

BASES DE DATOS

DSIC. Curso 1999-2000

Práctica Definición de Datos

(Triggers SQL99)

Objetivos

- Presentar el lenguaje de definición de datos del SQL/92
 - Presentar la sintaxis de creación de tablas
 - Presentar la sintaxis de la modificación de definición de una tabla
 - Presentar la sintaxis de la creación de una vista
 - Presentar la sintaxis para otorgar autorizaciones
 - Presentar la sintaxis para la creación de reglas de actividad (disparadores)
- Todo ello para el sistema Oracle 8.x

Definición de reglas de actividad: disparadores

Define el comportamiento activo del sistema. Aplicaciones:

- comprobación de restricciones de integridad
- control de la seguridad
- definición de reglas de funcionamiento de la organización
- mantenimiento de información derivada

Definición de reglas de actividad: disparadores

Forma de una regla de actividad:

Evento - Condición - Acción

acción que el sistema ejecuta cuando como respuesta a la ocurrencia de un *evento* cuando cierta *condición* se satisface. En Oracle:

- *evento*: operación de actualización
- *condición*: expresión lógica del SQL
- *acción*: procedimiento escrito en PL/SQL (incluye instrucción de manipulación de la BD)

Definición de reglas de actividad

definición_regla::=

```
{CREATE | REPLACE} TRIGGER nombre_regla
  {BEFORE | AFTER | INSTEAD OF} evento [disyunción_eventos]
ON {nombre_relación | nombre_vista}
  [ [REFERENCING OLD AS nombre_referencia
      [NEW AS nombre_referencia] ]
  [FOR EACH {ROW | STATEMENT} [WHEN ( condición ) ] ]
bloque PL/SQL
```

disyunción_eventos ::= OR *evento* [*disyunción_eventos*]

evento ::= INSERT | DELETE | UPDATE [OF comalista_nombre_atributo]

Definición de reglas de actividad

	FOR EACH STATEMENT	FOR EACH ROW
BEFORE	La regla se ejecuta una vez antes de la ejecución de la operación de actualización	La regla se ejecuta una vez antes de actualización de cada tupla afectada por la operación de actualización
AFTER	La regla se ejecuta una vez después de la ejecución de la operación de actualización	La regla se ejecuta una vez después de actualización de cada tupla afectada por la operación de actualización

Definición de reglas de actividad

EVENTOS

{BEFORE | AFTER | INSTEAD OF} *evento* [*disyunción_eventos*]
ON {nombre_relación | nombre_vista}

disyunción_eventos ::= OR *evento* [*disyunción_eventos*]

evento ::=

INSERT | DELETE | UPDATE [OF comalista_nombre_atributo]

Definición de reglas de actividad

EVENTOS

Parametrización de eventos:

- los eventos de las reglas FOR EACH ROW están parametrizados
- parametrización implícita:
 - evento INSERT o DELETE: n (n grado de la relación)
 - evento UPDATE: $2*n$
- nombre de parámetros:
 - evento INSERT: *NEW*
 - evento DELETE: *OLD*
 - evento UPDATE: *OLD* y *NEW*
- se pueden usar en la *condición de la regla*
- se pueden usar en el bloque PL/SQL

Definición de reglas de actividad

CONDICIONES

WHEN (condición)

- expresión lógica de sintaxis similar a la condición de la cláusula WHERE de la instrucción SELECT
- no puede contener subconsultas ni funciones agregadas
- sólo se puede hacer referencia a los parámetros del evento

Definición de reglas de actividad

ACCIONES

bloque PL/SQL

- bloque escrito en el lenguaje de programación de Oracle PL/SQL
- sentencias de manipulación de la BD: INSERT, DELETE, UPDATE, SELECT ... INTO ...
- sentencias de programa: asignación, selección, iteración
- sentencias de manejo de errores
- sentencias de entrada/salida

Definición de reglas de actividad

Lenguaje de reglas:

- Creación: `CREATE TRIGGER nombre_regla ...`
- Eliminación: `DROP TRIGGER nombre_reglas`
- Modificación: `REPLACE TRIGGER nombre_regla ...`
- Recompilación: `ALTER TRIGGER nombre_regla COMPILE`
- Des/Habilitar regla: `ALTER TRIGGER nombre_regla [ENABLE | DISABLE]`
- Des/Habilitar todas las reglas sobre una relación:
`ALTER TABLE nombre_relación [{ENABLE | DISABLE} ALL TRIGGERS]`

Ejemplo: Control de R.I. mediante Triggers

La restricción R2 :

R2) Px: Pieza, Sx: Suministro $\forall Px (\text{Pieza}(Px) \rightarrow \exists Sx (\text{Suministro}(Sx) \wedge Sx.\text{código}=Px.\text{código}))$

se define mediante una restricción general:

```
create assertion R2 check
```

```
not exists(select * from Pieza P
```

```
where not exists(select *
```

```
from Suministro S
```

```
where P.código=S.código));
```

¿Cómo controlar la restricción mediante reglas de actividad?

Ejemplo: Control de R.I. mediante Triggers

Detectar qué eventos pueden afectar a la R.I.:

TABLA,	OPERACIÓN,	ATRIBUTO
Suministro,	Borrado,	-
Suministro,	Modificación,	código
Pieza,	Inserción,	-

Construir Triggers para controlar estos eventos.

Ejemplo: Control de R.I. mediante Triggers

```
CREATE TRIGGER T1
AFTER DELETE ON Suministro OR UPDATE OF codigo ON Suministro
FOR EACH ROW
DECLARE
    N: NUMBER;
BEGIN
    SELECT COUNT(*) INTO N
    FROM Suministro S
    WHERE :old.codigo = S.codigo;
    IF N=0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'No se puede
        borrar el suministro, porque la pieza se quedaría sin suministros.');
```

Ejemplo: Control de R.I. mediante Triggers

```
CREATE TRIGGER T2
AFTER INSERT ON Pieza
FOR EACH ROW
DECLARE  N: NUMBER;
BEGIN
    SELECT COUNT(*) INTO N
        FROM Suministro S WHERE :new.codigo = S.codigo;
    IF N=0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'No se puede
            insertar una pieza, porque la pieza no tiene suministros.
            Cree las dos tuplas (la de pieza y la de suministro)
            dentro de una transacción deshabilitando este trigger.');
```

```
    END IF;
END;
```