1.1 BASES DE DATOS MYSQL

MySQL es un sistema de base de datos ampliamente utilizando, sobre todo por aplicaciones escritas en lenguajes como PHP, Python, C++, etc.

Acceder y manipular bases de datos son tareas de administración muy socorridas para un administrador de sistemas y Bash puede ayudar con dicha labor.

1.1.1 Ejemplo completo: base de datos de estudiantes

En este apartado se va a tomar como base un entorno real (una base de datos de estudiantes almacenada en un fichero de texto separado por comas –CSV-) y se creará y manipulará una base de datos que almacena dicha información.

1.1.1.1 Instalación de MySQL

Antes de nada, debe tener acceso al algún servidor MySQL. En caso de no disponer de ninguno puede llevar a cabo la instalación en Ubuntu mediante:

```
$ sudo apt-get install mysql-server
```

De cara a verificar que la instalación ha sido satisfactoria y el servidor MySQL se está ejecutando puede utilizar:

```
$ sudo netstat -tap | grep mysql
```

```
programacion@shell:~$ sudo netstat -tap | grep mysql
tcp        0      0 localhost:mysql         *:*           ESCUCHAR
31760/mysqld
```

O bien:

```
$ sudo service mysql status
```

```
programacion@shell:~$ sudo service mysql status
mysql start/running, process 31760
```

Para poder acceder desde un equipo cliente debe llevar a cabo la instalación de *mysql-client* (en el servidor se instala junto al paquete servidor) mediante:

```
$ sudo apt-get install mysql-client
```

Puede comprobar el acceso al servidor de bases de datos mediante:

```
$ mysql –u usuario -p
```

1.1.1.2 Fichero de estudiantes

Para realizar a cabo las pruebas, se dispone un fichero de estudiantes en formato *csv* con la siguiente información:

- Identificador.
- Nombre y Apellidos.
- Año de nacimiento.

- Ciudad de procedencia.

Un fichero de datos puede ser el siguiente (*estudiantes.csv*):

```
1,Gonzalo    Puga   Sabio,1982,Granada
2,Julio      Gomez  Lopez,1975,Murcia
3,Jose Perez  Perez,1958,Almeria
4,Magdalena Fernandez Fernandez,1980,Sevilla
```

1.1.1.3 Creación de base de datos

El primer paso a llevar a cabo es crear una base de datos con una estructura adecuada para almacenar los datos de los estudiantes. Todo el proceso se puede realizar mediante el script Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**-1:

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**-1 Crear base de datos y tabla de estudiantes**

```
#!/bin/bash
# crearBD.sh
# Crea la base de datos y la tabla de estudiantes

#Datos de conexión
USUARIO="usuario"
PASS="password"

#Conectar con la base de datos
mysql -u $USUARIO -p$PASS <<EOF 2> /dev/null

#Creación de la base de datos
create database estudiantes;
EOF
[ $? -eq 0 ] && echo "BD creada con éxito" || echo "La BD ya existe"

#Creación de la tabla de estudiantes
mysql -u $USUARIO -p$PASS estudiantes <<EOF 2> /dev/null
CREATE TABLE estudiantes(
id int,
nombre varchar(100),
nacimiento int,
ciudad varchar(50)
);
EOF
[ $? -eq 0 ] && echo "Tabla de estudiantes creada con éxito" || echo "La tabla
de estudiantes ya existe"

mysql -u $USUARIO -p$PASS estudiantes <<EOF
DELETE FROM estudiantes;
EOF
```

1.1.1.4 Volcado de datos

Una vez creada la base de datos y la tabla de estudiantes, será necesario procesar el fichero de estudiantes para almacenarlos ella. Esto se consigue mediante el script Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**-2:

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**-2 Insertar estudiantes en la base de datos**

```
#!/bin/bash
# escribirBD.sh
# Lee los datos del fichero csv y los almacena en la BD MySQL

#Datos de conexión
USUARIO="usuario"
PASS="password"
```

```
#Comprobación de los parámetros de entrada
if [ $# -ne 1 ]; then
       echo "Uso: `basename $0` ficheroDatos"
       echo
       exit 2
fi

#Lectura del fichero de entrada
datos=$1
while read line;
do
       oldIFS=$IFS
       IFS=,
       valores=($line)
       valores[1]="\"`echo ${valores[1]} | tr ' ' '#' `\""
       valores[3]="\"`echo ${valores[3]}`\""
       consulta=`echo ${valores[@]} | tr ' #' ', ' `
       IFS=$oldIFS
       mysql -u $USUARIO -p$PASS estudiantes <<EOF
       INSERT INTO estudiantes VALUES($consulta);
EOF
done< $datos

echo "Datos almacenados en la BD"
```

Es conveniente explicar algunos aspectos de este script. Como se ha indicado, se encarga de procesar el fichero de datos separados por coma y crear la consulta de inserción de usuarios en la base de datos.

Está formado por un bucle *while* que procesa los datos del fichero y los almacena en un vector. Para facilitar el proceso de detección de los diferentes campos, se ha jugado con el valor de *IFS* el cual se ha asignado al separador del fichero, las comas.

La información del usuario está compuesta por un identificador y el año de nacimiento, ambos enteros, y por el nombre y la ciudad de procedencia, cadenas de texto. Dichas cadenas de texto se deben escapar, pero además presentan un problema añadido: los espacios en blanco. Estos espacios pueden ser conflictivos con el separador de campos internos (*IFS*), por lo que para evitar dichos conflictos los espacios han sido reemplazados por el carácter '*#*', el cual se vuelve a reemplazar por el espacio a la hora de formular la consulta de inserción en la base de datos.

1.1.1.5 Consulta de la base de datos

Cuando se tienen todos los datos almacenados, se pueden llevar a cabo ciertas consultas sobre los mismos. Por ejemplo, puede utilizar el Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**-3 para mostrar los estudiantes ordenados por su ciudad de procedencia:

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**-3 Consultar la base de datos**

```
#!/bin/bash
# consultarBD.sh
# Consulta de la base de datos

#Datos de conexión
USUARIO="usuario"
PASS="password"

#declaración de la consulta
ciudades=`mysql -u $USUARIO -p$PASS estudiantes <<EOF | tail -n +2
SELECT DISTINCT ciudad FROM estudiantes;
```

```
EOF`

#Listado
for d in $ciudades;
do
      echo "Ciudad de procedencia : $d"
      resultado="`mysql -u $USUARIO -p$PASS estudiantes <<EOF
      SET @i:=0;
      SELECT @i:=@i+1 as numero,nombre,nacimiento FROM estudiantes WHERE
ciudad="$d" ORDER BY id DESC;
EOF`"
echo "$resultado"
echo
done
```

La primera consulta se utiliza para identificar el nombre de las ciudades de procedencia. Posteriormente se utiliza un bucle *while* para iterar a través de las distintas ciudades y mostrar la información de los usuarios que proceden de ella.

SET @i=0 es una construcción *SQL* utilizada para asignar el valor de la variable *i (i=0)*. Con cada nueva fila que se obtiene de la base de datos se incrementa el valor de dicha variable, de cara a reflejar el número de orden del estudiante.

Con esto se da por acabado el ejemplo completo propuesto sobre MySQL. En los siguientes apartados se van a presentar diversos scripts de indudable utilidad para todo administrador de bases de datos MySQL.

1.1.2 Backup de base de datos MySQL

El Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**-4 se puede utilizar para llevar a cabo una copia de seguridad completa de una base de datos MySQL:

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**-4 Backup de base de datos MySQL completa**

```
#!/bin/bash
# backupBD.sh
# Backup completo de una base de datos MySQL

### Parámetros del servidor MySQL ###
USUARIO='usuario'
PASS='pass'
HOST="127.0.0.1"

### Asigne el valor 1 si necesita observar el estado del proceso ###
VERBOSE=0

### Rutas de las utilidades necesarias ###
GZIP=/bin/gzip
MYSQL=/usr/bin/mysql
MYSQLDUMP=/usr/bin/mysqldump
RM=/bin/rm
MKDIR=/bin/mkdir
MYSQLADMIN=/usr/bin/mysqladmin
GREP=/bin/grep

### Directorio de destino ###
BACKUPDIR=/temporal/mysql

########################################
### ----[ No modifique abajo ]------###
########################################

### Formato de fecha ###
TIME_FORMAT='%H_%M_%S%P'
```

```
### Realiza el backup ###
function backup_mysql_BD
{
        local DBS="$($MYSQL -u $USUARIO -h $HOST -p$PASS -Bse 'show
databases')"
        local db="";
        [ ! -d $BACKUPDIR ] && ${MKDIR} -p $BACKUPDIR
        ${RM} -f $BACKUPDIR/* >/dev/null 2>&1
        [ $VERBOSE -eq 1 ] && echo "*** Volcando BD MySQL ***"
        [ $VERBOSE -eq 1 ] && echo -n "Base de datos> "
        for db in $DBS
        do
             local tTime=$(date +"${TIME_FORMAT}")
             local FILE="${BACKUPDIR}/${db}.${tTime}.gz"
             [ $VERBOSE -eq 1 ] && echo -n "$db.."
             ${MYSQLDUMP} -u ${USUARIO} -h ${HOST} -p${PASS} $db --skip-lock-
tables | ${GZIP} -9 > $FILE
        done
             [ $VERBOSE -eq 1 ] && echo ""
             [ $VERBOSE -eq 1 ] && echo "*** Backup completa [ Guardada en
$BACKUPDIR] ***"
}

### Salida ###
function salir
{
      echo "$@"
      exit 999
}

### Comprobar la existencia de los binarios
### En caso de error se sale del script
function verificarBinarios
{
      [ ! -x $GZIP ] && salir "$GZIP no existe. Compruebe la ruta del mismo en
$0."
      [ ! -x $MYSQL ] && salir "$MYSQL no existe. Compruebe la ruta del mismo
en $0."
      [ ! -x $MYSQLDUMP ] && salir "$MYSQLDUMP no existe. Compruebe la ruta
del mismo en $0."
      [ ! -x $RM ] && salir "$RM no existe. Compruebe la ruta del mismo en
$0."
      [ ! -x $MKDIR ] && salir "$MKDIR no existe. Compruebe la ruta del mismo
en $0."
      [ ! -x $MYSQLADMIN ] && salir "$MYSQLADMIN no existe. Compruebe la ruta
del mismo en $0."
      [ ! -x $GREP ] && salir "$GREP no existe. Compruebe la ruta del mismo en
$0."
}

### Comprobar la conexión con el servidor.
### En caso contrario se sale del script
function verificar_mysql_conexion
{
      $MYSQLADMIN  -u $USUARIO -h $HOST -p$PASS ping | $GREP 'alive'>/dev/null
      [ $? -eq 0 ] || salir "Error: No se puede conectar con el servidor
MySQL. Compruebe que su usuario y contraseña están correctamente establecidos
en $0"
}

### Principal ####
verificarBinarios
verificar_mysql_conexion
backup_mysql_BD
```

1.1.3 Borrar todas las tablas de una base de datos

El Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**-5 se puede utilizar para eliminar todas la tablas de una base de datos MySQL:

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**-5 Borrar todas las tablas de una base de datos**

```
#!/bin/bash
# dropTablas.sh
#
# Elimina todas las tablas de una base de datos MysQL.
# Uso: ./script USUARIO PASSWORD BD
# Uso: ./script USUARIO PASSWORD BD IP-Servidor
# Uso: ./script USUARIO PASSWORD BD mysql.servidor.es

#Datos de conexión
USUARIO="$1"
PASS="$2"
BD="$3"

HOST="localhost"

[ "$4" != "" ] && HOST="$4"

# Detección de los binarios
MYSQL=$(which mysql)
AWK=$(which awk)
GREP=$(which grep)

# Comprobación de los argumentos
if [ ! $# -ge 3 ] ; then
       echo "Uso: $0 {MySQL-USUARIO} {MySQL-PASSWORD} {MySQL-BD} [HOST]"
       echo "Elimina todas las tablas de la base de datos"
       exit 1
fi

# Comprobar la conexión con el servidor MySQL
$MYSQL -u $USUARIO -p$PASS -h $HOST -e "use $BD"  &>/dev/null
if [ $? -ne 0 ] ; then
       echo "Error - No se puede conectar al servidor MySQL usando la
información facilitada"
       echo "o bien la base de datos no existe"
       exit 2
fi

#Obtener las tablas de la base de datos
TABLAS=$($MYSQL -u $USUARIO -p$PASS -h $HOST $BD -e 'show tables' | $AWK '{
print $1}' | $GREP -v '^Tables' )

# Comprobar la existencia de tablas
if [ "$TABLAS" == "" ] ; then
       echo "Error - No se han encontrado tablas en la base de datos $BD"
       exit 3
fi

# Borrad de las tablas
for t in $TABLAS
do
       echo "Borrando la tabla $t de la base de datos $BD ..."
       $MYSQL -u $USUARIO -p$PASS -h $HOST $BD -e "drop table $t"
done
```

1.1.4 Backup de base de datos MySQL en un NAS

A la hora de llevar a cabo la copia de seguridad de una base de datos, lo normal es hacerlo en un equipo distinto al servidor, y en muchas ocasiones en servidores de almacenamiento de datos. El Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**-6 realiza la copia de una base de datos MySQL completa en un servidor NAS:

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**-6 Backup de base de datos MySQL completa en un NAS**

```
#!/bin/bash
# backupNAS.sh
```

```
# Realiza un volcado de la base de datos MySQL en un NAS
#
# Cada volcado se almacena del siguiente modo:
# Directorio:  /nas/mysql/dd-mm-yyyy
# Fichero: mysql-DBNAME.24-08-2011-18:40:12.gz
# Ruta completa: /nas/mysql/dd-mm-yyyy/mysql-DBNAME.24-08-2011-18:40:12.gz


FECHA=$(date +"%d-%m-%Y") # con formato dd-mm-yyyy
FICHERO=""                # usado en los bucles
NASBASE="/nas"            # Punto de montaje del NAS
BAK="${NAS}/mysql/${FECHA}" # Ruta de almacenaje del volcando en $NAS

### Configuración del servidor MySQL ###

#* Usuario *#
USUARIO="usario"

#* Password *#
PASS="password"

#* Host MySQL Server *#
HOST="127.0.0.1"

#* Binarios implicados *#
MYSQL="$(which mysql)"
MYSQLDUMP="$(which mysqldump)"
GZIP="$(which gzip)"

# Comprueba que el NAS este montado
mount | awk '{ print $3}' |grep -w $NASBASE >/dev/null
if [ $? -ne 0 ] ; then
       echo "Error: el NAS NO está montado en $NASBASE"
       echo "Por favor, monte el NAS en un directorio local e inténtelo de
nuevo"
       exit 99
fi

### El NAS debe montarse en modo Advance ###
# Se asume que /nas está montado vía /etc/fstab
if [ ! -d $BAK ]; then
  mkdir -p $BAK
else
 :
fi

# Listar la base de datos completa
DBS="$($MYSQL -u $USUARIO -h $HOST -p$PASS -Bse 'show databases')"

# Volcar las bases de datos una a una
for db in $DBS
do
 FICHERO=$BAK/mysql-$db.$FECHA-$(date +"%T").gz
 # Comprimir cada archivo de backup con gzip
 $MYSQLDUMP -u $USUARIO -h $HOST -p$PASS $db | $GZIP -9 > $FICHERO
done
```

Dada su funcionalidad, se podría programar mediante *cron* para llevar a cabo un volcado horario.

EJERCICIOS PROPUESTOS

1.	Desarrolle un script que cree copias de seguridad todas las bases de datos MySQL completas, salvo aquellas que se indiquen para su exclusión de la copia. El fichero resultante sólo podrá ser

	accedido por el root.
2.	Cree un script que permita añadir bases de datos, usuarios y contraseñas a un servidor MySQL, y que admita además acceso el mismo de modo inmediato mediante la asignación de los permisos pertinentes.
3.	Implemente un script que realice una copia de una base de datos MySQL y la envíe a otro servidor haciendo uso de SSH. Las condiciones que debe cumplir el script son las siguientes: • Exportar la estructura y los datos de la base de datos en un archivo SQL con MySqlDump. • Comprimir el archivo SQL. • Envío del archivo a un servidor remoto mediante SSH (opcional). • Guardar también la copia en un directorio local. Si el número de copias almacenadas supera un valor (variable *MaxBackups*) se eliminarán las versiones más antiguas.