BAZE DE DATE

CURS 11

Limbajul de prelucrare a datelor

- *SQL* furnizează comenzi ce permit consultarea (*SELECT*) şi actualizarea (*INSERT*, *UPDATE*, *DELETE*, *MERGE*) conţinutului bazei de date.
- Aceste comenzi definesc limbajul de prelucrare a datelor (*LMD*).
- Comenzile limbajului *LMD* pot fi:
  - formulate direct;
  - utilizate în utilitare ale sistemului *Oracle*;
  - încapsulate într-un program *PL/SQL*;

Comanda *SELECT*

- Una dintre cele mai importante comenzi ale limbajului de prelucrare a datelor este *SELECT;*
- Cu ajutorul ei pot fi extrase submulţimi de valori atât pe verticală (coloane), cât şi pe orizontală (linii) din unul sau mai multe tabele;
- Sintaxa comenzii este simplă, apropiată de limbajul natural;

Comanda *SELECT*

SELECT [*ALL | DISTINCT*] {* | *listă de atribute selectate | expr AS alias*}

FROM { [*schema.*]{*tabel* [*PARTITION* (*partition_name*)] |

(*subquery*)} [*alias_tabel*] }

[***WHERE*** *condiţie*]

[***START WITH*** *condiţie*]

[***CONNECT BY*** *condiţie*]

[***GROUP BY*** *listă de expresii* [***HAVING*** *condiţie*]]

[***ORDER BY*** {*expresie | poziţie | c_alias*} [***ASC*** | ***DESC***]] [***FOR UPDATE***

[***OF*** [*schema.*]{*table | view*}.*coloană*] [***NOWAIT***]

Comanda *SELECT*

- Clauzele *START WITH* şi *CONNECT BY* sunt utile pentru a construi cereri ierarhizate.
  - *START WITH ->* înregistrarea rădăcină a arborelui
  - *CONNECT BY ->* relaţia dintre coloanele (*părinte* şi *copil*)
  - Prin folosirea operatorului *PRIOR* se poate face referinţă la înregistrarea *părinte*.

Comanda *SELECT*

Operatorii utilizaţi sunt:

- operatori aritmetici (unari sau binari),
- operatorul de concatenare (||),
- operatorii de comparare (=, !=, ^=, < >, >, >=, <, <=, *IN* (echivalent cu =ANY, adică egal cu cel puţin una din valorile listei), *NOT IN* (echivalent cu *!=ALL*, adică diferit de toate elementele listei), *ALL*, [*NOT*] *BETWEEN x AND y*, [*NOT*] *EXISTS*, [*NOT*] *LIKE*, *IS* [*NOT*] *NULL*,
- operatori logici (*NOT*, *AND*, *OR*).

Comanda *SELECT*

- Limbajul permite prezenţa unor instrucţiuni *SELECT* **imbricate** în oricare din clauzele *SELECT*, *WHERE*, *HAVING* sau *FROM.*
- Instrucţiunile *SELECT* care apar în clauzele respective se numesc **subcereri**.
- În cazul folosirii subcererilor, pot fi utilizaţi:
  - operatorii *ALL*, *ANY*, *IN* (=*ANY*), *EXIST*, *NOT IN* (!=*ANY*), care sunt specifici cererilor ce returnează mai multe linii (*multiple-row subquery*)
  - operatorii de comparare =, <, >, >=, <=, <>, specifici cererilor care returnează o singură linie (*single-row subquery*).

Comanda *SELECT*

Executarea subcererilor se poate face:

- fie **cu sincronizare** (corelat -> evaluarea subcererii face referinţă la o coloană a cererii principale şi cererea interioară se execută pentru fiecare linie a cererii principale care o conţine);
- fie **fără sincronizare** (-> se execută mai întâi cererea interioară, iar rezultatul ei este transmis cererii de nivel imediat superior).

Comanda *INSERT*

INSERT INTO *nume_tabel* | *nume_view* [(*col1*[, *col2*[,…]])]

VALUES (*expresia1*[, *expresia2*[,…]]) | *subcerere*;

- *expresia1*, *expresia2* – reprezintă expresii a căror evaluare este atribuită coloanelor precizate (se inserează o linie);
- *subcerere* – reprezintă o interogare (se inserează una sau mai multe linii).

Comanda *INSERT*

- Dacă lipseşte specificaţia coloanelor se consideră că sunt completate toate câmpurile tabelului sau vizualizării.
- Dacă se introduc date doar în anumite coloane, atunci aceste coloane trebuie specificate. În restul coloanelor se introduce automat *null* (daca nu exista *DEFAULT*).
- Specificarea cererii din comanda *INSERT* determină copierea unor date dintr-un tabel în altul pe atâtea linii câte au rezultat din cerere.
- Dacă se introduc numai anumite câmpuri într-o înregistrare, atunci printre acestea trebuie să se găsească câmpurile cheii primare.
- Pentru a putea executa comanda *INSERT* este necesar ca utilizatorul care execută această instrucţiune să aibă privilegiul de a insera înregistrări în tabel sau în vizualizare.

Comanda *DELETE*

DELETE

[***FROM***] *tablename / viewname* [***AS*** *alias*] [***WHERE*** *condiţie*] [*clauza_returning*]

- Comanda *DELETE* nu şterge **structura** tabelului.
- Pentru a se putea executa instrucţiunea *DELETE*, utilizatorul care o lansează în execuţie trebuie să aibă acest privilegiu.
- În clauza *WHERE* pot fi folosite şi subcereri.
- Comanda nu poate fi folosită pentru ştergerea valorilor unui câmp individual. Acest lucru se poate realiza cu ajutorul comenzii *UPDATE*.

Comanda *UPDATE*

UPDATE *tablename* | *viewname*

SET (*column1*[,*column2*[,…]]) = (*subquery*) | *column = expr* / (*query*)

[***WHERE*** *condition*]

- Pentru a se putea executa instrucţiunea *UPDATE*, utilizatorul care o lansează în execuţie trebuie să aibă acest privilegiu.
- Dacă nu este specificată clauza *WHERE* se vor modifica toate liniile.
- Cererea trebuie să furnizeze un număr de valori corespunzător numărului de coloane din paranteza care precede caracterul de egalitate.

LIMBAJUL PENTRU CONTROLUL DATELOR

Controlul unei baze de date cu ajutorul *SQL*-ului se refera la:

- asigurarea confidentialitatii si securitatii datelor;
- organizarea fizica a datelor;
- realizarea unor performante;
- reluarea unor actiuni in cazul unei defectiuni;
- garantarea coerentei datelor in cazul prelucrarii concurente;

LIMBAJUL PENTRU CONTROLUL DATELOR

Sistemul de gestiune trebuie:

- să pună la dispoziţia unui număr mare de utilizatori o mulţime **coerentă** de date;
- să garanteze **coerenţa** datelor în cazul manipulării simultane de către diferiţi utilizatori;

LIMBAJUL PENTRU CONTROLUL DATELOR

- Coerenţa este asigurată cu ajutorul conceptului de **tranzacţie**.
- Tranzacţia este unitatea logică de lucru constând din una sau mai multe instrucţiuni *SQL*

LIMBAJUL PENTRU CONTROLUL DATELOR

- Limbajul pentru controlul datelor (*LCD*) permite salvarea informaţiei, realizarea fizică a modificărilor în baza de date, rezolvarea unor probleme de concurenţă.
- Limbajul conţine următoarele instrucţiuni:
  - *COMMIT* - folosită pentru permanentizarea modificărilor executate asupra BD (modificările sunt înregistrate şi sunt vizibile tuturor utilizatorilor);
  - *ROLLBACK* - folosită pentru refacerea stării anterioare a BD (sunt anulate toate reactualizările efectuate de la începutul tranzacţiei);
  - *SAVEPOINT* - folosită în conjuncţie cu instrucţiunea *ROLLBACK*, pentru definirea unor puncte de salvare în fluxul programului;

LIMBAJUL PENTRU CONTROLUL DATELOR

O **tranzacţie** constă:

- dintr-o singură instrucţiune *LDD*;
- dintr-o singură instrucţiune *LCD*;
- din instrucţiuni *LMD* care fac schimbări consistente în date;

LIMBAJUL PENTRU CONTROLUL DATELOR

- Tranzacţia **începe**:
  - după o comandă *COMMIT;*
  - după o comandă *ROLLBACK;*
  - după conectarea iniţială la Oracle;
  - când este executată prima instrucţiune *SQL;*
- Tranzacţia se **termină**:
  - dacă sistemul nu mai functioneaza;
  - dacă utilizatorul se deconectează;
  - dacă se dau comenzile *COMMIT* sau *ROLLBACK* ;
  - dacă se execută o comandă *LDD;*
- După ce se termină o tranzacţie, prima instrucţiune *SQL* executabilă va genera automat începutul unei noi tranzacţii.

LIMBAJUL PENTRU CONTROLUL DATELOR

- Din momentul în care s-a executat instrucţiunea ***COMMIT***, BD s-a modificat (permanent) în conformitate cu instrucţiunile *SQL* executate în cadrul tranzacţiei care tocmai s-a terminat. Din acest punct începe o nouă tranzacţie.
- Dacă se foloseşte utilitarul *SQL*Plus*, există posibilitatea ca după fiecare comandă *LMD* să aibă loc o permanentizare automată a datelor (un *COMMIT* implicit). Acest lucru se poate realiza folosind comanda:

  ***SET AUTO*[*COMMIT*] {*ON | OFF*}**

- Comanda ***ROLLBACK*** permite restaurarea unei stări anterioare a BD.

ROLLBACK [*TO* [*SAVEPOINT*] *savepoint*];

LIMBAJUL PENTRU CONTROLUL DATELOR

- Dacă nu se specifică nici un *savepoint*, toate modificările făcute în tranzacţia curentă sunt anulate;
- Dacă se specifică un anumit *savepoint*, atunci doar modificările de la acel *savepoint* până în momentul respectiv sunt anulate;
- Executarea unei instrucţiuni *ROLLBACK* presupune terminarea tranzacţiei curente şi începerea unei noi tranzacţii;

LIMBAJUL PENTRU CONTROLUL DATELOR

- Punctele de salvare pot fi considerate ca nişte etichete care referă o submulţime a schimbărilor dintr-o tranzacţie, marcând efectiv un punct de salvare pentru tranzacţia curentă. Punctele de salvare **NU sunt obiecte ale schemei**. Prin urmare, nu sunt referite in DD.
- *Server*-ul *Oracle* implementează **un punct de salvare implicit** pe care îl mută automat după ultima comandă *LMD* executată.
- Dacă este creat un punct de salvare având acelaşi nume cu unul creat anterior, cel definit anterior este şters automat.

***SAVEPOINT savepoint*;**

LIMBAJUL PENTRU CONTROLUL DATELOR

Starea datelor înainte de *COMMIT* sau *ROLLBACK* este următoarea:

- starea anterioară a datelor poate fi recuperată;
- utilizatorul curent poate vizualiza rezultatele operaţiilor *LMD* prin interogări asupra tabelelor;
- alţi utilizatori nu pot vizualiza rezultatele comenzilor *LMD* făcute de utilizatorul curent (***read consistency***);
- înregistrările (liniile) afectate sunt blocate şi, prin urmare, alţi utilizatori nu pot face schimbări în datele acestor înregistrări.

LIMBAJUL PENTRU CONTROLUL DATELOR

Execuţia unei comenzi *COMMIT* implică anumite modificări.

- Toate schimbările (*INSERT*, *DELETE*, *UPDATE*) din baza de date făcute după anterioara comandă *COMMIT* sau *ROLLBACK* sunt **definitive**. Comanda se referă numai la schimbările făcute de utilizatorul care dă comanda *COMMIT*.
- Toate punctele de salvare vor fi şterse.
- Starea anterioară a datelor este pierdută definitiv.
- Toţi utilizatorii pot vizualiza rezultatele.
- Blocările asupra liniilor afectate sunt eliberate; liniile pot fi folosite de alţi utilizatori pentru a face schimbări în date.

LIMBAJUL PENTRU CONTROLUL DATELOR

Execuţia unei comenzi *ROLLBACK* implică anumite modificări.

- **Anulează** tranzacţia în curs şi toate modificările de date făcute după ultima comandă *COMMIT.*
- Sunt **eliberate** blocările liniilor implicate.