一.什么是数据库

1. 数据库是存放数据的仓库。

2. 数据库是以一定方式储存在一起、能与多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合, 可视为电子化的文件柜——存储电子文件的处所, 用户可以对文件中的数据进行新增、查询、更新、删除等操作。

二.数据的发展

在数据库的发展历史上, 数据库先后经历了层次数据库、网状数据库和关系数据库等各个阶段的发展, 数据库技术在各个方面的快速的发展。 特别是关系型数据库已经成为目前数据库产品中最重要的一员, 80年代以来, 几乎所有的数据库厂商新出的数据库产品都支持关系型数据库, 即使一些非关系数据库产品也几乎都有支持关系数据库的接口。 这主要是传统的关系型数据库可以比较好的解决管理和存储关系型数据的问题。 随着云计算的发展和大数据时代的到来, 关系型数据库越来越无法满足需要, 这主要是由于越来越多的半关系型和非关系型数据需要用数据库进行存储管理, 以此同时, 分布式技术等新技术的出现也对数据库的技术提出了新的要求, 于是越来越多的非关系型数据库就开始出现, 这类数据库与传统的关系型数据库在设计和数据结构有了很大的不同, 它们更强调数据库数据的高并发读写和存储大数据, 这类数据库一般被称为NoSQL(Not only SQL)数据库。 而传统的关系型数据库在一些传统领域依然保持了强大的生命力。

三.关系数据库

1. 关系型数据库, 存储的格式可以直观地反映实体间的关系。关系型数据库和常见的表格比较相似, 关系型数据库中表与表之间是有很多复杂的关联关系的。 常见的关系型数据库有Mysql, SqlServer等。在轻量或者小型的应用中, 使用不同的关系型数据库对系统的性能影响不大, 但是在构建大型应用时, 则需要根据应用的业务需求和性能需求, 选择合适的关系型数据库。

2. 标准SQL语句

(1). 虽然关系型数据库有很多, 但是大多数都遵循SQL(结构化查询语言, Structured Query Language)标准。常见的操作有查询, 新增, 更新, 删除,求和, 排序等。

(2). 查询语句: SELECT param FROM table WHERE condition该语句可以理解为从 table 中查询出满足 condition 条件的字段 param。

(3). 新增语句: INSERT INTO table (param1， param2， param3) VALUES (value1, value2，value3)该语句可以理解为向table中的param1, param2, param3字段中分别插入value1，value2，value3。

(4). 更新语句: UPDATE table SET param=new\_value WHERE condition该语句可以理解为将满足condition条件的字段param更新为new\_value值。

(5). 删除语句: DELETE FROM table WHERE condition该语句可以理解为将满足condition条件的数据全部删除。

(6). 去重查询: SELECT DISTINCT param FROM table WHERE condition该语句可以理解为从表table中查询出满足条件condition的字段param, 但是param中重复的值只能出现一次。

(7). 排序查询: SELECT param FROM table WHERE condition ORDER BY param1该语句可以理解为从表table 中查询出满足condition条件的param，并且要按照param1升序的顺序进行排序。

(8). 总体来说, 数据库的SELECT，INSERT，UPDATE，DELETE对应了我们常用的增删改查四种操作。

(9). 关系型数据库对于结构化数据的处理更合适, 如学生成绩、地址等, 这样的数据一般情况下需要使用结构化的查询, 例如join, 这样的情况下, 关系型数据库就会比NoSQL数据库性能更优, 而且精确度更高。由于结构化数据的规模不算太大, 数据规模的增长通常也是可预期的, 所以针对结构化数据使用关系型数据库更好。关系型数据库十分注意数据操作的事务性、一致性, 如果对这方面的要求关系型数据库无疑可以很好的满足。

四. 非关系型数据库(NoSQL)

1. 随着近些年技术方向的不断拓展, 大量的NoSql数据库如MongoDB、Redis、Memcache出于简化数据库结构、避免冗余、影响性能的表连接、摒弃复杂、分布式的目的被设计。

2. NoSQL指的是分布式的、非关系型的、不保证遵循ACID原则的数据存储系统。NoSQL数据库技术与CAP理论、一致性哈希算法有密切关系。所谓CAP理论, 简单来说就是一个分布式系统不可能满足可用性、一致性与分区容错性这三个要,一次性满足两种要求是该系统的上限。而一致性哈希算则指的是NoSQL数据库在应用过程中, 为满足工作需求而在通常情况下产生的一种数据算法, 该算法能有效解决工作方面的诸多问题。但也存在弊端, 即工作完成质量会随着节点的变化而产生波动, 当节点过多时, 相关工作结果就无法那么准确。这一问题使整个系统的工作效率受到影响, 导致整个数据库系统的数据乱码与出错率大大提高, 甚至会出现数据节点的内容迁移, 产生错误的代码信息。但尽管如此, NoSQL数据库技术还是具有非常明显的应用优势, 如数据库结构相对简单, 在大数据量下的读写性能好; 能满足随时存储自定义数据格式需求, 非常适用于大数据处理工作。

3. NoSQL数据库适合追求速度和可扩展性、业务多变的应用场景。对于非结构化数据的处理更合适, 如文章、评论这些数据, 如全文搜索、机器学习通常只用于模糊处理, 并不需要像结构化数据一样, 进行精确查询, 而且这类数据的数据规模往往是海量的, 数据规模的增长往往也是不可能预期的, 而NoSQL数据库的扩展能力几乎也是无限的, 所以NoSQL数据库可以很好的满足这一类数据的存储。NoSQL数据库利用key-value可以大量的获取大量的非结构化数据, 并且数据的获取效率很高, 但用它查询结构化数据效果就比较差。

4. 目前NoSQL数据库仍然没有一个统一的标准, 它现在有四种大的分类:

(1). 键值对存储(key-value): 代表软件Redis, 它的优点能够进行数据的快速查询, 而缺点是需要存储数据之间的关系。

(2). 列存储: 代表软件Hbase, 它的优点是对数据能快速查询, 数据存储的扩展性强。而缺点是数据库的功能有局限性。

(3). 文档数据库存储: 代表软件MongoDB, 它的优点是对数据结构要求不特别的严格。而缺点是查询性的性能不好, 同时缺少一种统一查询语言。

(4). 图形数据库存储: 代表软件InfoGrid, 它的优点可以方便的利用图结构相关算法进行计算。而缺点是要想得到结果必须进行整个图的计算, 而且遇到不适合的数据模型时, 图形数据库很难使用。

五.NoSQL与关系型数据库的区别

1.存储方式

传统的关系型数据库采用表格的储存方式, 数据以行和列的方式进行存储, 要读取和查询都十分方便。而非关系型数据不适合这样的表格存储方式, 通常以数据集的方式, 大量的数据集中存储在一起, 类似于键值对、图结构或者文档。

2.存储结构

关系型数据库按照结构化的方法存储数据， 每个数据表都必须对各个字段定义好(也就是先定义好表的结构), 再根据表的结构存入数据, 这样做的好处就是由于数据的形式和内容在存入数据之前就已经定义好了, 所以整个数据表的可靠性和稳定性都比较高, 但带来的问题就是一旦存入数据后, 如果需要修改数据表的结构就会十分困难。 而NoSQL数据库由于面对的是大量非结构化的数据的存储, 它采用的是动态结构, 对于数据类型和结构的改变非常的适应, 可以根据数据存储的需要灵活的改变数据库的结构。

3.存储规范

关系型数据库为了避免重复、规范化数据以及充分利用好存储空间, 把数据按照最小关系表的形式进行存储, 这样数据管理的就可以变得很清晰、一目了然, 当然这主要是一张数据表的情况。 如果是多张表情况就不一样了, 由于数据涉及到多张数据表, 数据表之间存在着复杂的关系, 随着数据表数量的增加, 数据管理会越来越复杂。 而NoSQL数据库的数据存储方式是用平面数据集的方式集中存放, 虽然会存在数据被重复存储, 从而造成存储空间被浪费的问题(从当前的计算机硬件的发展来看, 这样的存储空间浪费的问题微不足道), 但是由于基本上单个数据库都是采用单独存放的形式, 很少采用分割存放的方式, 所以这样数据往往能存成一个整体, 这对于数据的读写提供了极大的方便。

4.扩展方式

当前社会和科学飞速发展, 要支持日益增长的数据库存储需求, 当然要求数据库有良好的扩展性能, 并且要求数据库支持更多数据并发量, 扩展方式是NoSQL数据库与关系型数据库差别最大的地方, 由于关系型数据库将数据存储在数据表中, 数据操作的瓶颈出现在多张数据表的操作中, 而且数据表越多这个问题越严重, 如果要缓解这个问题, 只能提高处理能力, 也就是选择速度更快性能更高的计算机, 这样的方法虽然可以一定的拓展空间, 但这样的拓展空间一定有非常有限的, 也就是关系型数据库只具备纵向扩展能力。 而NoSQL数据库由于使用的是数据集的存储方式，它的存储方式一定是分布式的, 它可以采用横向的方式来开展数据库, 也就是可以添加更多数据库服务器到资源池, 然后由这些增加的服务器来负担数据量增加的开销。

5.查询方式

关系型数据库采用结构化查询语言(即SQL)来对数据库进行查询, SQL早已获得了各个数据库厂商的支持, 成为数据库行业的标准, 它能够支持数据库的CRUD(增加, 查询, 更新, 删除)操作, 具有非常强大的功能, SQL可以采用类似索引的方法来加快查询操作。NoSQL数据库使用的是非结构化查询语言(UnQL),它以数据集(像文档)为单位来管理和操作数据, 由于它没有一个统一的标准, 所以每个数据库厂商提供产品标准是不一样的, NoSQL中的文档Id与关系型表中主键的概念类似, NoSQL数据库采用的数据访问模式相对SQL更简单而精确。

6.规范化

在数据库的设计开发过程中开发人员通常会面对同时需要对一个或者多个数据实体(包括数组、列表和嵌套数据)进行操作, 这样在关系型数据库中, 一个数据实体一般首先要分割成多个部分, 然后再对分割的部分进行规范化, 规范化以后再分别存入到多张关系型数据表中, 这是一个复杂的过程。 好消息是随着软件技术的发展, 相当多的软件开发平台都提供一些简单的解决方法, 例如: 可以利用ORM层(也就是对象关系映射)来将数据库中对象模型映射到基于SQL的关系型数据库中去以及进行不同类型系统的数据之间的转换。对于NoSQL数据库则没有这方面的问题, 它不需要规范化数据, 它通常是在一个单独的存储单元中存入一个复杂的数据实体。

7.事务性

关系型数据库强调ACID规则, 原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability), 可以满足对事务性要求较高或者需要进行复杂数据查询的数据操作, 而且可以充分满足数据库操作的高性能和操作稳定性的要求。并且关系型数据库十分强调数据的强一致性, 对于事务的操作有很好的支持。关系型数据库可以控制事务原子性细粒度, 并且一旦操作有误或者有需要可以马上回滚事务。 而NoSQL数据库强调BASE原则, 基本可用(Basically Availble)、软状态(Soft-state)、最终一致性(Eventual Consistency), 它减少了对数据的强一致性支持, 从而获得了基本一致性和柔性可靠性, 并且利用以上的特性达到了高可靠性和高性能, 最终达到了数据的最终一致性。NoSQL数据库虽然对于事务操作也可以使用, 但由于它是一种基于节点的分布式数据库, 对于事务的操作不能很好的支持, 也很难满足其全部的需求, 所以NoSQL数据库的性能和优点更多的体现在大数据的处理和数据库的扩展方面。

8.读写性能

关系型数据库十分强调数据的一致性, 并为此降低读写性能付出了巨大的代价,虽然关系型数据库存储数据和处理数据的可靠性很不错， 但一旦面对海量数据的处理的时候效率就会变得很差， 特别是遇到高并发读写的时候性能就会下降的非常厉害。 而NoSQL数据库相对关系型数据库优势最大的恰恰是应对大数据方面, 也就是对于大量的每天都产生非结构化的数据能够高性能的读写, 这是因为NoSQL数据库是按key-value类型进行存储的, 以数据集的方式存储的, 因此无论是扩展还是读写都非常容易, 并且NoSQL数据库不需要关系型数据库繁琐的解析, 所以NoSQL数据库大数据管理、检索、读写、分析以及可视化方面具有关系型数据库不可比拟的优势。

9.授权方式

关系型数据库常见的有Oracle、SQLServer、DB2、Mysql, 除了Mysql大多数的关系型数据库如果要使用都需要支付一笔价格高昂的费用, 即使是免费的Mysql性能也受到了诸多的限制。而对于NoSQL数据库, 比较主流的有redis、HBase、MongoDb、Memcache等产品, 通常都采用开源的方式, 不需要像关系型数据库那样, 需要一笔高昂的花费。