# 分布式搜索-ElasticSearch基础

## 前言

前面的微服务学习过程中，提到过，随着业务的发展，数据量越来越庞大，传统的Mysql数据库就渐渐地无法满足复杂的业务需求，所以，在微服务架构下，一般都会用到一种叫做分布式搜索的技术，今天就来学习最流行的分布式搜索技术：ElasticSearch

### ElasticSearch基础

初始ElasticSearch，了解ES;

安装ElasticSearch；

使用ElasticSearch：对索引库的操作，索引库就相当于Mysql的数据库；

文档操作：我们把数据导入索引库里，导入数据就是所谓的文档，，我们要实现对文档的CRUD；

RestAPI：最后学习ElasticSearch官方提供给我们的一套Restful风格的API，也就是所谓的JAVA客户端，来帮助我们更方便地操作ElasticSearch。

## 初始ElasticSearch

### 了解ES（ES就是对ElasticSearch的缩写、简称）

#### ES介绍

ES是一款非常强大的**开源的搜索引擎**，可以帮助我们从海量数据中快速找到需要的内容（关键字搜索、高亮显示、模糊搜索、距离）。

关键字搜索：github中搜索想要的代码程序仓库；

高亮显示：京东搜索想要的商品；

模糊搜索：百度搜索出错日志；

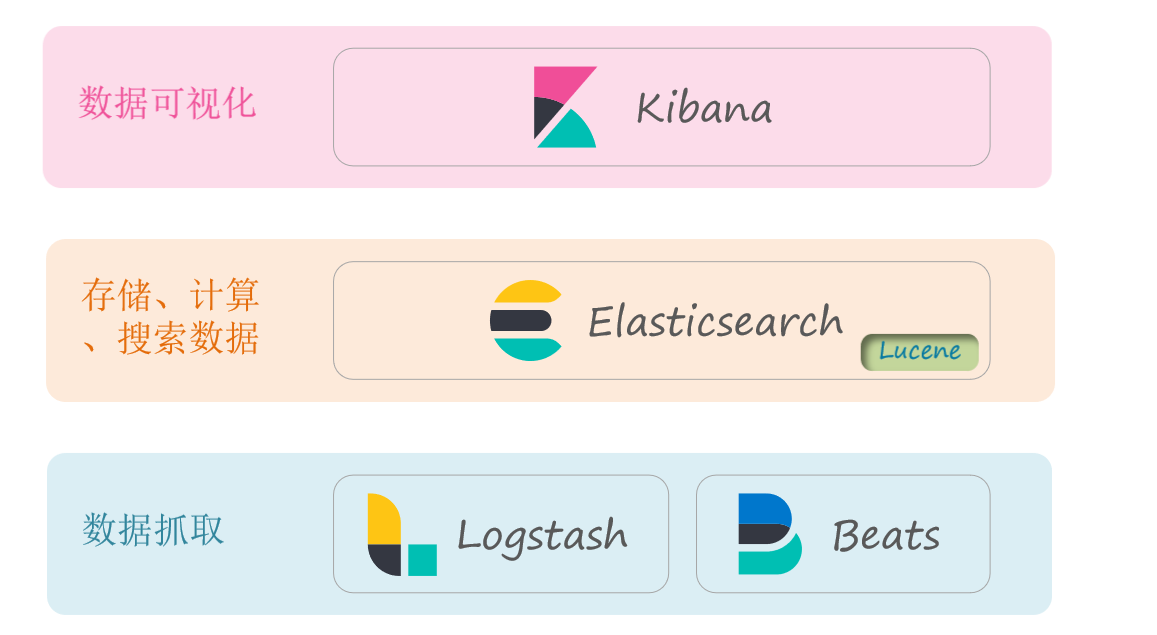
距离：美团骑车上显示附近可用单车；

。。。。。。

#### ELK Stack-Elastic技术栈

ELK = ElasticSearch + Logstash + Kibana/Beats，ElasticSearch结合kibana、Logstash、Beats，就是我们所说的elastic stack（ELK），即Elastic技术栈。被**广泛应用在微服务日志数据收集、分析、实时监控等领域**。

ELK就是Elastic技术栈Elastic Stack！包含ElasticSearch、Logstash、Kibana/Beats等技术，这套技术栈广泛应用在微服务日志数据收集、分析，和实时监控领域！



什么是日志数据收集、分析？就是说项目在运行的时候，会产生还来给你的日志信息，相信你也看过，这些日志信息其实就是用来方便我们定位系统所出现的问题！比如系统报错了，你怎么去找到错误所在呢？系统线上运行着，你是不可能去打断点debug啊，一般都是采用日志分析的方式，而ElasticSearch可以将日志数据可视化给你展示出来，所以你将来去做日志分析的时候就非常方便

实时监控呢？项目在运行过程中，项目的运行状态也是数据！项目节每一刻都会有状态，那么项目每一刻都有数据产生，比如CPU情况、内存情况、访问频率等等，这些信息也会被ElasticSearch收集、管理起来，将来也以可视化的形式给你展示出来，你就可以非常清晰地知道整个项目的运行情况了，这就叫实时监控！

以上，你可以看到，我们学习的搜索引擎ElasticSearch使用的场景非常广泛！

ELK技术栈里尽管有好多个组件（ElasticSearch、Logstash、Kibana/Beats），但核心就是ElasticSearch，负责数据的存储、计算、搜索、分析！而Logstash和Beats这两个组件组要是用来做数据抓取的，比如说你的日志数据就可以用这两个组件来抓取，而Kibana是一个数据可视化组件，也就是说将来你搜索出来的数据要展示，可以用kibana来展示，形成报表！但是数据可视化组件并不只有Kibana，很多大公司都有自己的数据可视化技术组件，所以数据可视化并不一定要用Kibana，完全可以自己去实现！数据抓取也一样，完全可以自己写JAVA代码，自己弄数据，然后往ElasticSearch去写！所以Kibana、Logstash、Beats都是可替换的组件，官方提供给你，你想用就用，不想用完全可以自己实现！但是不可替代的就是ElasticSearch这个核心！

#### Lucene：ES的底层实现

ElasticSearch的底层实现用到了Lucene。Lucene是一个Java语言的搜索引擎类库（简单来说就是jar包），是Apache公司的顶级项目。官网：

<https://www.elastic.co/cn/>

Lucene优点：

1：**易扩展**，只是实现了搜索引擎相关核心api，可以基于此做二次开放、定制，实现更高级的功能；

2：**高性能：**lucene基于倒排索引算法实现，查询性能相当优秀。

Lucene的缺点：

1：因为是java语言的类库，只限于java语言开发；

2：api设计得很复杂，学习曲线陡峭，一般人想基于lucene做搜索引擎开发时比较难的；

3：不支持高并发、集群扩展，这个类库只考虑到了如何去实现搜索，因此没有面对高并发的的场景，也不支持集群扩展的东西，要实现这些并须去做进一步的开发，做二次开发才行。

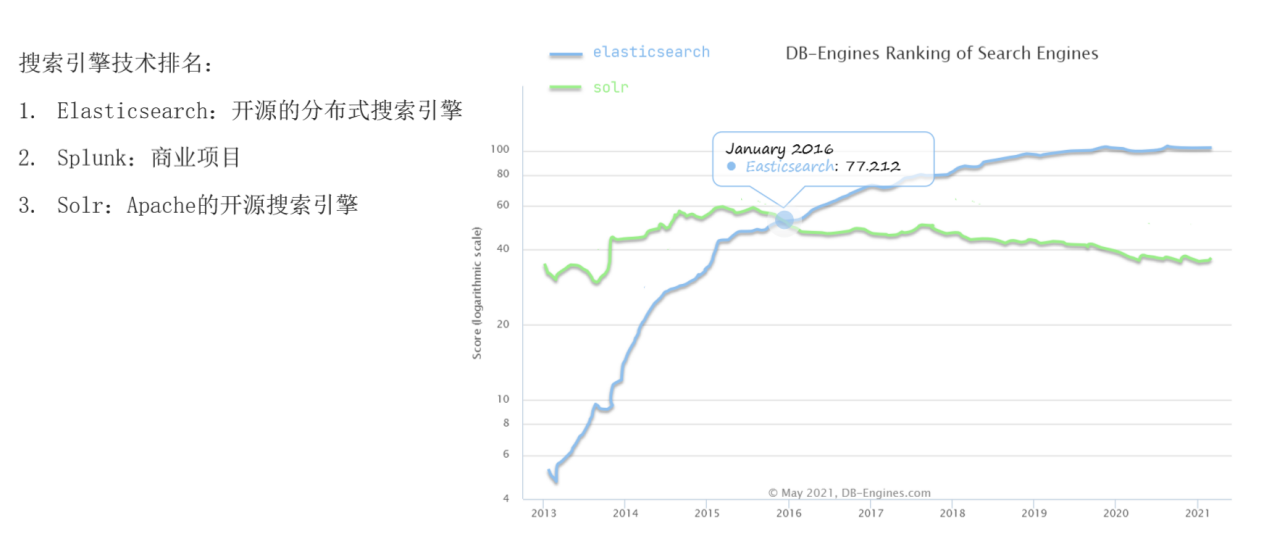
#### ElasticSearch：基于Lunece，对Lunece做了二次开发

除了具有Lunece的优点之外，还弥补了Lunece的缺点：

1：**天生就支持分布式，可水平扩展**（ES集群搭建非常简单，ES天生就是支持集群，所以支持海量数据和高并发场景）；

2：**提供Restful接口，跟语言无关**，ES对外暴露的是Restful接口，可被任何语言调用，使用场景更加广泛。

注意了ElasticSearch并不是唯一的搜索引擎技术，但是为什么ES的使用量这个高呢？因为使用量最高！所以使用多！



#### 小总结

什么是ElasticSearch?

一个开源的分布式搜索引擎，可以用来实现搜索、日志统计、分析、系统监控等功能

什么是elastic stack（ELGK）

是以ElasticSearch为核心的技术栈，包括beats、logstash、kibana、elasticsearch

什么是Lunece？

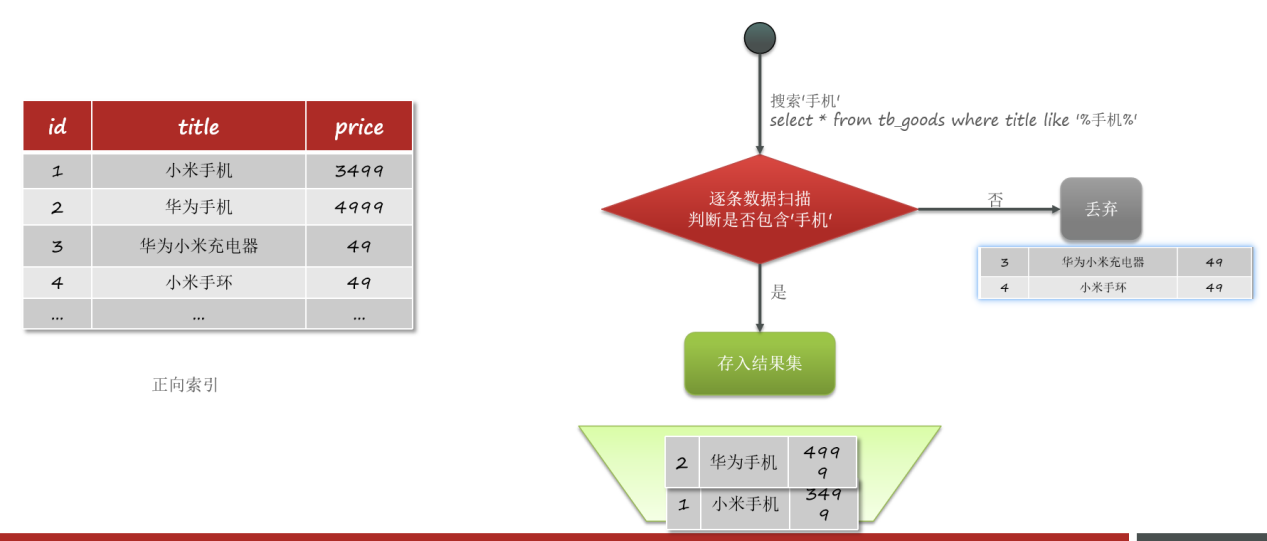
是Apache的开源的基于java的搜索引擎类库，提供了搜索引擎的核心api。

### 倒排索引-ElasticSearch的核心技术

在上节课我们知道了ES底层是基于Lunece来实现的，我们知道Lunece的核心技术就是倒排索引！倒排索引时与传统的数据库（如Mysql）的正向索引去对比得出的名称，可见代拍索引一定是与数据的正向索引有比较大的差异：

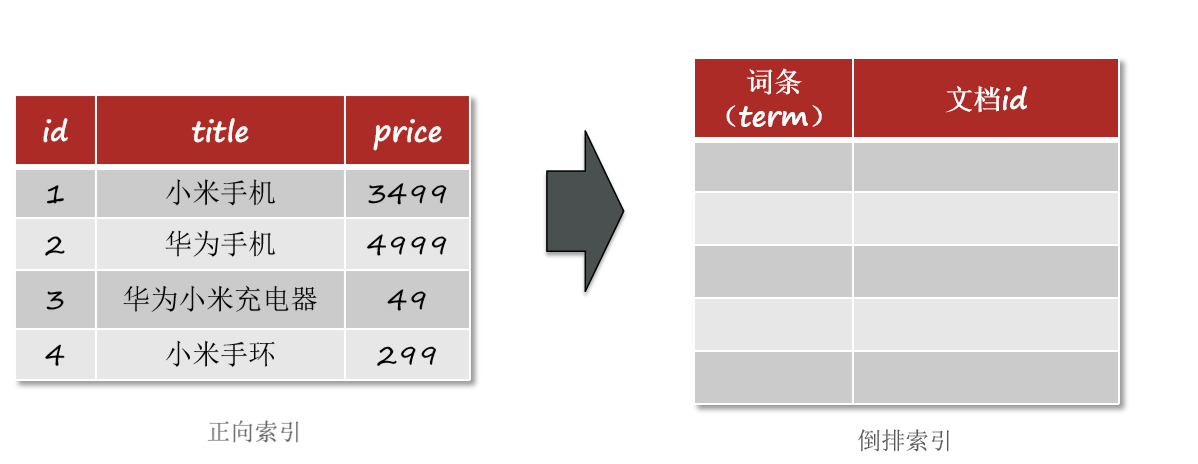
#### 正向索引

传统数据库（如mysql）采用的正向索引，例如常见的很多表中的id字段创建索引，形成一颗B+树，那么根据id检索的速度就会变得非常快。这种方式的索引就是正向索引，但是如果搜索的字段不是id，而是一个普通字段，如下图商品表的标题title字段，标题字段一般比较长，你一般不会给标题字段加索引，因为普通字段一般不会加索引，即便加了索引，但是如果搜索的不是精确的字段值，搜索的是部分字段值，那么就得使用模糊匹配（like），而**一但使用了like模糊匹配，就算字段有索引也不生效**，这种情况下（没有索引），就会采用逐条扫描的方式来判断表中每一行是否满足，不包含则丢弃，包含则把记录放入结果集。此时如果表的数据过多（如1000万条），那么就要扫描1000万次，全表扫描，性能可想而知地差。**正向索引做这种字段的局部内容检索的时候效率比较差！**



#### 倒排索引：

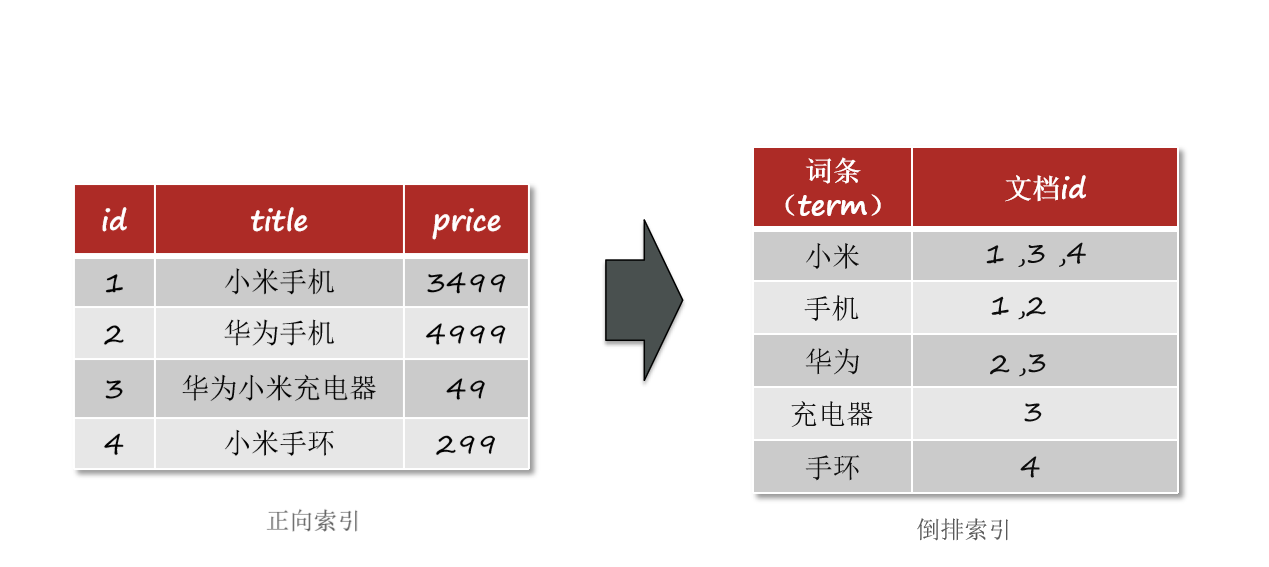
倒排索引创建的时候会形成一张新的表（倒排索引表），这张表里面有两个字段，一个字段是词条term，一个字段是文档id。



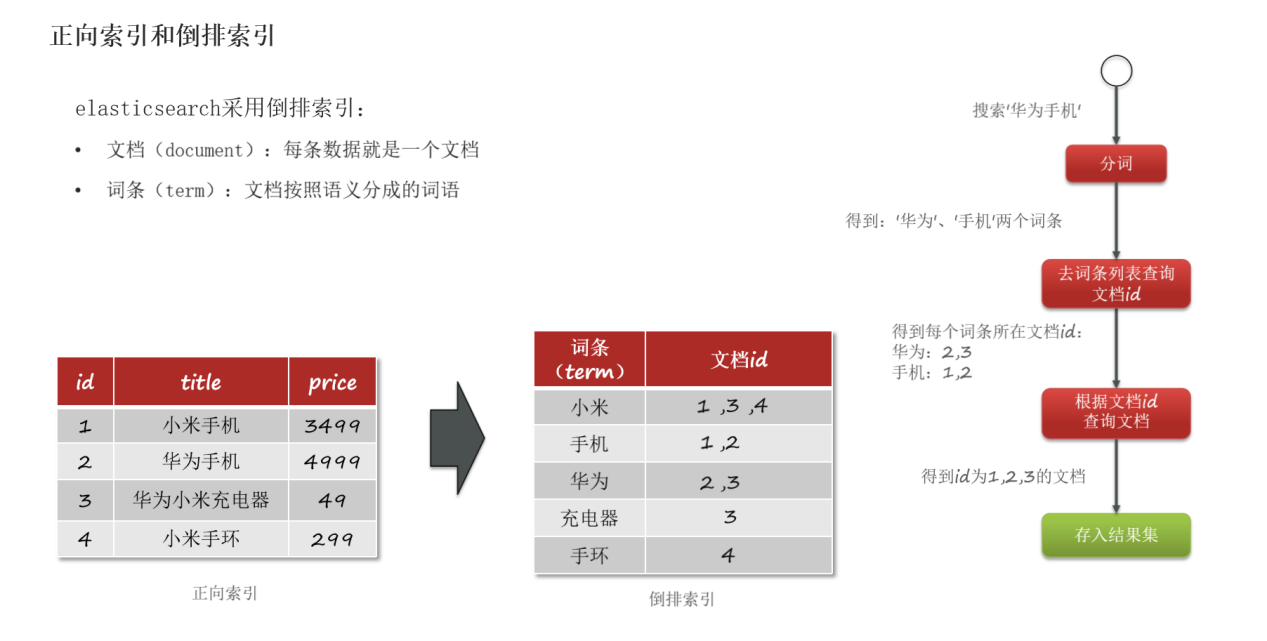
文档即document，每一条数据就是一个文档，参照数据库表中的一条记录，比方说商品表中每一条数据就是一个文档，订单表每一个订单就是一个文档，用户表每一个用户就是一个文档！

词条term：一个文档中有很多很多数据，文档内容按照语义分成的词语（重点是词语），比如如果是中文就按中文含义分词，如：华为小米充电器，可以分成（华为、小米、充电器、充电、电器），分出来得每个词语就是一个词条。

也就是说**倒排索引在存储是会先把文档中的内容分成词条去存**，比如说一个文档中包含标题title（华为小米充电器），要对标题title创建倒排索引，那么就要把标题内容做分词，得到五个词语：华为、小米、充电器、充电、电器，这个时候就把分词得到的每个词条存到倒排索引表中的term，然后记录对应的文档id。有多少个词语就存多个词条；如果新的文档分词后有想相同的词语，因为词条不能重复，那么就在对应词条的再加上文档的id，以此类推。所以不管是有四个文档，还是有四千万个文档，将来一定都可以分成一个一个的词条，并且这些词条肯定会有大量的重复，但是我们不用重复记录相同词条，只需要记录一个，并同时记录重复文档id即可，这样可以确保倒排索引当中词条term字段是绝对不会出现重复的，因此，因为**词条的唯一性，我们就可以给词条term创建索引了**，将来根据词条查找文档的速度就相当快了。



具体怎么查找呢？比如我们搜索“华为手机”，第一步会把“华为手机”进行分词，得到“华为”、“手机”两个词条，然后去倒排索引表中进行查询，因为倒排索引表中根据词条建立了索引，所以根据“华为”、“手机”这两个词条去查询的速度是很快的，然后查找这两个词条对应的文档id，于是就会得到两组文档id，这个时候就会知道包含“华为”，“手机”词条的所有文档了，将来还可以根据文档id出现的频率进行排序，然后根据文档id就可以去正向索引表查询文档，因为正向索引表根据id建立了索引，那么就可以快速地定位到对应的文档。



综上，可以发现，**倒排索引的过程其实是经过两次检索，第一次是根据用户输入内容的词条去词条列表中找到对应的文档id；第二次是根据文档id去找文档，虽然是两次，但每一次的检索都是通过索引去找的，所以搜索效率是很高的，是比全表扫描的效率要高得多得多的**。同时也能看出来了，倒排索引之所以叫倒排索引，因为在正向索引中是逐行去查找，是先找到文档，看文档是否符合词条要求；而倒排索引是反过来的，是基于词条去创建索引，然后去关联到对应的文档，查找的时候是先去找到对应的词条，再根据词条找到对应的文档，即倒排索引是根据词找文档，而正向是是根据文档匹配词。倒排索引适合、擅长基于文档的部分内容进行搜索，即更复杂的搜索需求，比如浏览器关键字搜索，商城商品搜索，都是复杂的搜索场景。

### 注意点（很重要，真的很重要）

一个文档库包含多个文档，一个文档库就看作是数据库中的一张表，

一个文档就看做是主句库中的一条记录，对文档库建立倒排索引就是对要分词的文档字段建立倒排索引表，比如对商品文档的商品名字段分词，就建立一张商品名的倒排索引表，对商品描述字段分此就建立一张商品描述倒排索引表，用户搜索的时候就先来要到倒排索引表查找到文档id，再根据文档id到文档库中获取文档！

创建索引（数据库中的表），所索引里添加一条一条的文档，在往索引加文档的时候，根据隐射里对字段的分词规则，建立文档字段的倒排索引库。

### 小总结：

1：什么是文档和词条：

每一条数据就是一个文档；

对文档中的内容分词，得到的词语就是词条

2：什么是正向索引：

基于文档id创建索引，查询词条时必须先找到文档，而后判断是否包含词条

3：什么是倒排索引：

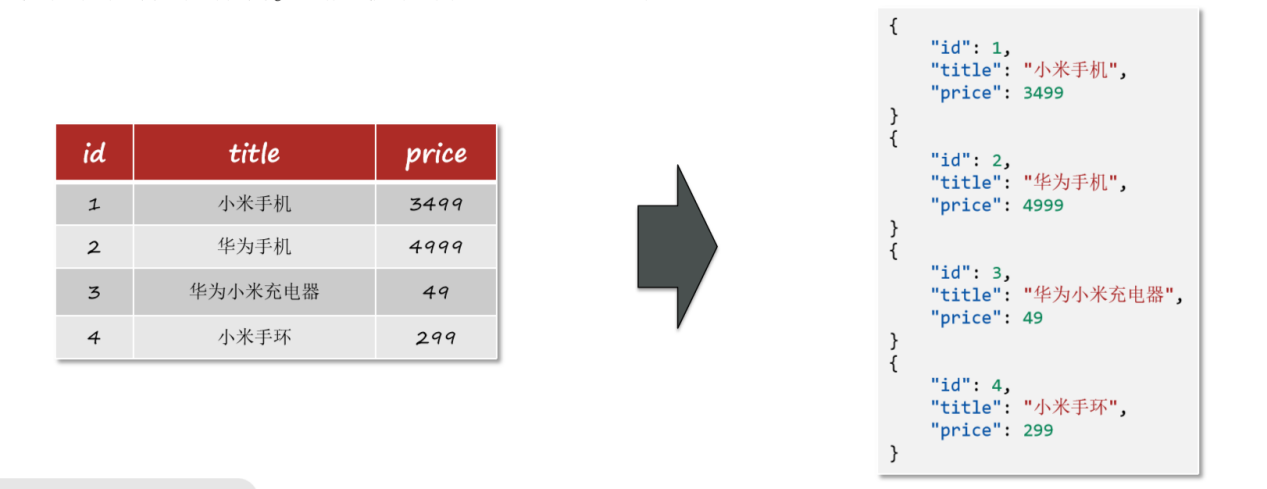
对文档内容分词，对词条创建索引，并记录词条所在文档的信息，查询时先根据词条查询到文档id，而后获取到文档

## ES的一些概念

了解一下ES当中的概念，为了更好地掌握，可以和Mysql中的概念做对比！

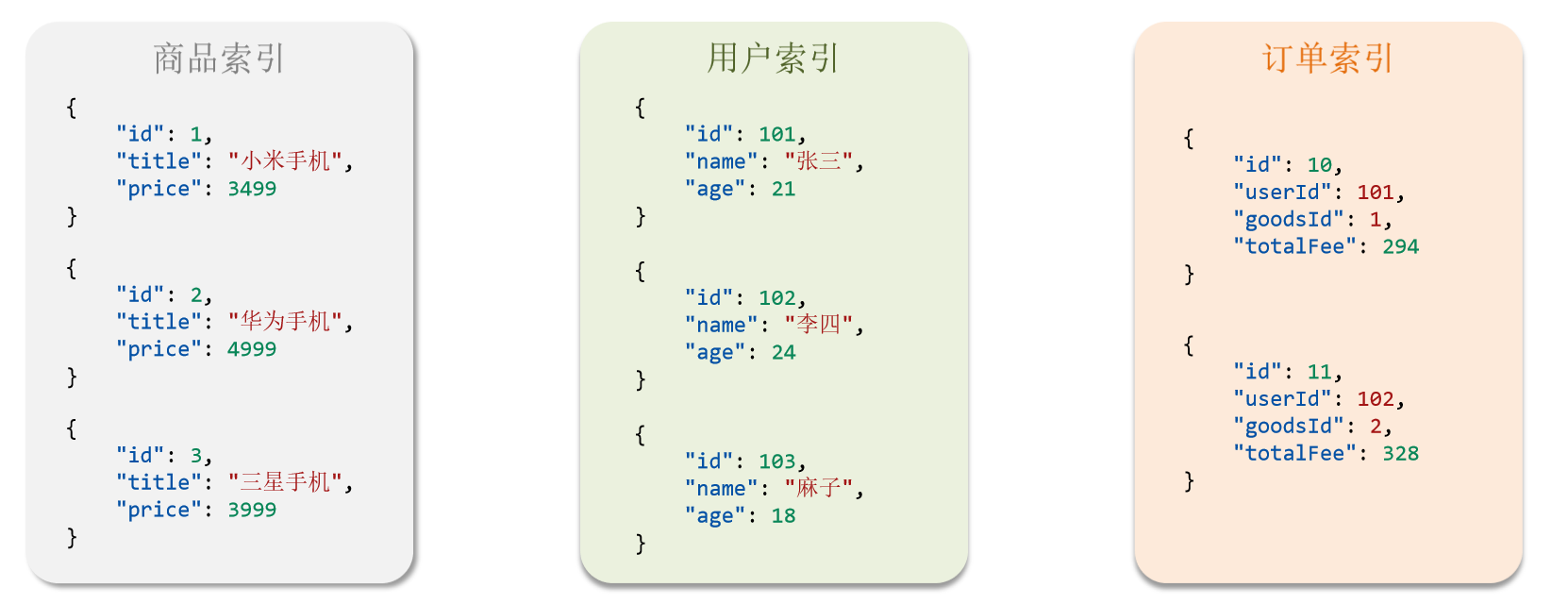
### 文档Document

ElasticSearch是面向文档存储的，可以是数据库中的一条商品数据，一个订单信息，这些东西都可以转化成文档存储在ES中。ES文档数据会被序列化为JSON格式后存储在elasticsearch中。也就是说商品数据也好，订单数据也好，如果要存储在ES中形成文档，一定会按照JSON格式，这是ES与数据库存储比较大的差异，格式上的差异。类似与数据库中表的行（Row），即一条记录。



### 索引Index

索引就是相同类型文档的集合，在ES中，文档肯定不会只有一条啊，开玩笑，ES是海量数据的搜索欸，会有非常非常多的文档，这些文档的结构是有差异的，索引就是相同类型的集合，类似于数据库中表的概念！



### 隐射Mapping：

索引中文档的字段约束信息（即字段名字叫什么、数据类型是什么。。），类似数据库中表的结构约束Schema。



### 字段Field：

就是Json文档中的字段，类型数据库中表的列（Column）

### DSL（domain specific language）：

DSL（domain specific language）就是操作ES的语句，dsl是ES提供的**JSON风格的请求语句**，用来操作elasticsearch，实现CRUD，所以dsl和sql是一样的，就是一种语法，这种语法就是用来描述要干的事，在Mysql中，sql写好了通过Connection把语句发给MySQL，MySQL就能进行相应的处理，在ElasticSearch中，写好DSL后，要怎么发出去呢，是通过基于HTTP请求发出去的，因为ES对外暴露的Restful的接口。这种接口的好处就是跟语言午关，也就是说任何的语言只要你能发HTTP请求，你都可以把DSL语句发给ES来操作ES。彻底脱离语言的束缚！

### ES概念和Mysql概念对比



所以将来操作ES的时候，和操作数据库相同的道理，第一件事是先去创建索引库，即索引Index，因为创建索引，那么就要指定约束映射Mapping，用来约束索引中的字段“长什么样子”，然后添加文档Document。然后是文档的CRUD。在ES中，写好DSL通过HTTP的请求发送过去给ES。

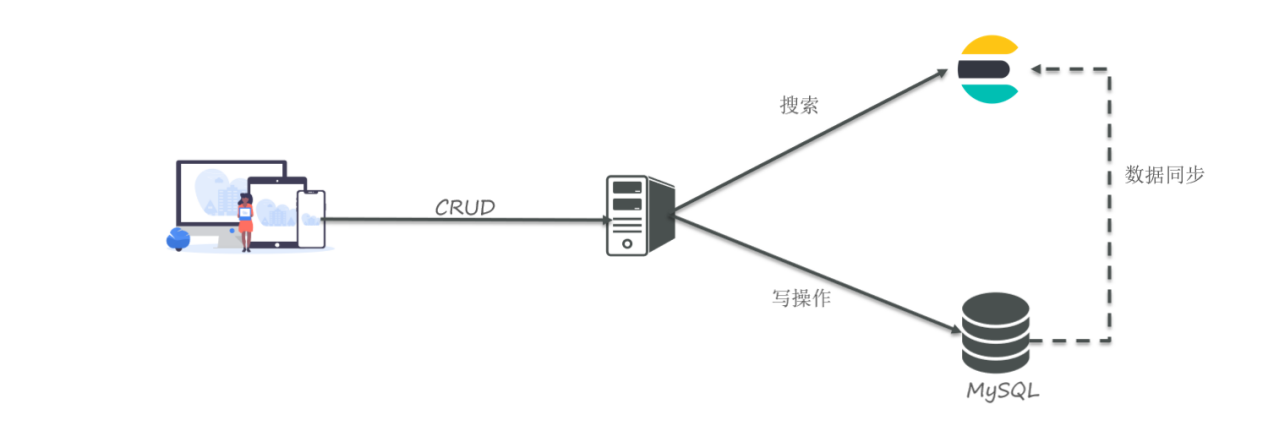
### Mysql和ES怎么选？

从上面的概念对比，是不是有了ES之后就可以完全替代MySql呢？并不是这样！因为这两种技术擅长的东西是不一样的！

Mysql：擅长事务类型操作，因为事务有ACID原则，可以确保数据的安全和一致性；

ES：ES没有事务的概念！无法保护ACID，擅长海量数据的搜索、分析、计算。

所以两者是各司其职的，如果你现在的业务对数据的安全性、一致性要求很高，事务要求很高，如下单付款，你就应该用Mysql去做数据存储；但是现在你做的是商品的搜索，页面的搜索，搜索比较复杂，你肯定是使用ES去做！所以，ES和Mysql两者是互补、不是替代关系！因此在我们的实际工作中、实际系统架构中，两者都会存在！



用户来做数据的CRUD操作，操作请求访问到服务器，服务器就可以做盘多，用户的请求是增删改的写操作吗？是的话就写到数据库，但是用户的请求是查询读操作，那肯定是去ES中去做搜索，因为ES快！这也就意味着ES、Mysql中都得有数据，那怎么确保两个都有数据呢？一般在Mysql写数据，但是Mysql可以基于某种方式，把数据同步给ES，从而实现数据双写！将来读数据，比如复杂搜素就从ES中读取。

### 小总结

文档：一条数据就是一个文档，es中是Json格式

字段：json文档中的字段

索引：同类文档的集合

映射：索引中文档的约束，就是读字段的约束，比如字段名称、类型

ES和数据库的关系：

数据库负责事务类型操作；

ES负责海量数据的搜索、分析、计算！