

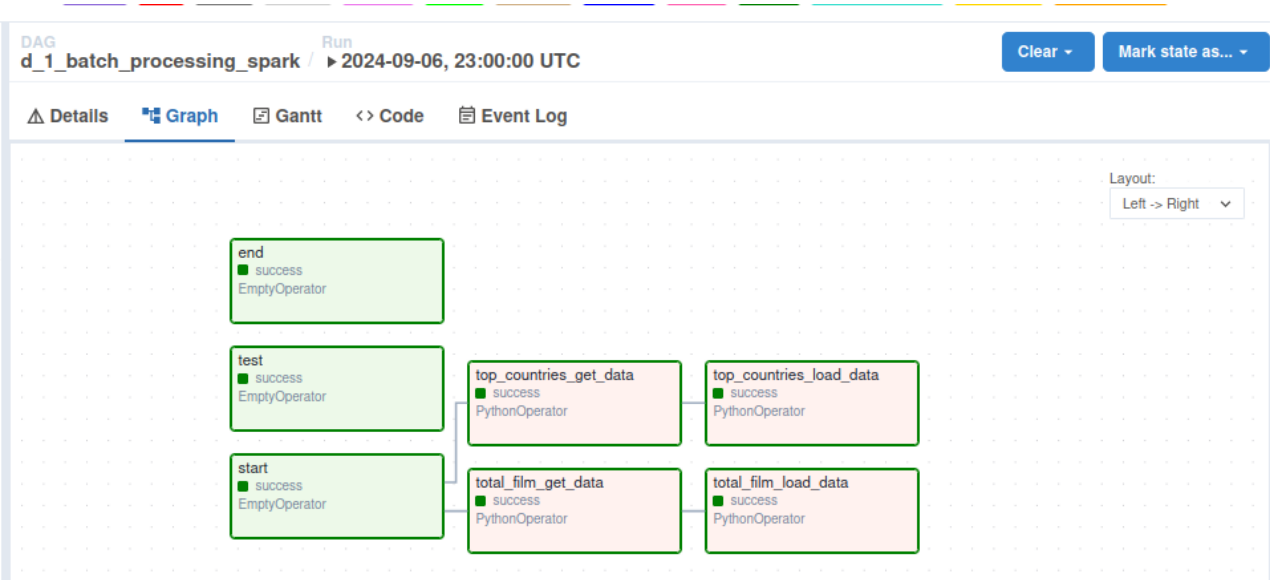
https://github.com/triutama133/project3_Data_Engineer.git

Create Table di TIDB

The screenshot shows the DBeaver interface with a table named 'top_country_trianka' selected. The table has the following columns: index, country, total, and date. The data is as follows:

index	country	total	date
1	Chad	1	2024-09-05
2	Anguilla	1	2024-09-05
3	Paraguay	3	2024-09-05
4	Yemen	4	2024-09-05
5	Senegal	1	2024-09-05
6	Sweden	1	2024-09-05
7	Philippines	20	2024-09-05
8	Tonga	3	2024-09-05
9	Malaysia	15	2024-09-05
10	Turkey	3	2024-09-05
11	Malawi	1	2024-09-05
12	Iraq	1	2024-09-05
13	Germany	7	2024-09-05
14	Afghanistan	1	2024-09-05
15	Cambodia	2	2024-09-05
16	Sudan	2	2024-09-05
17	France	4	2024-09-05
18	Greece	2	2024-09-05
19	Holy See (Vatican City State)	1	2024-09-05
20	Sri Lanka	1	2024-09-05
21	Taiwan	10	2024-09-05
22	Algeria	3	2024-09-05
23	Slovakia	1	2024-09-05
24	Argentina	13	2024-09-05
25	Angola	2	2024-09-05
26	Ecuador	3	2024-09-05
27	Congo, The Democratic Republic of the	2	2024-09-05

Bagan di Airflow Berfungsi



Data Service di TIDB

The screenshot shows the TiDB Cloud Data Service console. On the left, there's a sidebar with 'Data Service BETA' and a list of apps: 'GET /top_country', 'GET /hello_world', 'New App', and 'fortune-500-WyH1'. The main area displays a GET request to 'Cluster0' with the following SQL code:

```
1 /* Getting Started:
2 Enter "USE (database);" before entering your SQL statements.
3 Type "--your question" + Enter to try out AI-generated SQL queries
4 Declare a parameter like "Where id = ${arg}".
5 */
6 SELECT * From test.top_country_trianka;
```

The 'Result' tab shows the HTTP response as a table:

index	country	total	date
0	Chad	1	2024-09-08
1	Anguilla	1	2024-09-08
2	Paraguay	3	2024-09-08
3	Yemen	4	2024-09-08
4	Senegal	1	2024-09-08
5	Sweden	1	2024-09-08

The 'Properties' panel on the right shows the 'Basic' tab with the 'Path' as '/test/top_country' and the 'Endpoint URL' as 'https://ap-southeast-1.data.tidbcloud.com/api/v1beta/app/dataapp-OGXtAVuN/endpoint/test/top_country'. The 'Request Method' is set to 'GET'. The 'Advanced' tab shows a note about request quotas: 'Request quotas are set at 100 requests per minute (rpm) per API key, with a customizable maximum of 1000 rpm. Increase in quota beyond 1000 rpm, please contact us.'

API di Postman

The screenshot shows the Postman API client interface. The 'My Workspace' section on the left lists several collections. The main area displays a GET request to 'https://ap-southeast-1.data.tidbcloud.com/api/v1beta/app/dataapp-OGXtAVuN/endpoint/test/top_country'. The 'Body' tab shows the JSON response:

```
1 {
2   "type": "sql_endpoint",
3   "data": {
4     "columns": [
5       {
6         "col": "index",
7         "data_type": "BIGINT",
8         "nullable": true
9       },
10      {
11        "col": "country",
12        "data_type": "TEXT",
13        "nullable": true
14      },
15      {
16        "col": "total",
17        "data_type": "BIGINT",
18        "nullable": true
19      },
20    ]
21  }
22 }
```

The status bar at the bottom indicates a '200 OK' response with a response time of 1944 ms and a size of 69.3 KB.