

Vistas

Miguel Rodríguez Penabad

Laboratorio de Bases de Datos
Universidade da Coruña



Cláusula with check option: Exemplo

```
CREATE TABLE T1 (COL1 CHAR(10));
```

```
CREATE VIEW V1  
  AS SELECT COL1 FROM T1 WHERE COL1 LIKE 'A%';
```

```
CREATE VIEW V2  
  AS SELECT COL1 FROM V1 WHERE COL1 LIKE '%Z'  
  WITH LOCAL CHECK OPTION;
```

```
CREATE VIEW V3  
  AS SELECT COL1 FROM V2 WHERE COL1 LIKE 'AB%';
```

```
CREATE VIEW V4  
  AS SELECT COL1 FROM V3 WHERE COL1 LIKE '%YZ'  
  WITH CASCADED CHECK OPTION;
```

```
CREATE VIEW V5  
  AS SELECT COL1 FROM V4 WHERE COL1 LIKE 'ABC%';
```

Cláusula with check option: Exemplo (ii)

Insercións correctas e incorrectas

```
INSERT INTO V1 VALUES ('AAA') -- inserta
INSERT INTO V1 VALUES ('RRR') -- inserta
```

```
INSERT INTO V2 VALUES ('bbZ') -- inserta
INSERT INTO V2 VALUES ('AZ') -- inserta
INSERT INTO V2 VALUES ('AAA') -- falla
```

```
INSERT INTO V3 VALUES ('bbZ') -- inserta
INSERT INTO V3 VALUES ('AZ') -- inserta
INSERT INTO V3 VALUES ('AAA') -- falla
```

```
INSERT INTO V4 VALUES ('ABYZ') -- inserta
INSERT INTO V4 VALUES ('A') -- falla
```

```
INSERT INTO V5 VALUES ('ABYZ') -- inserta
INSERT INTO V5 VALUES ('A') -- falla
```

Vistas Materializadas en Oracle

- ▶ Tamén coñecidas como *snapshots*.
- ▶ Almacenan o resultado da consulta
 - ▶ As consultas son máis rápidas
 - ▶ Ocupan espacio en disco
 - ▶ Hai que actualizar o contido.

```
CREATE MATERIALIZED VIEW <nome>
  <opcións almacenamento>
  <opción creación>
  REFRESH <opcións refresco> <modo refresco>
  [ENABLE|DISABLE] QUERY REWRITE
AS <cláusula select>;
```

<opción creación>

Determina cando se crea a vista materializada.

- ▶ **BUILD IMMEDIATE**: créase cando se define
- ▶ **BUILD DEFERRED**: crearase con posterioridade
- ▶ **ON PREBUILT TABLE**: usa unha táboa para obter os datos da vista.
Úsase especialmente en data warehousing; ten limitacións (reescritura de queries,...)

<opcións refresco>

Determina como se actualiza o contido a vista materializada.

- ▶ **COMPLETE**: Recrea a vista por completo
 - ▶ Pode facerse en calquera momento
 - ▶ Caro (tempo e esforzo computacional)
- ▶ **FAST**: Refresca de forma incremental
 - ▶ Require “materialized view logs” nas táboas base
 - ▶ Máis rápido
- ▶ **FORCE**: Fará un refresco FAST se é posible, se non o é fai un refresco COMPLETE
 - ▶ Opción por defecto

<modo refresco>

Determina cando se actualiza o contido a vista materializada.

- ▶ **ON COMMIT:** A vista refréscase cando hai un commit nas táboas base
 - ▶ Non se permite para todos os tipos de vista
- ▶ **ON DEMAND:** O refresco iníciase de forma manual
 - ▶ Válido para todos os tipos de vistas
 - ▶ `DBMS_MVIEW.REFRESH(<mv_name>, <refresh_option>)`
 - ▶ `DBMS_MVIEW.REFRESH_ALL_MVIEWS()`
- ▶ **START WITH <data> [NEXT] <data>:** Inicia o refresco na data especificada, e indícase o intervalo entre refrescos

Vistas Materializadas en Oracle

Exemplos

```
CREATE MATERIALIZED VIEW LOG ON ITEMS
  TABLESPACE MV_LOGS STORAGE(INITIAL 10M NEXT 10M) WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON CUSTOMERS
  TABLESPACE MV_LOGS STORAGE(INITIAL 1M NEXT 1M) WITH ROWID;

CREATE MATERIALIZED VIEW cust_activity
  BUILD IMMEDIATE
  REFRESH FAST ON COMMIT
AS SELECT u.ROWID cust_rowid, l.ROWID item_rowid,
         u.cust_id, u.custname, u.email,
         l.categ_id, l.site_id, sum(gms), sum(net_rev_fee)
FROM customers u, items l
WHERE u.cust_id = l.seller_id
GROUP BY u.cust_id, u.custname, u.email, l.categ_id, l.site_id;

-- Outra opción
CREATE MATERIALIZED VIEW cust_activity
  BUILD IMMEDIATE
  REFRESH FORCE START WITH SYSDATE NEXT SYSDATE + 1
AS SELECT ...
```