**面试记录8/28：**

岳新亮：

答题精确，回答的内容基本准确，

眼神不会太迷离

手上动作幅度不会太大

但是频率有点多集合的回答很完整,几乎涵盖集合的主要分类

.IO流只提问序列化流，回答依然较为详细,回答trainsent关键字

戴耀：

很有礼貌，声音较小（可能是自信不足），

眨眼次数过于频繁，

对于不熟悉的问题明说忘了，

对于问题没有一个清晰的理论框架，需要面试官驱动，

对于许多知识点自己的理解不够深，可能是此原因，导致对于单个知识没有清晰的思路

html不够熟练，

treemap不够熟悉

对于MySQL事务方面不够了解

简单sql语句使用不熟练

对于做过的项目实现功能不够了解（怀疑项目具有水分）

莫航：

面试过程能冷静面对，回答有自己的逻辑。

对于排序算法基本原理和个别差别不够了解

对于索引有一定基本了解，但不熟悉

对各种数据库类型有一定基本了解

对于JDBC语句的使用能够阐述利弊

对于部分集合（treeset）的实现原理较为熟悉

不了解hashset迭代的快速失败和安全失败

对于TCP/IP协议的连接过程和断开过程不熟练

对于线程生命周期有较好了解，但不熟练

对于IO流中的缓冲流底层有较好的了解

有自己的职业规划安排

**面试记录8/30：**

**岳新亮：**

介绍自己时：流畅自然

对于数据库的了解：卡壳了一次，非关系数据库了解有点少，数据库的整体阐述不太流畅

数据库的索引，索引的底层阐述详细

幻读，详细

mysql怎么解决事务并发问题，非常详细

inoDB怎么解决，不了解

数据库在更新表数据时是表锁还是行锁？

行锁不熟悉

java创建对象的过程，什么是类，可以

内存泄露，内存溢出，可以

死锁的发生条件，详细

项目：项目缺失

前端和后端的偏好，选择后端，也是可以全栈

**莫航：**

简历过于简陋，项目介绍不完整。

熟悉git的使用

基本了解jvm虚拟机

不了解垃圾回收机制的三色标记法

能清晰阐述hashcode方法和equals方法的实现。

能清晰阐述快速失败和失败安全。

熟悉市面上常用的关系型数据库与非关系型数据库

对于SQL Server不够了解

不了解什么是聚集索引什么是非聚集索引

对于索引的种类了解很多，但不清楚具体内容

了解事务的基本概念

熟悉sql语句的五种组成部分（DQL,DDL,DCL,DML,TCL）

不了解事务的底层原理（难）

熟悉事务的四大特性（A原子性C一致性I隔离性D持久性）

不能第一时间理解回答问题的具体切入点（序列化的实现原因和作用）

了解序列化的概念与作用

不够了解包装类integer的自动拆装箱原理

**于枭森：**

面试时间20分钟

自我介绍清晰完整，叙述有条理

对于归并排序和快速排序有一定了解，但不够熟练，叙述中有一定错误（快速排序的实现情况）

了解Jvm的垃圾回收机制的多种方式

熟悉垃圾回收机制

不了解集合的浅拷贝和深拷贝

较为了解多线程安全中的synchronized和lock锁

对事务隔离级别名称不敏感

在RR级别（第三级别）解决幻读方法阐述不完善

熟悉索引的底层数据结构

能清晰阐述B树和B+树，但有所不足

不了解缓冲流的底层

不了解内存溢出和内存泄露

愿意接受测试岗位。

**戴耀：**

自我介绍有些偏少，个人信息较为模糊

手部小动作有些多

大致了解面向对象的原理

不了解面向对象的三个阶段

对于幻读的情况阐述不正确

不够了解sql语句的五种组成部分（DQL,DDL,DCL,DML,TCL）回答缺少DML

不了解While和having的区别

不了解存储过程的缺点

不了解Sql的优化规则（三点）只能说出2点

能清晰阐述Java跨平台原理，各别措辞不合理

能大致阐述如何重写equals方法

熟悉方法的重载和重写

知道字符串的分割方法，但无法应对实际应用情况（url地址分割获取信息）

熟悉string stringbuffer和stringbuilder

不了解为什么要同时重写hashcode方法和equals方法

了解基本变量类型

了解强制类型转换现象

不够了解实现线程的三种方式

**面试记录09/01**

岳新亮：

◦ Java的优点：面向对象，指针，异常处理

◦ 双向链表：全

◦ 事务并发错误，引到脏读，脏读描述的好

◦ 负载均衡，介绍的不错，除权重策略之外的策略不知道

◦ 二叉树：说了定义和缺点，画出最小二叉树（哈夫曼树）

◦ 怎么设计索引？答：组合索引，账号唯一，没思路

◦ 索引的底层结构：mysql，innodb，b+树，5版本以前是MyISAM引擎答哈希表，答案：b树

◦ 介绍b➕树，答了b树和b➕树

◦ 超市收银项目：账户，结账退货，订单表，订单项表

◦ 续上，再加一个账户，收银的转账，答：账户id设置成主键…

◦ 双向队列：答了队列，双向队列说的不多

◦ 多线程的锁，还能锁一个对象，答的不错，但是，当前对象runable的例子反推出一个错误🙅

◦ 表单数据：存大量数据，引入非关系型数据库，Hbase惊喜

◦ javase，html

哈夫曼树7个数据回答错误

于枭森

自我介绍很流畅。

基本了解git的使用。

对于合并冲突跑偏到回滚版本号。

Java虚拟机（JDK1.8）的分区很了解栈，堆，方法区改为元数据区

对于栈比较了解，但整体介绍有欠缺

对于实现项目登录功能介绍很全面

B+树的介绍很全面，但描述过程略有卡顿

多线程并发的三种方式介绍

雪花算法不了解，但提出了哈希算法实现

红黑树的特点了解不够

二分查找区分小数据量和大数据量分别使用冒泡排序和快速排序

哈夫曼树5个数据（1，5，6，7，8）画图错误。

莫航：

经常使用git，具有github的账户

WebIO流文件传送，采用网络协议TCP，并采用对象序列化。

生成验证码末尾和首位为数字，其他为字母，采用字符串拼接，stringbuilder和stringbuffer，采用rundom方法生成随机数，使用ASCII码生成字母，面对大用户量采用将使用字符存放在一个数组中，回答很全面。

树的介绍很全面了解二叉树的种类

Treeset和hashset存储底层了解

Hashset存储原理回答很全面，对于相同对象的存储处理熟悉

不了解OCP原则

Mysql事务的底层原理了解日志记录，MVCC，

MVCC的序列号，间隙锁十分了解。

Sql语句ALL和ANY使用解释全面

不定量的用户对一张表数据更新采用行锁还是表锁，基本回答正确。

哈夫曼树6个数据（1、4、5、2、3、7）回答错误

戴耀：

CSS中选择器不熟悉

Java的跨平台原理了解

JVM虚拟机的垃圾回收机制基本了解，但不熟悉

数组和集合之间的区别不了解

了解进程和线程的区别

不了解Get和post的区别

对于反射不太熟悉

基本知道一些索引的种类

基本了解事务的四个特性ACID

了解持久存储对象方法

不了解哈夫曼树

**面试记录09/04**

于枭森：

来自哪所大学；介绍学历，课程，项目经历，还有学习时关于数据库的项目内容。

问题：

对于银行金融行业后端技术点了解；

回答：

Mysql：

金额变化需要使用事务进行同步变动

数据量庞大需要使用索引减少Io次数

Java

维护和操作需要采用多线程并发、IO流、标记算法。

回答散乱，不够整体。

问题：

对数据的插入和更新时采用行锁还是表锁？

回答

对少量数据采用行级锁，大量数据采用表级锁（）

问题；

JVM虚拟机的了解，JVM类是如何加载到内存中的？

回答：

需要先导入到类加载器中，分三种类加载器，启动类，扩展类，应用类

加载类要采用双亲委派机制。流程回答全面，提出了采用此机制的好处

问题：

破坏双亲委派机制的想法？

回答：

使用tomcat绕过机制

问题；

垃圾回收机制的三色算法？

回答：

三色标记法：对象分为白色，灰色，黑色，各个区别解释全面

一开始扫描放在灰色中。

问题：

还知道其他算法吗

回答；

计数器算法，流程回答全面，提出了有缺陷，但没有明确解释清楚

问题

Jvm内存结构/运行时数据区，

回答

回答了五大区域全面

问题

解释一下老年代

回答

解释了新生代到老年代的转换

问题：

为什么15岁将新生代转换为老年代？

尽量减少大数量的新生代向老年代转换，减少full GC的执行。

JDK8版本之前方法区的实现是什么？

他不记得了

问题

为什么用元空间替代永久代？

元空间使用硬盘内存，空间无限，减少内存移除

问题：

更改Java虚拟机的内存大小？

没回答上来

岳新亮：

1. 对数据库的了解？

答： 了解数据库的类型： 关系型和非关系型

关系型： 内部表通过链接形式 二维链接 保持数据的一致性

非关系型数据库： 用于存储大量数据，解决关系型数据库存储大量数据效率低的问题。

主流的关系型数据库； oracle mysql sql server

非关系型数据库 non-sql

1. 在数据库中你知道什么时RR隔离级别吗？

答： repeatable-read 第三级别 可重复读 解决了其中两个问题 ： 脏读和不可重复读的问题

对于RR级别，用的什么引擎？

答： 5之后用的innodb

如何解决幻读？

答： 三种方法：

1. 版本号 为事务添加版本号 为数据行添加版本号
2. 快照读： 查询的数据是copy的数据。不会影响原表
3. 间隙锁 ：在查询前后上锁。保持事务的原子性

3.MVCC知道伐？？

答：知道。

4.什么是聚集索引和非聚集索引？

* + 1. 答：主键是聚集 其他的是非聚集

5.B+树了解吗？

答：”了解 是对B树的优化。b树是在节点中存数据和指针。B+树在中间节点只存放索引，只在叶子节点存放数据。

解释底层叶子节点的优化？

答； 双向链表。

B+树相比于红黑树的优点？？？？从树的高度来回答？

答： 红黑树是二叉树。二叉树深度会更深。IO次数增多。B+树是多叉树，深度会减少。效率会 变高。红黑树需要自平衡，带来性能消耗。

1. 希望你谈谈JVM。内存结构？？

答： 分为五大块 方法区 堆内存 虚拟机栈 本地方法站 程序计数器

方法去和堆内存是线程共享的 其他的是线程私有的

回答堆内存 新生代？？？？

答： 新生代和老年代 ： 新生-eden 幸存者0 幸存者1

Eden存放刚创建的对象

MinorGC 存入幸存者0

进行下一次majorGC 从幸存者0移动到幸存者1

15次后转入老年代

Object header 4bit 最大为15位

老年满了后触发Full GC

1. 解释HashSet的hashCode重复的问题？如何存储重复的键

答： 底层是 哈希表 链表和红黑树

存入对象时先用hashcode计算地址值，发生哈希冲突使用equals方法判断对象内容是否一样。如果则去除重复元素，如果不一样则散列 ： 开放寻址法和链式寻址法

链表长度大于8 哈希表内容大于64将转为红黑树结构

还有栈寻址法和建立溢出缓冲区

谢谢！；-09

莫航

1. 自我介绍

不够严肃，介绍课程卡顿，含糊其辞

1. 外卖平台，数据存储，点菜信息，MySQL语言有哪些

回答比较完整

1. 数据库三个经典安全问题

脏读的产生原因不了解，回滚

1. 不可重复读？幻读？

还算可以

1. 存储引擎

偏题，innoDB优点不了解

6.InnoDB事务隔离级别，是否解决幻读

答错了是否可以解决幻读问题

1. ACID四个特性

可以但不流畅

1. 三色标记法

白色没有答出来。锦辉叹气，灰色黑色还可以

1. 类加载器

很容易答偏，答到其他区域

1. 双亲委派机制

还可以，Classnofound异常类型判断错误

1. 快速失败与安全失败

偏题（自己意识到），回答可以但是 不全面

张椿泊

1. 自我介绍

不够严肃，项目方向不足

1. get和post

不全面

1. 项目经历询问（登录）

很详细

1. 表锁与行锁

不对

1. 事务实现原理

不知道

1. 事务的隔离级别

惨烈

1. innoDB解决幻读问题

还可以

1. String底层存储数据用的什么

分版本回答没有答出来

1. 三个字符串类的区别

可行

10同步锁

区别答的有点少

1. Integer

答得基本可以

1. 雪花算法

不了解

1. Git

不熟悉

1. 三色标记算法

理解略有偏差

1. 哈夫曼树

可行，但是没有满足左大右小

**面试记录09/08**

面试记录

于枭森

c++和java的区别？

有无指针

跨平台原理

异常处理机制

面向对象

c++是否有多态回答错误

不知道c++和JAVA的继承的区别

JAVA类加载的过程？

三种类加载器

双亲委派机制

网络传输IO的同步，异步，阻塞，非阻塞？

基本回答正确

阻塞IO经典应用？

soket套接字（没答对）

线程池？

存储线程，资源复用，提前生成

不知道线程存储在队列中

不了解最大线程数，和CPU有关，核数+1

线程锁？

只知道同步锁

什么是线程通信？

不了解，多个线程互相影响，改变生命周期

改变生命周期主要是资源共享

反射的优缺点？

能在运行过程中获取类的信息

不符合面对对象思想

安全性低

可读性差

单例模式？

饿汉式

枚举饿汉式

懒汉式

DCL双重检验懒汉式（说错了DLC），没有提出volatile关键字

没有提出静态内部类懒汉式

进行单用户文档排序时，排序计算是放在数据库中驾还是后端中？

放在后端，减少数据库的内存消耗

多用户文档排序，使用什么？

引用第三方，使用分布式（存储）

对于计算方面没有想法

分布式服务器，从每个服务器中计算出最高分，再放到主服务上计算，分布式计算。

张椿泊

数据库三大范式？

这个…不了解

计算机操作系统进程和线程的区别？

线程是进程的子单位

主流计算机操作系统？

学过一点lunix

修改权限的指令？

忘记了chmod-r777

网络经典三大层？

网络，传输，应用（应用没答上）

JAVA应用于传输层

什么是CPU计数器？

不了解

虚拟机程序计数器？

针对单线程，记录程序执行位置

了解图吗？迪杰斯特拉算法？

不记得了

链表？

单双链表，插入删除效率高

很多电器，想用一种是设计模式实现后端和数据库构建？

额…没答上

采用抽象工厂

简单工厂和工厂模式？

回答全面

序列化的作用？

将对象写入硬盘，静态属性不被序列化，transient关键字也不会被序列化，防止对象序列化用静态

两者的区别？

静态属于类不属于对象，

为什么使用transient不用静态？

transient是忽略序列化，静态是不能被序列化，transient属性会被序列化，但会是默认值。

分布式，用户对20台服务器访问，防止访问集中？

根据服务器id分布访问，根据服务器容量进行服务器转换下一个服务器

循环顺序访问，

进行访问阻塞（流量削峰）

提示：随机数访问，权重分配方式

动态调整权重分配

jdk8中问什么引入新的时间类

防止时间可变，月份偏移，第二代没有类型转换

岳新亮

自我介绍

-- 小项目技术点？难点？

结账和退货有点棘手，

使用hashmap作为购物车

-- hashmap存储结构？？

底层是哈希表 链表 红黑树

原理是重写hashcode方法和equals方法判断对象是否重复

重复对象存储使用链式寻址发 一个bucket中树高度超过8将变为红黑树 数量超过64 也会变为红黑树。Hashmap扩容机制是新建一个hashmap，将原hashmap中内容依次插入新的hashmap中。造成死循环。。

-- 解释一下死循环？

扩容 -》 头插法扩容，导致元素顺序变为倒序。多线程并发状况下引发死循环问题。

-- 如何解决？

头插改尾插

-- 如何创建一个双向链表？

具体实现不清楚。。。

-- 了解并发哈希图吗？

是线程安全的

-- 并发类下 的方法都是线程安全的吗?

不清楚

-- 为何不用并发哈希图？

效率低

-- 并发哈希图如何存储空值？

不知道

-- 小项目中使用的数据库？主流数据库？

Mysql

关系型 oracle mysql sql server

非关系型 。。。

-- sql优化 策略？

查询时不适用\*

用索引查询

用小结果集驱动大结果集

不能在索引中使用函数

-- 索引下推 ，回表 ，索引覆盖》？

索引下推：

将服务层处理的判断过程转移到引擎层实现

在查询时，将筛选过程交给引擎层

索引回表：

查询时无法一次性查出所有结果，要回到原表中再次查询

索引覆盖：

使用索引查询时，尽量将查询的列和索引中的列对应。索引覆盖通过覆盖索引来实现。

-- 小项目中，实现文章的搜索，根据文章标题，tag或作者来搜索，在sql中如何实现最高效？？

减少回表次数 ，基于主键创建一个索引，将其他列和主键联合构成一个复合索引。

-- 全文查询》？

扫描整个表

-- where条件怎么写？

没思路。。

（用全文索引加like的模糊搜索。

一般不会放在mysql中实现这种）

-- 什么是全文索引？

不知

-- 索引分类？

聚集索引 - 主键索引

1. 普通索引
2. 唯一索引
3. 全文索引
4. 组合索引

-- 表没有设主键 有索引吗？

自动生成索引

莫航

白盒测试和黑盒测试？

白盒测试允许对代码调试，黑盒测试是在一个用例中进行场景调试

BIO，NIO，AIO？

BIO阻塞IO，无数据则阻塞

NIO非阻塞IO轮询请求

AIO异步IO，由内核进行数据处理，进程可以进行其他工作

代理模式？

访问控制，

适配器模式，解决接口不兼容

数据库的版本？

5.8

存储引擎？

innoDB

存储结构?

B+ 树，

不知道innoDB对B+树的优化

将单向链表变成双向链表

InnoDB解决并发问题？

多版本并发控制，版本号，间隙锁，快照读

事务的底层原理？

日志系统，通过日志进行事务回滚

可重复读和可串行化区别？

事务隔离界别3和4

3不能解决幻读，4效率低，但3个问题都能解决

4多了锁，事务不提交其他事务无法访问

介绍下视图？

表的快照，，一般视图变化不影响原表，某些特殊情况会影响

在MVCC快照读，当前读行锁和间隙锁解决幻读

事务底层，日志，MVCC

在多台服务器中分布式服务id？

雪花算法，

算法有什么问题？

服务宕机会导致生成重复id

生成重复id抛出异常，但会造成损失

引入日志系统，记录之前机器码最大值，在此基础上递增

对机器码划分为3位时间序列和7位机器码，但会降低服务器并发数量

Arraylist和linkedlist的区别

1. 效率，arraylist是数组，随机访问效率高

Linkedlist访问效率满，插入和删除效率高

Arraylist扩容是数组复制，内容扩大1.5倍

存储内存，arrlist连续，linked不连续

进行尾部插入是arrlist效率更好，