

Inserciones multitables - INSERT INTO .. SELECT

by admin - jueves, octubre 29, 2020

<https://dbandtech.com/inserciones-multitables-insert-into-select/>

Las inserciones de varias tablas se introdujeron en Oracle 9i para permitir que una sola instrucción **INSERT INTO .. SELECT** se inserte de forma condicional o incondicional en varias tablas. Esta declaración reduce los escaneos de tablas y el código **PL/SQL** necesarios para realizar múltiples inserciones condicionales en comparación con las versiones anteriores. Su uso principal es para el **proceso ETL** en almacenes de datos donde se puede paralelizar y/o convertir datos no relacionales en un formato relacional.

La **documentación de Oracle** utiliza el término "**inserciones de varias tablas**", pero otros recursos separan el nombre con guiones, por lo que también lo verá escrito como "**inserciones de varias tablas**".

- Preparar
- **INSERT ALL Incondicional**
- **INSERT ALL Condicional**
- **INSERT FIRST**
- Restriciones

Preparar

Cree y complete una tabla de prueba para que actúe como fuente de datos para los ejemplos básicos, así como tres tablas de destino basadas en la tabla de origen, pero sin filas.

```
CREATE TABLE source_tab AS
SELECT level AS id,
       'Description of ' || level AS description
FROM   dual
CONNECT BY level <= 10;
CREATE TABLE dest_tab1 AS
SELECT * FROM source_tab WHERE 1=2;
```

```
CREATE TABLE dest_tab2 AS
SELECT * FROM source_tab WHERE 1=2;
```

```
CREATE TABLE dest_tab3 AS
SELECT * FROM source_tab WHERE 1=2;
```

```
SELECT * FROM source_tab;
ID DESCRIPTION
-- -----
1  Description of 1
2  Description of 2
```

```
3 Description of 3
4 Description of 4
5 Description of 5
6 Description of 6
7 Description of 7
8 Description of 8
9 Description of 9
10 Description of 10
```

10 rows selected.

SQL>

Cree y complete una tabla de prueba para que actúe como origen del ejemplo dinámico y una tabla de destino vacía.

```
CREATE TABLE pivot_source (
  id          NUMBER,
  mon_val    NUMBER,
  tue_val    NUMBER,
  wed_val    NUMBER,
  thu_val    NUMBER,
  fri_val    NUMBER
);
INSERT INTO pivot_source VALUES (1, 111, 222, 333, 444, 555);
INSERT INTO pivot_source VALUES (2, 111, 222, 333, 444, 555);

CREATE TABLE pivot_dest (
  id  NUMBER,
  day VARCHAR2(3),
  val NUMBER
);
```

Incondicional INSERT ALL

Cuando se usa una instrucción **INSERT ALL** incondicional, cada fila producida por la consulta de conducción da como resultado una nueva fila en cada una de las tablas enumeradas en las cláusulas **INTO**. En el siguiente ejemplo, la consulta de conducción devuelve 10 filas, lo que significa que veremos 30 filas insertadas, 10 en cada tabla.

```
INSERT ALL
  INTO dest_tab1 (id, description) VALUES (id, description)
  INTO dest_tab2 (id, description) VALUES (id, description)
```

```
    INTO dest_tab3 (id, description) VALUES (id, description)
SELECT id, description
FROM   source_tab;
```

30 rows inserted.

SQL>

Se puede usar una instrucción **INSERT ALL incondicional** para pivotar o dividir datos. En el siguiente ejemplo, convertimos cada fila que representa una semana de datos en filas separadas para cada día.

```
INSERT ALL
  INTO pivot_dest (id, day, val) VALUES (id, 'mon', mon_val)
  INTO pivot_dest (id, day, val) VALUES (id, 'tue', tue_val)
  INTO pivot_dest (id, day, val) VALUES (id, 'wed', wed_val)
  INTO pivot_dest (id, day, val) VALUES (id, 'thu', thu_val)
  INTO pivot_dest (id, day, val) VALUES (id, 'fri', fri_val)
SELECT *
FROM   pivot_source;
```

10 rows inserted.

```
SQL> SELECT *
FROM   pivot_dest;
      ID DAY      VAL
-----
      1 mon      111
      2 mon      111
      1 tue      222
      2 tue      222
      1 wed      333
      2 wed      333
      1 thu      444
      2 thu      444
      1 fri      555
      2 fri      555
```

10 rows selected.

SQL>

Condicional INSERT ALL

En una instrucción **INSERT ALL condicional**, se pueden agregar condiciones a las cláusulas **INTO**, lo

que significa que el número total de filas insertadas puede ser menor que el número de filas fuente multiplicado por el número de **cláusulas INTO**. Parece similar a una **expresión CASE**, pero cada condición siempre se prueba en función de la fila actual de la consulta de conducción.

En el siguiente ejemplo, insertamos en diferentes tablas dependiendo del rango del valor de ID.

```
INSERT ALL
  WHEN id <= 3 THEN INTO dest_tab1 (id, description)
VALUES (id, description) WHEN id BETWEEN 4 AND 7 THEN
  INTO dest_tab2 (id, description) VALUES (id, description)
  WHEN id >= 8 THEN
  INTO dest_tab3 (id, description) VALUES (id, description)
SELECT id, description
FROM source_tab;
```

10 rows inserted.

SQL>

Se puede utilizar una sola condición para varias cláusulas INTO.

```
INSERT ALL
  WHEN id <= 3 THEN INTO dest_tab1 (id, description)
VALUES (id, description)
  WHEN id BETWEEN 4 AND 7 THEN
  INTO dest_tab2 (id, description) VALUES (id, description)
  INTO dest_tab3 (id, description) VALUES (id, description)
SELECT id, description FROM source_tab; 11 rows inserted. SQL>
```

Puede usar una condición de "1 = 1" para forzar todas las filas en una tabla.

```
INSERT ALL
  WHEN id <= 3 THEN INTO dest_tab1 (id, description)
VALUES (id, description)
  WHEN id BETWEEN 4 AND 7 THEN
  INTO dest_tab2 (id, description) VALUES (id, description)
  WHEN 1=1 THEN INTO dest_tab3 (id, description)
VALUES (id, description) SELECT id, description
FROM source_tab; 17 rows inserted.
SQL>
```

INSERT FIRST

El uso de **INSERT FIRST** hace que la inserción de varias tablas funcione como una **expresión CASE**, por lo que las condiciones se prueban hasta que se encuentra la primera coincidencia y no se prueban más condiciones. También podemos incluir una **cláusula ELSE** opcional para capturar cualquier fila que no haya sido causada por una condición previa.

```
INSERT FIRST
  WHEN id <= 3 THEN
    INTO dest_tab1 (id, description)
    VALUES (id, description)
  WHEN id <= 5 THEN INTO dest_tab2 (id, description)
VALUES (id, description)
ELSE INTO dest_tab3 (id, description)
VALUES (id, description)
SELECT id, description
FROM source_tab; 10 rows inserted.
```

SQL>

```
INSERT FIRST
  WHEN id <= 3 THEN INTO dest_tab1 (id, description)
VALUES (id, description) ELSE INTO dest_tab2 (id, description)
VALUES (id, description) INTO dest_tab3 (id, description)
VALUES (id, description) SELECT id, description
FROM source_tab; 17 rows inserted.
```

SQL>

Restricciones

Las restricciones sobre las inserciones de varias tablas son las siguientes.

Las inserciones de varias tablas solo se pueden realizar en tablas, no en vistas o vistas materializadas.

- No puede realizar una inserción de varias tablas a través de un enlace DB.
- No puede realizar inserciones de varias tablas en tablas anidadas.
- La suma de todas las columnas **INTO** no puede **exceder 999**.
- Las secuencias no se pueden utilizar en la instrucción de inserción de varias tablas. Se considera una declaración única, por lo que solo se generará y usará un valor de secuencia para todas las filas.
- Las inserciones de varias tablas no se pueden utilizar con la estabilidad del plan.
- Si se usa la sugerencia **PARALLEL** para cualquier tabla de destino, se paralelizará toda la declaración. De lo contrario, la instrucción solo se paralelizará si las tablas tienen **PARALLEL**

definido.

- Las declaraciones de varias tablas no se paralelizarán si alguna de las tablas está organizada en índices o tiene índices de mapa de bits definidos en ellas.

Para más información, ver:

- [INSERT](#)

PDF generated by Kalin's PDF Creation Station