



ETSI Informática (Gestión)

Administración de B.D.

Relación de Problemas 5

Gestión de una BD y Triggers

Cree una BD con 5 tablas para el estudio de la **DESIGUALDAD SOCIAL** confines altruistas, teniendo en cuenta que debe establecer una llave primaria para cada tabla y establecer las restricciones de integridad referencial que considere convenientes, así como otras restricciones interesantes (que los números sean positivos donde sea pertinente, etc.):

1. Cree un usuario llamado **GESTOR**. Dicho usuario será el propietario y el administrador de esa BD, lo cual no quiere decir que tenga que tener permisos de **DBA**. O sea, el usuario **GESTOR** administra esa BD pero no puede efectuar tareas de administración de Oracle (matarse sesiones...). Endic haBD se desea almacenar la siguiente información:

- a) Sobre **países**: Un código único para cada país, nombre, capital y presupuesto para fines sociales en miles de euros.
- b) Sobre **regiones**: Se guardan datos como el país en el que están situadas (su código), su nombre (que puede repetirse si pertenecen a países distintos), su extensión (en Km²), su coeficiente de Gini (de tipo NUMBER(5, 2)), su población (en miles de habitantes, NUMBER(10, 3)) y el umbral de pobreza para esa región (en euros/año, NUMBER(9, 1)).
- c) Sobre el **censo**: En cada región se tienen datos sobre algunas de las personas que viven en ella, como nombre, sexo (V o H), país de su nacionalidad, región donde vive (donde está censado), ingresos (en euros/año) y un código (similar al DNI) para cada persona, que no puede repetirse dentro del mismo país en el que está nacionalizado (que no tiene que ser el mismo que el país en el que vive).
NOTA 1: Supondremos que en esta tabla se tiene información de una encuesta y no todos los datos de todas las personas, por lo que el número de personas censadas en una región no tiene que coincidir con el atributo "población" de la tabla regiones.
NOTA 2: No confunda el país de su nacionalidad con el país en donde está censado. Supondremos que sólo se puede estar nacionalizado en un país, y que el país en el que uno está censado es el país en el que vive. Observe que interesa saber no sólo el país en el que se vive sino también la región en la que se vive dentro de dicho país.
- d) También se gestionan diversas **actividades** (sanidad, escolarización, control de la natalidad...) con datos como el costo por persona al año (en euros). Cada actividad tendrá un código numérico único y un nombre irrepetible.
- e) Además, cada **región** puede **adoptar** un número indeterminado de esas actividades indicando su fecha de inicio y duración de la actividad.

2. Inserte información en TODAS las tablas abarcando todos los diversos tipos de información que incluya la tabla en cuestión (hombres y mujeres en el censo...).

3. Se necesitan crear distintas vistas con la información que se indica:

a) Crear una **vista** con las características de aquellas actividades que han sido adoptadas en regiones para las que o bien su país no tiene presupuesto social para llevarlas a cabo globalmente (según la población de esa región y el costo por persona de tal actividad), o bien la densidad de población en dichas regiones está entre 1000 y 5000 hab./Km².

b) Crear una **vista** con 4 atributos: Nombre del país, Capital, Concepto y Cantidad. El concepto se refiere al significado del atributo Cantidad. Se incluirá la información de cada concepto para cada país existente. Los conceptos son los siguientes seis:

- Personas censadas en el país.
- Extensión total de las regiones del país.
- Regiones contabilizadas del país.
- Menor umbral de pobreza del país.
- Población del país.
- Actividades adoptadas por el país.

País	Capital	Concepto	Cantidad
Bolivia	La Paz	Personas censadas	74324
Bolivia	La Paz	Número de Regiones	8
...
Honduras	Tegucigalpa	Personas censadas	38654
Honduras	Tegucigalpa	Número de Regiones	3

- 4. Como DBA otorgue permisos al usuario **GESTOR** para que éste pueda crear usuarios. Pruébelo creando un usuario llamado **AKSEN**, y otórguele permisos para consultar las dos vistas anteriores. ¿Puede **AKSEN** consultar dichas vistas sin tener permiso para consultar las tablas sobre las que se construyeron dichas vistas? _____
- 5. Otorgue a **AKSEN** todos los permisos necesarios para consultar, borrar o insertar todas las tablas, excepto en la de **Actividades**, en la que sólo podrá consultar. Compruébelo.
- 6. Programe, en un fichero de texto independiente, un disparador o *trigger* a nivel de sentencia llamado **ACTIV_SIN_5REGIONES**, que actúe ANTES de INSERTAR una nueva **actividad**. El disparador no permitirá que una nueva actividad sea insertada si no existen al menos 5 regiones en la tabla de **Regiones**. En caso contrario, la actividad se insertará sin problemas. Utilice **RAISE_APPLICATION_ERROR** para generar el error.
- 7. ¿Puede **AKSEN** crear ese disparador a pesar de que no tiene permisos para insertar en dicha tabla? _____. ¿Y si se le concede permiso para insertar? _____. ¿Y el administrador **SYSTEM**? _____. ¿Qué permiso debe dar el DBA para que **AKSEN** pueda crear dicho *trigger*? _____. Finalmente, instale el disparador como **GESTOR** y compruebe su funcionamiento cuando **GESTOR** intente insertar una nueva actividad.
- 8. Conceda el permiso necesario para que el usuario **AKSEN** pueda ejecutar el disparador anterior. ¿Qué ocurre? _____
- 9. Asegúrese de que **AKSEN** no tenga permiso para insertar en la tabla **Actividades**. ¿Se disparará ese *trigger* cuando el usuario **AKSEN** intente insertar algo? _____. Observe que **AKSEN** no tiene permiso para insertar, pero el disparador es de tipo **BEFORE**.
- 10. Otorgue permisos para insertar en la tabla **Actividades** al usuario **AKSEN**. Compruebe si el disparador anterior se ejecuta cuando **AKSEN** inserta en esa tabla. ¿Es necesario otorgar permisos de ejecución sobre el disparador? _____.

11. El usuario **GESTOR** crea una tabla llamada **CTRL_ACTIVIDADES** con los siguientes atributos: **USUARIO**, **FECHA** y **OPERACION**. Modifique el disparador anterior para que se inserte una fila en esta nueva tabla, donde **USUARIO** es el usuario que dispara su ejecución, **FECHA** es la fecha actual y **OPERACION** tomará el valor '**INSERT**' si la operación de inserción no es abortada por el disparador. En caso contrario, el atributo **OPERACION** tomará el valor siguiente: '**INTENTO FALLIDO**'. ¿Funciona correctamente cuando el intento es fallido? _____.
12. Observe que el usuario **AKSEN** no tiene ningún tipo de permiso sobre la tabla **CTRL_ACTIVIDADES**. A pesar de ello, cuando **AKSEN** intenta insertar una nueva actividad, ¿se insertan valores en la tabla **CTRL_ACTIVIDADES**? _____. ¿Puede **AKSEN** consultar esa tabla? _____.
13. Suponga que **GESTOR** quiere insertar una nueva actividad pero el disparador anterior no se lo permite (establezca las condiciones para ello). Para conseguirlo haga que **GESTOR** desactive el disparador, con la orden **ALTER TRIGGER**. Mientras el disparador está desactivado, ¿puede el usuario **AKSEN** insertar filas en la tabla de Actividades, sin que se ejecute el disparador? _____. Tras insertar la nueva actividad vuelva a activar el disparador y compruebe que realmente está activado para todos los usuarios.
14. Haga que el usuario **GESTOR** examine la información sobre ese disparador que existe en la vista **USER_TRIGGERS** del Diccionario de Datos de Oracle (utilice sólo las columnas **TRIGGER_NAME**, **STATUS**, **TRIGGER_TYPE**, **TRIGGERING_EVENT**, **TABLE_OWNER** y **TABLE_NAME**). ¿Puede el usuario **AKSEN** acceder a esa información? _____.
15. Haga que el usuario **AKSEN** consulte la información que tiene disponible en el Diccionario de Datos sobre la tabla de **Actividades**. Como dicha tabla no le pertenece, dicha tabla no aparece en la vista **USER_TABLES**. ¿Aparece esa información en otra vista llamada **ALL_TABLES**? _____. ¿Puede el usuario **AKSEN** consultar las restricciones que operan sobre la tabla de **Actividades**? _____. En caso afirmativo ¿cuántas restricciones existen sobre esa tabla y observe el tipo de cada una.
16. Asegúrese de que existen menos de 5 regiones, para hacer que el disparador genere un error. Haga que el usuario **AKSEN** intente insertar una nueva actividad que incumpla alguna de las restricciones de integridad de la tabla de **Actividades**. Lógicamente, la actividad no podrá ser insertada pero, ¿cuál es el mensaje de error recibido por el usuario, el que genera el disparador, el que genera Oracle por incumplir una restricción, o ambos? _____. ¿Qué información se ha insertado en la tabla **CTRL_ACTIVIDADES**?
17. Inserte nuevas regiones hasta conseguir un mínimo de 5. Tras esto vuelva a probar la inserción del ejercicio anterior. ¿cuál es el mensaje de error ahora? ¿Qué información se ha insertado ahora en la tabla **CTRL_ACTIVIDADES**?
18. Convierta el disparador en un disparador de tipo **AFTER** y conteste a la siguiente pregunta. Si al ejecutar una orden SQL sobre una tabla, un disparador **AFTER** genera un error con **RAISE_APPLICATION_ERROR**, ¿se ejecuta la orden SQL? ¿o se deshace esa orden a pesar de que el disparador se ejecuta DESPUÉS de la orden? _____.
19. Como usuario **GESTOR**, cree una tabla **CTRL_PAISES** con los mismos atributos que la tabla **CTRL_ACTIVIDADES**, y programe un disparador a **nivel de orden** de tipo **BEFORE** sobre la tabla de **Países**, que se active cuando se **borre** sobre esa tabla. El disparador insertará en dicha tabla los valores correctos de usuario y fecha y en el atributo **OPERACION** tomará el valor '**BEFORE DELETE**'. Compruebe que el disparador también funciona para el usuario **AKSEN**.
20. De manera similar al ejercicio anterior, cree un disparador a **nivel de orden** de tipo **AFTER** sobre la tabla de **Países**, que al **borrar** sobre esa tabla inserte en la tabla **CTRL_PAISES** los valores correspondientes de usuario y fecha, y en el atributo **OPERACION** insertará el valor '**AFTER DELETE**'.
21. Programe ahora un disparador a **nivel de fila** de tipo **BEFORE** sobre la tabla de **Países**, que al **borrar** cada país inserte en la tabla **CTRL_PAISES** los valores correspondientes de usuario y fecha, y en el atributo **OPERACION** insertará el valor '**BEFORE DELETE fila: PPPP con capital en CCCC**'. Los valores de **PPPP** y **CCCC** serán los valores del país y su capital correspondientes a la fila afectada. Para comprobar su funcionamiento borre un país y posteriormente examine la tabla **CTRL_PAISES**.
22. De manera similar al ejercicio anterior, cree un disparador a **nivel de fila** de tipo **AFTER** sobre la tabla de **Países**, que tras **borrar** un país inserte en la tabla **CTRL_PAISES** los valores correspondientes de usuario y fecha, y en el atributo **OPERACION** insertará el valor '**AFTER DELETE fila: PPPP con capital en CCCC**'.
23. Inserte en la tabla de países todos aquellos que empiecen por la letra '**U**': **Ucrania** (con capital en Kiev), **Uganda** (con capital en Kampala), **Uruguay** (con capital en Montevideo) y **Uzbekistán** (con capital en Tashkent). Localice en un mapa la situación de estos países si lo desconoce, y compruebe si se ha disparado algún *trigger* al insertar dicha información.
24. Borre con una sola sentencia **DELETE** todos los países que empiezan por la letra '**U**'. Compruebe que en la tabla **CTRL_PAISES** se han insertado 10 filas y examine cuántas veces se ha ejecutado cada uno de los 4 disparadores que existen sobre esa tabla.
25. Se desea asegurar que los países de la zona Euro no tengan un presupuesto para fines sociales (tabla de **Países**) inferior a un millón de Euros. Para ello, al insertar estos países o al actualizar su presupuesto deberá comprobar si el presupuesto que se pretende dejar en la tabla es menor a esa cantidad. En caso afirmativo debe poner el presupuesto mínimo e insertar en la tabla **CTRL_PAISES** un mensaje aclaratorio que indique el país y la cantidad que se le ha añadido a su presupuesto para llegar al mínimo.
- Los 12 países de la zona Euro son: España, Portugal, Italia, Grecia, Austria, Alemania, Finlandia, Irlanda, Bélgica, Holanda, Luxemburgo y Francia.
26. Modifique el disparador anterior para que al final haga un **COMMIT**. ¿Qué ocurre?
27. Borre las tablas prestando especial atención al orden en el que deben borrarse.