

---

**Úkoly**

1. Napište uloženou proceduru `P_ROZDEL_FUNKCE` s parametry `p_funkce_od DATE` a `p_organ_od DATE`. Procedura provede následně kroky. Zkontroluje správnost parametrů `p_funkce_od`, `p_funkce_do`, tj. `p_funkce_od` i `p_funkce_do` a ani jeden z údajů není v budoucnosti. Pokud ano, procedura vyhodí chybu a vypíše ji. Zkontroluje pomocí systémového katalogu (tabulka `user_tables`) existenci tabulky `zarazeni_funkce`. Pokud existuje, odstraní ji. Vytvoří novou tabulku `zarazeni_funkce` s atributy: (`id_osoba INT`, `id_funkce INT`, `funkce_od DATE`, `funkce_do DATE`). Do tabulky `zarazeni_funkce` se pomocí kurzoru přkopírují záznamy z tabulky `zarazeni` pro funkce (tj. `cl_funkce = 1`), dle datumu z parametrů `p_funkce_od` a `p_funkce_do`. Při vkládání použijte vázání proměnných. Procedura zkontroluje a vypíše počet přkopírovaných záznamů. Příklady volání

```
p_rozdel_funkce(to_date('2024-02-01', 'YYYY-MM-DD'), to_date('2022-02-01', 'YYYY-MM-DD'));
```

```
p_rozdel_funkce(to_date('2024-02-01', 'YYYY-MM-DD'), to_date('2025-02-01', 'YYYY-MM-DD'));
```

```
p_rozdel_funkce(to_date('2008-01-01', 'YYYY-MM-DD'), to_date('2022-12-31', 'YYYY-MM-DD'));
```

– Vypis: Vloženo 30 záznamu z 30

2. Napište uloženou proceduru jako transakci `P_POSLANEC_OMLUV` s parametry `p_poslanec_id INT`, `p_od TIMESTAMP` a `p_do TIMESTAMP`. Procedura zkontroluje, zda ID poslance je správné (tj. v tabulce `poslanec` existuje záznam s daným ID). Pokud ne, procedura vypíše chybu. Dále procedura zkontroluje, zda v daný den poslanec již není omluven (tj. zda-li existuje záznam pro omluvu poslance na daný den, čas ignorujte), pokud ano, procedura vypíše chybu. Dále musí procedura zkontrolovat, zda parametr `p_od` není větší než `p_do`. Pokud ano, procedura vypíše chybu. Procedura by také měla zkontrolovat, že poslanec nemá žádné hlasování v daný den a časovém rozsahu omluvy. Pokud ano, procedura vyhodí výjimku. Pokud ne, procedura vloží záznam pro omluvu poslance. Příklady volání

```
declare
```

```
    v_od TIMESTAMP;
```

```
    v_do TIMESTAMP;
```

```
begin
```

```
    v_od := to_timestamp('2024-03-01 09:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    v_do := to_timestamp('2024-03-01 16:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    p_poslanec_omluv(2050, v_od, v_do);
```

```
    -- OUTPUT: Poslanec neexistuje!
```

```
    p_poslanec_omluv(1940, v_do, v_od);
```

```
    -- OUTPUT: Hodnota DO je mensi nez hodnota OD
```

```
    v_od := to_timestamp('2023-09-15 13:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    v_do := to_timestamp('2023-09-15 14:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    p_poslanec_omluv(1940, v_od, v_do);
```

```
    -- OUTPUT: Poslanec je jiz omluven!
```

```
    v_od := to_timestamp('2023-09-26 13:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    v_do := to_timestamp('2023-09-26 15:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    p_poslanec_omluv(1940, v_od, v_do);
```

```
    -- OUTPUT: Poslanec ma v dane dobe hlasovani a nemuze byt omluven
```

```
    v_od := to_timestamp('2024-01-01 13:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    v_do := to_timestamp('2024-01-01 15:00:00', 'YYYY-MM-DD HH24:MI:SS');
```

```
    p_poslanec_omluv(1940, v_od, v_do);
```

```
    -- OUTPUT: Omluva zaznamenana.
```

```
end;
```

---

**Úkoly**

1. Napište uloženou proceduru `P_ROZDEL_ORGAN` s parametry `p_organ_od DATE` a `p_organ_do DATE`. Procedura provede kroky: Zkontroluje správnost parametrů `p_organ_od`, `p_organ_do`, tj. `p_organ_od` ; `p_organ_do` a ani jeden z údajů není v budoucnosti. Pokud ano, procedura vyhodí chybu a vypíše ji. Zkontroluje pomocí systémového katalogu (tabulka `user_tables`) existenci tabulky `zarazeni_organ`. Pokud existuje, odstraní ji. Vytvoří novou tabulku `zarazeni_organ` s atributy (`id_osoba INT FK`, `id_organ INT FK`, `organ_od DATE`, `organ_do DATE`). Do tabulky `zarazeni_organ` se pomocí kurzoru překopírují záznamy z tabulky `zarazeni` pro organ (tj. `cl_funkce = 0`), dle datumu z parametrů `p_organ_od` a `p_organ_do`. Při vkládání použijte vázání proměnných. Procedura zkontroluje a vypíše počet překopírovaných záznamů. Příklady volání  
`p_rozdel_organ(to_date('2024-02-01', 'YYYY-MM-DD'), to_date('2022-02-01', 'YYYY-MM-DD'));`  
– Vypis: Spatne zadane parametry `p_funkce_od` nebo `p_funkce_do`  
`p_rozdel_organ(to_date('2024-02-01', 'YYYY-MM-DD'), to_date('2025-02-01', 'YYYY-MM-DD'));`  
– Vypis: Spatne zadane parametry `p_funkce_od` nebo `p_funkce_do`  
`p_rozdel_organ(to_date('2008-01-01', 'YYYY-MM-DD'), to_date('2022-12-31', 'YYYY-MM-DD'));`  
– Vypis: Vloženo 30 záznamu z 30
2. Napište uloženou proceduru `P_POSLANEC_HLASUJ` s parametry `p_poslanec_id INT`, `p_hlasovani_id INT`, `p_vysledek CHAR` a `p_cas TIMESTAMP`. Procedura jako transakce provede následující: Zkontroluje, zda jsou parametry `p_poslanec_id` a `p_hlasovani_id` správné záznamy hlasování a poslance (existují záznamy v tabulkách `poslanec` a `hlasovani`). Pokud ne, procedura vypíše chybu. Zkontroluje, zda parametr `p_vysledek` je jedna z možností "A"(pro), "B"(proti), "X"(zdržel). Pokud ne, procedura vypíše chybu. Zkontroluje, zda parametr `p_cas` není v budoucnosti. Pokud ano, procedura vypíše chybu. Zkontroluje, zda není v daný den a čas poslanec omluven. Pokud je omluven, procedura vyhodí výjimku. Pokud není, smaže existující hlasování poslance, zapíše nové hlasování a aktualizuje sloupec odpovědi (pro, proti, zdrzel) v tabulce hlasování. Příklady volání

```
declare
    v_cas TIMESTAMP := to_timestamp('2023-09-06 16:00:00', 'YYYY-MM-DD HH24:MI:SS');
begin
    p_poslanec_hlasuj(2050, 81531, v_cas, 'X');
    -- OUTPUT: Poslanec neexistuje!
    p_poslanec_hlasuj(1940, 100000, v_cas, 'B');
    -- OUTPUT: Hlasování neexistuje
    p_poslanec_hlasuj(1940, 81531, v_cas, 'Z');
    -- OUTPUT: Výsledek nemá platnou hodnotu
    p_poslanec_hlasuj(1940, 81531, v_cas, 'A');
    -- OUTPUT: Poslanec je na dané hlasování omluven!
    v_cas := to_timestamp('2025-09-06 16:00:00', 'YYYY-MM-DD HH24:MI:SS');
    p_poslanec_hlasuj(1940, 81531, v_cas, 'A');
    -- OUTPUT: Čas hlasování je v budoucnosti!
    v_cas := to_timestamp('2023-09-26 14:18:00', 'YYYY-MM-DD HH24:MI:SS');
    p_poslanec_hlasuj(1940, 81531, v_cas, 'A');
    -- OUTPUT: Hlasování bylo zaznamenáno.
end;
```