

Mysql（数据库面试题）

1 主流数据库服务软件有哪些？开源且跨平台的数据库软件有哪些？

参考答案

主流数据库服务软

件有： 甲骨文公司

Oracle

IBM DB2

微软 SQL Server

美国 Sybase 公司 Sybase

加州大学伯克利分校计算机系开发的

PostgreSQL 开源且跨平台的数据库软件

有：

MySQL、PostgreSQL：开源且跨平台

Oracle、DB2：跨平台不

开源SQL Server：不跨

平台不开源

Sybase：跨平台不开源

2 MySQL 数据库的服务进程叫什么名字？监听端口是多少？默认数据库目录是？

参考答案

服务进程名是 `mysqld`；监听端口是 `3306`；默认数据库目录为 `/var/lib/mysql`。

3 MySQL 默认的3 个库叫什么名字？哪个库里的数据不占用物理磁盘空间？

参考答案

3 个默认库：`mysql`、`test` 和 `information_schema`。

其中，`information_schema` 库的数据不占用磁盘空间，仅保存在内存里。

4 请列出 MySQL 常用的数据类型，并写出定义这些数据类型所使用的关键字。

参考答案

MySQL 常用的数据类型：

数值类型：所用关键字为 `int`、

`float` 字符类型：所用关键字为

`char`、`varchar`

日期时间类型：所用关键字为 `year`、`time`、

`datetime` 枚举类型：所用关键字为 `set`、`enum`

5 简述索引的优点与缺点，默认情况下哪个文件保存表的索引信息？

参考答案

索引的优点与缺点

如下：索引就像一

本书的目录 加快查

询记录的速度

会降低插入、更新记录的速度

默认情况下“表名.MYI” 文件保存表的索引信息

6 简述在表中创建外键字段要满足那些条件？

参考答案

在表中创建外键字段要满足以下

条件：表必须都使用 `innodb` 存

储引擎

表中外键字段的类型

要匹配被参照字段要

有明确的索引

7 简述 MySQL 体系结构的组成，并描述每个组成部分的作用。

参考答案

主要包括 8 个部分：

连接池：进程数限制、内存检查、缓存检查等。

SQL 接口：用户通过 sql 客户端发过来的命令，由 sql 接口接收，sql 操作(DML 数据操作语言：查询、修改、升级数据等；DDL 数据定义语言：创建一个新的数据库、新的索引、删除一个用户等；存储过程、视图触发器。

分析器：分析查询语句 事务处理 对象访问

权限。优化器：优化访问路径、生成执行树。

缓存和缓冲：保存 sql 查询结果。

存储引擎：用于管理存储的文件系统，将逻辑结构转换为物理结构的程序；不同的存储引擎有不同的功能和存储方式。

管理工具：备份，恢复，安全，移植，集群等，这些工具一般和文件系统打交道，不需要和 mysql-server 打交道，它们对应的都是命令。

物理存储设备(文件系统)。

8 简述 MySQL 数据库访问的执行过程。

参考答案

- 1) 客户端发出请求。
- 2) 服务器端开辟线程响应客户端请求。
- 3) 客户端发起 sql 语句查询数据库。
- 4) 查询缓存：记录用户的 sql 查询语句，如果查询内容相同，直接从查询缓存回复。
- 5) 如果缓存没有进入分析器。
- 6) 分析器：分析用户命令语法是否正确，将用户的命令进行切片，一个词一个词用空格隔开，获得用户要查询的表、内容、用户的权限等。
- 7) 优化器：执行路径的选择，生成执行树。(每个 SQL 语句都有很多执行路径，优化的目的就是在这些执行路径里选择最优的执行路径)。
- 8) 存储引擎：用于管理存储的文件系统，不同的存储引擎有不同的功能和存储方式。

9 简述 MySQL 数据库中插入、更新、查询、删除表记录的指令格式。

连接到 MySQL 数据库服务器，练习以下表记录操

作：向表中插入记录的语法格式

更新表记录的语

法格式查询表记

录的语法格式删

除表记录的语法

格式参考答案

- 1) 向表中插入记录的语法格式
`insert (into) 表名 (字段名列表) values(字段名=值, 字段名=值, ...);`
- 2) 更新表记录的语法格式
`update 表名 (set) (字段名=值, 字段名=值, ...) where (条件表达式列表);`
- 3) 查询表记录的语法格式
`select (字段名列表) (from) 表名 (where) 条件表达式列表;`
- 4) 删除表记录的语法格式
`delete (from) 表名 (where) (条件表达式列表);`

10 简述用户授权命令的语法格式。

参考答案

grant 权限列表 on 数据库 to 用户名@“客户端地址”

identified by “密码” with grant option;

11 在 MySQL-MMM 集群中有几种角色，各自的功能是什么？

参考答案

共3 种角色：客户端、monitor 节点（管理节点）、agent 节点（数据库节点）

客户端：访问集群

管理节点：负责所有的监控工作的监控守护进程，决定故障节点的移除或恢复

数据库节点：运行在 MySQL 服务器上的代理守护进程，提供简单远程服务集、提供给监控节点（可用来更改只读模式、复制的主服务器等）

12 在 MySQL-MMM 集群中可以有多少台主数据库服务器、多少台从数据库服务器？

参考答案

在 MySQL-MMM 环境中主数据库有且只能有 2 台，从数据库理论上可以任意多台。

13 在 MySQL-MMM 集群中均衡模式和排他模式的作用是什么？

参考答案

均衡模式一般用于从数据库，可实现多个虚拟 IP 地址。

排他模式一般用于主数据库，只可设置一个虚拟 IP 地址。

14,简述 mysqldump 备份数据时数据库名的表示方式。

参考答案

--all-databases

所有库数据库名

指定单个库

数据库名. 表名 指定库里的指定表

-B 数据 1 数据库 2 备份多个库

15,什么是 MySQL 集群？

答案：MySQL 集群是一个无共享的（shared-nothing），分布式节点架构的存储方案，其目的是提供容错性和高性能。数据在单个数据节点（有时也称存储节点）上存储和复制，每个数据节点运行在独立的服务器上并维护数据的一份拷贝。每个集群还有管理节点。数据更新使用读已提交隔离级别（read-committed isolation）来保证所有节点数据的一致性，使用两阶段提交机制（two-phased commit）保证所有节点都有相同的数据（如果任何一个写操作失败，则更新失败）。

MySQL 集群的最初实现将所有信息都保存在主存内，没有任何永久性存储。后来 MySQL 集群允许数据存储在磁

盘上。通过存储引擎层 MySQL 服务器作为查询引擎，可以使 MySQL 集群的性能达到最佳。这样就可以将 MySQL 应用透明地迁移到 MySQL 集群中去。

无共享的对等节点使得某台服务器上的更新操作在其他服务器上立即可见。传播更新使用一种复杂的通信机制，这一机制专用来提供跨网络的高吞吐量。该架构通过多个 MySQL 服务器分配负载，从而最大程度地达到高性能，通过在不同位置存储数据保证高可用性和冗余。

16，MySQL 集群和 MySQL 有和不同？

答案：你可能会问：“集群和复制之间有什么区别那？” 集群的定义很多，通常认为集群包含成员、消息、冗余和自动故障转移等功能，而复制仅仅是一个服务器向别一个服务器发送消息（数据）的方式。我们先讨论集群内部的复制（又称本地复制）。

17，MySQL 集群的特点？

答案：为了实现最高性能。高可用性和冗余等目标。数据在集群内部的对等数据节点之间互相复制。数据复制采用同步机制，每个数据节点到所有其他数据节点上，数据在多个数据节点上存储。

18，MySQL 集群有一些创建高可用性系统的专用功能，主要包括？

答案：节点回复、日志、检查点、系统恢复、热备份恢复、无单点故障、故障转移、分区、联机操作

19、MySQL 中 myisam 与 innodb 的区别，至少 5 点

【评析】将 Mysql 常见的存储引擎的特点归纳表格如下

点	myisam	innodb	memory	archive
存储限制	256TB	64TB	有	无
事物安全	不	支持	不	不
支持索引	支持	支持	支持	不支持
锁颗粒（锁力度）	表锁	行锁（没有索引的情况是表锁）	表锁	行锁
数据压缩	支持	不支持	支持	不支持

支持外键	不	支持	不	不
------	---	----	---	---

20 什么是读写分离?

MySQL Proxy 最强大的一项功能是实现“读写分离(Read/Write Splitting)”。基本的原理是让主数据库处理事务性查询，而从数据库处理SELECT 查询。数据库复制被用来把事务性查询导致的变更同步到集群中的从数据库。当然，主服务器也可以提供查询服务。使用读写分离最大的作用无非是环境服务器压力。

21 读写分离的好处

1. 增加冗余
2. 增加了机器的处理能力
3. 对于读操作为主的应用，使用读写分离是最好的场景，因为可以确保写的服务器压力更小，而读又可以接受点时间上的延迟。

22 读写分离提高性能之原因

1. 物理服务器增加，负荷增加
 2. 主从只负责各自的写和读，极大程度的缓解 X 锁和S 锁争用
 3. 从库可配置 myisam 引擎，提升查询性能以及节约系统开销
 4. 从库同步主库的数据和主库直接写还是有区别的，通过主库发送来的 binlog 恢复数据，但是，最重要区别在于主库向从库发送 binlog 是异步的，从库恢复数据也是异步的
 5. 读写分离适用与读远大于写的场景，如果只有一台服务器，当 select 很多时，update 和delete 会被这些 select 访问中的数据堵塞，等待 select 结束，并发性能不高。对于写和读比例相近的应用，应该部署双主相互复制
 6. 可以在从库启动是增加一些参数来提高其读的性能，例如--skip-innodb、--skip-bdb、--low-priority-updates 以及--delay-key-write=ALL。当然这些设置也是需要根据具体业务需求来定得，不一定能用上
 7. 分摊读取。假如我们有 1 主3 从，不考虑上述 1 中提到的从库单方面设置，假设现在 1 分钟内有 10 条写入，150 条读取。那么，1 主3 从相当于共计 40 条写入，而读取总数没变，因此平均下来每台服务器承担了 10 条写入和 50 条读取（主库不承担读取操作）。因此，虽然写入没变，但是读取大大分摊了，提高了系统性能。另外，当读取被分摊后，又间接提高了写入的性能。所以，总体性能提高了，说白了就是拿机器和带宽换性能。MySQL 官方文档中有相关演算公式：官方文档 见 6.9FAQ 之“MySQL 复制能够何时和多大程度提高系统性能”
 8. MySQL 复制另外一大功能是增加冗余，提高可用性，当一台数据库服务器宕机后能通过调整另外一台从库来以最快的速度恢复服务，因此不能光看性能，也就是说 1 主1 从也是可以的。
-

23 varchar 与 char 的区别; varchar(50)中 50 的涵义; int(20)中 20 的涵义;

char 是定长变量, varchar 是变长变量。 varchar(50)表示这一行的变量最大的存储字节是 50 个字节, int(20)同理。

【评析】假设有一行是 name char(8), 如果有一个数据是叫 AAA, 那么它仅仅只有三个字节被存储进去, 但是依旧存储了 8 个字节, 多余的 5 个字节空着也就空着了。而是 name varchar(8), 同样是 AAA, 由于是变长, 所以只保存了 3 个字节, 剩下 5 个字节是弹性的, 有就用, 没有就不用。

在读取方面, char 的读取速度要比 varchar 快, 也就是常说的“用读取换容量”, 但是还是多用 varchar, 当数据库内容成万上亿的时候, 节省的容量是非常非常可观的。

24 计划, mysqldump 以及 xtrabackup 的实现原理; 备份恢复时间; 备份恢复失败如何处理。

mysqldump 是采用 sql 级别的备份机制, 将数据表导出成 sql 脚本文件, 在不用的 mysql 版本之间升级时相对比较合适。xtrabackup 是 innodb 的 hotbackup 工具, xtrabackup 在启动的时候会复制所有的数据文件, 同时会启动一个后台进程, 用于监视事务日志, 并且从事务日志复制最新的修改。所以 xtrabackup 在启动的开始, 就不懂的将事务日志的每个数据文件的修改都记录下来。

mysqldump 的备份和恢复时间都很慢, 任何数据的更新和变化都会被挂起。

xtrabackup 的恢复时间比 mysqldump 快一点, 但是会锁表。

备份恢复失败的话, 其实原因很多, 主要可能就是参数设置的不对, 检查一下参数。

【评析】使用 mysqldump 备份数据表的命令, 在 shell 下执行:

mysqldump -u 用户名 -p 密码 (可以直接 -p) -h 主机名 --databases 数据库名 > 要备份的文件路

径mysqldump -u 用户名 -p 密码 -h 主机名 --all-databases >要备份的文件路径

mysqldump -u 用户名 -p 密码 -h 主机名 --no-data 数据库名 >要备份的文件

路径这里并不全, 另写文章专门补充。

25 MySQL 中 InnoDB 引擎的行锁是通过加在什么上完成(或称实现)的? 为什么是这样子的?

InnoDB 的行锁是通过加在索引上实现的, 为什么这么设计, 我也不知道, 去问 mysql 的设计公司。

26 MySQL 数据库备份方式

增量备份、差异备份、完整备份

27 MySQL 主从复制原理？

答案：

分为同步复制和异步复制，实际复制架构中大部分为异步

复制。复制的基本过程如下：

1)、Slave 上面的 IO 进程连接上 Master，并请求从指定日志文件的指定位置（或者从最开始的日志）之后的日志内容；

2)、Master 接收到来自 Slave 的 IO 进程的请求后，通过负责复制的 IO 进程根据请求信息读取制定日志指定位置之后的日志信息，返回给 Slave 的 IO 进程。返回信息中除了日志所包含的信息之外，还包括本次返回的信息已经到 Master 端的 bin-log 文件的名称以及 bin-log 的位置；

3)、Slave 的 IO 进程接收到信息后，将接收到的日志内容依次添加到 Slave 端的 relay-log 文件的最末端，并将读取到的 Master 端的 bin-log 的文件名和位置记录到 master-info 文件中，以便在下次读取的时候能够清楚的告诉 Master “我需要从某个 bin-log 的哪个位置开始往后的日志内容，请发给我”；

4)、Slave 的 Sql 进程检测到 relay-log 中新增加了内容后，会马上解析 relay-log 的内容成为在 Master 端真实执行时候的那些可执行的内容，并在自身执行。

Mysql 为了解决这个风险并提高复制的性能，将 Slave 端的复制改为两个进程来完成。提出这个改进方案的人是 Yahoo! 的一位工程师 “Jeremy Zawodny”。这样既解决了性能问题，又缩短了异步的延时时间，同时也减少了可能存在的数据丢失量。当然，即使是换成了现在这样两个线程处理以后，同样也还是存在 slave 数据延时以及数据丢失的可能性的，毕竟这个复制是异步的。只要数据的更改不是在一个事物中，这些问题都是会存在的。如果要完全避免这些问题，就只能用 mysql 的 cluster 来解决了。不过 mysql 的 cluster 是内存数据库的解决方案，需要将所有数据都 load 到内存中，这样就对内存的要求就非常大了，对于一般的应用来说可实施性不是太大。

复制常用架构

Mysql 复制环境 90% 以上都是一个 Master 带一个或者多个 Slave 的架构模式，主要用于读压力比较大的应用的数据库廉价扩展解决方案。因为只要 master 和 slave 的压力不是太大（尤其是 slave 端压力）的话，异步复制的延时一般都很少很少。尤其是自 slave 端的复制方式改成两个进程处理之后，更是减小了 slave 端的延时。而带来的效益是，对于数据实时性要求不是特别的敏感度的应用，只需要通过廉价的 pc server 来扩展 slave 的数量，将读压力分散到多台 slave 的机器上面，即可解决数据库端的读压力瓶颈。这在很大程度上解决了目前很多中小型网站的数据库压力瓶颈问题，甚至有些大型网站也在使用类似方案解决数据库瓶颈。

28 mysql-mmm 简介

答案：

MMM(Master-Master replication manager for Mysql) 是一套灵活的脚本程序，用来对 mysql replication 进行监控和故障迁移，并能管理 mysql Master-Master 复制的配置

（同一时间只有一个节点是可写的）。附带的工具套件可以实现多个 slaves 的 read 负载均衡，因此你可以使用这个工具移除一组服务器中复制延迟较高的服务器的虚拟 IP，它还可以备份数据，两节点之间再同步等等。

Mysql-mmm 是一套脚本程序，基于 perl 实现，通过不同的 perl 脚本实现对 mysql 服务器的管理与维护。它仅仅是一个管理程序，自己本身并不提供 mysql 服务功能。

被管理的 mysql server 机需要安装相关的 agent 脚本，mysql-mmm 的监控端可以监管所以安装过此脚本的 mysql server。

当出现多台可写 mysql server 时，mmm 可以保证在同一时间点只使用一台 mysql server 进行写入操作，以保证数据有效性，防止写入冲突。所以它并不适用于有大并发写入要求的生产环境。相反，当有多台可读 mysql server 存在时，它可以通过一些其他软件的配合，实现负载均衡方式的读取，大大提高 mysql server 的读性能。

在运行过程中，如果某一台 mysql server 处于不可用状态时，mmm 可以将原有操作请求迁移至其他可用 mysql，从而实现服务的高可用性。包括写请求迁移，读请求迁移，主从同步的 master 迁移。

Mysql+mmm 同时也提供了一套很好用的运维管理工具，可以实现简单快捷的数据备份与维护。简化管理员的日常维护成本。

29 mysql-mmm 组成与原理

答案:

MySQL-mmm 的管理功能主要通过三个脚本来实现

`mmm_mond`

监控进程，负责所有的监控工作，决定和处理所有节点角色活动。此脚本需要在监管机上运行。

`mmm_agentd`

运行在每个 `mysql` 服务器上的代理进程，完成监控的探针工作和执行简单的远端服务设置。此脚本需要在被监管机上运行。

`mmm_control`

一个简单的脚本，提供管理 `mmm_mond` 进程的命令

`mysql-mmm` 的监管端会提供多个虚拟 IP（VIP），包括一个可写 VIP，多个可读VIP，通过监管的管理，这些 IP 会绑定在可用 `mysql` 之上，当某一台 `mysql` 宕机时，监管机会将 VIP 迁移至其他 `mysql`。

在整个监管过程中，需要在 `mysql` 中添加相关授权用户，以便让 `mysql` 可以支持监管机的维护。授权的用户包括一个 `mmm_monitor` 用户和一个 `mmm_agent` 用户，如果想使用 `mmm` 的备份工具则还要添加一个 `mmm_tools` 用户。

30 多源复制下，支持 master 是 5.6，slave 是 5.7 吗？

答：可以的

【老叶补充】非常不建议跨大版本的 MySQL Replication。

31 半同步复制设置 N 个 slave 应答，如果当前 Slave 小于 N 会怎样？

答：取决于 `rpl_semi_sync_master_wait_no_slave` 的设置。

- `rpl_semi_sync_master_wait_no_slave`

`ve = 0` 会立刻变成异步复制。

- `rpl_semi_sync_master_wait_no_slave`

`ve = 1` 仍然等待应答，直到超时。

32 gtid 如果出现不支持的语句，怎么解决

答：从应用中去掉不支持的语句/

事务：例如：

- `CREATE TABLE ... SELECT`

可以该

为：

`CREATE`

`TABLE`

`INSERT ... SELECT`

33 基于 GTID 的复制，可以指定 GTID 复制的起始位置么，还是只能根据现有的信息？

答：GTID 复制就是为了摆脱对binlog 文件名和位置的依赖。所以不能指定复制的起始位置，也完全没有必要指定。

34 对一个已经开启 GTID 的数据库再做一个从库，先把 Master 备份下来还 原到新 slave 上去，直接可以同步了还是先需要做 purge_gtid 的操作再同步呢？

答：需要设置 gtid_purged。整个过程可以通过 mysqldump 完成。

请 参 考 手 册 : <http://dev.mysql.com/doc/refman/5.7/en/replication-gtid-backup.html>

35 线上全是 5.5 的环境，有没有办法搭建 5.5 到 5.7 的复制？

答：可以，但不能开启 gtid 功能。

【老叶补充】非常不建议跨大版本的 MySQL Replication，更何况是垮了 2 个大版本，最好是先进行升级。

36 并行复制 logical_clock，如果不开启 gtid 是不是就不能并行了？

答：不开启 gtid, 也能使用 logical_clock 并行复制。

37 在主从复制过程中，是主向从推数据还是从拉数据，如果这个传送的过程中，出现网络闪断，会不会造成数据包丢失，会执行校验重传嘛？

答：Slave 的 IO 线程发起到 Master 的连接。

然后 master 开始发送 events, slave 只是被动的接收。

slave 和 master 之间使用的是tcp 连接。tcp 是可靠的连接。网络丢包等都是 tcp 层自动处理的。mysql 不需要处理。

如果 slave 长时间不能收到一个完整的 Event, 或者接收 event 时出错。slave 会进行相应的处理。

如果 slave 认为，重新建立连接能解决问题。slave 则自动的断开原来的连接，然后重新连接到 master 去。

如果 slave 认为，这个错误无法自动解决，slave 会停掉 io 线程，并报错。

38 设置了 binlog_group_commit_sync_delay 参数，在宕机的时候应该不会

影响 binlog 文件安全吧？

答：不影响

39 多源复制是只能在异步模式下使用么？因为半同步的状态是全局的，
—

一个通道关闭会导致其它通道出错， 有没有考虑将半同步状态改成每个 channel 的私有配置呢？

答：只有一个通道可以开启 seimsync，其他的都要使用异步通道。不光是 Semisync，异步通道的状态也要改成每个channel 的私有配置，都有考虑。