

1. Relatii baza de date - spatii tabel - fisiere de date:

O baza de date contine unul sau mai multe spatii tabel. Fiecare spatiu tabel contine unul sau mai multe fisiere numite fisiere de date

Baze de date > Spatii Tabel > Fisiere de Date

2.Modul de executie al unui trigger:

Un trigger (declansator) este un bloc PL/SQL cu nume, stocat in baza de date (independent), care se executa automat ori de cate ori are loc evenimentul „declansator“ de care este asociat.

Evenimentul declansator poate consta din:

- o modificarea unui tabel sau a unei vizualizari;
- o actiuni sistem;
- o actiuni utilizator.

Un trigger poate fi asociat cu un eveniment care are loc asupra unui tabel, unei vizualizari, unei scheme sau unei baze de date.

Fata de subprogramele stocate trigger-ii:

- pot fi activati sau dezactivati;
- nu pot fi invocati explicit;
- nu pot contine comenzile LCD COMMIT, SAVEPOINT sau ROLLBACK.

3.Modul de executie al unui cursor:

Un cursor este un pointer catre o zona de memorie (care stocheaza informatii pana la momentul incheierii acesteia), despre procesarea unei comenzi SELECT sau LMD.

Categorii:

- implicite (construite si gestionate automat de PL/SQL de fiecare data cand este rulata o comanda SELECT sau LMD, se inchide automat, valorile ramanand disponibile pana cand este rulata o alta comanda SELECT sau LMD)

- explicite (contruite si gestionate de catre utilizator specificate printr-un nume, asociat cu o comanda SELECT care de obicei intoarce mai multe linii. Proces: se deschis cursoarele (OPEN), se incarca liniile cursorului in variabile cu comanda FETCH sau FOR LOOP, apoi se inchide cursorul (CLOSE))

Cursor parametrizat: se realizeaza asemanator cu apelul unei functii, specificand lista parametrilor actuali ai cursorului (asocierea dintre parametrii formali si cei actuali facandu-se prin pozitie si nu)

Cursor dinamic: cursor a carui comanda SQL este cunoscuta la momentul compilarii blocului

Tipul de date este o multime de valori predefinite sau definita de utilizator. Constantele, variabilele si parametrii trebuie sa aiba specificat un tip de date, determinand formatul de stocare, valorile si operatiile permise.

1. Scalare (stocheaza o singura valoare care nu poate avea componente interne)
2. Compuse(stocheaza mai multe valori care pot avea componente interne ce pot fi accesate individual)

Subtipul este un tip de date care reprezinta o submultime a unui alt tip de date denumit tip de baza, acesta permite aceleasi operatii ca si tipul de baza.

Tipul de date înregistrare se definește în 2 pași: se definește un tip RECORD și se declară variabilele de acest tip.

Colectii:

- Tablouri indexate (vectori asociativi) – similari cu tabelele de dispersie
- Tablouri imbricate – similari cu multimile (sets, multisets)
- Vectori cu dimensiune variabilă

Declarare: se definește un tip de colecție, se declară o variabilă de acel tip

O colecție are o singură dimensiune. Pentru a modela o colecție multidimensională se definește o colecție ale cărei elemente sunt direct sau indirect colecții (multilevel collections). O numărul nivelurilor de imbricare este limitat doar de capacitatea de stocare a sistemului. □ Colecții pe mai multe niveluri permise: o vectori de vectori; o vectori de tablouri imbricate; o tablouri imbricate de tablouri imbricate; o tablouri imbricate de vectori; o tablouri imbricate sau vectori de un tip definit de utilizator care are un atribut de tip tablou imbricat sau vector. □ Pot fi utilizate ca tipuri de date pentru definirea: o coloanelor unui tabel relational; o variabilelor PL/SQL; o atributelor unui obiect într-un tabel obiect.

Tablouri indexate = mulțimi de perechi cheie-valoare, cu cheie unică și utilizată pentru a putea localiza valoarea asociată. Atunci când este creat un tablou indexat care nu are încă elemente, acesta este vid. Nu este inițializat automat (atomic) null, ca în cazul celorlalte tipuri de colecții. Atunci când o valoare este asociată pentru prima oară unei chei, cheia este adăugată în tablou. Tablourile indexate folosesc spațiu de stocare temporar. Nu au constructori

Tablouri imbricate = este tip al unei coloane care stochează un număr nespecificat de linii, în nicio ordine particulară (stochează o mulțime de valori). O Atunci când tabloul imbricat din baza de date este preluat de o variabilă PL/SQL, sistemul atribuie liniilor indicii consecutivi (începând cu valoarea 1). Astfel, se permite accesarea liniilor folosind indicii, în mod asemănător cu vectorii. O Indicii și ordinea liniilor dintr-un tablou imbricat ar putea să nu rămână stabile în timp ce tabloul este stocat sau regăsit din baza de date. □ Numărul maxim de linii este dat de capacitatea maximă 2 GB. Inițial, tabloul este dens, dar în urma prelucrării este posibil să nu mai aibă indici consecutivi. Inițializarea se realizează cu ajutorul unui constructor

Vectori . Se utilizează în special pentru modelarea relațiilor one-to-many, atunci când numărul maxim de elemente copil este cunoscut și ordinea elementelor este importantă. □ Reprezintă structuri dense. O Fiecare element are un index care precizează poziția sa în vector (primul index are valoarea 1). O Indexul este utilizat pentru accesarea elementelor din vector. □ Vectorii: o față de tablourile imbricate au o dimensiune maximă specificată la declarare; o pot fi stocați în baza de date; o pot fi prelucrați direct în instrucțiuni SQL; o trebuie inițializați și extinși pentru a li se adăuga elemente.

Procedul bulk collect implică doar două comutări între cele două motoare: o motorul PL/SQL comunică motorului SQL să obțină mai multe linii odată și să le plaseze într-o colecție; o motorul SQL regăsește toate liniile și le încarcă în colecție, după care predă controlul motorului PL/SQL.

Procedul bulk bind permite transferul liniilor din colecție printr-o singură operație.

Subprograme: proceduri și funcții PL/SQL. Sunt blocuri cu nume, au structură similară blocurilor fără nume, secțiune declarativă sau de tratare a excepțiilor este opțională, cea executabilă este însă obligatorie. Tipuri:

- locale (definite în partea declarativă a unui bloc sau a unui alt subprogram)
- Independente (stocate în baze de date și considerate drept obiecte ale acestora)
- Impachetate (definite într-un pachet)

Avantaje: codul e ușor de întreținut, reutilizabil, asigură siguranța datelor și integritatea lor, îmbunătățesc performanța

Atunci cand este apelat un subprogram stocat, server-ul Oracle parcurge etapele:

Verifica daca utilizatorul are privilegiul de executie asupra subprogramului.

Verifica daca p-code-ul subprogramului este in shared pool. Daca este prezent va fi executat, altfel va fi incarcat de pe disc in database buffer cache.

Verifica daca starea subprogramului este VALID sau INVALID. Starea unui subprogram este INVALID, fie pentru ca au fost detectate erori la compilarea acestuia, fie pentru ca structura unui obiect s-a schimbat de cand subprogramul a fost executat ultima oara. Daca starea subprogramului este INVALID, atunci este recompilat automat. Daca nu a fost detectata nicio eroare, atunci va fi executata noua versiune a subprogramului.

Daca subprogramul apartine unui pachet atunci toate subprogramele pachetului sunt de asemenea incarcate in database buffer cache (daca acestea nu erau deja acolo). Daca pachetul este activat pentru prima oara intr-o sesiune, atunci server-ul va executa blocul de initializare al pachetului.

Procedura = bloc PL/SQL cu nume care poate accepta parametrii, utilizate pentru a realiza anumite actiuni. Procedurile stocate pot fi apelate:

- din corpul altei proceduri sau al unui declansator;
- interactiv, de utilizator folosind un utilitar Oracle (de exemplu, SQL*Plus);
- explicit, dintr-o aplicatie (de exemplu, Oracle Forms sau un precompilator).

Parametrii formali ai unei proceduri pot fi: parametri de intrare (IN), parametri de iesire (OUT), de intrare/iesire (IN OUT).

Functie = este un bloc PL/SQL cu nume care trebuie sa intoarca un rezultat (o singura valoare). Functiile independente sunt compilate si stocate in baza de date ca obiecte ale schemei. Clauza RETURN este utilizata in locul parametrilor de tip OUT. O functie trebuie sa contina clauza RETURN in antet si cel putin o comanda RETURN in partea executabila. In interiorul functiei trebuie sa apara RETURN expresie, unde expresie este valoarea rezultatului furnizat de functie. Intr-o functie pot sa apara mai multe comenzi RETURN, dar numai una din acestea va fi executata, deoarece dupa ce valoarea este intoarsa, procesarea blocului inceteaza.

Atunci cand un subprogram este compilat, in dictionarul datelor se vor inregistra informatii despre toate obiectele referite

Un pachet (package) permite incapsularea intr-o unitate logica a: constantelor si variabilelor; tipurilor si exceptiilor; cursoarelor; procedurilor si functiilor.

Pachetele sunt: unitati de program compilate; obiecte ale bazei de date care grupeaza tipuri, obiecte si subprograme PL/SQL avand o legatura logica intre ele.

Atunci cand este referentiat un pachet (atunci cand este apelata pentru prima data o constructie a pachetului), intregul pachet este incarcat in SGA (zona globala sistem) si este pregatit pentru executie. Plasarea pachetului in SGA reprezinta avantajul vitezei de executie, deoarece server-ul nu mai trebuie sa aduca informatia despre pachet de pe disc, aceasta fiind deja in memorie.

Definirea unui pachet se realizeaza in doua etape: crearea specificatiei pachetului si crearea corpului acestuia. Pachetul este instantiat cand este apelat prima data.

Modelul signature determina momentul la care obiectele bazei distante trebuie recompilate. Atunci cand este creata o unitate PL/SQL, o semnatura este depusa in dictionarul datelor, alaturi de p-code. Aceasta contine numele blocului PLSQL (PROCEDURE, FUNCTION, PACKAGE), tipurile

parametrilor, ordinea parametrilor, numarul acestora si modul de transmitere (IN, OUT, IN OUT)), tipul de date intors de functie. Daca signatura nu este modificata, atunci executia continua (fara a fi necesara recompilarea).

Dependentele la distanta pot fi manipulate folosind modelul timestamp (implicit) sau modelul signature.

Modelul timestamp realizeaza compararea momentelor ultimei modificari a celor doua obiecte analizate. Daca un obiect (referit) are momentul ultimei modificari mai recent decat cel al obiectului dependent, atunci obiectul dependent va fi recompilat.

Structura logica ale unei baze de date este formata din tabelele spatiu (tablespaces), schema de obiectelor bazei de date, blocurile de date, extensiile si segmentele.

Tabelele spatiu sunt unitatile logice de memorie in care este impartita o baza de date si pot fi tabele spatiu de sistem si tabele spatiu de utilizator. Din punct de vedere al accesibilitatii aceste pot fi on line si offline.

Fisierele de date sunt structurile de memorie specifice unui sistem de operare pe care rezida tabelele spatiu ale unei baze de date.

Schema este o colectie de obiecte, iar schema de obiecte este o structura logica care se refera direct la datele unei baze de date (tabele, vederi, secvente, proceduri memorate, sinonime, indecsi, clustere si link-uri de baza de date)

Blocurile de date, extensiile si segmentele sunt elemente de control eficient al spatiului de memorie externa pe disc aferent unei baze de date.

Blocul de date este unitatea de memorie cea mai mica manipule de SGBD Oracle, iar marimea acestuia masurata in bytes se defineste la momentul creerii bazei de date.

Instanta Oracle este combinatia logica dintre structurile de memorie interna (SGA – system global area, PGA – program global area) si procesele Oracle de baza activate la momentul pornirii unei baze de date.

- 1) SGA este o regiune partajabila de memorie care contine datele si informatiile necesare unei instante Oracle si contine:
 - Database buffer cache (contine blocurile cele mai recent utilizate pentru a reduce utilizarea discului)
 - Redo log buffer(contine datele despre blocurile modificate)
 - Shared Pool(pentru prelucrarea instructiunilor SQL)
 - Cursorii folositi pentru manipularea instructiunilor unui limbaj gazda folosind facilitate ORACLE CALL INTERFACE

- 2) PGA este o zona de memorie care contine datele si informatiile de control ale unui process server

Procesul= mecanism al sistemului de operare care poate executa o serie de instructiuni. Un server Oracle are 2 tipuri de procese: utilizator si oracle

SQL

- este limbajul utilizat pentru a accesa o baza de date relationala, inclusiv Oracle.
- poate fi utilizat de fiecare instrument Oracle, cand accesul la baza de date este necesar.

*PL/SQL

- este limbajul Procedural al lui Oracle pentru scrierea aplicatiilor si pentru manipularea datelor în afara bazei de date.
- poate include un subset al comenzilor SQL, când accesul la baza de date este cerut
- este accesibil în fiecare din produsele CDE. De asemenea în însuși serverul Oracle (dacă opțiunea procedurală este instalată).

Procesele server (Server Processes) sunt utilizate de Oracle pentru a prelucra cererile proceselor utilizator. Oracle poate fi configurat astfel încât să permită unul sau mai multe procese utilizator. Din acest punct de vedere avem servere dedicate care au un singur proces utilizator și servere multi prelucrare (multi-threaded server configuration). Pe anumite sisteme procesele utilizator și procesele server sunt separate, iar în altele sunt combinate într-unul singur. Dacă folosim sistemul multi prelucrare sau dacă procesele utilizator și procesele server se află pe mașini diferite atunci aceste procese trebuie să fie separate. Sistemul client/server separă procesele utilizator de către procesele server și le execută pe mașini diferite.

Procesele background (Background processes) sunt create pentru fiecare instanță Oracle pentru a executa asincron anumite funcții.