Tehnici de programare cu baze de date

#1 PL/SQL
Concepte generale

Adrian Runceanu
www.runceanu.ro/adrian

2022

Câteva precizări

Structura cursului

- 2 ore curs – titular curs:

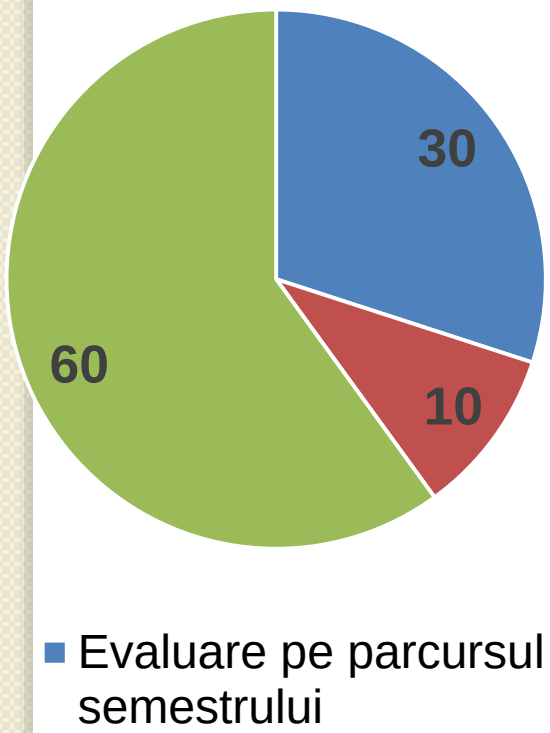
Sef lucr. dr. Adrian Runceanu

- 2 ore laborator – titular aplicaţii practice:

Sef lucr. dr. Adrian Runceanu

Câteva precizări

Procentaje evaluare



Forme de examinare:

- Examen final (verificare) – 60%
- Activitate la curs si laborator – 10%
- Verificare finală lucrări de laborator – 30%

Câteva precizări

Bibliografia necesară cursului:

1. Abiteboul S. etc:"Foundations of Databases", Addison Wesley, 95
2. Date C.J.":An Introduction to Database Systems, ed.8, Addison Wesley, 2004
3. Fotache M. etc. :"Oracle 9i – Ghidul dezvoltarii aplicatiilor profesionale", Polirom, 2003
4. Garcia-Molina H., Ullman J.D. :"Database Systems.The Complete Book", 2000
5. Popescu Ileana: "Modelarea bazelor de date", Editura Tehnica, Bucuresti, 2000
6. Felea V. :"Elemente ale implementarii modelului relational in sisteme de gestiune de baze de date". Ed.MatrixROM, 2007
7. Felea V., Matei C. si Balta M.:"Interogarea bazelor de date. Aplicatii in Oracle si SQL Server", Ed.MatrixROM, 2005

Referinţe electronice:

- http://thor.info.uaic.ro/~felea/
- Documentatia produselor Oracle

Câteva precizări

1) Hub-ul colaborativ Microsoft Teams: cursul FI-3-Tehnici de programare cu baze de date-2022-2023 va contine toate cursurile si laboratoarele.

2) Suport curs - varianta electronică disponibilă pe site-ul: www.runceanu.ro/adrian

Notă: Actualizarea site-ului se face saptamnal.

Tehnici de programare cu baze de date

Concepte generale

GENERALITĂŢI DESPRE BAZE DE DATE

- **Baza de date** este un ansamblu structurat de date coerente, fără redundanţă inutilă, astfel încât acestea pot fi prelucrate eficient de mai mulţi utilizatori într-un mod concurent.

GENERALITĂŢI DESPRE BAZE DE DATE

- Un **sistem de gestiune al bazei de date** (SGBD – *Data Base Management System*) este un *produs software* cu elemente hardware, care asigură interacţiunea cu o bază de date, permiţând *definirea*, *consultarea* şi *actualizarea datelor* din baza de date.

Hardware | Software | Date | Proceduri | Persoane

- Persoane
  - Proiectanţii bazei de date
    - Pr. b.d. logice
    - Pr. b.d. fizice
  - Administratorul de baze de date
  - Utilizatorii bazei de date
    - Programatorii de aplicaţii
    - Utilizatorii finali
      - Specialişti
      - Nespecialişti

PL/SQL

- ***P**rocedural **L**anguage/**S**tructured **Q**uery **L**anguage* (*PL/SQL*) este extensia **procedurală** a limbajului *SQL*.
- ***PL/SQL*** este un limbaj de programare sofisticat care asigură accesarea datelor unei **baze de date relaţionale orientate obiect** şi permite gruparea unei mulţimi de comenzi într-un **bloc** unic de tratare a datelor.
- Programul este format din unul sau mai multe blocuri care pot conţine blocuri imbricate.

PL/SQL include:

1. instrucţiuni *SQL* pentru manipularea datelor şi pentru gestiunea tranzacţiilor
2. instrucţiuni proprii

- Limbajul combină construcţiile procedurale ale unui limbaj de generatia 3 (*LG3)* cu puterea şi flexibilitatea lui *SQL* (*LG4*).
- Combinaţia a generat un limbaj puternic pentru modelarea aplicaţiilor complexe.

PL/SQL extinde *SQL* prin construcţii specifice limbajelor procedurale:

- definirea variabilelor
- declararea tipurilor
- utilizarea structurilor de control
- implementarea procedurilor şi funcţiilor
- introducerea tipurilor obiect şi a metodelor, etc.

PL/SQL oferă posibilităţi moderne de tratare a informaţiei:

- încapsularea datelor
- analiza specială a erorilor
- mascarea informaţiei
- orientarea obiect

Posibilităţile lui *SQL* sunt folosite pentru un acces rafinat la date, iar facilităţile oferite de ***PL/SQL*** sunt folosite pentru fluxul controlului procesării datelor.

Dintre funcţionalităţile limbajului ***PL/SQL*** care determină ca acesta să fie frecvent utilizat se remarcă următoarele facilităţi:

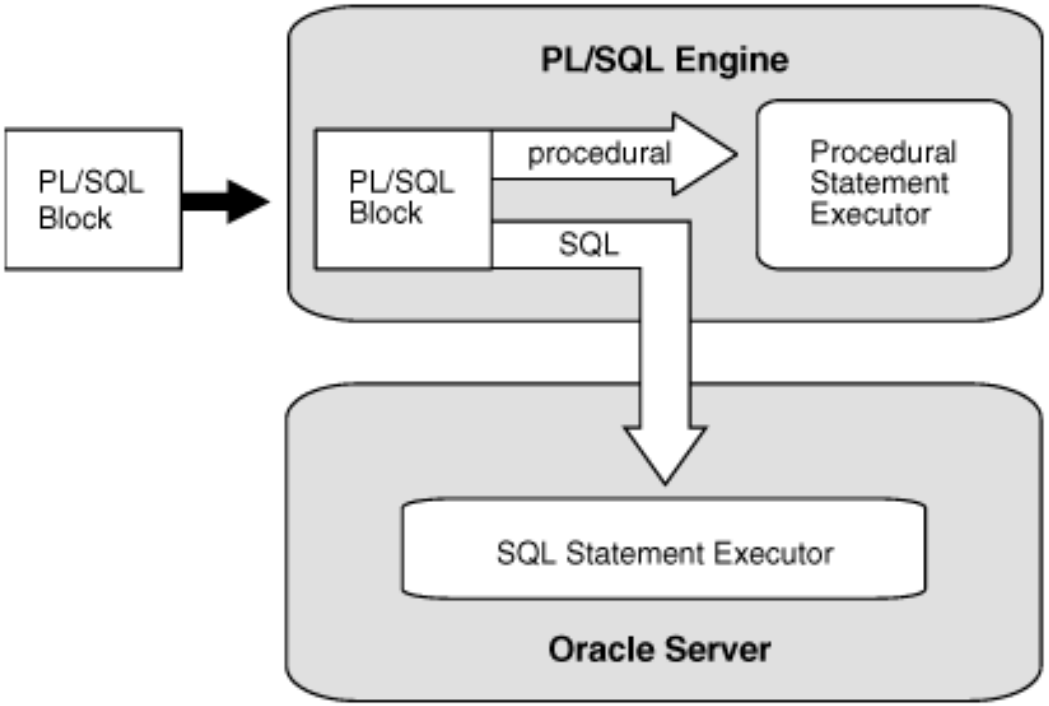
1. integrarea comenzilor *SQL* de bază
2. integrarea cu *server*-ul *Oracle* şi cu aplicatii *Oracle*
3. oferirea unui suport pentru programarea orientată obiect
4. asigurarea securităţii informaţiei
5. definirea şi gestiunea blocurilor de instrucţiuni

6. gestiunea variabilelor, constantelor şi a cursoarelor
7. modularizarea programelor (subprograme, pachete)
8. implementarea şi utilizarea declanşatorilor
9. utilizarea structurilor de control fundamentale
10. detectarea şi gestiunea erorilor de execuţie şi a situaţiilor excepţionale
11. dezvoltarea de *aplicaţii Web*

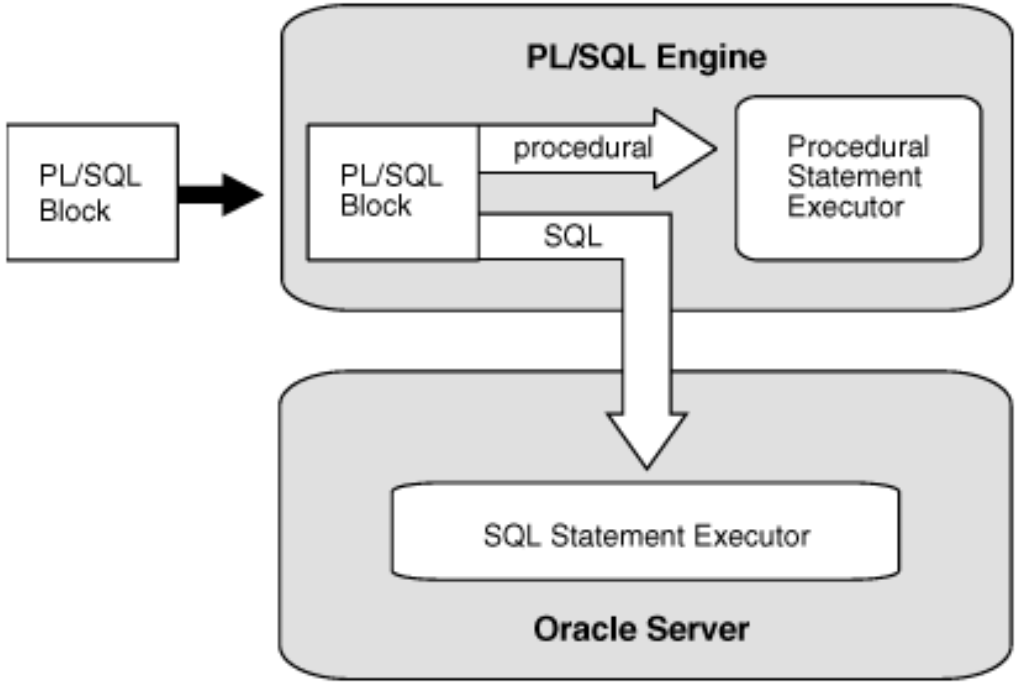
- *PL/SQL* este o tehnologie utilizată de *server*-ul *Oracle* şi de anumite *aplicatii Oracle*.

- Blocurile *PL/SQL* sunt transmise unui **motor *PL/SQL*** şi procesate (compilate şi executate) de acesta.
- *Motorul PL/SQL* poate să se afle pe *server*-ul *Oracle* sau într-un utilitar, iar utilizarea sa depinde de unde se invocă *PL/SQL*.
- Multe *aplicatii Oracle* (inclusiv *Developer/2000*) au propriul lor *motor PL/SQL* care este independent de motorul prezent pe *server*-ul *Oracle*.

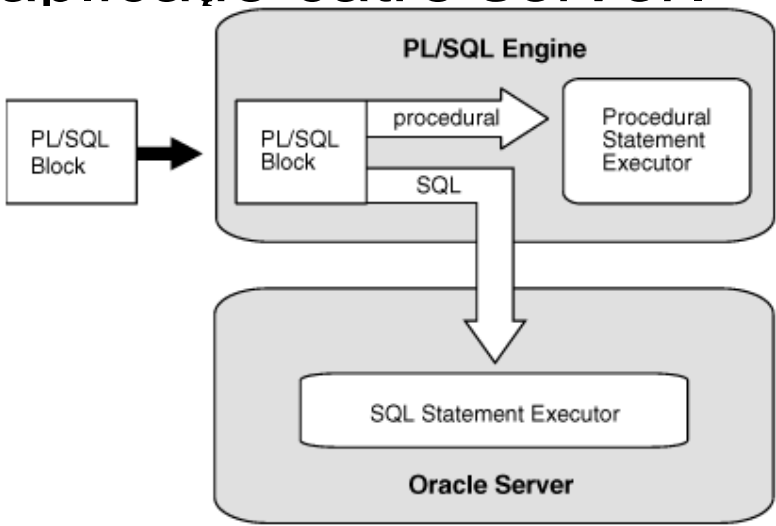
- Blocurile *PL/SQL* pot fi executate:
  1. pe staţia *client* fără interacţiune cu *server*-ul
  2. sau în întregime pe *server*.
- Când blocurile *PL/SQL* sunt referite dintr-un program *PRO**, din *iSQL*Plus*, sau de către *Server Manager*, motorul *PL/SQL* de pe *server*-ul *Oracle* va procesa aceste blocuri.



- Acesta descompune blocul în instrucţiuni *SQL* şi le trimite executorului de instrucţiuni *SQL* (*SQL Statement Executor*) de pe *server*-ul *Oracle*.
- Fără *PL/SQL*, instrucţiunile *SQL* ar fi procesate separat, fiecare la un moment dat, fiecare implicând un apel la *server*-ul *Oracle*.



- Restul comenzilor (procedurale) sunt procesate de către executorul instrucţiunilor procedurale (*PSE – Procedural Statement Executor*) care este în motorul *PL/SQL*.
- *PSE* procesează datele care sunt locale aplicaţiei, reducându-se astfel activitatea de transfer spre *server*-ul *Oracle* şi numărul de cursoare solicitate.
- În felul acesta, este necesar un singur transfer pentru a trimite blocul din aplicaţie către *server*.



O aplicaţie bază de date poate fi structurată în trei părţi:

1. interfaţa utilizator (utilizatorul introduce anumite informaţii şi obţine nişte rezultate în urma executării aplicaţiei)
2. aplicaţia logică efectivă
3. baza de date

Există două modele pentru proiectarea unei aplicaţii bază de date:

1. modelul *client-server* (*two-tier*)
2. modelul *three-tier*

1. Modelul *client-server*

- Multe dintre aplicaţiile baze de date sunt construite folosind modelul clasic *client-server*, descris succint anterior pentru *PL/SQL*.

- Modelul este caracterizat de cele două componente:
  1. *client*
  2. *server*
- *Client*-ul foloseste interfaţa,
- iar *server*-ul conţine baza de date.

1. Modelul *client-server*

- Aplicaţia logică este scindată între *client* şi *server*.
- De remarcat din *această caracteristică fundamentală a modelului că aplicaţia comunică direct cu server*-ul.
- Există un motor *PL/SQL* pe *server*, iar în anumite cazuri şi pe *client*.

- Dacă motorul *PL/SQL* este pe *server*, atunci aplicaţia (care poate fi scrisă în *Pro*C*, *JDBC*, *OCI* sau alte limbaje) care rezidă pe *client* trimite cereri la un *server* de date.
- Cererile sunt rezolvate utilizând *SQL*.
- Diferite cereri *SQL* pot fi grupate într-un bloc *PL/SQL* şi trimise ca o singură entitate *server*-ului.

2. Modelul *three-tier*

În modelul *three-tier* interfaţa utilizator, aplicaţia logică şi baza de date sunt scindate în trei părţi separate:

1. *Client Browser*
2. *Application Server* (*Oracle Internet Application Server – IAS*)
3. *Oracle Database Server*

- Clientul (cel mai frecvent un *browser*) apelează subprograme care generează ca rezultat pagini *HTML*, iar *Application Server* procesează solicitările.
- În *Oracle9i*, aplicaţiile *Forms* şi *Reports* se execută numai ca parte a unui model *three-tier*.
- De exemplu, *Oracle Web Forms* permite desfăşurarea unei aplicaţii într-un mediu *multi-tier Internet* fără a modifica o linie de cod.

- În general, comenzile *SQL* şi *PL/SQL* sunt trimise *server*-ului pentru a fi executate.
- Pentru a realiza acest lucru, trebuie stabilită conectarea la BD.
- Aceasta presupune ca baza să autentifice utilizatorul (parolă şi identificator).
- *PL/SQL* nu conţine o sintaxă pentru realizarea conectării la baza de date.
- Conectarea poate fi realizată de alte componente ale sistemului (de exemplu, *iSQL*Plus*).

- *PL/SQL* are un rol important atât la nivelul *server*-ului *Oracle* (prin subprograme stocate, declanşatori şi pachete), cât şi la nivelul utilitarelor *Oracle* (de exemplu, *Developer/2000* – componenta declanşatori).

- Partea procedurală este atât la nivel de *client*, cât şi la nivel de *server*.

- Cererile însă nu se execută pe staţii *client*, ci numai pe *server*.

PL/SQL constituie fundamentul pentru realizarea diverselor aplicaţii:

- fişiere *iSQL*Plus* (*script file*);
- proceduri şi funcţii stocate (*stored procedure*):
  1. o procedură sau o funcţie stocată este un subprogram *PL/SQL* care poate fi invocat de o aplicaţie *client*
  2. un declanşator BD
  3. un declanşator aplicaţie

1. Declanşatori bază de date (*database trigger*)

- un declanşator bază de date este un bloc *PL/SQL* care se execută la apariţia unui eveniment:
  - modificarea unui tabel al bazei
  - modificarea unei vizualizări
  - anumite acţiuni sistem
  - sau chiar anumite acţiuni ale utilizatorului

2. Pachete (*package*)

: un pachet este un bloc *PL/SQL* care încapsulează într-o unitate logică în baza de date o mulţime de proceduri, funcţii, variabile, constante, tipuri, cursoare şi excepţii;

3. Declanşatori aplicaţie (*application trigger*) - un declanşator aplicaţie este un bloc *PL/SQL* care se execută în urma unui eveniment provocat de sistem.

- Aplicaţiile *Oracle* (*Oracle Forms* şi *Oracle Reports*) sunt dotate cu motoare *PL/SQL* care permit construirea acestor declanşatori.

Portabilitatea PL/SQL

Referitor la portabilitatea limbajului, pot fi remarcate două aspecte:

1. Aplicaţiile *PL/SQL* se pot executa pe orice platformă sau sistem de operare pe care poate fi executat *Oracle*.
2. *PL/SQL* permite transfer de cod între *server*-ul *Oracle* şi aplicaţii. Pot fi scrise pachete program portabile şi crea biblioteci ce pot fi utilizate în diferite situaţii, de diverse aplicaţii.

Controlul execuţiei unui bloc *PL/SQL*

PL/SQL este un limbaj cu structură de bloc, adică programele sunt compuse din blocuri care pot fi complet separate sau imbricate.

- Structura unui bloc poate fi obţinută combinând subprograme, pachete, blocuri imbricate.

- Blocurile pot fi folosite în utilitarele *Oracle*.

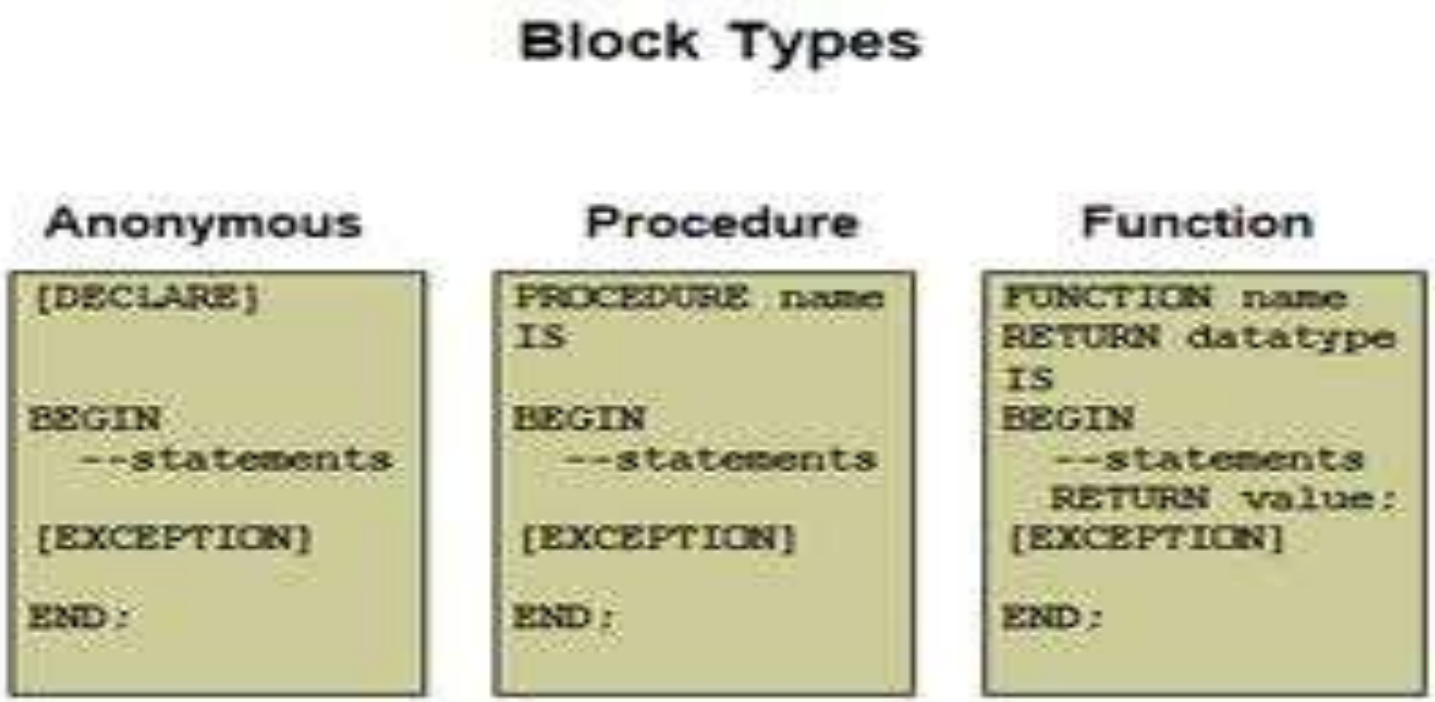
Pentru modularizarea unui program este necesară:

1. gruparea logică a instrucţiunilor în blocuri
2. imbricarea de subblocuri în blocuri mai mari
3. descompunerea unei probleme complexe într-o mulţime de module logice şi implementarea acestora cu ajutorul blocurilor
4. plasarea în biblioteci a codului *PL/SQL* reutilizabil, de unde poate fi folosit de aplicaţii
5. depunerea codului într-un *server Oracle*, de unde este accesibil oricărei aplicaţii care interacţionează cu baza de date *Oracle*

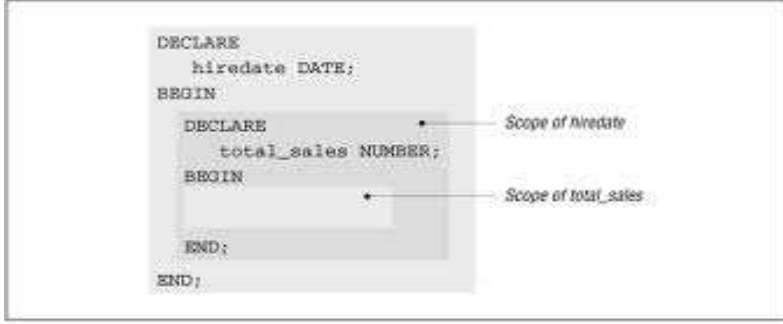
Un program *PL/SQL* poate cuprinde unul sau mai multe blocuri.

Un bloc poate fi:

1. anonim
2. neanonim



Blocuri anonime



1. Blocurile **anonime** sunt blocuri *PL/SQL* fără nume, care sunt construite dinamic şi sunt executate o singură dată.

- Acest tip de bloc nu are argumente şi nu returnează un rezultat.
- Ele sunt declarate într-un punct al aplicaţiei, unde vor fi executate (trimise motorului *PL/SQL*).
- În blocurile anonime pot fi declarate proceduri şi funcţii *PL/SQL*.

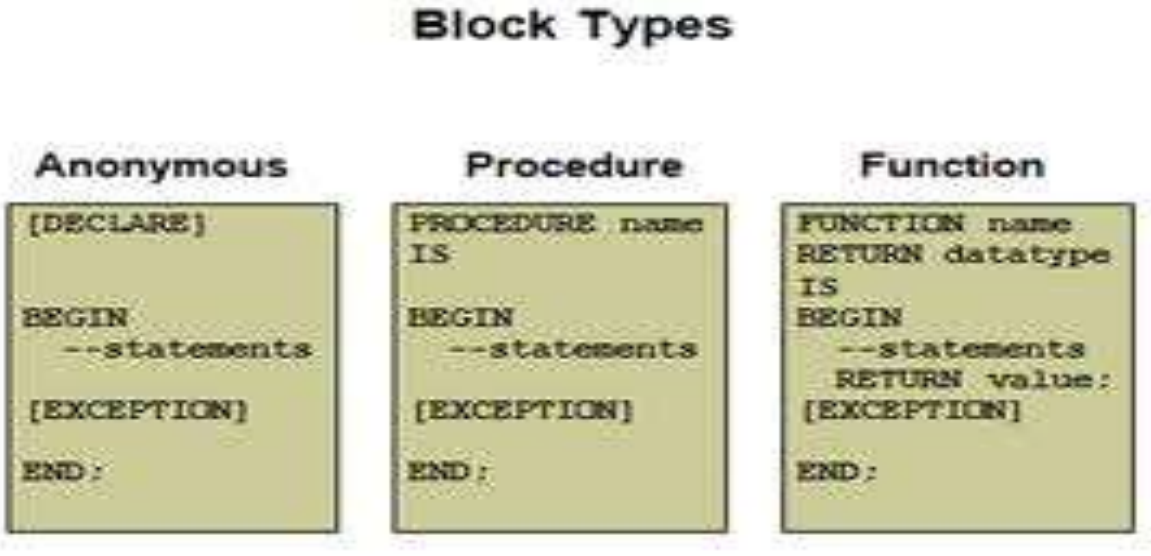
Blocuri neanonime

2. Blocurile **neanonime** sunt:

- fie blocuri cu nume (etichetate) construite static sau dinamic şi executate o singură dată,
- fie subprograme, pachete sau declanşatori.

1. Subprogramele sunt proceduri sau funcţii depuse în baza de date.

- Aceste blocuri sunt executate de mai multe ori şi, în general, nu mai sunt modificate după ce au fost construite.
- Procedurile şi funcţiile stocate sunt depuse pe *server*-ul *Oracle*, acceptă parametri şi pot fi apelate prin nume.

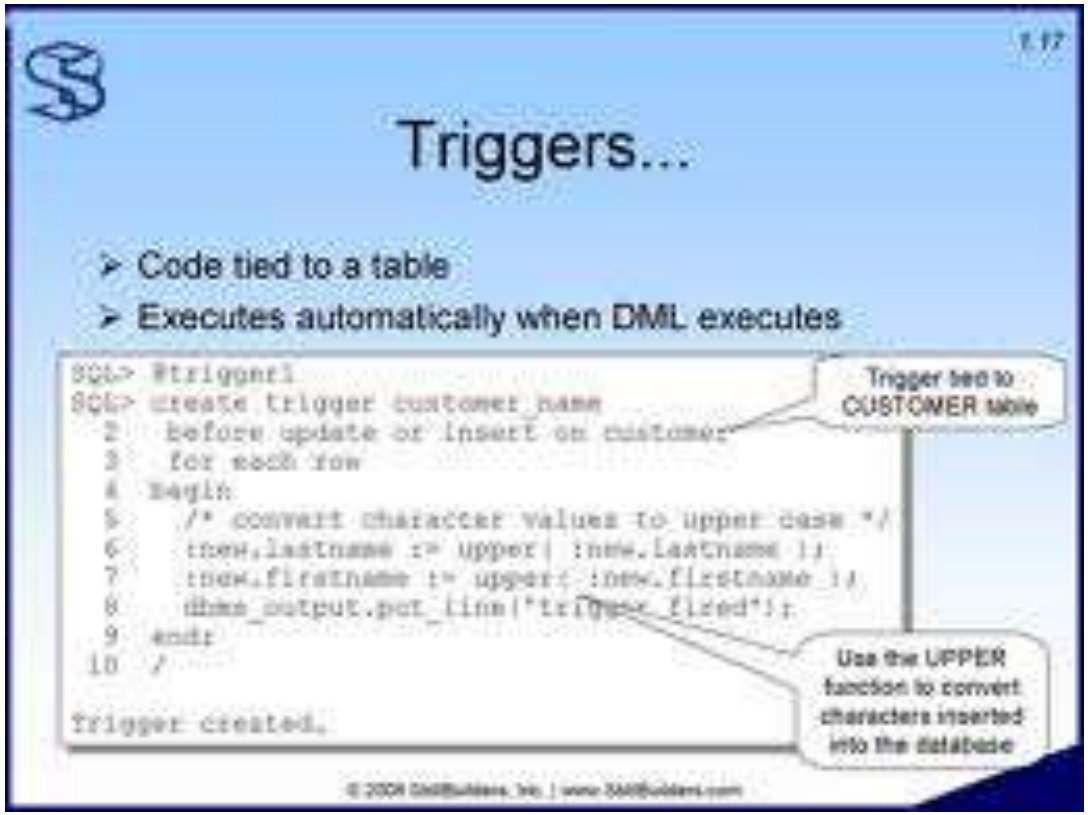


2. Pachetele (stocate sau aplicaţie)

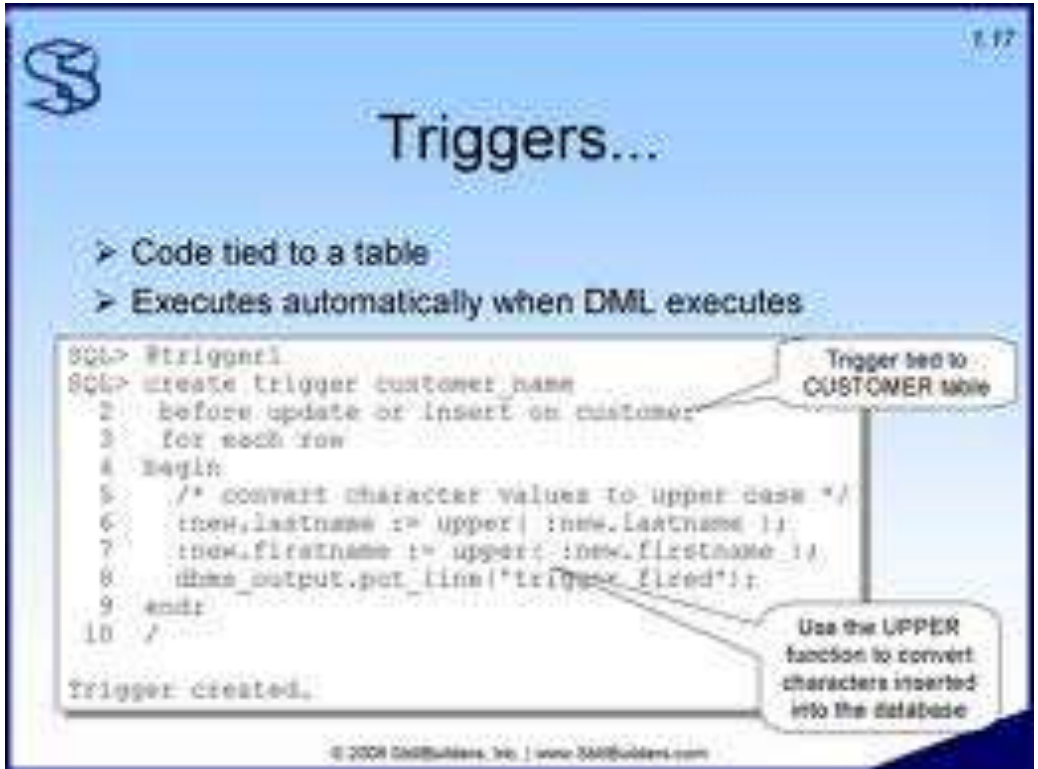
sunt blocuri neanonime care grupează:

1. proceduri
2. funcţii
3. cursoare
4. tipuri
5. constante
6. variabile - într-o unitate logică, în baza de date

3. Declanşatorii sunt blocuri *PL/SQL* neanonime depuse în baza de date, care pot fi asociaţi bazei, iar în acest caz sunt executaţi implicit ori de câte ori apare un anumit eveniment declanşator:



- De exemplu, instrucţiuni *INSERT*, *UPDATE* sau *DELETE* ce se execută asupra unui tabel al bazei de date
- sau pot fi asociaţi unei aplicaţii (de exemplu, declanşator *SQL*Forms*), ceea ce presupune că se execută automat, în funcţie de anumite condiţii sistem.



Structura unui bloc *PL/SQL*

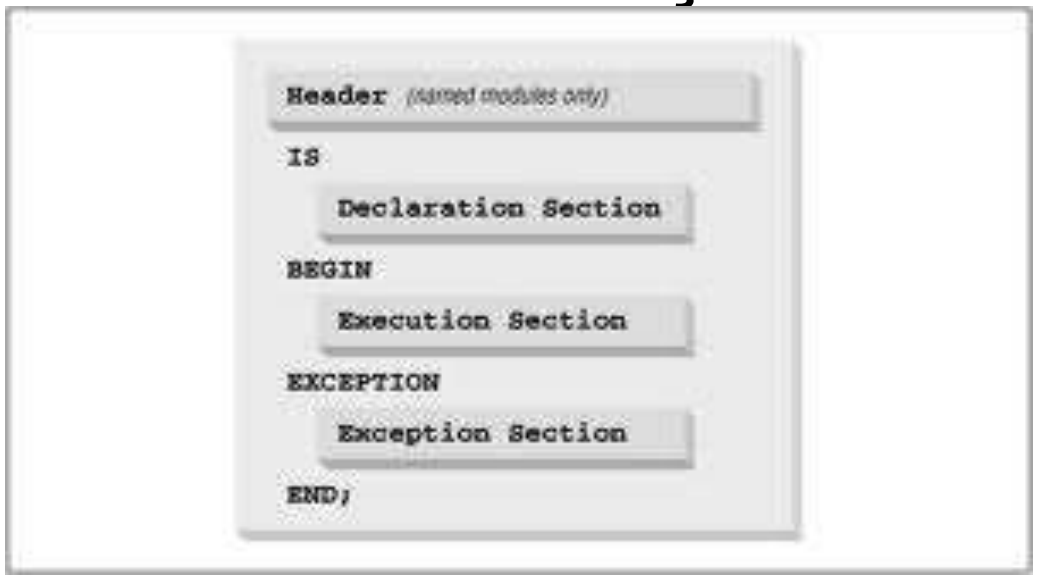
Un bloc *PL/SQL* este compus din trei secţiuni distincte:

1. Secţiunea declarativă (opţională) conţine declaraţii pentru toate variabilele, constantele, cursoarele şi erorile definite de utilizator la care se face referinţă în secţiunea executabilă sau chiar în cea declarativă.

- De asemenea, pot fi declarate subprograme locale care sunt vizibile doar în blocul respectiv.

2. Secţiunea executabilă conţine instrucţiuni neprocedurale *SQL* pentru prelucrarea datelor din baza de date şi instrucţiuni *PL/SQL* pentru prelucrarea datelor în cadrul blocului.

3. Secţiunea pentru tratarea erorilor (opţională) specifică acţiunile ce vor fi efectuate atunci când în execuţia blocului apar erori sau condiţii anormale.



Structura unui bloc PL/SQL

Blocul *PL/SQL* are următoarea structură generală:

[<<*nume_bloc*>>]
[***DECLARE***
instrucţiuni de declarare]
BEGIN
instrucţiuni executabile (*SQL sau PL/SQL*)
[***EXCEPTION***
tratarea erorilor]
END [*nume_bloc*];

Dacă blocul *PL/SQL* este executat fără erori, invariant va apărea mesajul:

PL/SQL procedure successfully completed

Compatibilitate cu *SQL*

Din punct de vedere al compatibilităţii dintre *PL/SQL* şi *SQL*, se remarcă următoarele reguli de bază:

- *PL/SQL* furnizează:
  - toate comenzile *LMD (limbajul de manipulare a datelor)* ale lui *SQL*
  - comanda *SELECT* cu clauza *INTO*
  - comenzile *LCD (limbajul de control al datelor)*
  - funcţiile
  - pseudocoloanele
  - şi operatorii *SQL*
- *PL/SQL* nu furnizează comenzile *LDD (limbajul de definire a datelor)*.

- Majoritatea funcţiilor *SQL* sunt disponibile în *PL/SQL*.
- Există însă funcţii specifice *PL/SQL*, cum sunt funcţiile *SQLCODE* şi *SQLERRM*.
- De asemenea, există funcţii *SQL* care nu sunt disponibile în instrucţiuni procedurale (*DECODE*, funcţiile grup), dar care sunt disponibile în instrucţiunile *SQL* dintr-un bloc *PL/SQL*.
- SQL nu poate folosi funcţii sau atribute specifice PL/SQL.

- Funcţiile de grup trebuie folosite cu atenţie, deoarece clauza ***GROUP BY*** nu are sens să apară în instrucţiunea ***SELECT … INTO***.

- *Oracle9i* introduce clauza ***OVER***, care permite ca funcţia grup căreia îi este asociată să fie considerată o funcţie analitică (poate returna mai multe linii pentru fiecare grup).

Următoarele funcţii ***SQL*** nu sunt permise în *PL/SQL*:

1. *WIDTH_BUCKET*
2. *BIN_TO_NUM*
3. *COMPOSE*
4. *DECOMPOSE*
5. *TO_LOB*
6. *DECODE*
7. *DUMP*
8. *EXISTSNODE*
9. *TREAT*
10. *NULLIF*
11. *SYS_CONNECT_BY_PATH*
12. *SYS_DBURIGEN*
13. *EXTRACT*

Întrebări?