**陈中原：**

第一部分：通用模块。此部分对 MySQL 整体概念、执行流程、数据库引擎、查询缓存、表空间、回表查询、数据类型间的区别、内存表、临时表、删除表的 n 种方式、枚举、视图、数据恢复等相关知识点对应的面试题进行解答。

第二部分：索引模块。索引的好坏直接影响数据库的性能，所以索引的面试题也是面试中必问的问题，此部分为索引对应的面试题合集。

第三部分：事务模块。事务决定了程序的稳定性，在 MySQL 中的地位也是首屈一指，也是面试中必问的面试题，此部分为事务对应的面试题合集。

第四部分：锁。锁包括：全局锁、表锁、行锁、死锁、乐观锁、悲观锁等，不同的数据库引擎支持的锁支持粒度也是不同的，此部分的面试题，让你彻底搞定锁相关的面试题。

第五部分：日志。日志看似不起眼，却是 MySQL 主备同步和容灾恢复以及问题排除的关键，当然也是面试中必问的问题，这部分会对不同的数据库引擎中的重点日志，进行详细的介绍。

第六部分：MySQL 操作命令和内置函数。MySQL 的操作命令，对于程序员或者 DBA 来说也是必须具备的一项技能，比如，用户和权限的创建、数据库相关信息的查询等，都离不开对 MySQL 命令行的掌握。对内置函数的掌握程度，代表了你对 MySQL 的掌握程度，善用 MySQL 提供的内置函数，会让你有事半功倍的效果，内置函数也是笔试中必考的面试题。

第七部分：性能优化和分布式。性能优化和分布式是面试中决定你高度的关键指标，其中性能优化包括了慢查询的分析和处理，对分布式的掌握体现了你的技术深度。

第八部分：开放性问题。很多大公司最后也会问一下没有标准答案的开放性问题，以考察面试者的技术能力边界和对待问题的分析思路，这部分助你更平稳的获得 offer。

**王福成：**

**【微服务】作为服务注册中心，Eureka比Zookeeper好在哪里?**

著名的CAP理论指出，一个分布式系统不可能同时满足C(一致性)、A(可用性)和P(分区容错性)。由于分区容错性P在是分布式系统中必须要保证的，因此我们只能在A和C之间进行权衡。

因此，Zookeeper 保证的是CP, Eureka 则是AP。

Zookeeper保证CP

当向注册中心查询服务列表时，我们可以容忍注册中心返回的是几分钟以前的注册信息，但不能接受服务直接down掉不可用。也就是说，服务注册功能对可用性的要求要高于一致性。但是zk会出现这样一种情况，当master节点因为网络故障与其他节点失去联系时，剩余节点会重新进行leader选举。问题在于，选举leader的时间太长，30~120s,且选举期间整个zk集群都是不可用的，这就导致在选举期间注册服务瘫痪。在云部署的环境下，因网络问题使得zk集群失去master节点是较大概率会发生的事，虽然服务能够最终恢复，但是漫长的选举时间导致的注册长期不可用是不能容忍的

Eureka保证AP

Eureka看明白了这一点，因此在设计时就优先保证可用性。Eureka各个节点都是平等的，几个节点挂掉不会影响正常节点的工作，剩余的节点依然可以提供注册和查询服务。而Eureka的客户端在向某个Eureka注册或时如果发现连接失败，则会自动切换至其它节点，只要有一台Eureka还在，就能保证注册服务可用(保证可用性)，只不过查到的信息可能不是最新的(不保证强一致性)。

除此之外，Eureka还有一种自我保护机制，如果在15分钟内超过85%的节点都没有正常的心跳，那么Eureka就认为客户端与注册中心出现了网络故障，此时会出现以下几种情况：

Eureka不再从注册列表中移除因为长时间没收到心跳而应该过期的服务

Eureka仍然能够接受新服务的注册和查询请求，但是不会被同步到其它节点上(即保证当前节点依然可用)

当网络稳定时，当前实例新的注册信息会被同步到其它节点中

因此， Eureka可以很好的应对因网络故障导致部分节点失去联系的情况，而不会像zookeeper那样使整个注册服务瘫痪。

**丁昌茂：**

a = [{"label": "1", "value": "打篮球", "name": "hobby"},

{"label": "2", "value": "读书", "name": "hobby"},

{"label": "3", "value": "听歌", "name": "hobby"},

{"label": "1", "value": "男", "name": "sex"},

{"label": "2", "value": "女", "name": "sex"}]

b = {"hobby": {"1": "打篮球", "2": "读书", "3": "听歌"}, "sex": {"1": "男", "2": "女"}}

**苏建飞：**

Python 实现布隆过滤器

附件：a.py