

Politechnika Świętokrzyska w Kielcach

Wydział Elektroniki, Automatyki i Informatyki

Projekt: Bazy Danych 2

Autorzy:
Adrian Chmielowiec
Mateusz Kubas

Grupa: **2ID12B**

Studia:
Stacjonarne 1 Stopnia

Ocena:

Kierunek:
Informatyka

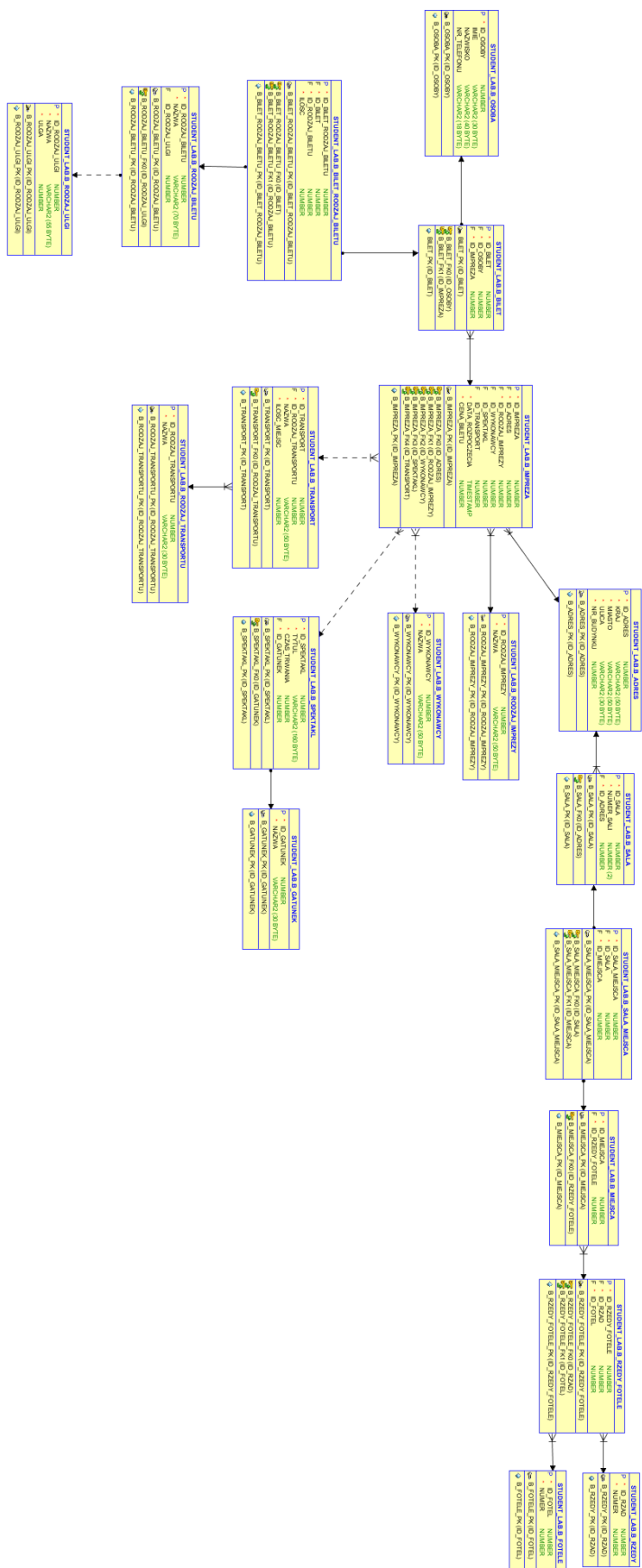
Spis treści

1. Opis problemu.....	3
2. Schemat ERD bazy.....	4
3. Opis tabel	5
4. Opis procesu transformacji bazy do hurtowni.....	7
5. Schemat ERD hurtowni	9
6. Opis procesu ładowania i transformacji danych z bazy do hurtowni	10
7. Porównanie wykonania zapytań	17
8. Uwagi i wnioski	46

1. Opis problemu

Naszym zadaniem było zaprojektowanie bazy, która będzie mogła być wykorzystywana jako system rezerwacji biletów. Głównym problemem podczas projektowania takiej bazy było zaprojektowanie jej w taki sposób aby mogła być uniwersalna i można było bez konieczności przebudowywania całej bazy dodawać nowe rodzaje imprez, wycieczek dla których to klient może zrobić rezerwacje. Utworzyliśmy więc główną tabelę o nazwie „Impreza”, która przechowuje klucze obce do słowników opisujących rodzaj imprezy, transportu, wykonawców, spektakli itd. Klucze te mogą przyjmować wartość NULL, więc możemy utworzyć ile chcemy imprez danego rodzaju np. tworząc przedstawienie teatralne tylko ten klucz obcy będzie posiadał wartość a np. klucz do tabeli transport, związany z wycieczkami będzie niepotrzebny więc będzie przyjmował wartość NULL. Kolejnym problemem było to aby 1 klient mógł zrobić rezerwacje na kilka imprez i wycieczek oraz by taka impreza lub wycieczka mogła mieć kilka rezerwacji od różnych osób. Rozwiązaliśmy ten problem tworząc tabele łączące o nazwie „Bilet” między tabelą „Impreza” a tabelą „Osoba”. Tym samym powstała nam relacja wiele do wielu między tymi dwoma tabelami. Następnym problemem z którym się zetknęliśmy było umożliwienie zakupu jednej osobie kilku biletów takiego samego lub różnego rodzaju na daną wycieczkę lub imprezę. Poradziliśmy sobie z nim tworząc tabele „Bilet_rodzaj_biletu”, która jest tabelą łączącą dla tabeli „Bilet” i „Rodzaj_biletu”. Poza przechowywaniem kluczy obcych, przechowuje ona także ilość biletów, które ktoś zakupił. Zawierając klucz obcy do tabeli „Rodzaj_biletu”, które przechowuje pole z nazwą i posiada ponad to słownik „Rodzaj_ulgi”, dzięki któremu możemy podać jaką ulgę i o ile zmniejsza cenę. Dzięki temu rozwiązaniu możliwe staje się kupienie przez jednego klienta kilku rodzajów biletów o dowolnej ilości. Ostatnim z problemów jakie napotkaliśmy była możliwość umieszczenia kilku osób w 1 Sali kinowej tak aby każdy miał swoje miejsce i była możliwość przypisania ich wiele do jednej Sali ale też by te miejsca mogły być w wielu salach a nie tylko w 1. Rozwiązaliśmy ten problem podobnie jak problem z kilkoma biletami czyli poprzez zastosowanie tabeli łączącej i relacji wiele do wielu. Tam gdzie było to możliwe zastosowaliśmy dla tabel słowniki by dane te nie musiały być powtarzane.

2. Schemat ERD bazy



3. Opis tabel

Pierwszą tabelą, którą opiszemy, jest tabela o nazwie „B_Osoba”, przechowuje ona dane na temat osób, które dokonały rezerwacji biletu lub jego zakupu. Zawiera informacje takie jak imię, nazwisko oraz numer telefonu.

Dane te są typu varchar2 i mogą być NULL. Tabela posiada swój klucz główny o nazwie „id_osoby”, po którym można zidentyfikować, kto dokonał rezerwacji lub zakupu biletu. Kolejną tabelą jest tabela „B_Bilet”, która to jest tabelą łączącą pomiędzy „B_Osoba” a „B_Impreza”. Przechowuje więc ona klucze obce do wspomnianych wcześniej tabel. Dane te nie mogą być NULL. „B_Bilet” sam też zawiera własny klucz główny, który jest reprezentantem biletu, która jakaś osoba zakupiła na jakąś wybraną przez siebie Imprezę lub wycieczkę. Następna utworzona przez nas tabela to „B_Bilet_Rodzaj_Biletu”. Przechowuje ona informacje o ilości zakupionych biletów, ponieważ jest powiązana relacją jeden do wielu z tabelą „B_Bilet”, a więc na możliwy jest zakup kilku biletów przez jedną osobę. Ponadto zawiera klucze obce do wspomnianej już tabeli „B_Bilet”, ale też do tabeli „B_Rodzaj_Biletu”. Jest więc ona tabelą łączącą pomiędzy „B_Rodzaj_Biletu” a „B_Bilet”. Wspomniana tabela „B_Rodzaj_Biletu” zawiera informacje, o tym jakiego rodzaju jest to bilet oraz klucz obcy do tabeli „B_Rodzaj_ulgi”, który może być NULL. Relacja, która łączy „B_Rodzaj_Biletu” z „B_Rodzaj_ulgi” to relacja jeden do wielu. Tabela „B_Rodzaj_ulgi” jest więc „słownikiem” dla „B_Rodzaj_Biletu” i przechowuje informacje na temat przysługujących ulg i ich zniżek. Skonstruowanie wyżej wspomnianych tabel w ten sposób pozwala osobie kupić kilka biletów różnego rodzaju z możliwą ulgą na kilka imprez lub wycieczek. Następną tabelą, którą opiszemy, jest nasza główna tabela, od której zaczęliśmy projektować naszą bazę danych, czyli „B_Impreza”. Przechowuje ona klucze obce do tabel takich jak „B_Adres”, „B_Rodzaj_Imprezy”, „B_Wykonawcy”, „B_Spektakl” i „B_Transport”. Wszystkie te tabele są w relacji jeden do wielu z tabelą „B_Impreza”. Posiada ona również pola takie jak data rozpoczęcia jakiegoś wydarzenia lub podróży o typie TIMESTAMP i posiada także informacje o cenie naszego biletu z

STUDENT_LAB.B.OSOBA		
P	ID_OSOBY	NUMBER
	IMIE	VARCHAR2(30 BYTE)
	NAZWISKO	VARCHAR2(40 BYTE)
	NR_TELEFONU	VARCHAR2(18 BYTE)
B_OSоба_PK (ID_OSOBY)		
B_OSоба_PK (ID_OSOBY)		

STUDENT_LAB.B.BILET		
P	ID_BILET	NUMBER
F	ID_OSOBY	NUMBER
F	ID_IMPENZA	NUMBER
B_BILET_PK (ID_BILET)		
B_BILET_FK0 (ID_OSOBY)		
B_BILET_FK1 (ID_IMPENZA)		
B_BILET_PK (ID_BILET)		

STUDENT_LAB.B.BILET.RODZAJ.BILETU		
P	ID_BILET.RODZAJ.BILETU	NUMBER
F	ID_BILET	NUMBER
F	ID.RODZAJ.BILETU	NUMBER
	ILOSC	NUMBER
B_BILET.RODZAJ.BILETU_PK (ID_BILET.RODZAJ.BILETU)		
B_BILET.RODZAJ.BILETU_FK0 (ID_BILET)		
B_BILET.RODZAJ.BILETU_FK1 (ID.RODZAJ.BILETU)		
B_BILET.RODZAJ.BILETU_PK (ID_BILET.RODZAJ.BILETU)		

STUDENT_LAB.B.RODZAJ.BILETU		
P	ID.RODZAJ.BILETU	NUMBER
	NAZWA	VARCHAR2(70 BYTE)
F	ID.RODZAJ_ULGI	NUMBER
B.RODZAJ.BILETU_PK (ID.RODZAJ.BILETU)		
B.RODZAJ.BILETU_FK0 (ID.RODZAJ_ULGI)		
B.RODZAJ.BILETU_PK (ID.RODZAJ.BILETU)		

STUDENT_LAB.B.RODZAJ_ULGI		
P	ID.RODZAJ_ULGI	NUMBER
	NAZWA	VARCHAR2(55 BYTE)
	ULGA	NUMBER
B.RODZAJ_ULGI_PK (ID.RODZAJ_ULGI)		
B.RODZAJ_ULGI_PK (ID.RODZAJ_ULGI)		

STUDENT_LAB.B.IMPREZA		
P	ID_IMPENZA	NUMBER
F	ID_ADRES	NUMBER
F	ID.RODZAJ_IMPENZA	NUMBER
F	ID_WYKONAWCY	NUMBER
F	ID_SPEKTAKL	NUMBER
F	ID_TRANSPORT	NUMBER
	DATA_ROZPOCZECIA	TIMESTAMP
	CENA_BILETU	NUMBER
B_IMPENZA_PK (ID_IMPENZA)		
B_IMPENZA_FK0 (ID_ADRES)		
B_IMPENZA_FK1 (ID.RODZAJ_IMPENZA)		
B_IMPENZA_FK2 (ID_WYKONAWCY)		
B_IMPENZA_FK3 (ID_SPEKTAKL)		
B_IMPENZA_FK4 (ID_TRANSPORT)		
B_IMPENZA_PK (ID_IMPENZA)		

uwzględnioną już zniżką. Z oczywistych względów klucze obce do tabel takich jak: „B_Adres”, „B_Rodzaj_Imprezy”, ale również pól z datą i ceną nie mogą być NULL w przeciwieństwie do kluczy obcych do tabel: „B_Wykonawcy”, „B_Spektakl” i „B_Transport”. Zrealizowaliśmy to w taki sposób, ponieważ dzięki temu rozwiązaniu możemy w prosty sposób wybrać, na co jest bilet, ale również łatwo można rozbudować w przyszłości o inne wydarzenia naszą bazę bez konieczności zmian w innych tabelach.

Kolejną tabelą, którą stworzyliśmy, jest „B_Transport”. Zawiera ona informacje na temat miejsc dla poszczególnych modeli samolotów, statków, autokarów, dla których to klient wykupił miejsce w celu podróży. Oprócz tego posiada też klucz obcy do tabeli „B_Rodzaj_Transportu”. Jest to relacja jeden do wielu. Tabela „B_Rodzaj_Transportu” jest tak zwanym „słownikiem” przechowującym nazwę rodzaju transportu (samolot, autokar itd.) dla tabeli „B_Transport”. Kolejna utworzona przez nas tabela to „B_Spektakl”, przechowuje ona informacje na temat tytułów i czasu trwania spektakli teatralnych, filmów w kinie i innych wydarzeń tego typu. Zawiera też klucz obcy do tabeli „B_Gatunek”. Jest to relacja jeden do wielu. Tabela „B_Gatunek” jest „słownikiem” dla tabeli „B_Spektakl” i przechowuje informacje na temat nazwy gatunku np. filmu.

Kolejną tabelą przez nas utworzoną jest „B_Wykonawcy”, posiada ona nazwy dla np. zespołów muzycznych, które występują. Następną tabelą, która opisujemy, jest „B_Rodzaj_Imprezy”. Ma ona za zadanie przechowywać nazwy na temat rodzaju imprezy, czyli na przykład czy jest to bilet do kina, czy do teatru a może na koncert. Następną tabelą to „B_Adres”. Tak jak jej nazwa wskazuje, przechowuje ona adres a konkretnie: kraj, miasto, ulice i numer budynku. Dane te są używane, chociażby by określić, do którego kina kupiliśmy nasz bilet czy wskazać gdzie odbędzie się koncert lub dokąd podróżujemy. Następną tabelą jest tabela o nazwie „B_Sala”. Przechowuje ona informacje o numerze sali, w której odbywa się nasz film czy też spektakl. Posiada też klucz obcy do tabeli „B_Adres” jest

STUDENT_LAB.B_TRANSPORT		
P	ID_TRANSPORT	NUMBER
F	ID_RODZAJ_TRANSPORTU	NUMBER
	NAZWA	VARCHAR2 (50 BYTE)
	ILOSC_MIEJSC	NUMBER
B_TRANSPORT_PK (ID_TRANSPORT)		
B_TRANSPORT_FK0 (ID_RODZAJ_TRANSPORTU)		
B_TRANSPORT_PK (ID_TRANSPORT)		

STUDENT_LAB.B_RODZAJ_TRANSPORTU		
P	ID_RODZAJ_TRANSPORTU	NUMBER
	NAZWA	VARCHAR2 (30 BYTE)
B_RODZAJ_TRANSPORTU_PK (ID_RODZAJ_TRANSPORTU)		
B_RODZAJ_TRANSPORTU_PK (ID_RODZAJ_TRANSPORTU)		

STUDENT_LAB.B_SPEKTAKL		
P	ID_SPEKTAKL	NUMBER
	TYTUL	VARCHAR2 (160 BYTE)
	CZAS_TRWANIA	NUMBER
F	ID_GATUNEK	NUMBER
B_SPEKTAKL_PK (ID_SPEKTAKL)		
B_SPEKTAKL_FK0 (ID_GATUNEK)		
B_SPEKTAKL_PK (ID_SPEKTAKL)		

STUDENT_LAB.B_GATUNEK		
P	ID_GATUNEK	NUMBER
	NAZWA	VARCHAR2 (30 BYTE)
B_GATUNEK_PK (ID_GATUNEK)		
B_GATUNEK_PK (ID_GATUNEK)		

STUDENT_LAB.B_WYKONAWCY		
P	ID_WYKONAWCY	NUMBER
	NAZWA	VARCHAR2 (50 BYTE)
B_WYKONAWCY_PK (ID_WYKONAWCY)		
B_WYKONAWCY_PK (ID_WYKONAWCY)		

STUDENT_LAB.B_RODZAJ_IMPREZY		
P	ID_RODZAJ_IMPREZY	NUMBER
	NAZWA	VARCHAR2 (50 BYTE)
B_RODZAJ_IMPREZY_PK (ID_RODZAJ_IMPREZY)		
B_RODZAJ_IMPREZY_PK (ID_RODZAJ_IMPREZY)		

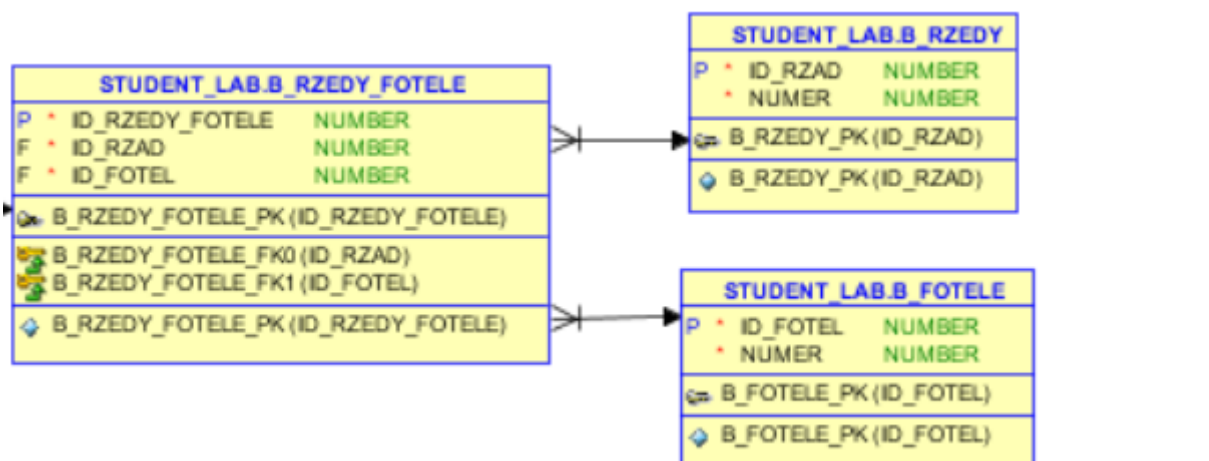
STUDENT_LAB.B_ADRES		
P	ID_ADRES	NUMBER
	KRAJ	VARCHAR2 (50 BYTE)
	MIASTO	VARCHAR2 (50 BYTE)
	ULICA	VARCHAR2 (30 BYTE)
	NR_BUDYNKU	NUMBER
B_ADRES_PK (ID_ADRES)		
B_ADRES_PK (ID_ADRES)		

STUDENT_LAB.B_SALA		
P	ID_SALA	NUMBER
	NUMER_SALI	NUMBER (2)
F	ID_ADRES	NUMBER
B_SALA_PK (ID_SALA)		
B_SALA_FK0 (ID_ADRES)		
B_SALA_PK (ID_SALA)		

więc utworzona relacja jeden do wielu, ponieważ w taki sposób do danego na przykład kina możemy przypisać wiele sali, tak jak ma to miejsce w rzeczywistości.

Następna tabela jest łącząca „B_Sala” i „B_Miejsca” o nazwie „B_Sala_miejsca”. Przechowuje ona tylko klucze obce do tych tabel, by możliwa była relacja wiele do wielu między nimi. Zrealizowaliśmy to w ten sposób, ponieważ, chcemy by jedna sala, mogła posiadać wiele miejsc, ale też dane miejsce mogło być w wielu salach. Tabela „B_Miejsca” przechowuje za to tylko klucz obcy do tabeli „B_Rzedy_fotele” jest to więc relacja jeden do wielu. Kolejna tabela to właśnie wspomniane wcześniej

„B_Rzedy_fotele”, które również przechowuje tylko klucze obce tylko tym razem do dwóch tabel: „B_Fotele” oraz „B_Rzedy”. Są to relacje jeden do wielu. Tabela „B_Fotele” przechowuje informacje o numerach foteli natomiast „B_Rzedy” informacje o numerze rzędu. Stworzyliśmy taki układ tabel i ich relacji, ponieważ w takiej konfiguracji możliwe jest by w jednym rzędzie, było wiele foteli i by dana sala miała wiele rzędów z wieloma fotelami tak jak, ma to miejsce w rzeczywistości i by można było dokonać rezerwacji na dany fotel we wskazanym przez nas rzędzie.

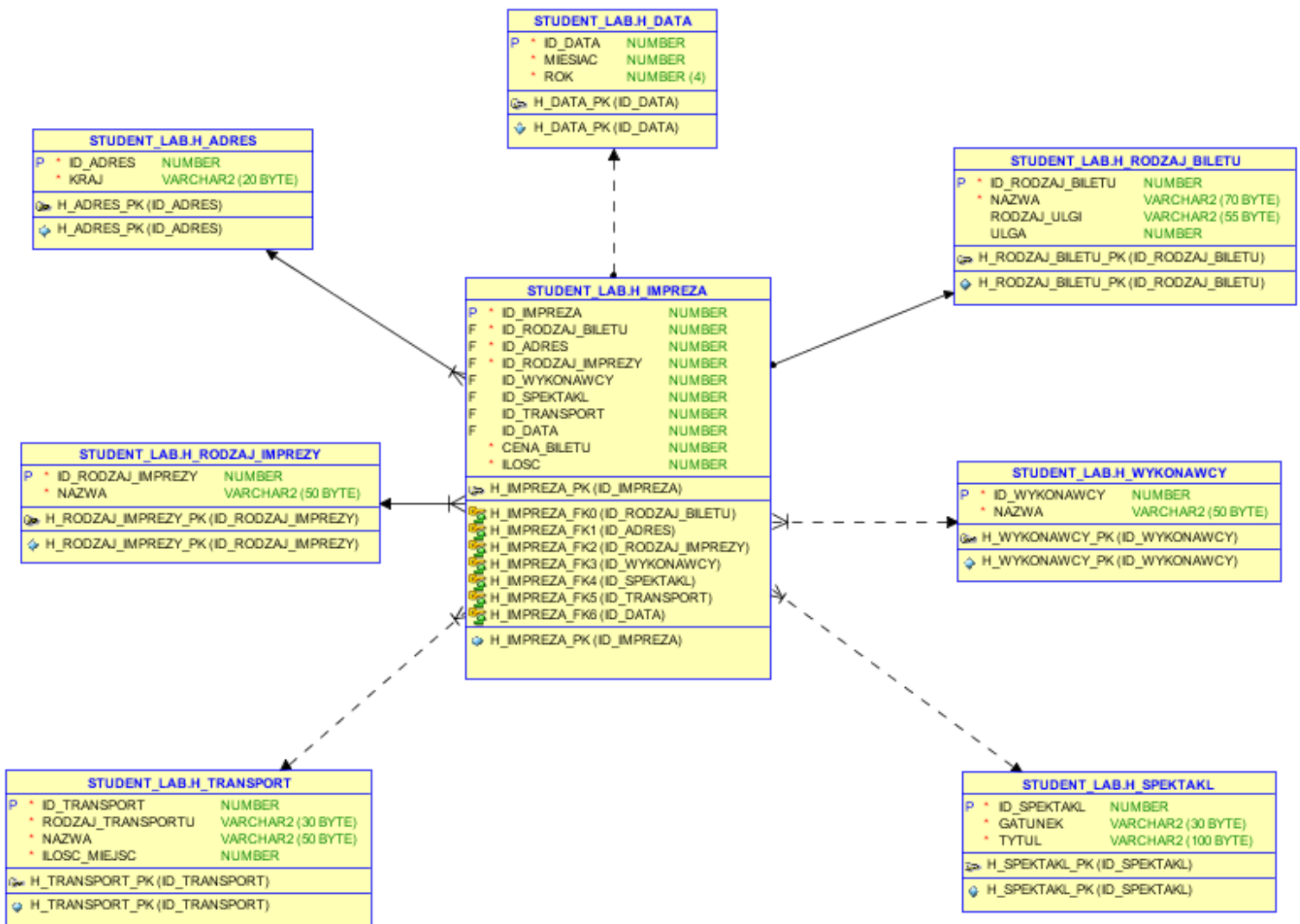


4. Opis procesu transformacji bazy do hurtowni

Transformację naszej bazy danych do hurtowni danych rozpoczęliśmy od sprawdzenia zapytań, które używamy na bazie. Po analizie ich i wypisania tabel wraz z kolumnami, które będą nam potrzebne, aby prawidłowo wykonać te same zapytania, tylko dla hurtowni by wynik był taki sam, przystąpiliśmy do realizacji. Naszą hurtownię oparliśmy o schemat gwiazdy, głównie ze względu na prostotę i szybkość wykonania na niej zapytań. Tabela faktów dla naszej hurtowni to „H_Impreza”. Odpowiednik „B_Impreza” dla bazy. Używamy na początku znaków „H_” i „B_” w celu rozróżnienia hurtowni od bazy i by możliwe było wykonywanie operacji jednocześnie na bazie i na hurtowni. Tabela „H_Impreza” zawiera

teraz klucze obce do wszystkich tabel. Została dodana również kolumna, które przechowuje ilość kupionych biletów, wcześniej wspomniana ilość znajdowała się w tabeli „B_Bilet_Rodzaj_Biletu”. Poza tą zmianą dodaliśmy również dwa nowe klucze obce. Jeden z nich odwołuje się do zupełnie nowej tabeli o nazwie „H_DATA”, jest ona w relacji jeden do wielu z tabelą „H_Impreza” i jej klucz obcy w tabeli „H_Impreza” może być NULL. Sama tabela „H_DATA” przechowuje informacje na temat daty a konkretnie miesiąca i roku, ponieważ ich tylko używamy w zapytaniach. Miesiąc i rok są przechowywane jako typ number. Poprzednio ta informacja była zawarta w kolumnie „Data_rozpoczecia” w tabeli „B_Impreza” i była typu TIMESTAMP. Kolejny nowy klucz obcy w tabeli faktów odnosi się do tabeli „H_Rodzaj_Biletu”. Jest to tabela, którą zrobiliśmy poprzez złączenie tabel „B_Rodzaj_Biletu” oraz jego słownika „B_Rodzaj_ulgi”. Zrobiliśmy tak w celu skrócenia czasu wykonywania zapytań. Tabela „H_Rodzaj_Biletu” zawiera więc teraz kolumnę z nazwą rodzaju biletu, czyli np. ulgowy, normalny. Zawiera także rodzaj ulgi np. studencka i zawiera również wysokość tej ulgi. Tabele „H_Rodzaj_Imprezy” oraz „H_Wykonawcy” pozostały bez zmian względem ich wersji dla bazy danych. Zmiany dotyczyły natomiast tabeli „H_Adres”, która przechowuje teraz tylko nazwę kraju, ponieważ tylko tę kolumnę używaliśmy w naszych zapytaniach. Kolejną tabelą, którą użyliśmy w naszej hurtowni, jest „H_Spektakl”. Różni ona się względem odpowiednika tabeli w bazie tym, że nie posiada już klucza obcego do tabeli, która przechowywała nazwy gatunków dla niej. Teraz nazwa gatunku znajduje się w tej tabeli pod nazwą „Gatunek”. Ostatnią już tabelą, którą użyliśmy, jest tabela „H_Transport”. Została ona zmodyfikowana w ten sam sposób co tabela „H_Spektakl”, czyli klucz obcy do tabeli, którą przechowywała rodzaj transportu, jak i cała tabela, zostały usunięte i teraz rodzaj transportu znajdują się w samej tabeli „H_Transport”. Reszta tabel została usunięta, ponieważ była zbędna w naszej hurtowni, gdyż nie były na nich wykonywane zapytania a kolumny, na których były z tych tabel zostały włączone do innych tabel w celu przyspieszenia działania bazy.

5. Schemat ERD hurtowni



6. Opis procesu ładowania i transformacji danych z bazy do hurtowni

Program do generacji samych danych został napisany w języku java i mimo tego, że generuje większość danych do bazy, to część danych generowana jest za pomocą internetowego generatora <https://www.mockaroo.com/>.

Tabele generowane przez nasz generator:

- Rodzaj transportu
- Rodzaj imprezy
- Sala
- Sala_miejsca
- Miejsca
- Transport
- Rodzaj biletu
- Bilet
- Bilet_rodzaj_biletu
- Impreza

Tabele generowane przez generator internetowy:

- Adres
- Fotele
- Osoba
- Rzędy
- Rzędy_fotele
- Spektakl
- Wykonawcy

Tabele, do których dane zostały napisane ręcznie:

- Gatunek
- Rodzaj ulgi

Sam generator jest zbyt obszerny, by znajdował się w sprawozdaniu, więc jego kod źródłowy znajduje się w folderze Generator-csv. Ponad to w folderze zawierającym generator znajduje się folder „wcześniej gotowe” zawierający dane przygotowane ręcznie, bądź za pomocą generatora internetowego.

Funkcja main znajduje się w pliku *FileOperations.java*.

Dane do utworzonej wcześniej bazy należy załadować za pomocą pliku *dane.bat*.

Informacje, które zawiera nasz plik przedstawione są poniżej:

```

cd Inserty/Dane
set user=STUDENT_DB/student
sqlldr USERID=%user% control=b_osoba.ctl
sqlldr USERID=%user% control=b_rodzaj_ulgi.ctl
sqlldr USERID=%user% control=b_rodzaj_biletu.ctl
sqlldr USERID=%user% control=b_rodzaj_transportu.ctl
sqlldr USERID=%user% control=b_transport.ctl
sqlldr USERID=%user% control=b_gatunek.ctl
sqlldr USERID=%user% control=b_spektakl.ctl
sqlldr USERID=%user% control=b_wykonawcy.ctl
sqlldr USERID=%user% control=b_rodzaj_imprezy.ctl
sqlldr USERID=%user% control=b_adres.ctl
sqlldr USERID=%user% control=b_sala.ctl
sqlldr USERID=%user% control=b_fotele.ctl
sqlldr USERID=%user% control=b_rzedy.ctl
sqlldr USERID=%user% control=b_rzedy_fotele.ctl
sqlldr USERID=%user% control=b_miejsca.ctl
sqlldr USERID=%user% control=b_sala_miejsca.ctl
sqlldr USERID=%user% control=b_impreza.ctl
sqlldr USERID=%user% control=b_bilet.ctl
sqlldr USERID=%user% control=b_bilet_rodzaj_biletu.ctl
pause

```

instrukcja **cd** kieruje skrypt do katalogu w którym trzymane są pliki z rozszerzeniem .ctl przechowujące informacje o sposobie ładowania danych za które odpowiada.

Przykładowy plik *b_adres.ctl*:

```

load data
infile 'b_adres.csv'
replace into table B_ADRES
fields terminated by ','
trailing nullcols
(ID_ADRES,KRAJ,MIASTO,ULICA,NR_BUDYNKU)

```

Wykonywane jest tu polecenie załadowania danych z pliku *b_adres.csv*, który takie dane przechowuje i załadowanie ich do tabeli B_ADRES, gdzie dane w pliku .csv rozdzielane są za pomocą znaku „ , ” (przecinka).

Wracając do pliku *dane.bat* przy poleceniu „set user=” należy podać nazwę użytkownika bazy, oraz poprzedzając znakiem „ / ” hasło tego użytkownika.

Poprawnie skonfigurowany powinien wgrać do bazy wszystkie podane dane po dwukrotnym naciśnięciu na plik *dane.bat*.

Transformacja danych odbywa się na zasadzie zapytań, które w odpowiedni sposób konwertują informacje z bazy do hurtowni.

Zapytania te znajdują się w pliku *B_do_H.sql*, a jego zawartość przybliżona zostanie poniżej:

B_ADRES -> H_ADRES

```
INSERT INTO H_ADRES(id_adres,kraj)
SELECT ROW_NUMBER() OVER(ORDER BY s.kraj), s.kraj
FROM
    (SELECT DISTINCT kraj FROM B_adres) s;
```

To zapytanie zamienia dane związane z adresami z bazy do hurtowni. W hurtowni używamy tylko krajów, więc wybierane zostają wszystkie występujące kraje i do posortowanych alfabetycznie krajów dopisywane jest ID, które jest kolejnym numerem wiersza.

B_RODZAJ_IMPREZY -> H_RODZAJ_IMPREZY

```
INSERT INTO H_RODZAJ_IMPREZY(id_rodzaj_imprezy, nazwa)
SELECT id_rodzaj_imprezy, nazwa FROM B_rodzaj_imprezy;
```

Dane zostają załadowane do hurtowni w niezmienionej formie.

B_SPEKTAKL + B_GATUNEK -> H_SPEKTAKL

```
INSERT INTO H_SPEKTAKL(id_spektakl,gatunek,tytul)
SELECT id_spektakl, nazwa, tytul FROM B_spektakl
JOIN B_gatunek ON B_spektakl.id_gatunek = B_gatunek.id_gatunek;
```

Dane umieszczane są w hurtowni w prawie niezmienionej formie. Tytuł i czas_trwania z B_SPEKTAKL pozostaje nie zmieniony, lecz zamiast id_gatunek zostaje wstawiona nazwa gatunku z tabeli B_GATUNEK.

data_rozpozecia w B_IMPREZA -> H_DATA

```
INSERT INTO H_DATA(id_data,miesiac,rok)
SELECT ROW_NUMBER() OVER(ORDER BY s.rok, s.miesiac), s.miesiac,
s.rok
FROM
    (SELECT DISTINCT EXTRACT(MONTH FROM data_rozpozecia) as
miesiac, EXTRACT(YEAR FROM data_rozpozecia) as rok FROM B_impreza)
s;
```

Wstawianie wartości id_data do hurtowni odbywa się tym samym sposobem jak z adreami, czyli wybierane są kolejne liczby znalezionych przez SELECT rekordów. Miesiąc i rok natomiast są „wyciągane” z pola data_rozpozecia z tabeli B_INMPREZA za pomocą metody EXTRACT i odpowiednich argumentów jak MONTH, do wyciągania miesiąca z daty i YEAR pobierający rok z daty.

B_RODZAJ_BILETU + B_RODZAJ_ULGI -> H_RODZAJ_BILETU

```
INSERT INTO h_rodzaj_biletu(id_rodzaj_biletu, nazwa, rodzaj_ulgi,
ulga)
SELECT rb.id_rodzaj_biletu AS id_rodzaj_biletu, rb.nazwa AS nazwa,
ru.nazwa AS rodzaj_ulgi, ru.ulga AS ulga
FROM B_Rodzaj_Biletu rb
LEFT JOIN B_Rodzaj_ulgi ru ON ru.id_rodzaj_ulgi = rb.id_rodzaj_ulgi;
```

Wartość id_rodzaj_biletu do hurtowni pochodzi w niezmienionej formie z id_rodzaj_biletu z tabeli B_RODZAJ_BILETU tak samo jak pole *nazwa*. Pozostałe pola pochodzą ze złączonej za pomocą LEFT JOIN tabeli B_RODZAJ_ULGI. Rodzaj_ulgi w hurtowni pozyskiwany jest z pola *nazwa* z tabeli B_RODZAJ_ULGI, a ulga bierze się z pola *ulga* z tej samej tabeli. LEFT JOIN jest tu konieczny, by zawrzeć w zwracanej tabeli bilety bez ulgi, które w miejscu na id_rodzaj_ulgi mają wartość NULL.

B_WYKONAWCY -> H_WYKONAWCY

```
INSERT INTO H_WYKONAWCY(id_wykonawcy, nazwa)
SELECT * FROM B_wykonawcy;
```

Wykonawcy w hurtowni wyglądają tak samo jak w bazie, więc przesyłanie odbywa się na zasadzie 1:1.

B_TRANSPORT + B_RODZAJ_TRANSPORTU -> H_TRANSPORT

```
INSERT INTO H_TRANSPORT(id_transport, rodzaj_transportu, nazwa,
ilosc_miejsc)
SELECT t.id_transport, rt.nazwa AS rodzaj_transportu, t.nazwa AS
nazwa, t.ilosc_miejsc AS ilosc_miejsc FROM B_transport t
JOIN B_rodzaj_transportu rt ON t.id_rodzaj_transportu =
rt.id_rodzaj_transportu;
```

Id_transport, nazwa, oraz ilość miejsc brane są z tabeli B_TRANSPORT i wkładane bezpośrednio, a jako rodzaj_transportu dla hurtowni pobierane jest pole *nazwa* z połączonej za pomocą JOIN tabeli B_RODZAJ_TRANSPORTU.

B_IMPENZA -> H_IMPENZA

```
INSERT INTO H_IMPENZA(id_impieza, id_rodzaj_biletu, id_adres,  
id_rodzaj_impiezy,  
id_wykonawcy, id_spektakl, id_transport, id_data, cena_biletu,  
ilosc)  
SELECT i.id_impieza,  
brb.id_rodzaj_biletu,  
getAdres(i.id_adres),  
getRodzajImpiezy(i.id_rodzaj_impiezy),  
getWykonawca(i.id_wykonawcy),  
getSpektakl(i.id_spektakl),  
getTransport(i.id_transport),  
getDate(i.id_impieza),  
i.cena_biletu,  
brb.ilosc  
FROM B_impieza i  
JOIN B_Bilet ON b_bilet.id_impieza = i.id_impieza  
JOIN B_Bilet_rodzaj_biletu brb ON brb.id_bilet = b_bilet.id_bilet;
```

Samo zapytanie dla imprezy wygląda właśnie w taki sposób. Id_impieza brane jest z tabeli B_IMPENZA tak samo jak cena biletu. Ilość biletów musi być jednak pobierana z tabeli B_BILET_RODZAJ_BILETU. Żeby się do niej dostać trzeba przejść najpierw przez tabelę B_BILET, więc ta właśnie tabela podłączona została do B_IMPENZA, a dopiero do niej tabela docelowa B_BILET_RODZAJ_BILETU.

Dane, szczególnie w wielkich ilościach mogą nie przenosić się dokładnie w sposób uporządkowany, a łącząc to z faktem, że id każdego wiersza wstawianego do hurtowni kontrolowane jest przez sekwencję doprowadza to do sytuacji, gdzie pierwotne odnośniki w postaci id w takiej tabeli jak H_IMPENZA mogą nie istnieć, bądź być pomieszane. W takiej sytuacji do akcji wchodzi funkcje specjalnie napisane, by szukać odpowiednich id danych w hurtowni na podstawie starych id z bazy.

Przedstawię je teraz i opisze wszystkie na raz, ponieważ działają w bardzo zbliżony sposób.

```
CREATE OR REPLACE FUNCTION getDate (impieza_id NUMBER)  
RETURN NUMBER  
IS data_id NUMBER;  
BEGIN  
    SELECT hd.id_data INTO data_id  
    FROM H_DATA hd, (SELECT id_impieza, EXTRACT(MONTH FROM  
B_impieza.data_roz poczeczia) as miesiac, EXTRACT(YEAR FROM  
B_impieza.data_roz poczeczia) as rok FROM B_impieza) s  
    WHERE hd.miesiac = s.miesiac AND hd.rok = s.rok AND  
s.id_impieza = impieza_id;  
    RETURN (data_id);
```

```

        END;
    /

CREATE OR REPLACE FUNCTION getAdres (B_adres_id NUMBER)
RETURN NUMBER
IS H_adres_id NUMBER;
BEGIN
    SELECT id_adres INTO H_adres_id FROM H_ADRES
    WHERE kraj =
    (
        SELECT kraj FROM B_ADRES WHERE id_adres = B_adres_id
    );
    RETURN (H_adres_id);
END;

/

CREATE OR REPLACE FUNCTION getSpektakl (old_id NUMBER)
RETURN NUMBER
IS new_id NUMBER;
BEGIN
    SELECT id_spektakl INTO new_id FROM H_SPEKTAKL
    WHERE tytuł LIKE
    (
        SELECT tytuł FROM B_SPEKTAKL WHERE id_spektakl = old_id
    );
    RETURN (new_id);
END;

/

CREATE OR REPLACE FUNCTION getTransport (old_id NUMBER)
RETURN NUMBER
IS new_id NUMBER;
BEGIN
    SELECT id_transport INTO new_id FROM H_TRANSPORT
    WHERE nazwa LIKE
    (
        SELECT nazwa FROM B_TRANSPORT WHERE id_transport =
old_id
    );
    RETURN (new_id);
END;

/

```



```

CREATE OR REPLACE FUNCTION getRodzajImprezy (old_id NUMBER)
RETURN NUMBER
IS new_id NUMBER;
BEGIN
    SELECT id_rodzaj_imprezy INTO new_id FROM H_RODZAJ_IMPREZY
    WHERE nazwa LIKE
    (
        SELECT nazwa FROM B_RODZAJ_IMPREZY WHERE
id_rodzaj_imprezy = old_id
    );
    RETURN (new_id);
END;
/

CREATE OR REPLACE FUNCTION getWykonawca (old_id NUMBER)
RETURN NUMBER
IS new_id NUMBER;
BEGIN
    SELECT id_wykonawcy INTO new_id FROM H_WYKONAWCY
    WHERE nazwa LIKE
    (
        SELECT nazwa FROM B_WYKONAWCY WHERE id_wykonawcy =
old_id
    );
    RETURN (new_id);
END;
/

```

Właśnie te funkcje są wykorzystywane przy wstawianiu odpowiednich danych do H_IMPREZA. Ich działanie opiera się na wywołaniu nazwy funkcji i podaniu w argumencie id szukanego adresu, wykonawcy, itd. z bazy. Funkcja wykonuje po tym zapytanie SELECT, którego zadaniem jest pobranie poprawnego id od tabeli w hurtowni i zapisanie go do zwracanej potem zmiennej. Id jest wybierane za pomocą instrukcji WHERE która operuje na tabeli zwracanej przez podzapytanie zwracające charakterystyczną dla danej tabeli kolumną jak np. *nazwa*, która znajduje się właśnie za pomocą podanego id, które jest porównywane do id w tabeli w bazie.

System ten upewnia się, że powiązania do odpowiednich danych nie pomieszą się w trakcie przenoszenia.

Po operacji wstawiania rekordów do H_IMPREZA funkcje są usuwane za pomocą DROP FUNCTION.

7. Porównanie wykonania zapytań

Zrzuty ekranu przedstawiane są w kolejności: baza, hurtownia.

Czas wykonania zapytań ROLLUP:

Opis zapytania: Podsumowanie finansowe 1

Kod zapytania bazy:

```
SELECT B_Rodzaj_Imprezy.Nazwa AS "Rodzaj", B_Transport.Nazwa AS "Pojazd", sum(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) "Suma kwota"
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_impirezy = B_Rodzaj_Imprezy.id_rodzaj_impirezy JOIN B_Transport ON B_Impreza.id_transport = B_Transport.id_transport JOIN B_Bilet ON B_Bilet.id_impireza = B_IMPREZA.id_impireza JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet = B_Bilet.id_bilet
group by rollup(B_Rodzaj_Imprezy.Nazwa,B_Transport.Nazwa);
```

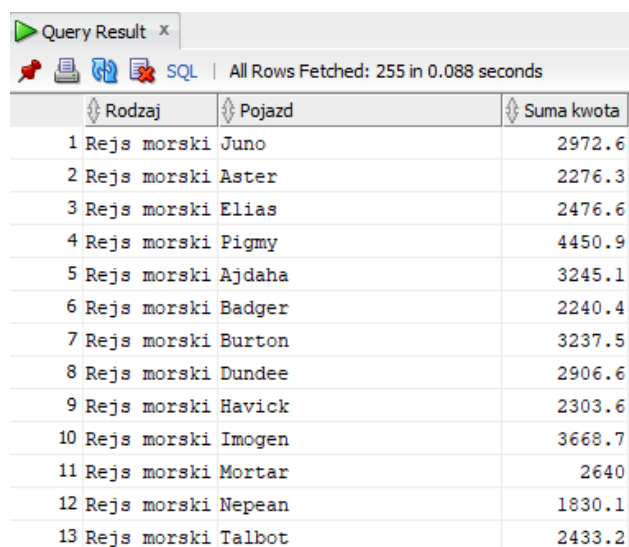
Kod zapytania hurtowni:

```
SELECT H_Rodzaj_Imprezy.Nazwa AS "Rodzaj", H_Transport.Nazwa AS "Pojazd", sum(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) "Suma kwota"
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_impirezy = H_Rodzaj_Imprezy.id_rodzaj_impirezy JOIN H_Transport ON H_Impreza.id_transport = H_Transport.id_transport
group by rollup(H_Rodzaj_Imprezy.Nazwa,H_Transport.Nazwa);
```

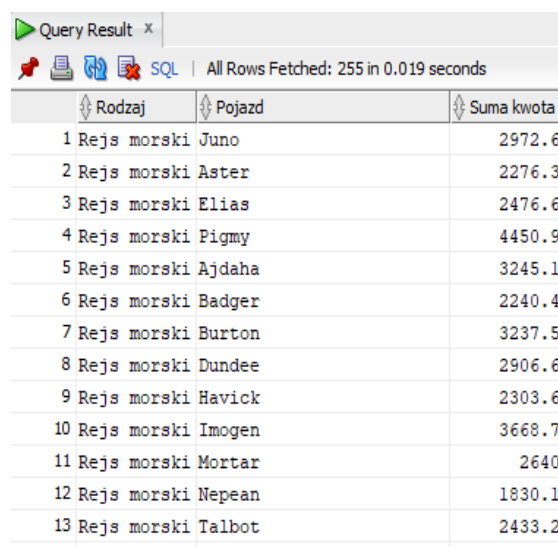
Czas dla bazy: 0.088s

Czas dla hurtowni: 0.019s

Zrzuty ekranów:



Rodzaj	Pojazd	Suma kwota
1 Rejs morski	Juno	2972.6
2 Rejs morski	Aster	2276.3
3 Rejs morski	Elias	2476.6
4 Rejs morski	Pigmy	4450.9
5 Rejs morski	Ajdaha	3245.1
6 Rejs morski	Badger	2240.4
7 Rejs morski	Burton	3237.5
8 Rejs morski	Dundee	2906.6
9 Rejs morski	Havick	2303.6
10 Rejs morski	Imogen	3668.7
11 Rejs morski	Mortar	2640
12 Rejs morski	Nepean	1830.1
13 Rejs morski	Talbot	2433.2



Rodzaj	Pojazd	Suma kwota
1 Rejs morski	Juno	2972.6
2 Rejs morski	Aster	2276.3
3 Rejs morski	Elias	2476.6
4 Rejs morski	Pigmy	4450.9
5 Rejs morski	Ajdaha	3245.1
6 Rejs morski	Badger	2240.4
7 Rejs morski	Burton	3237.5
8 Rejs morski	Dundee	2906.6
9 Rejs morski	Havick	2303.6
10 Rejs morski	Imogen	3668.7
11 Rejs morski	Mortar	2640
12 Rejs morski	Nepean	1830.1
13 Rejs morski	Talbot	2433.2

Opis zapytania: Podsumowanie finansowe 2

Kod zapytania bazy:

```
SELECT B_Rodzaj_Imprezy.Nazwa AS "Rodzaj", B_Wykonawcy.Nazwa AS  
"Wykonawca/y",  
sum(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) "Suma kwota"  
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_impreszy  
= B_Rodzaj_Imprezy.id_rodzaj_impreszy JOIN B_Wykonawcy ON  
B_Impreza.id_wykonawcy = B_Wykonawcy.id_wykonawcy  
JOIN B_Bilet ON B_Bilet.id_impresza = B_IMPREZA.id_impresza  
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =  
B_Bilet.id_bilet  
group by rollup(B_Rodzaj_Imprezy.Nazwa,B_Wykonawcy.Nazwa);
```

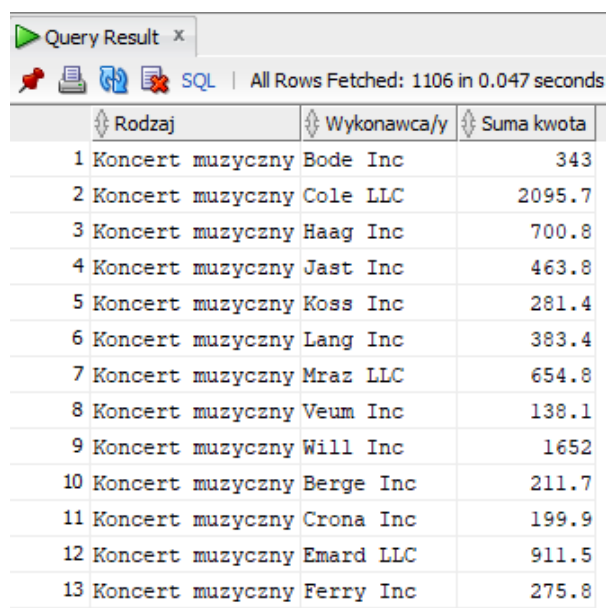
Kod zapytania hurtowni:

```
SELECT H_Rodzaj_Imprezy.Nazwa AS "Rodzaj", H_Wykonawcy.Nazwa AS  
"Wykonawca/y", sum(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) "Suma  
kwota"  
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_impreszy  
= H_Rodzaj_Imprezy.id_rodzaj_impreszy JOIN H_Wykonawcy ON  
H_Impreza.id_wykonawcy = H_Wykonawcy.id_wykonawcy  
group by rollup(H_Rodzaj_Imprezy.Nazwa,H_Wykonawcy.Nazwa);
```

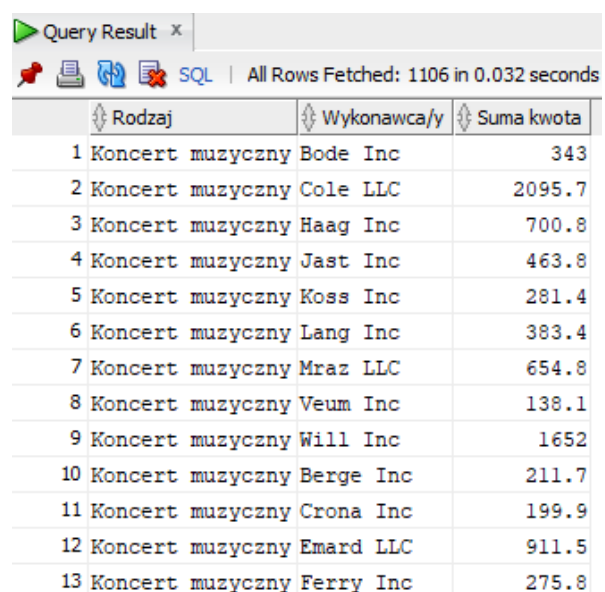
Czas dla bazy: 0.047s

Czas dla hurtowni: 0.032s

Zrzuty ekranów:



Rodzaj	Wykonawca/y	Suma kwota
1 Koncert muzyczny	Bode Inc	343
2 Koncert muzyczny	Cole LLC	2095.7
3 Koncert muzyczny	Haag Inc	700.8
4 Koncert muzyczny	Jast Inc	463.8
5 Koncert muzyczny	Koss Inc	281.4
6 Koncert muzyczny	Lang Inc	383.4
7 Koncert muzyczny	Mraz LLC	654.8
8 Koncert muzyczny	Veum Inc	138.1
9 Koncert muzyczny	Will Inc	1652
10 Koncert muzyczny	Berge Inc	211.7
11 Koncert muzyczny	Crona Inc	199.9
12 Koncert muzyczny	Emard LLC	911.5
13 Koncert muzyczny	Ferry Inc	275.8



Rodzaj	Wykonawca/y	Suma kwota
1 Koncert muzyczny	Bode Inc	343
2 Koncert muzyczny	Cole LLC	2095.7
3 Koncert muzyczny	Haag Inc	700.8
4 Koncert muzyczny	Jast Inc	463.8
5 Koncert muzyczny	Koss Inc	281.4
6 Koncert muzyczny	Lang Inc	383.4
7 Koncert muzyczny	Mraz LLC	654.8
8 Koncert muzyczny	Veum Inc	138.1
9 Koncert muzyczny	Will Inc	1652
10 Koncert muzyczny	Berge Inc	211.7
11 Koncert muzyczny	Crona Inc	199.9
12 Koncert muzyczny	Emard LLC	911.5
13 Koncert muzyczny	Ferry Inc	275.8

Opis zapytania: Podsumowanie finansowe 3

Kod zapytania bazy:

```
SELECT B_Rodzaj_Imprezy.Nazwa AS "Rodzaj", B_Gatunek.Nazwa AS
"Gatunek", sum(B_Impreza.cena_biletu*B_Bilet_Rodzaj_Biletu.ilosc)
"Suma kwota"
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_impreszy
= B_Rodzaj_Imprezy.id_rodzaj_impreszy JOIN B_Spektakl ON
B_Impreza.id_spektakl = B_Spektakl.id_spektakl JOIN B_Gatunek ON
B_Spektakl.id_gatunek = B_Gatunek.id_gatunek
JOIN B_Bilet ON B_Bilet.id_impresza = B_IMPRESZA.id_impresza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
group by rollup(B_Rodzaj_Imprezy.Nazwa,B_Gatunek.Nazwa);
```

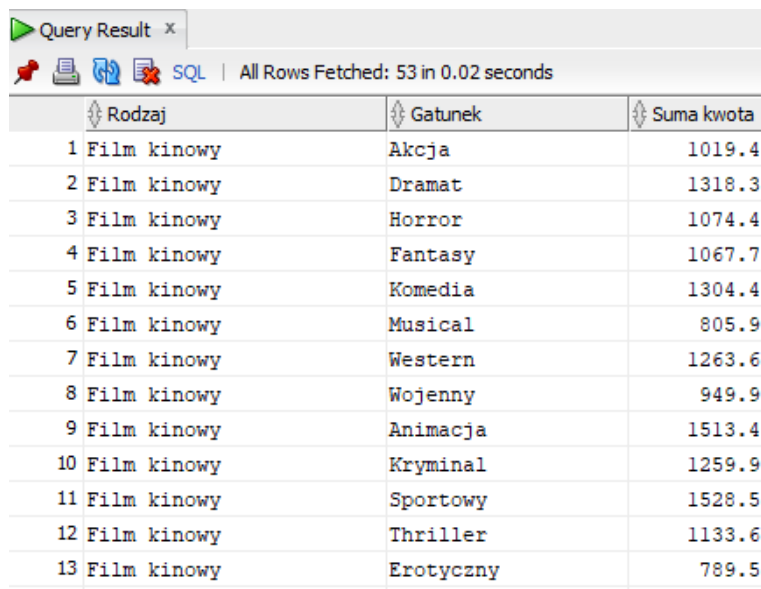
Kod zapytania hurtowni:

```
SELECT H_Rodzaj_Imprezy.Nazwa AS "Rodzaj", H_Spektakl.Gatunek,
sum(H_Impreza.cena_biletu*H_IMPRESZA.ilosc) "Suma kwota"
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_impreszy
= H_Rodzaj_Imprezy.id_rodzaj_impreszy JOIN H_Spektakl ON
H_Impreza.id_spektakl = H_Spektakl.id_spektakl
group by rollup(H_Rodzaj_Imprezy.Nazwa,H_Spektakl.Gatunek);
```

Czas dla bazy: 0.02s





Czas dla hurtowni: 0.009s

Zrzuty ekranów:



	Rodzaj	Gatunek	Suma kwota
1	Film kinowy	Akcja	1019.4
2	Film kinowy	Dramat	1318.3
3	Film kinowy	Horror	1074.4
4	Film kinowy	Fantasy	1067.7
5	Film kinowy	Komedia	1304.4
6	Film kinowy	Musical	805.9
7	Film kinowy	Western	1263.6
8	Film kinowy	Wojenny	949.9
9	Film kinowy	Animacja	1513.4
10	Film kinowy	Kryminal	1259.9
11	Film kinowy	Sportowy	1528.5
12	Film kinowy	Thriller	1133.6
13	Film kinowy	Erotyczny	789.5

Query Result x

    SQL | All Rows Fetched: 53 in 0.009 seconds

	Rodzaj	GATUNEK	Suma kwota
1	Film kinowy	Akcja	1019.4
2	Film kinowy	Dramat	1318.3
3	Film kinowy	Horror	1074.4
4	Film kinowy	Fantasy	1067.7
5	Film kinowy	Komedia	1304.4
6	Film kinowy	Musical	805.9
7	Film kinowy	Western	1263.6
8	Film kinowy	Wojenny	949.9
9	Film kinowy	Animacja	1513.4
10	Film kinowy	Kryminal	1259.9
11	Film kinowy	Sportowy	1528.5
12	Film kinowy	Thriller	1133.6
13	Film kinowy	Erotyczny	789.5

Opis zapytania: Zliczanie ilości biletów

Kod zapytania bazy:

```
SELECT B_Rodzaj_Imprezy.Nazwa AS "Rodzaj", B_Rodzaj_Biletu.Nazwa AS
"Rodzaj biletu", sum(B_Bilet_Rodzaj_Biletu.ILOSC) "Ilosc biletow"
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_impreszy
= B_Rodzaj_Imprezy.id_rodzaj_impreszy JOIN B_Bilet ON
B_Bilet.id_impresza = B_Impreza.id_impresza JOIN B_Bilet_Rodzaj_Biletu
ON B_Bilet_Rodzaj_Biletu.id_bilet = B_Bilet.id_bilet JOIN
B_Rodzaj_biletu ON B_Rodzaj_biletu.id_rodzaj_biletu =
B_Bilet_Rodzaj_Biletu.id_rodzaj_biletu
group by rollup(B_Rodzaj_Imprezy.Nazwa,B_Rodzaj_Biletu.Nazwa);
```

Kod zapytania hurtowni:





```
SELECT H_Rodzaj_Imprezy.Nazwa AS "Rodzaj", H_Rodzaj_Biletu.Nazwa AS
"Rodzaj biletu", sum(H_IMPRESZA.ilosc) "Ilosc biletow"
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_impreszy
= H_Rodzaj_Imprezy.id_rodzaj_impreszy JOIN H_Rodzaj_biletu ON
H_Rodzaj_biletu.id_rodzaj_biletu = H_IMPRESZA.id_rodzaj_biletu
group by rollup(H_Rodzaj_Imprezy.Nazwa,H_Rodzaj_Biletu.Nazwa);
```

Czas dla bazy: 0.022s

Czas dla hurtowni: 0.014s





Zrzuty ekranów:

Query Result x

 All Rows Fetched: 51 in 0.022 seconds

Rodzaj	Rodzaj biletu	Ilosc biletow
1 Film kinowy	Bilet ulgowy	1262
2 Film kinowy	Bilet grupowy	48
3 Film kinowy	Bilet normalny	42
4 Film kinowy	(null)	1352
5 Rejs morski	Bilet rejsowy ulgowy	1560
6 Rejs morski	Bilet rejsowy normalny	66
7 Rejs morski	Bilet przewozowy ulgowy	83
8 Rejs morski	Bilet przewozowy grupowy	40
9 Rejs morski	(null)	1749
10 Lot pasazerski	Bilet lotniczy ulgowy	1183
11 Lot pasazerski	Bilet lotniczy grupowy	62
12 Lot pasazerski	Bilet autokarowy ulgowy	459
13 Lot pasazerski	Bilet lotniczy normalny	61

Query Result x

 SQL | All Rows Fetched: 51 in 0.014 seconds

	Rodzaj	Rodzaj biletu	Ilosc biletow
1	Film kinowy	Bilet ulgowy	1262
2	Film kinowy	Bilet grupowy	48
3	Film kinowy	Bilet normalny	42
4	Film kinowy	(null)	1352
5	Rejs morski	Bilet rejsowy ulgowy	1560
6	Rejs morski	Bilet rejsowy normalny	66
7	Rejs morski	Bilet przewozowy ulgowy	83
8	Rejs morski	Bilet przewozowy grupowy	40
9	Rejs morski	(null)	1749
10	Lot pasazerski	Bilet lotniczy ulgowy	1183
11	Lot pasazerski	Bilet lotniczy grupowy	62
12	Lot pasazerski	Bilet autokarowy ulgowy	459
13	Lot pasazerski	Bilet lotniczy normalny	61

Opis zapytania: Zliczanie jaki pojazd jest najczęściej wybierany wśród klientów 1

Kod zapytania bazy:

```
SELECT B_Rodzaj_Imprezy.Nazwa AS "Rodzaj", B_Transport.nazwa AS  
"Pojazd", sum(B_Bilet_Rodzaj_Biletu.ILOSC) AS "Ilosc biletow"  
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_imprezy  
= B_Rodzaj_Imprezy.id_rodzaj_imprezy JOIN B_Bilet ON  
B_Bilet.id_impreza = B_Impreza.id_impreza JOIN B_Bilet_Rodzaj_Biletu  
ON B_Bilet_Rodzaj_Biletu.id_bilet = B_Bilet.id_bilet JOIN  
B_Transport ON B_Impreza.id_transport = B_Transport.id_transport  
group by rollup(B_Rodzaj_Imprezy.Nazwa,B_Transport.nazwa);
```

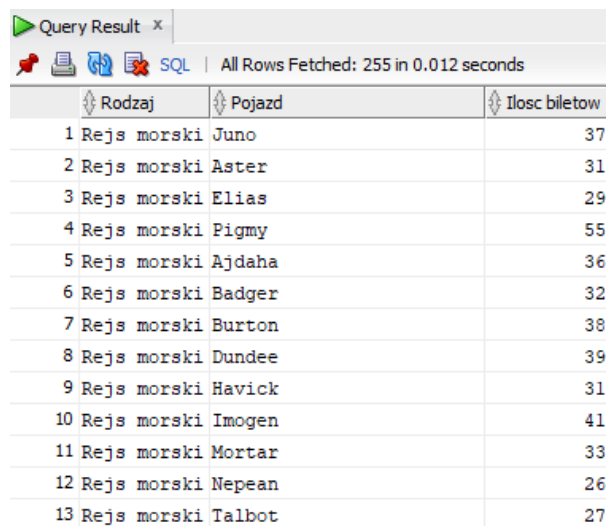
Kod zapytania hurtowni:

```
SELECT H_Rodzaj_Imprezy.Nazwa AS "Rodzaj", H_Transport.nazwa AS  
"Pojazd", sum(H_IMPENZA.ilosc) AS "Ilosc biletow"  
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_imprezy  
= H_Rodzaj_Imprezy.id_rodzaj_imprezy JOIN H_Transport ON  
H_Impreza.id_transport = H_Transport.id_transport  
group by rollup(H_Rodzaj_Imprezy.Nazwa,H_Transport.nazwa);
```

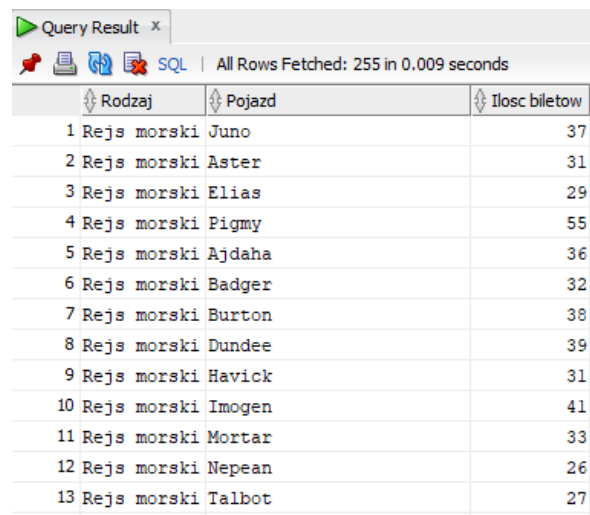
Czas dla bazy: 0.012s

Czas dla hurtowni: 0.009s

Zrzuty ekranów:



Rodzaj	Pojazd	Ilosc biletow
1 Rejs morski	Juno	37
2 Rejs morski	Aster	31
3 Rejs morski	Elias	29
4 Rejs morski	Pigmy	55
5 Rejs morski	Ajdaha	36
6 Rejs morski	Badger	32
7 Rejs morski	Burton	38
8 Rejs morski	Dundee	39
9 Rejs morski	Havick	31
10 Rejs morski	Imogen	41
11 Rejs morski	Mortar	33
12 Rejs morski	Nepean	26
13 Rejs morski	Talbot	27



Rodzaj	Pojazd	Ilosc biletow
1 Rejs morski	Juno	37
2 Rejs morski	Aster	31
3 Rejs morski	Elias	29
4 Rejs morski	Pigmy	55
5 Rejs morski	Ajdaha	36
6 Rejs morski	Badger	32
7 Rejs morski	Burton	38
8 Rejs morski	Dundee	39
9 Rejs morski	Havick	31
10 Rejs morski	Imogen	41
11 Rejs morski	Mortar	33
12 Rejs morski	Nepean	26
13 Rejs morski	Talbot	27

Opis zapytania: Zliczanie jaki pojazd jest najczęściej wybierany wśród klientów 2

Kod zapytania bazy:

```
SELECT B_Rodzaj_Transportu.Nazwa AS "Rodzaj", B_Transport.nazwa AS "Pojazd", sum(B_Bilet_Rodzaj_Biletu.ILOSC) AS "Ilosc biletow"
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_imprezy = B_Rodzaj_Imprezy.id_rodzaj_imprezy JOIN B_Bilet ON
B_Bilet.id_impreza = B_Impreza.id_impreza JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet = B_Bilet.id_bilet JOIN
B_Transport ON B_Impreza.id_transport = B_Transport.id_transport JOIN B_Rodzaj_Transportu ON B_Transport.id_rodzaj_transportu =
B_Rodzaj_Transportu.id_rodzaj_transportu
group by rollup(B_Rodzaj_Transportu.Nazwa,B_Transport.nazwa);
```

Kod zapytania hurtowni:

```
SELECT H_Transport.Rodzaj_Transportu AS "Rodzaj", H_Transport.nazwa AS "Pojazd", sum(H_IMPENZA.ilosc) AS "Ilosc biletow"
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_imprezy = H_Rodzaj_Imprezy.id_rodzaj_imprezy JOIN H_Transport ON
H_Impreza.id_transport = H_Transport.id_transport
group by rollup(H_Transport.Rodzaj_Transportu,H_Transport.nazwa);
```

Czas dla bazy: 0.03s

Czas dla hurtowni: 0.017s

Zrzuty ekranów:

Query Result x			Query Result x		
SQL All Rows Fetched: 204 in 0.03 seconds			SQL All Rows Fetched: 204 in 0.017 seconds		
Rodzaj	Pojazd	Ilosc biletow	Rodzaj	Pojazd	Ilosc biletow
1 Pociag	THE GHAN	22	1 Pociag	THE GHAN	22
2 Pociag	RENFE AVE	31	2 Pociag	RENFE AVE	31
3 Pociag	BLUE TRAIN	18	3 Pociag	BLUE TRAIN	18
4 Pociag	ROVOS TRAIL	12	4 Pociag	ROVOS TRAIL	12
5 Pociag	Fogland Line	35	5 Pociag	Fogland Line	35
6 Pociag	GOLDEN EAGLE	27	6 Pociag	GOLDEN EAGLE	27
7 Pociag	Eastern Route	57	7 Pociag	Eastern Route	57
8 Pociag	Ironbark Line	61	8 Pociag	Ironbark Line	61
9 Pociag	Lowland Route	37	9 Pociag	Lowland Route	37
10 Pociag	Savannah Line	21	10 Pociag	Savannah Line	21
11 Pociag	Somerset Line	27	11 Pociag	Somerset Line	27
12 Pociag	SHANGHAI MAGLEV	28	12 Pociag	SHANGHAI MAGLEV	28
13 Pociag	Serenity Tracks	36	13 Pociag	Serenity Tracks	36

Czas wykonania zapytań CUBE:

Opis zapytania: Preferencje gatunkowe klientów 1

Kod zapytania bazy:

```
SELECT B_Adres.Kraj AS "Kraj", B_Gatunek.nazwa AS "Gatunek",  
B_Spektakl.Tytul AS "Tytul", sum(B_Bilet_Rodzaj_Biletu.ILOSC) AS  
"Ilosc biletow"  
FROM B_Impreza JOIN B_Bilet ON B_Bilet.id_impresa =  
B_Impreza.id_impresa JOIN B_Bilet_Rodzaj_Biletu ON  
B_Bilet_Rodzaj_Biletu.id_bilet = B_Bilet.id_bilet JOIN B_ADRES ON  
B_IMPENZA.id_adres = B_ADRES.id_adres JOIN B_SPEKTAKL ON  
B_SPEKTAKL.ID_SPEKTAKL = B_IMPENZA.ID_SPEKTAKL JOIN B_GATUNEK ON  
B_SPEKTAKL.ID_GATUNEK = B_GATUNEK.ID_GATUNEK  
group by cube(B_Adres.Kraj, B_Gatunek.nazwa, B_Spektakl.Tytul);
```

Kod zapytania hurtowni:

```
SELECT H_Adres.Kraj AS "Kraj", H_SPEKTAKL.GATUNEK AS "Gatunek",  
H_Spektakl.Tytul AS "Tytul", sum(H_IMPENZA.ilosc) AS "Ilosc biletow"  
FROM H_Impreza JOIN H_ADRES ON H_IMPENZA.id_adres = H_ADRES.id_adres  
JOIN H_SPEKTAKL ON H_SPEKTAKL.ID_SPEKTAKL = H_IMPENZA.ID_SPEKTAKL  
group by cube(H_Adres.Kraj, H_Spektakl.GATUNEK, H_Spektakl.Tytul);
```

Czas dla bazy: 0.055s

Czas dla hurtowni: 0.05s

Zrzuty ekranów:

	Kraj	Gatunek	Tytul	Ilosc biletow
1	(null)	(null)	(null)	3386
2	(null)	(null)	15	15
3	(null)	(null)	360	2
4	(null)	(null)	Mac	5
5	(null)	(null)	1981	3
6	(null)	(null)	2:37	4
7	(null)	(null)	Anna	3
8	(null)	(null)	Ca\$h	1
9	(null)	(null)	Coma	2
10	(null)	(null)	Cops	4
11	(null)	(null)	DiG!	3
12	(null)	(null)	Dope	8
13	(null)	(null)	Duel	10

	Kraj	Gatunek	Tytul	Ilosc biletow
1	(null)	(null)	(null)	3386
2	(null)	(null)	15	15
3	(null)	(null)	360	2
4	(null)	(null)	Mac	5
5	(null)	(null)	1981	3
6	(null)	(null)	2:37	4
7	(null)	(null)	Anna	3
8	(null)	(null)	Ca\$h	1
9	(null)	(null)	Coma	2
10	(null)	(null)	Cops	4
11	(null)	(null)	DiG!	3
12	(null)	(null)	Dope	8
13	(null)	(null)	Duel	10

Opis zapytania: Preferencje gatunkowe klientów 2

Kod zapytania bazy:

```
SELECT B_Adres.Kraj AS "Kraj", B_Gatunek.nazwa AS "Gatunek",  
sum(B_Bilet_Rodzaj_Biletu.ILOSC) AS "Ilosc biletow"  
FROM B_Impreza JOIN B_Bilet ON B_Bilet.id_impreza =  
B_Impreza.id_impreza JOIN B_Bilet_Rodzaj_Biletu ON  
B_Bilet_Rodzaj_Biletu.id_bilet = B_Bilet.id_bilet JOIN B_ADRES ON  
B_IMPENZA.id_adres = B_ADRES.id_adres JOIN B_SPEKTAKL ON  
B_SPEKTAKL.ID_SPEKTAKL = B_IMPENZA.ID_SPEKTAKL JOIN B_GATUNEK ON  
B_SPEKTAKL.ID_GATUNEK = B_GATUNEK.ID_GATUNEK  
group by cube(B_Adres.Kraj, B_Gatunek.nazwa);
```

Kod zapytania hurtowni:

```
SELECT H_Adres.Kraj AS "Kraj", H_SPEKTAKL.GATUNEK AS "Gatunek" ,  
sum(H_IMPENZA.ilosc) AS "Ilosc biletow"  
FROM H_Impreza JOIN H_ADRES ON H_IMPENZA.id_adres = H_ADRES.id_adres  
JOIN H_SPEKTAKL ON H_SPEKTAKL.ID_SPEKTAKL = H_IMPENZA.ID_SPEKTAKL  
group by cube(H_Adres.Kraj, H_SPEKTAKL.GATUNEK);
```

Czas dla bazy: 0.017s

Czas dla hurtowni: 0.006s

Zrzuty ekranów:

Query Result x			
All Rows Fetched: 312 in 0.017 seconds			
Kraj	Gatunek	Ilosc biletow	
1 (null)	(null)	3386	
2 (null)	Akcja	149	
3 (null)	Dramat	187	
4 (null)	Horror	150	
5 (null)	Fantasy	153	
6 (null)	Komedia	146	
7 (null)	Musical	120	
8 (null)	Western	104	
9 (null)	Wojenny	155	
10 (null)	Animacja	133	
11 (null)	Kryminal	164	
12 (null)	Sportowy	152	
13 (null)	Thriller	123	

Query Result x			
All Rows Fetched: 312 in 0.006 seconds			
Kraj	Gatunek	Ilosc biletow	
1 (null)	(null)	3386	
2 (null)	Akcja	149	
3 (null)	Dramat	187	
4 (null)	Horror	150	
5 (null)	Fantasy	153	
6 (null)	Komedia	146	
7 (null)	Musical	120	
8 (null)	Western	104	
9 (null)	Wojenny	155	
10 (null)	Animacja	133	
11 (null)	Kryminal	164	
12 (null)	Sportowy	152	
13 (null)	Thriller	123	

Opis zapytania: Preferencje przewozowe klientów 1

Kod zapytania bazy:

```
SELECT B_Adres.Kraj AS "Kraj", B_Rodzaj_Transportu.nazwa AS "Rodzaj
Transportu", sum(B_Bilet_Rodzaj_Biletu.ILOSC) AS "Ilosc biletow"
FROM B_Impreza JOIN B_Bilet ON B_Bilet.id_impreza =
B_Impreza.id_impreza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
JOIN B_ADRES ON B_IMPENZA.id_adres = B_ADRES.id_adres
JOIN B_Transport ON B_Impreza.id_transport =
B_Transport.id_transport
JOIN B_Rodzaj_Transportu ON B_Transport.id_rodzaj_transportu =
B_Rodzaj_Transportu.id_rodzaj_transportu
group by cube(B_Adres.Kraj, B_Rodzaj_Transportu.nazwa);
```

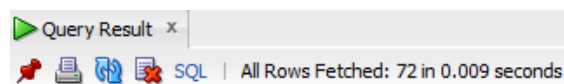
Kod zapytania hurtowni:

```
SELECT H_Adres.Kraj AS "Kraj", H_Transport.Rodzaj_Transportu AS
"Rodzaj Transportu", sum(H_IMPENZA.ilosc) AS "Ilosc biletow"
FROM H_Impreza JOIN H_ADRES ON H_IMPENZA.id_adres = H_ADRES.id_adres
JOIN H_Transport ON H_Impreza.id_transport =
H_Transport.id_transport
group by cube(H_Adres.Kraj, H_Transport.Rodzaj_Transportu);
```

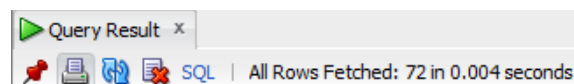
Czas dla bazy: 0.009s

Czas dla hurtowni: 0.004s

Zrzuty ekranów:



	Kraj	Rodzaj Transportu	Ilosc biletow
1	(null)	(null)	8456
2	(null)	Pociag	1757
3	(null)	Statek	1749
4	(null)	Autokar	3108
5	(null)	Samolot	1842
6	Italy	(null)	22
7	Italy	Statek	22
8	Japan	(null)	796
9	Japan	Pociag	118
10	Japan	Statek	149
11	Japan	Autokar	333
12	Japan	Samolot	196
13	Brazil	(null)	1318



	Kraj	Rodzaj Transportu	Ilosc biletow
1	(null)	(null)	8456
2	(null)	Pociag	1757
3	(null)	Statek	1749
4	(null)	Autokar	3108
5	(null)	Samolot	1842
6	Italy	(null)	22
7	Italy	Statek	22
8	Japan	(null)	796
9	Japan	Pociag	118
10	Japan	Statek	149
11	Japan	Autokar	333
12	Japan	Samolot	196
13	Brazil	(null)	1318

Opis zapytania: Preferencje przewozowe klientów 2

Kod zapytania bazy:

```
SELECT B_Adres.Kraj AS "Kraj", B_Rodzaj_Transportu.nazwa AS "Rodzaj
Transportu", B_Transport.Nazwa "Pojazd",
sum(B_Bilet_Rodzaj_Biletu.ILOSC) AS "Ilosc biletow"
FROM B_Impreza JOIN B_Bilet ON B_Bilet.id_impreza =
B_Impreza.id_impreza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
JOIN B_ADRES ON B_IMPENZA.id_adres = B_ADRES.id_adres
JOIN B_Transport ON B_Impreza.id_transport =
B_Transport.id_transport
JOIN B_Rodzaj_Transportu ON B_Transport.id_rodzaj_transportu =
B_Rodzaj_Transportu.id_rodzaj_transportu
group by cube(B_Adres.Kraj,
B_Rodzaj_Transportu.nazwa,B_Transport.Nazwa);
```

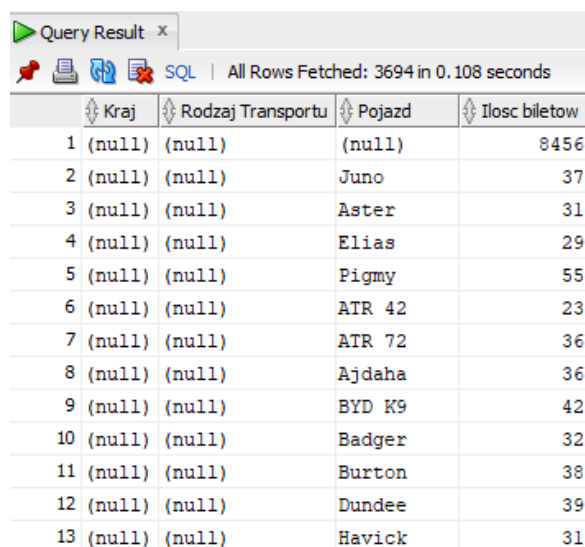
Kod zapytania hurtowni:

```
SELECT H_Adres.Kraj AS "Kraj", H_Transport.Rodzaj_Transportu AS
"Rodzaj Transportu", H_Transport.Nazwa "Pojazd",
sum(H_IMPENZA.ilosc) AS "Ilosc biletow"
FROM H_Impreza JOIN H_ADRES ON H_IMPENZA.id_adres = H_ADRES.id_adres
JOIN H_Transport ON H_Impreza.id_transport =
H_Transport.id_transport
group by cube(H_Adres.Kraj,
H_Transport.Rodzaj_Transportu,H_Transport.Nazwa);
```

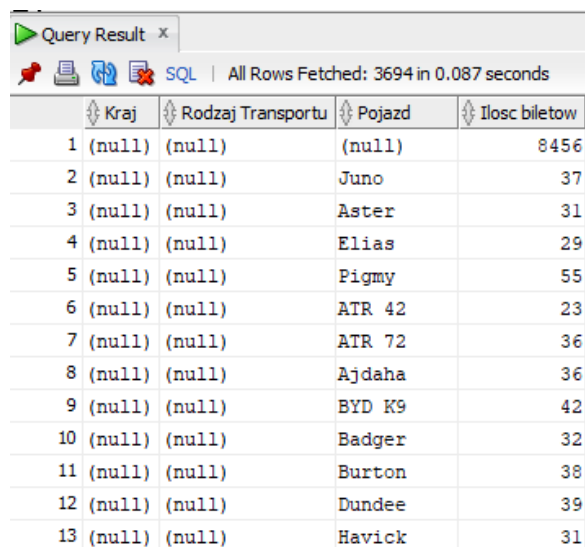
Czas dla bazy: 0.108s

Czas dla hurtowni: 0.087s

Zrzuty ekranów:



Kraj	Rodzaj Transportu	Pojazd	Ilosc biletow
(null)	(null)	(null)	8456
(null)	(null)	Juno	37
(null)	(null)	Aster	31
(null)	(null)	Elias	29
(null)	(null)	Pigmy	55
(null)	(null)	ATR 42	23
(null)	(null)	ATR 72	36
(null)	(null)	Ajdaha	36
(null)	(null)	BYD K9	42
(null)	(null)	Badger	32
(null)	(null)	Burton	38
(null)	(null)	Dundee	39
(null)	(null)	Havick	31



Kraj	Rodzaj Transportu	Pojazd	Ilosc biletow
(null)	(null)	(null)	8456
(null)	(null)	Juno	37
(null)	(null)	Aster	31
(null)	(null)	Elias	29
(null)	(null)	Pigmy	55
(null)	(null)	ATR 42	23
(null)	(null)	ATR 72	36
(null)	(null)	Ajdaha	36
(null)	(null)	BYD K9	42
(null)	(null)	Badger	32
(null)	(null)	Burton	38
(null)	(null)	Dundee	39
(null)	(null)	Havick	31

Opis zapytania: Który pojazd zarobił dla firmy najwięcej pieniędzy 1

Kod zapytania bazy:

```
SELECT B_Adres.Kraj AS "Kraj", B_Rodzaj_Transportu.nazwa AS "Rodzaj
Transportu", sum(B_Impreza.Cena_biletu*B_Bilet_Rodzaj_Biletu.ilosc)
AS "Suma kwota"
FROM B_Impreza JOIN B_ADRES ON B_IMPENZA.id_adres = B_ADRES.id_adres
JOIN B_Transport ON B_Impreza.id_transport =
B_Transport.id_transport
JOIN B_Rodzaj_Transportu ON B_Transport.id_rodzaj_transportu =
B_Rodzaj_Transportu.id_rodzaj_transportu
JOIN B_Bilet ON B_Bilet.id_impieza = B_IMPENZA.id_impieza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
group by cube(B_Adres.Kraj, B_Rodzaj_Transportu.nazwa);
```

Kod zapytania hurtowni:

```
SELECT H_Adres.Kraj AS "Kraj", H_Transport.Rodzaj_Transportu AS
"Rodzaj Transportu", sum(H_Impreza.Cena_biletu*H_IMPENZA.ilosc) AS
"Suma kwota"
FROM H_Impreza JOIN H_ADRES ON H_IMPENZA.id_adres = H_ADRES.id_adres
JOIN H_Transport ON H_Impreza.id_transport =
H_Transport.id_transport
group by cube(H_Adres.Kraj, H_Transport.Rodzaj_Transportu);
```

Czas dla bazy: 0.014s

Czas dla hurtowni: 0.005s

Zrzuty ekranów:

Query Result x			
SQL All Rows Fetched: 72 in 0.014 seconds			
	Kraj	Rodzaj Transportu	Suma kwota
1	(null)	(null)	3374963.9
2	(null)	Pociąg	208544.8
3	(null)	Statek	139551.3
4	(null)	Autokar	121661.4
5	(null)	Samolot	2905206.4
6	Italy	(null)	2004.1
7	Italy	Statek	2004.1
8	Japan	(null)	341582.6
9	Japan	Pociąg	13430.5
10	Japan	Statek	12982.7
11	Japan	Autokar	12238.2
12	Japan	Samolot	302931.2
13	Brazil	(null)	690309.6

Query Result x			
SQL All Rows Fetched: 72 in 0.005 seconds			
	Kraj	Rodzaj Transportu	Suma kwota
1	(null)	(null)	3374963.9
2	(null)	Pociąg	208544.8
3	(null)	Statek	139551.3
4	(null)	Autokar	121661.4
5	(null)	Samolot	2905206.4
6	Italy	(null)	2004.1
7	Italy	Statek	2004.1
8	Japan	(null)	341582.6
9	Japan	Pociąg	13430.5
10	Japan	Statek	12982.7
11	Japan	Autokar	12238.2
12	Japan	Samolot	302931.2
13	Brazil	(null)	690309.6

Opis zapytania: Który pojazd zarobił dla firmy najwięcej pieniędzy 2

Kod zapytania bazy:

```
SELECT B_Adres.Kraj AS "Kraj", B_Rodzaj_Transportu.nazwa AS "Rodzaj Transportu", B_Transport.Nazwa "Pojazd",  
sum(B_Impreza.Cena_biletu*B_Bilet_Rodzaj_Biletu.ilosc) AS "Suma kwota"  
FROM B_Impreza JOIN B_ADRES ON B_IMPENZA.id_adres = B_ADRES.id_adres  
JOIN B_Transport ON B_Impreza.id_transport =  
B_Transport.id_transport  
JOIN B_Rodzaj_Transportu ON B_Transport.id_rodzaj_transportu =  
B_Rodzaj_Transportu.id_rodzaj_transportu  
JOIN B_Bilet ON B_Bilet.id_imprensa = B_IMPENZA.id_imprensa  
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =  
B_Bilet.id_bilet  
group by cube(B_Adres.Kraj,  
B_Rodzaj_Transportu.nazwa,B_Transport.Nazwa);
```

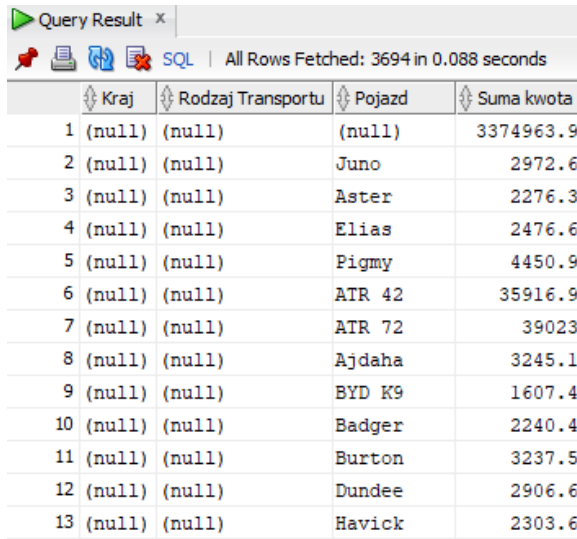
Kod zapytania hurtowni:

```
SELECT H_Adres.Kraj AS "Kraj", H_Transport.Rodzaj_Transportu AS  
"Rodzaj Transportu", H_Transport.Nazwa "Pojazd",  
sum(H_Impreza.Cena_biletu*H_IMPENZA.ilosc) AS "Suma kwota"  
FROM H_Impreza JOIN H_ADRES ON H_IMPENZA.id_adres = H_ADRES.id_adres  
JOIN H_Transport ON H_Impreza.id_transport =  
H_Transport.id_transport  
group by cube(H_Adres.Kraj,  
H_Transport.Rodzaj_Transportu,H_Transport.Nazwa);
```

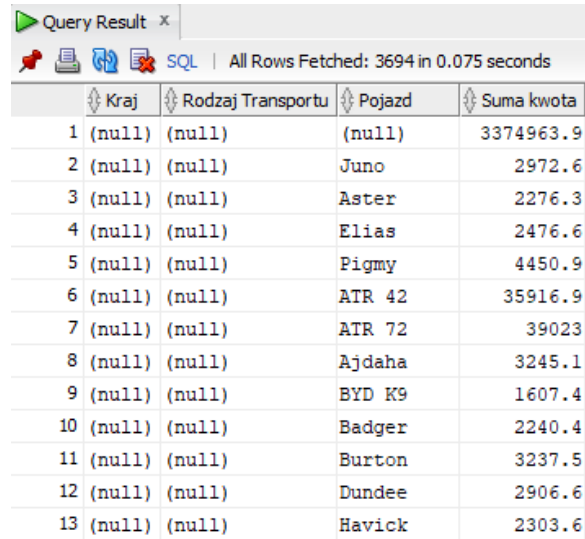

Czas dla bazy: 0.088s

Czas dla hurtowni: 0.075s

Zrzuty ekranów:



	Kraj	Rodzaj Transportu	Pojazd	Suma kwota
1	(null)	(null)	(null)	3374963.9
2	(null)	(null)	Juno	2972.6
3	(null)	(null)	Aster	2276.3
4	(null)	(null)	Elias	2476.6
5	(null)	(null)	Pigmy	4450.9
6	(null)	(null)	ATR 42	35916.9
7	(null)	(null)	ATR 72	39023
8	(null)	(null)	Ajdaha	3245.1
9	(null)	(null)	BYD K9	1607.4
10	(null)	(null)	Badger	2240.4
11	(null)	(null)	Burton	3237.5
12	(null)	(null)	Dundee	2906.6
13	(null)	(null)	Havick	2303.6



	Kraj	Rodzaj Transportu	Pojazd	Suma kwota
1	(null)	(null)	(null)	3374963.9
2	(null)	(null)	Juno	2972.6
3	(null)	(null)	Aster	2276.3
4	(null)	(null)	Elias	2476.6
5	(null)	(null)	Pigmy	4450.9
6	(null)	(null)	ATR 42	35916.9
7	(null)	(null)	ATR 72	39023
8	(null)	(null)	Ajdaha	3245.1
9	(null)	(null)	BYD K9	1607.4
10	(null)	(null)	Badger	2240.4
11	(null)	(null)	Burton	3237.5
12	(null)	(null)	Dundee	2906.6
13	(null)	(null)	Havick	2303.6

Czas wykonania zapytań partycji obliczeniowych:

Opis zapytania: % udziału rodzaju imprezy w danym kraju w odniesieniu do ogólnej kwoty dla całego kraju.

Kod zapytania bazy:

```
SELECT DISTINCT B_ADRES.KRAJ AS "Kraj", B_RODZAJ_IMPREZY.NAZWA AS  
"Rodzaj imprezy",  
SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) OVER  
(PARTITION BY B_ADRES.KRAJ,B_RODZAJ_IMPREZY.NAZWA) AS "Kwota dla  
rodzaju imprezy",  
SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) OVER  
(PARTITION BY B_ADRES.KRAJ) AS "Ogólna kwota dla kraju",  
ROUND(100*(SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ILOSC)OVE  
R (PARTITION BY B_ADRES.KRAJ,B_RODZAJ_IMPREZY.NAZWA))/  
(SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ILOSC)OVER  
(PARTITION BY B_ADRES.KRAJ)),5) "UDZIAL %" FROM B_IMPREZA JOIN  
B_ADRES ON B_IMPREZA.ID_ADRES = B_ADRES.ID_ADRES  
JOIN B_RODZAJ_IMPREZY ON B_IMPREZA.ID_RODZAJ_IMPREZY =  
B_RODZAJ_IMPREZY.ID_RODZAJ_IMPREZY  
JOIN B_Bilet ON B_Bilet.id_impreza = B_IMPREZA.id_impreza  
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =  
B_Bilet.id_bilet  
ORDER BY B_RODZAJ_IMPREZY.NAZWA, B_ADRES.KRAJ;
```

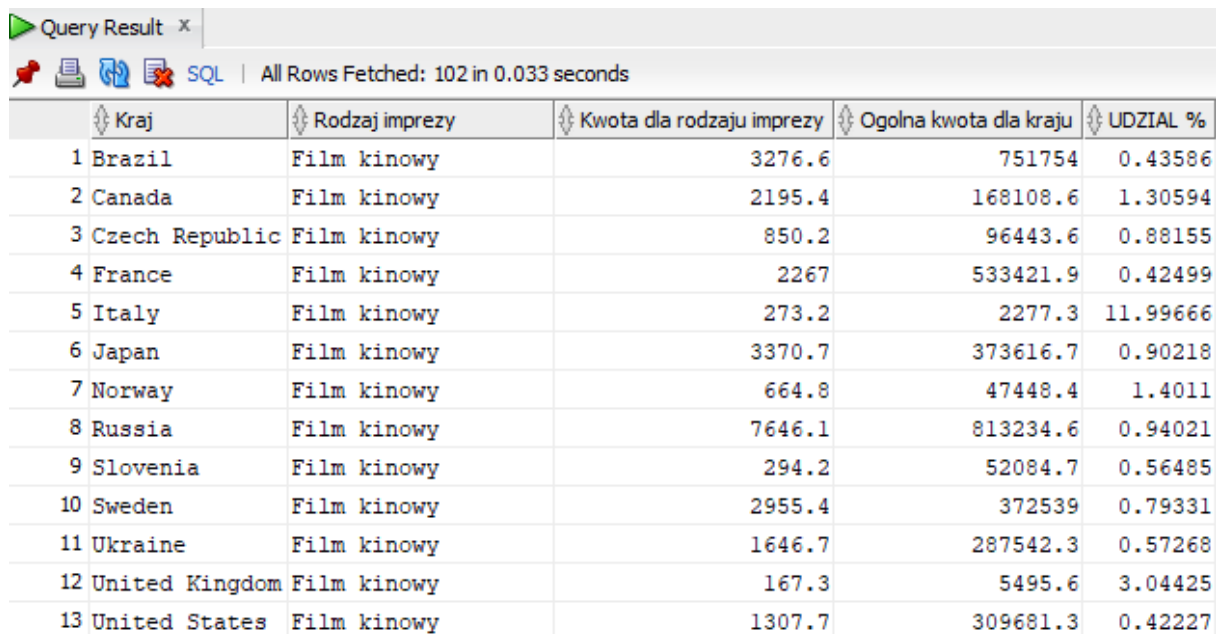
Kod zapytania hurtowni:

```
SELECT DISTINCT H_ADRES.KRAJ AS "Kraj", H_RODZAJ_IMPREZY.NAZWA AS  
"Rodzaj imprezy",  
SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) OVER (PARTITION BY  
H_ADRES.KRAJ,H_RODZAJ_IMPREZY.NAZWA) AS "Kwota dla rodzaju imprezy",  
SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) OVER (PARTITION BY  
H_ADRES.KRAJ) AS "Ogolna kwota dla kraju",  
ROUND(100*(SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc)OVER (PARTITION  
BY H_ADRES.KRAJ,H_RODZAJ_IMPREZY.NAZWA))/  
(SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc)OVER (PARTITION BY  
H_ADRES.KRAJ)),5) "UDZIAL %" FROM H_IMPREZA JOIN H_ADRES ON  
H_IMPREZA.ID_ADRES = H_ADRES.ID_ADRES  
JOIN H_RODZAJ_IMPREZY ON H_IMPREZA.ID_RODZAJ_IMPREZY =  
H_RODZAJ_IMPREZY.ID_RODZAJ_IMPREZY  
ORDER BY H_RODZAJ_IMPREZY.NAZWA, H_ADRES.KRAJ;
```

Czas dla bazy: 0.033s

Czas dla hurtowni: 0.021s

Zrzuty ekranów:



	Kraj	Rodzaj imprezy	Kwota dla rodzaju imprezy	Ogolna kwota dla kraju	UDZIAL %
1	Brazil	Film kinowy	3276.6	751754	0.43586
2	Canada	Film kinowy	2195.4	168108.6	1.30594
3	Czech Republic	Film kinowy	850.2	96443.6	0.88155
4	France	Film kinowy	2267	533421.9	0.42499
5	Italy	Film kinowy	273.2	2277.3	11.99666
6	Japan	Film kinowy	3370.7	373616.7	0.90218
7	Norway	Film kinowy	664.8	47448.4	1.4011
8	Russia	Film kinowy	7646.1	813234.6	0.94021
9	Slovenia	Film kinowy	294.2	52084.7	0.56485
10	Sweden	Film kinowy	2955.4	372539	0.79331
11	Ukraine	Film kinowy	1646.7	287542.3	0.57268
12	United Kingdom	Film kinowy	167.3	5495.6	3.04425
13	United States	Film kinowy	1307.7	309681.3	0.42227

Query Result x				
SQL All Rows Fetched: 102 in 0.021 seconds				
Kraj	Rodzaj imprezy	Kwota dla rodzaju imprezy	Ogolna kwota dla kraju	UDZIAL %
1 Brazil	Film kinowy	3276.6	751754	0.43586
2 Canada	Film kinowy	2195.4	168108.6	1.30594
3 Czech Republic	Film kinowy	850.2	96443.6	0.88155
4 France	Film kinowy	2267	533421.9	0.42499
5 Italy	Film kinowy	273.2	2277.3	11.99666
6 Japan	Film kinowy	3370.7	373616.7	0.90218
7 Norway	Film kinowy	664.8	47448.4	1.4011
8 Russia	Film kinowy	7646.1	813234.6	0.94021
9 Slovenia	Film kinowy	294.2	52084.7	0.56485
10 Sweden	Film kinowy	2955.4	372539	0.79331
11 Ukraine	Film kinowy	1646.7	287542.3	0.57268
12 United Kingdom	Film kinowy	167.3	5495.6	3.04425
13 United States	Film kinowy	1307.7	309681.3	0.42227

Opis zapytania: % udział krajów w zyskach ogólnych

Kod zapytania bazy:

```
SELECT DISTINCT B_ADRES.KRAJ AS "Kraj",
SUM(B_IMPENZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) OVER
(PARTITION BY B_ADRES.KRAJ) AS "Kwota dla kraju",
SUM(B_IMPENZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) OVER
(PARTITION BY NULL) AS "Ogolna kwota",
ROUND(100*(SUM(B_IMPENZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ILOSC)OVER
R (PARTITION BY B_ADRES.KRAJ))/
(SUM(B_IMPENZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ILOSC)OVER
(PARTITION BY NULL)),5) "UDZIAL %" FROM B_IMPENZA JOIN B_ADRES ON
B_IMPENZA.ID_ADRES = B_ADRES.ID_ADRES
JOIN B_Bilet ON B_Bilet.id_impreza = B_IMPENZA.id_impreza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
ORDER BY B_ADRES.KRAJ;
```

Kod zapytania hurtowni:

```
SELECT DISTINCT H_ADRES.KRAJ AS "Kraj",
SUM(H_IMPENZA.CENA_BILETU*H_IMPENZA.ilosc) OVER (PARTITION BY
H_ADRES.KRAJ) AS "Kwota dla kraju",
SUM(H_IMPENZA.CENA_BILETU*H_IMPENZA.ilosc) OVER (PARTITION BY NULL)
AS "Ogolna kwota",
ROUND(100*(SUM(H_IMPENZA.CENA_BILETU*H_IMPENZA.ilosc)OVER (PARTITION
BY H_ADRES.KRAJ))/
(SUM(H_IMPENZA.CENA_BILETU*H_IMPENZA.ilosc) OVER (PARTITION BY
NULL)),5) "UDZIAL %" FROM H_IMPENZA JOIN H_ADRES ON
```

H_IMPENZA.ID_ADRES = H_ADRES.ID_ADRES
ORDER BY H_ADRES.KRAJ;

Czas dla bazy: 0.026s

Czas dla hurtowni: 0.018s

Zrzuty ekranów:

Query Result x			
All Rows Fetched: 16 in 0.026 seconds			
Kraj	Kwota dla kraju	Ogólna kwota	UDZIAL %
1 Brazil	751754	3914865.9	19.20255
2 Canada	168108.6	3914865.9	4.29411
3 Czech Republic	96443.6	3914865.9	2.46352
4 France	533421.9	3914865.9	13.62555
5 Germany	98149.5	3914865.9	2.5071
6 Hong Kong	911.2	3914865.9	0.02328
7 Italy	2277.3	3914865.9	0.05817
8 Japan	373616.7	3914865.9	9.54354
9 Niger	2157.2	3914865.9	0.0551
10 Norway	47448.4	3914865.9	1.21201
11 Russia	813234.6	3914865.9	20.77299
12 Slovenia	52084.7	3914865.9	1.33043
13 Sweden	372539	3914865.9	9.51601

Query Result x			
All Rows Fetched: 16 in 0.018 seconds			
Kraj	Kwota dla kraju	Ogólna kwota	UDZIAL %
1 Brazil	751754	3914865.9	19.20255
2 Canada	168108.6	3914865.9	4.29411
3 Czech Republic	96443.6	3914865.9	2.46352
4 France	533421.9	3914865.9	13.62555
5 Germany	98149.5	3914865.9	2.5071
6 Hong Kong	911.2	3914865.9	0.02328
7 Italy	2277.3	3914865.9	0.05817
8 Japan	373616.7	3914865.9	9.54354
9 Niger	2157.2	3914865.9	0.0551
10 Norway	47448.4	3914865.9	1.21201
11 Russia	813234.6	3914865.9	20.77299
12 Slovenia	52084.7	3914865.9	1.33043
13 Sweden	372539	3914865.9	9.51601

Opis zapytania: % biletów każdego rodzaju w ogólnej sumie biletów

Kod zapytania bazy:

```
SELECT DISTINCT B_Rodzaj_Imprezy.Nazwa AS "Rodzaj",
B_Rodzaj_Biletu.Nazwa AS "Rodzaj biletu",
SUM(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY
B_Rodzaj_Biletu.Nazwa) AS "Ilosc biletow danego rodzjau",
sum(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY NULL) "Ilosc
biletow ogolna",
ROUND(100*SUM(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY
B_Rodzaj_Biletu.Nazwa)/
(SUM(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY NULL)),2)
"UDZIAL %"
FROM B_Impreza JOIN B_Rodzaj_Imprezy ON B_Impreza.id_rodzaj_impreny
= B_Rodzaj_Imprezy.id_rodzaj_impreny
JOIN B_Bilet ON B_Bilet.id_impreny = B_Impreza.id_impreny
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
JOIN B_Rodzaj_biletu ON B_Rodzaj_biletu.id_rodzaj_biletu =
B_Bilet_Rodzaj_Biletu.id_rodzaj_biletu
ORDER BY B_Rodzaj_Imprezy.Nazwa,B_Rodzaj_Biletu.Nazwa;
```

Kod zapytania hurtowni:

```
SELECT DISTINCT H_Rodzaj_Imprezy.Nazwa AS "Rodzaj",
H_Rodzaj_Biletu.Nazwa AS "Rodzaj biletu", SUM(H_IMPENZA.ilosc) OVER
(PARTITION BY H_Rodzaj_Biletu.Nazwa) AS "Ilosc biletow danego
rodzjau",
```

```

sum(H_IMPREZA.ilosc) OVER (PARTITION BY NULL) "Ilosc biletow
ogolna",
ROUND(100*SUM(H_IMPREZA.ilosc) OVER (PARTITION BY
H_Rodzaj_Biletu.Nazwa)/
(SUM(H_IMPREZA.ilosc) OVER (PARTITION BY NULL)),2) "UDZIAL %"
FROM H_Impreza JOIN H_Rodzaj_Imprezy ON H_Impreza.id_rodzaj_imprezy
= H_Rodzaj_Imprezy.id_rodzaj_imprezy
JOIN H_Rodzaj_biletu ON H_Rodzaj_biletu.id_rodzaj_biletu =
H_IMPREZA.id_rodzaj_biletu
ORDER BY H_Rodzaj_Imprezy.Nazwa,H_Rodzaj_Biletu.Nazwa;

```

Czas dla bazy: 0.024s

Czas dla hurtowni: 0.015s

Zrzuty ekranów:

Query Result x

SQL | All Rows Fetched: 42 in 0.024 seconds

Rodzaj	Rodzaj biletu	Ilosc biletow danego rodzajau	Ilosc biletow ogolna	UDZIAL %
1 Film kinowy	Bilet grupowy	206	13978	1.47
2 Film kinowy	Bilet normalny	174	13978	1.24
3 Film kinowy	Bilet ulgowy	4494	13978	32.15
4 Komunikacja miejska	Bilet autokarowy grupowy	102	13978	0.73
5 Komunikacja miejska	Bilet autokarowy normalny	63	13978	0.45
6 Komunikacja miejska	Bilet autokarowy ulgowy	2024	13978	14.48
7 Komunikacja miejska	Bilet rejsowy grupowy	14	13978	0.1
8 Komunikacja miejska	Bilet rejsowy ulgowy	1588	13978	11.36
9 Komunikacja miejska	Bilet tramwajowy grupowy	94	13978	0.67
10 Komunikacja miejska	Bilet tramwajowy normalny 3 dniowy	24	13978	0.17
11 Komunikacja miejska	Bilet tramwajowy normalny dobowy	14	13978	0.1
12 Komunikacja miejska	Bilet tramwajowy normalny jednorazowy	26	13978	0.19
13 Komunikacja miejska	Bilet tramwajowy normalny weekendowy	13	13978	0.09

Query Result x

SQL | All Rows Fetched: 42 in 0.015 seconds

Rodzaj	Rodzaj biletu	Ilosc biletow danego rodzajau	Ilosc biletow ogolna	UDZIAL %
1 Film kinowy	Bilet grupowy	206	13978	1.47
2 Film kinowy	Bilet normalny	174	13978	1.24
3 Film kinowy	Bilet ulgowy	4494	13978	32.15
4 Komunikacja miejska	Bilet autokarowy grupowy	102	13978	0.73
5 Komunikacja miejska	Bilet autokarowy normalny	63	13978	0.45
6 Komunikacja miejska	Bilet autokarowy ulgowy	2024	13978	14.48
7 Komunikacja miejska	Bilet rejsowy grupowy	14	13978	0.1
8 Komunikacja miejska	Bilet rejsowy ulgowy	1588	13978	11.36
9 Komunikacja miejska	Bilet tramwajowy grupowy	94	13978	0.67
10 Komunikacja miejska	Bilet tramwajowy normalny 3 dniowy	24	13978	0.17
11 Komunikacja miejska	Bilet tramwajowy normalny dobowy	14	13978	0.1
12 Komunikacja miejska	Bilet tramwajowy normalny jednorazowy	26	13978	0.19
13 Komunikacja miejska	Bilet tramwajowy normalny weekendowy	13	13978	0.09

Opis zapytania: % udział danego rodzaju biletu do ogólnej ilości biletów w kraju

Kod zapytania bazy:

```
SELECT DISTINCT B_ADRES.KRAJ AS "Kraj", B_Rodzaj_Biletu.Nazwa AS
"Rodzaj biletu", SUM(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY
B_ADRES.KRAJ,B_Rodzaj_Biletu.Nazwa) AS "ILOSC",
sum(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY B_ADRES.KRAJ)
Ilosc_biletow,
ROUND(100*SUM(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY
B_ADRES.KRAJ,B_Rodzaj_Biletu.Nazwa)/
(SUM(B_Bilet_Rodzaj_Biletu.ILOSC) OVER (PARTITION BY
B_ADRES.KRAJ)),5) "UDZIAL %"
FROM B_Impreza JOIN B_Bilet ON B_Bilet.id_impreza =
B_Impreza.id_impreza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
JOIN B_Rodzaj_biletu ON B_Rodzaj_biletu.id_rodzaj_biletu =
B_Bilet_Rodzaj_Biletu.id_rodzaj_biletu
JOIN B_ADRES ON B_IMPENZA.ID_ADRES = B_ADRES.ID_ADRES
ORDER BY B_ADRES.KRAJ, B_Rodzaj_Biletu.Nazwa;
```

Kod zapytania hurtowni:

```
SELECT DISTINCT H_ADRES.KRAJ AS "Kraj", H_Rodzaj_Biletu.Nazwa AS
"Rodzaj biletu", SUM(H_IMPENZA.ilosc) OVER (PARTITION BY
H_ADRES.KRAJ,H_Rodzaj_Biletu.Nazwa) AS "ILOSC", sum(H_IMPENZA.ilosc)
OVER (PARTITION BY H_ADRES.KRAJ) Ilosc_biletow,
ROUND(100*SUM(H_IMPENZA.ilosc) OVER (PARTITION BY
H_ADRES.KRAJ,H_Rodzaj_Biletu.Nazwa)/
(SUM(H_IMPENZA.ilosc) OVER (PARTITION BY H_ADRES.KRAJ)),5) "UDZIAL
%"
FROM H_Impreza
JOIN H_Rodzaj_biletu ON H_Rodzaj_biletu.id_rodzaj_biletu =
H_IMPENZA.id_rodzaj_biletu
JOIN H_ADRES ON H_IMPENZA.ID_ADRES = H_ADRES.ID_ADRES
ORDER BY H_ADRES.KRAJ, H_Rodzaj_Biletu.Nazwa;
```

Czas dla bazy: 0.027s

Czas dla hurtowni: 0.014s

Zrzuty ekranów:

Query Result x

SQL | All Rows Fetched: 296 in 0.027 seconds

	Kraj	Rodzaj biletu	ILOSC	ILOSC_BILETOW	UDZIAL %
1	Brazil	Bilet autokarowy grupowy	12	2011	0.59672
2	Brazil	Bilet autokarowy normalny	9	2011	0.44754
3	Brazil	Bilet autokarowy okresowy	11	2011	0.54699
4	Brazil	Bilet autokarowy ulgowy	338	2011	16.80756
5	Brazil	Bilet grupowy	22	2011	1.09398
6	Brazil	Bilet kolejowy normalny	14	2011	0.69617
7	Brazil	Bilet kolejowy ulgowy	206	2011	10.24366
8	Brazil	Bilet lotniczy grupowy	7	2011	0.34809
9	Brazil	Bilet lotniczy normalny	15	2011	0.7459
10	Brazil	Bilet lotniczy ulgowy	308	2011	15.31576
11	Brazil	Bilet normalny	21	2011	1.04426
12	Brazil	Bilet przewozowy grupowy	5	2011	0.24863
13	Brazil	Bilet przewozowy ulgowy	8	2011	0.39781

Query Result x

SQL | All Rows Fetched: 296 in 0.014 seconds

	Kraj	Rodzaj biletu	ILOSC	ILOSC_BILETOW	UDZIAL %
1	Brazil	Bilet autokarowy grupowy	12	2011	0.59672
2	Brazil	Bilet autokarowy normalny	9	2011	0.44754
3	Brazil	Bilet autokarowy okresowy	11	2011	0.54699
4	Brazil	Bilet autokarowy ulgowy	338	2011	16.80756
5	Brazil	Bilet grupowy	22	2011	1.09398
6	Brazil	Bilet kolejowy normalny	14	2011	0.69617
7	Brazil	Bilet kolejowy ulgowy	206	2011	10.24366
8	Brazil	Bilet lotniczy grupowy	7	2011	0.34809
9	Brazil	Bilet lotniczy normalny	15	2011	0.7459
10	Brazil	Bilet lotniczy ulgowy	308	2011	15.31576
11	Brazil	Bilet normalny	21	2011	1.04426
12	Brazil	Bilet przewozowy grupowy	5	2011	0.24863
13	Brazil	Bilet przewozowy ulgowy	8	2011	0.39781

Opis zapytania: % udział każdego tytułu spektaklu w ogólnych zyskach z danego rodzaju imprezy

Kod zapytania bazy:

```
SELECT DISTINCT B_RODZAJ_IMPREZY.NAZWA AS "Rodzaj imprezy",
B_SPEKTAKL.TYTUL AS "Tytuł",
SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) OVER
(PARTITION BY B_SPEKTAKL.TYTUL) AS "Kwota",
SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ilosc) OVER
(PARTITION BY B_RODZAJ_IMPREZY.NAZWA) SUMA_KWOTA,
ROUND(100*(SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ILOSC)OVER
R (PARTITION BY B_SPEKTAKL.TYTUL)))/
(SUM(B_IMPREZA.CENA_BILETU*B_Bilet_Rodzaj_Biletu.ILOSC) OVER
(PARTITION BY B_RODZAJ_IMPREZY.NAZWA)),5) "UDZIAL %"
FROM B_IMPREZA JOIN B_SPEKTAKL ON B_IMPREZA.ID_SPEKTAKL =
B_SPEKTAKL.ID_SPEKTAKL
JOIN B_RODZAJ_IMPREZY ON B_IMPREZA.ID_RODZAJ_IMPREZY =
B_RODZAJ_IMPREZY.ID_RODZAJ_IMPREZY
JOIN B_Bilet ON B_Bilet.id_impreza = B_IMPREZA.id_impreza
JOIN B_Bilet_Rodzaj_Biletu ON B_Bilet_Rodzaj_Biletu.id_bilet =
B_Bilet.id_bilet
ORDER BY B_RODZAJ_IMPREZY.NAZWA, B_SPEKTAKL.TYTUL;
```

Kod zapytania hurtowni:

```
SELECT DISTINCT H_RODZAJ_IMPREZY.NAZWA AS "Rodzaj imprezy",
H_SPEKTAKL.TYTUL AS "Tytuł",
SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) OVER (PARTITION BY
H_SPEKTAKL.TYTUL) AS "Kwota",
SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) OVER (PARTITION BY
H_RODZAJ_IMPREZY.NAZWA) SUMA_KWOTA,
ROUND(100*(SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc)OVER (PARTITION
BY H_SPEKTAKL.TYTUL)))/
(SUM(H_IMPREZA.CENA_BILETU*H_IMPREZA.ilosc) OVER (PARTITION BY
H_RODZAJ_IMPREZY.NAZWA)),5) "UDZIAL %"
FROM H_IMPREZA JOIN H_SPEKTAKL ON H_IMPREZA.ID_SPEKTAKL =
H_SPEKTAKL.ID_SPEKTAKL
JOIN H_RODZAJ_IMPREZY ON H_IMPREZA.ID_RODZAJ_IMPREZY =
H_RODZAJ_IMPREZY.ID_RODZAJ_IMPREZY
ORDER BY H_RODZAJ_IMPREZY.NAZWA, H_SPEKTAKL.TYTUL;
```

Czas dla bazy: 0.044s

Czas dla hurtowni: 0.038s

Zrzuty ekranów:

Query Result x

SQL | All Rows Fetched: 1063 in 0.044 seconds

Rodzaj imprezy	Tytuł	Kwota	SUMA_KWOTA	UDZIAL %
1 Film kinowy	10.000 Km (Long Distance)	15.7	26915.3	0.05833
2 Film kinowy	102 Minutes That Changed America	39.4	26915.3	0.14639
3 Film kinowy	11 09 01 - September 11	321.2	26915.3	1.19337
4 Film kinowy	15	679.4	26915.3	2.52421
5 Film kinowy	1981	141.7	26915.3	0.52647
6 Film kinowy	25th Hour	296.7	26915.3	1.10235
7 Film kinowy	35 and Ticking	255.1	26915.3	0.94779
8 Film kinowy	49 Up	188.2	26915.3	0.69923
9 Film kinowy	6 Souls (Shelter)	261.4	26915.3	0.97119
10 Film kinowy	7 Boxes (7 cajas)	312.9	26915.3	1.16254
11 Film kinowy	9 Month Stretch (9 mois ferme)	58.2	26915.3	0.21623
12 Film kinowy	Aakrosh	350.5	26915.3	1.30223
13 Film kinowy	Absolute Giganten	90.5	26915.3	0.33624

Query Result x

SQL | All Rows Fetched: 1063 in 0.038 seconds

Rodzaj imprezy	Tytuł	Kwota	SUMA_KWOTA	UDZIAL %
1 Film kinowy	10.000 Km (Long Distance)	15.7	26915.3	0.05833
2 Film kinowy	102 Minutes That Changed America	39.4	26915.3	0.14639
3 Film kinowy	11 09 01 - September 11	321.2	26915.3	1.19337
4 Film kinowy	15	679.4	26915.3	2.52421
5 Film kinowy	1981	141.7	26915.3	0.52647
6 Film kinowy	25th Hour	296.7	26915.3	1.10235
7 Film kinowy	35 and Ticking	255.1	26915.3	0.94779
8 Film kinowy	49 Up	188.2	26915.3	0.69923
9 Film kinowy	6 Souls (Shelter)	261.4	26915.3	0.97119
10 Film kinowy	7 Boxes (7 cajas)	312.9	26915.3	1.16254
11 Film kinowy	9 Month Stretch (9 mois ferme)	58.2	26915.3	0.21623
12 Film kinowy	Aakrosh	350.5	26915.3	1.30223
13 Film kinowy	Absolute Giganten	90.5	26915.3	0.33624

Czas wykonania zapytań okien czasowych:

Opis zapytania: Porównanie przychodów imprez na przestrzeni roku.

Kod zapytania bazy:

```
SELECT DISTINCT ri.nazwa,  
extract(year from data_roz poczeczia) AS "ROK",  
extract(month from data_roz poczeczia) AS "MIESIAC",  
sum(i.cena_biletu * brb.ilosc) over (partition by  
ri.id_rodzaj_imprezy order by extract(month from data_roz poczeczia)  
range between UNBOUNDED PRECEDING AND CURRENT ROW) AS "SUMA"  
FROM B_Impreza i JOIN B_rodzaj_imprezy ri ON i.id_rodzaj_imprezy =  
ri.id_rodzaj_imprezy  
JOIN B_bilet b ON b.id_impreza = i.id_impreza  
JOIN B_bilet_rodzaj_biletu brb ON brb.id_bilet = b.id_bilet  
ORDER BY ri.nazwa, extract(month from data_roz poczeczia);
```

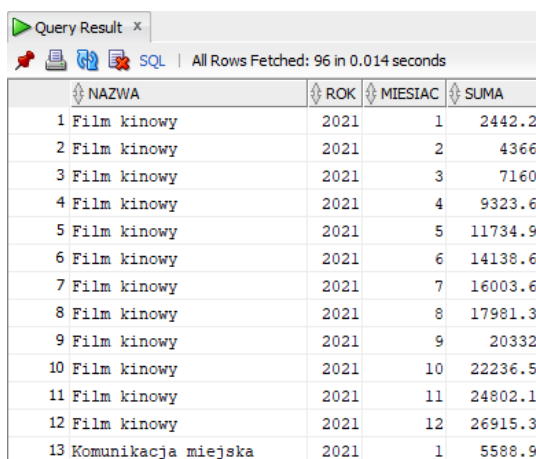
Kod zapytania hurtowni:

```
SELECT DISTINCT  
ri.nazwa,  
d.ROK,  
d.MIESIAC,  
sum(i.cena_biletu * i.ilosc) over (partition by ri.id_rodzaj_imprezy  
order by d.miesiac range between UNBOUNDED PRECEDING AND CURRENT  
ROW) AS "SUMA"  
FROM H_Impreza i JOIN H_rodzaj_imprezy ri ON i.id_rodzaj_imprezy =  
ri.id_rodzaj_imprezy  
JOIN H_DATA d ON i.ID_DATA = d.ID_DATA  
ORDER BY d.ROK, ri.nazwa, d.MIESIAC;
```

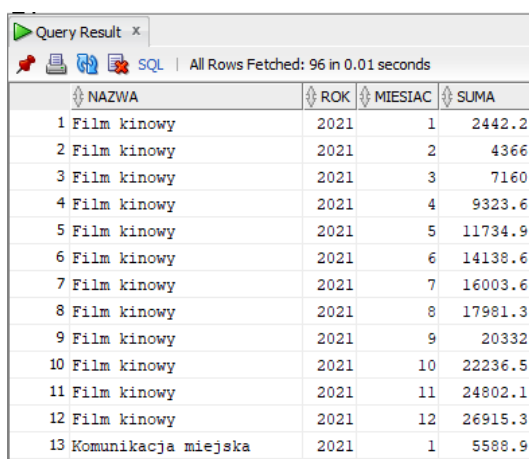
Czas dla bazy: 0.014s

Czas dla hurtowni: 0.01s

Zrzuty ekranów:



NAZWA	ROK	MIESIAC	SUMA
1 Film kinowy	2021	1	2442.2
2 Film kinowy	2021	2	4366
3 Film kinowy	2021	3	7160
4 Film kinowy	2021	4	9323.6
5 Film kinowy	2021	5	11734.9
6 Film kinowy	2021	6	14138.6
7 Film kinowy	2021	7	16003.6
8 Film kinowy	2021	8	17981.3
9 Film kinowy	2021	9	20332
10 Film kinowy	2021	10	22236.5
11 Film kinowy	2021	11	24802.1
12 Film kinowy	2021	12	26915.3
13 Komunikacja miejska	2021	1	5588.9



NAZWA	ROK	MIESIAC	SUMA
1 Film kinowy	2021	1	2442.2
2 Film kinowy	2021	2	4366
3 Film kinowy	2021	3	7160
4 Film kinowy	2021	4	9323.6
5 Film kinowy	2021	5	11734.9
6 Film kinowy	2021	6	14138.6
7 Film kinowy	2021	7	16003.6
8 Film kinowy	2021	8	17981.3
9 Film kinowy	2021	9	20332
10 Film kinowy	2021	10	22236.5
11 Film kinowy	2021	11	24802.1
12 Film kinowy	2021	12	26915.3
13 Komunikacja miejska	2021	1	5588.9

Opis zapytania: Zyski za wszystkie sprzedane bilety w danym miesiacu i wzrost tej kwoty w porownaniu do poprzedniego miesiaca

Kod zapytania bazy:

```
SELECT DISTINCT
extract(year from data_rozpoczecia) ROK,
extract(month from data_rozpoczecia) MIESIAC,
sum(i.cena_biletu * brb.ilosc) KWOTA,
sum(i.cena_biletu * brb.ilosc) - lag( (sum(i.cena_biletu *
brb.ilosc)), 1) over (order by
extract(month from data_rozpoczecia)) WZROST
FROM B_Impreza i
JOIN B_bilet b ON b.id_impieza = i.id_impieza
JOIN B_bilet_rodzaj_biletu brb ON brb.id_bilet = b.id_bilet
GROUP BY extract(year from data_rozpoczecia), extract(month from
data_rozpoczecia);
```

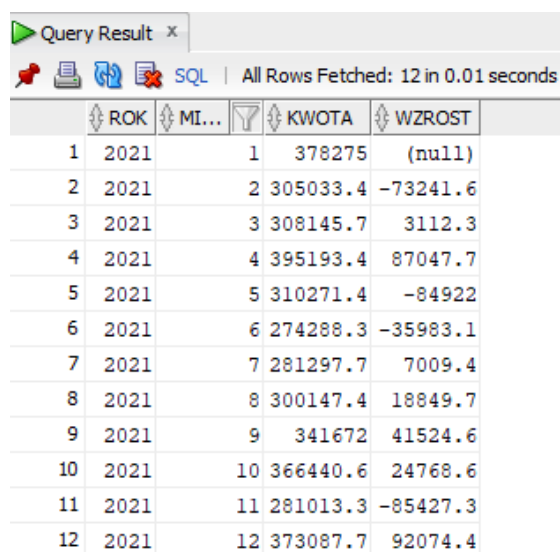
Kod zapytania hurtowni:

```
SELECT DISTINCT
H_DATA.ROK AS "ROK",
H_DATA.MIESIAC AS "MIESIAC",
sum(i.cena_biletu * i.ilosc) KWOTA,
sum(i.cena_biletu * i.ilosc) - lag( (sum(i.cena_biletu * i.ilosc)),
1) over (order by H_DATA.MIESIAC) AS "WZROST"
FROM H_Impreza i
JOIN H_DATA ON i.ID_DATA = H_DATA.ID_DATA
GROUP BY H_DATA.ROK, H_DATA.MIESIAC;
```

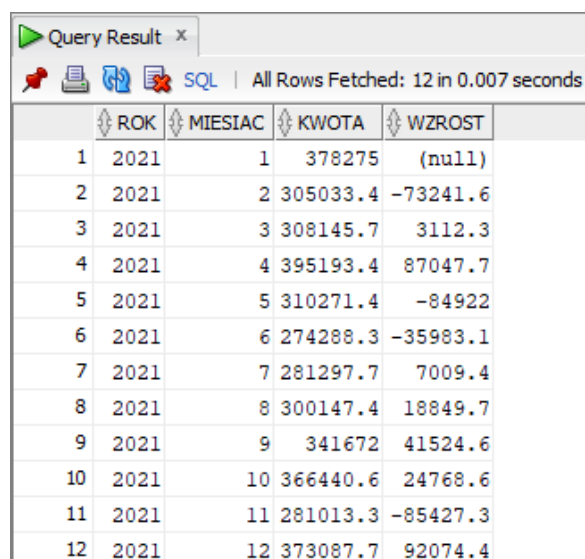
Czas dla bazy: 0.01s

Czas dla hurtowni: 0.007s

Zrzuty ekranów:



	ROK	MIESIAC	KWOTA	WZROST
1	2021	1	378275	(null)
2	2021	2	305033.4	-73241.6
3	2021	3	308145.7	3112.3
4	2021	4	395193.4	87047.7
5	2021	5	310271.4	-84922
6	2021	6	274288.3	-35983.1
7	2021	7	281297.7	7009.4
8	2021	8	300147.4	18849.7
9	2021	9	341672	41524.6
10	2021	10	366440.6	24768.6
11	2021	11	281013.3	-85427.3
12	2021	12	373087.7	92074.4



	ROK	MIESIAC	KWOTA	WZROST
1	2021	1	378275	(null)
2	2021	2	305033.4	-73241.6
3	2021	3	308145.7	3112.3
4	2021	4	395193.4	87047.7
5	2021	5	310271.4	-84922
6	2021	6	274288.3	-35983.1
7	2021	7	281297.7	7009.4
8	2021	8	300147.4	18849.7
9	2021	9	341672	41524.6
10	2021	10	366440.6	24768.6
11	2021	11	281013.3	-85427.3
12	2021	12	373087.7	92074.4

Opis zapytania: Porównanie ilości kupionych biletów na wycieczki w każdym kwartale

Kod zapytania bazy:

```
SELECT DISTINCT ri.nazwa,  
extract(month from data_roz poczeczia) AS "MIESIAC",  
FLOOR((extract(month from data_roz poczeczia) + 2) / 3) AS "KWARTAL",  
sum(i.cena_biletu * brb.ilosc) over (partition by  
FLOOR((extract(month from data_roz poczeczia) + 2) / 3) order by  
extract(month from data_roz poczeczia) range between UNBOUNDED  
PRECEDING AND CURRENT ROW) AS "SUMA"  
FROM B_Impreza i JOIN B_rodzaj_imp rezy ri ON i.id_rodzaj_imp rezy =  
ri.id_rodzaj_imp rezy  
JOIN B_bilet b ON b.id_imp reza = i.id_imp reza  
JOIN B_bilet_rodzaj_biletu brb ON brb.id_bilet = b.id_bilet  
WHERE ri.id_rodzaj_imp rezy = 6  
ORDER BY ri.nazwa, extract(month from data_roz poczeczia);
```

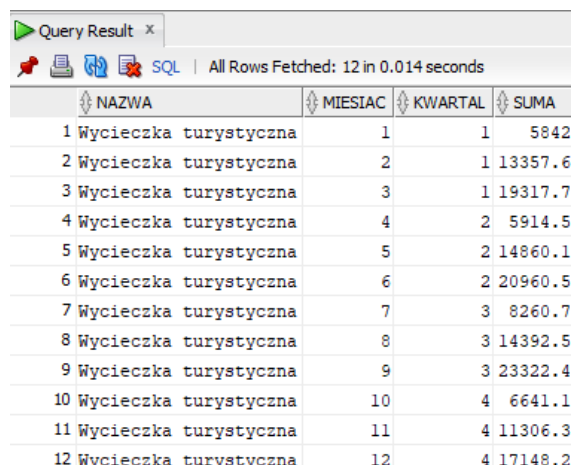
Kod zapytania hurtowni:

```
SELECT DISTINCT ri.nazwa,  
H_DATA.MIESIAC AS "MIESIAC",  
FLOOR((H_DATA.MIESIAC + 2) / 3) AS "KWARTAL",  
sum(i.cena_biletu * i.ilosc) over (partition by  
FLOOR(((H_DATA.MIESIAC + 2) / 3)) order by H_DATA.MIESIAC range  
between UNBOUNDED PRECEDING AND CURRENT ROW) AS "SUMA"  
FROM H_Impreza i JOIN H_rodzaj_imp rezy ri ON i.id_rodzaj_imp rezy =  
ri.id_rodzaj_imp rezy  
JOIN H_DATA ON i.ID_DATA = H_DATA.ID_DATA  
WHERE ri.id_rodzaj_imp rezy = 6  
ORDER BY ri.nazwa, H_DATA.MIESIAC;
```

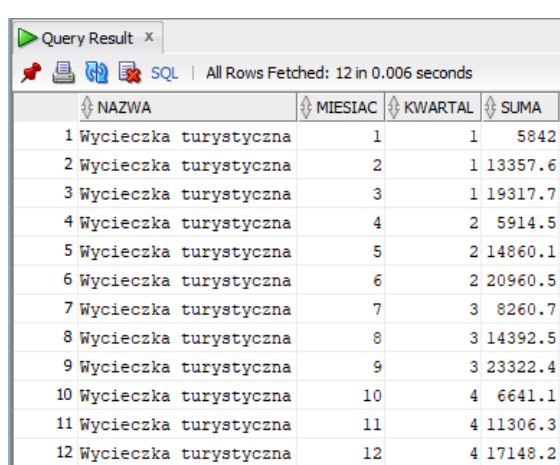
Czas dla bazy: 0.014s

Czas dla hurtowni: 0.006s

Zrzuty ekranów:



NAZWA	MIESIAC	KWARTAL	SUMA
1 Wycieczka turystyczna	1	1	5842
2 Wycieczka turystyczna	2	1	13357.6
3 Wycieczka turystyczna	3	1	19317.7
4 Wycieczka turystyczna	4	2	5914.5
5 Wycieczka turystyczna	5	2	14860.1
6 Wycieczka turystyczna	6	2	20960.5
7 Wycieczka turystyczna	7	3	8260.7
8 Wycieczka turystyczna	8	3	14392.5
9 Wycieczka turystyczna	9	3	23322.4
10 Wycieczka turystyczna	10	4	6641.1
11 Wycieczka turystyczna	11	4	11306.3
12 Wycieczka turystyczna	12	4	17148.2



NAZWA	MIESIAC	KWARTAL	SUMA
1 Wycieczka turystyczna	1	1	5842
2 Wycieczka turystyczna	2	1	13357.6
3 Wycieczka turystyczna	3	1	19317.7
4 Wycieczka turystyczna	4	2	5914.5
5 Wycieczka turystyczna	5	2	14860.1
6 Wycieczka turystyczna	6	2	20960.5
7 Wycieczka turystyczna	7	3	8260.7
8 Wycieczka turystyczna	8	3	14392.5
9 Wycieczka turystyczna	9	3	23322.4
10 Wycieczka turystyczna	10	4	6641.1
11 Wycieczka turystyczna	11	4	11306.3
12 Wycieczka turystyczna	12	4	17148.2

Czas wykonania zapytań funkcji rankingowych:

Opis zapytania: Ranking przedstawiający najczęściej wybierany transport ogólnie.

Kod zapytania bazy:

```
SELECT t.nazwa, count(b.id_bilet) AS "ILOSC BILETOW", DENSE_RANK()  
OVER (ORDER BY count(b.id_bilet) desc) AS "RANKING"  
FROM B_impreza i  
JOIN B_transport t ON i.id_transport = t.id_transport  
JOIN B_bilet b ON b.id_impreza = i.id_impreza  
GROUP BY t.nazwa;
```

Kod zapytania hurtowni:

```
SELECT t.nazwa, count(t.id_transport) AS "ILOSC BILETOW",  
DENSE_RANK() OVER (ORDER BY count(t.id_transport) desc) AS "RANKING"  
FROM H_impreza i  
JOIN H_transport t ON i.id_transport = t.id_transport  
GROUP BY t.nazwa;
```

Czas dla bazy: 0.014s

Czas dla hurtowni: 0.01s

Zrzuty ekranów:

Query Result x		
All Rows Fetched: 199 in 0.014 seconds		
NAZWA	ILOSC BILETOW	RANKING
1 Autosan H10	75	1
2 Neoplan N4009	66	2
3 Solaris Interlink	66	2
4 Wrightbus Pulsar Gemini	66	2
5 Eurabus 2.0	62	3
6 Volvo 7700	60	4
7 Solaris Trolilno	54	5
8 BOVA FUTURA 2009	53	6
9 Solaris Valletta	53	6
10 Volvo 8700	53	6
11 Jelcz M125M	51	7
12 Solaris Vacanza 12	51	7
13 MERCEDES 0818 TEAMSTAR 2005	48	8

Query Result x		
All Rows Fetched: 199 in 0.01 seconds		
NAZWA	ILOSC BILETOW	RANKING
1 Autosan H10	75	1
2 Wrightbus Pulsar Gemini	66	2
3 Solaris Interlink	66	2
4 Neoplan N4009	66	2
5 Eurabus 2.0	62	3
6 Volvo 7700	60	4
7 Solaris Trolilno	54	5
8 Solaris Valletta	53	6
9 Volvo 8700	53	6
10 BOVA FUTURA 2009	53	6
11 Jelcz M125M	51	7
12 Solaris Vacanza 12	51	7
13 Volvo Super Olympian	48	8

Opis zapytania: Ranking przedstawiający najczęściej wybierany transport w danym kraju.

Kod zapytania bazy:

```
SELECT a.kraj, t.nazwa, count(b.id_bilet) AS "ILOSC BILETOW",  
DENSE_RANK() OVER (PARTITION BY a.kraj ORDER BY count(b.id_bilet)  
desc) AS "RANKING"  
FROM B_impreza i  
JOIN B_transport t ON i.id_transport = t.id_transport  
JOIN B_adres a ON i.id_adres = a.id_adres  
JOIN B_bilet b ON b.id_impreza = i.id_impreza  
GROUP BY a.kraj, t.nazwa;
```

Kod zapytania hurtowni:

```
SELECT a.kraj, t.nazwa, count(t.id_transport) AS "ILOSC BILETOW",  
DENSE_RANK() OVER (PARTITION BY a.kraj ORDER BY  
count(t.id_transport) desc) AS "RANKING"  
FROM H_impreza i  
JOIN H_transport t ON i.id_transport = t.id_transport  
JOIN H_adres a ON i.id_adres = a.id_adres  
GROUP BY a.kraj, t.nazwa;
```

Czas dla bazy: 0.041s

Czas dla hurtowni: 0.03s

Zrzuty ekranów:

Query Result x			
All Rows Fetched: 1612 in 0.041 seconds			
KRAJ	NAZWA	ILOSC BILETOW	RANKING
1 Brazil	BOVA FUTURA 2009	16	1
2 Brazil	VW T6 CARAVELLE	14	2
3 Brazil	MERCEDES VITO TOURER VIP CLASS	13	3
4 Brazil	Airbus A320	13	3
5 Brazil	Nissan UA	12	4
6 Brazil	Autosan H10	12	4
7 Brazil	Indonesian Aerospace N-219	12	4
8 Brazil	Airbus A310	11	5
9 Brazil	BOVA FUTURA 2007	11	5
10 Brazil	Songbird Tracks	11	5
11 Brazil	Volvo 9700	11	5
12 Brazil	Volvo 7700	10	6
13 Brazil	Wrightbus Pulsar Gemini	10	6

Query Result x			
All Rows Fetched: 1612 in 0.03 seconds			
KRAJ	NAZWA	ILOSC BILETOW	RANKING
1 Brazil	BOVA FUTURA 2009	16	1
2 Brazil	VW T6 CARAVELLE	14	2
3 Brazil	Airbus A320	13	3
4 Brazil	MERCEDES VITO TOURER VIP CLASS	13	3
5 Brazil	Autosan H10	12	4
6 Brazil	Nissan UA	12	4
7 Brazil	Indonesian Aerospace N-219	12	4
8 Brazil	BOVA FUTURA 2007	11	5
9 Brazil	Songbird Tracks	11	5
10 Brazil	Airbus A310	11	5
11 Brazil	Volvo 9700	11	5
12 Brazil	Wrightbus Pulsar Gemini	10	6
13 Brazil	Neoplan N4009	10	6

Opis zapytania: Ranking przedstawiający, która ulga jest najczęściej wykorzystywana

Kod zapytania bazy:

```
SELECT ru.nazwa as "NAZWA ULGI", count(ru.id_rodzaj_ulgi) AS  
"ILOSC",  
DENSE_RANK() OVER (ORDER BY(count(ru.id_rodzaj_ulgi)) desc) AS  
"RANKING"  
FROM B_Bilet_rodzaj_biletu brb  
JOIN B_rodzaj_biletu rb ON brb.id_rodzaj_biletu =  
rb.id_rodzaj_biletu  
JOIN B_rodzaj_ulgi ru ON rb.id_rodzaj_ulgi = ru.id_rodzaj_ulgi  
GROUP BY ru.nazwa;
```

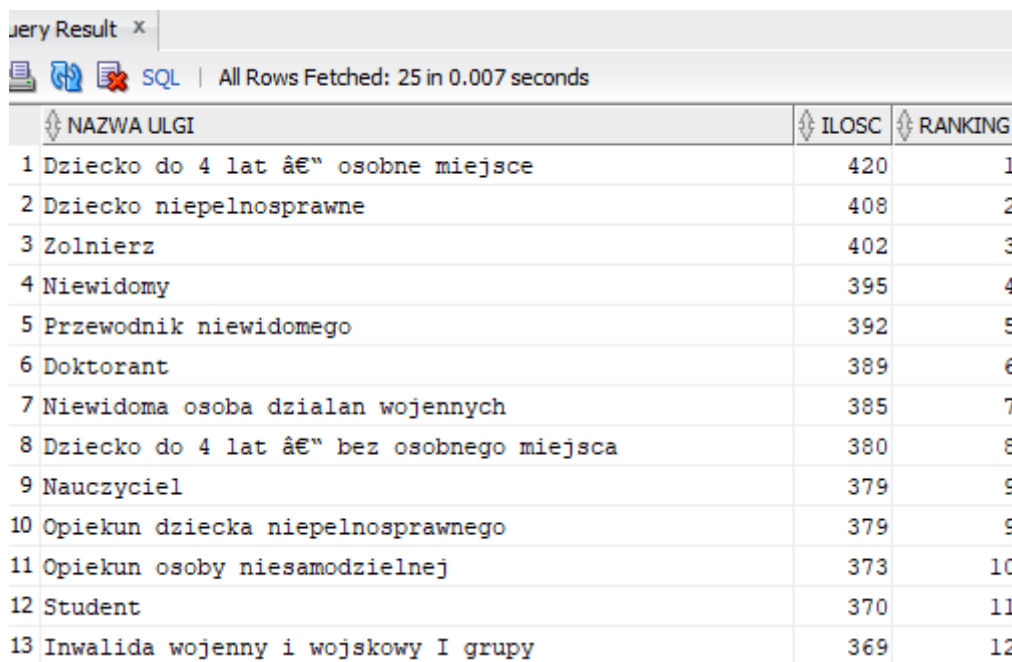
Kod zapytania hurtowni:

```
SELECT rb.rodzaj_ulgi as "NAZWA ULGI", count(rb.id_rodzaj_biletu) AS  
"ILOSC", DENSE_RANK() OVER (ORDER BY(count(rb.rodzaj_ulgi)) desc) AS  
"RANKING"  
FROM H_impieza i  
JOIN H_rodzaj_biletu rb ON i.id_rodzaj_biletu = rb.id_rodzaj_biletu  
WHERE rb.rodzaj_ulgi IS NOT NULL  
GROUP BY rb.rodzaj_ulgi;
```

Czas dla bazy: 0.007s

Czas dla hurtowni: 0.005s

Zrzuty ekranów:



NAZWA ULGI	ILOSC	RANKING
1 Dziecko do 4 lat – osobne miejsce	420	1
2 Dziecko niepełnosprawne	408	2
3 Żołnierz	402	3
4 Niewidomy	395	4
5 Przewodnik niewidomego	392	5
6 Doktorant	389	6
7 Niewidoma osoba działań wojennych	385	7
8 Dziecko do 4 lat – bez osobnego miejsca	380	8
9 Nauczyciel	379	9
10 Opiekun dziecka niepełnosprawnego	379	9
11 Opiekun osoby niesamodzielnej	373	10
12 Student	370	11
13 Inwalida wojenny i wojskowy I grupy	369	12

Query Result x		
SQL All Rows Fetched: 25 in 0.005 seconds		
NAZWA ULGI	ILOSC	RANKING
1 Dziecko do 4 lat " osobne miejsce	420	1
2 Dziecko niepełnosprawne	408	2
3 Żołnierz	402	3
4 Niewidomy	395	4
5 Przewodnik niewidomego	392	5
6 Doktorant	389	6
7 Niewidoma osoba działań wojennych	385	7
8 Dziecko do 4 lat " bez osobnego miejsca	380	8
9 Nauczyciel	379	9
10 Opiekun dziecka niepełnosprawnego	379	9
11 Opiekun osoby niesamodzielnej	373	10
12 Student	370	11
13 Inwalida wojenny i wojskowy I grupy	369	12

Opis zapytania: Ranking imprez, na które wydarzenia najchętniej ludzie kupują bilety.

Kod zapytania bazy:

```
SELECT ri.nazwa, sum(brb.ilosc) AS "ILOSC BILETOW",
DENSE_RANK() OVER (ORDER BY(sum(brb.ilosc)) desc) AS "RANKING"
FROM B_impreza i
JOIN B_Bilet b ON b.id_impreza = i.id_impreza
JOIN B_bilet_rodzaj_biletu brb ON brb.id_bilet = b.id_bilet
JOIN B_rodzaj_imprezy ri ON i.id_rodzaj_imprezy =
ri.id_rodzaj_imprezy
GROUP BY ri.nazwa;
```

Kod zapytania hurtowni:




```
SELECT ri.nazwa, sum(i.ilosc) AS "ILOSC BILETOW", DENSE_RANK() OVER
(ORDER BY(sum(i.ilosc)) desc) AS "RANKING"
FROM H_impreza i
JOIN H_rodzaj_imprezy ri ON i.id_rodzaj_imprezy =
ri.id_rodzaj_imprezy
GROUP BY ri.nazwa;
```

Czas dla bazy: 0.012s

Czas dla hurtowni: 0.006s




Zrzuty ekranów:

uery Result x

   SQL | All Rows Fetched: 8 in 0.012 seconds

NAZWA	ILOSC BILETOW	RANKING
1 Przedstawienie teatralne	2034	1
2 Wycieczka turystyczna	1991	2
3 Lot pasazerski	1842	3
4 Komunikacja miejska	1765	4
5 Przejazd kolejja	1757	5
6 Rejs morski	1749	6
7 Koncert muzyczny	1488	7
8 Film kinowy	1352	8

uery Result x

   SQL | All Rows Fetched: 8 in 0.006 seconds

NAZWA	ILOSC BILETOW	RANKING
1 Przedstawienie teatralne	2034	1
2 Wycieczka turystyczna	1991	2
3 Lot pasazerski	1842	3
4 Komunikacja miejska	1765	4
5 Przejazd kolejja	1757	5
6 Rejs morski	1749	6
7 Koncert muzyczny	1488	7
8 Film kinowy	1352	8

8. Uwagi i wnioski

Podczas wykonywania naszego projektu doszliśmy do wniosku, że bazy danych wykorzystywane są do przechowywania bieżącego stanu systemu, odpowiada np. za działalność operacyjną firmy (przyjmowanie, realizacja zamówień), wymaga ona również wysokiego poziomu niezawodności i odporności na awarie oraz łatwej modyfikacji bez konieczności przerabiania całej bazy. Baza danych powinna także w jakiś sposób kontrolować spójność danych i przechowywać niezbyt duże ilości danych, mieszczące się na jednym serwerze. Zapytania wykonują się zwykle dłużej na bazie danych, o ile w stosunku do hurtowni przechowuje tyle samo danych. Jest to spowodowane tym, że baza danych musi być logicznie z sobą powiązana i znormalizowana, zatem posiada dużo relacji, które zwiększają czas wykonania zapytań. Hurtownia danych za to służy głównie do celów analitycznych. Buduje się ją zwykle na podstawie schematu gwiazdy, ale może być również budowana jako płatek śniegu lub rozbudowany płatek śniegu. Hurtownie danych wykorzystuje się do przechowywania stanu historycznego, dane głównie są dopisywane, nie modyfikuje się ich, a kasuje hurtowo, gdy się przedawnią. Używając hurtowni, chcemy przejrzeć dużą ilość danych na raz, dlatego tak ważne jest to by, wykonywało się to szybko. Hurtownia nie musi być odporna na awarie czy kontrolować spójność danych. Sposób ładowania danych do bazy danych a hurtowni również się różni. Baza danych odwzorowuje model danych w każdym czasie. Hurtownię danych zasila się danymi pochodzącymi z baz danych, poza tym przechowuje ona dane historyczne, tak jak wspominaliśmy wyżej. Aby poprawnie zrealizować projekt hurtowni, powinniśmy zacząć od wybrania tabeli faktów, która będzie posiadała klucze obce do wszystkich tabel, które potrzebujemy. Tabele te powinniśmy wybierać, patrząc na zapytania, które chcemy użyć, po to by ich czas wykonania był jak najkrótszy. Dane w hurtowni nie muszą być znormalizowane. Porównując nasze czasy wykonywania zapytań dla bazy danych i dla hurtowni dla takiej samej ilości danych można dojść do wniosku, że zapytania wykonywane na hurtowni są zwykle o około 1,5-2 razy szybsze niż te same zapytania wykonywane na bazie. W przypadku małej ilości danych nie ma to aż takiego znaczenia, ale gdy posiada ich bardzo dużo to wtedy, może być to nawet

różnica kilku godzin, a nawet dni. Do ładowania danych do bazy wykorzystujemy sqlloader, ponieważ pozwala on na ładowanie danych bezpośrednio z plików, co jest bardzo przydatne, gdy nasze dane są generowane. Inne przydatne funkcje to np. tworzenie plików dziennika. Możemy tam znaleźć liczbę przetworzonych i zatwierdzonych wierszy. Dodatkowo w pliku dziennika możemy podejrzeć informacje o przebiegu ładowania danych. Natomiast w pliku tworzonym podczas niepowodzenia wprowadzania danych znajdziemy informacje, które wierszy nie zostały załadowane i z jakiego powodu. Dzięki temu znacznie szybciej możemy znaleźć problem w naszej bazie lub danych. Plik sterujący możemy dowolnie konfigurować, dzięki czemu możemy, np. używając warunku WHEN, określić by zostały załadowane tylko z jednego kraju. Do generowania danych używamy własnego generatora, który tworzy nam losowe dane w plikach csv o określonej ilości oraz generatorów internetowych, które dane następnie pobieramy. Kilka tabel zostało również wypełnionych w sposób ręczny. Ostatnią rzeczą o której chcemy wspomnieć jest to, że by możliwe było korzystanie jednocześnie z bazy i hurtowni na 1 serwerze musieliśmy do każdej tabeli dodać na początku „B_” (oznacza bazę danych) lub „H_” (oznacza hurtownię). Dzięki temu możemy jednocześnie wykonywać zapytania na bazie i na hurtowni oraz przechowywać dane. Wszystkie zapytania i skrypty zostały dostosowane do wspomnianego formatu. Schemat ERD bazy z uwagi na swoją obszerność, stał się mało czytelny w sprawozdaniu, dlatego schematy zostały zamieszczone w archiwum w odpowiednim folderze zarówno dla bazy jak i dla hurtowni, tak jak generator, który używaliśmy.