Práctica 6 Trigger, Assertions y Constraints

Trigger

Un trigger es un bloque PL/SQL con un nombre asignado en Oracle Database, el mismo se ejecuta automáticamente cuando un evento en específico ocurre. El evento puede ser de estos tipos:

- Lenguaje de manipulación de datos (data manipulation language, DML). Cuando en una tabla se ejecuta INSERT, UPDATE, o DELETE. Por ejemplo, si se define un trigger que se ejecuta antes que INSERT, se ejecutará antes que sea insertada una nueva fila.

- Lenguaje de definición de datos (data definition language, DDL) Cuando se ejecuta CREATE o ALTER. Estos son usados usualmente para auditar cambios en los campos de la tabla.

- Eventos del sistema como inicio o apagado de la base de datos.

- Eventos tales como login o logout.

```
CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER } triggering_event ON table_name
[FOR EACH ROW]
[FOLLOWS | PRECEDES another_trigger]
[ENABLE / DISABLE ]
[WHEN condition]
DECLARE
    declaration statements
BEGIN
    executable statements
EXCEPTION
    exception_handling statements
END;
```

El ejemplo crea un trigger sencillo que se ejecuta luego de realizar un cambio o una inserción en la tabla “votantes”.

```
CREATE OR REPLACE TRIGGER vot
    AFTER UPDATE or INSERT ON votantes
    FOR EACH ROW
BEGIN
     DBMS_OUTPUT.PUT_LINE(:new.nombrecompleto);
END;
```

Assertions

- SQL assertions pueden ser empleadas para crear restricciones teniendo en cuenta información de varias tablas.
- SQL assertion es una restricción CHECK a nivel de base de datos que permite realizar consultas.

En el ejemplo se valida que el número de localizaciones no sea mayor que 16.

```
create assertion limit_localidades as CHECK
((select count(*) from localidades ) <= 16)
```

Constraints

- Las restricciones validan que los valores en una columna cumplan con un criterio.
- Se deben crear durante la clásula CREATE TABLE o luego con ALTER TABLE.

```
--constraint en línea
CREATE TABLE table_name (
    ...
    column_name data_type UNIQUE
    ...
);

--constraint fuera de línea
CREATE TABLE table_name (
    ...,
    UNIQUE(column_name)
);

--asignar un nombre al constraint
CREATE TABLE table_name (
    ...
    column_name data_type CONSTRAINT unique_constraint_name UNIQUE
    ...
);
```

```
--asignar un nombre al constraint fuera de línea
CREATE TABLE table_name (
    ...
    column_name data_type,
    ...,
    CONSTRAINT unique_constraint_name UNIQUE(column_name)
);

--constraint para un grupo de columnas, para este caso la combinación
de valores de ambas columnas será único
CREATE TABLE table_name (
    ...
    column_name1 data_type,
    column_name2 data_type,
    ...,
    CONSTRAINT unique_constraint_name UNIQUE(column_name1, column_nam
e2)
);
```

```
--agregar un constraint a una tabla existente
ALTER TABLE table_name
ADD CONSTRAINT unique_constraint_name UNIQUE(column_name1, column_na
m2);

--habilitar o deshabilitar constraint
ALTER TABLE table_name
DISABLE|ENABLE CONSTRAINT unique_constraint_name;

--eliminar constraint
ALTER TABLE table_name
DROP CONSTRAINT unique_constraint_name;

--Oracle Check constraint
Un constraint de tipo check permite comprobar la integridad del domi
nio de los datos

CREATE TABLE table_name (
    ...
    column_name data_type CHECK (expression),
    ...
);
```

```
CREATE TABLE table_name (
    ...,
    CONSTRAINT check_constraint_name CHECK (expresssion)
);

CREATE TABLE parts (
    part_id NUMBER GENERATED BY DEFAULT AS IDENTITY,
    part_name VARCHAR2(255) NOT NULL,
    buy_price NUMBER(9,2) CHECK(buy_price > 0),
    PRIMARY KEY(part_id)
);

--Eliminar un check constraint
ALTER TABLE table_name
DROP CONSTRAINT check_constraint_name;

--Deshabilitar / Habilitar check constraint
ALTER TABLE table_name
DISABLE|ENABLE CONSTRAINT check_constraint_name;
```