:: DEVELOPER ZONE

MySQL 5.1 Reference Manual :: 18 Partitioning :: 18.2 Partition Types :: 18.2.3 HASH Partitioning :: 18.2.3.1 LINEAR HASH Partitioning

Search the MySQL manual:

Manual 5.1 Go

- MySQL 5.1 Reference Manual

- 18.2.3 HASH Partitioning
- 18.2.3.1 LINEAR HASH Partitioning

Get the MySQL Language Reference and MySQL Administrator's Guide from MySQL Press!

Previous / Next / Up / Table of Contents

18.2.3.1. `LINEAR HASH` Partitioning

MySQL also supports linear hashing, which differs from regular hashing in that linear hashing utilizes a linear powers-of-two algorithm whereas regular hashing employs the modulus of the hashing function's value.

Syntactically, the only difference between linear-hash partitioning and regular hashing is the addition of the `LINEAR` keyword in the `PARTITION BY` clause, as shown here:

```
CREATE TABLE employees (
    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,
    store_id INT
)
PARTITION BY LINEAR HASH(YEAR(hired))
PARTITIONS 4;
```

Given an expression *`expr`*, the partition in which the record is stored when linear hashing is used is partition number *N* from among *`num`* partitions, where *N* is derived according to the following algorithm:

1. Find the next power of 2 greater than *`num`*. We call this value *V*; it can be calculated as:

   ```
   V = POWER(2, CEILING(LOG(2, num)))
   ```

   (For example, suppose that *`num`* is 13. Then `LOG(2,13)` is 3.7004397181411. `CEILING(3.7004397181411)` is 4, and *V* = `POWER(2,4)`, which is 16.)

2. Set *N* = *F*(*`column_list`*) & (*V* - 1).

3. While *N* >= *`num`*:

   - Set *V* = CEIL(*V* / 2)

   - Set *N* = *N* & (*V* - 1)

For example, suppose that the table `t1`, using linear hash partitioning and having 6 partitions, is created using this statement:

```
CREATE TABLE t1 (col1 INT, col2 CHAR(5), col3 DATE)
    PARTITION BY LINEAR HASH( YEAR(col3) )
    PARTITIONS 6;
```

Now assume that you want to insert two records into `t1` having the `col3` column values `'2003-04-14'` and `'1998-10-19'`. The partition number for the first of these is determined as follows:

```
V = POWER(2, CEILING(LOG(2,7))) = 8
```

```
N = YEAR('2003-04-14') & (8 - 1)
   = 2003 & 7
   = 3

(3 >= 6 is FALSE: record stored in partition #3)
```

The number of the partition where the second record is stored is calculated as shown here:

```
V = 8
N = YEAR('1998-10-19') & (8-1)
  = 1998 & 7
  = 6

(6 >= 6 is TRUE: additional step required)

N = 6 & CEILING(5 / 2)
  = 6 & 3
  = 2

(2 >= 6 is FALSE: record stored in partition #2)
```

The advantage in partitioning by linear hash is that the adding, dropping, merging, and splitting of partitions is made much faster, which can be beneficial when dealing with tables containing extremely large amounts (terabytes) of data. The disadvantage is that data is less likely to be evenly distributed between partitions as compared with the distribution obtained using regular hash partitioning.

Previous / Next / Up / Table of Contents

User Comments

Add your own comment.

Top / Previous / Next / Up / Table of Contents