

Projekt: Bazy danych 2		
Temat: Sieć kwiaciarni		
Ocena:	Członkowie zespołu: Piotr Miernik Oskar Ołub	Grupa: 2ID14B
		Data oddania sprawozdania: 01.07.2023

SPIS TREŚCI

OPIS PROBLEMU I PRZYJĘTE ZAŁOŻENIA	4
SCHEMAT ERD BAZY	5
OPIS TABEL.....	6
TABELA POWIĄZAŃ.....	6
OPIS ELEMENTÓW MODELU	8
1. Adresy.....	8
2. Kwaciarnie	8
3. Magazyny	8
4. Gatunki	8
5. Magazynygatunki.....	9
6. Usługi.....	9
7. Kwaciarnieusługi.....	9
8. Dane_personalne	9
9. Dane_personalneAdresy	10
10. Pracownicy	10
11. Zatrudnienia	10
12. Klienci	10
13. Rachunki	11
14. Pozycja paragonu_gatunek.....	11
15. Pozycja paragonu_usługa	11
16. Dostawcy.....	12
17. Dostawy.....	12
18. Zamówienia.....	12
19. Zamówienia Pozycje	13
20. Dostawy Klientów.....	13
21. Oferty Specjalne	13
22. Reklamacje	14
TRANSFORMACJA BAZY DO HURTOWNI.....	15
1. Dim_Miasto.....	15
2. Dim_Kwaciarnia	15
3. Dim_Gatunek.....	15
4. Dim_Usluga	15
5. Dim_Pracownik	15
6. Dim_Klient	16
7. Fakt_Sprzedazy.....	16
Schemat hurtowni danych.....	17
SCHEMAT ERD HURTOWNI	18
ŁADOWANIE I TRANSFORMACJA DANYCH Z BAZY DO HURTOWNI.....	19
PORÓWNANIE CZASU WYKONANIA ZAPYTAŃ.....	21
ROLLUP.....	21
CUBE	24
Partycje obliczeniowe.....	27
Okna czasowe.....	30
Funkcje rankingowe	32
UWAGI I WNIOSKI	37

ELEMENTY PO POPRAWIE.....	37
<i>Schemat bazy</i>	37
<i>Zapytania baza</i>	37
<i>Schemat hurtownia</i>	37
SPIS LISTINGÓW	38

Opis problemu i przyjęte założenia

Temat naszego projektu to Sieć kwiaciarni. W ramach projektu stworzyliśmy bazę danych, która służy do zarządzania siecią kwiaciarni, ich asortymentem, transakcjami, a także do obsługi klientów. Oto kluczowe aspekty naszej bazy danych:

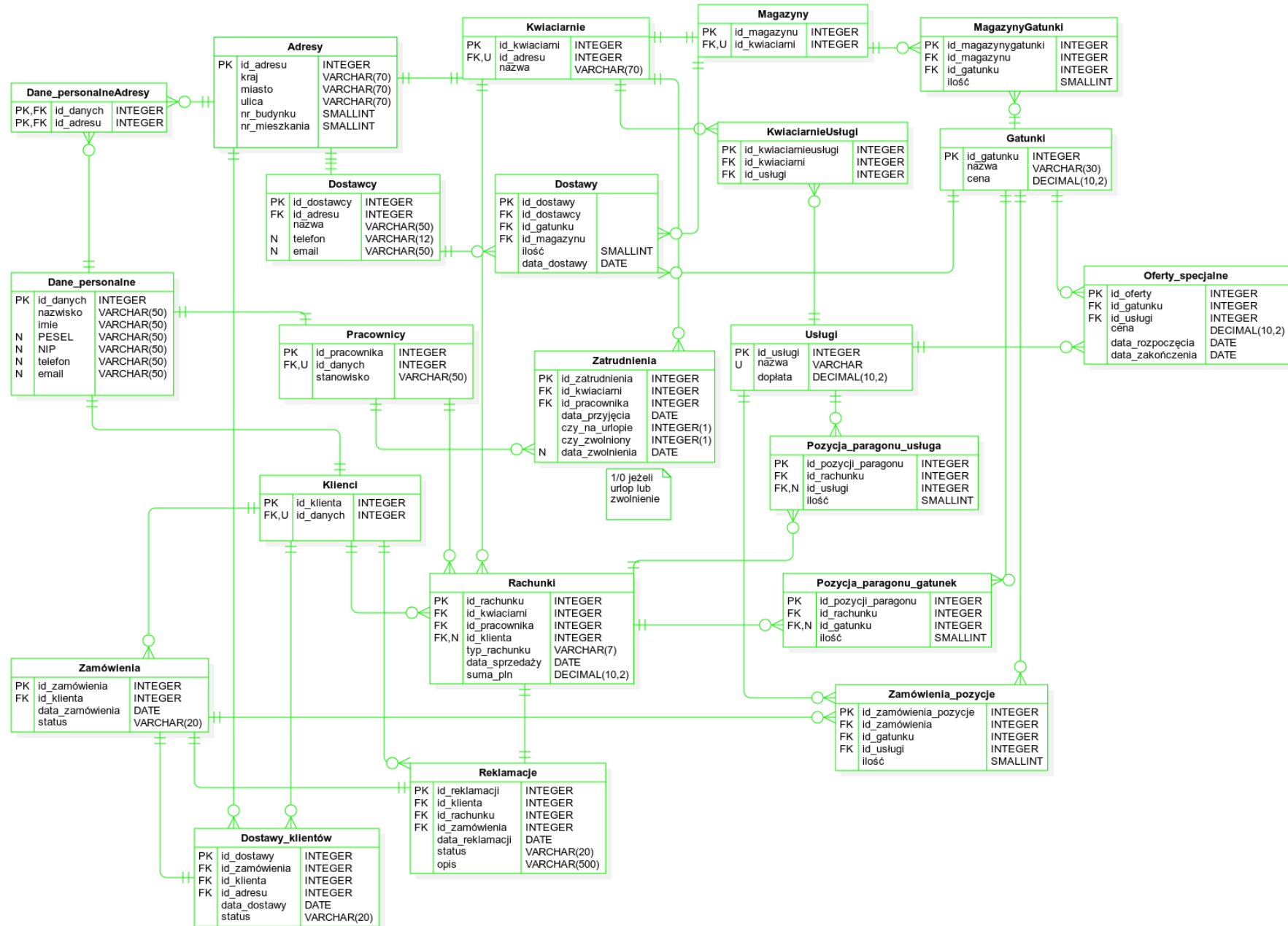
- 1. Zarządzanie kwiaciarniami i ich lokalizacją:** Baza danych pozwala na zarządzanie różnymi kwiaciarniami, które mogą być rozmieszczone w różnych lokalizacjach. Każda kwiaciarnia jest identyfikowana przez unikalny identyfikator i zawiera informacje takie jak nazwa kwiaciarni i adres.
- 2. Zarządzanie asortymentem:** Przechowujemy informacje o różnych gatunkach kwiatów dostępnych w kwiaciarniach, wraz z ich cenami. Dodatkowo, śledzimy ilość kwiatów w magazynach każdej kwiaciarni. Baza pozwala również na zarządzanie specjalnymi usługami, takimi jak aranżacje kwiatów, wraz z informacją o dodatkowej opłacie za usługę.
- 3. Obsługa klientów i zamówień:** Baza przechowuje informacje o klientach, włączając ich dane osobowe. Klienci mogą składać zamówienia na kwiaty i usługi, które są rejestrowane w systemie wraz z datą zamówienia i statusem. Klienci mają również możliwość składania zamówień z dostawą na wskazany adres.
- 4. Transakcje i rachunki:** Baza pozwala na generowanie rachunków po dokonaniu transakcji. Rachunki zawierają informacje o sprzedanych kwiatach i usługach, sumie do zapłaty, a także dane pracownika obsługującego transakcję.
- 5. Reklamacje:** Klienci mają możliwość składania reklamacji w przypadku problemów z zakupionymi produktami lub usługami. Reklamacje są rejestrowane w systemie wraz z datą, statusem i ewentualnym opisem problemu.
- 6. Oferty specjalne:** Baza pozwala na zarządzanie ofertami specjalnymi, które mogą zawierać rabaty na określone gatunki kwiatów lub usługi. Każda oferta ma określony okres ważności.
- 7. Relacje między tabelami:** Baza danych wykorzystuje odpowiednie klucze główne i obce, aby umożliwić skuteczne łączenie danych między różnymi tabelami, co umożliwia na przykład powiązanie zamówień z konkretnymi klientami i kwiaciarniami.

Te założenia stanowią podstawę dla naszej bazy danych, umożliwiając skuteczne zarządzanie siecią kwiaciarni, ich asortymentem, transakcjami i obsługą klienta. Poza tym, baza danych może być wykorzystywana do uzyskiwania wglądu w działalność poprzez analizę danych.

Baza danych jest w trzeciej postaci normalnej, ponieważ:

- Wszystkie tabele mają zdefiniowany klucz główny, co oznacza, że spełniają warunki 1NF (dane są w postaci tabelarycznej, każda kolumna ma atomową wartość).
- Wszystkie atrybuty, które nie są kluczami, zależą od całego klucza głównego, co oznacza, że spełniają warunki 2NF (nie ma zależności częściowej).
- Zależności transgraniczne (atrybut zależy od innego atrybutu, który nie jest kluczem) nie występują w żadnej z tabel, co oznacza, że spełniają warunki 3NF.

Schemat ERD bazy



Opis tabel

Tabela powiązań

Encja 1	Encja 2	Typ związku	Powód
adresy	kwiaciarnie	jeden do jeden	Każda kwiaciarnia ma jeden adres.
adresy	dostawcy	jeden do jeden	Każdy dostawca ma jeden adres.
adresy	dane_personalne_adresy	jeden do wielu	Jednym adresem może być powiązanych wiele danych personalnych.
kwiaciarnie	magazyny	jeden do jeden	Każda kwiaciarnia ma jeden magazyn.
kwiaciarnie	kwiaciarnieuslugi	jeden do wielu	Każda kwiaciarnia może oferować wiele usług.
kwiaciarnie	zatrudnienia	jeden do wielu	W jednej kwiaciarni może być zatrudnionych wielu pracowników.
kwiaciarnie	rachunki	jeden do wielu	Każda kwiaciarnia może wystawić wiele rachunków.
magazyny	magazynygatunki	jeden do wielu	W jednym magazynie może być przechowywanych wiele gatunków kwiatów.
magazyny	dostawy	jeden do wielu	Do jednego magazynu może być dostarczonych wiele dostaw.
gatunki	magazynygatunki	jeden do wielu	Jeden gatunek kwiatu może być przechowywany w wielu magazynach.
gatunki	pozycja_paragonu_gatunek	jeden do wielu	Jeden gatunek kwiatu może być na wielu pozycjach paragonu.
gatunki	dostawy	jeden do wielu	Jeden gatunek kwiatu może być dostarczany w wielu dostawach.
gatunki	zamowieniapozycje	jeden do wielu	Jeden gatunek kwiatu może być na wielu pozycjach zamówienia.
gatunki	oferty_specjalne	jeden do wielu	Jeden gatunek kwiatu może być częścią wielu ofert specjalnych.
uslugi	kwiaciarnieuslugi	jeden do wielu	Jedna usługa może być oferowana w wielu kwiaciarniach.
uslugi	pozycja_paragonu_usluga	jeden do wielu	Jedna usługa może być na wielu pozycjach paragonu.
uslugi	zamowieniapozycje	jeden do wielu	Jedna usługa może być na wielu pozycjach zamówienia.
uslugi	oferty_specjalne	jeden do wielu	Jedna usługa może być częścią wielu ofert specjalnych.
dostawcy	dostawy	jeden do wielu	Każdy dostawca może dokonać wielu dostaw do magazynu.
dane_personalne	dane_personalne_adresy	jeden do wielu	Jedne dane personalne mogą być powiązane z wieloma adresami.

Encja 1	Encja 2	Typ związku	Powód
dane_personalne	pracownicy	jeden do jeden	Każdy pracownik ma jedno dane personalne.
dane_personalne	klienci	jeden do jeden	Każdy klient ma jedno dane personalne.
pracownicy	zatrudnienia	jeden do wielu	Jeden pracownik może być zatrudniony w wielu kwiaciarniach.
pracownicy	rachunki	jeden do wielu	Jeden pracownik może wystawić wiele rachunków.
klienci	rachunki	jeden do wielu	Jeden klient może otrzymać wiele rachunków.
klienci	zamowienia	jeden do wielu	Jeden klient może złożyć wiele zamówień.
klienci	dostawy_klientow	jeden do wielu	Jeden klient może otrzymać wiele dostaw.
klienci	reklamacje	jeden do wielu	Jeden klient może złożyć wiele reklamacji.
rachunki	pozycja_paragonu_gatunek	jeden do wielu	Jeden rachunek może zawierać wiele pozycji z gatunkami kwiatów.
rachunki	pozycja_paragonu_usluga	jeden do wielu	Jeden rachunek może zawierać wiele pozycji z usługami.
rachunki	reklamacje	jeden do jeden	Każda reklamacja jest powiązana z jednym rachunkiem.
zamowienia	klienci	jeden do wielu	Klient może mieć kilka zamówień.
zamowienia	zamowieniapozycje	jeden do wielu	Jedno zamówienie może zawierać wiele pozycji (produktów lub usług).
zamowienia	dostawy_klientow	jeden do jeden	Każde zamówienie może mieć przypisaną jedną dostawę do klienta.
zamowienia	reklamacje	jeden do jeden	Każde zamówienie może być podstawą do jednej reklamacji.

Opis elementów modelu

1. Adresy

Zawiera dane adresowe.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_adresu	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
kraj	Nie	Nie	Nie	Nie	VARCHAR2(70)	Kraj
miasto	Nie	Nie	Nie	Nie	VARCHAR2(70)	Miasto
ulica	Nie	Nie	Nie	Nie	VARCHAR2(70)	Ulica
nr_budynku	Nie	Nie	Nie	Nie	SMALLINT	Numer budynku
nr_mieszkania	Nie	Nie	Nie	Tak	SMALLINT	Numer mieszkania

2. Kwiaciarnie

Zawiera budynki kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_kwaciarni	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_adresu	Nie	Tak (Adresy)	Tak	Nie	NUMBER	Klucz obcy do adresu
nazwa	Nie	Nie	Nie	Nie	VARCHAR2(70)	Nazwa budynku kwiaciarni

3. Magazyny

Zawiera informacje o magazynach.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_magazynu	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_kwaciarni	Nie	Tak (Kwaciarnie)	Tak	Nie	NUMBER	Klucz obcy do kwiaciarni

4. Gatunki

Zawiera informacje o gatunkach kwiatów.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_gatunku	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
nazwa	Nie	Nie	Nie	Nie	VARCHAR2(30)	Nazwa gatunku
cena	Nie	Nie	Nie	Nie	DECIMAL(10,2)	Cena gatunku

5. Magazynygatunki

Zawiera informacje o ilości danego gatunku kwiatów w magazynie.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_magazynygatunki	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_magazynu	Nie	Tak (Magazyny)	Nie	Nie	NUMBER	Klucz obcy do magazynu
id_gatunku	Nie	Tak (Gatunki)	Nie	Nie	NUMBER	Klucz obcy do gatunku
ilosc	Nie	Nie	Nie	Nie	SMALLINT	Ilość danego gatunku

6. Usługi

Zawiera informacje o usługach oferowanych przez kwiaciarnię.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_uslugi	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
nazwa	Nie	Nie	Tak	Nie	VARCHAR2(50)	Nazwa usługi
doplata	Nie	Nie	Nie	Nie	DECIMAL(10,2)	Dopłata za usługę

7. Kwiaciarnieuslugi

Zawiera informacje o usługach oferowanych przez poszczególne kwiaciarnie.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_kwiaciarnieuslugi	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_kwiaciarni	Nie	Tak (Kwiaciarnie)	Nie	Nie	NUMBER	Klucz obcy do kwiaciarni
id_uslugi	Nie	Tak (Uslugi)	Nie	Nie	NUMBER	Klucz obcy do usługi

8. Dane_personalne

Zawiera podstawowe dane personalne osób związanych z kwiaciarnią (klienci, pracownicy).

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_danych	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
nazwisko	Nie	Nie	Nie	Nie	VARCHAR2(30)	Nazwisko
imie	Nie	Nie	Tak	Tak	VARCHAR2(30)	Imię
PESEL	Nie	Nie	Tak	Tak	NUMBER(11)	Numer PESEL
NIP	Nie	Nie	Tak	Tak	NUMBER(10)	Numer NIP
telefon	Nie	Nie	Tak	Tak	VARCHAR2(50)	Numer telefonu
email	Nie	Nie	Tak	Tak	VARCHAR2(50)	Adres email

9. Dane_personalneAdresy

Zawiera informacje o adresach połączonych z danymi personalnymi.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_danych	Tak	Tak (Dane_personalne)	Tak	Nie	NUMBER	Identyfikator
id_adresu	Tak	Tak (Adresy)	Tak	Nie	NUMBER	Identyfikator

10. Pracownicy

Zawiera informacje o pracownikach kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_pracownika	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_danych	Nie	Tak (Dane_personalne)	Nie	Nie	NUMBER	Klucz obcy do danych personalnych
stanowisko	Nie	Nie	Nie	Tak	VARCHAR2(50)	Stanowisko pracownika

11. Zatrudnienia

Zawiera informacje o zatrudnieniach pracowników w kwiaciarniach.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_zatrudnienia	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_kwiaciarni	Nie	Tak (Kwiaciarnie)	Nie	Nie	NUMBER	Klucz obcy do kwiaciarni
id_pracownika	Nie	Tak (Pracownicy)	Nie	Nie	NUMBER	Klucz obcy do pracownika
data_zatrudnienia	Nie	Nie	Nie	Nie	DATE	Data zatrudnienia
data_zwolnienia	Nie	Nie	Nie	Tak	DATE	Data zwolnienia

12. Klienci

Zawiera informacje o klientach kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_klienta	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_danych	Nie	Tak (Dane_personalne)	Tak	Nie	NUMBER	Klucz obcy do danych personalnych

13. Rachunki

Zawiera informacje o rachunkach wystawionych przez kwiaciarnię.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_rachunku	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_kwiaciarni	Nie	Tak (Kwiaciarnie)	Nie	Nie	NUMBER	Klucz obcy do kwiaciarni
id_pracownika	Nie	Tak (Pracownicy)	Nie	Nie	NUMBER	Klucz obcy do pracownika
id_klienta	Nie	Tak (Klienci)	Nie	Tak	NUMBER	Klucz obcy do klienta
typ_rachunku	Nie	Nie	Nie	Nie	VARCHAR2(7)	Typ rachunku
data_sprzedazy	Nie	Nie	Nie	Nie	DATE	Data sprzedaży
suma_pln	Nie	Nie	Nie	Nie	DECIMAL(10,2)	Suma do zapłaty w PLN

14. Pozycja paragonu_gatunek

Zawiera szczegóły dotyczące każdej pozycji gatunku na paragonie.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_pozycji_paragonu	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_rachunku	Nie	Tak (Rachunki)	Nie	Nie	NUMBER	Klucz obcy do rachunku
id_gatunku	Nie	Tak (Gatunki)	Nie	Tak	NUMBER	Klucz obcy do gatunku
ilosc	Nie	Nie	Nie	Nie	SMALLINT	Ilość

15. Pozycja paragonu_usługa

Zawiera szczegóły dotyczące każdej pozycji usług na paragonie.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_pozycji_paragonu	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_rachunku	Nie	Tak (Rachunki)	Nie	Nie	NUMBER	Klucz obcy do rachunku
id_uslugi	Nie	Tak (Usługi)	Nie	Tak	NUMBER	Klucz obcy do usługi
ilosc	Nie	Nie	Nie	Nie	SMALLINT	Ilość

16. Dostawcy

Zawiera informacje o dostawcach dla kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_dostawcy	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
nazwa	Nie	Nie	Nie	Nie	VARCHAR2(50)	Nazwa dostawcy
telefon	Nie	Nie	Nie	Tak	VARCHAR2(50)	Numer telefonu
email	Nie	Nie	Nie	Tak	VARCHAR2(50)	Adres email
id_adresu	Nie	Tak (Adresy)	Nie	Nie	NUMBER	Klucz obcy do adresu

17. Dostawy

Zawiera informacje o dostawach roślin do magazynów kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_dostawy	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_dostawcy	Nie	Tak (Dostawcy)	Nie	Nie	NUMBER	Klucz obcy do dostawcy
id_gatunku	Nie	Tak (Gatunki)	Nie	Nie	NUMBER	Klucz obcy do gatunku
id_magazynu	Nie	Tak (Magazyny)	Nie	Nie	NUMBER	Klucz obcy do magazynu
ilosc	Nie	Nie	Nie	Nie	SMALLINT	Ilość dostarczonych roślin
data_dostawy	Nie	Nie	Nie	Nie	DATE	Data dostawy

18. Zamówienia

Zawiera informacje o zamówieniach klientów.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_zamowienia	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_klienta	Nie	Tak (Klienci)	Nie	Nie	NUMBER	Klucz obcy do klienta
data_zamowienia	Nie	Nie	Nie	Nie	DATE	Data zamówienia
status	Nie	Nie	Nie	Nie	VARCHAR2(20)	Status zamówienia

19. Zamówienia Pozycje

Zawiera szczegółowe informacje o zamówionych pozycjach (gatunki roślin i usługi).

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_zamowienia_pozycje	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_zamowienia	Nie	Tak (Zamówienia)	Nie	Nie	NUMBER	Klucz obcy do zamówienia
id_gatunku	Nie	Tak (Gatunki)	Nie	Tak	NUMBER	Klucz obcy do gatunku
id_uslugi	Nie	Tak (Usługi)	Nie	Tak	NUMBER	Klucz obcy do usługi
ilosc	Nie	Nie	Nie	Nie	SMALLINT	Ilość zamówionych elementów

20. Dostawy Klientów

Zawiera informacje o dostawach zamówień do klientów.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_dostawy	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_zamowienia	Nie	Tak (Zamówienia)	Nie	Nie	NUMBER	Klucz obcy do zamówienia
id_klienta	Nie	Tak (Klienci)	Nie	Nie	NUMBER	Klucz obcy do klienta
id_adresu	Nie	Tak (Adresy)	Nie	Nie	NUMBER	Klucz obcy do adresu
data_dostawy	Nie	Nie	Nie	Nie	DATE	Data dostawy
status	Nie	Nie	Nie	Nie	VARCHAR2(20)	Status dostawy

21. Oferty Specjalne

Zawiera informacje o specjalnych ofertach kwiatarni na określone gatunki roślin lub usługi.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_oferty	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_gatunku	Nie	Tak (Gatunki)	Nie	Tak	NUMBER	Klucz obcy do gatunku
id_uslugi	Nie	Tak (Usługi)	Nie	Tak	NUMBER	Klucz obcy do usługi
cena	Nie	Nie	Nie	Nie	DECIMAL(10,2)	Cena w ofercie
data_rozpozecia	Nie	Nie	Nie	Nie	DATE	Data rozpoczęcia oferty
data_zakonczenia	Nie	Nie	Nie	Nie	DATE	Data zakończenia oferty

22. Reklamacje

Zawiera informacje o reklamacjach złożonych przez klientów.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_reklamacji	Tak	Nie	Tak	Nie	NUMBER	Identyfikator
id_klienta	Nie	Tak (Klienci)	Nie	Nie	NUMBER	Klucz obcy do klienta
id_rachunku	Nie	Tak (Rachunki)	Nie	Tak	NUMBER	Klucz obcy do rachunku
id_zamowienia	Nie	Tak (Zamówienia)	Nie	Tak	NUMBER	Klucz obcy do zamówienia
data_reklamacji	Nie	Nie	Nie	Nie	DATE	Data złożenia reklamacji
status	Nie	Nie	Nie	Nie	VARCHAR2(20)	Status reklamacji
opis	Nie	Nie	Nie	Tak	VARCHAR2(500)	Opis reklamacji

Transformacja bazy do hurtowni

1. Dim_Miasto

Tabela zawiera informacje o miastach.

Nazwa	Klucz główny	Klucz obcy	Unique	Nullable	Typ/Dziedzina	Opis
id_miasta	Tak	Nie	Tak	Nie	NUMBER	Identyfikator miasta
nazwa_miasta	Nie	Nie	Nie	Nie	VARCHAR2(50)	Nazwa miasta

2. Dim_Kwiaciarnia

Tabela zawiera informacje o kwiaciarniach.

Nazwa	Klucz główny	Klucz obcy	Unique	Nullable	Typ/Dziedzina	Opis
id_kwiaciarni	Tak	Nie	Tak	Nie	NUMBER	Identyfikator kwiaciarni
nazwa	Nie	Nie	Nie	Nie	VARCHAR2(100)	Nazwa kwiaciarni
id_miasta	Nie	Tak	Nie	Tak	NUMBER	Klucz obcy odwołujący się do tabeli Dim_Miasto

3. Dim_Gatunek

Tabela zawiera informacje o gatunkach kwiatów.

Nazwa	Klucz główny	Klucz obcy	Unique	Nullable	Typ/Dziedzina	Opis
id_gatunku	Tak	Nie	Tak	Nie	NUMBER	Identyfikator gatunku
nazwa	Nie	Nie	Nie	Nie	VARCHAR2(50)	Nazwa gatunku

4. Dim_Usluga

Tabela zawiera informacje o dostępnych usługach.

Nazwa	Klucz główny	Klucz obcy	Unique	Nullable	Typ/Dziedzina	Opis
id_uslugi	Tak	Nie	Tak	Nie	NUMBER	Identyfikator usługi
nazwa	Nie	Nie	Nie	Nie	VARCHAR2(100)	Nazwa usługi
cena	Nie	Nie	Nie	Nie	DECIMAL(10, 2)	Cena usługi

5. Dim_Pracownik

Tabela zawiera informacje o pracownikach kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	Nullable	Typ/Dziedzina	Opis
id_pracownika	Tak	Nie	Tak	Nie	NUMBER	Identyfikator pracownika
imie	Nie	Nie	Nie	Nie	VARCHAR2(50)	Imię pracownika
nazwisko	Nie	Nie	Nie	Nie	VARCHAR2(50)	Nazwisko pracownika

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
stanowisko	Nie	Nie	Nie	Nie	VARCHAR2(50)	Stanowisko pracownika
data_zatrudnienia	Nie	Nie	Nie	Nie	DATE	Data zatrudnienia pracownika

6. Dim_Klient

Tabela zawiera informacje o klientach kwiaciarni.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id_klientai	Tak	Nie	Tak	Nie	NUMBER	Identyfikator klienta
imie	Nie	Nie	Nie	Nie	VARCHAR2(50)	Imie klienta
nazwisko	Nie	Nie	Nie	Nie	VARCHAR2(50)	Nazwisko klienta
Email	Nie	Nie	Nie	Nie	VARCHAR2(100)	Email klienta

7. Fakt_Sprzedaży

Tabela zawiera informacje o sprzedaży usług i produktów w kwiaciarniach.

Nazwa	Klucz główny	Klucz obcy	Unique	NULLable	Typ/Dziedzina	Opis
id	Tak	Nie	Tak	Nie	NUMBER	Identyfikator sprzedaży
Id_kwiaciarni	Nie	Tak	Nie	Nie	NUMBER	Klucz obcy odwołujący się do tabeli Dim_Kwiaciarnia
id_gatunku	Nie	Tak	Nie	Nie	NUMBER	Klucz obcy odwołujący się do tabeli Dim_Gatunek
id_uslugi	Nie	Tak	Nie	Nie	NUMBER	Klucz obcy odwołujący się do tabeli Dim_Usluga
id_pracownika	Nie	Tak	Nie	Nie	NUMBER	Klucz obcy odwołujący się do tabeli Dim_Pracownik
id_klienta	Nie	Tak	Nie	Nie	NUMBER	Klucz obcy odwołujący się do tabeli Dim_Klient
ilosc	Nie	Nie	Nie	Nie	NUMBER	Ilość sprzedanych produktów/usług
suma_pln	Nie	Nie	Nie	Nie	DECIMAL(10, 2)	Przychód ze sprzedaży
data_sprzedazy	Nie	Nie	Nie	Nie	DATE	Data sprzedaży

Schemat hurtowni danych

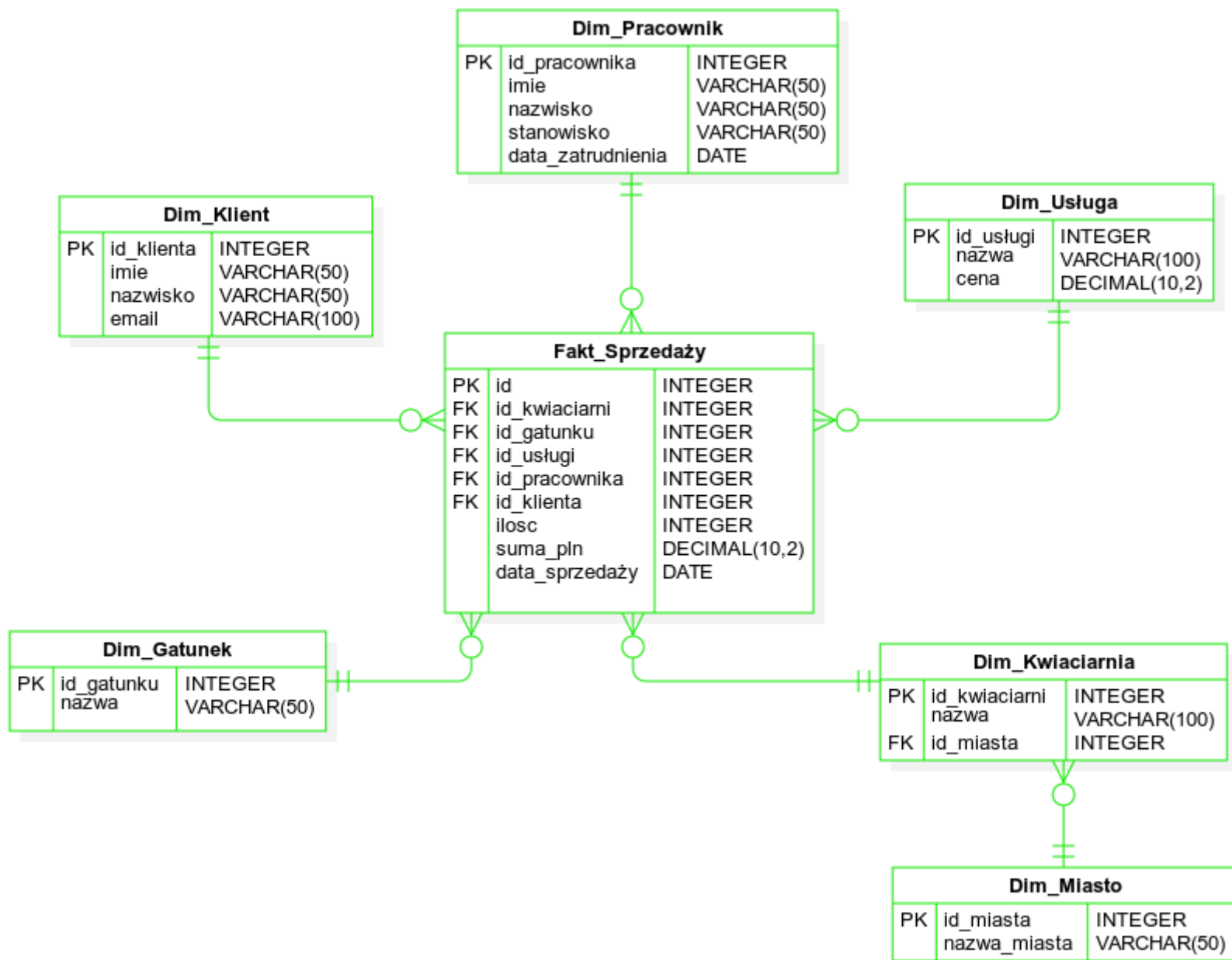
Hurtownia danych skupia się na przechowywaniu informacji na temat sprzedaży w kwiaciarniach. Składa się z siedmiu tabel, z których sześć jest tabelami wymiarowymi (*Dim_Miasto*, *Dim_Kwiaciarnia*, *Dim_Gatunek*, *Dim_Usluga*, *Dim_Pracownik* i *Dim_Klient*) oraz jedną tabelą faktów (*Fakt_Sprzedaży*).

Relacje:

- *Dim_Kwiaciarnia* zawiera klucz obcy *id_miasta*, który odnosi się do klucza głównego *id_miasta* w tabeli *Dim_Miasto*.
- *Fakty_Sprzedaż* zawiera pięć kluczy obcych, które odnoszą się do tabel wymiarowych:
id_kwiaciarni odwołuje się do *Dim_Kwiaciarnia*, *id_gatunku* odwołuje się do *Dim_Gatunek*,
id_uslugi odwołuje się do *Dim_Usluga*, *id_pracownika* odwołuje się do *Dim_Pracownik*,
id_klienta odwołuje się do *Dim_Klient*.

Hurtownia danych umożliwia analizę informacji na temat sprzedaży, takich jak przychody, ilość sprzedanych produktów, a także relacje między lokalizacją, rodzajem produktów i usług oraz pracownikami.

Schemat ERD hurtowni



Ładowanie i transformacja danych z bazy do hurtowni

Aby załadować dane z bazy danych do hurtowni danych składającej się z różnych tabel wymiarowych i tabeli faktów, stworzyliśmy procedurę **load_data** napisaną w PL/SQL. Procedura najpierw wprowadza dane do tabeli **Dim_Miasto** na podstawie informacji z tabeli **adresy**. Następnie wykonuje podobne operacje dla innych tabel wymiarowych (**Dim_Kwiaciarnia**, **Dim_Gatunek**, **Dim_Usluga**, **Dim_Pracownik**, **Dim_Klient**). Główną część procedury stanowi wypełnienie tabeli **Fakt_Sprzedazy**, gdzie dane są agregowane z różnych źródeł, takich jak **rachunki**, **pozycja_paragonu_gatunki** i **pozycja_paragonu_uslugi**. Na koniec, procedura wykonuje operację **COMMIT** aby zapisać zmiany, a w przypadku wystąpienia błędu, wypisuje informacje o tym błędzie.

Kod procedury:

```
CREATE OR REPLACE PROCEDURE load_data AS
BEGIN
    -- Wprowadź dane do Dim_Miasto
    INSERT INTO Dim_Miasto
        (id_miasta, nazwa_miasta)
    SELECT DISTINCT id_adresu, miasto
    FROM adresy;

    -- Wprowadź dane do Dim_Kwiaciarnia
    INSERT INTO Dim_Kwiaciarnia
        (id_kwiaciarni, nazwa, id_miasta)
    SELECT k.id_kwiaciarni, k.nazwa, a.id_adresu
    FROM kwiaciarnie k
    JOIN adresy a ON k.id_adresu = a.id_adresu;

    -- Wprowadź dane do Dim_Gatunek
    INSERT INTO Dim_Gatunek
        (id_gatunku, nazwa)
    SELECT id_gatunku, nazwa
    FROM gatunki;

    -- Wprowadź dane do Dim_Usluga
    INSERT INTO Dim_Usluga
        (id_uslugi, nazwa, cena)
    SELECT id_uslugi, nazwa, doplata
    FROM uslugi;

    -- Wprowadź dane do Dim_Pracownik
    INSERT INTO Dim_Pracownik
        (id_pracownika, imie, nazwisko, stanowisko, data_zatrudnienia)
    SELECT p.id_pracownika, dp.imie, dp.nazwisko, p.stanowisko, MAX(z.data_przy-
jacia)
    FROM pracownicy p
    JOIN dane_personalne dp ON p.id_danych = dp.id_danych
    JOIN zatrudnienia z ON p.id_pracownika = z.id_pracownika
    GROUP BY p.id_pracownika, dp.imie, dp.nazwisko, p.stanowisko;
```

Listing 1: Kod procedury, cz.1

```

-- Wprowadzanie danych do Dim_Klient
INSERT INTO Dim_Klient
(id_klienta, imie, nazwisko, email)
SELECT k.id_klienta, dp.imie, dp.nazwisko, dp.email
FROM klienci k
JOIN dane_personalne dp ON k.id_danych = dp.id_danych;

-- Wprowadź dane do Fakt_Sprzedazy
INSERT INTO Fakt_Sprzedazy
(id, id_kwiaciarni, id_gatunku, id_uslugi, id_pracownika, id_klienta,
ilosc, suma_pln, data_sprzedazy)
SELECT
rownum,
r.id_kwiaciarni,
pg.id_gatunku,
pu.id_uslugi,
r.id_pracownika,
r.id_klienta,
NVL(pg.ilosc, pu.ilosc),
r.suma_pln,
r.data_sprzedazy
FROM rachunki r
LEFT JOIN pozycja_paragonu_gatunek pg ON r.id_rachunku = pg.id_rachunku
LEFT JOIN pozycja_paragonu_usluga pu ON r.id_rachunku = pu.id_rachunku;

COMMIT;
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE
('An error occurred: ' || SQLCODE || ' ' || SQLERRM);
END;
/
EXECUTE load_data();

```

Listing 2: Kod procedury, cz.2

Porównanie czasu wykonania zapytań

ROLLUP

1. Pokazuje łączną ilość sprzedanych produktów i łączną kwotę sprzedaży dla każdej kwiaciarni w każdym mieście.

Baza:

```
SELECT a.id_adresu,  
       a.miasto,  
       k.id_kwiaciarni,  
       sum(ppg.ilosc) as laczna_ilosc,  
       sum(r.suma_pln) as laczna_kwota  
FROM adresy a  
     JOIN kwiaciarnie k ON a.id_adresu = k.id_adresu  
     JOIN rachunki r ON k.id_kwiaciarni = r.id_kwiaciarni  
     JOIN pozycja_paragonu_gatunek ppg ON r.id_rachunku = ppg.id_rachunku  
GROUP BY ROLLUP (a.id_adresu, a.miasto, k.id_kwiaciarni);
```

Listing 3: ROLLUP – I – baza

Hurtownia:

```
SELECT d.id_miasta,  
       d.nazwa_miasta,  
       k.id_kwiaciarni,  
       sum(f.ilosc) as laczna_ilosc,  
       sum(f.suma_pln) as laczna_kwota  
FROM Dim_Miasto d  
     JOIN Dim_Kwiaciarnia k ON d.id_miasta = k.id_miasta  
     JOIN Fakt_Sprzedazy f ON k.id_kwiaciarni = f.id_kwiaciarni  
GROUP BY ROLLUP (d.id_miasta, d.nazwa_miasta, k.id_kwiaciarni);
```

Listing 4: ROLLUP – I – hurtownia

LP.	BAZA	HURTOWNIA
1	0,014	0,013
2	0,017	0,015
3	0,017	0,017
ŚREDNIA	0,016	0,015

2. Pokazuje które kombinacje gatunków kwiatów i usług przyczyniają się do największych dochodów.

Baza:

```
SELECT
  prac.id_pracownika,
  dane.imie,
  dane.nazwisko,
  gatunki.id_gatunku,
  gatunki.nazwa as nazwa_gatunku,
  SUM(pozycja_gatunek.ilosc) as laczna_ilosc
FROM
  pracownicy prac
  JOIN dane_personalne dane ON prac.id_danych = dane.id_danych
  JOIN rachunki rach ON prac.id_pracownika = rach.id_pracownika
  JOIN pozycja_paragonu_gatunek pozycja_gatunek ON rach.id_rachunku = pozycja_gatunek.id_rachunku
  JOIN gatunki ON pozycja_gatunek.id_gatunku = gatunki.id_gatunku
GROUP BY ROLLUP (prac.id_pracownika, dane.imie, dane.nazwisko, gatunki.id_gatunku, gatunki.nazwa);
```

Listing 5: ROLLUP – 2 – baza

Hurtownia:

```
SELECT p.id_pracownika,
  p.imie,
  p.nazwisko,
  g.id_gatunku,
  g.nazwa,
  sum(f.ilosc) as laczna_ilosc
FROM Dim_Pracownik p
  JOIN Fakt_Sprzedazy f ON p.id_pracownika = f.id_pracownika
  JOIN Dim_Gatunek g ON f.id_gatunku = g.id_gatunku
GROUP BY ROLLUP (p.id_pracownika, p.imie, p.nazwisko, g.id_gatunku, g.nazwa);
```

Listing 6: ROLLUP – 2 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,086	0,073
2	0,081	0,081
3	0,079	0,081
ŚREDNIA	0,082	0,0783

3. Zapytanie ma na celu analizę wpływu różnych gatunków kwiatów na dochód. Pozwala to zrozumieć, które gatunki kwiatów przyczyniają się do większych dochodów w zależności od ich ceny i ilości sprzedanych sztuk.

Baza:

```
SELECT us.id_uslugi,
       us.nazwa AS nazwa_uslugi,
       dp.id_danych AS id_klienta,
       dp.imie,
       dp.nazwisko,
       SUM(ppu.ilosc) AS ilosc_uslug
FROM   uslugi us
       JOIN pozycja_paragonu_usluga ppu ON us.id_uslugi = ppu.id_uslugi
       JOIN rachunki r ON ppu.id_rachunku = r.id_rachunku
       JOIN klienci k ON r.id_klienta = k.id_klienta
       JOIN dane_personalne dp ON k.id_danych = dp.id_danych
GROUP BY ROLLUP (us.id_uslugi, us.nazwa, dp.id_danych, dp.imie, dp.nazwisko);
```

Listing 7: ROLLUP – 3 – baza

Hurtownia:

```
SELECT u.id_uslugi,
       u.nazwa,
       k.id_klienta,
       k.imie,
       k.nazwisko,
       sum(f.ilosc) as ilosc_uslug
FROM   Dim_Usluga u
       JOIN Fakt_Sprzedazy f ON u.id_uslugi = f.id_uslugi
       JOIN Dim_Klient k ON f.id_klienta = k.id_klienta
GROUP BY ROLLUP (u.id_uslugi, u.nazwa, k.id_klienta, k.imie, k.nazwisko);
```

Listing 8: ROLLUP – 3 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,112	0,121
2	0,114	0,106
3	0,12	0,106
ŚREDNIA	0,1153	0,111

CUBE

1. Zapytanie analizujące sprzedaż produktów według miasta i gatunku. Pozwala to na identyfikację, które miasta i gatunki produktów generują największe przychody.

Baza:

```
SELECT
    adresy.miasto,
    gatunki.id_gatunku,
    SUM(pozycja_paragonu_gatunek.ilosc) AS laczna_ilosc,
    SUM(pozycja_paragonu_gatunek.ilosc * gatunki.cena) AS laczna_kwota
FROM
    pozycja_paragonu_gatunek
JOIN
    rachunki ON pozycja_paragonu_gatunek.id_rachunku = rachunki.id_rachunku
JOIN
    kwiaciarnie ON rachunki.id_kwiaciarni = kwiaciarnie.id_kwiaciarni
JOIN
    adresy ON kwiaciarnie.id_adresu = adresy.id_adresu
JOIN
    gatunki ON pozycja_paragonu_gatunek.id_gatunku = gatunki.id_gatunku
GROUP BY
    CUBE (adresy.miasto, gatunki.id_gatunku);
```

Listing 9: CUBE – I – baza

Hurtownia:

```
SELECT dmiasto.id_miasta,
    dgatunek.id_gatunku,
    SUM(fs.ilosc) as laczna_ilosc,
    SUM(fs.suma_pln) as laczna_kwota
FROM Fakt_Sprzedazy fs
JOIN Dim_Kwiaciarnia dkwiaciarnia ON fs.id_kwiaciarni = dkwiaciarnia.id_kwiaciarni
JOIN Dim_Miasto dmiasto ON dkwiaciarnia.id_miasta = dmiasto.id_miasta
JOIN Dim_Gatunek dgatunek ON fs.id_gatunku = dgatunek.id_gatunku
GROUP BY CUBE (dmiasto.id_miasta, dgatunek.id_gatunku);
```

Listing 10: CUBE – I – hurtownia

LP.	BAZA	HURTOWNIA
1	0,045	0,038
2	0,046	0,04
3	0,048	0,043
ŚREDNIA	0,0463	0,0403

2. Zapytanie analizujące wpływy z usług w różnych kwaciarniach. Można dzięki temu zrozumieć, jakie usługi są najbardziej dochodowe w różnych lokalizacjach.

Baza:

```
SELECT
  kwaciarnie.id_kwaciarni,
  uslugi.id_uslugi,
  SUM(pozycja_paragonu_usluga.ilosc) AS laczna_ilosc,
  SUM(pozycja_paragonu_usluga.ilosc * uslugi.doplata) AS laczna_kwota
FROM
  pozycja_paragonu_usluga
  JOIN
  rachunki ON pozycja_paragonu_usluga.id_rachunku = rachunki.id_rachunku
  JOIN
  kwaciarnie ON rachunki.id_kwaciarni = kwaciarnie.id_kwaciarni
  JOIN
  uslugi ON pozycja_paragonu_usluga.id_uslugi = uslugi.id_uslugi
GROUP BY
  CUBE (kwaciarnie.id_kwaciarni, uslugi.id_uslugi);
```

Listing 11: CUBE – 2 – baza

Hurtownia:

```
SELECT dkwaciarnia.id_kwaciarni,
  dusluga.id_uslugi,
  SUM(fs.ilosc) as laczna_ilosc,
  SUM(fs.suma_pln) as laczna_kwota
FROM Fakt_Sprzedazy fs
  JOIN Dim_Kwaciarnia dkwaciarnia ON fs.id_kwaciarni = dkwaciarnia.id_kwaciarni
  JOIN Dim_Usluga dusluga ON fs.id_uslugi = dusluga.id_uslugi
GROUP BY CUBE (dkwaciarnia.id_kwaciarni, dusluga.id_uslugi);
```

Listing 12: CUBE – 2 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,048	0,047
2	0,044	0,038
3	0,044	0,04
ŚREDNIA	0,0453	0,0416

3. Zapytanie analizujące wpływy od różnych klientów w zależności od gatunku produktu. Pozwala to na zrozumienie, jacy klienci wydają najwięcej pieniędzy na jakie gatunki produktów.

Baza:

```
SELECT
    klienci.id_klienta,
    gatunki.id_gatunku,
    SUM(pozycja_paragonu_gatunek.ilosc) AS laczna_ilosc,
    SUM(pozycja_paragonu_gatunek.ilosc * gatunki.cena) AS laczna_kwota
FROM
    pozycja_paragonu_gatunek
JOIN
    rachunki ON pozycja_paragonu_gatunek.id_rachunku = rachunki.id_rachunku
JOIN
    klienci ON rachunki.id_klienta = klienci.id_klienta
JOIN
    gatunki ON pozycja_paragonu_gatunek.id_gatunku = gatunki.id_gatunku
GROUP BY
    CUBE (klienci.id_klienta, gatunki.id_gatunku);
```

Listing 13: CUBE – 3 – baza

Hurtownia:

```
SELECT dklient.id_klienta,
    dgatunek.id_gatunku,
    SUM(fs.ilosc) as laczna_ilosc,
    SUM(fs.suma_pln) as laczna_kwota
FROM Fakt_Sprzedazy fs
JOIN Dim_Klient dklient ON fs.id_klienta = dklient.id_klienta
JOIN Dim_Gatunek dgatunek ON fs.id_gatunku = dgatunek.id_gatunku
GROUP BY CUBE (dklient.id_klienta, dgatunek.id_gatunku);
```

Listing 14: CUBE – 3 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,035	0,033
2	0,035	0,033
3	0,041	0,038
ŚREDNIA	0,037	0,0346

Partycje obliczeniowe

1. Suma sprzedaży dla każdej kwiaciarni i gatunku kwiatu w danym miesiącu.

Pozwala na analizę, jakie gatunki kwiatów przynoszą największe przychody w poszczególnych kwiaciarniach, co jest przydatne przy planowaniu oferty.

Baza:

```
SELECT
  k.nazwa AS kwiaciarnia,
  g.nazwa AS gatunek,
  EXTRACT(MONTH FROM r.data_sprzedazy) AS miesiac,
  SUM(ppg.ilosc * g.cena) AS calkowita_sprzedaz
FROM
  rachunki r
  JOIN
    kwiaciarnie k ON r.id_kwiaciarni = k.id_kwiaciarni
  JOIN
    pozycja_paragonu_gatunek ppg ON r.id_rachunku = ppg.id_rachunku
  JOIN
    gatunki g ON ppg.id_gatunku = g.id_gatunku
GROUP BY
  k.nazwa, g.nazwa, EXTRACT(MONTH FROM r.data_sprzedazy);
```

Listing 15: Partycje obliczeniowe – I – baza

Hurtownia:

```
SELECT k.nazwa AS kwiaciarnia, g.nazwa AS gatunek, EXTRACT(MONTH FROM
f.data_sprzedazy) AS miesiac, SUM(f.suma_pln) AS calkowita_sprzedaz
FROM Fakt_Sprzedazy f
  JOIN Dim_Kwiaciarnia k ON f.id_kwiaciarni = k.id_kwiaciarni
  JOIN Dim_Gatunek g ON f.id_gatunku = g.id_gatunku
GROUP BY k.nazwa, g.nazwa, EXTRACT(MONTH FROM f.data_sprzedazy);
```

Listing 16: Partycje obliczeniowe – I – hurtownia

LP.	BAZA	HURTOWNIA
1	0,025	0,024
2	0,027	0,023
3	0,022	0,026
ŚREDNIA	0,0246	0,0243

2. Średnia cena sprzedaży dla każdego pracownika i usługi w danym miesiącu. Umożliwia monitorowanie wyników sprzedaży pracowników i efektywności różnych usług w celu optymalizacji zarządzania personelem i portfolio usług.

Baza:

```
SELECT
  dp.imie || ' ' || dp.nazwisko AS pracownik,
  u.nazwa AS usługa,
  EXTRACT(MONTH FROM r.data_sprzedazy) AS miesiac,
  AVG(ppu.ilosc * u.doplata) AS srednia_cena
FROM
  rachunki r
  JOIN
    pracownicy p ON r.id_pracownika = p.id_pracownika
  JOIN
    dane_personalne dp ON p.id_danych = dp.id_danych
  JOIN
    pozycja_paragonu_usluga ppu ON r.id_rachunku = ppu.id_rachunku
  JOIN
    usługi u ON ppu.id_uslugi = u.id_uslugi
GROUP BY
  dp.imie, dp.nazwisko, u.nazwa, EXTRACT(MONTH FROM r.data_sprzedazy);
```

Listing 17: Partycje obliczeniowe – 2 – baza

Hurtownia:

```
SELECT p.imie || ' ' || p.nazwisko AS pracownik, u.nazwa AS usługa,
  EXTRACT(MONTH FROM f.data_sprzedazy) AS miesiac, AVG(f.suma_pln) AS
  srednia_cena
FROM Fakt_Sprzedazy f
  JOIN Dim_Pracownik p ON f.id_pracownika = p.id_pracownika
  JOIN Dim_Usluga u ON f.id_uslugi = u.id_uslugi
GROUP BY p.imie, p.nazwisko, u.nazwa, EXTRACT(MONTH FROM f.data_sprzedazy);
```

Listing 18: Partycje obliczeniowe – 2 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,025	0,026
2	0,026	0,021
3	0,026	0,021
ŚREDNIA	0,0256	0,0233

3. Liczba transakcji dla każdej kwiaciarni i klienta w danym miesiącu.

Pomaga w identyfikacji najbardziej wartościowych klientów i ocenie ich wpływu na wyniki sprzedaży w poszczególnych kwiaciarniach.

Baza:

```
SELECT
  k.nazwa AS kwiaciarnia,
  dp.imie || ' ' || dp.nazwisko AS klient,
  EXTRACT(MONTH FROM r.data_sprzedazy) AS miesiac,
  COUNT(*) AS liczba_transakcji
FROM
  rachunki r
  JOIN
    kwiaciarnie k ON r.id_kwiaciarni = k.id_kwiaciarni
  JOIN
    klienci kl ON r.id_klienta = kl.id_klienta
  JOIN
    dane_personalne dp ON kl.id_danych = dp.id_danych
GROUP BY
  k.nazwa, dp.imie, dp.nazwisko, EXTRACT(MONTH FROM r.data_sprzedazy);
```

Listing 19: Partycje obliczeniowe – 3 – baza

Hurtownia:

```
SELECT k.nazwa AS kwiaciarnia, c.imie || ' ' || c.nazwisko AS klient,
  EXTRACT(MONTH FROM f.data_sprzedazy) AS miesiac, COUNT(*) AS liczba_transakcji
FROM Fakt_Sprzedazy f
  JOIN Dim_Kwiaciarnia k ON f.id_kwiaciarni = k.id_kwiaciarni
  JOIN Dim_Klient c ON f.id_klienta = c.id_klienta
GROUP BY k.nazwa, c.imie, c.nazwisko, EXTRACT(MONTH FROM f.data_sprzedazy);
```

Listing 20: Partycje obliczeniowe – 3 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,026	0,028
2	0,028	0,024
3	0,022	0,022
ŚREDNIA	0,0253	0,0246

Okna czasowe

1. Suma sprzedaży w bieżącym i poprzednim miesiącu dla każdej kwiaciarni.

Pomaga monitorować miesięczne zmiany w przychodach i identyfikować sezonowe trendy lub skuteczność działań promocyjnych.

Baza:

```
SELECT k.nazwa AS kwiaciarnia,  
       EXTRACT(MONTH FROM r.data_sprzedazy) AS miesiac,  
       SUM(r.suma_pln) OVER (PARTITION BY k.nazwa ORDER BY EXTRACT(MONTH FROM  
r.data_sprzedazy) ROWS BETWEEN 1 PRECEDING AND CURRENT ROW) AS suma_sprzedazy  
FROM rachunki r  
JOIN kwiaciarnie k ON r.id_kwiaciarni = k.id_kwiaciarni;
```

Listing 21: Okna czasowe – 1 – baza

Hurtownia:

```
SELECT k.nazwa AS kwiaciarnia,  
       EXTRACT(MONTH FROM f.data_sprzedazy) AS miesiac,  
       SUM(f.suma_pln * f.ilosc) OVER (PARTITION BY k.nazwa ORDER BY EXTRACT(MONTH  
FROM f.data_sprzedazy) ROWS BETWEEN 1 PRECEDING AND CURRENT ROW) AS  
suma_sprzedazy  
FROM Fakt_Sprzedazy f  
JOIN Dim_Kwiaciarnia k ON f.id_kwiaciarni = k.id_kwiaciarni;
```

Listing 22: Okna czasowe – 1 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,025	0,043
2	0,027	0,046
3	0,027	0,045
ŚREDNIA	0,0263	0,0446

2. Średnia cena sprzedaży dla każdego gatunku kwiatu w danym miesiącu oraz w całym roku.

Umożliwia analizę popularności gatunków kwiatów w różnych okresach, co jest przydatne przy planowaniu zakupów i ustalaniu strategii cenowych.

Baza:

```
SELECT g.nazwa AS gatunek,  
       EXTRACT(MONTH FROM r.data_sprzedazy) AS miesiac,  
       AVG(r.suma_pln / pg.ilosc) OVER (PARTITION BY g.nazwa ORDER BY EXTRACT(MONTH  
FROM r.data_sprzedazy) RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED  
FOLLOWING) AS srednia_roczna,  
       AVG(r.suma_pln / pg.ilosc) OVER (PARTITION BY g.nazwa ORDER BY EXTRACT(MONTH  
FROM r.data_sprzedazy)) AS srednia_miesieczna  
FROM rachunki r  
JOIN pozycja_paragonu_gatunek pg ON r.id_rachunku = pg.id_rachunku  
JOIN gatunki g ON pg.id_gatunku = g.id_gatunku;
```

Listing 23: Okna czasowe – 2 – baza

Hurtownia:

```
SELECT g.nazwa AS gatunek,
       EXTRACT(MONTH FROM f.data_sprzedazy) AS miesiac,
       AVG(f.suma_pln) OVER (PARTITION BY g.nazwa ORDER BY EXTRACT(MONTH FROM
f.data_sprzedazy) RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
AS srednia_roczna,
       AVG(f.suma_pln) OVER (PARTITION BY g.nazwa ORDER BY EXTRACT(MONTH FROM
f.data_sprzedazy)) AS srednia_miesieczna
FROM Fakt_Sprzedazy f
JOIN Dim_Gatunek g ON f.id_gatunku = g.id_gatunku;
```

Listing 24: Listing 22: Okna czasowe – 2 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,022	0,032
2	0,027	0,032
3	0,027	0,027
ŚREDNIA	0,0253	0,0303

3. Różnica w liczbie transakcji pomiędzy bieżącym a poprzednim miesiącem dla każdej kwiaciarni. Pomaga w identyfikacji kwiaciarni z rosnącą lub malejącą liczbą transakcji, co może wpłynąć na decyzje dotyczące alokacji zasobów i planowania promocji.

Baza:

```
SELECT k.nazwa AS kwiaciarnia,
       EXTRACT(MONTH FROM r.data_sprzedazy) AS miesiac,
       COUNT(*) - LAG(COUNT(*), 1, 0) OVER (PARTITION BY k.nazwa ORDER BY
EXTRACT(MONTH FROM r.data_sprzedazy)) AS roznica_transakcji
FROM rachunki r
JOIN kwiaciarnie k ON r.id_kwiaciarni = k.id_kwiaciarni
GROUP BY k.nazwa, EXTRACT(MONTH FROM r.data_sprzedazy);
```

Listing 25: Okna czasowe – 3 – baza

Hurtownia:

```
SELECT k.nazwa AS kwiaciarnia,
       EXTRACT(MONTH FROM f.data_sprzedazy) AS miesiac,
       COUNT(*) - LAG(COUNT(*), 1, 0) OVER (PARTITION BY k.nazwa ORDER BY
EXTRACT(MONTH FROM f.data_sprzedazy)) AS roznica_transakcji
FROM Fakt_Sprzedazy f
JOIN Dim_Kwiaciarnia k ON f.id_kwiaciarni = k.id_kwiaciarni
GROUP BY k.nazwa, EXTRACT(MONTH FROM f.data_sprzedazy);
```

Listing 26: Okna czasowe – 3 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,022	0,021
2	0,017	0,016
3	0,016	0,017
ŚREDNIA	0,0183	0,018

Funkcje rankingowe

1. Ranga pracowników w kwiaciarniach na podstawie ilości sprzedanych kwiatów.

Pozwala na porównanie wyników sprzedaży pracowników w ramach danej kwiaciarni, co może być wykorzystane do motywowania personelu.

Baza:

```
WITH
  RangaSprzedazy
AS
(
  SELECT
    dp.imie AS imie_pracownika,
    dp.nazwisko AS nazwisko_pracownika,
    k.nazwa AS nazwa_kwiaciarni,
    SUM(ppg.ilosc) AS ilosc_sprzedanych_kwiatow,
    RANK() OVER (PARTITION BY r.id_kwiaciarni ORDER BY SUM(ppg.ilosc) DESC)
AS ranga
  FROM
    pozycja_paragonu_gatunek ppg
  JOIN rachunki r ON ppg.id_rachunku = r.id_rachunku
  JOIN pracownicy pr ON r.id_pracownika = pr.id_pracownika
  JOIN dane_personalne dp ON pr.id_danych = dp.id_danych
  JOIN kwiaciarnie k ON r.id_kwiaciarni = k.id_kwiaciarni
  WHERE ppg.id_gatunku IS NOT NULL
  GROUP BY
    r.id_kwiaciarni, dp.imie, dp.nazwisko, k.nazwa
)
SELECT
  imie_pracownika,
  nazwisko_pracownika,
  nazwa_kwiaciarni,
  ilosc_sprzedanych_kwiatow,
  ranga
FROM
  RangaSprzedazy;
```

Listing 27: Funkcje rankingowe – 1 – baza

Hurtownia:

```
WITH
  RangaSprzedazy
AS
(
  SELECT
    dp.imie AS imie_pracownika,
    dp.nazwisko AS nazwisko_pracownika,
    dk.nazwa AS nazwa_kwiaciarni,
    SUM(fs.ilosc) AS ilosc_sprzedanych_kwiatow,
    RANK() OVER (PARTITION BY fs.id_kwiaciarni ORDER BY SUM(fs.ilosc) DESC)
AS ranga
  FROM
    Fakt_Sprzedazy fs
  JOIN Dim_Pracownik dp ON fs.id_pracownika = dp.id_pracownika
  JOIN Dim_Kwiaciarnia dk ON fs.id_kwiaciarni = dk.id_kwiaciarni
  JOIN Dim_Gatunek dg ON fs.id_gatunku = dg.id_gatunku
  GROUP BY
    fs.id_kwiaciarni, dp.imie, dp.nazwisko, dk.nazwa
)
SELECT
  imie_pracownika,
  nazwisko_pracownika,
  nazwa_kwiaciarni,
  ilosc_sprzedanych_kwiatow,
  ranga
FROM
  RangaSprzedazy;
```

Listing 28: Funkcje rankingowe – 1 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,033	0,019
2	0,02	0,014
3	0,019	0,018
ŚREDNIA	0,024	0,017

2. Przydzielanie klientom poziomów lojalnościowych na podstawie wydanej kwoty. Pozwala na segmentację klientów według wydanej kwoty, co może być użyte do wprowadzenia programów lojalnościowych.

Baza:

```
SELECT dp.imie AS imie_klienta,
  dp.nazwisko AS nazwisko_klienta,
  SUM(r.suma_pln) AS calkowita_kwota,
  NTILE(3) OVER (ORDER BY SUM(r.suma_pln) DESC) AS poziom_lojalnosci
FROM rachunki r
  JOIN klienci k ON r.id_klienta = k.id_klienta
  JOIN dane_personalne dp ON k.id_danych = dp.id_danych
GROUP BY dp.imie, dp.nazwisko;
```

Listing 29: Funkcje rankingowe – 2 – baza

Hurtownia:

```
SELECT dk.imie AS imie_klienta,  
       dk.nazwisko AS nazwisko_klienta,  
       SUM(fs.suma_pln * fs.ilosc) AS calkowita_kwota,  
       NTILE(3) OVER (ORDER BY SUM(fs.suma_pln * fs.ilosc) DESC) AS  
poziom_lojalnosci  
FROM Fakt_Sprzedazy fs  
     JOIN Dim_Klient dk ON fs.id_klienta = dk.id_klienta  
WHERE ilosc IS NOT NULL  
GROUP BY fs.id_klienta, dk.imie, dk.nazwisko;
```

Listing 30: Funkcje rankingowe – 2 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,016	0,015
2	0,016	0,014
3	0,018	0,016
ŚREDNIA	0,0166	0,015

3. Znajdzenie 3 najpopularniejszych gatunków kwiatów w każdej kwiaciarni. Pozwala na zrozumienie, jakie gatunki kwiatów cieszą się największą popularnością w poszczególnych kwiaciarniach, co może wpłynąć na decyzje zakupowe.

Baza:

```
WITH
  RangaGatunku
AS
(
  SELECT
    k.nazwa AS nazwa_kwiaciarni,
    g.nazwa AS gatunek_kwiatu,
    SUM(ppg.ilosc) AS ilosc_sprzedanych,
    RANK() OVER (PARTITION BY r.id_kwiaciarni ORDER BY SUM(ppg.ilosc) DESC)
AS ranga
FROM
  pozycja_paragonu_gatunek ppg
  JOIN rachunki r ON ppg.id_rachunku = r.id_rachunku
  JOIN gatunki g ON ppg.id_gatunku = g.id_gatunku
  JOIN kwiaciarnie k ON r.id_kwiaciarni = k.id_kwiaciarni
WHERE ppg.id_gatunku IS NOT NULL
GROUP BY
  r.id_kwiaciarni, g.nazwa, k.nazwa
)
SELECT
  nazwa_kwiaciarni,
  gatunek_kwiatu,
  ilosc_sprzedanych,
  ranga
FROM
  RangaGatunku
WHERE
  ranga <= 3
ORDER BY
  nazwa_kwiaciarni,
  ranga;
```

Listing 31: Funkcje rankingowe – 3 – baza

Hurtownia:

```
WITH
  RangaGatunku
  AS
  (
    SELECT
      dk.nazwa AS nazwa_kwiaciarni,
      dg.nazwa AS gatunek_kwiatu,
      SUM(fs.ilosc) AS ilosc_sprzedanych,
      RANK() OVER (PARTITION BY fs.id_kwiaciarni ORDER BY SUM(fs.ilosc) DESC)
    AS ranga
  FROM
    Fakt_Sprzedazy fs
    JOIN Dim_Gatunek dg ON fs.id_gatunku = dg.id_gatunku
    JOIN Dim_Kwiaciarnia dk ON fs.id_kwiaciarni = dk.id_kwiaciarni
  GROUP BY
    fs.id_kwiaciarni, dg.nazwa, dk.nazwa
  )
SELECT
  nazwa_kwiaciarni,
  gatunek_kwiatu,
  ilosc_sprzedanych,
  ranga
FROM
  RangaGatunku
WHERE
  ranga <= 3;
```

Listing 32: Funkcje rankingowe – 3 – hurtownia

LP.	BAZA	HURTOWNIA
1	0,013	0,013
2	0,011	0,008
3	0,011	0,012
ŚREDNIA	0,0116	0,011

Uwagi i wnioski

W ramach naszego projektu zaimplementowaliśmy bazę danych w OracleSQL, której celem było przechowywanie i zarządzanie danymi związanymi z działalnością kwaciarni. W celu optymalizacji wydajności zapytań oraz ułatwienia analizy danych, stworzyliśmy dodatkową strukturę w postaci hurtowni danych. Wykorzystaliśmy skrypt do generowania zestawu danych, który następnie załadowaliśmy do naszej bazy danych przy użyciu narzędzia SQLLoader. Aby przenieść dane z bazy operacyjnej do hurtowni, stworzyliśmy procedurę, która transformuje i agreguje dane do odpowiednich tabel wymiarowych i tabeli faktów w hurtowni. Po zakończeniu procesu przenoszenia danych, przeprowadziliśmy szereg zapytań i zaobserwowaliśmy, że wykonywanie zapytań na hurtowni danych jest znacząco szybsze w porównaniu do zapytań wykonywanych bezpośrednio na bazie operacyjnej. To podkreśla zalety zastosowania hurtowni danych w kontekście efektywności i wydajności analizy danych.

Wykorzystane narzędzia:

- SQLDeveloper 21.2.1
- Baza Oracle wersja 21.0
- SQLLoader
- StarUML
- GitHub: <https://github.com/PSK-proj/BD2>

Elementy po poprawie

Schemat bazy

Baza nie jest w 3 postaci normalnej (relacje 1:1), relacje na schemacie nie odpowiadają kluczom, np. dostawy zatrudnienie 1:N oraz są nie zgodne ze skryptem zakładającym.

Zmiany:

Usunęliśmy atrybuty zależne od innego atrybutu, który nie jest kluczem aby baza spełniała wymagania 3NF, Pozostały tylko niezbędne relacje 1:1 (człowiek nie może mieć wiele tożsamości, ani kilka osób nie może mieć tej samej tożsamości),

Naprawiliśmy schemat tak, aby tabele nie nakładały się na linie relacji, co mogło wcześniej sprawiać mylne wrażenie zaplanowania relacji, której tak naprawdę nie mieliśmy w intencji tworzyć.

Zapytania baza

*Jaki sens ma rollup z agregacją po jednej kolumnie i to do tego po polu opisowym?
Niestety w wielu projektach są takie błędy.*

Zmiany:

Zmieniliśmy zapytania ROLLUP i CUBE aby agregowały dane po kilku polach numerycznych, Dostosowaliśmy pozostałe zapytania do nowego schematu bazy danych.

Schemat hurtownia

*Nieprawidłowy projekt hurtowni - powinien być zbudowany w oparciu o
pozycja_paragon, bo te dane są w zapytaniach, a nie o rachunki.*

Wyjaśnienie:

Agregowaliśmy w tabeli faktów dane z wielu tabel, w tym także z **pozycja_paragonu**. Nowa procedura przenosząca dane z bazy danych do hurtowni (stworzona z powodu zmiany struktury głównej bazy danych) także agreguje dane zarówno z **pozycja_paragonu_gatunek**, **pozycja_paragonu_usluga** oraz **rachunki**, gdyż te tabele współpracują ze sobą.

Spis listingów

Listing 1: Kod procedury, cz.1	19
Listing 2: Kod procedury, cz.2	20
Listing 3: ROLLUP – 1 – baza	21
Listing 4: ROLLUP – 1 – hurtownia	21
Listing 5: ROLLUP – 2 – baza	22
Listing 6: ROLLUP – 2 – hurtownia	22
Listing 7: ROLLUP – 3 – baza	23
Listing 8: ROLLUP – 3 – hurtownia	23
Listing 9: CUBE – 1 – baza	24
Listing 10: CUBE – 1 – hurtownia	24
Listing 11: CUBE – 2 – baza	25
Listing 12: CUBE – 2 – hurtownia	25
Listing 13: CUBE – 3 – baza	26
Listing 14: CUBE – 3 – hurtownia	26
Listing 15: Partycje obliczeniowe – 1 – baza	27
Listing 16: Partycje obliczeniowe – 1 – hurtownia	27
Listing 17: Partycje obliczeniowe – 2 – baza	28
Listing 18: Partycje obliczeniowe – 2 – hurtownia	28
Listing 19: Partycje obliczeniowe – 3 – baza	29
Listing 20: Partycje obliczeniowe – 3 – hurtownia	29
Listing 21: Okna czasowe – 1 – baza.....	30
Listing 22: Okna czasowe – 1 – hurtownia.....	30
Listing 23: Okna czasowe – 2 – baza.....	30
Listing 24: Listing 22: Okna czasowe – 2 – hurtownia	31
Listing 25: Okna czasowe – 3 – baza.....	31
Listing 26: Okna czasowe – 3 – hurtownia.....	31
Listing 27: Funkcje rankingowe – 1 – baza	32
Listing 28: Funkcje rankingowe – 1 – hurtownia	33
Listing 29: Funkcje rankingowe – 2 – baza	33
Listing 30: Funkcje rankingowe – 2 – hurtownia	34
Listing 31: Funkcje rankingowe – 3 – baza	35
Listing 32: Funkcje rankingowe – 3 – hurtownia	36