

Limbajul SQL a fost pentru prima dată standardizat de către Organizația Internațională pentru Standardizare (ISO) în anul 1987. De atunci limbajul a suferit modificări/îmbunătățiri în mai multe rânduri, ultima versiune stabilă fiind SQL:2008 (denumită ISO/IEC 9075:2008).

Deși standardizat la nivel internațional, fiecare dezvoltator de baze de date a modificat și a extins versiunea standard pentru a exploata avantajele oferite de implementarea particulară a produsului propriu. Datorită dimensiunii mari și a complexității standardului cei mai mulți dezvoltatori nu l-au implementat în totalitate. Vice-versa, standardul nu specifică comportamentul bazei de date în anumite situații, deciziile rămânând la latitudinea dezvoltatorilor. Se vorbește astfel de dialecte SQL. Rezultatul este apariția de incompatibilități între SGBD-uri relaționale diferite. Unele incompatibilități sunt intenționat păstrate de dezvoltatori cu scopul de fidelizare a clienților.

Declarația de conformitate Oracle cu standardul ISO SQL:2008 poate fi studiată la adresa http://download.oracle.com/docs/cd/E14072_01/server.112/e10592/ap_standard_sql003.htm

În continuare sunt listate în forma Backus-Naur cele mai uzuale comenzi SQL. Convențiile de sintaxă în acest format sunt următoarele:

- | reprezintă conjuncția
- [...] reprezintă elemente opționale
- {...|...|...} reprezintă exact un element din listă

Comenzi DDL (de definire a datelor)

Crearea tabelelor

```
CREATE [{GLOBAL | LOCAL} TEMPORARY] TABLE  
    nume_tabel  
    (nume_coloană {data_type | domain_name} [constrânger]  
    [, column_name {data_type | domain_name} [constrânger]]  
    [...]  
    )  
    [ON COMMIT { DELETE | PRESERVE } ROWS ]
```

Opțiunea TEMPORARY specifică crearea unui tabel temporar, care își păstrează conținutul doar pe parcursul unei sesiuni (opțiunea ON DELETE PRESERVE ROWS) sau doar pe durata unei tranzacții (până la apariția comnzii COMMIT). În **Oracle există doar varianta GLOBAL** a tabelelor temporare: definiția tabelului temporar este vizibilă din toate sesiunile de lucru însă accesul la date este privat (datele inserate într-o sesiune pot fi accesate doar din sesiunea respectivă).

Constrângerile specificate la creare în dreptul fiecărei coloane:

- **NOT NULL** – coloana nu permite valoarea NULL

- **UNIQUE** - coloana permite doar valori unice
- **DEFAULT *valoare*** - *valoare* este inserată implicit
- **CHECK (*conditie*)** - permite doar valori ce satisfac anumite condiții
- **PRIMARY KEY** - coloana constituie cheie primară pentru tabel
- Constrangerile de tip cheie straina se declară la creare astfel:
REFERENCES *nume_tabel_parinte*(*lista_chei_candidat*)
[ON DELETE {SET NULL | CASCADE | NO ACTION}]
[ON UPDATE {SET NULL | CASCADE | NO ACTION}]

NO ACTION (utilizat în Oracle) este înlocuit în unele SGBD-uri cu RESTRICT

Alternativ constrangerile de tip UNIQUE, declararea cheii primare si a cheilor straine pot fi specificate la sfarsitul comenzii:

```
CREATE TABLE nume_tabel
    {(nume_coloana_tip [NOT NULL] [UNIQUE] [DEFAULT valoare_implicita]
      [CHECK (conditie)]
    [...]}

    [PRIMARY KEY (lista_coloane),]

    {[UNIQUE (lista_coloane),] [...],}

    {[FOREIGN KEY (lista_coloane_FK)
    REFERENCES nume_tabel_parinte [(lista_chei_candidat)],
    [ ON DELETE {SET NULL | CASCADE | NO ACTION}]
    [ ON UPDATE {SET NULL | CASCADE | NO ACTION}]

    {[CHECK (conditie)] [...] }}
```

Stergerea unui tabel

DROP TABLE *nume_tabel* [RESTRICT] CASCADE]

CASCADE permite ștergerea unui tabel care e referențiat de un VIEW sau de chei străine și șterge view-ul și restricția de tip cheie străină

În Oracle sintaxa pentru ștergerea tabelelor este:

DROP TABLE *nume_tabel* [CASCADE CONSTRAINTS] [PURGE]

În Oracle tabelele șterse sunt reciclate dacă nu este specificată opțiunea PURGE, ele putând fii recuperate cu comanda:

```
FLASHBACK TABLE nume_tabel TO  
    { { SCN | TIMESTAMP } expr | { ENABLE | DISABLE } TRIGGERS | |  
    BEFORE DROP | RENAME TO table } ;
```

Comenzi DML (de manipulare a datelor)

Inserarea înregistrărilor

```
INSERT INTO nume_tabel [ (lista_coloane) ]  
    VALUES (lista_valori)
```

Actualizarea valorilor

```
UPDATE nume_tabel  
    SET nume_coloana1 = valoare1  
    [, nume_coloana2 = valoare2...]  
    [WHERE conditie]
```

Ștergerea înregistrărilor

```
DELETE FROM nume_tabel  
    [WHERE conditie]
```

Interogarea bazei de date

```
SELECT [DISTINCT | ALL] { * | [expresie_coloana [AS nume_nou]] [,...] }  
    FROM nume_tabel [alias] [, ...]  
    [WHERE conditie]  
    [GROUP BY lista_coloane] [HAVING conditie]  
    [ORDER BY lista_coloane]
```