

## Procedury a funkce

Cíl: Procvičit tvorbu uložených procedur a funkcí v PL/SQL.

### Zadání

PF001: Vytvořte funkci, která vrátí počet letů pilota se zadaným id (pilot.pilot\_id). S použitím této funkce napište SQL dotaz, který vrátí jména a příjmení pilotů, kteří letěli alespoň 2x [6 pilotů].

PF002: Napište proceduru, která vypíše na konzoli všechny pasažéry, se kterými letěl pilot se zadaným pilot\_id [např. pilot s pilot\_id = 1 letěl s 8 pasažéry].

PF003: Napište proceduru, která vypíše jména všech zaměstnanců ve formátu: <příjmení><zadaný znak><jméno>. Jako výchozí hodnotu spojovacího znaku použijte '\_', aby nemusel uživatel procedury znak vždy zadávat. Zkuste proceduru zavolat se zadáním a bez zadání spojovacího znaku.

PF004: Napište proceduru, která povýší všechny „Dispečery letového provozu“, kteří do aerolinek nastoupili do zadaného data včetně (zadáno v parametru), na „Hlavní dispečery letového provozu“ (viz. zaměstnání v tabulce zamestnani). Proceduru vyzkoušejte na datu 1.1.2002. Po provedení procedury by měli být 2 dispečeri povýšeni na hlavní dispečery a v aerolinkách by mělo zůstat 13 „obyčejných“ dispečerů letového provozu.

Proceduru napište tak, aby používala příkaz SELECT FOR UPDATE a dotaz UPDATE s WHERE CURRENT OF (při UPDATu zaměstnání je potřeba zaměstnance projít ve FOR cyklu, nebo v LOOP cyklu s příkazem FETCH).

Můžete využít pomocný dotaz:

```
SELECT * FROM AERO.zamestnanec zc, AERO.zamestnani zi WHERE  
zc.zamestnani_id = zi.zamestnani_id  
AND zi.nazev_pozice = 'Dispečer letového provozu'  
AND zc.datum_nastupu <= TO_DATE('1.1.2002', 'DD.MM.YYYY');
```

Zkuste debugovat kód procedury v prostředí Oracle SQL Developer.

PF005: Napište proceduru, která pro zadané tabulky (jména tabulek) vytvoří záložní tabulky s názvem <jméno tabulky>\_bak a do těchto záložních tabulek zduplikuje data z originálních tabulek. Při tvorbě procedury použijte příkaz CREATE TABLE <název tabulky> AS <příkaz SELECT>; a příkaz EXECUTE IMMEDIATE. Seznam názvů tabulek, který bude vstupním parametrem procedury, je potřeba deklarovat jako globální typ – ještě před kódem procedury.

PF006: Napište funkci, která spočítá počet zaměstnanců, kteří byli přijati v zadaném rozmezí roků (např. v intervalu <2002; 2005>) - roky budou parametry funkce [v letech <2002; 2005> bylo přijato 18 zaměstnanců].

PF007: Napište proceduru, která na konzoli vypíše přehlednou tabulku s příjmeními, jmény, platem zaměstnanců a počtem letů, které daný zaměstnanec absolvoval – pokud se jedná o pilota, vypíše skutečný počet letů, u ostatních zaměstnanců vždy 0. Údaje vypíše pouze pro zaměstnance, kteří byli přijati v zadaném rozmezí roků (např. v intervalu <2002; 2005>) - roky budou parametry procedury. Tabulku vykreslete se záhlavím pomocí čárové grafiky. Pro za-

rovnávání údajů do sloupců tabulky použijte funkce LPAD, RPAD [např. pro interval roků nástupu <2002; 2005> se vypíše 18 zaměstnanců, mezi nimi: Messer, Alan, 125000, 1].

Ukázka tabulky:

PRIJMENI	JMENO	PLAT	POCET_LETU
Novák	Jan	50000	1
...			

PF008: Napište funkci, která pro daného zaměstnance (zamestnanec\_id) vrátí počet podřízených na zadané úrovni pod zaměstnancem – použijte parametr uroven\_podrizenych: 1 = přímí podřízení, 2 podřízení přímých podřízených atd. Pro průchod hierarchií lze použít dotaz SELECT – FROM – WHERE – START WITH – CONNECT BY (nebo pouhé spojování tabulek – tabulky zamestnanec jednou v roli nadřízených, jindy v roli podřízených zaměstnanců). Pokud by byla zadaná úroveň 0, vyvolejte ve funkci vlastní výjimku NEPLATNA\_UROVEN a v sekci EXCEPTION tuto výjimku ošetřete výpisem chybového hlášení na konzoli [např. pro zamestnanec\_id = 1 existuje 16 podřízených na úrovni 1 – přímých podřízených, 17 na úrovni 2].

PF009: Vytvořte proceduru vypis\_podrizene, která pro každého zaměstnance vypíše počet podřízených na první a druhé úrovni ve smyslu příkladu PF008 (můžete využít předchozí funkci) [jeden z řádků bude Boss, Michael, 16, 17].

PF010: Vytvořte rekurzivní funkci, která spočítá všechny podřízené daného zaměstnance (tj. od 1. úrovně „hlouběji“, úroveň chápejte ve smyslu zadání příkladu PF008). Nepoužívejte SELECT – FROM – WHERE – START WITH – CONNECT BY [pro zamestnanec\_id = 1 vrátí funkce celkem 33 podřízených].