

Tutoriat #9

Views

Vizualizări (Views)

- sunt **tabele virtuale** (i.e. nu există fizic în baza de date), relative la date din una sau mai multe tabele de bază
- o vizualizare este definită plecând de la o cerere SQL (în baza de date se stochează, de fapt, definiția cererii)
- de fiecare dată când se interoghează vizualizarea, se va executa cererea respectivă
- vizualizările au rolul de a "masca" tabelele de bază, creând un nivel de securitate a bazei de date prin restricționarea accesului la date, dar sunt folosite și pentru a simplifica cereri mai complexe

Sintaxa – crearea unei vizualizări

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW nume_viz  
[ ( alias [ , alias ]* ) ]  
AS subcerere  
[WITH CHECK OPTION [CONSTRAINT nume_constrangere]]  
[WITH READ ONLY [CONSTRAINT nume_constrangere]];
```

- **FORCE**: creează vizualizarea chiar dacă tabelul de bază nu există
- **WITH CHECK OPTION**: înainte de o modificare a datelor prin intermediul vizualizării, se va verifica faptul că acestea respectă **constrângerea de validare** (definită în clauza **WHERE** a subcererii)
- **WITH READ ONLY**: asigură că prin intermediul vizualizării nu se mai pot executa operații LMD
- **OR REPLACE**: recreează vizualizarea dacă aceasta există deja

Sintaxa – ștergerea unei vizualizări

```
DROP VIEW nume-viz;
```

Vizualizările pot fi:

- simple: definite pe baza unui singur tabel, nu conțin funcții sau grupări de date, permit operații LMD

Exemplu:

```
create or replace view info_emp as  
select employee_id, last_name, sysdate - hire_date as num_days  
from employees;
```

- complexe: definite pe baza mai multor tabele, pot conține funcții sau grupări de date, nu permit întotdeauna operații LMD

Actualizarea datelor prin intermediul vizualizărilor are loc doar dacă sunt respectate anumite restricții:

- Nu se pot realiza operații **LMD** (insert, update, delete) asupra vizualizărilor, dacă acestea conțin: funcții agregat (MIN, COUNT, etc.), clauza GROUP BY, pseudocoloana ROWNUM, keyword-ul DISTINCT, operatori pe mulțimi (UNION, INTERSECT, MINUS), coloane definite prin expresii (ex: CASE);
- Nu pot fi **inserate** linii printr-o vizualizare dacă tabelul de bază conține coloane care au constrângerea NOT NULL și nu apar în vizualizare;
- Nu se pot **actualiza coloane** ale căror valori rezultă prin calcul sau care sunt definite cu DECODE;
- Într-o vizualizare (complexă sau simplă), o operație LMD poate afecta **doar un singur tabel de bază**, iar acesta trebuie să fie key-preserved (protejat prin cheie).

*Un tabel de bază se numește **key-preserved** dacă are proprietatea că fiecare valoare a cheii sale primare sau a unei coloane având constrângerea de unicitate, este unică și în vizualizare.*

Exemplu:

```
CREATE OR REPLACE VIEW emp_dep AS
SELECT e.employee_id, e.first_name, e.last_name, e.salary,
d.department_name, d.location_id
FROM employees e JOIN departments d ON e.department_id = d.department_id;

--interogam vizualizarea user_updatable_columns din dictionarul datelor
--pentru a vedea care coloane pot fi actualizate
SELECT table_name, column_name, insertable, updatable, deletable
FROM user_updatable_columns
WHERE table_name LIKE 'emp_dep' ;
```

!! Dacă se actualizează tabelul, datele apar automat și în vizualizare; dacă se actualizează vizualizarea, tabelul de bază se actualizează doar dacă sunt respectate restricțiile !!

INSERT	UPDATE	DELETE
Toate coloanele în care sunt inserate valori trebuie să fie dintr-un tabel key-preserved	Toate coloanele updatable sunt dintr-un tabel key-preserved	Se pot șterge rânduri din view doar dacă acesta conține un singur tabel, iar acesta este key-preserved

!! Atenție, orice operație LMD aplicată unei vizualizări, trebuie mai întâi să respecte constrângerile impuse tabelului de bază asupra căreia se aplică modificarea !!

Interogări utile din dicționarul datelor:

- Din tabelul (vizualizarea) **user_views**, puteți obține codul corespunzător vizualizărilor utilizatorului curent

```
SELECT view_name, text
FROM user_views
WHERE view_name LIKE '...';
```

- Puteți consulta vizualizarea user_updatable_columns pentru a obține informații referitoare la coloanele modificabile ale anumitor vizualizări:

```
SELECT table_name, column_name, insertable, updatable, deletable
FROM user_updatable_columns
WHERE table_name LIKE '...';
```

Vizualizări inline

- Un inline view este o subcerere însoțită de un alias. Acestea pot apărea în comenzile SELECT, INSERT, UPDATE, DELETE
- Spre deosebire de vizualizările propriu-zise, nu reprezintă obiecte ale schemei bazei de date, ci sunt "vizibile" doar la nivelul comenzii în cadrul căreia sunt definite
- În comanda SELECT, un inline view este, de fapt, o subcerere nesincronizată în clauza FROM. Sunt folosite pentru a simplifica scrierea comenzilor.

Exemplu:

```
-- selectati numele departamentelor si orasul pt acele departamente cu
sediul in Regatul Unit

SELECT d.department_name, orase_uk.city
FROM departments d,
(SELECT loc.city, loc.location_id FROM locations loc WHERE loc.country_id
= 'UK') orase_uk
WHERE orase_uk.location_id = d.location_id;
```