

**Uniwersytet Gdański**



# **BAZY DANYCH**

**Projekt: Komis**

**Prowadzący:**

**Wykonała:  
Paulina Kimak**

**Spis treści**  
**Opis projektu**

**Spis treści**

1. Opis Projektu.....	4
1. Schemat ERD bazy danych dla komis samochodowego.....	4
2. Encje.....	4
3. Atrybuty encji.....	5
4. Związki encji.....	6
2. Fragmenty tabel.....	6
3. Przykłady zapytań.....	7
1. Zapytanie 1.....	7
2. Zapytanie 2.....	8
3. Zapytanie 3.....	8
4. Zapytanie 4.....	9
5. Zapytanie 5.....	9
6. Zapytanie 6.....	10
4. Trigery.....	11
1. Opis triggerów:.....	11
2. Trigger 1.....	11
3. Trigger 2.....	12
Założenia początkowe dla projektu (inserty).....	12

## Wymagania

Dokumentacja:

- 1.opis słowny projektu, jakie są najważniejsze encje i ich atrybuty, jaki charakter mają związki je łączące, jakie pytania mamy zamiar zadawać, ew. jakie przewidujemy wyzwalacze,
- 2.diagram związków i encji,
- 3.fragmenty tabel w projekcie, po dwa, najwyżej trzy wiersze z każdej tabeli (ta część powinna się zmieścić na jednej stronie wydruku),
- 4.przykłady zapytań i ew. fragmenty ich wyników.

Dokumentacja powinna być dostarczona w postaci jednego pliku pdf.

2.Plaki:

- 1.z definicją tabel,
- 2.z danymi (z polskimi literami! Jeśli atrybut nie jest kluczem, np. nazwisko, to powinien się powtórzyć. Dla związku wieloznacznego powinien być przykład, że nie jest jednoznaczny, dla jednoznacznego, że nie jest 1-1, Jeśli atrybut dopuszcza NULL, powinny być przykłady ),
- 3.pomocniczy plik do usuwania tabel,
- 4.plik z zapytaniami,
- 5.plik z definicjami procedur wyzwalanych,
- 6.plik z przykładami zastosowania tych procedur

Wymagania - zapytania

Zapytania w projekcie powinny wykazać następujące aspekty:

- Obecność złączenia, zarówno w składni ze słowami **INNER JOIN** jak i bez niej.
- **Użycie grupowania i funkcji agregujących.**
- Wyświetlanie wyniku **operacji np. operacji arytmetycznych na liczbach czy na datach.**
- Warunki odwołujące się do wzorców napisów (np. opisy wszystkich układanek, **LIKE 'układanka%'**).
- **Użycie zagnieżdżenia w obu odmianach, nieskorelowane i skorelowane.**
- Sprawdzanie **warunku NULL.**
- **Zapytania negatywne, np. klienci bez złożonych zamówień.** Co najmniej w dwu wersjach.

Wersje zapytań negatywnych:

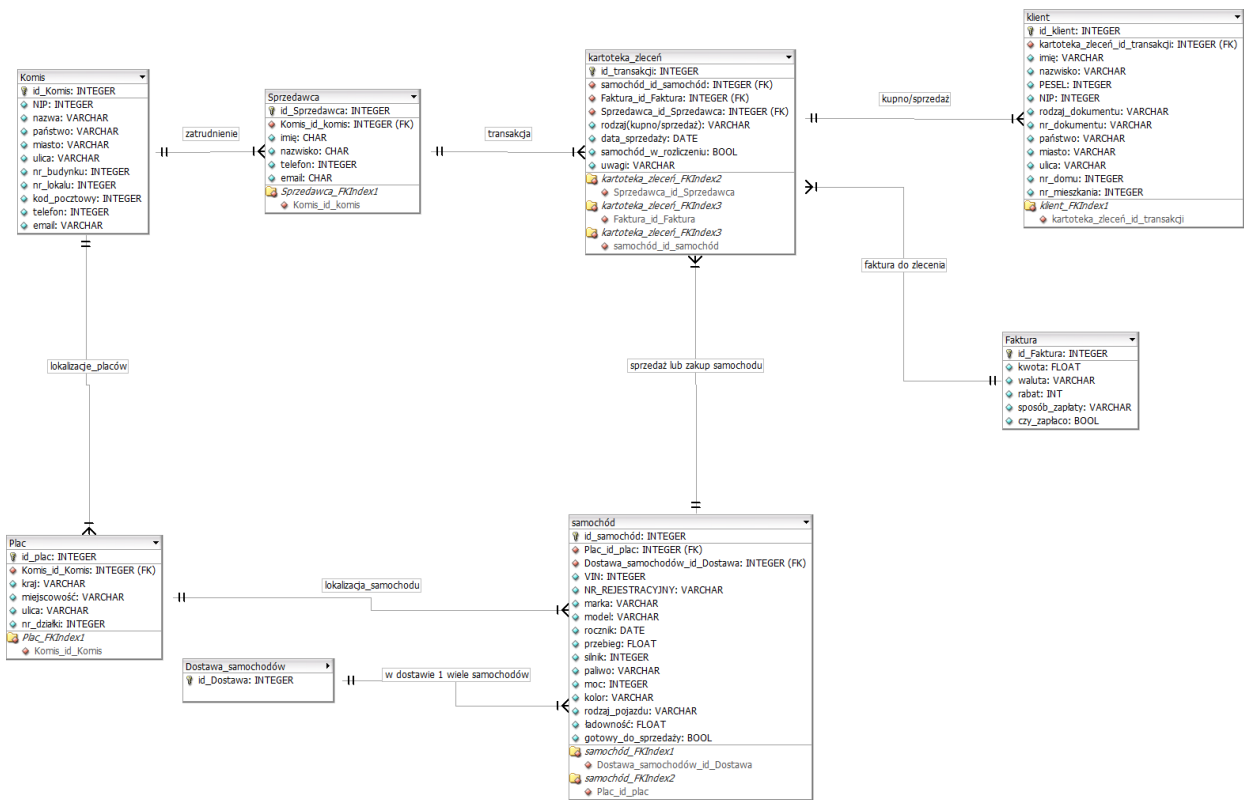
```
SELECT * FROM klient WHERE klient.nr NOT IN (SELECT klient_nr FROM zamowienie)
SELECT * FROM klient WHERE NOT EXISTS (SELECT * FROM zamowienie WHERE
klient.nr=klient_nr)
SELECT * FROM klient LEFT JOIN zamowienie ON klient.nr=klient_nr WHERE
zamowienie.nr IS NULL
SELECT klient.nr FROM klient EXCEPT SELECT klient_nr FROM zamowienie
```

Użycie widoku (perspektywy, VIEW).  
Usuwanie z tabeli.  
Aktualizacja wierszy w tabeli.

# 1. Opis Projektu

Celem projektu było stworzenie bazy danych dla komis samochodowego. 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 komis samochodowego



## 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 komis (Tabela 2.)
3. **sprzedawca**- posiada informacje o sprzedawcach zatrudnionych w komisie (Tabela 3.)
4. **samochód**- posiada informacje o samochodach na stanie komis (Tabela 4.)
5. **dostawa** – posiada informacje o dacie dostawy samocjodu, kraju pochodzenia samochodu, czy samochód uszkodzony oraz nr placu (Tabela 5.)
6. **klient** - posiada informacje o klientach komis (Tabela 6.)
7. **kartoteka\_transakcji** - posiada informacje o dokonanych transakcjach, łącząc dane o samochodzie, kliencie, sprzedawcy, fakturze oraz dacie transakcji (Tabela 7.)
8. **faktura** – posiada informacje m.in. o kwocie transakcji, rabacie, sposobie zapłaty(Tabela 8.)

## 3. Atrybuty encji

Tabela 1. **komis**

Nazwa pola:	Typ:
id_komis	serial
nip	int
nazwa	varchar(30)
panstwo	varchar(30)
miasto	varchar(30)
ulica	varchar(30)
nr_budynku	int
nr_lokalu	varchar(10)
kod_pocztowy	char(6)
telefon	varchar(15)
email	varchar(30)

Tabela 2. **plac**

Nazwa pola:	Typ:
id_plac	serial
krajvarchar(30)	varchar(30)

miestowosc	varchar(30)
ulica	varchar(30)
nr_działki	varchar(20)
ulica	varchar(30)
id_komis	int
nr_lokalu	varchar(10)
kod_pocztowy	char(6)
telefon	varchar(15)
email	varchar(30)

#### 4. Związki encji:

- **Komis <posiada> wiele placów, relacja 1 do N.** W stworzonej bazie danych istnieje 1 komis oraz 3 place, należące do komis, które znajdują się w różnych lokalizacjach.
- **Komis <zatrudnia> wielu sprzedawców, relacja 1 do N.** W stworzonej bazie danych istnieje 1 komis który zatrudnia 12 sprzedawców.
- **Na placu <pracuje> wielu sprzedawców, relacja 1 do N.** W stworzonej bazie danych 4 sprzedawców pracuje na każdym z 3 placów.
- **Samochód <znajduje się> na 1 z 3 placów, relacja 1 do N.** W stworzonej bazie danych do każdego samochodu przypisany jest numer placu.
- **W dostawie <dostarczane> są samochody, relacja 1 do N.** W stworzonej bazie danych w 1 dostawie może zostać dostarczonych różna ilość samochodów.
- **Dostawa <dostarczana> jest na 1 z 3 placów, relacja 1 do N.** W stworzonej bazie danych 1 dostawa może zostać dostarczona na 1 z 3 placów.
- **W kartotece transakcji zawarte są informacje o wielu transakcjach, relacja N do N.** W stworzonej bazie wiele klientów może kupić wiele samochodów, relacja N do N. Samochody są kupowane/sprzedawane przez wielu sprzedawców, relacja N do N. Na wielu fakturach może znajdować się wiele transakcji, , relacja N do N.

#### 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\_działki, 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\_sprzedaży, 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\_zapłaty, 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

## 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 (10)	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)	miestowosc 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_rejestracyjny character varying (10)	nr_vin character varying (17)	marka character varying (30)	model character varying (30)	rocznik integer	przebieg numeric (12,2)	silnik numeric (4,2)	paliwo character varying (10)	moc integer	kolor character varying (10)	rodzaj_pojazdu character varying (10)	ladownosc numeric (4,2)	gotowy_do_sprzedaży boolean	opis text	id_plac integer
1	1	ABC1234	1HGCM8...	Toyota	Corolla	2015	80000.00	1.80	Benzyna	140	Czarny	Sedan	5.00	true	Polski sal...	1
2	2	DEF5678	2HGCM8...	Honda	Civic	2018	60000.00	2.00	Benzyna	158	Biały	Sedan	4.50	false	Dodatk...	2
3	3	GHI9101	3HGCM8...	Ford	Focus	2020	30000.00	1.50	Diesel	120	Niebieski	Hatchback	4.00	true	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

Kartoteka transakcji tabela nr 7

	id_transakcja [PK] integer	rodzaj character varying (15)	data_transakcji 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
2	2	sprzedaż	2024-01-15	true	Samochód kupiony z rabatem		2	10	2	2
3	3	sprzedaż	2024-02-05	false	Poleka przed wydaniem		3	3	3	3

Faktura tabela nr 8

	id_faktura [PK] integer	nr_faktury character varying (10)	kwota numeric (12,2)	waluta character varying (3)	przelicznik_waluty numeric (6,3)	rabat smallint	sposob_zapłaty character varying (30)	czy_zapłacono boolean
1	1	F/2024/001	1500.00	PLN	1.000	5	Przelew	true
2	2	F/2024/002	2500.00	EUR	4.550	[null]	Gotówka	true
3	3	F/2024/003	1800.00	USD	3.755	10	Karta kredytowa	true



### 3. Przykłady zapytań

#### 1. Zapytanie 1

**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	QRS4456	5HGCM82633C445566	Kia	Ceed	2017	70000.00	1.60	Diesel	136

#### 2. Zapytanie 2

**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

**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

**OPIS:** Wyświetl pokaż klientów którzy nie dostali żadnego rabatu (rabat=NULL)

**WYMAGANIE:** Sprawdzanie warunku NULL

**ZAPYTANIE:**

```
SELECT  
    k.imie,  
    k.nazwisko,  
    f.rabat  
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 NULL;
```

**WYNIK ZAPYTANIA:**

	imie character varying (30)	nazwisko character varying (30)	rabat smallint
1	Grzegorz	Kaczmarek	[null]
2	Magdalena	Woźniak	[null]

## 5. Zapytanie 5

**OPIS:** Wyświetl tabelę faktura z kwotami podanymi w walucie polskiej

**WYMAGANIE:** Użycie widoku (perspektywy, VIEW).

**ZAPYTANIE:**

**Stworzenie view:**

```
CREATE VIEW kwoty_polskie
AS
SELECT
    id_faktura,nr_faktury,
    ROUND(kwota * przelicznik_waluty,2) AS kwota_PLN,
    rabat,sposob_zaplaty,czy_zaplacono
FROM faktura;
```

**Uruchomienie view:**

```
SELECT * FROM kwoty_polskie;
```

**WYNIK ZAPYTANIA:**

	id_faktura integer	nr_faktury character varying (10)	kwota_pln numeric	rabat smallint	sposob_zaplaty character varying (30)	czy_zaplacono boolean
1	1	F/2024/001	1500.00	5	Przelew	true
2	2	F/2024/002	11375.00	[null]	Gotówka	true
3	3	F/2024/003	6759.00	10	Karta kredytowa	true
4	4	F/2024/004	3200.00	7	Przelew	false
5	5	F/2024/005	20060.00	4	Gotówka	true
6	6	F/2024/006	4200.00	[null]	Przelew	false
7	7	F/2024/007	15925.00	9	Przelew	true
8	8	F/2024/008	10514.00	3	Karta kredytowa	true
9	9	F/2024/009	1900.00	4	Przelew	true
10	10	F/2024/010	4000.00	7	Gotówka	true

## 6. Zapytanie 6

**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

## 4. Trigery

### 1. Opis triggerów:

- **Trigger 1 (before\_insert\_faktura)** -Automatyczne obliczenie kwoty na fakturze po uwzględnieniu rabatu.
- **Trigger 2 (before\_insert\_kartoteka\_transakcji)** Sprawdzenie, czy samochód jest gotowy do sprzedaży przed dodaniem nowej transakcji do kartoteki transakcji.

### 2. Trigger 1

**Nazwa:** before\_insert\_faktura

**Opis:** Trigger przed wstawieniem nowego rekordu do tabeli faktura, który automatycznie oblicza kwotę po uwzględnieniu rabatu.

```
CREATE OR REPLACE FUNCTION apply_discount()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.rabat IS NOT NULL THEN
        NEW.kwota := NEW.kwota - (NEW.kwota * NEW.rabat / 100);
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER before_insert_faktura
BEFORE INSERT ON faktura
FOR EACH ROW
EXECUTE FUNCTION apply_discount();
```

### 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();
```

### Założenia początkowe dla projektu (inserty)

1. 1 komis
2. 3 place należące do komis
3. na każdym z placów pracuje 4 pracowników
4. klienci mogą otrzymać rabat max 10%

### Możliwe Checki:

- pesel
- nip
- vin