



AMPLIACION DE BASE DE DATOS

Examen Final Convocatoria Extraordinaria 2021-22

EJERCICIO 1 (4 puntos)

Hemos almacenado todo tipo de artículos en una colección llamada `Noticias`, dentro de una base de datos de MongoDB llamada `examenABD`, cuyos documentos (uno por cada artículo sobre alguna noticia) tienen los siguientes campos:

- *TipoArti*: EJ: política, elTiempo, deporte, programaTV, etc.
- *Agencia*: quien publica ese TipoArti. Ej: Reuters, Efe, etc.
- *Título*: Titular del artículo concreto. EJ: eleccionesXX, elTiempoHoy, partidoRealMadrid-Barça, etc.
- *Valoración*: de calidad del artículo (entre 0 y 10)
- *TotalMeGusta*: cuántos MeGusta lleva ese artículo.
- *Detalles*: array de pares (nombre, valor) de características particulares del artículo.

SE PIDE: Escribe lo pedido en cada apartado.

- a) Asume, para este apartado solo, que *Detalles* contiene: (autores, [nombre1,nombre2,ect.]). Sucede que, como los autores son pocos, se repiten mucho en la BD.
- A.1.- Describe qué podemos hacer para tener los datos de un autor solo una vez en la BD sin perder información. Y que se pueda acceder a los artículos de un autor y a los autores de un artículo.
- A.2.- Describe cada paso, en el orden adecuado, necesario para obtener lo que indicas en E.1, partiendo de los datos de la BD original: escribe el pseudocódigo y el *nombre* de la instrucción MongoDB para hacerlo (solo el nombre, no escribas la instrucción completa).
- b) Este apartado es sobre la colección original `Noticias`. Usando **aggregate**, obtener *para cada* Tipo de artículo y su Agencia, los siguientes campos y valores: TipoArti, Agencia, el total de todos los MeGusta de sus artículos (solo de los artículos valorados entre 5 y 8) y cuántos artículos hay que cumplen esa condición. Otra condición es que queremos esos valores solo para los Tipos de artículo que tengan más de 20 artículos. El resultado estará ordenado de modo ascendente por el Tipo de artículo y su Agencia.
- c) Este apartado es sobre la colección original `Noticias`. Usa solo una instrucción **update** (sin funciones que recorran la colección). Queremos actualizar el primer documento que cumpla las condiciones siguientes: *TipoArti*: "deporte", *Agencia*: "Efe", *Título*: "Perdió el Barça" y dentro de los *Detalles* debe estar el par : <puntuación , "5 a 0" >. Además, no queremos que otro proceso pueda escribir en la colección hasta terminar esta operación. Los cambios a realizar cuando se cumplen las condiciones son: *Valoración* = 3, *TotalMeGusta* = 0, y (nuevo) *Aceptado* = sí. Si no existe ningún documento que cumpla las condiciones, debe crear un documento con los valores indicados tanto en las condiciones como en los *cambios*.

EJERCICIO 2 (4 puntos)

Dado el siguiente esquema relacional (las tablas están creadas en Oracle y con datos) para pagar por cada *meGusta*

```
Agencia(Nombre, Estilo,Dirección)
Noticia_XXXXX(TipoArti, Título, Valoración, TotalMeGusta)
MalasNoticias(Agencia, TipoArti, Título, Valoración, TotalMeGusta)
Pago(TipoArti, Título, numBono, NomAgencia, Importe)
```

→ Hay una tabla de `Noticias` por cada `Agencia`: en el nombre de la tabla `Noticia_XXXX`, `XXXX` es el nombre de la agencia. Se asume que hay un número desconocido de agencias (y tablas).

SE PIDE: Hacer dos procedimientos en PL/SQL (debes usar dinámico solo donde corresponda) y un trigger:



- a) Escribe el procedimiento **EJ(NomAgencia)** que incluye los siguientes tratamientos y características:
- Vamos a tratar solo la agencia que se pasa por parámetro, con sus noticias.
 - El tratamiento de cada noticia se hace en una transacción distinta llamada **TranNoticia** (y que vea los cambios de otras trans. cuando terminen).
 - Para cada noticia de esa agencia hacemos esto: por cada MeGusta (el total se indica en el atributo **TotalMeGusta**), escribimos un Pago (un bono) con **importe** de 10 euros. El **numBono** es un número secuencial único para cada bono (se asume que ya está creada una secuencia *ej-sec*, úsala). El resto de atributos son los de la noticia.
 - Si una noticia tenía **TotalMeGusta** menor de 5, llamamos a otro procedimiento **procMalas** (descrito debajo)
 - Al terminar de crear los bonos para una noticia, actualizamos a cero el **TotalMeGusta** de esa noticia.
 - Si en este momento llevamos generados bonos que en total desde el principio del programa sumen más de 10.000 euros, queremos deshacer los bonos de esta última noticia y terminar de crear bonos.
 - Si hay cualquier error inesperado, debe dar un mensaje de aviso: "**error inesperado**"
- b) Escribe el Procedimiento **procMalas (Agencia, TipoArti, Título, Valoración, TotalMeGusta)** que inserte la noticia, usando los parámetros dados, en la tabla **MalasNoticias**. Esta fila debe quedarse creada incluso si deshacemos los bonos para esa noticia, e incluso si aborta la transacción desde donde se llama a este procedimiento.
- c) Escribe un trigger **TRborra** para que, después de cada inserción en la tabla **MalasNoticias**, se borre la fila en la tabla **Noticia_XXXX** de la agencia XXXX correspondiente. Este trigger se debe ejecutar después de otro llamado **TRantes** (que asumimos ya creado).
- d) Que podría suceder de particular por existir el trigger **TRborra** al ejecutar todo el proceso?. Razona tu respuesta.

EJERCICIO 3 (2 puntos)

- a) En Oracle, tenemos estas dos relaciones **Alumno (DNI, Nombre)** y **Matriculado (DNI, codAsignat, NumCred)** con sus PKs subrayadas. Como accedemos frecuentemente al total de los créditos en los que está matriculado un alumno, queremos facilitar el acceso a ese total. **a.1)** Para ello describe la desnormalización necesaria y como queda el resultado. **a.2)** Indica qué problema causa. **a.3)** Indica cómo solucionarlo conservando la desnormalización (solo descríbelo, no lo realices).
- b) Dada la relación **R(A,B,C,D,E,F)** y las siguientes DFs: { $A \rightarrow B$, $B \rightarrow A$, $C \rightarrow A$, $E \rightarrow F$, $E \rightarrow D$, $EC \rightarrow B$ } **se pide:**
- b.1)** Rellena las cuatro filas con los valores de B para los dos casos: que cumplan / no cumplan la DF : $EC \rightarrow B$

E	C	Valor de B caso: NO cumple $EC \rightarrow B$	Valor de B caso: SI cumple $EC \rightarrow B$
b1	c1		
b2	c3		
b1	c1		
b2	c3		

b.2) Encuentra la clave primaria con todas las DFs . Incluye la aplicación del algoritmo adecuado. Explica los pasos.

b.3) Aplicando las definiciones de FNs indica cual es la mejor FN en la que está la relación **REL(X1,X2,X3,X4,X5)** con $DFs = \{X1 \ X2 \rightarrow X3 ; X5 \rightarrow X4 ; X1 \ X2 \rightarrow X5 \}$. Además, indica qué condición cumple para estar en dicha FN y cual no cumple para estar en la siguiente mejor FN. Describe los conceptos que uses.