# Week7-ER模型\_关系模型\_SQL操作 Assignment

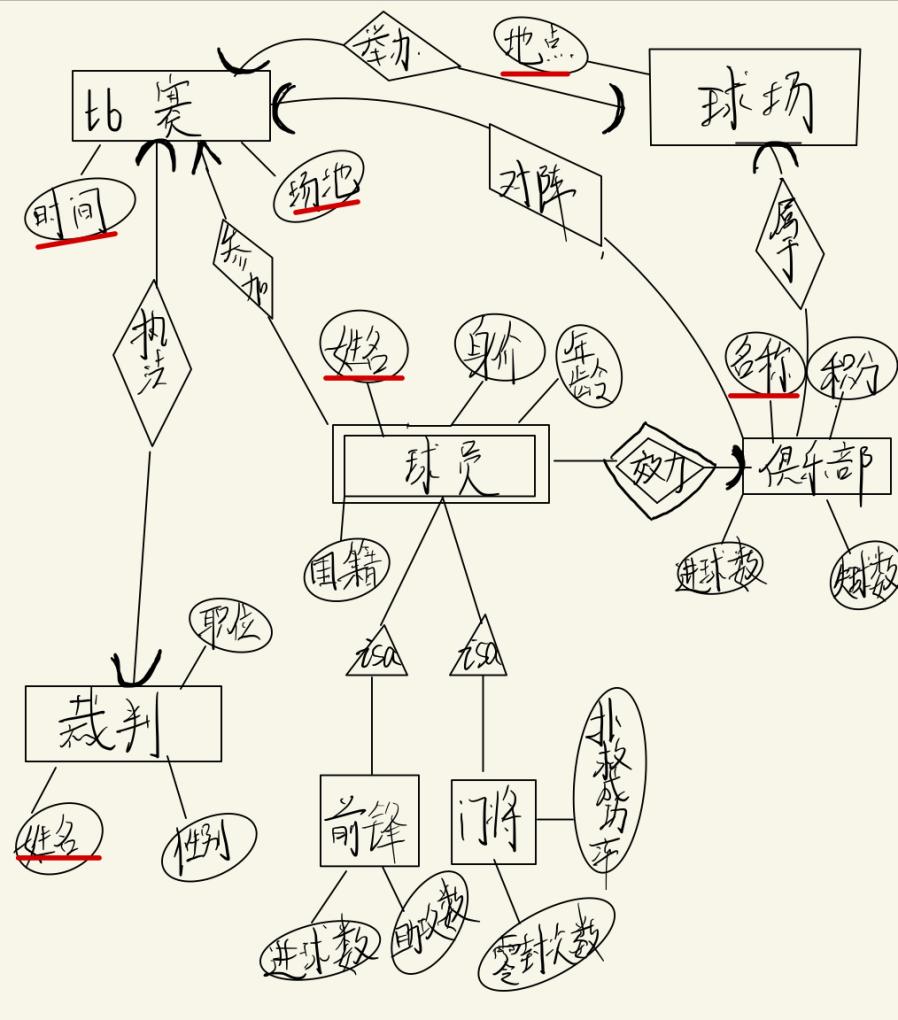
学号：2111408 姓名：周钰宸

1. 选择应用领域
2. 应用领域：英格兰超级联赛（简称英超）的信息管理系统。
3. 该应用领域的详细需求描述：英超是“[欧洲足球五大联赛](https://baike.baidu.com/item/%E6%AC%A7%E6%B4%B2%E8%B6%B3%E7%90%83%E4%BA%94%E5%A4%A7%E8%81%94%E8%B5%9B/9856716?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E8%8B%B1%E6%A0%BC%E5%85%B0%E8%B6%B3%E7%90%83%E8%B6%85%E7%BA%A7%E8%81%94%E8%B5%9B/_blank)”之一，由**20支球队**组成，由超级联盟负责具体运作。英超联赛一般由九月到五月，每队**踢38场比赛（19个主场和19个客场）。在比赛中，每支球队获得3分，如果没有双方取得胜利，就各获1分。**

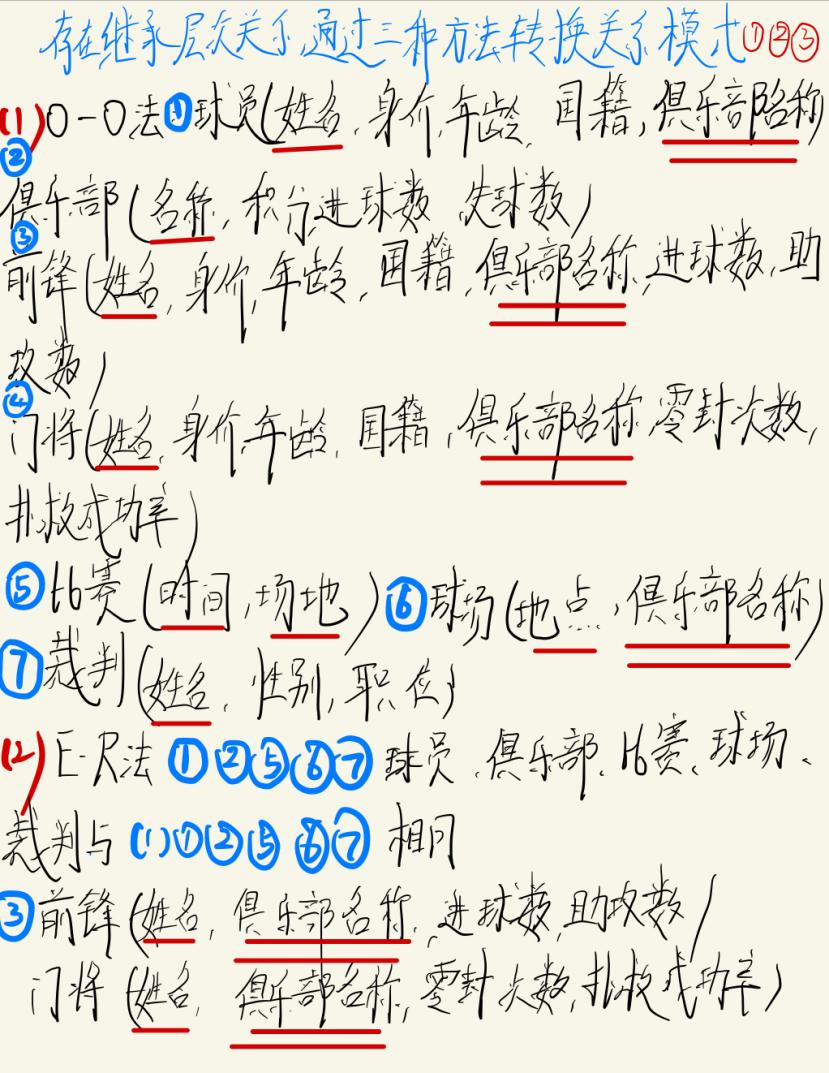
**赛季结束后积分榜末三位**降入[英格兰足球冠军联赛](https://baike.baidu.com/item/%E8%8B%B1%E6%A0%BC%E5%85%B0%E8%B6%B3%E7%90%83%E5%86%A0%E5%86%9B%E8%81%94%E8%B5%9B/7097321?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E8%8B%B1%E6%A0%BC%E5%85%B0%E8%B6%B3%E7%90%83%E8%B6%85%E7%BA%A7%E8%81%94%E8%B5%9B/_blank)，**前四位获得下赛季欧冠参加资格，前七位获得下赛季欧联参加资格**。同时每年需要根据**球队积分第一名颁发该赛季的英超冠军**。

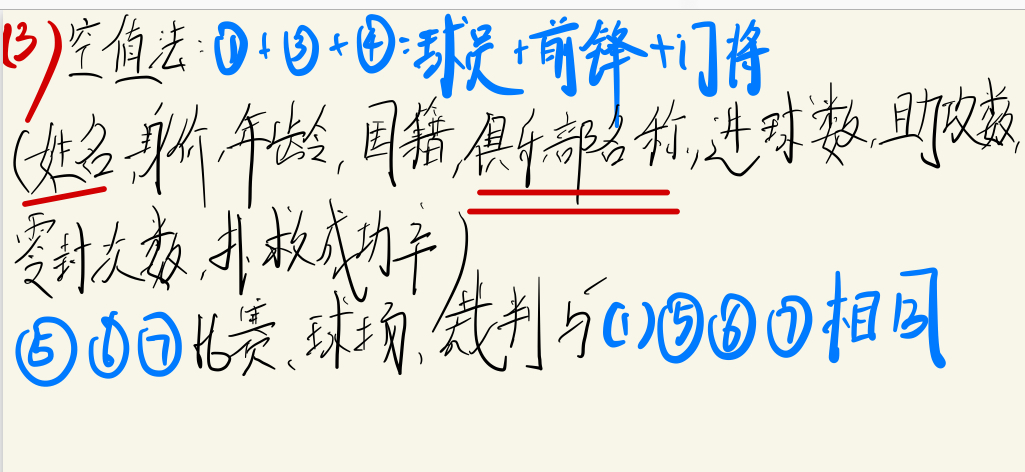
根据**球员进球数颁布该赛季的英超金靴**，根据**守门员的丢球数颁布该赛季的英超金手套**。

1. 设计模型
2. **画出该领域的概念模型ER图（至少有五个以上的实体，含有子类的形式，注意一定标明每个实体的主键）：**



1. **请按课堂上讲授的ER图转换成关系模式的方法，将上述ER图转换成关系模式，并标明每个关系的主键属性和外键属性：(红色单下划线表示主键，红色双下划线表示外键）**





1. **用SQL语句创建上述关系模式：**

(1)create table 球员

(

姓名 CHAR (256),

身价 longint,

年龄 int,

国籍 char(256),

primary key (姓名)

);

(2)create table 俱乐部

(

`名称` CHAR(256),

`积分` int,

`进球数` int，

丢球数 int

primary key (名称)

);

(3)create table 球场

(

`地点` CHAR(256),

);

(4)create table 比赛

(

`时间` char(256),

`场地` char(100),

primary key (时间，场地)

);

(5)create table 裁判

(

`裁判姓名` char(100),

`裁判性别` char(100),

`职位` char(100),

primary key (裁判姓名)

);

(6)create table 前锋

(

`姓名` char(100),REFERENCES `球员`（姓名）

`进球数`int,

`助攻数`int,

primary key (姓名)

);

(7)create table 门将

(

`姓名` char(100),REFERENCES `球员`（姓名）

`零封次数`int,

`扑救成功率`float,

primary key (姓名)

);

1. **给出该数据库模式上5个查询语句样例，分别为：单表查询、多表连接查询、多表嵌套查询和、EXISTS查询和聚合操作查询。**
2. **单表查询**

**查询球员“Mohamed Salah”的国籍**

SELECT ‘国籍’ from ‘球员’ WHERE ‘姓名’ = ‘Mohamed Salah’;

1. **多表连接查询**

**查询所有本赛季有进球的球员姓名**

SELECT ‘姓名’ FROM ‘球员’ ，’前锋’ WHERE

球员.姓名 = 前锋.姓名 ORDER BY 姓名 AND ‘进球数’>1;

1. **多表嵌套查询**

**查询地点在伦敦的球场举办的比赛时间**

SELECT ‘比赛时间’ FROM ‘比赛’ WHERE ‘场地’

IN (SELECT ‘地点’ FROM ‘球场’ WHERE ‘地点’ = ‘伦敦’);

1. **EXIST查询**

**查询在各个国家的球员中，在同一国家本赛季进球数唯一的球员的姓名**

SELECT ‘姓名’ FROM ‘球员’ p1 WHERE not EXISITS

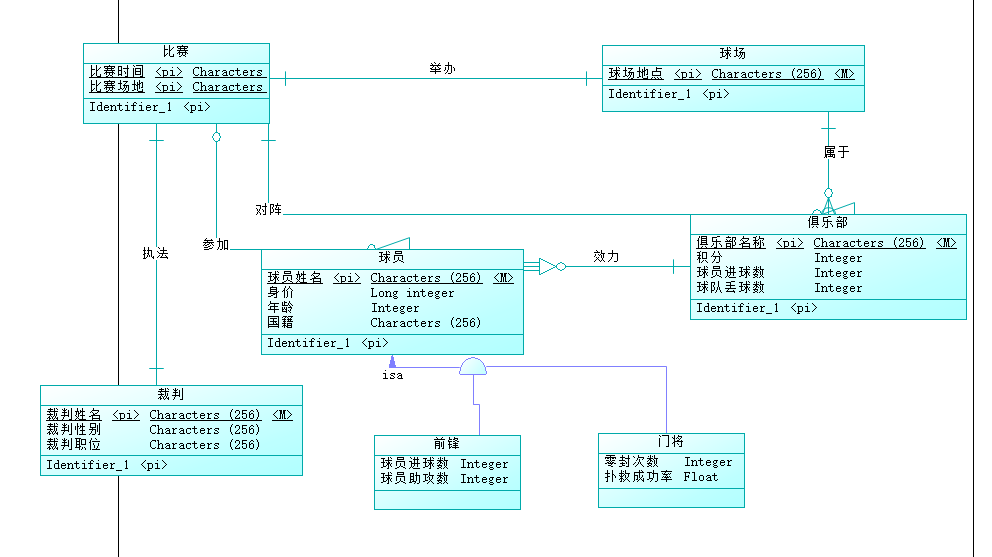
(SELECT \* FROM ‘球员’ p2 WHERE p2.’国家’ = p1.’国家’ AND p2.’进球数’ <> p1.’进球数’);

1. **聚合操作查询**

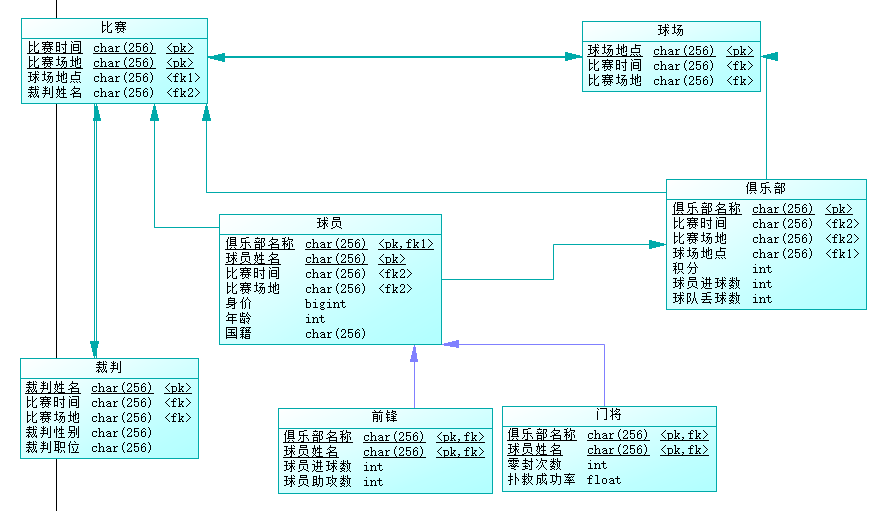
**查询在各个国家的球员数量**

SELECT ‘国籍’， COUNT（\*） FROM ‘球员’ GROUP BY ‘国籍’;

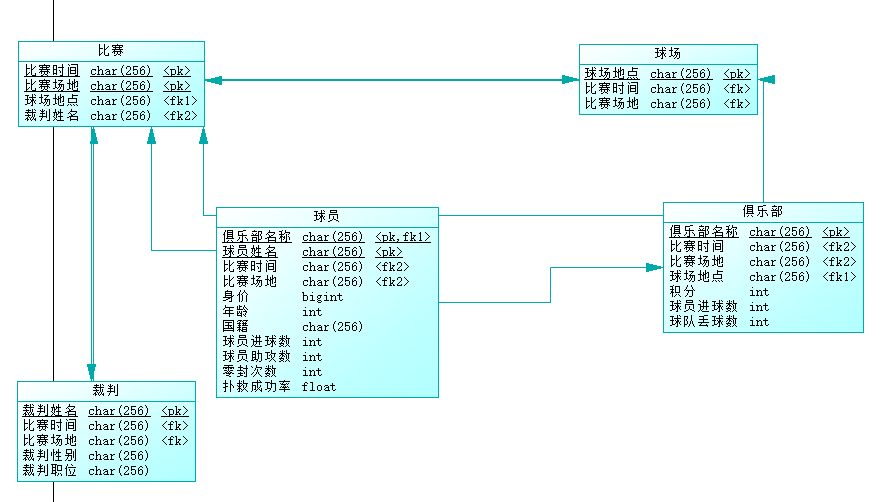
1. 使用PowerDesigner工具实现设计
2. **画出该领域的概念模型ER图，给出ER图截图：**



1. **使用PowerDesigner工具，将上述ER图转为关系模型图，给出关系模型图截图：（PowerDesigner工具不提供O-O的关系模式）**
2. E-R法：



1. 空值法：



1. **使用PowerDesigner工具，生成创建数据库的SQL语句：**

drop table if exists 俱乐部;

drop table if exists 比赛;

drop table if exists 球员;

drop table if exists 球场;

drop table if exists 裁判;

/\*==============================================================\*/

/\* Table: 俱乐部 \*/

/\*==============================================================\*/

create table 俱乐部

(

俱乐部名称 char(256) not null,

比赛时间 char(256) not null,

比赛场地 char(256) not null,

球场地点 char(256) not null,

积分 int,

球员进球数 int,

球队丢球数 int,

primary key (俱乐部名称)

);

/\*==============================================================\*/

/\* Table: 比赛 \*/

/\*==============================================================\*/

create table 比赛

(

比赛时间 char(256) not null,

比赛场地 char(256) not null,

球场地点 char(256) not null,

裁判姓名 char(256) not null,

primary key (比赛时间, 比赛场地)

);

/\*==============================================================\*/

/\* Table: 球员 \*/

/\*==============================================================\*/

create table 球员

(

俱乐部名称 char(256) not null,

球员姓名 char(256) not null,

比赛时间 char(256),

比赛场地 char(256),

身价 bigint,

年龄 int,

国籍 char(256),

球员进球数 int,

球员助攻数 int,

零封次数 int,

扑救成功率 float,

primary key (俱乐部名称, 球员姓名)

);

/\*==============================================================\*/

/\* Table: 球场 \*/

/\*==============================================================\*/

create table 球场

(

球场地点 char(256) not null,

比赛时间 char(256) not null,

比赛场地 char(256) not null,

primary key (球场地点)

);

/\*==============================================================\*/

/\* Table: 裁判 \*/

/\*==============================================================\*/

create table 裁判

(

裁判姓名 char(256) not null,

比赛时间 char(256) not null,

比赛场地 char(256) not null,

裁判性别 char(256),

裁判职位 char(256),

primary key (裁判姓名)

);

alter table 俱乐部 add constraint FK\_对阵 foreign key (比赛时间, 比赛场地)

references 比赛 (比赛时间, 比赛场地);

alter table 俱乐部 add constraint FK\_属于 foreign key (球场地点)

references 球场 (球场地点);

alter table 比赛 add constraint FK\_举办 foreign key (球场地点)

references 球场 (球场地点);

alter table 比赛 add constraint FK\_执法 foreign key (裁判姓名)

references 裁判 (裁判姓名);

alter table 球员 add constraint FK\_参加 foreign key (比赛时间, 比赛场地)

references 比赛 (比赛时间, 比赛场地);

alter table 球员 add constraint FK\_效力 foreign key (俱乐部名称)

references 俱乐部 (俱乐部名称);

alter table 球场 add constraint FK\_举办2 foreign key (比赛时间, 比赛场地)

references 比赛 (比赛时间, 比赛场地);

alter table 裁判 add constraint FK\_执法2 foreign key (比赛时间, 比赛场地)

references 比赛 (比赛时间, 比赛场地);

1. 分析比较采用上述两种方法
2. **两种关系模式的设计是否存在差异？如有差异，这种差异是否对后期的实现带来不同的影响？**

答：

E-R方式和空值方式的关系模式设计**存在差异。**

**差异体现在继承关系产生的子类上：**E-R法产生的子类只继承父类的主键属性，另外再加上自身新增的属性；空值方式将所有父类和子类都放在一个表中，如果对于某些子类其有些属性并不属于它，那么就用空值NULL填充。

**这种差异会对后期实现产生不同的影响。E-R法**可以清晰地描述实体之间的关系，以及这些关系的属性。而**空值法**是一种更加简单的关系模型设计方法，在空值法中，每个关系都被看作一个表格，其中行代表元组（tuple），列代表属性（attribute）。如果某个元组的某个属性值未知或不存在，则该位置填写空值（NULL）。**在后期实现过程中，需要考虑如何处理空值**，以保证查询结果的正确性。

1. **PowerDesigner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？**

答：

1. **特点：**

**①规范性：**PowerDesigner生成的 SQL 语句通常符合各种数据库系统的规范，具有良好的可读性、可维护性和可移植性。

**②可定制性：**支持自定义SQL脚本，用户可以在生成SQL语句时根据需要添加自己的个性化语句，以满足更复杂的需求。

**③自动化：**可以自动生成索引、外键约束等辅助性SQL语句，从而简化了数据库设计人员的工作量；可以自动将一些主键和外键的值设为空值，这样后期在进行表的增删改查时候如果出现违法的操作，会自动报错。

1. **附加语句：附加语句的出现是必要的。**

例如，创建表时可以使用"CREATE TABLE"语句来定义表结构，但是在某些情况下，需要添加一些额外的语句来实现特定的功能，如添加触发器、存储过程、函数等。**附加语句的作用是扩展表的功能，满足不同的业务需求。**