1. **Hiểnthị partition trong 1 table**

[**EXPLAIN**](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.6%2Fen%2Fexplain.html&token=529b313489c664a355074ddb38591b79) **PARTITIONS** [**SELECT**](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.6%2Fen%2Fselect.html&token=529b313489c664a355074ddb38591b79) [COUNT](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.6%2Fen%2Fgroup-by-functions.html%23function_count&token=529b313489c664a355074ddb38591b79)( \* ) **FROM** t1

1. **Lấydữliệutừ 1 partition ra**

[**SELECT**](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.6%2Fen%2Fselect.html&token=529b313489c664a355074ddb38591b79) \* **FROM** t1**PARTITION** (p2)

-Với t1: tên table

- p2: têncủa partition (vídụ ở đâytênnólà p2)

**SELECT id, CONCAT(fname, ' ', lname) AS name**

**FROM employees PARTITION (p0) ORDER BY lname;**

**SELECT \* FROM employees PARTITION (p0, p2)**

**WHERE lname LIKE 'S%';**

**SELECT store\_id, COUNT(department\_id) AS c**

**FROM employees PARTITION (p1,p2,p3)**

**GROUP BY store\_id HAVING c > 4;**

1. **Tạo table có partition và primary key auto\_crement**

**CREATE TABLE employees\_sub (**

**id INT NOT NULL AUTO\_INCREMENT,**

**fname VARCHAR(25) NOT NULL,**

**lname VARCHAR(25) NOT NULL,**

**store\_id INT NOT NULL,**

**department\_id INT NOT NULL,**

**PRIMARY KEY pk (id, lname)**

**)**

**PARTITION BY RANGE(id)**

**SUBPARTITION BY KEY (lname)**

**SUBPARTITIONS 2 (**

**PARTITION p0 VALUES LESS THAN (5),**

**PARTITION p1 VALUES LESS THAN (10),**

**PARTITION p2 VALUES LESS THAN (15),**

**PARTITION p3 VALUES LESS THAN MAXVALUE**

**);**

1. **Tạo table**

[**CREATE** **TABLE**](http://thoitrangbui.com:2082/cpsess8874072846/3rdparty/phpMyAdmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.1%2Fen%2Fcreate-table.html&token=1c7b0390c57bd3e999add8968aba210d) employees(

id INT [**NOT**](http://thoitrangbui.com:2082/cpsess8874072846/3rdparty/phpMyAdmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.1%2Fen%2Flogical-operators.html%23operator_not&token=1c7b0390c57bd3e999add8968aba210d) **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY** ,  
fname VARCHAR( 25 ) [**NOT**](http://thoitrangbui.com:2082/cpsess8874072846/3rdparty/phpMyAdmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.1%2Fen%2Flogical-operators.html%23operator_not&token=1c7b0390c57bd3e999add8968aba210d) **NULL** ,  
lname VARCHAR( 25 ) [**NOT**](http://thoitrangbui.com:2082/cpsess8874072846/3rdparty/phpMyAdmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.1%2Fen%2Flogical-operators.html%23operator_not&token=1c7b0390c57bd3e999add8968aba210d) **NULL** ,  
store\_id INT [**NOT**](http://thoitrangbui.com:2082/cpsess8874072846/3rdparty/phpMyAdmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.1%2Fen%2Flogical-operators.html%23operator_not&token=1c7b0390c57bd3e999add8968aba210d) **NULL** ,  
department\_id INT [**NOT**](http://thoitrangbui.com:2082/cpsess8874072846/3rdparty/phpMyAdmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.1%2Fen%2Flogical-operators.html%23operator_not&token=1c7b0390c57bd3e999add8968aba210d) **NULL**

) **PARTITION** **BY** **RANGE** (

id

)(

**PARTITION** p0 **VALUES** LESS THAN( 5 ) ,   
**PARTITION** p1  
**VALUES** LESS THAN( 10 ) ,   
**PARTITION** p2  
**VALUES** LESS THAN( 15 ) ,   
**PARTITION** p3  
**VALUES** LESS THAN  
**MAXVALUE**

)

1. CREATE TABLE employees (
2. id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
3. fname VARCHAR(25) NOT NULL,
4. lname VARCHAR(25) NOT NULL,
5. store\_id INT NOT NULL,
6. department\_id INT NOT NULL
7. )
8. PARTITION BY RANGE(id) (
9. PARTITION p0 VALUES LESS THAN (5),
10. PARTITION p1 VALUES LESS THAN (10),
11. PARTITION p2 VALUES LESS THAN (15),
12. PARTITION p3 VALUES LESS THAN MAXVALUE
13. );
14. INSERT INTO employees VALUES
15. ('', 'Bob', 'Taylor', 3, 2), ('', 'Frank', 'Williams', 1, 2),
16. ('', 'Ellen', 'Johnson', 3, 4), ('', 'Jim', 'Smith', 2, 4),
17. ('', 'Mary', 'Jones', 1, 1), ('', 'Linda', 'Black', 2, 3),
18. ('', 'Ed', 'Jones', 2, 1), ('', 'June', 'Wilson', 3, 1),
19. ('', 'Andy', 'Smith', 1, 3), ('', 'Lou', 'Waters', 2, 4),
20. ('', 'Jill', 'Stone', 1, 4), ('', 'Roger', 'White', 3, 2),
21. ('', 'Howard', 'Andrews', 1, 2), ('', 'Fred', 'Goldberg', 3, 3),
22. ('', 'Barbara', 'Brown', 2, 3), ('', 'Alice', 'Rogers', 2, 2),
23. ('', 'Mark', 'Morgan', 3, 3), ('', 'Karen', 'Cole', 3, 2);

| **Region** | **Store ID Numbers** |
| --- | --- |
| North | 3, 5, 6, 9, 17 |
| East | 1, 2, 10, 11, 19, 20 |
| West | 4, 12, 13, 14, 18 |
| Central | 7, 8, 15, 16 |

To partition this table in such a way that rows for stores belonging to the same region are stored in the same partition, you could use the [**CREATE TABLE**](http://dev.mysql.com/doc/refman/5.1/en/create-table.html) statement 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 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)

);

**5. Sửađổicấutrúccủa partition**

**- Sửađổicấutrúctrong 1 partition:**

**ALTER TABLE employees**

->**REORGANIZE PARTITION p3 INTO (**

->**PARTITION p3 VALUES LESS THAN (20),**

->**PARTITION p4 VALUES LESS THAN (25),**

->**PARTITION p5 VALUES LESS THAN MAXVALUE**

->**);**

ALTER TABLE t1

PARTITION BY HASH(id)

PARTITIONS 8;

ALTERTABLE expenses

PARTITIONBY LIST COLUMNS (category)

(

PARTITION p01 VALUESIN ('lodging', 'food'),

PARTITION p02 VALUESIN ('flights', 'ground transportation'),

PARTITION p03 VALUESIN ('leisure', 'customer entertainment'),

PARTITION p04 VALUESIN ('communications'),

PARTITION p05 VALUESIN ('fees')

);

**SELECT count(\*) as ct,`categories`.`Title`,`categories`.`URL` FROM `tbraovat`,`categories` WHERE**

**`tbraovat`.`ID\_CategoriesFather`=`categories`.`ID`**

**AND `tbraovat`.`ID\_CategoriesFather` IN(1203,1204,1205,1208,1209)**

**GROUP BY `tbraovat`.`ID\_CategoriesFather`**

CREATE TABLE ti (id INT, amount DECIMAL(7,2), tr\_date DATE)

ENGINE=INNODB

PARTITION BY HASH( MONTH(tr\_date) )

PARTITIONS 6;

SELECTmytable.STATE,

SUM(IIF(mytable.survey\_answer = 'low', 1, 0) AS low,

SUM(IIF(mytable.survey\_answer = 'moderate', 1, 0) AS moderate,

SUM(IIF(mytable.survey\_answer = 'high', 1, 0) AS high

FROMmytable

GROUPBYmytable.state;

**I** finally found a way to do it, I answer for the sake of other people who come across this question

//dropauto\_increment capability

altertable table1 change column id id BIGINT NOTNULL;

//in one line, dropprimarykeyand rebuild one

altertable table1 dropprimarykey, addprimarykey(id,fk\_id,begin);

//re add the auto\_increment capability, last value is remembered

altertable table1 change column id id BIGINT NOTNULL AUTO\_INCREMENT;

//build the partition

altertable table1 partitionby range (begin) (

PARTITION until\_2010\_07 VALUES LESS THAN (1280620800000),

PARTITION2010\_08 VALUES LESS THAN (1283299200000),

PARTITION2010\_09 VALUES LESS THAN (1285891200000),

PARTITION2010\_10 VALUES LESS THAN (1288569600000),

PARTITION2010\_11 VALUES LESS THAN (1291161600000),

PARTITION2010\_12 VALUES LESS THAN (1293840000000),

PARTITION from\_2011 VALUES LESS THAN MAXVALUE

);

**Công ty TNHH Công nghệ và truyền thông tam nguyên**

**Phong 1001 nhà OTC 3D khu đô thị cổ nhuế, xã xuân đỉnh, huyện từ liêm, hà nội**

**MST: 0102.957.334**