

Projekt bazy danych Miejskiego Przedsiębiorstwa Komunikacyjnego

1. Uczestnicy projektu

- a. Filip Turek
- b. Michał Sroka
- c. Marcin Słowiak

2. Cel i założenia projektu

Celem projektu było stworzenie bazy danych Miejskiego Przedsiębiorstwa Komunikacyjnego. Jako miasto wybraliśmy Kraków. Projekt przedstawia realne dane dotyczące taboru komunikacyjnego oraz linii kursujących po mieście.

W naszym projekcie zawarliśmy informacje dotyczące szczegółowych informacji na temat MPK takich jak wykonane naprawy pojazdów, zakupione części potrzebne od wykonania napraw. Projekt zawiera też istotne funkcjonalności tj. wykupione bilety, firmy sprzątające pojazdy, kontrole biletowe.

Przy projektowaniu założyliśmy, że wszyscy pracownicy znajdą się w jednej tabeli ale będą odpowiednio dopisywani do swoich funkcjonalności - tzn. żaden kontroler nie zostanie przypisany do wykonania naprawy itd.

3. Schemat bazy danych

Znajduje się w załączonym pliku Diagram_MPK.pdf

4. Opis tworzonych wyzwalaczy

a. **TR1**

Zapisuje do tabeli RejestrZmian dane użytkownika który wprowadził zmianę w cenie biletów, datę, typ biletu oraz starą i nową cenę

b. **TR2**

Zapobiega kasowaniu historii biletowej biletów okresowych

c. **TR3**

Zapobiega kasowaniu historii biletowej biletów jednorazowych

d. **TR4**

Ustawia cenę domyślną przy wstawianiu do biletów jednorazowych z tabeli typów biletów

e. **TR5**

Jeżeli dana firma sprząta w bieżący dzień (GETDATE()) to nie może zostać usunięta z bazy

5. Opis stworzonych procedur składowanych

a. *proc_wymienKierowce*

Procedura wymienia kierowcę z innym zadany kierowcą na jego trasie.
Funkcja przydatna przy dawaniu urlopu.

b. *proc_zmienNazwePrzystanku*

Procedura zmienia nazwę przystanku automatycznie w każdej możliwej tabeli i dodatkowo jeżeli na przystanku był automat to zmienia także jego adres.

c. *proc_dodajTrase*

Procedura dodająca nowy czas odjazdu dla istniejącej już linii (kierowca oraz pojazd przez niego obsługiwany muszą znajdować się już w bazie Obsługa_trasy)

d. *proc_dodajUsterke*

Procedura dodająca nową usterkę pojazdu wraz z opisem zaistniałego zdarzenia do tabeli Historia_usterek.

e. *proc_premiaUrlop*

Procedura zwiększa pensję o współczynnik 1,1 pracownikom, którzy mają najmniejszą ilość pobranych dni urlopowych. Pensja zostanie podwyższona pracownikom, którzy byli na urlopie mniejszą lub równą ilość dni, co osoba na podanej w argumencie pozycji rankingu wybranych dni urlopowych.

6. Opis stworzonych funkcji

a. *fun_rozkladNaLinii*

Funkcja wyświetla czasy odjazdu na danych przystankach dla zadanej linii

b. *fun_sprzedaneWOkresie*

Funkcja wyświetla statystyki sprzedaży biletów sprzedanych w danym przedziale czasowym

c. *fun_linieLiczbaPrzystankow*

Funkcja wypisuje nazwy linii, które mają więcej przystanków niż zadana liczba

d. *fun_wolniKierowcy*

Funkcja zwracająca tablicę imiona, nazwiska i id kierowców, którzy nie są przydzieleni do żadnej trasy.

e. *fun_sprzedaneWPunkcie*

Funkcja zwracająca ilość sprzedanych biletów każdego typu w danym punkcie sprzedaży.

f. *fun_rozkladPrzystanku*

Funkcja wyświetlająca rozkład jazdy dla danego przystanku (Wszystkie linie przejeżdżające przez podany przystanek)

g. *fun_czasPrzejazdu*

Funkcja zwraca czas przejazdu między podanymi przystankami, na danej linii.

h. *fun_naprawyMechanika*

Funkcja zwraca dane dotyczące mechanika, który wykonał ilość napraw mniejszą lub równą liczbie podanej w argumencie funkcji.

i. *fun_rankingDniUrlopowych*

Funkcja zwraca dane osób, które brały urlop w danym argumentem roku.
Zawiera też informację o ilości wybranych dni urlopowych w tym roku.

j. ***fun_rankingMandatow***

Funkcja zwraca tabelę z danymi osobowymi, ilością mandatów i kwotą do zapłaty dla wszystkich klientów.

7. Skrypt tworzący bazę

Znajduje się w załączonym pliku *generator_bazy.txt*

8. Typowe zapytania

- a. Wyświetl dane techniczne pojazdu, który miał najwięcej napraw (pierwszy rekord)

```
Select * from Dane_tekniczne_pojazdow where Model = ( Select Model from
Pojazdy where ID_Pojazdu = ( Select top(1) ID_Pojazdu Liczba from Naprawy Group
by ID_Pojazdu order by Count(ID_Pojazdu) DESC ))
```

- b. Oblicz średnią pensję kierowców zatrudnionych przed 2000-01-01

```
Select AVG(Pensja) AS [Średnia Pensja] from Dane_osobowe where
ID_Pracownika LIKE 'KA%' AND Data_zatrudnienia <'20000101'
```

- c. Wyświetl dane pojazdów, które nie mają aktualnych przeglądów

```
select * from Przeglady_pojazdow where Data_waznosci < GetDate()
```

- d. wyświetl nazwy przystanków, przez które przejeżdżają więcej niż 2 linie oraz posiadające automat biletowy

```
select PnL.Nazwa_przystanku from Przystanki as P Join Przystanki_na_linii as
PnL on P.Nazwa_przystanku = PnL.Nazwa_przystanku where Automat_biletowy !=0
group by PnL.Nazwa_przystanku having Count(PnL.Nazwa_przystanku) > 2
```

- e. Stwórz raport wyświetlający ilość zakupionych biletów okresowych (Typ biletu, nazwa, ilość)

```
select BO.Typ_Biletu as Typ, TB.Pelna_nazwa as Nazwa, Count (
BO.Typ_biletu) as Ilosc from Bilety_okresowe as BO join Typy_biletow as TB on
BO.Typ_biletu = TB.Typ_biletu group by BO.Typ_biletu, TB.Pelna_nazwa
```

- f. Wyświetl wszystkie linie dzienne

```
select * from Linie where [Linia dzienna] =1
```

- g. Wyświetl wszystkie bezwypadkowe pojazdy

```
select * from Dane_tekniczne_pojazdow as DTP join Pojazdy as P on
DTP.Model = P.Model where P.ID_Pojazdu not in (select ID_Pojazdu from
Historia_usterek )
```

- h. Ilość dni urlopowych wykorzystanych przez danego pracownika

```
select U.ID_Pracownika ID, D.Imie, D.Nazwisko, DATEDIFF(day,
Początek_urlopu,Koniec_urlopu), U.Typ from urlopy as U join Dane_osobowe as D on
U.ID_Pracownika = D.ID_Pracownika
```

9. Indeksy

- a. ***PrzystIndex*** - W tabeli Przystanki_na_linii indeks na kolumnie Nazwa_przystanku
- b. ***UrlopiIndex*** - W tabeli Urlopy indeks na kolumnie Początek_urlopu
- c. ***UsterkiIndex*** - W tabeli Historia_usterek indeks na kolumnie Data_zdarzenia
- d. ***NaprawyIndex*** - W tabeli Naprawy indeks na kolumnie Data_wykonania
- e. ***BiletyJedIndex*** - W tabeli Bilety_jednoprzejazdowe indeks na kolumnie Typ_biletu

10. Załączniki

- a. **Funkcje.txt** – zbiór wszystkich funkcji do projektu
- b. **Generator_bazy.txt** - kod generujący bazę MPK
- c. **Procedury.txt** – zbiór wszystkich procedur
- d. **Wyzwalacze.txt** - zbiór wszystkich wyzwalaczy
- e. **Diagram_MPK.pdf** – diagram obrazujący bazę