

Baza Danych Miejskiego Przedsiębiorstwa Komunikacji

Oskar Kuliński, Wieńczysław Włodyga

Założenia projektu:

Baza danych MPK, obsługująca głównie logistykę pojazdów i bilety. Pojazdy, rozkłady jazdy, linie tramwajowe i autobusowe, kierowców oraz innych pracowników, bilety papierowe i na kartach miejskich, podstawy księgowości takie jak roczny dochód z biletów, urlopy i zwolnienia pracowników, ich umowy, oraz wpłaty na karty miejskie.

Wykorzystany system bazodanowy: Postgres

Ograniczenia:

Brak dostępu do uporządkowanych danych, część danych pochodzi ze strony MPK (rozkłady jazdy, linie, modele tramwajów i autobusów), część z Wikipedii (modele tramwajów i autobusów, zajezdnie, pętle tramwajowe). Dane dotyczące biletów, kart miejskich oraz pracowników są danymi przykładowymi, ponieważ nie są one publicznie dostępne.

Brak dostępu do systemów lokalizacji pojazdów, które pozwalałyby na zdobycie danych o obecnym położeniu pojazdu. W miejsce danych, które powinny pochodzić z takiego systemu, brane jest przewidywane położenie pojazdu, które nie bierze pod uwagę możliwych opóźnień.

Strategia pielęgnacji bazy danych:

Codziennie automatyczny zapisy bazy, w godzinach nocnych. W razie awarii odtwarzanie danych z Write Ahead Log.

Backup na miejscu, na serwerze zapasowym oraz drugi w chmurze.

Utworzone indeksy:

Na każdym kluczu głównym, automatycznie tworzone przez system bazodanowy.

Do tego indeksy hashujące w:

- zwolnieniach na id pracownika, w celu szybszego wyszukiwania zwolnień dla danego pracownika,

- rozkładach autobusowych i tramwajowych na przystanku, w celu szybszego znajdowania rozkładu dla danego przystanku (wywieszane na kartkach na przystankach),

- rozkładach autobusowych i tramwajowych na idKursu i idLinii, w celu szybszego znajdowania godzin odjazdu dla konkretnego kursu konkretnej linii (informacja taka jest np. pokazywana na ekranach wewnątrz pojazdu),

- autobusach i tramwajach na model, w celu sprawnego wyszukiwania pojazdów danego modelu (nie koniecznie często robione, ale utrzymanie indeksu jest tanie, ze względu na rzadkość dodawania nowych pojazdów do floty miasta).

Widoki:

ŁączneZarobki – wyświetla sumę przychodów z biletów papierowych i z kart miejskich w kolejnych latach.

PodwładniPrzełożeni – wyświetla bezpośrednich podwładnych każdego pracownika.

PojazdySerwisowane – wyświetla obecnie serwisowane pojazdy, zarówno autobusy jak i tramwaje.

WszystkieLinie – wyświetla wszystkie linie, co ułatwia uzyskanie dostępu do pola idLinii.

Funkcje:

CzasPodrozy – skrypt napisany w Pythonie, który łączy się z API Google Maps i wyświetla jaki jest czas podróży między przystankami

PokazRocznyZarobek – zwraca zarobek łączny przedsiębiorstwa, który jest obliczany dla roku przekazanego argumentem typu INT.

LokalizacjaPojazdu – zwraca przystanek, na którym pojazd powinien być w momencie wywołania funkcji. Przekazujemy do niej argument typu VARCHAR(10), który odpowiada numerowi pojazdu.

PojazdyZastepcze – wyświetla potencjalne pojazdy zastępcze, jeśli pojazd dozna awarii na trasie. Przekazujemy do niej argument typu VARCHAR(10), który odpowiada numerowi zepsutego pojazdu.

CzyWazny – sprawdza ważność biletu zakupionego przez pasażera pojazdu. Przekazujemy do niej argument typu INT odpowiadający ID biletu.

NajbardziejZapelnionyPojazd – wyświetla dane pojazdu, w którym aktualnie przebywa najwięcej osób na bazie danych z tabeli Bilety.

IleZostaloMiejsc – sprawdza ile zostało miejsc w autobusie, którego ID przekazane jest argumentem typu VARCHAR(10).

PrzystankiNaLinii – wyświetla jakie są przystanki na linii, której ID przekazane jest argumentem typu INT.

Wyzwalacze:

1. Wyzwalacze sprawdzające, czy przed dodaniem lub aktualizowaniem tabel Tramwaje i Autobusy zajezdnia, do której dodajemy pojazd jest czynna i czy nie jest zapelniona.
2. Wyzwalacze sprawdzające, czy przed dodaniem lub aktualizowaniem tabel PrzejazdyAutobusowe i PrzejazdyTramwajowe pojazd przypisywany do przejazdu jest dostępny.
3. Wyzwalacze sprawdzające, czy przed dodaniem lub aktualizowaniem tabel PrzejazdyAutobusowe i PrzejazdyTramwajowe kierowca przypisywany do przejazdu jest dostępny.

4. Wyzwalacze zapewniające więzy integralności, które sprawdzają istnienie pary (linia, kurs)
5. Wyzwalacze dokonujące doładowania salda karty miejskiej lub pobrania adekwatnej do ceny biletu kwoty z tej karty.
6. Wyzwalacze zapewniające więzy integralności poprzez sprawdzenie, czy istnieje pojazd o danym ID, kiedy chcemy dodać bilet do tabeli Bilety.

Procedury:

DoladujKarte – doładowuje saldo karty miejskiej kwotą dodaną w tabeli TransakcjeKartyMiejskie.

KupBilet – pobiera z salda karty miejskiej kwotę równą cenie biletu, jeśli płatność była wykonywana poprzez kartę miejską.

SprawdzStanZajezdni – sprawdza stan zajezdni, do której dodawany jest pojazd.

UsunPrzejazdSprzed90Dni – usuwa przejazd, który jest starszy niż 90 dni.

SprawdzDostepnoscPojazdu – sprawdza dostępność pojazdu, który przypisujemy do danego przejazdu.

SprawdzDostepnoscKierowcy – sprawdza dostępność kierowcy, którego przypisujemy do danego przejazdu.

SprawdzPoprawnoscPojazdu – zapewnia więzy integralności sprawdzając, czy dany pojazd istnieje jeśli dodajemy bilet z danym numerem pojazdu.

SprawdzIstnienieLiniiIKursu – sprawdza istnienie pary (linia, kurs) zapewniając więzy integralności.