Partitioning

Sheeri K. Cabral
Database Administrator
The Pythian Group, www.pythian.com

cabral@pythian.com
January 12, 2009

Partitioning

- Why
- What
- How

How to Partition

CREATE TABLE tblname (

fld FLDTYPE ....)

PARTITION BY TYPE(fld) part_info

- Works with most storage engines!

Partition Definition Information

- 1024 max partitions/subpartitions
- Case-insensitive partition names
- Can use a function instead of a field
  - PARTITION BY RANGE(expr)

Partition Definition Requirements

- Only integer/NULL fields (except KEY)

- Partitioning function can only use fields from the PRIMARY or UNIQUE keys.

```
ERROR 1503 (HY000): A PRIMARY KEY must
  include all columns in the table's
  partitioning function
```

Partition Definition Restrictions

- No FULLTEXT index
- No SPATIAL data
- No foreign key or reference to

Partition Definition Restrictions

- No TEMPORARY, MERGE, FEDERATED, CSV
- No tables used for internal logging
- No key cache

Partition Definition Restrictions

- No INSERT DELAYED
- Table {DATA|INDEX} DIRECTORY ignored
- No mysqlcheck / myisamchk

RANGE Partitioning

- type = RANGE
- specify VALUES LESS THAN
- specify partition names

RANGE Partition Definition

```
CREATE TABLE tblname (
    fld FLDTYPE .... )
    PARTITION BY RANGE(fld)  (
      PARTITION pname VALUES LESS THAN (val)
     [,PARTITION pname VALUES LESS THAN (val)]
    )
```

RANGE Partition Definition

```
 CREATE TABLE nums_range (
  id INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (id)
) ENGINE=InnoDB PARTITION BY RANGE (id) (
PARTITION p0mil VALUES LESS THAN (1000000),
PARTITION p1mil VALUES LESS THAN (2000000),
PARTITION p2mil VALUES LESS THAN (3000000),
PARTITION p3mil VALUES LESS THAN (4000000),
PARTITION p4mil VALUES LESS THAN (5000000)
);
```

Minimum and Maximum values

- INFORMATION_SCHEMA.PARTITIONS
- What about a value >5,000,000 ?
- How can we fix that?

Minimum and Maximum values

- INFORMATION_SCHEMA.PARTITIONS
- What about a value >5,000,000 ?
- How can we fix that?
  - VALUES LESS THAN MAXVALUE

RANGE Partition Definition

index that goes across all partitions, does it do them in

```
CREATE TABLE nums_range_mod (
  id int NOT NULL AUTO_INCREMENT PRIMARY KEY)
    ENGINE=InnoDB
PARTITION BY RANGE (id % 4) (
  PARTITION zero VALUES LESS THAN (1),
  PARTITION one VALUES LESS THAN (2),
  PARTITION two VALUES LESS THAN (3),
  PARTITION three VALUES LESS THAN (4),
  PARTITION should_be_empty VALUES LESS THAN
    MAXVALUE);
```

LIST Partitioning

- Like RANGE, but exact value matches

  PARTITION BY LIST(expr)

  (PARTITION name VALUES IN (list))

- Order not important

- No catch-all

LIST Partition Definition

```
CREATE TABLE nums_list_mod (
  id int NOT NULL AUTO_INCREMENT PRIMARY KEY)
    ENGINE=InnoDB
PARTITION BY LIST (id % 4) (
  PARTITION zero VALUES IN (0),
  PARTITION one VALUES IN (1),
  PARTITION two VALUES IN (2),
  PARTITION three VALUES IN (3)
);
```

HASH Partitioning

- What to partition

- Number of partitions

PARTITION BY HASH (expr)

PARTITIONS x

HASH Partition Definition

```
CREATE TABLE nums_hash (
  id int NOT NULL AUTO_INCREMENT PRIMARY KEY)
    ENGINE=InnoDB
PARTITION BY HASH (id) PARTITIONS 4;
```

LINEAR HASH Partitioning

- Like HASH

- Does not use %

PARTITION BY LINEAR HASH (expr)

PARTITIONS x

KEY Partitioning Basics

- Like HASH, but field(s) only
- Part or all of PRIMARY KEY
- Can use non-integer/NULL fields

KEY Partitioning Algorithm

PARTITION BY KEY([fld1[, fld2]]...) PARTITIONS x

- Uses internal algorithm
- Storage engine dependent

KEY Partition Definition

```
CREATE TABLE nums_key (
  id int NOT NULL AUTO_INCREMENT PRIMARY KEY)
    ENGINE=InnoDB
PARTITION BY KEY() PARTITIONS 4;
```

LINEAR KEY Partitioning

- Like LINEAR HASH

PARTITION BY LINEAR KEY ([fld1 [, fld2]])

PARTITIONS x

Composite Partitioning

- Subpartition RANGE/LIST partitions...
- ...into HASH/KEY subpartitions

Composite Partitioning

```
CREATE TABLE nums_composite (
  id INT,  happened DATETIME NOT NULL,
PRIMARY KEY (id, happened)
) ENGINE=InnoDB PARTITION BY RANGE (HOUR(happened))
  SUBPARTITION BY HASH( id )
    SUBPARTITIONS 2 (
       PARTITION p0 VALUES LESS THAN (10),
       PARTITION p1 VALUES LESS THAN (20),
       PARTITION p2 VALUES LESS THAN MAXVALUE
    );
```

Loading the Data

```
mysql> INSERT INTO nums_composite (id, happened)
SELECT id,
'2009-01-01 00:00:00' + INTERVAL id HOUR FROM nums_hash;
Query OK, 500 rows affected (0.01 sec)
Records: 500  Duplicates: 0  Warnings: 0
```

Manual Subpartitions

- Can specify {DATA | INDEX} DIRECTORY
- Must specify name, for others, default
- Subpartition names unique across table

NULL

- RANGE = lowest value
- LIST = must contain VALUES IN (NULL)
- HASH/KEY = treated as 0

Removing Partitions/ing

ALTER TABLE tbl REMOVE PARTITIONING

ALTER TABLE tbl DROP PARTITION pname

- Only for named partitioning! LIST/RANGE

Changing Partitions

ALTER TABLE tbl PARTITION BY....

ALTER TABLE tbl
REORGANIZE PARTITION pname [, pname]
INTO (partition definition(s))

- REORGANIZE for LIST/RANGE only

Changing HASH/KEY Partitioning

ALTER TABLE tbl COALESCE PARTITION x

ALTER TABLE tbl ADD PARTITION PARTITIONS x

Partition Maintenance

  - ALTER TABLE tbl...
  -

- REBUILD PARTITION list
  - defrag

- OPTIMIZE PARTITION list
  - defrag, reclaim space

Partition Maintenance

  - ALTER TABLE tbl..

- CHECK PARTITION list
- REPAIR PARTITION list
- ANALYZE PARTITION list

EXPLAIN

- EXPLAIN PARTITIONS SELECT...

```
mysql> EXPLAIN PARTITIONS SELECT *
    -> FROM nums_range_mod limit 10\G
*************************** 1. row ***************************
id: 1
  select_type: SIMPLE
        table: nums_range_mod
   partitions: zero,one,two,three,should_be_empty
         type: index
possible_keys: NULL
          key: PRIMARY
      key_len: 4
          ref: NULL
         rows: 501
        Extra: Using index
1 row in set (0.00 sec)
```

Resources

- MySQL Manual
- Forums
- http://dev.mysql.com/doc/refman/5.1/en/partitioning-limitations.html

Sheeri Cabral

cabral@pythian.com

www.sheeri.com