Result analysis:

Extra hard:

Correctness rate: 0.65

Rate of needs to add Types and Values: 0.08

Rate of needs to add Types xor Values: 0.08

Rate of needs to add Types only: 0.25

Rate of needs to add Values only: 0.08

Rate of needs to not add Types or Values: 0.58

Reasons to failure:  
1. gpt try to compare student id to student last name  
2. gpt doesn’t drop duplicate of the country and series name  
3. aggergate expression don’t allow #num in the output. For example sum(#3.Population) is illigal. And the output of sum have to be from the form SUM(Population) AS Sum\_Population  
4. gpt calculate the id of the professionals of the second part of the question, but instead of finding their names and union it with the first half, it join the id with the first half  
5. gpt find the zip code in #1, doesn’t join it with anything and try to use it in the end  
6. gpt use in filter in wrong way. It try to act it as join.  
7. gpt generated very long qpl plan, therefore it made him do illegal actions

hard:

Correctness rate: 0.6

Rate of needs to add Types and Values: 0

Rate of needs to add Types xor Values: 0

Rate of needs to add Types only: 0

Rate of needs to add Values only: 0

Rate of needs to not add Types or Values: 1

Reasons to failure:  
1. in some plans it doesn’t output the id field and try to use it. In others there are ambigues with fields that appears in both tables (on join)  
2. the qpl want the except to be:  
#3 = Except [ #1 , #2 ] Predicate [ #2.Channel = #1.id ] Output [ #1.id ]  
and not  
#3 = Except [ #1 , #2 ] Predicate [ #1.id = #2.Channel ] Output [ #1.id ]

For conclusion:

We saw that the results with qpl are much better than sql.

In hard we can see that the types or values not helping so much to the gpt, but in extra hard types or values have impact to the qpl plan.