

Proiect de curs PL/SQL

Termen limita: 20 Iunie

Parte 1 : Crearea unui Framework de Debugging în PL/SQL

Scop:

Realizarea unui **framework modular de depanare** pentru PL/SQL, destinat monitorizării, urmăririi și diagnosticării erorilor și fluxurilor de execuție în aplicații de dimensiuni medii și mari.

Definiție – Ce reprezintă un Framework de Debugging în PL/SQL?

Un **framework de debugging** reprezintă un set organizat de:

- mecanisme de logare controlată,
- captură a variabilelor și mesajelor de stare,
- tratare uniformă a excepțiilor,
- structuri reutilizabile pentru monitorizarea execuției codului.

Acesta ajută la îmbunătățirea calității codului, la identificarea rapidă a problemelor și la întreținerea facilă a aplicației.

◆ Componente esențiale

1. Tabel pentru logare (debug_log)

Stochează înregistrările de debug cu detalii despre modul, linia și mesajul survenit în timpul execuției.

```
CREATE TABLE debug_log (  
    log_id          NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    log_time        TIMESTAMP DEFAULT SYSTIMESTAMP,  
    module_name     VARCHAR2(100),  
    line_no         NUMBER,  
    log_message     VARCHAR2(4000),  
    session_id      VARCHAR2(50) DEFAULT SYS_CONTEXT('USERENV', 'SESSIONID')  
);
```

2. Pachet utilitar: debug_utils

Un pachet PL/SQL ce oferă proceduri pentru logare și controlul regimului de debug:

```
-- Activare/dezactivare mod debug
PROCEDURE enable_debug;
PROCEDURE disable_debug;

-- Înregistrare mesaje și valori
PROCEDURE log_msg(p_message VARCHAR2);
PROCEDURE log_variable(p_name VARCHAR2, p_value VARCHAR2);
PROCEDURE log_error(p_proc VARCHAR2, p_err VARCHAR2);
```

Toate procedurile trebuie să verifice dacă debug_mode este activ, pentru a evita logarea inutilă în producție.

Funcționalități dorite

- Activarea/dezactivarea dinamică a modului debug
 - Captură condiționată a mesajelor în funcție de flag-ul global g_debug_mode
 - Înregistrarea numelui modulului, liniei și sesiunii curente
 - Posibilitatea de a extinde framework-ul cu niveluri de logare (INFO, WARN, ERROR)
-

Utilitate

Acest framework este util în:

- proceduri complexe care conțin logică ramificată sau calcule
- monitorizarea execuției în medii de test
- depanarea excepțiilor apărute în runtime
- auditul execuției în aplicații critice (ex. financiar, bancar)

Partea2 : Utilizarea frameworkului în cod

Scop:

Să se creeze o procedură PL/SQL care actualizează salariile angajaților în funcție de valoarea comisionului (commission_pct).



Comportament:

1. Pentru fiecare angajat:
 - Dacă commission_pct are o valoare definită (nu este NULL), salariul va fi mărit cu această valoare aplicată procentual.
 - Dacă commission_pct este NULL, salariul va fi mărit cu **2%** din valoarea actuală.
2. Procedura trebuie să utilizeze un **pachet de debugging (debug_util)**, astfel încât:
 - La fiecare pas important (citire date, calcul, actualizare), să se înregistreze un mesaj de log **doar dacă modul de debug este activ**.



Obiective tehnice

- Scrierea unei proceduri numite, de exemplu, adjust_salaries_by_commission.
- Integrarea cu un pachet de tip debug_util ce conține:
 - Flag de control g_debug_mode
 - Procedura log(p_module_name, p_line, p_message)
- Utilizarea de SQL dinamic sau cursor explicit, în funcție de preferințe.
- Tratarea situației în care commission_pct este NULL cu logică NVL sau IF.



Opțional: Niveluri de logare

Poți adăuga niveluri (DEBUG, INFO, WARN, ERROR) și flag-uri pentru a activa/dezactiva în execuție.

Avantaje ale unui PL/SQL Debugging Framework

Beneficiu	Descriere
✓ Trasabilitate	Urmărești clar ce și când s-a executat
✓ Detectarea rapidă a erorilor	Mesajele logate ajută la localizarea exactă a problemei
✓ Scalabilitate	Reutilizabil în orice procedură/funcție
✓ Control activ/inactiv	Debugul poate fi dezactivat în producție pentru performanță