

# MySQL 面试指南

作者: [Hefery](#)

原文链接: <https://ld246.com/article/1642245658602>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## MySQL 版本类问题

### 你之前工作中使用的是哪个版本的 MySQL-为什么选择这个版本-

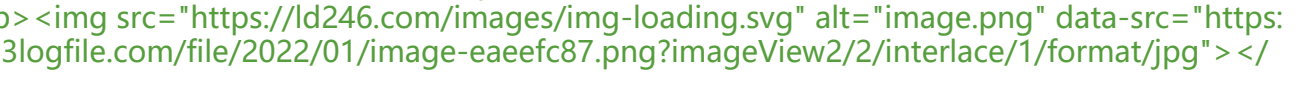
知识点

- 

MySQL 常见的发行版：官方社区版、官方企业版、Percona MySQL、MariaDB

- 
- 

各个发行版之间的区别及优缺点

 <https://ld246.com/images/img-loading.svg> alt="image.png" data-bbox="100 217 918 261"/> <https://b3logfile.com/file/2022/01/image-eaeefc87.png?imageView2/2/interlace/1/format/jpg>

- 



### 如何决定是否要对 MySQL 进行升级-如何进行升级-

在对 MySQL 进行升级前要考虑什么?

- 升级可以给业务带来的益处

- 是否可以解决业务上某一方面的痛点
- 是否可以解决运维上某一方面的痛点

- 
- 升级可能对业务带来的影响

- 对原业务程序的支持是否有影响
- 对原业务程序的性能是否有影响

- 
- 数据库升级方案的制定

- 评估受影响的业务系统
- 升级的详细步骤 (演练)
- 升级后的数据库环境检查
- 升级后的业务检查

- 
- 升级失败的回滚方案

- 升级失败回滚的步骤
- 回滚后的数据库环境检查
- 回滚后的业务检查

- 



MySQL 升级的步骤

- 对待升级数据库进行备份
- 升级 Slave 服务器版本
- 手动进行主从切换
- 升级 MASTER 服务器版本
- 升级完成后进行业务检查

</ul>

<h3 id="最新的MySQL版本是什么-它有什么特性比较吸引你-">最新的 MySQL 版本是什么? 它有什么特性比较吸引你? </h3>

<p>MySQL 8.0</p>

<p>特性</p>

<ul>

<li>服务器特性<br>

所有元数据使用 InnoDB 引擎存储, 无 frm 文件系统表。采用 InnoDB 存储并采用独立表空间<br>

支持定义资源管理组 (目前仅支持 CPU 资源) <br>

支持窗口函数<br>

支持在线修改全局参数持久化</li>

<li>用户安全<br>

默认使用 caching\_sha2\_password 认证插件<br>

新增支持定义角色 (role) <br>

新增密码历史记录功能, 限制重复使用密码</li>

</ul>

<h2 id="用户管理类问题">用户管理类问题</h2>

<h3 id="如何在给定场景下为某用户授权-">如何在给定场景下为某用户授权? </h3>

<p>知识点</p>

<ul>

<li>如何定义 MySQL 数据库账号? <br>

用户名 @ 可访问控制列表: hefery@<a href="https://ld246.com/member/" aria-name="" class="tooltipped\_user" target="\_blank"></a>192.168.1.%<br>

使用 CREATE USER 命令建立用户</li>

<li>MySQL 常用的用户权限

<ul>

<li>Admin<br>

Create User: 建立新的用户的权限<br>

Grant option: 为其他用户授权的权限<br>

Super: 管理服务器的权限</li>

<li>DDL<br>

Create: 新建数据库, 表的权限<br>

Alter; 修改表结构的权限<br>

Drop: 删除数据库和表的权限<br>

Index: 建立和删除索引的权限</li>

<li>DML<br>

Select: 查询表中数据的权限<br>

Insert: 向表中插入数据的权限<br>

Update: 更新表中数据的权限<br>

Delete: 删除表中数据的权限<br>

Execute: 执行存储过程的权限</li>

</ul>

</li>

<li>如何为用户授权?

<ul>

<li>遵循最小权限原则</li>

<li>使用 Grant 命令对用户授权

<ul>

<li>grant select,insert,update,delete on db.tb to <a href="https://ld246.com/forward?goto=ailto%3Auesr%40ip" target="\_blank" rel="nofollow ugc">uesr@</a> <a href="https://ld246.com/member/ip" aria-name="ip" class="tooltipped\_user" target="\_blank">ip</a></li>

<li>revoke delete on db.tb from <a href="https://ld246.com/forward?goto=mailto%3Auesr%40ip" target="\_blank" rel="nofollow ugc">uesr@</a> <a href="https://ld246.com/member/ip" aria-name="ip" class="tooltipped\_user" target="\_blank">ip</a></li>

</ul>  
</li>  
</ul>  
</li>  
</ul>

### 如何保证数据库账号的安全? </h3>

<p>知识点</p>

- <li>数据库用户管理流程规范</li>  - <li>密码管理策略</li>  - <li>数据库用户管理流程规范
- <li>最小权限原则</li>  - <li>密码强度策略</li>  - <li>密码过期原则</li>  - <li>限制历史密码重用原则</li>

</ul>

</li>

</ul>

### 如何从一个实例迁移数据库账号到另一个例? </h3>

<ul>

<li>

<p>对比数据库版本是否一致</p>

<ul>

<li>是就备份数据库，在目的实例恢复</li>

<li>否就导出授权语句 (pt-show-grants u=root,p=123456,h=localhost) ，在目的实例执行</li>

</ul>

</li>

</ul>

## 服务器配置类问题</h2>

### 请分析一个 Group By 语句的异常原因</h3>

<p>结果集并未出现预期结果</p>

<ul>

<li>SQL\_MODE: 配置 MySQL 处理 SQL 的方式<br>

set [session/global/persist] sql\_mode='xxxxxx'<br>

[mysqld] sql\_mode=xxxxxx</li>

<li>常见 SQL\_MODE<br>

ONLY\_FULL\_GROUP\_BY: 对于 GROUPBY 聚合操作，如果出现在 SELECT 中的列、HAVING 或者 RDERBY 子句的非聚合列，没有在 GROUP BY 中出现，那么这个 SQL 语法检查报错<br>

ANSI\_QUOTES: 禁止用双引号来引用字符串<br>

REAL\_AS\_FLOAT: Real 做为 float 的同义词<br>

PIPES\_AS\_CONCAT: 将 || 视为字符串的连接操作符而非或运算符<br>

STRICT\_TRANS\_TABLES/STRICT\_ALL\_TABLES: 在事务存储引擎/所有存储引擎上启用严格模式出，那么这个 SQL 语法检查报错<br>

ERROR\_FOR\_DIVISION\_BY\_ZERO: 不允许 0 做为除数<br>

NO\_AUTO\_CREATE\_USER: 在用户不存在时不允许 Grant 语句自动建立用<br>

NO\_ZERO\_IN\_DATE/NO\_ZERO\_DATE: 日期数据内/日期数据不能含 0<br>

NO\_ENGINE\_SUBSTITUTION: 当指定的存储引擎不可用时报错</li>

</ul>

### 如何比较系统运行配置和配置文件的配置是否一致? </h3>

<p>知识点</p>

<ul>

<li>使用 set 命令配置动态参数<br>

set [session | @@session.]system\_var\_name=expr<br>

set [global | @@global.]system\_var\_name=expr<br>

set [persist | @@persist.]system\_var\_name=expr</li>

<li>使用 pt-config-diff 工具比较配置文件<br>

pt-config-diff u=root,p=h=localhost /etc/my.cnf</li>

</ul>

<h3 id="举几个MySQL中的关键性能参数">举几个 MySQL 中的关键性能参数</h3>

<ul>

<li>服务器配置参数<br>

max\_connections: 设置 MySQL 允许访问的最大连接数量<br>

interactive\_timeout: 设置交互连接的 timeout 时间<br>

wait\_timeout: 设置非交互连接的 timeout 时间<br>

max\_allowed\_packet: 控制 MySQL 可以接收的数据包的大小<br>

sort\_buffer\_size: 设置每个会话使用的排序缓存区的大小<br>

join\_buffer\_size: 设置每个会话所使用的连接缓冲的大<br>

read\_buffer\_size : 指定了当对一个 MYISAM 进行表扫描时所分配的读缓冲池大小<br>

read\_rnd\_buffer\_size: 设置控制索引缓冲区的大小<br>

binlog\_cache\_size 设置每个会话用于缓存未提交的事务缓存大小</li>

<li>存储引擎参数<br>

innodb\_flush\_log\_atrx\_commit: <br>

0: 每秒进行一次重做日志的磁盘刷新操作。<br>

1: 每次事务提交都会刷新事务日志到磁盘中。<br>

2: 每次事务提交写入系统缓存每秒向磁盘刷新一次<br>

innodb\_buffer\_pool\_size: 设置 Innodb 缓冲池的大小, 应为系统可用内存的 75%<br>

innodb\_buffer\_pool\_instances: Innodb 缓冲池的实例个数, 每个实例的大小为总缓冲池大小/实例个数<br>

innodb\_file\_per\_table: 设置每个表独立使用一个表空间文件</li>

</ul>

<h2 id="日志类问题">日志类问题</h2>

<h3 id="常用的MySQL日志有那些-我们在什么情况下使用这些日志-">常用的 MySQL 日志有那些我们在什么情况下使用这些日志? </h3>

<p>知识点</p>

<ul>

<li>MySQL 常用的日志类型</li>

<li>各种日志的配置和使用场景</li>

</ul>

<h3 id="如何通过日志来审计用户活动-">如何通过日志来审计用户活动? </h3>

<h2 id="存储引擎相关问题">存储引擎相关问题</h2>

<h3 id="说一说你了解的MySQL存储引擎及其适用场景">说一说你了解的 MySQL 存储引擎及其适用场景</h3>

<p>知识点</p>

<ul>

<li>MySQL 常用的存储引擎

<ul>

<li>MYISAM: 不支持事务, 最常用的非事务型存储引擎</li>

<li>CSV: 不支持事务, 以 CSV 格式存储的非事务型存储引擎</li>

<li>Archive: 不支持事务, 只允许查询和新增, 不允许修改的非事务型存储引擎</li>

<li>Memory: 不支持事务, 读写速度非常快, 易失性非事务型存储引擎</li>

<li>InnoDB: 最常用的事务型存储引擎</li>

<li>NDB: MySQL 集群所使用的内存型事务存储引擎</li>

</ul>

</li>

<li>各种常见存储引擎的使用场景

```
<ul>
<li>MYISAM
<ul>
<li>特点<br>
非事务型存储引擎<br>
以堆表方式存储<br>
使用表级锁<br>
支持 Btree 索引, 空间索引, 全文索引</li>
<li>适用场景<br>
读操作远远大于写操作的场景<br>
不需要使用事务的场景</li>
</ul>
</li>
<li>CSV
<ul>
<li>特点<br>
非事务型存储引擎<br>
数据以 CSV 格式存储<br>
所有列都不能为 NULL<br>
不支持索引</li>
<li>适用场景<br>
做为数据交换的中间表使用</li>
</ul>
</li>
<li>Archive
<ul>
<li>特点<br>
非事务型存储引擎<br>
表数据使用 zlib 压缩<br>
只支持 Insert 和 Select<br>
只允许在自增 ID 上建立索引</li>
<li>适用场景<br>
日志和数据采集类应用<br>
数据归档存储</li>
</ul>
</li>
<li>Memory
<ul>
<li>特点<br>
非事务型存储引擎<br>
数据保存在内存中<br>
所有字段长度固定<br>
支持 Btree 和 Hash 索引</li>
<li>适用场景<br>
用于缓存字典映射表<br>
缓存周期性分析数据</li>
</ul>
</li>
<li>InnoDB
<ul>
<li>特点<br>
事务型存储引擎支持 ACID<br>
数据按主键聚集存储<br>
支持行级锁及 MVCC<br>
```

支持 Btree 和自适应 Hash 索引<br>

支持全文及空间索引</li>

<li>适用场景<br>

大多数 OLTP 场景</li>

</ul>

</li>

</ul>

</li>

</ul>

<h3 id="在什么情况下InnoDB无法在线修改表结构-">在什么情况下 InnoDB 无法在线修改表结构?</h3>

<p>知识点</p>

<ul>

<li>InnoDB 不支持在线修改表结构的场景<br>

加全文索引: CREATE FULLTEXT INDEX name ON table (column) <br>

加空间索引: ALTER TABLE geom ADD SPATIAL INDEX (g) <br>

删除主键: ALTER TABLE tbl\_name DROP PRIMARY KEY<br>

增加自增列: alter table t add column id int auto\_increment not null primary key<br>

修改列类型: ALTER TABLE tbl\_name CHANGE c1 c1 NEW\_TYPE<br>

改表字符集: ALTER TABLE tbl\_name CHARACTER SET=charset\_name</li>

<li>如何更安全的在线修改表结构<br>

pt-online-schema-change [OPTIONS] DSN</li>

</ul>

<h3 id="InnoDB是如何实现事务的-">InnoDB 是如何实现事务的? </h3>

<p>知识点</p>

<ul>

<li>什么是事务? <br>

原子性 (A) : 一个事务中的所有操作, 要么全部完成, 要么全部不完成, 不会结束在中间某个环节<br>

一致性 (C) : 在事务开始之前和事务结束以后, 数据库的完整性没有被破坏<br>

隔离性 (I) : 事务的隔离性要求每个读写事务的对象与其它事务的操作对象能相互分离, 即该事务提前对其它事务都不可见<br>

持久性 (D) : 事务一旦提交了, 其结果就是永久性的, 就算发生了宕机等事故, 数据库也能将数据复</li>

<li>事务的实现方式<br>

原子性 (A) : 回滚日志 (Undo log) : 用于记录数据修改前的状态<br>

一致性 (C) : 重作日志 (Redo Log) : 用于记录数据修改后的状态<br>

隔离性 (I) : 锁: 用于资源隔离, 分为共享锁和排它锁<br>

持久性 (D) : 重作日志 (Redo log) + 回滚日志 (Undo Log) </li>

</ul>

<p></p>

<h3 id="InnoDB读操作是否会阻塞写操作-">InnoDB 读操作是否会阻塞写操作? </h3>

<p>知识点</p>

<ul>

<li>InnoDB MVCC (多版本并发控制) 的实现方式</li>

</ul>

<p>查询需要对资源加共享锁 (S) <br>

数据修改需要对资源加排它锁 (x) </p>

<h2 id="MySQL架构类问题">MySQL 架构类问题</h2>

<h3 id="MySQL的主从复制是如何工作">MySQL 的主从复制是如何工作</h3>

<p>知识点</p>

<ul>

- <li>MySQL 主从复制的实现原理</li>
- <li>MySQL 主从复制的配置步骤<br>在 MASTER 服务器上的操作<br>开启 binlog (必须) 开启 gtid (可选) <br>建立同步所用的数据库账号<br>使用 master\_data 参数备份数据库<br>把备份文件传输到 Slave 服务器<br>在 Slave 服务器上的操作<br>开启 binlog (可选) 开启 gtid (可选) <br>恢复 Master 上的备份数据库<br>使用 Change master 配置链路<br>使用 start slave 启动复制</li>

### <p>知识点</p> <ul> - <li>什么是基于日志点的复制<br> 传统的主从复制方式<br> Slave 请求 Master 的增量日志依赖于日志偏移量<br> 配置链路时需指定 master\_log\_file 和 master\_log\_pos 参数</li> - <li>什么是基于 GTID 的复制<br> GTID=source id:transaction id<br> Slave 增量同步 Master 的数据依赖于其未同步的事务 ID<br> 配置复制链路时, SLAVE 可以根据已经同步的事务 ID 继续自动同步 </li> - <li>这两种复制方式各自的特点 </li> </ul> <p>知识点</p> <ul> - <li>MMM 和 MHA 两种架构的作用</li> - <li>MMM 架构的优缺点及适用场景</li> - <li>MHA 架构的优缺点及适用场景</li> </ul> <p>知识点</p> <ul> - <li>主从复制延迟产生的原因</li> - <li>减小主从延迟的处理方法</li> </ul> <p>知识点</p> <ul> - <li>什么是 MGR 复制<br> 不同于异步复制的多 Master 复制集群</li> - <li>如何使用 MGR 复制</li> </ul> <p>知识点</p> <ul> - <li>如何解决读负载大的问题? 读写分离</li> 原文链接: [MySQL 面试指南](#)



<li>如何解决写负载大的问题? 分库分表</li>

</ul>

<h2 id="备份恢复类问题">备份恢复类问题</h2>

<h3 id="在之前的工作中是如何对数据库进行备份的-">在之前的工作中是如何对数据库进行备份的</h3>

<p>知识点</p>

<ul>

<li>备份方式<br>

逻辑备份 (SQL 文件) 和物理备份<br>

全量备份和增量备份以及差异备份</li>

<li>常用的备份工具

<ul>

<li>mysqldump: 最常用的逻辑备份工具, 支持全量备份及条件备份<br>

备份结果为可读的 SQL 文件, 可用于跨版本跨平台恢复数据<br>

备份文件的尺寸小于物理备份, 便于长时间存储<br>

MySQL 发行版自带工具, 无需安装第三方软件<br>

只能单线程执行备份恢复任务, 备份恢复速度较慢<br>

为完成一致性备份需要对备份表加锁, 容易造成阻塞。<br>

会对 InnoDB Buffer Pool 造成污染</li>

<li>mysqlpump: 多线程逻辑备份工具, mysqldump 的增强版本<br>

支持基于库和表的并行备份, 可以提高逻辑备份的性能<br>

支持使用 ZLIB 和 Lz4 算法对备份进行压缩<br>

对大表性能较差</li>

<li>xtrabackup: InnoDB 在线物理备份工具, 支持多线程和增量备份<br>

支持 InnoDB 存储引擎的在线热备份, 对 InnoDB 缓冲没有影响<br>

支持并行对数据库的全备和增量备份<br>

备份和恢复效率比逻辑备份高<br>

做单表恢复时比较复杂<br>

完整的数据文件拷贝, 故备份文件比逻辑备份大<br>

对跨平台和数据库版本的备份恢复支持度不如逻辑备份</li>

</ul>

</li>

</ul>

<h3 id="如何对MYSQL进行增量备份和恢复-">如何对 MYSQL 进行增量备份和恢复? </h3>

<p>使用 xtrabackup 进行增量备份</p>

<h3 id="如何对binlog进行备份-">如何对 binlog 进行备份? </h3>

<ul>

<li>备份方式<br>

利用 cp 命令进行离线备份<br>

使用 mysqlbinlog 命令在线时实备份</li>

</ul>

<h2 id="管理及监控类问题">管理及监控类问题</h2>

<h3 id="说说你都对MySQL进行过那些监控-">说说你都对 MySQL 进行过那些监控? </h3>

<ul>

<li>性能指标<br>

QPS: 数据库每秒钟处理的请求数量<br>

TPS: 数据库每秒钟处理的事务数量<br>

并发数: 数据库实例当前并行处理的会话数量<br>

连接数: 连接到数据库会话的数量<br>

缓存命中率: InnoDB 的缓存命中率</li>

<li>功能指标<br>

可用性: 数据库是否可正常对外提供服务<br>

阻塞: 当前是否有阻塞的会话<br>

死锁: 当前事务是否产生了死锁<br>

主从延迟: 数据库主从延迟时间<br>

主从状态: 数据库主从复制链路是否正常</li>

</ul>

<h3 id="这些监控是如何实现的-">这些监控是如何实现的? </h3>

<ul>

<li>QPS: 数据库每秒钟处理的请求数量<br>

show global status like 'com%'<br>

Show global status like 'Queries'<br>

QPS= (Queries2-Queries1) /时间间隔</li>

<li>TPS: 数据库每秒钟处理的事务数量<br>

show global status where Variable\_name in (com\_insert,'com\_delete','com\_update');<br>

Tcscom\_insert+com\_delete+com\_update<br>

TPS\*(TC2-Tc1)/(time2-time1)</li>

<li>并发数: 数据库实例当前并行处理的会话数量<br>

SHOW GLOBAL STATUS LIKE "Threads\_running"</li>

<li>连接数: 连接到数据库会话的数量<br>

SHOW GLOBAL STATUS LIKE 'Threads\_connected'</li>

<li>InnoDB 缓存命中率<br>

(Innodb\_buffer\_pool\_read\_requests - Innodb\_buffer\_pool\_reads) / Innodb\_buffer\_pool\_read\_requests\*100%<br>

Innodb\_buffer\_pool\_read\_requests: 从缓冲池中读取的次数<br>

Innodb\_buffer\_pool\_reads: 表示从物理磁盘读取的次数<br>

show global status like 'innodb\_buffer\_pool\_read%';</li>

</ul>

<p>报警阈值: Threads\_connected/max\_connections>0.8</p>

<ul>

<li>可用性: 数据库是否可正常对外提供服务<br>

周期性连接数据库服务器并执行 select @@version<br>

mysqladmin-uxxxx-pxxxxx-hxxxxx ping</li>

<li>阻塞

```
<pre><code class="language-sql highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-k">SELECT</span><span class="highlight-w"></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"></span></span><span class="highlight-n">waiting_pid</span><span class="highlight-w"></span><span class="highlight-k">AS</span><span class="highlight-w"></span><span class="highlight-s1">'被阻塞的线程'</span><span class="highlight-err">,</span><span class="highlight-w"></span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"></span></span><span class="highlight-n">waiting_query</span><span class="highlight-w"></span><span class="highlight-k">AS</span><span class="highlight-err">'</span><span class="highlight-s1">阻塞的</span><span class="highlight-k">SQL</span><span class="highlight-s1">,</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-s1">blocking_pid AS '</span><span class="highlight-err">阻塞线程</span><span class="highlight-s1">,</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-s1">blocking_query AS '</span><span class="highlight-err">阻塞</span><span class="highlight-k">SQL</span><span class="highlight-s1">,</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-s1">wait age AS '</span><span class="highlight-err">阻塞时间</span><span class="highlight-s1">,</span></span></span></span>
```

```
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-s1">  sql_kill_blocking_query AS “建议操作”
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-s1">FROM
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-s1">  sys.innodb_lock_waits
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-s1">WHERE (UNIX_TIMESTAMP()-UNIX_TIMESTAMP(wait started))_&gt;30
</span></span></span></code></pre>
```

- </li>
- <li>监控慢查询<br>通过慢查询日志监控<br>通过 information\_schema.PROCESSLIST 表实时监控</li>
- <li>死锁<br>

show engine innodb status<br>pt-deadlock-logger u=dba,p=xxxxx,h=127.0.0.1 --create-dest-table --dest u=dba,p=xxxxx,h=27.0.0.1,D=crn,t=deadlock<br>set global innodb\_print\_all\_deadlocks=on;</li>

<li>主从延迟：数据库主从延迟时间<br>show slave status; (Seconds\_Behind\_Master) <br>pt-heartbeat --user=xx --password=xxx -h master --create -table --database xxx --update --demonize --interval=1<br>pt-heartbeat --user=xx --password=xxx -h slave --database crn --monitor --daemonize --log/mp/slave\_lag.log</li>

<li>主从状态：数据库主从复制链路是否正常<br>show slave status; (Slave\_IO\_Running:YES; Slave\_SQL\_Running:YES) </li></ul>

## <p>数据库服务器负载过大的问题</p> - <li>现象：QPS（低谷）、CPU 占有率（峰值）</li> - <li>原因<br>服务器磁盘 IO 超负荷<br>存在大量阻塞线程<br>存在大量并发慢查询<br>存在其它占用 CPU 的服务<br>服务器硬件资源原因</li> <li>解决</li> - </ul> - <p>慢查询造成的磁盘 IO 爆表</p> - <li>原因<br>MySQL 输出大量日志<br>MySQL 正在进行大批量写<br>慢查询产生了大量的磁盘临时表</li> - <li>解决<br>优化慢查询，减少使用磁盘临时表<br>增加 tmp\_table\_size 和 max heap\_table\_size 参数的大小</li> </ul> <p>主从数据库数据不一致</p> 原文链接: [MySQL 面试指南](#)

```
<ul>
<li>现象<br>
主从数据库延迟为 0<br>
IO_THREAD 和 SQL_THREAD 状态为 YES<br>
相同查询在主从服务器中查询结果不同</li>
<li>原因<br>
对从服务器进行了写操作<br>
使用 sql_slave_skip_counter 或注入空事务的方式修复错误<br>
使用了 statement 格式的复制</li>
<li>解决<br>
设置 read_only=ON<br>
设置 super_read_only=ON<br>
使用 row 格式的复制</li>
</ul>
```

### 处理过哪些MySQL主从复制异常? </h3>

```
<ul>
<li>主服务器连接不上<br>
主从服务器间网络是否畅通 (ping) <br>
是否存在防火墙, 过滤了数据库端口<br>
复制链路配置的用户名和密码是否正确, 是否有相应权限</li>
<li>主键冲突问题<br>
跳过故障数据<br>
检查主从数据一致性<br>
直接删除从库主键冲突数据</li>
<li>数据行不存在<br>
跳过故障数据<br>
使用 pt-table-sync 修复数据</li>
<li>relay_log 损坏<br>
找到已经正确同步的日志点<br>
使用 reset slave 删除 relay_log<br>
在正确同步日志点后重新同步日志</li>
</ul>
```

### 你会从哪些方面对MySQL数据库进行优化? </h3>

```
<p>  </p>
```