Zadania:

- 1. Napisz skrypt (bash lub powershell), który automatycznie wykona następujące kroki:
 - a. pobierze plik z internetu: https://home.agh.edu.pl/~wsarlej/Customers Nov2024.zip
 - b. rozpakuje pobrany plik,
 - c. sprawdzi poprawność pliku odrzucając błędne wiersze do pliku *Customers_Nov2024.bad_\${TIMESTAMP}*. Warunki walidacji:
 - zignoruj puste linie,
 - pozostaw tylko unikalne wiersze (deduplikacja),
 - pozostaw wiersze, które mają ilość kolumn taką jak nagłówek pliku,
 - wartość w kolumnie email nie jest pusta,
 - porównaj plik wejściowy z plikiem Customers_old.csv, pozostaw te wiersze, które nie występują w pliku Customers_old.csv: https://home.agh.edu.pl/~wsarlej/Customers_old.csv (pobrać manualnie),
 - d. w bazie danych PostgreSQL (lub innej) utworzy tabelę *CUSTOMERS_\${NUMERINDEKSU}*} tylko jeśli taka tabela już nie istnieje (warunek *if not exists*),
 - e. załaduje dane ze zweryfikowanego pliku do tabeli CUSTOMERS \${NUMERINDEKSU},
 - f. przeniesie przetworzony plik do podkatalogu PROCESSED dodając prefix \${TIMESTAMP}_ do nazwy pliku,
 - g. utworzy raport ze statystykami przetwarzania, nazwa pliku: CUSTOMERS LOAD \${TIMESTAMP}.dat, treść:
 - liczba wierszy w pliku pobranym z internetu: <wartość>,
 - liczba poprawnych wierszy (po czyszczeniu) : <wartość>,
 - liczba duplikatów w pliku wejściowym : <wartość>,
 - ilość danych załadowanych do tabeli CUSTOMERS_\${NUMERINDEKSU}:: <wartość>.
 - h. uruchomi kwerendę SQL, która znajdzie imiona i nazwiska klientów, którzy mieszkają w promieniu 50 kilometrów od punktu: 41.39988501005976, -75.67329768604034 (funkcja <u>ST DistanceSpheroid</u>) i zapisze je do tabeli BEST CUSTOMERS \${NUMERINDEKSU},
- i. wyeksportuje zawartość tabeli *BEST_CUSTOMERS_\${NUMERINDEKSU}* do pliku csv o takiej samej nazwie jak tabela źródłowa,
- j. skompresuje wyeksportowany plik csv,
- k. wyśle skompresowany plik do adresata poczty razem z raportem o treści: data utworzenia, ilość wierszy w pliku csv (w przypadku braku odpowiedniej konfiguracji serwera pocztowego zaproponuj komendę, ale jej nie uruchamiaj),
- po każdym kroku (a-k) sprawdzi czy komenda wykonała się poprawnie oraz zaloguje, że taki krok miał miejsce i wykonał się prawidłowo (wraz z krótkim opisem oraz czasem w którym nastąpiło wykonanie komendy (np. 20250101121212 – Creating report file- Successful).
 - Zalogowane zdarzenia będą zapisywane w pliku PROCESSED/nazwa skryptu \${TIMESTAMP}.log
- 2. Skrypt, plik .log z uruchomienia oraz kwerendę SQL proszę udostępnić w swoim repozytorium na github.

<u>Dodatkowe uwagi odnośnie skryptu:</u>

- na początku skryptu proszę dodać krótki changelog wraz z opisem skryptu oraz datą utworzenia,
- proszę w skrypcie sparametryzować wartości typu: numer indexu, automatycznie wyznaczana dzisiejsza data \${TIMESTAMP} w formacie MMDDYYYY, nazwa logu, adres pobranego pliku, hasło do archiwum, sql hostname, sql userid, sql hasło (hasła nie mogą być zapisywane do pliku .log!),
- wszelkie walidacje/zliczenia ilości wierszy muszą być automatyczne i działać również w przypadku otrzymania pliku z innymi danymi,
- adresat automatycznych maili to Państwa adres email (proszę uważać szczególnie w przypadku kiedy wysyłanie realizowane jest w pętli, proszę nie używać innych adresów email niż swój, proszę nie przesadzać z ilością wysyłanych maili),
- proszę dodać odpowiednie nagłówki do tworzonych plików i pomijać je przy zliczaniu ilości wierszy lub ładowaniu danych do bazy,
- proszę rozsądnie korzystać z plików tymczasowych (i usuwać je kiedy nie są potrzebne),
- proszę dopisać krótki komentarz przed kluczowymi krokami/komendami (np. # Deduplikacja),
- proszę nie używać wyrażeń regularnych.

Komponenty/opcje, które mogą okazać się przydatne: Bash; Powershell; linux: wget, cat, sed, awk, cut, psql, sort, uniq, diff, sdiff, mv, cp, wc, zip, unzip, mailx, echo, \$?, >, <; sql: CREATE TABLE IF EXISTS, ST_DistanceSpheroid