数据库进阶 day6

分库分表

-将放在一个数库（主机）中的数据，按照特定方式进行拆分，分散存放到多个数据库（主机）中，以达到分散单台设备负载的效果。

垂直分割

纵向切分

-将单个表，拆分成多个表，分散到不同的数据库

-将单个数据库的多该表进行分类，按业务类别分散到不同的数据库上

水平分割

横向切分

-按照表中某个字段的某种规则，把表中许多记录按行？切分，分散到多个数据库中

Mycat

-是基于java的分布式数据库系统中间层，为高并发环境的分布式访问提供解决反感

-支持jdbc形式连接

-支持mysql，oracle，sqlserver，mongodb等

-提供数据读写分离服务

-可以实现数据库服务器的高可用

-提供数据分片服务

-基于阿里巴巴cobar进行研发的开原软件

-适合数据大量写入数据的存储需求

分片规则

Mycat支持10种分片规则

-枚举法 sharding-by-intfile

-固定分片 rule1

-范围约定 auto-sharding-long

-求模法 mod-long

-日期列分区法 sharding-by-date

-通配取模 sharding-by-pattern

-ASCII码求模通配 sharding-by-prefixpattem

-编程指定 sharding-by-substring

-字符串拆分hash解析 sharding-by-string

-一致性hash sharbing-by-murmur

工作过程

-先了解这个sql查找涉及到的表

-然后看此表的定义，如果有分片规则，则获取sql里分片字段的值，并且匹配分片函数，获得分片列表

-然后将sql发网这些分片去执行

-最后收集和处理所有分片结果数据，并返回到客户端

环境部署

安装软件包

安装JDK #系统自带

安装mycat-server的包

目录结构

-bin #mycat命令 如启动 停止

-catlet #扩展功能

-conf #配置文件

-lib #mycat使用的jar

-log #mycat启动和运行日志

-wrapper.log #mycat服务启动日志

-mycat.log #记录sql脚本执行后的报错内容

配置文件

-Server.xml #设置连mycat的账户，以及访问权限

-schema.xml #设置mycat的真实库

-rule.xml #定义mycat分片规则以及调度算法

配置标签说明

<user>.....</user> #定义连接mycat用户信息

<datanode>.....</datanode> #指定数据节点

<datahost>.....</datahost> #指定数据库地址及用户信息

修改server.xml文件

<user name="admin"> #定义用户名

<property name="password">123456</property> #用户名密码

<property name="schemas">TESTDB</property> schemas的虚拟库

</user>

<user name="user">

<property name="password">user</property>

<property name="schemas">TESTDB</property>

<property name="readOnly">true</property> #这个用户连接时只有读的权限

修改配置schema

<table name="travelrecord" dataNode="dn1,dn2"

表名 数据节点

rule="auto-sharding-long" /

分片规则

<dataNode name="dn1" dataHost="localhost1" database="db1" />

数据节点 节点的名字 数据名称 库名

<dataHost name="localhost1" maxCon="1000" minCon="10"

数库名称 名称

balance="0"

writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">

<heartbeat>select user()</heartbeat>

<!-- can have multi write hosts -->

<writeHost host="hostM1" url="192.168.4.54:3306"

数据库的IP地址

user="root" password="123qqq...A">

连接时的用户 连接时的密码

<!-- can have multi read hosts -->

</writeHost>