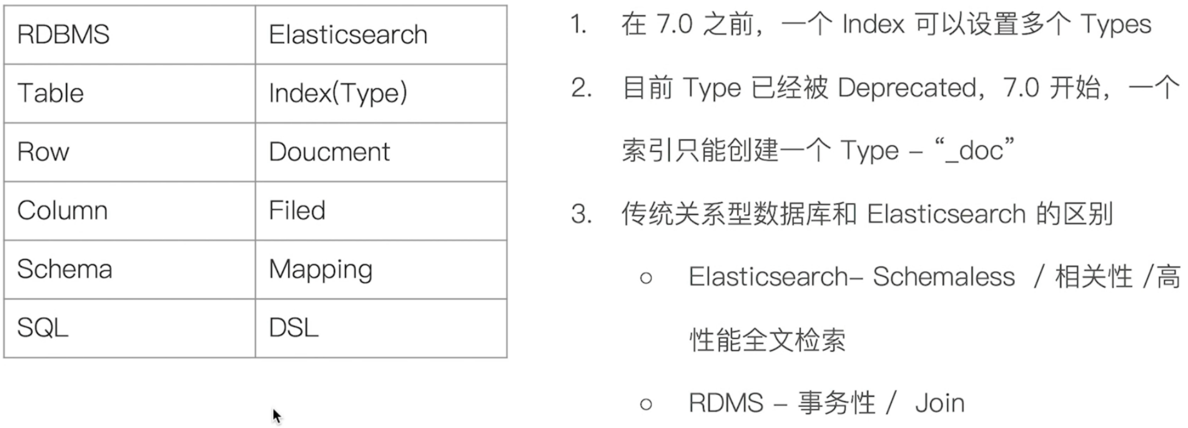
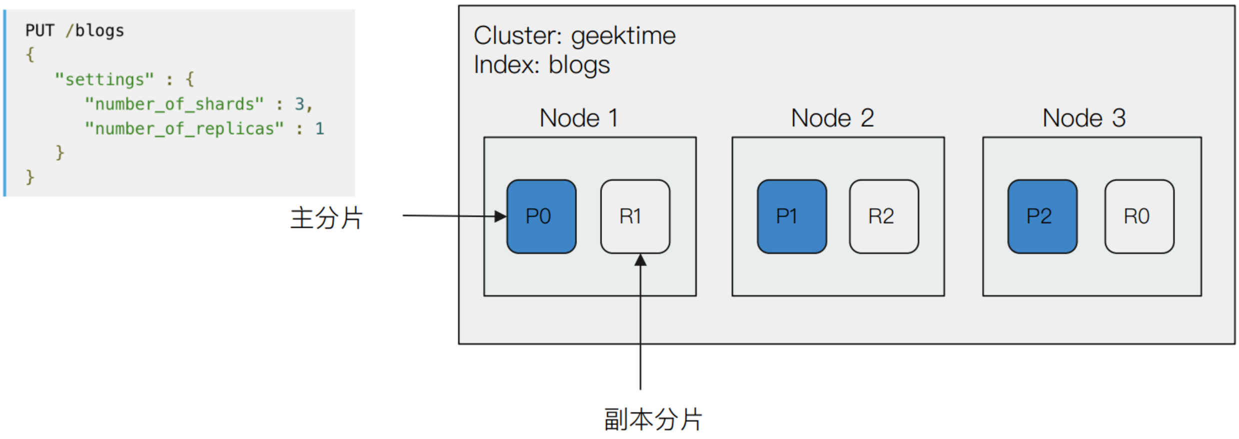


RDBMS	Elasticsearch
Table	Index(Type)
Row	Doucment
Column	Filed
Schema	Mapping
SQL	DSL

1. 在 7.0 之前，一个 Index 可以设置多个 Types
2. 目前 Type 已经被 Deprecated，7.0 开始，一个索引只能创建一个 Type – “_doc”
3. 传统关系型数据库和 Elasticsearch 的区别
   - Elasticsearch– Schemaless / 相关性 /高性能全文检索
   - RDMS – 事务性 / Join

- 高可用
  - 服务可用性 - 允许有节点停止服务
  - 数据可用性 - 部分节点丢失，不丢失数据
- 可扩展性
  - 请求量提升或者数据的不断增长，系统能够将数据分不到所有节点上，水平扩容
- es 的分布式好处
  - 存储水平扩容
  - 提高系统可用性，部分节点宕机，不影响集群的服务
- 节点是一个 es 实例 运行在 jvm 中的一个java 进程
  - master-eligible 能够参与选举成为 master
  - master node 只有master 才能够修改集群信息，如 mapping settings
  - data node 存储数据的节点
  - coordinating node 请求分发和结果汇总，每个节点都能起到 coordinating node的职责
- document
  - 日志文件中的日志
  - 一部电影的具体信息
  - 一首歌曲 、一篇pdf 具体内容
- index 一类文档的集合
  - 每个索引都有自己的 mapping 定义，定义文档的字段名和字段类型
  - mapping 定义文档字段类型
  - setting 定义不同的数据分布
- 分片



```
PUT /blogs
{
   "settings" : {
      "number_of_shards" : 3,
      "number_of_replicas" : 1
   }
}
```

倒排索引

- 单词词典（Term Dict），记录所有文档的单词，以及单词到倒排列表的关联关系
- 倒排列表（Posting List），记录单词对应的文档，由倒排索引项组成
    - 倒排索引项（Posting）
        - 文档 ID
        - 词频 - 单词在文档中出现的次数
        - 位置（Position）
        - 偏移（offset），记录开始结束位置

```
put product
get product
delete product
head product
get /_cat/indices?v
get _cat/nodes?v

put product/_doc/10003
{
  "name": "zhang"
}

post product/_update/10004
{
  "doc": {
    "price": 3999
  }
}

get product/_search?q=name:zhang
get product/_search
{
  "query": {
    "match": {
      "name": "zhang"
    }
  }
}
get product/_search
{
  "query": {
    "match_all": {}
  }
}

# term 不允许分词
{
  "query": {
    "term": {
      "name": {
        "value": "zhangsan"
      }
    }
  }
}
# 多词查询
{
  "query": {
    "terms": {
      "name": {
        "value": [
          "zhangsan","wangwu"
        ]
      }
    }
  }
}

{
  "_source": [
    "name","nickname"
  ],
  "query": {
    "match": {
      "name": "zhang"
    }
}

{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "name": "zhangsan"
          }
        }
      ],
      "must_not": [
        {
          "match": {
            "age": "40"
          }
        }
      ],
      "should": [
        {
          "match": {
            "sex": "男"
          }
        }
      ]
    }
  }
```

```
}

{
  "query": {
    "range": {
      "age": {
        "gte": 30,
        "lte": 35
      }
    }
  }
}

get product/_search
{
  "query": {
    "match_all": {}
  },
  "from": 0,
  "size": 4,
  "_source": ["name"],
  "sort": {
    "name": {
      "order": "desc"
    }
  }
}

{
  "query": {
    "match": {
      "name": "zhangsan"
    }
  },
  "sort": [
    {
      "age": {
        "order": "desc"
      }
    }
  ]
}

{
  "query": {
    "match_all": {}
  },
  "from": 0,
  "size": 4
}

get product/_search
{
  "aggs":{
    "price_group": {
      "terms": {
        "field": "price"
      }
    }
  },
  "size": 0
}

{
  "aggs":{
    "max_price": {
      "max": {
        "field": "price"
      }
    }
  },
  "size": 0
}

get product/_search
{
  "aggs":{
    "price_avg": {
      "avg": {
        "field": "price"
      }
    }
  },
  "size": 0
}
```

Bulk API

```
PUT example
{
  "settings": {
      "number_of_shards": 3,
      "number_of_replicas": 1
  }
}
PUT example/_mapping
{
  "properties": {
    "id": {
      "type": "long"
    },
    "name": {
      "type": "text"
    },
    "counter": {
      "type": "integer",
      "index": false
    },
    "tags": {
      "type": "text"
    }
  }
}

POST example/_bulk
{"index": {"_id": 1}}
{"id":1, "name": "admin", "counter":10, "tags":["red", "black"]}
```

```
{"index": {"_id": 2}}
{"id":2, "name": "张三", "counter":20, "tags":["green", "purple"]}
{"index": {"_id": 3}}
{"id":3, "name": "李四", "counter":30, "tags":["red", "blue"]}
{"index": {"_id": 4}}
{"id":4, "name": "tom", "counter":40, "tags":["orange"]}

POST example/_bulk
{"update": {"_id": 1}}
{"doc": {"id":1, "name": "admin-02", "counter":"11"}}
{"update": {"_id": 2}}
{"script":{"lang":"painless","source":"ctx._source.counter += params.num","params":{"num":2}}}
{"update":{"_id": 3}}
{"doc": {"name": "test3333name", "counter": 999}}
{"update":{"_id": 4}}
{"doc": {"name": "test444name", "counter": 888},  "doc_as_upsert" : true}

POST example/_bulk
{"delete": {"_id": 1}}
{"delete": {"_id": 2}}
{"delete": {"_id": 3}}
{"delete": {"_id": 4}}

get _mget
{
  "docs": [
    {
      "_index": "example",
      "_id": 2
    },
    {
      "_index": "example",
      "_id": 3
    }
  ]
}

post _sql?format=txt
{
  "query": """ select * from "user" """,
  "filter": {
    "range": {
      "telephone": {
        "gte": 111
      }
    }
  }
}
```