

Projekt Bazy Danych 2021

Baza danych Miejskiego Przedsiębiorstwa Komunikacyjnego

Grzegorz Przybylski, Paweł Pyciński

- 1** Podstawowe założenia projektu
- 2** Diagram ER
- 3** Diagram tabel
- 4** Widoki i funkcje w bazie
- 5** Procedury Składowane i wyzwalacze
- 6** Skrypt tworzący bazę danych
- 7** Utworzone indeksy i typowe zapytania
- 8** Strategia pielęgnacji bazy danych

Celem projektu jest utworzenie przejrzystej, łatwej w zarządzaniu bazy danych przedsiębiorstwa komunikacyjnego. Zasadniczo baza została podzielona na 5 sekcji:

- **Pojazdy**

W projekcie postanowiliśmy uwzględnić dwa typy pojazdów tj. tramwaje oraz autobusy. Typy pojazdów jakie posiadamy znajdują się w tabeli **Vehicle type**, są to min. autobusy typu Urbino 12/18 czy też tramwaje typu NGT6. Rozwinięciem tej tabeli jest tabela **Vehicles** oraz **Historic vehicles**, w tych tabelach znajdują się już konkretne modele pojazdów. Ostatnią tabelą sekcji jest **Service Part** w której są przechowywane informacje o ilości części na magazynie w tym przypadku została przyjęta zasada że tylko do jednego typu pojazdu pasuje dana część.

- **Linie i przystanki**

Do tej sekcji należą 4 tabele. Pierwsza z nich **Line** zawiera wszystkie linie które obecnie kursują po mieście. Założyliśmy, że ID line [0, 99] to będą linie tramwajowe a [100, 999] linie autobusowe. Każda linia ma dwa wyróżnione przystanki Początkowy oraz końcowy, ponadto jest też informacja która zajezdnia i jakie pojazdy ma wysłać na linię. W tabeli **Route**, znajdują się numery linii oraz wszystkie przystanki na których się zatrzymuje dana linia, ponadto jest także kierunek jazdy. Jeśli jest on "A" to znaczy, że pojazd linii jedzie przez ten przystanek jadąc z pętli 1 do pętli 2, natomiast "B" w przeciwnym kierunku. Tabela **Stop** zawiera wszystkie przystanki jakie mamy w mieście, natomiast tabela **City District** dzielnice miasta".

- **Pracownicy**

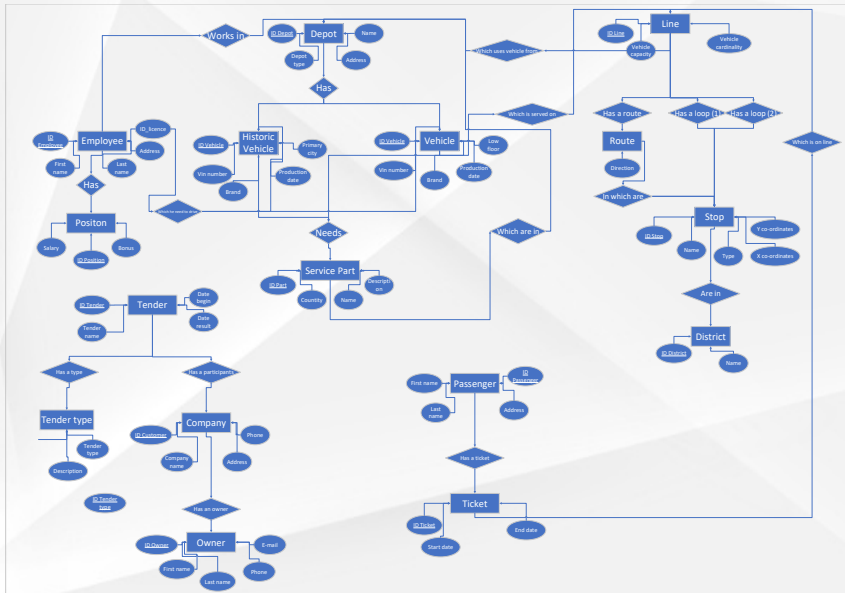
W tabeli **Employee** mamy dane wszystkich zatrudnionych, wiadomo, że nie są to tylko kierowcy. W skład pracowników wchodzi także osoby z zarządu czy kontroli biletów, dlatego w polu *Licence ID* zezwalamy na wartości *NULL*. W tabeli **Position** znajdują się wszystkie stanowiska w naszej firmie oraz odpowiednie do nich pensja i premia. Ponadto nie każdy pracownik jest przypisany do zajezdni, część z nich np odpowiedzialni za prace biurowe mogą pracować zdalnie, z domu

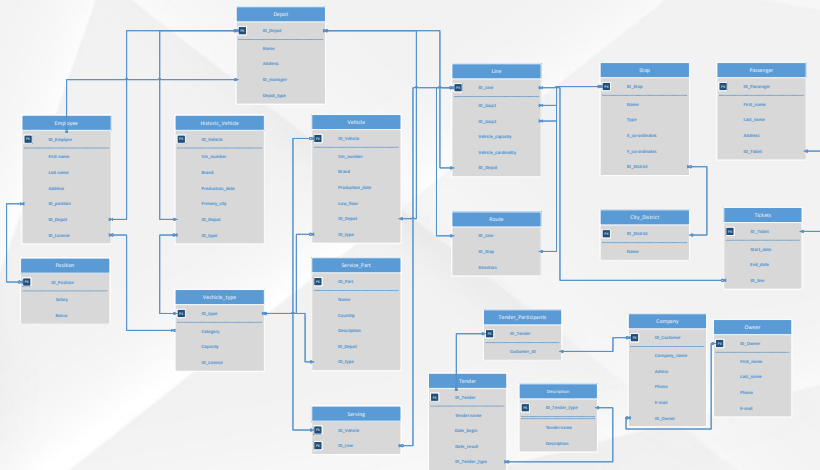
- **Pasażerowie**

Pasażerowie którzy mają wykupioną kartę biletową znajdują się w naszej bazie. W tabeli **Passenger** przechowujemy podstawowe informacje o nich, zaś w tabeli **Tickets** znajdują się wszystkie zakupione bilety.

- **Przetargi i kandydaci**

Każda większa spółka komunikacyjna zakup pojazdów czy części serwisowych lub środków czystości dokonuje w formie przetargowej wybierając jak najkorzystniejszą ofertę. W tabeli **Tender** przechowujemy wszystkie przeprowadzane przetargi. **Tender description** to tabela która zawiera kategorie przetargów tzn. Zakup pojazdów, remonty czy zakup części serwisowych oraz krótki opis sposobu przeprowadzenia takiego przetargu. **Tender participants** przechowuje informacje o zgłoszeniach do przetargów. Natomiast w tabeli **Company** znajdują się wszystkie zarejestrowane firmy zaś w **Owner** właściciele danych spółek.





Baza danych zawiera szereg widoków oraz funkcji pozwalających na prostszy dostęp do danych. Należą do nich:

1. Widok kierowców pracujących w Przedsiębiorstwie.
2. Funkcja jako argument biorąca numer pasażera, zwraca listę wszystkich przystanków, na które może dotrzeć dany pasażer. Funkcja wyznacza ten zasięg na podstawie posiadanych przez pasażera aktualnych biletów.
3. Widok wszystkich pojazdów znajdujących się w bazie (zarówno historycznych, jak i współczesnych)
4. Funkcja zwracająca dla podanego numeru menedżera zajezdni wszystkich pracowników tej zajezdni.
5. Statystyka popularności (potencjalnej ruchliwości) przystanków. Dla każdego przystanku zliczana jest liczba pasażerów, którzy mają bilety na ten przystanek.

6. Statystyki ilości zatrudnionych pracowników w stosunku do poziomu referencyjnego. Statystyka ta jest pogrupowana według zajezdni. Optymalne ilości zatrudnianych kierowców są na podstawie ilości autobusów wystawianych na linie przez daną zajezdnię. Dla pozostałych zawodów optymalne ilości pracowników są stałe.
7. Widok wypisujący wszystkie aktualnie przeprowadzane przetargi.
8. Statystyka potencjalnych pasażerów danych linii, podobna dla statystyki z punktu 5, jednak zawierająca również informacje o odpowiedzi na to obciążenie w postaci łącznej pojemności wchodzących skład danej linii pojazdów.
9. Funkcja przypisująca danej firmie wszystkie przetargi w których bierze ona udział.
10. Funkcja wypisująca dla podanej linii i kierunku kolejne przystanki w kolejności przejazdu linią.

1. Procedura ułatwia wstawianie trasy linii, bez konieczności pamiętania *ID stop* jest to znaczne ułatwienie przy wstawianiu, możemy na bieżąco monitorować jakie linie poruszają się po jakich trasach. Jako parametry podajemy numer linii oraz specjalny utworzony typ *StopTableTypeIdentity* o dwóch atrybutach *ORD* oraz *Stop name*.
2. Kolejna procedura umożliwia nam wstawienie nowego przystanku na trasie po zadaniu, jest to przydatne w sytuacjach gdy odległości pomiędzy przystankami są zbyt duże i należy dobudować nowy między nimi, lub gdy chcemy przedłużyć trasę linii o nowe części miasta. Procedura przyjmuje jako parametry *ID line*, *kierunek*, *ID stop* nowego przystanku oraz *ID stop* przystanku poprzedzającego.
3. Procedura umożliwia nam wstawienie nowego przystanku na trasie przed zadaniem, jest to również przydatne gdy powstanie przystanek tymczasowy lub trasa linii zostanie wydłużona i będziemy objeżdżać po trasie okrężnej dany fragment sieci. Procedura jako paramter przyjmuje *ID line*, *kierunek*, *ID stop* wstawianego przystanku, *ID stop* przystanku następującego.

4. Procedura pełni rolę aplikacji dla kanara w sieci, sprawdza czy pasażer podróżujący daną linią posiada na nią **ważny i aktualny** bilet, w przeciwnym wypadku nalicza karę + 100. Procedura przyjmuje jako parametry *ID passenger* oraz *ID line* którą podróżuje pasażer.
5. Procedura pozwala przepisać pojazd liniowy który zostaje wycofany z eksploatacji do tabeli pojazdów zabytkowych. Jako paramter przyjmuje *ID vehicle* wycowywanego pojazdu.

1. Pierwszy trigger *INSTEAD OF INSERT, UPDATE* ma za zadanie sprawdzać czy wstawiany kierowca ma poprawny numer licencji. Jeśli nie prawdopodobnie licencja jest nieważna lub podrabiana.
2. Kolejny trigger *INSTEAD OF INSERT* zachowuje spójność danych przy ręcznym wstawianiu przystanków, uniemożliwia rozpójnienia tras linii.
3. Trigger *AFTER INSERT, UPDATE, DELETE* aktualizuje dane dotyczące danej linii, po dodaniu lub usunięciu pojazdów do jej obsługi. Jest przydatny szczególnie gdy przekształcamy jakiś pojazd na historyczny lub dokupujemy nowe pojazdy i umieszczamy je na linii.
4. kolejny z kolei trigger *INSTEAD OF INSERT* poprawia błędne daty tak aby spełniały założenia bazy to znaczy data początkowa nie była wcześniejsza niż data zakończenia.
5. Trigger *AFTER UPDATE* zamawia nowe części gdy kończą się na magazynie, dzięki czemu nie dopuścimy do sytuacji gdzie braknie nam sprawnych pojazdów z powodu braku części zamiennych.

Skrypt tworzący bazę danych zawiera 18 poleceń tworzących tabelę wraz z odpowiednimi informacjami dotyczącymi ich pól oraz polecenia tworzące więzy klucza obcego. Kolejno mamy:

- Utworzenie tabeli **Employee** wraz z koniecznymi polami, zezwalamy NULL'e w polach *Address* nie każdy pracownik być może chce podać swój adres, *ID Depot*, niektórzy pracownicy nie są przypisani do zajezdni, mogą pracować w kilku lub wcale (zdalnie). *ID License* także może pozostać pusty, nie wszyscy pracownicy muszą być kierowcami.
- Utworzenie tabeli **Position** do przechowywania dostępnych stanowisk w firmie, wraz ze stawką i premią przewidzianą na stanowisku pole może pozostać puste.
- Utworzenie więzów klucza obcego między *id stanowiska* w tabeli **Employee** oraz stanowiska w tabeli **Position**.
- Utworzenie tabeli **Vehicle** do przechowywania konkretnych modeli pojazdów posiadanych przez spółkę wraz z potrzebnymi informacjami o pojeździe .
- Utworzenie tabeli **Historic Vehicle** do przechowywania konkretnych modeli historycznych pojazdów posiadanych przez spółkę wraz z potrzebnymi informacjami o pojeździe .

- Utworzenie tabeli **Vehicle type** przechowująca typy dostępnych pojazdów, podzielonych ze względu na pojemność oraz autobus / tramwaj. Tutaj wykorzystujemy dziedziczenie, tabele **Vehicle** oraz **Historic Vehicle** są jej pochodnymi.
- Utworzenie tabeli **Service part** w której przechowujemy części zamienne do pojazdów. Części znajdują się w konkretnej zajezdni i przypisane są do konkretnego typu pojazdu z tabeli **Vehicle type**.
- Utworzenie tabeli **Depot** przechowującej dane dotyczące zajezdni. Każda zajezdnia ma swojego managera i może być tramwajowa lub autobusowa.
- Utworzenie więzów klucza obcego pomiędzy *ID Depot* z tabeli **Vehicle**, **Historic Vehicle** oraz **Depot** aby powiązać pojazdy z zajezdniami.
- Utworzenie więzów klucza obcego pomiędzy *ID Type* z tabeli **Vehicle**, **Historic Vehicle** oraz **Vehicle type** aby powiązać pojazdy z ich typami.
- Utworzenie więzów klucza obcego pomiędzy *ID Depot* części serwisowych z tabeli **Service part** a zajezdnią w której się znajdują oraz pojazdami po prze pole *ID type* z tabeli **Vehicle type** do których dana część pasuje.

- Utworzenie więzów klucza obcego między tabelą **Depot** a **Employee** (*ID manager* -> *ID employee*).
- Utworzenie tabeli **Line** przechowującej linie kursujące po mieście wraz z informacją jaki typ pojazdu należy na nią skierować.
- Utworzenie tabeli **Stop** zawierającej informację o wszystkich przystankach w mieście wraz z niezbędnymi informacjami.
- Utworzenie tabeli **Route** przechowującej trasy pojazdów. Każdy rekord składa się z nr linii, identyfikatora przystanku, identyfikatora następnego przystanku i kierunku jazdy 'A' lub 'B'. Pojazd nie musi przejeżdżać przez ten sam zestaw przystanków w obie strony, na trasie mogą być przystanki jednostronne.
- Utworzenie tabeli **City district** przechowującej dzielnice w mieście, pozwoli to przyporządkować przystanki do konkretnej dzielnicy.
- Utworzenie tabeli **Serving** w której będzie znajdował się przydział konkretnych modeli pojazdów na linię.
- Utworzenie więzów klucza obcego między *id District* z tabeli **Stop** a *id District* z tabeli **City District**.

- Utworzenie więzów klucza obcego między *Id line* z tabeli **Route** a *Id line* z tabeli **Line**.
- Utworzenie więzów klucza obcego między *Id stop* z tabeli **Route** a *Id stop* z tabeli **Stop**.
- Utworzenie więzów klucza obcego między *Id next* z tabeli **Route**, które to definiuje następny przystanek z kolei na trasie a *Id stop* z tabeli **Stop**.
- Utworzenie więzów klucza obcego między *Id loop 1* oraz *Id loop 2* z tabeli **Line** a *Id stop* z tabeli **Stop**.
- Utworzenie więzów klucza obcego między *ID depot* w tabeli **Line** a *ID depot* w tabeli **Depot**.
- Utworzenie więzów klucza obcego między *ID vehicle* w tabeli **Serving** a *ID vehicle* w tabeli **Vehicle**.
- Utworzenie więzów klucza obcego między *ID line* w tabeli **Serving** a *ID line* w tabeli **Line**.

- Utworzenie tabeli **Tender** Przechowującej przetargi zorganizowane przez spółkę wraz z datą rozpoczęcia / zakończenia.
- Utworzenie tabeli **Tender description**, zawierającej opisy typów przetargów organizowanych przez spółkę. Typem przetargu może być dostawa tramwajów czy też dostawa części serwisowych natomiast przetarg na konkretną już część znajduje się tabeli **Tender**.
- Utworzenie tabel **Company** oraz **Company Owner**, przechowująca dane firmy biorące udział w przetargu oraz ich właścicieli, zakładamy, że jeden Owner może posiadać kilka spółek.
- Utworzenie tabeli **Tender participants**, przechowującej *ID tender* oraz *ID customer* - identyfikator przetargu oraz firma która w nim startuje.
- Utworzenie więzów klucza obcego między *Tender type* z tabeli **Tender** a *Tender type* z tabeli **Tender description**.
- Utworzenie więzów klucza obcego do elementów tabeli **Tender participants** kolejno z polem *ID tender* z tabeli **Tender** oraz *ID Customer* z tabeli **Customer**.
- Utworzenie więzów klucza obcego pomiędzy polem *ID owner* z tabeli **Company** a *ID owner* z tabeli **Company owner**.

- Utworzenie tabel **Passenger** oraz **Tickets**. W tabeli **Passenger** zezwalamy na *NULL* w polu *Address*.
- Utworzenie więzów klucza obcego z pola *ID Passenger* w tabeli **Ticket** do pola *ID Passenger* w tabeli **Passenger**.
- Utworzenie więzów klucza obcego z pola *ID line* w tabeli **Ticket** do *ID line* w tabeli **Line**.

1. Często będziemy sprawdzać czy pasażer który jest obecnie kontrolwany posiada **ważny** bilet na daną linię. Dlatego został założony indeks na kolumny *Last name* i *ID passenger* w tabeli **Passenger** oraz *End date* i *ID line* w tabeli **Tickets**.
2. Często również będziemy szukać zamiennych części serwisowych, w serwisach najczęściej wykorzystuje się do tego numer Vin pojazdu dlatego zostanie nałożony indeks na kolumnę *name* oraz *id type* z tabeli **Service parts** oraz *Vin number* z tabeli **Vehicle**.
3. Kolejnym ważnym zapytaniem które warto usprawnić indeksowaniem jest wyszukiwanie kierowców którzy mają licencję na dany pojazd. Jest to przydatne przy układaniu grafiku dla pracowników. Indeks zostanie nałożony na kolumnę *Last name*, *ID licence* z tabeli **Employee** oraz *Capacity* z tabeli **Vehicle type**.
4. Dla spóki ważny jest także dobór ofert przetargowych. Nałożony został indeks na kolumny *Tender name*, *ID tender* z tabeli **Tender** oraz *ID tender*, *ID Customer* z tabeli **Tender participants** i *ID customer*, *company name* z tabeli **Company**.

Baza danych jest dość rozległa, przechowujemy sporo danych, choć znaczna część z nich jest aktualizowana niezwykle rzadko. Nowe pojazdy nie są kupowane zbyt często, stanowiska jednorazowo ustanowione pozostają takie same przez cały okres funkcjonowania spółki, sporadycznie idąc z duchem czasu mogą pojawić się nowe takie jak *specjalista do spraw social media*. Codziennie za to jest aktualizowana lista pojazdów które wyjeżdżają na linię co wymusza na nas codzienne tworzenie kopii zapasowej. Codziennie będziemy wykonywać przyrostową kopię zapasową a raz na 3 miesiące całościową.