[mysql数据库主从配置](https://www.cnblogs.com/atcloud/p/10773855.html)

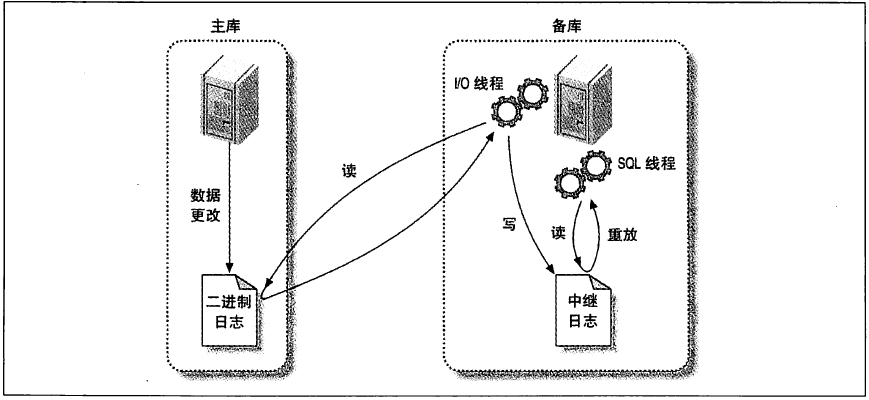
在一篇文章[《离线安装mysql数据库》](https://www.toutiao.com/i6680112559154004484/),讲解了离线安装mysql数据库的过程，本文将讲解mysql数据库的主从配置方法。mysql数据库进行主从配置后，可以实现数据库的备份、同时应用也可以实现读写分离，提高应用的并发量。

**1、主从原理**

从《高性能mysql》书中讲解主从原理大致有三个步骤：

* 在主库上把数据更改记录到二进制日志中（Binary Log）中，这些记录称为二进制日志事件。
* 从库通过IO线程将主库上的日志复制到自己的中继日志（Relay Log）中。
* 从库通过SQL线程读取中继日志中的事件，将其重放到自己数据上。

原理图为：



**2、主从配置**

演示的环境如下：

| **名称** | **IP** |
| --- | --- |
| msyql-master（主库） | 192.168.197.135 |
| mysql-slave（从库） | 192.168.197.136 |

首先在这两台服务器安装mysql数据库，安装方法参考[《离线安装mysql数据库》](https://www.toutiao.com/i6680112559154004484/" \t "_blank)。

**（1）、配置主库**

* 修改my.cnf文件，在[mysqld]加入下面的内容：
* # 服务的唯一编号
* server-id = 1
* # 开启mysql binlog功能
* log-bin = mysql-bin
* # binlog记录内容的方式，记录被操作的每一行
* binlog\_format = ROW
* # 减少记录日志的内容，只记录受影响的列
* binlog\_row\_image = minimal
* # 指定需要复制的数据库名为jgyw
* binlog-do-db = jgyw
* 修改好配置文件，重启mysql服务
* service mysqld restart
* 创建从库同步数据的账号
* grant replication slave on \*.\* to 'jgyw'@'192.168.197.136' identified by 'jgyw@123';
* flush privileges;

**注意：**上面这两个命令是在mysql的终端执行的。

* 查看主库的状态：

mysql的终端执行:

show master status\G;

返回的信息为：

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***File: mysql-bin.000002***

***Position: 2380***

***Binlog\_Do\_DB: jgyw***

***Binlog\_Ignore\_DB:***

***Executed\_Gtid\_Set:***

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

**（2）、配置从库**

* 修改my.cnf文件，在[mysqld]加入下面的内容：
* # 服务的唯一编号
* server-id = 2
* # 开启mysql binlog功能
* log-bin = mysql-bin
* # binlog记录内容的方式，记录被操作的每一行
* binlog\_format = ROW
* # 减少记录日志的内容，只记录受影响的列
* binlog\_row\_image = minimal
* # 指定需要复制的数据库名为jgyw
* replicate-do-db = jgyw
* 修改好配置文件，重启mysql服务
* service mysqld restart
* 执行同步命令

mysql的终端执行:

# 设置主服务器ip，同步账号密码，同步位置

change master to master\_host='192.168.197.135',master\_user='jgyw',master\_password='jgyw@123',master\_log\_file='mysql-bin.000002',master\_log\_pos=2380;

# 开启同步功能

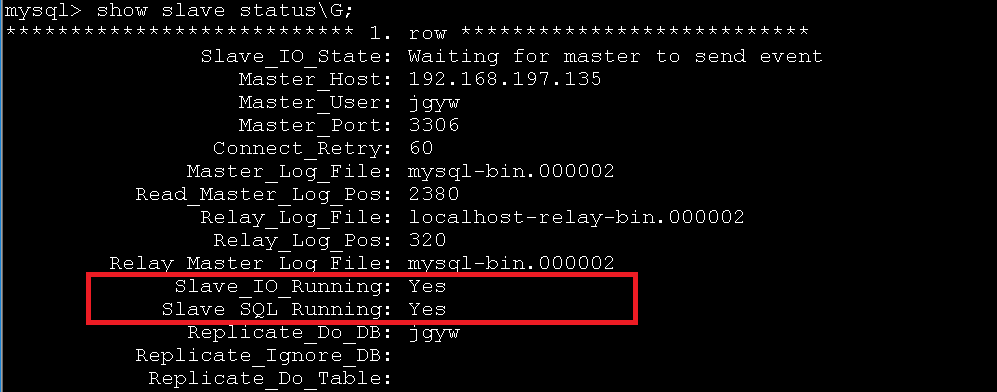
start slave;

* 查看从库的状态

mysql的终端执行:

show slave status\G;

返回信息为：



**注意：**Slave\_IO\_Running和Slave\_SQL\_Running的状态都为Yes时，说明从库配置成功。

**3、测试**

（1）、在**主库**上创建jgyw模式，即：

create schema jgyw;

（2）、在**主库**上的jgyw模式里面创建comm\_config表，即：

use jgyw;

CREATE TABLE comm\_config (configId varchar(200) NOT NULL ,configValue varchar(1024) DEFAULT NULL ,description varchar(2000) DEFAULT NULL ,PRIMARY KEY (configId)) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;

（3）、在**主库**上往comm\_config表中插入一条记录，即：

insert into comm\_config(configId, configValue, description) values('name', '架构与我', '测试一下');

（4）、在**从库**上查看模式，即：

show schemas;

结果为：

+--------------------+

| Database |

+--------------------+

| information\_schema |

| jgyw |

| mysql |

| performance\_schema |

| sys |

+--------------------+

5 rows in set (0.00 sec)

说明jgyw模式同步到从库了

（5）、在**从库**上查看jgyw模式下的表及数据，即：

use jgyw;

show tables;

结果为：

+----------------+

| Tables\_in\_jgyw |

+----------------+

| comm\_config |

+----------------+

1 row in set (0.00 sec)

说明表也同步好了，再查看数据，即：

select \* from comm\_config;

结果为：

+----------+--------------+--------------+

| configId | configValue | description |

+----------+--------------+--------------+

| name | 架构与我 | 测试一下 |

+----------+--------------+--------------+

1 row in set (0.00 sec)

说明数据也同步过来了。