

Data Science Apprenticeship Challenge Section B: Query Building

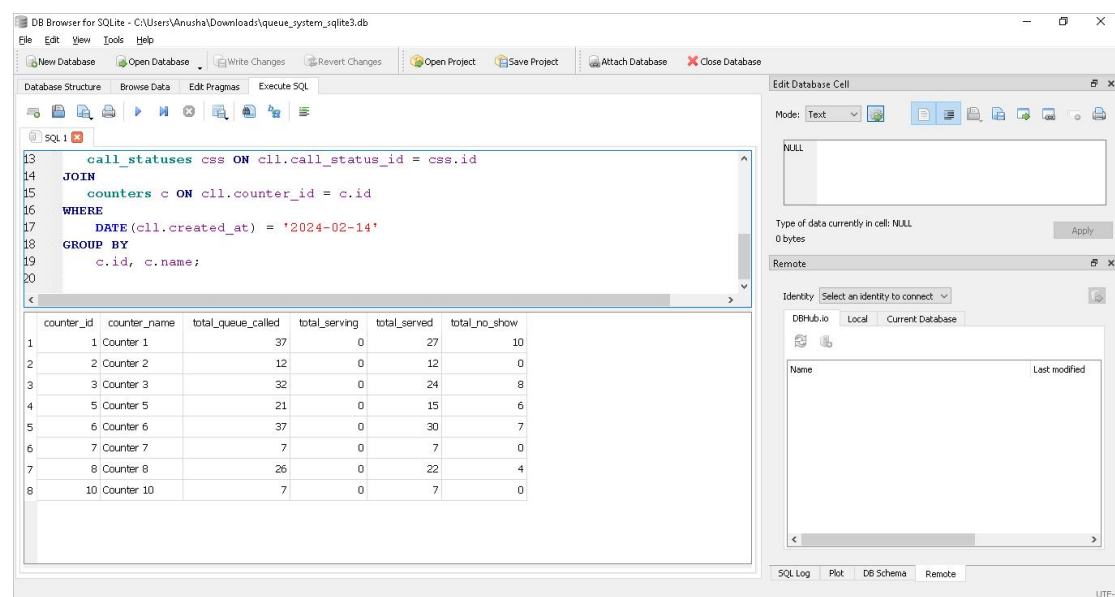
Counter Summary:

Objective: Summarize counter activities for a specific date.

Query:

```
SELECT
    c.id as counter_id,
    c.name as counter_name,
    COUNT(DISTINCT cll.id ) AS total_queue_called,
    SUM(css.name='serving') AS total_serving,
    SUM(css.name='served') AS total_served,
    SUM(css.name='noshow') AS total_no_show
FROM
    calls cll
JOIN
    queues q ON cll.queue_id = q.id
JOIN
    call_statuses css ON cll.call_status_id = css.id
JOIN
    counters c ON cll.counter_id = c.id
WHERE
    DATE(cll.created_at) = '2024-02-14'
GROUP BY
    c.id, c.name;
```

Output:



The screenshot shows the DB Browser for SQLite interface. The SQL editor contains the query from the previous block. The output pane displays the results of the query, which are summarized in the table below.

| counter_id | counter_name | total_queue_called | total_serving | total_served | total_no_show |
|------------|--------------|--------------------|---------------|--------------|---------------|
| 1 | Counter 1 | 37 | 0 | 27 | 10 |
| 2 | Counter 2 | 12 | 0 | 12 | 0 |
| 3 | Counter 3 | 32 | 0 | 24 | 8 |
| 4 | Counter 5 | 21 | 0 | 15 | 6 |
| 5 | Counter 6 | 37 | 0 | 30 | 7 |
| 6 | Counter 7 | 7 | 0 | 7 | 0 |
| 7 | Counter 8 | 26 | 0 | 22 | 4 |
| 8 | Counter 10 | 7 | 0 | 7 | 0 |

Service Summary:

Objective: Summarize service activities for a specific date.

Query:

```
SELECT
    s.id as service_id,
    s.name as service_name,
    COUNT(DISTINCT q.id) AS total_visitors,
    COUNT(DISTINCT CASE WHEN q.called = 0 THEN q.id END) AS total_queued,
    COUNT(DISTINCT CASE WHEN q.called = 1 THEN q.id END) AS total_called,
    SUM(css.name= 'serving') AS total_serving,
    SUM(css.name='served') AS total_served,
    SUM(css.name = 'noshow') AS total_no_show
FROM
    queues q
JOIN
    services s ON q.service_id = s.id
LEFT JOIN
    calls cll ON q.id = cll.queue_id
LEFT JOIN
    call_statuses css on cll.call_status_id = css.id
WHERE
    DATE(q.created_at) = '2024-02-14'
GROUP BY
    s.id, s.name
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL editor contains the query from the previous block. The output is displayed in a table with 8 columns: service_id, service_name, total_visitors, total_queued, total_called, total_serving, total_served, and total_no_show. The results are for three services: WIS (W), NEW REGISTRATION (NR), and TRAINING (T), all for the date 2024-02-14.

| service_id | service_name | total_visitors | total_queued | total_called | total_serving | total_served | total_no_show |
|------------|-----------------------|----------------|--------------|--------------|---------------|--------------|---------------|
| 1 | WIS (W) | 87 | 0 | 87 | 0 | 65 | 22 |
| 2 | NEW REGISTRATION (NR) | 79 | 0 | 79 | 0 | 68 | 11 |
| 3 | TRAINING (T) | 13 | 0 | 13 | 0 | 11 | 2 |

Service X Counter Summary:

Objective: Summarize service and counter activities for a specific date.

Query:

```
SELECT
  s.id as service_id,
  s.name as service_name,
  c.id as counter_id,
  c.name as counter_name,
  COUNT(DISTINCT q.id) AS total_visitors,
  COUNT(DISTINCT cl.id) AS total_queue_called,
  SUM(CASE WHEN css.name='serving' THEN 1 ELSE 0 END) AS total_serving,
  SUM(CASE WHEN css.name='served' THEN 1 ELSE 0 END) AS total_served,
  SUM(CASE WHEN css.name='noshow' THEN 1 ELSE 0 END) AS total_no_show
FROM
  queues q
JOIN
  services s on q.service_id= s.id
JOIN
  calls cl on q.id = cl.queue_id
JOIN
  counters c on cl.counter_id = c.id
JOIN
  call_statuses css on cl.call_status_id = css.id
WHERE
  DATE(q.created_at) = '2024-02-14'
GROUP BY
  s.id, s.name, c.id, c.name
ORDER BY
  s.id, c.id;
```

Output:

DB Browser for SQLite - C:\Users\Anusha\Downloads\queue_system_sqlite3.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

adddd11.sql

1
2
3

SELECT
s.id as service_id,
s.name as service_name,

| | service_id | service_name | counter_id | counter_name | total_visitors | total_queue_called | total_serving | total_served | total_no_show |
|----|------------|-----------------------|------------|--------------|----------------|--------------------|---------------|--------------|---------------|
| 1 | 1 | WIS (W) | 1 | Counter 1 | 35 | 35 | 0 | 26 | 9 |
| 2 | 1 | WIS (W) | 2 | Counter 2 | 4 | 4 | 0 | 4 | 0 |
| 3 | 1 | WIS (W) | 3 | Counter 3 | 15 | 15 | 0 | 9 | 6 |
| 4 | 1 | WIS (W) | 5 | Counter 5 | 19 | 19 | 0 | 13 | 6 |
| 5 | 1 | WIS (W) | 6 | Counter 6 | 10 | 10 | 0 | 10 | 0 |
| 6 | 1 | WIS (W) | 8 | Counter 8 | 4 | 4 | 0 | 3 | 1 |
| 7 | 2 | NEW REGISTRATION (NR) | 1 | Counter 1 | 2 | 2 | 0 | 1 | 1 |
| 8 | 2 | NEW REGISTRATION (NR) | 2 | Counter 2 | 8 | 8 | 0 | 8 | 0 |
| 9 | 2 | NEW REGISTRATION (NR) | 3 | Counter 3 | 4 | 4 | 0 | 4 | 0 |
| 10 | 2 | NEW REGISTRATION (NR) | 5 | Counter 5 | 2 | 2 | 0 | 2 | 0 |
| 11 | 2 | NEW REGISTRATION (NR) | 6 | Counter 6 | 27 | 27 | 0 | 20 | 7 |
| 12 | 2 | NEW REGISTRATION (NR) | 7 | Counter 7 | 7 | 7 | 0 | 7 | 0 |
| 13 | 2 | NEW REGISTRATION (NR) | 8 | Counter 8 | 22 | 22 | 0 | 19 | 3 |
| 14 | 2 | NEW REGISTRATION (NR) | 10 | Counter 10 | 7 | 7 | 0 | 7 | 0 |
| 15 | 3 | TRAINING (T) | 3 | Counter 3 | 13 | 13 | 0 | 11 | 2 |

Edit Database Cell

Mode: Text

1 1

Type of data currently in cell: Text / Numeric
1 character

Apply

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name

SQL Log Plot DB Schema Remote

UTF-8

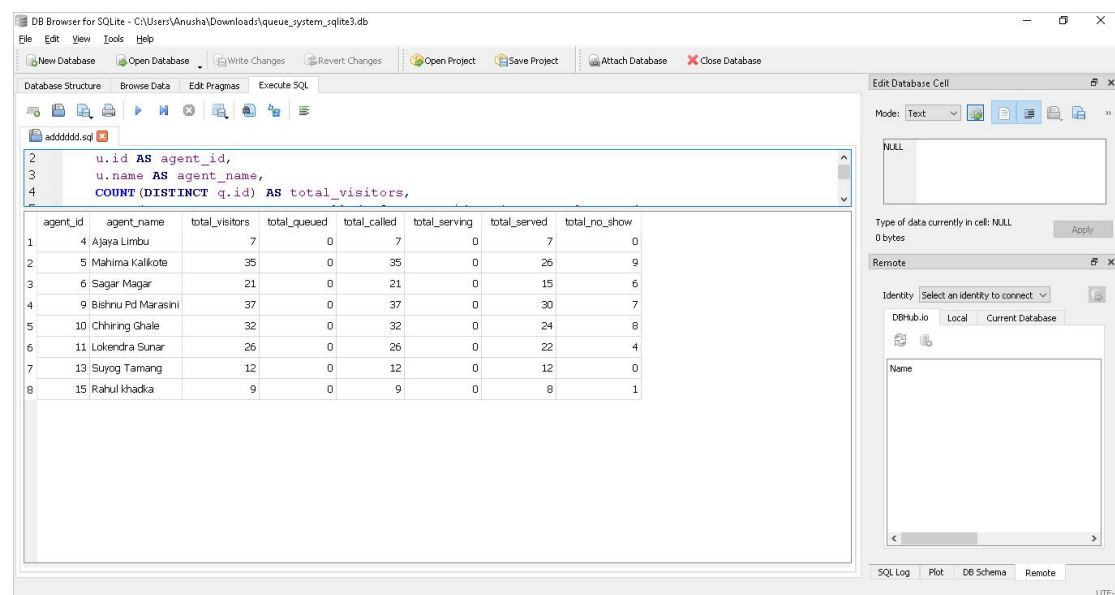
Agent Summary:

Objective: Summarize the activity of an agents for a specific date.

Query:

```
SELECT
    u.id AS agent_id,
    u.name AS agent_name,
    COUNT(DISTINCT q.id) AS total_visitors,
    COUNT(DISTINCT CASE WHEN q.called = 0 THEN q.id END) AS total_queued,
    COUNT(DISTINCT CASE WHEN q.called = 1 THEN q.id END) AS total_called,
    SUM(CASE WHEN css.name = 'serving' THEN 1 ELSE 0 END) AS total_serving,
    SUM(CASE WHEN css.name = 'served' THEN 1 ELSE 0 END) AS total_served,
    SUM(CASE WHEN css.name = 'noshow' THEN 1 ELSE 0 END) AS total_no_show
FROM
    queues q
JOIN
    calls cll ON q.id = cll.queue_id
JOIN
    call_statuses css ON cll.call_status_id = css.id
JOIN users u ON cll.user_id = u.id
WHERE
    DATE(q.created_at) = '2024-02-14'
GROUP BY
    u.id, u.name
ORDER BY
    u.id;
```

Output:



The screenshot shows the DB Browser for SQLite interface. The query is entered in the SQL editor, and the results are displayed in a table. The table has 8 columns: agent_id, agent_name, total_visitors, total_queued, total_called, total_serving, total_served, and total_no_show. The results are as follows:

| agent_id | agent_name | total_visitors | total_queued | total_called | total_serving | total_served | total_no_show |
|----------|--------------------|----------------|--------------|--------------|---------------|--------------|---------------|
| 4 | Ajaya Limbu | 7 | 0 | 7 | 0 | 7 | 0 |
| 5 | Mahima Kalkote | 35 | 0 | 35 | 0 | 26 | 9 |
| 6 | Sagar Magar | 21 | 0 | 21 | 0 | 15 | 6 |
| 9 | Bishnu Pd Marasini | 37 | 0 | 37 | 0 | 30 | 7 |
| 10 | Chhiring Ghale | 32 | 0 | 32 | 0 | 24 | 8 |
| 11 | Lokendra Sunar | 26 | 0 | 26 | 0 | 22 | 4 |
| 13 | Suryog Tamang | 12 | 0 | 12 | 0 | 12 | 0 |
| 15 | Rahul Khadka | 9 | 0 | 9 | 0 | 8 | 1 |

