# 复制

冗余是高可用的核心，MySQL也不例外，通过复制实现MySQL的高可用，本章学习MySQL复制相关的知识。

复制使来自一个MySQL数据库服务器（master-主）的数据可以复制到一个或多个MySQL数据库服务器（从站--slaves）。默认情况下，复制是异步的；从站不需要永久连接以接收主站的更新。**根据配置，你可以选择复制MySQL实例中的所有数据库，选定的数据库或选定的表。**

MySQL复制的优点包括：

* 支持横向扩展：在多个从站之间平衡负载以提高性能。在这种环境下，所有写入和更新都必须在主服务器上进行。但是，读取会发生在一个或多个从站上。该模型可以提高写入的性能（由于主服务器专用于更新），同时读取数据的速度可以随着增加从站的数量而增加。
* 数据安全性：由于数据被复制到从站，从站可以暂停复制过程，可以在从站上运行备份服务，而不会破坏或影响对应的主站数据。
* 分析：可以在主机上进行OLTP，而信息分析可以在从站上进行，而不会影主站的性能。
* 远程数据分发：你可以使用复制技术创建远程站点数据的本地副本以供使用，而无需永久访问主站。

有关如何在这些情况下使用复制的信息，请参见 [第16.3节“复制解决方案”](https://dev.mysql.com/doc/refman/5.7/en/replication-solutions.html)。

MySQL 5.7支持不同的复制方法。传统方法是基于从主机二进制日志中复制事件，并要求主站和从站之间的日志文件和位置保持同步。基于全局事务标识符（GTID）的新方法是事务性的，因此不需要使用这些文件中的日志文件或位置，这大大简化了许多常见的复制任务。只要在主服务器上提交的所有事务也已被应用在从服务器上，使用GTID的复制保证了主从服务器之间的一致性。

MySQL中，复制支持不同的同步类型。最原始的是单向异步复制，其中一个服务器充当主服务器，而一个或多个其他服务器充当从服务器。在MySQL 5.7中，除了内置异步复制之外，还支持半同步复制。使用半同步复制，直到至少一个从机确认（返回ACK给主机）已收到并记录事务事件之后，主机才会真正提交事务， [请参见第16.3.9节“半同步复制”](https://dev.mysql.com/doc/refman/5.7/en/replication-semisync.html)。MySQL 5.7还支持延迟复制，使得从属服务器故意滞后于主机至少指定的时间量; 请参见 [第16.3.10节“延迟复制”](https://dev.mysql.com/doc/refman/5.7/en/replication-delayed.html)。对于需要*同步*复制的场景 ，请使用NDB Cluster（请参见[第21章，*MySQL NDB Cluster 7.5和NDB Cluster 7.6*](https://dev.mysql.com/doc/refman/5.7/en/mysql-cluster.html)）。

复制格式有两种核心类型，即基于语句的复制（SBR）和基于行的复制（RBR）。你也可以使用第三种，混合型复制（MBR）。有关不同复制格式的更多信息，请参见 [第16.2.1节“复制格式”](https://dev.mysql.com/doc/refman/5.7/en/replication-formats.html)。

你可以使用复制来解决许多不同的问题，包括性能，数据库的备份，以及作为减轻系统故障的更大解决方案的一部分。有关如何解决这些问题的信息，请参见 [第16.3节“复制解决方案”](https://dev.mysql.com/doc/refman/5.7/en/replication-solutions.html)。

## 1、配置复制

本节介绍MySQL中支持的多种类型的复制相关的配置。主要内容有：

* 两个或多个服务器使用二进制日志文件位置进行复制的指南，请参见 [第16.1.2节“设置基于二进制日志文件位置的复制”](https://dev.mysql.com/doc/refman/5.7/en/replication-howto.html)，介绍服务器的配置，并提供在主机和从机之间复制数据的方法。
* 两个或多个服务器基于GTID进行复制的指南，请参见[第16.1.3节“使用全局事务标识符复制”](https://dev.mysql.com/doc/refman/5.7/en/replication-gtids.html) 处理服务器的配置。
* 二进制日志中记录的多种格式的事件。被称为基于语句的复制（SBR）、基于行的复制（RBR）。第三种为两种格式混合的复制（MIXED），在适当的情况下MySQL自动使用SBR或RBR复制来利用SBR和RBR格式的优点。[第16.2.1节“复制格式”](https://dev.mysql.com/doc/refman/5.7/en/replication-formats.html)中讨论了不同的格式 。
* 有关复制的不同配置选项和变量的详细信息，请参见 [第16.1.6节“复制和二进制日志相关的配置选项和变量”](https://dev.mysql.com/doc/refman/5.7/en/replication-options.html)。
* 一旦启动，复制过程就需要很少的管理或监控。有关可能要执行的常见任务的建议，请参见 [第16.1.7节“常见复制管理任务”](https://dev.mysql.com/doc/refman/5.7/en/replication-administration.html)。

### 1.1 基于二进制日志文件位置的复制概述

本节介绍MySQL服务器之间基于二进制日志文件位置的复制，其中主站（数据库更改的源头）将更新和修改作为“ 事件 ”写入二进制日志。根据正在记录的数据库更改，二进制日志中的信息以不同的日志格式存储。从站从主站读取二进制日志，并执行在从站本地数据库的二进制日志中的事件。

每个从站都会收到二进制日志的全部副本。从站的任务是决定应该执行二进制日志中的哪些语句。除非另有说明，否则主站二进制日志中的所有事件都将在从站上被执行。如果需要，你也可以将从站配置为仅处理特定数据库或表的事件。

**重要说明：**

虽然可以配置从站去执行某些数据库或者表的二进制日志，但是不能配置主站只记录某些事件，换句话说主站会记录所有对数据库更改的事件。

每个从站都保存有表示二进制日志坐标的记录，这个坐标的记录的内容为：已经从主站读取并处理了的二进制日志文件的文件名和文件内的位置。这意味着可以将多个从站连接到同一主站并执行同一二进制日志的不同部分。由于从站控制此过程，因此可以将单个从站与服务器连接并断开连接，而不会影响主站的操作。另外，由于每个从站记录二进制日志中的当前位置，所以可以将从站断开，重新连接，然后恢复处理。

主站和每个从站必须配置唯一的ID（使用[server-id](https://dev.mysql.com/doc/refman/5.7/en/replication-options.html#option_mysqld_server-id)选项）。此外，每个从站必须配置相关主站的主机名，日志文件名和该文件中的位置的信息。这些细节可以在从站的MySQL会话中使用[CHANGE MASTER TO](https://dev.mysql.com/doc/refman/5.7/en/change-master-to.html)语句进行控制。详细信息存储在从站的信息存储库中，可以是文件或表（请参见 [第16.2.4节“复制中继和状态日志”](https://dev.mysql.com/doc/refman/5.7/en/slave-logs.html)）。

### 1.2 设置基于二进制日志文件位置的复制

虽然设置复制有许多不同的方法，但是所有的设置都会有一些通用任务。

* 在主站上，必须启用二进制日志记录并配置唯一的服务器ID。这可能需要重新启动服务器。
* 在要连接到主站的每个从站上，必须配置唯一的服务器ID。这可能需要重新启动服务器。
* 最好为复制创建独立的用户（这是可选的，基于安全考虑，用户权限应该最小化）。
* 在创建数据快照或启动复制过程之前，请在主站上记录二进制日志中的当前位置。在配置从站时，你需要此信息，以便从站知道从二进制日志中的哪个地方开始执行事件。
* 如果你已经拥有主站上的数据，并希望使用它来同步从站，则需要创建数据快照将数据复制到从站。你使用的存储引擎对创建快照的方式有影响。使用[MyISAM](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html)时，必须停止主机的处理语句以获取读锁定，在允许主站继续执行语句之前获取主站当前的二进制日志坐标并转储其数据。如果你不停止执行语句，则数据转储和主站态信息将不匹配，从而导致从站数据库不一致或损坏。有关复制[MyISAM](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html)主站的更多信息 ， [请参见第16.1.2.3节“获取复制主站二进制日志坐标”](https://dev.mysql.com/doc/refman/5.7/en/replication-howto-masterstatus.html)。如果正在使用[InnoDB](https://dev.mysql.com/doc/refman/5.7/en/innodb-storage-engine.html)，则不需要读锁定，并且足够长的传输数据快照的事务就足够了。有关更多信息，请参见 [第14.19节“InnoDB和MySQL复制”](https://dev.mysql.com/doc/refman/5.7/en/innodb-and-mysql-replication.html)。
* 使用用于连接的主站的设置来配置从站，例如主站的主机名，登录凭据和二进制日志文件名称和位置。请参见 [第16.1.2.5.2节“在从站上设置主站配置”](https://dev.mysql.com/doc/refman/5.7/en/replication-setup-slaves.html#replication-howto-slaveinit)。

**注意**

设置过程中的某些步骤需要 [SUPER](https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_super)特权。如果没有此权限，则可能无法启用复制。

配置完基本的选项后，选择你需要的配置方案：

* 要设置全新安装的不包含数据的主站和从站的复制，请参见 [第16.1.2.5.3节“在新的主站和从站之间设置复制”](https://dev.mysql.com/doc/refman/5.7/en/replication-setup-slaves.html#replication-howto-newservers)。
* 要使用现有MySQL服务器的数据设置新主站的复制，请参见 [第16.1.2.5.4节“使用现有数据设置复制”](https://dev.mysql.com/doc/refman/5.7/en/replication-setup-slaves.html#replication-howto-existingdata)。
* 要将复制从站添加到现有的复制环境，请参见 [第16.1.2.6节“将从站添加到复制环境”](https://dev.mysql.com/doc/refman/5.7/en/replication-howto-additionalslaves.html)。

**1.2.1 设置主站：**

要将主服务器配置为使用基于二进制日志文件位置的复制，必须启用二进制日志记录并建立唯一的服务器ID。如果尚未完成，则需要重新启动服务器。二进制日志记录*必须*在主服务器上启用，因为二进制日志是从主服务器复制到其从服务器的基础。如果二进制日志记录未在主服务器上使用log-bin 选项启用，则复制是不可能的。复制组中的每个服务器都必须配置唯一的服务器标识。此ID用于标识组中的单个服务器，并且必须是1到（2 **32**）-1 之间的正整数。如何组织和选择数字是你的权利。

要配置二进制日志和服务器ID选项，请关闭MySQL服务器并编辑my.cnf文件，在配置文件的[mysqld]部分添加log-bin和 server-id选项，例如，要使用日志文件名前缀 mysql-bin启用二进制日志记录，并配置服务器ID为1，请如下配置：

[mysqld]

log-bin=mysql-bin

server-id=1

修改后重启服务器。

注意：

* 为了在使用InnoDB事务的复制设置中**尽最大可能**的做到持久性和一致性 ，你应该在主站的配置文件中使用innodb\_flush\_log\_at\_trx\_commit=1和sync\_binlog=1。
* 确保主站上未启用选项skip-networking。否则，如果网络被禁用，则从站不能与主站通信，复制将失败。
* server-id的范围必须为1到（2的32次方）-1 之间的正整数，如果你省略了server-id或者将其设置为0，则主站会拒绝从站的连接。

**1.2.2 创建用于复制的用户**

复制的时候，每个从站使用MySQL用户名和密码连接到主站，因此，主站上必须具有合适的用户名和密码来让从站连接。任何帐户都可以用于此操作，只要它已被授予[REPLICATION SLAVE](https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_replication-slave)权限，可以为每个从站创建不同的帐户，或者每个从站使用相同的帐户连接到主站。

虽然不必专门为复制创建帐户，但应注意复制用户名和密码以纯文本格式存储在文件或表中。因此，你可能需要创建一个单独的帐户，该帐户只具有复制权限，以尽可能减少对其他帐户的影响。

要创建一个新帐户，请使用[CREATE USER](https://dev.mysql.com/doc/refman/5.7/en/create-user.html)语句。要将此帐户授予复制所需的权限，请使用[GRANT](https://dev.mysql.com/doc/refman/5.7/en/grant.html) 语句。如果仅为了复制而创建一个帐户，该帐户只需要 [REPLICATION SLAVE](https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_replication-slave)权限。例如，要创建一个新用户repl，并可以在mydomain.com域内的任何主机上连接到主站进行复制 ，请在主服务器上执行以下语句：

mysql> CREATE USER 'repl'@'%.mydomain.com' IDENTIFIED BY 'slavepass';

mysql> GRANT REPLICATION SLAVE ON \*.\* TO 'repl'@'%.mydomain.com';

**1.2.3 获取主站二进制日志的坐标**

### 1.3 使用全局事务标识符进行复制

## 2、复制实现

## 3、复制解决方案

复制可以在许多不同的环境中用于一系列的目的。本节为使用复制达到各种目的的各种解决方案提供了注意事项和建议。包括：在备份环境中使用复制、在主机和从机上使用不同的存储引擎、使用复制作为横向扩展的基础、出于一些原因，将不同的数据库复制到不同的从站、随着复制从站数量的增加，主站上的负载可能增加并导致性能下降（因为需要将二进制日志复制到每个从站）。有关提高复制性能的提示、主站和从站的切换、加密复制过程以提高安全性。

3.1 使用复制进行备份

为了主站的高可用性，我们可以通过复制，将主站数据同步到从站，然后备份从站上的数据，这样的话主站将不会受到任何影响，主站可以正常服务，而对从站进行一系列操作。那么以何种方式备份从站的数据呢？这取决于数据库的大小以及要备份的内容（只备份数据还是连同从站的状态），依据不同的情况，有以下两种方式：

* 当要备份的从站数据库的大小不会太大，则[**mysqldump**](https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html)工具可能是合适的。
* 对于从站较大的数据库，[**mysqldump**](https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html)将不太合适，会比较低效，这时可以备份原始数据文件。使用原始数据文件也意味着你可以备份二进制和中继日志，这将使你能够在从站发生故障的情况下重新创建从站（因为备份所有数据，意味着将从站的状态也备份起来了）。
* 另外一种备份策略是将主站或者从站至于只读状态，备份之后，再回复状态，这会影响主站的可用性。

接下来分别介绍这三种情况。

3.1.1 使用mysqldump来备份从站

这个工具适用于数据不是很多的情况。使用mysqldump时，应该在启动转储过程之前停止从站上的复制，以确保转储包含一组一致的数据（其实就是备份过程中不可以修改正在备份的数据库实例），具体如下：

1、停止从站处理请求。可以使用[mysqladmin](https://dev.mysql.com/doc/refman/5.7/en/mysqladmin.html" \o "4.5.2 mysqladmin  - 用于管理MySQL服务器的客户端)在slave上完全停止复制：

shell> mysqladmin stop-slave

或者也可以只是停止从站的SQL线程来执行主站同步过来的二进制日志中的事件：

shell> mysql -e 'STOP SLAVE SQL\_THREAD;'

这使从站能够继续接受主站的二进制日志，并使用I / O线程将其存储在中继日志中，但可以防止从站执行这些事件并更改其数据。在忙碌的复制环境中，允许I / O线程在备份过程中运行，可能会在重新启动从站SQL线程时加快追赶过程。

2、运行mysqldump转储数据库。可以转储所有数据库或选择要转储的数据库。例如，要转储所有数据库：

shell> mysqldump --all-databases > fulldb.dump

3、转储完成后，再次启动从站操作：

shell> mysqladmin start-slave

通常情况下，需要再如上命令中添加用户名、密码之类的信息，并将其置于shell脚本中自动执行。使用此方法备份时要监听从站的复制过程，确保运行备份所需的时间不会影响从站执行主站事件的能力，如果从站收到影响，可以多个从站，来加入备份中。这通常需要将不同的数据库复制到不同的从站。

3.1.2 从从站备份原始数据

从从站备份所有数据应该在从站关闭时进行，因为这不仅备份了数据，还备份了从站的各种状态。同时这又不会影响主站的可用性。步骤如下：

1、关闭从站MySQL服务器：

shell> mysqladmin shutdown

2、复制数据文件。你可以使用任何合适的复制或归档实用程序，包括**cp**，**tar**或**WinZip等**。例如，假设数据目录位于当前目录下，则可以按如下方式归档整个目录：（因为从站关闭了，因此不存在服务器还没有将数据刷新到磁盘的情况，因此可以直接拷贝数据目录）

shell> tar cf /tmp/dbbackup.tar ./data

3、再次启动MySQL服务器。在Unix/Linux下：

shell> mysqld\_safe &

通常你应该备份站的整个数据目录。如果你想要恢复数据并作为从站运行（例如，在从站发生故障的情况下），则除了从站的数据之外，还应备份从站状态文件，从站上配置的主站信息、中继日志信息和中继日志文件。还原从站数据后，需要这些文件才能恢复复制（复制需要这些信息）。

如果丢失了中继日志但仍具有 relay-log.info文件，则可以检查它以确定SQL线程将主站的二进制日志执行的距离（即：还有多少没有执行）。然后你可以使用带有MASTER\_LOG\_FILE和 MASTER\_LOG\_POS选项的[CHANGE MASTER TO](https://dev.mysql.com/doc/refman/5.7/en/change-master-to.html)命令告诉从站从该点重新读取二进制日志。这要求二进制日志仍然存在于主服务器上。

如果你的从站正在执行 [LOAD DATA INFILE](https://dev.mysql.com/doc/refman/5.7/en/load-data.html)语句，你还应该备份SQL\_LOAD-\*用到的数据文件。从站需要这些文件来恢复被中断的[LOAD DATA INFILE](https://dev.mysql.com/doc/refman/5.7/en/load-data.html)操作 。该目录的位置是该[--slave-load-tmpdir](https://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#option_mysqld_slave-load-tmpdir)选项的值 。如果服务器未配置该选项，则目录位置是[tmpdir](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_tmpdir)系统变量的值 。

3.1.3 将主站和从站设置为只读模式来备份

**注意：**

在进行备份时，因为主站或者从站还在运行，因此可能存在服务器没有将更新刷新到磁盘的情况，因此，不要直接复制文件，而应该使用专用的工具。

假设有如下设置：

* 主站M1
* M1的从站S1
* 连接到M1的客户端C1
* 连接到S1的客户端C2

不管备份主站还是从站，都需要锁定全局读锁并将[read\_only](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_read_only)设置为on。这不会传播到任何从站。

**情况一：备份主站，则如下操作：**

1、机M1置于只读状态：

mysql> FLUSH TABLES WITH READ LOCK;

mysql> SET GLOBAL read\_only = ON;

当M1处于只读状态时，以下属性为真：

* 由于服务器处于只读模式，由C1发送到M1的更新请求将被阻止。
* 由C1向M1发送的查询结果请求将成功。
* 在M1上备份是安全的。
* 在S1上备份是不安全的。该服务器仍在运行，并且可能正在处理二进制日志或来自客户端C2的更新。

2、M1只读时，执行备份。例如，你可以使用[**mysqldump**](https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html)。

3、在M1的备份操作完成后，通过执行以下语句将M1恢复到正常工作状态：

mysql> SET GLOBAL read\_only = OFF;

mysql> UNLOCK TABLES;

**情况二：备份从站，步骤与备份主站一样。从站处于只读时具有如下属性：**

* **主机M1将继续运行，因此在主站上进行备份是不安全的。**
* **从站S1停止，因此在从站S1上进行备份是安全的。**

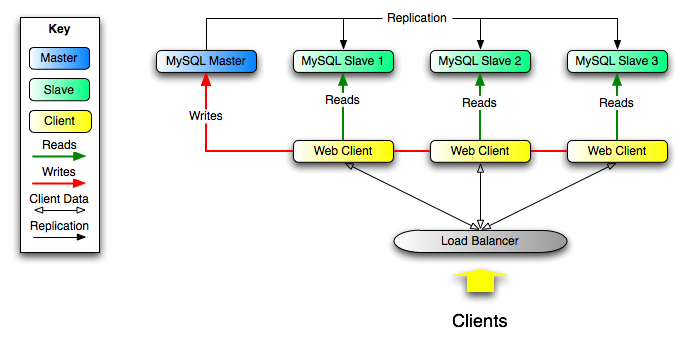
**这些属性为流行的备份方案提供了基础：让一个从站忙于执行备份一段时间不是问题，因为它不会影响整个网络，并且系统在备份期间仍在对外服务。特别是，客户端仍然可以在主站上进行更新，主站不会受到从站上备份活动的影响。在从站恢复正常操作后，它将通过追踪主站的二进制日志中的任何未完成的更新来再次与主站同步。确实、复制可以实现高可用。另外，这个也可以像第一种一样，来作为脚本定期备份。**

3.2 处理从站的意外暂停（比如崩溃）

3.3 主机和从机上使用不同的存储引擎

3.4 使用复制实现横向扩展

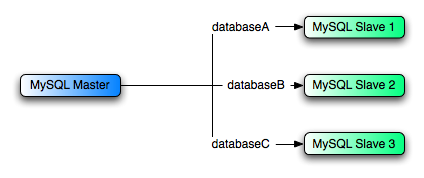
**复制可以作为横向扩展的基础，即：在多个MySQL服务器上平衡客户端的请求。因为复制是从一个主站分发到一个或多个从站，所以使用复制进行横向扩展非常适合具有大量读和少量写入/更新的场景。大多数网站都是这种场景，用户浏览网站，阅读文章，发布信息或查看产品。****下图是使用复制进行横向扩展以提高性能。**

****

如果负责数据库访问的代码已被抽象/模块化，则将其转换为使用基于复制的横向扩展应该非常顺利和容易。更改数据库访问的实现以将所有写入/更新发送到主站，并将读取发送到主站或从站。可见抽象还是非常有必要的。每一层之间必须进行模块化和抽象，以保持良好的接口，这样有利于后期升级等。

3.5 将不同的数据库复制到不同的从站

有些情况下，你有一个主库，并且希望将主站中不同的数据库复制到不同的从站。如下图：



可以通过配置主站和从站来实现这种需求，然后通过在每个从机上使用配置选项[--replicate-wild-do-table](https://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#option_mysqld_replicate-wild-do-table) 限制每个从机需要处理的二进制日志中的事件。

**重要提示：**

对于基于语句或者混合格式的二进制日志，不要使用 [--replicate-do-db](https://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#option_mysqld_replicate-do-db)，因为基于语句或者混合的方式下，某些语句的执行与选定的当前数据库有关，对于基于行的复制，[--replicate-do-db](https://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#option_mysqld_replicate-do-db)选项可以被安全的使用。

例如，为了支持如上图的分离，在执行 [START SLAVE](https://dev.mysql.com/doc/refman/5.7/en/start-slave.html)语句之前，要如下配置从站：

* 从站1应该使用 --replicate-wild-do-table=databaseA.%。
* 从站2应该使用 --replicate-wild-do-table=databaseB.%。
* 从站3应该使用 --replicate-wild-do-table=databaseC.%。

此时每个从站会接收主站的整个二进制日志，但仅执行二进制日志中被[--replicate-wild-do-table](https://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html" \l "option_mysqld_replicate-wild-do-table)选项包含的数据库和表的那些事件。

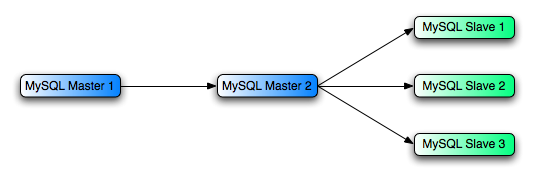
如果复制开始之前有必须被同步到从站的数据，你有多种选择：

* 将所有数据同步到每个从站，并删除不想保留的数据库，表或两者。
* 使用[**mysqldump**](https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html)为每个数据库创建一个单独的转储文件，并在每个从站上加载相应的转储文件。
* 使用原始数据文件转储，并仅包含每个从站所需的特定文件和数据库。

**注意：对于InnoDB来说，除非启用了**[**innodb\_file\_per\_table**](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_file_per_table)**，否则这无法工作，因为在**[**innodb\_file\_per\_table**](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_file_per_table)**未启用的情况下，多个数据库的多个表被放在同一个系统表空间文件中。**

3.6 提高复制的性能

随着连接到主站的从站数量的增加，主站的负载也会增加。此外，由于每个从站必须接收到主站二进制日志的完整副本，因此主站上的网络负载也可能会称为新的瓶颈。如果正在通过复制来实现横向扩展，同时多台从站连接到同一主站，这可能会式主站的负载过高。因此应该重新设计复制结构，来提高复制性能，正如计算机依据经典的话：“什么问题都可以通过增加一层来解决”，复制也可以，即：创建更深层次的复制结构，使主站只能复制到一个从站。如下图：



为此，必须按如下方式配置MySQL实例：

4、复制注释和提示