数据整理与清洗方案

# *顾高飞 2016.10.31*

# 数据整理放在 *PlatDataCleanUp* 数据库中进行，整理完成后，再更新到 *TechTradePlat* 库中。

# 首先做一些前期处理准备

## 首先对项目、专利、论文三张表的重要字段(name、organization、unit、leader、member、applicationdate、starttime….)的**空格进行去除**，因为空格会影响匹配和归类（需要注意的是，由于很多数据和都是爬取过来的，所以里面的空格和自己手动打出来的空格可能不是同一种空格，也许是编码问题，所以需要针对表中出现的空格来进行去除）

### SQL脚本( project 表为例)：

## UPDATE project SET xx = REPLACE(xx, ‘ ’, ‘’);

## 然后对去掉项目、专利、论文表**进行去重操作**，插入临时表project2，之后project2就作为新的project表

### SQL脚本( project 表为例，项目名字和负责人相同代表同一项)：

## INSERT INTO project2 (SELECT \* FROM project GROUP BY name, leader);

## **处理三张表中的成员字段**，其中 patent 表中的成员字段(inventors)比较规范：第一发明人包含在成员字段中，并且位于第一个；各个成员之间使用英文分号分隔；首尾不带分号

### 由于成员字段可能的情况较多，所以需要 Python 脚本处理，处理之后，确保字段样式如上所述。

## **抽取 patent 表中的 applicant 字段和 areacode 字段**，插入到 organization\_areacode\_map 表中，并进行处理（去掉人名和多单位的情况），再进行去重。**以后维护该表，用该表来个更新其他表的单位所属的地区。**

### 抽取的SQL脚本：

## INSERT INTO organization\_areacode\_map\_tmp (unit, areacode) (SELECT applicant, areacode FROM patent GROUP BY applicant);

### 单位的情况比较复杂：有可能有多个单位合在一起，一般使用英文分号分隔，这种情况只需要保留第一个分号之前的第一个单位；有可能只是人名，这种情况需要直接删除。处理的时候需要人工+SQL脚本+Python脚本。

### 上面这步处理之后，进行去重，SQL脚本：

## INSERT INTO organization\_areacode\_map (unit, areacode) (SELECT unit, areacode FROM organization\_areacode\_map\_tmp GROUP BY unit);

# 处理项目表

## **处理 project 表中的areacode字段**，用organization\_areacode\_map 表来更新。areacode字段中的数据为地区数字代码，代号对应的实际省份地区存在于areacode\_province\_map映射表中。

### SQL脚本：

## UPDATE project p, organization\_areacode\_map o SET p.areacode = o.areacode WHERE p.unit = o.unit;

## **处理project表中的starttime字段和endtime字段**，确保：starttime和endtime格式为xxxx-xx-xx或者xxxx-xx，即日期分隔号为 -

### 原始爬取数据中，有些数据的日期分隔号可能为 . 或者 / 或者直接使用年月日的方式分隔，则需要将这些字符替换为 - ，SQL脚本如下（以starttime为例）：

## UPDATE project SET starttime = REPLACE(starttime, ‘/’, ‘-’);

### 有些记录的starttime字段和endtime字段可能不存在，则需要判断对应的timespan字段是否存在，如果存在，就从中抽取相应的时间数据插入到相应的字段中。这需要人工+SQL脚本+Python脚本来处理。

## **处理project表中的year字段**，确保：year字段只是由四位年份数字组成

### 在原始爬取数据中，有些year字段并不只是年份，而是可能包括月份或者日期，对于这类数据，只需取其前4个字符即可（前提是已经去掉了空格）；有些year字段原始数据是空的，则判断其starttime字段是否存在，若存在，则取starttime字段的年份（前四个字符）作为year字段的数据。

## UPDATE project SET year = left(year, 4);

## UPDATE project SET year = left(starttime, 4) where year is null;

# 处理专利表

## **处理patent表中的year字段**

### 取patent表中的applicationdate字段，该字段的结构为xxxx-xx-xx，所以只要取其前四个字符，插入到year字段中即可。SQL脚本如下：

## UPDATE patent SET year = left(applicationdate, 4);

# 处理论文表

## **处理 paper表中的unit字段**，确保：该单位是最大单位（例如 ‘杭州电子科技大学’ 而不是 ‘杭州电子科技大学计算机学院’）；该单位数据存在于表organization\_areacode\_map中，否则为空

### 首先抽取原先organization字段中的单位数据，一般该字段都是有好几个字段组成的，使用分号分隔，则统一抽取第一个分号之前的第一个单位，插入到unit字段中，作为初始unit字段值。使用Python脚本处理。

### 因为现在导入的unit字段里面的单位数据并不是最大单位，因此还要继续处理unit的值。利用organization\_areacode\_map表中的单位，和unit中的单位进行前缀匹配，如果匹配上，就将unit更改为organization\_areacode\_map表中匹配上的单位值。*<这一步速度比较慢>*

## **处理 paper表中的areacode字段**，用organization\_areacode\_map 表来更新。areacode字段中的数据为地区数字代码，代号对应的实际省份地区存在于areacode\_province\_map映射表中。

### SQL脚本：

## UPDATE paper p, unit\_areacode\_map o SET p.area\_code = o.area\_code WHERE p.unit = o.unit;

## **处理 paper表中的year字段**，四位年份数字

### paper表的journal\_no字段中包含了当前论文在所在期刊的期刊号，而期刊号内包含了年份时间，因此只要取该年份时间数据作为year字段的数据即可。SQL脚本：

## UPDATE paper SET year = left(journal\_no, 4);

# 处理专家表

## 进行到此步骤时，理论上项目、专利、论文表中的第一作者及其单位都已经处理完毕，则**抽取项目、专利、论文表中的作者，插入到专家表中**，并进行**去重**操作。约定：名字和单位相同的代表同一个人。以下操作步骤的SQL脚本以 patent 表为例，其他表的操作类似。

### 首先将专利表中的作者和其对应的单位都插入到临时表expert2中

## INSERT INTO expert2 (name, unit) SELECT inventor, unit FROM patent GROUP BY inventor, unit;

### 然后将临时表expert2中unit字段为空的记录删除

## DELETE FROM expert2 WHERE unit IS NULL;

### 将临时表expert2中的数据插入专家表expert中，并插入临时表expert3去重，GROUP BY默认会保留顺序靠前的一条记录，而靠前的老记录一般会比后插入的只有名字和单位的记录更详细，所以一般不会丢失其他信息。之后expert3就作为新的expert表

## INSERT INTO expert (name, unit) SELECT name, unit FROM expert2;

## INSERT INTO expert3 SELECT \* FROM expert GROUP BY name, unit;

# 关联专家表与项目、专利、论文表

## 进行到此步骤时，理论上项目、专利、论文以及专家表最重要的一些字段已经处理完成了，则可以开始**进行表之间的关联**了。

### 以下操作步骤的SQL脚本以 patent表为例，其他表的操作类似

## INSERT INTO expert\_patent\_join(EXPERT\_ID, expert, PATENT\_ID, patent) SELECT e.EXPERT\_ID EXPERT\_ID, e.name expert, p.PATENT\_ID PATENT\_ID, p.name patent FROM expert e, patent p WHERE e.name = p.name AND e.unit = p.unit;