**数据采集**

**采集目标**：**1.历史数据**

Nba球队变更历史信息

Nba历史所有球员信息

Nba各个赛季比赛基本信息，技术统计，文字直播

Nba 各赛季球员薪水排名(用于推断统计)

Nba 今日热点球员获取

**2.在线同步数据**

正在进行比赛的数据与文字直播

**采集来源**：nba英文官网，nba中文官网，espn网站

**采集模式**：以比赛技术统计信息为例

ps所有的网站信息爬取类均基于WebSec类

WebSec类里有以下三个基本方法：

//获取destStr中与正则相匹配的List

**public** **static** ArrayList<String> getMatcherSubstrs(String destStr,String regex)

//打开指定url的网页内容

**public** **static** String getURLContent(String urlString,String charset)

//根据url和文件名保存网络图片

**public** **static** **void** SavePic(String url,String path,String fileName)

**采集步骤**：

Step1:使用网站分析工具，获取网站后台交换数据的url

例如：

<http://stats.nba.com/stats/scoreboardV2?DayOffset=0&LeagueID=00&gameDate=05%2F12%2F2015>

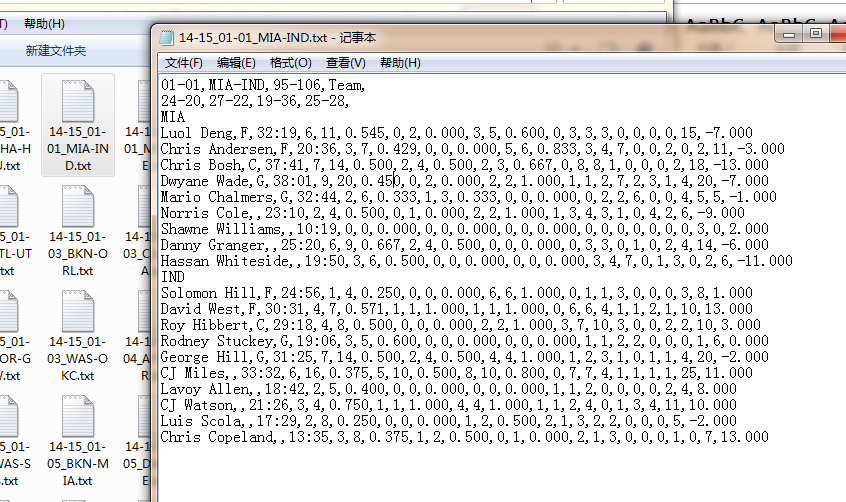
变量值为gameDate,然后通过构建不同的gameDate拼接新的url获取每天的比赛数据。

Step2:使用正则匹配，从该url网页内容中匹配出ArrayList

如String gameIDPattern="\"[\\d]{10}\"";

Step3:对拿到的List通过基本的replace，split等操作，组织成预先设定好的格式

如技术统计单元为：



如果官网上的技术统计数据不全，则作脏数据处理：在文件名上加缺失标识。

Step4:若为离线数据，按要求存储命名成txt，由数据库存入。

否则，直接传给界面构造显示

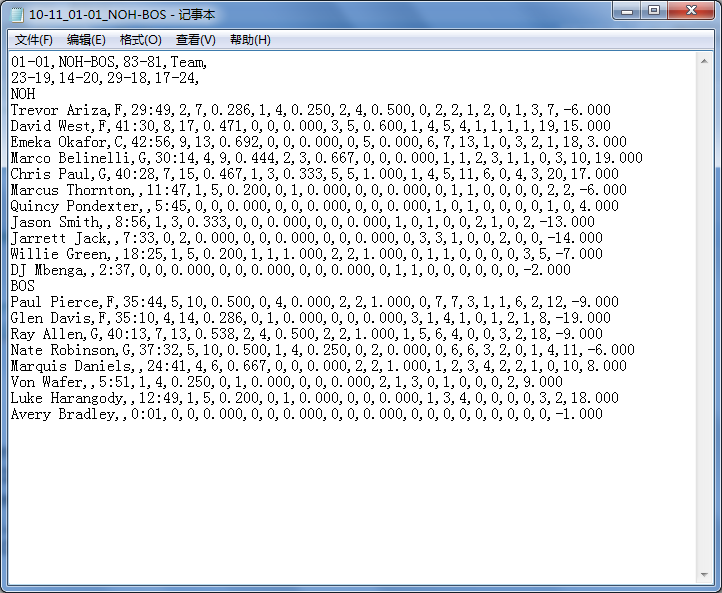
Ps:1.有些数据类型需要进入多个网页页面获取，例如从赛程界面拿到gameID,根据gameID进入到每个比赛的详情界面，拿到技术统计和文字直播。

2.英文网站与中文网站比赛日期相差一天，需做日期转化处理

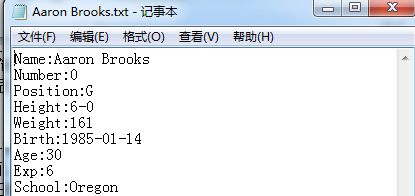
3.虽然中文网站后台数据多为json类型对象，但本次爬取站源多为英文官网，最终以正则匹配为主。

**采集的原始数据的结构：**

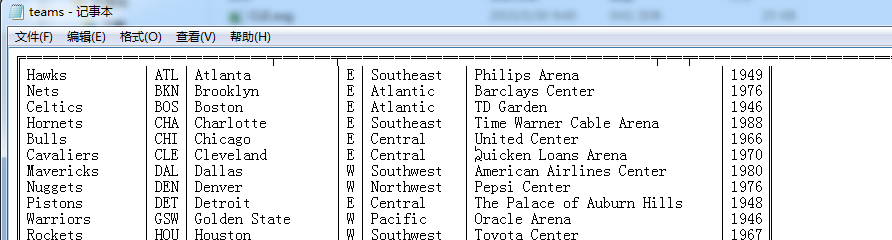
比赛的txt文件结构:



球员的txt文件结构:



球队的txt文件结构：



**脏数据处理：**

对于数据不全的一场比赛，检测到文件名的缺失标记后，在导入数据库的过程中会跳过不统计。

对球龄进行检查，如果是非数字就返回0，超过合理范围的返回-1。

对球衣号码进行检查，如果是非数字就返回-1。

检查每场比赛每位球员的得分是否合理 1、判断得分是否是数字 2、判断得分计算是否准确。如果得分合理，返回int型得分，否则根据球员该场比赛的表现计算出相应的得分并返回。

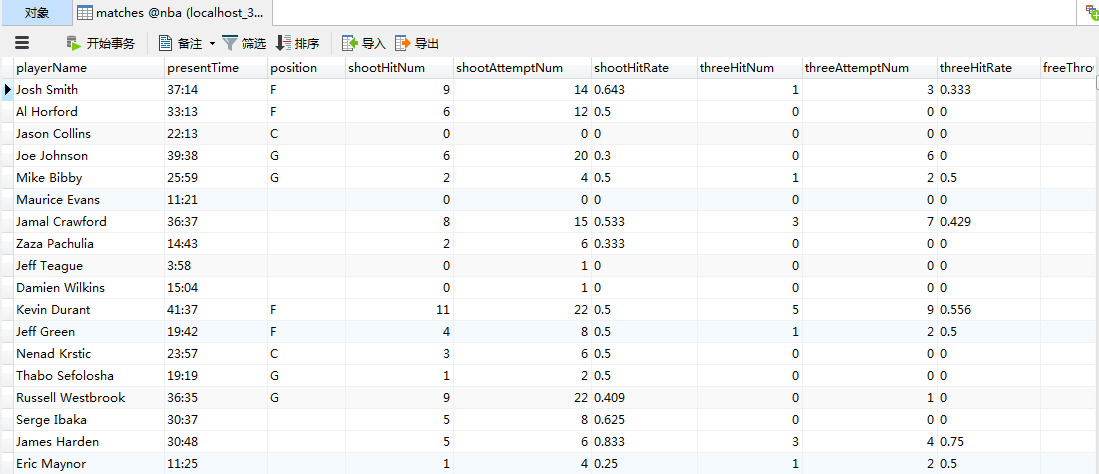
检查出手次数是否大于等于命中次数，正确的出手次数——出手次数大于等于命中次数，如果给出的出手次数小于命中次数，就将出手次数置为命中次数。

校验在场时间是否符合要求，如果在场时间是x:xx格式的，直接返回，如果是None（即该球员未上场比赛），统一置为“0：00”，其他数据缺失置为null的，读文件计算出正确时间返回。

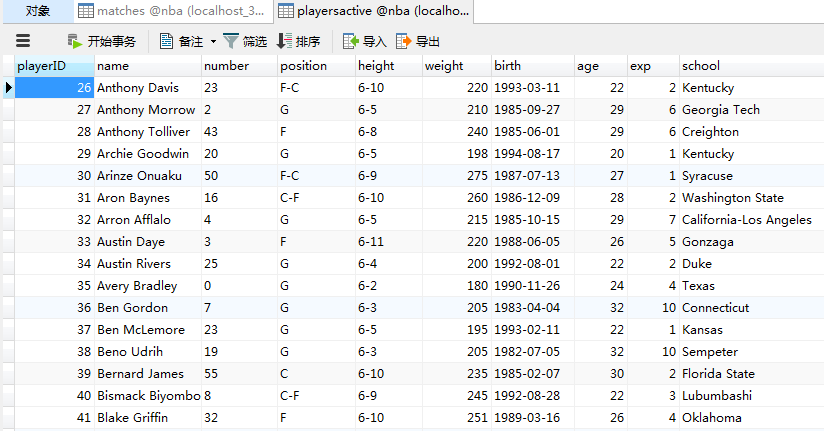
球员在场时间丢失，读取比赛记录，利用公式：所有球员在场时间之和=比赛总时间\*5，求得该时间。

**数据库数据结构:**

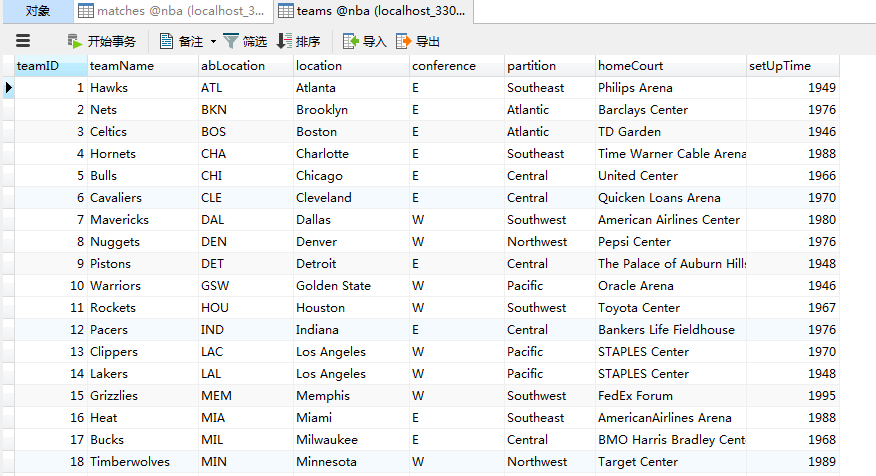
比赛的数据结构：



球员的数据结构：



球队的数据结构：



**数据库设计**

Player的基本信息(名字，身高，体重等)存一个表；

Team 的基本信息(简称，所在联盟，成立时间等)存一个表；

历史文字直播的数据与数据网页上的球员ID与球员名称对应分别存一个表；

比赛的各个球员的赛场数据存一个表；

比赛各节比赛的比分存一个表；

各个赛季的不同类型的比赛，球员球队的赛场数据分别存表；

球员不同赛季的薪水分别存表；