**DATA ENGINEER BATCH 4**
Mentee: Yovina Silvia
Mentor: Raja Fathurrahman

**TASK 2 - Airflow**

1. Create a DAG that contain several tasks:
   - A task that predict multiple names from gender-api with SimpleHTTPOperator, refer to this API documentation.
   - A task to create table in postgresql. Suppose the prediction result returned a json data like below.


```
[
  {
    "input": {
      "first_name": "sandra",
      "country": "US"
    },
    "details": {
      "credits_used": 1,
      "duration": "13ms",
      "samples": 9273,
      "country": "US",
      "first_name_sanitized": "sandra"
    },
    "result_found": true,
    "first_name": "Sandra",
    "probability": 0.98,
    "gender": "female"
  }
]
```


2. Create a table with columns: input, details, result_found, first_name, probability, gender and timestamp. Timestamp refers to the time where the data is loaded to Postgre. Please define the appropriate data type for each column.
3. A task that will load all the prediction result to table gender_name_prediction postgresql with PostgresHook. Duplication data are allowed.

- Mengimpor modul dan operator yang diperlukan untuk membuat DAG Airflow.
- definisikan DAG dengan nama `yovina-airflow-task2`, deskripsi, jadwal interval (setiap 5 jam), tanggal mulai, dan pengaturan `catchup`.

```
from airflow import DAG
from airflow.operators.python import PythonOperator
from airflow.providers.http.operators.http import SimpleHttpOperator
from airflow.providers.postgres.hooks.postgres import PostgresHook
from airflow.providers.postgres.operators.postgres import PostgresOperator
from airflow.utils.dates import days_ago
import json

dag = DAG(
        'yovina-airflow-task2',
        description='Hello, Ini DAG Task 2 Yovina Silvia',
        schedule_interval='0 */5 * * *',
        start_date=datetime(2023, 10, 18),
        catchup=False
)
```

- Tugas ini menggunakan `SimpleHttpOperator` untuk mengirim permintaan POST ke API gender-api dengan data berupa nama-nama yang akan diprediksi. Hasil respon dari API difilter menjadi objek JSON.

```
predict_names_task = SimpleHttpOperator(
    task_id='profile_from_gender',
    method='POST',
    http_conn_id='gender_api',
    endpoint='/gender/by-first-name-multiple',
    headers={"Content-Type": "application/json"},
    data=json.dumps([
  {
    "first_name": "Yovina",
```

```
    "country": "ID"
  },
  {
    "first_name": "Dimaz",
    "country": "ID"
  }
]),
    response_filter=lambda response: json.loads(response.text),
    log_response=True,
    dag=dag,
)
```

- Tugas ini menggunakan `PostgresOperator` untuk membuat tabel di PostgreSQL jika tabel tersebut belum ada. Tabel ini memiliki kolom-kolom yang sesuai dengan hasil JSON dari API.

```
create_table_task = PostgresOperator(
    task_id='create_table_to_postgres',
    postgres_conn_id='pg_conn_yovina',
    sql="""
    CREATE TABLE IF NOT EXISTS yovina_gender_name_prediction (
        input JSONB,
        details JSONB,
        result_found BOOLEAN,
        first_name VARCHAR(50),
        probability FLOAT,
        gender VARCHAR(10),
        timestamp TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
    );
    """,
    retries=3,
```

```
    retry_delay=timedelta(minutes=5),
    dag=dag,
    autocommit=True,
)
```

- Fungsi ini digunakan untuk mengambil hasil prediksi dari tugas sebelumnya (`profile_from_gender`) dan memasukkannya ke dalam tabel PostgreSQL menggunakan `PostgresHook`. Data dimasukkan ke tabel dengan format yang sesuai.

```
def load_predictions_to_postgres(**kwargs):
    ti = kwargs['ti']
    predictions = ti.xcom_pull(task_ids='profile_from_gender')
    pg_hook = PostgresHook(postgres_conn_id='pg_conn_yovina')
    for prediction in predictions:
        input_data = json.dumps(prediction['input'])
        details_data = json.dumps(prediction['details'])
        result_found = prediction['result_found']
        first_name = prediction['first_name']
        probability = prediction['probability']
        gender = prediction['gender']
        pg_hook.run("""
            INSERT INTO yovina_gender_name_prediction (input, details, result_found, first_name, probability, gender)
            VALUES (%s, %s, %s, %s, %s, %s);
        """, parameters=(input_data, details_data, result_found, first_name, probability, gender))
```

- Tugas ini menggunakan `PythonOperator` untuk menjalankan fungsi `load_predictions_to_postgres`, yang memuat data prediksi ke PostgreSQL.

```
load_predictions_task = PythonOperator(
    task_id='profile_gender_to_postgres',
    python_callable=load_predictions_to_postgres,
    provide_context=True,
    dag=dag,
)
```

- Bagian ini mengatur urutan eksekusi tugas: `predict_names_task` akan dijalankan terlebih dahulu, kemudian `create_table_task`, dan terakhir `load_predictions_task`.

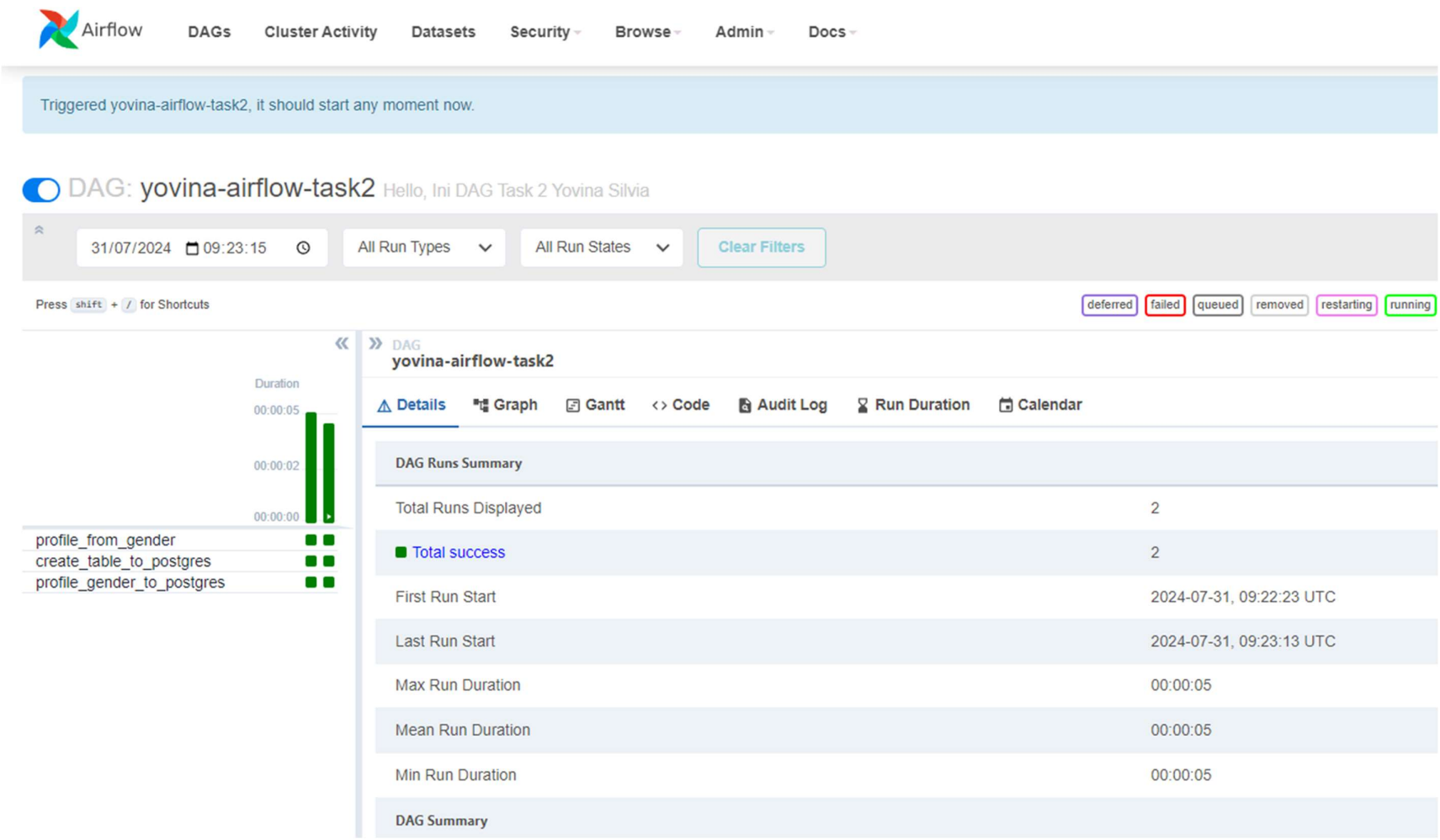
```
predict_names_task >> create_table_task >> load_predictions_task
```

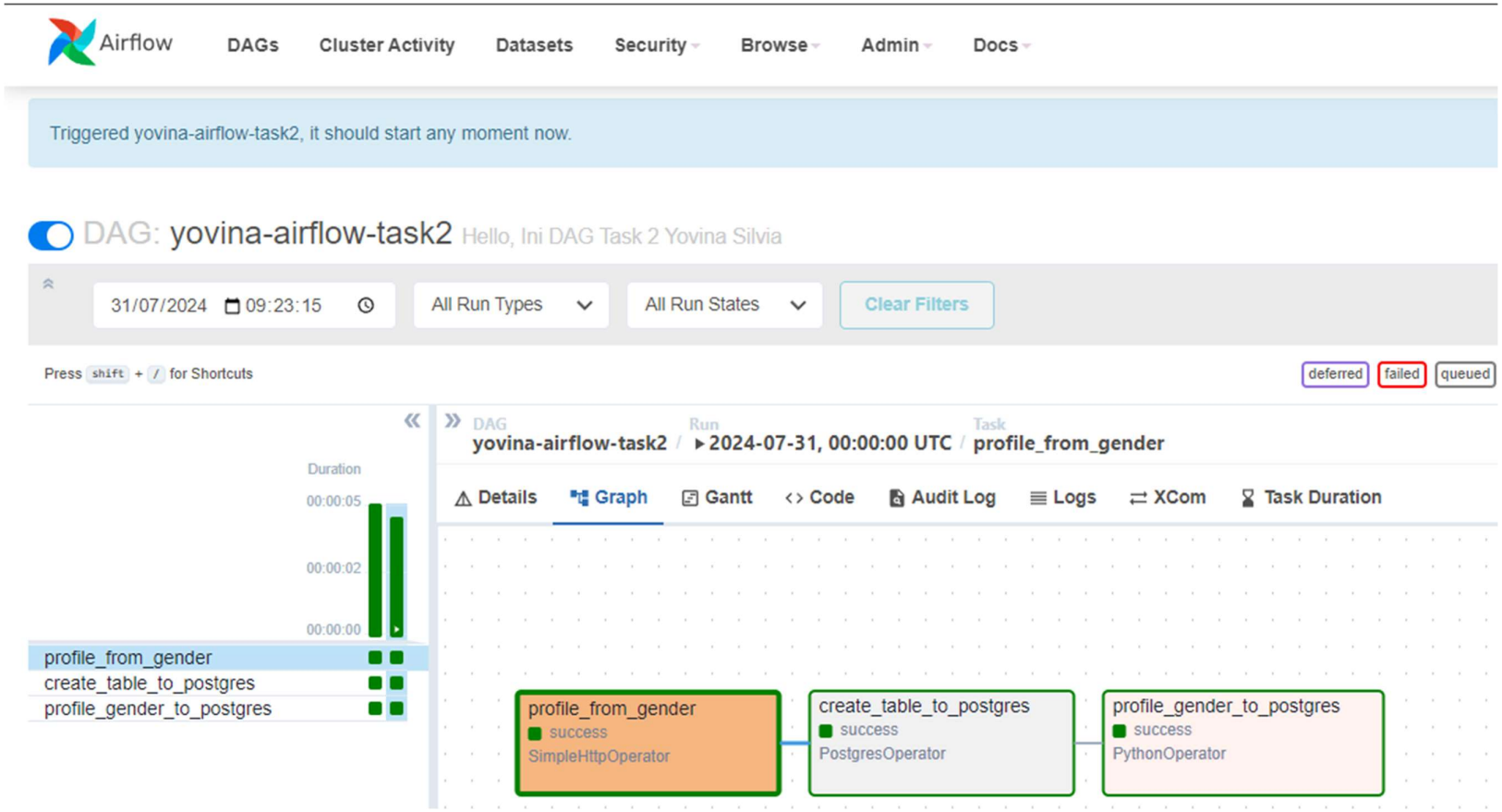
Dengan demikian, DAG ini melakukan prediksi gender dari beberapa nama menggunakan API, membuat tabel di PostgreSQL untuk menyimpan hasil prediksi, dan memuat data prediksi ke tabel tersebut.

- Tampilan pada saat memasukkan codingannya dengan prompt nano:

```
  GNU nano 7.2                                                    yovina_airflow_task1.py
from datetime import datetime
from airflow import DAG
from airflow.operators.empty import EmptyOperator
from airflow.operators.python_operator import PythonOperator

# 1. Create DAG that run in every 5 hours.
dag = DAG(
    'yovina-airflow-task1',
    description='Airflow Task 1',
    schedule_interval='0 */5 * * *',
    start_date=datetime(2023, 10, 18),
    catchup=False
)

start = EmptyOperator(
    task_id='start',
    dag=dag,
)
# ti = task instance
# 2. Suppose we define a new task that push a variable to xcom.
def push_variable_to_xcom(ti=None):
    ti.xcom_push(key='job_role', value='Backend Engineer')
    ti.xcom_push(key='job_role_1', value='Data Engineer')
    ti.xcom_push(key='job_role_2', value='Frontend Engineer')
    ti.xcom_push(key='job_role_3', value='Quality Assurance')

# 3. How to pull multiple values at once?
def pull_multiple_value_once(ti=None):
    job_role = ti.xcom_pull(task_ids='push_var_job_role', key='job_role')
    job_role_1 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_1')
    job_role_2 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_2')
    job_role_3 = ti.xcom_pull(task_ids='push_var_job_role', key='job_role_3')

    print(f'print job_role variable from xcom: {job_role}, {job_role_1}, {job_role_2}, {job_role_3}')

push_variable_to_xcom = PythonOperator(
    task_id = 'push_variable_to_xcom',
    python_callable = push_variable_to_xcom
)

pull_multiple_value_once = PythonOperator(
    task_id = 'pull_multiple_value_once',
    python_callable = pull_multiple_value_once
)

start >> push_variable_to_xcom >> pull_multiple_value_once
```

- Tampilan pada airflownya setelah di masukkan dari terminal:




Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
Triggered yovina-airflow-task2, it should start any moment now.
DAG: yovina-airflow-task2
Hello, Ini DAG Task 2 Yovina Silvia
31/07/2024 09:23:15
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
deferred
failed
queued
DAG
yovina-airflow-task2
Run
2024-07-31, 00:00:00 UTC
Task
profile_from_gender
Duration
00:00:05
00:00:02
00:00:00
profile_from_gender
create_table_to_postgres
profile_gender_to_postgres
Details
Graph
Gantt
Code
Audit Log
Logs
XCom
Task Duration
profile_from_gender
success
SimpleHttpOperator
create_table_to_postgres
success
PostgresOperator
profile_gender_to_postgres
success
PythonOperator

Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs

Triggered yovina-airflow-task2, it should start any moment now.

DAG: yovina-airflow-task2 Hello, Ini DAG Task 2 Yovina Silvia

31/07/2024 09:23:15 All Run Types All Run States Clear Filters

Press shift + / for Shortcuts

deferred failed queued

Duration
00:00:05
00:00:02
00:00:00

profile_from_gender
create_table_to_postgres
profile_gender_to_postgres

DAG yovina-airflow-task2 / Run 2024-07-31, 00:00:00 UTC / Task profile_from_gender

Details Graph Gantt Code Audit Log Logs XCom Task Duration

Parsed at: 2024-07-31, 09:23:05 UTC

```python
from datetime import datetime, timedelta
from airflow import DAG
from airflow.operators.python import PythonOperator
from airflow.providers.http.operators.http import SimpleHttpOperator
from airflow.providers.postgres.hooks.postgres import PostgresHook
from airflow.providers.postgres.operators.postgres import PostgresOperator
from airflow.utils.dates import days_ago
import json


dag = DAG(
        'yovina-airflow-task2',
        description='Hello, Ini DAG Task 2 Yovina Silvia',
        schedule_interval='0 */5 * * *',
        start_date=datetime(2023, 10, 18),
        catchup=False
)
predict_names_task = SimpleHttpOperator(
    task_id='profile_from_gender',
    method='POST',
    http_conn_id='gender_api',
    endpoint='/gender/by-first-name-multiple',
    headers={"Content-Type": "application/json"},
```

Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs

Triggered yovina-airflow-task2, it should start any moment now.

DAG: yovina-airflow-task2 Hello, Ini DAG Task 2 Yovina Silvia

31/07/2024 09:23:15 All Run Types All Run States Clear Filters

Press shift + / for Shortcuts deferred failed

Duration
00:00:05
00:00:02
00:00:00

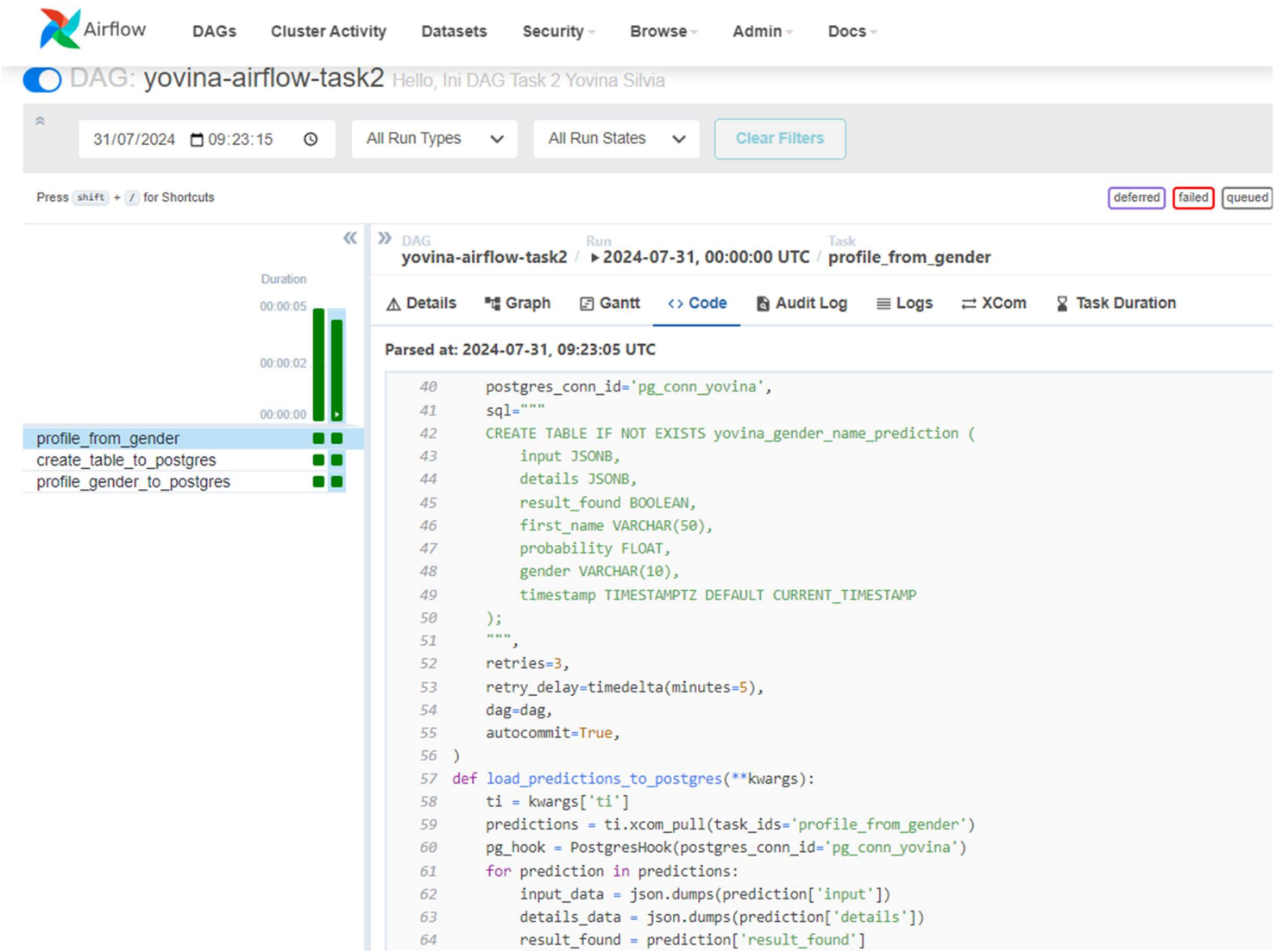
profile_from_gender
create_table_to_postgres
profile_gender_to_postgres

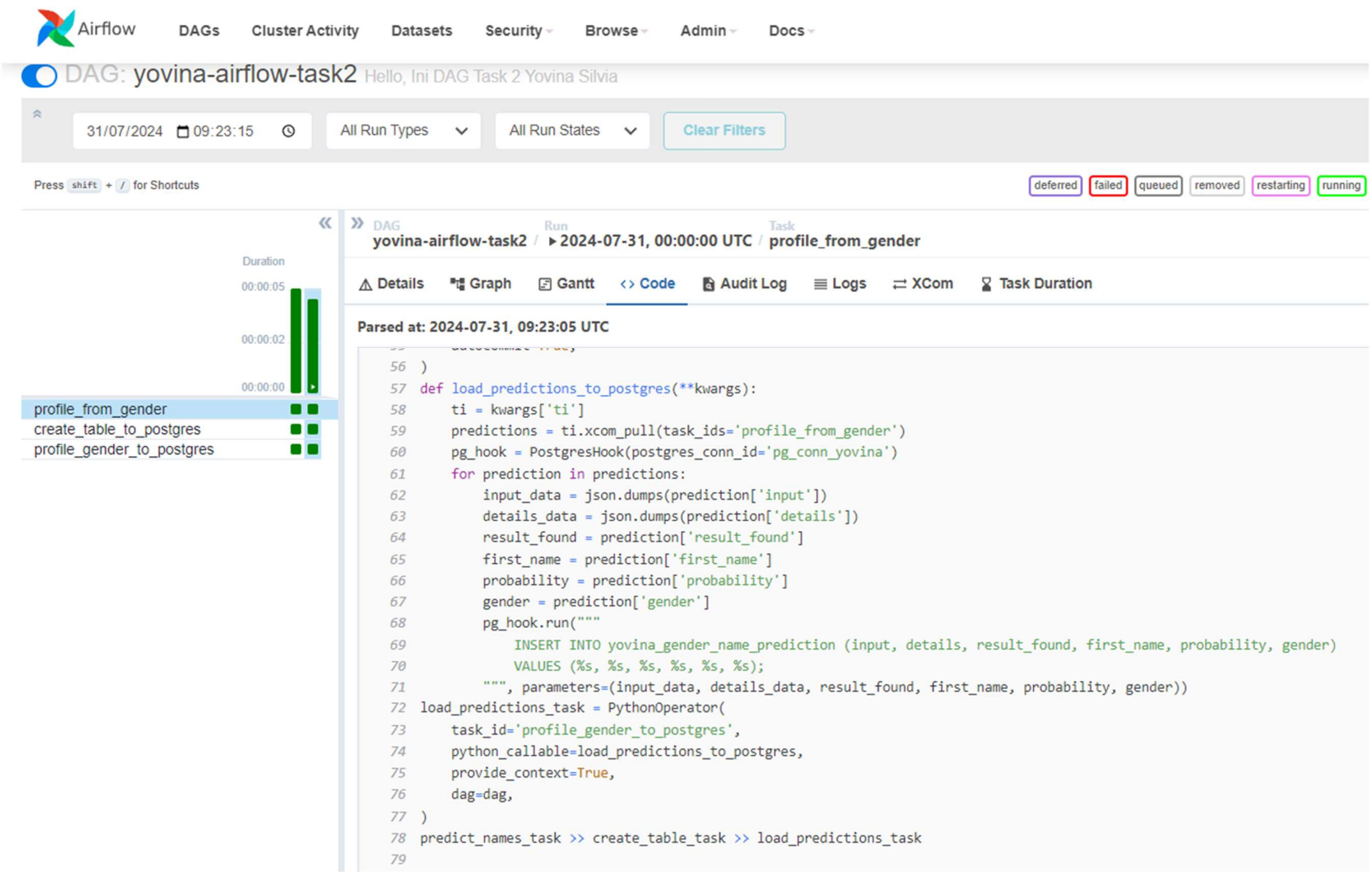
DAG yovina-airflow-task2 / Run 2024-07-31, 00:00:00 UTC / Task profile_from_gender

Details Graph Gantt Code Audit Log Logs XCom Task Duration

Parsed at: 2024-07-31, 09:23:05 UTC

```
    method='POST',
    http_conn_id='gender_api',
    endpoint='/gender/by-first-name-multiple',
    headers={"Content-Type": "application/json"},
    data=json.dumps([
  {
    "first_name": "Yovina",
    "country": "ID"
  },
  {
    "first_name": "Dimaz",
    "country": "ID"
  }
]),
    response_filter=lambda response: json.loads(response.text),
    log_response=True,
    dag=dag,
)
create_table_task = PostgresOperator(
    task_id='create_table_to_postgres',
    postgres_conn_id='pg_conn_yovina',
    sql="""
    CREATE TABLE IF NOT EXISTS yovina_gender_name_prediction (
```


Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
DAG: yovina-airflow-task2 Hello, Ini DAG Task 2 Yovina Silvia
31/07/2024 09:23:15
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
deferred
failed
queued
Duration
00:00:05
00:00:02
00:00:00
profile_from_gender
create_table_to_postgres
profile_gender_to_postgres
DAG
yovina-airflow-task2
Run
2024-07-31, 00:00:00 UTC
Task
profile_from_gender
Details
Graph
Gantt
Code
Audit Log
Logs
XCom
Task Duration
Parsed at: 2024-07-31, 09:23:05 UTC
40 postgres_conn_id='pg_conn_yovina',
41 sql="""
42 CREATE TABLE IF NOT EXISTS yovina_gender_name_prediction (
43 input JSONB,
44 details JSONB,
45 result_found BOOLEAN,
46 first_name VARCHAR(50),
47 probability FLOAT,
48 gender VARCHAR(10),
49 timestamp TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
50 );
51 """,
52 retries=3,
53 retry_delay=timedelta(minutes=5),
54 dag=dag,
55 autocommit=True,
56 )
57 def load_predictions_to_postgres(**kwargs):
58 ti = kwargs['ti']
59 predictions = ti.xcom_pull(task_ids='profile_from_gender')
60 pg_hook = PostgresHook(postgres_conn_id='pg_conn_yovina')
61 for prediction in predictions:
62 input_data = json.dumps(prediction['input'])
63 details_data = json.dumps(prediction['details'])
64 result_found = prediction['result_found']

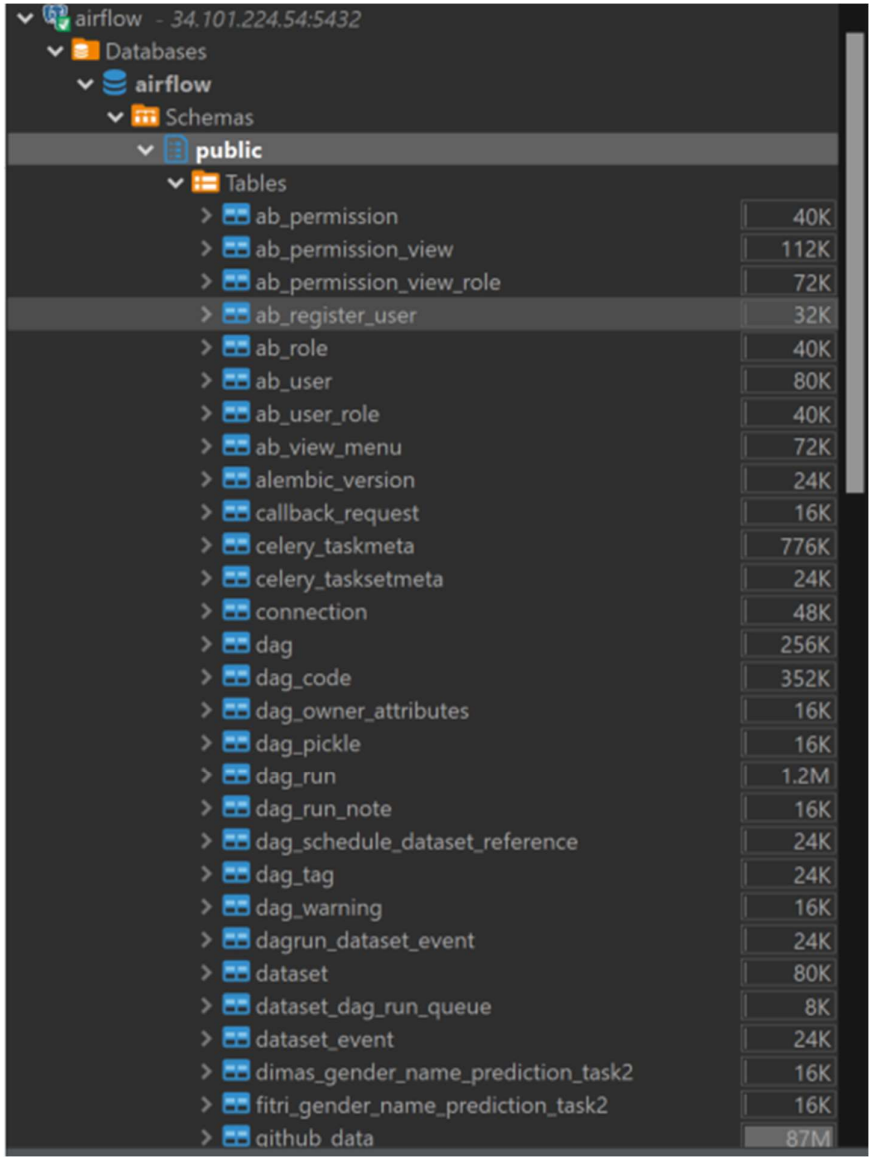

Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
DAG: yovina-airflow-task2 Hello, Ini DAG Task 2 Yovina Silvia
31/07/2024 09:23:15
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
deferred
failed
queued
removed
restarting
running
Duration
00:00:05
00:00:02
00:00:00
profile_from_gender
create_table_to_postgres
profile_gender_to_postgres
DAG
yovina-airflow-task2
Run
2024-07-31, 00:00:00 UTC
Task
profile_from_gender
Details
Graph
Gantt
Code
Audit Log
Logs
XCom
Task Duration
Parsed at: 2024-07-31, 09:23:05 UTC
56 )
57 def load_predictions_to_postgres(**kwargs):
58 ti = kwargs['ti']
59 predictions = ti.xcom_pull(task_ids='profile_from_gender')
60 pg_hook = PostgresHook(postgres_conn_id='pg_conn_yovina')
61 for prediction in predictions:
62 input_data = json.dumps(prediction['input'])
63 details_data = json.dumps(prediction['details'])
64 result_found = prediction['result_found']
65 first_name = prediction['first_name']
66 probability = prediction['probability']
67 gender = prediction['gender']
68 pg_hook.run("""
69 INSERT INTO yovina_gender_name_prediction (input, details, result_found, first_name, probability, gender)
70 VALUES (%s, %s, %s, %s, %s, %s);
71 """, parameters=(input_data, details_data, result_found, first_name, probability, gender))
72 load_predictions_task = PythonOperator(
73 task_id='profile_gender_to_postgres',
74 python_callable=load_predictions_to_postgres,
75 provide_context=True,
76 dag=dag,
77 )
78 predict_names_task >> create_table_task >> load_predictions_task
79

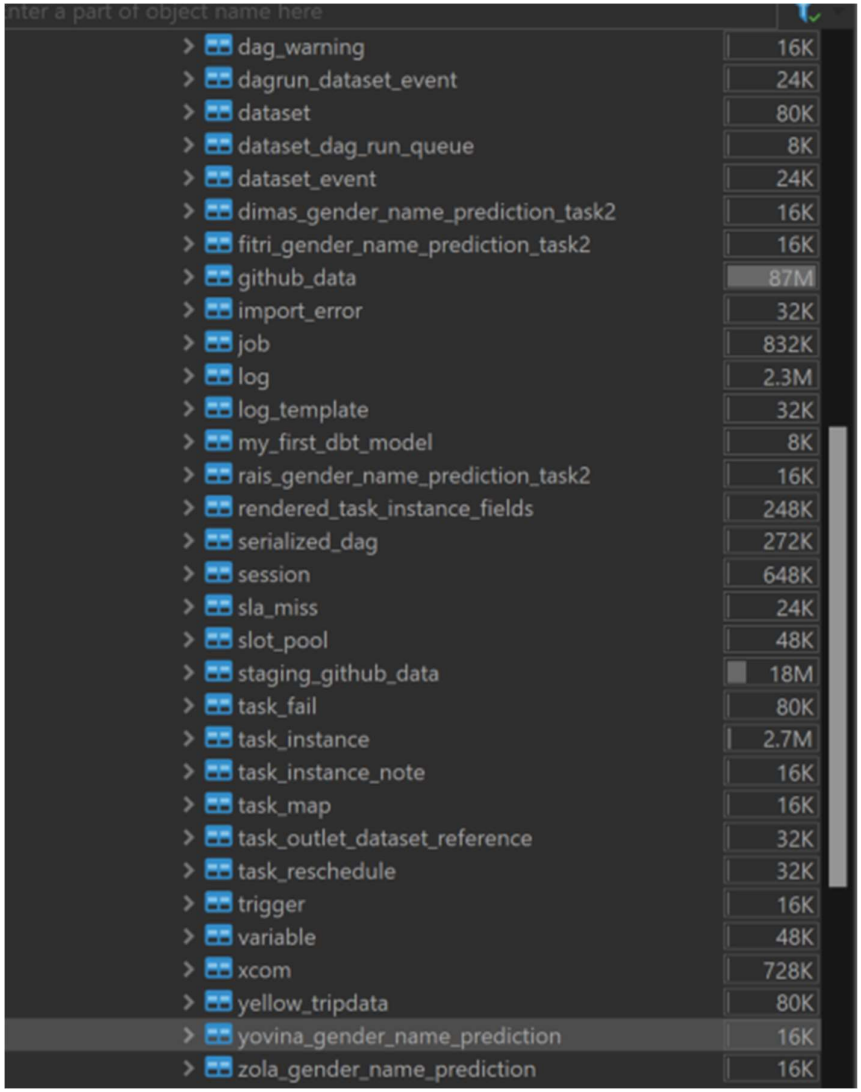
- Hasilnya pada airflow:

```
4b400f82801f
*** Found local files:
***   * /opt/airflow/logs/dag_id=yovina-airflow-task2/run_id=manual__2024-07-31T09:23:12.868591+00:00/task_id=profile_from_gender/attempt=1.log
[2024-07-31, 09:23:13 UTC] {local_task_job_runner.py:120} ▶ Pre task execution logs
[2024-07-31, 09:23:14 UTC] {http.py:173} INFO - Calling HTTP method
[2024-07-31, 09:23:14 UTC] {base.py:84} INFO - Using connection ID 'gender_api' for task execution.
[2024-07-31, 09:23:14 UTC] {base.py:84} INFO - Using connection ID 'gender_api' for task execution.
[2024-07-31, 09:23:15 UTC] {http.py:222} INFO - [
    {
        "input": {
            "first_name": "Yovina",
            "country": "ID"
        },
        "details": {
            "credits_used": 1,
            "duration": "15ms",
            "samples": 4,
            "country": "ID",
            "first_name_sanitized": "yovina"
        },
        "result_found": true,
        "first_name": "Yovina",
        "probability": 1,
        "gender": "female"
    },
    {
        "input": {
            "first_name": "Dimaz",
            "country": "ID"
```

```
            "duration": "15ms",
            "samples": 4,
            "country": "ID",
            "first_name_sanitized": "yovina"
        },
        "result_found": true,
        "first_name": "Yovina",
        "probability": 1,
        "gender": "female"
    },
    {
        "input": {
            "first_name": "Dimaz",
            "country": "ID"
        },
        "details": {
            "credits_used": 1,
            "duration": "16ms",
            "samples": 210,
            "country": "ID",
            "first_name_sanitized": "dimaz"
        },
        "result_found": true,
        "first_name": "Dimaz",
        "probability": 1,
        "gender": "male"
    }
]
[2024-07-31, 09:23:15 UTC] {taskinstance.py:441} ▶ Post task execution logs
```

- Tampilan pada postgresnya:





| | |
|---|---|
| dag_warning | 16K |
| dagrun_dataset_event | 24K |
| dataset | 80K |
| dataset_dag_run_queue | 8K |
| dataset_event | 24K |
| dimas_gender_name_prediction_task2 | 16K |
| fitri_gender_name_prediction_task2 | 16K |
| github_data | 87M |
| import_error | 32K |
| job | 832K |
| log | 2.3M |
| log_template | 32K |
| my_first_dbt_model | 8K |
| rais_gender_name_prediction_task2 | 16K |
| rendered_task_instance_fields | 248K |
| serialized_dag | 272K |
| session | 648K |
| sla_miss | 24K |
| slot_pool | 48K |
| staging_github_data | 18M |
| task_fail | 80K |
| task_instance | 2.7M |
| task_instance_note | 16K |
| task_map | 16K |
| task_outlet_dataset_reference | 32K |
| task_reschedule | 32K |
| trigger | 16K |
| variable | 48K |
| xcom | 728K |
| yellow_tripdata | 80K |
| yovina_gender_name_prediction | 16K |
| zola_gender_name_prediction | 16K |

- Table gender_name_prediction

DBeaver Sample Database (SQLite) | yellow_tripdata | yovina_gender_name_prediction

Properties | Data | ER Diagram

airflow Databases airflow Schemas public Tables yovina_gender_name_prediction

yovina_gender_name_prediction | Enter a SQL expression to filter results (use Ctrl+Space)

| | input | details | result_found | first_name | probability | gender | timestamp |
|---|---|---|---|---|---|---|---|
| 1 | {"country": "ID", "first_name": "Yovina"} | {"country": "ID", "samples": 4, "duration": "12ms", "credits_used": 1, "first_name_sanitiz | [v] | Yovina | 1 | female | 2024-07-31 16:22:28.605 +0700 |
| 2 | {"country": "ID", "first_name": "Dimaz"} | {"country": "ID", "samples": 210, "duration": "13ms", "credits_used": 1, "first_name_sani | [v] | Dimaz | 1 | male | 2024-07-31 16:22:28.626 +0700 |
| 3 | {"country": "ID", "first_name": "Yovina"} | {"country": "ID", "samples": 4, "duration": "15ms", "credits_used": 1, "first_name_sanitiz | [v] | Yovina | 1 | female | 2024-07-31 16:23:18.211 +0700 |
| 4 | {"country": "ID", "first_name": "Dimaz"} | {"country": "ID", "samples": 210, "duration": "16ms", "credits_used": 1, "first_name_sani | [v] | Dimaz | 1 | male | 2024-07-31 16:23:18.231 +0700 |