**1、使命不同**

Oracle 对关系型数据库的定义：

**关系型数据库**，是指采用了关系模型来组织数据的数据库，其以行和列的形式存储数据，以便于用户理解，关系型数据库这一系列的行（包含唯一 key 的记录）和列（存储了属性）被称为表，一组表组成了数据库。

Elasticsearch 的官方定义：

**Elasticsearch** 是一个分布式的开源搜索和分析引擎，适用于所有类型的数据，包括文本、数字、地理空间、结构化和非结构化数据。Elasticsearch 在 Apache Lucene 的基础上开发而成，由 Elasticsearch N.V.（即现在的 Elastic）于 2010 年首次发布。Elasticsearch 以其简单的 REST 风格 API、分布式特性、速度和可扩展性而闻名，是 Elastic Stack 的核心组件；Elastic Stack 是适用于数据采集、充实、存储、分析和可视化的一组开源工具。人们通常将 Elastic Stack 称为 ELK Stack（代指 Elasticsearch、Logstash 和 Kibana），目前 Elastic Stack 包括一系列丰富的轻量型数据采集代理，这些代理统称为 Beats，可用来向 Elasticsearch 发送数据。

A relational database can store data and also index it.

A search engine can index data but also store it.

如上可通俗解读为：

* 关系数据库可以存储数据并为其建立索引。
* 搜索引擎可以索引数据，但也可以存储数据。

**2、适用场景不同**

关系型数据库更适合 OLTP（是一种以事务元作为数据处理的单位、人机交互的计算机应用系统，最大优点：最大优点是可以即时地处理输入的数据，及时地回答）的业务场景；而 Elasticsearch不能当做纯数据库来使用。

* 原因 1：不支持事务，
* 原因 2：近实时而非准实时，由 refresh\_interval 控制，最快 1s 数据写入后可检索。

Elasticsearch 适合 OLAP的场景（它使分析人员能够迅速、一致、交互地从各个方面观察信息，以达到深入理解数据的目的。侧重分析）。

举例：

* 海量日志分析和检索、
* 海量大文本的全文检索等。

**3,存储类型不同**

关系型数据库一般只支持存储结构化数据（pgsql 支持 json）。

**结构化数据的特点**：

* 由二维表结构来逻辑表达和实现的数据
* 严格地遵循数据格式与长度规范。

举例：银行交易数据、个人信息数据等。

而 Elasticsearch 支持关系型和非结构化数据，如：json 由 object 或者 nested 类型或者父子 Join 存储。

**非结构化数据的特点**：

* 数据结构不规则或不完整；
* 没有预定义的数据模型，不方便用数据库二维逻辑表来表现的数据。

举例：包括所有格式的办公文档、文本、图片、XML, HTML、各类报表、图像和音频/视频信息等等。

脑海中想一下：是不是实战中遇到：数据结构不定、字段个数不定、字段类型不定、是否动态添加不定等多变的业务场景？

**4,可扩展性不同**

关系型数据库通病， 如：mysql 单表支持数据量有限，数据量大了就得分库分表，再大了考虑分布式，原生分布式的瓶颈如下：

* 分库分表非常麻烦，
* 业务依赖性高，
* 复杂查询会出现错误，
* 更重要的是分布式事务无法有效处理。

催生了很多第三方 NewSql 公司如：TIDB（开源+解决方案付费）。

而 Elasticsearh 支持横向扩展，天生支持多节点集群部署，扩展能力强，甚至支持跨集群检索；能支持 PB+的数据。

国内的：滴滴、携程、顺丰、今日头条、bat 等很多核心数据业务都已经通过 Elasticsearch 实现。

**5,解决问题不同**

关系型数据库针对核心：增删改查的业务场景，对于全文检索会慢的要死（很多客户迁移 Elasticsearch 就是这个原因，早期用 lucene 后用 solr，但发现 Elasticsearch 更好用）；而 Elasticsearch 的倒排索引机制更适合全文检索。

实际业务中：

* 如果数据量不大，建议使用简单的关系数据库结合简单的 SQL 查询就能解决问题。
* 如果您对性能没有问题，请保持架构简单并使用单个数据库存储，必要时加些缓存（如 redis）。
* 如果您在搜索中遇到性能问题，则可以将关系型数据库和 Elasticsearch 结合使用。

**6,数据模型不同**

关系型数据库通常针对复杂业务会多表设计、不同表不同模型，多表通过 join 关联或者视图查询。

而 Elasticsearch 支持复杂业务数据，通常不建议多表关联，确切说 Elasticsearch 倒排索引机制决定了它天然不适合多表关联。复杂业务数据通常解决方案：

* 1, 宽表（空间换时间）；
* 2, nested
* 3, 父子关联 join（针对频繁更新场景）。

对于聚合业务场景，的确大数据量（千万级以上）多重嵌套全量聚合 es 会很慢，业务选型可以考虑其他辅助方案。

**7、底层逻辑不同**

传统数据库的存储引擎为 B+树，包括 ES 的很多 NOSQL 数据库使用的 LSM Tree，对写操作支持更高效。

为什么 Elasticsearch/Lucene 检索可以比 mysql 快？

Mysq 的分词词典（term dictionary）是以 b-tree 排序的方式存储在磁盘上的。检索一个 term 需要若干次的随机访问磁盘操作。

而 Lucene 在分词词典的基础上添加了 term index（以 FST（finite state transducers）形式保存，非常节省内存）来加速检索，term index 以树的形式缓存在内存中。从 term index 查到对应的 term dictionary 的 block 位置之后，再去磁盘上找 term，大大减少了磁盘的随机访问次数。

**8、小结**

所以，没有最牛逼“一招先,吃遍天”的方案，只有最适合的方案。

适合自己业务场景的才是最好的！