



Project 3 - Batch Processing



Table of Content

What will We Learn Today?

1. Intro
2. Open DB Connection
3. Interacting With Database
4. Creating Table
5. SQL Transaction
6. Copying the Data





Intro



Batch Processing

Pemrosesan *batch* adalah metode yang digunakan komputer untuk menyelesaikan tugas data berulang dengan volume tinggi secara berkala. Tugas pemrosesan data tertentu, seperti pencadangan, pemfilteran, dan pengurutan dapat menjadi komputasi yang intensif dan tidak efisien untuk dijalankan pada transaksi data individual. Sebaliknya, sistem data memproses tugas-tugas seperti itu dalam *batch*, seringkali di waktu tidak sibuk ketika sumber daya komputasi lebih umum tersedia, seperti di penghujung hari atau semalam.





Prerequisites

- GitHub Account
- Postgresql / SQLite
- VisualStudioCode
- Python



An abstract graphic design featuring a central white rectangular area with the word "Start" in a bold, dark blue sans-serif font. The background is a light cream color with a subtle pattern of small, faint yellow and blue geometric shapes. Surrounding the central area are large, flowing organic shapes in vibrant orange and deep blue. These shapes are decorated with various patterns: white 'x' marks, small white circles, a grid of blue dots, and clusters of yellow and blue 'L' shapes. In the top right corner, there is a stylized logo consisting of a dark blue 'D' shape with an orange and white diagonal bar. The overall composition is dynamic and modern, using a limited color palette of orange, blue, and cream.



Install Library psychopg2

```
pip install psychopg2
```



Connecting to Postgresql

Hal pertama saat kita akan membuat batch processing adalah melakukan koneksi terhadap database. Disini kita akan menggunakan library yang kita install sebelumnya

```
import psycopg2
conn = psycopg2.connect("host=localhost dbname=postgres user=postgres password=1234")
```





Interacting with database

Setelah kita memastikan bahwa koneksi bisa berjalan, maka selanjutnya adalah kita coba berinteraksi dengan database. Pada contoh ini saya akan coba select ke table yang sudah ada di database

```
import psycopg2
conn = psycopg2.connect("host=localhost dbname=postgres user=postgres password=1234")
```





Mencoba Select Ke Database

```
#menggunakan kursor
cur = conn.cursor()
cur.execute('Select * from public.employee')

#menampilkan hasil
one = cur.fetchone()
all = cur.fetchall()
print(one)
```



CRUD Dari Python





Create Table

```
#create table
cur.execute("""
    CREATE TABLE IF NOT EXISTS users(
        id integer PRIMARY KEY,
        email text,
        name text,
        address text
    )
""")
```

```
##Sebelum di commit table belum akan terbuat, maka lakukan commit
conn.commit()
```



Inserting Data

```
with open('/Users/nugrohom/Documents/project-3/data/users.csv','r') as f:
    reader = csv.reader(f)
    next(reader)
    for row in reader:
        cur.execute(
            "INSERT INTO users VALUES (%s, %s, %s, %s) ON CONFLICT DO NOTHING",
            row
        )
    conn.commit()
```





Inserting Data - Copy

```
with open('/Users/nugrohom/Documents/project-3/data/users_w_postal_code.csv', 'r') as f:
    # Notice that we don't need the `csv` module.
    next(f) # Skip the header row.
    cur.copy_from(f, 'users', sep=',', columns=('email', 'name', 'phone', 'postal_code'))

conn.commit()
```



Table From Files

```
create_table_from_files.py > ...
1  import pandas as pd
2  from sqlalchemy import create_engine
3  import psycopg2
4
5  df = pd.read_csv('/Users/nugrohom/Documents/project-3/data/users_w_postal_code.csv', sep=',')
6
7
8  ##> {dialect}+{driver}://{user}:{password}@{host}:{port}/{database}
9  #host=localhost dbname=postgres user=postgres password=1234
10
11  engine = create_engine('postgresql://postgres:1234@127.0.0.1:5432/postgres')
12  df.to_sql("from_file_table", engine)
```

Thank YOU

