Some Hints on Labsheet 4

November 3, 2023

1. Result: 90 rows

   Hint: Easy

2. Result: 15 rows

   Hint: Easy

   Result: AS above

   Hint:

3. Result: Three rows (beer, crisps and nutella)

   Hint: Easy

4. Result: Eight rows (for AA, DD ansd FF)

   Hint: Use —— for concatenating name parts into single string

5. Result: Seven rows

   Hint: Use DISTINCT to suppress duplicates

6. Result: Two rows (for DD and EE)

   Hint: Easy

7. Result: Cork

   Hint: Don't forget DISTINCT

8. Result: 225 rows

   Hint: Easy

9. Result: 12 rows

   Hint: Self join likes x likes, linked by common person id; Use ¡ trick (on food attribute) to suppress pointless self pairs (X, X) and symmetric pairs (X, Y) and (Y, X).

10. Result: Empty. Could temporarily experiment with foods that individuals are known to like both of (e.g. pizza and crisps)

    Hint: Self join likes x likes, linked by common person id. Use WHERE to restrict to pairs where one if the likes relates to pizza, the other to nutella.

11. Result: 4 rows

    Hint: Despite seeming similarity with Q10 above ('or' instead of 'and' in the problem statement), this one is a lot simpler, requiring no joins at all!

12. Result: Two rows (AA and DD)

    Hint: Two way join (persons and likes)

13. Result: Empty. (AS with Q10 could experiment with e.g. pizza ann crisps when developing solution.)

    Hint: Need 3way join here: persons x likes(1) x likes(2) to capture person with two food preferences for that person. Trying to work WITH only one copy of likes is doomed to failure. All three elements of triple should agree on the person id aspect; Use WHERE clause to focus in on those triples/persons with fondness for pizza and nutella.

14. Result: Four rows

    Hint: AS with Q10 and Q11, this is easier than Q13 and only requires 2way join (persons x likes).

15. Result: 36 rows

    Hint:

16. Result: 15 rows

    Hint: Use ¡ trick (on person's id) to filter out self pairs (X, X) and symmetric pairs (X, Y) and (Y, X).

17. Result: Empty

    Hint: Use self join persons x persons and WHERE clause to captures pairs with same date of birth. Supress self pairs and symetric pairs as above.

18. Result: Empty

    Hint: Similar to Q18 except date criterion is more nuanced. To share the same birth day, individuals' date of birth must agree on their day and month parts but not necessarily on their year parts. To extract the month part from birth_date use strftime("birth_date); use "for the day.birth_date

19. Result: Eight rows,

    Hint: Easy

20. Result: Six rows (AA, CC, BB, EE, FF and DD)

    Hint: This is tricky to do using just joins. Far easier would be to use the EXCEPT device we saw in the subquery lecture i.e. Q1 EXCEPT Q2, where Q1 captures everyone in the table and Q2 just the beer lovers.

    To do this using just joins, form 3way joins persons x likes(1) x likes(2) Impose conditions so all three elements of each triple relate to the same individual. Stipulate that first food in each triple can be anything at all but that the second food in the triple can be anything *apart from beer*. Group by individual. Within each group the foods listed represent (1) all food liked the individual and (2) all non-beer foods liked. If we do a COUNT on first foods and separately on food two on a group by group basis, for non-beer lovers the two counts will be the same, but for the beer lovers the two counts will be different.

21. Result: Two rows (AA and DD)

    Hint: Two-way join persons x likes, linked on person id. Restrict to the three food mentioned in the question using the WHERE clause. This will gives us at most three rows per person. If we group by person and count rows within groups, the groups with a count of 2 or greater are the ones we want. (Uses GROUP BY, HAVING and COUNT).

22. Result: Eight pairs, including AA and DD

    Hint: Use 4way join persons(1) x likes(1) x persons(2) x likes(2) to capture pairs of individual-food couples. Use ON conditions to ensure persons(1) and likes(1) shares the same person id and similarly with persons(2) and likes(2). Ensure likes(1) and likes(2) relate to the same food.

23. Result: 12 rows, including two beer lovers in Cork

    Hint: 2way join persons x likes aligned in terms of person id. Group by county and foo (both).

24. Result: One row.

    Hint: 2way join persons x likes, aligned on person id and restricted only to beer. Use GROUP BY to bunch together results by county and ORDER BY to present result in desired order.

25. Result: Barry Barry

    Hint: This is much easier to do using subqueries, but if you like to do things the hard way, use a self join persons(1) x persons(2). Impose conditions so that the pairs (X, Y) we retain are such that the X is no older than Y i.e. same date of birth or later. If we group by person is, each group will contains pair of some individual AND those who are no older than him/her. The smallest group (size = 1), will contain one pair (X, X) for the youngest person as there are individuals younger than X. This approach assume that there is a unique youngest person; it breaks down if there is a tie for the youngest person slot.