

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”

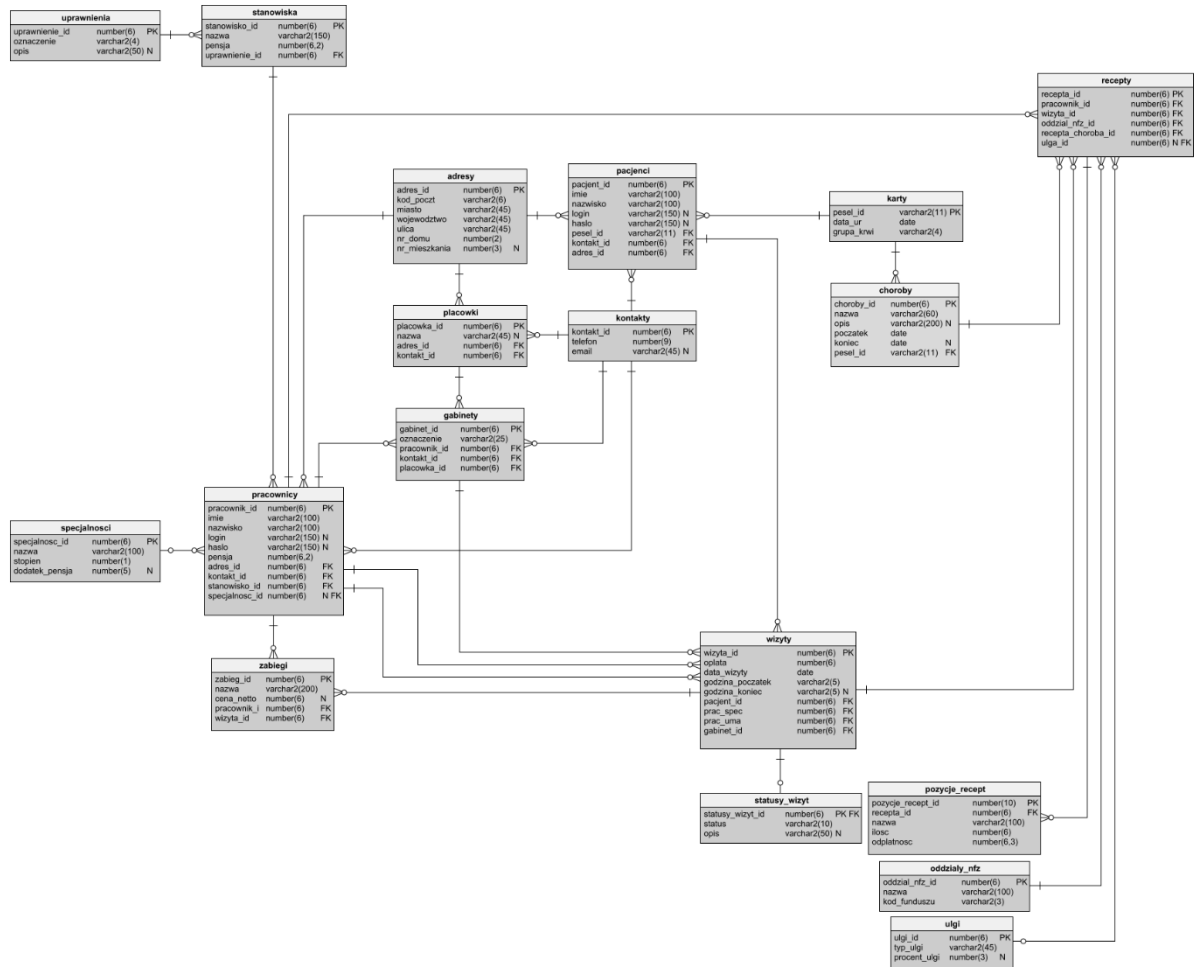
Podeszliśmy do tematu racjonalnie, projektując bazę danych ,która mogłaby faktycznie służyć w prawdziwych okolicznościach jako narzędzie do przechowywania danych ,potrzebnych do działania takich placówek lekarskich.

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



Vertabelo

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.

- Oddziały_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 pracownikowi, 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 odpłatność.

Fakt Wizyty ma klucze obce do tabel: Pracownicy, Statusy_Wizyt, Daty_Wizyt, Zabiegi, Pacjenci, Gabinety, Recepty oraz miary: opłata (za wizyte) i cena_netto_za_zabieg.

Opis wymiarów faktu Pozycje_recept:

- 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.

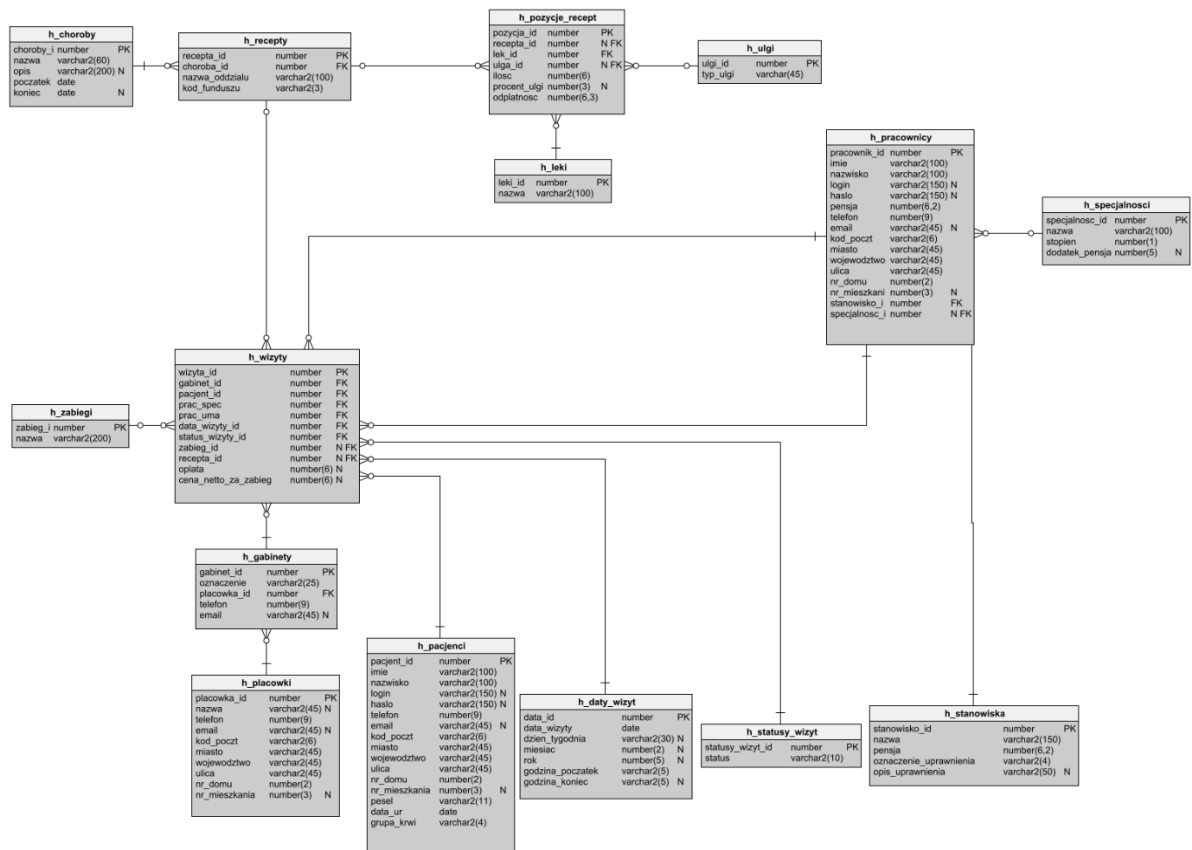
Opis wymiarów faktu Wizyty:

- Statusy_Wizyt – wymiar opisujący status wizyty.
- Pracownicy – wymiar opisujący pracownika związanego z wizytą. Zostały do niego wcielenne dane z dwóch tabel: Kontakty i Adresy.
- Recepty - wymiar współdzielony z tabelą faktów Pozycje_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 wcielenne 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 uprawnień). Wymiar Gabinety ma z kolei jedną tabelę zewnętrzną placówki (do placówek zostały włączone dane z tabel: Adresy i Kontakty). Natomiast wymiar Recepty posiada zewnętrzną tabelę Choroby.

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

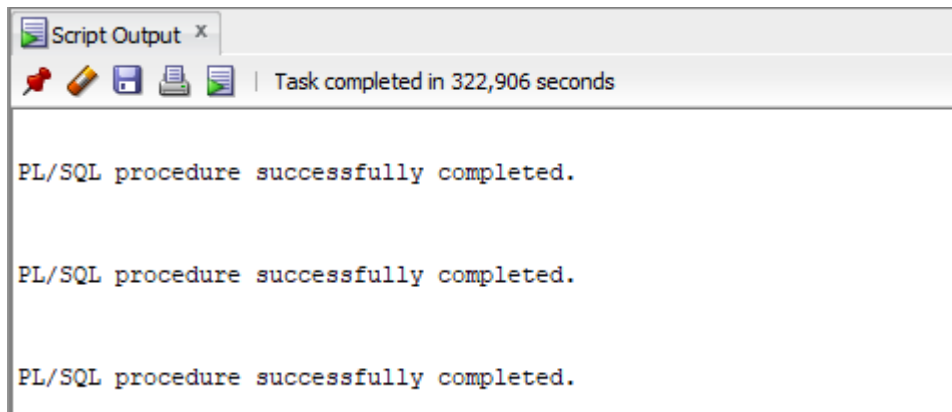
INSERT INTO h_pracownicy VALUES(P_ID,pracownik.imie,pracownik.nazwisko,pracownik.login,pracownik.haslo,pracownik.pensja,
kontakt.telefon,kontakt.email,
adres.kod_poczt,adres.miasto,adres.wojewodztwo,adres.ulica,adres.nr_domu,adres.nr_mieszkania,
pracownik.stanowisko_id,pracownik.specjalnosc_id);

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 cursor 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.



The screenshot shows a 'Script Output' window with a title bar 'Script Output x'. Below the title bar is a toolbar with icons for a red pushpin, a yellow pencil, a blue save icon, a printer icon, and a green play icon. To the right of the toolbar, it says 'Task completed in 322,906 seconds'. The main area of the window contains three lines of text, each on a new line: 'PL/SQL procedure successfully completed.', 'PL/SQL procedure successfully completed.', and 'PL/SQL procedure successfully completed.'.

```
Script Output x
Task completed in 322,906 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```

8. Porównanie wykonania zapytań

- Rollup nr1.
 - Baza

```
-- Ilosc wizyt ze statusem 'zakonczone' w kazdym gabinecie
SELECT DISTINCT
NVL (TO_CHAR ((SELECT oznaczenie FROM gabinety WHERE gabinet_id = g_id)), 'Laczna ilosc wizyt') oznaczenie_gabinetu,
NVL (TO_CHAR ((SELECT status FROM statusy_wizyt WHERE statusy_wizyt_id = sw_id AND status LIKE 'Zakonczone')), 'Ilosc wizyt w gabinecie') status_wizyty,
ilosc_wizyt
FROM (
SELECT w.gabinet_id g_id, sw.statusy_wizyt_id sw_id, COUNT (w.wizyta_id) ilosc_wizyt FROM statusy_wizyt sw
JOIN wizyty w ON w.wizyta_id = sw.statusy_wizyt_id
GROUP BY ROLLUP (w.gabinet_id, sw.statusy_wizyt_id)
);
```

Query Result 1 x

All Rows Fetched: 93 in 0,302 seconds

OZNACZENIE_GABINETU	STATUS_WIZYT	ILOSC_WIZYT
1 Wizyty	Ilosc wizyt w gabinecie	14
2 Wizyty kontrolne	Ilosc wizyt w gabinecie	16
3 Wizyty	Ilosc wizyt w gabinecie	13
4 Wizyty	Ilosc wizyt w gabinecie	9
5 Wizyty	Ilosc wizyt w gabinecie	4
6 Zabiegi	Ilosc wizyt w gabinecie	14
7 Zabiegi	Ilosc wizyt w gabinecie	13
8 Wizyty	Ilosc wizyt w gabinecie	6
9 Badania	Ilosc wizyt w gabinecie	4
10 Zabiegi	Ilosc wizyt w gabinecie	4
11 Wizyty	Ilosc wizyt w gabinecie	2
12 Badania	Ilosc wizyt w gabinecie	2
13 Wizyty kontrolne	Ilosc wizyt w gabinecie	26
14 Badania	Ilosc wizyt w gabinecie	1
15 Badania	Ilosc wizyt w gabinecie	11
16 Wizyty kontrolne	Ilosc wizyt w gabinecie	7
17 Badania	Ilosc wizyt w gabinecie	6
18 Wizyty	Ilosc wizyt w gabinecie	11

- Hurtownia

```
-- Ilosc wizyt ze statusem 'zakonczone' w kazdym gabinecie
SELECT DISTINCT
NVL (TO_CHAR ((SELECT oznaczenie FROM h_gabinety WHERE gabinet_id = g_id)), 'Laczna ilosc wizyt') oznaczenie_gabinetu,
NVL (TO_CHAR ((SELECT status FROM h_statusy_wizyt WHERE statusy_wizyt_id = sw_id AND status LIKE 'Zakonczone')), 'Ilosc wizyt w gabinecie') status_wizyty,
ilosc_wizyt
FROM (
SELECT gabinet_id g_id, status_wizyty_id sw_id, COUNT (wizyta_id) ilosc_wizyt FROM h_wizyty
GROUP BY ROLLUP (gabinet_id, status_wizyty_id)
);
```

Query Result x

All Rows Fetched: 93 in 0,269 seconds

OZNACZENIE_GABINETU	STATUS_WIZYT	ILOSC_WIZYT
1 Wizyty	Ilosc wizyt w gabinecie	14
2 Wizyty kontrolne	Ilosc wizyt w gabinecie	16
3 Wizyty	Ilosc wizyt w gabinecie	13
4 Wizyty	Ilosc wizyt w gabinecie	9
5 Wizyty	Ilosc wizyt w gabinecie	4
6 Zabiegi	Ilosc wizyt w gabinecie	14
7 Zabiegi	Ilosc wizyt w gabinecie	13
8 Wizyty	Ilosc wizyt w gabinecie	6
9 Badania	Ilosc wizyt w gabinecie	4
10 Zabiegi	Ilosc wizyt w gabinecie	4
11 Wizyty	Ilosc wizyt w gabinecie	2
12 Badania	Ilosc wizyt w gabinecie	2
13 Wizyty kontrolne	Ilosc wizyt w gabinecie	26
14 Badania	Ilosc wizyt w gabinecie	1
15 Badania	Ilosc wizyt w gabinecie	11
16 Wizyty kontrolne	Ilosc wizyt w gabinecie	7
17 Badania	Ilosc wizyt w gabinecie	6
18 Wizyty	Ilosc wizyt w gabinecie	11

Różnicą między zapytaniem z bazy, a zapytaniem z hurtowni jest to ,że w bazie wymaga ono o jedno więcej połączenie JOIN. W hurtowni to zapytanie wykonuje się o 0,033s szybciej.

- Rollup nr2.
 - Baza

```
-- Liczba pracowników pracujących na danym stanowisku, posiadających określone specjalności, mieszkających w danym mieście
SELECT DISTINCT
  NVL (TO_CHAR ((SELECT nazwa FROM stanowiska WHERE stanowisko_id = s_id)), 'Łączna ilość pracowników') nazwa_stanowiska,
  NVL (TO_CHAR ((SELECT nazwa FROM specjalnosci WHERE specjalnosc_id = sp_id)), 'Ilość pracowników na danym stanowisku') nazwa_specjalnosci,
  NVL (adres, 'Ilość pracowników na danym stanowisku z okreslna spec.') miasto,
  liczba_pracownikow
FROM (
  SELECT p.stanowisko_id s_id, p.specjalnosc_id sp_id, a.miasto adres, COUNT (p.pracownik_id) liczba_pracownikow FROM pracownicy p
  JOIN adresy a ON a.adres_id = p.adres_id
  GROUP BY ROLLUP (p.stanowisko_id, p.specjalnosc_id, a.miasto)
);
```

NAZWA_STANOWISKA	NAZWA_SPECJALNOSCI	MIASTO	LICZBA_PRACOWNIKOW
1 Pielęgniarz	Ilość pracowników na danym stanowisku	Zielona Góra	1
2 Konserwator	Medycyna pracy	Bydgoszcz	1
3 Konserwator	Medycyna pracy	Ilość pracowników na danym stanowisku z okreslna spec.	1
4 Denstysta	Seksuologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
5 Ginekolog	Chirurgia klatki piersiowej	Opole	1
6 Lekarz pediatra	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	1
7 Pielęgniarz	Otarynolaryngologia	Gdańsk	1
8 Denstysta	Endokrynologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
9 Urolog	Medycyna lotnicza	Ilość pracowników na danym stanowisku z okreslna spec.	1
10 Denstysta	Stomatologia dziecięca	Rzeszów	1
11 Neurolog	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	2
12 Laborant	Pediatra	Ilość pracowników na danym stanowisku z okreslna spec.	1
13 Kardiolog	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	1
14 Immunolog	Gastroenterologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
15 Immunolog	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	1
16 Konserwator	Hipertensjologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
17 Pielęgniarz	Chirurgia szczękowo-twarzowa	Poznań	1
18 Urolog	Medycyna lotnicza	Lublin	1

- Hurtownia

```
-- Liczba pracowników pracujących na danym stanowisku, posiadających określone specjalności, mieszkających w danym mieście
SELECT DISTINCT
  NVL (TO_CHAR ((SELECT nazwa FROM h_stanowiska WHERE stanowisko_id = s_id)), 'Łączna ilość pracowników') nazwa_stanowiska,
  NVL (TO_CHAR ((SELECT nazwa FROM h_specjalnosci WHERE specjalnosc_id = sp_id)), 'Ilość pracowników na danym stanowisku') nazwa_specjalnosci,
  NVL (adres, 'Ilość pracowników na danym stanowisku z okreslna spec.') miasto,
  liczba_pracownikow
FROM (
  SELECT p.stanowisko_id s_id, p.specjalnosc_id sp_id, p.miasto adres, COUNT (p.pracownik_id) liczba_pracownikow FROM h_pracownicy p
  GROUP BY ROLLUP (p.stanowisko_id, p.specjalnosc_id, p.miasto)
);
```

NAZWA_STANOWISKA	NAZWA_SPECJALNOSCI	MIASTO	LICZBA_PRACOWNIKOW
1 Pielęgniarz	Ilość pracowników na danym stanowisku	Zielona Góra	1
2 Konserwator	Medycyna pracy	Bydgoszcz	1
3 Konserwator	Medycyna pracy	Ilość pracowników na danym stanowisku z okreslna spec.	1
4 Denstysta	Seksuologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
5 Ginekolog	Chirurgia klatki piersiowej	Opole	1
6 Lekarz pediatra	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	1
7 Pielęgniarz	Otarynolaryngologia	Gdańsk	1
8 Denstysta	Endokrynologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
9 Urolog	Medycyna lotnicza	Ilość pracowników na danym stanowisku z okreslna spec.	1
10 Denstysta	Stomatologia dziecięca	Rzeszów	1
11 Neurolog	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	2
12 Laborant	Pediatra	Ilość pracowników na danym stanowisku z okreslna spec.	1
13 Kardiolog	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	1
14 Immunolog	Gastroenterologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
15 Immunolog	Ilość pracowników na danym stanowisku	Ilość pracowników na danym stanowisku z okreslna spec.	1
16 Konserwator	Hipertensjologia	Ilość pracowników na danym stanowisku z okreslna spec.	1
17 Pielęgniarz	Chirurgia szczękowo-twarzowa	Poznań	1
18 Urolog	Medycyna lotnicza	Lublin	1

Różnicą między zapytaniem z bazy, a zapytaniem z hurtowni jest to, że w bazie wymaga ono o jedno więcej połączenie JOIN. W hurtowni to zapytanie wykonuje się o 0,013s szybciej.

- Rollup nr3.
 - Baza

```
-- Liczba pacjentow umowionych na wizyte, wymagajaca zabiegu
SELECT DISTINCT
  NVL (TO_CHAR ((SELECT nazwa FROM zabiegi WHERE zabieg_id = z_id)), 'Laczna ilosc pacjentow') nazwa_zabiegu,
  NVL (TO_CHAR ((SELECT status FROM statusy_wizyt WHERE statusy_wizyt_id = sw_id AND status LIKE 'Oczekujaca')), 'Ilosc wizyt wymagajacych zabiegu') status_wizyty,
  liczba_pacjentow
FROM (
  SELECT z.zabieg_id z_id, sw.statusy_wizyt_id sw_id, COUNT(w.pacjent_id) liczba_pacjentow FROM zabiegi z
  RIGHT JOIN wizyty w ON w.wizyta_id = z.wizyta_id
  JOIN statusy_wizyt sw ON sw.statusy_wizyt_id = w.wizyta_id
  GROUP BY ROLLUP (z.zabieg_id, sw.statusy_wizyt_id)
);
```

Query Result x

SQL | All Rows Fetched: 192 in 0,21 seconds

NAZWA_ZABIEGU	STATUS_WIZYT	LICZBA_PACJENTOW
1 operacja Nussa	Ilosc wizyt wymagajacych zabiegu	1
2 operacja Milesa	Ilosc wizyt wymagajacych zabiegu	1
3 hemikorporektomia	Ilosc wizyt wymagajacych zabiegu	1
4 operacja Norwooda	Ilosc wizyt wymagajacych zabiegu	1
5 adrenalektomia	Ilosc wizyt wymagajacych zabiegu	1
6 owarektomia	Ilosc wizyt wymagajacych zabiegu	1
7 operacja Mogga	Ilosc wizyt wymagajacych zabiegu	1
8 operacja Jurasza	Oczekujaca	1
9 laryngofisura	Ilosc wizyt wymagajacych zabiegu	1
10 appendektomia	Ilosc wizyt wymagajacych zabiegu	1
11 adenotomia	Ilosc wizyt wymagajacych zabiegu	1
12 ureterosigmoidostomia	Ilosc wizyt wymagajacych zabiegu	1
13 splenektomia	Ilosc wizyt wymagajacych zabiegu	1
14 cholecystektomia	Oczekujaca	1
15 plastyka Heinekego-Mikulicza	Oczekujaca	1
16 trachelektomia	Oczekujaca	1
17 operacja Halstedta	Oczekujaca	1
18 nanilotomia	Oczekujaca	1

- Hurtownia

```
-- Liczba pacjentow umowionych na wizyte, wymagajaca zabiegu
SELECT DISTINCT
  NVL (TO_CHAR ((SELECT nazwa FROM h_zabiegi WHERE zabieg_id = z_id)), 'Laczna ilosc pacjentow') nazwa_zabiegu,
  NVL (TO_CHAR ((SELECT status FROM h_statusy_wizyt WHERE statusy_wizyt_id = sw_id AND status LIKE 'Oczekujaca')), 'Ilosc wizyt wymagajacych zabiegu') status_wizyty,
  liczba_pacjentow
FROM (
  SELECT zabieg_id z_id, status_wizyty_id sw_id, COUNT (pacjent_id) liczba_pacjentow FROM h_wizyty
  GROUP BY ROLLUP (zabieg_id, status_wizyty_id)
);
```

Query Result x

SQL | All Rows Fetched: 192 in 0,186 seconds

NAZWA_ZABIEGU	STATUS_WIZYT	LICZBA_PACJENTOW
1 operacja Nussa	Ilosc wizyt wymagajacych zabiegu	1
2 operacja Milesa	Ilosc wizyt wymagajacych zabiegu	1
3 hemikorporektomia	Ilosc wizyt wymagajacych zabiegu	1
4 operacja Norwooda	Ilosc wizyt wymagajacych zabiegu	1
5 adrenalektomia	Ilosc wizyt wymagajacych zabiegu	1
6 owarektomia	Ilosc wizyt wymagajacych zabiegu	1
7 operacja Mogga	Ilosc wizyt wymagajacych zabiegu	1
8 operacja Jurasza	Oczekujaca	1
9 laryngofisura	Ilosc wizyt wymagajacych zabiegu	1
10 appendektomia	Ilosc wizyt wymagajacych zabiegu	1
11 adenotomia	Ilosc wizyt wymagajacych zabiegu	1
12 ureterosigmoidostomia	Ilosc wizyt wymagajacych zabiegu	1
13 splenektomia	Ilosc wizyt wymagajacych zabiegu	1
14 cholecystektomia	Oczekujaca	1
15 plastyka Heinekego-Mikulicza	Oczekujaca	1
16 trachelektomia	Oczekujaca	1
17 operacja Halstedta	Oczekujaca	1
18 papilotomia	Oczekujaca	1

Różnicą między zapytaniem z bazy, a zapytaniem z hurtowni jest to, że w bazie wymaga ono o 2 połączenia JOIN więcej. W hurtowni to zapytanie wykonuje się o 0,024s szybciej.

- Cube nr1.
 - Baza

```
-- Srednia oplata wizyt danego pacjenta, w kazdym gabinecie
```

```
SELECT DISTINCT
TO_CHAR ((SELECT nazwisko FROM pacjenci WHERE pacjent_id = p_id)) nazwisko_pacjenta,
TO_CHAR ((SELECT oznaczenie FROM gabinety WHERE gabinet_id = g_id)) oznaczenie_gabinetu,
srednia_oplata
FROM (
SELECT pacjent_id p_id, gabinet_id g_id, AVG (oplata) srednia_oplata FROM wizyty
GROUP BY CUBE (pacjent_id, gabinet_id)
);
```

Query Result x

SQL | All Rows Fetched: 12837 in 0,716 seconds

NAZWISKO_PACJENTA	OZNACZENIE_GABINETU	SREDNIA_OPLATA
1 (null)	Wizyty kontrolne	800
2 (null)	Wizyty kontrolne	1012,5
3 (null)	Wizyty kontrolne	787,5
4 (null)	Wizyty	740
5 (null)	Wizyty kontrolne	762,5
6 (null)	Wizyty	772,727272727272727272727272727273
7 (null)	Zabiegi	820
8 (null)	Zabiegi	772,727272727272727272727272727273
9 (null)	Zabiegi	922,222222222222222222222222222222
10 (null)	Badania	669,230769230769230769230769230769
11 (null)	Wizyty kontrolne	941,666666666666666666666666666667
12 (null)	Wizyty	825
13 (null)	Zabiegi	1028,571428571428571428571428571429
14 (null)	Wizyty kontrolne	785,714285714285714285714285714286
15 (null)	Wizyty kontrolne	890,909090909090909090909090909091
16 (null)	Wizyty kontrolne	1011,111111111111111111111111111111
17 (null)	Badania	833,333333333333333333333333333333
18 (null)	Zabiegi	733,333333333333333333333333333333
19 (null)	Zabiegi	960

- Hurtownia

```
-- Srednia oplata wizyt danego pacjenta, w kazdym gabinecie
```

```
SELECT DISTINCT
TO_CHAR ((SELECT nazwisko FROM h_pacjenci WHERE pacjent_id = p_id)) nazwisko_pacjenta,
TO_CHAR ((SELECT oznaczenie FROM h_gabinety WHERE gabinet_id = g_id)) oznaczenie_gabinetu,
srednia_oplata
FROM (
SELECT pacjent_id p_id, gabinet_id g_id, AVG (oplata) srednia_oplata FROM h_wizyty
GROUP BY CUBE (pacjent_id, gabinet_id)
);
```

Query Result x

SQL | All Rows Fetched: 12837 in 0,621 seconds

NAZWISKO_PACJENTA	OZNACZENIE_GABINETU	SREDNIA_OPLATA
1 (null)	Wizyty kontrolne	800
2 (null)	Wizyty kontrolne	1012,5
3 (null)	Wizyty kontrolne	787,5
4 (null)	Wizyty	740
5 (null)	Wizyty kontrolne	762,5
6 (null)	Wizyty	772,727272727272727272727272727273
7 (null)	Zabiegi	820
8 (null)	Zabiegi	772,727272727272727272727272727273
9 (null)	Zabiegi	922,222222222222222222222222222222
10 (null)	Badania	669,230769230769230769230769230769
11 (null)	Wizyty kontrolne	941,666666666666666666666666666667
12 (null)	Wizyty	825
13 (null)	Zabiegi	1028,571428571428571428571428571429
14 (null)	Wizyty kontrolne	785,714285714285714285714285714286
15 (null)	Wizyty kontrolne	890,909090909090909090909090909091
16 (null)	Wizyty kontrolne	1011,111111111111111111111111111111
17 (null)	Badania	833,333333333333333333333333333333
18 (null)	Zabiegi	733,333333333333333333333333333333

Wersja dla hurtowni jak i dla bazy jest taka sama, mimo to zapytanie do hurtowni danych wykonało się o 0,095s szybciej

- Cube nr2.
 - Baza

```

-- Srednia oplata zabiegow danego pacjenta z okreslonego oddzialu, w kazdym gabinecie
SELECT DISTINCT
TO_CHAR ((SELECT nazwisko FROM pacjenci WHERE pacjent_id = p_id)) nazwisko_pacjenta,
TO_CHAR ((SELECT oznaczenie FROM gabinety WHERE gabinet_id = g_id)) oznaczenie_gabinetu,
TO_CHAR ((SELECT nazwa FROM oddzialy_nfz o JOIN recepty r ON r.oddzial_nfz_id = o.oddzial_nfz_id WHERE r.recepta_id = r_id)) nazwa_oddzialu,
srednia_oplata
FROM (
  SELECT w.pacjent_id p_id, w.gabinet_id g_id, r.recepta_id r_id, AVG (z.cena_netto) srednia_oplata FROM zabiegi z
  RIGHT JOIN wizyty w ON w.wizyta_id = z.wizyta_id
  LEFT JOIN recepty r ON r.wizyta_id = w.wizyta_id
  GROUP BY CUBE (w.pacjent_id, w.gabinet_id, r.recepta_id)
);

```

Query Result x

SQL | All Rows Fetched: 23622 in 1,948 seconds

	NAZWISKO_PACJENTA	OZNACZENIE_GABINETU	NAZWA_ODDZIALU	SREDNIA_OPLATA
1	(null)	(null)	Mazowiecki Oddzial Narodowego Funduszu Zdrowia w Warszawie	(null)
2	(null)	(null)	Pomorski Oddzial Narodowego Funduszu Zdrowia w Gdansk	500
3	(null)	(null)	Slaski Oddzial Narodowego Funduszu Zdrowia w Katowicach	1000
4	(null)	(null)	Malopolski Oddzial Narodowego Funduszu Zdrowia w Krakowie	3000
5	(null)	(null)	Opolski Oddzial Narodowego Funduszu Zdrowia w Opolu	2000
6	(null)	(null)	Swietokrzyski Oddzial Narodowego Funduszu Zdrowia w Kielcach	4000
7	(null)	(null)	Malopolski Oddzial Narodowego Funduszu Zdrowia w Krakowie	500
8	(null)	(null)	Warminsko-Mazurski Oddzial Narodowego Funduszu Zdrowia w Olsztynie	3500
9	(null)	(null)	Lubelski Oddzial Narodowego Funduszu Zdrowia w Lublinie	5000
10	(null)	(null)	Lodzki Oddzial Narodowego Funduszu Zdrowia w Lodzi	3000
11	(null)	(null)	Wielkopolski Oddzial Narodowego Funduszu Zdrowia w Poznaniu	4500
12	(null)	(null)	Malopolski Oddzial Narodowego Funduszu Zdrowia w Krakowie	1500
13	(null)	(null)	Dolnoslaski Oddzial Narodowego Funduszu Zdrowia we Wroclawiu	4500
14	(null)	(null)	Podkarpacki Oddzial Narodowego Funduszu Zdrowia w Rzeszowie	500
15	(null)	(null)	Swietokrzyski Oddzial Narodowego Funduszu Zdrowia w Kielcach	2500
16	(null)	(null)	Podlaski Oddzial Narodowego Funduszu Zdrowia w Bialymstoku	3500

- Hurtownia

```

-- Srednia oplata zabiegow danego pacjenta z okreslonego oddzialu, w kazdym gabinecie
SELECT DISTINCT
TO_CHAR ((SELECT nazwisko FROM h_pacjenci WHERE pacjent_id = p_id)) nazwisko_pacjenta,
TO_CHAR ((SELECT oznaczenie FROM h_gabinety WHERE gabinet_id = g_id)) oznaczenie_gabinetu,
TO_CHAR ((SELECT nazwa_oddzialu FROM h_recepty WHERE recepta_id = r_id)) nazwa_oddzialu,
srednia_oplata
FROM (
  SELECT pacjent_id p_id, gabinet_id g_id, recepta_id r_id, AVG (cena_netto_za_zabieg) srednia_oplata FROM h_wizyty
  GROUP BY CUBE (pacjent_id, gabinet_id, recepta_id)
);

```

Query Result x

SQL | All Rows Fetched: 23622 in 1,682 seconds

	NAZWISKO_PACJENTA	OZNACZENIE_GABINETU	NAZWA_ODDZIALU	SREDNIA_OPLATA
1	(null)	(null)	Mazowiecki Oddzial Narodowego Funduszu Zdrowia w Warszawie	(null)
2	(null)	(null)	Pomorski Oddzial Narodowego Funduszu Zdrowia w Gdansk	500
3	(null)	(null)	Slaski Oddzial Narodowego Funduszu Zdrowia w Katowicach	1000
4	(null)	(null)	Malopolski Oddzial Narodowego Funduszu Zdrowia w Krakowie	3000
5	(null)	(null)	Opolski Oddzial Narodowego Funduszu Zdrowia w Opolu	2000
6	(null)	(null)	Swietokrzyski Oddzial Narodowego Funduszu Zdrowia w Kielcach	4000
7	(null)	(null)	Malopolski Oddzial Narodowego Funduszu Zdrowia w Krakowie	500
8	(null)	(null)	Warminsko-Mazurski Oddzial Narodowego Funduszu Zdrowia w Olsztynie	3500
9	(null)	(null)	Lubelski Oddzial Narodowego Funduszu Zdrowia w Lublinie	5000
10	(null)	(null)	Lodzki Oddzial Narodowego Funduszu Zdrowia w Lodzi	3000
11	(null)	(null)	Wielkopolski Oddzial Narodowego Funduszu Zdrowia w Poznaniu	4500
12	(null)	(null)	Malopolski Oddzial Narodowego Funduszu Zdrowia w Krakowie	1500
13	(null)	(null)	Dolnoslaski Oddzial Narodowego Funduszu Zdrowia we Wroclawiu	4500
14	(null)	(null)	Podkarpacki Oddzial Narodowego Funduszu Zdrowia w Rzeszowie	500
15	(null)	(null)	Swietokrzyski Oddzial Narodowego Funduszu Zdrowia w Kielcach	2500
16	(null)	(null)	Podlaski Oddzial Narodowego Funduszu Zdrowia w Bialymstoku	3500
17	(null)	(null)	Swietokrzyski Oddzial Narodowego Funduszu Zdrowia w Kielcach	1500
18	(null)	(null)	Kujawsko-Pomorski Oddzial Narodowego Funduszu Zdrowia w Bydgoszczy	4500

Porównując te dwa zapytania widzimy ,że wersja dla hurtowni jest prostsza i posiada o 2 złączenia tabel mniej i licza średnią opłat prosto z tabeli faktów. Dodatkowo wykonuje się o 0,266s szybciej niż zapytanie do bazy danych

- Cube nr3.
 - Baza

```
-- Laczna odpłatność leków na każdej receptce z dana ulga
SELECT DISTINCT
  r_id numer_recepty,
  TO_CHAR ((SELECT typ_ulgi FROM ulgi WHERE ulgi_id = u_id)) typ_ulgi,
  laczna_odplatnosc
FROM (
  SELECT r.recepta_id r_id, r.ulga_id u_id, SUM (pr.odplatnosc) laczna_odplatnosc FROM pozycje_recept pr
  JOIN recepty r ON r.recepta_id = pr.recepta_id
  GROUP BY CUBE (r.recepta_id, r.ulga_id)
);
```

Query Result x

SQL | All Rows Fetched: 74747 in 1,463 seconds

	NUMER_RECEPTY	TYP_ULGI	LACZNA_ODPLATNOSC
1	(null)	Niepełnosprawnościowa	40254,44
2	(null)	Wiekowa50+	67115,61
3	(null)	Niepełnosprawnościowa	42348,61
4	(null)	Wiekowa50+	67246,75
5	4 (null)		628,91
6	9 (null)		439,99
7	14	Niepełnosprawnościowa	46,47
8	17 (null)		289,38
9	20 (null)		252,99
10	27 (null)		943,38
11	33 (null)		59,52
12	44 (null)		788,98
13	46 (null)		284,35
14	51 (null)		557,99
15	54 (null)		74,4
16	65 (null)		51,98
17	73 (null)		15,49
18	76 (null)		297,71

- Hurtownia

```
-- Laczna odpłatność leków na każdej receptce z dana ulga
SELECT DISTINCT
  r_id numer_recepty,
  TO_CHAR ((SELECT typ_ulgi FROM h_ulgi WHERE ulgi_id = u_id)) typ_ulgi,
  laczna_odplatnosc
FROM (
  SELECT recepta_id r_id, ulga_id u_id, SUM (odplatnosc) laczna_odplatnosc FROM h_pozycje_recept
  GROUP BY CUBE (recepta_id, ulga_id)
);
```

Query Result x

SQL | All Rows Fetched: 74747 in 1,324 seconds

	NUMER_RECEPTY	TYP_ULGI	LACZNA_ODPLATNOSC
1	(null)	Niepełnosprawnościowa	40254,44
2	(null)	Wiekowa50+	67115,61
3	(null)	Niepełnosprawnościowa	42348,61
4	(null)	Wiekowa50+	67246,75
5	4 (null)		628,91
6	9 (null)		439,99
7	14	Niepełnosprawnościowa	46,47
8	17 (null)		289,38
9	20 (null)		252,99
10	27 (null)		943,38
11	33 (null)		59,52
12	44 (null)		788,98
13	46 (null)		284,35
14	51 (null)		557,99
15	54 (null)		74,4
16	65 (null)		51,98
17	73 (null)		15,49
18	76 (null)		297,71

Różnicą między zapytaniem z bazy, a zapytaniem z hurtowni jest to, że w bazie wymaga ono o jedno więcej połączenie JOIN. W hurtowni to zapytanie wykonuje się o 0,139s szybciej.

- 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;
```

Query Result x

SQL | All Rows Fetched: 1748 in 0,169 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 Nazwa,
d.rok Rok,
p.grupa_krwi,
SUM(CNZ)OVER(PARTITION BY d.rok,p.grupa_krwi,z.nazwa) CNZ
FROM
(SELECT wizyta_id WID,pacjent_id PID,zabieg_id ZID,SUM(cena_netto_za_zabieg) CNZ,data_wizyty_id DID
FROM h_wizyty
WHERE zabieg_id IS NOT NULL
AND pacjent_id IN (SELECT pacjent_id
FROM h_pacjenci
WHERE h_pacjenci.grupa_krwi = 'A-'
)
GROUP BY wizyta_id,pacjent_id,zabieg_id,data_wizyty_id
),
h_zabiegi z,
h_daty_wizyt d,
h_pacjenci p
WHERE z.zabieg_id = ZID
AND d.data_id = DID
AND p.pacjent_id = PID
ORDER BY Nazwa,Rok DESC;
```

Query Result x

SQL | All Rows Fetched: 1748 in 0,058 seconds

	NAZWA	ROK	GRUPA_KRWI	CNZ
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

Różnicą między zapytaniem z bazy a hurtowni jest to, że wersja z hurtowni jest bardziej złożona i wymaga o 1 więcej złączenie tabel ,lecz końcowo zapytanie w hurtowni wykonuje się o 0.111s szybciej niż w zwykłej bazie.

- Partycje Obliczeniowe nr2.
 - Baza

Worksheet Query Builder

```

1 --Id pacjenta,jego imie ,nazwisko , pesel oraz jego Wydatki na leki wciagu jednego roku z ulga/bez ulgi
2 SELECT DISTINCT
3   p.pacjent_id,p.imie,p.nazwisko,p.pesel_id,
4   EXTRACT(YEAR FROM w.data_wizyty) AS Rok,
5   SUM(pr.odplatnosc) OVER (PARTITION BY p.pacjent_id,EXTRACT(YEAR FROM w.data_wizyty)) AS "Wydatki Pacjenta Na Leki Bez Ulgi",
6   (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"
7 FROM recepty r
8 JOIN pozycje_recept pr ON pr.recepta_id = r.recepta_id
9 JOIN wizyty w ON w.wizyta_id = r.wizyta_id
10 JOIN pacjenci p ON p.pacjent_id = w.pacjent_id
11 LEFT JOIN ulgi u ON u.ulgi_id = r.ulgi_id
12 ORDER BY p.pacjent_id ASC,Rok DESC;

```

Query Result x All Rows Fetched: 61288 in 1,665 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)

- Hurtownia

Worksheet Query Builder

```

--Id pacjenta,jego imie ,nazwisko , pesel oraz jego Wydatki na leki wciagu jednego roku z ulga/bez ulgi
SELECT DISTINCT PID AS Pacjent_ID,
imie,
nazwisko,
pesel AS Pesel_id,
rok AS Rok,
SUM(ODP) OVER (PARTITION BY PID,rok) AS "Wydatki Pacjenta Na Leki Bez Ulgi",
(SUM(ODP) OVER (PARTITION BY PID,rok) * (PUL / 100) ) AS "Wydatki Pacjenta Na Leki Z Wliczona Ulga"
FROM
  (SELECT p.pacjent_id PID,p.imie imie,p.nazwisko nazwisko,p.pesel pesel,d.rok rok,SUM(pr.odplatnosc) ODP,pr.procent_ulgi PUL
  FROM
    (SELECT pacjent_id PID,data_wizyty_id DID ,recepta_id RID
    FROM h_wizyty w
    WHERE recepta_id IS NOT NULL
    ),
    h_pacjenci p,
    h_daty_wizyt d,
    h_pozycje_recept pr
    WHERE p.pacjent_id = PID
    AND d.data_id = DID
    AND pr.recepta_id = RID
    GROUP BY p.pacjent_id,p.imie,p.nazwisko,p.pesel,d.rok,pr.procent_ulgi
  )
ORDER BY PID ASC,Rok DESC;

```

Query Result x All Rows Fetched: 61288 in 1,259 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)

Zapytanie w hurtowni pod względem trudności zapytania jest bardziej złożone niż wersja dla hurtowni, ale pomaga uniknąć niepotrzebnych złączeń przez co wykonuje się o wiele szybciej od wersji z bazy danych o całe 406ms/0.406s .

- Partycje Obliczeniowe nr3.
 - Baza

Worksheet Query Builder

```

1 -- Procentowy udział opłat za wizyty, w danym roku, w określonej placowce, znajdującej się w określonym mieście, na przestrzeni wszystkich lat
2 SELECT DISTINCT p.placowka_id, p.nazwa AS nazwa_placowki, a.miesto, EXTRACT (YEAR FROM w.data_wizyty) AS rok,
3 SUM (w.oplata) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty), p.placowka_id, a.miesto) suma_w_danym_roku,
4 SUM (w.oplata) OVER (PARTITION BY p.placowka_id, a.miesto) suma_na_przestrzeni_lat,
5 ROUND (100 * (SUM (w.oplata) OVER (PARTITION BY EXTRACT (YEAR FROM w.data_wizyty), p.placowka_id, a.miesto)) / SUM (w.oplata) OVER (PARTITION BY p.placowka_id, a.miesto))
6 "UDZIAL % danego roku" FROM wizyty w
7 JOIN gabinety g ON g.gabinet_id = w.gabinet_id
8 JOIN placowki p ON p.placowka_id = g.placowka_id
9 JOIN adresy a ON a.adres_id = p.adres_id
10 ORDER BY p.placowka_id ASC, rok DESC;

```

Query Result x

All Rows Fetched: 70570 in 1,698 seconds

	PLACOWKA_ID	NAZWA_PLACOWKI	MIASTO	ROK	SUMA_W_DANYM_ROKU	SUMA_NA_PRZESTRZENI_LAT	UDZIAL % danego roku
1	1	Centrum Medyczno-Diagnostyczne Białystok	2020		2100	9400	22
2	1	Centrum Medyczno-Diagnostyczne Białystok	2017		1300	9400	14
3	1	Centrum Medyczno-Diagnostyczne Białystok	2015		300	9400	3
4	1	Centrum Medyczno-Diagnostyczne Białystok	2013		1200	9400	13
5	1	Centrum Medyczno-Diagnostyczne Białystok	2009		1000	9400	11
6	1	Centrum Medyczno-Diagnostyczne Białystok	2007		1200	9400	13

- Hurtownia

Worksheet Query Builder

```

-- Procentowy udział opłat za wizyty, w danym roku, w określonej placowce, znajdującej się w określonym mieście, na przestrzeni wszystkich lat
SELECT DISTINCT
PID AS Placowka_id,
nazwa AS nazwa_placowki,
miasto AS miasto,
rok as rok,
SUM (OPL) OVER (PARTITION BY rok, PID, miasto) suma_w_danym_roku,
SUM (OPL) OVER (PARTITION BY PID, miasto) suma_na_przestrzeni_lat,
ROUND (100 * (SUM (OPL) OVER (PARTITION BY rok, PID, miasto)) / SUM (OPL) OVER (PARTITION BY PID, miasto))"UDZIAL % danego roku"
FROM
(SELECT placowka_id PID, p.nazwa nazwa,p.miesto miasto,d.rok rok, SUM(OPL) OPL
FROM
  (SELECT gabinet_id GID,SUM(oplata) OPL,data_wizyty_id DID
  FROM h_wizyty
  GROUP BY gabinet_id,data_wizyty_id
  ),
  h_placowki p,
  h_daty_wizyt d
WHERE p.placowka_id IN (SELECT placowka_id
  FROM h_gabinety
  WHERE gabinet_id = GID)
AND d.data_id = DID
GROUP BY placowka_id,p.nazwa,p.miesto,d.rok
)
ORDER BY PID ASC, rok DESC;

```

Query Result x

All Rows Fetched: 70570 in 1,528 seconds

	PLACOWKA_ID	NAZWA_PLACOWKI	MIASTO	ROK	SUMA_W_DANYM_ROKU	SUMA_NA_PRZESTRZENI_LAT	UDZIAL % danego roku
1	1	Centrum Medyczno-Diagnostyczne Białystok	2020		2100	9400	22
2	1	Centrum Medyczno-Diagnostyczne Białystok	2017		1300	9400	14
3	1	Centrum Medyczno-Diagnostyczne Białystok	2015		300	9400	3
4	1	Centrum Medyczno-Diagnostyczne Białystok	2013		1200	9400	13
5	1	Centrum Medyczno-Diagnostyczne Białystok	2009		1000	9400	11
6	1	Centrum Medyczno-Diagnostyczne Białystok	2007		1200	9400	13

Pod względem złożoności oba zapytania są podobne, wersja dla bazy wymaga 3 złączeń, natomiast wersja dla hurtowni aż 3 podzapytań, jednakże wydajność przemawia na stronę wersji dla hurtowni o jakieś 170ms/0.170s .

- Okna Czasowe nr1.
 - Baza

Worksheet Query Builder

```
--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;
```

Query Result x

All Rows Fetched: 10137 in 0,49 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

- Hurtowni

Worksheet Query Builder

```
--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(WID) 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(ZID) 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(RID) 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
(SELECT wizyta_id WID,zabieg_id ZID,recepta_id RID,data_wizyty_id DID
FROM h_wizyty
GROUP BY wizyta_id,zabieg_id,recepta_id,data_wizyty_id
),
h_daty_wizyt d
WHERE d.data_id = DID
ORDER BY Rok DESC , Miesiac ASC;
```

Query Result x

All Rows Fetched: 10137 in 0,464 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

Podstawową różnicą pomiędzy zapytania jest to ,że wersja dla bazy wymaga dwóch złączeń JOIN ,natomiast wersja dla hurtowni wymaga tylko jednego podzapytania co przekłada się na szybsze wykonanie o 26ms/0.026s .

- Okna Czasowe nr2.
 - Baza

Worksheet Query Builder

```
--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
1 SELECT DISTINCT
2   c.nazwa AS Nazwa_Choroby,c.choroby_id,EXTRACT(YEAR FROM c.początek) Rok,w.wizyta_id,
3   COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.początek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
4   AS "Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu wizyty",
5   ROUND(100 * COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.początek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
6   / COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.początek) ) , 3) "Udział % w tym roku",
7   COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.początek) ) "Łączna wartość zapadnięć na tę chorobę w tym roku",
8   ROUND(100 * COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.początek) ) / COUNT(*) OVER (PARTITION BY c.nazwa) , 3) "Udział % na tie lat",
9   COUNT(*) OVER (PARTITION BY c.nazwa) "Łączna wartość zapadnięć na tę chorobę na przestrzeni lat",
10  k.grupa_krwi "Grupa Krwi Pacjenta",p.imie,p.nazwisko
11 FROM choroby c
12 JOIN recepty r ON r.recepta_choroba_id = c.choroby_id
13 JOIN wizyty w ON w.wizyta_id = r.wizyta_id
14 JOIN pacjenci p ON p.pacjent_id = w.pacjent_id
15 JOIN karty k ON k.pesel_id = p.pesel_id
16 ORDER BY Rok DESC,COUNT(*) OVER (PARTITION BY c.nazwa,EXTRACT(YEAR FROM c.początek) ORDER BY c.choroby_id ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) DESC;
```

Query Result x All Rows Fetched: 69805 in 2,349 seconds

	NAZWA_CHOROBY	CHOROBY_ID	ROK	WIZYTA_ID	Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu wizyty	Udział % w tym roku	Łączna wartość zapadnięć na tę chorobę w tym roku	Udział % na tie lat	Łączna wartość zapadnięć na tę chorobę na przestrzeni lat	Grupa...	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

Worksheet Query Builder

```
--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
1 SELECT DISTINCT
2   Nazwa AS Nazwa_Choroby,
3   CID choroby_id,
4   EXTRACT(YEAR FROM CP) Rok,
5   WID AS wizyta_id,
6   COUNT(*) OVER (PARTITION BY Nazwa,EXTRACT(YEAR FROM CP) ORDER BY CID ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS "Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu wizyty",
7   ROUND(100 * COUNT(*) OVER (PARTITION BY Nazwa,EXTRACT(YEAR FROM CP) ORDER BY CID ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / COUNT(*) OVER (PARTITION BY Nazwa,EXTRACT(YEAR FROM CP) ) , 3) "Udział
8   COUNT(*) OVER (PARTITION BY Nazwa,EXTRACT(YEAR FROM CP) ) "Łączna wartość zapadnięć na tę chorobę w tym roku",
9   ROUND(100 * COUNT(*) OVER (PARTITION BY Nazwa,EXTRACT(YEAR FROM CP) ) / COUNT(*) OVER (PARTITION BY Nazwa) , 3) "Udział % na tie lat",
10  COUNT(*) OVER (PARTITION BY Nazwa) "Łączna wartość zapadnięć na tę chorobę na przestrzeni lat",
11  PGK AS "Grupa Krwi Pacjenta",
12  PIMIE AS Imie,
13  PNAZ AS Nazwisko
14 FROM
15   (SELECT WID WID,CID CID,Nazwa Nazwa,CP CP,p.grupa_krwi PGK,p.imie PIMIE,p.nazwisko PNAZ
16   FROM
17     (SELECT wizyta_id WID,pacjent_id PID,CID CID,Nazwa Nazwa,CP CP
18     FROM
19       (SELECT CID CID,Nazwa Nazwa,CP CP,recepta_id RID
20       FROM
21         (SELECT choroby_id CID,nazwa Nazwa,początek CP
22         FROM h_choroby
23         ),
24         h_recepty
25         WHERE h_recepty.choroba_id = CID
26         ),
27         h_wizyty
28         WHERE h_wizyty.recepta_id = RID
29         ),
30         h_pacjenci p
31         WHERE p.pacjent_id = PID
32         )
33     ORDER BY Rok DESC,COUNT(*) OVER (PARTITION BY Nazwa,EXTRACT(YEAR FROM CP) ORDER BY CID ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) DESC;
```

Query Result x All Rows Fetched: 69805 in 1,864 seconds

	NAZWA_CHOROBY	CHOROBY_ID	ROK	WIZYTA_ID	Ilość zapadnięć na tę chorobę w tym roku do aktualnego rekordu wizyty	Udział % w tym roku	Łączna wartość zapadnięć na tę chorobę w tym roku	Udział % na tie lat	Łączna wartość zapadnięć na tę chorobę na przestrzeni lat	Grupa...	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 B-	Emil	Malinowski	
3	Grypa	9984	2020	59949	293	99,322	295	6,071	4859 0-	Dagmara	Urbanska	
4	Grypa	9984	2020	56436	292	98,983	295	6,071	4859 AB-	Milena	Kucharska	
5	Grypa	9984	2020	56433	291	98,644	295	6,071	4859 AB-	Milena	Kucharska	
6	Grypa	9984	2020	56432	290	98,305	295	6,071	4859 AB-	Agata	Zakrzewska	

W tym przypadku widzimy pokąźną przewagę wydajnościową pomiędzy zapytaniami, wersja dla bazy danych wymaga dodatkowych 4 złączeń tabel a hurtownia aż 4 podzapytań, i tym sposobem wersja dla hurtowni wykonuje się o 485ms/0.485s szybciej.

- Okna Czasowe nr3.
 - Baza

Worksheet: Query Builder

```

1 --Oplaty, ceny leków i zabiegów każdej wizyty oraz sumaryczna wartość dotychczasowych wizyt ,ktore zakonczyly sie recepta lub recepta i zabiegiem
2 SELECT DISTINCT
3 w.wizyta_id,p.pesel_id,p.imie,p.nazwisko,w.oplata "Oplata za wizytę pacjenta",
4 (SUM(pr.odplatnosc) OVER (PARTITION BY w.wizyta_id)) "Cena leków",
5 (u.procent_ulgi / 100 * (SUM(pr.odplatnosc) OVER (PARTITION BY w.wizyta_id))) "Cena leków Po Uldze",
6 SUM(w.oplata) OVER (ORDER BY w.wizyta_id RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych opłat za wizyty pacjentów",
7 z.cena_netto "Oplata za zabieg pacjenta",
8 SUM(z.cena_netto) OVER (ORDER BY w.wizyta_id RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych opłat za zabiegi"
9 FROM wizyty w
10 JOIN pacjenci p ON p.pacjent_id = w.pacjent_id
11 LEFT JOIN zabiegi z ON w.wizyta_id = z.wizyta_id
12 JOIN recepty r ON r.wizyta_id = w.wizyta_id
13 LEFT JOIN posycje_recept pr ON pr.recepta_id = r.recepta_id
14 LEFT JOIN ulgi u ON u.ulgi_id = r.ulgi_id
15 ORDER BY w.wizyta_id ASC;

```

Query Result x

All Rows Fetched: 69805 in 2,058 seconds

WIZYTA_ID	PESEL_ID	IMIE	NAZWISKO	Oplata za wizytę pacjenta	Cena leków	Cena leków Po Uldze	Suma wszystkich dotychczasowych opłat za wizyty pacjentów	Oplata za zabieg pacjenta	Suma wszystkich dotychczasowych opłat za zabiegi
1	284134527845	Oktawian	Szewczyk	1500	616,93	(null)	4500	(null)	(null)
2	432763847807	Maja	Sikorska	1500	365,25	182,625	9000	(null)	(null)
3	521444450160	Jola	Kubiak	1000	1029,4	(null)	13000	(null)	(null)
4	640644227778	Alan	Gorski	300	628,91	(null)	14500	(null)	(null)
5	757345800338	Milena	Szczepanska	300	830,88	(null)	15700	(null)	(null)
6	1374663330250	Julian	Sikora	800	395,2	197,6	18900	(null)	(null)

- Hurtownia

Worksheet: Query Builder

```

1 --Oplaty, ceny leków i zabiegów każdej wizyty oraz sumaryczna wartość dotychczasowych wizyt ,ktore zakonczyly sie recepta lub recepta i zabiegiem
2 SELECT DISTINCT
3 WID wizyta_id,
4 p.pesel,
5 p.imie,
6 p.nazwisko,
7 OPL "Oplata za wizytę pacjenta",
8 SUM(pr.odplatnosc) OVER (PARTITION BY WID) "Cena leków",
9 (pr.procent_ulgi / 100 * (SUM(pr.odplatnosc) OVER (PARTITION BY WID))) "Cena leków Po Uldze",
10 SUM(OPL) OVER (ORDER BY WID RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych opłat za wizyty pacjentów",
11 CNZ "Oplata za zabieg pacjenta",
12 SUM(CNZ) OVER (ORDER BY WID RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) "Suma wszystkich dotychczasowych opłat za zabiegi"
13 FROM
14 (SELECT wizyta_id WID, oplata OPL, cena_netto_za_zabieg CNZ, pacjent_id PID, recepta_id RID
15 FROM h_wizyty
16 WHERE recepta_id IS NOT NULL
17 GROUP BY wizyta_id, pacjent_id, recepta_id, oplata, cena_netto_za_zabieg
18 ),
19 h_posycje_recept pr,
20 h_pacjenci p
21 WHERE pr.recepta_id = RID AND p.pacjent_id = PID
22 ORDER BY WID ASC;

```

Query Result x

All Rows Fetched: 69805 in 1,893 seconds

WIZYTA_ID	PESEL	IMIE	NAZWISKO	Oplata za wizytę pacjenta	Cena leków	Cena leków Po Uldze	Suma wszystkich dotychczasowych opłat za wizyty pacjentów	Oplata za zabieg pacjenta	Suma wszystkich dotychczasowych opłat za zabiegi
1	284134527845	Oktawian	Szewczyk	1500	616,93	(null)	4500	(null)	(null)
2	432763847807	Maja	Sikorska	1500	365,25	182,625	9000	(null)	(null)
3	521444450160	Jola	Kubiak	1000	1029,4	(null)	13000	(null)	(null)
4	640644227778	Alan	Gorski	300	628,91	(null)	14500	(null)	(null)
5	757345800338	Milena	Szczepanska	300	830,88	(null)	15700	(null)	(null)
6	1374663330250	Julian	Sikora	800	395,2	197,6	18900	(null)	(null)

Tak samo w tym przypadku jak i poprzednim wersja hurtowni zapytania deklasuje wersję dla bazy o 165ms/0.165 s , dzieje się tak ,ponieważ baza wymaga aż 5 złączeń JOIN a w hurtowni jest to realizowane poprzez jedno podzapytanie.

- Funkcje Rankingowe nr1.

- Baza

```
-- Ranking pacjentow, ktorzy wniesli najwiecej oplat za wizyte w danym gabinecie
```

```
SELECT DISTINCT
TO_CHAR ((SELECT nazwisko FROM pacjenci WHERE pacjent_id = p_id)) nazwisko_pacjenta,
g_id identyfikator_gabinetu,
opлата_za_wizyte,
ranking
FROM (
SELECT pacjent_id p_id, gabinet_id g_id, oplata oplata_za_wizyte,
RANK () OVER (PARTITION BY gabinet_id ORDER BY oplata DESC) ranking
FROM wizyty
)
ORDER BY ranking;
```

Query Result x

SQL | All Rows Fetched: 99371 in 1,898 seconds

	NAZWISKO_PACJENTA	IDENTYFIKATOR_GABINETU	OPLATA_ZA_WIZYTE	RANKING
1	Ziolkowska	1	1500	1
2	Duda	2	1500	1
3	Kozlowski	3	1200	1
4	Zawadzka	3	1200	1
5	Marciniak	4	1500	1
6	Pietrzak	4	1500	1
7	Gorkiewicz	5	1500	1
8	Gorska	6	1500	1
9	Przybylski	7	1300	1
10	Wisniewski	8	1400	1
11	Mazur	9	1200	1
12	Brzezinska	10	1500	1
13	Sikora	11	1200	1
14	Kwiatkowska	12	1400	1
15	Wisniewski	13	1200	1
16	Brzezinski	14	1200	1
17	Gorkiewicz	14	1200	1

- Hurtownia

```
-- Ranking pacjentow, ktorzy wniesli najwiecej oplat za wizyte w danym gabinecie
```

```
SELECT DISTINCT
TO_CHAR ((SELECT nazwisko FROM h_pacjenci WHERE pacjent_id = p_id)) nazwisko_pacjenta,
g_id identyfikator_gabinetu,
opлата_za_wizyte,
ranking
FROM (
SELECT pacjent_id p_id, gabinet_id g_id, oplata oplata_za_wizyte,
RANK () OVER (PARTITION BY gabinet_id ORDER BY oplata DESC) ranking
FROM h_wizyty
)
ORDER BY ranking;
```

Query Result x

SQL | All Rows Fetched: 99371 in 1,829 seconds

	IDENTYFIKATOR_GABINETU	NAZWISKO_PACJENTA	OPLATA_ZA_WIZYTE	RANKING
1	1	Ziolkowska	1500	1
2	2	Duda	1500	1
3	3	Kozlowski	1200	1
4	3	Zawadzka	1200	1
5	4	Marciniak	1500	1
6	4	Pietrzak	1500	1
7	5	Gorkiewicz	1500	1
8	6	Gorska	1500	1
9	7	Przybylski	1300	1
10	8	Wisniewski	1400	1
11	9	Mazur	1200	1
12	10	Brzezinska	1500	1
13	11	Sikora	1200	1
14	12	Kwiatkowska	1400	1
15	13	Wisniewski	1200	1
16	14	Brzezinski	1200	1

Struktura zapytania dla hurtowni jak i dla bazy jest taka sama, mimo to zapytanie do hurtowni danych wykonało się o 0,069s szybciej

- Funkcje Rankingowe nr2.

- Baza

```
-- Ranking najlepszej sumy sprzedanych lekow na recepte, z dana ulga
```

```
SELECT DISTINCT
  r_id numer_recepty,
  TO_CHAR ((SELECT typ_ulgi FROM ulgi WHERE ulgi_id = u_id)) typ_ulgi,
  odplatnosc_suma,
  ranking
FROM (
  SELECT r_id, u_id, odplatnosc_suma,
  RANK () OVER (ORDER BY odplatnosc_suma DESC) ranking
  FROM (
    SELECT r.recepta_id r_id, r.ulga_id u_id, SUM (odplatnosc) odplatnosc_suma FROM pozycje_recept pr
    JOIN recepty r ON pr.recepta_id = r.recepta_id
    GROUP BY r.recepta_id, r.ulga_id
  )
)
ORDER BY ranking;
```

Query Result x

SQL | All Rows Fetched: 69805 in 1,235 seconds

	NUMER_RECEPTY	TYP_ULGI	ODPLATNOSC_SUMA	RANKING
1	14873 (null)		2375,03	1
2	48200 (null)		2314,78	2
3	29127 (null)		2223,22	3
4	19329 Niepelnosprawnosciowa		2150,87	4
5	31959 (null)		2100,61	5
6	8316 (null)		2100,17	6
7	5854 (null)		2087,23	7
8	4110 Niepelnosprawnosciowa		2082,45	8
9	69782 (null)		2079,39	9
10	64290 (null)		2079,24	10
11	55289 (null)		2070,09	11
12	3900 (null)		2064,87	12
13	2921 (null)		2055,87	13
14	16389 (null)		2052,12	14

- Hurtownia

```
-- Ranking najlepszej sumy sprzedanych lekow na recepte, z dana ulga
```

```
SELECT DISTINCT
  r_id numer_recepty,
  TO_CHAR ((SELECT typ_ulgi FROM h_ulgi WHERE ulgi_id = u_id)) typ_ulgi,
  odplatnosc_suma,
  ranking
FROM (
  SELECT r_id, u_id, odplatnosc_suma,
  RANK () OVER (ORDER BY odplatnosc_suma DESC) ranking
  FROM (
    SELECT recepta_id r_id, ulga_id u_id, SUM (odplatnosc) odplatnosc_suma FROM h_pozycje_recept
    GROUP BY recepta_id, ulga_id
  )
)
ORDER BY ranking;
```

Query Result x

SQL | All Rows Fetched: 69805 in 1,202 seconds

	NUMER_RECEPTY	TYP_ULGI	ODPLATNOSC_SUMA	RANKING
1	14873 (null)		2375,03	1
2	48200 (null)		2314,78	2
3	29127 (null)		2223,22	3
4	19329 Niepelnosprawnosciowa		2150,87	4
5	31959 (null)		2100,61	5
6	8316 (null)		2100,17	6
7	5854 (null)		2087,23	7
8	4110 Niepelnosprawnosciowa		2082,45	8
9	69782 (null)		2079,39	9
10	64290 (null)		2079,24	10
11	55289 (null)		2070,09	11
12	3900 (null)		2064,87	12
13	2921 (null)		2055,87	13
14	16389 (null)		2052,12	14

Różnicą między zapytaniem z bazy, a zapytaniem z hurtowni jest to ,że w bazie wymaga ono o jedno więcej połączenie JOIN. W hurtowni to zapytanie wykonuje się o 0,033s szybciej.

- Funkcje Rankingowe nr3.

- Baza

```
-- Ranking najdrozej wykonanych zabiegow przez pracownika pracujacego w okreslonym gabinecie
```

```
SELECT DISTINCT
TO_CHAR ((SELECT nazwa FROM zabiegi WHERE zabieg_id = z_id)) nazwa_zabiegu,
TO_CHAR ((SELECT nazwisko FROM pracownicy WHERE pracownik_id = p_id)) nazwisko_pracownika,
identyfikator_gabinetu,
cena_netto_za_zabieg,
ranking
FROM (
SELECT z.zabieg_id z_id, w.prac_spec p_id, w.gabinet_id identyfikator_gabinetu, z.cena_netto cena_netto_za_zabieg,
RANK () OVER (PARTITION BY w.gabinet_id ORDER BY z.cena_netto DESC) ranking
FROM zabiegi z
RIGHT JOIN wizyty w ON w.wizyta_id = z.wizyta_id
)
ORDER BY nazwa_zabiegu, ranking;
```

Query Result x

SQL | All Rows Fetched: 31891 in 0,656 seconds

	NAZWA_ZABIEGU	NAZWISKO_PRACOWNIKA	IDENTYFIKATOR_GABINETU	CENA_NETTO_ZA_ZABIEG	RANKING
1	abdominoplastyka	Gorska	1203	5000	1
2	abdominoplastyka	Gorska	266	4500	1
3	abdominoplastyka	Sikora	8864	4500	1
4	abdominoplastyka	Sawicka	4541	5000	1
5	abdominoplastyka	Andrzejewski	4568	5000	1
6	abdominoplastyka	Malinowski	4736	4500	1
7	abdominoplastyka	Wlodarczyk	2914	5000	1
8	abdominoplastyka	Kubiak	3666	5000	1
9	abdominoplastyka	Zalewski	3835	4500	1
10	abdominoplastyka	Urbanska	8271	5000	1
11	abdominoplastyka	Sikorska	6889	5000	1
12	abdominoplastyka	Krawczyk	7587	5000	1
13	abdominoplastyka	Krawczyk	7660	5000	1
14	abdominoplastyka	Chmielewska	6129	5000	1

- Hurtownia

```
-- Ranking najdrozej wykonanych zabiegow przez pracownika pracujacego w okreslonym gabinecie
```

```
SELECT DISTINCT
TO_CHAR ((SELECT nazwa FROM h_zabiegi WHERE zabieg_id = z_id)) nazwa_zabiegu,
TO_CHAR ((SELECT nazwisko FROM h_pracownicy WHERE pracownik_id = p_id)) nazwisko_pracownika,
identyfikator_gabinetu,
cena_netto_za_zabieg,
ranking
FROM (
SELECT zabieg_id z_id, prac_spec p_id, gabinet_id identyfikator_gabinetu, cena_netto_za_zabieg,
RANK () OVER (PARTITION BY gabinet_id ORDER BY cena_netto_za_zabieg DESC) ranking
FROM h_wizyty
)
ORDER BY nazwa_zabiegu, ranking;
```

Query Result x

SQL | All Rows Fetched: 31891 in 0,633 seconds

	NAZWA_ZABIEGU	NAZWISKO_PRACOWNIKA	IDENTYFIKATOR_GABINETU	CENA_NETTO_ZA_ZABIEG	RANKING
1	abdominoplastyka	Gorska	1203	5000	1
2	abdominoplastyka	Gorska	266	4500	1
3	abdominoplastyka	Sikora	8864	4500	1
4	abdominoplastyka	Sawicka	4541	5000	1
5	abdominoplastyka	Andrzejewski	4568	5000	1
6	abdominoplastyka	Malinowski	4736	4500	1
7	abdominoplastyka	Wlodarczyk	2914	5000	1
8	abdominoplastyka	Kubiak	3666	5000	1
9	abdominoplastyka	Zalewski	3835	4500	1
10	abdominoplastyka	Urbanska	8271	5000	1
11	abdominoplastyka	Sikorska	6889	5000	1
12	abdominoplastyka	Krawczyk	7587	5000	1
13	abdominoplastyka	Krawczyk	7660	5000	1
14	abdominoplastyka	Chmielewska	6129	5000	1

Różnicą między zapytaniem z bazy, a zapytaniem z hurtowni jest to, że w bazie wymaga ono o jedno więcej połączenie JOIN. W hurtowni to zapytanie wykonuje się o 0,023s szybciej.

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.
- Wszystkie zapytania do hurtowni danych wykonują się szybciej niż do bazy danych. Największe różnice były widoczne przy zapytaniach z partycjami obliczeniowymi, oknami czasowymi i z operatorem CUBE.
- Struktura zapytań do hurtowni danych nie posiada żadnych złączeń typu JOIN, gdyż nie były one konieczne.
- 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.