

## 1. elasticsearch为什么比mysql快?

a. 数据库查询 like语句: mysql在使用like查询的时候只有使用后面的%时, 才会使用到索引。

索引: 在关系数据库中, 索引是一种单独的、物理的对数据库表中一列或多列的值进行排序的一种存储结构, 它是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单。索引的作用相当于图书的目录, 可以根据目录中的页码快速找到所需的内容。

b. elasticsearch为什么比mysql快?

<https://www.cnblogs.com/dasha/p/12450153.html>

## 2. windows环境下elasticsearch安装

### 教程

网址 :

<https://www.cnblogs.com/hualess/p/11540477.html>

### **ElasticSearch**

启动: 进入bin目录下, 双击执

elasticsearch.bat ( E:\软件备份

\elasticsearch\elasticsearch-7.6.0-windows-

x86\_64\elasticsearch-7.6.0\bin )

打开 : <http://localhost:9200/>

← → ↻ ⓘ localhost:9200

应用 百度 工作 文章 已导入 工具 GitHub gitee 技术

```
{
  "name" : "ADMIN-PC",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "ZdPWwyAuTYuWjVn9gO9B4Q",
  "version" : {
    "number" : "7.6.0",
    "build_flavor" : "default",
    "build_type" : "zip",
    "build_hash" : "7f634e9f44834fbc12724506cc1da681b0c3b1e3",
    "build_date" : "2020-02-06T00:09:00.449973Z",
    "build_snapshot" : false,
    "lucene_version" : "8.4.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

## ElasticSearch-head插件

启动：进入head安装目录（npm run start）

打开：<http://localhost:9100/>

← → ↻ ⓘ localhost:9100

应用 百度 工作 文章 已导入 工具 GitHub gitee 技术 官方网站

**Elasticsearch** <http://localhost:9200/> [连接](#) **elasticsearch** 集群健康: green (0 of 0)

概览 索引 数据浏览 基本查询 [\[+\]](#) 复合查询 [\[+\]](#)

数据浏览

所有索引 ▾ 查询 0 个分片中用的 0 个, 0 命中, 耗时 0.109 秒

	_index	_type	_id	_score ▲
--	--------	-------	-----	----------

索引  
类型  
字段

## 基础概念

节点 Node、集群 Cluster 和分片 Shards

ElasticSearch 是分布式数据库，允许多台服务器协同工作，每台服务器可以运行多个实例。单个实例称为一个节点（node），一组节点构成一个集群（cluster）。分片是底层的工作单元，文档保存在分片内，分片又被分配到集群内的各个节点里，每个分片仅保存全部数据的一部分。

索引 Index、类型 Type 和文档 Document

对比我们比较熟悉的 MySQL 数据库：

index → db

type → table

document → row

如果我们要访问一个文档元数据应该包括囊括 index/type/id 这三种类型，很好理解。

## ik分词器

**简介：**在对文档(Document)中的内容进行索引前，需要对域(Field)中的内容使用分析对象(分词器)进行分词。

**原理：**基于正向匹配的分词算法。

a. 默认的中文分词是将每个字看成一个词，显然不满足需求

b. ik分词器两种模式

基本可分为两种模式，一种为smart模式（ik\_smart），一种为非smart模式（ik\_max\_word）

非smart模式所做的就是将能够分出来的词全部输出；smart模式下，IK分词器则会根据内在方法输出一个认为最合理的分词结果

例如：张三说的确实在理

smart模式的下分词结果为：

张三 | 说的 | 确实 | 在理

而非smart模式下的分词结果为：

张三 | 三 | 说的 | 的确 | 的 | 确实 | 实在 | 在理

坑：

1. 一个index下只能有一个type；（跟版本挂钩）

2. index名不能大写；

3. 实体类@Mapping(mappingPath = "newsIndex.json") // 解决Ik分词不能使用问题

"analyzer": "ik\_max\_word", 插入文档时，将text类型的字段做分词然后插入倒排索引，

"search\_analyzer": "ik\_max\_word" 询时，先对要查询的text类型的输入做分词，再去倒排索引搜索

这个注解在写入跟查询时要同时存在，不能在写入时没加，查询时加会报错