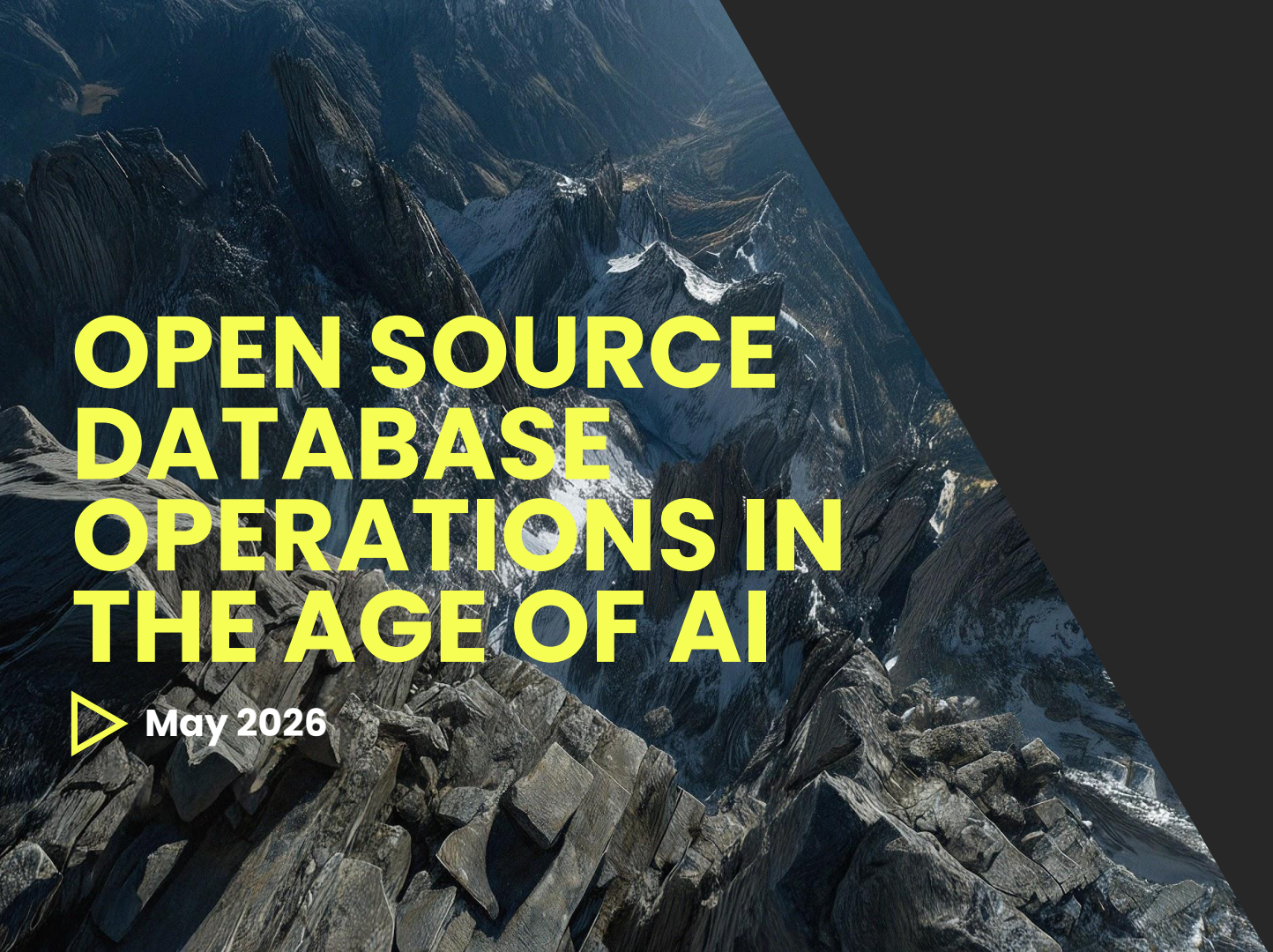


OPEN SOURCE DATABASE OPERATIONS IN THE AGE OF AI

May 2026

PERCONA

Percona Monitoring and Management

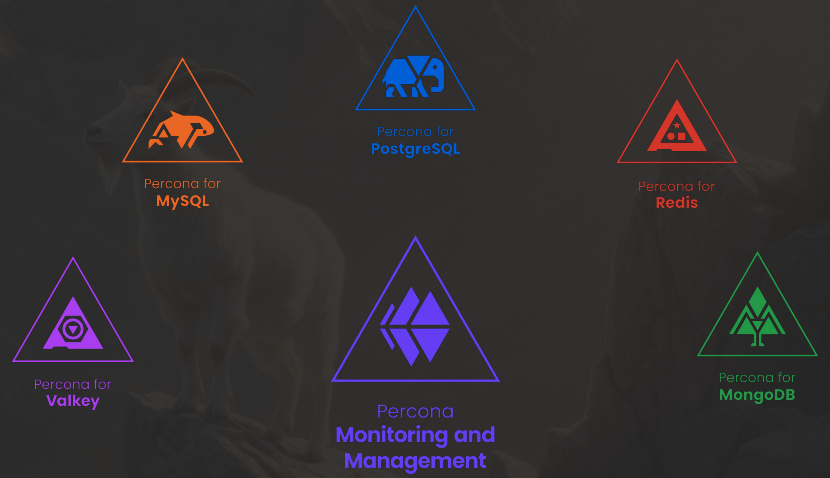
Percona
Monitoring and Management

The Leading Open Source Observability and Management Platform for the Modern Open Source Database Environments

PERCONA®

DATABASE OPERATIONS
HAVE FUNDAMENTALLY CHANGED

- More engines
- More platforms
- More telemetry
- Less operational time

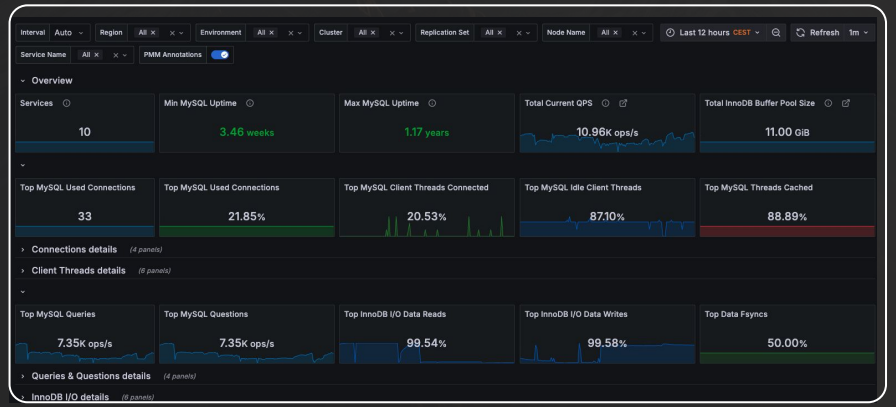


Public Cloud | Hybrid | Private Cloud

PERCONA®

DASHBOARDS
DON'T SOLVE INCIDENTS

TRADITIONAL MONITORING



REACTIVE

INTELLIGENT OPERATIONS

Root cause identified
Slow query on orders table

Impact
3 nodes - 12% queries affected

Recommendation
Add index on [customer_id, created_at]

Confidence
92%

PROACTIVE

PERCONA®

THE FUTURE IS OPERATIONAL INTELLIGENCE

Telemetry

Anomaly Detection

Query Optimization

Natural Language Troubleshooting

Remediation Suggestions

UNDERSTAND **PREDICT** **RECOMMEND** **AUTOMATE**

PERCONA®

Home page
MySQL
PostgreSQL
Operating system
All dashboards
eBPF
Query Analytics (QAN)
Investigations
Autonomous Database Reliability Engineer
Explore
Alerts
Advisors
Inventory
Backups
Configuration
Users and access
Account
Help

Alert: Slow Query

Summary

Slow query alert triggered by full table scan on sbtest2. Query used WHERE on column k, which is not indexed. Table has nearly 10 million rows, causing poor performance.

Confidence: HIGH (85) Show evidence map (4)

Computed from evidence count/diversity, report completeness, and uncertainty signals.

Time range: 5/6/2026, 8:51:23 AM — 5/6/2026, 8:51:23 AM Source: Alert Service: mysql-mysql

Report

Alert Explanation

Slow query alert triggered because SELECT c FROM sbtest2 WHERE k BETWEEN ? AND ? performed a full table scan on sbtest2, which has nearly 10 million rows. The column k in the WHERE clause is not indexed.

Key Findings

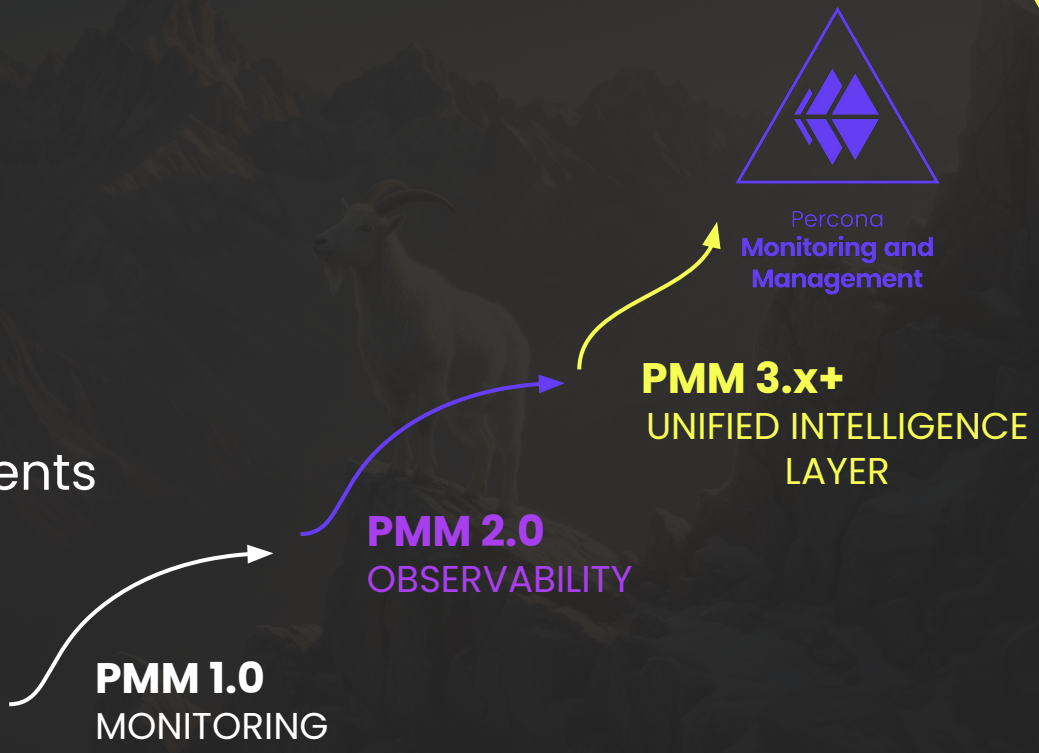
Query did a full table scan (type=ALL) on sbtest2. No index exists on k. Table has ~9.8M rows. Only index is on id. High execution time and rows examined.

Conclusions and Possible Root causes

Root cause: No index on k led to full table scans and slow performance. Adding an index on k will allow MySQL to optimize the query.

PMM IS EVOLVING
BEYOND MONITORING

- Real-time analytics
- Unified visibility
- HA & DR ready deployments
- AI-ready architecture
- Open integrations



PERCONA

OPEN SOURCE BECOMES STRATEGIC

CLOSED AI OPS PLATFORM

- Opaque decisions
- Vendor lock-in
- LImited flexibility
- No data portability
- Black box AI

VS.

OPEN AI-DRIVEN OBSERVABILITY

- Transparent by design
- Portable anywhere
- Extensible ecosystem
- Your data, your control
- Trustworthy AI

PERCONA®

OPEN MEANS CONTROL. CONTROL BUILD TRUST

PERCONA®

THE FUTURE
OPERATOR EXPERIENCE

Why is my query slow?

Root cause
Missing index on orders (customer_id, created_at)

Impacted
3 nodes - 12% queries affected

Recommendation
Add index on [customer_id, created_at]

Anomaly timeline
Started 14 min ago - Ongoing

PERCONA

PMM Query Analytics

Filters Show Selected Reset All

Copy Link Add column

Filter by...

Environment 1
- test 100%

Schema 2
- sbtest 99.13%
- n/a 0.87%

Node Name 1
- mysql 100%

Service Name 1
- mysql-mysql 100%

Client Host 1
- localhost 100%

User Name 2
- sbtest 99.13%
- pmm 0.87%

#	Query	Search by...	Load	Query Count	Query Time
	TOTAL		1.87 load	6.91 QPS	271.16 ms
1	select c from sbtest2 where k between ? and ?		0.84 load	<0.01 QPS	0:02:06
2	select c from sbtest2 where k=?		0.68 load	<0.01 QPS	0:02:02
3	select c from sbtest3 where k between ? and ?		0.08 load	0.01 QPS	5.80 sec
4	select c from sbtest4 where k between ? and ?		0.06 load	0.01 QPS	5.04 sec
5	select c from sbtest5 where k between ? and ?		0.06 load	0.01 QPS	5.06 sec

< 1 2 3 4 > 25 / page 1-25 of 92 items

Details Examples Explain Tables AI Insights Close

Last analyzed: Apr 12, 02:15 PM

Copy analysis Create ServiceNow ticket Re-run Analysis

Summary

Query `SELECT c FROM sbtest2 WHERE k BETWEEN 50305 AND 50404` on table `sbtest2` in database `sbtest` performs a full table scan (`type=ALL`) because there is no index on column `k` . This results in inefficient execution, especially as the table grows.

Evidence

EXPLAIN output:

```
id | select_type | table   | partitions | type | possible_keys | key | key_len | ref | rows    | filtered | Extra
1  | SIMPLE      | sbtest2 | NULL       | ALL  | NULL          | NULL| NULL    | NULL| 9859423 | 11.11    | Using where
```

SHOW CREATE TABLE:

```
CREATE TABLE `sbtest2` (
  `id` int NOT NULL AUTO_INCREMENT,
  `k` int NOT NULL DEFAULT '0',
  `c` char(120) NOT NULL DEFAULT '',
  `pad` char(60) NOT NULL DEFAULT '',
```

PERCONA

Workload Analyses for last 10h

Key Findings

MySQL handler metrics (mysql_global_status_handlers_total) are present and active; top 5 handlers dominate activity. Top slow queries are SELECTs on sbtest2 and sbtest3, with sbtest2 queries performing full table scans (ALL type, no key used). EXPLAIN for sbtest2 queries shows no index usage (ALL), high row estimates (9.8M+), and 'Using where' filter; sbtest3 query uses a range scan with index k_3.

Conclusions and Possible Root causes

sbtest2 queries perform full table scans due to missing index on column k. sbtest3 queries use index k_3 for range scans.

MySQL Handlers panel

Name	Mean	Max	Min
delete	0.08 ops/s	0.32 ops/s	0.00 ops/s
external_lock	9.65 ops/s	14.05 ops/s	8.03 ops/s
read_first	0.52 ops/s	0.67 ops/s	0.48 ops/s
read_key	52.13 ops/s	73.52 ops/s	49.61 ops/s
read_next	114.25 ops/s	546.18 ops/s	7.78 ops/s
read_rnd	0.47 ops/s	0.95 ops/s	0.00 ops/s
read_rnd_next	296.09K ops/s	989.68K ops/s	454.69 ops/s
update	1.16 ops/s	4.18 ops/s	0.00 ops/s
write	197.14 ops/s	256.86 ops/s	184.53 ops/s

SMARTER MONITORING. UNIFIED CONTROL
BUILT FOR THE FUTURE

AI-POWERED OPTIMIZATION

UNIFIED DATABASE INTELLIGENCE

AUTONOMOUS OPERATIONS

OPEN ECOSYSTEM

THE NEXT ERA OF DATABASE OPERATIONS IS OPEN

The future is not AI replacing operators.
The future is operators. becoming exponentially more capable.

PERCONA®

Percona

Monitoring and Management

THANK YOU !

PERCONA®