Database Explorer

Dokumentacja

Paweł Pollak 262772,
Karol Prusinowski 262773,
Karol Szczawiński 262782

1. Cel aplikacji

Celem aplikacji jest ułatwienie analizy zawartości baz danych. Głównym założeniem jest umożliwienie efektywnej, podglądowej analizy baz danych składających się z wielu tabel. Aplikacja powinna w łatwy sposób umożliwiać pobieranie najważniejszych informacji o tabelach i ich kolumnach.

2. Zakres aplikacji

2.1. Wymagania funkcjonalne

1. Aplikacja powinna umożliwiać wybranie silnika bazy danych, z którym użytkownik chce się połączyć.
2. Użytkownik powinien mieć możliwość uruchomienia aplikacji podając dane potrzebne do stworzenia pełnego ConnectionString, który zostanie wykorzystany do połączenia z bazą. Dane zostaną wprowadzone poprzez argumenty podane podczas uruchomienia aplikacji.

3. Za pomocą argumentów będzie można wybrać wersję generowanego raportu - podstawową lub rozszerzoną.
4. Dla każdej tabeli w bazie danych powinny być wygenerowane następujące informacje:
    - nazwa tabeli
    - liczba rekordów
    - liczba kolumn
5. Dla każdej kolumny w danej tabeli możliwe są dwa warianty generowanych danych:
    a. podstawowy:
        - nazwa kolumny
        - typ sql i ogólny (numeryczny, znakowy lub data)
        - dla danych numerycznych:
            - średnia
            - minimum
            - maksimum
        - dla danych tekstowych:
            - najczęstsze wartości
        - dla danych typu datetime:
            - minimum
            - maksimum
    b. rozszerzony:
        - to co w podstawowym zestawie
        - czy kolumna może przyjmować wartości puste, jeżeli tak to wyznaczony jest procent rekordów, które nie mają wartości
        - dla danych numerycznych:
            - liczba unikalnych wartości
            - kwantyle
        - dla danych tekstowych:
            - liczba unikalnych wartości (dla tekstów nie dłuższych niż 20 znaków)
        - dla danych typu datatime:
            - liczba unikalnych wartości
6. Dla pobranych danych powinien być generowany raport w postaci strony HTML:
    a. prezentacja statystyk dla tabel i ich kolumn
    b. wyszukiwanie:
        i. po nazwie tabeli
        ii. po nazwie kolumny
        iii. po wartościach statystyk

2.2. Wymagania niefunkcjonalne

- Aplikacja powinna umożliwiać połączenie z każdym z następujących silników baz danych: Redshift, Postgress, Mysql i Teradata.
- Aplikacja powinna działać na systemie Ubuntu w wersji co najmniej 16.04.

- Do generowania informacji oraz wyliczania analiz nie powinna bezpośrednio pobierać wszystkich danych z bazy. Wszelkie operacje powinny być wykonane zdalnie przez silnik bazodanowy.
- Generowany raport powinien w prosty sposób przedstawiać pobrane dane, aby umożliwić ich sprawną analizę.
- Raport powinien być wygenerowany w języku angielskim.

3. Opis technologiczny

Aplikacja będzie składać się z dwóch części:

1. Skrypt napisany w języku Python (wymagana jest co najmniej wersja 3.6), który będzie się łączyć z wybranym silnikiem i pobierać dane, z których wygeneruje raport. Do połączenia z bazą danych teradata wymagane jest zainstalowanie sterownika odbc (można pobrać go ze strony: https://downloads.teradata.com/download/connectivity).
   Do działania aplikacja wykorzystuje następujące biblioteki:
   a. pyodbc
   b. setuptools
   c. typing
   d. PyMySQL
   e. psycopg2
   f. simplejson
2. Raport generowany jako strona napisana w HTML i CSS. Do obsługi wyszukiwania i filtrowania użyty będzie JavaScript.

4. Prototyp wyglądu HTML

Prototyp wyglądu generowanego raportu w postaci HTML dla wersji podstawowej i rozszerzonej jest zamieszczony w osobnym pliku.

5. Testy akceptacyjne

Testy akceptacyjne będą się składać z następujących faz, które będą się powtarzać dla każdego silnika bazodanowego:

1. Uruchomienie aplikacji dla dwóch różnych baz danych i dla dwóch różnych schematów dla każdej bazy.
2. Dla każdego pliku wynikowego sprawdzenie kompletności wygenerowanych danych:
   a. liczby wszystkich tabel w wskazanym schemacie,
   b. dla pięciu losowych tabel sprawdzenie poprawności nazwy tabeli oraz liczby, nazw i typów kolumn w każdej tabeli.
3. Dla każdego raportu pięć losowych wyszukiwań po każdym dostępnym parametrze i zweryfikowanie poprawności w bazie danych.

4. Sprawdzenie poprawności wygenerowanych danych sześciu kolumn po dwie dla każdego rodzaju (numeryczna, data i tekstowa) dla obu rodzajów raportów.

6. Instrukcja obsługi

Po ściągnięciu kodu źródłowego należy przejść do katalogu, w którym się on znajduje i wywołać polecenie: “`pip install .`”. Następnie w celu uruchomienia programu należy uruchomić polecenie `dbexplorer` z odpowiednimi argumentami:

- `-e` lub `--extended` — generowanie raportu w wersji rozszerzonej (domyślnie generowany jest w wersji podstawowej), parametr nie przyjmuje wartości,
- `-s` lub `--server` — adres serwera bazodanowego,
- `-p` lub `--port` — port serwera bazodanowego,
- `-n` lub `--database_name` — nazwa bazy danych, dla której ma być wygenerowany raport,
- -u lub `--user` — nazwa użytkownika,
- `-pass` lub `--password`— hasło dla wskazanego użytkownika,
- `-t` lub `--database_type` — typ bazy danych (`Redshift`, `Postgress`, `Mysql` lub `Teradata`),
- `-o` lub `--output` — nazwa pliku wyjściowego z raportem,
- `-sc` lub `--schema` — schema, dla której ma być wygenerowany raport (tylko dla postgres),
- `-d` lub `--odbc_driver` — nazwa sterownika odbc dla połączenia z baza danych Teradata.

Przykładowe wywołanie programu dla bazy:

- Postgres: `dbexplorer -s 192.2.3.4 -p 5432 -n dvdrental -u dbadmin -pass password -t postgres -o out.html`,
- Teradata: `dbexplorer -e -t teradata -s 192.168.44.128 -u dbc -n sample1 -pass dbc -o test.html -d 'Teradata Database ODBC Driver 16.20'`