

TECNOLOGICO NACIONAL DE MEXICO CAMPUS CULIACAN

INGENIERIA EN SISTEMAS COMPUTACIONALES



MATERIA

ADMINISTRACION DE BASE DE DATOS

INTEGRANTES

LOZANO CORVERA DANIEL ANTONIO

GARCIA AGUILAR JOSE ALFREDO

MAESTRO

DANIEL ESPARZA SOTO

FECHA

22-OCTUBRE-2022

TAREA 2 COPIAS DE SEGURIDAD EN 3 GESTORES DE BASE DE DATOS

INVESTIGAR EN LOS SISTEMAS GESTORES DE BASE DE DATOS MYSQL, POSTGRES Y ORACLE C20 LO SIGUIENTE:

- **LOS MECANISMOS DE COPIA DE SEGURIDAD CON LOS QUE CUENTA**

MYSQL

4 tipos de copias de seguridad en una base de datos para MySQL:

Los tipos de copia de seguridad de datos se dividen principalmente en los siguientes grupos según sus propias características.

- Copia de seguridad completa: se refiere a la copia de seguridad de todo el conjunto de datos (es decir, la base de datos completa)
- Copia de seguridad parcial: se refiere a la copia de seguridad de parte del conjunto de datos (por ejemplo: solo se realiza una copia de seguridad de una tabla)

Y algunas copias de seguridad se dividen en los siguientes dos

- Copia de seguridad incremental: se refiere a la copia de seguridad de datos que ha cambiado desde la última copia de seguridad (incremental o completa)
 - Características:
 - Ahorra espacio
 - Restaura problemas
- Copia de seguridad diferencial: se refiere a la copia de seguridad de los datos que han cambiado desde la última copia de seguridad completa
 - Características:
 - El espacio perdido y la restauración son más simples que la copia de seguridad incremental

Datos de respaldo de MySQL

En MySQL, generalmente hacemos copias de seguridad de los datos de varias maneras.

- Copia de seguridad en caliente: Significa que cuando se realiza una copia de seguridad de la base de datos, se puede realizar la operación de lectura de la base de datos, pero no se puede realizar la operación de escritura
- Respaldo cálido: Significa que cuando se realiza una copia de seguridad de la base de datos, las operaciones de lectura y escritura de la base de datos no se ven afectadas
- Copia de seguridad en frío: significa que cuando se realiza una copia de seguridad de la base de datos, la base de datos no se puede leer ni escribir, es decir, la base de datos está fuera de línea

Necesitamos considerar cómo hacer una copia de seguridad de los datos en la base de datos MySQL después de considerar la copia de seguridad de datos, después de que la base de datos se esté ejecutando

POSTGRESQL

2 tipos de copias de seguridad en una base de datos para PostgreSQL:

- Tipo físico:
 - Las copias de seguridad a nivel de sistema de archivos (o copias de seguridad físicas) son esencialmente simples instantáneas de todos los archivos de la base de datos.
 - Pero no es tan fácil como podría parecer porque los archivos dentro de una base de datos suelen sufrir constantes reescrituras y cambios.
 - La copia de seguridad de la base de datos PostgreSQL se basa en dos métodos clave: la continuidad en el archivo y la recuperación puntual.
 - Por el bien de la consistencia, las copias de seguridad necesitan tener alguna forma de saber con seguridad que el proceso de copia de seguridad está copiando toda la base de datos o no está cambiando nada y dejando la base de datos como está.
 - PostgreSQL cuenta con una tecnología de registro de escritura anticipada: los segmentos de registro de escritura anticipada (WAL) son exactamente los que se están respaldando durante el proceso de archivado en curso. La información que se almacena en esos archivos permite tanto una recuperación más fácil después de un accidente como una mejor consistencia de los datos.
 - No es raro que las bases de datos sufran algunos cambios en el proceso de una copia de seguridad del sistema de archivos, pero algunos de esos cambios podrían dañar algunas partes de una copia de seguridad o hacerla irreparable en su conjunto. Para evitar estas nefastas consecuencias, PostgreSQL dispone de una API de bajo nivel para el proceso de copia de seguridad física. El uso de `pg_start_backups()` y `pg_stop_backup()` antes y después del proceso, respectivamente, asegura que no se realicen cambios peligrosos en la base de datos durante el proceso de copia de seguridad.
- Tipo lógico:
 - Algo parcialmente diferente a una copia de seguridad física es el volcado SQL (o copia de seguridad lógica).
 - Como su nombre indica, esta copia de seguridad consiste en utilizar los comandos de copia de seguridad de PostgreSQL para crear la estructura básica de la base de datos y luego llenarla con todos los datos.
 - Un volcado SQL representa sistemáticamente un estado de calma de la base de datos en un momento dado (ya que el proceso de volcado es casi el mismo que el de cualquier otra sesión de la base de datos).
 - El proceso es el siguiente: el software recorre todas las tablas disponibles y obtiene todas las filas. No es realmente complicado, pero es lo suficientemente inteligente como para mantener el orden de las cosas para restaurar todo tal y como se hizo la copia de seguridad, con todas las conexiones y demás.

- El uso de volcados SQL significa que tendrá que acostumbrarse a que los datos de varias tablas estén repartidos por la línea de tiempo. Significa que una tabla puede tener una marca de tiempo A, y la otra puede estar hecha en la marca de tiempo B. Esto es útil para tenerlo en cuenta en caso de que haya algún tipo de reglas en la base de datos sobre cómo deben interactuar las filas y las tablas entre sí.

Las herramientas a utilizar para realizar los backup en PostgreSQL, dependerán del tipo de copia de seguridad y de si es necesario almacenar en el respaldo las transacciones en ejecución.

ORACLE

- 2 tipos de copias de seguridad en una base de datos para Oracle:
 - Tipo físico: Se realizan cuando se copian los ficheros que soportan la BD.
 - Backup de SO: El más sencillo de ejecutar, aunque consume mucho tiempo y hace inaccesible al sistema mientras se lleva a cabo
 - Backup en frío: Implican parar la BD en modo normal y copiar todos los ficheros sobre los que se asienta. Antes de parar la BD hay que parar también todas las aplicaciones que estén trabajando con la BD
 - Backup en caliente: Se realiza mientras la BD está abierta y funcionando en modo ARCHIVELOG. Habrá que tener cuidado de realizarlo cuando la carga de la BD sea pequeña. Este tipo de Backup consiste en copiar todos los ficheros correspondientes a un tablespaces determinado, los ficheros REDO LOG archivados y los ficheros de control
 - Tipo lógico: sólo extraen los datos de las tablas utilizando comandos SQL y se realizan con la utilidad export/import.
 - Backup con Export/Import: Estas utilidades permiten al DBA hacer copias de determinados objetos de la BD, así como restaurarlos o moverlos de una BD a otra. Estas herramientas utilizan comandos del SQL para obtener el contenido de los objetos y escribirlos en/leerlos de ficheros

• LOS MECANISMOS DE RESTAURACIÓN DE COPIAS DE SEGURIDAD CON LOS QUE CUENTA.

MYSQL

Estrategia de respaldo

Para diferentes escenarios, debemos formular diferentes estrategias de respaldo para respaldar la base de datos. Generalmente, las estrategias de respaldo generalmente son las siguientes cuatro:

- Direct cp, archivos de base de datos de copia tar: Si la cantidad de datos es pequeña, se puede usar esta para copiar directamente el archivo de la base de datos

- **mysqldump copy BIN LOGS:** Si la cantidad de datos está bien, se puede usar esta, primero usar mysqldump para realizar una copia de seguridad completa de la base de datos y luego hacer una copia de seguridad del REGISTRO BINARIO para lograr el efecto de la copia de seguridad incremental
- **copia instantánea de lvm2 BIN LOGS:** Si la cantidad de datos es promedio, pero no afecta demasiado la operación del negocio, se puede usar esta, usar instantáneas de lvm2 para hacer una copia de seguridad de los archivos de datos, y luego hacer una copia de seguridad regularmente del REGISTRO BINARIO para lograr el efecto de la copia de seguridad incremental
- **xtrabackup:** Si la cantidad de datos es grande, pero no afecta demasiado la operación del negocio, puede usar esta, después de usar xtrabackup para respaldo completo, use xtrabackup regularmente para respaldo incremental o diferencial

POSTGRESQL

PostgreSQL, al igual que otros motores SQL guarda un log (WAL) que almacena todos los cambios que se realizan contra las bases de datos, algo similar al log binario de MySQL. Esto nos puede permitir en caso de fallo recuperar datos de una base de datos desde el punto concreto que necesitemos. En la documentación de PostgreSQL recomiendan combinar este sistema con el de copiado a nivel físico para una estrategia compleja de backups.

Con Bacula, puede hacer copias de seguridad y restaurar bases de datos PostgreSQL rápidamente y sin necesidad de complejos scripts. El software se encarga de toda la información esencial, a la vez que admite técnicas de copia de seguridad y restauración de punto en el tiempo (o PITR), incluyendo la copia de seguridad incremental de la base de datos PostgreSQL y la copia de seguridad diferencial de la base de datos PostgreSQL.

En Azure Database for PostgreSQL, al realizar una restauración se crea un nuevo servidor a partir de las copias de seguridad del servidor original. Hay dos tipos de restauración disponibles:

- **Restauración a un momento dado:** está disponible con cualquier opción de redundancia de copia de seguridad y crea un nuevo servidor en la misma región que el servidor original.
- **Restauración geográfica:** solo está disponible si ha configurado el servidor para almacenamiento con redundancia geográfica y permite restaurar el servidor en una región diferente.

ORACLE

Existen tres tipos de recuperación en Oracle:

- **Recuperación de bloques:** Es el mecanismo de recuperación más simple, y se realiza automáticamente. Se produce cuando un proceso muere justo cuando está cambiando un bloque, y se utilizan los registros redo log en línea para reconstruir el bloque y escribirlo en disco.

- Recuperación de threads: Se realiza automáticamente cuando Oracle descubre que una instancia muere dejando abierto un thread, entonces se restauran los bloques de datos modificados que estaban en el cache de la instancia muerta, y cerrando el thread que estaba abierto. La recuperación se efectúa automáticamente cuando la BD se levanta.
- Recuperación física: Se realiza como respuesta a un comando RECOVER. Se utiliza para convertir los ficheros de backup en actuales, o para restaurar los cambios que fueron perdidos cuando un fichero de datos fue puesto offline sin un checkpoint, aplicando los ficheros redo log archivados y en línea.

Utilice RMAN para restaurar el archivo de control desde la copia de seguridad y, a continuación, restaurar y recuperar la base de datos. RMAN recupera la base de datos utilizando redo logs archivados.

RMAN: es un administrador de copia de seguridad y recuperación suministrado para bases de datos Oracle creado por Oracle Corporation. Proporciona capacidades de respaldo, restauración y recuperación de bases de datos que abordan problemas de alta disponibilidad y recuperación ante desastres.

- **EL MANEJO DEL REGISTRO DE TRANSACCIONES SI CUENTA CON UNO.**

Una transacción en un Sistema de Gestión de Bases de Datos es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

Se dice que un SGBD es transaccional si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio.

MYSQL

- MySQL es compatible con varios motores de almacenamiento. InnoDB es totalmente compatible con ACID. Las transacciones confiables deben ser compatibles con estas cuatro propiedades
- Las operaciones dentro de una transacción deben ser atómicas. Esto significa que todas las operaciones tienen éxito o fallan. Esta es la regla de todo o nada.
- La consistencia garantiza que la base de datos se encuentre en un estado consistente una vez finalizada la transacción, los datos son válidos y no hay registros a medio terminar.
- El aislamiento es el requisito de que otras operaciones no puedan acceder a los datos que se han modificado durante una transacción que aún no se ha completado, el aislamiento ocurre en caso de transacciones concurrentes.
 - Niveles de aislamiento:
 - Dirty reads (Lecturas sucias): Es el problema más importante de todos. Supone que las transacciones en curso puedan leer el resultado de otras transacciones aún no confirmadas.

- Non-Repeatable reads (Lecturas no repetibles): Ocurre cuando una transacción activa vuelve a leer un dato cuyo valor difiere con respecto al de la anterior lectura.
- Phantom reads (Lecturas fantasma): Este supone el menor problema que se nos puede plantear con respecto a las transacciones. Sucede cuando una transacción en un momento lanza una consulta de selección con una condición y recibe en ese momento N filas y posteriormente vuelve a lanzar la misma consulta junto con la misma condición y recibe M filas con $M > N$

Los pasos para usar transacciones en MySQL son:

- Iniciar una transacción con el uso de la sentencia BEGIN.
- Actualizar, insertar o eliminar registros en la base de datos.
- Si se quieren los cambios a la base de datos, completar la transacción con el uso de la sentencia COMMIT. Únicamente cuando se procesa un COMMIT los cambios hechos por las consultas serán permanentes.
- Si sucede algún problema, podemos hacer uso de la sentencia ROLLBACK para cancelar los cambios que han sido realizados por las consultas que han sido ejecutadas hasta el momento.

POSTGRESQL

- Una transacción empaqueta varios pasos en una operación, de forma que se completen todos o ninguno.
- Los estados intermedios entre los pasos no son visibles para otras transacciones ocurridas en el mismo momento.
- En el caso de que ocurra algún fallo que impida que se complete la transacción, ninguno de los pasos se ejecuta y no afectan a los objetos de la base de datos.
- Los pasos dentro de una transacción son varias sentencias SQL que deben de completarse todas para que queden registradas.

Comandos:

- BEGIN: el sistema permite que se ejecuten todas las sentencias SQL que necesitemos y las registra en un fichero
- COMMIT: estamos confirmando que todas las sentencias son correctas. Hasta que no se ejecute el comando COMMIT, las sentencias no quedarán registradas. Si cerramos la conexión antes de ejecutar este comando no se verá afectada ninguna de las relaciones de la base de datos
- ROLLBACK: podemos desechar las transacciones que se hayan ejecutado. después de haber realizado y confirmado una transacción, PostgreSQL nos permite anular dicha transacción de forma que no se modifique los datos de nuestra base de datos

Para poder utilizar estos comando mencionados (BEGIN, COMMIT y ROLLBACK) debemos de desactivar el AUTOCOMMIT. Ésta opción es a nivel de cliente y por defecto

está activada. De forma que toda sentencia ejecutada queda confirmada y registrada en la base de datos.

ORACLE

- Una transacción se define como un conjunto de operaciones sobre la base de datos
- En Oracle si se ejecuta un conjunto de operaciones y una de ellas falla se aborta la transacción entera

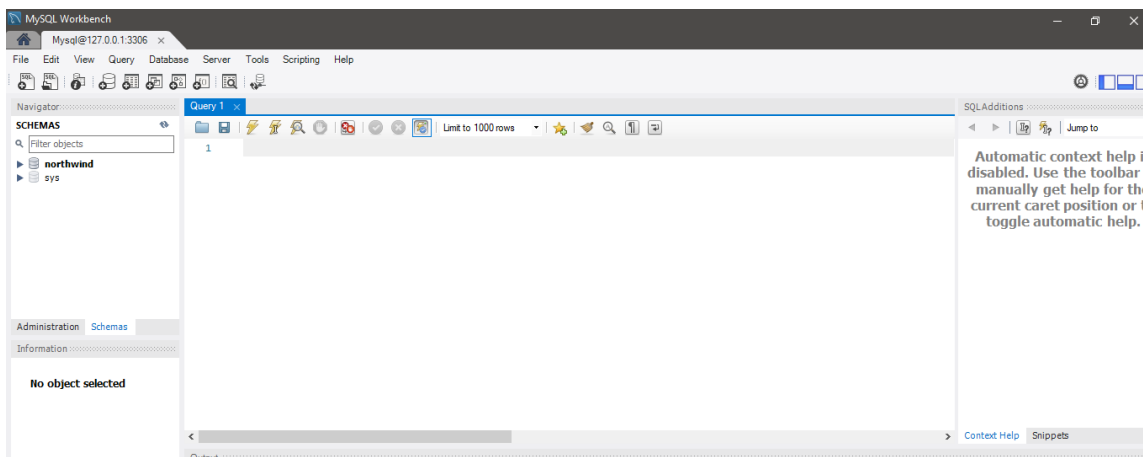
Comandos:

- COMMIT: Este comando da por concluida la transacción actual y hace definitivos los cambios realizados liberando las filas bloqueadas. Sólo después de que se ejecute commit tendremos acceso a los datos modificados.
- ROLLBACK: Este comando da por concluida la transacción actual y deshace los cambios que se pudiesen haber producido en la misma, liberando las filas bloqueadas. Se utiliza especialmente cuando no se puede concluir una transacción porque se han levantado excepciones.
- SAVEPOINT: Se utiliza para poner marcas o puntos de salvaguarda al procesar transacciones. Se utiliza junto con rollback permitiendo deshacer cambios hasta los savepoint. El número de savepoint está limitado a 5 por sesión pero lo podemos modificar con la siguiente sentencia: savepoint numero;
- ROLLBACK IMPLICITO: Este comando se ejecuta cuando un programa almacenado (procedimiento o función) falla y no se controla la excepción que produjo el fallo. Pero si en el programa tenemos un commit estos cambios no serán deshechos.
- ROLLBACK TO: deshace el trabajo realizado después del punto indicado. Pero no se confirma el trabajo hecho hasta el savepoint. La transacción no finaliza hasta que se ejecuta un comando de control de transacciones o hasta que finaliza la sesión.

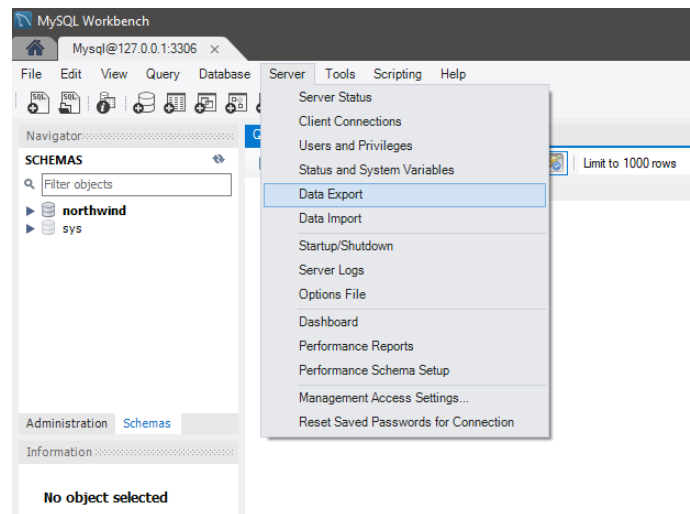
- **REALIZAR UNA COPIA DE SEGURIDAD EN CADA GESTOR.**

MYSQL

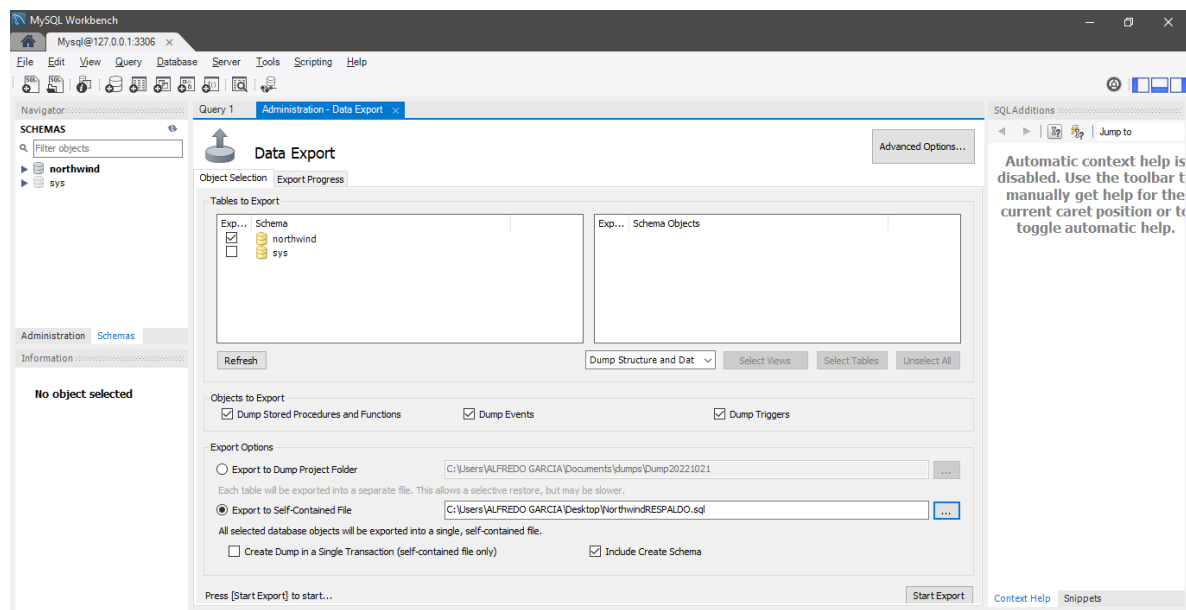
Nos conectamos



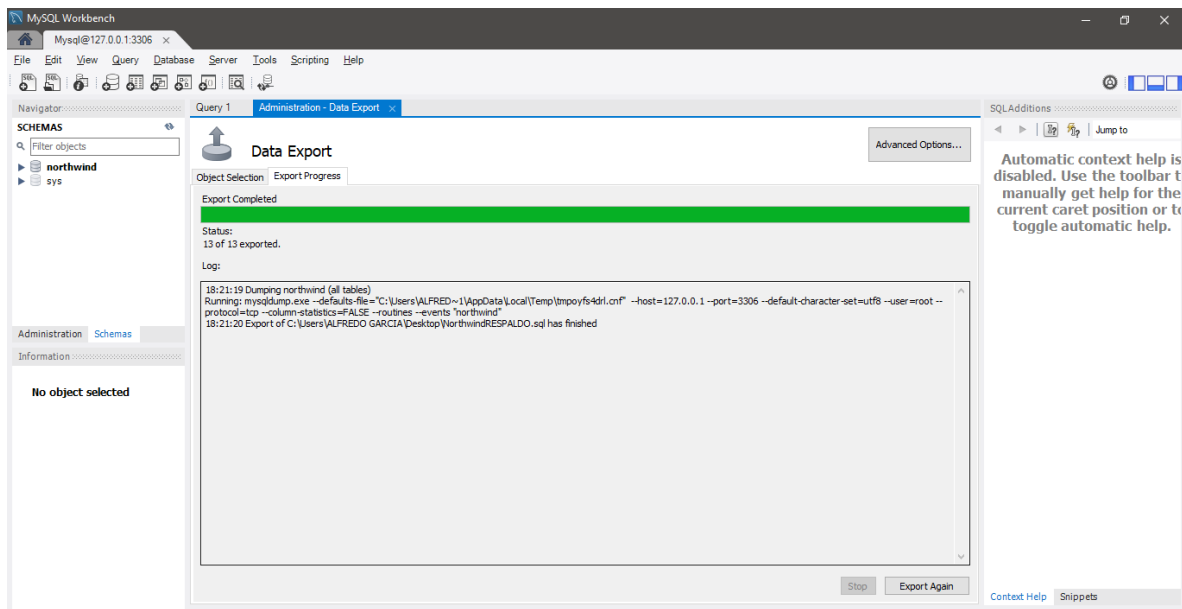
Vamos a Server y de ahí a Data Export



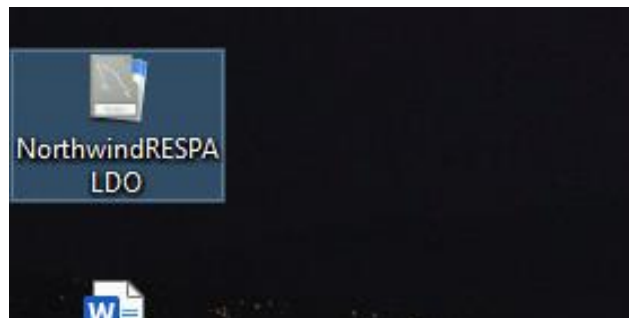
Marcamos la Northwind con la palomita, elegimos Dump Structure and Data, también los 3 Dump, elegimos Export to Self-Contained File y elegimos una ubicación local en este caso el escritorio, le pusimos NorthwindRESPALDO. Damos en Export Again



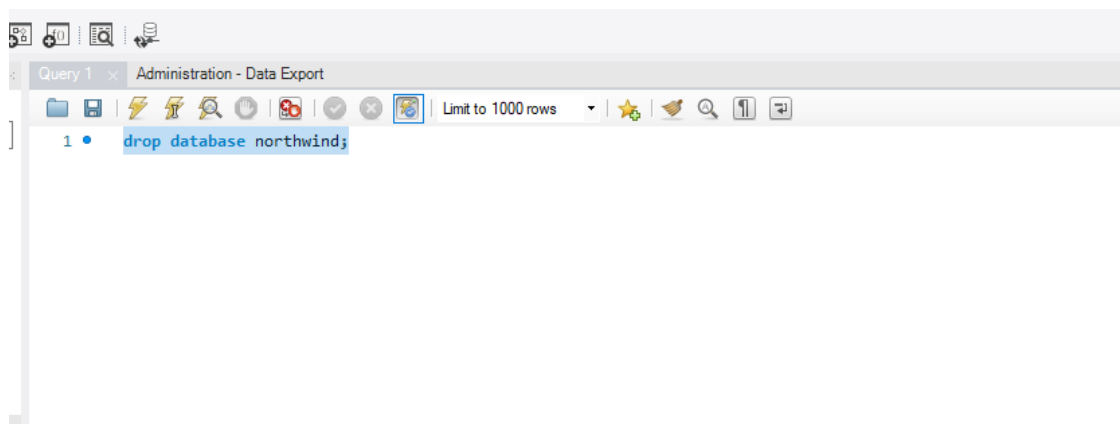
Esperamos a que se llene y que diga has finished



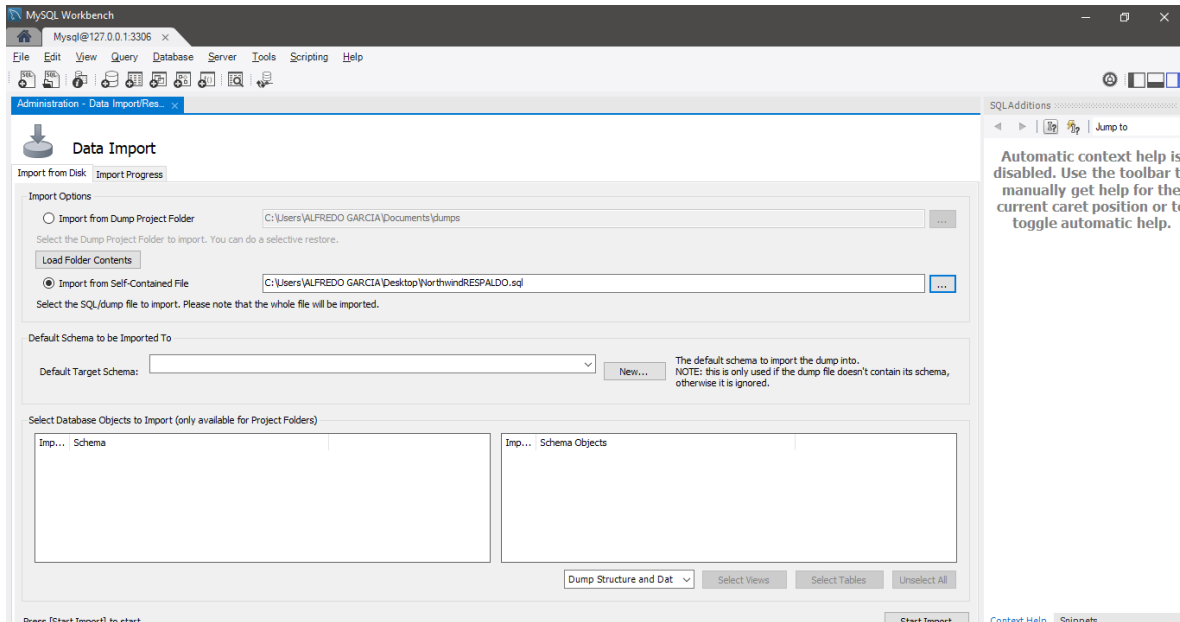
Comprobamos yendo al escritorio



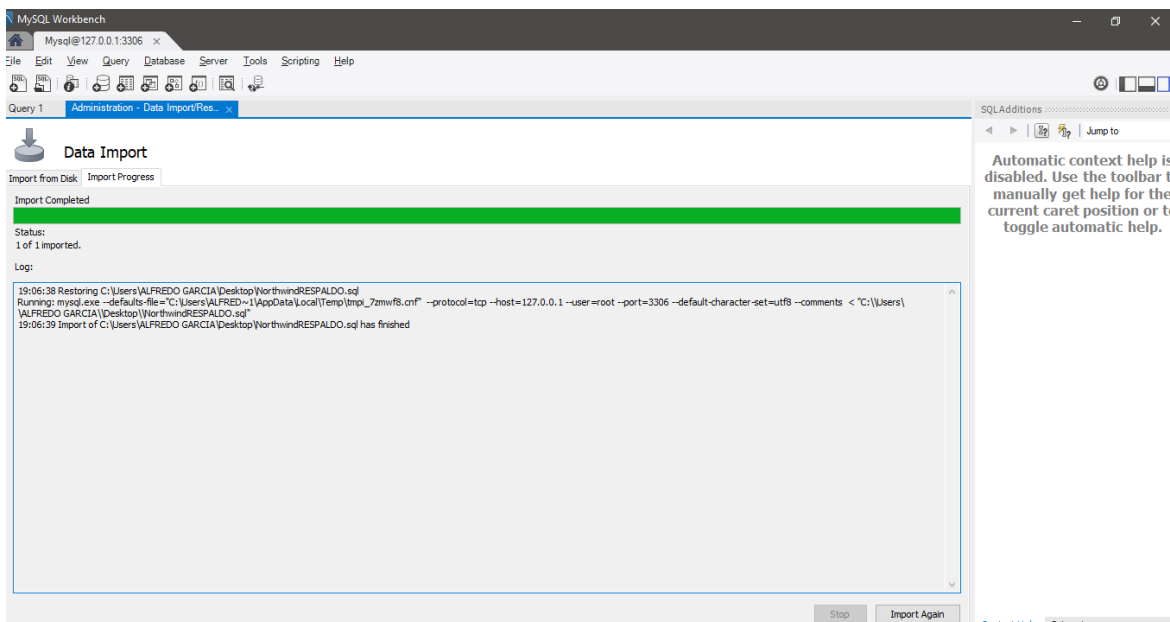
Borramos la bd original



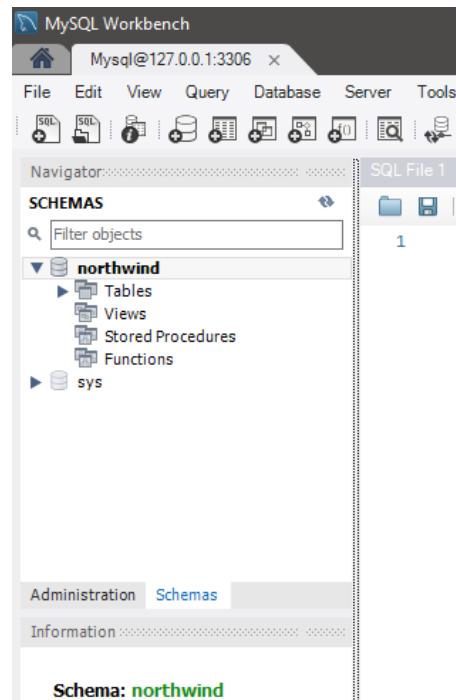
Vamos a Administración, luego Data Import/Restore, elegimos Import from Self/Contained File y buscamos el archivo que guardamos en este caso NorthwindRESPALDO.



Damos en Start Import



Checamos y sí está restaurada

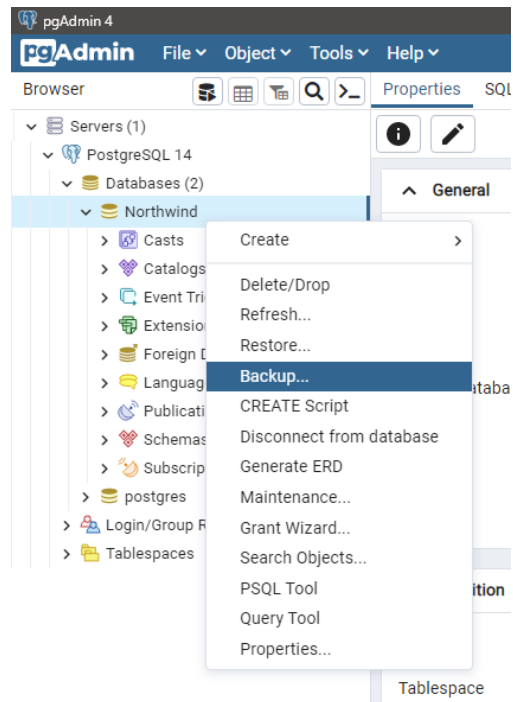


POSTGRESQL

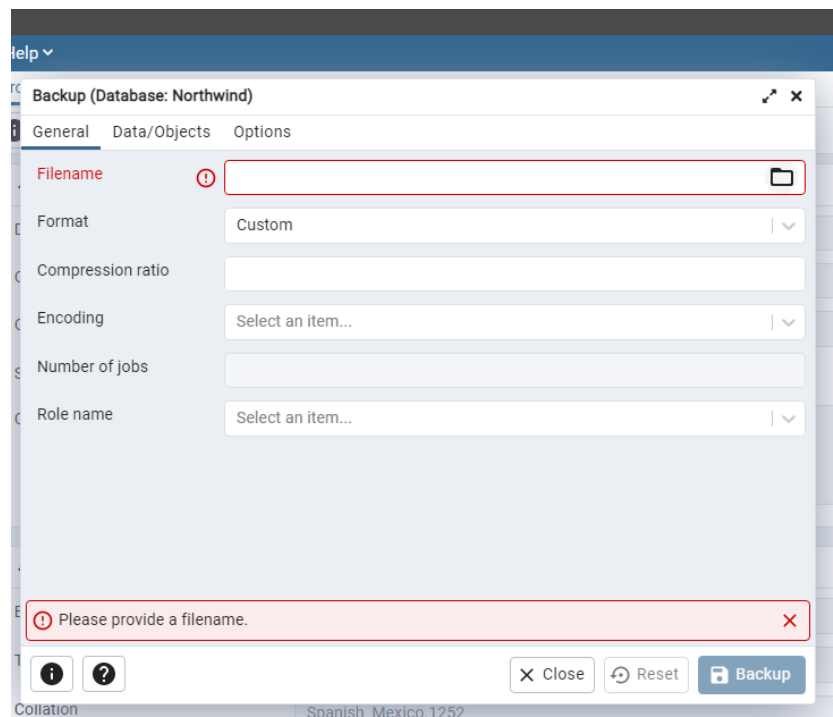
Entramos con nuestro Admin con clave



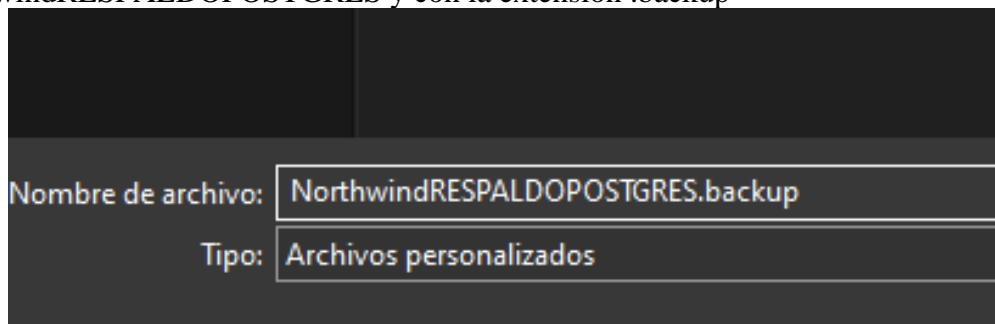
Vamos a la bd y le damos click derecho Backup



Nos aparecerá esta ventana y damos en la carpeta

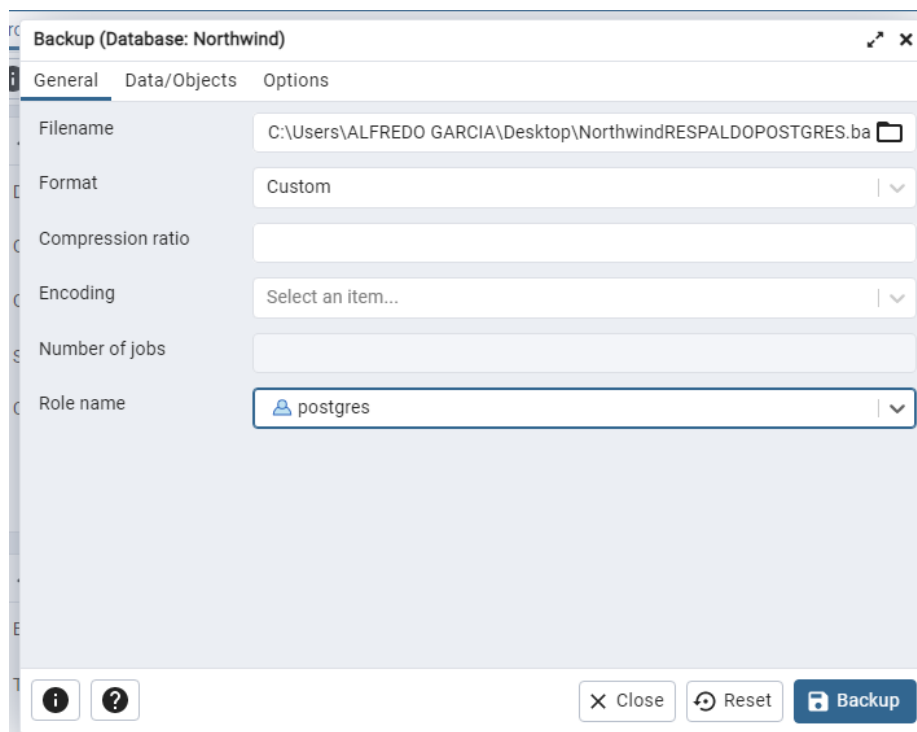


Elegimos un lugar en este caso el escritorio y le damos un nombre en este caso NorthwindRESPALDOPOSTGRES y con la extensión .backup



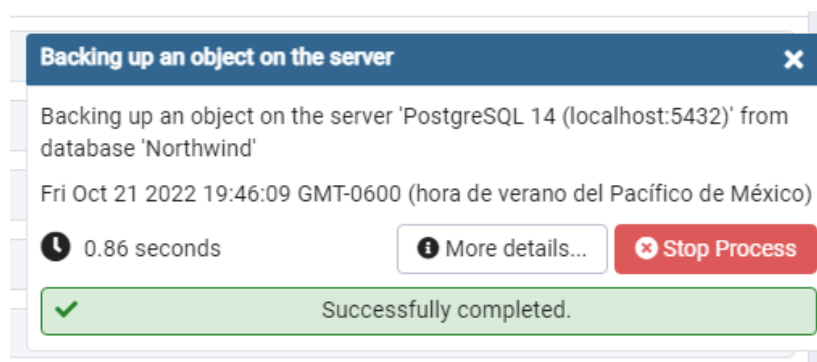
A dark-themed dialog box with two input fields. The first field is labeled 'Nombre de archivo:' and contains the text 'NorthwindRESPALDOPOSTGRES.backup'. The second field is labeled 'Tipo:' and contains the text 'Archivos personalizados'.

En Role name damos postgres. Damos Backup



A light-themed dialog box titled 'Backup (Database: Northwind)'. It has three tabs: 'General', 'Data/Objects', and 'Options'. The 'General' tab is active. It contains several fields: 'Filename' with a text box and a folder icon, 'Format' with a dropdown menu set to 'Custom', 'Compression ratio' with a text box, 'Encoding' with a dropdown menu set to 'Select an item...', 'Number of jobs' with a text box, and 'Role name' with a dropdown menu set to 'postgres'. At the bottom, there are three buttons: 'Close', 'Reset', and 'Backup'.

Esperamos a que se llene y nos mostrará así:



A status window titled 'Backing up an object on the server'. It contains the following text: 'Backing up an object on the server 'PostgreSQL 14 (localhost:5432)' from database 'Northwind'', 'Fri Oct 21 2022 19:46:09 GMT-0600 (hora de verano del Pacífico de México)', and '0.86 seconds'. There are two buttons: 'More details...' and 'Stop Process'. At the bottom, there is a green bar with a checkmark and the text 'Successfully completed.'.

Creamos una nueva base de datos para hacer el respaldo, le llamamos NorthwindRESPALDOPOSTGRES

The screenshot shows the 'Create - Database' dialog box with the 'General' tab selected. The 'Database' field contains 'NorthwindRESPALDOPOSTGRES'. The 'Owner' field shows a user icon and the text 'postgres'. The 'Comment' field is empty. At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

Field	Value
Database	NorthwindRESPALDOPOSTGRES
Owner	postgres
Comment	

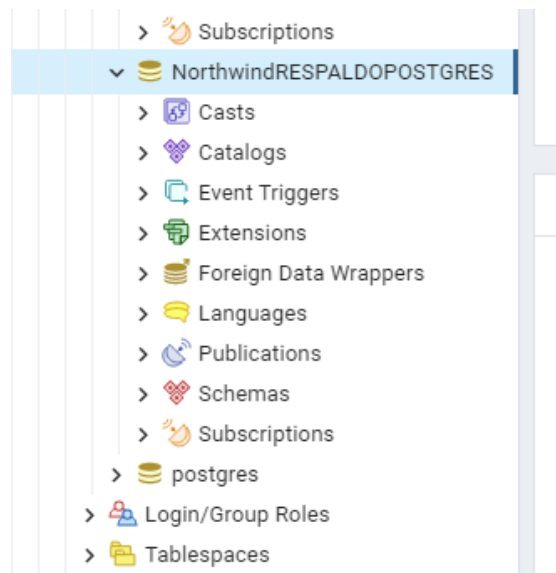
Damos en Security y le damos todos los privilegios. Damos Save

The screenshot shows the 'Create - Database' dialog box with the 'Security' tab selected. The 'Privileges' section shows a table with columns 'Grantee', 'Privileges', and 'Grantor'. The 'Grantee' is 'postgres' and the 'Grantor' is 'postgres'. The 'Privileges' column has a dropdown menu open showing 'C+T*c*' and a list of checked privileges: ALL, CREATE, TEMPORARY, CONNECT, and WITH GRANT OPTION. Below the table is a 'Security labels' section with columns 'Provider' and 'Security label'. At the bottom, there are buttons for 'Close', 'Reset', and 'Save'.

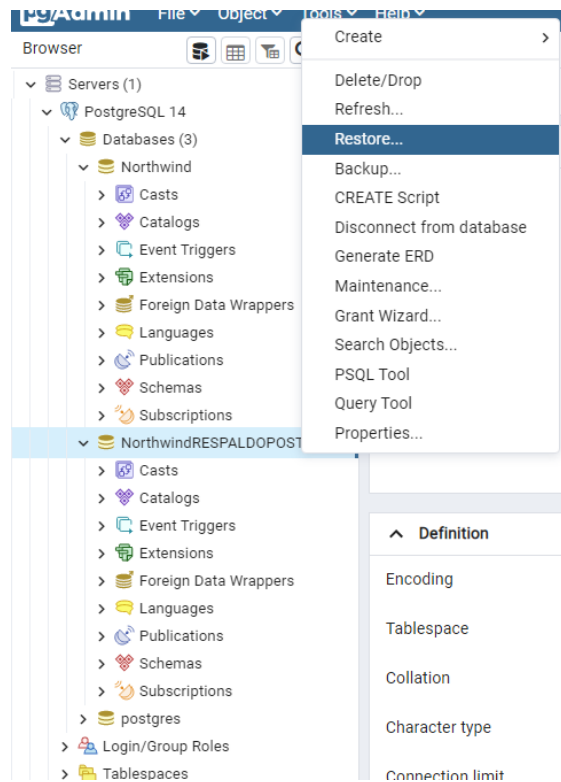
Grantee	Privileges	Grantor
postgres	C+T*c*	postgres

Provider	Security label
----------	----------------

Ya se nos creó



Click derecho en ella, Restore



Click en la carpetita

Restore (Database: NorthwindRESPALDOPOSTGRES)

General Data/Objects Options

Format Custom or tar

Filename !

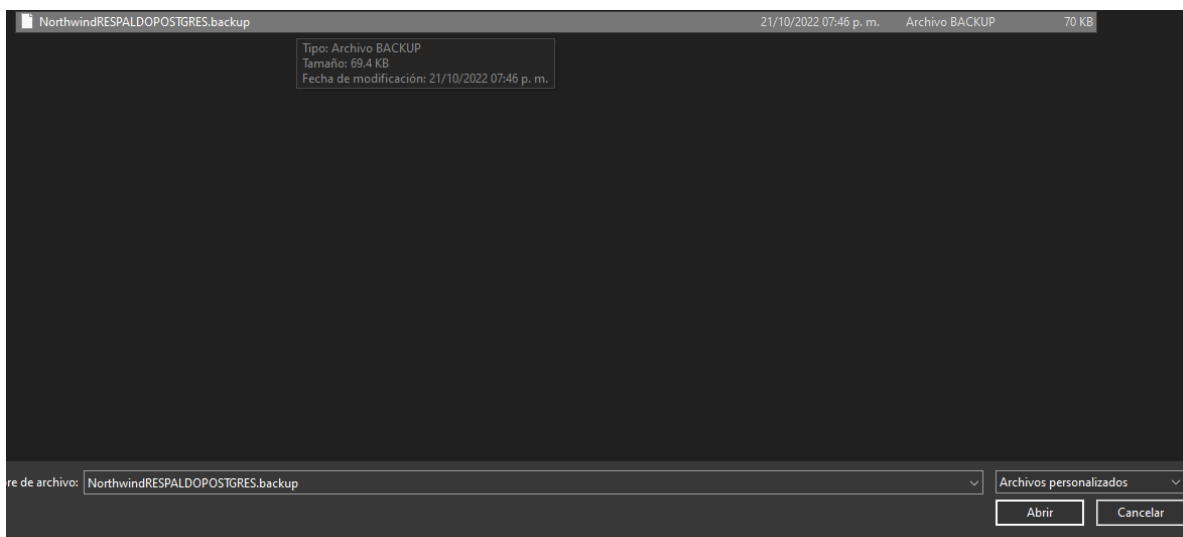
Number of jobs

Role name Select an item...

! Please provide a filename. X

Close Reset Restore

Elegimos el archivo NorthwindRESPALDOPOSTGRES.backup



Le damos Role name Postgres

Restore (Database: NorthwindRESPALDOPOSTGRES)

General Data/Objects Options

Format Custom or tar

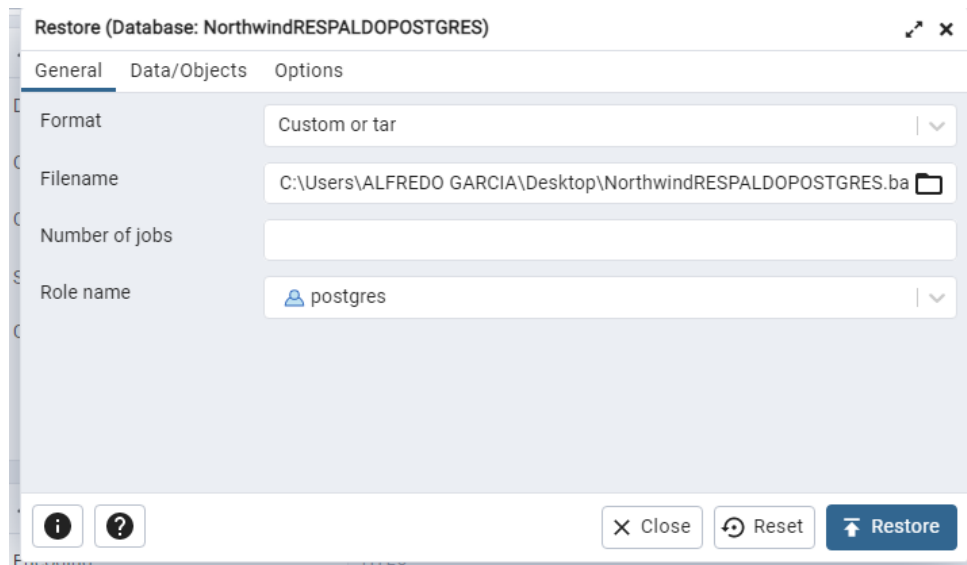
Filename C:\Users\ALFREDO GARCIA\Desktop\NorthwindRESPALDOPOSTGRES.ba

Number of jobs

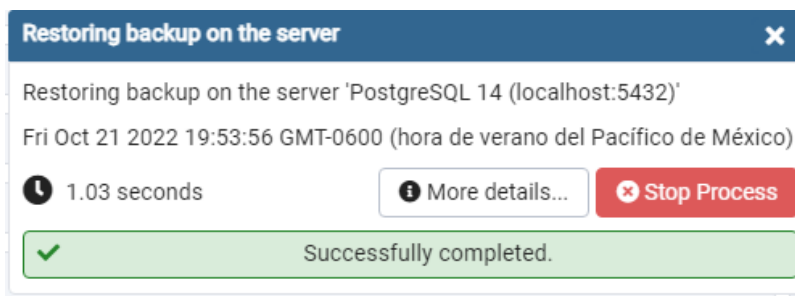
Role name postgres

Close Reset Restore

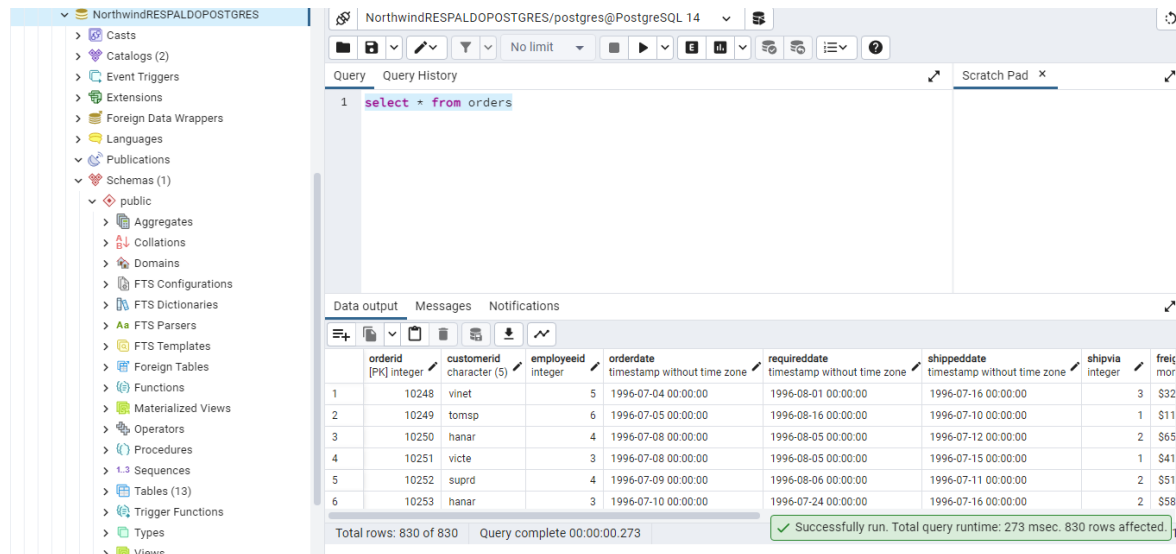
Damos Restore



Esperamos de nuevo a que se llene y listo

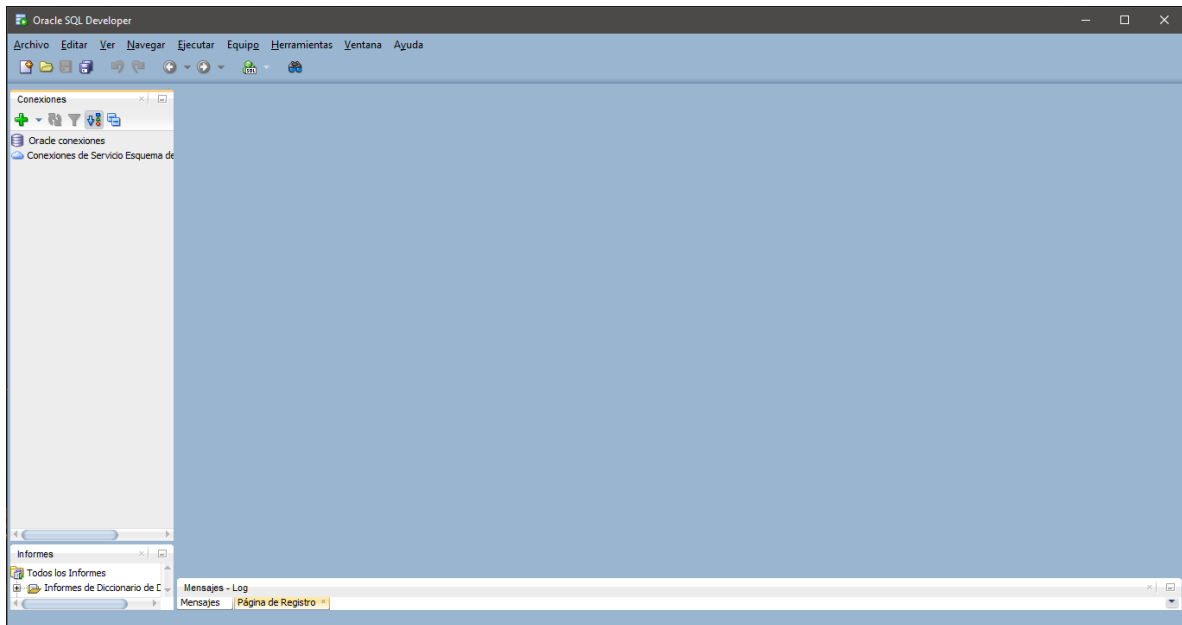


Hacemos un select sobre la base de datos restaurada o checamos sus tablas y efectivamente está bien

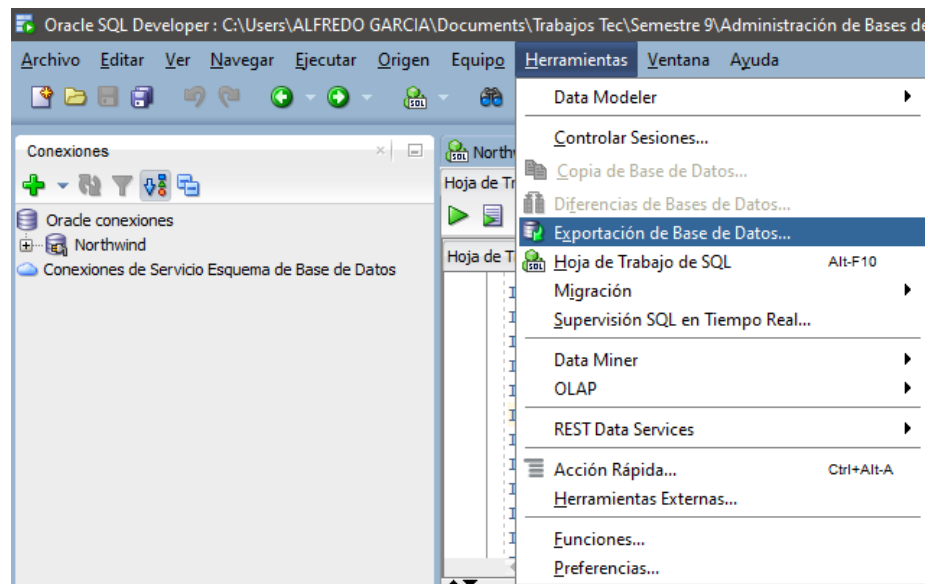


ORACLE

Entramos



Vamos a Herramientas y Exportación



Elegimos la Northwind y una ubicación local donde guardaremos el respaldo, damos Siguiente

Asistente de Exportación: Paso 1 de 5

Origen/Destino

Conexión: Northwind

☒ Exportar DDL

☒ Agregar Palabra Clave BYTE ☒ Forzar Vistas ☐ Borrados en Cascada ☐ Dependientes

☐ Borrados ☒ Permisos ☒ Partición ☒ Impresión con Sangría

☒ Mostrar Esquema ☒ Almacenamiento ☒ Tablespace ☒ Terminador

Versión: COMPATIBLE

☒ Exportar Datos

Formato: insert ☒ Mostrar Esquema

Terminador de Línea: valor por defecto de entorno

☐ Incluir Confirmación Cada 100 Filas

Guardar como: Archivo Único ☐ Comprimido Codificación: Cp1252

Archivo: REDO GARCIA\Desktop\respaldo oracle\NorthwindRESPALDOORACLE.sql Examinar...

☐ Ir al Resumen.

Ayuda < Atrás Siguiente > Terminar Cancelar

Marcamos solo las siguientes. Damos Siguiente

Asistente de Exportación: Paso 2 de 5

Tipos a Exportar

Limite el ámbito del proceso seleccionando los tipos de objeto que desea incluir. Si no limita más el proceso mediante la selección de objetos, se procesarán todos los objetos de los tipos seleccionados aquí.

Tipos de Objetos Estándar

☐ Conmutar Todos los Objetos Estándar

☒ Tablas ☐ Tipos ☐ Especificación del Paquete

☒ Vistas ☒ Secuencias ☐ Cuerpo del Paquete

☒ Índices ☐ Sinónimos ☒ Procedimientos

☒ Disparadores ☐ Enlaces de Base de Datos ☐ Funciones

☒ Restricciones ☐ Colas ☐ Operadores

☐ Restricciones Referenciales ☐ Tablas de Colas

☐ Vistas Materializadas

☐ Logs de Vistas Materializadas

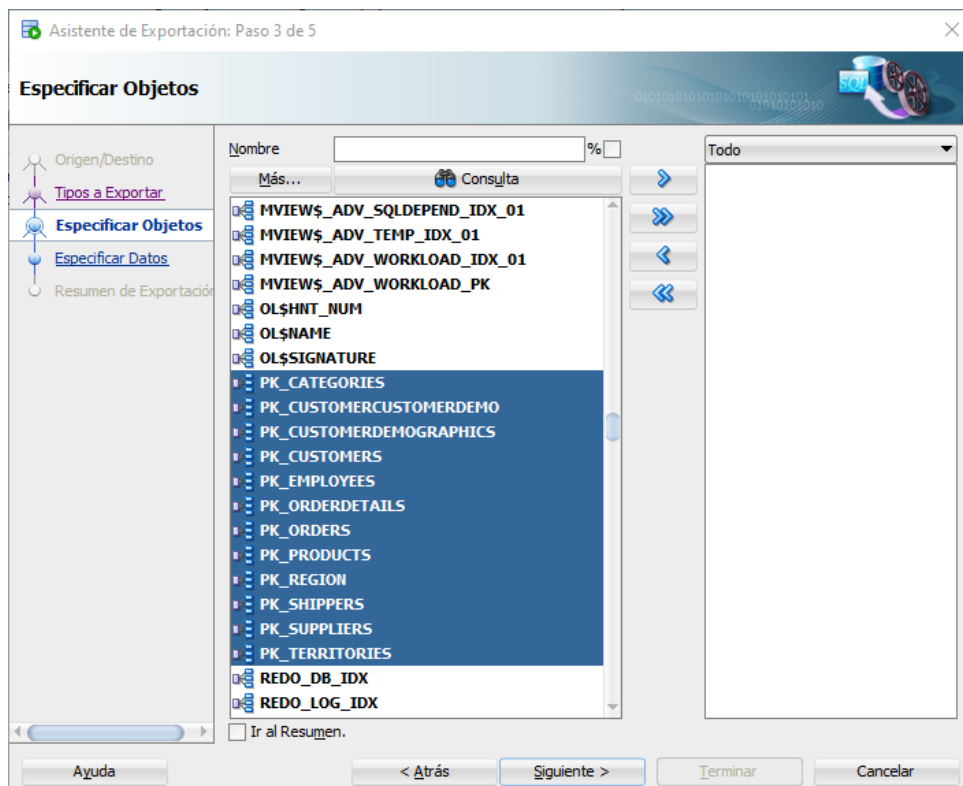
Tipos de Objetos de Extensión

☐ Conmutar Todos los Objetos de Extensión

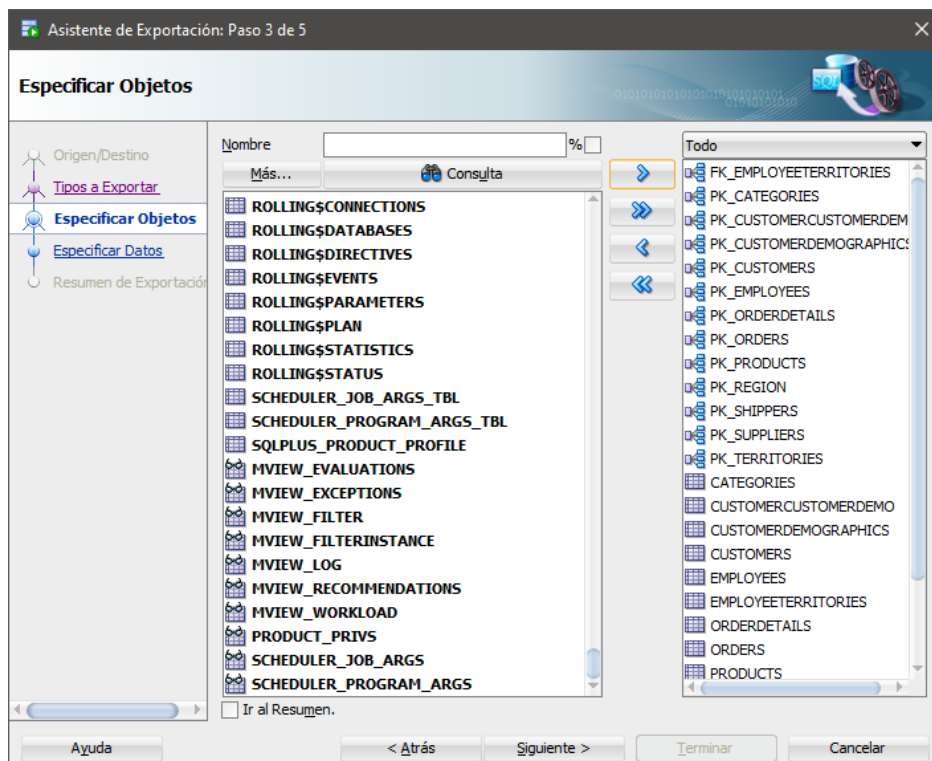
☐ Ir al Resumen.

Ayuda < Atrás Siguiente > Terminar Cancelar

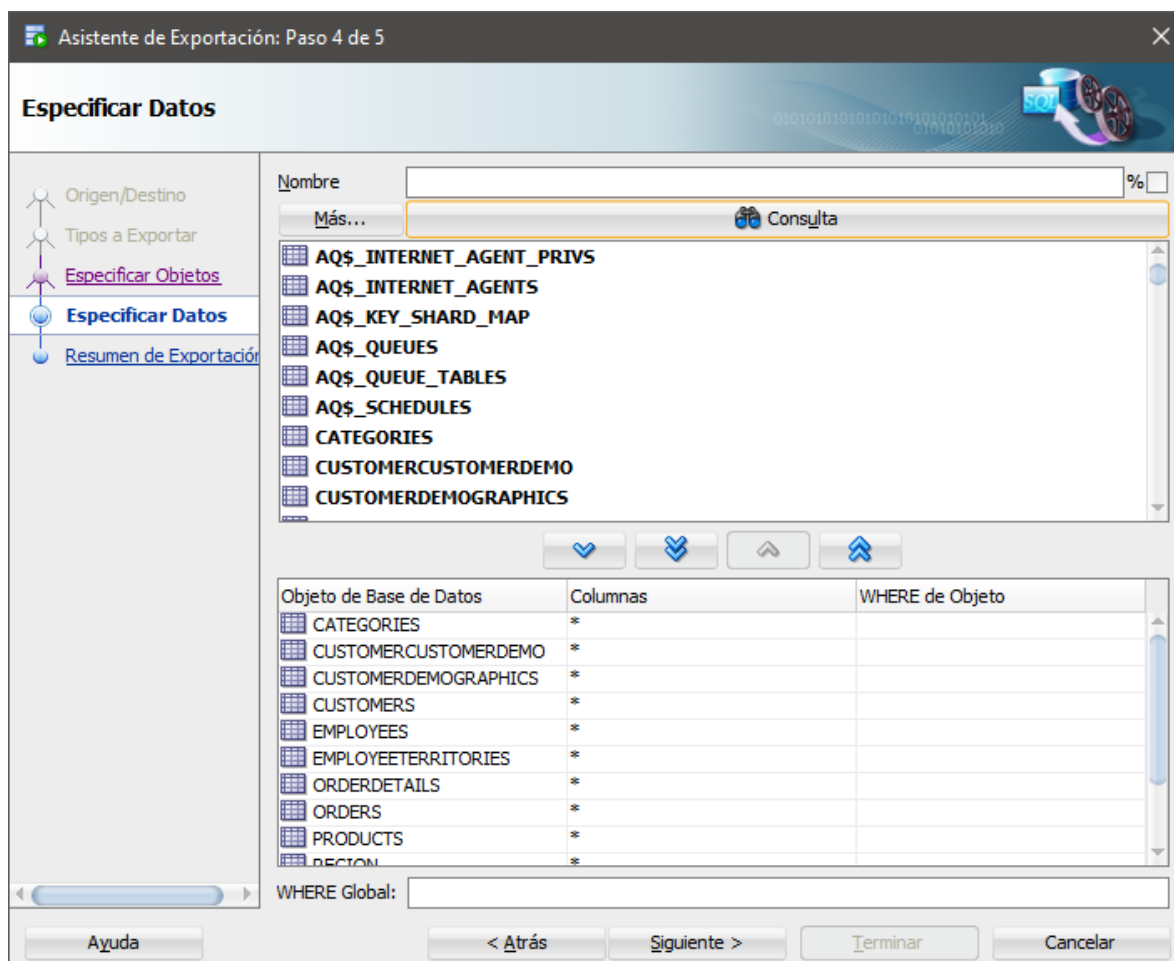
Damos en Consulta y elegimos los objetos que queremos respaldar



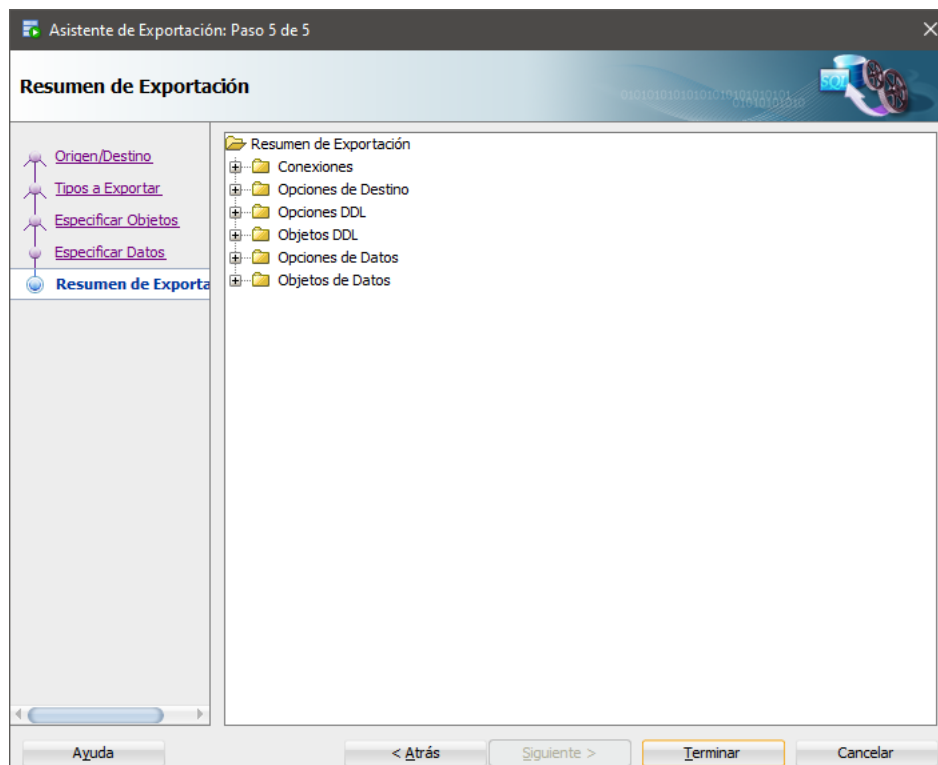
Los pasamos con la flecha derecha que está hasta arriba. Damos Siguiente



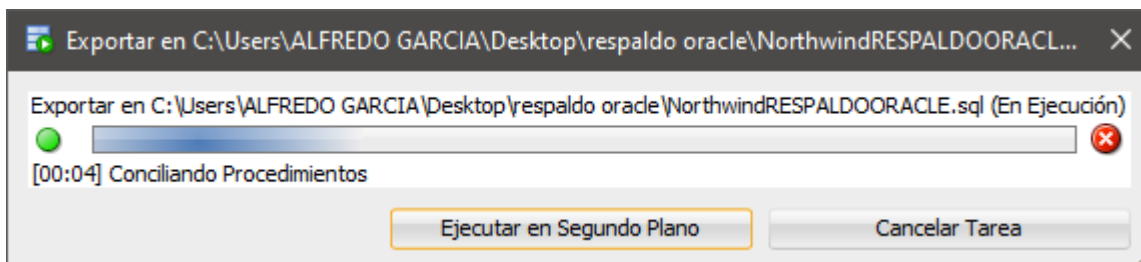
Siguiente



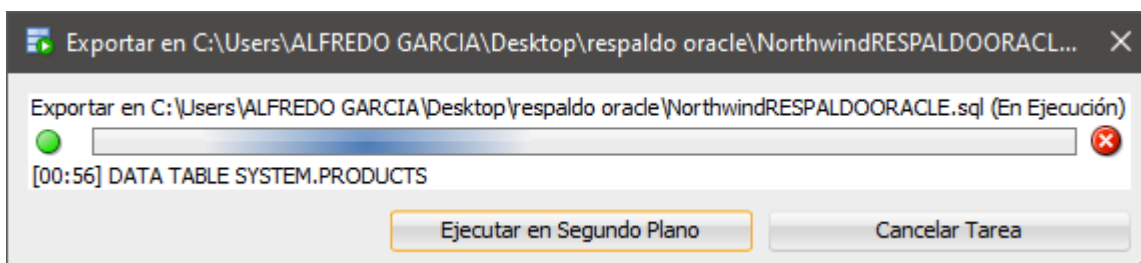
Terminar



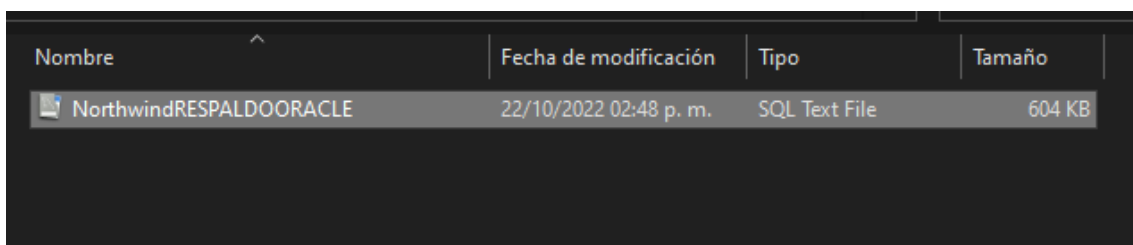
Esperamos a que se termine o cargue



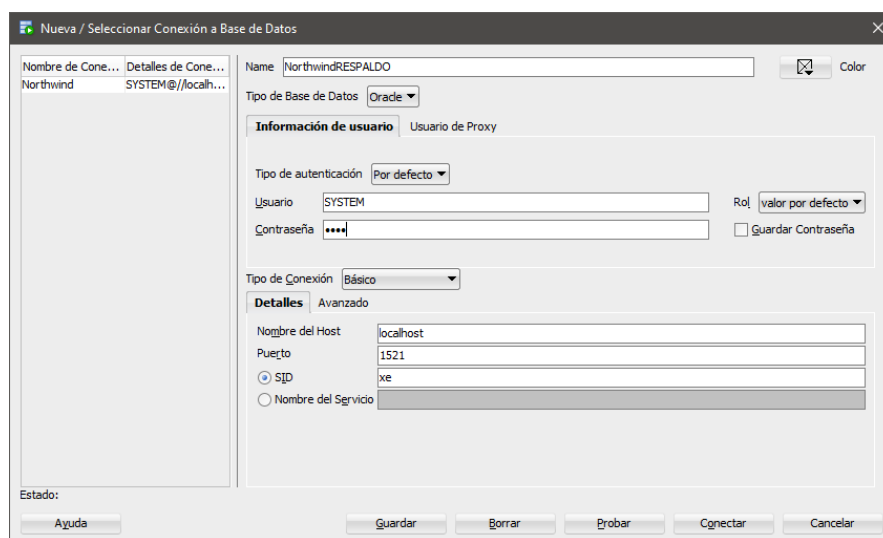
Puede que tarde mucho



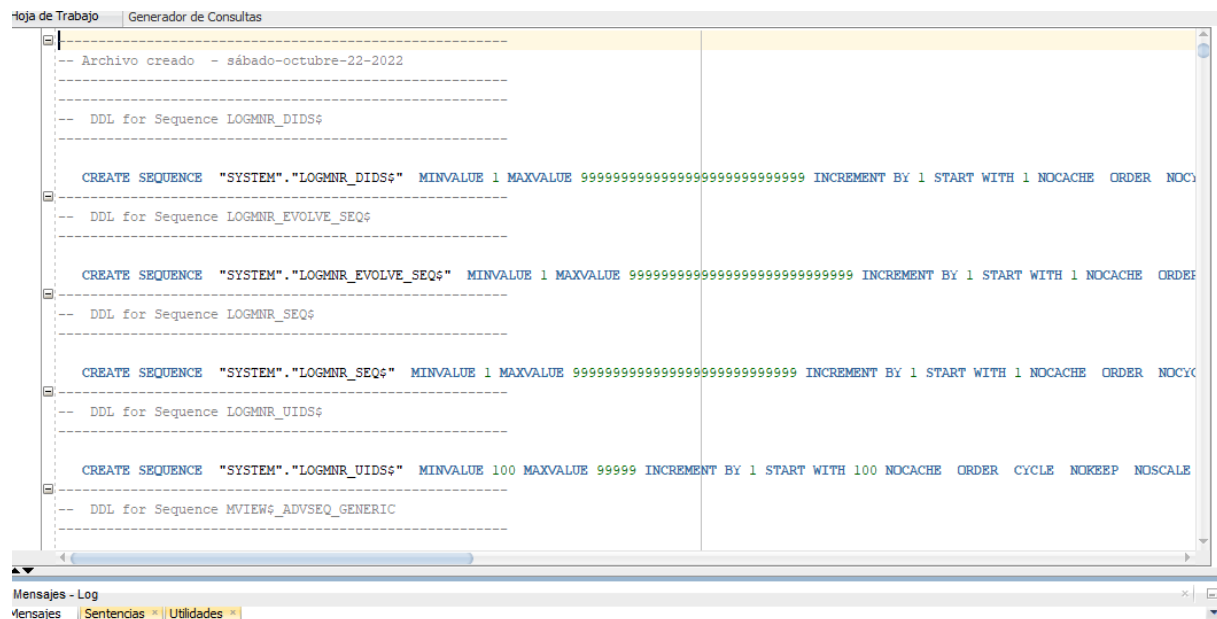
Se genera el archivo en la ubicación



Creamos una nueva conexión para hacer el respaldo



Corremos el archivo del respaldo sobre la nueva conexión



Hoja de Trabajo | Generador de Consultas

```
-- Archivo creado - sábado-octubre-22-2022

-- DDL for Sequence LOGMNR_DIDS$

CREATE SEQUENCE "SYSTEM"."LOGMNR_DIDS$" MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE;

-- DDL for Sequence LOGMNR_EVOLVE_SEQ$

CREATE SEQUENCE "SYSTEM"."LOGMNR_EVOLVE_SEQ$" MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE;

-- DDL for Sequence LOGMNR_SEQ$

CREATE SEQUENCE "SYSTEM"."LOGMNR_SEQ$" MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE;

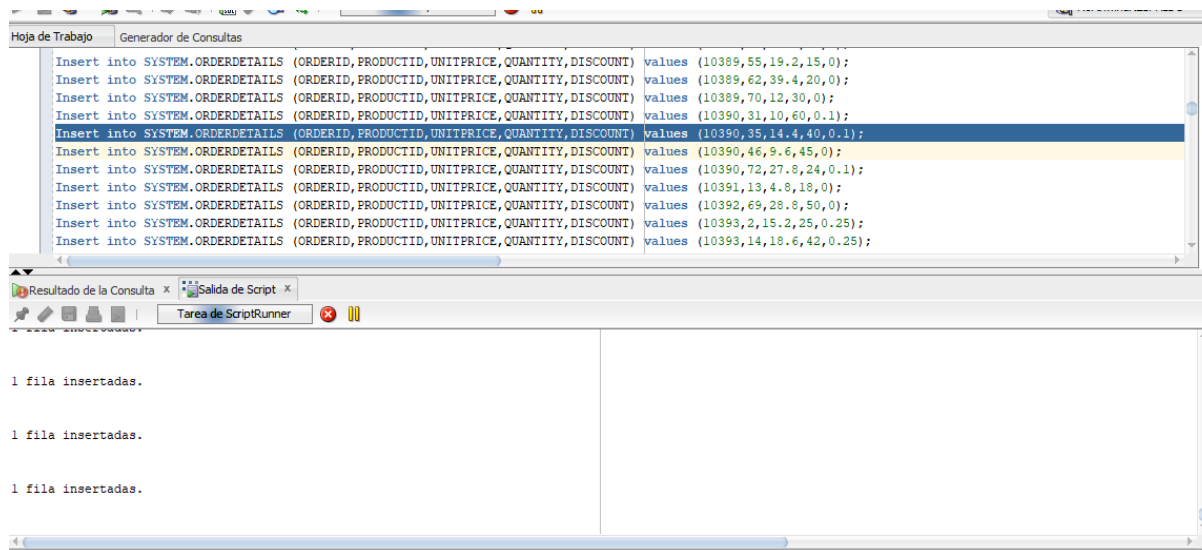
-- DDL for Sequence LOGMNR_UIDS$

CREATE SEQUENCE "SYSTEM"."LOGMNR_UIDS$" MINVALUE 100 MAXVALUE 99999 INCREMENT BY 1 START WITH 100 NOCACHE ORDER CYCLE NOKEEP NOSCALE;

-- DDL for Sequence MVIEW$ADVSEQ_GENERIC
```

Mensajes - Log

Mensajes | Sentencias * | Utilidades *



Hoja de Trabajo | Generador de Consultas

```
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10389,55,19.2,15,0);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10389,62,39.4,20,0);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10389,70,12,30,0);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10390,31,10,60,0.1);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10390,35,14.4,40,0.1);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10390,46,9.6,45,0);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10390,72,27.8,24,0.1);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10391,13,4.8,18,0);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10392,69,28.8,50,0);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10393,2,15.2,25,0.25);
Insert into SYSTEM.ORDERDETAILS (ORDERID,PRODUCTID,UNITPRICE,QUANTITY,DISCOUNT) values (10393,14,18.6,42,0.25);
```

Resultado de la Consulta | Salida de Script

Tarea de ScriptRunner

```
1 fila insertadas.

1 fila insertadas.

1 fila insertadas.
```


Una vez terminado hacemos un select para comprobar que se restauraron bien los datos

Hoja de Trabajo | Generador de Consultas

```
-- Archivo creado - sábado-octubre-22-2022

-- DDL for Sequence LOGMNR_DIDS$

select * from employees

CREATE SEQUENCE "SYSTEM"."LOGMNR_DIDS$" MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE NOKEEP
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 9 en 0.102 segundos

	EMPLOYEEID	LASTNAME	FIRSTNAME	TITLE	TITLEOF COURTESY	BIRTHDATE	HIREDATE	ADDRESS	CITY	REGION	POSTALCODE	
1	1	Devolio	Nancy	Sales Representative	Ms.	12-08-1948	05-01-1992	507 - 20th Ave. E.Apt. 2A	Seattle	WA	98122	U
2	2	Fuller	Andrew	Vice President, Sales	Dr.	02-19-1952	08-14-1992	908 W. Capital Way	Tacoma	WA	98401	U
3	3	Leverling	Janet	Sales Representative	Ms.	08-30-1963	04-01-1992	722 Moss Bay Blvd.	Kirkland	WA	98033	U
4	4	Peacock	Margaret	Sales Representative	Mrs.	09-19-1937	05-03-1993	4110 Old Redmond Rd.	Redmond	WA	98052	U
5	5	Buchanan	Steven	Sales Manager	Mr.	03-04-1955	10-17-1993	14 Garrett Hill	London	(null)	SW1 8JR	U
6	6	Suyama	Michael	Sales Representative	Mr.	07-02-1963	10-17-1993	Coventry House Miner Rd.	London	(null)	EC2 7JR	U
7	7	King	Robert	Sales Representative	Mr.	05-29-1960	01-02-1994	Edgeham Hollow Winchester Way	London	(null)	RG1 9SP	U
8	8	Callahan	Laura	Inside Sales Coordinator	Ms.	01-09-1958	03-05-1994	4726 - 11th Ave. N.E.	Seattle	WA	98105	U
9	9	Dodsworth	Anne	Sales Representative	Ms.	01-27-1966	11-15-1994	7 Houndstooth Rd.	London	(null)	W62 7LT	U

Mensajes - Log

```
select * from ORDERDETAILS
```

```
select count(distinct productid) from ORDERDETAILS
```

loja de Trabajo | Generador de Consultas

```
-- Archivo creado - sábad-octubre-22-2022
--
-- DDL for Sequence LOGMNR_DIDS$

select * from ORDERDETAILS
select count(distinct productid) from ORDERDETAILS

CREATE SEQUENCE "SYSTEM"."LOGMNR_DIDS$" MINVALUE 1 MAXVALUE 99999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE NOKEEP
```

Salida de Script x | Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0.006 segundos

COUNT(DISTINCTPRODUCTID)
77