哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

开课学期：2021 秋季

课程名称：数据库系统

实验名称：足球联赛管理系统设计与实现

实验性质：设计型

实验学时：4 地点：T2102

学生班级：19 级计科 10 班

学生学号：190111026

学生姓名：郭毅安

评阅教师：

报告成绩：

实验与创新实践教育中心制

2021 年 10 月

1 实验环境

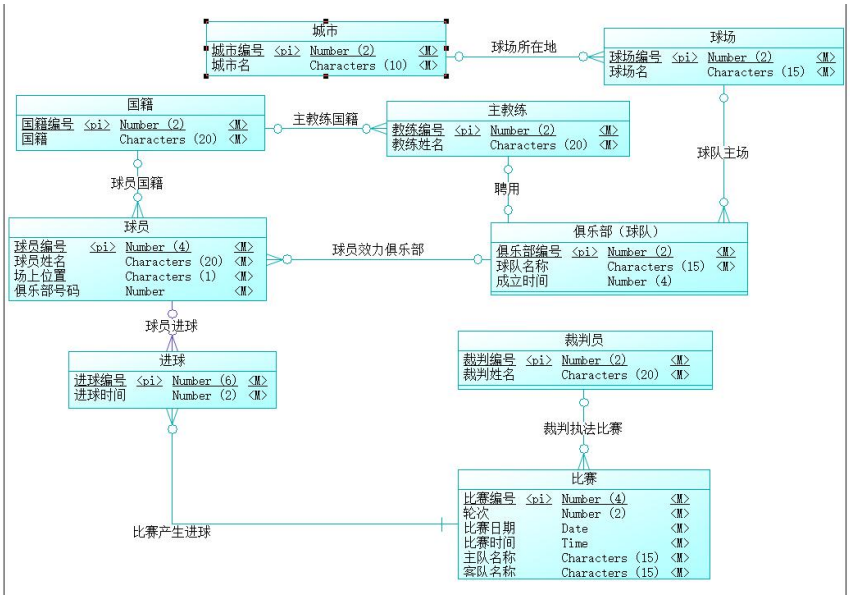
（1）操作系统：windows
（2）开发工具：JAVA 语言+IDEA
（3）技术选型：Mysql 8.0 +Java swing 图形界面

2 实验过程

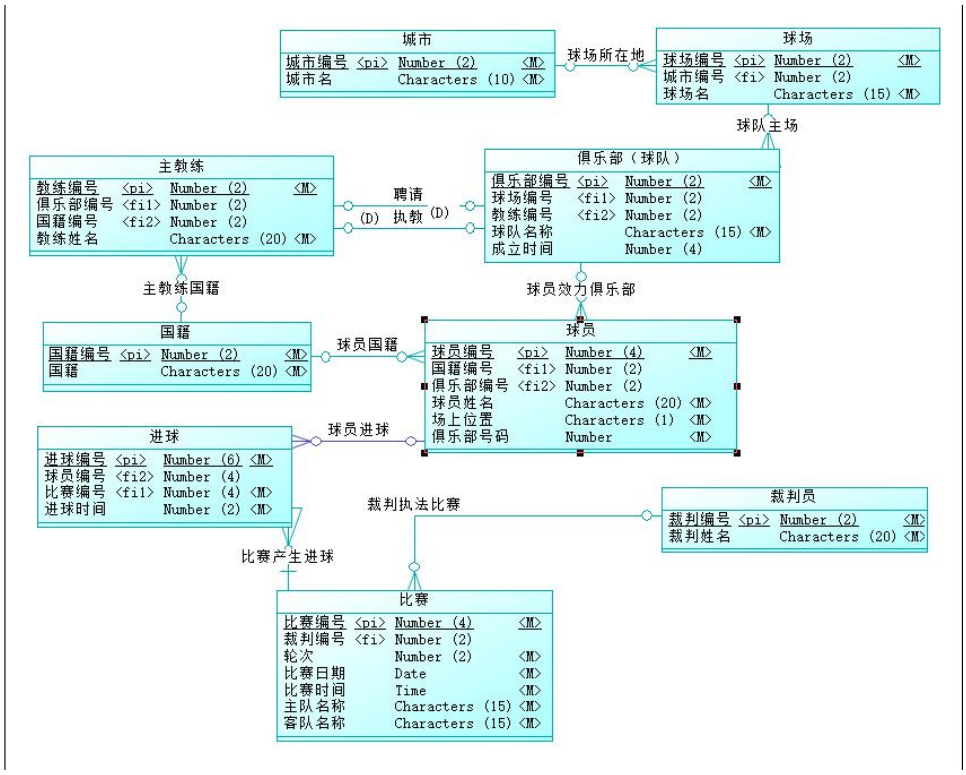
2.1 系统功能

（1） 包含了联赛系统的所有基本组成部分：包括球员、俱乐部、比赛、进球、教练裁判等。
（2） 每个组成部分都有一个对应实体，使得整个数据库系统包含了一个联赛的所有基本信息
（3） 该数据库系统能够对联赛最基本的实体——球员进行增删查改工作
（4） 作为联赛系统的维护和运营最重要的工作之一，该数据库系统能够录入最新比赛的进球。
（5） *该系统能自动生成联赛的最新积分榜和射手榜
（6） 该系统能为一般用户提供球队信息查询、赛程查询和赛况查询等基本服务。

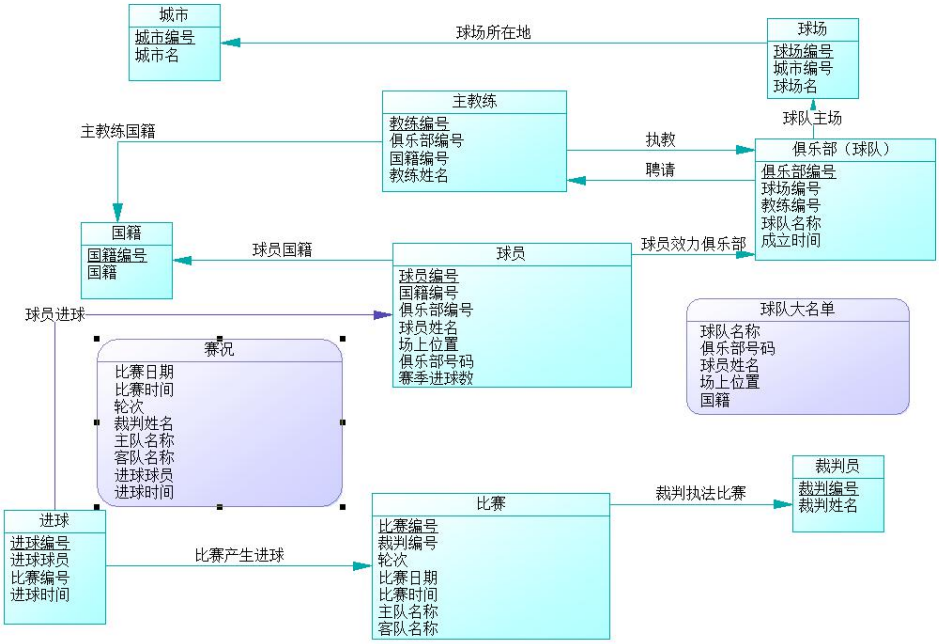
2.1.1 ER 图



2.1.2 LDM 图


城市
城市编号 <pi> Number (2) <M>
城市名 Characters (10) <M>
球场所在地
球场
球场编号 <pi> Number (2) <M>
城市编号 <fi> Number (2)
球场名 Characters (15) <M>
球队主场
主教练
教练编号 <pi> Number (2) <M>
俱乐部编号 <fi1> Number (2)
国籍编号 <fi2> Number (2)
教练姓名 Characters (20) <M>
聘请
(D) 执教 (D)
俱乐部（球队）
俱乐部编号 <pi> Number (2) <M>
球场编号 <fi1> Number (2)
教练编号 <fi2> Number (2)
球队名称 Characters (15) <M>
成立时间 Number (4)
主教练国籍
球员效力俱乐部
国籍
国籍编号 <pi> Number (2) <M>
国籍 Characters (20) <M>
球员国籍
球员
球员编号 <pi> Number (4) <M>
国籍编号 <fi1> Number (2)
俱乐部编号 <fi2> Number (2)
球员姓名 Characters (20) <M>
场上位置 Characters (1) <M>
俱乐部号码 Number <M>
进球
进球编号 <pi> Number (6) <M>
球员编号 <fi2> Number (4)
比赛编号 <fi1> Number (4) <M>
进球时间 Number (2) <M>
球员进球
裁判执法比赛
裁判员
裁判编号 <pi> Number (2) <M>
裁判姓名 Characters (20) <M>
比赛产生进球
比赛
比赛编号 <pi> Number (4) <M>
裁判编号 <fi> Number (2)
轮次 Number (2) <M>
比赛日期 Date <M>
比赛时间 Time <M>
主队名称 Characters (15) <M>
客队名称 Characters (15) <M>

2.1.3 PDM 图



2.1.4 数据库表结构

1、 表结构

Result Grid | Filter Rows: | Edit: | Export/Import:

goal_id	player_id	game_id	score_min
10145	226	101	45
10150	418	101	50
10155	210	101	55
10172	418	101	72
10174	227	101	74
10184	209	101	84
10214	121	102	14
10226	320	102	26
20108	127	201	8
20174	130	201	74
20204	418	202	4

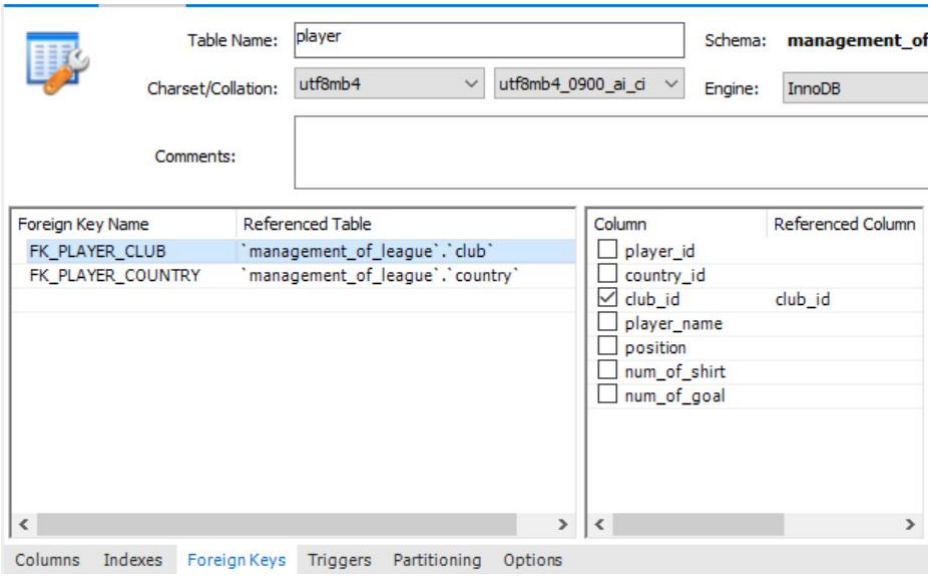


Table Name: player
Schema: management_of
Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci
Engine: InnoDB
Comments:
Foreign Key Name
Referenced Table
FK_PLAYER_CLUB
`management_of_league`.`club`
FK_PLAYER_COUNTRY
`management_of_league`.`country`
Column
Referenced Column
player_id
country_id
club_id
club_id
player_name
position
num_of_shirt
num_of_goal
Columns
Indexes
Foreign Keys
Triggers
Partitioning
Options

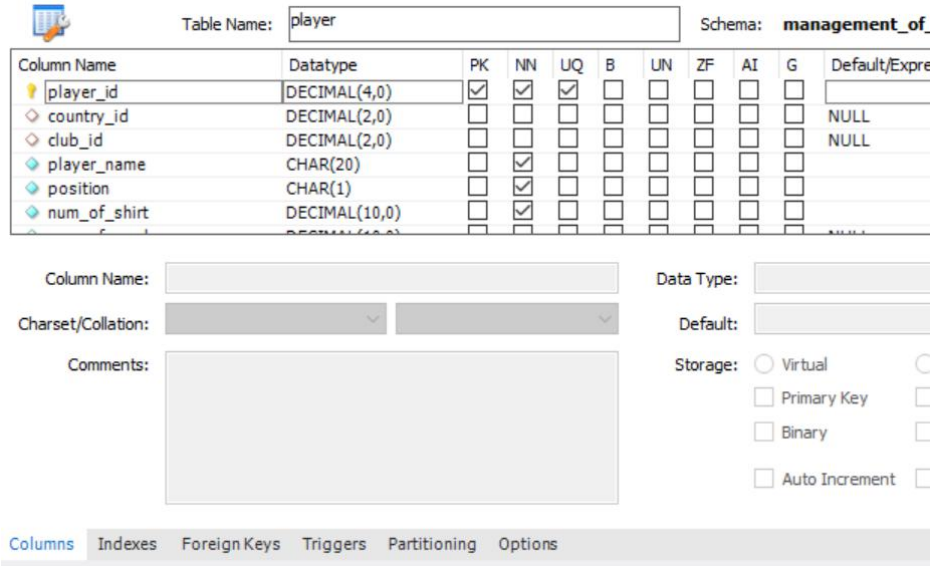
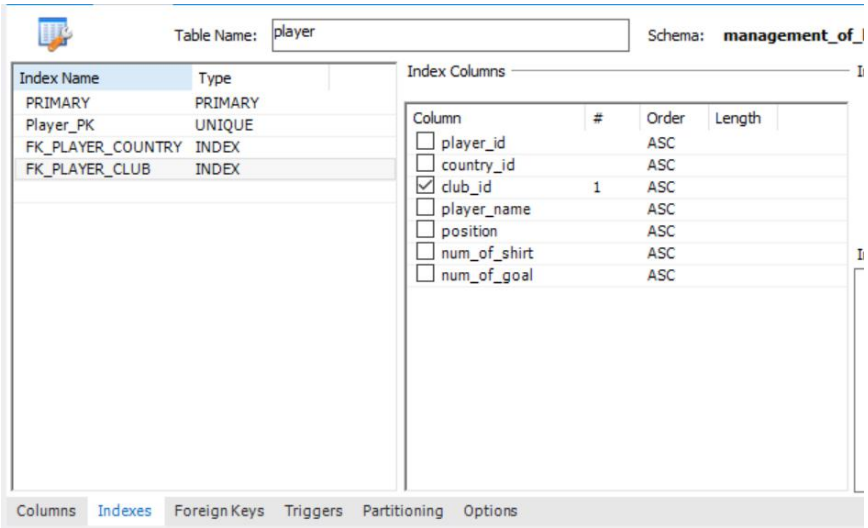


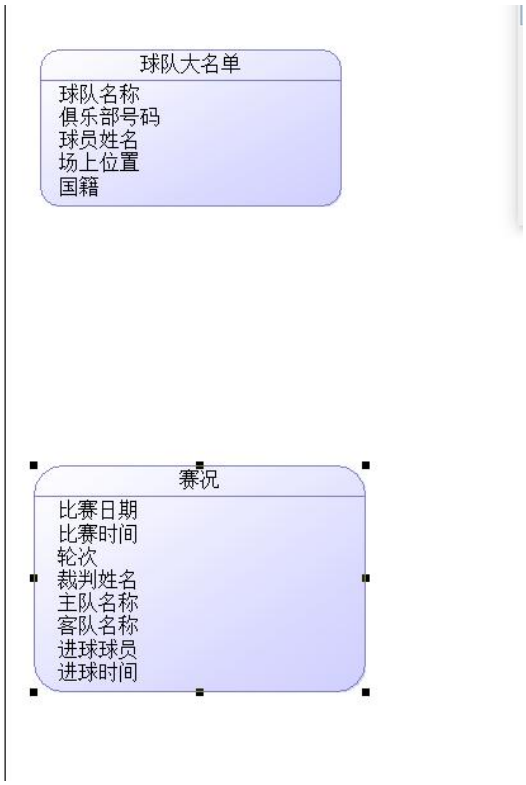
Table Name: player
Schema: management_of_
Column Name
Datatype
PK
NN
UQ
B
UN
ZF
AI
G
Default/Expre
player_id
DECIMAL(4,0)
country_id
DECIMAL(2,0)
NULL
club_id
DECIMAL(2,0)
NULL
player_name
CHAR(20)
position
CHAR(1)
num_of_shirt
DECIMAL(10,0)
Column Name:
Data Type:
Charset/Collation:
Default:
Comments:
Storage:
Virtual
Primary Key
Binary
Auto Increment
Columns
Indexes
Foreign Keys
Triggers
Partitioning
Options

2、 索引



由于主外键约束，很多实体在别的实体中只能以编号形式存在，索引可以部分解决这一问题。

3、 视图



球队大名单：

```
select
    Club.club_name,
    Player.num_of_shirt,
    Player.player_name,
```

```
    Player.position,
    Country.country_name
from
    Player,
    Club,
    Country
where
    Player.club_id = Club.club_id
    and Player.country_id = Country.country_id
group by
    Club.club_name
order by
    Player.num_of_shirt DESC
```

赛况：

```
select
     Game.day_of_match,
     Game.time_of_match,
     Game.round,
     Referee.referee_name,
     hostname, guestname,
     goal. player_id, score_min
from
     Game, Referee, goal
where
     Referee.referee_id = Game.referee_id
     and goal.game_id = game.game_id
```

这两个视图主要用来直观地了解球队的球员信息以及比赛的进球情况等，而这些信息由于主外键的限制无法在表中直观展现。

4、 触发器

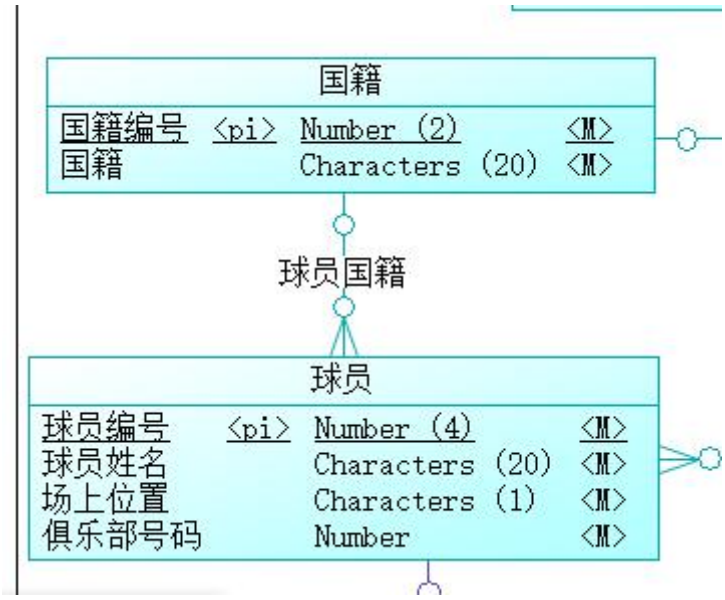
Player 表中有一属性为进球数，可以通过在 goal 表中加一个 after insert 的触发器，Goal 表每次增加一个元组，即进球，则进球者的进球数可实现自动加 1，大大方便了数据库的管理。

实现：

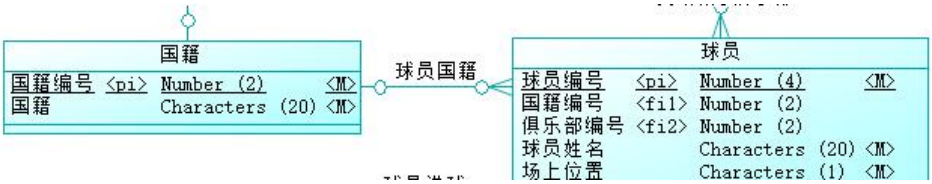
```
CREATE TRIGGER `goal_AFTER_INSERT` AFTER INSERT ON `goal`
FOR EACH ROW BEGIN
    update player
    set player.num_of_goal = player.num_of_goal + 1
    where player.player_id = new.player_id;
END
```

2.1.5 分析

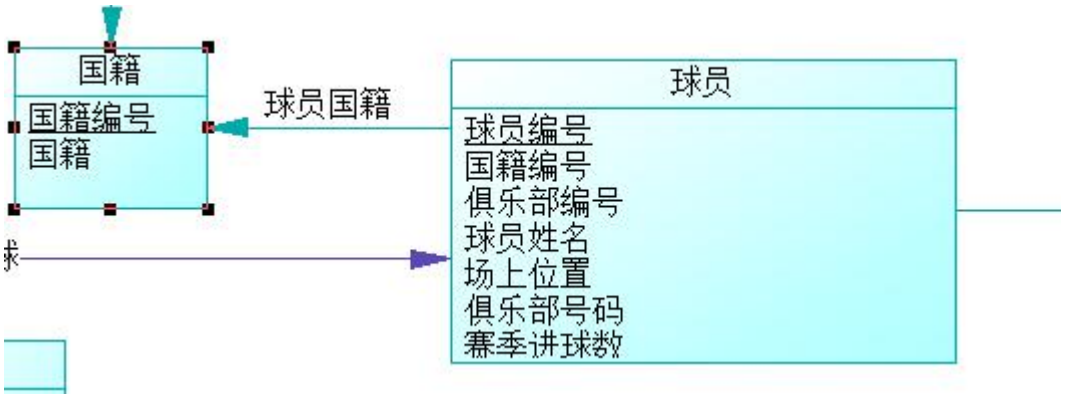
ER 图：



LDM 图：



PDM 图：



ER 图中的一对多或多对一模型在 LDM 图中在各自的实体中自动加入了外键。一般谁多谁加外键，在 PDM 中，一对多和多对一自动简化成立一个带箭头指向的单条线，外键不变。

3 收获和反思

（1） 收获：

通过这次的数据库设计，我对 ER 模型以及主外键等有了更深的了解，对数据库的创建、管理和操作有了进一步的掌握。同时，对 java 的使用也更加熟练，积累了更多数据库开发乃至小型系统开发的相关经验。我相信这些对于我的未来发展有很大帮助。

（2） 反思：

在设计 ER 模型设计初始阶段，对 ER 模型理解不深刻，一开始就加入了外键，导致浪费了一些时间，在复习相关知识后，设计思路就更加明确了。