

Úkoly

1. Vytvořte PL/SQL proceduru nazvanou `PCreateOsoba2`, která má dva vstupní parametry: `p_minBornDate` a `p_maxBornDate`. Oba parametry budou textovou reprezentací data ve formátu `YYYY-MM-DD`.

Procedura by měla provést následující kroky:

- Zkontrolovat, zda je zadané časové rozmezí platné (tedy není přehozeno datum od a do). Pokud není, vypsat chybu a vyvolat výjimku.
- Zjistit, zda již existuje tabulka s názvem `Osoba2`. Pokud ano, tabulku smazat (využijte systémovou tabulku `user_tables`, která obsahuje informace o všech tabulkách v databázi).
- Vytvořit tabulku `Osoba2` se stejnou strukturou jako má tabulka `Osoba`.
- Zkopírovat záznamy z tabulky `Osoba` do nově vytvořené tabulky `Osoba2` tak, aby narození každé osoby bylo mezi `p_minBornDate` a `p_maxBornDate`.
- Spočítat počet zkopírovaných záznamů do tabulky `Osoba2` a vypsat ho.
- Ukázkový výstup pro období od 1. 1. 1980 do 1. 1. 1990:
Počet zkopírovaných záznamů: 88

(8 bodů)

2. Napište funkci `FExportOmluv`, která má tři parametry: `f_idOsoba`, `f_minDate` a `f_maxDate`. Funkce má vrátit řetězec obsahující informace o osobě a seznam dat, kdy byla omluvena (viz. níže).

- První řádek výsledného řetězce bude obsahovat jméno a příjmení osoby.
- Druhý řádek bude obsahovat pohlaví (MUŽ, nebo ŽENA). Pro odřádkování použijte `CHR(10)` (vrátí symbol pro nový řádek).
- Na dalších řádcích budou vypsaný všechny dny omluv dané osoby (z tabulky `Omluva`, které jsou v období mezi daty `f_minDate` a `f_maxDate` (včetně). Pracujte pouze se sloupcem `DEN` z tabulky `Omluva`.
- Pokud osoba neexistuje, bude vrácen řetězec `NO_DATA`.
- Ukázkový výstup pro osobu s ID 6153 a období od 1. 1. 2015 do 1. 6. 2015:
Martin Sedlář
MUŽ
28/04/15
29/04/15
30/04/15

(8 bodů)

Úkoly

1. Vytvořte proceduru `pCreateSchuze`, která splňuje následující:

- Přijímá jeden parametr: `p_idOrgan` stejného typu jako atribut `id_organ` v tabulce `Organ`.
- Nejdříve bude vypsán název orgánu a jeho zkratka. V případě neexistence orgánu dojde k vypsání hlášky o tom, že neexistuje.
- Zkontroluje jestli existuje tabulka `Schuze_<id orgánu>` a pokud ano, tak ji smaže (využijte systémovou tabulku `user_tables`, která obsahuje informace o všech tabulkách v databázi).
- Vytvoří tabulku s názvem `Schuze_<id orgánu>` se stejnou strukturou jako má tabulka `Schuze`.
- Dojde ke zkopírování všech záznamů z tabulky `Schuze`, které patří k danému orgánu, do nově vytvořené tabulky.
- Nakonec dojde k vypsání počtu zkopírovaných záznamů.
- Ukázkový výstup pro ID orgánu 172:
Orgán: Poslanecká sněmovna (PSP8)
Počet zkopírovaných záznamů: 120

(8 bodů)

2. Napište funkci s názvem `fExportHlasovani`, která přijímá jeden parametr `f_idHlasovani` stejného typu jako má sloupec `id_hlasovani` v tabulce `Hlasovani`. Tato funkce vrací text viz. níže.

- První řádek textu bude obsahovat název orgánu a datum daného hlasování. Pro odřádkování použijte `CHR(10)` (odpovídající novému řádku).
- Další řádky budou obsahovat vždy jméno, příjmení hlasujícího osoby a jak hlasoval (atribut `vysledek` v tabulce `Hlasovani_poslanec`). Ale pouze pokud není hodnota sloupce `vysledek` rovna `"@"`.
- Pokud dané hlasování neexistuje, bude vrácen text `"NO_DATA"`.
- Ukázkový výstup pro hlasování s ID 53086:
Poslanecká sněmovna (02/02/11) Marek Benda: B
Petr Braný: F
Jiří Petrů: F

(7 bodů)