

DBA基础

NSD DBA1

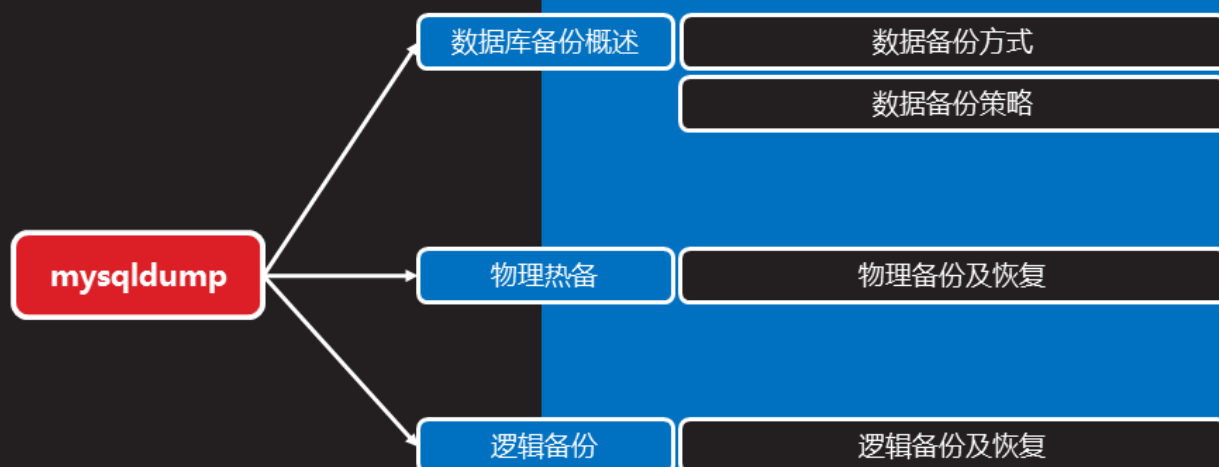
DAY05

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	mysqldump
	10:30 ~ 11:20	实时增量备份/恢复
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	innobackupex
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



mysqldump备份/恢复



数据库备份概述

数据备份方式

- 物理备份
 - 冷备：cp、tar、...
- 逻辑备份
 - mysqldump
 - mysql



数据备份策略

知识讲解

- 完全备份
 - 备份所有数据
- 增量备份
 - 备份上次备份后，所有新产生的数据
- 差异备份
 - 备份完全备份后，所有新产生的数据



物理备份

物理备份及恢复

知识讲解

- 备份操作
 - `cp -rp /var/lib/mysql/数据库 备份目录/文件名`
 - `tar -zcvf xxx.tar.gz /var/lib/mysql/数据库/*`
- 恢复操作
 - `cp -rp 备份目录/文件名 /var/lib/mysql/`
 - `tar -zxvf xxx.tar.gz -C /var/lib/mysql/数据库名/`



逻辑备份

逻辑备份及恢复

知识讲解

- 备份操作
 - `mysqldump -uroot -p 密码 库名 > 路径/xxx.sql`
- 恢复操作
 - `mysql -uroot -p 密码 库名 < 路径/xxx.sql`



逻辑备份及恢复（续1）

知识讲解

- 库名表示方式
 - `--all-databases` 或 `-A` 所有库
 - 数据库名 单个库
 - 数据库名 表名 单张表
 - `-B 数据库1 数据库2` 多个库
- 注意事项
 - 无论备份还是恢复，都要验证用户权限



逻辑备份及恢复（续2）

知识讲解

- 应用示例1
 - 将所有的库备份为mysql-all.sql文件
 - 将userdb库备份为用户db.sql文件

```
[root@dbsvr1 ~]# mysqldump -u root -p --all-databases > alldb.sql
Enter password: //验证密码
[root@dbsvr1 ~]# mysqldump -u root -p userdb > userdb.sql
Enter password: //验证密码
[root@dbsvr1 ~]# ls -lh *.sql
-rw-r--r--. 1 root root 595K 1月  2 13:54 alldb.sql
-rw-r--r--. 1 root root 4.1K 1月  2 13:55 userdb.sql
```



逻辑备份及恢复（续3）

知识讲解

- 应用示例2
 - 从备份文件userdb.sql重建userdb3库
- 实现思路
 - 先建空库，再执行mysql导入

```
mysql> CREATE DATABASE userdb3;
Query OK, 1 row affected (0.04 sec)

mysql> exit
Bye
[root@dbsvr1 ~]# mysql -u root -p userdb3 < userdb.sql
Enter password:
[root@dbsvr1 ~]#
```



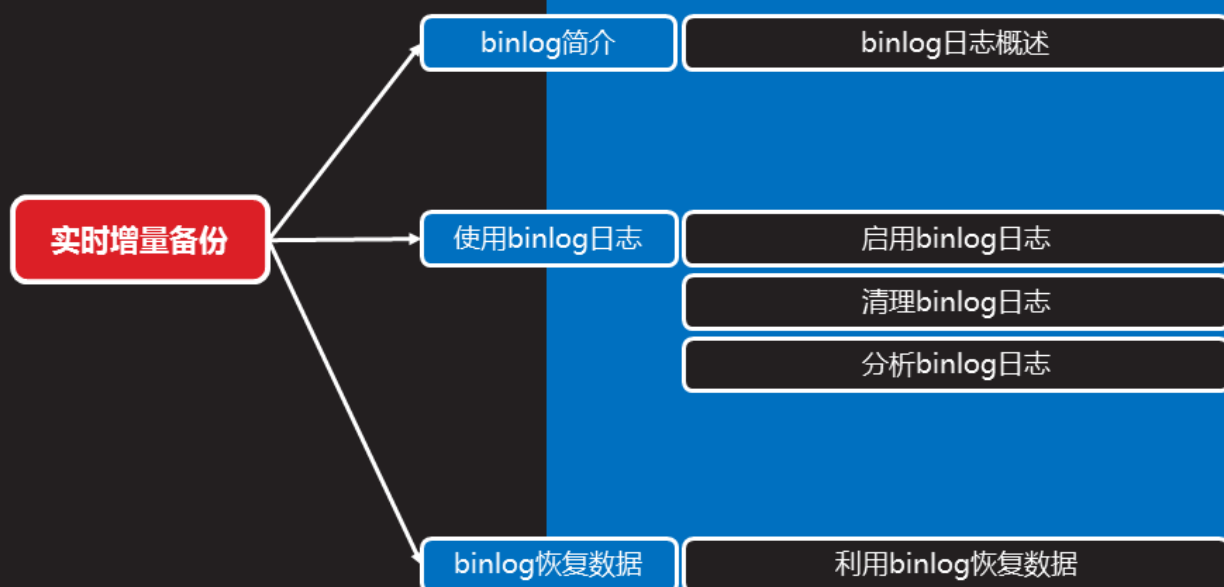
案例1：数据备份与恢复

1. 逻辑备份工具 mysqldump
2. 使用 mysql 恢复数据库

课堂练习



实时增量备份/恢复



binlog简介

binlog日志概述

- 二进制日志用途及配置方式

类 型	用 途	配 置
二进制日志	记录所有更改数据的操作	log_bin[=dir/name] server_id=数字 max_binlog_size=数字m



使用binlog日志

启用binlog日志

知识讲解

- 采用binlog日志的好处
 - 记录除查询之外的所有SQL命令，
 - 可用于数据恢复
 - 配置mysql主从同步的必要条件

```
[root@dbsvr1 ~]# vim /etc/my.cnf
[mysqld]
```

```
...
```

```
log_bin
```

```
server_id=100
```

```
//启用binlog日志
//指定id值
```

```
[root@dbsvr1 ~]# systemctl restart mysqld
```



启用binlog日志 (续1)

知识讲解

- binlog相关文件
 - 主机名-bin.index 记录已有日志文件名
 - 主机名-bin.000001 第1个二进制日志
 - 主机名-bin.000002 第2份二进制日志
 -

手动生成新的日志文件：

1. 重启mysql服务
2. 执行SQL操作 `mysql> flush logs;`
3. `mysqldump --flush-logs`
4. `mysql -uroot -p密码 -e 'flush logs'`



清理binlog日志

知识讲解

- 删除早于指定版本的binlog日志
 - `purge master logs to "binlog文件名";`
- 删除所有binlog日志，重建新日志
 - `reset master;`

```
mysql> PURGE MASTER LOGS TO 'mysql-bin.000003';
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 4
Current database: test
```

```
Query OK, 0 rows affected (0.28 sec)
```

```
mysql> RESET MASTER;
Query OK, 0 rows affected (0.00 sec)
```



分析binlog日志

知识讲解

- 查看日志当前记录格式

```
mysql> show variables like "binlog_format";
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_format | ROW   |
+-----+-----+
```

三种记录格式：

1. statement：每一条修改数据的sql命令都会记录在binlog日志中。
2. row：不记录sql语句上下文相关信息，仅保存哪条记录被修改。
3. mixed：是以上两种格式的混合使用。

- 修改日志记录格式

```
[root@localhost ~]# vim /etc/my.cnf
[mysqld]
```

```
...
binlog_format=mixed
```

```
[root@localhost ~]# systemctl restart mysqld
```



分析binlog日志（续1）

知识讲解

- 使用 mysqlbinlog 工具
 - 格式：mysqlbinlog [选项] binlog日志文件名
- 常用选项
 - --start-datetime="yyyy-mm-dd hh:mm:ss"
 - --stop-datetime="yyyy-mm-dd hh:mm:ss"
 - --start-position=数字
 - --stop-position=数字



分析binlog日志 (续2)

- 应用示例
 - 查看从2014年1月2日15:30开始的更改操作

知识讲解

```
[root@dbsvr1 ~]# mysqlbinlog \  
--start-datetime="2014-01-02 15:30" /var/lib/mysql/mysql-bin.000001  
...  
# at 318 ...  
create database newdb;  
# at 418 ...  
drop database newdb;  
...
```



binlog恢复数据

利用binlog恢复数据

知识讲解

- 基本思路
 - 使用 mysqlbinlog 提取历史SQL操作
 - 通过管道交给 mysql 命令执行
- 应用示例
 - 执行第1份binlog所记录的更改操作

```
# cd /var/lib/mysql/  
# mysqlbinlog mysql-bin.000001 | mysql -uroot -p123456
```



案例2：使用binlog日志

利用binlog恢复库表，要求如下：

- 1) 启用binlog日志
- 2) 创建db1库tb1表，插入3条记录
- 3) 删除tb1表中刚插入的3条记录
- 4) 使用mysqlbinlog恢复删除的3条记录

课堂练习



innobackupex 备份/恢复

MySQL备份工具

常用的MySQL备份工具

XtraBackup工具 ?

Tedu.cn
达内教育

MySQL备份工具

常用的MySQL备份工具

知识讲解

- 物理备份缺点
 - 跨平台性差
 - 备份时间长、冗余备份、浪费存储空间
- mysqldump备份缺点
 - 效率较低，备份和还原速度慢
 - 备份过程中，数据插入和更新操作会被挂起



XtraBackup工具？

知识讲解

- 一款强大的在线热备份工具
 - 备份过程中不锁库表，适合生产环境
 - 由专业组织Percona提供（改进MySQL分支）
- 主要含两个组件
 - xtrabackup：C程序，支持InnoDB/XtraDB
 - innobackupex：以Perl脚本封装xtrabackup，还支持MyISAM



PERCONA
XTRABACKUP



安装percona

安装percona

- 下载适配的RPM包并安装

- percona-xtrabackup-24-2.4.7-1.el7.x86_64.rpm

```
# yum -y install perl-Digest-MD5.x86_64 rsync perl-DBD-MySQL
```

```
# rpm -ivh libev-4.15-1.el6.rf.x86_64.rpm
```

```
# rpm -ivh percona-xtrabackup-24-2.4.7-1.el7.x86_64.rpm
```

```
# rpm -ql percona-xtrabackup-24
```

```
/usr/bin/innobackupex //备份innodb、xtradb、myisam引擎的表
```

```
/usr/bin/xbcloud
```

```
/usr/bin/xbcloud_osenv
```

```
/usr/bin/xbcrypt
```

```
/usr/bin/xbstream
```

```
/usr/bin/xtrabackup //备份innodb、xtradb引擎的表
```

```
.. ..
```

innobackupex基本选项

知识讲解

常用选项	含 义
--host	主机名
--user	用户名
--port	端口号
--password	密 码
--databases	数据库名
--no-timestamp	不用日期命名备份文件存储的子目录名

--databases="库名" //单个库
--databases="库1 库2" //多个库
--databases="库.表" //单个表



innobk应用案例

完全备份与恢复

- 应用示例：将所有库完全备份到 /backup

```
[root@dbsvr1 ~]# innobackupex --user root --password 123456  
/backup --no-timestamp //完全备份
```

```
[root@dbsvr1 ~]# innobackupex --user root --password 123456 -  
-apply-log /backup //准备恢复数据
```

```
[root@dbsvr1 ~]# rm -rf /var/lib/mysql //恢复时要求空的库目录
```

```
[root@dbsvr1 ~]# mkdir /var/lib/mysql
```

```
[root@dbsvr1 ~]# innobackupex --user root --password 123456 -  
-copy-back /backup //恢复数据
```

```
[root@dbsvr1 ~]# chown -R mysql:mysql /var/lib/mysql
```

```
[root@dbsvr1 ~]# systemctl restart mysqld
```

```
[root@dbsvr1 ~]# mysql -uroot -p123456
```

```
mysql> show databases;
```

增量备份与恢复

• 应用示例

- 增量备份时，必须先有一次备份,通常是完全备份

知识讲解

```
#innobackupex --user root --password 123456 \
--databases = "库名列表" /fullbak --no-timestamp //完全备份
```

```
#innobackupex --user root --password 123456 \
--databases = "库名列表" --incremental /new1 \
--incremental-basedir="/fullbak" --no-timestamp //第1次增量备份
```

```
#innobackupex --user root --password 123456 \
--databases="库名列表" --incremental /new2 \
--incremental-basedir="/new1" --no-timestamp //第2次增量备份
```



增量备份与恢复（续1）

知识讲解

```
# rm -rf /var/lib/mysql ; mkdir /var/lib/mysql/
# innobackupex --user root --password 123456 \
--databases = "库名列表" --apply-log --redo-only /fullbak //完整恢复
```

```
# innobackupex --user root --password 123456 \
--databases= "库名列表" --apply-log --redo-only /fullbak \
--incremental-dir="/new1" //恢复增量
```

```
# innobackupex --user root --password 123456 \
--databases= "库名列表" --apply-log --redo-only /fullbak \
--incremental-dir="/new2" //恢复增量
```

```
# innobackupex --user root --password 123456 \
--databases= "库名列表" --copy-back /fullbak //拷贝文件
```

```
# chown -R mysql:mysql /var/lib/mysql/
# systemctl stop mysqld ; systemctl start mysqld
```



在完全备份文件中恢复单个表

- 应用示例
 - 完全备份数据库到/allbak目录
 - 导出表信息

知识讲解

```
[root@dbsvr1 ~]# innobackupex --user root --password 123456
--databases="gamedb" /allbak --no-timestamp
```

```
mysql> drop table gamedb.a;
```

```
[root@dbsvr1 ~]# innobackupex --user root --password 123456 --
databases="gamedb" --apply-log --export /allbak //导出表信息
```

```
mysql> create table gamedb.a(id int); //创建表
```

```
mysql> alter table gamedb.a discard tablespace; //删除表空间
```



从完全备份中恢复单个表（续1）

```
mysql> system cp /allbak/gamedb/a.{ibd,cfg,exp}
/var/lib/mysql/gamedb //拷贝表信息文件
```

```
mysql> system chown mysql:mysql /var/lib/mysql/bbsdb/a.*
//修改所有者
```

```
mysql> alter table gamedb.a import tablespace; //导入表空间
```

```
mysql> select * from gamedb.a;
```

```
+-----+
| id    |
+-----+
| 1001  |
| 1002  |
+-----+
```

知识讲解



案例3：innobackupex备份工具

课堂练习

1. 安装percona软件包
2. innobackupex完整备份、增量备份操作
3. 恢复数据



总结和答疑





