Practical Exam: Hotel Operations

LuxurStay Hotels is a major, international chain of hotels. They offer hotels for both business and leisure travellers in major cities across the world. The chain prides themselves on the level of customer service that they offer.

However, the management has been receiving complaints about slow room service in some hotel branches. As these complaints are impacting the customer satisfaction rates, it has become a serious issue. Recent data shows that customer satisfaction has dropped from the 4.5 rating that they expect.

You are working with the Head of Operations to identify possible causes and hotel branches with the worst problems.

Data

The following schema diagram shows the tables available. You have only been provided with data where customers provided a feedback rating.

Task 1

Before you can start any analysis, you need to confirm that the data is accurate and reflects what you expect to see.

It is known that there are some issues with the `branch` table, and the data team have provided the following data description.

Write a query to return data matching this description. You must match all column names and description criteria.

Column Name	Criteria
id	Nominal. The unique identifier of the hotel. Missing values are not possible due to the database structure.
location	Nominal. The location of the particular hotel. One of four possible values, 'EMEA', 'NA', 'LATAM' and 'APAC'. Missing values should be replaced with “Unknown”.
total_rooms	Discrete. The total number of rooms in the hotel. Must be a positive integer between 1 and 400. Missing values should be replaced with the default number of rooms, 100.
staff_count	Discrete. The number of staff employeed in the hotel service department. Missing values should be replaced with the total_rooms multiplied by 1.5.
opening_date	Discrete. The year in which the hotel opened. This can be any value between 2000 and 2023. Missing values should be replaced with 2023.
target_guests	Nominal. The primary type of guest that is expected to use the hotel. Can be one of 'Leisure' or 'Business'. Missing values should be replaced with 'Leisure'.

Unknown integration DataFrame as `df`

```sql
SELECT
    id,
    COALESCE(location, 'Unknown') AS location,
    CASE
        WHEN total_rooms BETWEEN 1 AND 400 THEN total_rooms
        ELSE 100
    END AS total_rooms,
    CASE
        WHEN staff_count IS NOT NULL THEN staff_count
        ELSE total_rooms * 1.5
    END AS staff_count,
    CASE
        WHEN opening_date = '-' THEN '2023'
        WHEN opening_date BETWEEN '2000' AND '2023' THEN opening_date
        ELSE '2023'
    END AS opening_date,
    CASE
        WHEN target_guests IS NULL THEN 'Leisure'
        WHEN LOWER(target_guests) LIKE 'b%' THEN 'Business'
        ELSE target_guests END AS target_guests
FROM
    branch;
```

index	id	location	total_rooms	staff_count	opening_date	target_guests
0	1	LATAM	168	178	2017	Business
1	2	APAC	154	82	2010	Leisure
2	3	APAC	212	467	2003	Leisure
3	4	APAC	230	387	2023	Business
4	5	APAC	292	293	2002	Business
5	6	NA	260	590	2022	Leisure
6	7	EMEA	259	442	2018	Business
7	8	NA	259	285	2023	Business
8	9	NA	157	274	2001	Business
9	10	EMEA	205	138	2013	Leisure
10	11	EMEA	191	255	2005	Business
11	12	NA	177	248	2012	Business
12	13	EMEA	126	255	2010	Leisure
13	14	EMEA	366	703	2000	Business
14	15	APAC	365	688	2002	Business
15	16	LATAM	228	274	2021	Business

Rows: 100

Task 2

The Head of Operations wants to know whether there is a difference in time taken to respond to a customer request in each hotel. They already know that different services take different lengths of time.

Calculate the average and maximum duration for each branch and service. Your output should include the columns `service_id`, `branch_id`, `avg_time_taken` and `max_time_taken`. Values should be rounded to two decimal places where appropriate.

Unknown integration DataFrame as `average_time_service`

```sql
-- Write your query for task 2 in this cell
SELECT service_id,
branch_id,
ROUND(AVG(time_taken),2) AS avg_time_taken,
ROUND(MAX(time_taken),2) AS max_time_taken
FROM request
GROUP BY service_id, branch_id;
```

	s...	b	avg_tim...	max_ti...
0	2	46	13.09	16
1	4	99	9.13	13
2	1	8	2.56	10
3	2	13	13.53	17
4	1	46	2.08	4
5	3	15	6.73	7
6	2	35	13.17	16
7	1	1	2.44	12
8	3	13	6.8	8
9	1	57	2.29	5
10	1	41	2.27	8
11	2	32	13.23	19
12	4	66	9	10
13	1	23	2.46	9
14	3	22	7.15	9
15	3	57	7.81	13

Rows: 385

Task 3

The management team want to target improvements in `Meal` and `Laundry` service in Europe (`EMEA`) and Latin America (`LATAM`).

Write a query to return the `description` of the service, the `id` and `location` of the branch, the id of the request as `request_id` and the `rating` for the services and locations of interest to the management team.

Use the original branch table, not the output of task 1.

Unknown integration DataFrame as `t`

```sql
-- Write your query for task 3 in this cell
SELECT
    s.description,
    b.id AS branch_id,
    b.location,
    r.id AS request_id,
    r.rating
FROM request AS r
JOIN service AS s
    ON s.id = r.service_id
JOIN branch as b
    ON b.id = r.branch_id
WHERE
    description IN ('Meal', 'Laundry') AND
    location IN ('EMEA', 'LATAM');
```

	des…	b…		r…	
0	Laundry	63	EMEA	3	4
1	Laundry	69	LATAM	6	5
2	Meal	44	EMEA	18	4
3	Laundry	57	LATAM	19	3
4	Meal	1	LATAM	21	4
5	Meal	26	LATAM	26	5
6	Laundry	34	EMEA	27	4
7	Laundry	60	LATAM	35	4
8	Meal	21	EMEA	37	4
9	Meal	1	LATAM	38	4
10	Meal	26	LATAM	41	5
11	Laundry	30	EMEA	44	5
12	Meal	21	EMEA	51	4
13	Laundry	69	LATAM	55	5
14	Meal	70	LATAM	63	4
15	Meal	23	EMEA	66	5

Rows: 5,047

Task 4

So that you can take a more detailed look at the lowest performing hotels, you want to get service and branch information where the average rating for the branch and service combination is lower than 4.5 - the target set by management.

Your query should return the `service_id` and `branch_id`, and the average rating (`avg_rating`), rounded to 2 decimal places.

Unknown integration DataFrame as

```sql
-- Write your query for task 4 in this cell
SELECT
    service_id,
    branch_id,
    ROUND(AVG(rating),2) as avg_rating
FROM request
GROUP BY service_id, branch_id
HAVING AVG(rating) < 4.5;
```

	s…	b…	a…
0	2	46	3.78
1	4	99	3.83
2	1	8	3.64
3	1	46	3.81
4	3	15	4
5	2	35	3.76
6	1	1	3.66
7	1	57	3.64
8	1	41	3.77