

SQL Advanced Topics

Manpreet S. Katari

Outline

- User variables
- SQL Routines
- SQL Triggers
- NO SQL

<https://orion.bio.nyu.edu/phpmyadmin/>

Username: ga1009

Password: mkatari@nyu

Setting and Selecting variables

```
SET @var_name = expr [, @var_name = expr] ...
```

```
mysql> SET @t1=1, @t2=2, @t3:=4;
mysql> SELECT @t1, @t2, @t3, @t4 := @t1+@t2+@t3;
+-----+-----+-----+-----+
| @t1  | @t2  | @t3  | @t4 := @t1+@t2+@t3 |
+-----+-----+-----+-----+
|     1 |     2 |     4 |          7 |
+-----+-----+-----+-----+
```

In select statements, you have to assign using `:=` because `=` by itself is a comparison in select statements.

SQL Routines

- There are two main types of Routines:
 - Procedures
 - Functions
- Procedures
 - A procedure is executed using a CALL function and only return the output value
- Function
 - Just like functions in programming languages, can be called from any part of the code or statement.

```
CREATE  
    [DEFINER = { user | CURRENT_USER }]  
    PROCEDURE sp_name ([proc_parameter[,...]])  
    [characteristic ...] routine_body
```

```
CREATE  
    [DEFINER = { user | CURRENT_USER }]  
    FUNCTION sp_name ([func_parameter[,...]])  
    RETURNS type  
    [characteristic ...] routine_body
```

proc_parameter:

```
[ IN | OUT | INOUT ] param_name type
```

func_parameter:

```
param_name type
```

type:

```
Any valid MySQL data type
```

Procedure Example

```
DROP PROCEDURE `feature_nrows`;  
CREATE DEFINER=`msk8`@`%` PROCEDURE `feature_nrows`(OUT `nrow` INT(255) UNSIGNED)  
NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER  
BEGIN  
SELECT COUNT(*) INTO nrow FROM feature;  
END
```

Edit routine



Details

Routine name

Type

PROCEDURE

Parameters

Direction	Name	Type	Length/Values	Options
OUT	nrow	INT	255	UNSIGNED Drop

Add parameter

```
1 BEGIN
2   SELECT COUNT(*) INTO nrow FROM feature;
3 END
```

Definition

Is deterministic



Definer

Security type

DEFINER



SQL data access

NO SQL



Comment

Go

Close

Function Example

```
DROP FUNCTION `gene_chr`;  
CREATE DEFINER=`msk8`@`%` FUNCTION `gene_chr`(`genename` VARCHAR(255)) RETURNS  
VARCHAR(255) NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER  
BEGIN  
DECLARE chromosome VARCHAR(255);  
select seqname into chromosome from locationlist, feature, name  
where locationlist.id = feature.seqid and name.id = feature.id  
and name.name = genename; return (chromosome);  
END
```

Edit routine

Details

Routine name

Type

FUNCTION

Parameters

Name	Type	Length/Values	Options
genename	VARCHAR	255	Chars <input type="button" value="Drop"/>

Add parameter

Return type

VARCHAR

Return length/values

255

Return options

Charset

```
1 BEGIN
2   DECLARE chromosome VARCHAR(255);
3   select seqname into chromosome
4   from locationlist, feature, name
5   where locationlist.id = feature.seqid
6   and name.id = feature.id
7   and name.name = genename;
8
9   return (chromosome);
10 END
```

Definition

Is deterministic

Definer

msk8@%

Security type

DEFINER

SQL data access

NO SQL

Comment

Go

Close

To run the routines, simply click on execute

Name	Action	Type	Returns
feature_nrows	Edit Execute Export Drop PROCEDURE		
gene_chr	Edit Execute Export Drop FUNCTION		varchar(255)

Creating triggers

CREATE

[DEFINER = { **user** | CURRENT_USER }]

TRIGGER **trigger_name**

trigger_time trigger_event

ON **tbl_name** FOR EACH ROW

[**trigger_order**]

trigger_body

trigger_time: { BEFORE | AFTER }

trigger_event: { INSERT | UPDATE | DELETE }

trigger_order: { FOLLOWS | PRECEDES } **other_trigger_name**

Example use case of trigger

```
mysql> CREATE TABLE account (acct_num INT, amount DECIMAL(10,2));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TRIGGER ins_sum BEFORE INSERT ON account
-> FOR EACH ROW SET @sum = @sum + NEW.amount;
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> SET @sum = 0;
mysql> INSERT INTO account VALUES(137,14.98),(141,1937.50),(97,-100.00);
mysql> SELECT @sum AS 'Total amount inserted';
+-----+
| Total amount inserted |
+-----+
| 1852.48               |
+-----+
```



Recent Favorites



- + New
- + bp56
- + godb
- + hg19
- + information_schema
- + msk8
- + mysql
- + performance_schema
- tair10
 - + Functions
 - + Procedures
 - + Tables
 - + New
 - + attribute
 - + attributelist
 - + feature
 - + Columns
 - + Indexes
 - + New

← Server: localhost » Database: tair10 » Table: attribute

Browse Structure SQL Search Insert Export Import Privileges

Showing rows 0 - 24 (369504 total, Query took 0.0007 seconds.)

```
SELECT * FROM `attribute`
```

Profiling [Edit inline] [Edit] [Explain SQL]

1 > >> Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

				id	attribute_id	attribute_value
<input type="checkbox"/>				1	1	Chr1
<input type="checkbox"/>				2	1	AT1G01010
<input type="checkbox"/>				2	2	protein_coding_gene
<input type="checkbox"/>				3	1	AT1G01010.1
<input type="checkbox"/>				3	3	AT1G01010
<input type="checkbox"/>				4	1	AT1G01010.1-Protein
<input type="checkbox"/>				4	4	AT1G01010.1



Recent

Favorites

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#)[Table structure](#)[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)			No	None		Change Drop Primary Unique Index
2	attribute_id	int(10)			No	None		Change Drop Primary Unique Index
3	attribute_value	text	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index

[Check All](#) With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)[Print view](#) [Propose table structure](#) [Move columns](#) [Improve table structure](#) [Add](#) column(s) [after attribute_value](#) [Go](#)[+ Indexes](#)**Information**

Space usage	
Data	15.5 MiB
Index	0 B
Total	15.5 MiB

Row statistics	
Format	Compact
Collation	latin1_swedish_ci
Creation	Mar 06, 2018 at 06:02 PM



Recent Favorites

- + New
- + bp56
- + godb
- + hg19
- + information_schema
- + msk8
- + mysql
- + performance_schema
- tair10
 - + Functions
 - + Procedures
 - + Tables
 - + New
 - + attribute
 - + attributelist
 - + feature
 - + Columns
 - + Indexes
 - + New
 - + PRIMARY

Server: localhost » Database: tair10 » Table: attribute

Browse Structure SQL Search Insert Export Import Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)			No	None		Change Drop Primary Unique Index
2	attribute_id	int(10)			No	None		Change Drop Primary Unique Index
3	attribute_value	text	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index

Check All With selected: Browse Change Drop Primary Unique Index

Print view Propose table structure Move columns Improve table structure

Add 1 column(s) after attribute_value Go

- Indexes

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	369885	A	No	
					attribute_id	369885	A	No	

Create an index on 1 columns



Recent

Favorites



Browse

Structure

SQL

Search

Insert

Export

Import

Privileges



Show query box

Showing rows 0 - 1 (2 total, Query took 0.4502 seconds.)

```
SELECT * FROM `attribute` WHERE `attribute_value` = "AT1G77710"
```

Profiling [Edit inline] [Edit] [Explain SQL]

Show all

Number of rows:

25

Filter rows:

Search this table

Sort by key: None

+ Options



id attribute_id attribute_value

1 AT1G77710

3 AT1G77710



Edit



Copy



Delete

148747



Edit



Copy



Delete

148748

 Check All

With selected:



Edit



Delete



Show all

Number of rows:

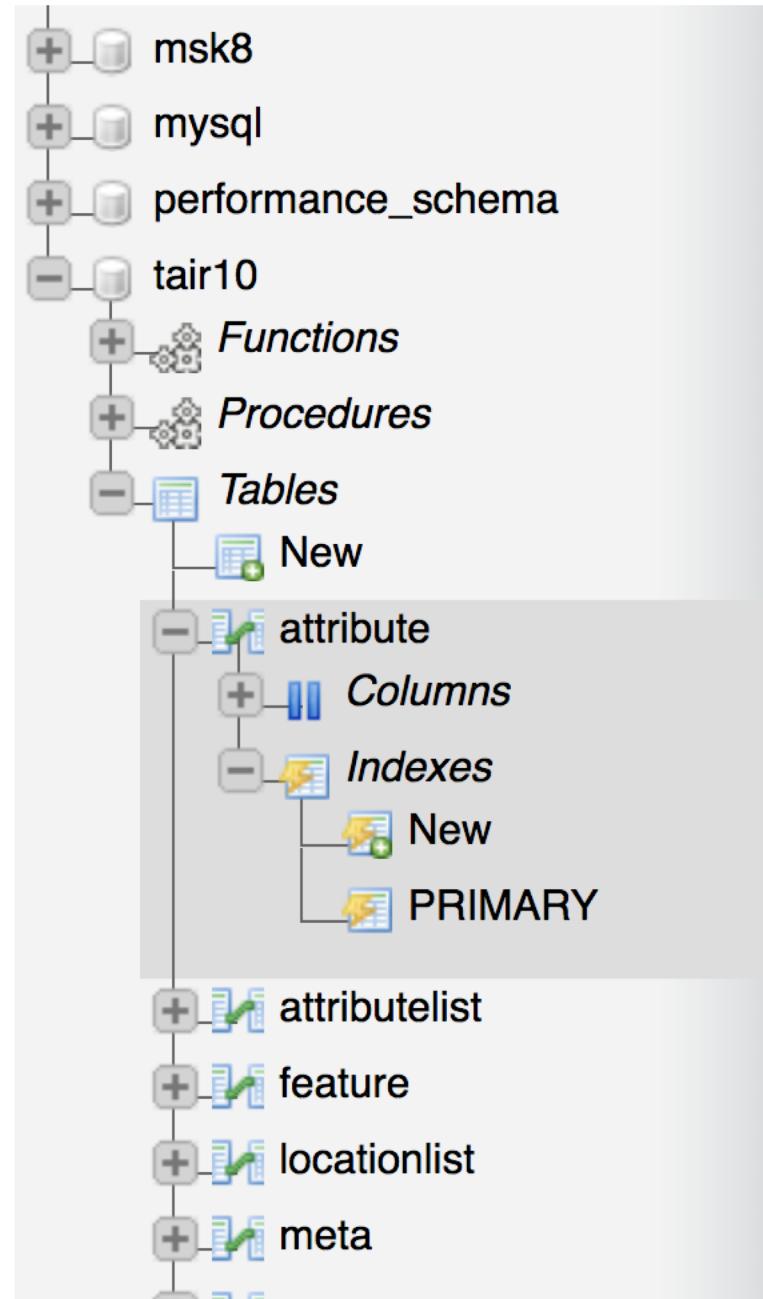
25

Filter rows:

Search this table

Query results operations

- New
- bp56
- godb
- hg19
- information_schema
- msk8
- mysql
- performance_schema
- tair10
 - Functions
 - Procedures
 - Tables
 - New
 - attribute
 - attributelist
 - feature
 - Columns
 - Indexes
 - New
 - PRIMARY



Add Index

Index name: AttributeValue

Index choice: INDEX

[+ Options](#)

Column	Size
attribute_value [text]	10
attribute_id [int(10)]	

Add 1 column(s) to index

Go Preview SQL Cancel



Recent Favorites



- New
- bp56
- godb
- hg19
- information_schema
- msk8
- mysql
- performance_schema
- tair10
 - Functions
 - Procedures
 - Tables
 - New
 - attribute
 - Columns
 - Indexes
 - New
 - AttributeValue
 - PRIMARY

Show query box

Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

```
SELECT * FROM `attribute` WHERE `attribute_value` = "AT1G77710"
```

Profiling [Edit inline] [Edit] [Explain SQL]

Show all

Number of rows:

25

Filter rows:

Search this table

Sort by key: None

+ Options



id

attribute_id

attribute_value



148747

1

AT1G77710



148748

3

AT1G77710



Check All

With selected:



Show all

Number of rows:

25

Filter rows:

Search this table



Recent

Favorites



- New
- + bp56
- + godb
- + hg19
- + information_schema
- + msk8
- + mysql
- + performance_schema
- tair10
 - + Functions
 - + Procedures
 - Tables
 - New
 - attribute
 - + Columns
 - + Indexes
 - New
 - AttributeValue
 - PRIMARY

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Your SQL query has been executed successfully.

```
EXPLAIN SELECT * FROM `attribute` WHERE `attribute_value` = "AT1G77710"
```

[Edit inline] [Edit] [Skip Explain SQL]

+ Options

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	attribute	ref	AttributeValue	AttributeValue	13	const	2	Using where

Query results operations

[Print view](#) [Print view \(with full texts\)](#) [Create view](#)

NoSQL – non SQL or non relational

- Different types of structures:
 - Key-value
 - Wide column
 - Graph (Neo4J)
 - Document (MongoDB)
- Motivation is that many of the high performance application need databases to perform only selects with very few joins.
- Thus there is no need to keep relationships between data objects and there is a lot of repetition, but disk space is cheap.

MongoDB (Document)

- <http://www.mongodb.org>
- Data is stored in some standard format, for example JSON/BSON.
- No Schemas
- No Foreign Keys

SQL Terms/Concepts

database

table

row

column

index

table joins

primary key

Specify any unique column or column combination as primary key.

MongoDB Terms/Concepts

database

collection

document or **BSON** document

field

index

embedded documents and linking

primary key

In MongoDB, the primary key is automatically set to the `_id` field.

Comparison Examples

```
CREATE TABLE users (
    id MEDIUMINT NOT NULL
        AUTO_INCREMENT,
    user_id Varchar(30),
    age Number,
    status char(1),
    PRIMARY KEY (id)
)
```

Implicitly created on first `insert()` operation. The primary key `_id` is automatically added if `_id` field is not specified.

```
db.users.insert( {
    user_id: "abc123",
    age: 55,
    status: "A"
} )
```

SQL INSERT Statements

```
INSERT INTO users(user_id,  
                  age,  
                  status)  
VALUES ("bcd001",  
       45,  
      "A")
```

MongoDB insert() Statements

```
db.users.insert(  
    { user_id: "bcd001", age: 45, status: "A" })
```

```
SELECT *  
FROM users  
WHERE status = "A"  
OR age = 50
```

```
db.users.find(  
    { $or: [ { status: "A" } ,  
            { age: 50 } ] })
```

```
SELECT *  
FROM users  
WHERE user_id like "%bc%"
```

```
db.users.find( { user_id: /bc/ } )
```