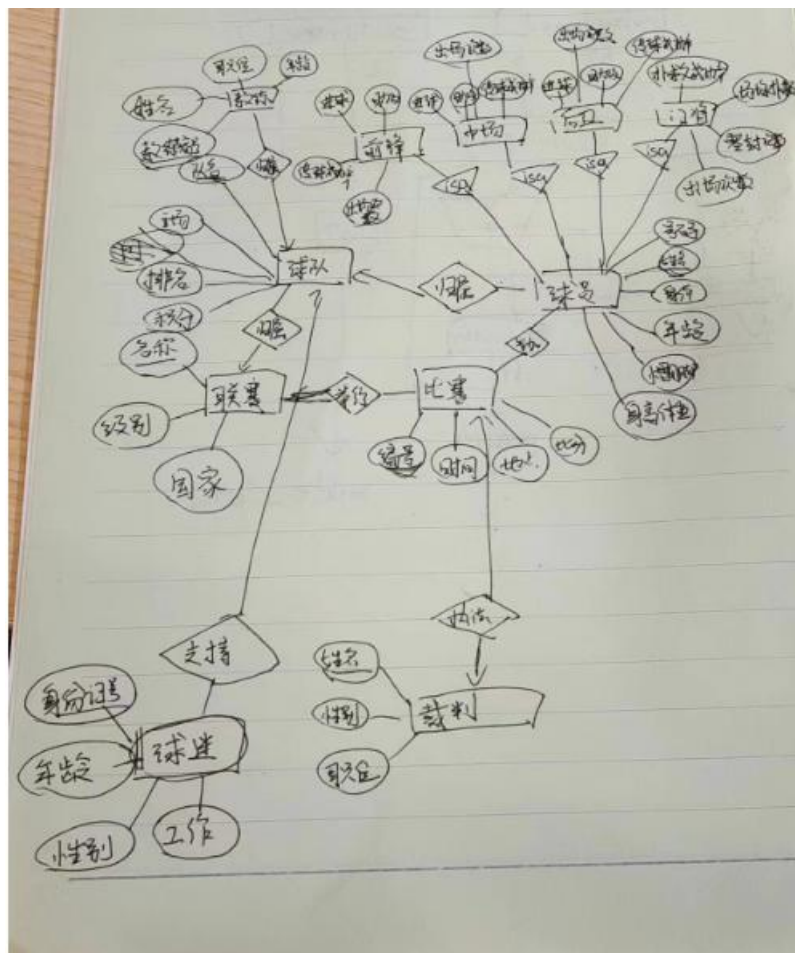
- 1.提供全面的球队信息：德甲联赛球队数据库可以收集和整理球队的基本信息、球员信息、比赛数据、荣誉、财务数据等多方面的数据，让球迷和研究人员可以全面了解球队的情况。
- 2.帮助球迷更好地了解球队情况：通过德甲联赛球队数据库，球迷可以更加全面地了解球队的情况，包括球队的历史、战术、球员、比赛数据等，从而更好地支持自己喜爱的球队。
- 3.提供数据支持：德甲联赛球队数据库可以收集和整理球队的比赛数据、财务数据等，为研究人员提供数据支持，进行数据分析和研究。
- 4.提高媒体报道效率：德甲联赛球队数据库可以为媒体提供全面的球队信息，提高媒体报道的效率和质量。
- 5.促进足球产业发展：德甲联赛球队数据库可以为足球产业提供数据支持，促进足球产业的发展和壮大。

可能的需求有

1. 球队基本信息：包括球队名称、所在城市、球场名称、球队成立时间、主教练、球队价值等。
2. 球队球员信息：包括球员姓名、年龄、国籍、身高、体重、位置、合同期限、转会费用等。
- 3.球队比赛数据：包括球队赛季赛程、比赛时间、对手、比分、进球数、失球数、胜率、积分等。
- 4.球队荣誉：包括球队曾经获得的德甲联赛冠军、德国杯冠军、欧洲冠军联赛冠军、欧洲联赛冠军等。
- 5.球队财务数据：包括球队的收入、支出、薪资总额、转会费用、赞助商和广告收入等。

二、主观创建

1. ER 图



2. 关系模式（主键和外键）

E/R:

球队（队名，联赛名称，主场，排名，积分）

教练（编号，球队队名，姓名，职位）

联赛（名称，级别，国家）

球员（姓名，球队队名，号码，身价，年龄，惯用脚，身高体重）

比赛（编号，时间，地点，比分）

裁判（姓名，性别，职业）

前锋（姓名， 进球， 助攻， 传球成功率， 出场次数）

中场（姓名， 进球， 助攻， 传球成功率， 出场次数）

后卫（姓名， 进球， 助攻， 传球成功率， 出场次数）

门将（姓名， 扑救成功率， 场均扑救， 出场次数， 零封次数）

3. SQL 创建

```
create table 联赛
(
    名称    CHAR (100) ,
    级别    int    ,
    国家    char(100)  ,
    primary key (名称)
);
```

```
create table 球队
(
    队名 char(100),
    归属 varchar(100) REFERENCES 联赛（归属）,
    荣誉 Char (500) ,
    排名 int,
    财务数据 varchar(100),
    积分 int,
    primary key (队名)
);
```

```
create table 教练
(
    姓名    int,
    编号    int,
    年龄    int,
    职位    int,
    归属 varchar(100) REFERENCES 球队（归属）,
    Primary key(编号)
);
```

```
create table 球员
(
    姓名    char(100),
    归属    varchar(100) REFERENCES 球队（归属）,
    号码    int,
```

```
    身价 int,  
    年龄 int,  
    惯用脚 int,  
    身高体重 int,  
    primary key (姓名)  
);
```

```
create table 比赛  
(  
    编号 int,  
    时间 time,  
    地点 char(100),  
    比分 varchar(100),  
    执法 char(100), REFERENCES 裁判 (姓名)  
    primary key (编号)  
);
```

```
create table 球迷  
(  
    身份证号 int,  
    归属 varchar(100) REFERENCES 球队 (归属),  
    年龄 int,  
    性别 int,  
    工作 char (100),  
    primary key (身份证号)  
);
```

```
Create table 前锋  
(  
    进球 int,  
    助攻 int,  
    传球成功率 float,  
    出场次数 int,  
    Foreign key(姓名) REFERENCES 球员(姓名),  
)
```

```
Create table 中场  
(  
    进球 int,  
    助攻 int,  
    传球成功率 float,  
    出场次数 int,  
    Foreign key(姓名) REFERENCES 球员(姓名),  
)
```

Create table 后卫

```
(  
    进球 int,  
    助攻 int,  
    传球成功率 float,  
    出场次数 int,  
    Foreign key(姓名) REFERENCES 球员(姓名),  
)
```

Create table 门将

```
(  
    场均扑救 float,  
    扑救成功率 float,  
    零封次数 int,  
    出场次数 int,  
    Foreign key(姓名) REFERENCES 球员(姓名),  
)
```

4. 查询样例

(1) 单表查询

查询联赛“德甲”的级别

```
SELECT '级别' from '联赛' WHERE '名称' = '德甲';
```

(2) 多表连接查询

查询拜仁慕尼黑的所有球员和教练

```
SELECT '姓名' FROM '球员' , '教练' WHERE  
球队.队名 =教练.队名='拜仁慕尼黑';
```

(3) 多表嵌套查询

查询德甲所有球队的积分

```
SELECT '积分' FROM '球队' WHERE '名称'  
IN (SELECT '名称' FROM '联赛' WHERE " = '德甲');
```

(4) EXISIT 查询

查询在各个球队中，年龄唯一的球员姓名

```
SELECT '姓名' FROM '球员' e1 WHERE not EXISITS  
(SELECT * FROM '球员' e2 WHERE e2.'队名' = e1.'队名' AND  
e2.'年龄' = e1.'年龄');
```

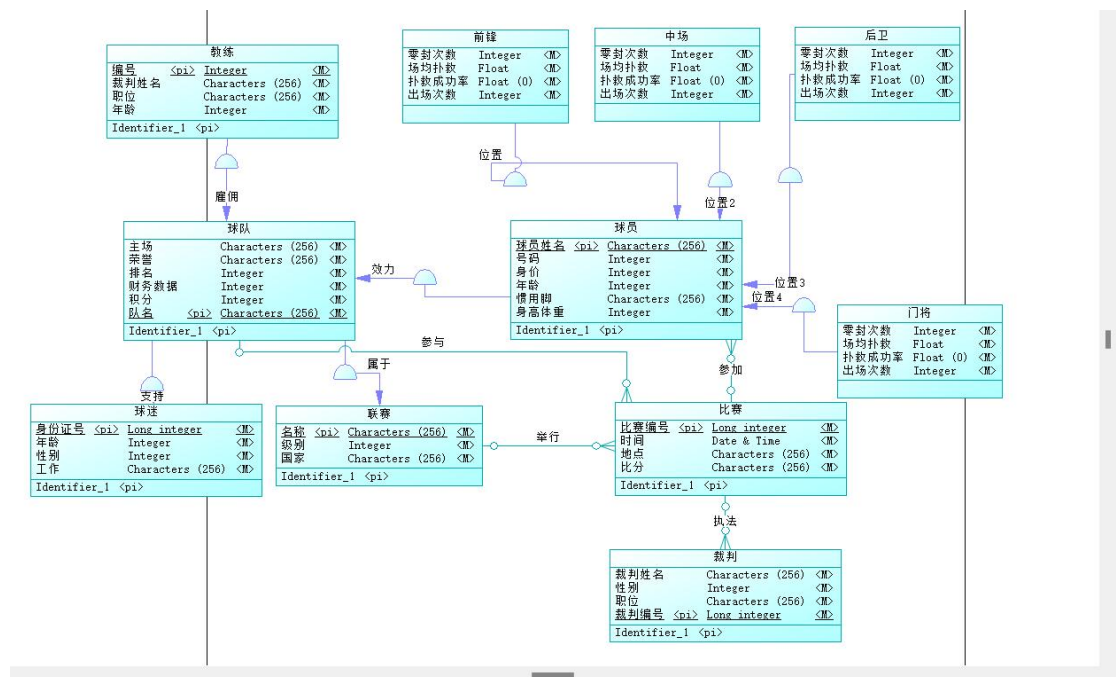
(5) 聚合操作查询

查询在各个球队的球员数量

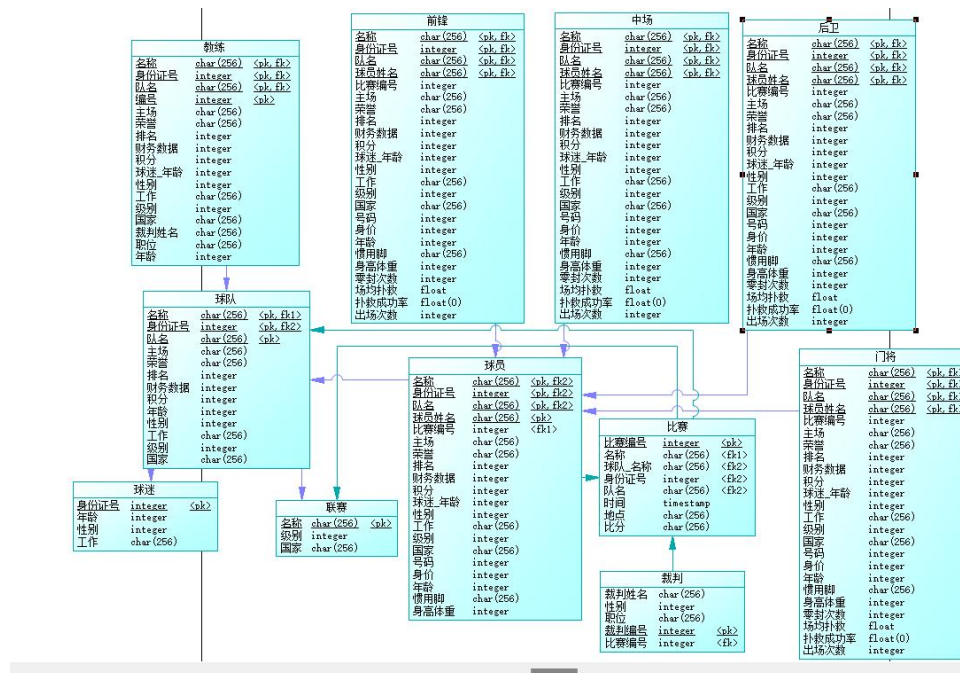
```
SELECT '球队', COUNT (*) FROM '球员' GROUP BY '球队';
```

三、Power Designer 工具实现

1. ER 图



2. 关系模式图



3. 创建 SQL 语句

```

/*=====*/
/* DBMS name:      Sybase SQL Anywhere 12                      */
/* Created on:      2023/4/9 20:32:54                          */
/*=====*/

```

```

if exists(select 1 from sys.sysforeignkey where role='FK_中场_位置 2_球员')
then

```

```

    alter table 中场
        delete foreign key FK_中场_位置 2_球员
end if;

```

```

if exists(select 1 from sys.sysforeignkey where role='FK_前锋_位置_球员') then

```

```

    alter table 前锋
        delete foreign key FK_前锋_位置_球员
end if;

```

```

if exists(select 1 from sys.sysforeignkey where role='FK_后卫_位置 3_球员')
then

```

```

    alter table 后卫
        delete foreign key FK_后卫_位置 3_球员
end if;

```

```

if exists(select 1 from sys.sysforeignkey where role='FK_教练_雇佣_球队') then

```

```

    alter table 教练

```

```

        delete foreign key FK_教练_雇佣_球队
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_比赛_举行_联赛') then
    alter table 比赛
        delete foreign key FK_比赛_举行_联赛
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_比赛_参与_球队') then
    alter table 比赛
        delete foreign key FK_比赛_参与_球队
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_球员_参加_比赛') then
    alter table 球员
        delete foreign key FK_球员_参加_比赛
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_球员_效力_球队') then
    alter table 球员
        delete foreign key FK_球员_效力_球队
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_球队_属于_联赛') then
    alter table 球队
        delete foreign key FK_球队_属于_联赛
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_球队_支持_球迷') then
    alter table 球队
        delete foreign key FK_球队_支持_球迷
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_裁判_执法_比赛') then
    alter table 裁判
        delete foreign key FK_裁判_执法_比赛
end if;

if exists(select 1 from sys.sysforeignkey where role='FK_门将_位置4_球员')
then
    alter table 门将
        delete foreign key FK_门将_位置4_球员
end if;

```


drop index if exists 中场.中场_PK;

drop table if exists 中场;

drop index if exists 前锋.前锋_PK;

drop table if exists 前锋;

drop index if exists 后卫.后卫_PK;

drop table if exists 后卫;

drop index if exists 教练.雇佣_FK;

drop index if exists 教练.教练_PK;

drop table if exists 教练;

drop index if exists 比赛.参与_FK;

drop index if exists 比赛.举行_FK;

drop index if exists 比赛.比赛_PK;

drop table if exists 比赛;

drop index if exists 球员.效力_FK;

drop index if exists 球员.参加_FK;

drop index if exists 球员.球员_PK;

drop table if exists 球员;

drop index if exists 球迷.球迷_PK;

drop table if exists 球迷;

drop index if exists 球队.支持_FK;

drop index if exists 球队.属于_FK;

drop index if exists 球队.球队_PK;

```
drop table if exists 球队;
```

```
drop index if exists 联赛.联赛_PK;
```

```
drop table if exists 联赛;
```

```
drop index if exists 裁判.执法_FK;
```

```
drop index if exists 裁判.裁判_PK;
```

```
drop table if exists 裁判;
```

```
drop index if exists 门将.门将_PK;
```

```
drop table if exists 门将;
```

```
/*=====*/
/* Table: 中场 */
/*=====*/
create table 中场
(
    名称                char(256)                not null,
    身份证号            integer                    not null,
    队名                char(256)                not null,
    球员姓名            char(256)                not null,
    1                    integer                    null,
    主场                char(256)                not null,
    荣誉                char(256)                not null,
    排名                integer                    not null,
    财务数据            integer                    not null,
    积分                integer                    not null,
    球迷_年龄            integer                    not null,
    性别                integer                    not null,
    工作                char(256)                not null,
    级别                integer                    not null,
    国家                char(256)                not null,
    号码                integer                    not null,
    身价                integer                    not null,
    年龄                integer                    not null,
    惯用脚              char(256)                not null,
    身高体重            integer                    not null,
    零封次数            integer                    not null,
    场均扑救            float                    not null,
    扑救成功率          float(0)                 not null,
```

```

        出场次数                integer                not null,
        constraint PK_中场 primary key clustered (名称, 身份证号, 队名, 球员姓名)
);

```

```

/*=====*/
/* Index: 中场_PK                                */
/*=====*/

```

```

create unique clustered index 中场_PK on 中场 (
名称 ASC,
身份证号 ASC,
队名 ASC,
球员姓名 ASC
);

```

```

/*=====*/
/* Table: 前锋                                    */
/*=====*/

```

```

create table 前锋
(
    名称                char(256)                not null,
    身份证号            integer                not null,
    队名                char(256)                not null,
    球员姓名            char(256)                not null,
    1                    integer                null,
    主场                char(256)                not null,
    荣誉                char(256)                not null,
    排名                integer                not null,
    财务数据            integer                not null,
    积分                integer                not null,
    球迷_年龄            integer                not null,
    性别                integer                not null,
    工作                char(256)                not null,
    级别                integer                not null,
    国家                char(256)                not null,
    号码                integer                not null,
    身价                integer                not null,
    年龄                integer                not null,
    惯用脚                char(256)                not null,
    身高体重            integer                not null,
    零封次数            integer                not null,
    场均扑救            float                not null,
    扑救成功率            float(0)                not null,
    出场次数            integer                not null,
    constraint PK_前锋 primary key clustered (名称, 身份证号, 队名, 球员姓名)
);

```

```
);
```

```
/*=====*/
/* Index: 前锋_PK */
/*=====*/
create unique clustered index 前锋_PK on 前锋 (
名称 ASC,
身份证号 ASC,
队名 ASC,
球员姓名 ASC
);
```

```
/*=====*/
/* Table: 后卫 */
/*=====*/
create table 后卫
(
名称 char(256) not null,
身份证号 integer not null,
队名 char(256) not null,
球员姓名 char(256) not null,
1 integer null,
主场 char(256) not null,
荣誉 char(256) not null,
排名 integer not null,
财务数据 integer not null,
积分 integer not null,
球迷_年龄 integer not null,
性别 integer not null,
工作 char(256) not null,
级别 integer not null,
国家 char(256) not null,
号码 integer not null,
身价 integer not null,
年龄 integer not null,
惯用脚 char(256) not null,
身高体重 integer not null,
零封次数 integer not null,
场均扑救 float not null,
扑救成功率 float(0) not null,
出场次数 integer not null,
constraint PK_后卫 primary key clustered (名称, 身份证号, 队名, 球员姓名)
);
```

```

/*=====*/
/* Index: 后卫_PK */
/*=====*/
create unique clustered index 后卫_PK on 后卫 (
名称 ASC,
身份证号 ASC,
队名 ASC,
球员姓名 ASC
);

/*=====*/
/* Table: 教练 */
/*=====*/
create table 教练
(
    名称                char(256)                not null,
    身份证号            integer                    not null,
    队名                char(256)                not null,
    编号                integer                    not null,
    主场                char(256)                not null,
    荣誉                char(256)                not null,
    排名                integer                    not null,
    财务数据            integer                    not null,
    积分                integer                    not null,
    球迷_年龄            integer                    not null,
    性别                integer                    not null,
    工作                char(256)                not null,
    级别                integer                    not null,
    国家                char(256)                not null,
    裁判姓名            char(256)                not null,
    职位                char(256)                not null,
    年龄                integer                    not null,
    constraint PK_教练 primary key (名称, 身份证号, 队名, 编号)
);

/*=====*/
/* Index: 教练_PK */
/*=====*/
create unique index 教练_PK on 教练 (
名称 ASC,
身份证号 ASC,
队名 ASC,
编号 ASC
);

```

```

/*=====*/
/* Index: 雇佣_FK */
/*=====*/
create index 雇佣_FK on 教练 (
名称 ASC,
身份证号 ASC,
队名 ASC
);

/*=====*/
/* Table: 比赛 */
/*=====*/
create table 比赛
(
1
integer not null,
名称 char(256) null,
球队_名称 char(256) null,
身份证号 integer null,
队名 char(256) null,
时间 timestamp not null,
地点 char(256) not null,
比分 char(256) not null,
constraint PK_比赛 primary key (1)
);

/*=====*/
/* Index: 比赛_PK */
/*=====*/
create unique index 比赛_PK on 比赛 (
1 ASC
);

/*=====*/
/* Index: 举行_FK */
/*=====*/
create index 举行_FK on 比赛 (
名称 ASC
);

/*=====*/
/* Index: 参与_FK */
/*=====*/
create index 参与_FK on 比赛 (

```

球队_名称 ASC,
身份证号 ASC,
队名 ASC
);

```
/*=====*/  
/* Table: 球员 */  
/*=====*/
```

```
create table 球员  
(  
    名称                char(256)                not null,  
    身份证号            integer                    not null,  
    队名                char(256)                not null,  
    球员姓名            char(256)                not null,  
    1                    integer                    null,  
    主场                char(256)                not null,  
    荣誉                char(256)                not null,  
    排名                integer                    not null,  
    财务数据            integer                    not null,  
    积分                integer                    not null,  
    球迷_年龄            integer                    not null,  
    性别                integer                    not null,  
    工作                char(256)                not null,  
    级别                integer                    not null,  
    国家                char(256)                not null,  
    号码                integer                    not null,  
    身价                integer                    not null,  
    年龄                integer                    not null,  
    惯用脚              char(256)                not null,  
    身高体重            integer                    not null,  
    constraint PK_球员 primary key (名称, 身份证号, 队名, 球员姓名)  
);
```

```
/*=====*/  
/* Index: 球员_PK */  
/*=====*/
```

```
create unique index 球员_PK on 球员 (  
    名称 ASC,  
    身份证号 ASC,  
    队名 ASC,  
    球员姓名 ASC  
);
```

```
/*=====*/
```

```

/* Index: 参加_FK */
/*=====*/
create index 参加_FK on 球员 (
1 ASC
);

```

```

/*=====*/
/* Index: 效力_FK */
/*=====*/
create index 效力_FK on 球员 (
名称 ASC,
身份证号 ASC,
队名 ASC
);

```

```

/*=====*/
/* Table: 球迷 */
/*=====*/
create table 球迷
(
    身份证号            integer            not null,
    年龄                integer            not null,
    性别                integer            not null,
    工作                char(256)          not null,
    constraint PK_球迷 primary key (身份证号)
);

```

```

/*=====*/
/* Index: 球迷_PK */
/*=====*/
create unique index 球迷_PK on 球迷 (
身份证号 ASC
);

```

```

/*=====*/
/* Table: 球队 */
/*=====*/
create table 球队
(
    名称                char(256)          not null,
    身份证号            integer            not null,
    队名                char(256)          not null,
    主场                char(256)          not null,
    荣誉                char(256)          not null,

```



```

        排名                integer                not null,
        财务数据            integer                not null,
        积分                integer                not null,
        年龄                integer                not null,
        性别                integer                not null,
        工作                char(256)              not null,
        级别                integer                not null,
        国家                char(256)              not null,
        constraint PK_球队 primary key (名称, 身份证号, 队名)
    );

/*=====*/
/* Index: 球队_PK */
/*=====*/
create unique index 球队_PK on 球队 (
    名称 ASC,
    身份证号 ASC,
    队名 ASC
);

/*=====*/
/* Index: 属于_FK */
/*=====*/
create index 属于_FK on 球队 (
    名称 ASC
);

/*=====*/
/* Index: 支持_FK */
/*=====*/
create index 支持_FK on 球队 (
    身份证号 ASC
);

/*=====*/
/* Table: 联赛 */
/*=====*/
create table 联赛
(
    名称                char(256)                not null,
    级别                integer                not null,
    国家                char(256)                not null,
    constraint PK_联赛 primary key (名称)
);

```

```

/*=====*/
/* Index: 联赛_PK */
/*=====*/
create unique index 联赛_PK on 联赛 (
名称 ASC
);

```

```

/*=====*/
/* Table: 裁判 */
/*=====*/
create table 裁判
(
    裁判姓名          char(256)          not null,
    性别              integer             not null,
    职位              char(256)           not null,
    裁判编号          integer             not null,
    1                  integer            null,
    constraint PK_裁判 primary key (裁判编号)
);

```

```

/*=====*/
/* Index: 裁判_PK */
/*=====*/
create unique index 裁判_PK on 裁判 (
裁判编号 ASC
);

```

```

/*=====*/
/* Index: 执法_FK */
/*=====*/
create index 执法_FK on 裁判 (
1 ASC
);

```

```

/*=====*/
/* Table: 门将 */
/*=====*/
create table 门将
(
    名称              char(256)          not null,
    身份证号          integer             not null,
    队名              char(256)           not null,
    球员姓名          char(256)           not null,

```

```

1                integer                null,
主场            char(256)              not null,
荣誉            char(256)              not null,
排名            integer                not null,
财务数据        integer                not null,
积分            integer                not null,
球迷_年龄        integer                not null,
性别            integer                not null,
工作            char(256)              not null,
级别            integer                not null,
国家            char(256)              not null,
号码            integer                not null,
身价            integer                not null,
年龄            integer                not null,
惯用脚          char(256)              not null,
身高体重        integer                not null,
零封次数        integer                not null,
场均扑救        float                 not null,
扑救成功率      float(0)              not null,
出场次数        integer                not null,
constraint PK_门将 primary key clustered (名称, 身份证号, 队名, 球员姓名)
);

```

```

/*=====*/
/* Index: 门将_PK                                     */
/*=====*/
create unique clustered index 门将_PK on 门将 (
名称 ASC,
身份证号 ASC,
队名 ASC,
球员姓名 ASC
);

```

```

alter table 中场
add constraint FK_中场_位置2_球员 foreign key (名称, 身份证号, 队名, 球员姓名)
references 球员 (名称, 身份证号, 队名, 球员姓名)
on update restrict
on delete restrict;

```

```

alter table 前锋
add constraint FK_前锋_位置_球员 foreign key (名称, 身份证号, 队名, 球员姓名)
references 球员 (名称, 身份证号, 队名, 球员姓名)

```

```
on update restrict
on delete restrict;
```

```
alter table 后卫
```

```
add constraint FK_后卫_位置3_球员 foreign key (名称, 身份证号, 队名, 球员
姓名)
references 球员 (名称, 身份证号, 队名, 球员姓名)
on update restrict
on delete restrict;
```

```
alter table 教练
```

```
add constraint FK_教练_雇佣_球队 foreign key (名称, 身份证号, 队名)
references 球队 (名称, 身份证号, 队名)
on update restrict
on delete restrict;
```

```
alter table 比赛
```

```
add constraint FK_比赛_举行_联赛 foreign key (名称)
references 联赛 (名称)
on update restrict
on delete restrict;
```

```
alter table 比赛
```

```
add constraint FK_比赛_参与_球队 foreign key (球队_名称, 身份证号, 队名)
references 球队 (名称, 身份证号, 队名)
on update restrict
on delete restrict;
```

```
alter table 球员
```

```
add constraint FK_球员_参加_比赛 foreign key (1)
references 比赛 (1)
on update restrict
on delete restrict;
```

```
alter table 球员
```

```
add constraint FK_球员_效力_球队 foreign key (名称, 身份证号, 队名)
references 球队 (名称, 身份证号, 队名)
on update restrict
on delete restrict;
```

```
alter table 球队
```

```
add constraint FK_球队_属于_联赛 foreign key (名称)
references 联赛 (名称)
on update restrict
```

```

        on delete restrict;

alter table 球队
    add constraint FK_球队_支持_球迷 foreign key (身份证号)
        references 球迷 (身份证号)
        on update restrict
        on delete restrict;

alter table 裁判
    add constraint FK_裁判_执法_比赛 foreign key (1)
        references 比赛 (1)
        on update restrict
        on delete restrict;

alter table 门将
    add constraint FK_门将_位置4_球员 foreign key (名称, 身份证号, 队名, 球员
姓名)
        references 球员 (名称, 身份证号, 队名, 球员姓名)
        on update restrict
        on delete restrict;

```

四、分析比较采用以上两种方法：

1. 两种关系模式的设计是否存在差异？若有，这种差异是否对后期的实现带来不同的影响？

手写 MySQL 代码和使用 PowerDesigner 设计 E/R 图生成 MySQL 代码之间存在一定的差异。

手写 MySQL 代码需要程序员自己编写 SQL 语句，包括创建表、定义字段、设置约束、插入数据等，需要考虑到语法正确性、性能优化等问题。而使用 PowerDesigner 设计 E/R 图生成 MySQL 代码则是通过图形化界面来设计表结构和关系，系统会自动生成对应的 MySQL 代码，程序员只需要进行一些简单的配置和调整即可。

这种差异对后期的实现可能会带来不同的影响。手写 MySQL 代码需要程序员对 MySQL 语法有一定的了解和掌握，从而能够编写出高效、稳定的代码。相比之下，使用 PowerDesigner 设计 E/R 图生成 MySQL 代码可能会忽略一些细节问题，比如

数据类型、主键、外键等，从而导致实际的数据库表结构和数据之间的关系出现问题。

此外，手写 MySQL 代码可以根据实际需求进行灵活的调整和优化，可以更好地满足具体的业务需求。而使用 PowerDesigner 设计 E/R 图生成 MySQL 代码则可能会受到系统的限制和约束，无法满足一些特殊的需求。

因此，在选择手写 MySQL 代码和使用 PowerDesigner 设计 E/R 图生成 MySQL 代码时，需要根据实际情况进行权衡和选择，确保生成的 MySQL 代码能够满足业务需求，并且具有高效、稳定的特点。

2. Power Designer 工具生成的 SQL 语句有什么样的特点？为什么会出现一些附加语句？它的作用是什么？

Power Designer 生成的 SQL 语句更标准，因此可读性和可移植性更强，语句也比较简洁，只包含必要的信息。

同时它生成的附加语句包括一些注释、创建过程、触发器、外键和索引等。其作用如下：

- 1、 创建索引可以在数据库操作时提高查询速度
- 2、 使用触发器可以在数据操作时保证数据完整性。
- 3、 生成注释则使得语句的可读性更强。