

VIZUALIZĂRI MATERIALIZATE

TEHNICI DE RESCRIERE A CERERILOR

1. a. Definiți vizualizarea materializată *vm_clienti_facturi1_**** care să conțină pentru fiecare client codul, numele (tabela *clienti_****), respectiv numărul de facturi emise pentru acesta (tabela *vanzari_factura_**** care conține valoarea vânzărilor la nivel de factură).

- Vizualizarea va fi populată cu date la crearea acesteia.

BUILD IMMEDIATE

- Reactualizările ulterioare ale vizualizării se vor realiza prin rularea completă a cererii din definiție vizualizării.

COMPLETE

- Reactualizarea se va realiza la permanentizarea tranzacției.

COMMIT

```
CREATE MATERIALIZED VIEW nume_vizualizare
    BUILD moment_populare_cu_date
    REFRESH mod_de_reactualizare
    ON moment_de_reactualizare
    AS cerere_SQL;
```

- b. Afisați informațiile stocate în această vizualizare.

- c. Inserați o linie nouă în tabela *vanzari_factura_**** care să conțină date referitoare la o factură a clientului 100.

- d. Verificați dacă în vizualizare datele au fost reactualizate.

- e. Permanentizați tranzacția și apoi consultați informațiile stocate în vizualizare.

2. a. Definiți vizualizarea materializată *vm_clienti_facturi2_**** care să conțină pentru fiecare client codul, numele (tabela *clienti_****), respectiv numărul de facturi emise pentru acesta (tabela *vanzari_factura_**** care conține valoarea vânzărilor la nivel de factură).

- Vizualizarea nu va fi populată cu date la crearea acesteia.

BUILD DEFERRED

- Reactualizările ulterioare ale vizualizării se vor realiza prin rularea completă a cererii din definiția vizualizării.

COMPLETE

- Prima reactualizare a vizualizării se va realiza peste 5 de secunde (o zi are 86400 secunde).

START WITH data_start

- Următoarele reactualizări se vor realiza din 15 în 15 secunde.

NEXT data_urmatoare

```
CREATE MATERIALIZED VIEW nume_vizualizare
    BUILD moment_populare_cu_date
    REFRESH mod_de_reactualizare
    START WITH data_start
    NEXT data_urmatoare
    AS cerere_SQL;
```

- b.** Afişaţi informaţiile stocate în această vizualizare. Aşteptaţi 5 secunde şi încercaţi iar.
 - c.** Inseraţi o linie în tabela *vanzari_factura_**** care să conţină date referitoare la aceeaşi factură (inserată în exerciţiul anterior) a clientului 100, dar cu alt produs.
 - d.** Verificaţi dacă în vizualizare datele au fost reactualizate.
 - e.** Aşteptaţi 15 secunde şi apoi consultaţi informaţiile stocate în vizualizare. Ce observaţi?
 - f.** Permanentizaţi tranzacţia şi verificaţi iar.
- 3. a.** Definiţi vizualizarea materializată *vm_produse_facturi1_**** care să conţină pentru fiecare produs codul, denumirea (tabela *produse_****), respectiv cantitatea totală vândută (tabela *vanzari_****).

- Vizualizarea va fi populată cu date la crearea acesteia.

BUILD IMMEDIATE

- Reactualizările ulterioare ale vizualizării se vor realiza prin reexecuţia completă a cererii din definiţie vizualizării.

COMPLETE

- Reactualizarea se va realiza la cerere.

DEMAND

```
CREATE MATERIALIZED VIEW nume_vizualizare
    BUILD moment_populare_cu_date
    REFRESH mod_de_reactualizare
    ON moment_de_reactualizare
    AS cerere_SQL;
```

- b.** Afişaţi informaţiile stocate în această vizualizare.
- c.** Inseraţi o linie nouă în tabela *vanzari_**** care să conţină date referitoare la produsul având codul 321 de pe factura având codul 11 a clientului 100. Permanentizaţi tranzacţia.

- d. Verificați dacă în vizualizare datele au fost reactualizate.
- e. Actualizați vizualizarea utilizând procedura REFRESH din pachetul DBMS_MVIEW.
- ```
EXECUTE DBMS_MVIEW.REFRESH ('vm_produse_facturi1_***');
```
- f. Verificați dacă în vizualizare datele au fost reactualizate.
4. a. Definiți vizualizarea materializată *vm\_produse\_facturi2\_\*\*\** care să conțină pentru fiecare produs codul, denumirea (tabela *produse\_\*\*\**), respectiv cantitatea totală vândută (tabela *vanzari\_\*\*\**).
- Vizualizarea va fi populată cu date la crearea acesteia.
    - BUILD IMMEDIATE
  - Reactualizarea vizualizării se va realiza doar pentru înregistrările modificate ulterior ultimei reactualizări.
    - FAST
  - Reactualizarea se va realiza la permanentizarea actualizărilor pe tabelele de bază.
    - COMMIT

Rulați următoarea comandă. Ce observați?

```
CREATE MATERIALIZED VIEW nume_vizualizare
 BUILD moment_populare_cu_date
 REFRESH mod_de_reactualizare
 ON moment_de_reactualizare
 AS
 select id_produs,max(denumire), sum(cantitate),
 max(a.rowid) prod_rowid, max(b.rowid) vanz_rowid,
 count(*), count(id_produs)
 from produse_*** a, vanzari_*** b
 where id_produs=produs_id
 group by id_produs;
```

**Obs.** Pentru a optimiza actualizarea rapidă, lista SELECT trebuie să conțină toate coloanele din clauza GROUP BY (dacă este prezentă) și o funcții COUNT(\*), respectiv COUNT(coloana) pentru fiecare coloană agregată.

- b. Definiți log-uri pentru cele două tabele referite în vizualizare.

```
CREATE MATERIALIZED VIEW LOG ON nume_tabelă
 WITH ROWID, PRIMARY KEY, SEQUENCE
 (listă_coloane_tabelă)
 INCLUDING NEW VALUES;
```

*Observații:*

- Logurile vizualizării materializate (*materialized view logs*) trebuie să fie prezente pe toate tabelele referite de cererea ce definește vizualizarea.
  - Pentru a putea defini log-ul tabela trebuie să aibă definită cheia primară care poate fi în stare RELY.
- c.** Încercați să definiți din nou vizualizarea.
- d.** Inserați o linie nouă în tabela *vanzari\_\*\*\** care să conțină date referitoare la produsul având codul 25 de pe factura având codul 11 a clientului 100.
- e.** Verificați dacă în log a fost înregistrată o linie nouă (tabela *mlog\$\_vanzari\_\*\*\**).
- f.** Verificați dacă vizualizarea reflectă modificarea realizată.
- g.** Permanentizați tranzacția. Reluați punctul *f*.
- 5. a.** Definiți vizualizarea materializată *vm\_clienti\_vanzari1\_\*\*\** care să conțină pentru fiecare client codul, numele, respectiv valoarea totală a facturilor (tabela *vanzari\_factura\_\*\*\**).
- Vizualizarea va fi populată cu date la crearea acesteia.  
BUILD IMMEDIATE
  - Reactualizarea vizualizării se va realiza complet.  
COMPLETE
  - Reactualizarea se va realiza la cerere.  
DEMAND
  - Vizualizarea va putea fi utilizată pentru rescrierea cererilor.  
ENABLE QUERY REWRITE
- ```
CREATE MATERIALIZED VIEW nume_vizualizare
    BUILD moment_populare_cu_date
    REFRESH mod_de_reactualizare
    ON moment_de_reactualizare
    ENABLE QUERY REWRITE
    AS cerere_SQL;
```
- b.** Afishați informațiile conținute în vizualizare.
- c.** Folosind tabela de bază *vanzari_factura_**** afișați pentru fiecare client codul, numele, respectiv valoarea totală a facturilor.
- d.** Colectați statistici pentru tabela *vanzari_factura_****.
- ```
ANALYZE TABLE nume_tabelă COMPUTE STATISTICS;
```

e. Generați și analizați planul de execuție al cererii anterioare. Cererea este rescrisă?

```
EXPLAIN PLAN
SET STATEMENT_ID ='st_5e_***' FOR comanda_SQL;

SELECT plan_table_output
FROM table(dbms_xplan.display('plan_table',
 'st_5e_***','serial'));
```

f. Observați următorul mesaj care apare ca notă în planul de execuție al cererii anterioare:

```
Note

- dynamic sampling used for this statement (level=2)
```

*Level = 2* implică faptul că unul sau mai multe tabele utilizate de cerere nu are statistici colectate.

g. Colectați statistici pentru vizualizarea *vm\_clienti\_vanzari1\_\*\*\**. Repetați punctul e și verificați dacă în planul de execuție apare mesajul anterior.

h. Indicați optimizatorului să nu utilizeze rescrierea pentru cererea anterioară. Analizați planul de execuție al cererii.

```
EXPLAIN PLAN
SET STATEMENT_ID ='st_5h_***'
FOR
SELECT /*+ NOREWRITE */ ...;

SELECT plan_table_output
FROM table(dbms_xplan.display('plan_table',
 'st_5h_***','serial'));
```

i. Modificați cererea anterioară astfel încât să afișați doar clienții pentru care valoarea totală este mai mare decât 6000000. Generați și analizați planul de execuție al cererii anterioare. Cererea este rescrisă?

j. Folosind tabela de bază *vanzari\_factura\_\*\*\** afișați pentru fiecare client valoarea medie a facturilor (utilizați funcția *AVG*). Generați și analizați planul de execuție al cererii anterioare. Cererea este rescrisă?

k. Modificați cererea anterioară astfel încât formula mediei să fie dată explicit (*suma\_valorilor/numărul\_valorilor*). Generați și analizați planul de execuție al cererii anterioare. Cererea este rescrisă?

l. Definiți vizualizarea materializată *vm\_clienti\_vanzari2\_\*\*\** cu aceeași definiție ca și a vizualizării materializate *vm\_clienti\_vanzari1\_\*\*\**, dar adăugați și coloana *COUNT(\*)*,

respectiv coloana  $COUNT(valoare)$ . Colectați statistici pentru această vizualizare. Repetați punctul *j*.

- m.** Repetați punctul *g*, dar de această dată afișați în loc de codul clientului, adresa de email a acestuia.
- n.** Adăugați constrângerea de cheie primară tablei *clienti\_\*\*\** fără a determina costuri suplimentare (*rely disable novalidate*). Definiți relația dintre tabelele *vanzari\_factura\_\*\*\** și *clienti\_\*\*\** (*rely disable novalidate*). Repetați punctul *m*. Cererea este rescrisă?
- 6. a.** Analizați structura vizualizării *user\_mviews*.
- b.** Afișați informații despre vizualizările definite anterior.