

# Uložené procedury, funkce a balíky procedur a funkcí, kompilace, spouštění.

Thursday, May 30, 2013 8:16 AM

*Podprogram* je pojmenovaný blok, který může být opakovaně volán a může přebírat aktuální parametry. Typy podprogramů jsou **funkce a procedury**. Procedury a funkce lze sdružovat do logických celků – balíků (package).

Všechny anonymní bloky PL/SQL lze natrvalo uložit do databáze, a to buď ve tvaru procedury, nebo funkce.

## Uložené podprogramy – procedury a funkce

Procedury a funkce mohou být trvale uloženy do DB a mohou být použity jakoukoli aplikací, která s databázovým strojem komunikuje. Jednou přeložený a uložený program patří mezi databázové objekty a může být referován libovolným počtem aplikací.

Uložené podprogramy mohou být samostatné, nebo součástí balíku.

Podprogramy lze volat z:

- DB triggerů
- Jiných uložených podprogramů
- Aplikačních programů zapsaných ve vyšším programovacím jazyce, pro něž existuje předkompilátor (ORACLE Pro)
- Interaktivně (SQL\*Plus) : EXECUTE jmeno\_procedury(parametry)

Uložené procedury a funkce mohou mít parametry, které mohou být:

- IN - vstupní
- OUT - výstupní
- IN OUT – vstupní i výstupní

## Uložená procedura

**Procedura se vstupními parametry:**

- Nelze volat z SQL příkazů
- Na rozdíl od funkce může vrátit více parametrů (IN/OUT)

```
CREATE OR REPLACE PROCEDURE nastav_plat (id IN NUMBER, novy_plat
IN NUMBER) AS
zam_plat NUMBER(5,2);
nema_plat EXCEPTION;
BEGIN
SELECT plat INTO zam_plat FROM osoby WHERE os_cislo = id;
IF zam_plat IS NULL THEN
RAISE nema_plat;
END IF;
EXCEPTION
WHEN nema_plat THEN
UPDATE osoby SET plat = novy_plat WHERE os_cislo = id;
COMMIT;
END;
```

Pro zrušení (smazání) uložené procedury použijeme příkaz:

```
DROP PROCEDURE nastav_plat;
```

## Uložená funkce

Způsob vytvoření uložené funkce (s parametry) ukazuje následující příklad:

```
CREATE OR REPLACE FUNCTION secti(a IN INTEGER, b IN INTEGER)
RETURN INTEGER AS
```

```
BEGIN
RETURN (a + b);
END;
```

Pro zrušení (smazání) uložené funkce použijeme příkaz:  
DROP FUNCTION secti;

## Uložené balíky

Uložené balíky procedur a funkcí slouží ke sdružení logicky spolu souvisejících procedur a funkcí, ale i typů a objektů. Mohou obsahovat i globální proměnné, jejichž platnost je omezena délkou aktuálního spojení s databází.

Definice balíku představuje definici rozhraní balíku (deklarace typů, proměnných, konstant, podmínek definujících nestandardní stavy, kurzorů, podprogramů dostupných zvenčí)(co tam je za funkce parametry, ale bez těla) a těla balíku (implementuje specifikaci)(normálně napsané procedury včetně implementace).

Př. rozhraní:

```
CREATE OR REPLACE PACKAGE arithmetic AS
usage INTEGER;

FUNCTION add(a IN INTEGER, b IN INTEGER) RETURN INTEGER;
FUNCTION sub(a IN INTEGER, b IN INTEGER) RETURN INTEGER;
PROCEDURE inc(a IN OUT INTEGER);
END;
/
```

Př. Těla:

```
CREATE OR REPLACE PACKAGE BODY arithmetic AS

FUNCTION add(a IN INTEGER, b IN INTEGER) RETURN INTEGER IS
BEGIN
usage := usage + 1;
RETURN (a + b);
END;

FUNCTION sub(a IN INTEGER, b IN INTEGER) RETURN INTEGER IS
BEGIN
usage := usage + 1;
RETURN (a - b);
END;

PROCEDURE inc(a IN OUT INTEGER) IS
BEGIN
usage := usage + 1;
a := a + 1;
END;
BEGIN
usage := 0;
END;
/
```

## Kompilace

- Pokud je volána procedura/funkce ve stavu INVALID, kompiluje se automaticky.
  - Pokud se kompilace nezdaří, dojde k výjimce.
- Ruční kompilace
  - ALTER PROCEDURE[FUNCTION] jmproc COMPILE;
  - Pokud se kompilace nezdaří, dojde k výjimce
- SQL\*Plus a chyby kompilace – pomocí něj můžeme zjistit, jaké chyby se vyskytly při kompilaci. Pokud uložení procedury/funkce neproběhlo bez chyb, nelze ji používat a je nutné ji opravit. Pomocí SQL\*Plus příkazů:
  - SHOW ERROR – vypíše popis poslední chyby, na kterou při ukládání (kompilaci) narazil

- SHOW ERR typ jméno – např. SHOW ERR FUNCTION F – pro uvedený objekt
- Pomocí dotazu na tabulku USER\_ERRORS

## Spouštění

Proceduru můžeme zavolat (spustit různými způsoby:

- Pomocí direktivy EXEC – EXEC nastav\_plat(123, 10000);
- V těle jiného PL/SQL bloku

```
BEGIN
...
Nastav_plat(123, 10000);
...
END;
/
```

Funkci můžeme volat také dvěma způsoby:

- V příkazu SELECT – SELECT secti(2, 3) FROM dual;
- V těle jiného PL/SQL bloku

```
DECLARE
a INTEGER;
b INTEGER;
BEGIN
a := 5;
b := secti(a, 2);
dbms_output.put_line('Vysledek : '||b);
END;
/
```

Při volání funkcí/procedur z balíků používáme tečkovou notaci (package.funkce()) pro kvalifikaci jejich jména.

Zabaleny podprogram lze volat z db triggeru, jiného uloženého programu, aplikace napsané pro některý z předkompilátorů, standardních klientských nástrojů (SQL\*Plus)

Standardní balík **STANDARD** = definuje prostředí PL/SQL.

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>