

Uniwersytet Gdański



BAZY DANYCH

Projekt: Komis samochodowy

Prowadzący:

dr Andrzej Borzyszkowski;
mgr Michał Zakrzewski;
dr Adam Kostulak;
dr Tomasz Borzyszkowski

Wykonała:

Paulina Kimak
Nr albumu 292511

Gdańsk, 2024

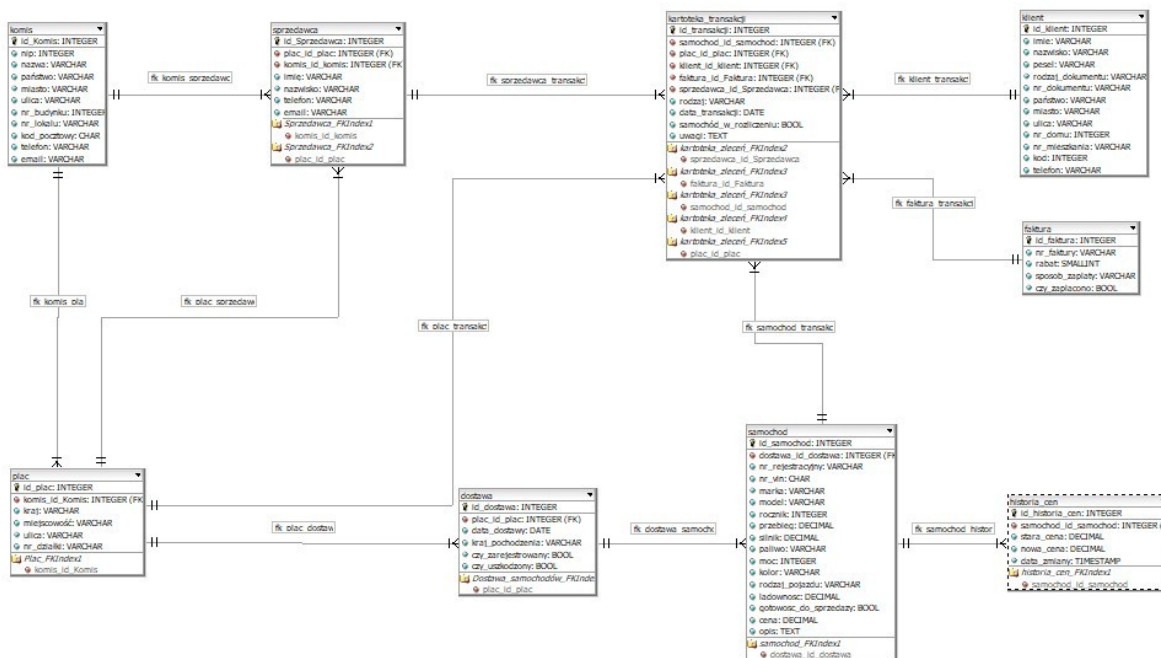
Spis treści

1. Opis Projektu.....	3
1. Schemat ERD bazy danych dla komis samochodowego.....	3
2. Encje.....	4
3. Atrybuty encji.....	4
4. Związki encji:.....	7
5. Klucze.....	8
2. Fragmenty tabel.....	9
3. Przykłady zapytań.....	10
1. Zapytanie 1 – SELECT.....	10
2. Zapytanie 2 – SELECT.....	11
3. Zapytanie 3 – SELECT.....	11
4. Zapytanie 4 – SELECT.....	12
5. Zapytanie 5 – SELECT.....	12
6. Zapytanie 6 – SELECT.....	13
7. Zapytanie 7 – SELECT.....	14
8. Zapytanie 8 – SELECT.....	14
9. Zapytanie 9 – SELECT.....	15
10. Zapytanie 10 – CREATE VIEW.....	15
11. Zapytanie 11 – CREATE VIEW.....	16
12. Zapytanie 12 - UPDATE.....	17
13. Zapytanie 13 - UPDATE.....	17
14. Zapytanie 13 - DELETE.....	18
4. Trigery.....	18
1. Opis triggerów:.....	18
2. Trigger 1.....	19
Założenia:.....	19
3. Trigger 2.....	21

1. Opis Projektu

Celem projektu było stworzenie bazy danych dla komisów samochodowych. Dzięki zaprojektowanej bazie danych możliwa jest kontrola sprzedaży/zakupów samochodów różnego rodzaju od osób fizycznych jak i i firm. W bazie znajdziemy m.in. aktualne informacje o samochodach które są wystawione na sprzedaż oraz zrealizowanych zamówieniach klientów.

1. Schemat ERD bazy danych dla komisów samochodowych



2. Encje

W bazie danych znajduje się osiem encji:

1. **komis** - posiada informacje o komisie (Tabela 1)
2. **plac** - posiada informacje placach, które należą do komisum (Tabela 2)
3. **sprzedawca**- posiada informacje o sprzedawcach zatrudnionych w komisie (Tabela 3)
4. **samochód**- posiada informacje o samochodach na stanie komisum (Tabela 4)
5. **dostawa** – posiada informacje dostawie samochodów (Tabela 5)
6. **klient** - posiada informacje o klientach komisum (Tabela 6)
7. **kartoteka_transakcji** - posiada informacje o dokonanych transakcjach (Tabela 7)
8. **faktura** – posiada informacje dotyczące płatności (Tabela 8)
9. **historia_cen** – posiada informacje historii zmian cen wybranych samochodów (Tabela 9)

3. Atrybuty encji

Tabela 1. **komis**

Nazwa pola:	Typ:	
id_komis	serial	PRIMARY KEY
nip	int	not null, UNIQUE
nazwa	varchar(30)	not null
panstwo	varchar(30)	not null
miasto	varchar(30)	not null
ulica	varchar(30)	not null
nr_budynku	int	not null
nr_lokalu	varchar(10)	
kod_pocztowy	char(6)	not null
telefon	varchar(15)	
email	varchar(30)	

Tabela 2. **plac**

Nazwa pola:	Typ:	
id_plac	serial	PRIMARY KEY
kraj	varchar(30)	not null
miescowosc	varchar(30)	not null
ulica	varchar(30)	not null
nr_dzialki	varchar(20)	not null
id_komis	int	FOREIGN KEY

Tabela 3. sprzedawca

Nazwa pola:	Typ:	
id_sprzedawca	serial	PRIMARY KEY
imie	varchar(30)	not null
nazwisko	varchar(30)	not null
nr_telefonu	varchar(30)	
email	varchar(50)	
id_komis	int	FOREIGN KEY
id_plac	int	FOREIGN KEY

Tabela 4. samochod

Nazwa pola:	Typ:	
id_samochod	serial	PRIMARY KEY
nr_rejestracyjny	varchar(7)	UNIQUE
nr_vin	char(17)	not null, UNIQUE
marka	varchar(30)	not null
model	varchar(30)	not null
rocznik	int	not null
przebieg	decimal(12,2)	not null
silnik	decimal(4,2)	not null
paliwo	varchar(30)	not null
moc	int	not null
kolor	varchar(30)	not null
rodzaj_pojazdu	varchar(30)	not null
ladownosc	decimal(4,2)	
gotowy_do_przedazy	bool	not null
cena	decimal(20,2)	not null
opis	text	
id_plac	int	FOREIGN KEY

Tabela 5. dostawa

Nazwa pola:	Typ:	
id_dostawa	serial	PRIMARY KEY
data_dostawy	date	not null
kraj_pochodzenia	varchar(0)	not null
czy_zarejestrowany	bool	not null
czy_uszkodzony	bool	not null
id_plac	int	FOREIGN KEY
id_samochod	int	FOREIGN KEY

Tabela 6. klient

Nazwa pola:	Typ:	
id_klient	serial	PRIMARY KEY
imie	varchar(30)	not null
nazwisko	varchar(30)	not null
pesel	varchar(11)	not null, UNIQUE
rodzaj_dokumentu	varchar(50)	not null
nr_dokumentu	varchar(30)	not null
panstwo	varchar(50)	
miasto	varchar(50)	
ulica	varchar(30)	
nr_domu	int	
nr_mieszkania	varchar(10)	
kod	varchar(00)	
telefon	varchar(30)	

Tabela 7. kartoteka transakcji

Nazwa pola:	Typ:	
id_transakcji	serial	PRIMARY KEY
rodzaj	varchar(15)	not null
data_transakcji	date	not null
samochod_w_rozliczeniu	bool	not null
uwagi	text	
id_samochod	int	FOREIGN KEY
id_klient	int	FOREIGN KEY
id_sprzedawca	int	FOREIGN KEY
id_plac	int	FOREIGN KEY
id_faktura	int	FOREIGN KEY

Tabela 8. faktura

Nazwa pola:	Typ:	
id_faktura	serial	PRIMARY KEY
nr_faktury	varchar(10)	not null, UNIQUE
rabat	smallint	
sposob_zaplaty	varchar(30)	not null
czy_zaplacono	bool	null

Tabela 9. historia cen

Nazwa pola:	Typ:	
id_historia	serial	PRIMARY KEY
id_samochod	int	not null
stara_cena	decimal(20,2)	
nowa_cena	decimal(20,2)	not null
data_zmiany	timestamp	not null DEFAULT CURRENT_TIMESTAMP

4. Związki encji:

Numer	Nazwa związku	Encja 1	Encja 2	Łączność związku	Opis
1	fk_komis_plac	komis	plac	1:n	Komis może posiadać wiele placów w różnych lokalizacjach
2	fk_komis_sprzedawca	komis	sprzedawca	1:n	Komis może zatrudniać wielu sprzedawców.
3	fk_plac_sprzedawca	plac	sprzedawca	1:n	Na placu może pracować wielu sprzedawców
4	fk_plac_dostawa	plac	dostawa	1:n	Na jeden plac może być wiele dostaw.
5	fk_dostawa_samochod	dostawa	samochod	1:n	W jednej dostawie może być dostarczonych kilka samochodów.
6	fk_samochod_transakcja	samochod	transakcja	1:n	Samochód może być przedmiotem wielu transakcji.
7	fk_klient_transakcja	klient	transakcja	1:n	Klient może dokonać wiele transakcji.
8	fk_plac_transakcja	plac	transakcja	1:n	Na jednym placu może być dokonanych wiele transakcji.
9	fk_sprzedawca_transakcja	sprzedawca	transakcja	1:n	Sprzedawca może dokonać wielu transakcji.
10	fk_faktura_transakcja	faktura	transakcja	1:n	Na fakturze może być wiele transakcji
11	fk_samochod_historia	samochod	historia	1:n	Samochod może mieć wiele zmian w historii.

5. Klucze

- **komis** (id_komis, nip, nazwa, panstwo, miasto, ulica, nr_budynku, nr_lokalu, kod_pocztowy, telefon, e_mail)
(id_komis) **PRIMARY KEY**
- **plac** (id_plac, kraj, miejscowosc, ulica, nr_dzialki, id_komis)
(id_plac) **PRIMARY KEY**
(id_komis) **FOREIGN KEY REF** komis
- **samochod** (id_samochod, nr_rejestracyjny, nr_vin, marka, model, rocznik, przebieg, silnik, paliwo, moc, kolor, rodzaj_pojazdu, ladownosc, gotowy_do_sprzedazy, opis, id_plac)
(id_samochod) **PRIMARY KEY**
(id_plac) **FOREIGN KEY REF** plac
- **dostawa** (id_dostawa, data_dostawy, kraj_pochodzenia, czy_zarejestrowany, czy_uszkodzony, id_plac, id_samochod)
(id_dostawa) **PRIMARY KEY**
(id_plac) **FOREIGN KEY REF** plac
(id_samochod) **FOREIGN KEY REF** samochód
- **sprzedawca** (id_sprzedawca, imie, nazwisko, nr_telefonu, e_mail, id_komis, id_plac)
(id_sprzedawca) **PRIMARY KEY**
(id_komis) **FOREIGN KEY REF** komis
(id_plac) **FOREIGN KEY REF** plac
- **faktura** (id_faktura, nr_faktury, kwota, waluta, przelicznik_waluty, rabat, sposob_zaplaty, czy_zaplacono)
(id_faktura) **PRIMARY KEY**
- **klient** (id_klient, imie, nazwisko, pesel_nip, rodzaj_dokumentu, nr_dokumentu, panstwo, miasto, ulica, nr_domu, nr_mieszkania, kod, telefon)
(id_klient) **PRIMARY KEY**
- **kartoteka_transakcji** (id_transakcja, rodzaj, data_transakcji, samochod_w_rozliczeniu, uwagi, id_samochod, id_klient, id_sprzedawca, id_plac, id_faktura)
(id_transakcja) **PRIMARY KEY**
(id_samochod) **FOREIGN KEY REF** samochód
(id_klient) **FOREIGN KEY REF** klient

(id_plac) **FOREIGN KEY REF** plac

(id_sprzedawca) **FOREIGN KEY REF** sprzedawca

(id_faktura) **FOREIGN KEY REF** faktura

- **historia_zmian** (id_historia, id_samochod, stara_cena, nowa_cena, data_zmiany)
- (id_historia) **PRIMARY KEY**
- (id_samochod) **FOREIGN KEY REF** samochód

2. Fragmenty tabel

Komis tabela 1

	id_komis [PK] integer	nip integer	nazwa character varying (30)	panstwo character varying (30)	miasto character varying (30)	ulica character varying (30)	nr_budynku character varying (50)	nr_lokalu character varying (50)	kod_pocztowy character varying (30)	telefon character varying (30)	e_mail character varying (30)
1	1	1234567890	AutoKomis XYZ	Polska	Warszawa	Marszałkowska	123	4A	00-001	123-456-789	kontakt@autokomisxyz.pl

Plac tabela 2

	id_plac [PK] integer	kraj character varying (30)	miescowosc character varying (30)	ulica character varying (30)	nr_dzialki character varying (20)	id_komis integer
1	1	Polska	Kraków	Floriańska	10	1
2	2	Polska	Gdańsk	Długa	25	1
3	3	Polska	Poznań	Święty Marcin	8	1

Sprzedawca tabela 3

	id_sprzedawca [PK] integer	imie character varying (30)	nazwisko character varying (30)	nr_telefonu character varying (30)	e_mail character varying (50)	id_komis integer	id_plac integer
1	1	Jan	Kowalski	123-456-789	jan.kowalski@example.com	1	1
2	2	Anna	Nowak	234-567-890	anna.nowak@example.com	1	1
3	3	Piotr	Wiśniewski	345-678-901	piotr.wisniewski@example.com	1	1

Samochód tabela 4

	id_samochod [PK] integer	nr_rejestracji character varying (30)	nr_vin character varying (30)	marka character varying (30)	model character varying (30)	rocznik integer	przebieg numeric (12,2)	silnik numeric (4,2)	paliwo character varying (30)	moc integer	kolor character varying (30)	rodzaj_pojazdu character varying (30)	ladownosc numeric (4,2)	gotowy_do boolean	cena numeric (20,2)	opis text	id_plac integer
1	1	ABC1234	1HGCM8...	Toyota	Corolla	2015	80000.00	1.80	Benzyna	140	Czarny	Sedan	5.00	true	15000.00	Polski sal...	1
2	2	DEF5678	2HGCM8...	Honda	Civic	2018	60000.00	2.00	Benzyna	158	Biały	Sedan	4.50	false	145999.00	Dodatk...	2
3	3	GHI9101	3HGCM8...	Ford	Focus	2020	30000.00	1.50	Diesel	120	Niebieski	Hatchback	4.00	true	65000.00	Brak drug...	1

Dostawa tabela nr 5

	id_dostawa [PK] integer	data_dostawy date	kraj_pochodzenia character varying (50)	czy_zarejestrowany boolean	czy_uszkodzony boolean	id_plac integer	id_samochod integer
1	1	2022-01-05	Polska	true	false	1	1
2	2	2022-02-12	Niemcy	false	true	2	2
3	3	2022-03-20	Polska	true	false	3	3
4	4	2022-04-15	Francja	false	true	1	4

Klient tabela nr 6

	id_klient [PK] integer	imie character varying (30)	nazwisko character varying (30)	pesel character varying (11)	rodzaj_dokumentu character varying (30)	nr_dokumentu character varying (30)	panstwo character varying (30)	miasto character varying (30)	ulica character varying (30)	nr_domu integer	nr_mieszkania integer	kod character varying (30)	telefon character varying (30)
1	1	Jan	Kowalski	123456789...	Dowód osobisty	ABC123456	Polska	Warszawa	Krakowska	10	2	00-001	123456789
2	2	Anna	Nowak	987654321...	Dowód osobisty	DEF987654	Polska	Kraków	Piotrkows...	5	[null]	30-002	987654321
3	3	Piotr	Wiśniewski	345678901...	Paszport	GHI654321	Polska	Gdańsk	Stonieczna	15	4	80-003	654321987

Kartoteka transakcji tabela nr 7

	id_transaksija [PK] integer	rodzaj character varying (15)	data_transaksiji date	samochod_w_rozliczeniu boolean	uwagi text	id_samochod integer	id_klient integer	id_sprzedawca integer	id_plac integer	id_faktura integer
1	1	sprzedaż	2024-01-10	false	Dodać opony zimowe	1	7	1	1	1
2	2	sprzedaż	2024-01-15	true	Samochód kupiony z rabatem	2	10	2	1	2
3	3	sprzedaż	2024-02-05	false	Polerka przed wydaniem	3	3	3	1	3

Faktura tabela nr 8

	id_faktura [PK] integer	nr_faktury character varying (10)	rabat smallint	sposob_zaplaty character varying (30)	czy_zaplacono boolean
1	1	F/2024/001	5	Przelew	true
2	2	F/2024/002	[null]	Gotówka	true
3	3	F/2024/003	10	Karta kredytowa	true

Historia zmian tabela nr 9

	id_historia [PK] integer	id_samochod integer	stara_cena numeric (20,2)	nowa_cena numeric (20,2)	data_zmiany timestamp without time zone
1	1	1	15000.00	80000.00	2024-06-02 13:51:28.586066
2	2	16	85000.00	80000.00	2024-06-02 14:02:31.444755

3. Przykłady zapytań

1. Zapytanie 1 – SELECT

OPIS: Wyświetl wszystkie samochody, które pochodzą z Polski

WYMAGANIE: Użycie zagnieżdżenia w obu odmianach, nieskorelowane i skorelowane.

ZAPYTANIE:

```
SELECT
    dos.kraj_pochodzenia,
    s.*
FROM samochod as s
JOIN dostawa as dos ON dos.id_samochod=s.id_samochod
WHERE s.id_samochod IN (SELECT d.id_samochod FROM dostawa AS d WHERE
kraj_pochodzenia = 'Polska')
```

WYNIK ZAPYTANIA:

	kraj_pochodzenia character varying (50)	id_samochod integer	nr_rejestracyjny character varying (7)	nr_vin character	marka character varying (30)	model character varying (30)	rocznik integer	przebieg numeric (12,2)	silnik numeric (4,2)	paliwo character varying (30)	moc integer
1	Polska	1	ABC1234	1HGCM82633A123456	Toyota	Corolla	2015	80000.00	1.80	Benzyna	140
2	Polska	3	GHI9101	3HGCM82633A987654	Ford	Focus	2020	30000.00	1.50	Diesel	120
3	Polska	5	MNO6789	5HGCM82633A112233	BMW	3 Series 4x4	2017	70000.00	2.00	Diesel	190
4	Polska	7	STU3344	7HGCM82633A223344	Mercedes	C-Class	2018	65000.00	2.00	Diesel	180
5	Polska	9	YZA7788	9HGCM82633A667788	Volvo	S60 4x4	2016	85000.00	2.00	Diesel	190
6	Polska	11	EFG1235	1HGCM82633B112233	Opel	Astra	2020	25000.00	1.60	Benzyna	130
7	Polska	13	KLM7890	3HGCM82633B778899	Peugeot	308	2018	55000.00	1.20	Benzyna	130
8	Polska	15	QRC4456	5HGCM82633C445566	Kia	Ceed	2017	70000.00	1.60	Diesel	136

2. Zapytanie 2 – SELECT

OPIS: Wyświetl samochody zarejestrowane, które mają napęd 4x4

WYMAGANIE: Warunki odwołujące się do wzorców napisów (np. opisy wszystkich układanek, LIKE 'układanka%').

ZAPYTANIE:

```
SELECT
    s.nr_vin,
    s.marka,
    s.model,
    s.rocznik
FROM samochod AS s WHERE model LIKE '%4x4%';
```

WYNIK ZAPYTANIA:

	nr_vin character	marka character varying (30)	model character varying (30)	rocznik integer
1	5HGCM82633A112233	BMW	3 Series 4x4	2017
2	9HGCM82633A667788	Volvo	S60 4x4	2016
3	0HGCM82633A889900	Mazda	6 4x4	2017

3. Zapytanie 3 – SELECT

OPIS: Wyświetl informacje o sprzedawcach, którzy sprzedali najwięcej samochodów

WYMAGANIE: INNER JOIN+grupowanie+funkcja agregująca count

ZAPYTANIE:

```
SELECT s.id_sprzedawca, s.imie, s.nazwisko, COUNT(k.id_samochod) AS
liczba_sprzedanych_samochodow
FROM sprzedawca AS s
JOIN kartoteka_transakcji AS k ON s.id_sprzedawca = k.id_sprzedawca
GROUP BY s.id_sprzedawca, s.imie, s.nazwisko
ORDER BY liczba_sprzedanych_samochodow DESC
--LIMIT 1;-- wybór najlepszego sprzedawcy
```

WYNIK ZAPYTANIA:

	id_sprzedawca [PK] integer	imie character varying (30)	nazwisko character varying (30)	liczba_sprzedanych_samochodow bigint
1	9	Robert	Zieliński	3
2	5	Marek	Kowalczyk	2
3	10	Agnieszka	Szymańska	1
4	1	Jan	Kowalski	1
5	8	Magdalena	Dąbrowska	1
6	4	Katarzyna	Wójcik	1
7	3	Piotr	Wiśniewski	1

4. Zapytanie 4 – SELECT

OPIS: Wyświetl samochody, które posiadają opis.

WYMAGANIE: Sprawdzanie warunku NULL

ZAPYTANIE:

```
SELECT
    s.nr_vin,
    s.marka,
    s.model,
    s.rocznik,
    s.opis
FROM samochod AS s WHERE opis IS NOT NULL;
```

WYNIK ZAPYTANIA:

	nr_vin character	marka character varying (30)	model character varying (30)	rocznik integer	opis text
1	1HGCM82633A123456	Toyota	Corolla	2015	Polski salon
2	2HGCM82633A654321	Honda	Civic	2018	Dodatkowy komplet opon
3	3HGCM82633A987654	Ford	Focus	2020	Brak drugiego kluczyka
4	6HGCM82633A332211	Audi	A4	2019	Dodatkowy komplet opon
5	8HGCM82633A445566	Skoda	Octavia	2021	Dodatkowo CB radio
6	3HGCM82633B778899	Peugeot	308	2018	Brak drugiego kluczyka
7	4HGCM82633C112233	Hyundai	Elantra	2021	Dodatkowy komplet opon
8	5HGCM82633C445566	Kia	Ceed	2017	Polski salon

5. Zapytanie 5 – SELECT

OPIS: Wyświetl klientów, którzy nie dostali żadnego rabatu

WYMAGANIE: Zapytania negatywne, np. klienci bez złożonych zamówień. Co najmniej w dwu wersjach

ZAPYTANIE:

Wersja 1 (WHERE NOT IN)

```
SELECT
    kl.imie,
    kl.nazwisko,
    kl.id_klient
FROM klient AS kl
WHERE kl.id_klient NOT IN (SELECT
    k.id_klient
FROM klient AS k
JOIN kartoteka_transakcji as t ON t.id_klient=k.id_klient
JOIN faktura AS f ON f.id_faktura=t.id_faktura
WHERE f.rabat IS NOT NULL)
ORDER BY kl.id_klient;
```

Wersja 2 (LEFT JOIN)

```
SELECT
    kl.imie,
    kl.nazwisko,
    kl.id_klient
FROM klient AS kl
LEFT JOIN kartoteka_transakcji as t ON t.id_klient=kl.id_klient
LEFT JOIN faktura AS f ON f.id_faktura=t.id_faktura
WHERE f.rabat IS NULL
ORDER BY kl.id_klient;
```

WYNIK OBU ZAPYTAŃ:

	imie character varying (30)	nazwisko character varying (30)	id_klient [PK] integer
1	Piotr	Wiśniewski	3
2	Ewa	Kamińska	6
3	Magdalena	Woźniak	8

6. Zapytanie 6 – SELECT

OPIS: Wyświetl średnią wartość rabatu udzielanego klientom w procentach i kwotowo

WYMAGANIE: Wyświetlanie wyniku operacji np. operacji arytmetycznych

ZAPYTANIE:

```
SELECT
    ROUND(AVG(rabat),2) AS sredni_rabat_procent
FROM faktura;
```

WYNIK ZAPYTANIA:

	sredni_rabat_procent numeric
1	6.13

7. Zapytanie 7 – SELECT

OPIS: Wyświetl transakcje z kartoteki zleceń z miesiąca marca

WYMAGANIE: Wyświetlanie wyniku operacji na datach.

ZAPYTANIE:

```
SELECT * FROM public.kartoteka_transakcji
      WHERE data_transaksi BETWEEN '2024-03-01' AND '2024-03-31'
ORDER BY id_transaksi ASC
```

WYNIK ZAPYTANIA:

	id_transaksi [PK] integer	rodzaj character varying (1)	data_transaksi date	samochod_w_r boolean	uwagi text	id_samochod integer	id_klient integer	id_sprzedawca integer	id_plac integer	id_faktura integer
1	5	sprzedaż	2024-03-01	true	Brak uwag		5	2	5	
2	6	sprzedaż	2024-03-15	false	Brak uwag		6	8	9	2

8. Zapytanie 8 – SELECT

OPIS: Wyświetl pozycje samochodów na konkretnej fakturze wraz z cenami

WYMAGANIE: Obecność złączenia, zarówno w składni ze słowami INNER JOIN

ZAPYTANIE:

```
SELECT
    kt.id_samochod,
    s.cena,
    f.*
FROM faktura AS f
JOIN kartoteka_transakcji AS kt ON kt.id_faktura=f.id_faktura
JOIN samochod AS s ON s.id_samochod=kt.id_samochod
WHERE f.id_faktura=1
```

WYNIK ZAPYTANIA:

	id_samochod integer	cena numeric (20,2)	id_faktura integer	nr_faktury character varying (10)	rabat smallint	sposob_zaplaty character varying (30)	czy_zaplacono boolean
1	1	15000.00	1	F/2024/001	5	Przelew	true
2	2	145999.00	1	F/2024/001	5	Przelew	true

9. Zapytanie 9 – SELECT

OPIS: Wyświetl sumę wszystkich pozycji na fakturze nr 1 (z uwzględnieniem rabatów)

WYMAGANIE: Zagnieżdzenie w klauzuli FROM + funkcje agregujące

ZAPYTANIE:

```
SELECT
    SUM(sub.cena_po_rabacie) AS suma_cen
FROM
    (SELECT
        s.cena - (s.cena * ROUND((CAST(COALESCE(f.rabat, 0) AS decimal) / 100), 2))
    AS cena_po_rabacie
    FROM faktura AS f
    JOIN kartoteka_transakcji AS kt ON kt.id_faktura = f.id_faktura
    JOIN samochod AS s ON s.id_samochod = kt.id_samochod
    WHERE f.id_faktura = 1
    ) AS sub;
```

WYNIK ZAPYTANIA:

	suma_cen numeric
1	152949.0500

10. Zapytanie 10 – CREATE VIEW

OPIS: Wyświetl tabelę faktura z ostatecznymi kwotami po rabacie.

WYMAGANIE: Użycie widoku (perspektywy, VIEW) + Wyświetlanie wyniku operacji arytmetycznych

ZAPYTANIE:

Stworzenie view:

```
CREATE VIEW kwota_po_rabacie
AS
SELECT
    kt.id_samochod,
    s.cena AS cena_samochodu,
    f.id_faktura, f.nr_faktury,
    COALESCE(f.rabat, 0) AS rabat_procenty,
    ROUND((CAST(COALESCE(f.rabat, 0) AS decimal)/100),2) AS rabat_decimal,
    s.cena * ROUND((CAST(COALESCE(f.rabat, 0) AS decimal)/100),2) AS
kwota_rabatu,
    s.cena - (s.cena * ROUND((CAST(COALESCE(f.rabat, 0) AS decimal)/100),2))
AS kwota_po_rabacie,
    f.sposob_zaplaty, f.czy_zaplacono
FROM faktura AS f
JOIN kartoteka_transakcji AS kt ON kt.id_faktura=f.id_faktura
JOIN samochod AS s ON s.id_samochod=kt.id_samochod;
FROM faktura;
```

Uruchomienie view:

```
SELECT * FROM kwota_po_rabacie;
```

WYNIK ZAPYTANIA:

	id_samochod integer	cena_samochodu numeric (20,2)	id_faktura integer	nr_faktury character varyii	rabat_procenty integer	rabat_decima numeric	kwota_rabatu numeric	kwota_po_rabacie numeric	sposob_zaplat character varyi	czy_zaplacono boolean
1	1	15000.00	1	F/2024/001	5	0.05	750.0000	14250.0000	Przelew	true
2	2	145999.00	1	F/2024/001	5	0.05	7299.9500	138699.0500	Przelew	true
3	3	65000.00	2	F/2024/002	0	0.00	0.0000	65000.0000	Gotówka	true
4	4	90500.00	2	F/2024/002	0	0.00	0.0000	90500.0000	Gotówka	true
5	5	25000.00	5	F/2024/005	4	0.04	1000.0000	24000.0000	Gotówka	true
6	6	25000.00	6	F/2024/006	0	0.00	0.0000	25000.0000	Przelew	false
7	7	12000.00	7	F/2024/007	9	0.09	1080.0000	10920.0000	Przelew	true
8	8	15000.00	8	F/2024/008	3	0.03	450.0000	14550.0000	Karta kred...	true
9	9	10000.00	9	F/2024/009	4	0.04	400.0000	9600.0000	Przelew	true
10	10	115000.00	9	F/2024/009	4	0.04	4600.0000	110400.0000	Przelew	true

11. Zapytanie 11 – CREATE VIEW

OPIS: Wyświetl pokazuje ceny po rabacie dla faktury nr 1

WYMAGANIE: Użycie grupowania i funkcji agregujących.+Użycie widoku (perspektywy, VIEW)

ZAPYTANIE:

```
CREATE VIEW zestawienie_faktura_1
AS
SELECT
    f.id_faktura,
    f.nr_faktury,
    s.id_samochod,
    SUM(s.cena -(s.cena * ROUND((CAST(COALESCE(f.rabat, 0) AS
decimal)/100),2))) AS cena_auta_po_rabacie
FROM faktura AS f
JOIN kartoteka_transakcji AS kt ON kt.id_faktura=f.id_faktura
JOIN samochod AS s ON s.id_samochod=kt.id_samochod
WHERE f.id_faktura=1
GROUP BY f.id_faktura,f.nr_faktury,s.cena, s.id_samochod
```

WYNIK ZAPYTANIA:

	id_faktura integer	nr_faktury character varying (10)	id_samochod integer	cena_auta_po_rabacie numeric
1	1	F/2024/001	1	14250.0000
2	1	F/2024/001	2	138699.0500

12. Zapytanie 12 - UPDATE

OPIS: Zmiana atrybutu gotowy_do_sprzedaży dla samochodu o id=4

WYMAGANIE: Aktualizacja atrybutu

ZAPYTANIE:

```
UPDATE samochod SET gotowy_do_sprzedaży = TRUE WHERE id_samochod = 4;
```

WYNIK ZAPYTANIA:

	paliwo character varying (30)	moc integer	kolor character varying (30)	rodzaj_pojazdu character varying (30)	ladownosc numeric (4,2)	gotowy_do_sprzedaży boolean	cena numeric (20,2)	opis text
1	Benzyna	140	Czarny	Sedan	5.00	true	15000.00	Pol:
2	Benzyna	158	Biały	Sedan	4.50	false	145999.00	Doc
3	Diesel	120	Niebieski	Hatchback	4.00	true	65000.00	Bral
4	Diesel	190	Czerwony	Sedan	4.80	true	25000.00	[nul
5	Benzyna	200	Czarny	Sedan	4.70	false	25000.00	Doc
6	Diesel	180	Biały	Sedan	4.60	true	12000.00	[nul

13. Zapytanie 13 - UPDATE

OPIS: Aktualizacja ceny (obniżenie pierwotnej) dla samochodu o id=15

WYMAGANIE: Aktualizacja atrybutu

ZAPYTANIE:

```
UPDATE samochod SET cena=65000.00 TRUE WHERE id_samochod = 15;
```

WYNIK ZAPYTANIA:

	moc integer	kolor character varying (30)	rodzaj_pojazdu character varying (30)	ladownosc numeric (4,2)	gotowy_do_sprzedaży boolean	cena numeric (20,2)	opis text
7	150	Zielony	Hatchback	4.20	false	15000.00	Dodatkowo CB radio
8	190	Srebrny	Sedan	4.90	true	10000.00	[null]
9	175	Niebieski	Sedan	4.50	false	115000.00	[null]
10	130	Czerwony	Hatchback	4.10	true	17000.00	[null]
11	115	Biały	Hatchback	4.00	false	20000.00	[null]
12	130	Czarny	Hatchback	4.20	true	15000.00	Brak drugiego kluczyka
13	147	Srebrny	Sedan	4.50	false	45000.00	Dodatkowy komplet opon
14	110	Srebrny	Hatchback	4.30	true	90500.00	[null]
15	136	Niebieski	Hatchback	4.30	true	65000.00	Polski salon

14. Zapytanie 13 - DELETE

OPIS: Zwolnienie pracownika z id=15 (Zofia Kozłowska)

WYMAGANIE: Usunięcie rekordu

ZAPYTANIE:

```
DELETE FROM sprzedawca WHERE id_sprzedawca = 12;
```

WYNIK ZAPYTANIA:

	id_sprzedawca [PK] integer	imie character varying (30)	nazwisko character varying (30)	nr_telefonu character varying (30)	e_mail character varying (50)	id_komis integer	id_plac integer
1	1	Jan	Kowalski	123-456-789	jan.kowalski@exa...	1	1
2	2	Anna	Nowak	234-567-890	anna.nowak@exa...	1	1
3	3	Piotr	Wiśniewski	345-678-901	piotr.wisniewski@...	1	1
4	4	Katarzyna	Wójcik	456-789-012	katarzyna.wojcik@...	1	1
5	5	Marek	Kowalczyk	567-890-123	marek.kowalczyk...	1	2
6	6	Ewa	Kamińska	678-901-234	ewa.kaminska@ex...	1	2
7	7	Tomasz	Lewandowski	789-012-345	tomasz.lewandow...	1	2
8	8	Magdalena	Dąbrowska	890-123-456	magdalena.dabro...	1	2
9	9	Robert	Zieliński	901-234-567	robert.zielinski@e...	1	3
10	10	Agnieszka	Szymańska	012-345-678	agnieszka.szyman...	1	3
11	11	Paweł	Woźniak	123-456-789	pawel.wozniak@e...	1	3

4. Trigery

1. Opis triggerów:

- **Trigger 1 (log_cena_changes)** - Zapisywanie historii zmian ceny samochodu
- **Trigger 2 (before_insert_kartoteka_transakcji)** Sprawdzenie, czy samochód jest gotowy do sprzedaży przed dodaniem nowej transakcji do kartoteki transakcji.

2. Trigger 1

Nazwa: log_cena_changes

Opis: Ten trigger będzie zapisywać historię zmian ceny samochodu w osobnej tabeli historia_cen, za każdym razem, gdy cena samochodu zostanie zaktualizowana.

```
CREATE TABLE historia_cen (  
    id_historia serial PRIMARY KEY,  
    id_samochod int NOT NULL,  
    stara_cena decimal(20,2),  
    nowa_cena decimal(20,2),  
    data_zmiany timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    CONSTRAINT fk_samochod_historia FOREIGN KEY (id_samochod)  
        REFERENCES samochod(id_samochod)  
        ON DELETE CASCADE  
);  
  
CREATE OR REPLACE FUNCTION log_cena_changes()  
RETURNS TRIGGER AS $$  
BEGIN  
    INSERT INTO historia_cen (id_samochod, stara_cena, nowa_cena)  
    VALUES (OLD.id_samochod, OLD.cena, NEW.cena);  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER log_cena_update  
AFTER UPDATE OF cena ON samochod  
FOR EACH ROW  
WHEN (OLD.cena IS DISTINCT FROM NEW.cena)  
EXECUTE FUNCTION log_cena_changes();
```

PRZYKŁAD:

Założenia:

1. Mamy tabelę samochod, w której przechowywane są informacje o samochodach, w tym ich ceny.
2. Mamy tabelę historia_cen, która przechowuje historię zmian cen dla każdego samochodu.
3. Stworzono funkcję log_cena_changes i trigger log_cena_update, które zapisują zmianę ceny do tabeli historia_cen za każdym razem, gdy cena samochodu zostanie zaktualizowana.

WPROWADZENIE DANYCH: Wprowadzenie nowego samochodu z ceną 85 tys.

```
INSERT INTO samochod (nr_rejestracyjny, nr_vin, marka, model, rocznik, przebieg, silnik, paliwo, moc, kolor, rodzaj_pojazdu, ladownosc, gotowy_do_sprzedaży, cena, opis, id_plac)
VALUES ('ABC1234', '1HGBH41JXMN109186', 'Toyota', 'Camry', 2020, 15000.00, 2.5, 'Benzyna', 178, 'Biały', 'Sedan', 1.50, TRUE, 85000.00, 'Samochód w bardzo dobrym stanie', 1);
```

	kolor character varying (30)	rodzaj_pojazdu character varying (30)	ladownosc numeric (4,2)	gotowy_do_sprzedaży boolean	cena numeric (20,2)	opis text	id_pla integi
7	Biały	Sedan	4.60	true	12000.00	[null]	
8	Zielony	Hatchback	4.20	false	15000.00	Dodatkowo CB radio	
9	Srebrny	Sedan	4.90	true	10000.00	[null]	
10	Niebieski	Sedan	4.50	false	115000.00	[null]	
11	Czerwony	Hatchback	4.10	true	17000.00	[null]	
12	Biały	Hatchback	4.00	false	20000.00	[null]	
13	Czarny	Hatchback	4.20	true	15000.00	Brak drugiego kluczyka	
14	Srebrny	Sedan	4.50	false	45000.00	Dodatkowy komplet opon	
15	Niebieski	Hatchback	4.30	true	65000.00	Polski salon	
16	Biały	Sedan	1.50	true	85000.00	Samochód w bardzo dobrym stanie	

AKTUALIZACJA: Obniżenie ceny samochodu

```
UPDATE samochod
SET cena = 80000.00
WHERE id_samochod = 16;
```

	moc integer	kolor character varying (30)	rodzaj_pojazdu character varying (30)	ladownosc numeric (4,2)	gotowy_do_sprzedaży boolean	cena numeric (20,2)	opis text
7	180	Biały	Sedan	4.60	true	12000.00	[null]
8	150	Zielony	Hatchback	4.20	false	15000.00	Dodatkowo CB radio
9	190	Srebrny	Sedan	4.90	true	10000.00	[null]
10	175	Niebieski	Sedan	4.50	false	115000.00	[null]
11	130	Czerwony	Hatchback	4.10	true	17000.00	[null]
12	115	Biały	Hatchback	4.00	false	20000.00	[null]
13	130	Czarny	Hatchback	4.20	true	15000.00	Brak drugiego kluczyka
14	147	Srebrny	Sedan	4.50	false	45000.00	Dodatkowy komplet opon
15	136	Niebieski	Hatchback	4.30	true	65000.00	Polski salon
16	178	Biały	Sedan	1.50	true	80000.00	Samochód w bardzo dobrym st

WERYFIKACJA: Weryfikacja historii zmian ceny

```
SELECT * FROM historia_cen
WHERE id_samochod = 16;
```

REZULTAT

	id_historia [PK] integer	id_samochod integer	stara_cena numeric (20,2)	nowa_cena numeric (20,2)	data_zmiany timestamp without time zone
1	2	16	85000.00	80000.00	2024-06-02 14:02:31.444755

3. Trigger 2

Nazwa: before_insert_kartoteka_transakcji

Opis: Trigger przed wstawieniem nowego rekordu do tabeli kartoteka_transakcji, który sprawdza, czy samochód jest gotowy do sprzedaży. Jeśli nie, wstawienie rekordu zostanie zablokowane.

```
CREATE OR REPLACE FUNCTION check_car_ready_for_sale()
RETURNS TRIGGER AS $$
BEGIN
    IF (SELECT gotowy_do_sprzedaży FROM samochod WHERE id_samochod =
NEW.id_samochod) = FALSE THEN
        RAISE EXCEPTION 'Samochód nie jest gotowy do sprzedaży';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER before_insert_kartoteka_transakcji
BEFORE INSERT ON kartoteka_transakcji
FOR EACH ROW
EXECUTE FUNCTION check_car_ready_for_sale();
```

PRZYKŁAD:

AKTUALIZACJA: Aktualizacja atrybutu na False

```
UPDATE samochod SET gotowy_do_sprzedaży = FALSE WHERE id_samochod = 1;
```

	moc integer	kolor character varying (30)	rodzaj_pojazdu character varying (30)	ladownosc numeric (4,2)	gotowy_do_sprzedaży boolean	cena numeric (20,2)	opis text
1	158	Biały	Sedan	4.50	false	145999.00	Dodatkowy komplet opon
2	120	Niebieski	Hatchback	4.00	true	65000.00	Brak drugiego kluczyka
3	190	Czerwony	Sedan	4.80	true	25000.00	[null]
4	200	Czarny	Sedan	4.70	false	25000.00	Dodatkowy komplet opon
5	180	Biały	Sedan	4.60	true	12000.00	[null]

WPROWADZENIE DANYCH: Wprowadzenie transakcji

```
INSERT INTO kartoteka_transakcji (rodzaj, data_transakcji, samochod_w_rozliczeniu,
id_samochod, id_klient, id_sprzedawca, id_plac, id_faktura)
VALUES ('Sprzedaż', '2024-05-01', FALSE, 1, 1, 1, 1, 1);
```

WYNIK DZIAŁANIA TRIGGERA

```
ERROR:  Samochód nie jest gotowy do sprzedaży
CONTEXT:  funkcja PL/pgSQL check_car_ready_for_sale(), wiersz 4 w RAISE

BŁĄD:  Samochód nie jest gotowy do sprzedaży
SQL state: P0001
```