

การวิเคราะห์ปริมาณน้ำฝน ที่เหมาะสมกับการปลูกพืช เศรษฐกิจ

DS525

Data Warehouse and Business Intelligence

สมาชิก

- | | | |
|----|------------------------------|-------------|
| 1. | นางสาวเจนจิรา วงศ์อุคะ | 66199160144 |
| 2. | นายธนกร กอวรพันธุ์ | 66199160152 |
| 3. | นายธราเทพ สุตวิไล | 66199160154 |
| 4. | นางสาวนัยน์ภัค สถาพรพินิจกิจ | 66199160158 |
| 5. | นายนิติชัย ทองคำ | 66199160160 |
| 6. | นายวงศ์กร วงศ์เดชงาม | 66199160177 |

ที่มาและความสำคัญ

ประเทศไทยมีจำนวนครัวเรือนที่เป็นเกษตรกรกว่า ร้อยละ 35 เนื่องจากมีสภาพภูมิอากาศที่เหมาะสมแก่การ เพาะปลูก ไม่ว่าจะเป็นในเรื่องของสภาพดิน แร่ธาตุในดิน ความสูงของพื้นที่ เส้นทางน้ำ อุณหภูมิ และอีกปัจจัยที่สำคัญคือ **ปริมาณน้ำฝน** ซึ่งถือว่าเป็นปัจจัยที่มีความ สำคัญสูงสุด จึงมีความจำเป็นที่จะต้องวิเคราะห์ข้อมูล ปริมาณน้ำฝนที่เหมาะสมกับพืชแต่ละชนิด โดยพืชที่จะนำ มาใช้วิเคราะห์ได้แก่ ข้าว, ข้าวโพดเลี้ยงสัตว์, ปาล์มน้ำมัน, มันสำปะหลัง, ยางพารา และอ้อย ซึ่งเป็นพืชเศรษฐกิจที่ สำคัญของไทย



ปัญหา

1. พืชล้มตายและไม่มีคุณภาพ ส่งผลให้ราคาพืชลดลง
2. ปัญหาหนี้สินของเกษตรกรจากการลงทุนทำการเกษตร ซึ่งอาจไม่ได้ผลผลิตตามที่คาดไว้



วัตถุประสงค์

1. วิเคราะห์ปริมาณน้ำฝนแต่ละจังหวัดที่เหมาะสมกับการเพาะปลูกพืชแต่ละชนิด เพื่อให้ผลผลิตที่ได้มีคุณภาพ และป้องกันพืชล้มตลาด
2. วิเคราะห์ช่วงเวลา (เดือน) ที่เหมาะสมกับการเพาะปลูกพืชแต่ละชนิด เพื่อวางแผนการทำการเกษตรกรรมในปีถัดไป



Data Modeling

Tables

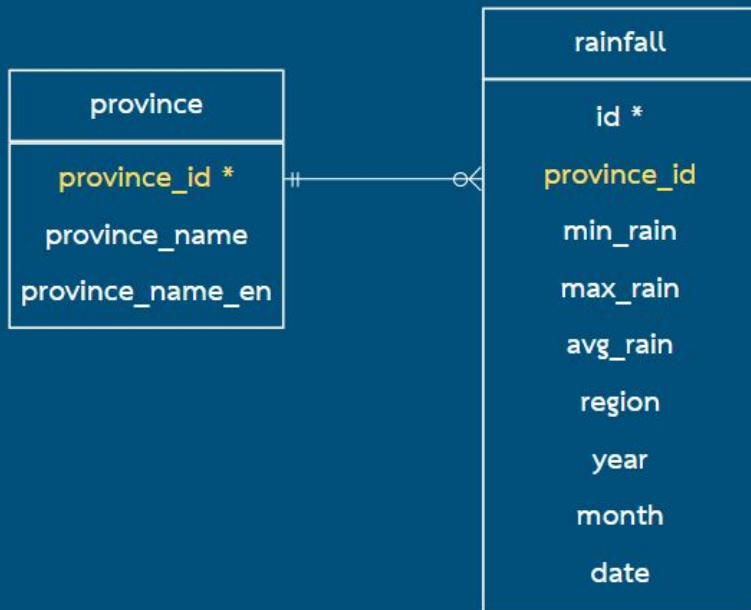
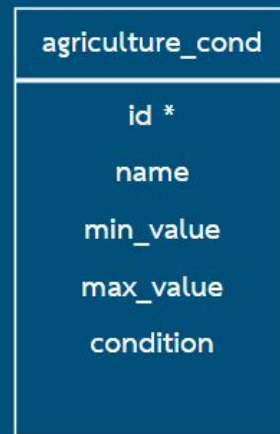
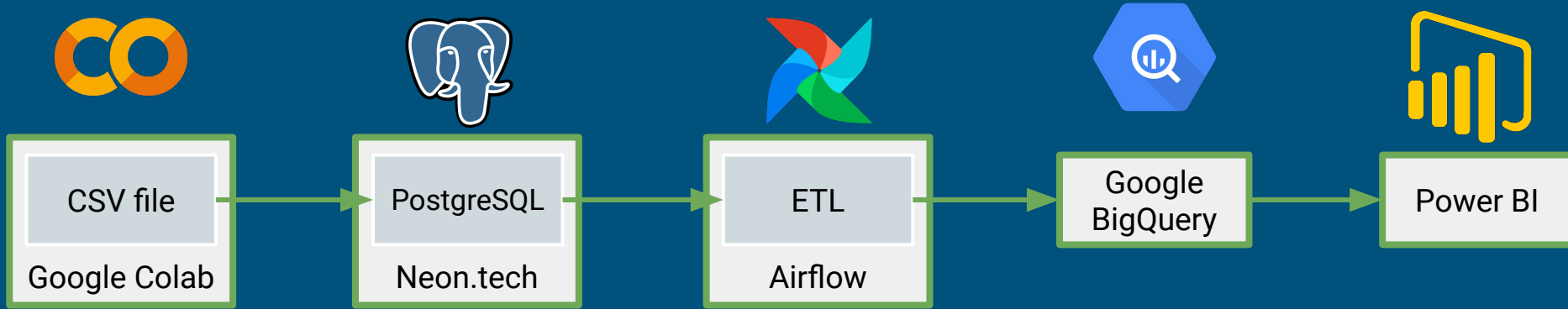


Table Condition

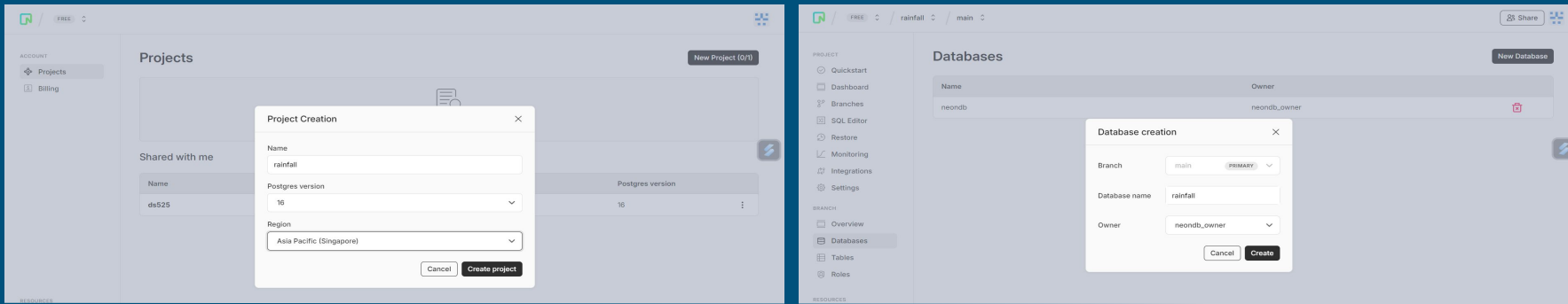


Data Ingestion - Workflow Orchestration




ขั้นตอนการทำงาน - สร้างฐานข้อมูล Postgresql


1. สร้าง Project “rainfall” และ Database “rainfall” ใน NEON จะได้ url ของ Postgres





The first screenshot shows the 'Projects' page in the NeonDB interface. A 'Project Creation' modal is open, allowing the user to create a new project. The modal fields are: Name (rainfall), Postgres version (16), and Region (Asia Pacific (Singapore)). The 'Create project' button is visible.


The second screenshot shows the 'Databases' page in the NeonDB interface. A 'Database creation' modal is open, allowing the user to create a new database. The modal fields are: Branch (main), Database name (rainfall), and Owner (neondb_owner). The 'Create' button is visible.



Postgres



NeonDriver



NextJS



Prisma



Node.js


Django


SQLA


Java


Symfony


Go

postgresql://

ขั้นตอนการทำงาน - Insert ข้อมูลใน NEON โดย Google colab

2. เปิดไฟล์ load.ipynb ด้วย Google colab และนำ url ที่ได้จากขั้นตอนที่ 1 มาใช้เพื่อ connect กับ database ของ NEON และอัปโหลด rainfall.csv เข้า Google colab ซึ่งในขั้นตอนนี้จะทำการสร้าง Table ชื่อ "rainfall" และ insert ข้อมูลจากไฟล์ เพื่อนำข้อมูลเข้าไปเก็บใน postgres database ที่อยู่ใน NEON

The screenshot displays a Google Colab notebook with the following content:

```
import os
import psycopg2
# Provide your Neon connection string
connection_string = "postgresql://[redacted]"
# Connect using the connection string
connection = psycopg2.connect(connection_string)
# Create a new cursor object
cursor = connection.cursor()

create_table_sql = '''
CREATE TABLE IF NOT EXISTS rainfall (
  id BIGSERIAL PRIMARY KEY,
  ProvinceId INTEGER,
  ProvinceName CHAR(60),
  ProvinceName2 CHAR(60),
  MinRain double precision,
  MaxRain double precision,
  AvgRain double precision,
  region CHAR(20),
  Year CHAR(4),
  Month CHAR(2),
  Date DATE,
  Month_TH CHAR(20)
);
'''

cursor.execute(create_table_sql)
result = connection.commit()
print(result)
```

Below the code, a file named 'rainfall.csv' is shown in the file browser. To the right, a code cell titled 'Insert data to Neon' contains the following code:

```
import pandas as pd
import numpy as np

pd = pd.read_csv("rainfall.csv");

import datetime
format = '%d/%m/%y'
```

At the bottom, the Neon database interface is visible, showing the 'rainfall' table with the following data:

#	id	provinceid	provincename	provincename2	minrain	maxrain	avgrain
1	1	10	กรุงเทพมหานคร	Bangkok	54.3	257.23	142.119
2	2	11	สมุทรปราการ	Samut Prakan	76.25	256.1	137.302
3	3	12	นนทบุรี	Nonthaburi	38.36	161.47	113.434
4	4	13	ปทุมธานี	Pathum Thani	51.44	116.5	82.902
5	5	14	พระนครศรีอยุธยา	Phra Nakhon Si Ayutthaya	8.85	88.59	39.96

ขั้นตอนการทำงาน - สร้าง Data Warehouse โดยใช้ Google BigQuery

3. สร้าง Project และ Dataset สำหรับเก็บข้อมูล Tables และทำการ create table ชื่อ "agriculture_cond"

Rich 'View all products' to view more products and add them to the navigation menu

Google Cloud

Explorer

Q. Type to search

Viewing resources. SHOW STARRED ONLY

Queries

Notebooks

Data canvases

External connections

rainfalldtest

Nothing currently selected

Summary

New code management

Create table

Source

Create table from

Empty table

Destination

Project *

Dataset * rainfalldtest

Table * agriculture_cond

Table type

Native table

Schema

Edit as text

Partition and cluster settings

Partitioning

No partitioning

Clustering order

CREATE TABLE CANCEL

Edit Schema

Current schema

ADD POLICY TAG

Filter Enter property name or value

	Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
	No results to display						

New fields

Edit as text

1	Field name * id	Type * INTEGER	Mode * NULLABLE				Description
2	Field name * name	Type * STRING	Mode * NULLABLE		Max length 64		Description ชื่อพื้นที่
3	Field name * min_value	Type * FLOAT	Mode * NULLABLE				Description
4	Field name * max_value	Type * FLOAT	Mode * NULLABLE				Description
5	Field name * condition	Type * STRING	Mode * NULLABLE		Max length 255		Description
6	ADD FIELD						

SAVE CLOSE

ขั้นตอนการทำงาน - สร้าง Data Warehouse บนระบบ Cloud โดยใช้ Google Cloud BigQuery (ต่อ)

4. ทำการ Insert Data เข้าสู่ Table “agriculture_cond” จะได้ Table Condition ปริมาณน้ำฝนที่เหมาะสมกับพืชแต่ละชนิด

The screenshot shows the Google Cloud BigQuery interface. The left sidebar contains navigation options like Analysis, Migration, and Administration. The main area displays a query editor with a query to insert data into the 'agriculture_cond' table. The query is as follows:

```
1 SELECT FROM [redacted] rainfalltest.agriculture_cond LIMIT 1000
2
3 insert into 'rainfalltest.agriculture_cond' ('name', 'min_value', 'max_value', 'condition') values ('ข้าว', 83, 166,
4 'min_value <= x <= max_value'), ('ข้าวเหนียว', 41, 50, 'min_value <= x <= max_value'), ('ข้าวเจ้า', 150, 166,
5 'min_value <= x <= max_value'), ('ข้าวเหนียว', 83, 250, 'min_value <= x <= max_value'), ('ข้าวเจ้า', 100, 116, 'min_value <= x <=
max_value'), ('ข้าว', 100, 133, 'min_value <= x <= max_value')
6
7 SELECT FROM [redacted] rainfalltest.agriculture_cond LIMIT 1000
```

Below the query editor, the 'Query results' section is visible, showing a table with 6 rows of data. The table has columns: Row, Id, name, min_value, max_value, and condition. The data is as follows:

Row	Id	name	min_value	max_value	condition
1	null	ข้าวเจ้า	100.0	116.0	min_value <= x <= max_value
2	null	ข้าว	100.0	133.0	min_value <= x <= max_value
3	null	ข้าวเหนียว	41.0	50.0	min_value <= x <= max_value
4	null	ข้าว	83.0	166.0	min_value <= x <= max_value
5	null	ข้าวเจ้า	83.0	250.0	min_value <= x <= max_value
6	null	ข้าวเหนียว	150.0	166.0	min_value <= x <= max_value

ขั้นตอนการทำงาน - สร้าง Data Warehouse บนระบบ Cloud โดยใช้ Google Cloud BigQuery (ต่อ)

5. ทำการ Create service account ที่ IAM & Admin เพื่อสร้าง Key และกำหนด Role จากนั้นให้ download file key ในรูปแบบ .json แล้ว upload ใน folder capstone project ใน github

The screenshot shows the 'Create service account' wizard in the Google Cloud IAM & Admin console. The left sidebar lists navigation options like IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts, Workload Identity Federation, Workforce Identity Federation, Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, Manage Resources, and Release Notes. The main content area is titled 'Create service account' and contains three steps: 1. Service account details, 2. Grant this service account access to project (optional), and 3. Grant users access to this service account (optional). Step 1 is currently active, showing fields for 'Service account name' (rainfall), 'Service account ID' (rainfall), 'Email address' (rainfall@iam.gserviceaccount.com), and 'Service account description'. A 'CREATE AND CONTINUE' button is at the bottom.

Google Cloud

Search (/) for resources, docs, products, and more

IAM & Admin

Create service account

1 Service account details

Service account name
rainfall

Display name for this service account

Service account ID *
rainfall

Email address
rainfall@iam.gserviceaccount.com

Service account description

Describe what this service account will do

CREATE AND CONTINUE

2 Grant this service account access to project (optional)

3 Grant users access to this service account (optional)

DONE CANCEL

The screenshot shows the 'Keys' tab for a service account named 'rainfall' in the Google Cloud IAM & Admin console. The left sidebar lists navigation options like DETAILS, PERMISSIONS, KEYS, METRICS, and LOGS. The main content area is titled 'Keys' and contains a warning message: 'Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation. You can learn more about the best way to authenticate service accounts on Google Cloud here.' Below the warning, there is a section for 'Add a new key pair or upload a public key certificate from an existing key pair.' and a link to 'Block service account key creation using organization policies'. A table below shows the list of keys.

My Project 95710

Search (/) for resources, docs, products, and more

rainfall

DETAILS PERMISSIONS KEYS METRICS LOGS

Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).

[Learn more about setting organization policies for service accounts](#)

ADD KEY

Type	Status	Key	Creation date	Expiration date
🔑	Active	[REDACTED]	May 6, 2024	Jan 1, 10000

ขั้นตอนการทำงาน - เตรียมการเชื่อมต่อ Airflow

6. ใน Github ปรับแก้ไฟล์ ETL_Postgres_to_Bigquery.py ดังนี้

- แก้ dbname, user, password, host เพื่อเชื่อมต่อกับฐานข้อมูล NEON Postgres
- แก้ keyfile, project_id และ dataset_id เพื่อเชื่อมต่อกับ Data Warehouse Google Cloud BigQuery

```
ETL_Postgres_to_Bigquery.py M X  README.md capstone_project M  docker-compose.yaml  RE
capstone_project > dags > ETL_Postgres_to_Bigquery.py
29 def _create_tables():
30
31
32
33
34
35
36
37
38
39
40
41 # changes dbname
42 # changes user
43 # changes password
44 # changes host
45 def _neon_to_rainfall_csv():
46     # เชื่อมต่อกับฐานข้อมูล Neon Postgres
47     conn = psycopg2.connect(
48         dbname="rainfall",
49         user="neondb_owner",
50         password="XXXXXXXXXX",
51         host="XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
52         port="5432"
53     )
54
```

```
# changes keyfile
# แต่เพื่อความง่ายเราสามารถกำหนด File Path ไปได้เลยตรง ๆ
keyfile = "/opt/airflow/dags/XXXXXXXXXXXXXXXXXXXX.json"
service_account_info = json.load(open(keyfile))
credentials = service_account.Credentials.from_service_account_info(service_account_info)

# โค้ดส่วนนี้จะเป็นการสร้าง Client เชื่อมต่อไปยังโปรเจกต์ GCP ของเรา โดยใช้ Credentials ที่
# สร้างจากโค้ดข้างต้น

# changes project_id
project_id = "XXXXXXXXXX"
client = bigquery.Client(
    project=project_id,
    credentials=credentials,
)

main(dataset_id="rainfalltest", table_id="province", file_path="province.csv") # changes dataset_id
```

ขั้นตอนการทำงาน - เตรียมการเชื่อมต่อ Airflow (ต่อ)

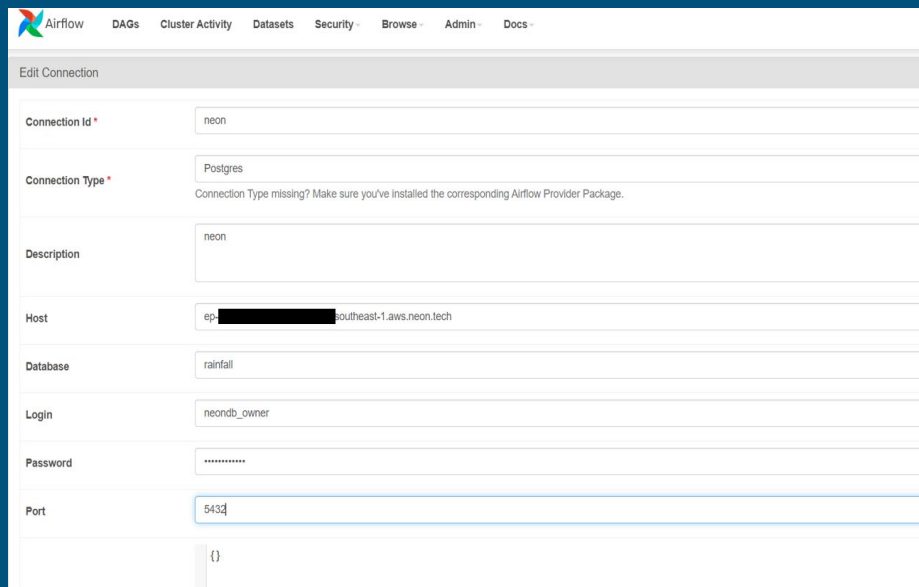
7. เปิด terminal โดยเข้าสู่โฟลเดอร์ที่ต้องการด้วยคำสั่ง `cd` ชื่อโฟลเดอร์ และเปิดใช้งาน Airflow ด้วยคำสั่ง `docker-compose up`

```
docker-compose up
```

8. เปิด Port 8080 จะเข้าไปที่หน้า Airflow และใส่ username password โดยดูจากไฟล์ `docker-compose.yaml`

ขั้นตอนการทำงาน - สร้าง Data Pipeline ด้วย Airflow

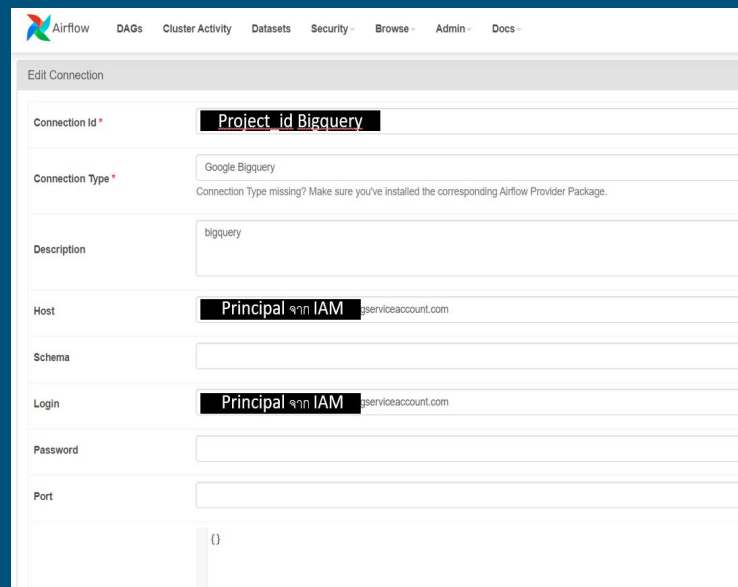
9. add connection เพื่อเชื่อมต่อ NEON และ Google BigQuery



The screenshot shows the 'Edit Connection' form in the Airflow web interface. The form fields are as follows:

- Connection Id ***: neon
- Connection Type ***: Postgres
- Description**: neon
- Host**: ep-[REDACTED]:southeast-1.aws.neon.tech
- Database**: rainfall
- Login**: neondb_owner
- Password**: [REDACTED]
- Port**: 5434

Below the form, there is a JSON configuration snippet: `{}`.



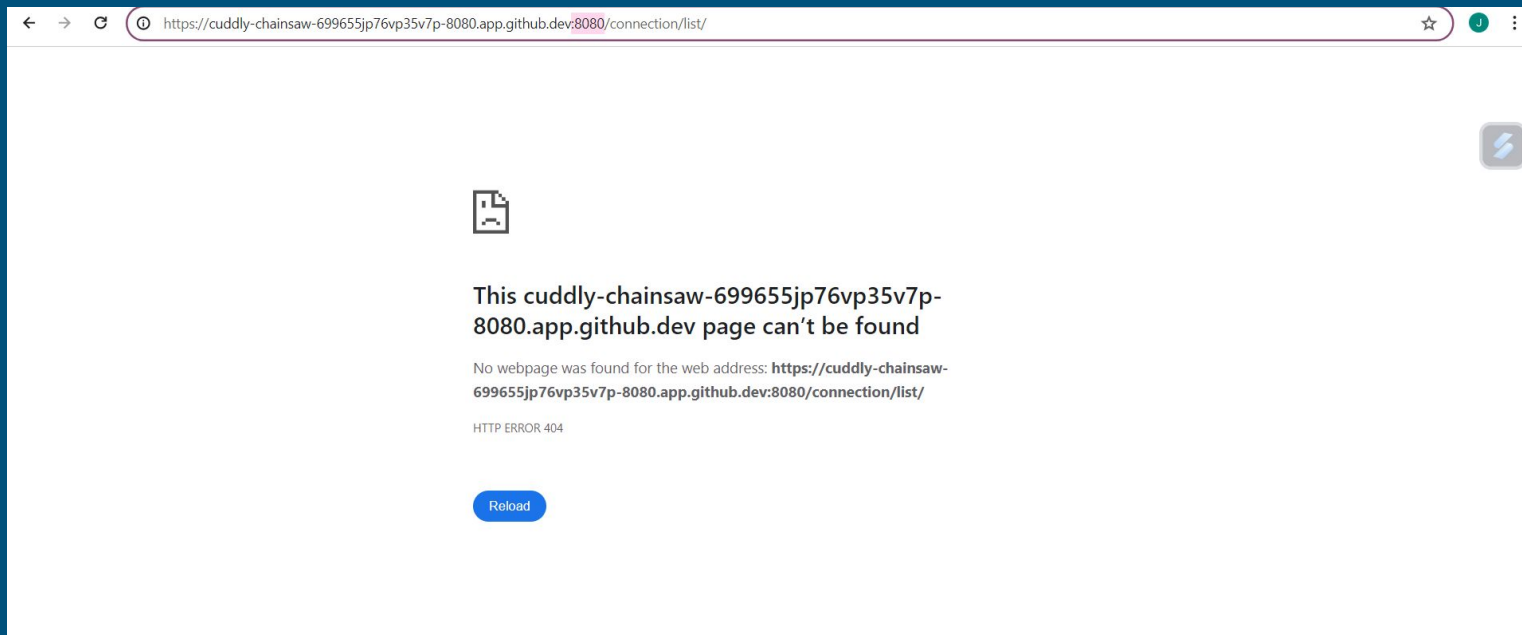
The screenshot shows the 'Edit Connection' form in the Airflow web interface for a Google BigQuery connection. The form fields are as follows:

- Connection Id ***: Project_id Bigquery
- Connection Type ***: Google BigQuery
- Description**: bigquery
- Host**: Principal [REDACTED] IAM [REDACTED]@serviceaccount.com
- Schema**: [REDACTED]
- Login**: Principal [REDACTED] IAM [REDACTED]@serviceaccount.com
- Password**: [REDACTED]
- Port**: [REDACTED]

Below the form, there is a JSON configuration snippet: `{}`.

ขั้นตอนการทำงาน - สร้าง Data Pipeline ด้วย Airflow (ต่อ)

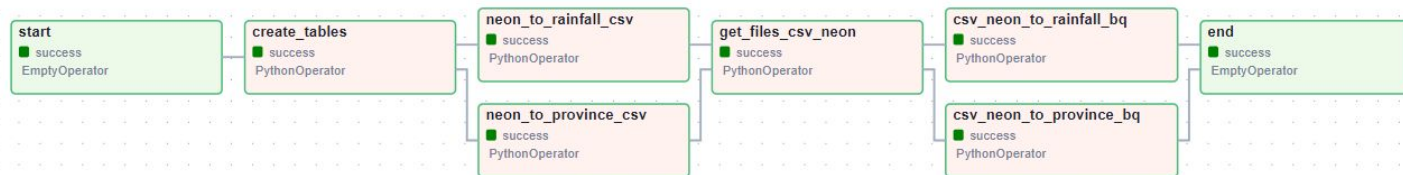
10. หาก save แล้วขึ้นว่า page can't be found ให้ลบ :8080 ออกแล้ว refresh



ขั้นตอนการทำงาน - สร้าง Data Pipeline ด้วย Airflow (ต่อ)

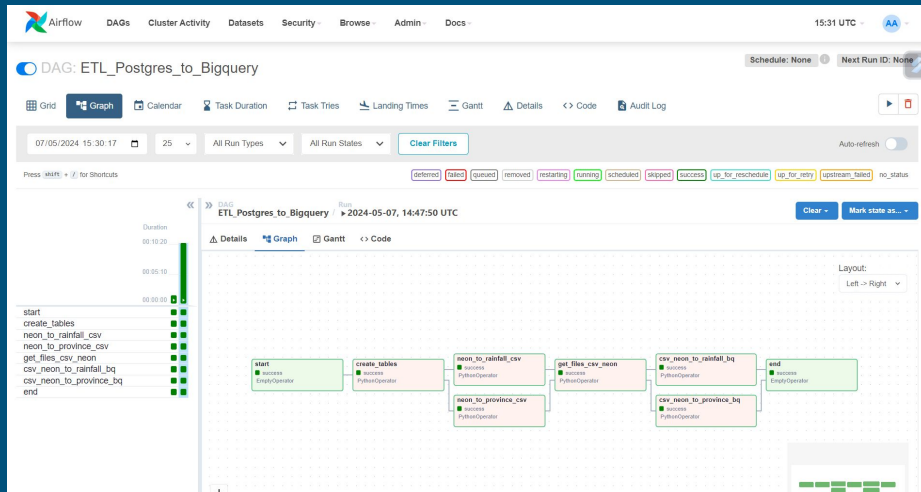
11. ใน Github ให้ปรับแก้ไฟล์ ETL_Postgres_to_Bigquery.py โดยแก้ postgres_conn_id="neon" ซึ่งได้มาจากการเพิ่ม connection NEON ใน UI Airflow tab Admin >> connection จากนั้นจะได้ Data Pipeline ใน DAGS

```
26
27 // # changes postgres_conn_id="conn id" ได้มาจากการเพิ่ม connection ใน UI Airflow tab Admin >> connection
28
29 def _create_tables():
30 //     hook = PostgresHook(postgres_conn_id="neon") #ใช้ PostgresHook เพื่อเชื่อมต่อกับฐานข้อมูล PostgreSQL
31     conn = hook.get_conn()
32     cur = conn.cursor()
33
```



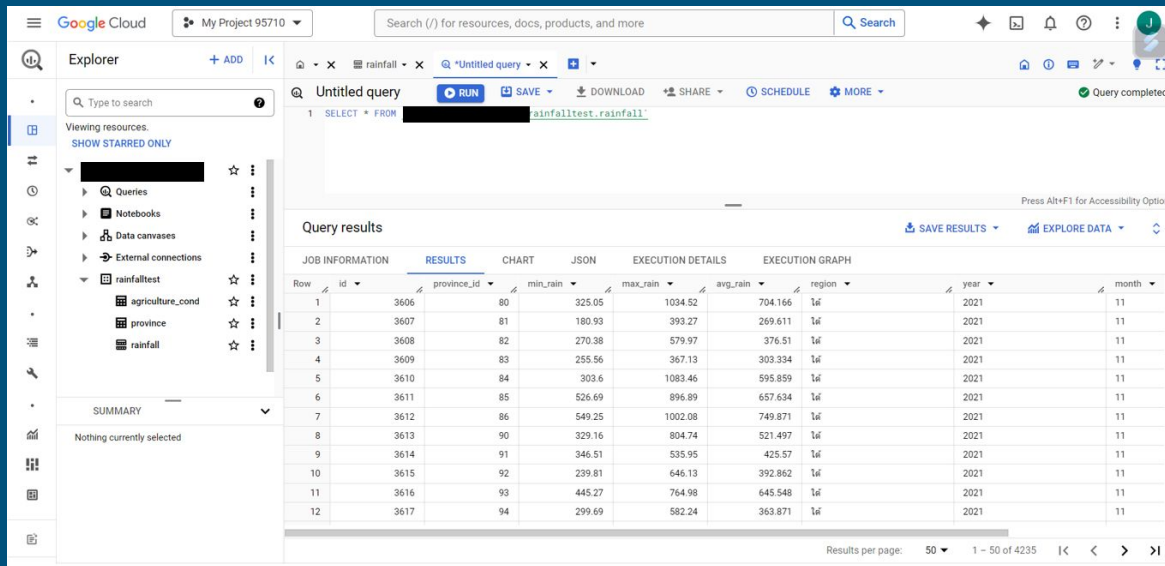
ขั้นตอนการทำงาน - ทำการ Data Transformation ด้วย Airflow

12. ในหน้า UI Airflow ให้ทำการ run ETL_Postgres_to_Bigquery
13. จะได้ไฟล์ rainfall.csv กับ province.csv เข้ามาอยู่ใน dags เพื่อที่นำข้อมูลเข้าไปยัง Google BigQuery



ขั้นตอนการทำงาน - ทำการ Data Transformation ด้วย Airflow (ต่อ)

14. เมื่อเข้าไปยัง Google BigQuery จะพบว่ามี Tables rainfall และ province เพิ่มขึ้น

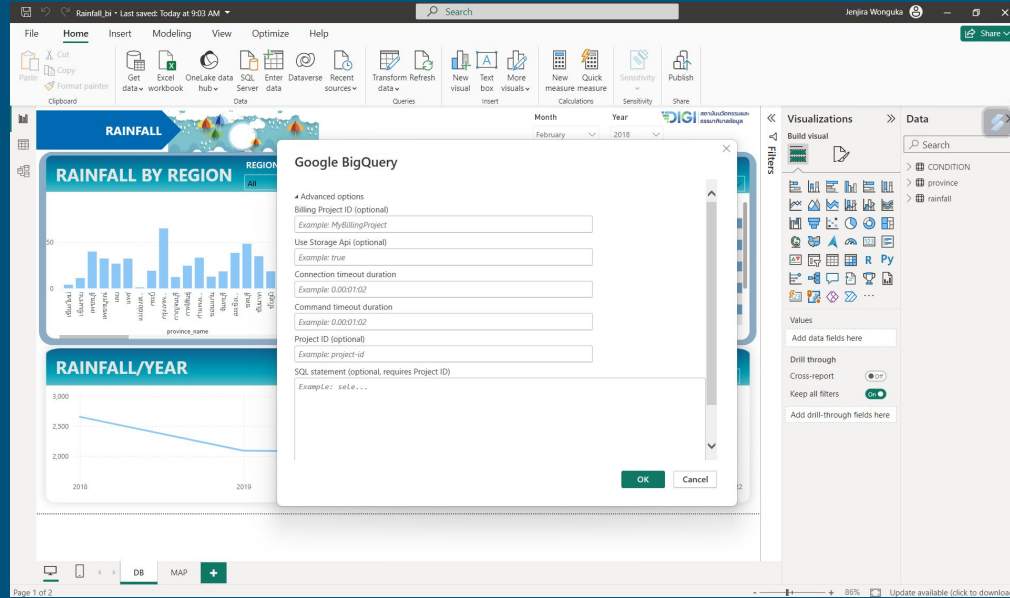


The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer pane shows a project named 'My Project 95710' with a folder 'rainfalltest' containing tables 'agriculture_cond', 'province', and 'rainfall'. The main area displays a query titled 'Untitled query' with the SQL statement: `SELECT * FROM [redacted].rainfalltest.rainfall`. The query has been executed successfully, as indicated by the 'Query completed' status. Below the query, the 'Query results' section shows a table with 12 rows and 10 columns: Row, Id, province_id, min_rain, max_rain, avg_rain, region, year, and month. The data represents rainfall statistics for various provinces in Thailand for the year 2021.

Row	Id	province_id	min_rain	max_rain	avg_rain	region	year	month
1	3606	80	325.05	1034.52	704.166	ใต้	2021	11
2	3607	81	180.93	393.27	269.611	ใต้	2021	11
3	3608	82	270.38	579.97	376.51	ใต้	2021	11
4	3609	83	255.56	367.13	303.334	ใต้	2021	11
5	3610	84	303.6	1083.46	595.859	ใต้	2021	11
6	3611	85	526.69	896.89	657.634	ใต้	2021	11
7	3612	86	549.25	1002.08	749.871	ใต้	2021	11
8	3613	90	329.16	804.74	521.497	ใต้	2021	11
9	3614	91	346.51	535.95	425.57	ใต้	2021	11
10	3615	92	239.81	646.13	392.862	ใต้	2021	11
11	3616	93	445.27	764.98	645.548	ใต้	2021	11
12	3617	94	299.69	582.24	363.871	ใต้	2021	11

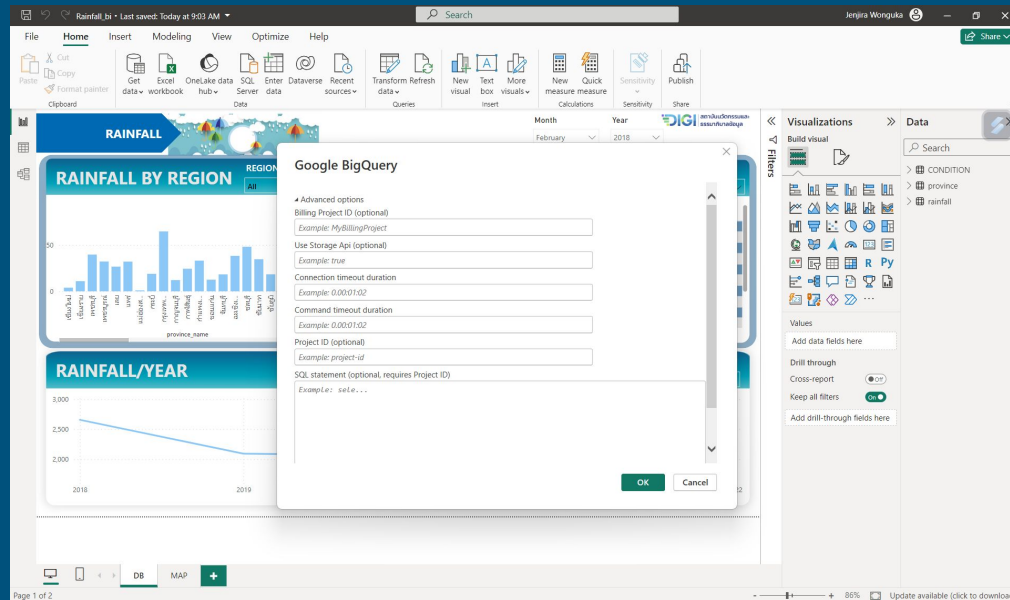
ขั้นตอนการทำงาน - สร้าง Data Visualization ด้วย Power BI

15. ทำการเชื่อมต่อกับ Google BigQuery เพื่อดึงข้อมูลเข้ามายัง Power BI โดยใช้ Project id ที่ต้องการเชื่อมต่อ



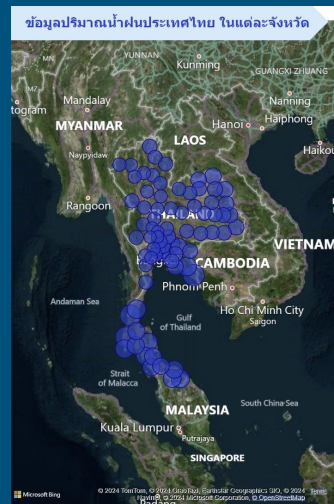
ขั้นตอนการทำงาน - สร้าง Data Visualization ด้วย Power BI (ต่อ)

16. ทำการ Transform data และเลือกข้อมูลที่ต้องการ (ในที่นี้ใช้ข้อมูล rainfall) ที่อยู่ใน Project Id ที่ได้ทำการเชื่อมต่อไว้



ขั้นตอนการทำงาน - สร้าง Data Visualization ด้วย Power BI (ต่อ)

17. ทำการสร้าง Visualization ตามวัตถุประสงค์ที่ต้องการ



เปรียบเทียบปริมาณน้ำฝนเฉลี่ยรายเดือนรายปีในภาคเหนือ

region	2018	2019	2020	2021	2022	Total
เชียงใหม่	110.29	83.47	85.04	105.20	117.64	96.75
ลำพูน	191.54	162.36	203.03	205.60	200.25	191.66
ลำปาง	102.11	75.18	92.09	114.40	108.52	97.54
สุโขทัย	103.60	79.12	93.65	120.55	101.09	99.47
สุพรรณบุรี	161.26	142.10	159.87	183.69	153.48	160.68
สุราษฎร์ธานี	116.80	106.97	109.08	125.13	136.33	117.27
Total	128.62	106.60	122.11	139.39	137.09	125.82

เปรียบเทียบปริมาณน้ำฝนเฉลี่ยรายเดือนรายปีในภาคกลาง

province_name	2018	2019	2020	2021	2022	Total
เชียงใหม่	110.69	81.45	71.68	98.72	109.06	93.42
เชียงใหม่	148.53	80.33	95.01	130.47	148.50	118.20
เชียงใหม่	90.40	66.66	89.80	101.02	85.49	88.68
เชียงใหม่	91.51	76.43	86.42	105.51	99.33	91.16
เชียงใหม่	94.33	64.76	93.15	121.89	109.30	95.54
เชียงใหม่	93.64	85.01	98.87	95.46	110.44	95.43
เชียงใหม่	110.14	85.07	70.50	99.53	105.29	93.09
เชียงใหม่	177.52	139.93	197.80	177.82	169.90	172.84
เชียงใหม่	138.18	80.44	103.37	123.99	137.40	114.79
เชียงใหม่	106.73	85.43	102.91	129.34	110.02	106.60
เชียงใหม่	106.46	111.09	101.29	104.54	138.27	109.97
เชียงใหม่	91.45	85.12	70.88	122.74	124.72	96.64
เชียงใหม่	97.22	86.71	122.24	103.00	116.09	99.47
เชียงใหม่	109.96	206.34	210.44	246.19	194.18	211.25
เชียงใหม่	145.52	91.19	100.21	130.81	128.64	122.79
เชียงใหม่	115.86	79.79	115.49	131.99	111.74	110.90
Total	128.62	106.60	122.11	139.39	137.09	125.82

