

Práctica 6: Triggers y constraints

Universidad de Córdoba

Diciembre 2021

Cree una tabla para almacenar el registro de ejecuciones de los triggers durante la práctica. Deberá tener dos columnas, “datos” y “tabla”, en el campo datos almacenará detalles del cambio y el campo tabla almacenará el nombre de la tabla donde ocurrió el cambio.

Para ello, recordemos, usar: *create table audit_table(datos varchar2(100 BYTE), tabla varchar2(100 BYTE))*

1. Cree un trigger que guarde los cambios que se produzcan en la columna ‘nombre’ de la tabla ‘eventos’. Deberá almacenar: ‘NombreAnterior NuevoNombre’

A modo de ejemplo, al modificar el nombre del evento con ID10:

10	10Madrid	22/01/18	Locales	Elecciones al ayuntamiento de Madrid
----	----------	----------	---------	--------------------------------------

Con el siguiente valor:

10	10Madrid 2020	22/01/18	Locales	Elecciones al ayuntamiento de Madrid
----	---------------	----------	---------	--------------------------------------

En la tabla de *audit_table* quedaría:

	DATOS	TABLA
1	Madrid Madrid 2020	eventos

2. Implemente un trigger que valide el número de habitantes de las localidades. Si se intenta modificar dicho valor, el número de habitantes nunca podrá ser mayor que 4000000 y no podrá ser menor que 1. En caso de ocurrir lo anterior deberá modificar el valor que se intentó insertar, y en su lugar deberá mantener el valor de la columna.

Por ejemplo, al intentar modificar con el valor 0:

16	16 Valencia	786189	6
*16	16 Valencia	0	6

Se obtendría el mismo valor, pues el trigger ejecuta la validación y modifica el valor que finalmente es guardado:

16	16 Valencia	786189	6
----	-------------	--------	---

3. Implemente una funcionalidad que cuando se realicen nuevas consultas (o se modifiquen) a los votantes, valide que las nuevas consultas tengan una fecha válida (menor o igual que la fecha actual). Además, registre dicho cambio en la tabla de auditoría.

A modo de ejemplo, si intentamos cambiar la fecha a '11-07-2025' en el caso de la consulta con ID 2, observaremos como la fecha que queda en la tabla es la del día actual (en el momento de realizar el ejercicio, el 24 de noviembre de 2021). Además, en la tabla de auditoría el cambio queda registrado.

	IDCONSULTA	EVENTO	VOTANTE	FECHA
1	1	1	30983712	22/03/18
2	2	1	41867538	24/11/21

DATOS	TABLA
1 El votante con dni 30983712 cambia su fecha a 23/02/92	votantes
2 hubo cambios de en la fecha de 22/03/18 a 24/11/21	consultas

4. Elabore un procedimiento que no permita que se interten números de teléfono inválidos de los votantes. El rango permitido es [600000000, 799999999].

+18	50000000	Alberto Guerrero	Ninguno	Parado	alberto@gmail.com	1	28/01/00	800000000
-----	----------	------------------	---------	--------	-------------------	---	----------	-----------

```
Se ha producido un error al guardar los cambios en la tabla "Z72PEPEE"."VOTANTES":
Fila 18: ORA-02290: restricción de control (Z72PEPEE.CHECK_COMP) violada
ORA-06512: en línea 1
```

5. Restrinja que el tipo de los eventos deban comenzar con letra inicial mayúscula y debe terminar en 's'. De no cumplirse, deberá impedir que se inserte en la tabla.

Primero hay que deshabilitar CK_NOMBRE durante este ejercicio.

Ejemplo: si se intentar insertar el registro

+12	12	Europa2019	01/01/19	Otra	Reeleccion
-----	----	------------	----------	------	------------

Deberá lanzar el siguiente mensaje:

```
INSERT INTO "██████"."EVENTOS" (IDEVENTO, NOMBRE, FECHA, TIPO, DETALLE) VALUES ('12', 'Europa2019', TO_DATE('2019-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Otra', 'Reeleccion')
ORA-02290: restricción de control (██████.CHECK_UPPER) violada
ORA-06512: en línea 1
```

6. Haga una funcionalidad que permita controlar la situación laboral y/o la fecha de nacimiento de los votantes. Si existe un votante que tenga más de 59 años y aún no está jubilado, deberá quedar registrado su DNI en la tabla de control de ejecuciones, así como de cuál tabla proviene la información.

Ejemplo de registro insertado en la tabla de control:

1	votante con DNI 3333333 tiene mas de 59 y no esta jubilado	votantes
---	--	----------

7. Implemente una funcionalidad que no permita que se inserten direcciones de correo inválidas en la tabla VOTANTES. Para que sea una dirección válida debe comenzar con una secuencia de caracteres (longitud > 0), seguido del carácter '@', seguido de una secuencia de caracteres (longitud > 0), luego '.', y finalmente una secuencia de caracteres (longitud > 1). Puede asumir que en el caso del carácter '.' solo aparecerá como máximo en 1 ocasión.

Ejemplos de direcciones inválidas: @gmail.com, juangmail.com, juan@.com, juan@gmailcom, juan@gmail.

8. Elabore un procedimiento para controlar el tipo de correo asociado a los votantes. Para ser considerados válidos, los correos tendrán algunas particularidades. La primera está relacionada con el alias (antes del '@'). En esta parte el correo debe contar exactamente con dos letras 'a' (no puede haber menos ni más). Además, tras el '@' y antes del '.' se debe contar con una de las siguientes extensiones: "uco", "gmail", "hotmail". Cualquiera que no sea una de esas extensiones hará que el correo se considere inválido. Para finalizar, tras el '.' debe existir al menos una letra más.

Algunos ejemplos de correos inválidos son:

- araca@gmail.com: contiene 3 veces la letra 'a' en el alias.
- eduardo@gmail.com: contiene 1 vez la letra 'a' en el alias.
- alba@yahoo.com: "yahoo" es una extensión no válida.
- alba@uco.: No hay nada tras '.'.

Ejemplo de correo válido: alba@uco.es

Nota: Cuando se añada el constraint, para que los correos ya introducidos en la tabla que no cumplan esta restricción no den problemas, hay que añadir al final de la línea 'ENABLE NOVALIDATE'.