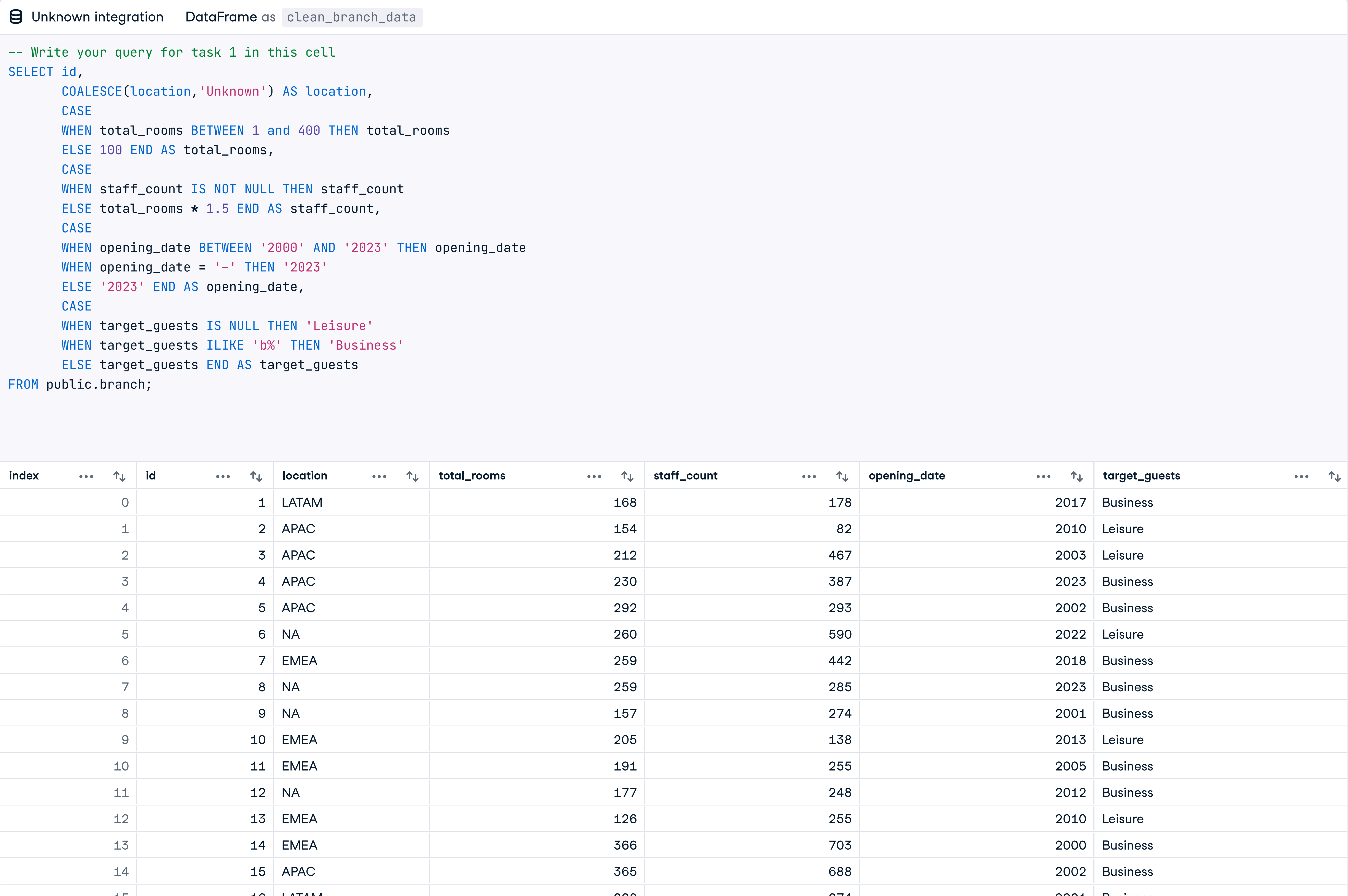
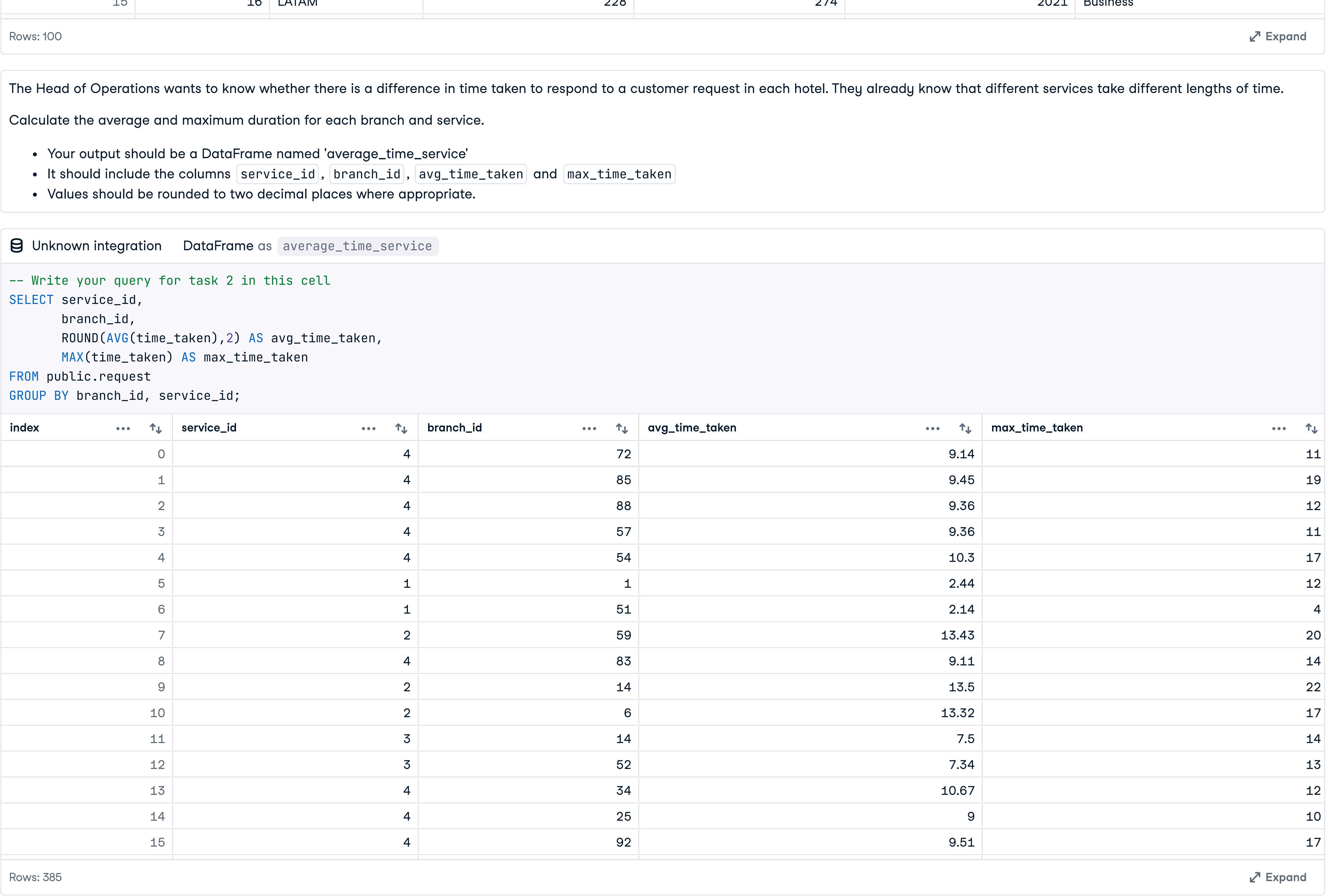


```sql
-- Write your query for task 1 in this cell
SELECT id,
       COALESCE(location,'Unknown') AS location,
       CASE
       WHEN total_rooms BETWEEN 1 and 400 THEN total_rooms
       ELSE 100 END AS total_rooms,
       CASE
       WHEN staff_count IS NOT NULL THEN staff_count
       ELSE total_rooms * 1.5 END AS staff_count,
       CASE
       WHEN opening_date BETWEEN '2000' AND '2023' THEN opening_date
       WHEN opening_date = '-' THEN '2023'
       ELSE '2023' END AS opening_date,
       CASE
       WHEN target_guests IS NULL THEN 'Leisure'
       WHEN target_guests ILIKE 'b%' THEN 'Business'
       ELSE target_guests END AS target_guests
FROM public.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	[illegible]	[illegible]	[illegible]	[illegible]



15	16	LATAM	228	274	2021	Business

Rows: 100

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.

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

index	service_id	branch_id	avg_time_taken	max_time_taken
0	4	72	9.14	11
1	4	85	9.45	19
2	4	88	9.36	12
3	4	57	9.36	11
4	4	54	10.3	17
5	1	1	2.44	12
6	1	51	2.14	4
7	2	59	13.43	20
8	4	83	9.11	14
9	2	14	13.5	22
10	2	6	13.32	17
11	3	14	7.5	14
12	3	52	7.34	13
13	4	34	10.67	12
14	4	25	9	10
15	4	92	9.51	17

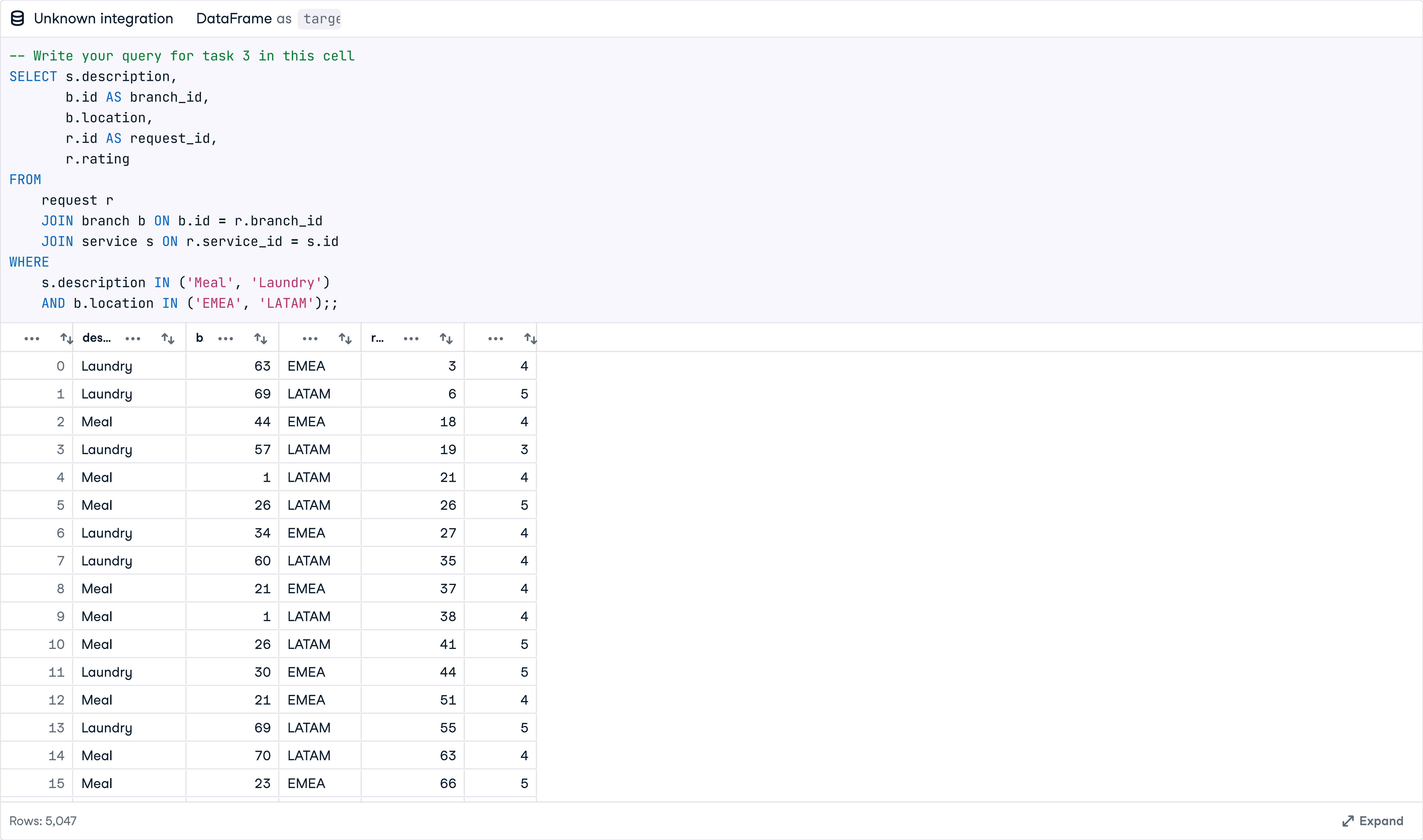
Rows: 385

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.



```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 r
    JOIN branch b ON b.id = r.branch_id
    JOIN service s ON r.service_id = s.id
WHERE
    s.description IN ('Meal', 'Laundry')
    AND b.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

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.

Unknown integration DataFrame as a

```
-- Write your query for task 4 in this cell
SELECT service_id,
       branch_id,
       ROUND(AVG(rating),2) AS avg_rating
FROM public.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
9	3	57	3.53
10	4	64	3.56
11	1	11	3.75
12	1	9	3.68
13	3	53	3.66
14	2	31	3.75
15	1	5	3.66

Rows: 215

Expand