

Úkoly

1. Napište trigger `TG_ZARAZENI`, který při vložení záznamu do tabulky `ZARAZENI` zkontroluje, zda je kombinace atributů `cl_funkce` a `id_of` smysluplná. To znamená, že pokud `cl_funkce = 0`, pak `id_of` musí být existujícím ID orgánu (`ORGAN.id_organ`). Pokud `cl_funkce = 1`, pak `id_of` musí být existujícím ID funkce (`FUNKCE.id_funkce`). Dále musí platit, že `cl_funkce` $\in \{0,1\}$. Při nesplnění těchto podmínek dojde k výpisu chybové hlášky na konzolový výstup a vygenerování výjimky.

Demonstrujte funkcionalitu triggeru na následující posloupnosti `INSERT` příkazů:

- (a) Zařaďte osobu s ID 6755 do existujícího orgánu s ID 1229. Vložení proběhne v pořádku.
 - (b) Zařaďte osobu s ID 6755 do neexistující funkce s ID 1229. Trigger zařazení zabrání a vypíše chybové hlášení o neexistující funkci.
 - (c) Pokuste se vytvořit zařazení s `cl_funkce = 2`. Trigger musí záznamu zabránit s chybovým hlášením o neplatném `cl_funkce`.
2. Napište uloženou proceduru `P_PRINT_HLASOVANI` s parametrem `p_id_hlasovani`, která vypíše statistiky hlasování s daným ID. Výpis bude obsahovat seznam poslanců seskupený dle výsledku hlasování. Procedura nesmí používat zanořený kurzor (tj. kurzor v kurzoru).

Příklad výstupu pro `p_id_hlasovani = 29714`:

```
vysledek @:
* Pavel Dostál
* Vladimír Mlynář
* Josef Houzák
* Jan Vidím
vysledek A:
* Petra Buzková
* Květoslava Čelišová
* Václav Exner
* Václav Frank
... (celkem 194 poslanců pro výsledek A)
vysledek B:
* Evžen Snítilý
vysledek F:
* Milan Urban
```

Úkoly

1. Napište uloženou proceduru `P_SET_ZARAZENI_FCE` s parametry `p_id_osoba`, `p_id_funkce` a `p_set`. Procedura neprve zkontroluje, zda `p_id_funkce` je existující ID funkce. Pokud ne, vypíše se chybové hlášení a bude vygenerována výjimka. Dále, při hodnotě `p_set = 'Y'` procedura zařadí danou osobu `p_id_osoba` k dané funkci `p_id_funkce` s tím, že atribut `od_o` bude nastaven dle aktuálního data. K zařazení dojde pouze v případě, že daná osoba k dané funkci aktuálně není přiřazena. Při hodnotě `p_set = 'N'` naopak procedura zruší zařazení osoby k dané funkci, opět, pouze pokud takové zařazení existuje. Procedura vypíše, zda došlo k vytvoření zařazení, zrušení zařazení, nebo neproběhla žádná operace.

Demonstrujte proceduru na následující posloupnosti volání:

- (a) Zařaďte osobu s ID 6755 do funkce s ID 2930. Dojde k výpisu hlášky o vytvoření zařazení.
- (b) Zařaďte osobu s ID 6755 do funkce s ID 2930. Dojde k výpisu hlášky, že neproběhla žádná operace.
- (c) Zrušte zařazení osoby s ID 6755 z funkce s ID 2930. Dojde k výpisu hlášky o zrušení zařazení.
- (d) Zrušte zařazení osoby s ID 6755 z funkce s ID 2930. Dojde k výpisu hlášky, že neproběhla žádná operace.

2. Napište uloženou funkci `F_COPY_SCHUZE` s parametry `p_id_schuze`, `p_od_schuze` a `p_do_schuze`. Funkce zkopíruje schůzi v tabulce `SCHUZE` s daným `p_id_schuze` včetně jejích bodů v tabulce `BOD_SCHUZE`. Zkopírovaný záznam schůze bude mít atributy `od_schuze` a `do_schuze` nastavené dle předaných parametrů `p_od_schuze` a `p_do_schuze`. Pro ID nového záznamu použijte nejvyšší ID v tabulce `schuze` zvětšené o 1. Dále je potřeba přidělit hodnotu `schuze` představující pořadí schůze pro daný orgán. Opět použijte nejvyšší hodnotu `schuze` pro daný orgán zvětšenou o 1. Názvy zkopírovaných bodů (`BOD_SCHUZE.uplny_naz`) budou oproti originálu navíc obsahovat příponu `'... (kopie)'`. Funkce vrátí ID nového záznamu.

Demonstrujte volání vytvoření funkce v anonymní proceduře pro schůzi s ID 734 a libovolná data `od / do`. Anonymní procedura vypíše nově vzniklé ID schůze a počet záznamů v tabulce `BOD_SCHUZE` odpovídajících nové schůzi, kterých bude pro kopii schůze 734 celkem 10.