FINAL PROJECT

MUH SANVI FAHRUDIN

LATAR BELAKANG

DALAM PROYEK KALI INI, ANDA DIHARAPKAN MAMPU UNTUK MENDESIGN SEBUAH DATAWAREHOUSE

MENGGUNAKAN STAR SCHEMA DARI DATA YANG DISIAPKAN. DARI DATA YANG ADA, ANALISA PENGARUH CURAH

HUJAN DAN SUHU TERHADAP REVIEW / RATING DI RESTORAN.

SUMBER DATA

1. DATA CLIMATE BERUPA PRECIPITATION DALAM BENTUK CSV
2. DATA CLIMATE BERUPA TEMPERATURE DALAM BENTUK CSV
3. DATA BUSINESS RESTORAN DALAM BENTUK JSON
4. DATA CHECKIN RESTORAN DALAM BENTUK JSON
5. DATA REVIEW RESTORAN DALAM BENTUK JSON
6. DATA TIP RESTORAN DALAM BENTUK JSON
7. DATA USER RESTORAN DALAM BENTUK JSON

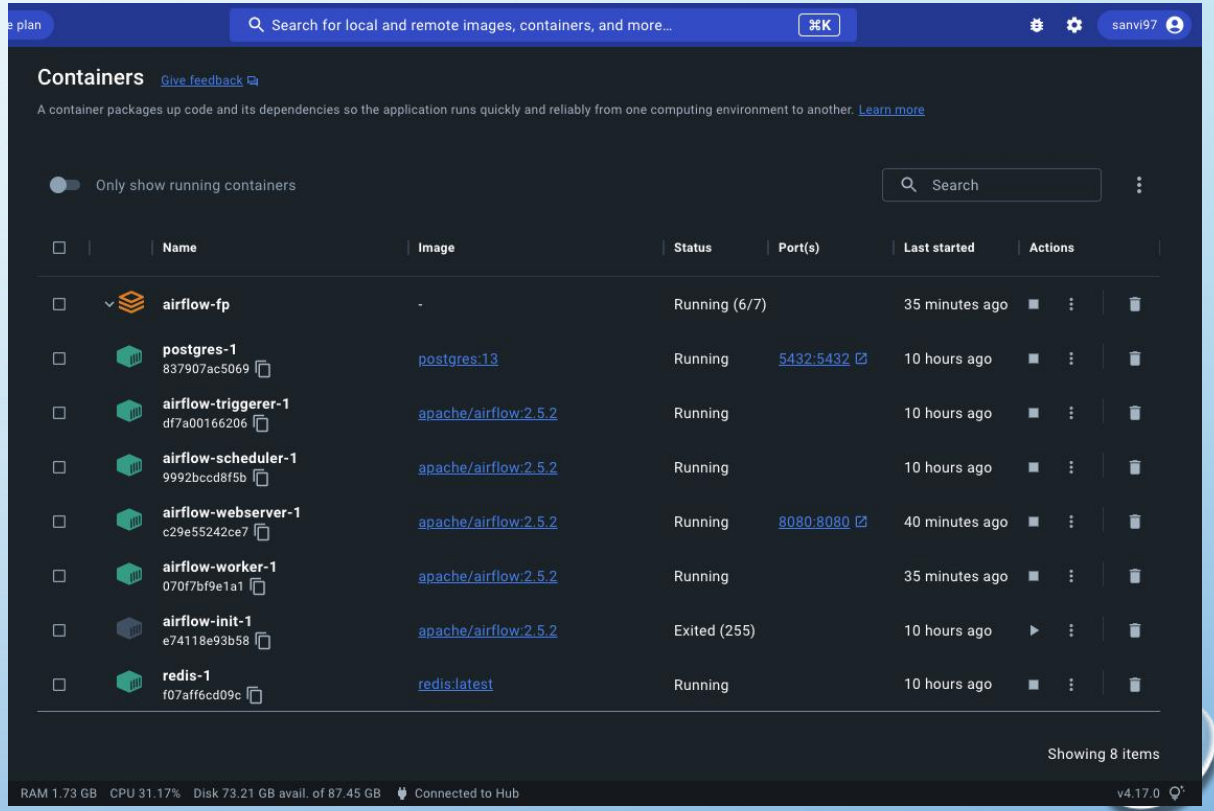
LANGKAH PENGERJAAN

1) BUATLAH DOCKER COMPOSE (AIRFLOW, POSTGRESQL) DI LOCAL COMPUTER ANDA.
2) BUATLAH DATA ARCHITECTURE DIAGRAM YANG DAPAT MENGGAMBARKAN BAGAIMANA CARA ANDA MENGEKSTRACT RAW DATA JSON & CSV KEDALAM DATABASE POSTGRESQL.
3) CASE DIATAS, BUATKAN DATAFLOW DIAGRAM YANG MENGGAMBARKAN DATAWAREHOUSE LAYER.
4) GAMBARKAN / BUATKAN ER DIAGRAM (ENTITY RELATIONSHIP) YANG MENGGAMBARKAN HUBUNGAN ANTARA TABLE – TABLE YANG DI INGEST KE MYSQL (ODS LAYER)
5) DARI RAW DATA TERSEBUT, BUAT DATA MODELING MENGGUNAKAN KIMBAL STAR SCHEMA. TENTUKAN MANA YANG MERUPAKAN FACT TABLE, MANA YANG MERUPAKAN DIM TABLE DAN GAMBARKAN DIAGRAMNYA.
6) BUATLAH SATU DAG DI AIRFLOW DENGAN TASK :
    A. EXTRACT LOAD DATA DARI RAW DATA KE ODS LAYER
    B. EXTRACT DAN LOAD DATA DARI ODS LAYER KE DWH LAYER (DIM & FACT)
    C. TRANSFORM DATA DARI DWH LAYER KE SERVING LAYER
7) PADA SERVING LAYER BUAT SATU TABLE AGREGASI YANG MELAKUKAN JOIN FACT DAN DIM TABLE YELP DATA DAN CLIMATE DATA
8) (QUERY BISA DIJALANKAN SEBAGAI TASK AIRFLOW PADA POIN 6C)
9) BUATLAH SEBUAH ANALISA DARI TABLE AGREGASI PADA POIN 7, UNTUK MENCARI KORELASI ANTARA REVIEW, TIPS DAN CUACA.
10) NILAI TAMBAH JIKA MAMPU MENAMPILKAN HASIL ANALISA KEDALAM BENTUK VISUAL
11) COMMIT CODE YANG DIBUAT KE GITHUB ANDA MASING

1. DOCKER COMPOSE

v-fp > docker-compose.yaml

```yaml
    # for other purpose (development, test and especially production usage) build/
    _PIP_ADDITIONAL_REQUIREMENTS: ${_PIP_ADDITIONAL_REQUIREMENTS:-}
  volumes:
    - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
    - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
    - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
    # - ${AIRFLOW_PROJ_DIR:-.}/sql:/opt/airflow/sql
  user: "${AIRFLOW_UID:-50000}:0"
  depends_on:
    &airflow-common-depends-on
    redis:
      condition: service_healthy
    postgres:
      condition: service_healthy

services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data
    ports:
      - "5432:5432"
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]
      interval: 10s
      retries: 5
      start_period: 5s
    restart: always
```

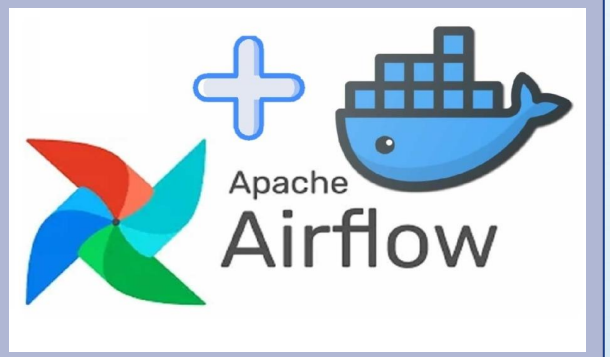


2. ARCHITECTURE EXTRACT RAW DATA TO DATABASE

CLIMATE DATA
–
PRECIPITATIO
N
–
TEMPERATUR

RESTAURANT DATA
–BUSINESS
–CHECKIN
–REVIEW
–TIP
–USER

RAW DATA

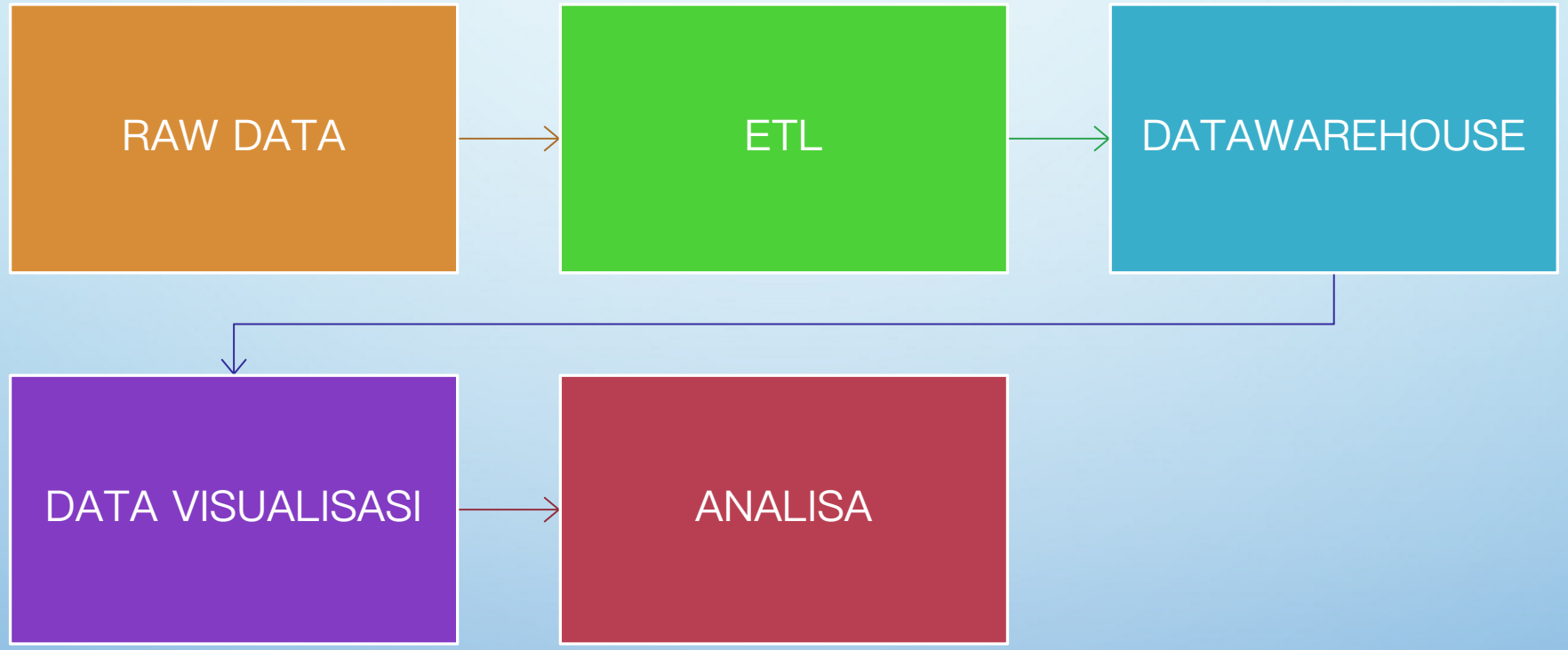


BUILD AIRFLOW FROM DOCKER AND CREATE DAG TO ETL PROCESS

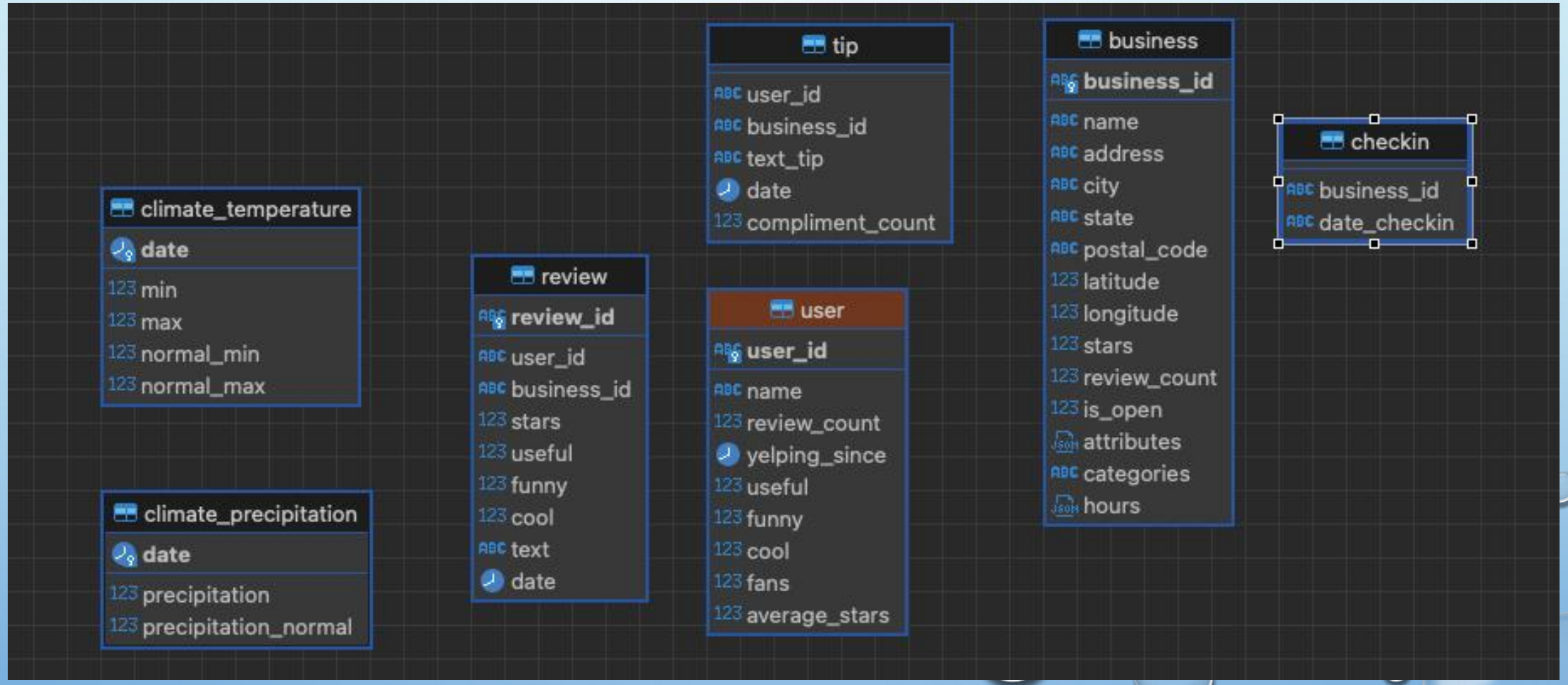
IMPORT PYTHON OPERATOR TO USING PANDAS, SQLALCHEMY, AND PSYCOPG2

LOAD DATA TO DATABASE POSTGRESQL AND OPEN USING DBEAVER

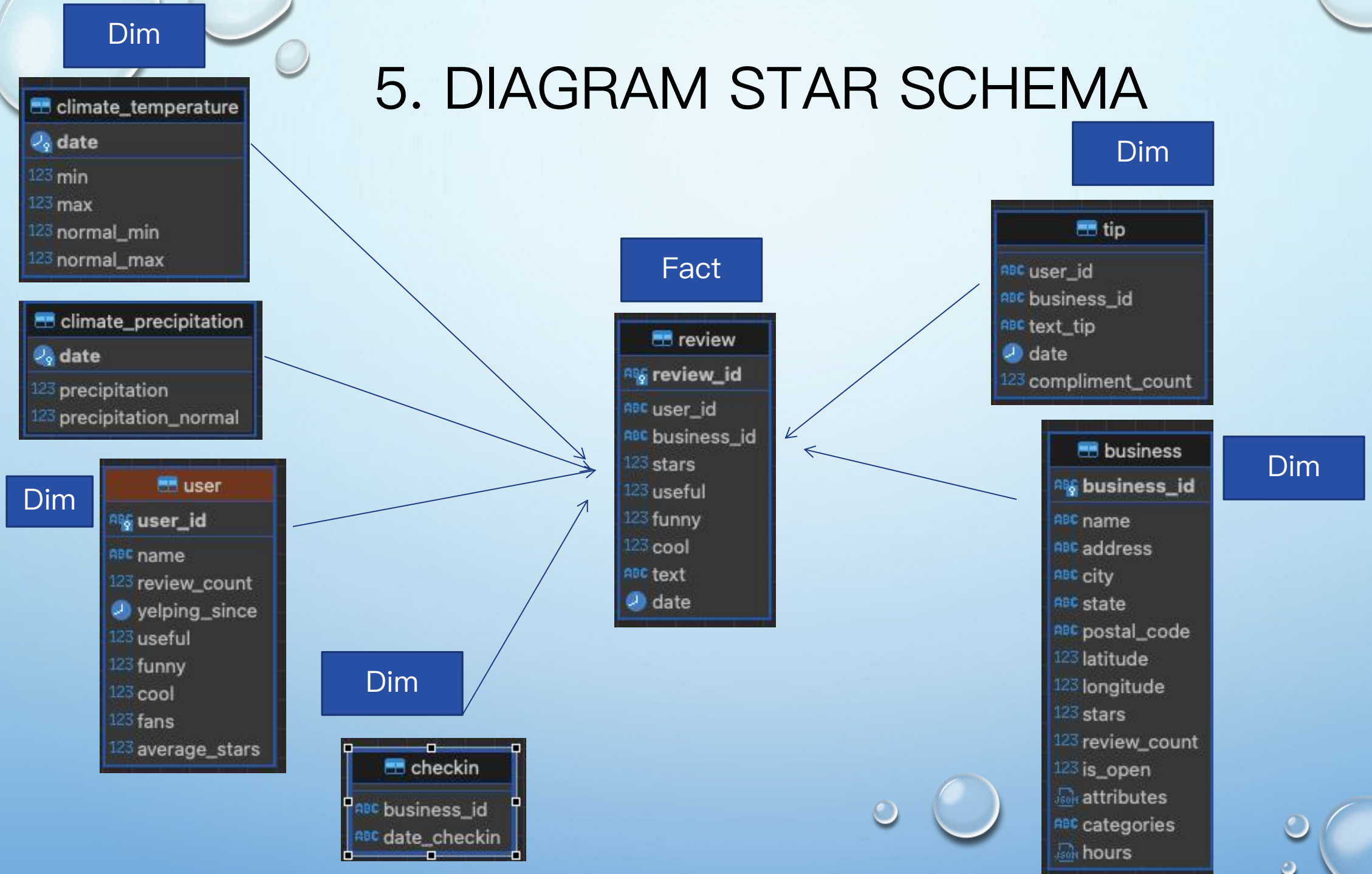
3. DATAWAREHOUSE LAYER



4. ER DIAGRAM (ENTITY RELATIONSHIP)



5. DIAGRAM STAR SCHEMA

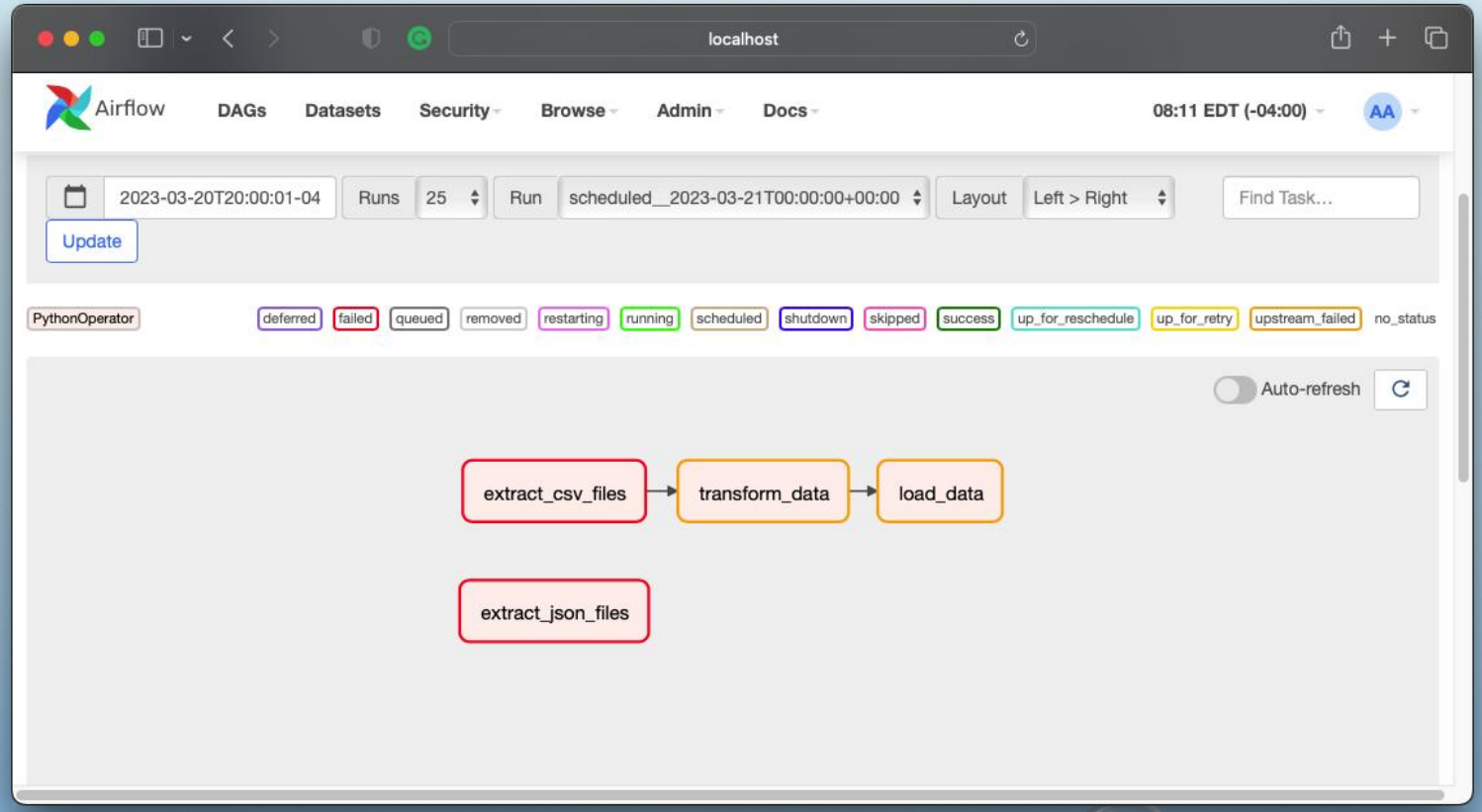


6. DAG AIRFLOW ETL PROCESS

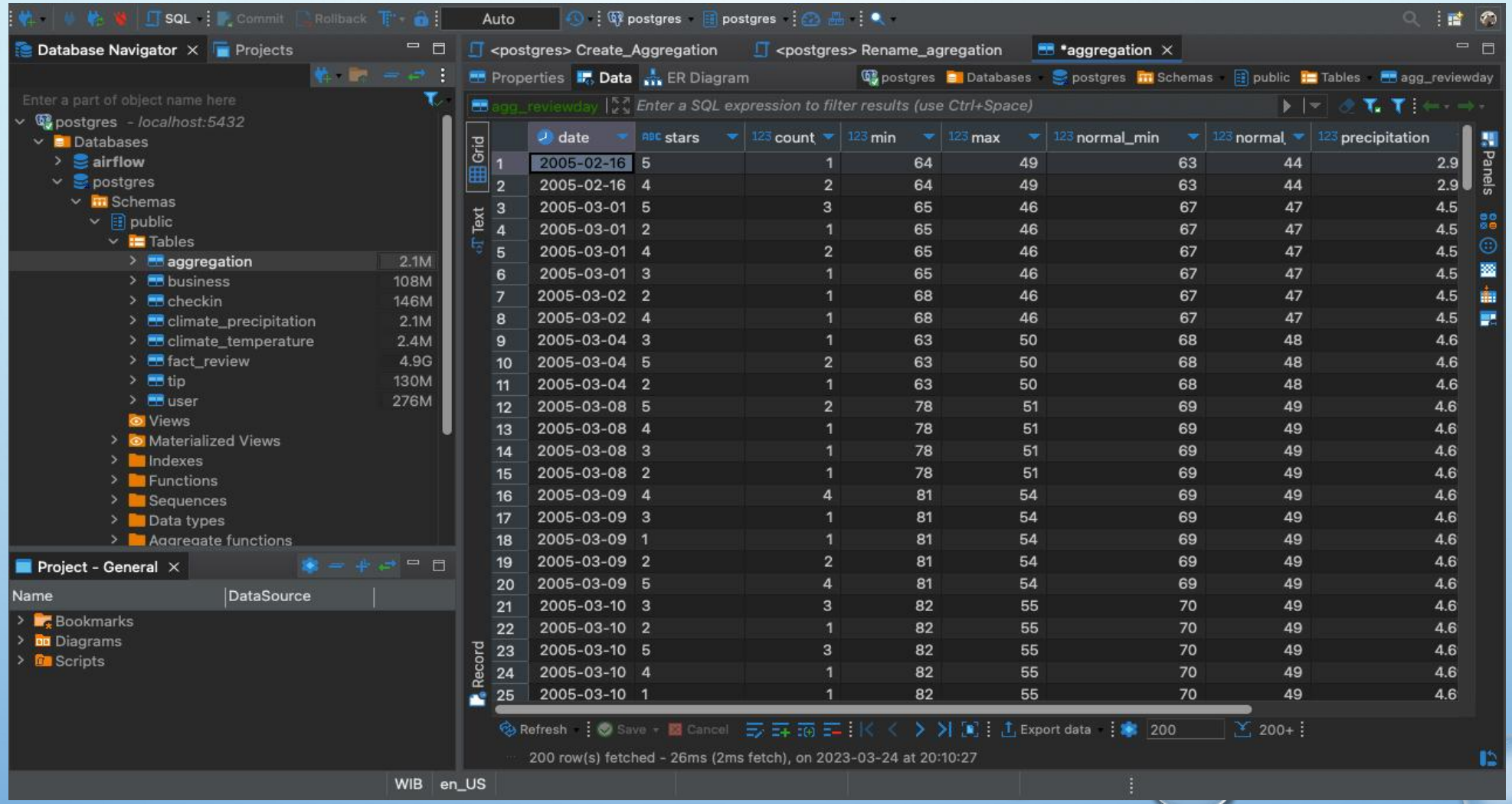
```python
# Import airflow modules
from datetime import datetime, timedelta
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.operators.postgres_operator import PostgresOperator

# Import modules
import pandas as pd
import json
import psycopg2
import os

# Arguments
default_args = {
    'owner': 'Sanvi',
    'depends_on_past': False,
    'start_date': datetime(2023, 3, 21),
    'retries': 1,
    'retry_delay': timedelta(minutes=5)
}

dag = DAG(
    'csv_json_to_postgres',
    default_args=default_args,
    description='Extract CSV and JSON files, transform data, and load into Postgres',
    schedule_interval='@once'
)

# Extract CSV files
cwd = os.getcwd()  # Get the current working directory (cwd)
files = os.listdir(cwd)  # Get all the files in that directory
def extract_csv_files():
    df1 = pd.read_csv('/Users/sanvi/Documents/final project/airflow-fp/dags/data/precipitation.csv')
    df2 = pd.read_csv('/Users/sanvi/Documents/final project/airflow-fp/dags/data/temperature.csv')
```

```python
                , {item['date']}, {item['value']}, {item['val
    # Commit changes and close connection
    conn.commit()
    cur.close()
    conn.close()

# Define DAG tasks
extract_csv_task = PythonOperator(
    task_id='extract_csv_files',
    python_callable=extract_csv_files,
    dag=dag
)

extract_json_task = PythonOperator(
    task_id='extract_json_files',
    python_callable=extract_json_files,
    dag=dag
)

transform_task = PythonOperator(
    task_id='transform_data',
    python_callable=transform_data,
    dag=dag
)

load_task = PythonOperator(
    task_id='load_data',
    python_callable=load_data,
    dag=dag
)

# Set task dependencies
extract_csv_task >> transform_task >> load_task
```

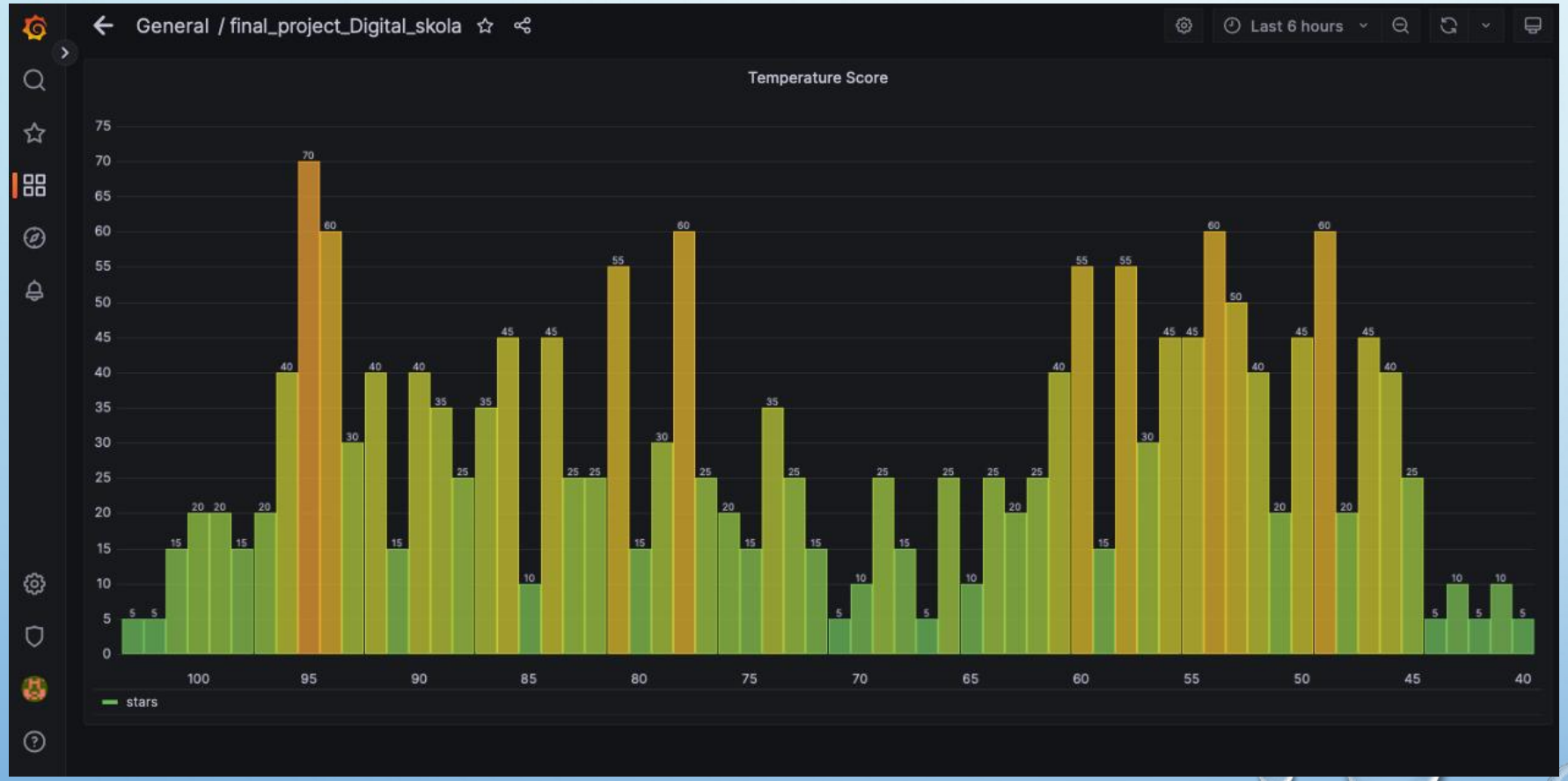
6. DAG AIRFLOW ETL PROCESS



7. SATU TABLE AGREGASI YANG MELAKUKAN JOIN FACT DAN DIM TABLE YELP DATA DAN CLIMATE DATA

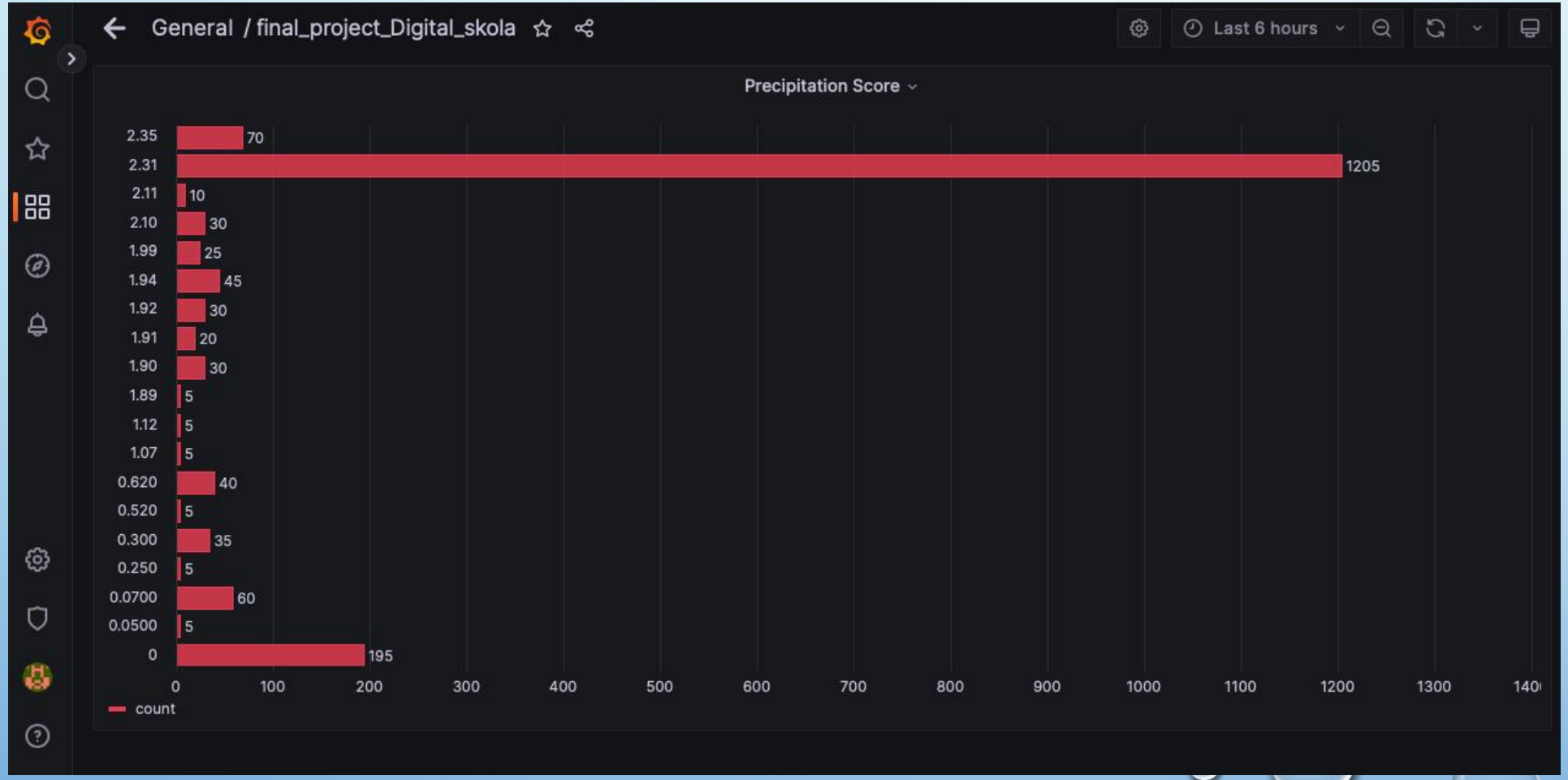


8. KORELASI ANTARA REVIEW, TIPS DAN CUACA



VISUALISASI DATA MENGGUNAKAN TOOLS *GRAFANA.* BAR CAHRT TERSEBUT MENUNJUKKAN KORELASI TOTAL BINTANG(REVIEW) YANG DIDAPAT PADA Y AXIS DAN SUHU PADA X AXIS

8. KORELASI ANTARA REVIEW, TIPS DAN CUACA



VISUALISASI DATA MENGGUNAKAN TOOLS *GRAFANA.* BAR CHART TERSEBUT MENUNJUKKAN KORELASI TOTAL BINTANG(REVIEW) YANG DIDAPAT PADA X AXIS DAN PERCIPITATION PADA Y AXIS

KESIMPULAN

HUBUNGAN ANTARA REVIEW BERUPA TOTAL BINTANG PADA RESTORAN DENGAN SUHU ADALAH PADA SUHU 95° MENDAPATKAN TOTAL BINTANG PALING BANYAK SEBANYAK 70 SEHINGGA BANYAK PENGUNJUNG YANG LEBIH MENYUKAI RESTORAN DENGAN SUHU PANAS

HUBUNGAN ANTARA REVIEW BERUPA TOTAL BINTANG PADA RESTORAN DENGAN PRECIPITATION ADALAH SEMAKIN TINGGI NILAI RECIPITATION, MAKA SEMAKIN DISUKAI OLEH PENGUNJUNG, HAL TERSEBUT DAPAT DILIHAT PADA NILAI PRECIPITATION 2.31 MENDAPATKAN TOTAL BINTANG TERTINGGI YAITU 1205

9. COMMIT CODE YANG DIBUAT KE GITHUB

HTTPS://GITHUB.COM/SANVIF-DOT/FINAL_PROJECT_AIRFLOW_DE