

1. Rozważmy bazę danych o następującym schemacie:

```
pracownicy (id_pracownika INTEGER GENERATED ALWAYS AS IDENTITY,  
            przelozony INTEGER, imie TEXT, nazwisko TEXT,  
            nr_stanowiska INTEGER, data_zatrudnienia DATE, pensja DECIMAL(7,2))
```

```
pensje (nr_stanowiska INTEGER, dolna DECIMAL(7,2), gorna DECIMAL(7,2))
```

- (a) Stwórz bazę danych działu kadr o tym schemacie i załóż w niej podstawowe więzy integralności (klucze główne, obce, ...). Wypełnij tabelę **pensje** stanowiskami o numerach od 0 do 4 o wybranych przez siebie granicach pensji.
- (b) Napisz funkcję **zwolnij**, która usuwa danego pracownika z bazy.
- (c) Napisz funkcję **zatrudnij**, która dodaje nowego pracownika do bazy. Numer przełożonego powinien być argumentem funkcji, data zatrudnienia powinna być ustawiona na bieżącą. Nowy pracownik powinien być zawsze zatrudniany na stanowisku o numerze 0 i mieć najniższą możliwą pensję.
- (d) Napisz funkcję **awans**, która awansuje pracownika na dane stanowisko. Pensja powinna być ustawiona jako najniższa możliwa na tym stanowisku.
- (e) Napisz funkcję zmieniającą pensję u danego pracownika. Zadbaj o to, żeby ta funkcja nie mogła wprowadzić niespójności między numerem stanowiska a pensją.
- (f) Napisz wyzwalacz, który przed modyfikacją tabeli **pracownicy** sprawdza, czy pensja jest w odpowiednim przedziale dla danego stanowiska. Zauważ, że taki wyzwalacz pozwoli uprościć funkcję z poprzedniego punktu.
- (g) Napisz wyzwalacz, który sprawdza, że przełożony zarabia więcej niż podwładny.
- (h) Napisz wyzwalacz, który sprawdza, że przełożony ma dłuższy staż niż podwładny.

2. Rozważmy bazę danych o następującym schemacie:

```
klienci (id_klienta INTEGER GENERATED ALWAYS AS IDENTITY,  
         imie TEXT, nazwisko TEXT)
```

```
produkty (id_produktu INTEGER GENERATED ALWAYS AS IDENTITY,  
          cena DECIMAL(7,2), opis TEXT)
```

```
egzemplarze (id_egzemplarza INTEGER GENERATED ALWAYS AS IDENTITY,  
             id_produktu INTEGER, data_waznosci DATE)
```

```
kupiono (id_egzemplarza INTEGER, id_produktu INTEGER, data_waznosci DATE,  
         id_klienta INTEGER, czas TIMESTAMPTZ)
```

```
stan (id_produktu INTEGER, liczba_egzemplarzy INTEGER)
```

- (a) Stwórz bazę danych sklepu o tym schemacie i załóż w niej podstawowe więzy integralności (klucze główne, obce, ...). Załóż, że egzemplarz po zakupie jest usuwany z tabeli **egzemplarze** i trafia do tabeli **kupiono**.
- (b) Napisz funkcję **dostawa**, która dodaje do bazy **n** nowych egzemplarzy danego produktu z określoną datą ważności.
- (c) Napisz wyzwalacz, który uaktualnia zawartość tabeli **stan** po modyfikacji tabeli **egzemplarze**.
- (d) Zauważ, że dużo wygodniej byłoby zamienić tabelę **stan** na widok. Zamień ją na odpowiedni widok (dbając o to, żeby nie zgubić produktów, których liczba egzemplarzy wynosi 0).
- (e) Napisz funkcję **sprzedaz**, która sprzedaje danemu klientowi **n** egzemplarzy danego produktu. Funkcja powinna w odpowiedniej pętli wpisywać wiersze do tabeli **kupiono** oraz usuwać wiersze z tabeli **egzemplarze**.
- (f) Napisz wyzwalacz na tabeli **kupiono**, który przed wpisaniem krotki sprawdza, czy nie wygasła data ważności sprzedawanego egzemplarza. Jeśli wygasła, to wiersz nie powinien zostać wprowadzony.