

Anexo V: Pruebas

De forma iterativa e incremental, cada procedimiento lleva su propio conjunto de pruebas y se le han aplicado de forma ordenada, atendiendo al orden numérico que se ha dado a los *scripts*; asimismo, el mismo proceso de pruebas de cada uno ha llevado en ciertos casos a reiterar sobre el conjunto del que depende o al que afecta.

Según a esta pauta, por cada procedimiento se centran las pruebas en el uso del conjunto dedicado a alta, modificación y baja por cada tabla con objeto de forzar todas y cada una de las condiciones que estos tienen implementadas. Según esto se avanza respetando las interdependencias entre tablas y los propios procedimientos almacenados. Cabe señalar en este punto que las funciones de uso interno fueron probadas en su momento de desarrollo al igual que las demás; como se exponía en el apartado dedicado a los procedimientos almacenados, toda la implementación se hace de forma incremental según las interdependencias entre tablas.

Finalmente, con objeto de verificar el buen funcionamiento del *data warehouse*, se eliminan datos en sentido contrario al descrito; por esta razón, se entrega un segundo conjunto de pruebas centrado en este aspecto.

La automatización de pruebas se diseña, en todos los casos, haciendo una inserción de un registro a tal efecto. Una vez insertado el registro se obtiene su identificador, en los casos en que no puede indicarse, y se realizan sobre el todas las pruebas pertinentes; por último, se elimina. En otros caso, cuando las inserciones dependen de la existencia de otros registros, se hacen consultas en las tablas de las que depende, tomando uno aleatoriamente.

A continuación se describe el contenido de los *scripts* de pruebas, luego se dan detalles de las pruebas no incluidas en los mismos. Estas últimas son por lo general rápidas, centradas en las restricciones definidas a nivel del DDL.

10_pruebas_01_tablasLookup.sql

De nuevo se comienza por las funciones dedicadas a estas tablas. Se tiene un bloque DO, que se asimila a una transacción, por cada tripleta mínima de procedimientos almacenados; es decir, por cada conjunto de alta, baja y modificación dedicado a cada tabla. En cuanto al proceso de pruebas real, paralelo al desarrollo, se probó también que no aceptaran cadenas vacías o nulos; y, que no admitiera valores repetidos en la función dedicada a la modificación y con diferente capitalizaciones, cosas que en el *script* no se han incluido.

Las tablas Modalidad, Descriptor, Tema, Instrumento y Actividad son todas similares en estructura y restricciones; por tanto, los bloques DO son prácticamente idénticos. Primero, se da alta a un elemento, luego se localiza su identificador, se modifica, se muestra el estado final del registro y se elimina. Luego, se busca un valor cualquiera y se procede a intentar el alta del mismo, con objeto de verificar que no admitirá duplicados. El conjunto de pruebas recién descritos se ha definido para su ejecución de forma automatizada en el *script* 10_pruebas_tablasLookup.sql.

10_pruebas_02_Documento.sql

Durante el desarrollo los siguientes procedimientos almacenados los modelados con carácter interno no se probaran directamente; no obstante, para la automatización si se dan las pruebas para todos estos. Por otro parte, cabe aclarar que se utiliza siempre el identificador 1, teniendo en cuenta que se borra el registro tras las pruebas de cada subclase de Entidad.

- Alta_Documento, Modifica_Documento y Baja_Documento

Como en el caso anterior, se crea un documento y se le aplican diversas modificaciones para forzar errores que demuestren que las inserciones y las modificaciones no permitirán la inserción de registros semánticamente incorrectos. Por este orden, se prueba que:

- El alta sin descriptor, que es el único lugar donde se puede forzar la obligatoriedad de tener un descriptor.
- El alta de una 'Unidad documental simple' no puede tener un tipo de documento que naturalmente es compuesto.
- El alta con el valor enDeposito habilitado es correcta siempre que lleve datos acerca de la ubicación.
- Una modificación sencilla, luego la presentación del nuevo estado del registro.
- Otra modificación por la que se retira el hecho de tener un documento enDeposito, y la presentación del efecto.

- Establecimiento del booleano anterior para forzar el error por faltas de referencias a la ubicación física.
- Se repite respecto a la electrónica; esta última, seguida de la modificación correcta, con el valor estaDigitalizado y un valor de URL.
- La baja no es posible si se tienen datos acerca de la ubicación del documento.

Respecto a la baja, en este caso no llevará a más errores porque se está creando un documento desvinculado de todo, cosa que no debe ocurrir; por tanto, las restricciones de que el documento no mantenga relación con ningún otro objeto estará satisfecha.

Hay que volver a realizar esta prueba más adelante.

- Modifica_ID_Documento

- Se busca el fallo intentando establecer como nuevo identificador un valor de entre los que puede dar la secuencia.
- Luego, se repite con un valor correcto y, la prueba, consiste en demostrar que el Código de Referencia ha sido adaptado al cambio.

- Alta_DepositoLegal_Documento, Modifica_DepositoLegal_Documento y Baja_DepositoLegal_Documento

- En primer lugar, se crea un documento de ejemplo. Este no tendrá información de documento tangible, de forma que el alta del depósito legal fallará.
- Seguidamente, se hace modifica el documento para poder dar el alta. Luego se prueba la modificación
- Se prueba la inserción de un segundo y la inserción de un repetido, que lleva a error.
- Por último, se elimina uno de los dos y se deja el otro; con este último se demostrará, al eliminar el documento, que el registro dejado se elimina en cascada.

No se retira el documento de prueba, será utilizado a continuación para los descriptores.

- Alta_Descriptor_Documento, Modifica_Descriptor_Documento y Baja_Descriptor_Documento

Para probar este conjunto: se añade un registro; luego el mismo, que indicará que ya está asociado; seguidamente, se intenta modificar el documento en lugar del descriptor; se prosigue con una modificación coherente; y, finalmente, se intentarán borrar todos. El primero se borra, el segundo no porque sería el último y mínimo necesario.

10_pruebas_03_Entidad.sql

- Alta_Empresa, Modifica_Empresa y Baja_Empresa

Estos no se prueban en exceso, ya que todas las restricciones están en el DDL; lo único, como en el alta de documentos, que se requiere una actividad como mínimo. Por otro lado, la dedicada a las bajas se prueba tras las siguientes.

- Alta_Actividad_Empresa, Modifica_Actividad_Empresa y Baja_Actividad_Empresa

Para estos se reutilizan las pautas indicadas, y el código, de las pruebas sobre la tabla DescriptorDocumento.

- Alta_Persona, Modifica_Persona y Baja_Persona

En este caso las pruebas se centran en la correcta introducción de los valores mínimos necesarios para identificar unívocamente a una persona.

- En primer lugar se pone a prueba el procedimiento almacenado para altas. En esta ocasión se empieza por las llamadas que generan error, con todas las combinaciones de valores que resultan insuficientes para el hecho anterior; tras esto, se insertan tres registros en Persona correctos.
- Seguidamente se intentan modificaciones incorrectas sobre ellos, concretamente sobre el que solo tiene nombre y apellidos. Se trata de retirar el apellido para insertar domicilio incompleto y DNI erróneo. Luego se realizan algunas ediciones correctas; y, finalmente, se eliminan los tres registros creados.

- Alta_Agrupacion, Modifica_Agrupacion y Baja_Agrupacion
 - La prueba se inicia como la anterior, tratando de hacer llamadas donde faltan valores mínimos, aunque esto se define en el DDL. Luego se insertan correctos, con y sin identificador, y por último se eliminan.
 - Se hace prueba de eliminación incorrecta; naturalmente da error porque intentará eliminar un registro de una tabla en que no existe, ya que debe dar baja en el subtipo antes que en el supertipo Entidad, donde quizás si exista el identificador.
- Modifica_ID_Agrupacion
 - En este caso, como ocurría en uno de los anteriores, la prueba se podrá realizar más tarde; sin embargo, el control de la coherencia está implementado a través de la integridad referencial en el DDL. En el caso de los documentos y obras el cambio de identificadores se debe reflejar en otros campos.

De forma paralela a estas pruebas es fácil observar la evolución del estado de la tabla DWH_Entidad, hasta este punto se ha podido verificar el correcto funcionamiento de las funciones del *script* **08_DWH_SP_funcionesentidad.sql** centradas en dicha tabla.

10_pruebas_04_Evento.sql

Los dos primeros conjuntos de funciones se asemejan a muchos de los vistos, son sencillos sobre una tabla o en herencia sobre dos; seguidamente, el tercer conjunto es también similar, actuando sobre dos tablas por la relación de composición entre Premios y Concurso

- Alta_Evento, Modifica_Evento y Baja_Evento
Estos tres son simples, las condiciones son que el tipo de evento no sea 'Concurso' y que el nombre no se repita atendiendo a la capitalización. Como se hizo anteriormente no se dará la baja hasta probar la gestión de documentos, razón por la que se hace esto a continuación.
- Doc_Nuevo_A_Evento y Modifica_Documento_Evento,
Se crea un documento y se modifica. En realidad, estas funciones solo verifican la existencia de los identificadores de Evento y Documento y componen el Código de Referencia requerido por ISAD(G); por lo demás, toda la lógica está en las funciones internas dedicadas a Documento.
- Doc_Existente_A_Evento y Baja_Documento_Evento
Se crea un documento más, de ejemplo, que se asocia mediante al evento anterior mediante la primera de estas. En cuanto al referente a bajas, de estos dos, se utilizará como parte del proceso de baja del evento.
La estructura de la prueba definida para las bajas de eventos empieza por solicitar la baja de un evento que tiene documentos vinculados; esto lleva a error, luego se utiliza Baja_Documento_Evento sobre todos los documentos que se le han vinculado y, por último, se vuelve a solicitar la baja del evento, ya con éxito.
No se considera como prueba de eliminación de documentos todavía, porque los eventos no implican la existencia de un mínimo de un documento asociado, como es el caso de las obras y sus componentes.
- Alta_Concurso, Modifica_Concurso y Baja_Concurso
Las pruebas sobre estos son sencillas, mientras que no exista ningún registro en participa, por lo tanto se procede con algunas llamadas aleatorias a los tres; además, hace uso para el alta de Alta_Evento, por lo que este caso no lleva mayores pruebas funcionales. La modificación sí.
Por otro lado, nada más que probar acerca de los documentos asociables a concursos, se utilizarían los procedimientos anteriores.
- Alta_Premio_Concurso, Modifica_Premio_Concurso y Baja_Premio_Concurso
Casuística similar al anterior, si no hay agrupaciones inscritas en los concursos no se pueden forzar datos erróneos; por tanto, otros tantos registros creados y modificados sin más.

10_pruebas_05_AgrupacionAnio.sql

- Alta_AgrupacionAnio, Modifica_AgrupacionAnio, Baja_AgrupacionAnio
 - Para poner a prueba las inserciones y modificaciones se intentará establecer el mismo nombre a dos agrupaciones, aunque con diferente capitalización.

- El igual que en el caso de los documentos, se probará el funcionamiento de los booleanos de tipo marca o señal. Concretamente, que de indicar alguno de estos a *true* se haya introducido algún valor en fuentes contenga algún dato.
- Por otro lado, nuevamente en este escenario, no es posible probar la asignación de obras por interpretación, puesto que no se ha llegado a dichos procedimientos almacenados aún.
- Para finalizar, se prueba el funcionamiento de la creación de agrupaciones a través del alta de estas anuales. Es decir, se hacen llamadas sin el identificador de la agrupación, más enviando solo localidad o provincia, con intención se provocar el error. Luego se crea uno y se visualizan los registros resultantes.
- Modifica_ID_AgrupacionAnio
 - La única prueba posible es indicar un identificador mayor que 100000, en esta ocasión tampoco afecta al contenido de otros campos.

- Baja_Literaria_AgrupacionAnio, Baja_Musical_AgrupacionAnio y Baja_Tipo_AgrupacionAnio, y Verifica_AgrupacionAnio_Obra

Estas funciones será probadas una vez que se puedan asociar obras, más adelante a lo largo de las pruebas

- Alta_Agrupacion_Usa_Instrumento, Modifica_Agrupacion_Usa_Instrumento y Baja_Agrupacion_Usa_Instrumento

Las pruebas de estos tres son análogas a las ya vistas sobre tablas destinadas a modelar relaciones N:M, con los condicionantes que se vienen siguiendo para las relaciones definidas como adjetivas. Solo el instrumento puede cambiar en un registro de la tabla usa; en consecuencia, la prueba consiste en intentar cambiar un instrumento de agrupación (solicitar el cambio de la agrupación). Se realizan algunas altas, modificaciones y bajas.

En este punto se puede probar el correcto funcionamiento del procedimiento almacenado Modifica_ID_AgrupacionAnio en cuanto a la actualización en cascada.

- Alta_Rol

Las pruebas en esta función interna se hará a partir de las siguientes; concretamente, para verificar el correcto funcionamiento de esta, simplemente se solicita la creación de roles de tipo Orquesta y Voz para una misma persona, solo admitirá el primero; luego, para la misma persona y agrupación puede admitir varios roles de tipo Cargo.

- Alta_Cargo_AgrupacionAnio, Modifica_Cargo_AgrupacionAnio y Baja_Cargo_AgrupacionAnio

Esta tabla es la que admite varios pares de persona y agrupación, para tipos de cargo distinto; por tanto, se prueba la creación de varios por par. Por otro lado, se prueba la restricción de no tener varios cargos de 'Presidente', 'Director' o 'Representante legal' para una misma agrupación.

Por último, se mostrará que tras la eliminación de roles no queda el registro huérfano en la tabla rol.

Para los dos siguientes conjuntos, la restricción se controla por DDL mediante la restricción de unicidad y en el procedimiento Alta_Rol que no exista uno de cada clase para el par de persona y agrupación dado. A modo de prueba se utilizarán todos, pero la verificación refiere a esta última función centrada en la tabla rol.

- Alta_Voz_AgrupacionAnio, Modifica_Voz_AgrupacionAnio y Baja_Voz_AgrupacionAnio
- Alta_Orquesta_AgrupacionAnio, Modifica_Orquesta_AgrupacionAnio y Baja_Orquesta_AgrupacionAnio
- Alta_Participa, Modifica_Participa y Baja_Participa
 - La primera prueba consiste en verificar que no permite la participación de agrupaciones en concursos de diferente año al propio y que coincida la categoría.
 - Seguidamente, tras una inserción correcta, se trata de introducir una agrupación con el mismo premio para la misma modalidad en el mismo concurso.
 - Se prosigue con unas pruebas de modificación; en primer lugar, se intenta cambiar la agrupación, esto genera también un error; luego, se intenta cambiar la categoría y el año, una vez inscrita la agrupación a un concurso.

Tras estos último resta probar los dedicados a Premio y Concurso, puesto que a falta de participantes no se podía verificar la consistencia de todo el código. En cuanto a los concursos, se trata e intentar modifica la

categoría y el año a uno con agrupaciones inscritas; por otro lado, para los premios, intentar modificarles la modalidad

Antes de proseguir con las pruebas de los procedimientos almacenados dedicados a la BBDD, en este punto, debe haber información precalculada en las tablas de *data warehouse*: DWH_FuncionesEntidad y DWH_PremiosConcurso. En todos estos *scripts* de prueba se ha ido eliminando lo creado; no obstante, si se comenta todo el bloque final dedicado a la eliminación de los registros insertados, se puede hacer una consulta en estas tablas. Por ejemplo, al llegar al punto de pruebas donde se han insertado y modificado los roles (sin probar la participación aún), si se consulta la tabla DWH_FuncionesEntidad se obtienen los siguientes resultados:

5

6

7

8

SELECT * FROM general_acc.dwh_funciones_entidad;

Data Output

Explain

Messages

History

<input type="checkbox"/>	id integer	entidad integer	funcion character varying (100)	primero integer	segundo integer	tercero integer	cuarto integer	veces integer	anio_inicio integer	anio_fin integer	<input type="checkbox"/>
<input type="checkbox"/>	125	101366	Director	0	0	0	0	1	2011	2011	
<input type="checkbox"/>	129	101367	Pito de caña	0	0	0	0	1	2011	2011	
<input type="checkbox"/>	130	101366	Baritono	0	0	0	0	1	2011	2011	
<input type="checkbox"/>	131	101366	Postulante	0	0	0	0	1	2011	2011	

Tras las pruebas de los procedimientos almacenados de la tabla participa, donde se asignan premios a la agrupación de estas personas, se observan los siguientes contenidos:

6
7
8
9

SELECT * FROM general_acc.dwh_funciones_entidad;

Data Output Explain Messages History

	id integer	entidad integer	funcion character varying (100)	primero integer	segundo integer	tercero integer	cuarto integer	veces integer	anio_inicio integer	anio_fin integer
	477	101509	Director	0	0	0	0	1	2011	2011
	482	101509	Baritono	0	0	0	0	1	2011	2011
	483	101509	Postulante	0	1	0	0	1	2011	2011
	481	101510	Pito de caña	0	1	0	0	1	2011	2011

</

Como recién se indicaba, para poder ver estos datos hubo que comentar los bloques del *script* de pruebas, esto a su vez implica la eliminación manual de algunos registros residuales, de las personas de prueba concretamente.

05_SP_06_Obra.sql

Sobre este *script* de funciones internas no se harán las pruebas directamente, como se hizo en el caso de los documentos, sino que se procede directamente con las pruebas en los procedimientos almacenados dedicados a los subtipos de Obra. En principio, los tres siguientes *scripts* prueban las funciones de este, hasta por tres veces porque la implementación de todos ellos es idéntica. Solo los procedimientos Doc_Nuevo_a_Componente y Baja_Documento_Componente quedarán sin probar, será durante las pruebas centradas en las tabla Letra, Musica y Disenio cuando será probados.

Por otro lado, para hacer una verificación efectiva, en el primero de os tres *scripts* de pruebas siguientes, se atenderá más específicamente a los condicionantes de las funciones que se vienen tratando como internas; luego, los dos restantes, relativos a las obras musicales y gráficas, tendrán un conjunto más reducido de pruebas.

10_pruebas_061_ObraLiteraria.sql

- Alta_Literaria

Se comienza por intentar introducir obras con valores que generarán error, casi todas las restricciones están definidas en el DDL, excepto:

- La composición del Código de Referencia; por tanto, mediante el uso de este procedimiento almacenado se probará el funcionamiento de Verifica_Codigo_Referencia de y Componer_Codigo_Referencia, el cual ya está verificando los intermedios. Para ello se indican los identificadores, en caso contrario se compondría.

Se introducen, primero, valores aleatorios para el Código de Referencia y, luego, un valor que debe aceptar; con esto se crearía un documento y la obra, pero se tienen erróneos los siguientes.

A continuación, para verificar la función interna Componer_Codigo_Referencia se crea un segundo documento para la misma haciendo uso de Doc_Nuevo_a_Literaria, e indicándole el identificador del nuevo documento, con objeto de proseguir con la eliminación del mismo haciendo uso de Baja_Documento_Literaria. Antes de la baja se muestran los documentos resultantes.

- El Nivel de Descripción, solo las gráficas admiten como valor la 'Unidad documental simple' si es de tipo vídeo; por tanto, introducirle este valor es la prueba.
- El tipo de documento que no admite son las imágenes.

Tras estas pruebas se inserta un registro correcto acorde a la agrupación que se creará para las pruebas de vinculación. Una de las cuestiones que ya se puede verificar es la coincidencia entre el nombre de la obra y el de la agrupación; por tanto, se modifica la agrupación para asociarlas, al igual que no se pueden asociar obras con diferente nombre a las agrupaciones, no se les puede cambiar el nombre a las obras una vez asociadas. Con estas últimas pruebas se da por probado el procedimiento almacenado Verifica_AgrupacionAnio_Obra. Finalmente, tras todo lo anterior, se verifica el funcionamiento de Baja_Literaria_AgrupacionAnio.

Por extensión, se da por comprobado el comportamiento de Alta_Obra y de Doc_Nuevo_A_Obra, utilizado por el anterior.

- Modifica_Literaria

En este caso se intenta modificar de nuevo el Código de Referencia, el Nivel de Descripción y el tipo de documento, que vuelven a ser las restricciones a los datos dentro del nivel semántico. Con estas pruebas, además, queda verificado el funcionamiento de Modifica_Obra –que en realidad hace uso del verificador del Código de Referencia y del procedimiento almacenado Modifica_Documento–.

- Baja_Literaria

En este caso toda la implementación está en Baja_Obra, que a su vez hace uso de Baja_Documento, en esencia lo que hace es desvincular la obra de todo documento. El primero, si existe el registro en la tabla DocumentoObra, lo elimina y llama al siguiente, el cual ya ha sido puesto a prueba anteriormente –no admite la eliminación si es el único vínculo del documento en la BBDD y tiene algún representante tangible referenciado–.

En consecuencia, la prueba consiste en verificar que no existe ya la obra en la BBDD, resta decir que si el documento aún mantiene relaciones con otro objeto no habrá sido eliminado; sin embargo, este no es el caso.

- Modifica_Signatura_Literaria

Este procedimiento será probado en el tercero y último *script* de entre estos dedicados a las obras, ya que uno de los dos errores de nivel semántico que lanza se debe al tipo de obra; es decir, solo será lanzado de enviarle una signatura que no sea literaria.

- Doc_Nuevo_a_Literaria, Doc_Existente_a_Literaria y Baja_Documento_Literaria

El único que aún no se ha verificado es el segundo, y se hará en el siguiente *script*.

10_pruebas_062_ObraMusical.sql y 10_pruebas_063_ObraGrafica.sql

Estos dos *scripts* contienen un conjunto reducido de las pruebas del anterior, puesto que se dan por verificadas las funciones internas definidas en 05_06_Obra.sql. Además, se incluye el uso de

Modifica_Signatura_Literaria con una signatura musical, luego se muestran los resultados de la modificación.

Por otro lado, como especificidades de cada una se hacen pruebas en las de altas respecto a el nivel de descripción y el tipo de documento.

- Alta_Musical
 - El Nivel de Descripción, tampoco admiten como valor la 'Unidad documental simple'; de nuevo, introducirle este valor es la prueba.
 - El tipo de documento que no admite son las imágenes.

También en este caso se prosigue con la verificación de la coincidencia entre el nombre de la obra y el de la agrupación. Se modifica la agrupación para asociarlas y luego se verifica el funcionamiento de Baja_Musical_AgrupacionAnio.

Tras estas pruebas se crea un registro válido con un documento compatible a las literarias, de tipo audio, para poder verificar el funcionamiento de Doc_Existente_a_Literaria. Seguidamente se da la baja mediante el uso de Baja_Documento_Literaria, se modifica convirtiéndole el tipo a 'Imagen' para, finalmente, volver a probar con Doc_Existente_a_Literaria.

- Alta_Grafica
 - El Nivel de Descripción, admite como valor la 'Unidad documental simple' si es de tipo vídeo; por tanto, introducirle este valor en nivel de descripción sin el tipo de documento igual a 'Video' es la prueba.
 - El tipo de documento que no admite es 'Audio'.

Por último, se modifica la agrupación para asociarlas y luego se verifica el funcionamiento de Baja_Grafica_AgrupacionAnio.

Una vez realizado estas pruebas, de no haber eliminado nada, se tienen varios datos en las tablas de *data warehouse*. De hecho, las pruebas contenidas en estos *scripts* han sido planteadas con objeto de poder verificar el comportamiento de las funcionalidades estadísticas; por tanto, simplemente comentando los tres bloques de eliminación de los registros insertados, se pueden observar los siguientes estados en el DWH. No obstante, para ir conformando un *script* que pruebe el DWH se toman estos bloques, más el de el *script* de pruebas centrado en AgrupacionAnio.

También en este estado se procede a probar el efecto de Modifica_Signatura_Literaria, ya que se tienen varios documentos de obra insertados.

10_pruebas_071_Letra.sql

Al igual que se ha hecho en las obras literarias, en este primer *script* de componente de obra, se hará un mayor esfuerzo en la validación del comportamiento de los procedimientos en **05_SP_06_Obra.sql** aún no probados, __Doc_Nuevo_a_Componente y __Baja_Documento_Componente.

- Alta_Letra, Modifica_Letra y Baja_Letra
 - Respecto a la gestión de los documento que se le asocian a una letra, el mínimo y los demás, no hay restricciones más allá del tipo de documento que admiten; de hecho, solo es necesario tratar de introducir un documento de tipo imagen y uno de nivel de descripción de valor 'unidad documental compuesta'. Estos dos hechos se gestionan directamente en estos procedimientos, por la especialización del tipo de entidad, Letra. Luego los procedimientos __Doc_Nuevo_a_Componente y __Baja_Documento_Componente simplemente verifican el Código de referencia, en el primer caso, y acti seguido llaman a las funciones __Alta_Documento y __Baja_Documento.
 - Por otro lado, se mostrará el contenido de la tabla Documento, tras la inserción de varias letras, con objeto de verificar que la signaturización a nivel de componente de obra es correcta; es decir, que se da una signatura secuencial con respecto de las ya registradas dentro de la obra.
 - Resta verificar que el alta obliga a la introducción del tema, al igual que se hacía en las altas de Documento respecto a los descriptores, y que no se admiten dos letras de género popurrí y presentación por cada obra.
- __Alta_Documento_Letra, Doc_Existente_a_Letra y Baja_Documento_Letra
 - El primero de estos queda verificado por los anteriores, se procederá con el uso de los otros dos; de nuevo, entra en juego el tipo de documento como restricción. En lo referente a las bajas,

al igual que para la gestión de los documentos descriptores de obras, se intenta la baja para controlar que no haya documentos aislados.

- Doc_Nuevo_a_Letra
 - La lógica implementada en este procedimiento también queda ya verificada; no bstante se realizan un par de llamadas.
- Alta_Tema_Letra, Modifica_Tema_Letra y Baja_Tema_Letra
 - Para probar estos simplemente hay que utilizarlos con temas que no existen, pares que ya se han vinculado y el intento de eliminación del último par de referencias que aluda a un letra determinada; puesto que es obligatorio tener marcadas las letras con un mínimo de un tema.

10_pruebas_072_Musica.sql

Para poner a prueba el código contenido en el *script* **05_SP_072_Musica.sql** se hace uso de las pruebas descritas en el bloque inmediatamente anterior; exceptuando las referentes a el etiquetado mediante temas y las que fuerzan el tipo de documento, ya que en este caso los admite todos.

10_pruebas_073_Disenio.sql

Para este último conjunto de pruebas se vuelve a hacer uso de la batería utilizada para los procedimientos de la tabla Letra, en este caso se pueden reutilizar los del etiquetado mediante temas y se añaden los dedicados a la relación crea entre Entidad y Disenio.

- Alta_Crea_Disenio, Modifica_Crea_Disenio y Baja_Crea_Disenio
 - La única restricción a forzar es que no se cree una relación ya existente entre pares. En el caso de la modificación, esta es la única excepción de tabla N:M donde se puede modificar cualquiera de las dos referencias; no obstante, no representa ninguna complejidad.

Pruebas al *data warehouse*

Como forma de automatización se aprovecha parte de los bloques DO utilizados en las pruebas anteriores; concretamente se crea un último *script* que contiene todas las inserciones correctas realizadas y, además, se omiten las eliminaciones. Con esto se rellenan todas las tablas de hechos

Antes de ejecutar este último *script*, de nombre **10_pruebas_080_DWH.sql**, teniendo en cuenta que para todas las inserciones de las pruebas se ha dado un borrado posterior, se muestra el estado de cada una de las tablas a partir de las vistas de cada una.

Soporte al proceso documental de las agrupaciones

Estado antes de las inserciones.

carnaval on postgres@grandullon

1

2

```
SELECT * FROM general_accs."DWH_SoporteDocumental";
```

Data Output

Explain

Messages

History

<div><div></div><div>anio</div><div>integer</div></div>	<div><div>agrupacion</div><div>character varying</div></div>	<div><div>tipo_obra</div><div>general_accs.enum_obra</div></div>	<div><div>tipo_documento</div><div>general_accs.enum_tipodocumento</div></div>	<div><div>tipo_soporte</div><div>general_accs.enum_tiposoporte</div></div>	<div><div>total</div><div>integer</div></div>
---	--	--	--	--	---

Tras las inserciones se han creado los siguientes registros.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_accs."DWH_SoporteDocumental";
2
```

anio integer	agrupacion character varying	tipo_obra general_accs.enum_obra	tipo_documento general_accs.enum_tipodocumento	tipo_soporte general_accs.enum_tiposoporte	total integer
2011	En las mismas de siempre	Literaria	Texto	Papel	1
2011	En las mismas de siempre	Literaria	Video	VHS	1
2011	En las mismas de siempre	Musical	Audio	Casete	1
2011	En las mismas de siempre	Musical	Video	VHS	1
2011	En las mismas de siempre	Grafica	Video	VHS	1
2011	En las mismas de siempre	Grafica	Expediente	Papel	1

Temas recurrentes en letras por año y modalidad

Estado antes de las inserciones.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_accs."DWH_TemasRecurrentes_Anio";
2
```

tema character varying	tipo_obra general_accs.enum_obra	anio integer	total bigint
LA BANDURRIA	Literaria	2011	1
NUEVO CEMENTERIO	Literaria	2011	2
MONSERRAT CABALLE	Literaria	2011	1
RESURGIR DEL TANGUILLO	Literaria	2011	1
ANTONIO BURGOS BERLINCHON	Literaria	2011	2
COMIDA PARA AFRICA	Literaria	2011	1
RELOJ DE AÑO NUEVO	Literaria	2011	2
MEDICINAS	Literaria	2011	1
CALETA	Literaria	2011	2
EMBASES	Literaria	2011	1
CARMEN ROMERO	Grafica	2011	1
ARREGLAR LA SOCIEDAD	Literaria	2011	1
EURO	Literaria	2011	1
MIGUEL INDURAIN	Literaria	2011	1
SOCIALISTAS	Literaria	2011	1

Total query runtime: 77 msec.
43 rows retrieved.

Tras la ejecución del *script* de inserciones la tabla de temas no recibe ningún dato, que es lo correcto, puesto que las letras y diseños registrados no pertenecen a una obra que esté siendo interpretada.

Por tanto, para forzar el comportamiento, se crea una agrupación con el nombre de las tres últimas obras de prueba, correspondientes a la agrupación de nombre "La Eternidad". En la misma creación se le adscriben las tres obras que fueran generadas para las pruebas de inserción de componentes de obras (letras, músicas y diseños). Con esto se debe cumplimentar la tabla de hechos relativos a soporte documental y la de los temas.

carnaval on postgres@grandullon

1

2

SELECT * FROM general_accs."DWH_SoporteDocumental";

Data Output

Explain

Messages

History

	anio integer	agrupacion character varying	tipo_obra general_accs.enum_obra	tipo_documento general_accs.enum_tipodocumento	tipo_soporte general_accs.enum_tiposoposte	total integer
<input type="checkbox"/>	2011	En las mismas de siempre	Literaria	Texto	Papel	1
<input type="checkbox"/>	2011	En las mismas de siempre	Literaria	Video	VHS	1
<input type="checkbox"/>	2011	En las mismas de siempre	Musical	Audio	Casete	1
<input type="checkbox"/>	2011	En las mismas de siempre	Musical	Video	VHS	1
<input type="checkbox"/>	2011	En las mismas de siempre	Grafica	Video	VHS	1
<input type="checkbox"/>	2011	En las mismas de siempre	Grafica	Expediente	Papel	1
<input type="checkbox"/>	2017	La Eternidad	Literaria	Texto	Papel	1
<input type="checkbox"/>	2017	La Eternidad	Literaria	Audio	Electrónico	1
<input type="checkbox"/>	2017	La Eternidad	Literaria	Video	Electrónico	1
<input type="checkbox"/>	2017	La Eternidad	Musical	Imagen	Electrónico	1
<input type="checkbox"/>	2017	La Eternidad	Musical	Audio	Electrónico	1
<input type="checkbox"/>	2017	La Eternidad	Grafica	Video	Electrónico	1
<input type="checkbox"/>	2017	La Eternidad	Grafica	Video	DVD	1

Para los temas se tenían 43 registros, agregados por año, y ahora se tienen 98.

carnaval on postgres@grandullon

1

SELECT * FROM general_accs."DWH_TemasRecurrentes_Año";

2

Data Output

Explain

Messages

History

	tema character varying	tipo_obra general_accs.enum_obra	anio integer	total bigint	
<input type="checkbox"/>	LA BANDURRIA	Literaria	2011	1	
<input type="checkbox"/>	NUEVO CEMENTERIO	Literaria	2011	2	
<input type="checkbox"/>	MONSERRAT CABALLE	Literaria	2011	1	
<input type="checkbox"/>	RESURGIR DEL TANGUILLO	Literaria	2011	1	
<input type="checkbox"/>	ANTONIO BURGOS BERLINCHON	Literaria	2011	2	
<input type="checkbox"/>	COMIDA PARA AFRICA	Literaria	2011	1	
<input type="checkbox"/>	RELOJ DE AÑO NUEVO	Literaria	2011	2	
<input type="checkbox"/>	APARCAMIENTOS	Grafica	2017	1	
<input type="checkbox"/>	MEDICINAS	Literaria	2011	1	
<input type="checkbox"/>	CALETA	Literaria	2011	2	
<input type="checkbox"/>	EMBATES	Literaria	2011	1	
<input type="checkbox"/>	CARMEN ROMERO	Grafica	2011	1	
<input type="checkbox"/>	CALETA	Literaria	2017	1	
<input type="checkbox"/>	ARREGLAR LA SOCIEDAD	Literaria	2011	1	
<input type="checkbox"/>	CAJERO AUTOMATICO	Literaria	2017	1	

Total query runtime: 98 msec.

55 rows retrieved.

Agrupaciones ganadoras por concurso y año

Estado antes de las inserciones.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_acc."DWH_PremiosConcurso";
2
```

	anio integer	concurso text	categoria character varying (20)	modalidad character varying (20)	premio integer	agrupacion text	autor chara
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	1	CIEN MARIPOSAS (CADIZ)	[null]
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	2	CHICAGO DE NOCHE (CADIZ)	[null]
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	3	FIGURITAS (CADIZ)	[null]
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	4	LA GRAN RESERVA (CADIZ)	[null]
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Comparsa	1	HOGUERAS (CADIZ)	[null]
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Comparsa	2	CAÑA Y MIMBRE (CADIZ)	[null]
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	1	NI PICAOS PAGAN (CADIZ)	[null]
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	2	SI TIRITO TATARI TARITO (CADIZ)	[null]
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	3	NOS QUEJAMOS DE PIEDRA (CADIZ)	[null]
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	4	EL RAYO VATICANO (CADIZ)	[null]
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Comparsa	1	CENTENARIOS (CADIZ)	[null]
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Comparsa	2	DE IDA Y VUELTA (CADIZ)	[null]

Total query runtime: 90 msec.
12 rows retrieved.

Estado tras la ejecución del script.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_acc."DWH_PremiosConcurso";
2
3
4
5
```

	anio integer	concurso text	categoria character varying (20)	modalidad character varying (20)	premio integer	agrupacion text	a c
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	1	CIEN MARIPOSAS (CADIZ)	
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	2	CHICAGO DE NOCHE (CADIZ)	
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	3	FIGURITAS (CADIZ)	
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Chirigota	4	LA GRAN RESERVA (CADIZ)	
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Comparsa	1	HOGUERAS (CADIZ)	
<input type="checkbox"/>	2011	COAC (Cádiz)	Adultos	Comparsa	2	CAÑA Y MIMBRE (CADIZ)	
<input type="checkbox"/>	2011	COAC (Localidad 1)	Adultos	Comparsa	2	En las mismas de siempre (Localid...	
<input type="checkbox"/>	2011	COAC (Localidad 1)	Adultos	Comparsa	3	HOGUERAS (CADIZ)	
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	1	NI PICAOS PAGAN (CADIZ)	
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	2	SI TIRITO TATARI TARITO (CADIZ)	
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	3	NOS QUEJAMOS DE PIEDRA (CA...	
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Chirigota	4	EL RAYO VATICANO (CADIZ)	
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Comparsa	1	CENTENARIOS (CADIZ)	
<input type="checkbox"/>	2015	COAC (Cádiz)	Adultos	Comparsa	2	DE IDA Y VUELTA (CADIZ)	

Dedicación oficial al Carnaval por persona

Estado antes de las inserciones.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_accs."DWH_DedicacionEntidad";
```

Data Output Explain Messages History

	localidad character varying (50)	nombre character varying (200)	tipo_entidad general_accs.enum_entidad	funcion character varying (100)	veces integer	anio_inicio integer	anio_fin integer
<input type="checkbox"/>	ALGECIRAS	SALVADOR CASAS (EL BORI)	Persona	Director	1	2011	
<input type="checkbox"/>	CADIZ	FRANCISCO ABEIJON (CARAPALO)	Persona	Diseño Disfraz	2	2011	
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Director	1	2011	
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Diseño Caracterización	2	2011	
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Guitarra	1	2011	
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Letra Pasodoble	1	2011	
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Octavilla	1	2011	
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Segunda	1	2011	
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Baritono	1	2011	
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Director	1	2011	
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Guitarra	1	2011	
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Cuplé	1	2011	
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Pasodoble			
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Popurrí			

Total query runtime: 193 msec.
55 rows retrieved.

Estado tras la ejecución del *script*. En este caso se desplaza el *scroll* para poder mostrar los que son los registros de prueba, además tienen marcado el *checkbox*.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_accs."DWH_PremiosEntidad";
```

Data Output Explain Messages History

	localidad character var...	nombre character varying (200)	tipo_entidad general_acc...	funcion character varying (100)	primero integer	segundo integer	tercero integer	cuarto integer
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Guitarra	1	0	1	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Letra Pasodoble	0	0	0	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Letra Popurrí	0	0	0	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Cuplé	0	0	0	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Pasodoble	0	0	0	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Popurrí	0	0	0	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Presentación	0	0	0	0
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Segunda	0	0	0	0
<input type="checkbox"/>	[null]	Desconocido - -	Persona	Guitarra	0	1	0	0
<input type="checkbox"/>	[null]	Desconocido - -	Persona	Segunda	0	0	0	0
<input checked="" type="checkbox"/>	[null]	Antonio Martínez Ares	Persona	Director	0	1	0	0
<input checked="" type="checkbox"/>	[null]	Antonio Martínez Ares	Persona	Octavilla	0	0	0	0
<input checked="" type="checkbox"/>	[null]	Antonio Martínez Ares	Persona	Representante legal	0	0	0	0
<input checked="" type="checkbox"/>	[null]	Micotondria Cortés Serrano	Persona	Guitarra	0	1	0	0

Premios por entidad y función

Estado antes de las inserciones.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_accs."DWH_PremiosEntidad";
```

2

Data Output Explain Messages History

	localidad character varying (50)	nombre character varying (200)	tipo_entidad general_acc...	funcion character varying (100)	primero integer	segundo integer	tercero integer	cuarto integer
<input type="checkbox"/>	ALGECIRAS	SALVADOR CASAS (EL BORI)	Persona	Director	0	0	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO ABEIJON (CARAPALO)	Persona	Diseño Disfraz	1	0	0	0
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Director	0	0	0	1
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Diseño Caracterización	1	0	0	0
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Guitarra	0	0	1	0
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Letra Pasodoble	0	0	0	0
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Octavilla	0	0	0	0
<input type="checkbox"/>	CADIZ	EMILIO ALVAREZ (EMILIN)	Persona	Segunda	0	0	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Baritono	0	0	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Director	0	0	1	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Guitarra	0	1	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Cuplé	0	0	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Pasodoble	1	0	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Popurrí	0	0	0	0
<input type="checkbox"/>	CADIZ	FRANCISCO BARROSO (EL ASTRO)	Persona	Música Presentación	0	0	0	0

Estado tras la ejecución del script.

carnaval on postgres@grandullon

```
1 SELECT * FROM general_accs."DWH_DedicacionEntidad";
```

2

3

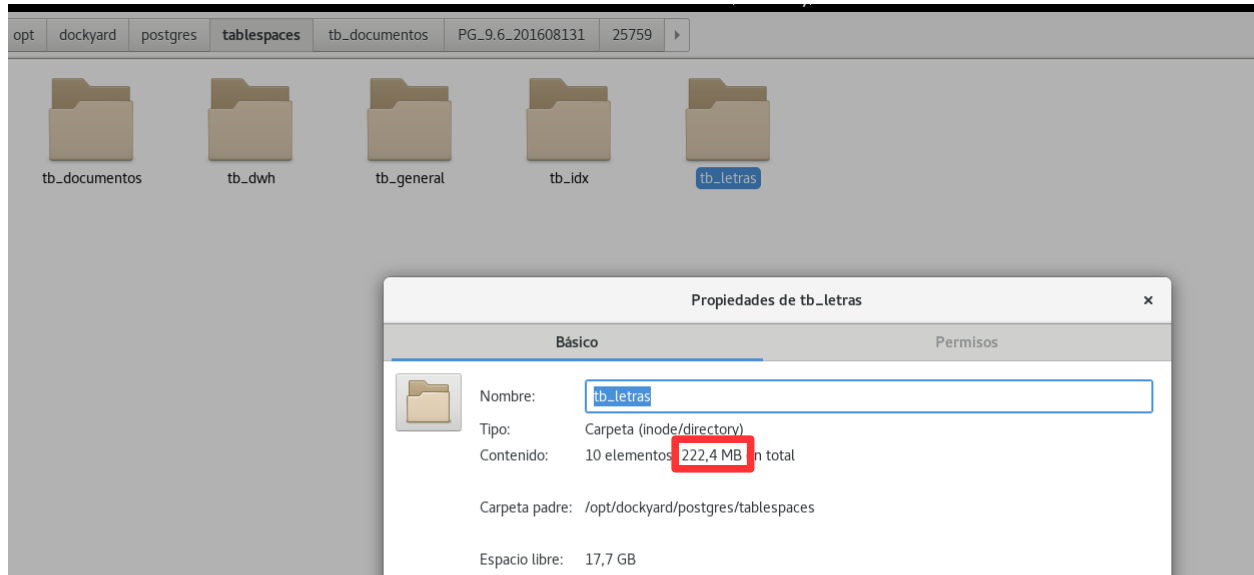
4

Data Output Explain Messages History

	localidad character ...	nombre character varying (200)	tipo_entidad general_accs.enum_entidad	funcion character varying (100)	veces integer	anio_inicio integer	anio_fin integer
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Guitarra	1	2011	2011
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Letra Pasodoble	3	[null]	[null]
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Letra Popurrí	1	[null]	[null]
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Cuplé	2	[null]	[null]
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Pasodoble	2	[null]	[null]
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Popurrí	2	[null]	[null]
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Música Presentación	2	[null]	[null]
<input type="checkbox"/>	[null]	Anónimo - -	Persona	Segunda	1	2011	2011
<input type="checkbox"/>	[null]	Desconocido - -	Persona	Guitarra	1	2011	2011
<input type="checkbox"/>	[null]	Desconocido - -	Persona	Segunda	1	2011	2011
<input type="checkbox"/>	[null]	Antonio Martínez Ares	Persona	Director	1	2011	2011
<input type="checkbox"/>	[null]	Antonio Martínez Ares	Persona	Octavilla	1	2011	2011
<input checked="" type="checkbox"/>	[null]	Antonio Martínez Ares	Persona	Representante legal	1	2011	2011
<input checked="" type="checkbox"/>	[null]	Micotondria Cortés Serrano	Persona	Guitarra	1	2011	2011

Pruebas de rendimiento

Esta prueba requiere de la inserción de varios miles de registros en una de las tablas que contienen campos de texto de longitud indeterminada, y posteriormente hacer filtrados en el mismo. Se ha elegido la tabla Letra, luego se ha creado un *script* para de inserciones en esta tabla. Este *script* contiene unas 500 inserciones y se ejecuta unas docenas de veces. Tomando como ventaja el hecho de tener esta tabla en un *tablespace* dedicado, se muestra a continuación el tamaño del mismo en disco:



A continuación se muestra la consulta de pruebas, en ella se observa el tiempo de respuesta y el número de registros devueltos.

camaval on postgres@grandullon

```

1 SELECT id, obra_literaria, signature, autoria_letra, genero, transcripcion
2 FROM general_acc.lettra
3 WHERE upper(transcripcion) LIKE '% ' || upper('falla') || '%';

```

id integer	obra_litera... integer	signature integer	autoria_letra integer	genero general_acc.enum_genero_letra	transcripcion text
1257	15	908	100041	Pasodoble	POPURRI
1258	15	909	100041	Pasodoble	POPURRI
1260	15	911	100041	Pasodoble	Si caminito del Falla.
368	42	88	101149	Pasodoble	POPURRI
369	42	90	101149	Pasodoble	POPURRI
370	42	91	101149	Pasodoble	POPURRI
371	42	92	101149	Pasodoble	POPURRI
372	42	93	100101	Pasodoble	POPURRI
373	42	94	101040	Pasodoble	POPURRI
374	42	95	101189	Pasodoble	POPURRI
48	22	607	100126	Pasodoble	POPURRI
49	22	609	100126	Pasodoble	POPURRI
50	22	610	100126	Pasodoble	POPURRI
51	22	611	100126	Pasodoble	POPURRI
52	22	612	100028	Pasodoble	POPURRI
53	22	613	101168	Pasodoble	POPURRI
54	26	282	101129	Pasodoble	POPURRI

Total query runtime: 9 secs.
28865 rows retrieved.

Por último, pasando el cursor sobre el campo en PgAdmin se accede al valor del mismo, se muestra en la siguiente captura de pantalla la palabra buscada.

carnaval on postgres@grandullon

```

1 SELECT id, obra_literaria, signature, autoria_letra, genero, transcrip
2 FROM general_accs.lettra
3 WHERE upper(transcripcion) LIKE '%' || upper('falla') || '%';

```

Data Output Explain Messages History

	id integer	obra_litera... integer	signature integer	autoria_letra integer	genero general_accs.enum_genero_letra	transcrip text
<input type="checkbox"/>	1257	15	908	100041	Pasodoble	POPURRI
<input type="checkbox"/>	1258	15	909	100041	Pasodoble	POPURRI
<input type="checkbox"/>	1260	15	911	100041	Pasodoble	Si camina...
<input type="checkbox"/>	15	21	205	101164	Pasodoble	Por mas...
<input type="checkbox"/>	19	21	210	101164	Cuple	Nosotros...
<input type="checkbox"/>	23	21	214	100149	Cuple	Van a por...
<input type="checkbox"/>	368	42	88	101149	Pasodoble	POPURRI
<input type="checkbox"/>	369	42	90	101149	Pasodoble	POPURRI
<input type="checkbox"/>	370	42	91	101149	Pasodoble	POPURRI
<input type="checkbox"/>	371	42	92	101149	Pasodoble	POPURRI
<input type="checkbox"/>	28	24	396	100828	Pasodoble	Querido Don Carlos Día
<input type="checkbox"/>	372	42	93	100101	Pasodoble	POPURRI

más que nunca te prometo que al tercer día cantarás. Siémbrete y crece con tu pueblo frente a la gloria de su mar. Aquí no hay fin ni hay principio. No hay presente ni pasado. No hay futuro. ¡Ay, mi niño! Aunque tampoco lo hay abajo. ¡Barquerooooo! Alguien me llama en la otra orilla del cielo. Es una niña bonita. ¿Paga dinero? ¡Paga dinero! Amigo mío déjame que te diga, volver a Cádiz, a Cádiz, bien que vale la vida. Hoy habrá una gran fiesta en el otro barrio, almas antiguas, tu gente, ¡Corre! te están reclamando. Siémbrete y crece compañero y recuerda a quien dejaste llorando y llorando en la tierra. El eterno recuerdo. Te buscaré por los mundos perdidos de la eternidad. Te buscaré con mi triste figura vagando entre la soledad. Buscaré no sé cuándo ni dónde, te encontraré. Yo sé que tú aunque dormida me miras, lo sé, yo lo sé. Tanto te echo de menos que ya no me duele el dolor. Duele este inmenso vacío de uno que un día fueron dos. Cuando más falta me hacías volaste, eso sí, aquí sigues prendida, encendida y ardiendo, latiendo muy dentro de mí... ¡Amor, ay dónde estás! Dices... que debo aprender a vivir... caminar... Dices... que debo dejar tu memoria... al mar... Yo solo escucho una voz y es la tuya muy lejos de aquí, me espero a la noche y te invento y salimos los dos por ahí. Te buscaré y venderé lo que quede de mí a cualquier dios. Claro que existe el infierno el infierno es donde vivo yo. Te buscaré en esta vida, en la otra y jamás lloraré. Hasta la muerte, qué tonta, lo sabe... yo te encontraré. Te buscaré por los mundos perdidos de la eternidad... La historia eterna. Tu vasta historia te contempla en una esquinita del sur. Te pongo al día caminante para que sepas quién eres tú. Eres múltiple de arena, el vaso de la marea siempre llena. Así lo quisieron los dioses para tragarnos mejor las penas. Una bailaora romana y esclava, la mezquita del parque cristiano. Manuel de Falla dando las horas al vuelo libre del panavato. Maremoto, explosión. La lirio. ¡Dioses...