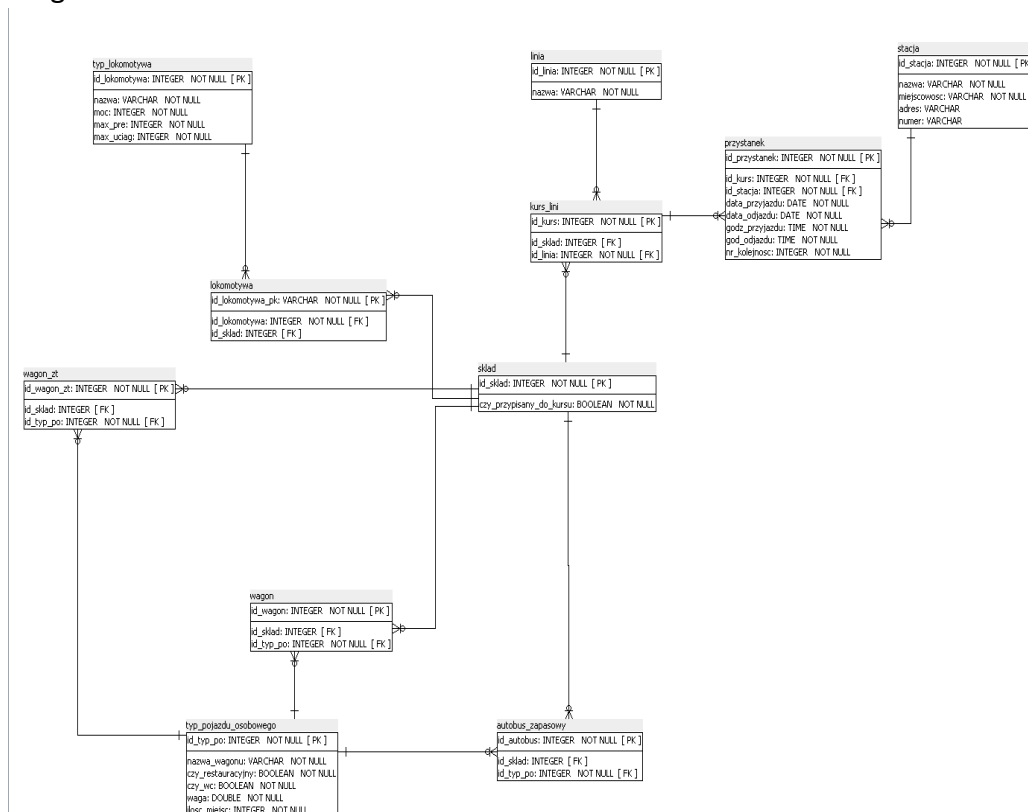


## 1) Projekt koncepcji, założenia.

- a) Temat projektu: Projekt zakłada stworzenie bazy danych połączeń kolejowych dla oprogramowania umożliwiającego zestawianie połączeń i składów kolejowych. Do głównych funkcjonalności aplikacji należeć będzie możliwość zarządzania i przeglądania połączeń kolejowych, taboru oraz składów pojazdów szynowych.
- b) Wymagania użytkownika:
  - i) Wprowadzanie pojazdów do systemu taboru kolejowego, modyfikacja istniejących.
  - ii) Tworzenie i zarządzanie liniami kolejowymi.
  - iii) Zestawianie składów pociągów – walidacja kompatybilności pojazdów.
  - iv) Prezentacja informacji zawartych w bazie danych.
- c) Aplikacja będzie umożliwiała zestawianie składów pociągowych, tj. lokomotyw, zespołów trakcyjnych, wagonów bądź autobusów zastępczych oraz przydzielanie ich do odpowiednich kursów linii kolejowych. Oprogramowanie będzie czuwało nad kompatybilnością stworzonych składów (np. moc pociągowa). Bazę pojazdów szynowych będzie można przeszukiwać, modyfikować i filtrować wg. zadanych parametrów.

## 2) Projekt logiczny

### a) Diagram ERD:



### b) Diagram przedstawia tabele oraz relacje między nimi.

- i) `typ_pojazdu_osobowego` – przechowuje min. nazwę, ilość miejsc dostępnych w wybranym pojeździe pasażerskim czy el. listy enumerowanej określający typ pojazdu. Zawiera informacje, które są wspólne dla bardziej szczegółowych egzemplarzy takich pojazdów (wagonów, pojazdów trakcyjnych, autobusów zapasowych).
- ii) `wagon_zt` – tabela reprezentuje egzemplarz – fizyczny obiekt – będący pojazdem szynowym (typu [EN57](#) itp.). W przypadku dalszej rozbudowy aplikacji tabela ta, jak również 2 poniższe mogłyby zawierać dodatkowe informacje min. o stanie technicznym, aktualnej lokalizacji czy roku produkcji danego egzemplarza.
- iii) `wagon` – tabela reprezentuje egzemplarz – fizyczny obiekt - będący wagonem (np. 156A)
- iv) `autobus_zapasowy` - reprezentuje egzemplarz – fizyczny obiekt – będący autobusem zapasowym w posiadaniu przewoźnika, wykorzystywany jako ZKS.
- v) `typ_lokomotywa` – podobnie jak tabela ‘`typ_pojazdu_osobowego`’ – przechowuje informacje o modelu danej lokomotywy tj. nazwa, moc (W), waga (tony), prędkość maksymalną (km/h) czy średni maksymalny uciąg lokomotywy (tony).
- vi) `lokomotywa` - tabela reprezentuje egzemplarz – fizyczny obiekt – będący lokomotywą (np. [EU07](#))
- vii) `sklad` – tabela wiążąca egzemplarze lokomotyw, wagonów, pojazdów szynowych i autobusów, spajające je w jeden skład pociągowy, możliwy do przydzielenia do kursu linii kolejowej.
- viii) `linia` – tabela reprezentuje linię kolejową rozumiane jako połączenie między danymi stacjami
- ix) `kurs_linii` – tabela zawiera informacje o przydzielonym składzie do kursu danej linii kolejowej (linia). Klucz główny tej tabeli odgrywa równie rolę klucza obcego w tabeli przystanki. W ten sposób tabela ‘`kurs_linii`’ łączy przydzielony skład pociągowy (`id_sklad`) z wybraną trasą linii kolejowej
- x) `przystanek` – tabela reprezentuje niejako miejsce (`id_stacji`) jak również czas i datę przyjazdu i odjazdu danego pociągu. Każdy element tej tablicy jest unikalny dla danego kursu – nie reprezentuje fizycznego miejsca lecz jest jedynie punktem na trasie pociągu.
- xi) `stacja` – tabela zawiera informacje o danej stacji kolejowej

### 3) Projekt logiczny.

- a) Do implementacji kluczy głównych w każdej tabeli użyto typu SERIAL, co pozwoliło na auto-inkrementację pól podczas insertowania wierszy do poszczególnych tabel.
  - i) `train.linia`
    - (1) `id_linia`: SERIAL NOT NULL - Unikalny identyfikator linii (klucz główny).
    - (2) `nazwa`: VARCHAR NOT NULL - Nazwa linii.
  - ii) `train.stacja`
    - (1) `id_stacja`: SERIAL NOT NULL - Unikalny identyfikator stacji (klucz główny).
    - (2) `nazwa`: VARCHAR NOT NULL - Nazwa stacji.
    - (3) `miestowosc`: VARCHAR NOT NULL - Miejscowość, w której znajduje się stacja.
    - (4) `adres`: VARCHAR - Adres stacji.

(5) numer: VARCHAR - Numer stacji.

iii) train.sklad

- (1) id\_sklad: SERIAL NOT NULL - Unikalny identyfikator składu (klucz główny).
- (2) czy\_przypisany\_do\_kursu: BOOLEAN DEFAULT false NOT NULL - Informacja, czy skład jest przypisany do kursu.

iv) train.kurs\_linii

- (1) id\_kurs: SERIAL NOT NULL - Unikalny identyfikator kursu (klucz główny).
- (2) id\_sklad: INTEGER - Identyfikator składu przypisanego do kursu.
- (3) id\_linia: INTEGER NOT NULL - Identyfikator linii, do której należy kurs.

v) train.przystanek

- (1) id\_przystanek: SERIAL NOT NULL - Unikalny identyfikator przystanku (klucz główny).
- (2) id\_kurs: INTEGER NOT NULL - Identyfikator kursu, do którego należy przystanek.
- (3) id\_stacja: INTEGER NOT NULL - Identyfikator stacji, gdzie znajduje się przystanek.
- (4) data\_przyjazd: DATE NOT NULL - Data przyjazdu na przystanek.
- (5) data\_odjazdu: DATE NOT NULL - Data odjazdu z przystanku.
- (6) godz\_przyjazdu: TIME NOT NULL - Godzina przyjazdu na przystanek.
- (7) godz\_odjazdu: TIME NOT NULL - Godzina odjazdu z przystanku.
- (8) nr\_kolejnosc: INTEGER NOT NULL - Numer kolejności przystanku w ramach kursu.

vi) train.typ\_lokomotywa

- (1) id\_lokomotywa: SERIAL NOT NULL - Unikalny identyfikator typu lokomotywy (klucz główny).
- (2) nazwa: VARCHAR NOT NULL - Nazwa typu lokomotywy.
- (3) moc: INTEGER NOT NULL - Moc lokomotywy.
- (4) max\_pre: INTEGER NOT NULL - Maksymalne prędkości, jakie może osiągnąć lokomotywa.
- (5) max\_uciag: INTEGER NOT NULL - Maksymalny uciąg lokomotywy.

vii) 7. train.lokomotywa

- (1) id: SERIAL NOT NULL - Unikalny identyfikator lokomotywy (klucz główny).
- (2) id\_lokomotywa: SERIAL NOT NULL - Identyfikator typu lokomotywy.
- (3) id\_sklad: SERIAL - Identyfikator składu, do którego należy lokomotywa.

viii) train.typ\_pojazdu\_osobowego

- (1) id\_typ\_po: SERIAL NOT NULL - Unikalny identyfikator typu pojazdu osobowego (klucz główny).
- (2) nazwa\_wagonu: VARCHAR NOT NULL - Nazwa typu pojazdu osobowego.
- (3) czy\_restauracyjny: BOOLEAN NOT NULL - Informacja, czy pojazd jest wagonem restauracyjnym.
- (4) czy\_wc: BOOLEAN NOT NULL - Informacja, czy w pojeździe jest WC.
- (5) waga: DOUBLE PRECISION NOT NULL - Waga pojazdu.

(6) `ilosc_miejsc`: INTEGER NOT NULL - Ilość miejsc w pojeździe.

ix) `train.autobus_zapasowy`

- (1) `id_autobus`: SERIAL NOT NULL - Unikalny identyfikator autobusu zapasowego (klucz główny).
- (2) `id_sklad`: SERIAL NOT NULL - Identyfikator składu, do którego należy autobus.
- (3) `id_typ_po`: SERIAL NOT NULL - Identyfikator typu pojazdu osobowego.

x) `train.wagon_zt`

- (1) `id_wagon_zt`: SERIAL NOT NULL - Unikalny identyfikator pojazdu szynowego (klucz główny).
- (2) `id_skla`: SERIAL - Identyfikator składu, do którego należy wagon.
- (3) `id_typ_po`: SERIAL NOT NULL - Identyfikator typu pojazdu osobowego.

xi) `train.wagon`

- (1) `id_wagon`: SERIAL NOT NULL - Unikalny identyfikator wagonu (klucz główny).
- (2) `id_sklad`: SERIAL - Identyfikator składu, do którego należy wagon.
- (3) `id_typ_po`: SERIAL NOT NULL - Identyfikator typu pojazdu osobowego.

b) Zdefiniowane zostały również funkcje pomocnicze jak:

- i) `train.waga_wagonu(id_wagon INTEGER)` – zwracająca wagę wagonu o zadanym id
- ii) `train.sklady_bez_kursu()` – zwraca wszystkie składy bez przydzielonego kursu
- iii) `train.kursy_bez_skladu()` – zwraca wszystkie kursy bez przydzielonego składu
- iv) `train.get_tables(your_schema_name VARCHAR)` – zwraca nazwy dostępnych table w schemie 'your\_schema\_name'
- v) `train.pojazdy_kursu_detailed(id_kurs_param INTEGER)` – zwraca wszystkie pojazdy i lokomotywy zawarte w składzie, który został przydzielony do kursu o id 'id\_kurs\_param'
- vi) `train.pojazdy_skladu_detailed(id_sklad_param INTEGER)` – funkcja o funkcjonalności podobnej do powyższej, jednak argumentem jest id\_skladu
- vii) `train.przystanki_kursu(id_kurs_param INTEGER)` – funkcja zwracająca dokładne informacje (stacja, data i godzina odjazdu/przyjazdu) o trasie (przystankach) danego kursu o id: 'id\_kurs\_param'

viii) `train.update_sklad_czy_przypisany_do_kursu()` – funkcja wykorzystywana w triggerze – ustawia pole składu 'czy\_przypisany\_do\_kursu' gdy wywoływane jest polecenie UPDATE na tabeli `kurs lini`. Odpowiednio ustawia 'true' na nowo przydzielanym składzie oraz 'false' na starym (o ile takowy istnieje).

c) Utworzone zostały również widoki:

- i) `train.lokomotywa_ilosc` – grupujący różne typy lokomotyw, podaje ich parametry oraz zlicza ilość zarejestrowanych egzemplarzy
- ii) `train.wagon_ilosc` – jw. lecz dla wagonów
- iii) `train.wagon_zt_ilosc` – jw. lecz dla zespołów trakcyjnych
- iv) `train.wagon_autobus_zapasowy_ilosc` – jw. lecz dla autobusów
- v) `train.lokomotywa_dane` – prezentuje poszczególne egzemplarze lokomotyw wraz z informacjami na ich temat, posortowane wg. nazwy

#### 4) Interfejs aplikacji użytkownika

a) Interfejs użytkownika został zaprojektowany przy użyciu framework'u Java Spring oraz środowiska IntelliJ umożliwiającego projektowanie aplikacji okienkowych. Aplikacja zawiera:

- i) 4 proste formularze umożliwiające stworzenie typu pojazdu oraz lokomotywy (tabele: typ\_pojazdu\_osobowego, typ\_lokomotywa, linia, stacja).
- ii) 2 proste formularze umożliwiające rejestrację pojazdów (tabele: wagon\_zt, wagon, autobus\_zapasowy, lokomotywa).
- iii) 2 złożone formularze umożliwiające:
  - (1) Stworzenie składu pociągu (po wcześniejszej rejestracji pojazdów).
  - (2) Stworzenie kursu linii kolejowej (po wcześniejszej rejestracji linii i stacji).
- iv) Formularz przyporządkowujący skład pociągowy do kursu.

The screenshot shows the 'Bazy danych' application window with the 'Wagony' tab selected. The 'Dodaj typ' sub-tab is active, showing a form for adding a new vehicle type. The form includes a dropdown menu for 'Typ pojazdu' (currently set to 'Ezt'), input fields for 'Nazwa wagonu', 'Waga (tony)', and 'Ilość miejsc', and checkboxes for 'Przedział restauracyjny' and 'Posiada WC'. A 'Zapisz typ pojazdu' button is located at the bottom of the form. The left sidebar contains buttons for 'Podgląd tabel' and 'Raporty'.

ad. 4.a.i – przykładowy prosty formularz

The screenshot shows the 'Bazy danych' application window with the 'Wagony' tab selected. The 'Dodaj egzemplarz' sub-tab is active, showing a form for adding a new vehicle example. The form includes a dropdown menu for 'Typ pojazdu' (currently set to 'wagon'), a dropdown menu for 'Egzemplarz' (currently set to 'B10bmnuoz2 - 193'), and a 'Dodaj pojazd' button. Below the form, the 'Aktualny uciąg składu: 650.0 t' and 'Aktualna waga składu: 506.0 t' are displayed. The right side shows a table of available vehicles with columns: ID, Typ pojazdu, Nazwa pojazdu, and Masa. The table lists 14 vehicles, including locomotives and wagons. A 'Zapisz skład' button is located at the top right.

ID	Typ pojazdu	Nazwa pojazdu	Masa
340	Lokomotywa	EP09	
69	Wagon	A9mnou21	45
307	Wagon	B11mnou2	46
129	Wagon	WRmnou2	49
194	Wagon	B10bmnuoz2	45
49	Wagon	A9mnou21	45
406	Wagon	B11bmnuoz2	47
407	Wagon	B11bmnuoz2	47
365	Wagon	B11mnou2	46
370	Wagon	B11mnou2	46
193	Wagon	B10bmnuoz2	45
197	Wagon	B10bmnuoz2	45

ad. 4.a.iii.1 – złożony formularz – tworzenie składu pociągowego

Bazy danych

Wagony Lokomotywy Składy Linie

Dodaj stację Stwórz linię Utwórz kurs danej linii Przypisz pojazd do linii

Podgląd tabel

EIP 5300 Chopin Linia

Gdańsk Główny Stacja

17/01/2024 Data przyjazdu

18/01/2024 Data odjazdu

00:11 Godzina przyjazdu

00:11 Godzina odjazdu

Dodaj przystanek

Przenieś niżej Przenieś wyżej Usuń zaznaczony Zapisz kurs

Kolejność	Stacja	Czas przyjazdu	Czas odjazdu
0	Gdańsk Główny	18:39:00	18:39:00
1	Kutno	19:41:00	19:44:00
2	Warszawa Wchodnia	20:55:00	20:59:00
3	Warszawa Centralna	21:03:00	21:10:00
4	Warszawa Zachodnia	21:13:00	21:18:00
5	Kielce Dworzec PKP	22:54:00	22:56:00
6	Kraków Główny	23:59:00	00:05:00
7	Kraków Płaszów	00:11:00	00:11:00

Raporty

ad. 4.a.iii.2 – złożony formularz – tworzenie kursu

- Dane wprowadzane do bazy danych zostają podane weryfikacji po stronie bazy danych jak również po stronie aplikacji (typ wprowadzanych danych, poprawność logiczna)
- Raporty oraz prezentacja danych została wykonana za pomocą obiektów jTable wyświetlanych w oddzielnych oknach. Aplikacja pozwala wyświetlać zawartość wszystkich tabel oraz generować raporty takie jak:
  - Lista wszystkich lokomotyw opatrzona wszystkimi danymi technicznym poszczególnego egzemplarza.
  - Raport generujący skład pociąg znając numer kursu/składu.
  - Raport grupujący pojazdy i lokomotywy wg. nazwy oraz zliczający ilość zarejestrowanych egzemplarzy
  - Raport generujący trasę danego kursu linii kolejowej

Raport Lokomotyw Raport Składów Raport Typów Pojazdów Raport Lini

Lokomotywy Wagony EZT Autobusy

Generuj raport

id_typu_lokomotywy	nazwa_lokomotywy	ilosc_egzemplarzy
8	EU44	34
10	EP08	66
6	EU07A	91
5	EU07	37
9	111Eb	27
7	EP09	67

ad. 4.c.iii – przykładowy raport dla lokomotyw

Raport Lokomotyw   Raport Składów   Raport Typów Pojazdów   Raport Lini					
Kurs		Skład			
Generuj raport		29	Numer kursu		
liczba_porzadkowa	nazwa	typ_pojazdu	czy_restauracyjny	czy_wc	ilosc_miejsc
1	EU44	lokomotywa			
1	A9mnouz1	wagon	f	t	54
2	B10bmnouz2	wagon	f	t	56
3	WRmnouz	wagon	t	f	0
4	B10bmnouz2	wagon	f	t	56
5	B10bmnouz2	wagon	f	t	56
6	B11mnouz2	wagon	f	t	66
7	B10bmnouz2	wagon	f	t	56
8	B11bmnouz2	wagon	f	t	56
9	B11bmnouz2	wagon	f	t	56
10	B11bmnouz2	wagon	f	t	56

ad. 4.c.ii – przykładowy raport składu pociągowego

## 5) Przykładowe użytkowanie aplikacji

### a) Tworzenie składu pociągu:

- Należy utworzyć potrzebne typy wagonów (Wagony -> Dodaj typ), bądź zarejestrować ich egzemplarze (Wagony -> Dodaj egzemplarz)
- Analogicznie w przypadku lokomotyw: (Lokomotywy -> Dodaj typ/Dodaj egz.)
- W zakładce Składy należy skomponować skład pociągowy, dodawane pojazdy pokażą się na tymczasowej liście pojazdów, zatwierdzić skład klikając przycisk „Zapisz skład”