

# DBA进阶

**NSD RDBMS2**

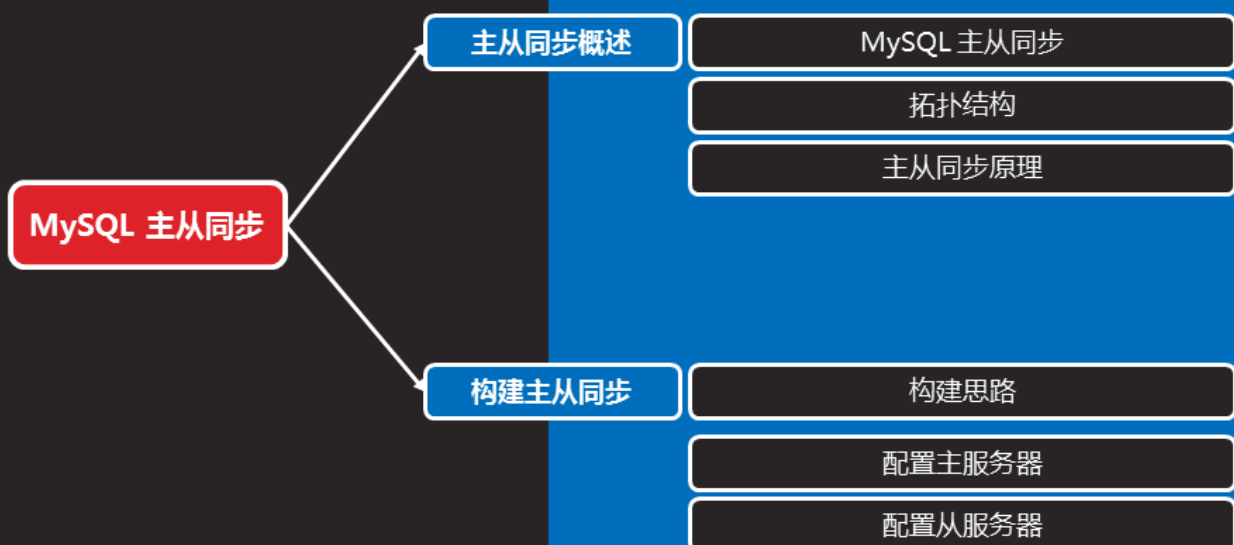
**DAY01**

# 内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	MySQL 主从同步
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	主从同步模式
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



## MySQL 主从同步



# 主从同步概述

## MySQL 主从同步

知识讲解

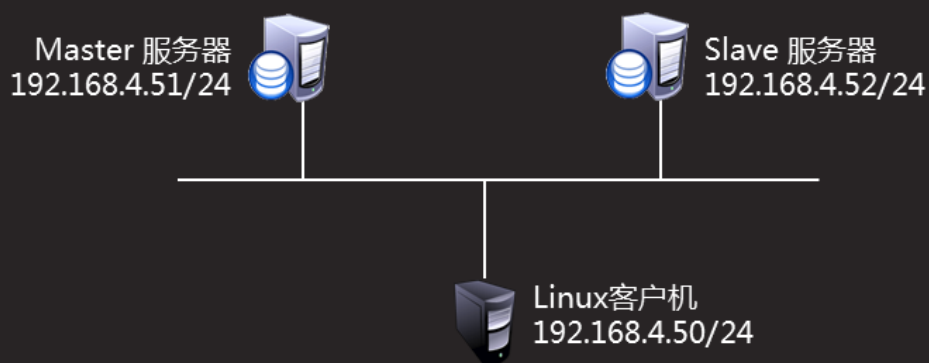
- 主从同步介绍
  - 实现数据自动同步的服务结构
  - 主服务器：接受客户端访问连接
  - 从服务器：自动同步主服务器数据



# 拓扑结构

知识讲解

- 主服务器： 192.168.4.51
- 从服务器： 192.168.4.52
- 客户端： 192.168.4.50





## 构建思路

知识讲解

- 配置主库
  - 启用binlog日志、授权用户、查看binlog日志信息
- 配置从服务器
  - 确保与主服务器数据一致。
  - 设置server\_id 、指定主库信息、启动slave程序



## 配置主服务器

知识讲解

- 修改主配置文件
  - 启用binlog日志 并重启服务

```
[root@mysql51 mysql]# vim /etc/my.cnf
[mysqld]
log_bin=日志名           //启用binlog日志
server_id=id值           //指定服务器ID号
...
[root@dbsvr1 mysql]# systemctl restart mysqld
```



## 配置主服务器（续1）

知识讲解

- 授权用户
  - 用户权限 replication slave
  - 用户名自定义
  - 客户端地址允许从库连接
  - 密码自定义

```
mysql> grant replication slave on *.* to  
        repluser@ "%" identified by "123qqq...A" ;
```



## 配置主服务器（续2）

知识讲解

- 查看日志信息

```
mysql> show master status\G;  
***** 1. row *****  
        File: mysql51-bin.000001  
        Position: 334  
        Binlog_Do_DB:  
        Binlog_Ignore_DB:  
        Executed_Gtid_Set:  
        .. ..
```



## 配置从服务器

- 修改主配置文件my.cnf
  - 指定server\_id 不允许与主库server\_id值相同

知识讲解

```
[root@mysql52 ~]# vim /etc/my.cnf
[mysqld]
server_id = 52                                //指定服务器ID
.. ..
[root@mysql52 ~]# systemctl restart mysqld
```



## 配置从服务器（续1）

- 确保与主库数据一致
  - 在主库上备份数据，备份文件拷贝给从库
  - 从库使用备份文件恢复数据
  - 从库查看备份数据对应的binlog日志信息

知识讲解

```
//主库备份数据，--master-data记录当前备份数据对应的日志信息
]# mysqldump -uroot -p密码 --master-data 数据库名 > /allbak.sql
]# scp /allbak.sql root@192.168.4.52:/root/

//从库恢复数据
]# mysql -uroot -p密码 数据库名 < /root/allbak.sql

]# vim /root/allbak.sql //查看binlog日志名及偏移量
```





## 配置从服务器（续2）

### • 指定主库信息

知识讲解

```
mysql> change master to
-> master_host= "192.168.4.51" ,           //主库ip地址
-> master_user= "repluser" ,               //主库授权用户
-> master_password= "123qqq...A" ,         //授权用户密码
-> master_log_file= "mysql51-bin.000001" , //主库日志文件
-> master_log_pos=334;                     //日志偏移量

mysql> start slave;                         //启动slave进程
```

1. Master信息会自动保存到 /var/lib/mysql/master.info 文件
2. 若更改主库信息时，  
应先执行stop slave；修改后 再执行 start slave;



## 配置从服务器（续3）

### • 查看Slave状态

- 确认IO线程、SQL线程都是 Yes 状态

知识讲解

```
mysql> show slave status\G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.4.51 //主库ip地址
Master_User: repluser
...
Slave_IO_Running : Yes      //IO线程已运行
Slave_SQL_Running : Yes     //SQL线程已运行
```



## 配置从服务器（续4）

- 相关文件
  - 存放在数据库目录下

知识讲解

文件名	说明
master.info	主库信息
relay-log.info	中继日志信息
主机名-relay-bin.xxxxxxx	中继日志
主机名-relay-bin.index	索引文件



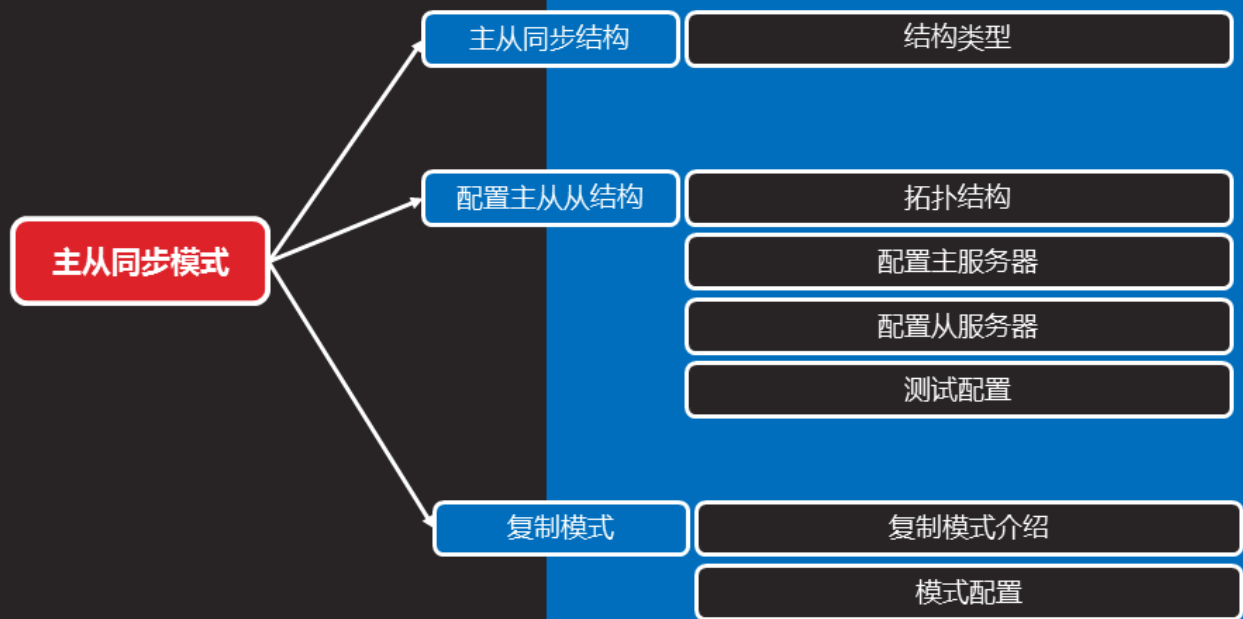
## 案例1：MySQL一主一从

- 具体要求如下：
  - 数据库服务器192.168.4.51配置为主数据库服务器
  - 数据库服务器192.168.4.52配置为主数据库服务器
  - 客户端192.168.4.50测试配置

课堂练习



## 主从同步模式



## 主从同步结构

# 结构类型

- 基本应用

- 单向复制：一主 <-- 一从

- 扩展应用

- 一主多从：从 <-- 主 --> 从

↓

从

- 链式复制：主 <-- 从 <-- 从

- 互为主从：主 <--> 主

知识讲解



## 配置主服务器

- 配置步骤

- 启用binlog日志
  - 重启服务数据库服务
  - 用户授权
- ```

]# vim /etc/my.cnf
[mysqld]
log-bin=日志名
server_id=id号
:wq

]# systemctl restart mysqld
  
```

```
mysql> grant replication slave on *.* to 用户名@" 从库IP地址"
identified by "密码" ;
```

知识讲解



## 配置从服务器

- 配置从服务器54

- 修改配置文件
  - 用户授权
  - 指定主库信息
  - 启动slave进程
- ```

]# vim /etc/my.cnf
[mysqld]
server_id=id 号
log-bin=日志名
log_slave_updates //允许级联复制
]# systemctl restart mysqld
  
```

```
mysql> grant replication slave on *.* to 用户名@ '%'
identified by '密码' ;
```

```
mysql> change master to master_host= "192.168.4.54" ,
-> master_user = "用户名" ,
-> master_password = "密码" ,
-> master_log_file = "binlog日志文件名" ,
-> master_log_pos = 偏移量;
```

```
mysql> start slave; //启动slave进程
mysql> show slave status\G; //检查状态
```

知识讲解



## 配置从服务器（续1）

知识讲解

- 配置从服务器55
  - 修改配置文件 `]# vim /etc/my.cnf`
  - 指定主库信息 `[mysqld]`  
`server_id=id 号`
  - 启动slave进程 `:wq`  
`]# systemctl restart mysqld`

```
mysql> change master to master_host= "192.168.4.54" ,
-> master_user = "用户名" ,
-> master_password = "密码" ,
-> master_log_file = "binlog日志文件名" ,
-> master_log_pos = 偏移量;
```

```
mysql> start slave; //启动slave进程
mysql> show slave status\G; //检查状态
```



## 测试配置

知识讲解

- 测试步骤
  - 在主库授权访问数据的连接用户
  - 客户端连接主库，执行与权限匹配的sql操作
  - 授权用户连接第1台从库，可以看到主库的数据
  - 授权用户连接第2台从库，可以看到主库的数据

```
mysql> grant all on 库.* to 用户@'客户端地址'
identified by '密码';
```

```
]# mysql -h数据库IP地址 -u用户名 -p 密码
```

```
mysql> select * from 库.表;
```



## 案例2：配置主从从结构

具体要求如下：

- 配置主机192.168.4.53为主库服务器
- 配置主机192.168.4.54为53主机的从库服务器
- 配置主机192.168.4.55为54主机的从库服务器
- 客户端测试配置。

课堂练习



## 复制模式

## 复制模式介绍

知识讲解

- 异步复制 ( Asynchronous replication )
  - 主库执行完一次事务后，立即将结果返回给客户端，并不关心从库是否已经接收并处理。
- 全同步复制 ( Fully synchronous replication )
  - 当主库执行完一次事务，且所有从库都执行了该事务后才将结果返回给客户端。
- 半同步复制 ( Semisynchronous replication )
  - 介于异步复制和全同步复制之间。
  - 主库在执行完一次事务后，等待至少一个从库接收到并写到relay log中才将结果返回给客户端。



## 模式配置

知识讲解

- 查看是否允许动态加载模块
  - 默认允许

```
mysql> show variables like 'have_dynamic_loading';
```

```
mysql> show variables like "have_dynamic_loading";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_dynamic_loading | YES |
+-----+-----+
1 row in set (0.00 sec)
```

```
1 row in set (0.00 sec)
```





## 模式配置（续1）

- 命令行加载插件
  - 使用数据库管理员root用户访问服务

```
mysql> install plugin rpl_semi_sync_master
-> SONAME "semisync_master.so" ; //主服务器上执行
```

```
mysql> install plugin rpl_semi_sync_slave
-> SONAME "semisync_slave.so" ; //从服务器上执行
```

```
mysql> select plugin_name, plugin_status from
information_schema.plugins where plugin_name like '%semi%'; //查看
```

PLUGIN_NAME	PLUGIN_STATUS
rpl_semi_sync_master	ACTIVE
rpl_semi_sync_slave	ACTIVE

知识讲解



## 模式配置（续2）

- 启用半同步复制
  - 半同步复制，默认是关闭的。

```
mysql> set global rpl_semi_sync_master_enabled=1; //主服务器执行
```

```
mysql> set global rpl_semi_sync_slave_enabled=1; //从服务器执行
```

```
mysql> show variables like "rpl_semi_sync_%_enabled" ; //查看
```

```
mysql> show variables like "rpl_semi_sync_%_enabled";
```

Variable_name	Value
rpl_semi_sync_master_enabled	ON
rpl_semi_sync_slave_enabled	ON

知识讲解



## 模式配置（续3）

知识讲解

- 永久启用半同步复制
  - 修改主配置文件 /etc/my.cnf
  - 在[mysqld] 下方添加

//主服务器

```
plugin-load=rpl_semi_sync_master=semisync_master.so  
rpl_semi_sync_master_enabled=1
```

//从服务器

```
plugin-load=rpl_semi_sync_slave=semisync_slave.so  
rpl_semi_sync_slave_enabled=1
```



## 模式配置（续4）

知识讲解

- 是master和slave两种角色时
  - 需同时启用

```
plugin-load =
```

```
"rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=semisync_slave.so"
```

```
rpl_semi_sync_master_enabled=1
```

```
rpl_semi_sync_slave_enabled=1
```

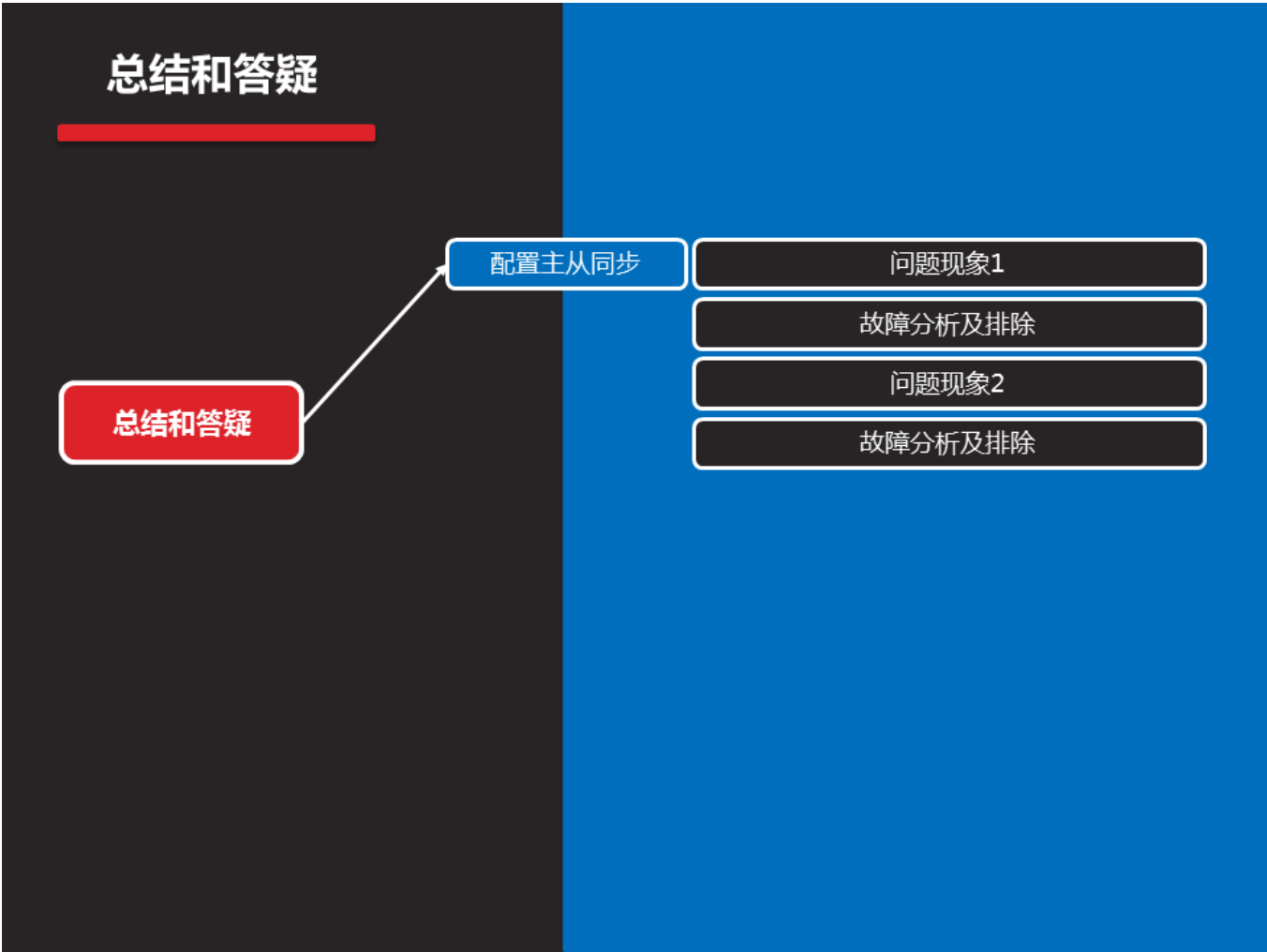
```
mysql> show variables like "rpl_semi_sync_%_enabled";
```

Variable_name	Value
rpl_semi_sync_master_enabled	ON
rpl_semi_sync_slave_enabled	ON

```
2 rows in set (0.00 sec)
```

```
2 rows in set (0.00 sec)
```





# 配置主从同步

## 问题现象1

- Slave\_IO 线程没有运行
  - 报错：Slave\_IO\_Running: No

```
mysql > show slave status \G;
```

```
Slave_IO_Running: No
```

```
Last_IO_Error : 报错信息.....
```

# 故障分析及排除

## 知识讲解

- 原因分析
  - 连接不上 master数据库服务器
- 解决办法
  - 检查物理连接（ping）、检查授权用户
  - 禁用防火墙、关闭SELinux
  - 或是binlog日志文件指定错误（日志名或pos节点）

```
mysql> stop slave;  
mysql> change master to 选项=值;  
mysql> start slave;
```



# 故障分析及排除

## 知识讲解

- 原因分析
  - 执行本机中继日志里的sql命令时，sql命令使用的库、表或记录在本机不存在

- 解决办法

```
mysql> stop slave;  
mysql> ...  
mysql> start slave;
```

//创建或恢复需要用到的库或表

