etl_load

August 23, 2024

[1]:
```
# for database connections
import sqlalchemy
from sqlalchemy import create_engine, inspect, text

# data
import pandas as pd
```

[2]:
```
SQL_USERNAME = "postgres"
SQL_PASSWORD = "postgres"
SQL_IP = "localhost"
PORT = 5432
DATABASE = "crowdfunding_db"
```

[3]:
```
# create a dictionary with the tables and csv files \n",
TABLE_PATHS = {
    "category": "Resources/category.csv",
    "contact": "Resources/contacts.csv",
    "subcategory": "Resources/subcategory.csv",
    "campaign": "Resources/campaign.csv"
}
```

[4]:
```
connection_string = f"postgresql+psycopg2://{SQL_USERNAME}:
 ↪{SQL_PASSWORD}@{SQL_IP}:{PORT}/{DATABASE}"
engine = create_engine(connection_string)
```

[5]:
```
# explore and understand the data

# Create the inspector and connect it to the engine
inspector = inspect(engine)
```

[6]:
```
# Define the table to use in a for loop
def load_csv(table_name, csv_path):
    df = pd.read_csv(csv_path)
    df.to_sql(table_name, schema="public", con=engine, index=False,␣
 ↪if_exists="append", method="multi")
```

[7]:

```
# Make the for loop
for table in TABLE_PATHS:
    load_csv(table, TABLE_PATHS[table])
```

[8]:

```
# This will also work
# for table, path in TABLE_PATHS.items():
    #load_csv(table, path)
```