Sprawozdanie Projektowe

Bazy Danych 2

1. Dane zespołu
    - Patryk Grzywacz Gr.2ID12A 090111
    - Dominik Grudzień Gr.2ID12A 090798
2. Opis Problemu
    - Temat „Sieć Gabinetów Lekarskich”

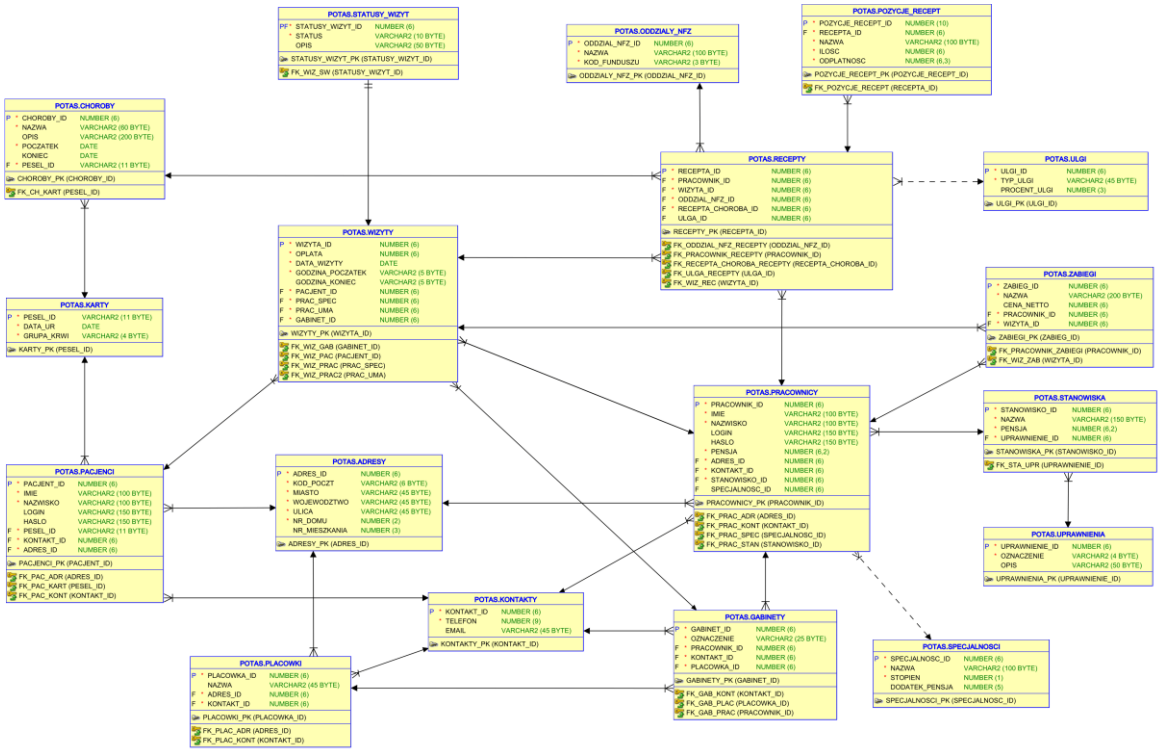
Podeszliśmy do tematu racjonalnie, projektując bazę danych ,która mogłaby faktycznie służyć w prawdziwych okolicznościach jako narzędzie do przechowywania danych ,potrzebnych do działania takich placówek lekarskich.

Rozdzieliliśmy sekcję gabinetów na placówki do, których one przynależą, a przynajmniej w prawdziwym świecie tak jest.

Baza danych zbudowana jest z 18 tabel ,które przechowują istotne elementy, od danych pacjentów, lekarzy i innych pracowników ,po przeróżne wizyty , zabiegi oraz wystawione recepty na leki .

Projektując relacje pomiędzy poszczególnymi tabelami staraliśmy się kierować logiką , aby nie tworzyć niepotrzebne połączenia , czy jak się tyczy samych połączeń, aby ich typy były zgodne z logiką ich późniejszych zastosowań , stosując 2 z 3 dostępnych typów relacji tj: 1-1 czy też 1-n .

3. Schemat ERD Bazy



4. 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 recepcie.
- 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 pracowniku, który go przeprowadzał.

5. Opis procesu transformacji bazy do hurtowni

W celu transformacji relacyjnej bazy danych do hurtowni posłużyliśmy schematem konstelacyjnym, w którym mogą się znaleźć różne rodzaje faktów, a pewne wymiary są współdzielone. Do naszych tabel faktów zaliczyły się tabele Pozycje_Recept oraz Wizyty.
Fakt Pozycje_Recept posiada klucze obce do tabel: Recepty, Leki, Ulgi oraz miary: ilość (danego leku), procent_ulgi i odplatnosc.
Fakt Wizyty ma klucze obce do tabel: Pracownicy, Statusy_Wizyt, Daty_Wizyt, Zabiegi, Pacjenci, Gabinety, Recepty oraz miary: oplata (za wizyte) i cena_netto_za_zabieg.

<u>Opis wymiarów faktu Pozycje_recept:</u>

- Ulgi – wymiar określający nazwy typów ulg. Kolumna procent_ulgi została przeniesiona do faktu pozycje_recept i pełni rolę miary w tym fakcie.
- Leki – wymiar określający nazwy leków.
- Recepty – wymiar współdzielony z tabelą faktów Wizyty. Określa receptę, która jest powiązana z lekami i wizytą.

<u>Opis wymiarów faktu Wizyty:</u>

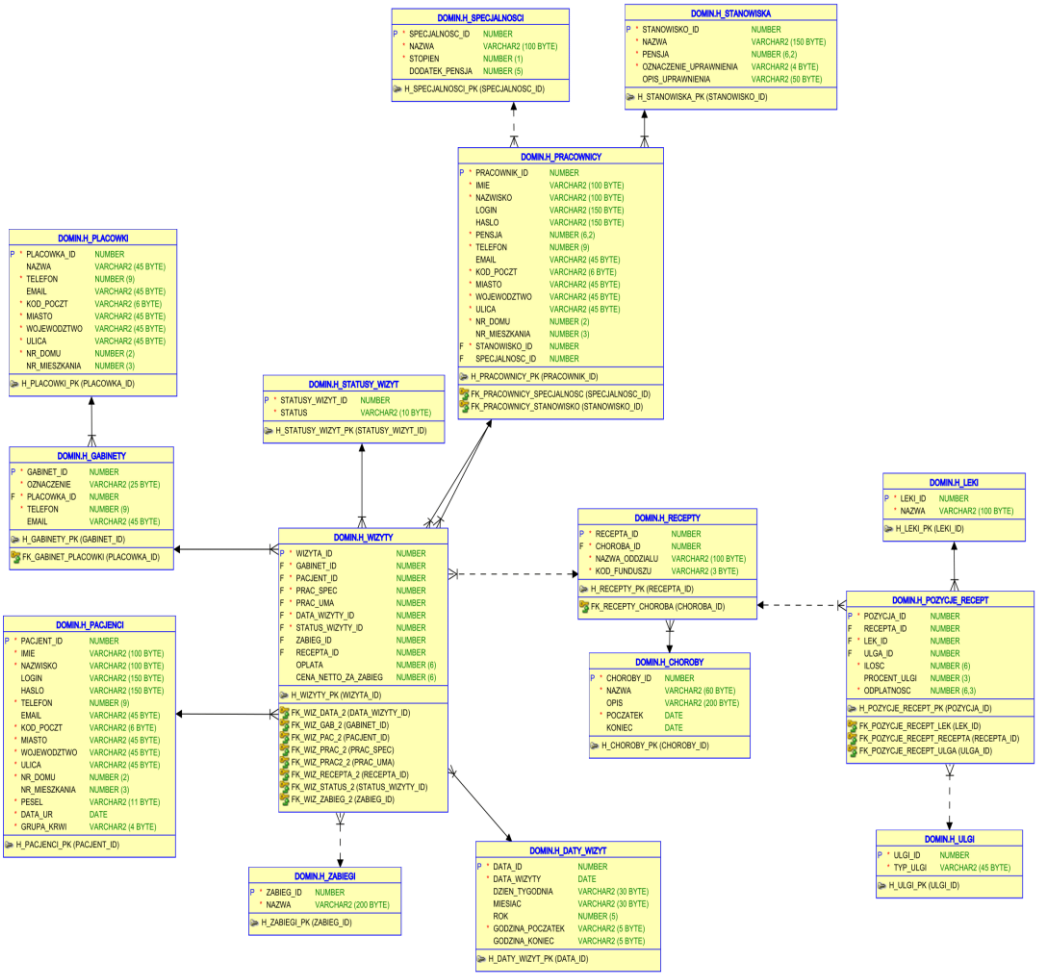
- Statusy_Wizyt – wymiar opisujący status wizyty.
- Pracownicy – wymiar opisujący pracownika związanego z wizytą. Zostały do niego wcielone dane z dwóch tabel: Kontakty i Adresy.
- Recepty - wymiar współdzielony z tabelą faktów Pozcyje_Recept. Określa receptę, która jest powiązana z lekami i wizytą.
- Daty_wizyt – wymiar opisujący dokładny czas odbycia się wizyty.
- Zabiegi – wymiar określający nazwę zabiegu związanego z wizytą.
- Pacjenci – wymiar opisujący pacjenta związanego z wizytą. Zostały do niego wcielone dane z trzech tabel: Kontakty, Adresy oraz Karty.

- Gabinety – wymiar opisujący gabinet, w którym odbyła się wizyta. Została do niego włączona tabela kontakty.

Wymiary Pracownicy, Gabinety i Recepty posiadają zewnętrzne tabele wymiarów. Mają one na celu stworzyć hierarchię tabel wymiarów. Wymiar Pracownicy posiada dwie tabele zewnętrzne: specjalności i stanowiska (do stanowisk zostały włączone dane z tabeli uprawnienia). Wymiar Gabinety ma z kolei jedną tabele zewnętrzną placówki (do placówek zostały włączone dane z tabel: Adresy i Kontakty). Natomiast wymiar Recepty posiada zewnętrzną tabele Choroby.

W schemacie konstelacyjnym współdzielonym wymiarem została tabela Recepty. Zostały do niej włączone dane z tabeli oddzialy_nfz.

6. Schemat ERD Hurtowni

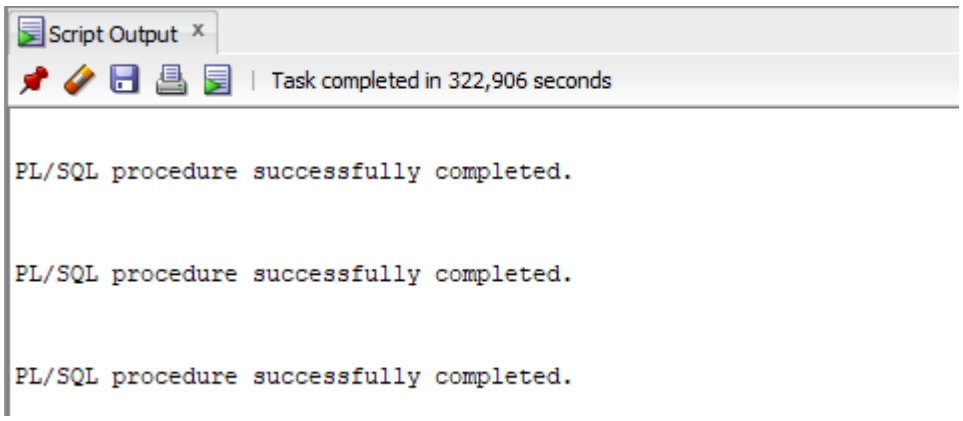


7. Opis procesu ładowania i transformacji danych z bazy do hurtowni
    - Do załadowania i przetransformowania danych z bazy do hurtowni użyliśmy własnoręcznie napisanych procedur, każda procedura wypełnia 1 tabelę z hurtowni danych, dla przykładu procedura „transformacja_pracownicy”

```
create or replace NONEDITIONABLE PROCEDURE transformacja_pracownicy IS
P_ID pracownicy.pracownik_id%type := 1;

CURSOR pobierz_pracownika(IDP IN NUMBER) IS
SELECT p.kontakt_id,p.adres_id,p.stanowisko_id,p.specjalnosc_id,p.imie,p.nazwisko,p.login,p.haslo,p.pensja FROM pracownicy p WHERE  p.pracownik_id = IDP;

CURSOR pobierz_kontakt(IDK IN NUMBER) IS SELECT
k.telefon,k.email FROM kontakty k WHERE k.kontakt_id = IDK;

CURSOR pobierz_adres(IDA IN NUMBER) IS SELECT
a.kod_poczt,a.miasto,a.wojewodztwo,a.ulica,a.nr_domu,a.nr_mieszkania FROM adresy a WHERE a.adres_id = IDA;

pracownik pobierz_pracownika%ROWTYPE;
kontakt pobierz_kontakt%ROWTYPE;
adres pobierz_adres%ROWTYPE;

BEGIN
LOOP

    OPEN pobierz_pracownika(P_ID);
        FETCH pobierz_pracownika INTO pracownik;
    CLOSE pobierz_pracownika;

    OPEN pobierz_kontakt(pracownik.kontakt_id);
        FETCH pobierz_kontakt INTO kontakt;
    CLOSE pobierz_kontakt;

    OPEN pobierz_adres(pracownik.adres_id);
        FETCH pobierz_adres INTO adres;
    CLOSE pobierz_adres;

    INSERT INTO h_pracownicy VALUES(P_ID,pracownik.imie,pracownik.nazwisko,pracownik.login,pracownik.haslo,pracownik.pensja,
    kontakt.telefon,kontakt.email,
    adres.kod_poczt,adres.miasto,adres.wojewodztwo,adres.ulica,adres.nr_domu,adres.nr_mieszkania,
    pracownik.stanowisko_id,pracownik.specjalnosc_id);

    kontakt := null;
    adres := null;

    P_ID := P_ID + 1;
    EXIT WHEN P_ID = 10001;
END LOOP;
END;

/
```

W tej procedurze deklarujemy 4 zmienne oraz 3 kursory, Iterator po tablicy pracownicy z bazy, kursory ,które pobierają wszystkie dane z tabeli podstawowej oraz tych ,które zostały wcielone w hurtowni do danej tabeli oraz 3 zmienne typu zwracanego rekordu kursora . Następnie w pętli iterujemy po tablicy pracownicy i wyciągamy interesujące nas dane poprzez kursor pobierz_pracownika ,później pobieramy dane z adresów i kontaktów za pomocą kluczy obcych wyjętych 1 kursorem i następnie łączymy te wszystkie potrzebne dane w instrukcji insert do tabeli h_pracownicy ,wypełniając ją danymi z bazy.

Wszystkie procedury transformacji danych trwają około 5 minut i 23 sekund.



8. Porównanie wykonania zapytań
    - Rollup nr1.
        - Baza

```
-- Liczba pacjentow pochodzacych z okreslonego oddzialu NFZ, chorujacych na dana chorobe,
-- ktorzy naleza do placowki Prodimed
SELECT c.nazwa AS nazwa_choroby, o.nazwa AS nazwa_oddzialu_nfz, p.nazwa AS nazwa_placowki, COUNT (w.pacjent_id) AS liczba_pacjentow FROM recepty r
JOIN oddzialy_nfz o ON o.oddzial_nfz_id = r.oddzial_nfz_id
JOIN choroby c ON c.choroby_id = r.recepta_choroba_id
JOIN wizyty w ON w.wizyta_id = r.wizyta_id
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
WHERE p.nazwa LIKE 'Prodimed'
GROUP BY ROLLUP (c.nazwa, o.nazwa, p.nazwa);
```

Script Output | Query Result

All Rows Fetched: 463 in 0,072 seconds

	NAZWA_CHOROBY	NAZWA_ODDZIALU_NFZ	NAZWA_PLACOWKI	LICZBA_PACJENTOW
1	Astma	Lodzki Oddzial Narodowego Funduszu Zdrowia w Lodzi	Prodimed	12
2	Astma	Lodzki Oddzial Narodowego Funduszu Zdrowia w Lodzi	(null)	12
3	Astma	Opolski Oddzial Narodowego Funduszu Zdrowia w Opolu	Prodimed	44
4	Astma	Opolski Oddzial Narodowego Funduszu Zdrowia w Opolu	(null)	44
5	Astma	Pomorski Oddzial Narodowego Funduszu Zdrowia w Gdansku	Prodimed	17
6	Astma	Pomorski Oddzial Narodowego Funduszu Zdrowia w Gdansku	(null)	17

        - Hurtownia

```
-- Liczba pacjentow pochodzacych z okreslonego oddzialu NFZ, chorujacych na dana chorobe,
-- ktorzy naleza do placowki Prodimed
SELECT c.nazwa AS nazwa_choroby, r.nazwa_oddzialu AS nazwa_oddzialu_nfz, p.nazwa, COUNT (w.pacjent_id) AS liczba_pacjentow FROM h_wizyty w
JOIN h_recepty r ON r.recepta_id = w.recepta_id
JOIN h_choroby c ON c.choroby_id = r.choroba_id
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
WHERE p.nazwa LIKE 'Prodimed'
GROUP BY ROLLUP (c.nazwa, r.nazwa_oddzialu, p.nazwa);
```

Script Output | Query Result

All Rows Fetched: 463 in 0,564 seconds

	NAZWA_CHOROBY	NAZWA_ODDZIALU_NFZ	NAZWA	LICZBA_PACJENTOW
1	Astma	Lodzki Oddzial Narodowego Funduszu Zdrowia w Lodzi	Prodimed	12
2	Astma	Lodzki Oddzial Narodowego Funduszu Zdrowia w Lodzi	(null)	12
3	Astma	Opolski Oddzial Narodowego Funduszu Zdrowia w Opolu	Prodimed	44
4	Astma	Opolski Oddzial Narodowego Funduszu Zdrowia w Opolu	(null)	44
5	Astma	Pomorski Oddzial Narodowego Funduszu Zdrowia w Gdansku	Prodimed	17
6	Astma	Pomorski Oddzial Narodowego Funduszu Zdrowia w Gdansku	(null)	17

Różnicą w tych zapytaniu dla bazy oraz hurtowni jest to ,że w bazie wymaga ono o jedno więcej połączenie JOIN , natomiast wykonuje się ono w hurtowni aż 683% wolniej niż w bazie danych.

- Rollup nr2.
  - Baza

Worksheet | Query Builder

```
-- Liczba pracownikow pracujacych na danym stanowisku, posiadajacych okreslone uprawnienia,
-- mieszkajacych w Kielcach
SELECT s.nazwa AS nazwa_stanowiska, u.opis AS typ_uprawnienia, a.miasto, COUNT (*) AS liczba_pracownikow FROM pracownicy p
JOIN stanowiska s ON s.stanowisko_id = p.stanowisko_id
JOIN uprawnienia u ON u.uprawnienie_id = s.uprawnienie_id
JOIN adresy a ON a.adres_id = p.adres_id
WHERE a.miasto LIKE 'Kielce'
GROUP BY ROLLUP (s.nazwa, u.opis, a.miasto);
```

Script Output | Query Result

SQL | All Rows Fetched: 244 in 0,075 seconds

	NAZWA_STANOWISKA	TYP_UPRAWNIENIA	MIASTO	LICZBA_PRACOWNIKOW
1	Urolog	Uprawnienia stopnia 4	Kielce	3
2	Urolog	Uprawnienia stopnia 4	(null)	3
3	Urolog	Uprawnienia stopnia 5	Kielce	5
4	Urolog	Uprawnienia stopnia 5	(null)	5
5	Urolog	Uprawnienia stopnia 6	Kielce	4
6	Urolog	Uprawnienia stopnia 6	(null)	4

  - Hurtownia

Worksheet | Query Builder

```
-- Liczba pracownikow pracujacych na danym stanowisku, posiadajacych okreslone uprawnienia,
-- mieszkajacych w Kielcach
SELECT s.nazwa AS nazwa_stanowiska, s.opis_uprawnienia AS typ_uprawnienia, p.miasto, COUNT (*) AS liczba_pracownikow FROM h_pracownicy p
JOIN h_stanowiska s ON s.stanowisko_id = p.stanowisko_id
WHERE p.miasto LIKE 'Kielce'
GROUP BY ROLLUP (s.nazwa, s.opis_uprawnienia, p.miasto);
```

Script Output | Query Result

SQL | All Rows Fetched: 244 in 0,054 seconds

	NAZWA_STANOWISKA	TYP_UPRAWNIENIA	MIASTO	LICZBA_PRACOWNIKOW
1	Urolog	Uprawnienia stopnia 4	Kielce	3
2	Urolog	Uprawnienia stopnia 4	(null)	3
3	Urolog	Uprawnienia stopnia 5	Kielce	5
4	Urolog	Uprawnienia stopnia 5	(null)	5
5	Urolog	Uprawnienia stopnia 6	Kielce	4
6	Urolog	Uprawnienia stopnia 6	(null)	4

W tym zapytaniu natomiast wersja dla hurtownia wymaga o 2 mniej złączenia i wykonuje się 39% szybciej niż wersja dla zwykłej bazy.

- Rollup nr3.
  - Baza

```
-- Liczba wizyt pacjentow z danym schorzeniem , w danej placowce oraz w danym miescie
SELECT c.nazwa AS nazwa_choroby, p.nazwa AS nazwa_placowki, a.miasto, COUNT (w.wizyta_id) AS liczba_wizyt FROM wizyty w
JOIN recepty r ON w.wizyta_id = r.wizyta_id
JOIN choroby c ON c.choroby_id = r.recepta_choroba_id
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
JOIN adresy a ON a.adres_id = p.adres_id
WHERE g.oznaczenie LIKE 'Wizyty%'
GROUP BY ROLLUP (c.nazwa, p.nazwa, a.miasto);
```

Script Output | Query Result

All Rows Fetched: 2633 in 0,195 seconds

	NAZWA_CHOROBY	NAZWA_PLACOWKI	MIASTO	LICZBA_WIZYT
1	Astma	Adamed	Lodz	17
2	Astma	Adamed	Opole	16
3	Astma	Adamed	Gdansk	8
4	Astma	Adamed	Kielce	14
5	Astma	Adamed	Krakow	7
6	Astma	Adamed	Lublin	32

  - Hurtownia

```
-- Liczba wizyt pacjentow ze schorzeniem osteoporozy, w danej placowce oraz w danym miescie
SELECT c.nazwa AS nazwa_choroby, p.nazwa AS nazwa_placowki, p.miasto, COUNT (w.wizyta_id) AS liczba_wizyt FROM h_wizyty w
JOIN h_recepty r ON w.recepta_id = r.recepta_id
JOIN h_choroby c ON c.choroby_id = r.choroba_id
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
WHERE g.oznaczenie LIKE 'Wizyty%'
GROUP BY ROLLUP (c.nazwa, p.nazwa, p.miasto);
```

Script Output | Query Result

All Rows Fetched: 2633 in 0,356 seconds

	NAZWA_CHOROBY	NAZWA_PLACOWKI	MIASTO	LICZBA_WIZYT
1	Astma	Adamed	Lodz	17
2	Astma	Adamed	Opole	16
3	Astma	Adamed	Gdansk	8
4	Astma	Adamed	Kielce	14
5	Astma	Adamed	Krakow	7
6	Astma	Adamed	Lublin	32

W tym zapytaniu wersja dla hurtowni wymaga o 1 mniej złączenie ale wykonuje się o 82% wolniej niż dla bazy.

- Cube nr1.
  - Baza

Worksheet | Query Builder

```
-- Srednia oplata wizyt danego pacjenta, w placowce, w danym miescie
SELECT pac.nazwisko AS nazwisko_pacjenta, p.nazwa AS nazwa_placowki, a.miasto, AVG (w.oplata) AS srednia_oplata FROM wizyty w
JOIN pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
JOIN adresy a ON a.adres_id = p.adres_id
WHERE g.oznaczenie LIKE 'Wizyty%'
GROUP BY CUBE (pac.nazwisko, p.nazwa, a.miasto);
```

Script Output | Query Result

SQL | All Rows Fetched: 12688 in 0,401 seconds

	NAZWISKO_PACJENTA	NAZWA_PLACOWKI	MIASTO	SREDNIA_OPLATA
1	(null)	(null)	(null)	801,540022682504626037127678624723930042
2	(null)	(null)	Lodz	795,655102728191310205456382620410912765
3	(null)	(null)	Opole	803,725422559503276992066229734391169369
4	(null)	(null)	Gdansk	800,659413122321134190570392350807781075
5	(null)	(null)	Kielce	811,970074812967581047381546134663341646
6	(null)	(null)	Krakow	803,423591658885776532835356364768129474

  - Hurtownia

Worksheet | Query Builder

```
-- Srednia oplata wizyt danego pacjenta, w placowce, w danym miescie
SELECT pac.nazwisko AS nazwisko_pacjenta, p.nazwa AS nazwa_placowki, p.miasto, AVG (w.oplata) AS srednia_oplata FROM h_wizyty w
JOIN h_pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
WHERE g.oznaczenie LIKE 'Wizyty%'
GROUP BY CUBE (pac.nazwisko, p.nazwa, p.miasto);
```

Script Output | Query Result

SQL | All Rows Fetched: 12688 in 0,653 seconds

	NAZWISKO_PACJENTA	NAZWA_PLACOWKI	MIASTO	SREDNIA_OPLATA
1	(null)	(null)	(null)	801,540022682504626037127678624723930042
2	(null)	(null)	Lodz	795,655102728191310205456382620410912765
3	(null)	(null)	Opole	803,725422559503276992066229734391169369
4	(null)	(null)	Gdansk	800,659413122321134190570392350807781075
5	(null)	(null)	Kielce	811,970074812967581047381546134663341646
6	(null)	(null)	Krakow	803,423591658885776532835356364768129474

Wersja dla hurtowni tego zapytania różni się od zwykłej tym ,że posiada o 1 mniej złączenie JOIN ,co za tym idzie dane adresowe są wyciągane prosto z placówki , natomiast nadal wykonuje się o 63% wolniej niż dla zwykłej bazy.

- Cube nr2.
  - Baza

Worksheet | Query Builder

```
-- Srednia oplata zabiegow danego pacjenta, w placowce, w danym miescie
SELECT pac.nazwisko AS nazwisko_pacjenta, p.nazwa AS nazwa_placowki, a.miasto, AVG (z.cena_netto) AS srednia_oplata FROM wizyty w
JOIN zabiegi z ON w.wizyta_id = z.wizyta_id
JOIN pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
JOIN adresy a ON a.adres_id = p.adres_id
WHERE g.oznaczenie LIKE 'Zabiegi'
GROUP BY CUBE (pac.nazwisko, p.nazwa, a.miasto);
```

Script Output | Query Result

SQL | All Rows Fetched: 11259 in 0,441 seconds

	NAZWISKO_PACJENTA	NAZWA_PLACOWKI	MIASTO	SREDNIA_OPLATA
1	(null)	(null)	(null)	2756,06314670156639810232710318596376426 3
2	(null)	(null)	Lodz	2738,962472406181015452538631346578366446
3	(null)	(null)	Opole	2713,956550362080315997366688610928242265
4	(null)	(null)	Gdansk	2781,806615776081424936386768447837150127
5	(null)	(null)	Kielce	2734,268125854993160054719562243502051984
6	(null)	(null)	Krakow	2800,818124606670862177470106985525487728

  - Hurtownia

Worksheet | Query Builder

```
-- Srednia oplata zabiegow danego pacjenta, w placowce, w danym miescie
SELECT pac.nazwisko AS nazwisko_pacjenta, p.nazwa AS nazwa_placowki, p.miasto, AVG (w.cena_netto_za_zabieg) AS srednia_oplata FROM h_wizyty w
JOIN h_pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
WHERE g.oznaczenie LIKE 'Zabiegi'
GROUP BY CUBE (pac.nazwisko, p.nazwa, p.miasto);
```

Script Output | Query Result

SQL | All Rows Fetched: 11259 in 0,525 seconds

	NAZWISKO_PACJENTA	NAZWA_PLACOWKI	MIASTO	SREDNIA_OPLATA
1	(null)	(null)	(null)	2756,06314670156639810232710318596376426 3
2	(null)	(null)	Lodz	2738,962472406181015452538631346578366446
3	(null)	(null)	Opole	2713,956550362080315997366688610928242265
4	(null)	(null)	Gdansk	2781,806615776081424936386768447837150127
5	(null)	(null)	Kielce	2734,268125854993160054719562243502051984
6	(null)	(null)	Krakow	2800,818124606670862177470106985525487728

Porównując te dwa zapytania widzimy ,że wersja dla hurtowni jest prostsza i posiada o 2 złączenia tabel mniej i zlicza średnią opłat prosto z tabeli faktów , nie mniej jednak wykonuje się 19% wolniej niż wersja dla zwykłej bazy danych.

- Cube nr3.
  - Baza

Worksheet | Query Builder

```
-- Srednia oplata za leki danego pacjenta, w danym miescie, chorujacego na okreslona chorobe, powiazanego z danego oddzialu
SELECT pac.nazwisko AS nazwisko_pacjenta, a.miasto, c.nazwa AS nazwa_choroby, AVG (pr.odplatnosc) AS srednia_oplata_za_leki FROM pozycje_recept pr
JOIN recepty r ON pr.recepta_id = r.recepta_id
JOIN wizyty w ON w.wizyta_id = r.wizyta_id
JOIN pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN adresy a ON a.adres_id = pac.adres_id
JOIN choroby c ON c.choroby_id = r.recepta_choroba_id
GROUP BY CUBE (pac.nazwisko, a.miasto, c.nazwa);
```

Script Output | Query Result

All Rows Fetched: 13711 in 0,576 seconds

	NAZWISKO_PACJENTA	MIASTO	NAZWA_CHOROBY	SREDNIA_OPLATA_ZA_LEKI
1	(null)	(null)	(null)	151,05659847473784556720686367969494756 9
2	(null)	(null)	Astma	150,53333312322722975102426725496375669 7
3	(null)	(null)	Grypa	149,99764388979814511729405346426623022 4
4	(null)	(null)	Wrzody	151,15545259572320696398158077335520090 8
5	(null)	(null)	Alergia	150,87050718629076838032061912658927584 3
6	(null)	(null)	Cukrzyca	151,24131070764098287034066850580612048 5

  - Hurtownia

Worksheet | Query Builder

```
-- Srednia oplata za leki danego pacjenta, w danym miescie, chorujacego na okreslona chorobe
SELECT pac.nazwisko AS nazwisko_pacjenta, pac.miasto, c.nazwa AS nazwa_choroby, AVG (pr.odplatnosc) AS srednia_oplata_za_leki FROM h_pozycje_recept pr
JOIN h_recepty r ON pr.recepta_id = r.recepta_id
RIGHT JOIN h_wizyty w ON w.recepta_id = r.recepta_id
JOIN h_pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN h_choroby c ON c.choroby_id = r.choroba_id
GROUP BY CUBE (pac.nazwisko, pac.miasto, c.nazwa);
```

Script Output | Query Result

All Rows Fetched: 13711 in 1,064 seconds

	NAZWISKO_PACJENTA	MIASTO	NAZWA_CHOROBY	SREDNIA_OPLATA_ZA_LEKI
1	(null)	(null)	(null)	151,05659847473784556720686367969494756 9
2	(null)	(null)	Astma	150,53333312322722975102426725496375669 7
3	(null)	(null)	Grypa	149,99764388979814511729405346426623022 4
4	(null)	(null)	Wrzody	151,15545259572320696398158077335520090 8
5	(null)	(null)	Alergia	150,87050718629076838032061912658927584 3
6	(null)	(null)	Cukrzyca	151,24131070764098287034066850580612048 5

W tym zapytaniu wersja dla hurtowni wymaga jedno mniej złączenie natomiast wykonuje się aż 84% wolniej.

- Partycje Obliczeniowe nr1.
  - Baza

Worksheet | Query Builder

```
--Łączna wartość roczna za każdy zabieg przeprowadzany na pacjentach z grupą krwi A-
SELECT DISTINCT z.nazwa,
EXTRACT(YEAR FROM w.data_wizyty) AS Rok,
k.grupa_krwi,
SUM(z.cena_netto) OVER (PARTITION BY EXTRACT(YEAR FROM w.data_wizyty),k.grupa_krwi,z.nazwa) AS "Wartosc_Roczna"
FROM zabiegi z
JOIN wizyty w ON w.wizyta_id = z.wizyta_id
JOIN pacjenci p ON w.pacjent_id = p.pacjent_id
JOIN karty k ON p.pesel_id = k.pesel_id
WHERE k.grupa_krwi = 'A-'
ORDER BY z.nazwa,Rok DESC;
```

Script Output | Query Result

SQL | All Rows Fetched: 1748 in 0,091 seconds

	NAZWA	ROK	GRUPA_KRWI	Wartosc_Roczna
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

  - Hurtownia

Worksheet | Query Builder

```
--Łączna wartość roczna za każdy zabieg przeprowadzany na pacjentach z grupą krwi A-
SELECT DISTINCT z.nazwa,
d.rok AS Rok,
p.grupa_krwi,
SUM(w.cena_netto_za_zabieg) OVER (PARTITION BY d.rok,p.grupa_krwi,z.nazwa) AS "Wartosc_Roczna"
FROM h_wizyty w
JOIN h_daty_wizyt d ON d.data_id = w.data_wizyty_id
RIGHT JOIN h_zabiegi z ON w.zabieg_id = z.zabieg_id
JOIN h_pacjenci p ON w.pacjent_id = p.pacjent_id
WHERE p.grupa_krwi = 'A-'
ORDER BY z.nazwa,Rok DESC;
```

Script Output | Query Result

SQL | All Rows Fetched: 1748 in 0,253 seconds

	NAZWA	ROK	GRUPA_KRWI	Wartosc_Roczna
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

Te dwie wersje zapytania praktycznie nie różnią się złożonością , lecz różnią się tym skąd dane są wyciągane z tabel, np.: w hurtowni Suma za zabiegi jest liczona z danych zawartych w tabeli faktów a partycjonowanie jest w pierwszej kolejności po polu rok z tabeli h_daty_wizyt a nie jak w wersji dla zwykłej bazy ,gdzie partycjonowane po roku ,który jest wyłuskiwany z pola typu Date z wizyt. Końcowo zapytanie w hurtowni wykonuje się 178% wolniej niż w zwykłej bazie.

- Partycje Obliczeniowe nr2.
  - Baza

```
--Id pacjenta,jego imie ,nazwisko , pesel oraz jego Wydatki na leki wciągu jednego roku z ulgą/bez ulgi
SELECT DISTINCT
p.pacjent_id,p.imie,p.nazwisko,p.pesel_id,
EXTRACT(YEAR FROM w.data_wizyty) AS Rok,
SUM(pr.odplatnosc) OVER (PARTITION BY p.pacjent_id,EXTRACT(YEAR FROM w.data_wizyty)) AS "Wydatki Pacjenta Na Leki Bez Ulgi",
(SUM(pr.odplatnosc) OVER (PARTITION BY p.pacjent_id,EXTRACT(YEAR FROM w.data_wizyty)) * (u.procent_ulgi / 100) ) AS "Wydatki Pacjenta Na Leki Z Wliczona Ulga"
FROM recepty r
JOIN pozycje_recept pr ON pr.recepta_id = r.recepta_id
JOIN wizyty w ON w.wizyta_id = r.wizyta_id
JOIN pacjenci p ON p.pacjent_id = w.pacjent_id
LEFT JOIN ulgi u ON u.ulgi_id = r.ulga_id
ORDER BY p.pacjent_id ASC,Rok DESC;
```

Script Output | Query Result

All Rows Fetched: 61288 in 1,623 seconds

	PACJENT_ID	IMIE	NAZWISKO	PESEL_ID	ROK	Wydatki Pacjenta Na Leki Bez Ulgi	Wydatki Pacjenta Na Leki Z Wliczona Ulga
1	1	Emil	Brzezinski	13088027337	2021	1217,96	(null)
2	1	Emil	Brzezinski	13088027337	2017	874,78	(null)
3	1	Emil	Brzezinski	13088027337	2015	310,48	(null)
4	1	Emil	Brzezinski	13088027337	2012	190,8	(null)
5	1	Emil	Brzezinski	13088027337	2008	44,99	(null)
6	1	Emil	Brzezinski	13088027337	2006	224,95	(null)
7	2	Kewin	Marciniak	61124651421	2018	301,15	(null)

  - Hurtownia

```
--Id pacjenta,jego imie ,nazwisko , pesel oraz jego Wydatki na leki wciągu jednego roku z/bez ulgą
SELECT DISTINCT
p.pacjent_id,p.imie,p.nazwisko,p.pesel,d.rok AS Rok,
SUM(pr.odplatnosc) OVER (PARTITION BY p.pacjent_id,d.rok) AS "Wydatki Pacjenta Na Leki Bez Ulgi",
(SUM(pr.odplatnosc) OVER (PARTITION BY p.pacjent_id,d.rok) * (pr.procent_ulgi / 100) ) AS "Wydatki Pacjenta Na Leki Z Wliczona Ulga"
FROM h_wizyty w
JOIN h_daty_wizyt d ON d.data_id = w.data_wizyty_id
JOIN h_recepty r ON w.recepta_id = r.recepta_id
JOIN h_pozycje_recept pr ON pr.recepta_id = r.recepta_id
JOIN h_pacjenci p ON p.pacjent_id = w.pacjent_id
ORDER BY p.pacjent_id ASC,Rok DESC;
```

Script Output | Query Result

All Rows Fetched: 61288 in 1,62 seconds

	PACJENT_ID	IMIE	NAZWISKO	PESEL	ROK	Wydatki Pacjenta Na Leki Bez Ulgi	Wydatki Pacjenta Na Leki Z Wliczona Ulga
1	1	Emil	Brzezinski	13088027337	2021	1217,96	(null)
2	1	Emil	Brzezinski	13088027337	2017	874,78	(null)
3	1	Emil	Brzezinski	13088027337	2015	310,48	(null)
4	1	Emil	Brzezinski	13088027337	2012	190,8	(null)
5	1	Emil	Brzezinski	13088027337	2008	44,99	(null)
6	1	Emil	Brzezinski	13088027337	2006	224,95	(null)
7	2	Kewin	Marciniak	61124651421	2018	301,15	(null)

Zapytanie to w obu wersjach praktycznie nie różni się czasem wykonania, zawiera tyle samo JOIN'ów ,jedynie czym się różni wersja hurtowni od zwykłej jest prostota wyciągania kolumn po ,których się wyciąga dane i partycjonuje sumowanie.

- Partycje Obliczeniowe nr3.
  - Baza

Worksheet | Query Builder

```
-- Procentowy udzial oplat za wizyte, w danym roku, w okreslonej placowce, znajdujacej sie w okreslonym miescie, na przestrzeni wszystkich lat
SELECT DISTINCT p.placowka_id, p.nazwa AS nazwa_placowki, a.miasto, EXTRACT (YEAR FROM w.data_wizyty) AS rok,
SUM (w.oplata) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty), p.placowka_id, a.miasto) suma_w_danym_roku,
SUM (w.oplata) OVER (PARTITION BY p.placowka_id, a.miasto) suma_na_przestrzeni_lat,
ROUND (100 * (SUM (w.oplata) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty), p.placowka_id, a.miasto)) / SUM (w.oplata) OVER (PARTITION BY p.placowka_id, a.miasto))
"UDZIAL % danego roku" FROM wizyty w
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
JOIN adresy a ON a.adres_id = p.adres_id
ORDER BY p.placowka_id ASC, rok DESC;
```

Script Output | Query Result

All Rows Fetched: 70570 in 1,56 seconds

	PLACOWKA_ID	NAZWA_PLACOWKI	MIASTO	ROK	SUMA_W_DANYM_ROKU	SUMA_NA_PRZESTRZENI_LAT	UDZIAL % danego roku
1	1	Centrum Medyczno-Diagnostyczne	Bialystok	2020	2100	9400	22
2	1	Centrum Medyczno-Diagnostyczne	Bialystok	2017	1300	9400	14
3	1	Centrum Medyczno-Diagnostyczne	Bialystok	2015	300	9400	3
4	1	Centrum Medyczno-Diagnostyczne	Bialystok	2013	1200	9400	13
5	1	Centrum Medyczno-Diagnostyczne	Bialystok	2009	1000	9400	11
6	1	Centrum Medyczno-Diagnostyczne	Bialystok	2007	1200	9400	13
7	1	Centrum Medyczno-Diagnostyczne	Bialystok	2005	1200	9400	13
8	1	Centrum Medyczno-Diagnostyczne	Bialystok	2001	200	9400	2
9	1	Centrum Medyczno-Diagnostyczne	Bialystok	1975	900	9400	10
10	3	Adamed	Bialystok	2023	1000	19400	5

  - Hurtownia

Worksheet | Query Builder

```
-- Procentowy udzial oplat za wizyte, w danym roku, w okreslonej placowce, znajdujacej sie w okreslonym miescie, na przestrzeni wszystkich lat
SELECT DISTINCT p.placowka_id, p.nazwa AS nazwa_placowki, p.miasto, d.rok,
SUM (w.oplata) OVER (PARTITION BY d.rok, p.placowka_id, p.miasto) suma_w_danym_roku,
SUM (w.oplata) OVER (PARTITION BY p.placowka_id, p.miasto) suma_na_przestrzeni_lat,
ROUND (100 * (SUM (w.oplata) OVER (PARTITION BY d.rok, p.placowka_id, p.miasto)) / SUM (w.oplata) OVER (PARTITION BY p.placowka_id, p.miasto))
"UDZIAL % danego roku" FROM h_wizyty w
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
JOIN h_daty_wizyt d ON d.data_id = w.data_wizyty_id
ORDER BY p.placowka_id ASC, d.rok DESC;
```

Script Output | Query Result

All Rows Fetched: 70570 in 1,838 seconds

	PLACOWKA_ID	NAZWA_PLACOWKI	MIASTO	ROK	SUMA_W_DANYM_ROKU	SUMA_NA_PRZESTRZENI_LAT	UDZIAL % danego roku
1	1	Centrum Medyczno-Diagnostyczne	Bialystok	2020	2100	9400	22
2	1	Centrum Medyczno-Diagnostyczne	Bialystok	2017	1300	9400	14
3	1	Centrum Medyczno-Diagnostyczne	Bialystok	2015	300	9400	3
4	1	Centrum Medyczno-Diagnostyczne	Bialystok	2013	1200	9400	13
5	1	Centrum Medyczno-Diagnostyczne	Bialystok	2009	1000	9400	11
6	1	Centrum Medyczno-Diagnostyczne	Bialystok	2007	1200	9400	13
7	1	Centrum Medyczno-Diagnostyczne	Bialystok	2005	1200	9400	13
8	1	Centrum Medyczno-Diagnostyczne	Bialystok	2001	200	9400	2
9	1	Centrum Medyczno-Diagnostyczne	Bialystok	1975	900	9400	10
10	3	Adamed	Bialystok	2023	1000	19400	5

Te dwie wersje zapytania posiadają tyle samo złączeń tabel, różnią się tym ,że wersja hurtowni ma prostszy zapis lecz wykonuje się prawie 18% dłużej.

- Okna Czasowe nr1.
  - Baza

```
--Zlicza ilość wizyt przeprowadzonych ,recept wypisanych oraz zabiegów w danym roku od 15 dni przed do 15 dni po aktualnej dacie wizyty
SELECT DISTINCT
EXTRACT (YEAR FROM w.data_wizyty) AS Rok ,EXTRACT (MONTH FROM w.data_wizyty) AS Miesiac ,w.data_wizyty AS Data,
COUNT(w.wizyta_id) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty) ORDER BY w.data_wizyty DESC RANGE BETWEEN INTERVAL '15' DAY PRECEDING AND INTERVAL '15' DAY FOLLOWING) AS Ilosc_wizyt,
COUNT(z.zabieg_id) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty) ORDER BY w.data_wizyty DESC RANGE BETWEEN INTERVAL '15' DAY PRECEDING AND INTERVAL '15' DAY FOLLOWING) AS Ilosc_zabiegow,
COUNT(r.recepta_id) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty) ORDER BY w.data_wizyty DESC RANGE BETWEEN INTERVAL '15' DAY PRECEDING AND INTERVAL '15' DAY FOLLOWING) AS Ilosc_recept
FROM wizyty w
LEFT JOIN recepty r ON r.wizyta_id = w.wizyta_id
LEFT JOIN zabiegi z ON z.wizyta_id = w.wizyta_id
ORDER BY Rok DESC , Miesiac ASC;
```

Script Output | Query Result — All Rows Fetched: 10137 in 0,471 seconds

	ROK	MIESIAC	DATA	ILOSC_WIZYT	ILOSC_ZABIEGOW	ILOSC_RECEPT
1	2023	1	23/01/01	183	49	134
2	2023	1	23/01/02	202	54	148
3	2023	1	23/01/03	211	58	153
4	2023	1	23/01/04	220	61	157
5	2023	1	23/01/05	235	64	170
6	2023	1	23/01/06	249	69	181
7	2023	1	23/01/07	253	70	183
8	2023	1	23/01/08	266	73	193
9	2023	1	23/01/09	280	76	200
10	2023	1	23/01/10	293	81	209
11	2023	1	23/01/11	310	84	221

  - Hurtownia

```
--Zlicza ilość wizyt przeprowadzonych ,recept wypisanych oraz zabiegów w danym roku od 15 dni przed do 15 dni po aktualnej dacie wizyty
SELECT DISTINCT
d.rok AS Rok ,d.miesiac AS Miesiac ,d.data_wizyty AS Data,
COUNT(w.wizyta_id) OVER (PARTITION BY d.rok ORDER BY d.data_wizyty DESC RANGE BETWEEN INTERVAL '15' DAY PRECEDING AND INTERVAL '15' DAY FOLLOWING) AS Ilosc_wizyt,
COUNT(w.zabieg_id) OVER (PARTITION BY d.rok ORDER BY d.data_wizyty DESC RANGE BETWEEN INTERVAL '15' DAY PRECEDING AND INTERVAL '15' DAY FOLLOWING) AS Ilosc_zabiegow,
COUNT(w.recepta_id) OVER (PARTITION BY d.rok ORDER BY d.data_wizyty DESC RANGE BETWEEN INTERVAL '15' DAY PRECEDING AND INTERVAL '15' DAY FOLLOWING) AS Ilosc_recept
FROM h_wizyty w
JOIN h_daty_wizyt  d ON d.data_id = w.data_wizyty_id
ORDER BY Rok DESC , Miesiac ASC;
```

Script Output | Query Result — All Rows Fetched: 10137 in 0,429 seconds

	ROK	MIESIAC	DATA	ILOSC_WIZYT	ILOSC_ZABIEGOW	ILOSC_RECEPT
1	2023	1	23/01/01	183	49	134
2	2023	1	23/01/02	202	54	148
3	2023	1	23/01/03	211	58	153
4	2023	1	23/01/04	220	61	157
5	2023	1	23/01/05	235	64	170
6	2023	1	23/01/06	249	69	181
7	2023	1	23/01/07	253	70	183
8	2023	1	23/01/08	266	73	193
9	2023	1	23/01/09	280	76	200
10	2023	1	23/01/10	293	81	209
11	2023	1	23/01/11	310	84	221
12	2023	1	23/01/12	316	85	226

Te dwie wersje zapytań różnią się tym ,że zapytanie w hurtowni wymaga o 1 złączenie mniej i mniej zabiegów przy wyciąganiu danych, ponieważ nie potrzeba wyłuskiwać z Daty rok i miesiąc, także zapytanie dla hurtowni wykonuje się 9% szybciej.

- Okna Czasowe nr2.
  - Baza

```
--Wypisuje nazwę choroby, jej ID , Rok przeszukiwań danych,Ilość znalezionych rekordów od początku tabeli do aktualnego rekordu z uwzględnieniem nazwy oraz roku, oraz dane Pacjenta
SELECT DISTINCT
c.nazwa AS Nazwa_Choroby,c.choroby_id,EXTRACT(YEAR FROM c.poczatek) Rok,w.wizyta_id,
COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
AS "Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu wizyty",
ROUND(100 * COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
/ COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ) , 3) "Udział % w tym roku",
COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ) "Łaczna wartość zapadnięć na tę chorobę w tym roku",
ROUND(100 * COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ) / COUNT(*) OVER (PARTITION BY c.nazwa) , 3) "Udział % na tle lat",
COUNT(*) OVER (PARTITION BY c.nazwa) "Łaczna wartość zapadnięć na tę chorobę na przestrzeni lat",
k.grupa_krwi "Grupa Krwi Pacjenta",p.imie,p.nazwisko
FROM choroby c
JOIN recepty r ON r.recepta_choroba_id = c.choroby_id
JOIN wizyty  w ON w.wizyta_id = r.wizyta_id
JOIN pacjenci p ON p.pacjent_id = w.pacjent_id
JOIN karty k ON k.pesel_id = p.pesel_id
ORDER BY Rok DESC,COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) DESC;
```

Script Output | Query Result

All Rows Fetched: 69805 in 2,193 seconds

	NAZW...	CH...	ROK	WIZYTA_ID	Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu wizyty	Udział %...	Łaczna wartość zapadnięć na tę chorobę w tym roku	Udział % ...	Łaczna ...	Grup...	IMIE	NAZWISKO
1	Grypa	9984	2020	11531	295	100	295	6,071	4859	AB-	Milena	Kucharska
2	Grypa	9984	2020	56436	294	99,661	295	6,071	4859	B-	Emil	Malinowski
3	Grypa	9984	2020	56433	293	99,322	295	6,071	4859	0-	Dagmara	Urbanska
4	Grypa	9984	2020	56432	292	98,983	295	6,071	4859	AB-	Milena	Kucharska
5	Grypa	9984	2020	59949	291	98,644	295	6,071	4859	AB-	Milena	Kucharska
6	Grypa	9984	2020	44615	290	98,305	295	6,071	4859	AB-	Agata	Zakrzewska

  - Hurtownia

```
--Wypisuje nazwę choroby, jej ID , Rok przeszukiwań danych,Ilość znalezionych rekordów od początku tabeli do aktualnego rekordu z uwzględnieniem nazwy oraz roku, oraz dane Pacjenta
SELECT DISTINCT
c.nazwa AS Nazwa_Choroby,c.choroby_id,EXTRACT(YEAR FROM c.poczatek) Rok,w.wizyta_id,
COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
AS "Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu  wizyty",
ROUND(100 * COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
/ COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ) , 3) "Udział % w tym roku",
COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ) "Łaczna wartość zapadnięć na tę chorobę w tym roku",
ROUND(100 * COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ) / COUNT(*) OVER (PARTITION BY c.nazwa) , 3) "Udział % na tle lat",
COUNT(*) OVER (PARTITION BY c.nazwa) "Łaczna wartość zapadnięć na tę chorobę na przestrzeni lat",
p.grupa_krwi "Grupa Krwi Pacjenta",p.imie,p.nazwisko
FROM h_choroby c
JOIN h_recepty r ON r.choroba_id = c.choroby_id
JOIN h_wizyty w ON w.recepta_id = r.recepta_id
JOIN h_daty_wizyt d ON d.data_id = w.data_wizyty_id
JOIN h_pacjenci p ON p.pacjent_id = w.pacjent_id
ORDER BY Rok DESC,COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.poczatek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) DESC;
```

Script Output | Query Result

All Rows Fetched: 69805 in 2,116 seconds

	NAZW...	CH...	ROK	WIZ...	Ilość zapadnięć na tę choro...	Udział % ...	Łaczna wartość zapadni...	Udział % ...	Łaczna wartoś...	Gr...	IMIE	NAZWISKO
1	Grypa	9984	2020	11531	295	100	295	6,071	4859	AB-	Milena	Kucharska
2	Grypa	9984	2020	70696	294	99,661	295	6,071	4859	AB-	Milena	Kucharska
3	Grypa	9984	2020	59949	293	99,322	295	6,071	4859	AB-	Milena	Kucharska
4	Grypa	9984	2020	56436	292	98,983	295	6,071	4859	B-	Emil	Malinowski
5	Grypa	9984	2020	56433	291	98,644	295	6,071	4859	0-	Dagmara	Urbanska
6	Grypa	9984	2020	56432	290	98,305	295	6,071	4859	AB-	Milena	Kucharska

Te dwa zapytania są praktycznie identyczne pod względem złożoności zapytania ,natomiast wersja dla hurtowni wykonuje się około 4.5% szybciej od wersji dla bazy.

- Okna Czasowe nr3.
  - Baza

```
--Oplaty,ceny leków i zabiegów dla każdej wizyty oraz sumaryczna wartość dotychczasowych wizyt
SELECT DISTINCT
w.wizyta_id,p.pesel_id,p.imie,p.nazwisko,w.oplata "Oplata za wizytę pacjenta",
SUM(pz.odplatnosc) OVER (PARTITION BY w.wizyta_id) "Cena leków",
(u.procent_ulgi / 100 * (SUM(pz.odplatnosc) OVER (PARTITION BY w.wizyta_id))) "Cena leków Po Uldze",
SUM(w.oplata) OVER ( ORDER BY w.wizyta_id RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych oplat za wizyty pacjentów",
z.cena_netto "Oplata za zabieg pacjenta",
SUM(z.cena_netto ) OVER (ORDER BY w.wizyta_id RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych oplat za zabiegi"
FROM wizyty w
LEFT JOIN zabiegi z ON w.wizyta_id = z.wizyta_id
JOIN pacjenci p ON p.pacjent_id = w.pacjent_id
LEFT JOIN recepty r ON r.wizyta_id = w.wizyta_id
LEFT JOIN pozycje_recept pz ON pz.recepta_id = r.recepta_id
LEFT JOIN ulgi u ON u.ulgi_id = r.ulga_id
ORDER BY w.wizyta_id ASC;
```

All Rows Fetched: 100000 in 2,898 seconds

	W...	PESEL_ID	IMIE	NAZWISKO	Oplata...	Cena leków	Cena leków Po Uldze	Suma wszystkich dotychczasowy...	Oplata za zabieg pacjenta	Suma wszystkich dotychcza...
1	1	60785675534	Pawel	Kubiak	200	(null)	(null)	200	(null)	(null)
2	2	84134527845	Oktawian	Szewczyk	1500	616,93	(null)	4700	(null)	(null)
3	3	71851685454	Andrzej	Wysocki	100	(null)	(null)	4800	(null)	(null)
4	4	32763847807	Maja	Sikorska	1500	365,25	182,625	9300	(null)	(null)
5	5	21444450160	Jola	Kubiak	1000	1029,4	(null)	13300	(null)	(null)
6	6	40644227778	Alan	Gorski	300	628,91	(null)	14800	(null)	(null)
7	7	57345800338	Milena	Szczepanska	300	830,88	(null)	16000	(null)	(null)
8	8	82332500460	Konrad	Urbanski	1300	(null)	(null)	17300	(null)	(null)
9	9	07755777815	Franciszka	Pietrzak	1500	(null)	(null)	18800	(null)	(null)
10	10	66580430051	Eryk	Walczak	500	(null)	(null)	19300	(null)	(null)
11	11	88657163435	Alicja	Piotrowska	1500	(null)	(null)	20800	2000	2000
12	12	30014284085	Bartosz	Duda	500	(null)	(null)	21300	(null)	2000

  - Hurtownia

```
--Oplaty,ceny leków i zabiegów dla każdej wizyty oraz sumaryczna wartość dotychczasowych wizyt
SELECT DISTINCT
w.wizyta_id,p.pesel,p.imie,p.nazwisko,w.oplata "Oplata za wizytę pacjenta",
SUM(pz.odplatnosc) OVER (PARTITION BY w.wizyta_id) "Cena leków",
(pz.procent_ulgi / 100 * (SUM(pz.odplatnosc) OVER (PARTITION BY w.wizyta_id))) "Cena leków Po Uldze",
SUM(w.oplata) OVER ( ORDER BY w.wizyta_id RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych oplat za wizyty pacjentów",
w.cena_netto_za_zabieg "Oplata za zabieg pacjenta",
SUM(w.cena_netto_za_zabieg ) OVER (ORDER BY w.wizyta_id RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych oplat za zabiegi"
FROM h_wizyty w
JOIN h_pacjenci p ON p.pacjent_id = w.pacjent_id
LEFT JOIN h_recepty r ON r.recepta_id = w.recepta_id
LEFT JOIN h_pozycje_recept pz ON pz.recepta_id = r.recepta_id
ORDER BY w.wizyta_id ASC;
```

All Rows Fetched: 100000 in 2,954 seconds

	WIZYTA_ID	PESEL	IMIE	NAZWISKO	Oplata za ...	Cena leków	Cena leków Po Uldze	Suma wszystkich dotyc...	Oplata za zabieg pac...	Suma wszystkich dotychczasowych opl...
1	1	60785675534	Pawel	Kubiak	200	(null)	(null)	200	(null)	(null)
2	2	84134527845	Oktawian	Szewczyk	1500	616,93	(null)	4700	(null)	(null)
3	3	71851685454	Andrzej	Wysocki	100	(null)	(null)	4800	(null)	(null)
4	4	32763847807	Maja	Sikorska	1500	365,25	182,625	9300	(null)	(null)
5	5	21444450160	Jola	Kubiak	1000	1029,4	(null)	13300	(null)	(null)
6	6	40644227778	Alan	Gorski	300	628,91	(null)	14800	(null)	(null)
7	7	57345800338	Milena	Szczepanska	300	830,88	(null)	16000	(null)	(null)
8	8	82332500460	Konrad	Urbanski	1300	(null)	(null)	17300	(null)	(null)
9	9	07755777815	Franciszka	Pietrzak	1500	(null)	(null)	18800	(null)	(null)
10	10	66580430051	Eryk	Walczak	500	(null)	(null)	19300	(null)	(null)
11	11	88657163435	Alicja	Piotrowska	1500	(null)	(null)	20800	2000	2000
12	12	30014284085	Bartosz	Duda	500	(null)	(null)	21300	(null)	2000

Te dwie wersje zapytań różnią się tym, że baza posiada o 2 złączenia więcej ,natomiast hurtownia wykonuje się prawie 2% wolniej.

- Funkcje Rankingowe nr1.
  - Baza

```sql
-- Ranking pacjentow, ktorzy wniesli najwiecej oplat za wizyte w danej placowce, w danym miescie
SELECT pac.nazwisko AS nazwisko_pacjenta, p.nazwa AS nazwa_placowki, a.miasto, w.oplata AS oplata_za_wizyte,
RANK () OVER (PARTITION BY p.nazwa, a.miasto ORDER BY w.oplata DESC)
ranking FROM wizyty w
JOIN pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
JOIN adresy a ON a.adres_id = p.adres_id
ORDER BY pac.nazwisko, ranking;
```

Query... SQL | All Rows Fetched: 100000 in 2,961 seconds

	NAZWISKO_PACJENTA	NAZWA_PLACOWKI	MIASTO	OPLATA_ZA_WIZYTE	RANKING
1	Adamska	Adamed	Krakow	1500	1
2	Adamska	Adamed	Lublin	1500	1
3	Adamska	Centrum Medyczno-Diagnostyczne	Lublin	1500	1
4	Adamska	Adamed	Wroclaw	1500	1
5	Adamska	Adamed	Opole	1500	1
6	Adamska	Klinika Kardiochirurgi	Kielce	1500	1
7	Adamska	Centrum Medyczno-Diagnostyczne	Szczecin	1500	1
8	Adamska	Centrum Medyczno-Diagnostyczne	Olsztyn	1500	1
9	Adamska	Centrum Medyczno-Diagnostyczne	Poznan	1500	1
10	Adamska	Klinika Kardiochirurgi	Zielona Gora	1500	1
11	Adamska	Klinika Kardiochirurgi	Lublin	1500	1
12	Adamska	Klinika Kardiochirurgi	Rzeszow	1500	1
13	Adamska	Klinika Kardiochirurgi	Bydgoszcz	1500	1
14	Adamska	Medican	Lublin	1500	1
15	Adamska	Medican	Olsztyn	1500	1
16	Adamska	Medican	Bydgoszcz	1500	1
17	Adamska	Red-Med	Zielona Gora	1500	1
18	Adamska	Red-Med	Lodz	1500	1
19	Adamska	Red-Med	Lublin	1500	1

  - Hurtownia

```sql
-- Ranking pacjentow, ktorzy wniesli najwiecej oplat za wizyte w danej placowce, w danym miescie
SELECT pac.nazwisko AS nazwisko_pacjenta, p.nazwa AS nazwa_placowki, p.miasto, w.oplata AS oplata_za_wizyte,
RANK () OVER (PARTITION BY p.nazwa, p.miasto ORDER BY w.oplata DESC)
ranking FROM h_wizyty w
JOIN h_pacjenci pac ON pac.pacjent_id = w.pacjent_id
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
ORDER BY pac.nazwisko, ranking;
```

Query Result SQL | All Rows Fetched: 100000 in 2,229 seconds

	NAZWISKO_PACJENTA	NAZWA_PLACOWKI	MIASTO	OPLATA_ZA_WIZYTE	RANKING
1	Adamska	Adamed	Krakow	1500	1
2	Adamska	Adamed	Lublin	1500	1
3	Adamska	Centrum Medyczno-Diagnostyczne	Lublin	1500	1
4	Adamska	Adamed	Wroclaw	1500	1
5	Adamska	Adamed	Opole	1500	1
6	Adamska	Klinika Kardiochirurgi	Kielce	1500	1
7	Adamska	Centrum Medyczno-Diagnostyczne	Szczecin	1500	1
8	Adamska	Centrum Medyczno-Diagnostyczne	Olsztyn	1500	1
9	Adamska	Centrum Medyczno-Diagnostyczne	Poznan	1500	1
10	Adamska	Klinika Kardiochirurgi	Zielona Gora	1500	1
11	Adamska	Klinika Kardiochirurgi	Lublin	1500	1
12	Adamska	Klinika Kardiochirurgi	Rzeszow	1500	1
13	Adamska	Klinika Kardiochirurgi	Bydgoszcz	1500	1
14	Adamska	Medican	Lublin	1500	1
15	Adamska	Medican	Olsztyn	1500	1
16	Adamska	Medican	Bydgoszcz	1500	1
17	Adamska	Red-Med	Zielona Gora	1500	1
18	Adamska	Red-Med	Lodz	1500	1
19	Adamska	Red-Med	Lublin	1500	1

Powyższe zapytania różnią się tym, że zapytanie z bazy posiada o 1 złączenie więcej niż zapytanie z hurtowni. W hurtowni dane z tabeli Adresy zostały zawarte w tabeli zewnętrznej wymiaru Gabinety. Zapytanie z hurtowni wykonuje się o 25% szybciej niż zapytanie z bazy.

- Funkcje Rankingowe nr2.
  - Baza

```sql
-- Ranking najlepszej sredniej sprzedazy lekow na dana chorobe z dana ulga
SELECT pr.nazwa AS nazwa_leku, c.nazwa AS nazwa_choroby, u.typ_ulgi, AVG (pr.ilosc),
RANK () OVER (PARTITION BY c.nazwa, u.typ_ulgi ORDER BY AVG (pr.ilosc) DESC)
ranking FROM pozycje_recept pr
JOIN recepty r ON r.recepta_id = pr.recepta_id
JOIN ulgi u ON u.ulgi_id = r.ulga_id
JOIN choroby c ON c.choroby_id = r.recepta_choroba_id
GROUP BY ROLLUP (pr.nazwa, c.nazwa, u.typ_ulgi)
ORDER BY c.nazwa, ranking;
```

Query Result

SQL | All Rows Fetched: 560 in 0,523 seconds

	NAZWA_LEKU	NAZWA_CHOROBY	TYP_ULGI	AVG(PR.ILOSC)	RANKING
1	Efferalgan Codeine	Alergia	(null)	3,211267605633802816901408450704225352 11	1
2	Gynoxin Uno	Alergia	Wiekowa50+	3,341463414634146341463414634146341	1
3	Efferalgan Codeine	Alergia	Niepelnosprawnosciowa	3,44	1
4	Abaktal	Alergia	(null)	3,196969696969696969696969696969696969697	2
5	Abaktal	Alergia	Niepelnosprawnosciowa	3,32	2
6	Keppra	Alergia	Wiekowa50+	3,270833333333333333333333333333333333	2
7	Cachexan	Alergia	Niepelnosprawnosciowa	3,304347826086956521739130434782608695 65	3
8	Keppra	Alergia	(null)	3,154929577464788732394366197183098591 55	3
9	Ibuprom Max	Alergia	Wiekowa50+	3,270270270270270270270270270270270270 27	3
10	Gynoxin Uno	Alergia	(null)	3,152777777777777777777777777777777777778	4
11	Lakcid	Alergia	Niepelnosprawnosciowa	3,193548387096774193548387096774193548 39	4
12	Abaktal	Alergia	Wiekowa50+	3,121951219512195121951219512195121951 22	4
13	Jovesto	Alergia	Niepelnosprawnosciowa	3,142857142857142857142857142857142857 14	5
14	Malidum	Alergia	Wiekowa50+	3,108695652173913043478260869565217391 3	5
15	Malidum	Alergia	(null)	3,109589041095890410958904109589041095 89	5
16	Lakcid	Alergia	(null)	3,093333333333333333333333333333333333333	6
17	Malidum	Alergia	Niepelnosprawnosciowa	3,111111111111111111111111111111111111111	6
18	Efferalgan Codeine	Alergia	Wiekowa50+	3,086956521739130434782608695652173913 04	6
19	Fastum	Alergia	Niepelnosprawnosciowa	3,04	7

  - Hurtownia

```sql
-- Ranking najlepszej sredniej sprzedazy lekow na dana chorobe z dana ulga
SELECT l.nazwa AS nazwa_leku, c.nazwa AS nazwa_choroby, u.typ_ulgi, AVG (pr.ilosc),
RANK () OVER (PARTITION BY c.nazwa, u.typ_ulgi ORDER BY AVG (pr.ilosc) DESC)
ranking FROM h_pozycje_recept pr
JOIN h_leki l ON l.leki_id = pr.lek_id
JOIN h_ulgi u ON u.ulgi_id = pr.ulga_id
JOIN h_recepty r ON r.recepta_id = pr.recepta_id
JOIN h_choroby c ON c.choroby_id = r.choroba_id
GROUP BY ROLLUP (l.nazwa, c.nazwa, u.typ_ulgi)
ORDER BY c.nazwa, ranking;
```

Query Result

SQL | All Rows Fetched: 560 in 0,082 seconds

	NAZWA_LEKU	NAZWA_CHOROBY	TYP_ULGI	AVG(PR.ILOSC)	RANKING
1	Efferalgan Codeine	Alergia	(null)	3,211267605633802816901408450704225352 11	1
2	Gynoxin Uno	Alergia	Wiekowa50+	3,341463414634146341463414634146341	1
3	Efferalgan Codeine	Alergia	Niepelnosprawnosciowa	3,44	1
4	Abaktal	Alergia	(null)	3,196969696969696969696969696969696969697	2
5	Abaktal	Alergia	Niepelnosprawnosciowa	3,32	2
6	Keppra	Alergia	Wiekowa50+	3,270833333333333333333333333333333333	2
7	Cachexan	Alergia	Niepelnosprawnosciowa	3,304347826086956521739130434782608695 65	3
8	Keppra	Alergia	(null)	3,154929577464788732394366197183098591 55	3
9	Ibuprom Max	Alergia	Wiekowa50+	3,270270270270270270270270270270270270 27	3
10	Gynoxin Uno	Alergia	(null)	3,152777777777777777777777777777777777778	4
11	Lakcid	Alergia	Niepelnosprawnosciowa	3,193548387096774193548387096774193548 39	4
12	Abaktal	Alergia	Wiekowa50+	3,121951219512195121951219512195121951 22	4
13	Jovesto	Alergia	Niepelnosprawnosciowa	3,142857142857142857142857142857142857 14	5
14	Malidum	Alergia	Wiekowa50+	3,108695652173913043478260869565217391 3	5
15	Malidum	Alergia	(null)	3,109589041095890410958904109589041095 89	5
16	Lakcid	Alergia	(null)	3,093333333333333333333333333333333333333	6
17	Malidum	Alergia	Niepelnosprawnosciowa	3,111111111111111111111111111111111111111	6
18	Efferalgan Codeine	Alergia	Wiekowa50+	3,086956521739130434782608695652173913 04	6
19	Fastum	Alergia	Niepelnosprawnosciowa	3,04	7

Powyższe zapytania różnią się tym, że zapytanie z bazy posiada o 1 złączenie mniej niż zapytanie z hurtowni. Do hurtowni został dołączony wymiar Leki, który określa nazwę danego leku. Niemniej jednak zapytanie z hurtowni wykonuje się o 537% szybciej.

- Funkcje Rankingowe nr3.
  - Baza

```sql
-- Ranking zabiegow, ktore zostaly wykonane przez neurologa, w danej placowce oraz w danym miescie
SELECT z.nazwa AS nazwa_zabiegu, pr.nazwisko, p.nazwa AS nazwa_placowki, a.miasto, z.cena_netto AS oplata_za_zabieg,
RANK () OVER (PARTITION BY p.nazwa, a.miasto ORDER BY z.cena_netto DESC)
ranking FROM wizyty w
JOIN zabiegi z ON z.wizyta_id = w.wizyta_id
JOIN pracownicy pr ON pr.pracownik_id = w.prac_spec
JOIN stanowiska s ON s.stanowisko_id = pr.stanowisko_id
JOIN gabinety g ON g.gabinet_id = w.gabinet_id
JOIN placowki p ON p.placowka_id = g.placowka_id
JOIN adresy a ON a.adres_id = p.adres_id
WHERE s.nazwa LIKE 'Neurolog'
ORDER BY z.nazwa, ranking;
```

Query Result

SQL | All Rows Fetched: 1786 in 0,059 seconds

	NAZWA_ZABIEGU	NAZWISKO	NAZWA_PLACOWKI	MIASTO	OPLATA_ZA_ZABIEG	RANKING
1	abdominoplastyka	Urbanski	Unimed	Warszawa	5000	1
2	abdominoplastyka	Sikorska	Centrum Medyczno-Diagnostyczne	Opole	4000	2
3	abdominoplastyka	Nowak	Podimed	Bydgoszcz	4500	3
4	abdominoplastyka	Wojcik	Centrum Medyczno-Diagnostyczne	Zielona Gora	4500	3
5	abdominoplastyka	Wozniak	Unimed	Olsztyn	3000	4
6	abdominoplastyka	Nowak	Podimed	Opole	3500	4
7	abdominoplastyka	Wysocki	Red-Med	Katowice	4000	4
8	abdominoplastyka	Jaworska	Centrum Medyczno-Diagnostyczne	Bydgoszcz	3500	6
9	abdominoplastyka	Sikora	Optimed	Rzeszow	4500	6
10	abdominoplastyka	Szewczyk	Centrum Medyczno-Diagnostyczne	Poznan	2000	7
11	abdominoplastyka	Witkowski	Red-Med	Warszawa	1500	7
12	abdominoplastyka	Kubiak	NZOZ Danea	Szczecin	1000	9
13	abdominoplastyka	Kozlowski	NZOZ Danea	Gdansk	2500	9
14	abdominoplastyka	Pietrzak	Medican	Zielona Gora	3000	9
15	abdominoplastyka	Chmielewska	NZOZ Danea	Lublin	4500	9
16	abdominoplastyka	Malinowski	Voxel Centrum	Warszawa	2000	11
17	abdominoplastyka	Wysocki	Red-Med	Katowice	500	14

  - Hurtownia

```sql
-- Ranking zabiegow, ktore zostaly wykonane przez neurologa, w danej placowce oraz w danym miescie
SELECT z.nazwa AS nazwa_zabiegu, pr.nazwisko, p.nazwa AS nazwa_placowki, p.miasto, w.cena_netto_za_zabieg AS oplata_za_zabieg,
RANK () OVER (PARTITION BY p.nazwa, p.miasto ORDER BY w.cena_netto_za_zabieg DESC)
ranking FROM h_wizyty w
JOIN h_zabiegi z ON w.zabieg_id = z.zabieg_id
JOIN h_pracownicy pr ON pr.pracownik_id = w.prac_spec
JOIN h_stanowiska s ON s.stanowisko_id = pr.stanowisko_id
JOIN h_gabinety g ON g.gabinet_id = w.gabinet_id
JOIN h_placowki p ON p.placowka_id = g.placowka_id
WHERE s.nazwa LIKE 'Neurolog'
ORDER BY z.nazwa, ranking;
```

Query Result

SQL | All Rows Fetched: 1786 in 0,32 seconds

	NAZWA_ZABIEGU	NAZWISKO	NAZWA_PLACOWKI	MIASTO	OPLATA_ZA_ZABIEG	RANKING
1	abdominoplastyka	Urbanski	Unimed	Warszawa	5000	1
2	abdominoplastyka	Sikorska	Centrum Medyczno-Diagnostyczne	Opole	4000	2
3	abdominoplastyka	Nowak	Podimed	Bydgoszcz	4500	3
4	abdominoplastyka	Wojcik	Centrum Medyczno-Diagnostyczne	Zielona Gora	4500	3
5	abdominoplastyka	Wozniak	Unimed	Olsztyn	3000	4
6	abdominoplastyka	Nowak	Podimed	Opole	3500	4
7	abdominoplastyka	Wysocki	Red-Med	Katowice	4000	4
8	abdominoplastyka	Jaworska	Centrum Medyczno-Diagnostyczne	Bydgoszcz	3500	6
9	abdominoplastyka	Sikora	Optimed	Rzeszow	4500	6
10	abdominoplastyka	Szewczyk	Centrum Medyczno-Diagnostyczne	Poznan	2000	7
11	abdominoplastyka	Witkowski	Red-Med	Warszawa	1500	7
12	abdominoplastyka	Kubiak	NZOZ Danea	Szczecin	1000	9
13	abdominoplastyka	Kozlowski	NZOZ Danea	Gdansk	2500	9
14	abdominoplastyka	Pietrzak	Medican	Zielona Gora	3000	9
15	abdominoplastyka	Chmielewska	NZOZ Danea	Lublin	4500	9
16	abdominoplastyka	Malinowski	Voxel Centrum	Warszawa	2000	11

Powyższe zapytania różnią się tym, że zapytanie z bazy posiada o 1 złączenie więcej niż zapytanie z hurtowni. W hurtowni dane z tabeli Adresy zostały zawarte w tabeli zewnętrznej wymiaru Gabinety. Zapytanie z hurtowni wykonuje się o 81,6% wolniej niż zapytanie z bazy.

9. Uwagi i wnioski
    - Zdecydowaliśmy się na schemat konstelacji z dwoma tabelami faktów, ponieważ w ten sposób możemy osobno badać fakty wizyt i pozycji recept (leków), co przekłada się na wydajniejsze operacje ,kiedy badamy tylko konkretną tabelę faktów.
    - Pomiary szybkości zapytań wskazują na korzyść bazy do hurtowni w stosunku 9:6, choć mogą one się delikatnie różnić w zależności od wydajności systemu operacyjnego i ilości przerwań systemowych podczas wykonywania zapytań.
    - Czas trwania procedur transformacyjnych jest tak duży, ponieważ w naszej bazie danych znajduje się bardzo duża ilość rekordów w tabelach , i z tego powodu system potrzebuje więcej czasu na przerobienie danych i wstawienie do tabel w hurtowni.
    - Do wygenerowania tak dużej ilości rekordów użyliśmy własnoręcznie napisanego programu generującego w języku Java . Znaczna ilość danych w tabelach jest powiązana dosyć logicznie poprzez generator , niemniej jednak spora liczba mniej istotnych danych jest losowa.