**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

**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 ~

**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)

**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
3. SQL Server (instancja stacji)
   1. Transakcje paliwowe i spożywcze
4. SQL Server (zarząd)
   1. Widoki z transakcji
   2. Średnia okresowa
   3. Maksymalna wartość transakcji
   4. Ilość transakcji jednego dnia, wartość etc.

2 ostatnie zajęcia rozliczanie projektu

**Procedury i funkcje**:

1. Menadżer
   1. Procedury
      1. Hire\_employee
      2. Fire\_employee
      3. Update\_Petrol\_prices – metoda z parametrem, które paliwo. Ustawia cenę paliw
      4. Set\_emp\_salary
      5. Set\_emp\_shift(empID,datetime from, datetime to) – ustawia pracownikowi zmianę
      6. Unset\_emp\_shift(scheduleID)
   2. Funkcje
2. Pracownik
   1. Procedury
      1. Add\_product\_transaction
      2. Add\_petrol\_transacton
      3. Show\_products\_stock
      4. Show\_petrol\_prices
      5. Show\_petrol\_stock
      6. Make\_order\_spożywka
      7. Make\_order\_paliwko
   2. Funkcje
      1. Show\_product\_on\_stock(productID)
3. Klient
   1. Procedury
      1. Table Check\_transaction\_history() – historia transakcji klienta
   2. Funkcje
      1. INT Check\_loyalty\_points(INT ClientID) – sprawdzenie ilości zgromadzonych punktów
4. Techniczne
   1. Funkcje:
      1. On\_schedule\_add – sprawdza czy pracownik się nie przepracowuje, sprawdza czy nie przekroczono limitu pracowników na zamianę
      2. On\_price change – sprawdza czy cena w widełkach
   2. Procedury
      1. Add\_Points\_to\_customer(CustId,points\_added)
5. Zarząd
   1. Funkcje
      1. ?
   2. Procedury
      1. ?

**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)

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

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

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

**Dodatkowe Informacje:**