

## คู่มือการใช้งานโปรแกรมคัดกรองฐานข้อมูล

### 1. ความต้องการขั้นพื้นฐาน

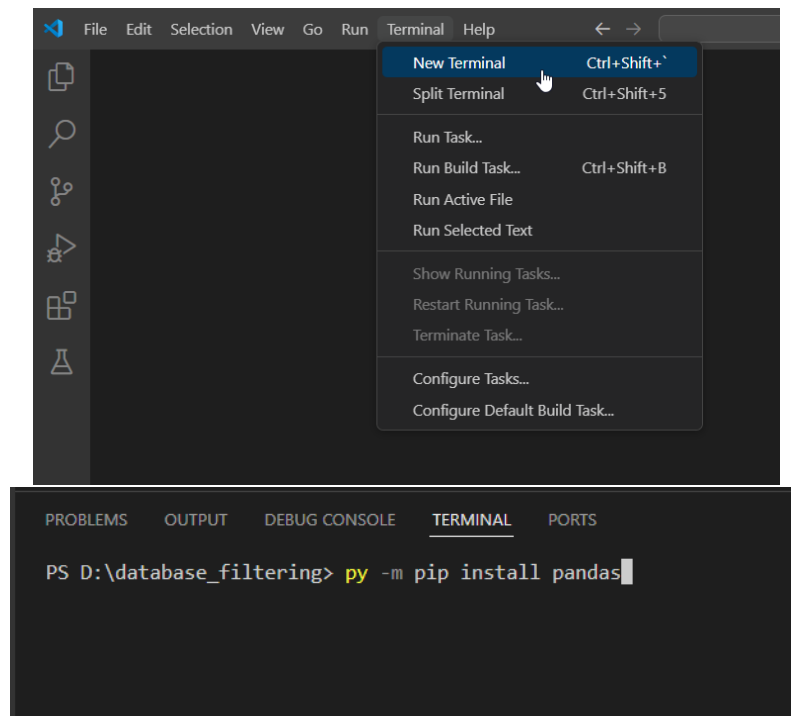
1.1 โปรแกรม Visual Studio Code

1.2 Python version 3.12 ขึ้นไป

1.3 ติดตั้ง Library

- Pandas ด้วยคำสั่ง `py -m pip install pandas`

- Openpyxl ด้วยคำสั่ง `py -m pip install openpyxl`



### 2. การใช้โปรแกรม filtering\_program.py

โปรแกรมนี้ออกแบบมาเพื่อคัดกรองข้อมูลตามเงื่อนไขที่กำหนด โดยผู้ใช้สามารถเพิ่มและปรับแต่งเงื่อนไขได้ด้วยการใช้ Regular Expression

#### 2.1 ไฟล์เอกสารที่ต้องเตรียมก่อนเริ่มโปรแกรม

1. เอกสารที่ต้องการจะกรองข้อมูล โดยให้ใช้ชื่อว่า data.csv
2. เอกสาร Excel ที่ใช้เพื่อระบุเงื่อนไข โดยใช้ชื่อว่า condition.xlsx
3. เอกสาร รหัสศูนย์ต้นทุน-รหัสบัญชี.xlsx

ไฟล์ทั้งหมดจะต้องอยู่ในโฟลเดอร์เดียวกับ filtering\_program.py

## 2.2 การใช้งาน condition.xlsx

การเขียนเงื่อนไข สามารถดูตัวอย่างเงื่อนไขเบื้องต้นได้ในไฟล์ example\_condition.xlsx

1. ระบุรหัสที่ต้องการค้นหาลงในคอลัมน์ รหัส โดยสามารถเปลี่ยนตัวเลขเป็น X เพื่อให้เป็นตัวเลขใดก็ได้
2. เขียนเงื่อนไขที่ต้องการกรองลงในคอลัมน์ find ในรูปแบบ Regular expression โดยโปรแกรมจะทำการกรองข้อมูลตรงกับ Regular expression ออกมา
3. หากมีรหัส G/L ที่ไม่ต้องการให้กรอง สามารถใส่รหัส G/L ได้ในคอลัมน์ exclude G/L

## 2.3 เงื่อนไขพิเศษ

1. “act” ตามด้วยรหัสกิจกรรมที่ต้องการตรวจสอบ เช่น act N0301 และ รหัส 51XXXXXX จะค้นหารหัสกิจกรรม N0301 ที่ไม่ใช้รหัสบัญชี 51XXXXXX

51XXXXXX		รหัสกิจกรรม N0301 รหัสบัญชี 51XXXXXX เท่านั้น	act N0301
54XXXXXX		รหัสกิจกรรม N0104 รหัสบัญชี 54XXXXXX เท่านั้น	act N0104
54XXXXXX		รหัสกิจกรรม N0105 รหัสบัญชี 54XXXXXX เท่านั้น	act N0105

2. “segment” ตามด้วยรหัสผลิตภัณฑ์ที่ต้องการตรวจสอบ โปรแกรมจะค้นหารหัสเซกเมนต์ที่มีรหัส G/L ไม่ตรงกับที่กำหนดและ G/L ที่มีรหัสเซกเมนต์ไม่ตรงกับที่กำหนด

59XXXXXX	ค่าใช้จ่าย	บันทึกเข้า เซกเมนต์ 19202	segment 19202
----------	------------	---------------------------	---------------

3. “cancel\_product” โปรแกรมจะค้นหาข้อมูลที่มีรหัสผลิตภัณฑ์ยกเลิก
4. “cancel\_act” โปรแกรมจะค้นหาข้อมูลที่มีรหัสกิจกรรมยกเลิก

5XXXXXXX	ค่าใช้จ่าย	ห้ามบันทึกรหัส Product ยกเลิก	cancel_product
5XXXXXXX	ค่าใช้จ่าย	ห้ามบันทึกรหัสกิจกรรมยกเลิก	cancel_act

## 2.4 เริ่มทำงานโปรแกรม

เมื่อเตรียมไฟล์ทั้งหมดแล้ว ผู้ใช้สามารถเปิดไฟล์ filtering\_program.py ผ่าน Visual Studio Code และเริ่มการทำงานได้โดยการกด run หรือปุ่ม ctrl+F5

## 3. ผลลัพธ์

หลังจากที่โปรแกรมทำงานเสร็จสิ้น ผลลัพธ์ทั้งหมดจะถูกเก็บไว้ในโฟลเดอร์ Program/Filter\_result โดยจำนวนผลลัพธ์จะขึ้นอยู่กับจำนวนแถวของเงื่อนไขใน condition.xlsx เรียงจากบน

ลงล่าง ผลลัพธ์ที่ได้จะเป็นข้อมูลที่ตรงตามเงื่อนไขที่กำหนด เช่น รหัส 5X604XXX ที่บันทึกรหัสกระบวนการทางธุรกิจ ผลลัพธ์ที่ได้ก็จะเป็นข้อมูล รหัส 5X604XXX ที่ไม่บันทึกรหัสกระบวนการทางธุรกิจ เป็นต้น

#### 4. ภาคผนวก

การเขียน Regular expression เบื้องต้น

[medium.com/@siwat.s/192d1559bf50](https://medium.com/@siwat.s/192d1559bf50)

การใช้งาน ChatGPT

[blog.tobejourney.com/6730e7531f39](https://blog.tobejourney.com/6730e7531f39)

Source filtering\_program.py

[https://github.com/Righty75/database\\_filtering](https://github.com/Righty75/database_filtering)

```
import pandas as pd
import shutil
import os
import re
from pathlib import Path

# ตารางเงื่อนไข
condition_table = pd.read_excel('./Program/condition.xlsx', dtype=str)

# ข้อมูลที่ต้องการนำมากรอง
main_df = pd.read_csv('./Program/data_202405.csv', dtype=str)

# ตารางรหัสศูนย์ต้นทุน-รหัสบัญชี
account_name = pd.read_excel('./Program/รหัสศูนย์ต้นทุน-รหัสบัญชี.xlsx', 'G/L', dtype=str)

# ตารางรวมรหัส กิจกรรม, Product ยกเลิก
cancel_product = pd.read_excel('./Program/condition.xlsx', 'รหัส Product ยกเลิก', dtype=str)['รหัส']
cancel_act = pd.read_excel('./Program/condition.xlsx', 'รหัสกิจกรรมยกเลิก')['Act']

# -----#

# สำหรับใส่ชื่อ G/L ในคอลัมน์ stat
main_df = pd.merge(main_df.drop(columns=['Stat']), account_name, on='G/L', how='left')
col = main_df.pop('Stat')
main_df.insert(1, col.name, col)

def setup_output_directory(directory_path):
    shutil.rmtree(directory_path, ignore_errors=True)
    Path(directory_path).mkdir(parents=True, exist_ok=True)

# เปลี่ยน X,x ในตารางเงื่อนไขเป็น \d เพื่อใช้ใน regex
condition_table['รหัส'] = condition_table['รหัส'].apply(lambda x: re.sub(r'[Xx]', r'\d', x))

def apply_conditions(main_df, condition_table, cancel_product, cancel_act, output_directory):
    write_header = True
    for index, row in condition_table.iterrows():
        find_pattern = row['find']
        exclude_pattern = row['exclude G/L']
        filtered_df = main_df.loc[main_df['G/L'].str.contains(row['รหัส'], na=False, regex=True)]

        if pd.notna(exclude_pattern):
            filtered_df = filtered_df.loc[~filtered_df['G/L'].str.contains(exclude_pattern, na=False)]

        if pd.notna(find_pattern):
            if 'segment' in find_pattern.lower():
                segment_code = ''.join(filter(str.isdigit, find_pattern))
                filtered_segment = main_df.loc[main_df['segment'].str.contains(segment_code, na=False)]
                filtered_segment = filtered_segment.loc[~filtered_segment['G/L'].str.contains(row['รหัส'], na=False, regex=True)]
                filtered_df = filtered_df.loc[~filtered_df['segment'].str.contains(segment_code, na=False)]
                filtered_df = pd.concat([filtered_segment, filtered_df])
```

```

elif 'act ' in find_pattern.lower():
    act_code = find_pattern.split()[1]
    filtered_act = main_df.loc[main_df['Bus. Process'].str.contains(act_code, na=False)]
    filtered_df = filtered_act.loc[~filtered_act['G/L'].str.contains(row['รหัส'], na=False, regex=True)]

elif 'cancel_product' in find_pattern.lower():
    filtered_df = filtered_df.loc[filtered_df['ผลิตภัณฑ์/'].str.contains('|'.join(cancel_product), na=False)]
elif 'cancel_act' in find_pattern.lower():
    filtered_df = filtered_df.loc[filtered_df['Bus. Process'].str.contains('|'.join(cancel_act), na=False)]
else:
    filtered_df = filtered_df.loc[filtered_df['Bus. Process'].str.contains(find_pattern, na=False, regex=True)]
else:
    filtered_df = filtered_df.loc[filtered_df['Bus. Process'].isna()]

#เพิ่มคอลัมน์ 'เงื่อนไข' ลงใน DataFrame ที่กรองแล้ว
filtered_df = filtered_df.assign(เงื่อนไข1 = row['เงื่อนไข 1'])
filtered_df = filtered_df.assign(เงื่อนไข2 = row['เงื่อนไข 2'])

output_path = os.path.join(output_directory, f'result_{index + 1}.csv')
output_path_combine = os.path.join(output_directory, f'combine_result.csv')
filtered_df.to_csv(output_path, index=False)
filtered_df.to_csv(output_path_combine, mode='a', index = False, header=write_header)
write_header = False

def main():
    #ตั้งค่าไฟล์เตอร์ผลลัพธ์
    output_directory = './Program/Filter_result'
    setup_output_directory(output_directory)
    apply_conditions(main_df, condition_table, cancel_product, cancel_act, output_directory)
    print("Filtering and saving completed.")

if __name__ == "__main__":
    main()

```