

MS SQL SERVER

Comenzile limbajului DDL se utilizează pentru crearea și modificarea structurii bazei de date:

CREATE TABLE – adăugarea unei noi tabel în baza de date

ALTER TABLE – modificarea structurii

DROP TABLE – șterge tabel

CREATE VIEW – crearea unei tabele virtuale

DROP VIEW – ștergerea tabelei virtuale

Obs. Sintaxele comenzilor următoare respectă standardul SQL, dar nu sunt complete și nu pot fi folosite în totalitate în MS SQL Server

Comanda CREATE TABLE

Comanda realizează fie definirea unui tabel urmând ca introducerea de date să se efectueze ulterior, fie definirea și umplerea cu date a unui tabel chiar în momentul creării lui prin specificarea unei cereri.

CREATE TABLE <nume_tabel>

```
(  
<nume_camp> <tip_de_data> [[CONSTRAINT <nume_restricție> [DEFAULT] <valoare_implicită>] [PRIMARY KEY  
<nume_camp_cheie_primară>] [UNIQUE <nume_camp>] [CHECK <condiție>] [REFERENCES  
<nume_tabela_referențială>(<coloana_referențială>)]]  
)
```

Comanda ALTER TABLE

Comanda realizează modificarea structurii unui tabel, creat anterior prin intermediul comenzii CREATE TABLE.

ALTER TABLE <nume_tabela><acțiune_modificare_tabela>

Unde <acțiune_modificare_tabela> : : =

ADD [COLUMN] <definiție_coloana>

|ALTER [COLUMN] <nume_coloana><acțiune_modificare_coloana>

|DROP [COLUMN] <nume_coloana><comportament_la_stergere>

|ADD <definiție_restricție_tabela>

|DROP CONSTRAINT<nume_restricție><comportament_la_stergere>

<comportament_la_stergere> : : = RESTRICT|CASCADE

Comanda DROP TABLE

Comanda realizează ștergerea unui tabel creat anterior prin intermediul comenzii CREATE TABLE.

DROP <nume_tabela><comportament_la_stergere>

Unde

<comportament_la_stergere> : : = RESTRICT|CASCADE

Comenzile limbajului DML se utilizează pentru introducerea, regăsirea, modificarea și actualizarea datelor din baza de date:

INSERT

SELECT

UPDATE

DELETE

Comanda INSERT

INSERT INTO <nume_tabela> [(nume_camp1, nume_camp2,...)]

VALUES (valoare_camp1, valoare_camp2,...)

Comanda SELECT

SELECT[ALL|DISTINCT`{*|[utilizator]{nume_tabela|viziune}*|expresie_sinonim_coloana}}],[{[utilizator.]
{nume_tabela|viziune}.expresie}{sinonim_coloana}]]

FROM[utilizator.]{nume_tabela|viziune}[[@legatura][sinonim_tabela][,...urmeaza_optiunile

[WHERE condiție]

[CONNECT BY condiție[STARTWITH condiție]]

[GROUP BY expr, exp...]

[HAVING condiție]

[{UNION|INTERSECT|MINUS}subcerere] ORDER BY {exp|numar}poziție}[ASC|DESC]

[FOR UPDATE OF nume_coloana[,nume_coloana]...[NOWAIT]];

OBS.

1. Caracterul ";" se pune obligatoriu la terminarea comenzii
2. Clauzele enumerate și sintaxa comenzii trebuie utilizate în ordinea specificată excepție făcând clauzele CONNECT BY, STARTWITH care pot fi specificate în orice ordine.
3. ALL afișează toate înregistrările tabelului. DISTINCT elimină înregistrările duplicate.
4. În SELECT în locul numelui de tabel se pot folosi sinonimele acestora stabilite în clauza FROM.
5. Orice sinonim specificat este folosit în scopul utilizării expresiei precedente din tabelul afișat și are un rol bine definit în referirea coloanei unui raport ce conține expresia specificată. Dacă sinonimul conține spații libere sau caractere speciale el trebuie inclus între apostrofuri.
6. Clauza FROM poate fi folosită în orice tabelă care conține un singur rând pentru a selecta o expresie care nu are nimic în comun cu tabelul.
7. WHERE specifică o condiție definită cu ajutorul operatorilor de relație sau logici.
8. CONNECT BY indică faptul că rândurile formează o structură arborescentă. Prin această clauză sunt definite relațiile necesare pentru a conecta rândurile tabelului într-un arbore cu o ierarhie de tip părinte-copil.
9. Clauzele GROUP BY și HAVING permit prelucrarea datelor pe submulțimi, adică pe grupuri de rânduri care au aceeași valoare în una sau mai multe coloane. Folosirea comenzii SELECT cu aceste clauze permite utilizarea funcțiilor agregat (SUM, COUNT, AVG)
10. Pentru a combina rezultatele comenzii SELECT într-un singur rezultat se folosesc clauzele UNION, INTERSECT, MINUS. Pentru a putea folosi aceste clauze este necesar ca numărul și tipul coloanelor selectate de fiecare comandă SELECT să fie același.
11. Clauza ORDER BY specifică ordinea în care trebuie afișate rândurile distincte ale unei tabeli (sortare).
12. FOR UPDATE OF determină blocarea rândurilor selectate ale tabelului, astfel încât acestea nu vor mai putea fi utilizate de alți utilizatori până la deblocarea lor cu una dintre comenzile COMMIT sau ROLLBACK. Comanda SELECT FOR UPDATE OF trebuie urmată de una sau mai multe comenzi UPDATE ...WHERE. Dacă se folosește clauza NOWAIT selecția este considerată terminată în loc să se aștepte chiar dacă rândurile selectate de FOR UPDATE nu pot fi blocate din cauză că un alt utilizator lucrează la ele.

Comanda UPDATE

Comanda se utilizează pentru modificarea unui sau mai multor atribute pe una sau mai multe linii dintr-o tabelă

UPDATE tabela

SET atribut1 = expresie1 [, atribut2 = expresie2 ...]

WHERE predicat

Comanda DELETE

Comanda se utilizează pentru ștergerea unei sau mai multor linii dintr-o tabelă

DELETE

FROM <nume_tabela>

WHERE predicat

1. Să se creeze baza de date *Exemplu_<numar_subgrupa>* cu un singur tabel *Ex1_<nume_student>* care are câmpul, de tip întreg, indexat, coloana *a*
 - a. să se adauge *coloana_b*, de tip text, de 20 de caractere, ce poate avea valori nule;
 - b. să se elimine *coloana_b* (1 p)
 - c. să se adauge *coloana_b*, indexată, de tip text, cu 20 de caractere, ce poate avea valori nule;
 - d. să se interzică ca în *coloana_a* să fie valori mai mici de 1;
 - e. să se elimine *coloana_b* (1 p);
 - f. să se adauge *coloana_b*, de tip întreg, cheie primară;
 - g. să se adauge *coloana_c*, de tip întreg, ce poate primi valori nule și care este asociată unei alte coloane din același tabel;
 - h. să se adauge *coloana_d*, de tip numeric, cu trei cifre la partea întreagă și trei zecimale, nenulă cu valoare implicită 0.081.