lucene原理

一、教学目标

1、什么是lucene

2、lucene的使用场景

3、索引的算法

4、lucene的原理

5、备注

二、什么是lucene

```
1. lucene就是apache下的一个全文检索工具，一堆的jar包，我们可以使用lucene做一个谷歌和百度一样的搜索引擎系统。
2. Lucene是有Doug Cutting 2000年时开发出的第一个版本，后捐献给apache基金会，doug cutting是Lucene、Hadoop（大数据领域的）等项目的发起人。
3. lucene(原理)--jdbc  solr--mybatis，elasticsearch(es)
```

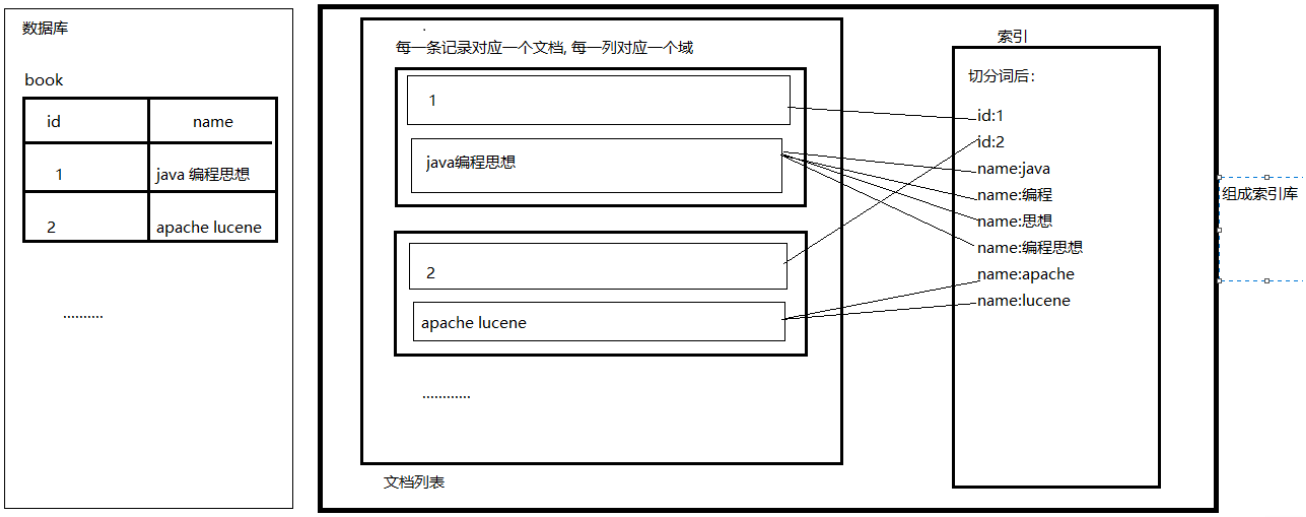
三、lucene的使用场景

```
互联网搜索：百度，必应，谷歌
站内搜索：淘宝，京东，贴吧
```

四、常见的算法

```
顺序扫描法
   描述：拿着关键字逐字匹配，一条一条的比较，直到找到为止
   举例：数据库：like查询
   缺点：慢，效率低，会随着内容的增长速度明显降低
   优点：准确率高
全文检索算法（倒排索引算法）
   描述：将数据库所有的数据查询出来，进行切分词，组成索引，把正文部分组成文档，把索引和文档组成索引库(就可以电脑上的任意一个文件夹)，查找的时候，先找到索引，索引与文档之间有联系，通过索引能快速找到文档，返回文档,这就是全文检索算法
   举例：字典,找到目录中的部首，找到大概位置
   缺点：以空间换时间，索引库占用大量的空间
   优点：效率高,准确率高
```

五、lucene的原理



六、代码实现

1、引入依赖

```
<dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.9</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.6</version>
        </dependency>
      <!-- ik中文分词器 -->
        <dependency>
            <groupId>com.janeluo</groupId>
            <artifactId>ikanalyzer</artifactId>
            <version>2012_u6</version>
        </dependency>
        <dependency>
            <groupId>org.apache.lucene</groupId>
            <artifactId>lucene-analyzers-common</artifactId>
            <version>4.10.3</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.apache.lucene/lucene-core -->
        <dependency>
            <groupId>org.apache.lucene</groupId>
            <artifactId>lucene-core</artifactId>
            <version>4.10.3</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.apache.lucene/lucene-queryparser -->
        <dependency>
            <groupId>org.apache.lucene</groupId>
            <artifactId>lucene-queryparser</artifactId>

            <version>4.10.3</version>
```

```
        </dependency>
    </dependencies>
```

2、数据准备

```

```

3、创建索引

```

```

4、界面查看测试索引

```

```

5、使用索引查询

```

```

6、删除索引

```

```

7、更新索引

```

```

七、中文分词器

```
1. lucene的分词器对中文不友好，一个字就是一个词
2. 中文分词器：可以分析中文语法,一个词就是一个词
   指定词元：传智播客
   停止词元：编程,分词后就不是一个词,
3. IK中文分词器
```

六、Field域的类型

Field类型	数据类型	Analyzed是否分词	Indexed是否索引	Stored是否存储
StringField(FieldName, FieldValue, Store.YES))	字符串	N	Y	Y or N
LongField(FieldName, FieldValue,Store.YES)	Long型	Y	Y	Y or N
StoredField(FieldName, FieldValue)	重载方法，支持多种类型	N	N	Y
TextField(FieldName, FieldValue, Store.NO)或TextField(FieldName, reader)	字符串或流	Y	Y	Y or N

七、备注（名词解释）

1．切分词：把内容中的不重要的内容去掉，留下重要的词，去掉：的，得 ，地，a，an ，the ,空格等等,转换成小写
2．索引：就是为了查询
3．文档：就是包含了内容,正文部分（数据库中的内容）
4．索引库：索引+文档
5．域的类型
特殊的定义：如果要使用范围进行检索，必须分词，必须索引
是否分词：分词的目的：索引，分词后是否有意义
是：分词后意义
举例：name，description,price
否：分词后没有意义
举例：id， pic
是否索引：查询使用的，需要索引
是：查询的需要使用
举例：id,price,name,description
否:查询的不需要使用
举例：pic
是否存储：存储到索引库中，需要在查询页面展示的，就需要存储
是：需要查询页面中存在
举例：id，price，name ,pic
否：不需要在查询页面中存在
举例：description
描述域：内容量较大，一般不存储，如果需要使用描述中的内容，那么可以根据id从数据库查询该描述信息，id本身就是一个主键，索引类