# 数据采集和数据库设计说明文档

组号：22 组员：朱静怡，王宁，于钦，季灵杰

1. **数据采集：**

**(1)采集模式：**使用python爬虫从<http://stats.nba.com/>爬取数据

**(2)采集目标：**为数据库建立提供基本数据支持

**(3)采集内容：**05-15赛季所有常规赛和季后赛的详细信息(boxscore)，05-15赛季球员比赛数据信息(按常规赛和季后赛分开)，05-15赛季所有球队中球员的基本信息，05-15赛季球队比赛数据信息(按常规赛和季后赛分开)，05-15赛季所有球队基本信息

**(4)采集的问题：**数据量过大，数据类别过多，分类数据时容易出错

**(5)采集数据的结构：**比赛详细信息（比赛时间，比赛球队，总比分，单节比分，上场球员的名字、位置、时间、投篮次数、命中数、三分投篮次数、命中数、罚球次数、命中数、前场篮板、后场篮板、总篮板、犯规，失误，盖帽、抢断、助攻）

球员基本信息按赛季和球队划分（球员名称，球衣号码，位置，身高，体重，出生日期，年龄，球龄，毕业大学）

球队基本信息按赛季划分（全名，缩写，地址，东西部，赛区，主场，建立日期）

球员比赛信息按赛季划分（投篮数，篮板数，助攻数，效率等）

球队比赛信息按赛季划分（投篮数，篮板数，助攻数，进攻效率等）

**(6)采集数据的分析：**

从[http://stats.nba.com/stats/boxscoretraditionalv2?EndPeriod=10&EndRange=**28800**&GameID=**0020500001**&RangeType=2&Season=2005-06&SeasonType=Regular+Season&StartPeriod=1&StartRange=0](http://stats.nba.com/stats/boxscoretraditionalv2?EndPeriod=10&EndRange=28800&GameID=0020500001&RangeType=2&Season=2005-06&SeasonType=Regular+Season&StartPeriod=1&StartRange=0)获取一场比赛的boxscore；

从<http://stats.nba.com/stats/teamyearbyyearstats?LeagueID=00&PerMode=Totals&SeasonType=Regular+Season&TeamID=1610612766>获取球队历年基本信息；

从<http://stats.nba.com/stats/commonteamroster?LeagueID=00&Season=2014-15&TeamID=1610612737>获取球队中球员的基本信息

从<http://stats.nba.com/stats/leaguedashplayerstats?Conference=&DateFrom=&DateTo=&Division=&GameScope=&GameSegment=&LastNGames=0&LeagueID=00&Location=&MeasureType=Usage&Month=0&OpponentTeamID=0&Outcome=&PORound=0&PaceAdjust=N&PerMode=Totals&Period=0&PlayerExperience=&PlayerPosition=&PlusMinus=N&Rank=N&Season=2014-15&SeasonSegment=&SeasonType=Playoffs&ShotClockRange=&StarterBench=&TeamID=0&VsConference=&VsDivision>=获取赛季球员的使用率信息

从<http://stats.nba.com/stats/leaguedashplayerstats?Conference=&DateFrom=&DateTo=&Division=&GameScope=&GameSegment=&LastNGames=0&LeagueID=00&Location=&MeasureType=Base&Month=0&OpponentTeamID=0&Outcome=&PORound=0&PaceAdjust=N&PerMode=Totals&Period=0&PlayerExperience=&PlayerPosition=&PlusMinus=N&Rank=N&Season=2014-15&SeasonSegment=&SeasonType=Playoffs&ShotClockRange=&StarterBench=&TeamID=0&VsConference=&VsDivision>=获取赛季球员的基本信息

从<http://stats.nba.com/stats/leaguedashplayerstats?Conference=&DateFrom=&DateTo=&Division=&GameScope=&GameSegment=&LastNGames=0&LeagueID=00&Location=&MeasureType=Advanced&Month=0&OpponentTeamID=0&Outcome=&PORound=0&PaceAdjust=N&PerMode=Totals&Period=0&PlayerExperience=&PlayerPosition=&PlusMinus=N&Rank=N&Season=2014-15&SeasonSegment=&SeasonType=Playoffs&ShotClockRange=&StarterBench=&TeamID=0&VsConference=&VsDivision>=获取赛季球员的高阶数据

从<http://stats.nba.com/stats/leaguedashteamstats?Conference=&DateFrom=&DateTo=&Division=&GameScope=&GameSegment=&LastNGames=0&LeagueID=00&Location=&MeasureType=Base&Month=0&OpponentTeamID=0&Outcome=&PORound=0&PaceAdjust=N&PerMode=PerGame&Period=0&PlayerExperience=&PlayerPosition=&PlusMinus=N&Rank=N&Season=2011-12&SeasonSegment=&SeasonType=Regular+Season&ShotClockRange=&StarterBench=&TeamID=0&VsConference=&VsDivision>=获取赛季球队的基本数据

从<http://stats.nba.com/stats/leaguedashteamstats?Conference=&DateFrom=&DateTo=&Division=&GameScope=&GameSegment=&LastNGames=0&LeagueID=00&Location=&MeasureType=Advanced&Month=0&OpponentTeamID=0&Outcome=&PORound=0&PaceAdjust=N&PerMode=PerGame&Period=0&PlayerExperience=&PlayerPosition=&PlusMinus=N&Rank=N&Season=2011-12&SeasonSegment=&SeasonType=Regular+Season&ShotClockRange=&StarterBench=&TeamID=0&VsConference=&VsDivision>=获取赛季球队的高阶数据

**(7)数据清洗：**通过正则表达式分割，再按信息格式使用split，replace等函数进行数据清洗

**(8)数据处理：**根据球员的唯一ID合并球员的基本，高阶和使用率数据

根据球队的唯一ID合并球队的基本，高阶信息

1. **数据库设计：**

**(1)数据库结构：**纵向上将nba分为各个赛季，每个赛季分为常规赛和季后赛两种；横向上将每个赛季的数据分为球员数据、球队数据、比赛数据，其中比赛数据分为总体比赛描述以及比赛中球员具体表现。另外设置了球员和球队的总表，用于记录每个球员、球队的特有ID及基本信息。

**(2)实体关系：**球员隶属于球队，每场比赛由球队参与，包含一定球员。球员总表包含球员ID（唯一固定），以便在每个赛季的球员数据中存储球员信息，一位球员的基本信息存于球员总表，赛季表现存于该赛季球员数据表，二者通过ID联系。球队与球员类似，球队总表存储该球队最新的基本信息，于每张分表里另存储该球队当时赛季的基本信息及数据。比赛存于比赛总表之中，并以唯一固定的比赛ID标识，于具体比赛信息表中可根据该ID查看各球员具体表现。

**(3)设计考虑：**首先由于存储多年数据，每年数据应以不同命名但一定格式存储，方便处理。

将球置于总表之中，方便查询，联系各张赛季分表可查看每个球员赛季表现。将球队同样以存于总表，方便获得球队最新赛季的基本情况，也可以通过查询赛季分表得到赛季表现。

比赛分为总述及球员具体数据两种表，方便比赛的查询，并且不影响每场比赛具体数据的查看。

附：数据层接口说明文档

**1. 数据层模块的职责**

数据层模块的职责如下表所示。

|  |  |
| --- | --- |
| **模 块** | **职 责** |
| PlayerData | 持久化数据库的接口，提供球员数据载入、保存、增、删、改、查等服务 |
| MatchData | 持久化数据库的接口，提供比赛数据载入、保存、增、删、改、查等服务 |
| TeamData | 持久化数据库的接口，提供球队数据载入、保存、增、删、改、查等服务 |

1. **数据层模块的接口规范**

PlayerData模块接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务(供接口)** | | |
| PlayerDataService.getSinglePlayerInfo | 语法 | Public playerPO getSinglePlayerInfo(String playerName,String season) |
| 前置条件 | 输入合法球员姓名 |
| 后置条件 | 查找并返回该球员该赛季（可能是全部赛季，下同）的信息 |
| PlayerDataService.getAllPlayerInfo | 语法 | Public ArrayList<playerPO> getAllPlayerInfo(String season) |
| 前置条件 | 查询所有球员该赛季的信息 |
| 后置条件 | 返回所有球员信息 |
| PlayerDataService.getTodayPlayerInfo | 语法 | Public ArrayList<todayPlayerPO> getTodayPlayerInfo(String date,String season) |
| 前置条件 | 输入合理查询比赛日期 |
| 后置条件 | 返回当日比赛球员信息 |

TeamData模块接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务(供接口)** | | |
| TeamDataService.getSingleTeamInfo | 语法 | Public TeamPO getSingleTeamInfo(String teamName,String season) |
| 前置条件 | 输入合法球队名 |
| 后置条件 | 查找并返回该球队该赛季信息 |
| TeamDataService.getAllTeamInfo | 语法 | Public ArrayList<teamPO> getAllTeamInfo(String season) |
| 前置条件 | 查询所有球队该赛季信息 |
| 后置条件 | 返回所有球队信息 |

MatchData模块接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务(供接口)** | | |
| MatchDataService.getSingleMatchInfo | 语法 | Public MatchPO getSingleMatchInfo(String teamName,String date,String season) |
| 前置条件 | 输入合法球队名及比赛日期 |
| 后置条件 | 查找并返回该场比赛信息 |
| MatchDataService.getAllMatchInfo | 语法 | Public ArrayList<Match PO> getAllMatchInfo(String season) |
| 前置条件 | 输入赛季合法 |
| 后置条件 | 返回该赛季所有比赛信息 |
| MatchDataService.getTeamRecentMatchInfo | 语法 | Public ArrayList<MatchPO> getTeamRecentMatchInfo(int number,String teamName) |
| 前置条件 | 输入合法 场数及球队名 |
| 后置条件 | 返回筛选的比赛信息 |
| MatchDataService.getPlayerRecentMatchInfo | 语法 | Public ArrayList<MatchPO> getPlayerRecentMatchInfo(int number,String playerName) |
| 前置条件 | 输入合法场数及球员名 |
| 后置条件 | 返回筛选的比赛信息 |
| MatchDataService.getTodayMatchInfo | 语法 | Public ArrayList<MatchPO> getTodayMatchInfo(String date,String season) |
| 前置条件 | 输入合法 |
| 后置条件 | 返回筛选的比赛信息 |
| MatchDataService.getMatchOfTeam | 语法 | Public ArrayList<MatchPO> getMatchOfTeam(String teamName,String season) |
| 前置条件 | 输入合法场数及球员名 |
| 后置条件 | 返回筛选的比赛信息 |