

Politechnika Świętokrzyska w Kielcach

Wydział Elektrotechniki, Automatyki i Informatyki

Katedra Informatyki, Elektroniki i Elektrotechniki

Kierunek Informatyka	Laboratorium nr 6. Bazy Danych 2	
Grupa dziekańska 2ID12A	Temat ćwiczenia <ul style="list-style-type: none">● Bazy danych NOSQL Neo4j – Zapytania złożone dla projektu	Wykonał: Patryk Grzywacz Dominik Grudzień
Data wykonania 31.05.2021r	Data oddania 31.05.2021r	Ocena i podpis

Temat: Sieć Gabinetów Lekarskich



1) Opis tabel

- Adresy – Przechowuje dane dotyczące adresu fizycznego domu/budynku/placówki etc.
- Kontakty – Przechowuje dane odnośnie możliwości kontaktowych zarówno osób jak i placówek medycznych.
- Specjalności – Przechowuje dane o specjalnościach lekarzy.
- Uprawnienia – Znajdują się w niej dane o uprawnieniach przypisywanych do stanowisk.
- Stanowiska – Przechowują dane odnośnie stanowisk przydzielanych do poszczególnych pracowników.
- Pracownicy – Przechowują dane osobiste jak i firmowe o pracownikach , a także odnośniki do adresów, kontaktów, stanowisk oraz ewentualnie do specjalności.
- Placówki – Zawierają dane o placówkach medycznych ,adresie oraz kontakcie z nimi.

- Gabinety – Zawierają dane o gabinetach lekarskich mieszczących się w placówkach ,do których przypisany jest odpowiedni pracownik, najczęściej jakiś lekarz.
- Karty – Przechowują dane osobowe tj. data urodzenia czy też grupę krwi.
- Pacjenci – Przechowują dane osobowe jak i klienckie, odnośniki do kart ,adresów oraz kontaktów.
- Choroby – Przechowują dane odnośnie choroby/chorób przypisanych do danej Karty pacjenta
- Ulgi – Zawierają dane ulg przeznaczonych do wykorzystania przy receptach.
- Pozycje_Recept – Zawierają dane materialne odnośnie leków przepisanych w ramach danej recepty.
- Oddzialy_NFZ – Zawierają dane specjalistyczne wymagane na każdej receptce.
- Statusy_Wizyt – Zawierają dane określające status danej wizyty.
- Wizyty – Przechowuje dane o wizytach pacjentów ,o dacie i czasie trwania, miejscu, opłacie a także kto umówił/przyjmował pacjenta.
- Recepty – Zawiera same odnośniki do pracownika ,który ją wystawił , do wizyty podczas ,której została wystawiona a także do choroby , ulgi i oddziału nfz.
- Zabiegi – Zawiera dane o zabiegu wykonanym w ramach danej wizyty , jego cenie oraz pracownika, który go przeprowadzał.

2. Napisane skrypty (Wszystkie skrypty będą w osobnych plikach .cyp wraz z sprawozdaniem , a także pliki .csv do zasilenia bazy danymi)

1) Skrypty na utworzenie bazy grafowej (węzłów i relacji)

```

MATCH(n:Adresy) DETACH DELETE n;
MATCH(n:Choroby) DETACH DELETE n;
MATCH(n:Gabinety) DETACH DELETE n;
MATCH(n:Karty) DETACH DELETE n;
MATCH(n:Kontakty) DETACH DELETE n;
MATCH(n:Oddzialy_nfz) DETACH DELETE n;
MATCH(n:Pacjenci) DETACH DELETE n;
MATCH(n:Placowki) DETACH DELETE n;
MATCH(n:Pozycje_recept)-[r]-() DETACH DELETE r;
MATCH(n:Pozycje_recept) DETACH DELETE n;
MATCH(n:Recepty) DETACH DELETE n;
MATCH(n:Pracownicy) DETACH DELETE n;
MATCH(n:Specjalnosci) DETACH DELETE n;
MATCH(n:Stanowiska) DETACH DELETE n;
MATCH(n:Statusy_wizyt) DETACH DELETE n;
MATCH(n:Ulgi) DETACH DELETE n;
MATCH(n:Uprawnienia) DETACH DELETE n;
MATCH(n:Wizyty) DETACH DELETE n;
MATCH(n:Zabiegi) DETACH DELETE n;
//Rekordy w tabelach
LOAD CSV WITH HEADERS FROM 'file:///ulgi.csv' AS row
CREATE (:Ulgi{ulgi_id:toInteger(row.ulgi_id),typ_ulgi:row.typ_ulgi,procent_ulgi:toInteger(row.procent_ulgi)});

LOAD CSV WITH HEADERS FROM 'file:///oddzialy_nfz.csv' AS row

```

```

CREATE (:Oddzialy_nfz{oddzial_nfz_id:toInteger(row.oddzial_nfz_id),nazwa:row.nazwa,kod_funduszu:row.kod_funduszu});

LOAD CSV WITH HEADERS FROM 'file:///adresy.csv' AS row
CREATE (:Adresy {adres_id:toInteger(row.adres_id),kod_poczt:row.kod_poczt,miasto:row.miasto,wojewodztwo:row.wojewodztwo,ulica:row.ulica,nr_domu:toInteger(row.nr_domu),nr_mieszkania:toInteger(row.nr_mieszkania)});

LOAD CSV WITH HEADERS FROM 'file:///kontakty.csv' AS row
CREATE (:Kontakty{kontakt_id:toInteger(row.kontakt_id),telefon:row.telefon,email:row.email});

LOAD CSV WITH HEADERS FROM 'file:///specjalnosci.csv' AS row
CREATE (:Specjalnosci{specjalnosc_id:toInteger(row.specjalnosc_id),nazwa:row.nazwa,stopien:row.stopien,dodatek_pensja:toInteger(row.dodatek_pensja)});

LOAD CSV WITH HEADERS FROM 'file:///uprawnienia.csv' AS row
CREATE (:Uprawnienia{uprawnienie_id:toInteger(row.uprawnienie_id),oznaczenie:row.oznaczenie,opis:row.opis});

LOAD CSV WITH HEADERS FROM 'file:///stanowiska.csv' AS row
CREATE (:Stanowiska{stanowisko_id:toInteger(row.stanowisko_id),nazwa:row.nazwa,pensja:toInteger(row.pensja),uprawnienie_id:toInteger(row.uprawnienie_id)});

LOAD CSV WITH HEADERS FROM 'file:///karty.csv' AS row
CREATE (:Karty{pesel_id:row.pesel_id,data_ur:row.data_ur,grupa_krwi:row.grupa_krwi});

LOAD CSV WITH HEADERS FROM 'file:///choroby.csv' AS row FIELDTERMINATOR '*'
CREATE (:Choroby{choroby_id:toInteger(row.choroby_id),nazwa:row.nazwa,opis:row.opis,początek:row.początek,koniec:row.koniec,pesel_id:row.pesel_id});

LOAD CSV WITH HEADERS FROM 'file:///placowki.csv' AS row
CREATE (:Placowki{placowka_id:toInteger(row.placowka_id),nazwa:row.nazwa,adres_id:toInteger(row.adres_id),kontakt_id:toInteger(row.kontakt_id)});

LOAD CSV WITH HEADERS FROM 'file:///pracownicy.csv' AS row
CREATE (:Pracownicy{pracownik_id:toInteger(row.pracownik_id),imie:row.imie,nazwisko:row.nazwisko,login:row.login,haslo:row.haslo,pensja:toInteger(row.pensja),adres_id:toInteger(row.adres_id),kontakt_id:toInteger(row.kontakt_id),stanowisko_id:toInteger(row.stanowisko_id),specjalnosc_id:toInteger(row.specjalnosc_id)});

LOAD CSV WITH HEADERS FROM 'file:///pacjenci.csv' AS row
CREATE (:Pacjenci{pacjent_id:toInteger(row.pacjent_id),imie:row.imie,nazwisko:row.nazwisko,login:row.login,haslo:row.haslo,pesel_id:row.pesel_id,kontakt_id:toInteger(row.kontakt_id),adres_id:toInteger(row.adres_id)});

LOAD CSV WITH HEADERS FROM 'file:///gabinety.csv' AS row
CREATE (:Gabinety{gabinet_id:toInteger(row.gabinet_id),oznaczenie:row.oznaczenie,pracownik_id:toInteger(row.pracownik_id),kontakt_id:toInteger(row.kontakt_id),placowka_id:toInteger(row.placowka_id)});

LOAD CSV WITH HEADERS FROM 'file:///wizyty.csv' AS row

```

```

CREATE (:Wizyty{wizyta_id:toInteger(row.wizyta_id),opлата:toInteger(row.opлата),data_wizyty:row.data_wizyty,godzina_poczatek:row.godzina_poczatek,godzina_koniec:row.godzina_koniec,pacjent_id:toInteger(row.pacjent_id),prac_spec:toInteger(row.prac_spec),prac_uma:toInteger(row.prac_uma),gabinet_id:toInteger(row.gabinet_id)});

LOAD CSV WITH HEADERS FROM 'file:///recepty.csv' AS row
CREATE (:Recepty{recepta_id:toInteger(row.recepta_id),pracownik_id:toInteger(row.pracownik_id),wizyta_id:toInteger(row.wizyta_id),oddzial_nfz_id:toInteger(row.oddzial_nfz_id),recepta_choroba_id:toInteger(row.recepta_choroba_id),ulga_id:toInteger(row.ulga_id)});

LOAD CSV WITH HEADERS FROM 'file:///pozycje_recept.csv' AS row
CREATE (:Pozycje_recept{pozycje_recept_id:toInteger(row.pozycje_recept_id),recepta_id:toInteger(row.recepta_id),nazwa:row.nazwa,ilosc:toInteger(row.ilosc),odplatnosc:toFloat(row.odplatnosc)});

LOAD CSV WITH HEADERS FROM 'file:///statusy_wizyt.csv' AS row
CREATE (:Statusy_wizyt{statusy_wizyt_id:toInteger(row.statusy_wizyt_id),status:row.status,opis:row.opis});

LOAD CSV WITH HEADERS FROM 'file:///zabiegi.csv' AS row
CREATE (:Zabiegi{zabieg_id:toInteger(row.zabieg_id),nazwa:row.nazwa,cena_netto:toInteger(row.cena_netto),pracownik_id:toInteger(row.pracownik_id),wizyta_id:toInteger(row.wizyta_id)});

//Relacje
MATCH(s:Stanowiska) MATCH(u:Uprawnienia)
WHERE u.uprawnienie_id = s.uprawnienie_id
CREATE (s)-[:fk_Stanowiska_Uprawnienia]->(u);

MATCH(c:Choroby) MATCH(k:Karty)
WHERE c.pesel_id = k.pesel_id
CREATE (c)-[:fk_Choroby_Karty]->(k);

MATCH(p:Placowki)MATCH(a:Adresy)MATCH(k:Kontakty)
WHERE p.adres_id = a.adres_id AND p.kontakt_id = k.kontakt_id
CREATE (p)-[:fk_Placowki_Adresy]->(a),(p)-[:fk_Placowki_Kontakty]->(k);

MATCH(p:Pracownicy)MATCH(a:Adresy)MATCH(k:Kontakty)MATCH(st:Stanowiska)
WHERE p.adres_id = a.adres_id AND p.kontakt_id = k.kontakt_id AND p.stanowisko_id = st.stanowisko_id
CREATE (p)-[:fk_Pracownicy_Adresy]->(a),(p)-[:fk_Pracownicy_Kontakty]->(k),(p)-[:fk_Pracownicy_Stanowiska]->(st);

MATCH(p:Pracownicy)MATCH(sp:Specjalnosci)
WHERE p.specjalnosc_id = sp.specjalnosc_id
CREATE (p)-[:fk_Pracownicy_Specjalnosci]->(sp);

MATCH(p:Pacjenci)MATCH(a:Adresy)MATCH(ko:Kontakty)MATCH(ka:Karty)
WHERE p.adres_id = a.adres_id AND p.kontakt_id = ko.kontakt_id AND p.pesel_id = ka.pesel_id
CREATE (p)-[:fk_Pacjenci_Adresy]->(a),(p)-[:fk_Pacjenci_Kontakty]->(ko),(p)-[:fk_Pacjenci_Karty]->(ka);

```

```

MATCH(g:Gabinety)MATCH(p:Pracownicy)MATCH(k:Kontakty)MATCH(pl:Placowki)
WHERE g.kontakt_id = k.kontakt_id AND g.pracownik_id = p.pracownik_id AND g.placowka_id = pl.placowka_id
CREATE (g)-[:fk_Gabinety_Kontakty]->(k),(g)-[:fk_Gabinety_Pracownicy]->(p),(g)-[:fk_Gabinety_Placowki]->(pl);

MATCH(w:Wizyty)MATCH(g:Gabinety)MATCH(ps:Pracownicy)MATCH(pu:Pracownicy)MATCH(pac:Pacjenci)
WHERE w.gabinet_id = g.gabinet_id AND w.prac_spec = ps.pracownik_id AND w.prac_uma = pu.pracownik_id AND w.pacjent_id = pac.pacjent_id
CREATE (w)-[:fk_Wizyty_Gabinety]->(g),(w)-[:fk_Wizyty_Pacjenci]->(pac),(w)-[:fk_Wizyty_Pracownicy_Spec]->(ps),
(w)-[:fk_Wizyty_Pracownicy_Uma]->(pu);

MATCH(r:Recepty)MATCH(p:Pracownicy)MATCH(w:Wizyty)MATCH(o:Oddzialy_nfz)MATCH(c:Choroby)
WHERE r.pracownik_id = p.pracownik_id AND r.wizyta_id = w.wizyta_id AND r.oddzial_nfz_id = o.oddzial_nfz_id AND
r.recepta_choroba_id = c.choroby_id
CREATE (r)-[:fk_Recepty_Pracownicy]->(p),(r)-[:fk_Recepty_Wizyty]->(w),(r)-[:fk_Recepty_Oddzialy]->(o),(r)-[:fk_Recepty_Choroby]->(c);

MATCH(r:Recepty)MATCH(u:Ulgi)
WHERE r.ulga_id = u.ulgi_id
CREATE (r)-[:fk_Recepty_Ulgi]->(u);

MATCH(pr:Pozycje_recept)MATCH(r:Recepty)
WHERE pr.recepta_id = r.recepta_id
CREATE (pr)-[:fk_Pozycje_Recepty]->(r);

MATCH(w:Wizyty)MATCH(s:Statusy_wizyt)
WHERE w.wizyta_id = s.statusy_wizyt_id
CREATE (s)-[:fk_Statusy_Wizyty]->(w);

MATCH(z:Zabiegi)MATCH(w:Wizyty)MATCH(p:Pracownicy)
WHERE z.wizyta_id = w.wizyta_id AND z.pracownik_id = p.pracownik_id
CREATE (z)-[:fk_Zabiegi_Wizyty]->(w),(z)-[:fk_Zabiegi_Pracownicy]->(p);

```

2) Skrypty zapytań

```
//Łączna wartość roczna za każdy zabieg przeprowadzany na pacjentach z grupą krwi
A-
MATCH(z:Zabiegi)-[:fk_Zabiegi_Wizyty]->(w:Wizyty)-[:fk_Wizyty_Pacjenci]-
>(p:Pacjenci)-[:fk_Pacjenci_Karty]->(k:Karty)
WHERE k.grupa_krwi = "A-"
RETURN z.nazwa,date(w.data_wizyty).year as rok,k.grupa_krwi,
SUM(z.cena_netto) AS SumaZaZabiegi
ORDER BY z.nazwa,rok DESC;

//Id pacjenta,jego imie ,nazwisko , pesel oraz jego Wydatki na leki wciągu jedneg
o roku z ulgą/bez ulgi
MATCH(r:Recepty)-[:fk_Recepty_Wizyty]->(w:Wizyty)-[:fk_Wizyty_Pacjenci]-
>(p:Pacjenci)
MATCH(pz:Pozycje_recept)-[:fk_Pozycje_Recepty]->(r)
OPTIONAL MATCH(r)-[:fk_Recepty_Ulgi]->(u:Ulgi)
RETURN p.pacjent_id ,p.imie ,p.nazwisko,p.pesel_id,date(w.data_wizyty).year AS ro
k,
SUM(pz.odplatnosc) AS WydatkiNaLekiBezUlg,(SUM(pz.odplatnosc) * u.procent_ulgi /
100) AS WydatkiNaLeki_z_Ulgą ORDER BY p.pacjent_id ASC;

//Summaryczna kwota za wizyty w danej placówce w danym roku
MATCH(w:Wizyty)-[:fk_Wizyty_Gabinety]->(g:Gabinety)-[:fk_Gabinety_Placowki]-
>(p:Placowki)-[:fk_Placowki_Adresy]->(a:Adresy)
call{
    WITH p,a,w
    RETURN p.placowka_id AS placowka_id, p.nazwa AS placowka,a.miasto AS miasto,d
ate(w.data_wizyty).year AS rok,SUM(w.oplata) AS Suma_w_DanymRoku
}
RETURN placowka_id,placowka,miasto,rok,SUM(Suma_w_DanymRoku) AS Suma_w_DanymRoku
ORDER BY placowka_id ASC,rok DESC;

//Średnia opłat za wizyty w każdym roku, pacjentów pochodzących z danego miasta
MATCH (w:Wizyty)-[:fk_Wizyty_Pacjenci]->(p:Pacjenci)-[:fk_Pacjenci_Adresy]-
>(a:Adresy)
RETURN date(w.data_wizyty).year AS rok, a.miasto, avg(w.oplata) ORDER BY rok

//Suma dochodów z zabiegów w każdym gabinecie
MATCH (z:Zabiegi)-[:fk_Zabiegi_Wizyty]->(w:Wizyty)-[:fk_Wizyty_Gabinety]-
>(g:Gabinety)
RETURN w.gabinet_id AS identyfikator_gabinetu, sum(z.cena_netto)
ORDER BY identyfikator_gabinetu

//Suma opłat za leki z każdej recepty, w danym roku
MATCH (pr:Pozycje_recept)-[:fk_Pozycje_Recepty]->(r:Recepty)-
[:fk_Recepty_Wizyty]->(w:Wizyty)
RETURN pr.recepta_id AS id_recepty, date(w.data_wizyty).year AS rok, sum(pr.odpla
tnosc) AS suma ORDER BY rok
```

3. Wyniki wykonania skryptów

1)

Database Information

Use database

neo4j - default

Node Labels

"(1,064,056)"

Adresy

Choroby

Gabinety

Karty

Kontakty

Oddzialy_nfz

Pacjenci

Placowki

Pozycje_recept

Pracownicy

Recepty

Specjalnosci

Stanowiska

Statusy_wizyt

Ulg

Uprawnienia

Wizyty

Zabiegi

Relationship Types

"(1,182,715)"

fk_Choroby_Karty

fk_Gabinety_Kontakty

fk_Gabinety_Placowki

fk_Gabinety_Pracownicy

fk_Pacjenci_Adresy

fk_Pacjenci_Karty

fk_Pacjenci_Kontakty

fk_Placowki_Adresy

neo4j\$

```
1 //łączna wartość roczna za każdy zabieg przeprowadzany na pacjentach z grupą krwi A-
2 MATCH(z:Zabiegi)-[:fk_Zabiegi_Wizyty]-(w:Wizyty)-[:fk_Wizyty_Pacjenci]-(p:Pacjenci)-[:fk_Pacjenci_Karty]-(k:Karty)
3 WHERE k.grupa_krwi = "A-"
4 RETURN z.nazwa,date(w.data_wizyty).year as rok,k.grupa_krwi,
5 SUM(z.cena_netto) AS SumaZaZabiegi
6 ORDER BY z.nazwa,rok DESC;
```

Table

Text

Code

	z.nazwa	rok	k.grupa_krwi	SumaZaZabiegi
1	"abdominoplastyka"	2022	"A-"	13500
2	"abdominoplastyka"	2020	"A-"	3000
3	"abdominoplastyka"	2019	"A-"	10000
4	"abdominoplastyka"	2018	"A-"	6000
5	"abdominoplastyka"	2016	"A-"	6000
6	"abdominoplastyka"	2014	"A-"	2500
7

Started streaming 1748 records after 22 ms and completed after 25 ms, displaying first 1000 rows.

2)

Database Information

Use database

neo4j - default

Node Labels

"(1,064,056)"

Adresy

Choroby

Gabinety

Karty

Kontakty

Oddzialy_nfz

Pacjenci

Placowki

Pozycje_recept

Pracownicy

Recepty

Specjalnosci

Stanowiska

Statusy_wizyt

Ulg

Uprawnienia

Wizyty

Zabiegi

Relationship Types

"(1,182,715)"

fk_Choroby_Karty

fk_Gabinety_Kontakty

fk_Gabinety_Placowki

fk_Gabinety_Pracownicy

fk_Pacjenci_Adresy

fk_Pacjenci_Karty

fk_Pacjenci_Kontakty

fk_Placowki_Adresy

neo4j\$

```
1 //Id pacjenta,jego imie ,nazwisko , pesel oraz jego Wydatki na leki wciagu jednego roku z ulga/bez ulgi
2 MATCH(r:Recepty)-[:fk_Recepty_Wizyty]-(w:Wizyty)-[:fk_Wizyty_Pacjenci]-(p:Pacjenci)
3 MATCH(pz:Pozycje_recept)-[:fk_Pozycje_Recepty]-(r)
4 OPTIONAL MATCH(r)-[:fk_Recepty_Ulgi]-(u:Ulg)
5 RETURN p.pacjent_id ,p.imie ,p.nazwisko,p.pesel_id,date(w.data_wizyty).year AS rok,
6 SUM(pz.odplatnosc) AS WydatkiNaLekiBezUlg,(SUM(pz.odplatnosc) * u.procent_ulgi / 100) AS WydatkiNaLeki_z_Ulgą
ORDER BY p.pacjent_id ASC;
```

Table

Text

Code

	p.pacjent_id	p.imie	p.nazwisko	p.pesel_id	rok	WydatkiNaLekiBezUlg	WydatkiNaLeki_z_Ulgą
1	1	"Emil"	"Brzezinski"	"13088027337"	2021	1217.96	null
2	1	"Emil"	"Brzezinski"	"13088027337"	2015	310.48	null
3	1	"Emil"	"Brzezinski"	"13088027337"	2008	44.99	null
4	1	"Emil"	"Brzezinski"	"13088027337"	2017	874.78	null
5	1	"Emil"	"Brzezinski"	"13088027337"	2006	224.95000000000002	null
6	1	"Emil"	"Brzezinski"	"13088027337"	2012	190.8	null
7

Started streaming 61288 records after 14 ms and completed after 18 ms, displaying first 1000 rows.

3)

Database Information

Use database

neo4j - default

Node Labels

{1,064,056}

Adresy

Choroby

Gabinety

Karty

Kontakty

Oddzialy_nfz

Pacjenci

Placowki

Pozycje_recept

Pracownicy

Recepty

Specjalnosci

Stanowiska

Statusy_wizyt

Uligi

Uprawnienia

Wizyty

Zabiegi

Relationship Types

{1,182,715}

fk_Choroby_Karty

fk_Gabinety_Kontakty

fk_Gabinety_Placowki

fk_Gabinety_Pracownicy

fk_Pacjenci_Adresy

fk_Pacjenci_Karty

fk_Pacjenci_Kontakty

fk_Placowki_Adresy

fk_Placowki_Kontakty

fk_Pozycje_Recepty

neo4j\$

```

1 //Sumaryczna kwota za wizyty w danej placówce w danym roku
2 MATCH (w:Wizyty)-[:fk_Wizyty_Gabinety]-(g:Gabinety)-[:fk_Gabinety_Placowki]-(p:Placowki)-[:fk_Placowki_Adresy]-(a:Adresy)
3 call{
4   WITH p,a,w
5   RETURN p.placowka_id AS placowka_id, p.nazwa AS placowka,a.miasto AS miasto,date(w.data_wizyty).year AS
   rok,SUM(w.oplata) AS Suma_w_DanymRoku
6 }
7 RETURN placowka_id,placowka,miasto,rok,SUM(Suma_w_DanymRoku) AS Suma_w_DanymRoku
8 ORDER BY placowka_id ASC,rok DESC;

```

	placowka_id	placowka	miasto	rok	Suma_w_DanymRoku
1	1	"Centrum Medyczno-Diagnostyczne"	"Białystok"	2020	2100
2	1	"Centrum Medyczno-Diagnostyczne"	"Białystok"	2017	1300
3	1	"Centrum Medyczno-Diagnostyczne"	"Białystok"	2015	300
4	1	"Centrum Medyczno-Diagnostyczne"	"Białystok"	2013	1200
5	1	"Centrum Medyczno-Diagnostyczne"	"Białystok"	2009	1000
6	1	"Centrum Medyczno-Diagnostyczne"	"Białystok"	2007	1200
7	-	-	-	-	-

Started streaming 70570 records after 6 ms and completed after 13 ms, displaying first 1000 rows.

4)

Database Information

Use database

neo4j - default

Node Labels

{1,064,056}

Adresy

Choroby

Gabinety

Karty

Kontakty

Oddzialy_nfz

Pacjenci

Placowki

Pozycje_recept

Pracownicy

Recepty

Specjalnosci

Stanowiska

Statusy_wizyt

Uligi

Uprawnienia

Wizyty

Zabiegi

Relationship Types

{1,182,715}

fk_Choroby_Karty

fk_Gabinety_Kontakty

fk_Gabinety_Placowki

fk_Gabinety_Pracownicy

fk_Pacjenci_Adresy

neo4j\$

```

1 //Średnia opłat za wizyty w każdym roku, pacjentów pochodzących z danego miasta
2 MATCH (w:Wizyty)-[:fk_Wizyty_Pacjenci]-(p:Pacjenci)-[:fk_Pacjenci_Adresy]-(a:Adresy)
3 RETURN date(w.data_wizyty).year AS rok, a.miasto, avg(w.oplata) ORDER BY rok

```

	rok	a.miasto	avg(w.oplata)
1	1970	"Poznan"	775.0
2	1970	"Krakow"	1020.0
3	1970	"Gdansk"	933.3333333333334
4	1970	"Lublin"	583.3333333333334
5	1970	"Bydgoszcz"	425.0
6	1970	"Szczecin"	900.0
7	-	-	-

Started streaming 861 records after 8 ms and completed after 435 ms.

5)

Database Information

Use database

neo4j - default

Node Labels

*(1,064,056)

Adresy Choroby

Gabinety Karty Kontakty

Oddzialy_nfz Pacjenci Placowki

Pozycje_recept Pracownicy

Recepty Specjalnosci

Stanowiska Statusy_wizyt Ulgi

Uprawnienia Wizyty Zabiegi

Relationship Types

*(1,182,715)

fk_Choroby_Karty

fk_Gabinety_Kontakty

fk_Gabinety_Placowki

fk_Gabinety_Pracownicy

fk_Pacjenci_Adresy

fk_Pacjenci_Karty

neo4j\$

```

1 //Suma dochodów z zabiegów w każdym gabinecie
2 MATCH (z:Zabiegi)-[:fk_Zabiegi_Wizyty]-(w:Wizyty)-[:fk_Wizyty_Gabinety]-(g:Gabinety)
3 RETURN w.gabinet_id AS identyfikator_gabinetu, sum(z.cena_netto)
4 ORDER BY identyfikator_gabinetu

```

	identyfikator_gabinetu	sum(z.cena_netto)
1	11	15500
2	13	34000
3	14	32500
4	16	28000
5	29	29000
6	33	12500
7

Started streaming 2443 records after 11 ms and completed after 22 ms, displaying first 1000 rows.

6)

Database Information

Use database

neo4j - default

Node Labels

*(1,064,056)

Adresy Choroby

Gabinety Karty Kontakty

Oddzialy_nfz Pacjenci Placowki

Pozycje_recept Pracownicy

Recepty Specjalnosci

Stanowiska Statusy_wizyt Ulgi

Uprawnienia Wizyty Zabiegi

Relationship Types

*(1,182,715)

fk_Choroby_Karty

fk_Gabinety_Kontakty

fk_Gabinety_Placowki

fk_Gabinety_Pracownicy

fk_Pacjenci_Adresy

neo4j\$

```

1 //Suma opłat za leki z każdej recepty, w danym roku
2 MATCH (pr:Pozycje_recept)-[:fk_Pozycje_Recepty]-(r:Recepty)-[:fk_Recepty_Wizyty]-(w:Wizyty)
3 RETURN pr.recepta_id AS id_recepty, date(w.data_wizyty).year AS rok, sum(pr.odplatnosc) AS suma ORDER BY rok

```

	id_recepty	rok	suma
1	3052	1970	308.6
2	7626	1970	377.51
3	8327	1970	278.70000000000005
4	15585	1970	452.75
5	16318	1970	738.5500000000001
6	64727	1970	481.08
7

Started streaming 69805 records after 2 ms and completed after 15 ms, displaying first 1000 rows.

4. Graficzne przedstawienie bazy

1) Opis węzłów:

Każdy typ węzła przedstawia określoną tabelę z bazy danych. Dodatkowo węzły te noszą tę samą nazwę co dana tabela.

Adresy – węzeł zawierający pola z tabeli Adresy

Choroby - węzeł zawierający pola z tabeli Choroby

Gabinety - węzeł zawierający pola z tabeli Gabinety

Karty - węzeł zawierający pola z tabeli Karty

Kontakty - węzeł zawierający pola z tabeli Kontakty

Oddzialy_nfz - węzeł zawierający pola z tabeli Oddzialy_nfz

Pacjenci - węzeł zawierający pola z tabeli Pacjenci

Placowki - węzeł zawierający pola z tabeli Placowki

Pozycje_recept - węzeł zawierający pola z tabeli Pozycje_recept

Pracownicy - węzeł zawierający pola z tabeli Pracownicy

Recepty - węzeł zawierający pola z tabeli Recepty

Specjalnosci - węzeł zawierający pola z tabeli Specjalnosci

Stanowiska - węzeł zawierający pola z tabeli Stanowiska

Statusy_wizyt - węzeł zawierający pola z tabeli Statusy_wizyt

Ulgi - węzeł zawierający pola z tabeli Ulgi

Uprawnienia - węzeł zawierający pola z tabeli Uprawnienia

Wizyty - węzeł zawierający pola z tabeli Wizyty

Zabiegi - węzeł zawierający pola z tabeli Zabiegi

2) Opis relacji:

Każdy typ relacji odwzorowuje połączenie między dwoma tabelami za pomocą kluczy obcych i kluczy głównych. Klucze te zostały zawarte w polach danego węzła.

Fk_Stalowiska_Uprawnienia – połączenie klucza obcego węzła typu Stanowiska, z kluczem głównym węzła typu Uprawnienia

Fk_Choroby_Karty – połączenie klucza obcego węzła typu Choroby, z kluczem głównym węzła typu Karty

Fk_Placowki_Adresy - połączenie klucza obcego węzła typu Placowki, z kluczem głównym węzła typu Adresy

Fk_Placowki_Kontakty - połączenie klucza obcego węzła typu Placowki, z kluczem głównym węzła typu Kontakty

Fk_Pracownicy_Adresy - połączenie klucza obcego węzła typu Pracownicy, z kluczem głównym węzła typu Adresy

Fk_Pracownicy_Kontakty - połączenie klucza obcego węzła typu Pracownicy, z kluczem głównym węzła typu Kontakty

Fk_Pracownicy_Stalowiska - połączenie klucza obcego węzła typu Pracownicy, z kluczem głównym węzła typu Stanowiska

Fk_Pracownicy_Specjalnosci - połączenie klucza obcego węzła typu Pracownicy, z kluczem głównym węzła typu Specjalnosci

Fk_Pacjenci_Adresy - połączenie klucza obcego węzła typu Pacjenci, z kluczem głównym węzła typu Adresy

Fk_Pacjenci_Kontakty - połączenie klucza obcego węzła typu Pacjenci, z kluczem głównym węzła typu Kontakty

Fk_Pacjenci_Karty - połączenie klucza obcego węzła typu Pacjenci, z kluczem głównym węzła typu Karty

Fk_Gabinety_Kontakty – połączenie klucza obcego węzła typu Gabinety, z kluczem głównym węzła typu Kontakty

Fk_Gabinety_Pracownicy - połączenie klucza obcego węzła typu Gabinety, z kluczem głównym węzła typu Pracownicy

Fk_Gabinety_Placowki - połączenie klucza obcego węzła typu Gabinety, z kluczem głównym węzła typu Placowki

Fk_Wizyty_Gabinety - połączenie klucza obcego węzła typu Wizyty, z kluczem głównym węzła typu Gabinety

Fk_Wizyty_Pacjneci - połączenie klucza obcego węzła typu Wizyty, z kluczem głównym węzła typu Pacjneci

Fk_Wizyty_Pracownicy_Spec - połączenie klucza obcego węzła typu Wizyty, z kluczem głównym węzła typu Pracownicy

Fk_Wizyty_Pracownicy_Uma - połączenie klucza obcego węzła typu Wizyty, z kluczem głównym węzła typu Pracownicy

Fk_Recepty_Pracownicy - połączenie klucza obcego węzła typu Recepty, z kluczem głównym węzła typu Pracownicy

Fk_Recepty_Wizyty - połączenie klucza obcego węzła typu Recepty, z kluczem głównym węzła typu Wizyty

Fk_Recepty_Oddzialy - połączenie klucza obcego węzła typu Recepty, z kluczem głównym węzła typu Oddzialy_nfz

Fk_Recepty_Choroby - połączenie klucza obcego węzła typu Recepty, z kluczem głównym węzła typu Choroby

Fk_Recepty_Ulgi - połączenie klucza obcego węzła typu Recepty, z kluczem głównym węzła typu Ulgi

Fk_Pozycje_Recepty - połączenie klucza obcego węzła typu Pozycje_recept, z kluczem głównym węzła typu Recepty

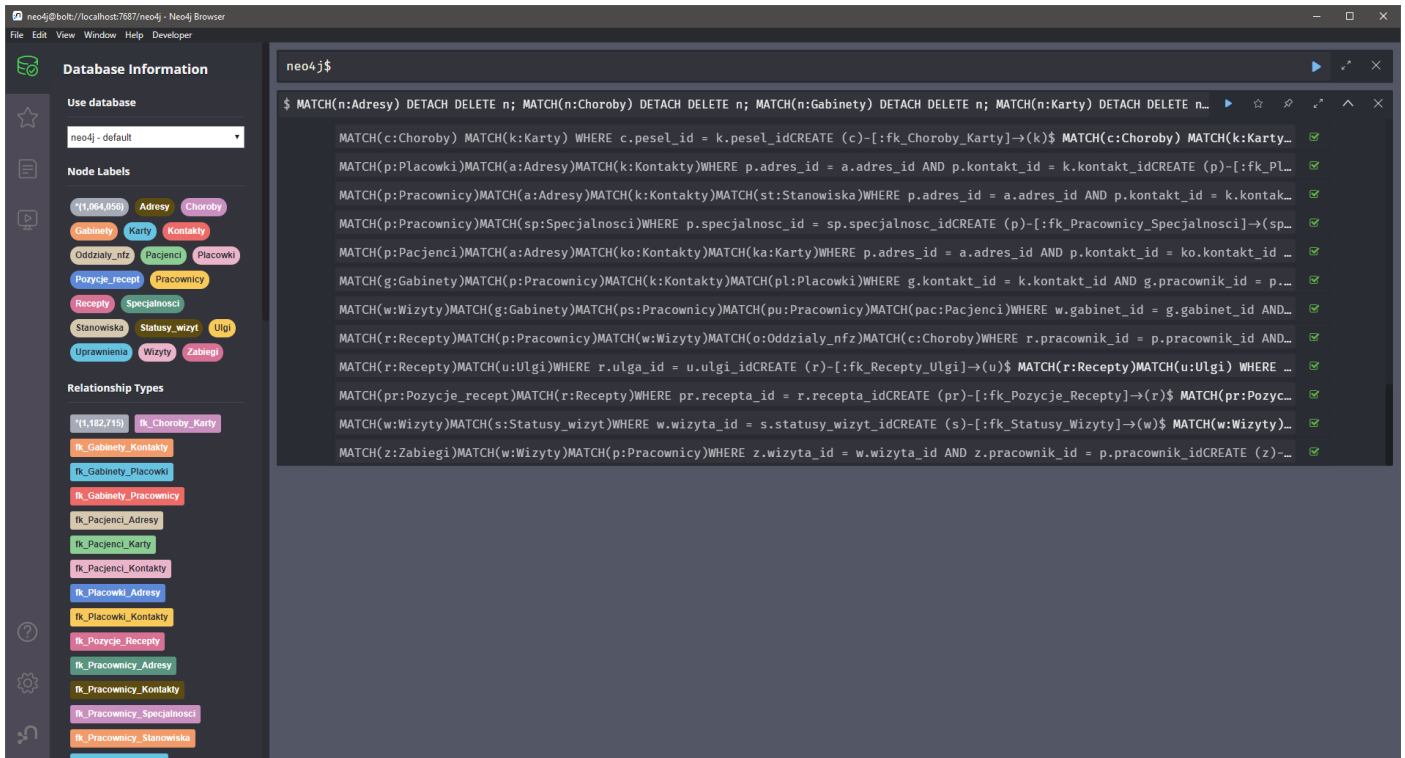
Fk_Statusy_Wizyty - połączenie klucza obcego węzła typu Statusy_wizyt, z kluczem głównym węzła typu Wizyty

Fk_Zabiegi_Wizyty - połączenie klucza obcego węzła typu Zabiegi, z kluczem głównym węzła typu Wizyty

Fk_Zabiegi_Pracownicy - połączenie klucza obcego węzła typu Zabiegi, z kluczem głównym węzła typu Pracownicy

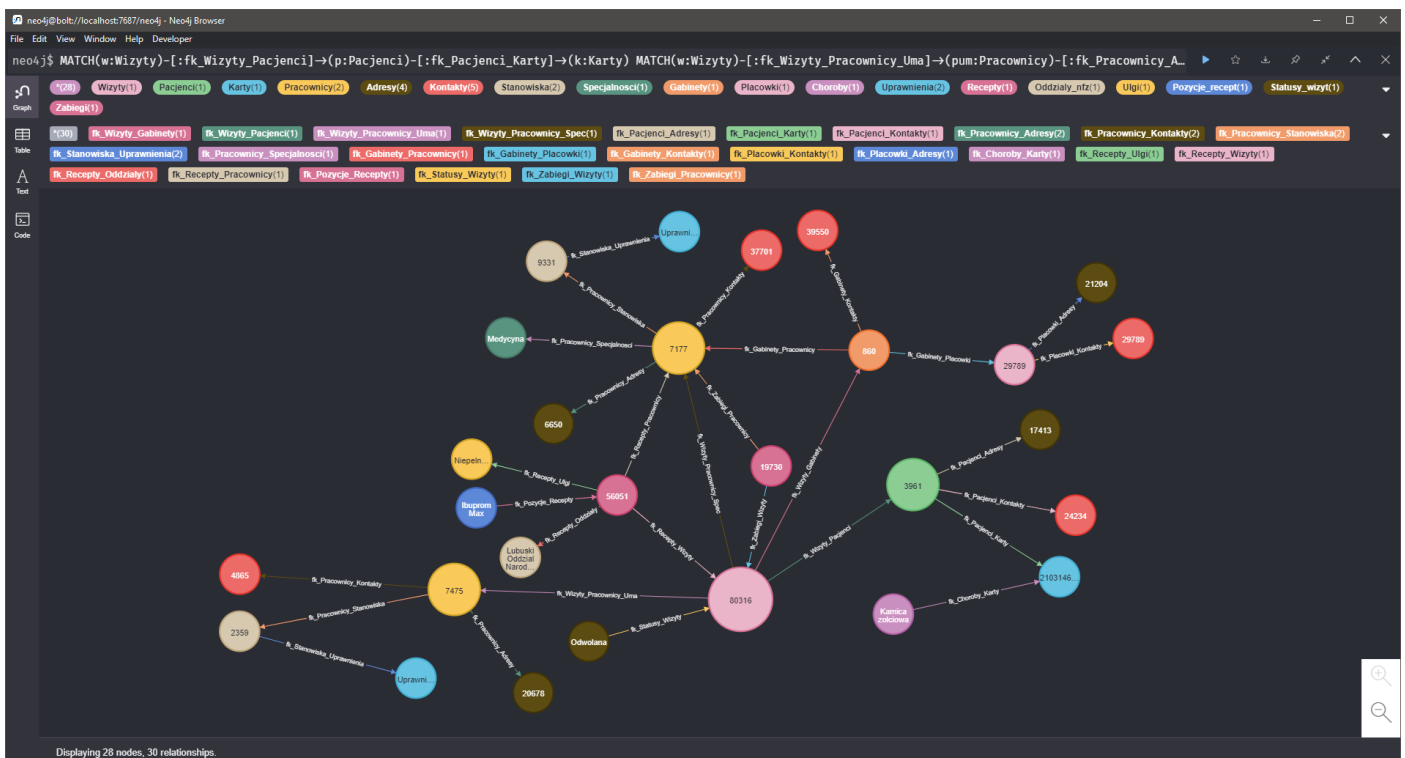
3) Sposób tworzenia bazy

- 1) Przeniesienie plików .csv do pod folderu import dla projektu w neo4j
- 2) Uruchomienie skryptów create.cyp



Po wykonaniu skryptów z create.cyp mamy bazę grafową z ponad milionem węzłów i relacji.

4) Przykładowe powiązanie danych



Podgląd 1 węzła wizyty wraz z resztą węzłów z całej bazy skorelowanych relacjami z nim.