通过管道传输pipe将MySQL数据批量导入Redis

自Redis 2.6以上版本起，Redis支持快速大批量导入数据，即官网的Redis Mass Insertion,即Pipe传输，

通过将要导入的命令转换为Resp格式，然后通过MySQL的concat()来整理出最终导入的命令集合，以达到快速导入的目的。

1. 根据需求设计好Redis的hash结构，关键是Key的设计

Redis其实就是内存数据库，而其中最常用的就是hash结构，key-value，查询时需要使用到key，所以key的设计决定了查询的效率，

我们以虚拟机的stress库下面的order为例

批量设值

HMSET myhash field1 "Hello" field2 "World"

紫色框引用了select出来的数据，然后转换成符合RESP协议格式：

第一行的 \*10\r\n : \*表示数组，10表示数组元素个数\*（可以看下面的SQL查询语句标红的字段有10个，所以为10）， \r\n是规定分隔符

第二行的 '$',LENGTH(redis\_cmd),'\r\n',redis\_cmd,'\r\n',

: $表示长字符串，LENGTH(redis\_cmd)表示字符串长度，

redis\_cmd字符串变量，

\r\n还是规定字符串(window的redis是\n)

测试表结构：

|  |
| --- |
| Create Table |
| CREATE TABLE `order` (   `orderid` varchar(38) DEFAULT NULL,   `ordertime` datetime DEFAULT NULL,   `ordermoney` decimal(20,0) DEFAULT NULL,   `orderstatus` char(1) DEFAULT NULL,   `version` int(11) DEFAULT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8 |

查询SQL语句：

SELECT CONCAT(

'\*10\r\n',

'$', LENGTH(redis\_cmd), '\r\n', redis\_cmd, '\r\n',

'$', LENGTH(redis\_key), '\r\n', redis\_key, '\r\n',

'$', LENGTH(hkey1),'\r\n',hkey1,'\r\n','$',LENGTH(hval1),'\r\n',hval1,'\r\n',

'$', LENGTH(hkey2),'\r\n',hkey2,'\r\n','$',LENGTH(hval2),'\r\n',hval2,'\r\n',

'$', LENGTH(hkey3),'\r\n',hkey3,'\r\n','$',LENGTH(hval3),'\r\n',hval3,'\r\n',

'$', LENGTH(hkey4),'\r\n',hkey4,'\r\n','$',LENGTH(hval4),'\r\n',hval4,'\r'

)

FROM (

SELECT

'HSET' AS redis\_cmd,

CONCAT('order:info:',orderid) AS redis\_key,

'ordertime' AS hkey1, ordertime AS hval1,

'ordermoney' AS hkey2, ordermoney AS hval2,

'orderstatus' AS hkey3, orderstatus AS hval3,

'version' AS hkey4, `version` AS hval4

FROM `order`

) AS t

将以上查询SQL查询语句在LINUX上存储order.sql文件

执行指令：

mysql -utest -ptest stress --default-character-set=utf8 --skip-column-names --raw < order.sql | redis-cli --pipe

-utest -ptest stress 代表连接test用户 密码为test 数据库为stress，并登录数据库

--raw < order.sql 表示将order.sql的sql语句内容粘贴到mysql里执行（以上指令代表已登录MYSQL）

redis-cli --pipe 表示使用redis pipe管道把数据导入到redis中