

# E06 Queries on KB

---

17341203 Yixin Zhang

October 13, 2019

## Contents

<b>1</b>	<b>Problem Description</b>	<b>2</b>
<b>2</b>	<b>Codes and Results</b>	<b>2</b>
2.1	Six Queries . . . . .	2
2.2	Same District Relation . . . . .	6
<b>3</b>	<b>Experience</b>	<b>8</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

### 2.1 Six Queries

1. What restaurants have branches in beigang?

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

2. What districts have restaurants of yuecai and xiangcai?

For convenience, I define a new auxiliary rule ‘categoryDistrict’ which gives names of districts where there is at least one restaurant offering food of a specific category. To do such thing, ‘assertz/1’ is used. Then this auxiliary rule is used to find the districts that have both yuecai and xiangcai.

Note that in my code ‘R’ means restaurants, ‘C’ means categories, ‘A’ means areas, ‘D’ means districts.

```
?- assertz((categoryDistrict(C,D):-restaurant(R,_,C),branch(R,A), district(A,D))).
true.

?- setof(D, (categoryDistrict(yuecai,D),categoryDistrict(xiangcai,D)), Result).
Result = [haizhu, liwan, panyu, tianhe, yuexiu].
```

3. What restaurants have the least number of branches?

(a) Step 1: Find branches of each restaurant

```
?- setof(A,branch(R,A),As).
R = ajukejiacai,
As = [shatainan, xintiandi, yongfu] ;
R = dagangxianmiaoshaoji,
As = [beishan, cencun, changxing, dongpu, fangcun, gaosheng, huadong, kecun,
      nanopudadao|...] ;
R = diandude,
As = [bainahui, huachengdadao, huifudong, linhe, panfu, shiqiao, tianhebei,
      yangji, youtuobangshiguang|...] ;
R = hongmenyan,
As = [xintiandi, zhilanwan] ;
... % omitted
```

(b) Step 2: Count their branches

```
?- setof(
|      Count,
|      As^( restaurant(R,_,_), setof(A,branch(R,A),As), length(As,Count)),
|      Counts
|      ).
```

```

R = ajukejiacai,
Counts = [3] ;
R = dagangxianmiaoshaoji,
Counts = [11] ;
R = diandude,
Counts = [10] ;
R = hongmenyan,
Counts = [2] ;
R = huangmenjimifan,
Counts = [7] ;
R = mixuebingcheng,
Counts = [12] ;
R = muwushaokao,
Counts = [10] ;
R = shaxianxiaochi,
Counts = [3] ;
R = tongxianghui,
Counts = [10] ;
R = yangguofu,
Counts = [4].

```

(c) Step 3: Make a list of counts

```

?- findall(
|      Counts,
|      setof(
|          Count,
|          As^( restaurant(R,_,_), setof(A,branch(R,A),As), length(As,Count)),
|          Counts
|      ),
|      CountList
|  ).
CountList = [[3], [11], [10], [2], [7], [12], [10], [3], [10],[4]].

```

(d) Step 4: Find the minimum value in the list

```

min_list(CountList, MinCount).

```

(e) Step 5: Find the restaurant whose branch count equals to the minimum value

```
?- setof(
|     R,
|     Count^As^CountList^(
|         findall(
|             Counts,
|             setof(
|                 Count,
|                 As^( restaurant(R,_,_), setof(A,branch(R,A),As),
|                 length(As,Count)),
|             Counts
|         ),
|         CountList
|     ),
|     min_list(CountList, MinCount), setof(A,branch(R,A),As),
|     length(As,Count), Count = MinCount
|     ),
|     Result
| ).
MinCount = 2,
Result = [hongmenyan].
```

(f) Result: The result is hongmenyan. It has only 2 branches.

#### 4. What areas have two or more restaurants?

We use nested 'setof' to accomplish this. The inner one gives those restaurants ('Rs') which are in area 'A'; Then the outer one counts the restaurants corresponding to different areas, and select those areas with a quantity greater than 2. To count the number of elements in a list, function 'length/2' is used.

```
?- setof(
|     A,
|     Count^Rs^(
|         setof(R, branch(R,A), Rs),
|         length(Rs, Count), Count>=2
|     ),
|     Result
```

```
|    ).
Result = [bainahui, beigang, dongpu, shiqiao, tianhebei, xintiandi, yongfu,
          yuancun].
```

## 5. Which restaurant has the longest history?

The idea is similar to Question 3, but this one is much easier. No more explanations needed.

```
?- setof(
|      R,
|      YearList^Y^(
|          findall(
|              Ys,
|              setof(Y,C^restaurant(R,Y,C),Ys),
|              YearList
|          ), min_list(YearList, MinYear), restaurant(R,Y,_), Y = MinYear
|      ),
|      Result
|  ).
MinYear = 1935,
Result = [huangmenjimifan].
```

## 6. What restaurants have at least 10 branches?

The idea is quite similar to Question 4.

```
?- setof(
|      R,
|      Count^ As^ (
|          setof(A, (branch(R,A)), As),
|          length(As, Count), Count>=10
|      ),
|      Result
|  ).
Result = [dagangxianmiaoshaoji, diandude, mixuebingcheng, muwushaokao, tongxianghui].
```

## 2.2 Same District Relation

Define the same district rule as below.

```
sameDistrict(R1,R2):-branch(R1,A1),branch(R2,A2),district(A1,D),district(A2,D),R1\=R2.
```

Test it:

```
?- sameDistrict(hongmenyan,ajukejiacai).
```

```
true .
```

```
?- sameDistrict(muwushaokao,diandude).
```

```
true .
```

We can also print all the restaurant pairs where they have one or more branches in the same district.

```
assertz((sameDistrict(R1,R2):-branch(R1,A1),branch(R2,A2),district(A1,D),district(A2,D),R1\=R2)).
```

```
assertz((printList([]))).
```

```
assertz((printList([H|T]):-write(H),nl,printList(T))).
```

```
?- setof((R1,R2), sameDistrict(R1,R2), Result), printList(Result).
```

```
ajukejiacai,dagangxianmiaoshaoji
```

```
ajukejiacai,diandude
```

```
ajukejiacai,hongmenyan
```

```
ajukejiacai,huangmenjimifan
```

```
ajukejiacai,mixuebingcheng
```

```
ajukejiacai,muwushaokao
```

```
ajukejiacai,shaxianxiaochi
```

```
ajukejiacai,tongxianghui
```

```
ajukejiacai,yangguofu
```

```
dagangxianmiaoshaoji,ajukejiacai
```

```
dagangxianmiaoshaoji,diandude
```

```
dagangxianmiaoshaoji,hongmenyan
```

```
dagangxianmiaoshaoji,huangmenjimifan
```

```
dagangxianmiaoshaoji,mixuebingcheng
```

```
dagangxianmiaoshaoji,muwushaokao
```

```
dagangxianmiaoshaoji,shaxianxiaochi
```

```
dagangxianmiaoshaoji,tongxianghui
```

```
dagangxianmiaoshaoji,yangguofu
```

```
diandude,ajukejiacai
```

```
diandude,dagangxianmiaoshaoji
```

```
diandude,hongmenyan
diandude,huangmenjimifan
...
% too much, omitted
```

### 3 Experience

When solving Qestion 2, I met a problem.

For example, I want to find out which districts have restaurants of yuecai. I tried the code below, but the result it produced was strange.

```
?- setof(D, R^A^(restaurant(R,_,yuecai), branch(R,A), district(A,D)), Result).
Result = [panyu, tianhe, yuexiu] ;
Result = [baiyun, panyu, yuexiu] ;
Result = [baiyun, haizhu, huadu, liwan, panyu, tianhe].
```

Why three different lists? However, the correct code is:

```
?- setof(D, R^Y^A^(restaurant(R,Y,yuecai), branch(R,A), district(A,D)), Result).
Result = [baiyun, haizhu, huadu, liwan, panyu, tianhe, yuexiu].
```

Note that the variable ‘Y’ cannot be replaced by the anonymous variable ‘\_’. Otherwise the result will be three lists instead of one, because there are restaurants founded in there different years having both yuecai and xiangcai, grouped by area ‘A’ and restaurant ‘R’. Thanks our TAs to help me understand this trick!