# 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.
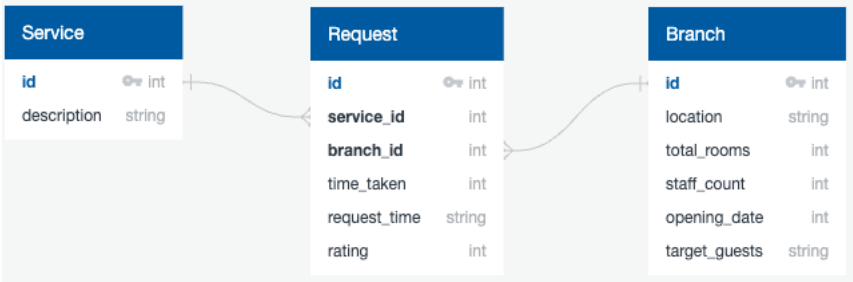
Task 1

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, including identifying and cleaning all invalid values. You must match all column names and description criteria. Your output should be a DataFrame named 'clean_branch_data'.

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 employed 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'.

Certification Hotel Operations   DataFrame as df

```sql
SELECT
    id,

    -- Clean location
    CASE
        WHEN location IN ('EMEA', 'NA', 'LATAM', 'APAC') THEN location
        ELSE 'Unknown'
    END AS location,

    -- Clean total_rooms
    CASE
        WHEN total_rooms::text ~ '^\d+$' AND total_rooms::int BETWEEN 1 AND 400
        THEN total_rooms::int
        ELSE 100
    END AS total_rooms,

    -- Clean staff_count
    CASE
        WHEN staff_count::text ~ '^\d+$' THEN staff_count::int
        ELSE
            (CASE
                WHEN total_rooms::text ~ '^\d+$' AND total_rooms::int BETWEEN 1 AND 400
                THEN total_rooms::int
                ELSE 100
            END) * 1.5
    END AS staff_count,

    -- Clean opening_date
    CASE
        WHEN opening_date::text ~ '^\d+$' AND opening_date::int BETWEEN 2000 AND 2023
        THEN opening_date::int
        ELSE 2023
    END AS opening_date,

    -- Clean target_guests
    CASE
        WHEN target_guests IN ('Leisure', 'Business') THEN target_guests
        ELSE 'Leisure'
    END AS target_guests

FROM branch;
```

ind...	id	locati...	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	Leisure
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	Leisure

Rows: 100   ⤢ Expand

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 be a DataFrame named 'average_time_service'
- It 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.

🦉 Certification Hotel Operations   DataFrame as a

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

⋯	s...	⋯	b.	⋯	avg_tim...	⋯	max_ti...	⋯
0	1	1	2.44	12
1	2	1	12.73	15
2	3	1	7	7
3	1	2	2.14	6
4	2	2	13.46	15
5	1	3	2.28	6
6	2	3	13	15
7	3	3	8	8
8	1	4	2.18	9
9	2	4	13.17	14
10	3	4	6.5	7
11	1	5	2.52	10
12	2	5	13.83	19
13	3	5	7.14	8
14	1	6	2.14	6
15	2	6	13.32	17

Rows: 385   ⤢ Expand

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.

Your output should be a DataFrame named 'target_hotels'.

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

🐘 Certification Hotel Operations   DataFrame as t

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

	↑↓	service_descripti... ··· ↑↓	b. ··· ↑↓	··· ↑↓	r... ··· ↑↓	··· ↑↓	
	0	Laundry	7	EMEA	12290	4	
	1	Laundry	7	EMEA	2462	3	
	2	Laundry	7	EMEA	5870	3	
	3	Laundry	7	EMEA	2921	4	
	4	Laundry	7	EMEA	11967	4	
	5	Laundry	7	EMEA	14033	4	
	6	Laundry	7	EMEA	11878	4	
	7	Laundry	7	EMEA	13711	4	
	8	Laundry	7	EMEA	6710	4	
	9	Laundry	7	EMEA	11311	3	
	10	Laundry	7	EMEA	7015	4	
	11	Laundry	7	EMEA	2945	3	
	12	Laundry	7	EMEA	10607	3	
	13	Laundry	7	EMEA	10508	4	
	14	Laundry	7	EMEA	14364	3	
	15	Laundry	7	EMEA	16856	4	

Rows: 5,047   ↗ Expand

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 output should be a DataFrame named 'average_rating'
- It should return the `service_id` and `branch_id`, and the average rating (`avg_rating`)
- Values should be rounded to 2 decimal places where appropriate.

🐘 Certification Hotel Operations   DataFrame as a

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

···	s… ···	b. ···	a… ···
0	1	1	3.66
1	2	1	3.91
2	3	1	4
3	3	3	4
4	1	4	3.75
5	2	4	3.72
6	3	4	4
7	1	5	3.66
8	2	5	3.56
9	3	5	3.86
10	1	7	3.7
11	2	7	3.64
12	3	7	4
13	1	8	3.64
14	2	8	3.67
15	3	8	3.38

Rows: 215   ↗ Expand

-- Write your query for task 4 in this cell

SELECT
    service_id,
    branch_id,