

Úkoly

1. Vytvořte proceduru s názvem `PSplitVotes`, která rozdělí data hlasování do tabulek podle zadaného rozmezí let.

Specifikace:

- Procedura přijímá dva parametry: `p_fromYear` (začáteční rok) a `p_untilYear` (koncový rok).
- Pokud je zadaný neplatný interval (kdy `p_fromYear` je větší než `p_untilYear`), procedura vypíše chybovou zprávu a vyvolá výjimku.
- Z databáze se odstraní všechny tabulky s názvem začínajícím `Hlasovani_rok_` (využijte systémovou tabulku `user_tables`, která obsahuje informace o všech tabulkách v databázi).
- Pro každý rok v zadaném intervalu:
 - Vytvoří se tabulka s názvem `Hlasovani_rok_X`, kde `X` je rok. Tabulka bude obsahovat pouze atributy `datum`, `pro` a `proti`.
 - Do této tabulky se zkopírují záznamy hlasování z tabulky `hlasovani` pro daný rok (nápopěda: `EXTRACT (YEAR FROM ...)`).
 - Počet záznamů v každé vytvořené tabulce se vypíše.
- Ukázkový výstup pro roky 2010 až 2012:
`HLASOVANI_ROK_2010`: 1791 záznamů
`HLASOVANI_ROK_2011`: 2342 záznamů
`HLASOVANI_ROK_2012`: 1797 záznamů

(8 bodů)

2. Napište PL/SQL kód pro trigger s názvem `TPreventTooManyAbsence`, který je vyvolán před vložením záznamu do tabulky `Omluva`. Trigger by měl provést následující akce:

- Zkontrolovat jestli na daný den již neexistuje jiná omluva daného poslance (pracujte pouze s atributem `den`). Pokud již existuje, dojde k vyvolání výjimky s informací o duplicitě.
- Spočítat počet omluv pro daného poslance v rozsahu jeden měsíc kolem data nově zadané omluvy (nápopěda: `ADD_MONTHS (...)`).
- Pokud počet omluv překročí 10, dojde k vložení záznamu o překročení tohoto limitu do tabulky `Too_many_absence_log`, kterou vytvořte (manuálně, mimo trigger). Tabulka bude obsahovat sloupce `id_poslanec`, `den_omluvy` a `datum_pridani`. Následně bude na výstup vypísána hláška o překročení maximálního počtu omluv (společně s daným počtem). Trigger nebude bránit vložení takovéto omluvy.

(7 bodů)

Úkoly

1. Napište PL/SQL proceduru s názvem `PSplitOsoba`, která má dva vstupní parametry typu `DATE`, `p_fromDate` a `p_untilDate`. Procedura by měla provádět následující kroky:

- Zkontrolujte, zda `p_fromDate` není větší než `p_untilDate`. Pokud ano, vypište chybovou na výstup a vyvolejte výjimku.
- Aktualizujte tabulku `Osoba`, nastavte hodnotu atributu `pohlavi` na 'Z', pokud je aktuálně 'Ž'.
- Projděte všechny tabulky v databázi, které začínají prefixem `Osoba_`, a smažte je (využijte systémovou tabulku `user_tables`, která obsahuje informace o všech tabulkách v databázi).
- Pro každé jedinečné pohlaví v tabulce `Osoba`, které má datum narození v rozmezí mezi `p_fromDate` a `p_untilDate`, proveďte následující kroky:
 - Vytvořte novou tabulku s názvem `Osoba_<pohlaví>` pro dané pohlaví. Tabulka bude obsahovat pouze atributy `jmeno`, `prijmeni` a `narozeni`.
 - Vložte záznamy z tabulky `Osoba`, které svým datem narození spadají do daného období `p_fromDate` a `p_untilDate` do nově vytvořené tabulky pro dané pohlaví.
 - Vypište název nově vytvořené tabulky a počet záznamů do ní vložených.
- Ukázkový výstup pro období od 1. 1. 1980 do 1. 1. 1990:
OSOBA_Z : 22 záznamů
OSOBA_M : 66 záznamů

(8 bodů)

2. Vytvořte tabulku `Too_many_inclusion_log` obsahující sloupce `id_osoba`, `info` typu `VARCHAR(300)` a datum přidání typu `TIMESTAMP`.

Vytvořte trigger `TPreventTooManyInclusion`, který se spouští před vložením záznamu do tabulky `zarazeni`.

Trigger má provést následující kroky:

- Zjistit počet zařazení (včetně nově vkládaného). Jedná se o všechna zařazení dané osoby pro které platí že `od_o` je menší než nově vkládaný `do_o` a `do_o` větší než nově vkládaný `od_o`.
- Pokud je počet zařazení větší jak věk dané osoby podělený 10, dojde k vložení záznamu o nastalé situaci do tabulky `Too_many_inclusion_log`. Do sloupce `info` bude vložen text obsahující jméno a příjmení dané osoby, její věk a celkový počet zařazení. Tento text bude také vypsán na výstup (vždy).
Nápověda: `MONTHS_BETWEEN(...), FLOOR(...)`
- V případě že daná osoba neexistuje, dojde vypsání informativní hlášky o tom, že daná osoba nebyla nalezena.

(7 bodů)