

Zad. 1. (10+10) Zaprojektuj bazę danych *13 posterunek*. Posterunek podzielony jest na wydziały (każdy ma swoją unikatową nazwę, kryptonim służący nadawaniu nazw sprawom, lokalizację itp.), a każdy wydział podzielony jest na podwydziały (każdy ma swoją unikatową nazwę, kryptonim służący nadawaniu nazw sprawom itp.) Każdy z policjantów zatrudnionych na tym posterunku pracuje w jednym wydziale, chociaż w trakcie pracy może on być przeniesiony do innego wydziału. Baza umożliwia obliczanie pensji policjantów – zależy ona od stopnia, jaki ma dany policjant i od długości czasu pracy policjanta na danym stopniu (każdy stopień ma swoją stawkę i swój przelicznik), od stażu pracy policjanta i od stałego, ale zależnego od pełnionej funkcji (np. komendant, zastępca komendanta, szef wydziału czy podwydziału) dodatku funkcyjnego. Policjanci pracują nad sprawami – do każdej zgłoszonej sprawy, której nadawana jest nazwa zależna od rodzaju sprawy, wyznaczana jest para policjantów, z których jeden jest szefem tej sprawy, która ma swój status (np. jest w trakcie, zawieszona, umorzona, rozwiązana, itp.)

Zad. 2 (5) Stwórz widok *głowy*, który dla bazy *world* pokazywał będzie kraj, jego największe i najmniejsze miasto oraz rządzącego tym krajem. Dwie początkowe litery nazwy rządzącego i dwie końcowe litery (jako jeden ciąg znaków) mają zawierać się w nazwie dowolnego kraju (niekoniecznie tego rządzącego) – nazwa tego kraju również ma być pokazywana.

Zad. 3 (5) Do bazy *world* napisz procedurę, której argumentami będzie ciąg znaków i litera. Przy pierwszym wywołaniu procedura utworzy tabelę oraz (ta część jest wspólna dla każdego wywołania) doda do tej tabeli wpis: kraj, którego nazwa rozpoczyna się podanym (pierwszy argument) ciągiem znaków, miasto tego kraju, które w porządku alfabetycznym byłoby pierwsze na podaną literę (drugi argument). Jeśli nie ma takiego miasta, to wpisane będzie miasto następne w kolejności alfabetycznej, a jeśli ciągle takiego nie ma, to tu rozwiązujący ma wolną rękę. Ostatnią dodaną wartością jest data i czas wywołania procedury. Procedura wypisuje również tak na nowo uzupełnioną tabelę.

Zad. 4 (5) Napisz wyzwalacz, który przy próbie zmiany imienia w tabeli *osoby*, jeśli zmienione imię kończy się samogłoską, to dodaje je „od końca” (za pełną pulę punktów nie używamy funkcji *reverse*), np. *róża* -> *azór*, a jeśli literą *q*, *x* lub *v*, to takie imię zamienia wg schematu *bEAtrIx* -> *BeatriX*.

Zad. 5 (5) Podaj przykład tabeli, która nie spełnia 2NF i ma dwu- lub trójargumentowy klucz główny. Podaj użyteczności, której nie posiada wyzwalacz. Podaj przykład widoku dyndającego. Podaj przykład zależności całkowicie nietrywialnej, która nie jest zależnością nietrywialną. Podaj przykład tabeli posiadającej klucz kandydujący.