



南开大学
Nankai University

分区分表

数据库系统上机

计算机学院&网络空间安全学院 乜鹏

<https://dbis.nankai.edu.cn/2019/0417/c12139a128118/page.htm>



声明：上机课程的内容偏向举例，通俗化，一些术语的准确定义请查看理论课程。



为什么要分区分表?



南开大学
Nankai University

日常开发中我们经常会遇到大表的情况，所谓的大表是指存储了百万级乃至千万级条记录的表。这样的表过于庞大，导致数据库在查询和插入的时候耗时太长，性能低下，如果涉及联合查询的情况，性能会更加糟糕。分表和表分区的目的就是减少数据库的负担，提高数据库的效率，通常点来讲就是提高表的增删改查效率。

允公允能 日新月异

NANKAI UNIVERSITY



分区

- 分区, partition, 分区是将数据分段划分在多个位置存放, 可以是同一块磁盘也可以在不同的机器。
- 分区后, 表面上还是一张表, 但数据散列到多个位置了。
- 用户读写的时候操作的还是大表名字, 数据库自动去组织分区的数据。
- 其实每个分区, 就是独立的表。都要存储该分区数据的数据, 索引等信息。

分表

- 分表是将一个大表按照一定的规则分解成多张具有独立存储空间的实体表, 我们可以称为子表, 每个表都对应三个文件, MYD数据文件, .MYI索引文件, .frm表结构文件。
- 这些子表可以分布在同一块磁盘上, 也可以在不同的机器上。
- app读写的时候根据事先定义好的规则得到对应的子表名, 然后去操作它。

注: MYSQL5.1后, 出现了分区的功能

投票 最多可选1项



MySQL分表用于哪种引擎更多一些?

- A InnoDB
- B MyISAM

允公允能 日新月异

NANKAI UNIVERSITY



分表有两种分割方式，一种垂直分割，一种水平分割。

- **垂直拆分**是指数据表列的拆分，把一张列比较多的表拆分为多张表。垂直分割一般用于拆分大字段和访问频率低的字段，分离冷热数据。
- **水平拆分**是指数据表行的拆分，表的行数超过500万行或者单表容量超过10GB时，查询就会变慢，这时可以把一张的表的数据拆成多张表来存放。水平分表尽可能使每张表的数据量相当，比较均匀。举个例子：我们可以将用户信息表拆分成多个用户信息表，这样就可以避免单一表数据量过大对性能造成影响。



单库单表

单库单表是最常见的数据库设计，例如，有一张用户(user)表放在数据库DB中，所有的用户都可以在DB库中的user表中查到。

单库多表

随着用户数量的增加，user表的数据量会越来越大，可以通过某种方式将user进行水平的切分，产生两个表结构完全一样的user_0000,user_0001等表，user_0000 + user_0001 + ...的数据刚好是一份完整的数据。

多库多表

随着数据量增加也许单台DB的存储空间不够，随着查询量的增加单台数据库服务器已经没办法支撑。这个时候可以再对数据库进行水平拆分。

投票 最多可选1项



大型在线游戏的用户数据表应该通过什么方式进行存取?

- A 单库单表
- B 单库多表
- C 多库多表

允公允能 日新月异

NANKAI UNIVERSITY

MySQL支持的分区算法有四种：

- RANGE分区：基于属于一个给定连续区间的列值，把多行分配给分区。
- LIST分区：类似于按RANGE分区，区别在于LIST分区是基于列值匹配一个离散值集合中的某个值来进行选择。
- HASH分区：基于用户定义的表达式的返回值来进行选择的分区，该表达式使用将要插入到表中的这些行的列值进行计算。这个函数可以包含MySQL中有效的、产生非负整数值的任何表达式。
- KEY分区：类似于按HASH分区，区别在于KEY分区只支持计算一列或多列，且MySQL服务器提供其自身的哈希函数。必须有一列或多列包含整数值。



分区算法示例



南开大学
Nankai University

- RANGE分区:

```
drop table if exists member;
create table member(
  firstname varchar(25) not null,
  lastname varchar(25) not null,
  username varchar(16) not null,
  email varchar(35),
  joined date not null
)
partition by range columns(joined)(
  partition p0 values less than ('1960-01-01'),
  partition p1 values less than ('1970-01-01'),
  partition p2 values less than ('1980-01-01'),
  partition p3 values less than ('1990-01-01'),
  partition p4 values less than maxvalue
)
```

- LIST分区:

```
drop table if exists staff;
create table staff(
  id int not null,
  fname varchar(30),
  lname varchar(30),
  hired date not null default '1970-01-01',
  separated date not null default '9999-12-31',
  job_code int not null default 0,
  store_id int not null default 0
)
partition by list(store_id)(
  partition pNorth values in (3,5,6,9,17),
  partition pEast values in (1,2,10,11,19,20),
  partition pWest values in (4,12,13,14,18),
  partition pCentral values in (7,8,15,16)
);
```

允公允能 日新月异

NANKAI UNIVERSITY



分区算法示例



南开大学
Nankai University

- HASH分区:

```
drop table if exists staff;  
create table staff(  
  id int not null,  
  fname varchar(30),  
  lname varchar(30),  
  hired date not null default '1970-01-01',  
  separated date not null default '9999-12-31',  
  job_code int not null default 0,  
  store_id int not null default 0  
)  
partition by hash(store_id)  
partitions 4;
```

- KEY分区:

```
drop table if exists staff;  
create table staff(  
  id int not null,  
  fname varchar(30),  
  lname varchar(30),  
  hired date not null default '1970-01-01',  
  separated date not null default '9999-12-31',  
  job_code int not null default 0,  
  store_id int not null default 0  
)  
partition by key(store_id)  
partitions 4;
```

Hash分区也存在与传统Hash分表一样的问题, 可扩展性差。MySQL也提供了一个类似于一致Hash的分区方法 - 线性Hash分区, 只需要在定义分区时添加LINEAR关键字。

允公允能 日新月异

NANKAI UNIVERSITY



实例操作



南开大学
Nankai University

1. 下载数据集

1. [迷你版zip下载](#) (829KB)
2. [精简版zip下载](#) (17MB)
3. [完整版zip下载](#) (149MB)

2. 新建表 test0 和 test1, test1进行RANGE分区

```
drop table if exists test0;  
create table test0(  
  word varchar(25),  
  count int,  
  id int(11)  
);
```

```
drop table if exists test1;  
create table test1(  
  word varchar(25),  
  count int,  
  id int(11)  
)  
partition by range columns(count)(  
  partition p0 values less than (10),  
  partition p1 values less than (20),  
  partition p2 values less than (50),  
  partition p3 values less than (100),  
  partition p4 values less than maxvalue  
);
```

允公允能 日新月异

NANKAI UNIVERSITY



实例操作



南开大学
Nankai University

3. 分别导入相同数据，对比导入时间（迷你数据集）

表:	1
已处理:	100000
错误:	0
已添加:	100000
已更新:	0
已删除:	0
时间:	60.571s

表:	1
已处理:	100000
错误:	0
已添加:	100000
已更新:	0
已删除:	0
时间:	58.481s

采用RANGE分区前后插入时间对比



实例操作



南开大学
Nankai University

4. 执行同一条查询语句，对比执行时间（迷你数据集）

```
SELECT * from test0 where count=50;  
受影响的行: 0  
时间: 0.197s
```

```
SELECT * from test1 where count=50;  
受影响的行: 0  
时间: 0.133s
```

采用RANGE分区前后查询时间对比

投票 最多可选4项



数据库的数据量达到一定程度之后，为避免带来系统性能上的瓶颈，采用的手段可以是（）

- ☐ A 分区
- ☐ B 分表
- ☐ C 分库
- ☐ D 分片

允公允能 日新月异

NANKAI UNIVERSITY