

# CREAREA ȘI UTILIZAREA VEDERILOR. UTILIZAREA OBIECTELOR DE TIP SECVENȚĂ PENTRU GENERARE AUTOMATĂ DE VALORI

## 1. Obiective

Prin parcurgerea acestui referat studentul va dobândi cunoștințele necesare pentru descrierea unei vederi;

- crearea unui obiect de tip vedere;
- extragerea datelor dintr-o vedere;
- modificarea structurii unei vederi;
- inserarea, actualizarea și ștergerea datelor dintr-o vedere;
- ștergerea unei vederi;
- crearea și utilizarea generatoarelor de secvențe.

## 2. CREAREA ȘI UTILIZAREA VEDERILOR.

O *vedere* este un tabel logic bazat pe un alt tabel sau pe o altă vedere și nu conține date proprii. Tabelele pe baza cărora sunt create vederile se numesc *tabele de bază*. La nivelul bazei de date o vedere este stocată sub forma unei instrucțiuni SELECT în dicționarul de date.

### 2.2. Avantajele utilizării vederilor

Vederile prezintă o serie de vederi:

- restricționează accesul la baza de date deoarece o vedere pune la dispoziția utilizatorilor doar o parte a datelor din baza de date;
- permit utilizatorilor să apeleze la interogări simple pentru a obține rezultatele unor interogări complexe. De exemplu, vederile permit utilizatorilor să extragă informații din mai multe tabele fără a cunoaște modul de realizare a joncțiunilor..
- asigură independența datelor pentru utilizatori și aplicații. O singură vedere poate fi utilizată pentru a obține date din mai multe tabele;
- furnizează grupurilor de utilizatori accesul la date în conformitate cu anumite criterii, stabilite de administratorul bazei de date.

Vederile se clasifică în două grupe: simple și complexe. Diferența de bază între cele două grupe este legată de operațiile LMD (inserare, actualizare și ștergere).

Caracteristici	Vederi simple	Vederi complexe
Număr de tabele	1	1 sau mai multe
Conține funcții	Nu	Da
Conține grupuri de date	Nu	Da
Operații LMD	Da	Nu întotdeauna

O vedere simplă este o vedere care:

- furnizează date dintr-un singur tabel;
- nu conține funcții sau grupuri de date;
- permite execuția unor operații LMD.

O vedere complexă este o vedere care :

- extrage date din mai multe tabele;

- conține funcții sau grupuri de date;
- nu permite întotdeauna operații LMD.

### 2.3. Crearea unei vederi

Instrucțiunea ce permite crearea unei vederi este `CREATE VIEW`, având următoarea sintaxă.

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW nume_vedere
[(alias[, alias...])]
AS subinterogare
[WITH CHECK OPTION [CONSTRAINT nume_constr]]
[WITH READ ONLY]
```

unde:

**OR REPLACE** : - recrează vederea dacă aceasta există deja

**FORCE** : - creează vederea indiferent dacă există sau nu tabelele de bază

**NOFORCE** :- creează vederea numai dacă există tabelele de bază; este opțiunea implicită

*nume\_vedere* : - este numele vederii

*alias* : - specifică noul nume pentru o coloană a vederii;

numărul de alias-uri trebuie să corespundă numărului de expresii selectate de vedere

*subinterogare* - este o instrucțiune `SELECT`; se pot utiliza alias-uri pentru coloanele din clauza `SELECT`

**WITH CHECK OPTION** - specifică faptul că pot fi actualizate doar liniile accesibile vederii

*nume\_constr* : - este numele constrângerii

**WITH READ ONLY** : - interzice execuția unei instrucțiuni LMD pe vedere.

Pentru a crea o vedere numită `EMPVU10` care conține detalii despre angajații din departamentul 10 se va executa următoarea instrucțiune:

```
CREATE VIEW empvu10
AS SELECT empno, ename, job
FROM emp
WHERE deptno=10;
```

#### Observații:

- subinterogarea care definește o vedere poate conține o instrucțiune `SELECT` complexă (inclusiv joncțiuni, grupuri și subinterogări);

- subinterogarea care definește o vedere nu poate conține clauza `ORDER BY`. Această clauză va fi specificată în momentul interogării vederii;

- opțiunea `OR REPLACE` poate fi utilizată pentru a putea modifica vederea fără a o șterge și recrea.

Redenumirea coloanelor ce apar în vedere se poate face în două moduri:

1. se includ alias-urilor de coloană în subinterogare:

```
CREATE VIEW      salvu30
AS SELECT empno EMPLOYEE_NUMBER, ename NAME, sal SALARY
FROM            emp
```

```
WHERE deptno=30;
```

Exemplul anterior creează o vedere care conține numărul, numele și salariul angajaților din departamentul 30. Coloanele acestei vederi au numele EMPLOYEE\_NUMBER, NAME și SALARY.

2. se includ alias-uri de coloană în clauza CREATE VIEW.

```
CREATE VIEW salvu30 (EMPLOYEE_NUMBER, NAME, SALARY)
AS SELECT empno, ename, sal
FROM emp
WHERE deptno=30;
```

#### 2.4. Interogarea vederilor

Extragerea datelor din vederi se face în același mod ca în cazul tabelelor.

```
SQL> SELECT * FROM salvu30;
```

EMPLOYEE_NUMBER	NAME	SALARY
7698	BLAKE	2850
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
7521	WARD	1250

Odată creată vederea, se poate interoga tabelul USER\_VIEWS din dicționarul de date pentru a obține numele și definiția vederii. Textul instrucțiunii SELECT care implementează vederea este memorat într-o coloană de tip LONG (șir de caractere).

Accesarea datelor din baza de date corespunzătoare unei vederi presupune execuția următoarelor operații de către serverul Oracle:

1. extrage definiția vederii din dicționarul de date (mai exact din tabelul USER\_VIEWS);
2. verifică privilegiile de acces pentru tabelele de bază;
3. convertește interogarea vederii într-o operație echivalentă asupra tabelului (tabelelor) de bază. Cu alte cuvinte, datele sunt extrase din, sau actualizate în tabelul (tabelele) de bază.

#### 2.5. Modificarea unei vederi

Opțiunea OR REPLACE permite crearea unei vederi, chiar dacă există deja o vedere cu același nume, înlocuindu-se astfel vechea versiune a vederii cu cea nouă. Aceasta înseamnă că o vedere poate fi modificată fără a fi necesară ștergerea și recrearea ei.

Următorul exemplu modifică vederea EMPVU10 utilizând clauza CREATE OR REPLACE VIEW. Suplimentar se adaugă un alias pentru fiecare coloană.

```
CREATE OR REPLACE VIEW empvu10 (employee_number,
employee_name, job_title)
AS SELECT empno, ename, job
FROM emp
WHERE deptno=10;
```

**Notă:** Când se furnizează nume alternative (alias-uri) coloanelor în clauza `CREATE VIEW`, alias-urile trebuie listate în aceeași ordine ca și coloanele din subinterogare.

## 2.6. Crearea unei vederi complexe

```
CREATE VIEW dept_sum_vu(name, minsal, maxsal, avgsal)
AS SELECT d.dname, MIN(e.sal), MAX(e.sal), AVG(e.sal)
FROM emp e, dept d
WHERE e.deptno=d.deptno
GROUP BY d.dname;
```

Exemplul anterior creează o vedere complexă care conține numele, salariul minim, salariul maxim și salariul mediu pentru fiecare departament. De notat că au fost specificate alias-uri pentru vedere. **Acest lucru este necesar dacă una din coloanele vederii rezultă din evaluarea unei funcții sau expresii.**

## 2.7. Reguli pentru executarea operațiilor LMD asupra unei vederi

Pentru a putea executa operații LMD asupra datelor din baza de date prin intermediul vederii trebuie avute în vedere o serie de reguli.

1. Nu se poate șterge o linie aparținând unei vederi dacă definiția vederii conține :
  - funcții grup;
  - clauză `GROUP BY`;
  - cuvântul cheie `DISTINCT`.
2. Nu se pot modifica datele dintr-o vedere dacă vederea conține:
  - oricare din elementele de la punctul 1;
  - coloane definite prin expresii (de exemplu, `SALARY*12`);
  - pseudocoloana `ROWNUM`.
3. Nu pot fi inserate date într-o vedere dacă:
  - vederea conține oricare din elementele de la punctul 1 și 2;
  - există coloane `NOT NULL` fără valoare implicită în tabelul de bază și care nu au fost selectate în vedere.

## 2.8. Utilizarea clauzei `WITH CHECK OPTION`

Există posibilitatea ca, prin intermediul vederilor, să se efectueze verificări de integritate referențială asupra datelor. Se pot, de asemenea, forța constrângeri la nivelul bazei de date. O vedere poate fi utilizată pentru a asigura integritatea datelor, dar într-o manieră destul de limitată.

Clauza `WITH CHECK OPTION` specifică faptul că instrucțiunile `INSERT` și `UPDATE` executate asupra vederii nu pot crea linii noi pe care vederea nu le poate selecta. Prin urmare, vederile pot forța verificări asupra datelor ce vor fi inserate sau actualizate. Dacă se încearcă execuția unei operații LMD pe linii care nu au fost selectate de vedere se va afișa un mesaj de eroare, împreună cu numele constrângerii (dacă acesta a fost specificat).

Pentru exemplificare să considerăm vederea

```
CREATE OR REPLACE VIEW empvu20
AS SELECT *
FROM emp
WHERE deptno=20
WITH CHECK OPTION CONSTRAINT empvu20_ck;
```

Orice încercare de a modifica numărul departamentului pentru orice linie din vedere va eșua deoarece nu respectă constrângerea WITH CHECK OPTION.

```
UPDATE empvu20
SET deptno=10
WHERE empno=7788;
```

```
update empvu20
      *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

Nici o linie nu este actualizată pentru că, dacă numărul departamentului ar deveni 10, vederea nu ar mai putea selecta acel angajat. Datorită faptului că vederea poate “vedea” doar angajații din departamentul 20, utilizarea clauzei WITH CHECK OPTION nu permite ca numărul departamentului pentru acești angajați să fie schimbat prin intermediul vederii.

## 2.9 Interzicerea operațiilor LMD

Dacă se dorește ca asupra vederii să nu poată fi executată nici o operație LMD, la crearea vederii trebuie specificată opțiunea WITH READ ONLY.

```
CREATE OR REPLACE VIEW empvu10
(employee_number, employee_name, job_title)
AS SELECT empno, ename, job
FROM emp
WHERE deptno=10
WITH READ ONLY;
```

Exemplul de mai sus modifică vederea EMPVU10 pentru a împiedica orice operație LMD asupra vederii. Orice încercare de ștergere a unei linii din vedere va genera o eroare.

```
DELETE FROM empvu10
WHERE employee_number=7782;
```

```
DELETE FROM empvu10
      *
ERROR at line 1:
ORA_01752: Cannot delete from view without exactly one
key-preserved table
```

## 2.10. Ștergerea definiției unei vederi

Pentru a șterge o vedere se va utiliza instrucțiunea:

```
DROP VIEW nume_vedere;
```

Exemplu:

```
DROP VIEW empvu10;
```

Această instrucțiune elimină definiția vederii din baza de date. Ștergerea vederilor nu are nici un efect asupra tabelelor pe care se bazează vederea. Vederile sau aplicațiile bazate pe o vedere ștearsă devin invalide. Ștergerea unei vederi poate fi efectuată fie de creatorul vederii fie de un utilizator ce deține privilegiul `DROP ANY VIEW`.

## 3. Obiecte de tip secvență

Multe aplicații necesită folosirea de valori distincte pentru o cheie primară. Pentru a îndeplini această cerință fie se scriu linii de cod, fie se recurge la utilizarea obiectelor de tip secvență pentru a genera automat aceste numere distincte. O secvență este un obiect creat de un utilizator și poate fi folosit de mai mulți utilizatori. Secvența este generată și incrementată (sau decrementată) de o rutină internă Oracle. Numerele secvenței sunt stocate și generate independent de tabel. Prin urmare, aceeași secvență poate fi folosită în mai multe tabele.

### 3.1 Crearea unui obiect de tip secvență

Instrucțiunea `CREATE SEQUENCE` definește o secvență ce va genera automat numere secvențiale. Sintaxa acestei instrucțiuni este următoarea:

```
CREATE SEQUENCE secventa  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE n | NOMAXVALUE}]  
[{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]  
[{CACHE n | NOCACHE}] ;
```

unde:

`secventa`: - este numele generatorului de secvență;

`INCREMENT BY n`: - specifică intervalul dintre două numere consecutive ale secvenței, unde  $n$  este un întreg. Dacă această clauză lipsește,  $n$  este implicit 1;

`START WITH n`: - specifică primul număr al secvenței. Dacă această clauză lipsește, implicit  $n = 1$ ;

`MAXVALUE n`: - specifică valoarea maximă pe care secvența o poate genera;

`NOMAXVALUE`: - specifică o valoare maximă de  $10^{27}$  pentru o secvență crescătoare și  $-1$  pentru o secvență descrescătoare. Aceste valori sunt implicite;

`MINVALUE n`: - specifică valoarea minimă a secvenței;

`NOMINVALUE`: - specifică o valoare minimă de 1 pentru o secvență crescătoare și  $(-10^{27})$  pentru una descrescătoare. Aceste valori sunt implicite;

CYCLE | NOCYCLE: - specifică dacă secvența continuă să genereze valori după ce a fost atinsă valoarea maximă / minimă, sau dacă nu mai generează nici o valoare. Opțiunea NOCYCLE este implicită;

CACHE *n* | NOCACHE: - specifică serverului Oracle pentru câte valori să aloce spațiu în memorie. Implicit *n* are valoarea 20.

Exemplul următor creează secvența DEPT\_DEPTNO, folosită pentru coloana DEPTNO din tabelul DEPT. Secvența începe cu valoarea 91, nu permite alocare de memorie și nu permite cicluri de generare.

```
CREATE SEQUENCE dept_deptno
INCREMENT BY 1
START WITH 91
MAXVALUE 100
NOCACHE
NOCYCLE;
```

Sequence created.

**Notă:** Nu se recomandă folosirea clauzei CYCLE atunci când secvența este folosită pentru a genera valorile unei chei primare, excepție făcând cazul când se dispune de un mecanism sigur de îndepărtare a vechilor înregistrări și mai rapid decât ciclurile secvenței.

### 3.2. Confirmarea existenței obiectelor de tip secvență

Odată creată o secvență, ea este înscrisă în dicționarul de date și poate fi identificată cu ajutorul tabelului USER\_OBJECTS. De asemenea, se pot confirma setările unei secvențe prin interogarea tabelului USER\_SEQUENCES.

```
SELECT sequence_name, min_value, max_value,
increment_by, last_number
FROM user_sequences;
```

- Coloana LAST\_NUMBER conține următorul număr disponibil al secvenței.

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
CUSTID	1	1.000E+27	1	109
DEPT_DEPTNO	1	100	1	91
ORDID	1	1.000E+27	1	622
PRODID	1	1.000E+27	1	200381

### 3.3 Pseudocoloanele NEXTVAL și CURRVAL

Odată creată secvența, ea poate fi folosită pentru a genera numere ce vor fi utilizate ulterior în tabele. Referirea următoarei valori din secvență și a valorii curente se face cu ajutorul pseudocoloanelor NEXTVAL și CURRVAL.

Pseudocoloana NEXTVAL este folosită pentru a extrage numere succesive dintr-o anumită secvență. Referirea la NEXTVAL se face împreună cu numele secvenței. Atunci când se face referire la secvența.NEXTVAL se generează un nou număr al secvenței, ce va fi memorat în CURRVAL.

Pseudocoloana CURRVAL este folosită pentru a referi numărul unei secvențe ce tocmai a fost generat. Înainte de a se face referire la CURRVAL trebuie utilizat NEXTVAL

pentru a genera un nou număr. Referirea lui CURRVAL se face împreună cu numele secvenței. Când `sequence.CURRVAL` este referită, va fi afișată ultima valoare generată de secvență.

### 3.4 Reguli pentru utilizarea pseudocoloanelor NEXTVAL și CURRVAL

Folosirea pseudocoloanelor NEXTVAL și CURRVAL este permisă în următoarele situații:

- în clauza SELECT a unei instrucțiuni SELECT ce nu face parte dintr-o subinterogare;
- în clauza SELECT a unei subinterogări dintr-o instrucțiune INSERT;
- în clauza VALUES a unei instrucțiuni INSERT;
- în clauza SET a unei instrucțiuni UPDATE.

Nu este permisă utilizarea pseudocoloanelor NEXTVAL și CURRVAL în următoarele situații:

- în clauza SELECT a unei vederi;
- într-o instrucțiune SELECT ce conține cuvântul cheie DISTINCT;
- într-o instrucțiune SELECT ce conține clauzele GROUP BY, HAVING sau ORDER BY;
- într-o subinterogare din instrucțiuni SELECT, DELETE sau UPDATE;
- într-o expresie DEFAULT în instrucțiuni CREATE TABLE sau ALTER TABLE.

### 3.5 Utilizarea unei secvențe

Exemplul următor inserează un nou departament în tabelul DEPT. Se utilizează secvența DEPT\_DEPTNO pentru generarea unui nou număr pentru departament.

```
INSERT INTO dept(deptno, dname, loc)
VALUES (dept_deptno.NEXTVAL, 'MARKETING', 'SAN DIEGO');
```

```
1 row created.
```

Pentru vizualizarea valorii curente pentru secvența DEPT\_DEPTNO se va executa următoarea instrucțiune:

```
SQL> SELECT dept_deptno.CURRVAL
2 FROM dual;
```

```
CURRVAL
-----
91
```

Stocarea secvențelor în memorie permite o accesare mai rapidă a valorilor secvenței, stocarea realizându-se la prima referire a secvenței. După ce a fost utilizată ultima valoare din secvență, următoarea cerere de generare determină o altă stocare a secvenței în memorie.

Deși generatorul de secvențe furnizează numere consecutive fără goluri, această acțiune se petrece independent de un commit sau un rollback. De aceea, dacă se execută un rollback pe o instrucțiune ce utilizează o valoare dintr-o secvență, acea valoare este pierdută. Un alt eveniment ce poate provoca goluri în secvență este o cădere a sistemului. Dacă valorile secvenței sunt stocate în memorie, acestea se pierd în cazul unei căderi de sistem.



Este posibilă vizualizarea următoarei valori disponibile din secvență fără a o incrementa, doar dacă secvența a fost creată cu opțiunea NOCACHE, prin interogarea tabelului USER\_SEQUENCES.

### 3.6 Modificarea unei secvențe

Dacă se atinge valoarea MAXVALUE într-o secvență definită cu opțiunea NOCYCLE nu va mai fi atribuită nici o valoare suplimentară și se va genera un mesaj de eroare ce indică faptul că secvența a atins valoarea maximă. Pentru a putea folosi în continuare această secvență, va trebui să modificăm definiția secvenței cu ajutorul instrucțiunii ALTER SEQUENCE.

```
ALTER SEQUENCE nume_secv
[INCREMENT BY n]
[{MAXVALUE n | NOMAXVALUE}]
[{MINVALUE n | NOMINVALUE}]
[{CYCLE | NOCYCLE}]
[{CACHE n | NOCACHE}];
```

unde:

*nume\_secv* este numele generatorului de secvențe

Exemplul următor modifică valoarea incrementului, valoarea maximă și minimă, opțiunea de ciclare și de stocare pentru secvența DEPT\_DEPTNO.

```
ALTER SEQUENCE dept_deptno
INCREMENT BY 1
MAXVALUE 999999
NOCACHE
NOCYCLE;
```

```
Sequence altered.
```

Pentru verificare se vor afișa informațiile din vederea USER\_SEQUENCES.

#### Observații:

- o secvență poate fi modificată doar de utilizatorul care a creat secvența sau de un utilizator ce deține privilegiul ALTER pentru secvența în cauză;
- sunt afectate doar numerele ce urmează să fie generate;
- secvența trebuie distrusă și recreată pentru a începe secvența cu un alt număr (specificat în clauza START WITH).

### 3.7 Ștergerea unei secvențe

Pentru ștergerea unei secvențe din dicționarul de date se utilizează instrucțiunea DROP SEQUENCE. Această operație poate fi realizată doar de proprietarul secvenței sau de utilizatorul cu privilegiul DROP ANY SEQUENCE.

```
DROP SEQUENCE nume_secv;
```

Exemplu:

```
SQL> DROP SEQUENCE dept_deptno;
```

```
Sequence dropped.
```

## Probleme

1. Creați o vedere EMP\_VU care să conțină numărul, numele și departamentul angajaților din tabelul emp. Redenumiți coloana corespunzătoare numelui EMPLOYEE. Afișați conținutul vederii EMP\_VU.
2. Vizualizați informațiile despre vederea creată la problema 1 din vederea USER\_VIEWS (coloanele view\_name și text).
3. Folosind vederea EMP\_VU scrieți o interogare pentru a afișa numele tuturor angajaților, împreună cu departamentele în care lucrează.
4. Creați vederea DEPT20 care să conțină următoarele informații despre angajații ce lucrează în departamentul 20: număr angajat, nume angajat și număr departament. Etichetați coloanele vederii astfel: EMPLOYEE\_ID, EMPLOYEE și DEPARTMENT\_NO. Nu permiteți ca unui angajat să i se poată modifica numărul departamentului prin intermediul vederii.
5. Afișați structura și conținutul vederii DEPT20.
6. Încercați să înlocuiți numărul departamentului pentru angajatul Smith cu valoarea 30. Explicați cauza ce a condus la generarea erorii semnalată de server.
7. Creați vederea SALARY\_VU care să conțină numele angajatului, numele departamentului, salariul și gradul de salarizare pentru toți angajații din emp. Etichetați coloanele astfel: Employee, Department, Salary și Grade.
8. Să se creeze o secvență ce va fi folosită împreună cu cheia primară a tabelului DEPARTMENT. Secvența trebuie să înceapă cu 60 și să aibă o valoare maximă de 200. Incrementul secvenței este de 10. Numele secvenței va fi DEPT\_ID\_SEQ.
9. Să se afișeze următoarele informații despre secvența de la problema 8: nume, valoarea maximă, mărimea incrementului și ultimul număr creat.
10. Să se realizeze un program interactiv pentru a insera o înregistrare în tabelul DEPARTMENT. Pentru introducerea valorilor în coloana ID se va utiliza secvența creată la problema 8. Se va cere introducerea de la tastatură a numelui departamentului. Să se adauge două departamente numite 'Education' și 'Administration'. Să se confirme adăugarea celor două înregistrări.

Structura tabelului DEPARTMENT este următoarea:

Column Name	ID	NAME
Nulls/Unique		
Datatype	Number(7)	Varchar2(25)