

Tritium MySQL Database How To

Tyler Hague

How to connect

- Log into an a-onl machine and type:
`mysql -u triton-user -p -h halladb triton`

What tables are there?

```
MySQL [triton]> SHOW TABLES;
+-----+
| Tables_in_triton |
+-----+
| COMMISSIONINGrunlist |
| EKrunlist |
| EPrunlist |
| MARATHONrunlist |
| PRECOMMISSIONINGrunlist |
| SRCrunlist |
| TESTrunlist |
+-----+
7 rows in set (0.01 sec)
```

What is in each table?

- run_number
- run_type
- start_time
- end_time
- target
- raster_x
- raster_y
- beam_energy
- momentum
- angle
- prescale_T1
- prescale_T2
- prescale_T3
- prescale_T4
- prescale_T5
- prescale_T6
- prescale_T7
- prescale_T8
- comment
- end_comment

Let's take a look at a table

- Here's the PRECOMMISSIONINGrunlist table

```
MySQL [triton]> SELECT * FROM PRECOMMISSIONINGrunlist;
```

run_number	run_type	start_time	end_time	target	raster_x	raster_y	beam_energy	momentum	angle	prescale_T1	prescale_T2	prescale_T3	prescale_T4	prescale_T5	prescale_T6	prescale_T7	prescale_T8	comment	
643	Cosmics	2017-12-01 17:50:28	2017-12-01 18:01:07	Titanium	2.00	2.00	4318.4902	0.0026261699	17.005400					1					
0	0	0	0	cosmics T1-T2-T3=1, 2x2 raster test						1			1						
90386	Cosmics	2017-12-01 17:50:31	2017-12-02 14:05:05	Titanium	2.00	2.00	4318.4902	0.0036617301	0.197000	0	0	0	0	1					
1	1	0	0	Cosmics T4-T5-T6=1															
644	Cosmics	2017-12-01 18:12:13	2017-12-02 14:04:45	Titanium	2.00	2.00	4318.4902	0.0026317399	17.005400	1	1	1	1	0					
0	0	0	0	cosmics T1-T2-T3=1, 2x2 raster test															
90387	Cosmics	2017-12-02 15:27:08	2017-12-02 15:47:42	Titanium	2.00	2.00	4318.4902	0.0000000000	0.197000	0	0	0	0	1					
1	1	0	0	Cosmics T4-T5-T6=1															
90388	Cosmics	2017-12-02 15:48:33	2017-12-02 15:49:06	Titanium	2.00	2.00	4318.4902	0.0000000000	0.197000	0	0	0	0	1					
1	1	0	100	Cosmics T4-T5-T6=1															
90389	Cosmics	2017-12-02 16:19:23	2017-12-02 16:24:52	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	0	0	0	0	1					
1	1	0	100	Cosmics T4-T5-T6=1															
90390	Cosmics	2017-12-02 16:25:17	2017-12-02 16:35:02	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	0	0	0	0	1					
1	1	0	1000	Cosmics T4-T5-T6=1															
90391	Cosmics	2017-12-02 16:44:14	2017-12-02 16:47:51	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	500	Cosmics T4-T5-T6=1															
90392	Cosmics	2017-12-02 16:51:21	2017-12-02 16:56:49	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	500	Cosmics T1-T2-T4-T5-T6=1 & T8-500															
90393	Cosmics	2017-12-02 16:57:10	2017-12-02 17:13:40	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	143	Cosmics T1-T2-T4-T5-T6=1 & T8-500															
90394	Cosmics	2017-12-02 17:14:12	2017-12-02 17:25:43	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	83	Cosmics T1-T2-T4-T5-T6=1 & T8-143															
90395	Cosmics	2017-12-02 17:26:30	2017-12-02 17:34:53	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	59	Cosmics T1-T2-T4-T5-T6=1 & T8-83															
90396	Cosmics	2017-12-02 17:35:30	2017-12-02 17:42:32	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	45	Cosmics T1-T2-T4-T5-T6=1 & T8-59															
90397	Cosmics	2017-12-02 17:43:06	2017-12-02 17:50:44	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	37	Cosmics T1-T2-T4-T5-T6=1 & T8-45															
645	Cosmics	2017-12-02 17:45:29	NULL	Titanium	2.00	2.00	4318.4902	0.0026696001	17.005400	1	1	1	1	0					
0	0	0	0	cosmics T1-T2-T3=1															
90398	Cosmics	2017-12-02 17:52:16	2017-12-02 17:56:58	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	37	Cosmics T1-T2-T4-T5-T6=1 & T8-37															
90399	Cosmics	2017-12-02 17:57:40	2017-12-02 18:03:31	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	59	Cosmics T1-T2-T4-T5-T6=1 & T8-37 (with veto)															
90400	Cosmics	2017-12-02 18:04:30	2017-12-02 18:06:15	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	59	Cosmics T1-T2-T4-T5-T6=1 & T8-59 (with veto)															
90401	Cosmics	2017-12-02 18:06:40	NULL	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	143	Cosmics T1-T2-T4-T5-T6=1 & T8-59 (with veto)															
90402	Cosmics	2017-12-02 18:19:56	2017-12-02 18:28:22	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	143	Cosmics T1-T2-T4-T5-T6=1 & T8-143 (with veto)															
90403	Cosmics	2017-12-02 18:30:54	2017-12-02 18:33:28	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	143	Cosmics T1-T2-T4-T5-T6=1 & T8-143 (with veto)															
90404	Cosmics	2017-12-02 18:34:04	2017-12-02 18:34:46	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	1	1	1	0	1					
1	1	0	45	Cosmics T1-T2-T4-T5-T6=1 & T8-143															
90405	Cosmics	2017-12-02 18:35:22	2017-12-03 12:58:53	Titanium	2.00	2.00	4318.4902	1.0000000000	0.197000	0	0	0	0	1					
1	1	0	0	Cosmics T1-T2-T4-T5-T6=1 & T8-45															
646	Cosmics	2017-12-03 13:52:58	2017-12-03 14:43:43	Titanium	2.00	2.00	4318.4902	0.0030039200	17.005400	1	1	1	1	0					
0	0	0	0	cosmics T1-T2-T3=1, 2x2 raster															
90406	Cosmics	2017-12-03 13:54:27	2017-12-03 14:46:18	Titanium	2.00	2.00	4318.4902	0.0167568000	0.197000	0	0	0	0	1					
1	1	0	0	Cosmics T4-T5-T6=1 (2x2 raster test)															

Oh no... that's pretty overwhelming. Let's see how we can cut down on this info.

I want to see some runs with the clock on

- I want to know what runs have the clock on. Also, I'm only interested in the run number and target. How do I do that?

I want to see some runs with the clock on

- I want to know what runs have the clock on. Also, I'm only interested in the run number and target. How do I do that?

```
MySQL [triton]> SELECT run_number, target FROM PRECOMMISSIONINGrunlist WHERE prescale_T8>0;
```

run_number	target
90388	Titanium
90389	Titanium
90390	Titanium
90391	Titanium
90392	Titanium
90393	Titanium
90394	Titanium
90395	Titanium
90396	Titanium
90397	Titanium
90398	Titanium
90399	Titanium
90400	Titanium
90401	Titanium
90402	Titanium
90403	Titanium
90404	Titanium
90419	Titanium

```
18 rows in set (0.00 sec)
```

Much better.

Another example

- I noticed something weird in some data for run 90418. Now I want to see if the shifters put any comment info and look at logbook entries around the same time. What ever am I to do?

Another example

- I noticed something weird in some data for run 90418. Now I want to see if the shifters put any comment info and look at logbook entries around the same time. What ever am I to do?

```
MySQL [triton]> SELECT start_time, end_time, comment, end_comment FROM PRECOMMISSIONINGrunlist WHERE run_number=90418;
+-----+-----+-----+-----+
| start_time | end_time | comment | end_comment |
+-----+-----+-----+-----+
| 2017-12-03 17:06:41 | 2017-12-03 17:11:13 | run(90418) Cosmics T4=T5=T6=1 | run(90418) Cosmics T4=T5=T6=1 extra comment here :-)
```

Now I can narrow down a time window in the logbook to search within.
I'm so glad the shifter was on the ball and gave me and extra comment!

What if I don't want it sorted by run number?

- I want to see run number and target info for the clock trigger again. But now I want it sorted by the prescale value of the clock trigger. Surely I don't have to keep scrolling until I find the highest clock rate?!

What if I don't want it sorted by run number?

- I want to see run number and target info for the clock trigger again. But now I want it sorted by the prescale value of the clock trigger. Surely I don't have to keep scrolling until I find the highest clock rate?!

```
MySQL [triton]> SELECT run_number, target, prescale_T8 FROM PRECOMMISSIONINGrunlist WHERE prescale_T8>0 ORDER BY prescale_T8 ASC;
```

run_number	target	prescale_T8
90397	Titanium	37
90398	Titanium	37
90396	Titanium	45
90404	Titanium	45
90395	Titanium	59
90399	Titanium	59
90400	Titanium	59
90419	Titanium	59
90394	Titanium	83
90388	Titanium	100
90389	Titanium	100
90393	Titanium	143
90401	Titanium	143
90402	Titanium	143
90403	Titanium	143
90391	Titanium	500
90392	Titanium	500
90390	Titanium	1000

18 rows in set (0.00 sec)

What if I don't want it sorted by run number?

- I want to see run number and target info for the clock trigger again. But now I want it sorted by the prescale value of the clock trigger. Surely I don't have to keep scrolling until I find the highest clock rate?!

```
MySQL [triton]> SELECT run_number, target, prescale_T8 FROM PRECOMMISSIONINGrunlist WHERE prescale_T8>0 ORDER BY prescale_T8 ASC;
```

run_number	target	prescale_T8
90397	Titanium	37
90398	Titanium	37
90396	Titanium	45
90404	Titanium	45
90395	Titanium	59
90399	Titanium	59
90400	Titanium	59
90419	Titanium	59
90394	Titanium	83
90388	Titanium	100
90389	Titanium	100
90393	Titanium	143
90401	Titanium	143
90402	Titanium	143
90403	Titanium	143
90391	Titanium	500
90392	Titanium	500
90390	Titanium	1000

18 rows in set (0.00 sec)

... But I wanted slowest clock rate first ☹️

There's a command for that.

There's a command for that.

```
MySQL [triton]> SELECT run_number, target, prescale_T8 FROM PRECOMMISSIONINGrunlist WHERE prescale_T8>0 ORDER BY prescale_T8 DESC;
```

run_number	target	prescale_T8
90390	Titanium	1000
90391	Titanium	500
90392	Titanium	500
90393	Titanium	143
90401	Titanium	143
90402	Titanium	143
90403	Titanium	143
90388	Titanium	100
90389	Titanium	100
90394	Titanium	83
90395	Titanium	59
90399	Titanium	59
90400	Titanium	59
90419	Titanium	59
90396	Titanium	45
90404	Titanium	45
90397	Titanium	37
90398	Titanium	37

```
18 rows in set (0.00 sec)
```

There's a command for that.

```
MySQL [triton]> SELECT run_number, target, prescale_T8 FROM PRECOMMISSIONINGrunlist WHERE prescale_T8>0 ORDER BY prescale_T8 DESC;
```

run_number	target	prescale_T8
90390	Titanium	1000
90391	Titanium	500
90392	Titanium	500
90393	Titanium	143
90401	Titanium	143
90402	Titanium	143
90403	Titanium	143
90388	Titanium	100
90389	Titanium	100
90394	Titanium	83
90395	Titanium	59
90399	Titanium	59
90400	Titanium	59
90419	Titanium	59
90396	Titanium	45
90404	Titanium	45
90397	Titanium	37
90398	Titanium	37

```
18 rows in set (0.00 sec)
```

#EasyPeasy

In conclusion

SQL is a powerful tool for storing and searching through a vast quantity of data. A python script, triggered by starting and stopping CODA, ensures that our runlist is kept up to date and ready to help us through our analysis.

Through the use of SELECT commands, you can find any type of run that you need.

This is only a brief overview of a few things that you can do with SQL. Google has a vast wealth of knowledge available to teach you how to query SQL as much as your heart desires.

The account that you use to access the database is read-only. MySQL doesn't confirm commands. An errant command could delete all info in the database without batting an eye.

Now go forth and run your experiment. And don't forget the semicolons. ;)

But I really want to make changes/add info/do something that requires write-privileges

- If that's the case, you should install mysql on your personal machine.
- Individual tables (or the whole database) can be exported and then imported into another mysql installation.
- If you want to have something exported for you to take and play with, let me know.
- I will try to export individual experiment tables after each run period. If I forget (which I probably will), remind me.