|  |  |
| --- | --- |
| Projekt **Bazy Danych 2**  Wydział Elektrotechniki Automatyki i Informatyki  Politechnika Świętokrzyska | |
| Studia: **Stacjonarne I stopnia** | Kierunek: **Informatyka** |
| **Temat:**  **Hurtownia danych dla sieci** **gabinetów dentystycznych** | Grupa: **2ID13A** |
| Skład osobowy: |

**1. Wprowadzenie.**

Tematem naszego projektu była hurtowania danych obsługująca sieć salonów dentystycznych. Każdy z gabinetów umiejscowiony jest w danej przychodni. W każdej przychodni znajduje się kilka gabinetów. Centralną tabelą naszej hurtowni (tabela faktów) jest tabela Zabieg, przechowuje ona informacje o gabinecie w którym został wykonany zabieg, pacjencie, dentyście oraz czasie wykonania zabiegu.

Zadaniem hurtowni jest gromadzenie danych nt. funkcjonowania przychodni. Dane służą kontrolowaniu działania placówek, archiwizacji wykonanych zabiegów oraz danych pacjentów oraz ocenie działalności każdej placówki.

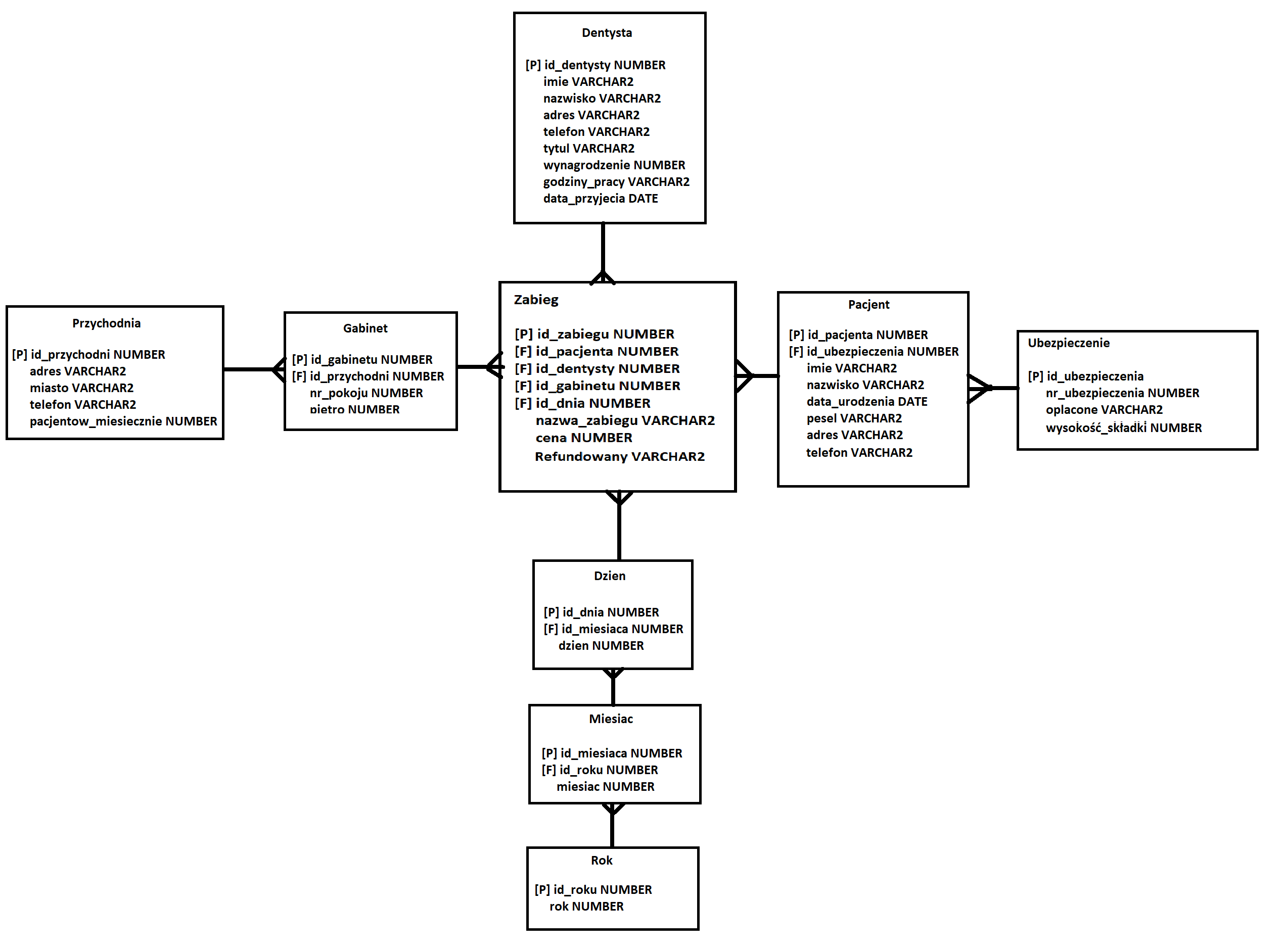
Dodatkowo możliwe jest też sprawdzenie liczby wykonywanych zabiegów wraz z zyskiem jakie przyniosło gabinetowi ich wykonanie. Pomaga to w określeniu rentowności danego gabinetu.

W przypadku gdy dany gabinet odwiedza duża ilość klientów, jest to znak, że można go doposażyć zwiększając tym samym liczbę wykonywanych zabiegów.

Możliwa jest też ocena dentystów na podstawie liczby wykonanych zabiegów i zysków jakie udało mu się wypracować. Hurtownia posiada zapytanie zwracające ranking dentystów pod względem wypracowanych zysków.

Mimo niewielkich rozmiarów nasza hurtownia pozwala łatwiej podejmować decyzje na temat funkcjonowania przychodni oraz personelu. Upraszcza to analizę sytuacji i pozwala łatwiej reagować na straty i zorientować się gdzie można podnieść zyski.

**2. Schemat ERD.**

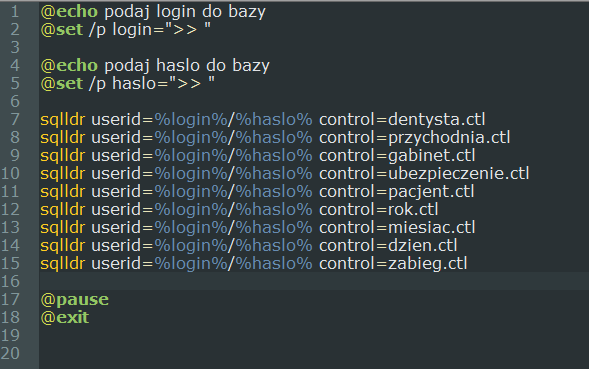


Schemat w pełnej rozdzielczości znajduje się w tym samym folderze co sprawozdanie. Plik schemat bazy.png

**3. Ładowanie danych**

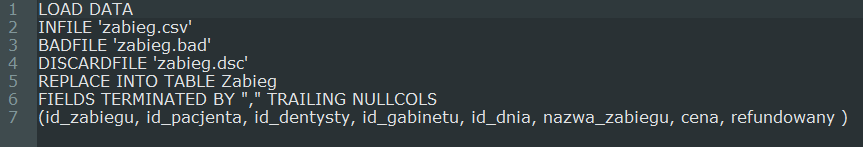
W folderze Pliki danych znajdującym się w głównym katalogu tej płyty znajduje się plik „Skrypt SQLLDR.bat” będący skryptem powłoki systemu Windows. Pomaga szybciej ładować dane. Jego działanie polega na pobraniu loginu i hasła do bazy, następnie za pomocą odpowiednich poleceń wywołuje plik sqlldr.exe ładując każdą z tabel danymi z odpowiadającego jej pliku.

„Skrypt SQLLDR.bat” ładujący dane

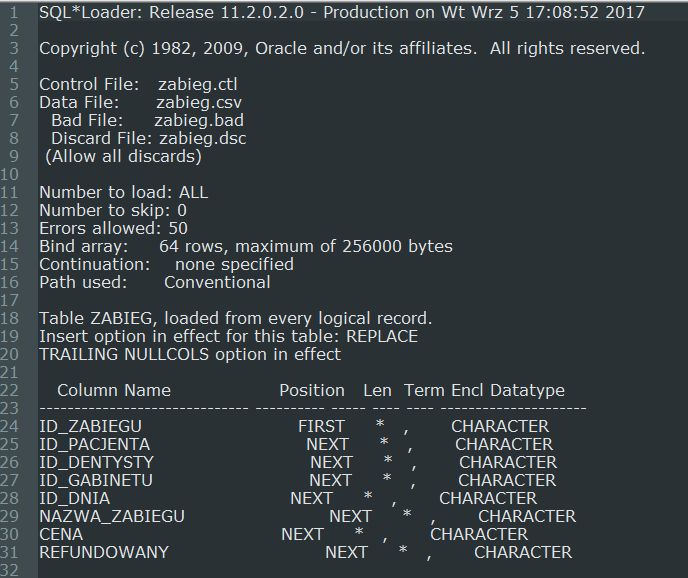


Program do swojego działania potrzebuje plików kontrolnych .ctl w których zapisano informacje na temat tabeli i pliku danych.

Przykładowy plik .ctl



Pliki \*.log opisują jak przebiegł proces ładowania danych do tabeli Przykładowy plik .log:



Na potrzeby projektu stworzyliśmy generator danych w języku C++. Pozwala on szybko generować dane dla każdej z tabel i zapisywać je do formatu .csv skąd mogą być zaimportowane do bazy. Generator znajduje się w katalogu \Pliki danych\generator\.

Przykładowy kod z początku programu.



**4. Zapytania.**

Znajdują się w folderze \Pliki SQL\

**Dane zostały wygenerowane przez program, stąd niektóre z zapytań mają takie same wartości w kolejnych wierszach. Z uwagi na 15 000 rekordów tabeli faktów suma zysków jest bardzo duża.**

* **ROLLUP**

*SELECT Nvl(To\_Char(id\_dnia),'Wszystkie dni') AS id\_dnia,*

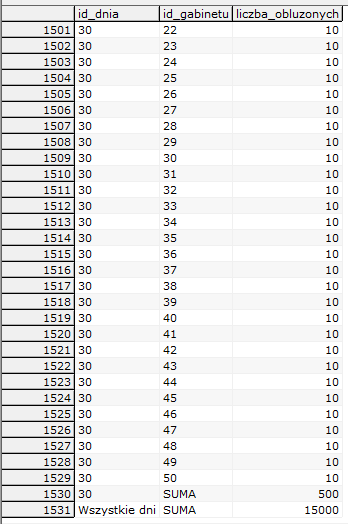
*Nvl(To\_Char(id\_gabinetu),'SUMA') AS id\_gabinetu,*

*Count(id\_pacjenta) AS liczba\_obluzonych*

*FROM zabieg*

*GROUP BY rollup (id\_dnia, id\_gabinetu);*

Pozwala określić liczbę pacjentów przyjętych w danym dniu, w danym gabinecie, a także podsumowanie dla liczby pacjentów w danym dniu dla wszystkich dni, oraz ogół liczby przyjętych pacjentów.



*SELECT Nvl(To\_Char(id\_dnia),'Wszystkie dni') AS id\_dnia,*

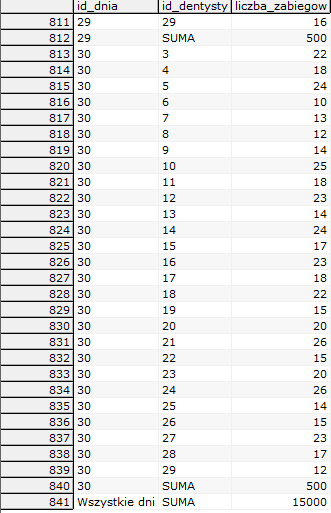
*Nvl(To\_Char(id\_dentysty),'SUMA') AS id\_dentysty,*

*Count(id\_zabiegu) AS liczba\_zabiegow*

*FROM zabieg*

*GROUP BY rollup (id\_dnia, id\_dentysty);*

Pozwala określić liczbę wykonanych przez danego dentystę, w danym dniu zabiegów, a także podsumowanie ilości zabiegów w danym dniu oraz liczę wszystkich wykonanych zabiegów.



* **CUBE**

*SELECT Nvl(To\_Char(id\_gabinetu),'SUMA GABINET') AS id\_gabinetu,*

*Nvl(To\_Char(id\_dentysty),'SUMA DENTYSTA') AS id\_dentysty,*

*Nvl(To\_Char(nazwa\_zabiegu),'WSZYSTKICH ZABIEGOW') AS nazwa\_zabiegu,*

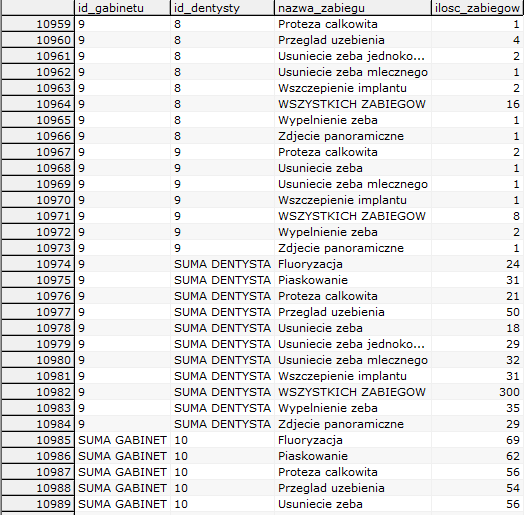
*Count(id\_zabiegu) AS ilosc\_zabiegow*

*FROM zabieg*

*GROUP BY cube(id\_gabinetu,id\_dentysty,nazwa\_zabiegu)*

*ORDER BY id\_gabinetu asc,id\_dentysty asc,nazwa\_zabiegu ASC;*

Zapytanie dostarcza informacji na temat liczby wykonanych przez danego dentystę zabiegów danego typu, w danym gabinecie, a także: liczbę zabiegów danego dentysty, liczbę zabiegów w danym gabinecie, liczbę zabiegów danego typu i liczbę zabiegów ogółem.

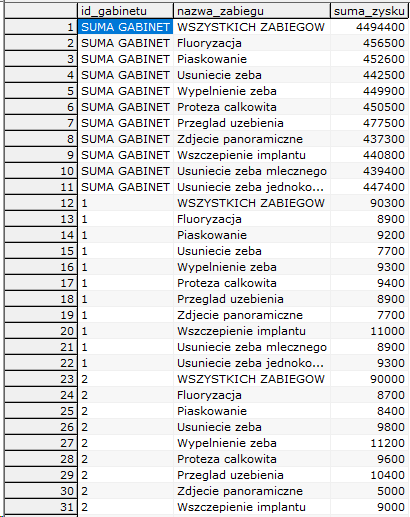


*SELECT Nvl(To\_Char(id\_gabinetu),'SUMA GABINET') AS id\_gabinetu, Nvl(To\_Char(nazwa\_zabiegu),'WSZYSTKICH ZABIEGOW') AS nazwa\_zabiegu, Sum(cena) AS suma\_zysku*

*FROM zabieg*

*GROUP BY cube(id\_gabinetu,nazwa\_zabiegu);*

Określa sumę zysku dla każdego gabinetu na każdym z rodzajów zabiegów, sumę zysku dla danego zabiegu oraz ogólną sumę zysku



* **GRUPING SETS**

*SELECT Nvl(To\_Char(id\_gabinetu),'SUMA GABINET') AS id\_gabinetu,*

*Nvl(To\_Char(id\_dnia),'SUMA DNIA') AS id\_dnia,*

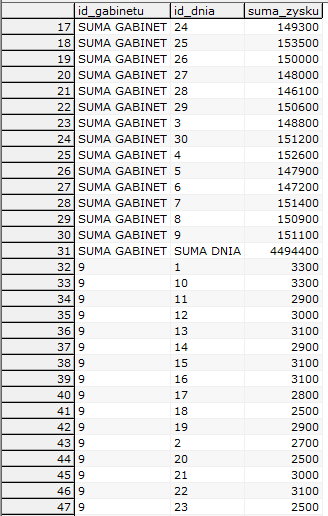
*Sum(cena) AS suma\_zysku*

*FROM zabieg*

*GROUP BY Grouping sets ((id\_gabinetu,id\_dnia), id\_dnia ,())*

*ORDER BY id\_gabinetu desc, id\_dnia;*

Zapytanie pokazuje sumę zysków w danym gabinecie danego dnia, sumę ogólną zysków w danym dniu, oraz zysk całkowity



*SELECT Nvl(To\_Char(id\_dentysty),'SUMA DENTYSTA') AS id\_dentysty,*

*Nvl(To\_Char(id\_dnia),'SUMA DNIA') AS id\_dnia,*

*Nvl(To\_Char(nazwa\_zabiegu),'WSZYSTKICH ZABIEGOW') AS nazwa\_zabiegu,*

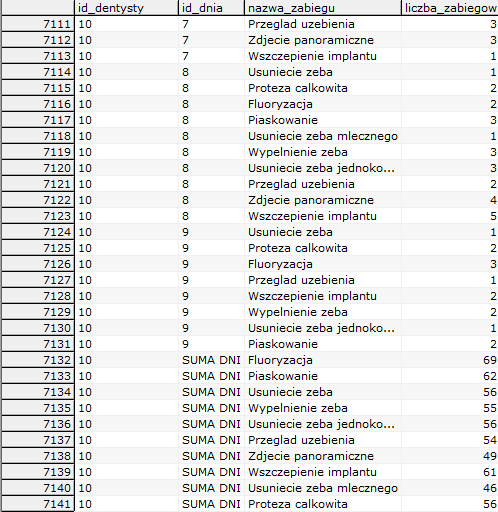
*Count(id\_zabiegu) AS liczba\_zabiegow*

*FROM zabieg*

*GROUP BY Grouping sets ((id\_dentysty,id\_dnia),nazwa\_zabiegu, id\_dnia ,())*

*ORDER BY id\_dentysty desc , id\_dnia;*

Zapytanie pokazuje liczbę zabiegów danego typu które wykonał dany lekarz w danym dniu, sumę zabiegów danego dnia i sumę poszczególnych zabiegów wykonanych przez danego dentystę.



* **GRUPING SETS**

*SELECT distinct id\_gabinetu, id\_dnia,*

*Sum(cena) OVER(*

*PARTITION BY id\_gabinetu*

*ORDER BY id\_dnia*

*RANGE BETWEEN unbounded preceding AND CURRENT ROW*

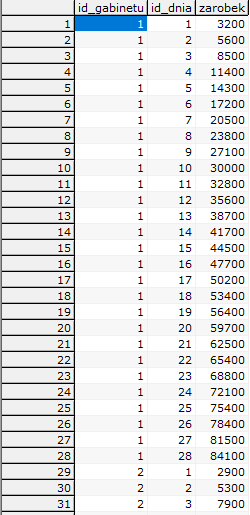
*) AS zarobek*

*FROM zabieg*

*WHERE id\_dnia <29*

*ORDER BY id\_gabinetu;*

Zapytanie z wykorzystaniem partycji, pozwala prześledzić zarobek danego gabinetu w kolejnych dniach.



*SELECT distinct id\_gabinetu, id\_dentysty, id\_pacjenta, id\_dnia,*

*Count(id\_zabiegu) OVER(*

*PARTITION BY id\_gabinetu,id\_dentysty*

*ORDER BY id\_dnia*

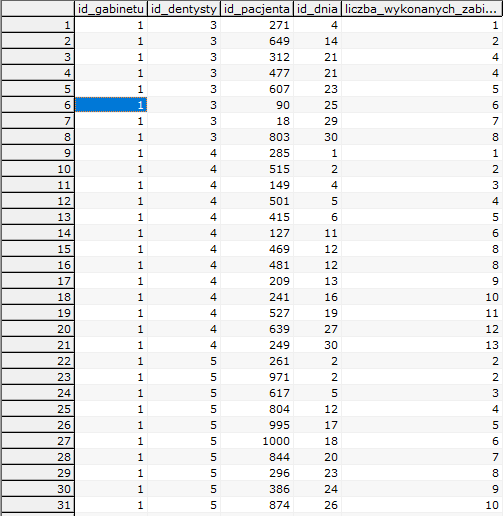
*RANGE BETWEEN unbounded preceding AND CURRENT ROW*

*) AS liczba\_wykonanych\_zabiegow*

*FROM zabieg*

*ORDER BY id\_gabinetu,id\_dentysty;*

Zapytanie partiotion pozwala prześledzić wykonane przez danego dentystę zabiegi z uwzględnieniem: gabinetu pacjenta i dnia. Zapytanie ma charakter informacyjny



* **Funkcje rankingowe**

*SELECT id\_dentysty,*

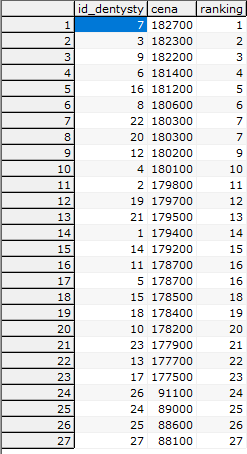
*SUM(cena) AS cena,*

*Rank() OVER (PARTITION BY NULL ORDER BY Sum(cena) desc) AS ranking*

*FROM zabieg*

*GROUP BY id\_dentysty;*

Ranking dentystów którzy wykazali największy zysk w wykonanych zabiegach.



*SELECT id\_gabinetu,*

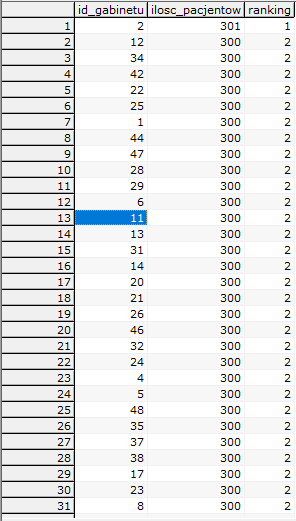
*Count(id\_pacjenta) AS ilosc\_pacjentow,*

*Rank() OVER (PARTITION BY NULL ORDER BY Count(id\_pacjenta)desc) AS ranking*

*FROM zabieg*

*GROUP BY id\_gabinetu;*

Zapytanie wyświetla ranking gabinetów, w kolejności liczby obsłużonych pacjentów.



**5. Opis interfejsu.**

**6. Podsumowanie.**

Podczas wykonywania projektu nauczyliśmy się ogólnej zasady funkcjonowania i projektowania hurtowni danych, różnic między hurtownią danych a relacyjną bazą danych a także zapytań specyficznych dla hurtowni danych.

Zapytania które można kierować do hurtowni są przydatne i mogłyby, według nas, wpłynąć na sprawniejsze funkcjonowanie rzeczywistej sieci gabinetów dentystycznych.

Uważamy że udało nam się spełnić wszystkie cele jakie postawiliśmy sobie w tym projekcie.