

# E06 Queries on KB

---

17341189 姚森舰

2019 年 10 月 13 日

## 目录

<b>1 Problem Description</b>	<b>2</b>
<b>2 Codes and Results</b>	<b>2</b>

# 1 Problem Description

Given a KB `Restaurants.pl`, which describes the distribution of branches of 10 well-known restaurants in Guangzhou.

For example, `restaurant(ajukejiacai,2007,yuecai)` means that `ajukejiacai` was founded in 2007 and is a restaurant of `yuecai`. `branch(ajukejiacai,xintiandi)` means that `ajukejiacai` has a branch in `xintiandi`. `district(xintiandi,panyu)` means that `xintiandi` is an area of `panyu` district.

Please formulate each of the following questions as a query using Prolog's notation, pose it to Prolog, and obtain Prolog's answer:

1. What restaurants have branches in beigang?
2. What districts have restaurants of yuecai and xiangcai?
3. What restaurants have the least number of branches?
4. What areas have two or more restaurants?
5. Which restaurant has the longest history?
6. What restaurants have at least 10 branches?

Please define the new relation below using Prolog and test it.

- `sameDistrict(Restaurant1, Restaurant2)`: Restaurant1 and Restaurant2 have one or more branches in the same district.

You should write down a listing that shows the queries you submitted to Prolog, and the answer returned. Hand in a file named `E06_YourNumber.pdf`, and send it to `ai_201901@foxmail.com`

# 2 Codes and Results

为了使查询更简洁，需要在 `Restaurant.pl` 中添加的语句：

```
% 2 区Dis 有种类为Type的菜
has_type(Dis,Type):-restaurant(R,_,Type),branch(R,Area),district(Area,Dis).

% 3 饭店R 有N个分支
num_branches(R,N):-setof(B,(restaurant(R,_,_),branch(R,B)),Branches), length(Branches,N).

% 4 地区A 有两个或更多饭店
more_than_2R(A):-setof(R,branch(R,A),L),length(L,Len),Len>=2.

% 6 饭店R有10个分支
more_than_10B(R):-setof(B,branch(R,B),L),length(L,Len),Len>=10.

% 7 饭店R1, R2, 在同一个区都有分支
sameDistrict(R1, R2):-branch(R1, A1), branch(R2, A2), district(A1, D), district(A2, D), R1\=R2.
```

6 个问题的查询结果依次列举如下:

```
?- setof(R,branch(R,beigang),Restaurant).
Restaurant = [huangmenjimifan, mixuebingcheng, shaxianxiaochi].

?- setof(D,(has_type(D,yuecai),has_type(D,xiangcai)),Districts).
Districts = [haizhu, liwan, panyu, tianhe, yuexiu].

?- findall(R, (findall(N1,num_branches(R,N1), Rsbns), num_branches(R,N), min_list(Rsbns,N)), Ans).
Ans = [hongmenyan].

?- setof(A,more_than_2R(A),Areas).
Areas = [bainaozhui, beigang, dongpu, shiqiao, tianhebei, xintiandi, yongfu, yuancun].

?- findall(R, (findall(Y1,restaurant(R,Y1,_),Found_year), restaurant(R,Y,_), min_list(Found_year,Y)), Ans).
Ans = [huangmenjimifan].

?- setof(R,more_than_10B(R),Ans).
Ans = [dagangxianmiaoshaoji, diandude, mixuebingcheng, muvushaokao, tongxianghui].
```

其中 3 的实现思路是先找到所有饭店的分支数构成的列表 (上面的 Rsbns), 然后用 min\_list 判断每个饭店的分支数是不是等于列表中最少的, 若是则是我们要找到答案。5 的实现类似。最后一个 sameDistrict(R1,R2) 的实现在上面已经列出来了, 以下是测试结果:

```
?- sameDistrict(diandude,yangguofu).
true.

?- setof(X,sameDistrict(diandude,X),Ans).
Ans = [ajukejiacai, dagangxianmiaoshaoji, hongmenyan, huangmenjimifan, mixuebingcheng, muvushaokao, shaxianxiaochi, tongxianghui, yangguofu].

?- setof(pair(X,Y),sameDistrict(X,Y),Ans),write(Ans),nl,fail.
[[pair(ajukejiacai,dagangxianmiaoshaoji),pair(ajukejiacai,diandude),pair(ajukejiacai,hongmenyan),pair(ajukejiacai,huangmenjimifan),pair(ajukejiacai,mixuebingcheng),pair(ajukejiacai,muvushaokao),pair(ajukejiacai,shaxianxiaochi),pair(ajukejiacai,tongxianghui),pair(ajukejiacai,yangguofu),pair(dagangxianmiaoshaoji,ajukejiacai),pair(dagangxianmiaoshaoji,diandude),pair(dagangxianmiaoshaoji,hongmenyan),pair(dagangxianmiaoshaoji,huangmenjimifan),pair(dagangxianmiaoshaoji,mixuebingcheng),pair(dagangxianmiaoshaoji,muvushaokao),pair(dagangxianmiaoshaoji,shaxianxiaochi),pair(dagangxianmiaoshaoji,tongxianghui),pair(dagangxianmiaoshaoji,yangguofu),pair(diandude,ajukejiacai),pair(diandude,dagangxianmiaoshaoji),pair(diandude,hongmenyan),pair(diandude,huangmenjimifan),pair(diandude,mixuebingcheng),pair(diandude,muvushaokao),pair(diandude,shaxianxiaochi),pair(diandude,tongxianghui),pair(diandude,yangguofu),pair(hongmenyan,ajukejiacai),pair(hongmenyan,dagangxianmiaoshaoji),pair(hongmenyan,diandude),pair(hongmenyan,huangmenjimifan),pair(hongmenyan,mixuebingcheng),pair(hongmenyan,shaxianxiaochi),pair(hongmenyan,tongxianghui),pair(hongmenyan,yangguofu),pair(huangmenjimifan,ajukejiacai),pair(huangmenjimifan,dagangxianmiaoshaoji),pair(huangmenjimifan,diandude),pair(huangmenjimifan,hongmenyan),pair(huangmenjimifan,mixuebingcheng),pair(huangmenjimifan,muvushaokao),pair(huangmenjimifan,shaxianxiaochi),pair(huangmenjimifan,tongxianghui),pair(huangmenjimifan,yangguofu),pair(mixuebingcheng,ajukejiacai),pair(mixuebingcheng,dagangxianmiaoshaoji),pair(mixuebingcheng,diandude),pair(mixuebingcheng,hongmenyan),pair(mixuebingcheng,huangmenjimifan),pair(mixuebingcheng,muvushaokao),pair(mixuebingcheng,shaxianxiaochi),pair(mixuebingcheng,tongxianghui),pair(mixuebingcheng,yangguofu),pair(muvushaokao,ajukejiacai),pair(muvushaokao,dagangxianmiaoshaoji),pair(muvushaokao,diandude),pair(muvushaokao,huangmenjimifan),pair(muvushaokao,mixuebingcheng),pair(muvushaokao,tongxianghui),pair(muvushaokao,yangguofu),pair(shaxianxiaochi,ajukejiacai),pair(shaxianxiaochi,dagangxianmiaoshaoji),pair(shaxianxiaochi,diandude),pair(shaxianxiaochi,hongmenyan),pair(shaxianxiaochi,huangmenjimifan),pair(shaxianxiaochi,mixuebingcheng),pair(shaxianxiaochi,tongxianghui),pair(shaxianxiaochi,yangguofu),pair(tongxianghui,ajukejiacai),pair(tongxianghui,dagangxianmiaoshaoji),pair(tongxianghui,diandude),pair(tongxianghui,hongmenyan),pair(tongxianghui,huangmenjimifan),pair(tongxianghui,mixuebingcheng),pair(tongxianghui,muvushaokao),pair(tongxianghui,shaxianxiaochi),pair(tongxianghui,yangguofu),pair(yangguofu,ajukejiacai),pair(yangguofu,dagangxianmiaoshaoji),pair(yangguofu,diandude),pair(yangguofu,hongmenyan),pair(yangguofu,huangmenjimifan),pair(yangguofu,mixuebingcheng),pair(yangguofu,muvushaokao),pair(yangguofu,shaxianxiaochi),pair(yangguofu,tongxianghui)]]
```

要稍微注意的是, 里面会有这种重复, 比如 [R1,R2] 在同一个地方有分支, 则 [R2,R1] 也会再出现一次。