

Anexo IV: Data Warehouse

El desarrollo de las funcionalidades estadísticas se basa en el conjunto de resúmenes de datos solicitados por el cliente, descritos como parte del requisito funcional RF07; y la definición, ya refinada, propuesta como tercer esquema externo en el apartado “3.1.3 Esquema *data warehouse*” de la Memoria del Proyecto.

Dentro del ámbito del *data warehouse* cabe señalar que las funcionalidades que se han desarrollado no están orientadas al control de rendimiento ni a la monitorización de procesos, sino al registro de eventos puntuales con objeto de gestionar tablas de resumen. Estas tablas de hecho son las conocidas como tablas de cobertura, donde lo importante es el evento más que su evolución.

Para la construcción se plantea el uso de un último *tablespace*; en principio, más por estructuración general y eficiencia que por cuestión de seguridad. En esta ocasión se entrega un único *script* SQL que contiene estos componentes virtuales y las tablas, luego los procedimientos almacenados y los disparadores por separado. Respecto a los preparativos, al igual que para el resto de *tablespaces*, se accede al contenedor y desde el se crea el directorio que se le asociará mediante SQL. Se ejecutan las siguientes instrucciones, y se incluye el *prompt* por diferenciar el entorno de ejecución:

```
[root@localhost uoc]# docker exec -it postgresql-tfg-uoc /bin/bash
root@85c9fd8b0723:/# mkdir -p /var/lib/postgresql/tablespace/tb_dwh
root@85c9fd8b0723:/# chown postgres:postgres /var/lib/postgresql/tablespace/tb_dwh
```

Este anexo se estructura a partir de las distintas funcionalidades que se han propuesto, para cada funcionalidad se detallan los resultados esperados y las implicaciones semánticas de los mismos, luego la implementación y, por último, el esquema externo concreto.

Luego, en lo referente a la composición de los nombres, se continúa en la misma línea que hasta ahora. En este caso se utiliza para todos los casos, tablas y estructuras de control, el prefijo 'dwh_'; tras este, según si es disparador o procedimiento almacenado, se utiliza 'tr_' o 'sp_' y nada si es tabla, simplemente el nombre de la misma. Tras estos prefijos se utiliza el nombre de la funcionalidad de *data warehouse* de que se trate, el nombre de la tabla donde está el *trigger* y, finalmente, el evento que activo al anterior. Además, como en el diseño físico de la BBDD se han generado funciones externas que se identifican mediante dos guiones bajos antes de todo el patrón ya descrito.

Por otro lado, para poder controlar el comportamiento de estas funcionalidades se ha definido una tabla que contendrá registros por cada error que generen las funciones que se describen a lo largo de este anexo. Esta tabla recibirá registros la la ejecución provoca la entrada en el bloque de gestión de excepciones. Hasta aquí, lo que es la creación de las tablas, más las propuestas como vistas finales, se entregan en el *script* **07_DWH_tablas_vistas.sql**

Una última cuestión, relativa a la implementación, procede describirse en este punto. Como forma de estructuración del código, en las tablas en que se debía crear más de un *trigger* se crea uno solo, luego la función asociada a este llama a las correspondientes a dicha tabla de forma ordenada. Concretamente se crean los siguientes *triggers* que utilizan las funciones indicadas, todos estos se entregan en el *script* **07_DWH_triggers.sql**.

- dwh_tr_agrupacionanio_in
 - dwh_sp_agrupacionanio_in

Esta función llama a las siguientes, ya contenidas en el *script* dedicado, con los valores que las mismas necesitan únicamente:

 - dwh_sp_funcionesentidad_agrupacionanio_in
 - dwh_sp_temasrecurrentes_agrupacionanio_in
 - dwh_sp_soportedocumental_agrupacionanio_in
- dwh_tr_agrupacionanio_up
 - dwh_sp_agrupacionanio_up

De la misma forma que en el caso anterior, esta función llama a las siguientes:

 - dwh_sp_premiosconcurso_agrupacionanio_up
 - dwh_sp_funcionesentidad_agrupacionanio_up
 - dwh_sp_temasrecurrentes_agrupacionanio_up
 - dwh_sp_soportedocumental_agrupacionanio_up

- dwh_tr_agrupacionanio_del
 - dwh_sp_agrupacionanio_del
Similar a los anteriores, se llama a las siguientes solo con los valores necesarios:
 - dwh_sp_soportedocumental_agrupacionanio_del
- dwh_tr_letra_del
 - dwh_sp_letra_del
 - dwh_sp_funcionesentidad_letra_del
 - dwh_sp_temasrecurrentes_letra_del
- dwh_tr_disenio_del
 - dwh_sp_disenio_del
 - dwh_sp_soportedocumental_disenio_del
 - dwh_sp_temasrecurrentes_disenio_del

En estos dos últimos casos, de letras y diseños, hubo que establecer el momento de activación del disparador a *before delete* porque, por otras decisiones de diseño, se eliminaba toda relación con los temas, evitando así la posibilidad de decrementar los contadores.

Agrupaciones ganadoras por concurso y año

La idea básica es poder generar un cuadro donde por cada concurso y año se tenga la lista de las agrupaciones premiadas por cada modalidad. Del concurso se debe indicar el nombre y la localidad; en cuanto a los premios, se deben gestionar los casos en que quedan desiertos; y, respecto a las agrupaciones se debe incluir el autor, para lo cual se calculará cual es el principal por mayoría y solo respecto de las composiciones literarias.

Hay que considerar que, en la actualidad, cada año participan alrededor de 150 agrupaciones en el COAC; por tanto, generar estos datos dentro del concepto de *data warehouse* es más que pertinente, teniendo en cuenta, además de la participación, la necesidad de acceder a las relaciones AgrupacionAnio, participa, Concurso y Premio; además de localizar la autoría predominante a partir del conjunto de Letras interpretadas por cada agrupación.

DWH_PremiosConcurso (anio, concurso, localidad, categoría, modalidad, agrupacionAnio,
autorLetra, autorMusica, localidad, premio)

En cuanto a desnormalización, se incluye la localidad del concurso por formar parte de la clave candidata de los mismos, más la de las agrupaciones por interés analítico. No se plantea extraer la redundancia que se generará en el par concurso-localidad porque como máximo se darán unas 50 repeticiones; esto último, derivado de la combinación entre categorías, modalidades y el total de premios del concurso en concreto.

Implementación

Con base en los disparadores necesarios para la actualización de esta tabla, se comenta la implementación de las funciones que cada uno utiliza para realizar los cálculos; estas, se entregan en el *script* **06_DWH_premiosconcurso.sql**.

- dwh_tr_premiosconcurso_evento_up
 - dwh_sp_premiosconcurso_evento_up
Esta función, puesto que ya no procede tratarlo como procedimiento almacenado, únicamente actualiza los valores concurso y localidad en caso de que alguno de estos cambie en la tabla Evento.
- dwh_tr_premiosconcurso_participa_in
 - dwh_sp_premiosconcurso_participa_in
Asociado a la tabla participa, llamará al procedimiento encargado de poblar esta relación. Si el valor de clasificacion es menor que 5 procederá a llamar a la función interna que se detalla al final de este apartado, a esta les indica los identificadores agrupacion_anio y concurso, más el valor de clasificacion.

- dwh tr premiosconcurso participa up

- dwh sp premiosconcurso participa up

En esta se evalúan varios escenarios. En primer lugar, tras verificar que el valor de clasificacion era nulo, o mayor que tres y pasa a ser menor que 4, se procede a llamar a la función que inserta en la tabla de hechos. En segundo lugar, si el premio pasa de ser menor que 4 a mayor que 3, se elimina el registro para esa agrupación en ese concurso. Por último, si la clasificación de la agrupación solamente varía entre el primero y el tercero, se actualiza la tabla de hechos directamente.

- dwh tr premiosconcurso participa del

Este es uno de los casos en que, solamente por ser el *trigger* relativo al evento *on delete*, se plantea que la función asociada se ejecute antes de la eliminación del registro en participa.

- dwh sp premiosconcurso participa del

Simplemente elimina el registro para la agrupación en ese concurso. Naturalmente, la pertinencia del borrado ya se habrá gestionado en la BBDD.

Aparte de estas, tal cual se expuso anteriormente, se tienen una más, la utilizada por el disparador en ubicados en tablas que afectan a más de una funcionalidad: concretamente las siguientes:

- dwh sp premiosconcurso agrupacionanio up

Similar al caso de los eventos, si cambia el nombre de la agrupación, se actualiza en la tabla.

Por último, como funciones internas se tienen las que representarían a los procedimientos almacenados de alta y baja en DWH PremiosConcurso.

- dwh alta premiosconcurso

Este recibe la agrupación el concurso y la clasificación de la misma, a partir de estos datos obtiene todos los necesarios para cumplimentar el registro. Prácticamente todos los obtiene por consulta sencilla, excepto las autorías.

Para el cálculo del autor se plantean dos consultas, cada una se ejecuta en caso de haber obra asociada. Las consultas que agrupan las letras y las composiciones musicales, respectivamente, y devuelven el autor con mayor número de piezas aportadas. Una vez obtenido dicho autor componen el que será el valor representativo del mismo en esta tabla de hechos, atendiendo al tipo de entidad.

La función de baja no es necesaria puesto que solo implica la eliminación del registro, la cual hace la función del disparador que se activa en las eliminaciones de participa.

Agrupaciones Ganadoras Concurso

La siguiente vista genera una lista ordenada por año y concurso, luego por modalidad y premio.

<anio, concurso, localidad, modalidad, premio, agrupacion, autores, localidad>

Dedicación oficial al Carnaval por persona

Este módulo estadístico satisface el requisito identificado como **RF07.3**. La intención en este caso es conocer la trayectoria de de los participantes oficiales en el Carnaval de Cádiz, concretamente el porcentaje de participación. Para esto solo se considera necesario conocer el nombre con apellidos y el número de veces por cada tipo de rol, incluido el detalle del mismo, que cada persona ha intervenido en el carnaval; por tanto, además de las instancias de rol que se creen, habrá recoger las de autoría y las de creación de diseños. Por dar más coherencia al proceso analítico se introduce el valor del año de la primera aparición de la persona y el valor del último, teniendo en cuenta que se conservan datos desde en siglo XIX.

Cabe aclarar, con base en este detalle, que la autoría contendrá como valor el total de letras, músicas y diseños considerándolos individualmente; de hecho, en este concepto es de interés el genero y la modalidad a la hora de componer y diseñar.

Se propone el siguiente modelo lógico:

DWH Entidad (ID, tipoEntidad, nombre, apellido1, apellido2, localidad, provincia, entidad)

DWH_FuncionesEntidad (entidad, funcion, primero, segundo, tercero, cuarto, veces,
anioInicio, anioFin)
{entidad} es clave externa a DWH Entidad (ID)

La relación DWH_Entidad tendrá carácter de tabla de dimensión respecto a la siguiente. Su objetivo es la sincronización entre los valores de la tabla Entidad de la BBDD y esta; por tanto, únicamente es necesario un *trigger* por cada subtipo de entidad que, cuando haya acciones de tipo inserción o actualización en dicha entidad, envíe a esta los datos registrados en aquellas. Como se puede observar, la eliminación de entidades en el origen no implicará la eliminación en esta, básicamente por el carácter histórico de la misma. Por otro lado, como parte del procesamiento de los datos se genera el contenido del atributo valor mediante la concatenación del resto.

La tabla de hechos, DWH_FuncionesEntidad, gestionará un contador por cada tipo de rol que se le conozca, incluido el posible atributo específico; es decir, puede tener tantos valores como roles posibles, y valores diferentes de cada uno, pueda adoptar una persona en cuanto a las agrupaciones. Caso de campo obligatorio bajo la premisa de que toda persona registrada es por tener algún tipo de relación con el carnaval; sin embargo, en primera instancia, se considera que tras el registro de las personas no tiene porque habersele declarado un rol, esto supondría declarar roles antes que personas. Además, en futuras ampliaciones podrían registrarse personas no participantes en agrupaciones.

Esta tabla de hechos se actualizará por cuatro vías; en primer lugar, por cada inserción en la tabla rol de la BBDD se creará en esta una tupla que indique el tipo de rol con el atributo conveniente al mismo, el año de la participación y el valor 1 en el atributo veces, esto si la tupla no existe; en caso contrario, si la tupla existe se sumará 1 al valor de veces. En segundo lugar, al asignar una autoría a una Letra, Musica o Diseño se procederá de forma similar; y, en tercero, por cada nuevo registro en la tabla crea. La última vía, por las particularidades del modelo, proviene cuando se declara a autoría de una obra o un rol en una agrupación se deberá recalcular este atributo multivaluado.

Por otro lado, una quinta vía, fuera del alcance de la presente funcionalidad estadística, es la debida a la introducción de premios, que se almacenan en esta tabla de hechos, pero se desarrolla en el siguiente apartado como funcionalidad independiente.

En relación al nivel de desnormalización de este par de tablas; por una parte, se plantea en este caso tener un atributo por cada premio, en lugar de una tupla por cada ocurrencia de premio y de participación, puesto que esta tabla de hechos satisfará los requisitos RF07.2 y RF07.3. En este caso se extraen los datos de las entidades a modo de dimensión de la tabla de hechos; principalmente porque la lista mostrará concatenados los datos que componen la clave candidata de la entidad que sea, la cual, a su vez, puede ser de cualquiera de los tres subtipos de entidad; por tanto, si en la tabla de origen uno de estos datos fuera actualizados crearía inconsistencias en estos resultados. Por otro lado, hay que tener en cuenta que una persona rara vez cambia más de tres o cuatro veces de función, por tanto la combinación entre las relaciones basada en índice será bastante eficiente.

Implementación

Aunque el modelo lógico cuenta con dos tablas, la funcionalidad las considera como una, ya que la primera es en realidad una dimensión de la segunda. De nuevo, tomando como estructura la lista de *triggers* necesarios para la actualización de esta tabla, se comenta la implementación de las funciones utilizadas por los mismos; las cuales están contenidas en el *script* **06_DWH_funcionesentidad.sql**. El primer bloque se centra en la gestión de la tabla de dimensión DWH_Entidad.

- dwh_tr_entidad_persona_in
 - dwh_sp_entidad_persona_in
Cuando se insertan registros en Persona se hace la composición del valor nombre en DWH_Entidad, concretamente usando el nombre y los apellidos.
- dwh_tr_entidad_persona_up
 - dwh_sp_entidad_persona_up
Cuando se actualizan registros en Persona, si se modifica alguno de los valores utilizados en DWH_Entidad, se gestiona el contenido de esta última para el identificador de entidad dado. En el caso de ser uno de los intervinientes en el compuesto, se vuelve a realizar la composición.
- dwh_tr_entidad_agrupacion_in
 - dwh_sp_entidad_agrupacion_in
En este caso no hay composición, simplemente se utiliza el alias. En caso de ser nulo, cosa que la tabla Agrupacion admite para este campo, se pone un valor aclaratorio.
- dwh_tr_entidad_agrupacion_up
 - dwh_sp_entidad_agrupacion_up
Similar a la anterior, solamente comprueba si han cambiado alguno de los valores de interés, en cuyo caso actualiza en DWH_Entidad.

- dwh_tr_entidad_empresa_in
 - dwh_sp_entidad_empresa_in
Cuando se insertan registros en Empresa se hace la composición del valor nombre en DWH_Entidad, concretamente usando el nombre y los apellidos.
- dwh_tr_entidad_empresa_up
 - dwh_sp_entidad_empresa_up
Igual a los anteriores, en este caso sin gestión de nulos, ya que todos los que interesan son obligatorios en origen.
- dwh_tr_entidad_entidad_del
 - dwh_sp_entidad_entidad_del
Para la eliminación en DWH_Entidad se considera más eficiente el evento de eliminación cuando ocurre en la que es superclase de las tres anteriores, Entidad.

A continuación comienzan los procedimientos propios de esta funcionalidad estadística.

- dwh_tr_funcionesentidad_cargo_in
 - dwh_sp_funcionesentidad_cargo_in
Conocido el tipo de cargo localiza el identificador de persona y agrupación, más el año de la misma por combinación. Acto seguido se llama a Alta_FuncionesEntidad, que se describe más adelante; y, seguidamente, se pasan los identificadores y el tipo de cargo a la función CambiaPremiados_FuncionesEntidad, que se describe en el siguiente apartado, dedicado a los premios por persona.
- dwh_tr_funcionesentidad_cargo_up
 - dwh_sp_funcionesentidad_cargo_up
Si el tipo de cargo ha cambiado se procede como en el anterior, localizando los identificadores, luego se llama a la función Baja_FuncionesEntidad con los identificadores más los valores de la estructura OLD y a Alta_FuncionesEntidad. De forma alterna entre estos se llama CambiaPremiados_FuncionesEntidad, que tiene un modo para incremento y otro para decremento de premios.
- dwh_tr_funcionesentidad_cargo_del
 - dwh_sp_funcionesentidad_cargo_del
Finalmente, este utiliza Baja_FuncionesEntidad y CambiaPremiados_FuncionesEntidad, con la petición de decremento; previa localización los identificadores.
- dwh_tr_funcionesentidad_orquesta_in
 - dwh_sp_funcionesentidad_orquesta_in
Muy similar a la descrita para el cargo, añade el acceso a la tabla Instrumento para utilizar el valor alfanumérico asociado al identificador conocido.
- dwh_tr_funcionesentidad_orquesta_up
 - dwh_sp_funcionesentidad_orquesta_up
En este caso es ante el cambio de instrumento cuando se procede con las llamadas a Alta_FuncionesEntidad, Baja_FuncionesEntidad y, para decremento de la función a cambiar más incremento de las que se va a establecer CambiaPremiados_FuncionesEntidad.
- dwh_tr_funcionesentidad_orquesta_del
 - dwh_sp_funcionesentidad_orquesta_del
Similar a dwh_sp_funcionesentidad_cargo_del.
- dwh_tr_funcionesentidad_voz_in
 - dwh_sp_funcionesentidad_voz_in
Tanto esta como las dos funciones siguientes son exactamente iguales a las descritas para Cargo, teniendo en cuenta que ambas tablas basan su campo principal en un dato de tipo enumeración.
- dwh_tr_funcionesentidad_voz_up
 - dwh_sp_funcionesentidad_voz_up

- dwh_tr_funcionesentidad_voz_del
 - dwh_sp_funcionesentidad_voz_del
- dwh_tr_funcionesentidad_letra_in
 - dwh_sp_funcionesentidad_letra_in

En este caso y los siguientes el tipo de función a registrar es la autoría, el alta en la tabla Letra y las demás requieren establecer el autor. Por tanto, en esta ocasión no es necesario acceder a más valores que los provistos, más los obvios, para poder hacer uso de la función Alta_FuncionesEntidad.

En cuanto a la agrupación que interpreta la letra, accesible a través de la signatura de la obra, podría no existir; por tanto, a la función CambiaPremiados_FuncionesEntidad sería llamada en caso de que se haya localizado una agrupación.

- dwh_tr_funcionesentidad_letra_up
 - dwh_sp_funcionesentidad_letra_up

En este caso se actúa si cambia el autor del registro editado en Letra. En este caso se intenta localizar el año y la agrupación, de los cuales ninguno es obligatorio para llamar a Baja_FuncionesEntidad ni Alta_FuncionesEntidad. Luego, tal cual se hizo en el anterior, si se localiza la agrupación interprete se llama a CambiaPremiados_FuncionesEntidad, de forma alterna, primero para decremento y luego para incremento.

- dwh_tr_funcionesentidad_musica_in
 - dwh_sp_funcionesentidad_musica_in

Al igual que lo detallado para la gestión de autorías de literarias, la relación con las agrupaciones –por tanto, con año y premio– es indirecta; no obstante, como quizás se den composiciones musicales nunca interpretadas por agrupaciones, se opta por hacer el registro igualmente. De nuevo en este caso, se trata de localizar la agrupación interprete, junto con el año de esta última; y luego, con año o sin el, se llama a Alta_FuncionesEntidad y, si hubo agrupación identificada, a CambiaPremiados_FuncionesEntidad.

- dwh_tr_funcionesentidad_musica_up
 - dwh_sp_funcionesentidad_musica_up

Análoga a la descrita para las letras.

- dwh_tr_funcionesentidad_musica_del
 - dwh_sp_funcionesentidad_musica_del

Utiliza Baja_FuncionesEntidad y CambiaPremiados_FuncionesEntidad, en modo decremento. Cabe señalar que este último ignora las llamadas que no indican la agrupación, por tanto no es necesario el control de este hecho de forma previa –cosa que se ha implementado para las inserciones y actualizaciones, porque se podía obtener el valor en una consulta compartida–.

Por otra parte, no se describen en esta sección las funciones relativas al *trigger* asociado a *delete* en Letras y Disenio porque son disparadores compartidos por varias funcionalidades estadísticas; como se hiciera anteriormente, ambas se describen al final de este apartado.

- dwh_tr_funcionesentidad_disenio_in
 - dwh_sp_funcionesentidad_disenio_in

De nuevo, se obvia la descripción de este y el siguiente por ser exactamente iguales a los anteriores; exceptuando que en lugar de utilizar el género como descriptor de la función, utiliza el campo tipoDiseno que es el correlativo.

- dwh_tr_funcionesentidad_disenio_up
 - dwh_sp_funcionesentidad_disenio_up

- dwh_tr_funcionesentidad_crea_in
 - dwh_sp_funcionesentidad_crea_in

En este último caso, estrictamente centrado en las funciones adoptadas por entidad, hay que acceder al valor del tipo de diseño; que, como hasta ahora, se utilizará como parte de la descripción de la función que la entidad realiza. Luego procede con las llamadas a Alta_FuncionesEntidad y CambiaPremiados_FuncionesEntidad, si hubo agrupación identificada.

- dwh_tr_funcionesentidad_crea_up
 - dwh_sp_funcionesentidad_crea_up
Esta verifica si cambia alguno de los participantes de la relación, ya sea el diseño o el diseñador; si hubo cambios, se procede como en el anterior, obtiene el tipo de diseño a sustituir y el sustituto, a partir de los registros *old* y *new*. Luego se da la baja, se obtiene la agrupación y se solicita el alta (o aumento del contador) para el nuevo par de entidad creadora y diseño; así mismo, se solicita la gestión de premios por entidad si hay agrupación conocida.
- dwh_tr_funcionesentidad_crea_del
 - dwh_sp_funcionesentidad_crea_del
Este último funciona exactamente igual que los descritos para los componentes de las diversas obras.

Como en la funcionalidad anterior, se requería de un disparador en AgrupacionAnio para poder gestionar la información objetivo de esta tabla de hechos; en esta ocasión, además, se tiene respecto a Letra y Disenio, ambas compartidas con la funcionalidad de temas recurrentes.

- dwh_sp_funcionesentidad_agrupacionanio_up
Como en la mayoría de las funcionalidades de *data warehouse* propuestas, al editar las agrupaciones de nivel anual, hay que controlar que se cambien las obras que estas utilizan. En este caso concreto, al cambiar o introducir alguna obra hay que tratar las autorías porque estos roles se introdujeron sin gestionarle los posibles premios y tampoco se les pudo asignar el año.
Por tanto, hay que recorrer los componentes de la obra que se retira y de la que se asigna, luego se llamará a CambiaPremiados_FuncionesEntidad para incremento o decremento y se gestionarán los años de inicio y final de la trayectoria según los conocidos.
- dwh_sp_funcionesentidad_letra_del
Para las bajas de funciones de tipo autor de la letra, no se hace ningún tipo de preparativo, simplemente se llama a la función Baja_FuncionesEntidad. Luego, respecto a la gestión de premios, si la agrupación interprete es nula, la función CambiaPremiados_FuncionesEntidad descartará la llamada.
- dwh_sp_funcionesentidad_disenio_del
Idéntica a la función recién comentada respecto al *trigger on delete* en Letra.

Finalmente, se procede con el detalle de los procedimientos de alta y baja para esta tabla de hechos. Cabe comentar que en este caso las modificaciones son en realidad altas o bajas del contador existente, por ende es innecesaria la función de modificación.

- __dwh_alta_funcionesentidad
Esta implementada de forma sencilla, a modo de procedimiento almacenado de alta, aunque con las implicaciones que conlleva la gestión de contadores; es decir, se valora la existencia de la combinación entre entidad y función.
Recibe la entidad, la función y el año de adopción del rol; luego, si existe la función para la entidad en la tabla de hechos, se aumenta el contador y se establece como último año conocido el indicado; respecto a esto último, se controlará que el años recibido sea mayor al registrado como último, con objeto e no reducir la trayectoria de la entidad. En caso contrario, se crea el registro indicando como primer y último año el mismo.
- __dwh_baja_funcionesentidad
Este tiene una implementación complementaria al anterior, para las bajas se evalúa el valor del contador, si el mayor que 1 se reduce 1, si es 1 se elimina el registro de la entidad para la función indicada. En esta ocasión se desestima el año.

DedicacionEntidad

<entidad, funcion, veces, anioInicio, anioFin>
--

Como consulta a la relación DWH_FuncionesEntidad es similar a la anterior, solo que centrada en otros campos con otro sentido; además, y por el contrario, no se observa la necesidad de componer el resultado a modo de matriz, como en el caso anterior. Lo que si requiere es la composición de esta tabla de hechos con la tabla de dimensión DWH_Entidad.

Premios por entidad y función

Este es un caso de información de interés de cara a la contratación por parte de los directores de agrupación, considerando que las agrupaciones son anuales, las personas y agrupaciones son quién acumulan premios; además, permitirá averiguar el prestigio de las entidades participantes en función de la actividad que desarrollan. En principio se tiene como suficiente gestionar contadores por entidad y rol específico para los premios 1º, 2º y 3º.

En el requisito funcional RF07.2, se hace alusión a las personas, agrupaciones y empresas; una vez realizado el diseño completo de la BBDD, se tiene la posibilidad de utilizar simplemente el identificador de nivel Entidad, puesto que las anteriores son subtipos de esta última relación. Por otro lado, se introduce el concepto de función para las entidades, dadas las posibilidades del modelo obtenido en este punto; se consideran funciones: el rol entre personas y agrupaciones, la autoría entre entidades y obras y la ejecución de los diseños. Respecto a esto último, hay un concepto no incluido en el alcance: el premio “Aguja de Oro”, que si será admitido por este resumen de datos cuando el modelo se amplíe a los eventos de tipo premio.

Tal como se ha indicado anteriormente, esta funcionalidad hace uso de las tablas de hechos y dimensión anteriores, así como del mismo *script* de creación de funciones. El uso que se hace de dicho par consiste en añadir una unidad al atributo de premio que corresponda al registrarse un premio para una agrupación; por tanto, debe averiguar el rol de la persona respecto de la agrupación premiada y aumentar el contador correspondiente. Se propone, en este caso, que el disparador esté asociado al evento tras inserción/actualización de la relación DWH_PremiosConcurso; de esta manera, solo se recibirán registros con premios menores al cuarto –que son los insertados en dicha tabla– y se reduce la carga en el lado de la BBDD operacional.

Implementación

Los siguientes tres disparadores, en realidad sus correspondientes funciones, actualizan la tabla de hechos DWH_FuncionesEntidad de forma totalmente independiente al resto de las descritas para la funcionalidad estadística anterior, dedicada a las funciones realizadas por las diversas entidades. Para ello todas utilizan la misma función, descrita justamente tras ellas, __dwh_sp_cambiapremioagrupacion_funcionesentidad; esta utiliza el identificador de la agrupación y el valor del premio, que por estar el *trigger* asociado a la tabla de hechos DWH_PremiosConcurso, se encuentra entre 1 y 3.

- dwh_tr_funcionesentidad_dwhpremiosconcurso_in
 - dwh_sp_funcionesentidad_dwhpremiosconcurso_in
Esta utiliza la función recién mencionada para solicitar el incremento del contador correspondiente al valor de premio en toda entidad asociada a la agrupación, que también se le pasa como parámetro.
- dwh_tr_funcionesentidad_dwhpremiosconcurso_up
 - dwh_sp_funcionesentidad_dwhpremiosconcurso_up
Al igual que la anterior, llama para decremento del contador respecto al premio que la agrupación tenía asignado y, posteriormente, al incremento del nuevo que se le asigna. Sin embargo, a diferencia de las inserciones, se verifica si ha cambiado el premio.
Esto último es de gran importancia por la carga computacional que supondría, hacer el recorrido de todos los participantes de la agrupación, si no se ha modificado dicho valor; aunque, el estado final de los datos fuera coherente con la realidad.
- dwh_tr_funcionesentidad_dwhpremiosconcurso_del
 - dwh_sp_funcionesentidad_dwhpremiosconcurso_del
Finalmente, en caso de eliminación, sin necesidad de verificación alguna se procede a solicitar el decremento del premio para todos los involucrados en la agrupación.

A continuación se describen las dos funciones de uso interno, la primera de ellas es la utilizada por estos tres *triggers*, dentro del carácter autónomo de esta funcionalidad; mientras que, la segunda, es utilizada por las funciones centradas en el módulo estadístico anterior. Con esta implementación se logra la reducción del código a la vez que se aprovechan los eventos, teniendo en cuenta que ambas funcionalidades se centran en las entidades y sus ocurrencias en el dominio; de hecho, con una sola función se gestiona tanto el incremento como el decremento de los contadores.

- __dwh_sp_cambiapremioagrupacion_funcionesentidad
Cuando en una agrupación se modifica, se introduce o se elimina un premio hay que recorrer todos los roles asociados a la misma para reflejar este cambio. En este caso, donde se llama a la función desde el punto de vista de la agrupación y el premio, hay que localizar a las entidades e identificar sus posibles funciones; para ello:

- Se implementa un bucle que itera sobre el conjunto de registros en rol que tiene la agrupación. En cada iteración:
 - En función del tipo de rol que indica el registro activo se accede al descriptor del rol en concreto; es decir, del registro del subtipo de rol indicado por el atributo discriminador de la superclase. Todo esto siguiendo las mismas pautas de extracción del descriptor, o función, que se hicieron para la inserción en esta tabla de hechos.
 - Una vez obtenido el valor llamado función, que es idéntico al contenido en el campo funcion de la tabla de hechos, se procede a añadir el valor recibido para el contador según el valor del premio. El valor que se recibe para añadir al contador puede ser 1 o -1; de esta forma se hace válida la función para ambas necesidades. En definitiva, si el premio es igual a 1, se incrementa o decrementa el campo llamado primero de la tabla.
 - Seguidamente procede a evaluar los involucrados a la agrupación dada como autor. Esto se hace de forma similar al caso anterior, en lo referente al aumento de los campos primero, segundo y tercero, según sea el valor de premio recibido y según si la unidad a incrementar es positiva o negativa. Para las autorías de los tres tipos de obras y de la creación de los diseños se actúa de la misma forma.

Primero se hace una consulta con la clausula *distinct* respecto al autor sobre el conjunto de registros de Letra asociados a la obra literaria indicada en interpretaLiteraria. Luego, se añade el incremento, o decremento, al premio según cual sea el recibido.

El funcionamiento y uso general de esta función, al igual que la siguiente, es con decremento y luego incremento; es decir, de baja y alta del premio para la agrupación. En cierto modo, actúa de la forma que lo hacen las funciones ejecutadas ante la edición de agrupaciones de carácter anual, por el tema de la dependencia entre las agrupaciones y las obras, por el hecho de conformar una relación indirecta entre varios conceptos.

- dwh_sp_cambiapremiados_funcionesentidad

Para finalizar, esta es la función que se utiliza como soporte a esta funcionalidad estadística centrada en premios desde la anterior, centrada en funciones; por tanto, esta función conoce de partida a la agrupación (al igual que la anterior), a la entidad y su función dentro de dicha agrupación. En consecuencia, esta solo hace el incremento para una entidad individual, ya que conoce las propiedades pertinentes; recordando la implementación del módulo anterior, se la llama por cada alta o baja de función de una entidad.

La implementación es análoga a la anterior, en lo referente a la gestión de los tres contadores en que se centra. No necesita recorrer los roles por agrupación, por el contrario, lo que requiere es conocer las clasificaciones en los posibles concursos que tenga su agrupación.

Una vez descritas estas dos últimas funciones, e incluyendo el hecho de la interdependencia que se ha introducido entre este módulo y el anterior –ambos contruidos sobre las mismas tablas de hecho y dimensión–; se hace notar la complejidad que alcanza mantener la coherencia en la información así como se editan datos en uno y otro lugar de los implicados en estas funcionalidades. Por ejemplo, que si una agrupación deja de interpretar una obra es necesario gestionar todos los contadores; o, un hecho similar, si una agrupación pierde un premio o se le cambia.

A todo esto, de nuevo en función de la relación entre las agrupaciones y las obras, ocurrirá para las funcionalidades de temas recurrentes y soporte al proceso documental. En el primero de los casos, si no hay agrupación no se gestionan los temas, por no conocer el año y la modalidad; en el segundo, porque el objeto de control son las agrupaciones, a partir de ellas que documentos se les asocian solo pueden ser obtenidos a través de las obras que utilizan.

PremiosEntidad

<entidad, funcion, primeros, segundos, terceros, cuartos>

La consulta consistirá en combinar las dos relaciones implicadas, al igual que en el caso anterior, utilizando el campo entidad de la relación DWH_Entidad, y la proyección de los atributos indicados en esta estructura, que no implican más operaciones ni cálculos. De hecho, sobre esta se podrían agregar los acumuladores a premios por persona, obviando el rol.

Temas recurrentes en letras por año y modalidad

Con esta funcionalidad estadística se solventa el requisito identificado como RF07.5 Para los temas incluir el año es de interés por el contraste histórico, teniendo en cuenta que se pueden hacer consultas al mismo conjunto de datos obviando el año, se obtiene una doble utilidad. Por otro lado, hay que diferenciar entre los temas que etiquetan a las letras y los que lo hacen a los tipos que adoptan las agrupación; así como la modalidad, el caso de las chirigotas y los cuartetos adoptan un carácter humorístico y las comparsas más serio; por tanto, la modalidad es de interés para la consulta de letras en función del tema.

Se propone a nivel lógico la siguiente relación:

DWH_TemasRecurrentes (id, id_tema, tema, tipoObra, anio, modalidad, categoria, total)

Como comentario al proceso de desnormalización, contendrá redundancia en el caso de valor de tema, se dará una vez por cada tipo de obra (Letra y Disenio), año, modalidad y categoría. Cabe recordar que las modalidades realmente utilizadas son un total de 5; de hecho, documentadas oficialmente 4; además, el valor tema es bastante corto. A esta, durante el proceso de diseño físico y de desarrollo se añade el identificador del tema; por tanto, en parte duplica el valor ya repetido.

Implementación

Se tienen las siguientes funciones y *triggers*:

- dwh_tr_temasrecurrentes_tema_up
 - dwh_sp_temasrecurrentes_tema_up
Mantiene la coherencia entre el contenido de la tabla Tema y la tabla de hechos de este módulo estadístico.
- dwh_tr_temasrecurrentes_temalettra_in
 - dwh_sp_temasrecurrentes_temalettra_in
En primer lugar, mediante una consulta que combina la obra a la que pertenece la letra con las agrupaciones, trata de obtener los valores que deben insertarse en esta tabla de hechos; si los localiza llamará a dwh_alta_temasrecurrentes, que se describe más abajo como viene haciéndose para las funciones definidas como internas. Además de los valores obtenidos de la consulta, pasa el valor 'Literaria' como tipo de obra.
Como puede observarse, aunque la función llamada para el alta del registro no realiza actualización alguna si le faltan valores, esta no la llama de no obtener dichos valores en su consulta; de esta manera, aunque en la función dwh_alta_temasrecurrentes se centraliza la verificación de los requisitos necesarios, se evita la consulta inicial que esta última añadiría antes de finalizar la ejecución completa que requiere el disparador.
- dwh_tr_temasrecurrentes_temalettra_up
 - dwh_sp_temasrecurrentes_temalettra_up
De la misma manera que el anterior, realiza una consulta para localizar los valores necesarios contenidos en la tabla AgrupacionAnio, si los obtiene llama a dwh_baja_temasrecurrentes y luego a dwh_alta_temasrecurrentes. En el primero indica los valores contenidos en el registro *old* del disparador y, en el segundo, los contenidos en *new*; también en este caso añade el valor de tipo de obra, 'Literaria'.
- dwh_tr_temasrecurrentes_temalettra_del
 - dwh_sp_temasrecurrentes_temalettra_del
Análogo a los anteriores, busca los valores de la agrupación y procederá con la llamada a la función de bajas si obtuvo resultados.

Los tres *triggers* siguientes actúan exactamente igual, con la misma implementación que los recién expuestos, exceptuando el valor de tipo de obra que indican, 'Gráfica' en este caso.

- dwh_tr_temasrecurrentes_temadisenio_in
 - dwh_sp_temasrecurrentes_temadisenio_in
- dwh_tr_temasrecurrentes_temadisenio_up
 - dwh_sp_temasrecurrentes_temadisenio_up
- dwh_tr_temasrecurrentes_temadisenio_del
 - dwh_sp_temasrecurrentes_temadisenio_del

Las dos siguientes son las de uso interno, al igual que se viene estructurando, a modo de procedimientos de alta y baja en la tabla de hechos más haciendo gestión de incremento o decremento de los contadores previa verificación de la existencia.

- dwh_alta_temasrecurrentes

Esta función recibe como parámetros todos los valores necesarios para gestionar la tabla de hechos sobre la que actúa, excepto el valor de tipo cadena que describe el tema y la modalidad, los cuales consulta en las tablas correspondientes en el momento del uso. Se considera que este acceso no introduce una carga computacional excesiva, en todo caso sería el filtrado en la tabla Tema pero este solo se hace una vez en todo el procedimiento; en consecuencia, se decide implementarlo directamente a modo de subconsulta.

Con base en estos argumentos recibidos, procede a comprobar si en la tabla de hechos existe un registro para el tema, el año, la modalidad y la categoría recibidas. En caso de localizarlo, mediante una sentencia *update* incrementa el acumulador contenido en el campo total. En caso contrario, primero verifica que haya recibido el año, la modalidad y la categoría –lo que implica que el tema está indirectamente asociado a una agrupación–, si los tiene insertará el conjunto de valores, con la obtención de las cadenas descriptoras de la modalidad y el tema a modo de subconsultas, e inicializa el total a 1.

Como en funcionalidades anteriores, la gestión de temas recurrentes es totalmente dependiente de la participación de las agrupaciones; por ende, hace falta recorrer las letras y diseños cuando a una agrupación se le asigna una obra literaria o gráfica.

- dwh_baja_temasrecurrentes

En este caso, como en el resto de los dedicados a gestión de contadores, se verifica que el valor del contador del registro al que aluden los argumentos recibidos, total en esta tabla de hechos. Si es mayor, mediante *update* se restará una unidad, y si es igual a 1 se eliminará el registro.

Como parte de las restricciones de esta funcionalidad, también se controla el hecho de la vinculación del tema a alguna agrupación, solo que en la implementación queda implícita. Los parámetros recibidos son los mismos que para el alta, luego al hacer la consulta no obtendrá resultados si alguno es nulo; por esta razón, los condicionales si son explícitos, tanto en esta función como en la de altas –no obstante, de no ser explícitas, el *delete* simplemente generaría una excepción por no existencia, y el *insert* generaría otra por no poder introducir nulos–.

Por último, las funciones utilizadas por la definida para los *triggers* de uso compartido:

- dwh_sp_temasrecurrentes_agrupacionanio_in

Mediante esta función, llamada desde dwh_sp_agrupacionanio_in, se incrementan los contadores de los temas que no fueron gestionados por no tener sus letras y diseños una agrupación asociada; es decir, se consideran todos los temas que aún no se han incluido en la tabla de hechos DWH_TemasRecurrentes.

En este evento, de inserción en AgrupacionAnio, se actúa si el valor de interpretaLiteraria o utilizaTipo es distinto de nulo. Para cada uno de estos, si no es nulo accede a los temas vinculados a cada componente de la obra dada –en las tablas Letra y Disenio está la signature de la obra–; a partir del conjunto de datos obtenidos, itera llamando para todos los temas obtenidos dwh_alta_temasrecurrentes. Desde este punto de vista ya se cuenta con los valores necesarios de la agrupación, los cuales son obligatorios.

- dwh_sp_temasrecurrentes_agrupacionanio_up

Para la actualización, como se viene realizando, verifica que los valores de interés hayan sido modificados interpretaLiteraria o utilizaTipo; luego, realiza las iteraciones como recién se ha comentado. En primer lugar, si cambia la obra hace una primera iteración según la signature de obra que se retira; luego, si se ha asignado una nueva obra, ya sea gráfica o literaria, se itera por todos los temas asociados a los componentes de la obra. De esta forma, si no hay nueva obra asignada se finaliza, lo que contempla la posibilidad de que se establezcan dichos campos a *null*.

- dwh_sp_temasrecurrentes_letra_del

Este y el siguiente son los casos en que se requería el uso de *triggers* definidos en el momento antes de la operación, concretamente del borrado, porque las relaciones existentes entre los temas y las letras o diseños se eliminaban en cascada; en consecuencia, los contadores no disminuían nunca.

Dicho esto, basado en los parámetros que recibe, contenidos en el registro *old* que conoce la función del *trigger* que llama a esta, itera sobre los temas asociados al registro de Letra que se va a eliminar y, para todos ellos, llama a la función dwh_baja_temasrecurrentes.

- dwh_sp_temasrecurrentes_disenio_del
Idéntico al anterior respecto a la tabla Disenio.

TemasRecurrentes

<tema, tipoObra, anio, modalidad, categoría, total>

En esta consulta no es necesario ningún tipo de operación de combinación ni agregación; por tanto, con respuestas en tiempo constante igual a 1; sin embargo, por la composición de campos se darán posibilidades analíticas, por ejemplo agrupar por modalidad obviando el año y la categoría, etc.

Soporte al proceso documental de las agrupaciones

A lo largo del proceso de diseño se decidió adscribir esta funcionalidad de soporte, o aporte de información, al sistema de *data warehouse* por el carácter de resumen que tienen los datos implicados. Concretamente, se trata de tener acumuladores por cada agrupación y obra que indiquen para cada tipo de documento y soporte el total de estos registrados.

El modelo lógico propuesto para esta tabla de hechos es el siguiente:

DWH_SoporteDocumental (agrupacion, tipoObra, tipoDocumento, tipoSoporte, total)

Con esta tabla de hechos simplemente se pretende acceder a la tipología de documentos, correspondientes a obras, que a su vez están asociadas a las agrupaciones. La desnormalización se da por la redundancia que introducirán los tres primeros campos respecto del tercero.

Se repite el mecanismo descrito para gestión de acumuladores, al crear registros en Documento se ejecutará el procedimiento almacenado dedicado a gestionar esta relación. La complejidad se deriva del hecho de no existir relación de interpretación, entre la obra asociada al documento y alguna agrupación – cosa que se ha dado en varios de los módulos estadísticos diseñados–.

Para el objetivo concreto de esta información, se solventa esta problemática implementando la inserción desde dos vías. Por un lado, se tendrán disparadores activados tras inserción en las tablas DocumentoObra, DocumentoLetra, DocumentoMusica y DocumentoDisenio y se gestionarán los datos necesarios si la Obra o componente de obra implicado tienen agrupación vinculada. Por otro lado, se tendrá un disparador en AgrupacionAnio que llame a un segundo procedimiento almacenado si los atributos interpretaLetra, interpretaMusica o utilizaTipo pasan de nulo a contener una referencia. En este caso el procedimiento accederá a los documentos asociados a la Obra a la que ahora apunta la agrupación e insertará todas las tuplas pertinentes; de hecho, este mismo procedimiento, llamado por el mismo disparador tendrá que actualizar la relación entre agrupación y documentos en caso de cambios entre agrupaciones y obras.

Con este esquema de actuación se registrarán de forma directa solo los documentos que tengan relación a agrupaciones y, los que no la tengan, serán registrados en lote en el momento en que la Obra a la que se adscriben sea vinculada mediante una relación de interpretación o utilización a cualquier agrupación.

Implementación

Una vez expuesto el mecanismo general, y tras todos los argumentos dados para las funcionalidades *data warehouse* anteriores, los detalles que se dan a continuación se centran en lo específico.

- dwh_tr_soportedocumental_documento_up
 - dwh_sp_soportedocumental_documento_up

Cuando se actualizan los documentos, para este módulo estadístico, interesa si se modifica el tipo de documento o soporte; obviamente, que cambie el tipo de documento de forma correcta según el tipo de obra queda controlado en el lado de la BBDD.

Si uno u otro valor cambia, en primer lugar, se recorre la tabla que relaciona los documentos con las obras y se crea un *recordset* con todas las referencias al documento activo durante la llamada al *trigger* que están relacionados a obras en estado de interpretación; es decir, combinando las tablas Documento, Obra, DocumentoObra y AgrupacionAnio se filtran las que contienen el identificador del documento en cuestión. Luego, iterando por el conjunto de registros resultantes, se realizan llamadas a dwh_baja_soportedocumental, con los valores del *old*, y a dwh_alta_soportedocumental con los del *new*; en este último caso, se hará el alta si el tipo de soporte es distinto a 'No disponible'.

Por último, se busca el identificador del documento en el resto de tablas de relación donde podría estar, DocumentoLetra, DocumentoMusica y DocumentoDiseño; entre estas, de forma separada y también combinadas con las agrupaciones. En cualquiera de las mencionadas, si aparece será solo una vez por restricción de clave primaria sobre documento; por tanto, no se hace uso de bucles sino de dos variables donde contener los valores de agrupación de los que no se dispone en esta función. De existir la agrupación se procede a solicitar el decremento del contador, más de ser de soporte distinto a 'No disponible' se solicita el alta.

Todas las descritas a continuación tienen una implementación similar, aunque en el primer caso el nivel que alcanza la consulta inicial –para obtención de los datos necesarios, ya que en estas solo se tiene dos identificadores– es menor que en los tres últimos pares, donde se combina además la tabla del componente correspondiente. Por otro lado, respecto a esto de los pares, por construcción de la BBDD no se considera la actualización de uno de los elementos en estas tablas; es decir, de una de las referencias a obra, componente o documento. Dicho esto, se describe el primero de estos pares centrados en las tablas de relación.

- dwh_tr_soportedocumental_documentoobra_in
 - dwh_sp_soportedocumental_documentoobra_in
Tal como se acaba de introducir, se hace una consulta a la combinación de AgrupacionAnio, Documento y Obra, que es la implicada por la tabla donde se define este *trigger*. Concretamente busca los valores de agrupación, identificador y nombre, el tipo de documento, de soporte y de obra; si se obtiene registro (porque tiene relación con alguna agrupación) y el tipo de soporte no es 'No disponible', se pasará el conjunto de valores a dwh_alta_soportedocumental.
- dwh_tr_soportedocumental_documentoobra_del
 - dwh_sp_soportedocumental_documentoobra_del
La consulta es exactamente la misma que la anterior, si localiza un registro coincidente solicita la baja sin atender al tipo de soporte.
- dwh_tr_soportedocumental_documentolettra_in
 - dwh_sp_soportedocumental_documentolettra_in
- dwh_tr_soportedocumental_documentolettra_del
 - dwh_sp_soportedocumental_documentolettra_del
- dwh_tr_soportedocumental_documento musica_in
 - dwh_sp_soportedocumental_documento musica_in
- dwh_tr_soportedocumental_documento musica_del
 - dwh_sp_soportedocumental_documento musica_del
- dwh_tr_soportedocumental_documento disenio_in
 - dwh_sp_soportedocumental_documento disenio_in
- dwh_tr_soportedocumental_documento disenio_del
 - dwh_sp_soportedocumental_documento disenio_del

Las tres siguientes son las diseñadas como internas con carácter de procedimiento almacenado para alta y baja de la tabla de hechos.

- dwh_alta_soportedocumental
Recibe como parámetros el identificador y el nombre de la agrupación más el tipo de obra, de documento y de soporte. Seguidamente comprueba que el tipo de soporte sea distinto a 'No disponible', lo cual indica que se cuenta con un documento tangible de la índole que sea.
A partir de este punto prosigue como de costumbre: accede a la tabla de hechos, DWH_SoporteDocumental, para verificar si existe la combinación de valores de los argumentos recibidos; si existe incrementa el acumulador en total; y, en caso contrario, de haber recibido lo inserta.
- dwh_baja_soportedocumental
Con los mismos parámetros que el anterior, eliminará si existe registro y el contador el igual a 1, restará una unidad si el contador es mayor que 1 y no hará nada de no localizarlo; esto, probablemente, indique que no se han recibido identificadores de la agrupación.
- dwh_alta_soportedocumental_obra

Esta es la utilizada por las tres siguientes para realizar un recorrido por todos los documentos existentes, que tengan relación con la obra que recibe como parámetro o con cualquiera de sus componentes. Durante dicho recorrido hace uso de __dwh_alta_soportedocumental.

- dwh_sp_soportedocumental_agrupacionanio_in

Según la tarea que realiza la inmediatamente anterior, cuando se da alta a una agrupación para un año, si tiene alguna obra asignada en los campos interpretaLetra, interpretaMusica o utilizaTipo, se llama a __dwh_alta_soportedocumental_obra indicándole la signatura de la obra y su tipo, más el identificador y el nombre de la agrupación.

- dwh_sp_soportedocumental_agrupacionanio_up

De nuevo para la actualización se retira la información de las obras que se van a sustituir y se colecciona la de la obra que se asigna, en los casos en que las signaturas cambien.

Cabe señalar que no ha sido necesario implementar una función específica para bajas, teniendo en cuenta la información está centrada en las agrupaciones, si una agrupación se desvincula de una obra simplemente se retiran todos los registros de DWH_SoporteDocumental que correspondan al tipo de obra retirada.

- dwh_sp_soportedocumental_agrupacionanio_del

De manera parecida a lo que se acaba de indicar respecto a la retirada de obras, si se elimina una agrupación de carácter anual en la BBDD sencillamente se eliminan todos los registros relativos a la agrupación.

SoporteDocumentalAgrupaciones

<agrupacionAnio, tipoObra, tipoDocumento, tipoSoporte, total>

Al igual que la primera vista, AgrupacionesGanadorasConcurso, se trata de mostrar una lista ordenada, según el orden de los atributos indicados.