

达梦分区表的使用

在大型的企业应用或企业级的数据库应用中，要处理的数据量通常达到 TB 级，对于这样的大型表执行全表扫描或者 DML 操作时，效率是非常低的。

为了提高数据库在大数据量读写操作和查询时的效率，达梦数据库提供了对表和索引进行分区的技术，把表和索引等数据库对象中的数据分割成小的单位，分别存放在一个个单独的段中，用户对表的访问转化为对较小段的访问，以改善大型应用系统的性能。

达梦数据库分区表主要包括范围分区、哈希分区和列表分区三种方式，企业可以使用合适的分区方法，如日期（范围）、区域（列表），对大量数据进行分区。由于达梦数据库划分的分区是相互独立且可以存储于不同的存储介质上的，完全可满足企业高可用性、均衡 IO、降低维护成本、提高查询性能的要求。今天我们主要讨论水平分区。

我们今天的测试环境

操作系统：中标麒麟 6 64 位

数据库版本：达梦 8.1

```
[dmdba@NeoKylin6-dm8 ~]$ uname -ra
Linux NeoKylin6-dm8 2.6.32-220.el6.x86_64 #1 SMP Tue Apr 10 05:55:10 EDT 2012 x86_64 x86_64 x86_64 GNU/Linux
[dmdba@NeoKylin6-dm8 ~]$ uname -m
x86_64
[dmdba@NeoKylin6-dm8 ~]$ disql
disql V8.1.0.147-Build(2019.03.27-104581)ENT
用户名:sysdba
密码:

服务器[LOCALHOST:5236]:处于普通打开状态
登录使用时间: 4.100(毫秒)
SQL>
```

一、 创建分区表

1.创建范围分区表

```
create table r_t1 (pid int primary key ,id int)
partition by range (pid)
(partition p1 values less than (101),
partition p2 values less than (201));
```

```
SQL> create table r_t1 (pid int primary key ,id int)
partition by range (pid)
(partition p1 values less than (101),
partition p2 values less than (201));
2 3 4 警告：范围分区未包含MAXVALUE,可能无法定位到分区
操作已执行
已用时间：211.739(毫秒)。执行号:3.
SQL>
```

将 1-200 的值录入到 t_r1 中。

```
begin
for i in 1..200 loop
insert into r_t1 values (i,i+1);
end loop;
end;
```

```
SQL> begin
for i in 1..200 loop
insert into r_t1 values (i,i+1);
end loop;
end;
2 3 4 5 6 /
DMSQL 过程已成功完成
已用时间：35.980(毫秒)。执行号:4.
SQL>
```

查看表 r_t1 的类型，显示为分区表

```
select table_name,PARTITIONED from dba_tables where
table_name='R_T1';
```

```
SQL> select table_name,PARTITIONED from dba_tables where table_name='R_T1';

行号      TABLE_NAME PARTITIONED
-----
1          R_T1          YES

已用时间：00:00:01.042。执行号:5.
SQL>
```

查看分区表的分区，显示为二个分区 p1 和 p2

```
select table_name,partition_name from dba_tab_partitions where
table_name='R_T1';
```

```
SQL> select table_name,partition_name from dba_tab_partitions where table_name='R_T1';
```

行号	TABLE_NAME	PARTITION_NAME
1	R_T1	P2
2	R_T1	P1

```
已用时间: 49.354(毫秒). 执行号:6.
SQL>
```

查询分区表中的记录数

```
SQL> select count(*) from r_t1 partition (p1);
```

```
SQL> select count(*) from r_t1 partition (p1);
```

行号	COUNT(*)
1	100

```
已用时间: 0.679(毫秒). 执行号:7.
SQL>
```

这样一个范围分区就建立好了。

2.创建列表分区

```
create table l_t1 (id int,name varchar(20))
partition by list (name)
(partition p1 values ('湖北'),
partition p2 values ('湖南'),
partition p3 values ('安徽'));
```

```
SQL> create table l_t1 (id int,name varchar(20))
partition by list (name)
(partition p1 values ('湖北'),
partition p2 values ('湖南'),
partition p3 values ('安徽'));
2 3 4 5 警告: 列表分区未包含DEFAULT,可能无法定位到分区
操作已执行
已用时间: 57.799(毫秒). 执行号:8.
SQL>
```

查看分区表的分区，显示为三个分区 p1, p2, p3

```
SQL>select table_name,partition_name from dba_tab_partitions
where table_name='L_T1';
```

```
SQL> select table_name,partition_name from dba_tab_partitions where table_name='L_T1';
```

行号	TABLE_NAME	PARTITION_NAME
1	L_T1	P3
2	L_T1	P1
3	L_T1	P2

已用时间: 92.839(毫秒). 执行号:10.

SQL>

3.创建哈希分区

```
SQL>CREATE TABLE sales01(sales_id INT, saleman CHAR(20),
saledate DATETIME, city CHAR(10))
PARTITION BY HASH(city)
(PARTITION p1,
PARTITION p2,
PARTITION p3,
PARTITION p4 );
```

```
SQL> CREATE TABLE sales01(sales_id INT, saleman CHAR(20),
saledate DATETIME, city CHAR(10))
PARTITION BY HASH(city)
(PARTITION p1,
PARTITION p2,
PARTITION p3,
PARTITION p4 );
```

2 3 4 5 6 7 操作已执行
已用时间: 53.912(毫秒). 执行号:11.

SQL>

如果不需指定分区表名，可以通过指定哈希分区个数来建立哈希分区表。

```
SQL>CREATE TABLE sales02(sales_id int, saleman char(20),
saledate DATETIME, city CHAR(10)) PARTITION BY HASH(city)
PARTITIONS 4;
```

```
SQL> CREATE TABLE sales02(sales_id int, saleman char(20), saledate DATETIME, city CHAR(10)) PARTITION BY HASH(city)
PARTITIONS 4;
```

2 操作已执行
已用时间: 39.960(毫秒). 执行号:12.
SQL>

激活

PARTITIONS 后的数字表示哈希分区的分区数，STORE IN 子句中指定了哈希分区依次使用的表空间。使用这种方式建立的哈希分区表分区名是匿名的，DM7 统一使用 DMHASHPART+分区号（从 0 开始）作为分区名。例如，需要查询 sales02 第一个分区的数据，可执行以下语句：

```
SQL>SELECT * FROM sales02 PARTITION (DMHASHPART1);
```

```
SQL> SELECT * FROM sales02 PARTITION (DMHASHPART1);  
未选定行
```

```
已用时间: 0.678(毫秒). 执行号:14.
```

二、 增加分区

```
SQL> alter table r_t1 add partition p3 values less than(301);
```

```
SQL> alter table r_t1 add partition p3 values less than(301);  
操作已执行  
已用时间: 108.880(毫秒). 执行号:15.  
SQL>
```

查看分区数，可以看到新增了一个分区 p3

```
SQL> select table_name,partition_name from dba_tab_partitions where table_name='R_T1';
```

行号	TABLE_NAME	PARTITION_NAME
1	R_T1	P2
2	R_T1	P3
3	R_T1	P1

```
已用时间: 8.947(毫秒). 执行号:16.  
SQL>
```

三、 删除分区

```
SQL> alter table r_t1 drop partition p3;
```

```
SQL> alter table r_t1 drop partition p3;  
操作已执行  
已用时间: 140.116(毫秒). 执行号:17.  
SQL> select table_name,partition_name from dba_tab_partitions where table_name='R_T1';
```

行号	TABLE_NAME	PARTITION_NAME
1	R_T1	P2
2	R_T1	P1

```
已用时间: 8.074(毫秒). 执行号:18.  
SQL>
```

可以看到，新增的分区 P3 被删除了。

四、 合并分区

```
Sql>alter table r_t1 merge partitions p1,p2 into partition p1_2;
```

```
SQL> alter table r_t1 merge partitions p1,p2 into partition p1_2;
```

操作已执行

已用时间: 54.174(毫秒). 执行号:19.

```
SQL> select table_name,partition_name from dba_tab_partitions where table_name='R_T1';
```

行号	TABLE_NAME	PARTITION_NAME
1	R_T1	P1_2

已用时间: 8.892(毫秒). 执行号:20.

```
SQL>
```

五、 拆分区

```
SQL> alter table r_t1 split partition p1_2 at (100) into  
(partition p1,partition p2);
```

```
SQL> alter table r_t1 split partition p1_2 at (100) into (partition p1,partition p2);
```

操作已执行

已用时间: 36.493(毫秒). 执行号:21.

```
SQL> select table_name,partition_name from dba_tab_partitions where table_name='R_T1';
```

行号	TABLE_NAME	PARTITION_NAME
1	R_T1	P2
2	R_T1	P1

已用时间: 8.965(毫秒). 执行号:22.

```
SQL>
```

六、 水平分区表的限制

- 1.分区列类型必须是数值型、字符型或日期型，不支持 BLOB、CLOB、IMAGE、TEXT、 LONGVARCHAR、BIT、BINARY、VARBINARY、LONGVARBINARY、时间间隔类型和用户自定义类型为分区列。
- 2.范围分区和哈希分区的分区键可以多个，最多不超过 16 列；LIST 分区的分区键 必须唯一。
- 3.水平分区表指定主键和唯一约束时，分区键必须都包含在主键和唯一约束中。
- 4.水平分区表不支持临时表。

- 5.不能在水平分区表上建立自引用约束。
- 6.普通环境中，水平分区表的各级分区数的总和上限是 65535；MPP 环境下，水平分区表的各级分区总数上限取决于 INI 参数 MAX_EP_SITES，上限为 $2^{(16 - \log_2 \text{MAX_EP_SITES})}$ 。比如：当 MAX_EP_SITES 为默认值 64 时，分区总数上限为 1024。
- 7.不允许对分区子表执行任何 DDL 操作。
- 8.哈希分区支持重命名、删除约束、设置触发器是否启用的修改操作。
- 9.范围分区支持分区合并、拆分、增加、删除、交换、重命名、删除约束、设置触发器是否生效操作。
- 10.LIST 分区支持分区增加、删除、交换、重命名、删除约束、设置触发器是否生效操作。
- 11.LIST 分区范围值不能为 NULL。

今天的分享就到这里了，感兴趣的同学赶紧去体验一下达梦的分区表吧！