

SzybkaPaczka

Jacek Markiewicz, Karol Bielaszka
9 czerwca 2024

Tematyka i założone cele

Tematem projektu jest baza danych dla firmy przewożącej paczki pomiędzy paczkomatami. Baza przechowuje informacje dotyczące klientów, pracowników, pojazdów oraz przesyłanych paczek. Aplikacja będzie udostępniać interfejs umożliwiający komunikację pomiędzy bazą a klientami, kurierami i aplikacją do zarządzania logistyką (aplikacji logistycznej nie będziemy implementować).

Schemat bazy

Schemat bazy możemy podzielić na trzy główne części:

1. **Osoby** - składa się z osób, uprawnień jakie posiadają, stanowisk jakich pracują oraz informacji o stawieniu się pracownika w pracy i zakończeniu pracy w danym dniu.
2. **Pojazdy** - składa się z pojazdów, które posiada firma, ich rodzajów, pojemności, wymaganych serwisów (uzależnionych od rodzaju pojazdu), przeprowadzonych serwisów, usterek i ich napraw oraz wymaganych uprawnień do prowadzenia pojazdu.
3. **Zlecenia i kursy** - składają się z informacji o zleceniach i transporcie paczek pomiędzy paczkomatami i magazynami. Dokładny opis przewozu paczki znajduje się poniżej.

Historia zlecenia

1. Zgłoszenie zlecenia przez klienta, odnotowane w tabeli zlecenia;
2. Włożenie paczki przez klienta do skrytki, odnotowane w nadaniach;
3. Zabranie paczki przez kuriera (odnotowane w zlecenia_kursy) i przewiezienie do magazynu (odnotowane w zlecenia_magazyny) do którego przynależy dany paczkomat;
4. Następnie paczka zostaje przewieziona do odpowiedniego magazynu, w pobliżu docelowego paczkomatu (zapakowanie do pojazdu odnotowane w zlecenia_kursy i załadowanie do magazynu w zlecenia_magazyny);
5. W międzyczasie aplikacja logistyczna rezerwuje pustą skrytkę w paczkomacie, do którego ma trafić przesyłka (odnotowane w odbiory, obie daty równe NULL, jest to równoznaczne z zarezerwowaniem skrytki na dane zlecenie);
6. Zawieszenie paczki do odpowiedniego paczkomatu i włożenie paczki do zarezerwowanej skrytki (odnotowane w odbiory jako data_dostarczenia);
7. Odebranie paczki z paczkomatu przez odbiorcę (odnotowane w odbiory jako data_odbioru).

Historia kursu

1. Warstwa logistyczna tworzy zlecenie do którego przyporządkowuje magazyn początkowy (magazyn_start), magazyn docelowy, kierowcę i pojazd znajdujących się w magazynie_start, każdy kurs odbywa się pomiędzy dwoma magazynami (być może między jednym i tym samym, oznacza to wtedy, że dany pojazd zbiera/rozwozi paczki między magazynem a pobliskimi paczkomatami);
2. Do pojazdu przypisanego do kursu zostają zapakowane odpowiednie paczki (odnotowane w zlecenia_kursy);
3. Kurs wyjeżdża z magazynu (odnotowane w kursy jako data_wyjazdu);
4. *Jeżeli kurs zbiera/rozwozi paczki z/do paczkomatów:
 - Kurs zatrzymuje się przy paczkomacie (odnotowane w kursy_paczkomaty jako data_przyjazdu);
 - Następuje przepakowanie odpowiednich paczek (odnotowane w zlecenia_kursy lub odbiory);
 - Kurs odjeżdża od paczkomatu i rusza w dalszą trasę (odnotowane w kursy_paczkomaty jako data_odjazdu);
5. Kurs przyjeżdża do docelowego magazynu (odnotowane w kursy jako data_przyjazdu);
6. Następuje wypakowanie wszystkich paczek do magazynu (odnotowane w zlecenia_magazyny);
7. Kierowca deklaruje, że cały pojazd został rozpakowany (odnotowane w deklaracja_wypakowan), nie możemy tego wnioskować z tabeli zlecenia_magazyny, ponieważ paczka mogła się zgubić podczas transportu.

Problemy i postawione założenia

- Zdecydowaliśmy nie zajmować się systemem wypłat oraz harmonogramem pracowników.
- Początkowo chcieliśmy trzymać graf magazynów i obliczać optymalną trasę dla każdego zlecenia, jednak zdecydowaliśmy pozostawić to warstwie logistycznej, która mogła by pracować na naszej bazie.
- Zakładamy, że każdy magazyn może pomieścić nieskończoną liczbę paczek.
- Zakładamy, że pesel jest niezmienny.
- Zakładamy, że rejestracja pojazdu jest niezmienna.
- Jeśli pojazd z pewnych przyczyn zniknie (np. zostanie ukradziony w trakcie odbywania kursu) to jest to traktowane jako usterka (której naprawą jest odzyskanie pojazdu).
- Każdy kurs ma zawsze przypisanego kierowcę. Pozycję pojazdów stwierdzamy na podstawie ostatniego przejazdu. Niestety trudno stwierdzić, gdzie się znajduje pojazd, zanim odbędzie jakiegokolwiek kurs. W tym celu w tabeli *osoby* znajduje się osoba "N.N." odpowiedzialna za wykonanie kursów z magazynu do siebie samego każdym pojazdem. Rozwiązuje to problem stwierdzenia początkowej lokalizacji pojazdów.

Użyte struktury

W bazie zastosowaliśmy następujące narzędzia:

- Indeksy - zwiększają wydajność wyszukiwania informacji w często sprawdzanych tabelach.
- Wyzwalacze - nie dopuszczają do niepoprawnego wstawienia/zedytowania krotki w bazie; są również odpowiedzialne za automatyczne wstawianie powiązanych krotek do innych tabel
- Zasady - nie pozwalają na usuwanie czy edycję rekordów z tabel.
- Funkcje - w związku ze stopniem zaawansowania bazy przydatne było napisanie paru pomocniczych funkcji.

Nie zdecydowaliśmy się na zastosowanie widoków.

Uruchomienie aplikacji

W celu uruchomienia aplikacji będzie konieczna Java oraz IDE umożliwiające otworenie projektu i uruchomienie go, w naszym przypadku jest to IntelliJ IDEA. Aby uruchomić aplikację należy:

1. Najpierw należy załadować dane z *create.sql* do psql (tj. wykonać komendę "psql < create.sql")
2. Rozpakować *src.tar*
3. Otworzyć środowisko IntelliJ IDEA
4. Otworzyć w nim projekt (Open Folder -> nasz rozpakowany folder)
5. Szukamy w projekcie po lewej pliku DesktopLauncher (./desktop/src/com/mygdx/sp/DesktopLauncher.java) i klikamy na niego dwukrotnie, by był aktywnym plikiem
6. Jeśli IntelliJIDEA zapyta o skonfigurowanie SDK (powiadomienie w prawym górnym rogu) bądź że znalazł skrypty Gradle (powiadomienie w prawym dolnym rogu) to akceptujemy
7. Zieloną strzałką u góry po prawej uruchamiamy DesktopLauncher (jeśli nie jest to należy wybrać obok strzałki opcję "Current file")
8. Po chwili powinna uruchomić się aplikacja

Aplikacja

Aplikacja ma prostym interfejs do komunikacji z bazą danych. W pierwszej kolejności należy wybrać rodzaj operacji/zapytania jakie chcemy przedsięwziąć. Następnie należy wypełnić ewentualne dane do zapytania. Po wciśnięciu przycisku "Submit" powinien ukazać się wynik zapytania (albo "UPDATE"/"INSERT" w przypadku zmiany stanu bazy).

Mimo swojej prostoty aplikacja umożliwia wszystkie operacje potrzebne do aktualizowania bazy. W szczególności możliwe jest wykonanie całego procesu przewozu paczki: od momentu jej nadania aż do dostarczenia jej do paczkomatu docelowego i odebrania przez adresta.