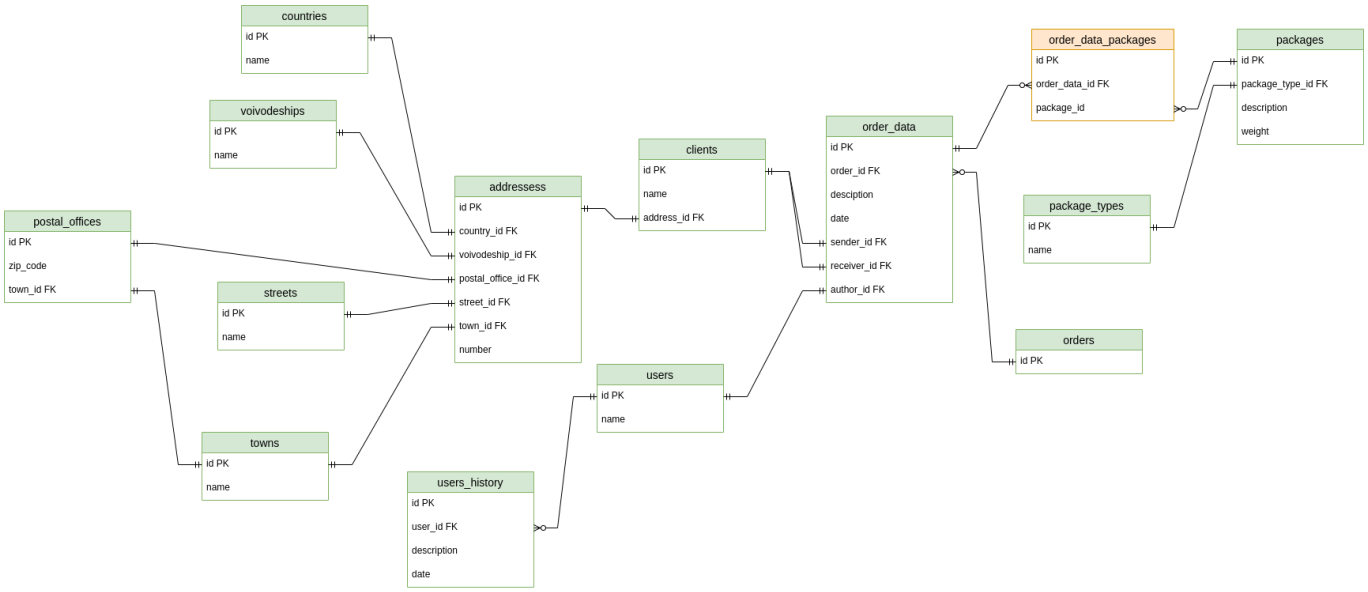
Relacyjne systemy zarządzania bazami danych

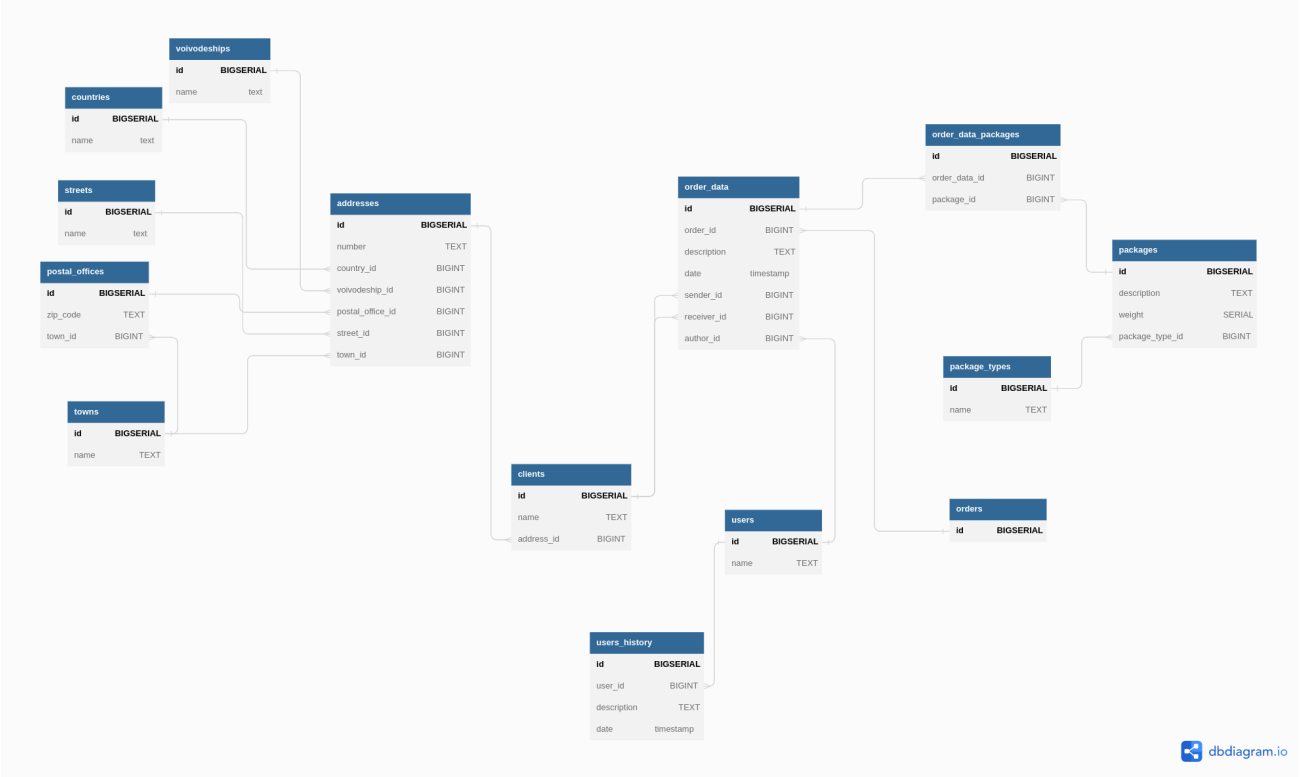
Jan Łukomski 291089
Informatyka Stosowana,
Studia Magisterskie,
Wydział Elektryczny

Sprawozdanie z projektu 2

Bazy danych



Pominięte w nim zostały typy danych, żeby nie zaciemniały obrazu nadmiarem informacji. Schemat z typami danych zaprezentowany jest poniżej.



Funkcje, procedury, wyzwalacze

Dla zliczania liczby dokonanych zmian przez wybranego użytkownika w zamówieniu służy funkcja count_changes(order_id, user_id).

```
drop function if exists count_changes(integer, integer);
create function count_changes(order_id integer, user_id integer) returns integer as $$
    select count(*) from order_data as od where od.order_id=order_id and od.author_id = user_id;
$$ language sql;      You, 4 days ago • configure cron database dumps
```

Przygotowałem procedurę do aktualizacji zamówienia. Zgodnie z architekturą bazy - zaktualizowanie polega na dodaniu nowego rekordu do tabeli order_data oraz podpięcie odpowiednich elementów z tabeli packages.

```
create or replace procedure update_order(order_id integer, description text, sender_id integer, receiver_id integer, author_id integer, package_ids
integer[]) AS $$
    declare
        package_id integer;
        order_data_id bigint;
    begin
        insert into order_data (order_id, description, sender_id, receiver_id, author_id, date) values (order_id, description, sender_id, receiver_id,
        author_id, now()) returning id into order_data_id;
        raise NOTICE 'create order_data_id: %',order_data_id;
        foreach package_id in array package_ids
        loop
            insert into order_data_packages(order_data_id, package_id) values (order_data_id, package_id);
        end loop;
    end;
$$ language plpgsql;
```

Przykładowy wyzwalacz służący do generowania tablicy z logami dotyczącymi zmian w tabeli users.

```
create or replace function update_users() returns trigger as $$
    declare
        description text = 'Name changed from ' || old.name || ' to ' || new.name;
    begin
    insert into users_history( date, user_id, description) values (now(), old.id, description);
    return NEW;
    end;
$$ language plpgsql;

drop trigger if exists log_users_update on users;      You, 4 days ago • configure cron database
create trigger log_users_update
    after update on users
    FOR EACH ROW
    WHEN (OLD.* IS DISTINCT FROM NEW.*)
    EXECUTE FUNCTION update_users();
```

Widoki

Dane z tabel mogą być przekazywane użytkownikom za pomocą widoków. Przedstawiam podstawowe bardziej skomplikowane zapytanie zapakowane w widoki. Dotyczy ono wyciągnięcia obecnych zamówień (z ostatnią datą modyfikacji).

```
drop view if exists last_order_data_id_for_order_id;
create view last_order_data_id_for_order_id as
    select od.order_id, (select odi.id from order_data as odi where odi.order_id = od.order_id order by
    odi.date desc limit 1) as order_data_id from order_data as od
        group by od.order_id
        order by od.order_id
;

drop view if exists orders_status;
create view orders_status as
    select o.id as order_id, od.id as order_data_id, od.description, od.date, cs.name as sender_name, cr.
    name as receiver_name, u.name as worker from orders as o
        join last_order_data_id_for_order_id as l on l.order_id = o.id
        join order_data as od on l.order_data_id = od.id
        join clients as cs on cs.id = od.sender_id
        join clients as cr on cr.id = od.receiver_id
        join users as u on u.id = od.author_id
;       You, 5 days ago • add views to select current orders status …
```

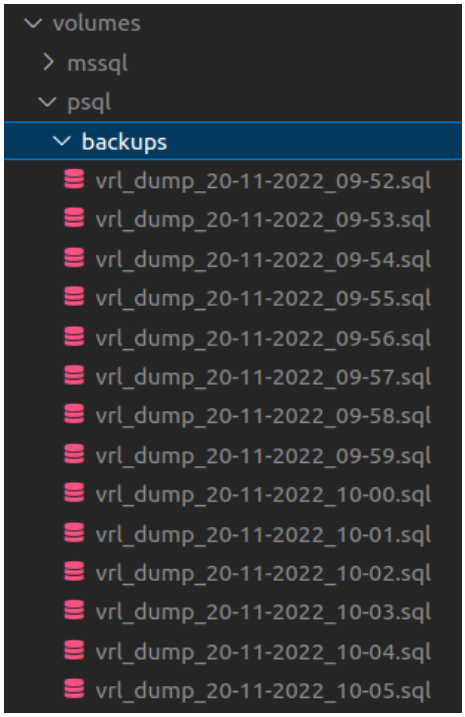
Pierwszy widok przygotowuje tabelę z ID najnowszych order_data. Jak widać nie jest to trywialne zapytanie, dlatego zdecydowałem się opakować go w widok. Jest ono używane w drugim widoku, które zwraca połączone dane o aktualnych zamówieniach.

Tworzenie kopii zapasowej bazy

Najlepiej jak kopia zapasowa wykonuje się automatycznie, dlatego przygotowałem skrypt zapisujący zrzut bazy. Skrypt ten jest uruchamiany co minutę (w realnym świecie mogłoby to być codziennie np).

```
#!/bin/sh

now=$(date +"%d-%m-%Y_%H-%M")
pg_dump -U postgres vrl  > "/backups/vrl_dump_$now.sql"

exit 0      You, 4 days ago • configure cron database dumps
```

Pliki poprawnie się generują i można ich użyć do przywrócenia bazy.



W celu uruchomienia crona na dockerze postawionym z obrazu postgresa musiałem zainstalować crona oraz zmodyfikować skrypt docker-entrypoint.sh dodając w nim linijkę go uruchamiającą.

```
# Run cron service on init
service cron start
```

Migracja bazy do MsSQL

Migracja odbywa się na podstawie pliku zrzutu bazy PostgresSQL.

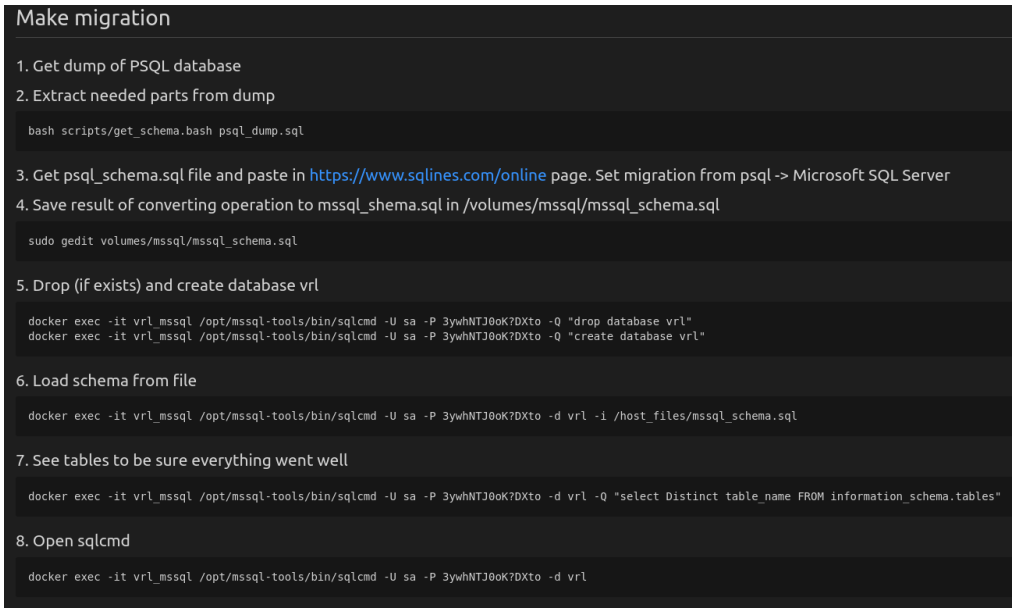
Jako, że nie wszystkie elementy znajdujące się w zrzucie PSQL mogą być łatwo przetworzone na MsSQL, dlatego przygotowałem skrypt wyciągający potrzebne informacje ze zrzutu bazy.

```bash
#!/bin/bash

if [[ $# -eq 0 ]]; then
    echo 'Need parameter with path to PSQL database dump'
    exit 1
fi

dump_file="$1"
outputfile=psql_schema.sql
echo $dump_file
cat $dump_file | awk 'RS="";/CREATE TABLE[^;]*;/' > $outputfile
cat $dump_file | awk 'RS="";/ALTER TABLE ONLY .* PRIMARY KEY[^;]*;/' >> $outputfile
cat $dump_file | awk 'RS="";/ALTER TABLE ONLY .* FOREIGN KEY[^;]*;/' >> $outputfile
sed -i 's/public\.//' $outputfile;
sed -i 's/ONLY //' $outputfile;

echo 'Go to "https://www.sqlines.com/online" and paste there "'$outputfile'" file.'
```

W następnym kroku - wygenerowany plik (z wyciętymi interesującymi nas fragmentami (oraz wycinający przeszkadzające słowa) wklejamy do convertera online: `https://www.sqlines.com/online`

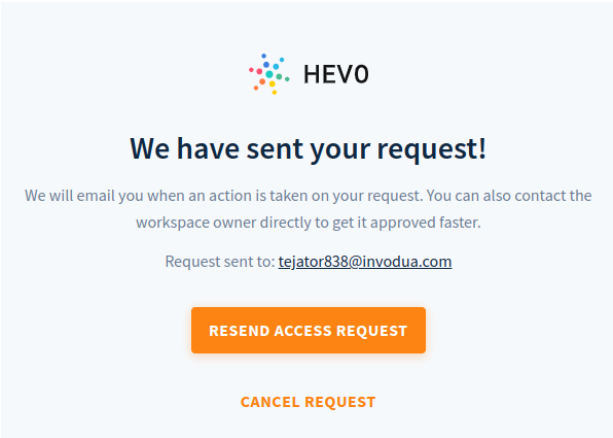
W converterze ustawiałem konwersje z PostgresSQL na Microsoft SQL.

Wygenerowany plik wklejałem do pliku znajdującego się w kontenerze uruchiononego dockera z mssql, następnie zgodnie z opisem Migration.md.



Różnice bazy danych

Próbowałem do tego początkowo wykorzystać narzędzie ETL.
Zarejestrowałem się na Hevo Data. Zatrzymałem się na kroku oczekiwania na potwierdzenie nadania dostępu.



Ostatecznie zdecydowałem się to robić przy użyciu narzędzia sqlines.

Podczas robienia migracji napotkałem wiele różnic między bazami. Jeden po drugim starałem się je rozwiązać. Przeszkadzały przestrzenie “public” przed nazwami tabel.

Inny słówkiem, które należało wyrzucić z pliku było “ONLY” przy poleceniu: “ALTER TABLE ONLY public.addresses ADD CONSTRAINT addresses_pkey PRIMARY KEY (id);”

Nazwy ograniczeń też musiały być unikatowe (w psql nie było problemu z powtarzającymi się nazwami) np.”ALTER TABLE ONLY public.addresses ADD CONSTRAINT addresses_pkey PRIMARY KEY (id);”