

Herramientas Útiles del Percona Toolkit para DBAs



Fernando Mattera - Percona



Buenos Aires, Argentina 2023

PERCONA
UNIVERSITY



Agenda

- Introducción
- Instalación
- Herramientas disponibles
- Herramientas que salvarán tu día
- Recursos
- Preguntas

Introducción

- Conjunto de herramientas de línea de comando.
- Gratuito y de código abierto.
- Desarrollado y soportado por Percona
- Herramientas para MySQL, Postgres y MongoDB
- Indispensable para DBAs



Instalación

Configurar el repositorio de Percona

RHEL/CentOS

```
$ sudo yum install -y https://repo.percona.com/yum/percona-release-latest.noarch.rpm
```

Ubuntu/Debian

```
$ sudo apt update  
$ sudo apt install curl  
$ curl -O https://repo.percona.com/apt/percona-release_latest.generic_all.deb  
$ sudo apt install gnupg2 lsb-release ./percona-release_latest.generic_all.deb  
$ sudo apt update
```

Instalar Percona Toolkit

```
$ sudo apt-get install percona-toolkit # (Ubuntu/Debian)  
  
$ sudo yum install -y percona-toolkit # (RHEL/CentOS)
```


Herramientas disponibles

- **pt-align**
- **pt-archiver**
- **pt-config-diff**
- pt-deadlock-logger
- pt-diskstats
- **pt-duplicate-key-checker**
- pt-fifo-split
- pt-find
- pt-fingerprint
- pt-fk-error-logger
- pt-heartbeat
- pt-index-usage
- pt-ioprofile
- pt-k8s-debug-collector
- **pt-kill**
- pt-mext
- pt-mongodb-index-check
- pt-mongodb-query-digest
- pt-mongodb-summary
- **pt-mysql-summary**
- **pt-online-schema-change**
- pt-pg-summary
- pt-pmp
- pt-query-digest
- pt-secure-collect
- **pt-show-grants**
- pt-sift
- pt-slave-delay
- pt-slave-find
- pt-slave-restart
- pt-stalk
- **pt-summary**
- pt-table-checksum
- pt-table-sync
- pt-table-usage
- pt-upgrade
- **pt-variable-advisor**
- **pt-visual-explain**



pt-align

- Permite emprolijar/alinear/dar mejor formato a un archivo para armar algún tipo de reporte sin recurrir a artilugios tales como grep/awk/sed/etc.
- Muy fácil de usar.

pt-align - ejemplo

Supongamos que tenemos este archivo

```
$ cat /tmp/random_file.txt
```

IP_ADDRESS	SERVER_CLUB_GRANDE	COPAS_GANADAS
=====	=====	=====
1.2.3.4	server.quilmesaclub.com	100
2.2.3.4	server.bocajuniors.com	33
3.2.3.4	server.riverplate.com	23
4.2.3.4	server.racingclub.com	-10
5.2.3.4	server.sanlorenzo.com	8
6.2.3.4	server.independiente.com	4

pt-align - ejemplo

Lo podemos “alinear” de la siguiente forma, utilizando **pt-align**

```
$ cat /tmp/random_file.txt | pt-align
```

IP_ADDRESS	SERVER_CLUB_GRANDE	COPAS_GANADAS
=====	=====	=====
1.2.3.4	server.quilmesaclub.com	100
2.2.3.4	server.bocajuniors.com	33
3.2.3.4	server.riverplate.com	23
4.2.3.4	server.racingclub.com	-10
5.2.3.4	server.sanlorenzo.com	8
6.2.3.4	server.independiente.com	4



pt-config-diff

- Permite comparar la diferencia en los parámetros entre dos instancias, o entre los parámetros “vivos” (online) de una instancia y los persistidos en el archivo de configuración
- Para esto último, todas las variables deben estar en un solo archivo de configuración (ejemplo: `/etc/my.cnf`)
- Muy fácil de usar.

pt-config-diff - ejemplo (comparando dos instancias MySQL)

```
$ pt-config-diff h=192.168.0.14 h=192.168.0.59 --report-width=1500 --noverison-check
```

16 config differences

Variable	primary	replica
audit_log_policy	NONE	ALL
event_scheduler	ON	OFF
general_log_file	/var/lib/mysql/master.log	/var/lib/mysql/replica.log
hostname	master	replica
innodb_flush_log_at_trx_commit	1	0
innodb_log_file_size	209715200	50331648
read_only	OFF	ON
relay_log_space_limit	104857600	0
report_host	192.168.0.14	192.168.0.59
server_id	1	2
server_uuid	d3784d76-9dc9-11ea-8155-080027762df4	4a12b992-9002-11eb-bacf-080027eae162
slow_query_log_file	/var/lib/mysql/master-slow.log	/var/lib/mysql/replica-slow.log
super_read_only	OFF	ON
sync_binlog	1	0
userstat	ON	OFF

pt-config-diff - ejemplo (parametros online y persistidos)

```
$ pt-config-diff h=localhost /etc/my.cnf --noverison-check
```

4 config differences

Variable	primary	/etc/my.cnf
=====	=====	=====
binlog_format	ROW	MIXED
expire_logs_days	0	8
open_files_limit	5000	65535
read_only	OFF	1

ALERT! Eso quiere decir que necesita sincronizarse lo que está online con lo persistido 😊



pt-variable-advisor

- examina la salida del comando “SHOW VARIABLES” en busca de valores y configuraciones incorrectos de acuerdo con las “REGLAS” que se describen a continuación. Informa sobre variables que coinciden con las reglas, por lo que puede encontrar configuraciones incorrectas en su servidor MySQL.

pt-variable-advisor - ejemplo

```
$ pt-variable-advisor h=192.168.0.14
```

```
# NOTE max_binlog_size: The max_binlog_size is smaller than the default of 1GB.
# WARN relay_log_space_limit: Setting relay_log_space_limit can cause replicas to stop
fetching binary logs from their master immediately.
# NOTE sort_buffer_size-1: The sort_buffer_size variable should generally be left at its
default unless an expert determines it is necessary to change it.
# WARN expire_logs_days: Binary logs are enabled, but automatic purging is not enabled.
# NOTE innodb_data_file_path: Auto-extending InnoDB files can consume a lot of disk space
that is very difficult to reclaim later.
# NOTE innodb_flush_method: Most production database servers that use InnoDB should set
innodb_flush_method to O_DIRECT to avoid double-buffering, unless the I/O system is very low
performance.
# WARN myisam_recover_options: myisam_recover_options should be set to some value such as
BACKUP,FORCE to ensure that table corruption is noticed.
```


pt-duplicate-key-checker

- Detecta índices duplicados e incluso índices de un solo campo que en otros índices son el primer campo, con lo cual hacen al primero un subset del último, por lo tanto candidato a la eliminación 🌀🌀🌀
- Muy fácil de usar.

pt-duplicate-key-checker - ejemplo

```
CREATE TABLE mitabla (  
  id int(11) NOT NULL AUTO_INCREMENT,  
  nombre varchar(50) DEFAULT NULL,  
  edad int(11) DEFAULT NULL,  
  email varchar(100) DEFAULT NULL,  
  col_1 int(11) DEFAULT NULL,  
  col_2 int(11) DEFAULT NULL,  
  otro_campo varchar(255) DEFAULT NULL,  
  PRIMARY KEY (id),  
  KEY idx_indice_repetido_1 (col_1),  
  KEY idx_indice_repetido_2 (col_1),  
  KEY idx_indice_repetido_3 (col_1,col_2)  
  ) ENGINE=InnoDB;
```

pt-duplicate-key-checker - ejemplo

```
$ pt-duplicate-key-checker --noversion-check --databases test
```

```
# #####  
# test.mitabla  
# #####
```

```
# idx_indice_repetido_1 is a left-prefix of idx_indice_repetido_3
```

```
# Key definitions:
```

```
# KEY `idx_indice_repetido_1` (`col_1`),
```

```
# KEY `idx_indice_repetido_3` (`col_1`,`col_2`)
```

```
# Column types:
```

```
# `col_1` int(11) default null
```

```
# `col_2` int(11) default null
```

```
# To remove this duplicate index, execute:
```

```
ALTER TABLE `test`.`mitabla` DROP INDEX `idx_indice_repetido_1`;
```

```
(sigue...)
```

pt-duplicate-key-checker - ejemplo

```
# idx_indice_repetido_2 is a left-prefix of idx_indice_repetido_3
# Key definitions:
#   KEY `idx_indice_repetido_2` (`col_1`),
#   KEY `idx_indice_repetido_3` (`col_1`,`col_2`)
# Column types:
#   `col_1` int(11) default null
#   `col_2` int(11) default null
# To remove this duplicate index, execute:
ALTER TABLE `test`.`mitabla` DROP INDEX `idx_indice_repetido_2`;
```

```
# #####
# Summary of indexes
# #####
```

```
# Size Duplicate Indexes      10000
# Total Duplicate Indexes     2
# Total Indexes                4
```

pt-kill

- Permite matar sesiones bajo ciertos criterios.
- Agrupar las sesiones para poder terminarlalas.
- Puede mostrar lo que matará antes de hacerlo (--print)
- Permite registrar sus acciones.



pt-kill - ejemplo

```
mysql> begin;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> update mitabla set email='fernando@qac.com';  
... query is running
```


pt-kill - ejemplo

```
$ /usr/bin/pt-kill --busy-time=5 --match-user=percona \  
--match-db=test --interval=2 --kill --print \  
--match-info="UPDATE"
```

```
# 2023-09-29T10:09:12 KILL 44624 (Sleep 24 sec) NULL
```

Donde

- **--busy-time** : Establece el tiempo en segundos que debe transcurrir antes de que pt-kill considere que una consulta está "ocupada" y la marque para matar si es necesario.
- **--match-user** : Filtra las consultas para que solo afecten al usuario "percona".
- **--match-db** : Filtra las consultas para que solo afecten a la base de datos "test".
- **--interval** : Establece el intervalo en segundos entre cada ejecución de pt-kill.
- **--match-info** : Filtra el comando que se está ejecutando.

Se puede agregar **--verbose** (para ver más detalles)

pt-mysql-summary / pt-summary

- Muestran un reporte de la instancia de base de datos a nivel estados de base de datos (pt-mysql-summary) y las características del servidor (pt-summary)
- Muy fácil de usar.

pt-mysql-summary - ejemplo

```
$ pt-mysql-summary
```

```
# Percona Toolkit MySQL Summary Report #####
      System time | 2023-09-29 14:36:05 UTC (local TZ: EDT -0400)
# Instances #####
Port  Data Directory      Nice OOM Socket
=====
                                0      0

# MySQL Executable #####
      Path to executable | /usr/sbin/mysqld
      Has symbols | No
# Slave Hosts #####
***** 1. row *****
Server_id: 2
      Host: 192.168.0.59
      Port: 3306
Master_id: 1

(sigue...)
```

pt-mysql-summary - ejemplo

```
Slave_UUID: 4a12b992-9002-11eb-bacf-080027eae162
# Report On Port 3306 #####
      User | percona@%
      Time | 2023-09-29 10:36:02 (EDT)
  Hostname | primary
   Version | 5.7.42-46-log Percona Server (GPL), Release 46,
Revision e1995a8bb71
   Built On | Linux x86_64
   Started | 2023-09-29 08:01 (up 0+02:34:19)
 Databases | 7
   Datadir | /var/lib/mysql/
 Processes | 5 connected, 4 running
Replication | Is not a slave, has 2 slaves connected
   Pidfile | /var/run/mysqld/mysqld.pid (exists)
```

pt-mysql-summary - ejemplo

```
# Processlist #####
```

Command	COUNT(*)	Working	SUM(Time)	MAX(Time)
Binlog Dump	2	2	17500	9000
Daemon	1	1	9000	9000
Query	1	1	0	0
Sleep	2	0	6	3

User	COUNT(*)	Working	SUM(Time)	MAX(Time)
event_scheduler	1	1	9000	9000
percona	1	1	0	0
pmm	2	0	0	0
replicator	1	1	9000	9000
root	1	1	9000	9000

etc...

pt-mysql-summary - ejemplo

Host	COUNT(*)	Working	SUM(Time)	MAX(Time)
localhost	5	2	1250	1250
replica	2	2	2250	1250

db	COUNT(*)	Working	SUM(Time)	MAX(Time)
NULL	7	4	3500	1250

State	COUNT(*)	Working	SUM(Time)	MAX(Time)
	3	0	0	0
Master has sent all binlog to	2	2	2250	1250
starting	1	1	0	0
Waiting on empty queue	1	1	1250	1250

Status Counters (Wait 10 Seconds)

pt-summary - ejemplo

```
$ pt-summary
```

```
# Percona Toolkit System Summary Report #####
    Date | 2023-09-29 14:36:59 UTC (local TZ: EDT -0400)
    Hostname | primary
    Uptime | 2:36, 1 user, load average: 0.71, 0.27, 0.18
    Platform | Linux
    Release | CentOS Linux release 7.8.2003 (Core)
    Kernel | 3.10.0-1127.8.2.el7.x86_64
Architecture | CPU = 64-bit, OS = 64-bit
    Threading | NPTL 2.17
    SELinux | Disabled
    Virtualized | KVM
# Processor #####
    Processors | physical = 1, cores = 2, virtual = 2, hyperthreading = no
    Speeds | 2x2393.998
    Models | 2xIntel(R) Xeon(R) CPU E5620 @ 2.40GHz
    Caches | 2x12288 KB
```

pt-summary - ejemplo

\$ pt-summary

```
# Percona Toolkit System Summary Report #####
    Date | 2023-09-29 14:36:59 UTC (local TZ: EDT -0400)
    Hostname | primary
    Uptime | 2:36, 1 user, load average: 0.71, 0.27, 0.18
    Platform | Linux
    Release | CentOS Linux release 7.8.2003 (Core)
    Kernel | 3.10.0-1127.8.2.el7.x86_64
Architecture | CPU = 64-bit, OS = 64-bit
    Threading | NPTL 2.17
    SELinux | Disabled
    Virtualized | KVM
# Processor #####
    Processors | physical = 1, cores = 2, virtual = 2, hyperthreading = no
    Speeds | 2x2393.998
    Models | 2xIntel(R) Xeon(R) CPU E5620 @ 2.40GHz
    Caches | 2x12288 KB
```

pt-summary - ejemplo

```
# Memory #####
```

```
    Total | 1.8G
```

```
    Free | 644.7M
```

```
    Used | physical = 528.9M, swap allocated = 1024.0M, swap used = 0.0,  
virtual = 528.9M
```

```
    Shared | 20.0M
```

```
    Buffers | 664.2M
```

```
    Caches | 1.1G
```

```
    Dirty | 152 kB
```

```
    UsedRSS | 559.8M
```

```
    Swappiness | 1
```

```
    DirtyPolicy | 30, 10
```

```
    DirtyStatus | 0, 0
```

Locator	Size	Speed	Form Factor	Type	Type Detail
=====	=====	=====	=====	=====	=====

pt-summary - ejemplo

```
# Mounted Filesystems #####
  Filesystem                Size Used Type        Opts
Mountpoint
  /dev/mapper/centos-root    8.0G  43% xfs          rw,relatime,attr2,inode64,noquota
/
  /dev/sda1                  1014M  22% xfs          rw,relatime,attr2,inode64,noquota
/boot
  devtmpfs                   908M   0% devtmpfs
rw,nosuid,size=929212k,nr_inodes=232303,mode=755 /dev
  tmpfs                      184M   0% tmpfs       rw,nosuid,nodev
/run/user/1000
  tmpfs                      184M   0% tmpfs       rw,nosuid,nodev,mode=755
/run/user/1000
  tmpfs                      184M   0% tmpfs
rw,nosuid,nodev,relatime,size=188196k,mode=700,uid=1000,gid=1000 /run/user/1000
  tmpfs                      184M   0% tmpfs       ro,nosuid,nodev,noexec,mode=755

etc...
```

pt-summary - ejemplo

```
# Disk Partioning #####
# Kernel Inode State #####
dentry-state | 19910  974545  0    38570
      file-nr | 1024   0    183629
      inode-nr | 16245 308
# LVM Volumes #####
Unable to collect information
# LVM Volume Groups #####
Unable to collect information
# RAID Controller #####
Controller | No RAID controller detected
# Network Config #####
FIN Timeout | 60
Port Range | 60999
# Interface Statistics #####
interface  rx_bytes rx_packets  rx_errors   tx_bytes tx_packets  tx_errors
=====
lo          25000000    30000        0    25000000    30000        0
enp0s3      10000000    90000        0    22500000    80000        0
bond0              0         0         0         0         0         0
```

pt-summary - ejemplo

```
# Top Processes #####
PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
2571 percona    20   0   161992    2108    1528 R   4.3   0.1   0:00.16  top
   1 root       20   0   190904    3828    2592 S   0.0   0.2   0:07.84  systemd
   2 root       20   0         0         0         0 S   0.0   0.0   0:00.01  kthreadd
   4 root        0 -20         0         0         0 S   0.0   0.0   0:00.00  kworker/0:
   6 root       20   0         0         0         0 S   0.0   0.0   0:00.98  ksoftirqd/0
   7 root       rt    0         0         0         0 S   0.0   0.0   0:00.00  migration/0
   8 root       20   0         0         0         0 S   0.0   0.0   0:00.00  rcu_bh
   9 root       20   0         0         0         0 S   0.0   0.0   0:04.11  rcu_sched
  10 root        0 -20         0         0         0 S   0.0   0.0   0:00.00  lru-add-dr+

# Notable Processes #####
PID    OOM    COMMAND
999    -17    sshd

# Simplified and fuzzy rounded vmstat (wait please) #####
procs  ---swap--  ----io----  ---system---  -----cpu-----
 r  b    si   so   bi   bo    ir    cs   us   sy   il   wa   st
 1  0     0    0  100  20    400   300   7    7   85    1    0
 5  0     0    0    0    0   2500   800  39   52    9    0    0
 6  0     0    0   50    0   2500   600  35   64    1    0    0
```


pt-show -grants

- Realiza un reporte de los usuarios, roles y permisos dentro de la instancia de base de datos
- Muy fácil de usar.

pt-show-grants - ejemplo

```
$ pt-show-grants -h 192.168.0.14
```

```
-- Grants for 'percona'@'%'
CREATE USER IF NOT EXISTS 'percona'@'%';
ALTER USER 'percona'@'%' IDENTIFIED WITH 'mysql_native_password' AS
'*A964945931CBFE32A105AE525648A6C41EEFB50C' REQUIRE NONE PASSWORD EXPIRE DEFAULT ACCOUNT UNLOCK;
GRANT ALL PRIVILEGES ON *.* TO 'percona'@'%' WITH GRANT OPTION;
-- Grants for 'percona'@'monitor'
CREATE USER IF NOT EXISTS 'percona'@'monitor';
ALTER USER 'percona'@'monitor' IDENTIFIED WITH 'mysql_native_password' AS
'*A964945931CBFE32A105AE525648A6C41EEFB50C' REQUIRE NONE PASSWORD EXPIRE DEFAULT ACCOUNT UNLOCK;
GRANT ALL PRIVILEGES ON *.* TO 'percona'@'monitor' WITH GRANT OPTION;
-- Grants for 'root'@'%'
CREATE USER IF NOT EXISTS 'root'@'%';
ALTER USER 'root'@'%' IDENTIFIED WITH 'mysql_native_password' AS '*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B'
REQUIRE NONE PASSWORD EXPIRE DEFAULT ACCOUNT UNLOCK;
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;
-- Grants for 'root'@'localhost'
CREATE USER IF NOT EXISTS 'root'@'localhost';
ALTER USER 'root'@'localhost' IDENTIFIED WITH 'mysql_native_password' AS
'*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B' REQUIRE NONE PASSWORD EXPIRE DEFAULT ACCOUNT UNLOCK;
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION;
```

No se han preguntado algo hasta ahora?



En ningún comando se agregaron o mencionamos las credenciales!



Cómo es posible? Simple, así:

```
$ cat /home/percona/.my.cnf  
[client]  
user = percona  
password = Quilmes1978
```

De esta forma ocultamos credenciales en líneas de comandos (hello auditores!!)



pt-online-schema-change

- Realiza operaciones DDL en línea.
- Muy útil cuando la tabla es muy grande o con alta concurrencia 🔥🔥🔥
- Monitorea la performance de la replicación (`--max-lag=n`) o de un PXC (Percona XtraDB) mediante el flag `--max-flow-ctl=0`



pt-online-schema-change

Cómo funciona?

- Creando una copia vacía de la tabla para modificarla, modificándola según se desee, y luego copiando filas de la tabla original en la nueva tabla mediante `INSERTS`.
- Crea triggers de `INSERT`, `DELETE` Y `UPDATE` en la tabla original hacia la nueva tabla.
- Cuando se completa la copia, aleja la tabla original y la reemplaza por la nueva. De forma predeterminada, también elimina la tabla original.
- El proceso de copia de datos se realiza en pequeños fragmentos de datos, que se varían para intentar que se ejecuten en un período de tiempo específico (`--chunk-time`).
- Cualquier modificación de los datos en las tablas originales durante la copia se reflejará en la nueva tabla, por los triggers creados en la tabla original para actualizar las filas correspondientes en la nueva tabla.

El uso de triggers por parte del pt-osc significa que la herramienta no funcionará si hay triggers previamente creados en la tabla.

pt-online-schema-change - ejemplo

Primary :192.168.0.14
Replica :192.168.0.59

```
$ pt-online-schema-change \  
--print --progress time,10 \  
--pause-file=/tmp/pt-osc_mitabla.pause \  
--no-swap-tables --no-drop-old-table --no-drop-new-table --no-drop-triggers \  
--alter="DROP INDEX idx_indice_repetido_1,DROP INDEX idx_indice_repetido_2" \  
--tries create_triggers:10000:1,drop_triggers:10000:1,copy_rows:10000:1 \  
--set-vars tx_isolation='READ-COMMITTED',lock_wait_timeout=60 \  
h=192.168.0.14,D=test,t=mitabla \  
--recursion-method=hosts \  
--critical-load threads_running=99999 --max-load Threads_running=50 \  
--chunk-time=1 --new-table-name=_%T_new \  
--max-lag 300 --execute
```


pt-online-schema-change - explicación de flags

`--print --progress time,10:` Mostrará un progreso en pantalla cada 10 segundos

`--pause-file=/tmp/pt-osc_mitable.pause:` permite pausar el pt-osc creando el archivo.

`--no-swap-tables --no-drop-old-table --no-drop-new-table --no-drop-triggers`

`--tries create_triggers:10000:1,drop_triggers:10000:1,copy_rows:10000:1:` # intentos y el intervalo en que se ejecutarán las etapas del proceso. Realizará un máximo de 10.000 intentos para cada fase de la operación con un intervalo de 1 segundo entre cada intento.

`--set-vars tx_isolation='READ-COMMITTED',lock_wait_timeout=60:` Variables de sesión.

`h=192.168.0.14,D=test,t=mitable:` los atributos del server, esquema y tabla

`--recursion-method=hosts:` Indica que replicas monitorear

`--critical-load threads_running=99999 --max-load Threads_running=50:` Controlan la carga crítica y máxima, si se superan, el pt-osc se pausa

`--chunk-time=1:` tiempo máximo que se permitirá para cada cacho de trabajo. En este caso, se limita a 1 segundo

`--new-table-name=_%T_new:` Cómo se llamará la nueva tabla

`--max-lag 300:` EL pt-osc se pondrá en suspenso si el lag supera los 300 segundos

pt-online-schema-change - output

```
Found 1 slaves:
replica -> 192.168.0.59:3306
Will check slave lag on:
replica -> 192.168.0.59:3306
Operation, tries, wait:
  analyze_table, 10, 1
  copy_rows, 10000, 1
  create_triggers, 10000, 1
  drop_triggers, 10000, 1
  swap_tables, 10, 1
  update_foreign_keys, 10, 1
Altering `test`.`mitabla`...
Creating new table...
CREATE TABLE `test`.`_mitabla_new` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) DEFAULT NULL,
  `edad` int(11) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `col_1` int(11) DEFAULT NULL,
  `col_2` int(11) DEFAULT NULL,
  `otro_campo` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_indice_repetido_1` (`col_1`),
  KEY `idx_indice_repetido_2` (`col_1`),
  KEY `idx_indice_repetido_3` (`col_1`,`col_2`)
) ENGINE=InnoDB AUTO_INCREMENT=1001 DEFAULT CHARSET=latin1
```

pt-online-schema-change - output

Created new table test._mitabla_new OK.

Altering new table...

```
ALTER TABLE `test`.`_mitabla_new` DROP INDEX idx_indice_repetido_1,DROP INDEX idx_indice_repetido_2
```

Altered `test`.`_mitabla_new` OK.

2023-10-07T15:20:36 Creating triggers...

Event : DELETE

Name : pt_osc_test_mitabla_del

```
SQL : CREATE TRIGGER `pt_osc_test_mitabla_del` AFTER DELETE ON `test`.`mitabla` FOR EACH ROW BEGIN DECLARE  
CONTINUE HANDLER FOR 1146 begin end; DELETE IGNORE FROM `test`.`_mitabla_new` WHERE  
`test`.`_mitabla_new`.`id` <=> OLD.`id`; END
```

Suffix: del

Time : AFTER

Event : UPDATE

Name : pt_osc_test_mitabla_upd

```
SQL : CREATE TRIGGER `pt_osc_test_mitabla_upd` AFTER UPDATE ON `test`.`mitabla` FOR EACH ROW BEGIN DECLARE  
CONTINUE HANDLER FOR 1146 begin end; DELETE IGNORE FROM `test`.`_mitabla_new` WHERE !(OLD.`id` <=> NEW.`id`)  
AND `test`.`_mitabla_new`.`id` <=> OLD.`id`; REPLACE INTO `test`.`_mitabla_new` (`id`, `nombre`, `edad`,  
`email`, `col_1`, `col_2`, `otro_campo`) VALUES (NEW.`id`, NEW.`nombre`, NEW.`edad`, NEW.`email`,  
NEW.`col_1`, NEW.`col_2`, NEW.`otro_campo`); END
```

Suffix: upd

Time : AFTER

pt-online-schema-change - output

```
-----
Event : INSERT
Name  : pt_osc_test_mitabla_ins
SQL   : CREATE TRIGGER `pt_osc_test_mitabla_ins` AFTER INSERT ON `test`.`mitabla` FOR EACH ROW BEGIN DECLARE
CONTINUE HANDLER FOR 1146 begin end; REPLACE INTO `test`.`_mitabla_new` (`id`, `nombre`, `edad`, `email`,
`col_1`, `col_2`, `otro_campo`) VALUES (NEW.`id`, NEW.`nombre`, NEW.`edad`, NEW.`email`, NEW.`col_1`,
NEW.`col_2`, NEW.`otro_campo`);END
Suffix: ins
Time   : AFTER
-----

2023-10-07T15:20:36 Created triggers OK.
2023-10-07T15:20:36 Copying approximately 1000 rows...
INSERT LOW_PRIORITY IGNORE INTO `test`.`_mitabla_new` (`id`, `nombre`, `edad`, `email`, `col_1`, `col_2`,
`otro_campo`) SELECT `id`, `nombre`, `edad`, `email`, `col_1`, `col_2`, `otro_campo` FROM `test`.`mitabla`
LOCK IN SHARE MODE /*pt-online-schema-change 13482 copy table*/
2023-10-07T15:20:38 Copied rows OK.
Not dropping old table because --no-drop-triggers was specified.
Not dropping triggers because --no-drop-triggers was specified. To drop the triggers, execute:
DROP TRIGGER IF EXISTS `test`.`pt_osc_test_mitabla_del`
DROP TRIGGER IF EXISTS `test`.`pt_osc_test_mitabla_upd`
DROP TRIGGER IF EXISTS `test`.`pt_osc_test_mitabla_ins`
Not dropping the new table `test`.`_mitabla_new` because --no-drop-new-table was specified. To drop the new
table, execute:
DROP TABLE IF EXISTS `test`.`_mitabla_new`;
Successfully altered `test`.`mitabla`.
```

pt-online-schema-change - post tarea

Una vez que se confirma que el pt-osc terminó OK

```
$ echo $?  
0
```

Se procede a realizar el swap de la tabla y DROP de triggers de la siguiente manera

```
mysql test> SET lock_wait_timeout=5;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql test> RENAME TABLE test.mitabla TO test.mitabla_old, test._mitabla_new TO test.mitabla;  
Query OK, 0 rows affected (0.39 sec)
```

```
mysql test> DROP TRIGGER IF EXISTS test.pt_osc_test_mitabla_del ;  
Query OK, 0 rows affected (0.04 sec)
```

```
mysql test> DROP TRIGGER IF EXISTS test.pt_osc_test_mitabla_upd ;  
Query OK, 0 rows affected (0.13 sec)
```

```
mysql test> DROP TRIGGER IF EXISTS test.pt_osc_test_mitabla_ins ;  
Query OK, 0 rows affected (0.04 sec)
```

pt-online-schema-change - chequeo final

Verificamos que los índices redundantes no están más

```
mysql test> show create table test.mitabla\G
***** 1. row *****
      Table: mitabla
Create Table: CREATE TABLE `mitabla` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) DEFAULT NULL,
  `edad` int(11) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `indice_repetido_1` int(11) DEFAULT NULL,
  `indice_repetido_2` int(11) DEFAULT NULL,
  `otro_campo` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_indice_repetido_3` (`indice_repetido_1`,`indice_repetido_2`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

pt-archiver

- Permite “archivar” filas de una tabla hacia otra tabla o archivo.
- Puede borrar filas de una tabla post pt-osc.
- Podés definir lógica para filtrar las filas a archivar.
- Se puede regular la velocidad de archivado.



pt-archiver – ejemplo

```
$ pt-archiver \  
--source h=192.168.0.14,D=test,t=mitabla_old \  
--progress=50000 --retries=5 --limit=5000 --bulk-delete \  
--sentinel=/tmp/pt-archiver-mitabla_old --where "1=1" \  
--max-lag=60 --check-slave-lag=h=192.168.0.59 \  
--primary-key-only \  
--purge
```

TIME	ELAPSED	COUNT
2023-10-07T15:24:30	0	0
2023-10-07T15:24:30	0	999

(Explicación de cada flag en la próxima página)

pt-archiver – explicación de flags

--source h=192.168.0.14,D=test,t=mitabla_old: Especifica la fuente de datos desde la cual se archivarán los datos. En este caso, se conecta a la base de datos en el host (h) 192.168.0.14, la base de datos (D) "test", y la tabla (t) "mitabla_old".

--progress=50000: Se mostrará información de progreso cada vez que se procesen 50,000 filas.

--retries=5: Número máximo de intentos de reconexión si la conexión con la base de datos se pierde.

--limit=5000: La cantidad de filas procesadas en cada iteración. Aquí, de a 5000 filas a la vez.

--bulk-delete: Realiza eliminaciones masivas en lugar de fila por fila, más eficiente en ciertos casos.

--sentinel=/tmp/pt-archiver-mitabla_old: Especifica la ubicación del archivo de control.

--where "1=1": En este caso, "1=1" selecciona todas las filas, se archivarán todas las filas de la tabla.

--max-lag=60: Se archivará si el delay es inferior a 60 segundos.

--check-slave-lag=h=192.168.0.59: Especifica la réplica que debe ser monitoreada para el retraso de replicación antes de realizar la operación de archivado.

--primary-key-only: Indica que solo se deben archivar las filas que contienen valores clave primaria.

--purge: Este flag permite realizar una purga después del archivado, lo que elimina las filas archivadas de la tabla de origen.

pt-archiver – post tarea

Una vez que se confirma que el pt-archiver terminó OK

```
$ echo $?  
0
```

Se procede a realizar el DROP de la tabla archivada de la siguiente manera

```
mysql test> SELECT COUNT(1) FROM test.mitabla_old;  
+-----+  
| COUNT(1) |  
+-----+  
|          1 |  
+-----+  
1 row in set (0.01 sec)
```

```
mysql test> DROP TABLE IF EXISTS test.mitabla_old;  
Query OK, 0 rows affected (0.18 sec)
```

pt-archiver – post tarea

Una vez que se confirma que el pt-archiver terminó OK

```
$ echo $?  
0
```

Se procede a realizar el DROP de la tabla archivada de la siguiente manera

```
mysql test> SELECT COUNT(1) FROM test.mitabla_old;  
+-----+  
| COUNT(1) |  
+-----+  
|          1 |  
+-----+  
1 row in set (0.01 sec)
```

```
mysql test> DROP TABLE IF EXISTS test.mitabla_old;  
Query OK, 0 rows affected (0.18 sec)
```

pt-visual-explain

- Transforma la salida de EXPLAIN en una representación de árbol del plan de consulta.

pt-visual-explain - ejemplo (Classic Explain)

```
$ mysql test -e "explain select a.id, a.nombre, a.edad, b.detalle from mitabla a,
hija b where a.id=b.id_mitabla and a.id=100;"
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	a	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL
1	SIMPLE	b	NULL	ref	idx_fk_id_mitabla	idx_fk_id_mitabla	5	const	200	100.00	NULL

pt-visual-explain - ejemplo (con pt-visual-explain)

```
$ mysql -e "explain select a.id, a.nombre, a.edad, b.detalle from mitabla a, hija b  
where a.id=b.id_mitabla and a.id=100;" | pt-visual-explain
```

```
JOIN  
+- Bookmark lookup  
| +- Table  
| | table          b  
| | possible_keys  idx_fk_id_mitabla  
| +- Index lookup  
|   key            b->idx_fk_id_mitabla  
|   possible_keys  idx_fk_id_mitabla  
|   key_len        5  
|   ref            const  
|   rows            200  
+- Bookmark lookup  
  +- Table  
  | table          a  
  | possible_keys  PRIMARY  
  +- Constant index lookup  
    key            a->PRIMARY  
    possible_keys  PRIMARY  
    key_len        4  
    ref            const  
    rows            1
```

Dónde encuentro esta presentación?





Gracias!