

Constrângeri de integritate în SQL

1. Impunerea constrângerilor de integritate în SQL

Serverul Oracle utilizează *constrângerile de integritate* pentru prevenirea introducerii unor date invalide în tabele. Constrângerile pot fi utilizate pentru:

- forțarea unor reguli la nivel de tabel atunci când o linie este inserată, actualizată sau ștearsă; constrângerea trebuie satisfăcută pentru ca operația să se termine cu succes.
- prevenirea ștergerii unui tabel dacă există dependențe din alte tabele.
- oferirea unor reguli pentru alte medii Oracle, cum ar fi Developer/2000.

Există cinci tipuri de constrângeri de integritate ce pot fi impuse:

- NOT NULL - specifică faptul că o coloană nu poate conține o valoare NULL;
- UNIQUE - specifică o coloană (o combinație de coloane) a cărei valoare (combinație de valori) trebuie să fie unică pentru toate înregistrările din tabel;
- PRIMARY KEY - identifică unic fiecare înregistrare;
- FOREIGN KEY - stabilește o legătură pe baza unei chei externe între o coloană din tabel și o coloană din tabelul referit;
- CHECK - specifică o condiție ce trebuie satisfăcută de fiecare înregistrare.

Toate constrângerile definite de un utilizator sunt păstrate în dicționarul de date. Pentru vizualizarea constrângerilor impuse unui tabel putem interoga dicționarul de date, mai exact vederea USER_CONSTRAINTS.

2. Definirea constrângerilor

Sintaxa instrucțiunii CREATE ce impune constrângeri asupra unei coloane sau asupra unui tabel este:

```
CREATE TABLE [schema.] tabel
( coloana tip_date [DEFAULT expr]
  [constrangere_coloana],
  .....
  [constrangere_tabel] );
```

unde:

| | |
|-----------------------------|--|
| <i>schema</i> | numele utilizatorului proprietar; |
| <i>tabel</i> | numele tabelului; |
| DEFAULT <i>expr</i> | specifică o valoare implicită, dacă este omisă o valoare pentru atribut la inserare; |
| <i>coloana</i> | numele coloanei; |
| <i>tip_date</i> | tipul de date și dimensiunea; |
| <i>constrangere_coloana</i> | constrângere de integritate ca parte a definiției de coloană; |
| <i>constrangere_tabel</i> | constrângere de integritate ca parte a definiției tabelului. |

```
CREATE TABLE angajati(
    empno      NUMBER(4),
    ename      VARCHAR(2),
    deptno     NUMBER(7,2) NOT NULL,
    CONSTRAINT angajati_empno_pk PRIMARY KEY(EMPNO));
```

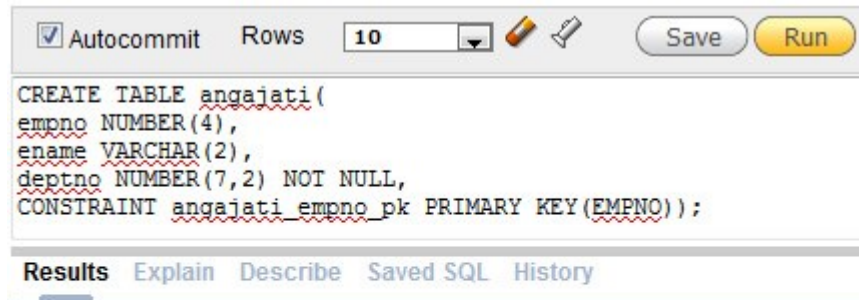


Table created.

3.62 seconds

În exemplul de mai sus, denumirea constrângerii poate să lipsească (CONSTRAINT angajati_empno_pk).

Constrângerile sunt de obicei definite în același timp cu tabelul. Ele pot fi adăugate și după crearea tabelului cu ajutorul instrucțiunii ALTER TABLE. Constrângerile pot fi definite la unul din următoarele două nivele:

| Nivel constrângere | Descriere |
|--------------------|---|
| Coloană | face referire la o singură coloană și poate defini orice tip de constrângere |
| Tabel | face referire la una sau mai multe coloane și este definită separat de definițiile coloanelor din tabel; poate defini orice tip de constrângere, cu excepția constrângerii NOT NULL |

Constrângere la nivel de coloană

```
coloana tip_constrangere,
```

Constrângere la nivel de tabel

```
coloana , ...
tip_constrangere (coloana , ...), ...
```

unde:

tip_constrangere este tipul constrângerii.

În definiția constrângerii utilizatorul poate furniza un nume sugestiv pentru aceasta (numele constrângerii este opțional). În acest caz constrângerea trebuie definită la nivel de tabel. Dacă unei constrângerii nu i se furnizează un nume, serverul Oracle generează un nume după formatul SYS_Cn, unde n este un întreg care determină unicitatea numelui.

2.1 Constrângerea NOT NULL

Constrângerea NOT NULL indică faptul că valoarea NULL nu este permisă în coloana afectată de constrângere. Coloanele fără constrângeri NOT NULL pot conține, implicit, valori NULL.

EMP

| EMPNO | ENAME | JOB | ... | COMM | DEPTNO |
|-------|-------|-----------|-----|------|--------|
| 7839 | KING | PRESIDENT | | | 10 |
| 7698 | BLAKE | MANAGER | | | 30 |
| 7782 | CLARK | MANAGER | | | 10 |
| 7566 | JONES | MANAGER | | | 20 |
| ... | | | | | |

Constrângere NOT NULL (nici o înregistrare nu poate conține o valoare NULL în această coloană)

Absența constrângerii NOT NULL (orice înregistrare poate conține valoarea NULL pentru această coloană)

Constrângere NOT NULL

Constrângerea NOT NULL poate fi specificată numai la nivel de coloană.

```
CREATE TABLE angajati (
    empno      NUMBER(4),
    ename      VARCHAR2(10) NOT NULL,
    job        VARCHAR2(9),
    mgr        NUMBER(4),
    hiredate   DATE,
    sal        NUMBER(7,2),
    comm       NUMBER(7,2),
    deptno     NUMBER(7,2) NOT NULL);
```

În exemplul de mai sus se aplică constrângerea NOT NULL coloanelor ename și deptno din tabelul angajati. Deoarece aceste constrângeri nu au nume, server-ul Oracle va crea automat nume pentru ele.

```

Autocommit Rows 10 Save Run
CREATE TABLE angajati(
empno NUMBER(4),
ename VARCHAR2(10) NOT NULL,
job VARCHAR2(9),
mgr NUMBER(4),
hiredate DATE,
sal NUMBER(7,2),
comm NUMBER(7,2),
deptno NUMBER(7,2) NOT NULL);
Results Explain Describe Saved SQL History

```

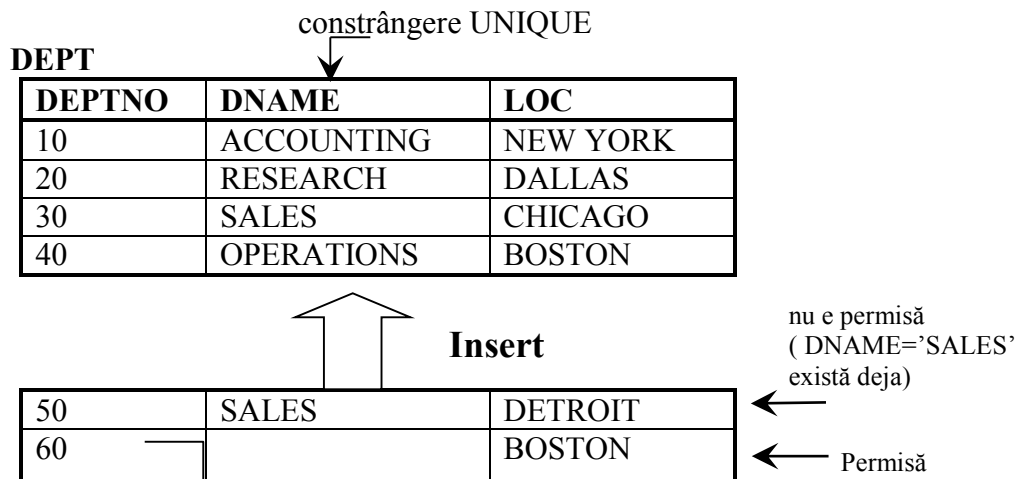
Table created.

0.04 seconds

Numele unei constrângeri poate fi indicat în definiția constrângerii, doar dacă aceasta apare la nivelul tabelului (după definirea tuturor coloanelor):

... deptno NUMBER(7,2) NOT NULL ...

2.2 Constrângerea UNIQUE



O constrângere de integritate de tip cheie unică cere ca fiecare valoare din coloană sau din mulțimea de coloane să fie unică – două înregistrări ale tabelului nu pot avea valori duplicate corespunzătoare cheii unice. Coloana (mulțimea de coloane) inclusă în definiția cheii unice se numește *cheie unică*. Dacă o cheie unică conține mai multe coloane se numește *cheie unică compusă*.

O constrângere de tip cheie unică permite introducerea valorilor NULL dacă nu a fost definită o constrângere NOT NULL pentru acea coloană. De fapt, orice număr de înregistrări pot include valori NULL în coloane fără constrângeri NOT NULL, deoarece valorile NULL nu sunt egale cu nimic. O valoare NULL într-o coloană (sau în toate coloanele unei chei unice compuse) va satisface întotdeauna o constrângere de cheie unică.

Notă: Din cauza mecanismului de căutare a constrângerilor UNIQUE în una sau mai multe coloane, nu este permisă existența unor valori identice în coloanele NOT NULL a unei constrângeri de cheie unică compusă.

Constrângerea de tip cheie unică (UNIQUE) definită la nivel de tabel sau de coloană

Constrângerile de tip cheie unică pot fi definite la nivel de coloană sau de tabel. O cheie unică compusă este creată utilizând definiția la nivel de tabel (după definirea tuturor coloanelor).

```
CREATE TABLE departament(  
    deptno    NUMBER(2) ,  
    dname     VARCHAR2(14) ,  
    loc       VARCHAR2(13) ,  
    UNIQUE (dname) ) ;
```

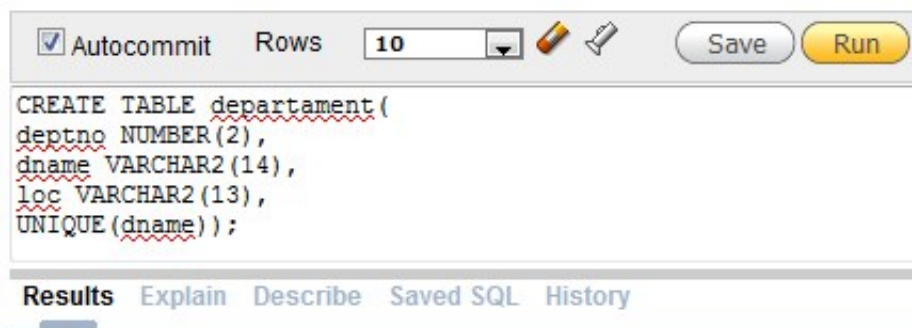


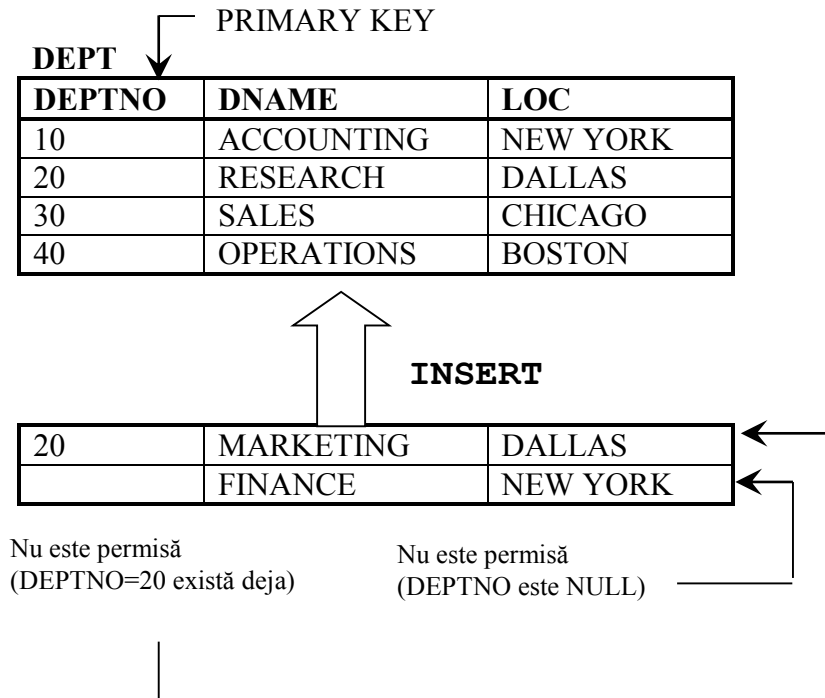
Table created.

0.33 seconds

În exemplul anterior se aplică constrângerea de cheie unică coloanei dname din tabelul departament.

Notă: Serverul Oracle forțează implicit constrângerea de cheie unică în momentul creării unui index unic după cheia unică.

2.3 Constrângerea PRIMARY KEY



Constrângerea de cheie primară creează o cheie primară pentru tabel. Doar o singură cheie primară poate fi creată pentru un anumit tabel. O cheie primară este o coloană sau mulțime de coloane ce identifică unic fiecare înregistrare din tabel. Această constrângere forțează unicitatea coloanei sau a mulțimii de coloane și asigură că nici o coloană din cheia primară nu poate conține valoarea NULL.

Constrângerea cheii primare definită la nivel de coloană sau de tabel

Constrângerea de cheie primară (PRIMARY KEY) poate fi definită la nivel de tabel (dacă implică mai multe coloane sau dacă i se atribuie un nume) sau de coloană (implică o singură coloană și este nedenumită). O cheie primară compusă este creată utilizând definiția la nivel de tabel.

```
CREATE TABLE departament(  
    deptno    NUMBER(2),  
    dname     VARCHAR2(14),  
    loc       VARCHAR2(13),  
    UNIQUE (dname),  
    PRIMARY KEY(deptno));
```

Exemplul de mai sus definește o cheie primară formată din coloana deptno din tabelul departamentt.

```

Autocommit Rows 10 Save Run
CREATE TABLE departament(
deptno NUMBER(2),
dname VARCHAR2(14),
loc VARCHAR2(13),
UNIQUE(dname),
PRIMARY KEY(deptno));
Results Explain Describe Saved SQL History

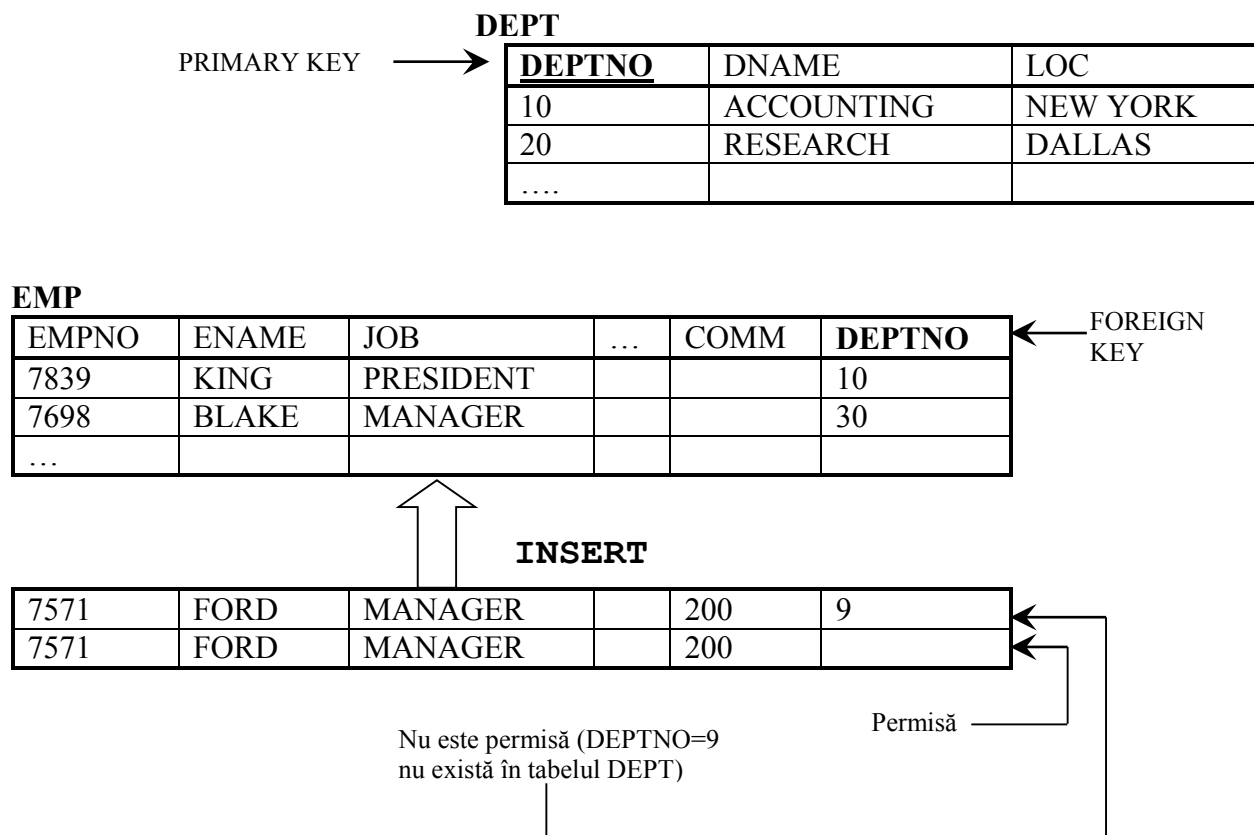
```

Table created.

0.01 seconds

Notă: Un index unic este creat automat pentru o coloană cheie primară.

2.4 Constrângerea FOREIGN KEY (sau de referință)



Cheia externă sau constrângerea de integritate referențială desemnează o coloană sau o combinație de coloane pe post de cheie externă și stabilește o relație cu o cheie primară sau o cheie unică din același tabel sau din alt tabel. În exemplul de mai sus, coloana deptno a fost definită

cheie externă în tabelul emp (denumit dependent sau tabel copil); ea referă coloana DEPTNO din tabelul dept (denumit referit sau tabel părinte).

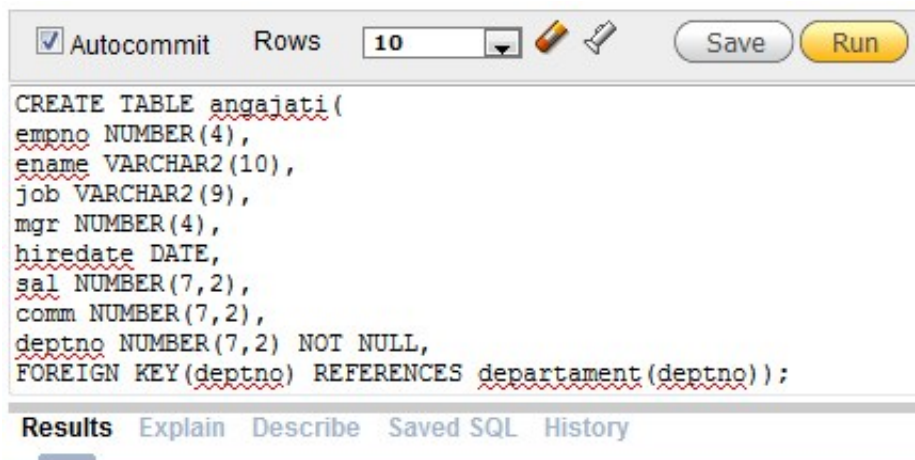
Valoarea unei chei externe trebuie să coincidă cu o valoare deja existentă în tabelul părinte sau să fie NULL.

Observație: Cheile externe sunt bazate pe valorile datelor din cele două tabele și sunt pointeri pur logici, nu fizici.

Constrângere de referință definită la nivel de coloană sau tabel

Constrângerile de chei externe pot fi definite la nivelul unui tabel sau unei coloane. O cheie externă compusă este creată folosind definiția la nivel de tabel.

```
CREATE TABLE angajati(  
    empno      NUMBER(4) ,  
    ename      VARCHAR2(10) NOT NULL ,  
    job        VARCHAR2(9) ,  
    mgr        NUMBER(4) ,  
    hiredate   DATE ,  
    sal        NUMBER(7,2) ,  
    comm       NUMBER(7,2) ,  
    deptno     NUMBER(7,2) NOT NULL ,  
    FOREIGN KEY (deptno) REFERENCES departament (deptno)) ;
```



```
CREATE TABLE angajati(  
empno NUMBER(4),  
ename VARCHAR2(10),  
job VARCHAR2(9),  
mgr NUMBER(4),  
hiredate DATE,  
sal NUMBER(7,2),  
comm NUMBER(7,2),  
deptno NUMBER(7,2) NOT NULL,  
FOREIGN KEY(deptno) REFERENCES departament(deptno));
```

Results Explain Describe Saved SQL History

Table created.

0.05 seconds

Exemplul de mai sus definește o constrângere de tip cheie externă pe coloana deptno din tabelul angajati.

Sintaxa constrângerii de referință este următoarea:


```
FOREIGN KEY (col, ...)
REFERENCES tabel_referit (col, ...) [ON DELETE <CASCADE | SET
NULL>]
```

- *FOREIGN KEY*
Definește coloana din tabelul copil la nivelul constrângerii de tabel.
- *REFERENCES*
Identifică tabelul părinte și coloana din tabelul părinte.
- *ON DELETE CASCADE*
Indică faptul că în situația în care se șterge o linie din tabelul părinte, liniile dependente din tabelul copil vor fi șterse.
- *ON DELETE SET NULL*
Dacă se șterge o linie din tabelul părinte, liniile dependente din tabelul copil vor primi valori NULL în coloanele implicate în constrângere.

Fără opțiunea *ON DELETE CASCADE | SET NULL* linia din tabelul părinte nu va putea fi ștearsă dacă este referită în tabelul copil.

2.5. Constrângerea CHECK

Constrângerea de tip CHECK definește o condiție ce trebuie îndeplinită de fiecare linie dintr-un tabel. Condiția poate utiliza aceleași construcții ca și condițiile de interogare, cu următoarele excepții:

- referiri la pseudocoloanele CURRVAL, NEXTVAL, LEVEL și ROWNUM.
- apeluri de funcții SYSDATE, UID, USER și USERENV.
- interogări care fac referire la alte valori din alte rânduri.

Nu există o limitare pentru numărul de constrângeri CHECK ce pot fi definite pe o coloană. Constrângerile CHECK pot fi definite la nivel de coloană sau de tabel.

Exemplul de mai jos definește o constrângere CHECK pentru coloana deptno din tabelul emp, condiția constrângerii fiind ca valoarea din coloană să fie cuprinsă între 10 și 99.

```
..., deptno NUMBER(2),
      CHECK (DEPTNO BETWEEN 10 AND 99)...
```

3. Adăugarea unei constrângeri

Adăugarea unei constrângeri la un tabel deja existent se face cu instrucțiunea ALTER TABLE cu clauza ADD.

```
ALTER TABLE tabel
ADD tip (coloana);
```

unde:

| | |
|--------------------------|---------------------------|
| <i>tabel</i> | este numele tabelului |
| <i>nume_constrangere</i> | este numele constrângerii |
| <i>tip</i> | este tipul constrângerii |

coloana este numele coloanei afectată de constrângere

Atentie! Deși denumirea constrângerilor este opțională este recomandată furnizarea unor nume. În caz contrar va fi generat un nume de către sistem.

Observații:

- se poate adăuga, șterge, activa sau dezactiva o constrângere, dar nu se poate modifica definiția acesteia.
- se poate adăuga o constrângere de tip NOT NULL la o coloană existentă folosind clauza MODIFY din instrucțiunea ALTER TABLE.

Exemplul următor creează o constrângere de referință (FOREIGN KEY) în tabelul emp. Impunerea acestei constrângeri externe indică faptul că un manager trebuie să existe ca angajat activ în tabelul EMP.

```
ALTER TABLE emp
  ADD FOREIGN KEY (mgr) REFERENCES emp (empno);
```

4. Eliminarea unei constrângeri

Pentru a elimina o constrângere trebuie identificat numele constrângerii. Această identificare se poate face cu ajutorul vederilor USER_CONSTRAINTS și USER_CONS_COLUMNS din dicționarul de date. Odată identificat numele constrângerii se va utiliza instrucțiunea ALTER TABLE împreună cu clauza DROP.

```
ALTER TABLE tabel
  DROP PRIMARY KEY | UNIQUE (coloana) |
  CONSTRAINT nume_constrangere [CASCADE];
```

unde:

tabel este numele tabelului
coloana este numele coloanei afectată de constrângere
nume_constrangere este numele constrângerii.

Opțiunea CASCADE din clauza DROP are ca efect eliminarea tuturor constrângerilor dependente.

Atentie: Când se elimină o constrângere de integritate, aceasta nu mai este folosită de către serverul Oracle și nu mai este disponibilă în dicționarul de date.

Pentru eliminarea constrângerii în raport cu managerii din tabelul emp se va utiliza instrucțiunea

```
ALTER TABLE emp
  DROP CONSTRAINT emp_mgr_fk;
```

Pentru eliminarea constrângerii de tip cheie primară din tabelul `dept`, împreună cu constrângerea externă asociată coloanei `deptno` se poate folosi instrucțiunea:

```
ALTER TABLE dept  
DROP PRIMARY KEY CASCADE;
```

5. Dezactivarea constrângerilor

Se poate dezactiva o constrângere fără să fie necesară eliminarea acesteia sau recrearea sa folosind instrucțiunea `ALTER TABLE` împreună cu clauza `DISABLE`.

```
ALTER TABLE tabel  
DISABLE CONSTRAINT nume_constrangere [CASCADE];
```

unde:

tabel este numele tabelului
nume_constrangere este numele constrângerii.

Notă:

- clauza `DISABLE` se poate folosi atât în instrucțiunea `CREATE TABLE`, cât și în instrucțiunea `ALTER TABLE`.
- clauza `CASCADE` dezactivează constrângerile de integritate dependente.

Exemplul următor dezactivează constrângerea de tip cheie primară din tabelul `emp`. Prin utilizarea clauzei `CASCADE` vor fi dezactivate toate constrângerile dependente.

```
ALTER TABLE emp  
DISABLE CONSTRAINT emp_empno_pk CASCADE;
```

Pentru verificare se va afișa valoarea din coloana `STATUS` din vederea `USER_CONSTRAINTS` pentru constrângerea în cauză. Dacă este afișată valoarea `ENABLED` atunci constrângerea este activată. În cazul în care constrângerea este dezactivată se va afișa valoarea `DISABLED`.

Activarea unei constrângere dezactivate se face prin utilizarea instrucțiunii `ALTER TABLE` împreună cu clauza `ENABLE`.

```
ALTER TABLE tabel  
ENABLE CONSTRAINT nume_constrangere;
```

unde:

tabel este numele tabelului.
nume_constrangere este numele constrângerii.

Notă:

- dacă se activează o constrângere, constrângerea va fi aplicată tuturor datelor din tabel. Toate datele din tabel trebuie să satisfacă constrângerea.

- dacă se activează o constrângere de tip UNIQUE sau PRIMARY KEY, atunci este creat în mod automat un index UNIQUE sau PRIMARY KEY.

- clauza ENABLE se poate folosi atât în instrucțiunea CREATE TABLE, cât și în instrucțiunea ALTER TABLE.

Exemplu:

```
ALTER TABLE emp
    ENABLE CONSTRAINT emp_empno_pk;
```

6. Vizualizarea constrângerilor

După crearea unui tabel, se poate confirma existența sa prin folosirea instrucțiunii DESCRIBE. Singura constrângere care poate fi verificată în acest mod este constrângerea NOT NULL. Pentru a vedea toate constrângerile din tabel, este necesară interogarea tabelului USER_CONSTRAINTS.

Exemplul următor afișează toate constrângerile definite în tabelul EMP.

```
SELECT constraint_name, constraint_type,
       search_condition
FROM user_constraints
WHERE table_name='EMP';
```

Notă: Constrângerilor care nu primesc un nume de la proprietarul tabelului li se atribuie un nume automat de către sistem. Coloana C indică tipul constrângerii: C provine de la CHECK, P de la PRIMARY KEY, R de la integritate referențială și U de la UNIQUE. De observat faptul că o constrângere NOT NULL este de fapt o constrângere de tip CHECK.

Vizualizarea coloanelor asociate constrângerilor

Pentru vizualizarea numelor coloanelor implicate în constrângeri se va interoga vederea USER_CONS_COLUMNS din dicționarul de date. Această vizualizare este utilă în cazul utilizării mecanismului de asociere automată a numelor pentru constrângeri de către sistem.

```
SELECT constraint_name, column_name,
       FROM user_cons_columns
       WHERE table_name='EMP';
```

Probleme:

1. Adăugați o constrângere de tip PRIMARY KEY la tabelul EMP folosind coloana EMPNO. Constrângerea trebuie să fie activată la creare.
2. Creați o constrângere PRIMARY KEY în tabelul DEPARTMENT folosind coloana ID. Constrângerea trebuie să fie activată la creare.
3. Adăugați o referință de tip cheie externă la tabelul EMP care va asigura că angajații nu sunt asignați la un departament inexistent.
4. Confirmați constrângerile adăugate interogând tabelul USER_CONSTRAINTS. Observați numele și tipul constrângerilor.

| CONSTRAINT_NAME | C |
|-------------------|---|
| ----- | - |
| DEPARTEMENT_ID_PK | P |
| EMP_EMPNO_PK | P |
| EMP_DEPT_ID_FK | R |

5. Afișați numele și tipul obiectelor din vederea USER_OBJECTS pentru tabelele EMP și DEPARTMENT.

| OBJECT_NAME | OBJECT_TYPE |
|------------------|-------------|
| ----- | ----- |
| DEPARTMENT | TABLE |
| DEPARTMENT_ID_PK | INDEX |
| EMPLOYEE | TABLE |
| EMPLOYEE_ID_PK | INDEX |