Some Hints on Labsheet 7

December 1, 2023

Bear in mind that because the data is so sparse, running queries can be relied upon only to validate that queries are syntactically correct, but are less useful in verifying correctness.

1. Easy
2. One name. Easy.
3. One room. Easy.
4. None. Use IS NULL.
5. Six. Easy.
6. Three hotels, one room each.
7. Three hotels.
8. One. Inner query joining hotels and rooms to determine hotels with relevant rooms; outer query performs count.
9. Around 85. Easy.
10. Around 96. Easy.
11. Around 104. As above. Easy.
12. None. Easy.
13. Four rooms. Easy.
14. One hotel, two rooms. Join hotels and rooms; group by hotel num.
15. None. Join hotels bookings and guests. Interpret "booking for month of January" to mean arrival date falling within that month.
16. None. Self join hotels with itself. Remember to filter out symmetric pairs etc.
17. None. Join guests, bookings and hotels. Identify bookings where DATE('now') between arrival date and departure date.

18. None. Five-way join guests-bookings-hotels-bookings-hotels. Impose conditions to ensure all elements relate to same individual. Ensure both bookings relate to same arrival and departure dates. Ensure bookings relate to different hotels and cities.

19. None. Join guests, bookings and hotels. Group by guest and use HAVING to enforce three-stay requirement.

20. Zero. Structure query in form SELECT (Subquery1)/ (Subquery2). Subquery1 returns the number of occupied rooms for HC. Use join of hotels and bookings and identify bookings that straddle New Year's Day. Subquery2 captures the total number of rooms using join of hotels and rooms.

21. Three guests (Murphy, Kelly and Smith). Subquery to capture maximum number of relevant bookings among all guests; outer query to get guest(s) with that number of bookings.

22. One hotel. Similar to above.

23. None. Self join guests with itself. Link on same address. Filter for symmetric pairs.

24. Assume one guest per booking i.e. ignore fact that booking for double room means two guests not one. Apply grouping to bookings based on arrival date and count. Wrap an outer query to find max and the associated date.

25. 19. Apply JulianDay to departure and arrival date of each booking, difference gives the length of the stay (as float). Use MAX to get largest.

26. Subquery to find bookings for specified day (i.e. arrival and departure dates straddle that day). Use an outer query and NOT IN to find rooms in HM not among those booked.