HIVE基础知识

- 一、HIVE概述
  - （一）什么是HIVE
    - 1、基于HADOOP 的数据仓库工具（编译工具）
    - 2、核心工作就是把SQL语句翻译成MR程序，和HDFS和MAP-REDUCE结合使用之后，可以在HADOOP上构建数据仓库
  - （二）和传统数据仓库的比较
    - 1、相同之处
      - ① 主要用来访问和管理数据
      - ②也提供类如SQL查询语言，HQL
    - 2、不同之处
      - ① 可以处理超大规模数据
      - ②可拓展性和容错性非常强
  - （三）HIVE-数据分析引擎
    - 1、MAP-REDUCE不支持SQL语法，其执行逻辑一般的使用JAVA语言
    - 2、HIVE会自动把SQL编译成Java语言发送给MAP-REDUCE引擎执行
  - （四）HIVE典型的应用场景
    - 1、HIVE主要应用在数据的离线分析
    - 2、日志分析
      - 统一网站一段时间内的PV、UV
      - 多维度数据分析
      - 大部分互联网公司使用HIVE进行日志分析，
    - 3、其他场景
      - 海量结构化数据离线分析（平安集团、vivo等）
  - （五）HIVE 不能做什么
    - 1、 Hive不是一个OLAP(On-Line Analytical Processing)系统
      - 响应时间慢
      - 无法实时更新数据
    - 2、Hive不是一个OLTP(On-line Transaction Processing)系统
      - 对事务的支持很弱
    - 3、Hive的表达能力有限
      - 不支持迭代式计算

- 有些复杂运算用SQL不易表达

二、HIVE基础知识

(一)常用的数据类型

- 1、string 字符串
- 2、decimal 数值
- 3、 date 日期

（二）数据的存储

- 1、Hive中所有的数据都存储在 HDFS 中，没有专门的数据存储格式（可支持Text，SequenceFile，ParquetFile，RCFILE，ORCFILE等）
- 2、Hive 的数据模型
  - ① db：在hdfs中表现为${hive.metastore.warehouse.dir}目录下一个文件夹
  - ② table：在hdfs中表现所属db目录下一个文件夹
  - ③ external table：与table类似，不过其数据存放位置可以在任意指定路径
  - ④ partition：在hdfs中表现为table目录下的子目录
  - ⑤ bucket：在hdfs中表现为同一个表目录下根据hash散列之后的多个文件

（三）HIVE常用命令

- 1、show databases; # 查看某个数据库
- 2、 use 数据库; # 进入某个数据库
- 3、show tables; # 展示所有表
- 4、desc 表名; # 显示表结构
- 5、show partitions 表名; # 显示表名的分区
- 6、show create table_name; # 显示创建表的结构

（四）HQL和SQL的差异

- 1、 join 条件仅支持等值关联且不支持or条件
- 2、 HQL中没有UNION，可使用distinct+ union all 实现 UNION
- 3、日期判断，建议使用to_date(),如：to_date(orderdate)='2016-07-18'
- 4、 SORT BY 与order BY 区别：
  - 和传统sql中的order by 一样，对数据做全局排序，加上排序，会新启动一个job进行排序，会把所有数据放到同一个reduce中进行处理，不管数据多少，不管文件多少，都启用一个reduce进行处理。
  - sort by 是局部排序，会在每个reduce端做排序，每个reduce端是排序的，也就是每个reduce出来的数据是有序的，但是全部不一定有序，除非一个reduce，一般情况下可以先进行局部排序完成后，再进行全局排序，会提高不少效率。
- 5、DISTRIBUTE BY 和SORT BY ： DISTRIBUTE BY控制map的输出在reducer中是如何划分的.

- 6、CLUSTER BY： 等价于DISTRIBUTE BY 和SORT BY同时对一个字段使用
- 7、不支持数据植入现有的表或者分区，仅支持覆盖重写整个表

（五） 数据的存储格式

1、传统行式存储

- ① 数据的按行存储的
- ② 没有索引的查询，使用大量的I/O
- ③ 建立索引和物化视图需要大量时间和资源
- ④ 面向查询的需求，数据库必须被大量膨胀才能满足性能需求

2、列式存储

- ① 数据是按列存储的，每一列单独存放
- ② 数据即是索引
- ③ 只访问需要查询的列，大量降低系统I/O
- ④ 每一列由一个线索来处理--查询的并发处理
- ⑤ 数据类型一致，数据特征相似-高效压缩