北京国际大学篮球联赛管理系统

张敏濠 | 2020080308 | 土木03

# 项目说明

每年，北京地区大学的国际生都会参加北京国际大学篮球联赛来竞争。因为参加球队有特别多，所以联赛委员会一定有准备一个管理系统。因此，本项目旨在通过利用所学的知识和技能来实现管理系统的基本功能。

这个管理系统的基本目的访问或更新联赛的信息。根据管理系统有各种用户，用户可以划分为两类——**委员会账户**和**球队账户**。想要参加的球队能先在系统新建一个球队账户，并且报名球队相关的细节。然后，委员会能使用之前已固定的委员会账户来检查球队的报名信息，接着批准和增加新球队到联赛数据库。关于游客的访问，管理系统没有提供新建账户的功能，但游客能通过 “**访客访问”**的功能来检查联赛时间表。

# 要求分析

篮球联赛管理系统的潜在使用者，根据不同身份能基本划分为三类：游客，球队，委员会。这三类身份的对应要求不一致，具体分析如下：

## 游客身份

游客的范畴包含所有除了球队和委员会的群体，实际中潜在的用户会集中于联赛观众和粉丝。接着，他们的权限仅限可以查看在一段时间的联赛时间表（一周时期），因为委员会可能对整体时间表会做更新。管理系统会提供“访客访问”的按钮功能，让游客查看时间表。游客也能通过这个功能买比赛的座位票。

## 球队身份

* 为了将游客与球队区分，要求球员首先用账户登录。球员身份继承了游客身份的整体功能。
* 球队能查看本身球员的个人信息，并能做修改或增加。
* 新建账户后，球队能报名参加联赛，填要求的球队整体信息，包括球员个人信息等等。只要委员会批准后才能查看报名成功的状态（管理系统会表示成功）。报名有截止时间；超过截止就不能报名了。
* 球队也能查看个人日程（或时间表）和积分榜。

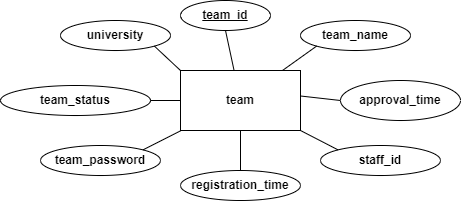
## 委员会身份

* 委员会跟球队一样要首先登录，但之前不用新建账户 （已固定）。
* 当球队报名时，委员会会受到申请表，并且会查看报名球队的信息，通过批准。
* 到报名截止时，委员会能关掉球队的报名功能，会有主任委员有特殊权力（例如：停止申请时期）
* 当报名结束后，委员会能安排联赛配对和时间表。
* 当一场比赛结束后，委员会能插入比赛比分和更新比赛状态变成“已完成”。
* 委员会能看到所门票的信息和球队的积分榜。

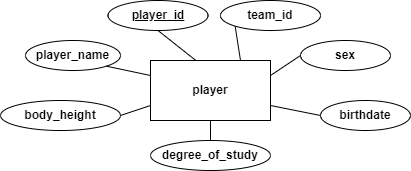
# 概念设计

上述需求分析中出现的重要名词包括：球队，球员，委员会，比赛，座位票等，他们都具有一组属性且部分属性能唯一标识每一个实体，而且他们需要存储到数据库中使用，因此可以建模为实体集。下面给出各实体集、其属性以及总体ER图。

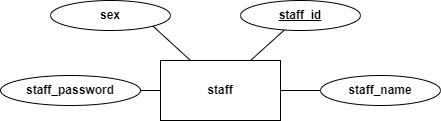
## 球队 （team）



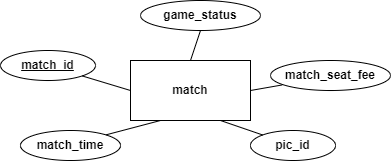
## 球员 （player）



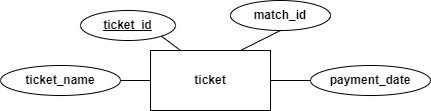
## 委员会 （staff）



## 比赛 （match）



## 座位票 （ticket）



## 联系

a. 球队与球员之间存在1对n的拥有(have)关系；

b. 球队与比赛之间存在m对n的进行(ongoing)关系；这个联系体包括球队的得分，评论, 和输赢状态

e. 委员会与比赛之间存在1对n的管理(handle)关系；

f. 比赛与座位票之间存在1对n的销售(sales)关系 ；

g. 比赛与委员会之间存在1对n的审批(approval)关系；这个联系体会包括审批的时间，评论，和结果

## 

# 逻辑设计

根据 E-R 模型向关系模型转换的规则将上述 E-R 模型转换为如下关系模型。

1）球队表(team)：由team实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **team\_id** | char(10) | 球队身份证号 |
| **staff\_id** | char(10) | 审批委员会身份证号 |
| name | varchar2(40) | 球队名称 |
| university | varchar2(40) | 球队大学名称 |
| team\_status | char(7) | 球队入塞状态 |
| approval\_time | datetime | 球队审批次数 |
| registration\_time | datetime | 球队注册次数 |
| team\_password | varchar2(40) | 球队登录密码 |

2）球员表(player)：由player实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **player\_id** | char(10) | 球员身份证号 |
| team\_id | char(10) | 球队身份证号 |
| sex | char(2) | 球员性别 |
| player\_name | varchar2(40) | 球员全名 |
| body\_height | int | 球员身高 |
| birthdate | date | 球员出生日期 |
| degree\_study | varchar2(40) | 球员学位水平 |

3）委员会表(staff)：由staff实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **staff\_id** | char(10) | 委员会身份证号 |
| staff\_name | varchar2(40) | 委员会全名 |
| sex | char(2) | 委员会性别 |
| staff\_password | varchar2(40) | 委员会登录密码 |

4）比赛表(match)：由match实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **match\_id** | char(10) | 比赛身份证号 |
| pic\_id | char(10) | 委员会身份证号 |
| game\_status | char(7) | 比赛完成状态 |
| match\_time | date | 比赛时间 |
| match\_seat\_fee | float | 比赛票费 |

5）座位票表(ticket)：由ticket实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **ticket\_id** | char(10) | 票身份证号 |
| match\_id | char(10) | 比赛身份证号 |
| holder\_name | varchar2(40) | 顾客姓名 |
| payment\_date | date | 票价支付日期 |

7）进行联系表(ongoing)：由多对多联系集ongoing转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **team\_id** | char(10) | 球队身份证号 |
| **match\_id** | char(10) | 比赛身份证号 |
| total\_points | int | 比赛个队得分 |
| remarks | varchar2(1000) | 球队任何评论 |
| win\_status | boolean | 球队在比赛中输赢状态 |

8）审批联系表(approval)：由一对多联系集approval转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **team\_id** | char(10) | 球队身份证号 |
| **staff\_id** | char(10) | 委员会身份证号 |
| registration\_time | datetime | 审批时间 |
| result | boolean | 审批结果 |

# 关系模式规范化处理

以上所有表的属性都是不能细分的数据项，所以满足1NF； 每个表的所有非主属性都是完整的。全函数依赖于它的主属性，所以满足2NF； 每个表的所有非主属性并不完全依赖于另一个非主属性因此，它满足3NF； 此外，每个表中的任何属性都没有对ID（身份证号）的部分或传递依赖。主属性之间不存在传递依赖，所以也满足BCNF范式。