

1.1.1 Copias de seguridade

1.1.1.1 Definición

De forma xenérica, as copias de seguridade, copias de respaldo ou backups (polo seu nome en inglés) son unha medida preventiva na seguridade informática que consiste en copias dos datos orixinais, ou mesmo programas, para dispor dun xeito de recuperalos en caso de perda, calquera que sexa a circunstancia que a provoque. No caso concreto das bases de datos, as copias de seguridade aplícanse aos datos nelas almacenados.

As copias de seguridade deben garantir dúas características da información:

- **Integridade:** refírese á calidade da información, xa que esta debe ser precisa e completa e non sufrir ningunha manipulación nin alteración respecto á orixinal sen a debida autorización.
- **Dispoñibilidade:** a información debe estar accesible en calquera momento para calquera usuario que teña permisos para poder acceder a ela.

1.1.1.2 Planificación das copias de seguridade

Para estar preparados ante calquera desastre que elimine ou corrompa os datos almacenados nunha base de datos, débese planificar unha política de realización de copias de seguridade periódicas. Esta planificación axeitada das copias de seguridade forma parte do plan de continxencia dunha empresa, xa que a perda de datos pode pór en perigo a continuidade do negocio.

Algúns dos requisitos que debe cumprir a planificación de copias de seguridade son:

- Identificar as bases de datos, ou datos dentro delas, que requiran ser preservados.
- Establecer a frecuencia coa que se van realizar os procesos de copia, así como o tipo de copia. Esta frecuencia inflúe na cantidade de información que se pode perder con respecto á fonte orixinal. Este parámetro é de suma importancia e require unha análise exhaustiva.

Por exemplo, supóñase unha empresa na que se realizan as copias de seguridade cada noite as 3:00hs. Se a base de datos falla ás 12 horas da mañá, toda a información xerada dende a noite anterior ata esa hora non estará gardada na copia de seguridade.

- Establecer o esquema de rotación: a rotación refírese á forma en que se almacenan e resgardan as copias de seguridade. Un esquema de rotación indica cantas cintas (ou outro tipo de soporte de almacenamento) se utilizan para realizar a copia de seguridade.



Nas seguintes ligazóns pódese obter información adicional sobre esquemas de rotación en copias de seguridade.

- <http://searchdatacenter.techtarget.com/es/consejo/Como-optimizar-su-estrategia-de-rotacion-de-cintas-de-respaldo>.

- Dispor o almacén físico para as copias. Este almacén determínase en función da seguridade que requira a información, entre almacéns no mesmo edificio ou remotos en edificios externos. Por exemplo, cando se produce un incendio no edificio da empresa, a información almacenada nun edificio externo segue a estar dispoñible.
- Procurar unha probabilidade de erro mínima, asegurándose de que os datos se copien nuns soportes fiables e en bo estado. Non se deben utilizar soportes que estean preto de cumprir a súa vida útil, para evitar que fallen cando se vaia recuperar a información que conteñen.

- Controlar os soportes que conteñen as copias, gardándoos nun lugar seguro e restrinxindo o seu acceso só ás persoas autorizadas.
- Planificar a restauración das copias:
 - Formando o persoal técnico encargado de realizalas.
 - Dispondo de soportes para restaurar a copia, diferentes dos de produción.
 - Establecendo os medios para dispor da devandita copia no menor tempo posible.
- Probar o sistema de forma exhaustiva, para comprobar a súa correcta planificación e a eficacia dos medios dispostos.
- Definir a vixencia das copias, establecendo un período no que estas deixan de ter validez e poden substituírse por outras máis actuais.
- Controlar a obsolescencia dos dispositivos de almacenamento. Para o caso das copias que almacenen información histórica da organización, por exemplo proxectos xa pechados, débese ter en conta o tipo de dispositivo en que se realiza a copia, para evitar que no momento que se requira a restauración da devandita información non existan xa lectores axeitados para o dispositivo.

Cando se desboten os soportes de almacenamento, porque cheguen ao límite da súa vida útil fixado na política de copias de seguridade, é importante realizar un proceso de borrado seguro ou destrución, para asegurar que a información que conteñan non poida ser recuperada posteriormente.
- Especificar os sistemas de copias de seguridade, sobre todo cando cumpra realizar copias de seguridade en sistemas que non sexa posible deter para realizar a copia en frío, polo que se debe indicar a estratexia que se vaia utilizar para realizar a copia en quente.

1.1.1.3 Tipos de copias de seguridade

A grandes liñas, existen tres tipos de copias de seguridade: completa, diferencial e incremental.

- Copia completa. Neste tipo de copia créase unha copia de tódolos datos existentes. A primeira vez que se realiza unha copia sobre a información adoita ser deste tipo.
- Copia diferencial. A copia de seguridade diferencial conterá todos os datos que se crearon ou modificaron desde a última copia completa. Para restaurar os datos necesítase a última copia completa e a última copia diferencial.
- Copia incremental. Neste caso só se xera unha copia dos datos que foron creados ou modificados desde a última copia completa ou diferencial realizada. Para restaurar os datos, necesítase a última copia completa e todas as copias incrementais realizadas desde entón.

Independentemente destas definicións xenéricas, cada SXBDs ten o seu xeito particular e as súas propias ferramentas para proporcionar ao administrador a suficiente flexibilidade á hora de realizar copias de seguridade.

1.1.1.4 Copias de seguridade con MySQL

Mysqldump

MySQL dispón dunha utilidade chamada *mysqldump*, que se instala có propio MySQL e que permite crear un script co esquema e os datos da base de datos. Este script pódese utilizar como copia de seguridade, para mover bases de datos dun servidor a outro, ou para crear unha base de datos de probas baseada nunha xa existente. A sintaxe é a seguinte: *mysqldump [opcións] nome_base_de_datos [nome_táboa]*.

Algúnha das *opcións* que se poden utilizar son:

- *--help*. Mostra unha mensaxe de axuda con todas as opcións (-?)
- *--add-locks*. Engade LOCK TABLES, e UNLOCK TABLE para copiar cada táboa.
- *--databases*. Permite indicar o nome de varias bases de datos (-B)
- *--all-databases*. Envorca o contido de todas as bases de datos do servidor (-A)
- *--flush-logs*. Baleira os ficheiros de rexistro de log antes de comezar o envorcado (-F)
- *--lock-tables*. Bloquea todas as táboas antes de comezar o envorcado (-l)
- *--non-create-db*. Non inclúe a sentenza CREATE DATABASE (-n)
- *--non-create-info*. Copia soamente os datos. Non inclúe as sentenzas de creación (base de datos nin táboas)
- *--non-data*. Non incluíra ningunha información sobre as filas da táboa. Esta opción serve para crear unha copia do esquema da base de datos
- *--opt*. Fai unha operación de envorcado rápida. É o mesmo que especificar: *--add-drop-table --add-locks --?create-options --disable-keys --extended-insert --lock-tables --quick --set-charset*.
- *--quick*. Acelera o proceso en táboas grandes xa que non carga en memoria os resultados, senón que recupera e grava fila a fila (-q)
- *--result-file=* Garda a saída no ficheiro indicado (-r *ficheiro*)

Para táboas transaccionales (InnoDB, BDB):

--single-transaction. Envorca as táboas transaccionales nun estado consistente, sen bloquear ningunha aplicación. É incompatible con *--lock-tables*. Para táboas grandes é recomendable combinar con *--quick*

Nota: Estas opcións poden gravarse no ficheiro de configuración, dentro da sección [mysqldump]

Veranse a continuación algúns exemplos¹ de uso.

Copia de seguridade básica

O seguinte comando realiza unha copia de seguridade completa dunha base de datos chamada NOME_BASE_DE_DATOS utilizando o usuario de nome USUARIO coa contrasinal CONTRASINAL:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL NOME_BASE_DE_DATOS > copia_seguridade.sql
```

Se por exemplo o usuario é *root*, a contrasinal tamén é *root* e a base de datos chámase *acme*, o comando que se debe executar é o seguinte:

```
$ mysqldump --user=root --password=root acme > copia_seguridade.sql
```

Se por motivos de seguridade non se quere escribir o contrasinal como parte do comando, pódese cambiar a opción *--password=XX* por *-p*. Ao facelo, MySQL pedirá que se escriba o contrasinal a man cada vez que se realice unha copia de seguridade:

```
$ mysqldump --user=root -p acme > copia_seguridade.sql
Enter password: *****
```



Tarefa 1. Copia de seguridade básica con *mysqldump*.

¹¹ <http://librosweb.es/tutorial/copias-de-seguridad-avanzadas-para-bases-de-datos-mysql/>
<http://librosweb.es/tutorial/como-hacer-copias-de-seguridad-de-una-base-de-datos-mysql/>

Recuperando unha copia de seguridade

Supoñendo que os datos a recuperar se atopan no arquivo *copia_seguridade.sql*, o comando que se debe executar para recuperar a información da base de datos é o seguinte:

```
$ mysql --user=USUARIO --password=CONTRASINAL NOME_BASE_DE_DATOS < copia_seguridade.sql
```

Neste caso execútase o comando *mysql* e non o comando *mysqldump*. Utilizando os mesmos datos que no exemplo anterior, o comando a executar sería:

```
$ mysql --user=root --password=root NOME_BASE_DE_DATOS < copia_seguridade.sql
```

Neste comando non fai falla indicar o nome da base de datos que se está recuperando, porque os arquivos xerados por *mysqldump* xa conteñen esa información. De feito, ao executar este comando de recuperación bórrase a base de datos orixinal e toda a información das súas táboas, para despois inserir toda a información contida no arquivo *copia_seguridade.sql*.

Se a copia de seguridade se fai cunha versión de MySQL moderna e a recuperación da información realízase cunha versión un pouco antiga, é mellor engadir a opción *--skip-opt* ao realizar a copia de seguridade, para desactivar algunhas opcións modernas e incompatibles:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL --skip-opt NOME_BASE_DE_DATOS > copia_seguridade.sql
```



Tarefa 2. Restaurar unha copia de seguridade con *mysql*.

Copias de seguridade de máis dunha base de datos

Normalmente o comando *mysqldump* utilízase para realizar a copia de seguridade dunha única base de datos. No entanto, en ocasións é necesario copiar varias bases de datos. Para iso, utilízase a opción *--databases* e indícase o nome de todas as bases de datos separados por un espazo en branco:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL --databases NOME_BASE_DE_DATOS_1 NOME_BASE_DE_DATOS_2 NOME_BASE_DE_DATOS_3 > copia_seguridade.sql
```

Se o que se quere é realizar unha copia de seguridade de todas as bases de datos, utilízase no seu lugar a opción *--all-databases*:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL --all-databases > copia_seguridade.sql
```



Tarefa 3. Copia de seguridade de varias bases de datos con *mysqldump*.

Incluindo soamente algunhas táboas na copia de seguridade

Por defecto, o comando *mysqldump* envorca todas as táboas da base de datos indicada. Para incluír soamente unha ou máis táboas, haberá que indicar os seus nomes despois do nome da base de datos:

```
# envorcando unha soa táboa
$ mysqldump --user=USUARIO --password=CONTRASINAL
NOME_BASE_DE_DATOS NOME_TÁBOA > copia_seguridade.sql
# envorcando tres táboas
$ mysqldump --user=USUARIO --password=CONTRASINAL NOME_BASE_DE_DATOS
NOME_TÁBOA_1 NOME_TÁBOA_2 NOME_TÁBOA_3 > copia_seguridade.sql
```



Tarefa 4. Copia de seguridade con *mysqldump* dalgunhas táboas dunha base de datos.

Excluindo algunhas táboas da copia de seguridade

Para non incluír na copia de seguridade algunha táboa concreta, haberá que indicalo coa opción `--ignore-table`, cuxo valor debe indicarse como `NOME_BASE_DE_DATOS.NOME_TÁBOA` (cando se indica soamente o nome da táboa prodúcese un erro):

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--ignore-table=NOME_BASE_DE_DATOS.NOME_TÁBOA NOME_BASE_DE_DATOS > copia_seguridade.sql
```

Por exemplo, para facer unha copia de seguridade dunha base de datos chamada *tenda* sen que se inclúa unha táboa chamada *vendas*, haberá que executar o seguinte comando:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--ignore-table=tenda.vendas tenda > copia_seguridade.sql
```

Para excluír varias táboas, indicaranse os seus nomes con outras tantas opcións `--ignore-table`:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--ignore-table=NOME_BASE_DE_DATOS.NOME_TÁBOA_1
--ignore-table=NOME_BASE_DE_DATOS.NOME_TÁBOA_2
--ignore-table=NOME_BASE_DE_DATOS.NOME_TÁBOA_3
NOME_BASE_DE_DATOS > copia_seguridade.sql
```

Limitando o número de rexistros de cada táboa

Por defecto, o comando *mysqldump* envorca todos os rexistros de todas as táboas. Cando se queira filtrar previamente os rexistros, engadirase a opción `--where`, que permite indicar unha condición de tipo WHERE igual á das consultas SELECT:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--where="idade > 18 AND idade < 65"
NOME_BASE_DE_DATOS > copia_seguridade.sql
```

Se o que interesa é simplemente limitar o número de rexistros envorcados para cada táboa, pódese utilizar a opción `--where` xunto co seguinte truco extraído do sitio *StackOverflow*:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--where="1 limit 1000"
NOME_BASE_DE_DATOS > copia_seguridade.sql
```

A opción `--where="1 limit 1000"` fai que só se extraian os primeiros 1.000 rexistros de cada táboa. Para axustar este valor, soamente hai que cambiar o valor 1000 pero sen cambiar a primeira parte (*1 limit*).



Tarefa 5. Copia de seguridade con *mysqldump* dalgúns rexistros dunha táboa.

Envorcando a estrutura das táboas pero non os seus datos

Por defecto, o comando *mysqldump* envorca tanto a estrutura das táboas como toda a súa información. Se o que interesa é envorcar a estrutura das táboas e columnas, hai que utilizar a opción `--non-data`. Así poderase crear outra base de datos exactamente igual pero baleira:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--non-data NOME_BASE_DE_DATOS > copia_seguridade.sql
```

Creando arquivos con liñas máis curtas

Por defecto, o comando *mysqldump* combina centos de instrucións INSERT individuais nunha soa gran instrución INSERT para inserir moitos rexistros á vez:

```
INSERT INTO `NOME_DE_TÁBOA`
VALUES (1, '...', '...', '...'), (2, '...', '...', '...'),
(3, '...', '...', '...'), (4, '...', '...', '...'),
(5, '...', '...', '...'), (6, '...', '...', '...');
```

Este é o comportamento recomendado na maioría das situacións, pero pode provocar erros con sistemas antigos incapaces de procesar liñas de miles de bytes de lonxitude. Igualmente, pódese ter problemas co editor de textos ao intentar abrir un arquivo de copia de seguridade que contén estas liñas tan longas.

Se este é o caso, compre engadir a opción `--extended-insert=false` para facer que cada INSERT se execute coa súa propia instrución:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--extended-insert=false NOME_BASE_DE_DATOS > copia_seguridade.sql
```

O abrir agora o arquivo da copia de seguridade, verase como xa non existen instrucións INSERT múltiples:

```
INSERT INTO `NOME_DE_TÁBOA` VALUES (1, '...', '...', '...');
INSERT INTO `NOME_DE_TÁBOA` VALUES (2, '...', '...', '...');
INSERT INTO `NOME_DE_TÁBOA` VALUES (3, '...', '...', '...');
INSERT INTO `NOME_DE_TÁBOA` VALUES (4, '...', '...', '...');
INSERT INTO `NOME_DE_TÁBOA` VALUES (5, '...', '...', '...');
INSERT INTO `NOME_DE_TÁBOA` VALUES (6, '...', '...', '...');
```

O principal problema desta opción é que a recuperación de datos é varias ordes de magnitude máis lenta que cando se combinan varios rexistros nunha única instrución INSERT.

Evitando o bloqueo das táboas

Por defecto, o comando *mysqldump* bloquea as táboas da base de datos antes de facer o envorcado da información. Este é o comportamento recomendado para evitar inconsistencias ao recuperar a información.

Con todo, cando se está a facer unha copia de seguridade do servidor de produción e este ten moita actividade, este comportamento pode ser inaceptable. Para evitalo, engádese a opción `--lock-tables=false`:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--lock-tables=false NOME_BASE_DE_DATOS > copia_seguridade.sql
```

Creando bases de datos con datos binarios

Por defecto, o comando *mysqldump* envorca toda a información tal e como se almacena na base de datos. Cando se gardan arquivos binarios (como imaxes ou arquivos PDF) nalgunha táboa, poden producirse erros ao procesar despois a información binaria ou ao recuperala.

Para evitar estes problemas, pódese forzar a que MySQL converta a información binaria a un formato hexadecimal máis seguro para transmitilo e recuperalo. Para iso, engádese a opción `--hex-blob`:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--hex-blob NOME_BASE_DE_DATOS > copia_seguridade.sql
```

O principal problema desta opción é que pode aumentar moito o tamaño do arquivo da copia de seguridade.

Cambiando o formato de saída

Por defecto, o comando *mysqldump* fai un envorcado de información en formato SQL. Para utilizar o formato XML, pódese engadir a opción `--xml`:

```
$ mysqldump --user=USUARIO --password=CONTRASINAL
--xml NOME_BASE_DE_DATOS > copia_seguridade.sql
```

O contido do arquivo xerado será algo como o seguinte:

```
<?xml version="1.0"?>
<mysqldump xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <database name="NOME_BASE_DE_DATOS">
    <table_structure name="XXX">
```

```

        <field Field="ID" Type="int(11)" Null="NO" Key="PRI"
            Extra="auto_increment" />
        <!-- ... -->
    </table_structure>
    <table_data name="XXX">
        <row>
            <field name="ID">1</field>
            <!-- ... -->
        </row>
    </table_data>
</database>
</mysqldump>

```



Manual oficial de *mysqldump*:

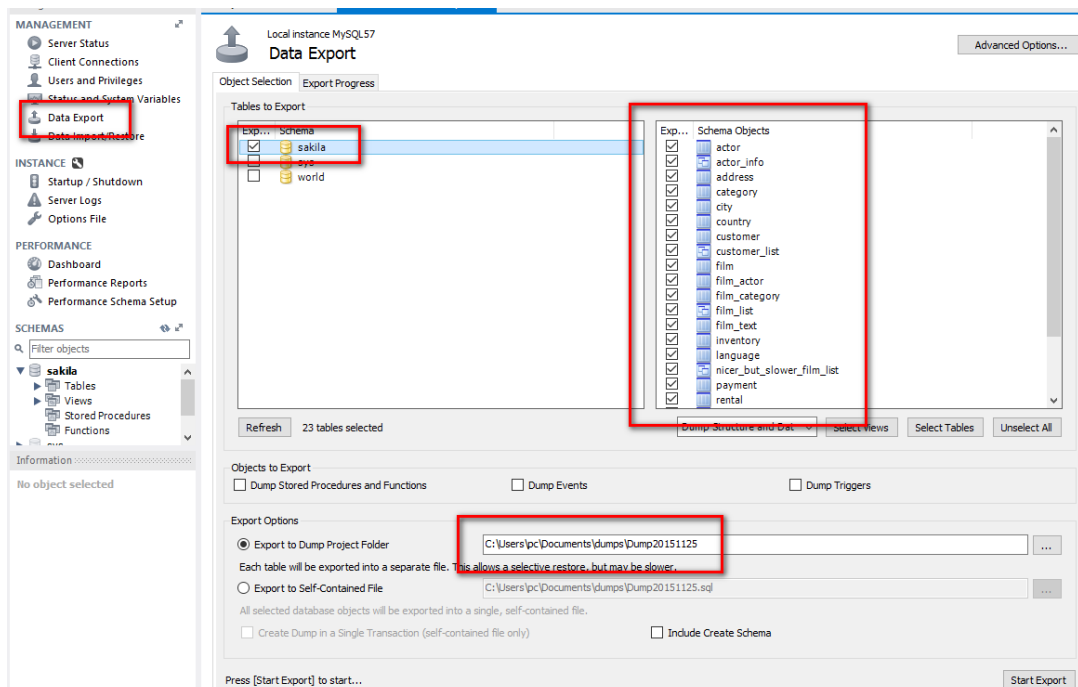
<http://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>

MySQL Workbench

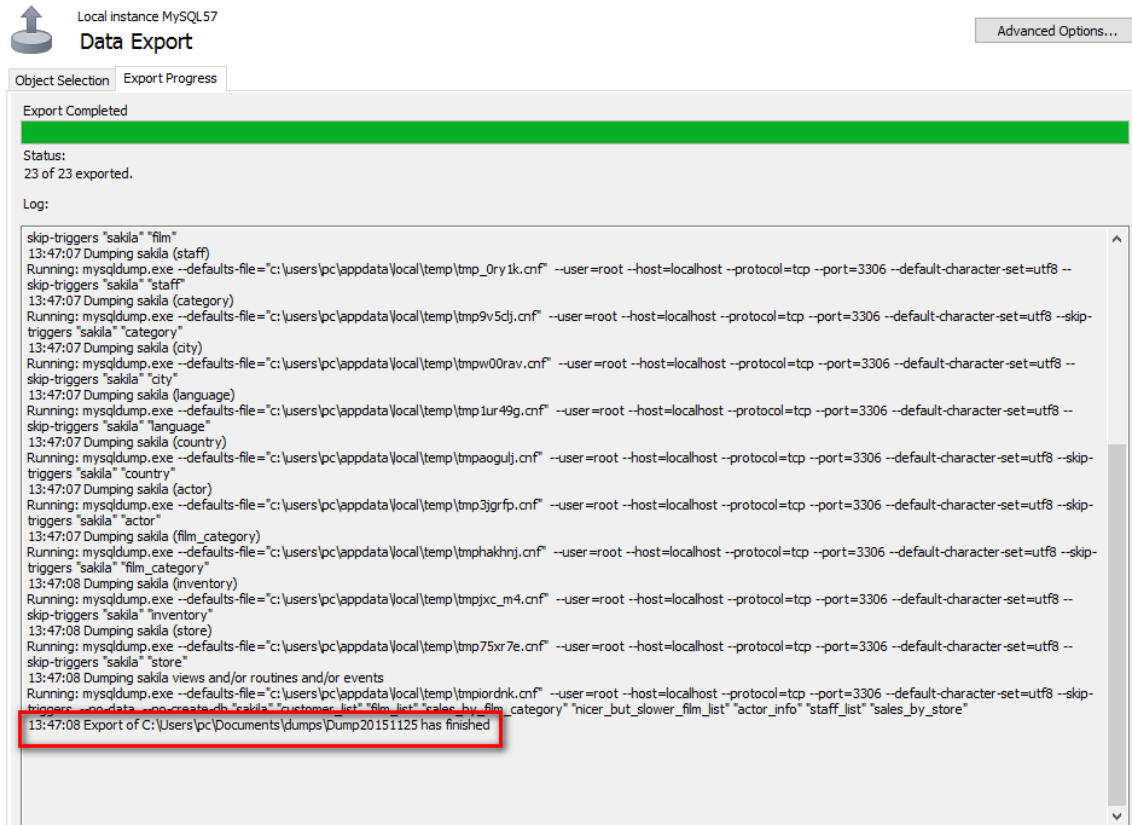
Desde a ferramenta gráfica *MySQL Workbench*, tamén existe a posibilidade de facer copias de seguridade das bases de datos existentes. Para iso utilízanse as opcións de exportación e importación de bases de datos.

Exportar

Mediante a opción *Data Export* seleccionárase a base de datos e as táboas das cales se quere facer unha copia de seguridade así como a situación do ficheiro de copia.

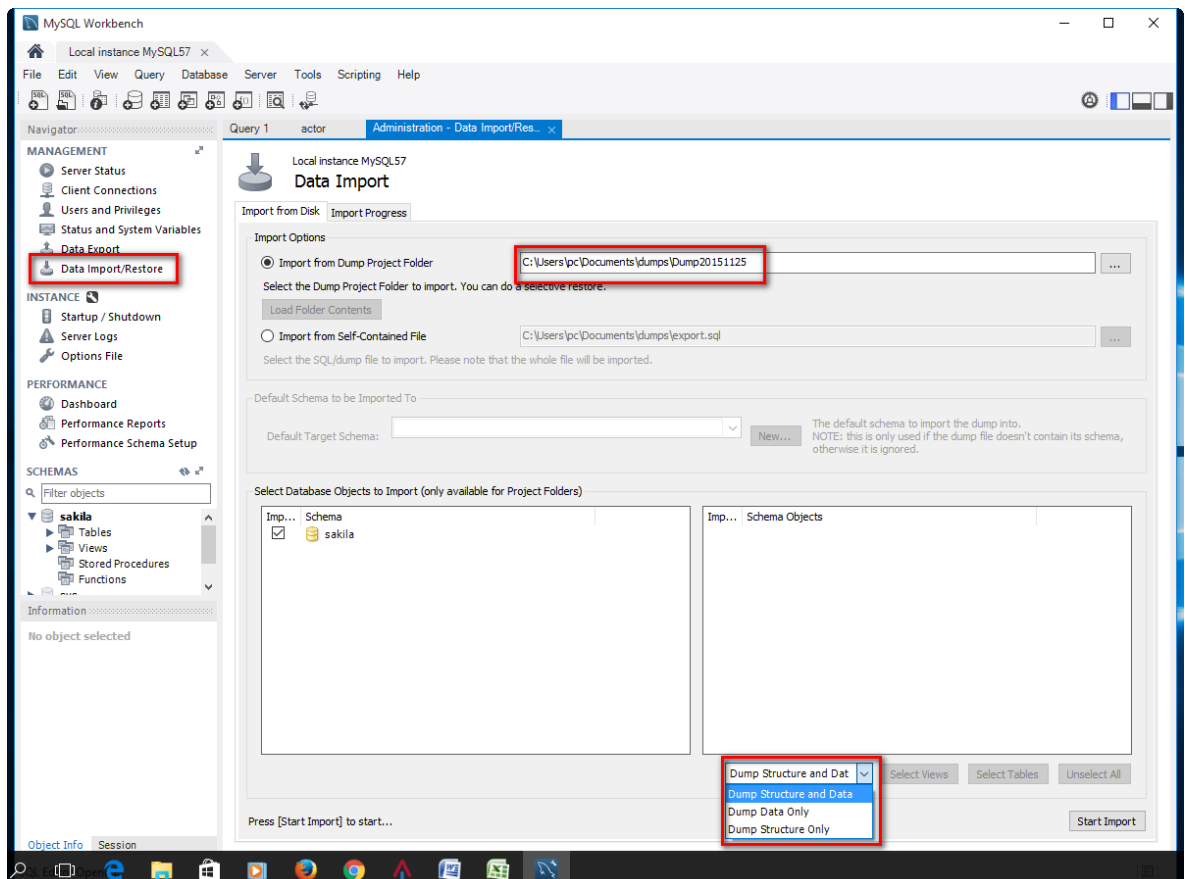


Unha vez seleccionadas as opcións desexadas, pulsárase o botón "*Data Export*" é, se non houbo erros, concluírase a copia de seguridade:

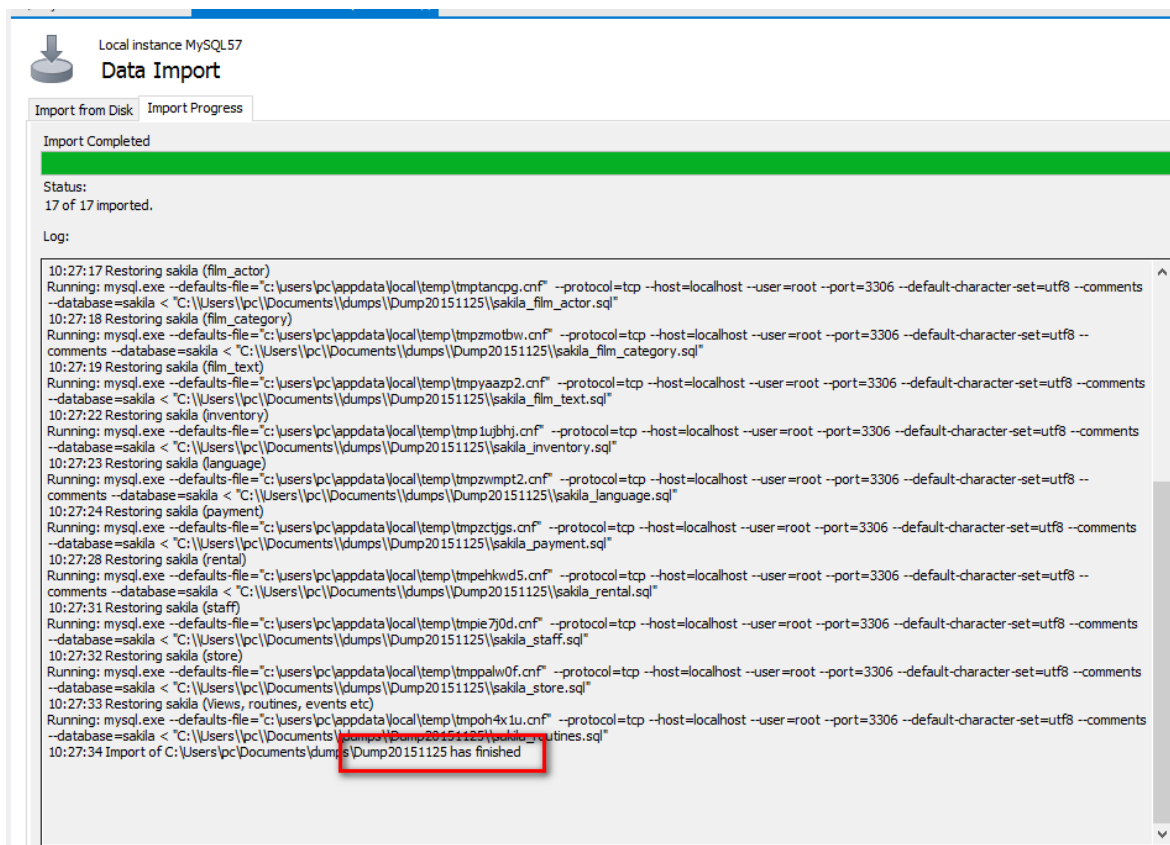


Importar

Para restaurar unha copia de seguridade feita mediante a opción *DataExport* de *MySQL Workdench*, utilizarase a opción *Data Import/Restore*. Unha vez dentro, seleccionárase o directorio onde se gardou a copia de seguridade. É importante saber que se pode seleccionar entre restaurar solo o esquema, os datos ou as dúas cousas:



Unha vez seleccionadas as opcións desexadas, pulsarase o botón "*Start Import*" é, se non houbo erros, concluirase a restauración da copia de seguridade:



1.1.2 Intercambio de datos entre SXBDs

Nalgunhas ocasións é necesario realizar un intercambio de datos, total ou parcial, entre distintos SXBDs. Para facilitar esta tarefa existen ferramentas, externas ou ben integradas có propio SXBD, que permiten a exportación e importación de datos entre esquemas de bases de datos, utilizando para iso distintas estratexias.

1.1.2.1 Intercambio de datos mediante scripts SQL

Unha forma doada de intercambiar datos entre SXBDs, tanto se son iguais ou de distinto fabricante, é mediante scripts SQL.

Como xa se viu no apartado anterior, existen ferramentas e comandos que permiten crear copias de seguridade que dan como resultado ficheiros con sentenzas SQL de creación da estrutura da base de datos e de inserción de datos nas táboas. Importando estes ficheiros noutro SXBD poderíase replicar toda ou parte da base de datos orixinal así como os datos almacenados nela.

No caso de que se quixera facer a importación/exportación entre modelos de SXBD diferentes, habería que asegurarse que existe unha completa compatibilidade entre a versión SQL empregada polos dous sistemas xestores. Por exemplo, é habitual que haxa problemas cos distintos tipos de datos utilizados para almacenar datos.

Nas tarefas 1-5 víronse exemplos de utilización de *mysqldump* que xeran scripts SQL que poden ser utilizados por outros SXBDs para replicar a estrutura e os datos da base de datos orixinal.



Tarefa 7. Intercambiar datos mediante scripts SQL con MySQL Workbench.

1.1.2.2 Intercambio de datos mediante outros formatos intermedios

Importar datos dende un ficheiro de texto

É posible importar datos nas táboas dunha base de datos dende ficheiros de texto plano. Practicamente tódalas aplicacións deseñadas para procesar datos (follas de cálculo, sistemas de xestión de bases de datos, ...), teñen a capacidade de importar e exportar datos dende/a ficheiros de texto (ASCII) de campos delimitados.

Un exemplo de ficheiro de texto de campos delimitados podería ser o seguinte (persoal.txt):

```
1112345;"Martínez Salas, Fernando";"PROFESOR";2200.00;10
4123005;"Bueno Zarco, Elisa";"PROFESOR";2200.00;10
4122025;"Montes García, M.Pilar";"PROFESOR";2200.00;10
1112346;"Rivera Silvestre, Ana";"PROFESOR";2050.00;15
9800990;"Ramos Ruiz, Luis";"PROFESOR";2050.00;15
8660990;"De Lucas Fdez, M.Angel";"PROFESOR";2050.00;15
7650000;"Ruiz Lafuente, Manuel";"PROFESOR";2200.00;22
43526789;"Serrano Laguía, María";"PROFESOR";2050.00;45
4480099;"Ruano Cerezo, Manuel";"ADMINISTRATIVO";1800.00;10
1002345;"Albarrán Serrano, Alicia";"ADMINISTRATIVO";1800.00;15
7002660;"Muñoz Rey, Felicia";"ADMINISTRATIVO";1800.00;15
5502678;"Marín Marín, Pedro";"ADMINISTRATIVO";1800.00;22
```

Estes ficheiros se caracterizan por estar almacenados en formato ASCII, habitualmente coas extensións: *.txt* ou *.csv*. Teñen os rexistros de lonxitude variable separados polo carácter de nova liña: `\n` (ASII 10) en Unix e Mac e `\r\n` (ASCII 10 e 13) en Windows, os campos están delimitados por un carácter especial (neste caso o “;”) e habitualmente os datos de tipo texto van pechados entre comiñas.

MySQL proporciona dous métodos para importar directamente datos dende este tipo de ficheiros nas táboas. Unha delas é a utilidade *mysqlimport*, e o outro método consiste en executar a sentenza `LOAD DATA INFILE`. Veranse a continuación os dous métodos.

LOAD DATA INFILE

A sintaxe da sentenza `LOAD FILE` é a seguinte:

```
LOAD DATA [LOCAL] INFILE 'path/nome_ficheiro.txt'
[REPLACE | IGNORE]
INTO TABLE nome_táboa
[FIELDS
  [TERMINATED BY '\t']
  [[OPTIONALLY] ENCLOSED BY '']
  [ESCAPED BY '\\ ' ]
]
[LINE
  [STARTING BY '']
  [TERMINATED BY '\n']
]
[IGNORE número LINES]
[(nome_columna,...)]
```

A maioría das cláusulas desta sentenza son optativas. Comentaranse a continuación as máis interesantes.

- Cando se especifica a cláusula `LOCAL`, significa que o ficheiro está gardado no equipo no que se está executando a aplicación cliente de MySQL dende a que se está executando a sentenza `LOAD DATA`. Cando se omite a cláusula `LOCAL`, significa que o ficheiro está gardado no equipo no que está instalado o servidor MySQL.
- As cláusulas `REPLACE` e `IGNORE` controlan o tratamento das filas cando a importación provoca que aparezan valores duplicados en columnas con claves únicas. Cando se especifica `REPLACE`, as filas importadas que teñan valores duplicados

substitúen ás existentes. Cando se especifica IGNORE, as filas importadas que teñan valores duplicados ignoranse. No caso de non especificar ningunha opción, o comportamento depende de que se especifique a cláusula LOCAL ou non; cando non se escribe a cláusula LOCAL, prodúcese un erro cando atopa un valor de clave duplicado, e o resto do ficheiro de texto non se ten en conta; cando se escribe a cláusula LOCAL, o comportamento é o mesmo que cando se especifica a opción IGNORE.

Habrá que ter coidado, ao facer unha carga masiva de datos con esta sentenza, para evitar entradas duplicadas en campos marcados como UNIQUE ou PRIMARY KEY, e ter en conta as posibles restricións de clave foránea que poidan existir nas táboas.

- As cláusulas FIELDS e LINES fan referencia aos caracteres empregados para marcar o inicio e remate dos campos e das liñas. No caso dos campos, tamén se pode indicar cal é o carácter que se vai a empregar como delimitador para os campos alfanuméricos.
- A cláusula IGNORE se utiliza, normalmente, para eliminar as liñas de cabeceira que poda ter o ficheiro, indicando o número de liñas que hai que omitir. De forma optativa, pódese indicar o nome das columnas que corresponden a cada campo e o orden que teñen nas liñas do ficheiro de texto, poñéndoas entre paréntese.

```
/* Exemplo de importación de datos dende o ficheiro persoal.txt do exemplo anterior:
*/
USE TEST;
CREATE TABLE IF NOT EXISTS persoal(
    dni int(10),
    apellidos varchar(30),
    funcion enum('PROFESOR','ADMINISTRATIVO','CONSERJE'),
    salario float(6,2),
    centro_traballo int(5),
    CONSTRAINT pk_codigo PRIMARY KEY (dni) );
LOAD DATA LOCAL INFILE 'C:\\temp\\persoal.txt'
    INTO TABLE persoal
    FIELDS TERMINATED BY ';' ENCLOSED BY '"'
    LINES TERMINATED BY '\r\n';
```

É moi importante construír correctamente as cláusulas FIELDS e LINES da sentenza LOAD DATA. No exemplo, na cláusula FIELDS indícase que o carácter delimitador é o “;” e que as cadeas de texto van pechadas entre comiñas; e na cláusula LINES indícase que o carácter de final de liña é ‘\r\n’. (Os ficheiros de texto de Windows soen levar \r\n como carácter de final de liña mentres que os ficheiros de texto en Linux levan unicamente \n)².

Cando se queira ignorar as restricións de clave foránea durante a operación de carga, pódese executar o comando SET FOREIGN_KEY_CHECKS=0 antes de executar LOAD DATA.

É importante saber que para poder executar unha sentenza LOAD DATA é necesario ser administrador da base de datos ou ter concedido o privilexio FILE.



Tarefa 8. Importar datos con LOAD DATA INFILE.

Mysqlimport

Mysqlimport proporciona unha interface de liña de comandos para o comando LOAD DATA INFILE. De feito, varias das opcións de *mysqlimport* correspóndense directamente con cláusulas de LOAD DATA INFILE.

² Para ver os caracteres de final de liña dun ficheiro pódese utilizar o editor Notepad++ e seleccionar, no menú principal, a opción Ver – Mostrar símbolo – Mostrar salto de liña.

A sintaxe de *mysqlimport* é a seguinte:

```
shell> mysqlimport [opcións] nome_de_base_de_datos fichero_de_texto1 [fichero_de_texto2 ...]
```

Do nome de cada ficheiro de texto especificado na liña de comandos, *mysqlimport* elimina calquera extensión, e utiliza o resultado para determinar o nome da táboa á que importar o contido do ficheiro. Por exemplo, os ficheiros con nomes *patient.txt*, *patient.text* e *patient* importaríanse todos á táboa chamada *patient*.

mysqlimport soporta as seguintes opcións:

- *--fields-terminated-by=...*, *--fields-enclosed-by=...*, *--fields-optionally-enclosed-by=...*, *--fields-escaped-by=...*, *--lines-terminated-by=...*, *--ignore,-i* *--replace,-r* *--ignore-lines=n* *.-* Estas opcións teñen o mesmo significado que as cláusulas correspondentes de LOAD DATA INFILE.
- *--help, -?* *.-* Mostra unha mensaxe de axuda.
- *--columns=lista_de_columnas, -c lista_de_columnas.* *.-* Esta opción admite unha lista de nomes de columnas separados por comas. A orde dos nomes de columna indica como emparellar as columnas dos ficheiros de datos coas columnas da táboa.
- *--compress, -C* *.-* Comprime toda a información enviada entre o cliente e o servidor, se ambos soportan compresión.
- *--debug[=opcións_de_depuración], -# [opcións_de_depuración].-* Escribe un log de depuración.
- *--delete, -D* *.-* Baleira a táboa antes de importar o ficheiro de texto.
- *--force, -f* *.-* Ignora erros. Por exemplo, cando unha táboa para un ficheiro de texto non existe, segue procesando o resto de ficheiros. Sen *--force*, *mysqlimport* finaliza se a táboa non existe.
- *--host=nome_de_equipo, -h nome_de_equipo* *.-* Importa datos ao servidor MySQL no equipo dado. O equipo por defecto é localhost.
- *--local, -L* *.-* Le os ficheiros de entrada localmente do equipo cliente.
- *--lock-tables, -l* *.-* Bloquea todas as táboas para escritura antes de procesar calquera ficheiro de texto. Isto asegura que todas as táboas estean sincronizadas no servidor.
- *--password[=contrasinal], -p[contrasinal].-* O contrasinal a usar cando se conecta ao servidor. Cando se usa a opción na súa forma curta (-p), non pode haber un espazo entre a opción e o contrasinal. Cando se omite o valor de contrasinal a continuación de *--password* ou *-p* na liña de comandos, aparece un prompt para que o introduza.
- *--port=número_de_porto, -P número_de_porto* *.-* O porto TCP/IP para usar na conexión.
- *--protocol={TCP | SOCKET | PIPE | MEMORY}.* *.-* O protocolo de conexión a usar.
- *--user=nome_de_usuario, -u nome_de_usuario* *.-* O nome de usuario MySQL a usar cando se conecta co servidor.
- *--verbose, -v.* *.-* Modo explícito. Mostra máis información sobre o que fai o programa.
- *--version, -V* *.-* Mostra información de versión.

Un exemplo do uso de *mysqlimport* podería ser:

```
shell> mysqlimport --local test impctest.txt
```



Tarefa 9. Importar datos con *mysqlimport*.