

Stacja Paliw

Dokumentacja



Krzysztof Kaniuka – 240697

Miłosz Wyrębkiewicz – 240840

Igor Tylak - 240827

Spis treści

[Opis projektu 2](#_Toc169954558)

[Założenia: 2](#_Toc169954559)

[Ograniczenia 2](#_Toc169954560)

[Baza danych tabele 2](#_Toc169954561)

[Zależności między tabelami 2](#_Toc169954562)

[Role 3](#_Toc169954563)

[Podział źródeł danych 3](#_Toc169954564)

[Procedury i funkcje 3](#_Toc169954565)

[Menadżer 3](#_Toc169954566)

[Pracownik 3](#_Toc169954567)

[Klient 4](#_Toc169954568)

[Techniczne 4](#_Toc169954569)

[Zarząd 4](#_Toc169954570)

[Triggery 5](#_Toc169954571)

# Opis projektu

Baza danych dla stacji paliw jest zaprojektowana w celu zarządzania wszystkimi aspektami działalności stacji paliw, w tym sprzedażą paliw, zarządzaniem zapasami, obsługą klientów, rejestrowaniem transakcji oraz generowaniem raportów.

# Założenia:

1. Maks 3 lub 4 pracowników na zmianie
2. Ten sam pracownik nie może mieć 2 zmian jednego dnia
3. Maksymalnie 1 dostawa danego typu jednego dnia
4. Ilość paliwa nie może spać poniżej 5% (wywala błąd bazy, nie można zrealizować transakcji)
5. Transakcje mogą być rejestrowane dla klientów bez programu lojalnościowego (KlientID = Null)
6. Mechanizm zliczania liczby punktów na podstawie transakcji (procedura)
7. Uzupełnianie stanu paliwa w momencie realizacji zamówienia (set fuel order status) i wykonania transakcji paliwowej
8. Redukcja produktów w momencie wykonania zamówienia spożywczego (błąd gdy brakuje produktów w magazynie)

# Ograniczenia

1. Opodatkowanie produktów
2. Regulacje prawne związane z pensją pracowników
3. Zamówienia składane ręcznie (nie zakładamy automatycznego harmonogramu zamówień)
4. Brak rozbudowanej struktury systemu punktów lojalnościowych
5. Brak sklepu punktów lojalnościowych

# Baza danych tabele

* Paliwa
* Pracownicy
* Dostawcy paliwowi
* Dostawcy spożywczy
* Produkty spożywcze
* Zamówienia paliwowe
* Zamówienia spożywcze
* Klienci (karty lojalnościowe)
* Transakcje paliwowe
* Transakcje spożywcze
* Harmonogram zmian

# Zależności między tabelami

* Zamówienia paliwowe – Paliwa (1:\*)
* Zamówienie spożywcze – produkty spożywcze (1:\*)
* Klient (karty lojalnościowe) – transakcje paliwowe (1:\*)
* Klient (karty lojalnościowe) – transakcje spożywcze (1:\*)
* Dostawcy paliwowi – zamówienia paliwowe (1:\*)
* Dostawcy spożywczy – zamówienia spożywcze(1:\*)
* Harmonogram zmian

# Role

* Menadżer stacji (zarządzanie pracownikami i cenami na stacji)
* Pracownik stacji (Sprawdza ilość produktów na magazynie i paliwa, składa zamówienia, realizuje transakcje)
* Klient (sprawdza liczbę punktów)
* Zarząd (prowadzenie analizy, wgląd w statystyki)

# Podział źródeł danych

1. Excel
   1. Pracownicy (mało, łatwo się wykonuje operacje księgowe)
2. Oracle (zaopatrzenie)
   1. Zamówienia spożywcze i paliwowe
   2. Dostawy spożywcze i paliwowe
   3. Klienci (system lojalnościowy) – nieprzypisani do stacji
   4. Produkty spożywcze i paliwa
3. SQL Server (instancja stacji)
   1. Transakcje paliwowe i spożywcze
   2. Harmonogram zmian
4. SQL Server (zarząd) (być  może jako jeden SQL Server)
   1. Widoki z transakcji
   2. Średnia okresowa
   3. Maksymalna wartość transakcji
   4. Ilość transakcji jednego dnia, wartość etc.

# Procedury i funkcje

## Menadżer

* 1. Procedury
     1. + Hire\_employee
     2. - Fire\_employee (nieobsługiwane przez tą wersję ISAM)
     3. + Update\_Petrol\_prices – metoda z parametrem, które paliwo. Ustawia cenę paliw
     4. + Update\_petrol\_points – aktualzuje punkty dla danego palwia
     5. -Set\_emp\_salary -modyfikowanie danych w Excel przez SQL Server nie jest wspierane
     6. Set\_product\_price (na Oracle) – sprawdzamy czy cena nie większa/mniejsza od starej ceny o 50% - wartość ustawiona na sztywno
     7. Set\_product\_points (na Oracle)
     8. add\_shift\_employee(@id\_emp INT,@shift\_date DATETIME)
     9. delete\_employee\_shift(@shift\_id INT)
     10. show\_employee\_schedule(@emp\_id INT)
  2. Funkcje
  3. Widoki
     1. + Show\_employees

## Pracownik

1. Procedury
   * 1. Add\_product\_transaction –jednocześnie usuwa podaną w zamówieniu ilość produktu oraz sprawdza przed tą operacją czy mamy wystarczającą ilość produktów
     2. Add\_petrol\_transacton – sprawdza czy paliwa po operacji zostanie powyżej określonego pułapu na podstawie wartości maksimum (5% z maksa co najmniej musi zostać)
     3. + Make\_order\_spożywka – zamówienie na produkt, sprawdza czy nie ma dostawy danego dnia
     4. + Make\_order\_paliwko – zamowienie na paliwo, sprawdza czy żądana ilość nie przekroczy maksimum i czy nie ma dostawy danego dnia
     5. Register\_client – stwórz nowego klienta (wartość punktów 0) na podstawie imienia, nazwiska i maila
     6. show\_employee\_schedule(@emp\_id INT)
2. Funkcje
   * 1. Show\_product\_on\_stock(productID) (ewentualnie jako procedura)
3. Widoki
   * 1. + Show\_products\_prices
     2. + Show\_products\_stock
     3. + Show\_petrol\_prices
     4. + Show\_petrol\_stock

## Klient

1. Procedury
   * 1. + Table Check\_transaction\_history() – historia transakcji klienta (osobno spozywcze I paliwowe, albo razem)
     2. + INT Check\_loyalty\_points(INT ClientID) – sprawdzenie ilości zgromadzonych punktów
2. Funkcje

## Techniczne

1. Funkcje:
   * 1. On\_price change – sprawdza czy cena w widełkach – sprawdzanie w dedykowanej procedurze
     2. is\_shift\_valid(@id\_emp INT = 0,@shift\_date DATETIME)
     3. check\_employee\_on\_shift(@shift\_time DATETIME)
2. Procedury
   * 1. + Add\_Points\_to\_customer(CustId,points\_added)
     2. add\_shift\_employee(@id\_emp INT,@shift\_date DATETIME)
     3. delete\_employee\_shift(@shift\_id INT)
     4. show\_employee\_schedule(@emp\_id INT)

## Zarząd

1. Funkcje
   * 1. -
2. Procedury
   * 1. show\_transaction\_analize\_daily\_fuel(@fuel\_id INT)
     2. show\_transaction\_analize\_products\_daily(@product\_id INT)
     3. show\_history\_of\_fuel\_price(@fuel\_id INT)
     4. show\_orders\_of\_fuel(@fuel\_id INT)
     5. show\_number\_of\_new\_customer
     6. update\_price\_of\_fuel(@fuel\_name VARCHAR(MAX),@new\_price NUMERIC(3,2))
     7. update\_points\_of\_fuel(@fuel\_name VARCHAR(MAX),@new\_points INT)
     8. show\_statistic\_car\_wash
     9. show\_employee\_hours(@emp\_id INT = 1)
     10. show\_employee\_works\_all

# Triggery

- on\_petrol\_transaction - aktualizacja stanu paliwa po wykonanej transakcji, error gdy poniżej progu, nabicie klientowi punktów za tankowanie (progi punktowe do ustalenia w triggerze) –ustalone w procedurze add\_petrol\_transaction

- on\_product\_transaction - sprawdza stan produktu spożywczego (błąd jeżeli wynik <0) oraz aktualizacja punktów klienta, jeżeli ten znajduje się w bazie – ustalone w procedurze add\_product\_transaction

- on\_petrol\_delivery – sprawdza czy nie ma już dostawy danego dnia

- on\_products\_delivery – sprawdza czy nie ma już dostawy danego dnia

- FK\_triggery dla tabeli **transakcje\_spozywcze** (do tabeli Produkty i Klienci)i **transakcje paliwowe** (do tabeli Paliwa i Klienci).

- FK trigger dla tabeli **Harmonogram** (do Pracowników w Excelu)