

数据库管理

NSD DBA 基础

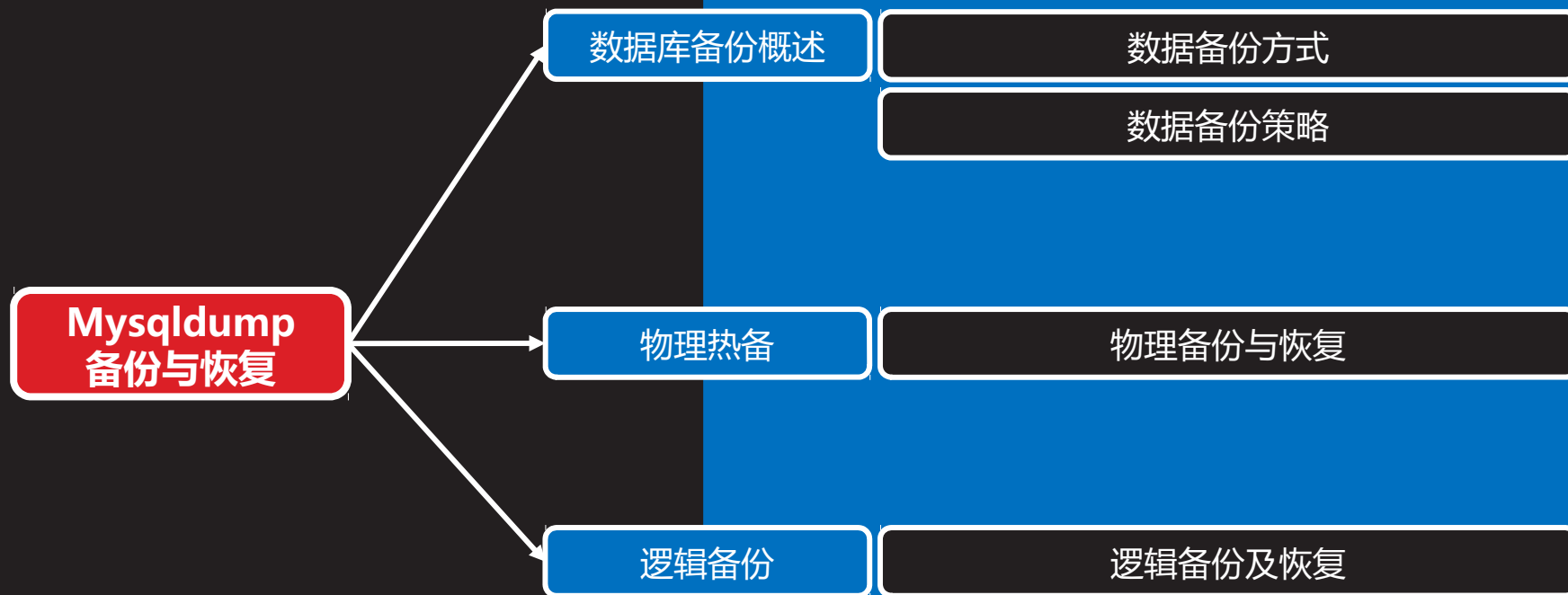
DAY05

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	mysqldump 备份 / 恢复
	10:30 ~ 11:20	实时增量备份 / 恢复
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	innobackupex 备份 / 恢复
	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

- 完全备份
 - 备份所有数据（一台服务器、一个库、一张表）
- 增量备份
 - 备份自上一次备份（包含完全备份、差异备份、增量备份）之后有变化的数据
- 差异备份
 - 备份自上一次完全备份之后有变化的数据



物理备份



物理备份与恢复

- 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

- 备份操作
 - _ mysqldump 源库名 > 路径 /xxx.sql
- 恢复操作
 - _ mysql 目标库名 < 路径 /xxx.sql



逻辑备份及恢复（续 1）

- 源库名的表示
 - --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 备份多个库
- 注意事项
 - 无论备份还是恢复，都要验证用户及权限



逻辑备份及恢复（续 2）

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

- 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 ~]#
```

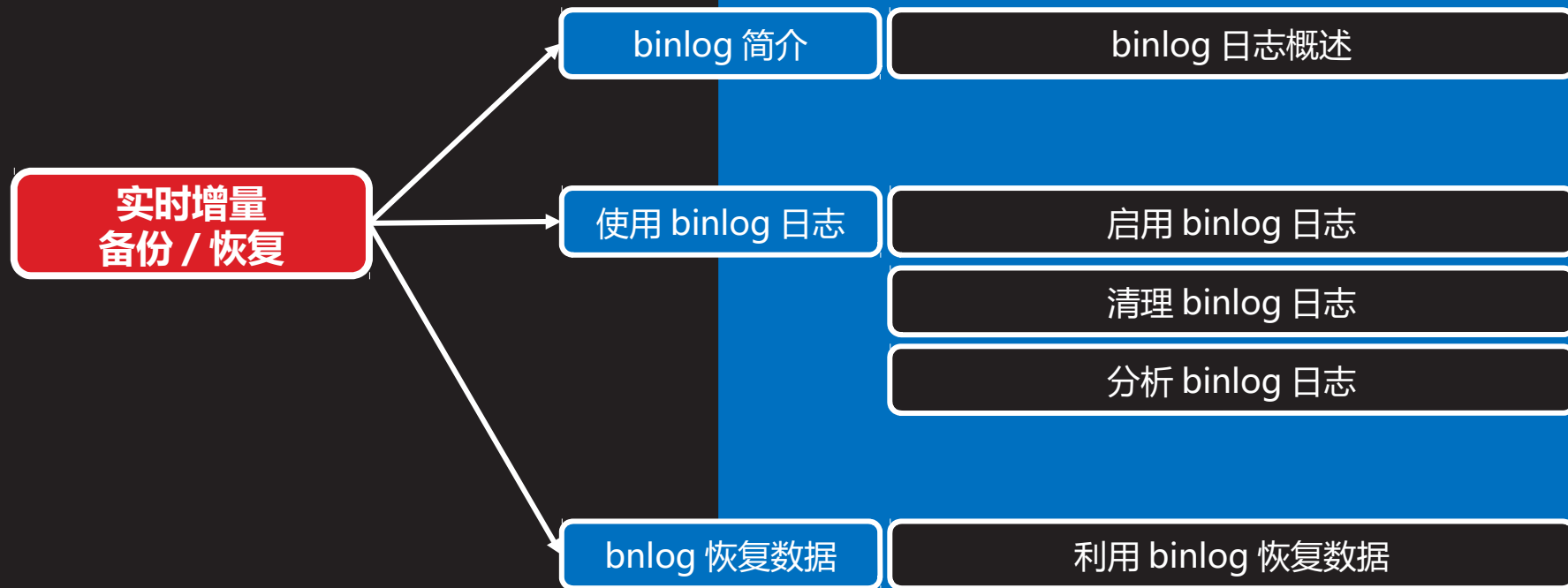


案例 1：数据备份与恢复

1. 逻辑备份工具 mysqldump
2. 使用 mysql 恢复数据库



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

- 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 // 启用 binlog 日志
server_id=100 // 指定 id 值
[root@dbsvr1 ~]# systemctl restart mysqld
```



启用 binlog 日志 (续 1)

• 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

- 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

- 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)

- 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

```
[root@dbsvr1 ~]# mysqlbinlog \
--start-datetime="2014-01-02 15:30" /var/lib/mysql/mysql-bin.000001
...
# at 318 ...
create database newdb2;
# at 418 ...
drop database newdb;
...
```



binlog 恢复数据



利用 binlog 恢复数据

- 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
```



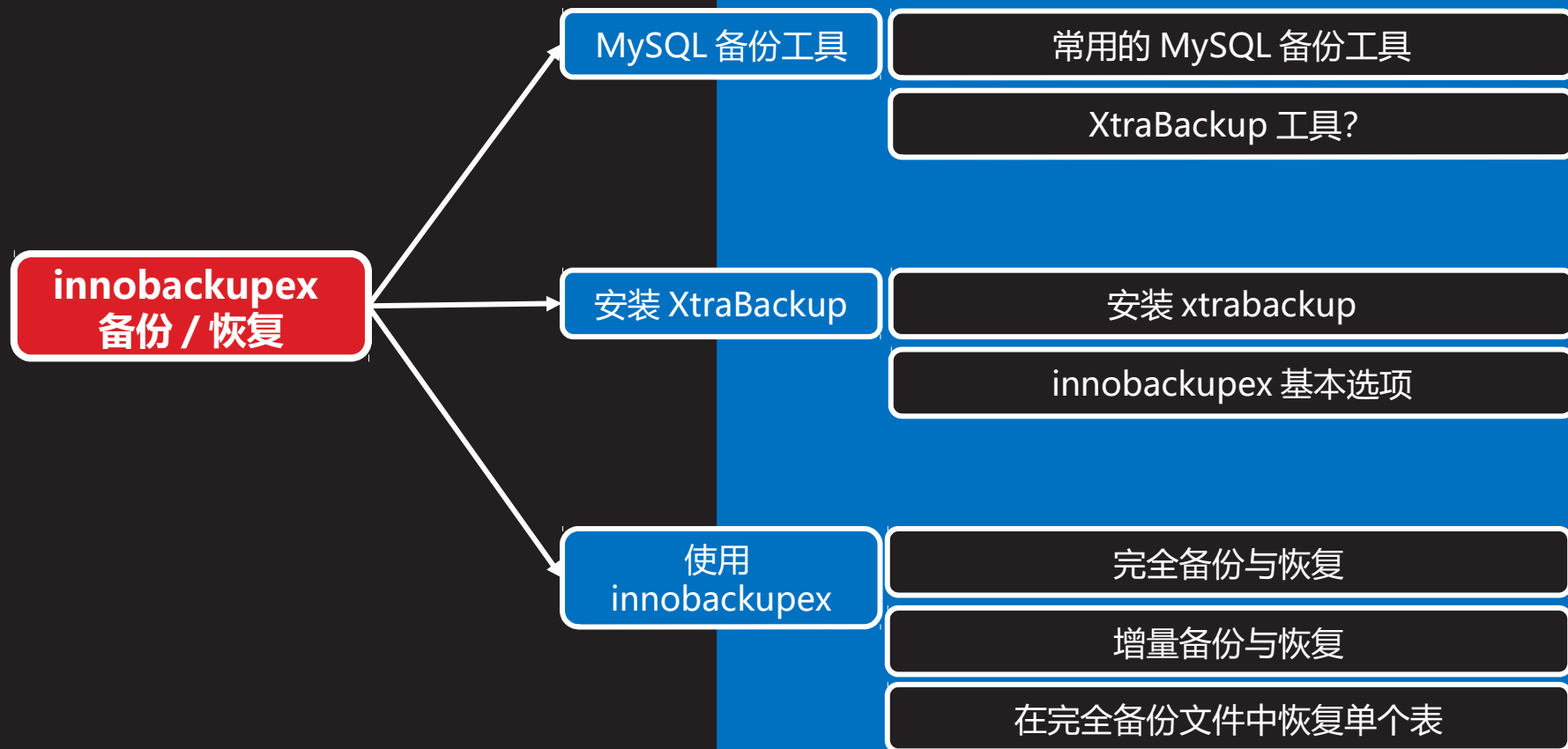
案例 2：使用 binlog 日志

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



innobackupex 备份 / 恢复



MySQL 备份工具



常用的 MySQL 备份工具

- 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?

- 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 XtraBackup

安装 xtrabackup

- 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 、 xtrdb 、 myisam 引擎的表
/usr/bin/xbcloud
/usr/bin/xbcloud_osenv
/usr/bin/xbcrypt
/usr/bin/xbstream
/usr/bin/xtrabackup // 备份 innodb 、 xtrdb 引擎的表
.....
```



innobackupex 基本选项

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
```



innobackupex 基本选项 (续 1)

常用选项	含 义
--redo-only	日志合并
--apply-log	准备还原 (回滚日志)
--copy-back	恢复数据
--incremental 目录名	增量备份
--incremental--basedir= 目录名	增量备份时, 指定上一次备份数据存储的目录名
--incremental-dir= 目录名	准备恢复数据时, 指定增量备份数据存储的目录名
--export	导出表信息
import	导入表空间



使用 innobackupex

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 // 同步日志  
[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 stop mysqld  
[root@dbsvr1 ~]# systemctl start mysqld  
[root@dbsvr1 ~]# mysql -uroot -p123456  
mysql> show databases;
```



增量备份与恢复

- 应用示例
 - 必须先有一次完全备份, 备份到 /allbak
 - 第 1 次增量备份到 /new1
 - 第 2 次增量备份到 /new2

```
#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）

```
#rm -rf /var/lib/mysql ; mkdir /var/lib/mysql/  
#innobackupex --user root --password 123456 \  
--databases = " 库名列表 " --apply-log /fullbak // 恢复完全备份
```

```
#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

```
[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
```



在完全备份文件中恢复单个表 (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  |
+-----+
```



案例 3：innobackupex 备份工具

1. 安装 XtraBackup 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

总结和答疑

启用 binlog 日志

问题现象

故障分析及排除

启用 binlog 日志

问题现象

- 创建 binlog 日志文件失败
 - 报错 1 : Can't create/write to file
 - 报错 2 : Permission denied

```
mysql>  
ERROR 1 (HY000): Can't create/write to file  
  
..... (Errcode: 13 - Permission denied)
```



故障分析及排除

- 原因分析
 - mysql 用户对目标目录没有 w 权限
 - 没有关闭 SELinux
- 解决办法
 - chown mysql 目录名
 - setenforce 0

