


ORACLE®



ORACLE
OPEN
WORLD

MySQL Performance Tuning 101

Hands-on-Lab

Mirko Ortensi
Senior Support Engineer
MySQL Support @ Oracle
October 3, 2017

ORACLE®

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

MySQL™

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Objectives



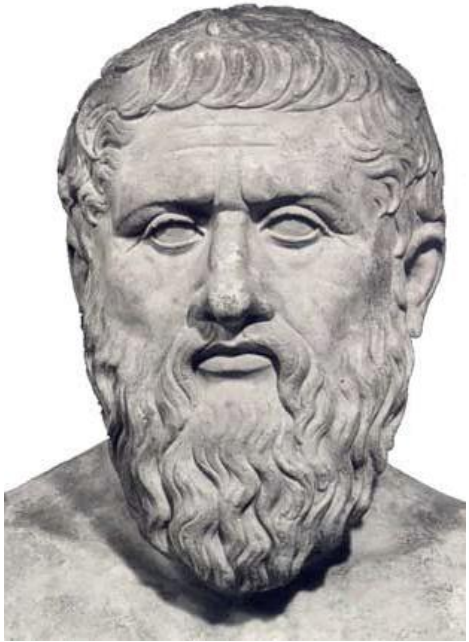
Understand what are the most important configuration parameters to get started with MySQL



Find out what to look at when performance is not the expected



Better hardware does not always mean scalability. Find out what are the typical bottlenecks



“The beginning is the most important part of the work”

– Plato

Hands-On Lab Agenda

- 1 Introduction
- 2 Connections
- 3 Threads model
- 4 REDO log
- 5 InnoDB buffer pool
- 6 Execution plan

A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking at a tablet. The background is a bokeh of city lights at night. The image is overlaid with a teal geometric pattern.

Introduction

Get ready for scaling in 5 steps

Introduction

Login

LOGIN

```
shell> mysql -uroot
```

CHANGE PASSWORD

```
mysql> alter user root@localhost identified by 'new-  
password';
```

ERROR 1819 (HY000): Your password does not satisfy the current policy requirements

Introduction

Reset password

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
```

1. Either `uninstall plugin validate_password;`
2. Or set a good password like: **Oracle1***

To reinstall the plugin:

```
mysql> INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

Introduction

Miscellaneous

- **shell\$** yum install sysstat *(installed)*
- set -o history in bash.rc *(optional)*
- Log for RPM based distribution is in /var/log/mysqld.log
- **shell\$** wget <https://tinyurl.com/employeesdb>
(provisioned)
- **shell\$** sudo service mysqld restart

Introduction

Test Database and Benchmarking

TEST DATABASE

The [Employees](#) sample database provides a combination of a large base of data (approximately 160MB) spread over six separate tables and consisting of 4 million records in total.

BENCHMARKING strategies: single operation vs workload

[mysqlslap](#) is a diagnostic program designed to emulate client load for a MySQL server and to report the timing of each stage. It works as if multiple clients are accessing the server.

InnoDB storage engine will be used for the following exercises

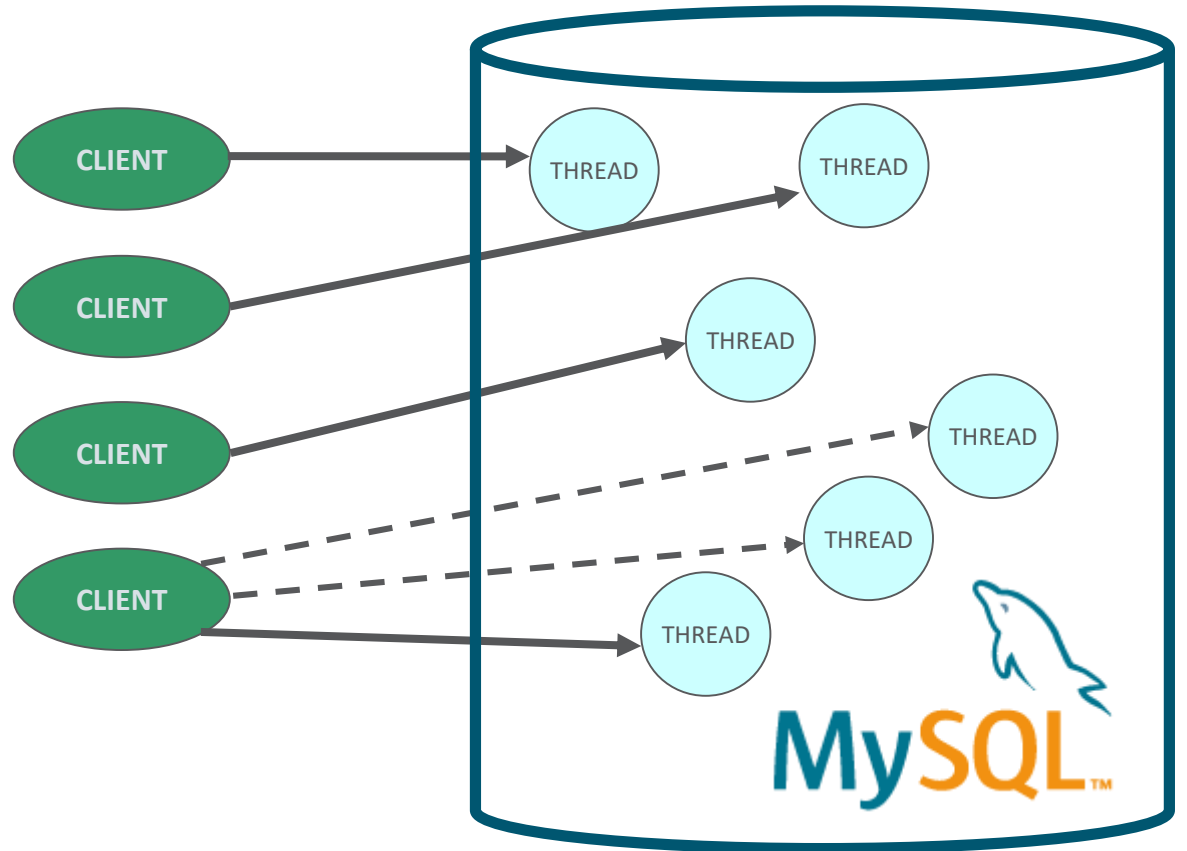
A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking at a tablet. The background is a bokeh of city lights at night. The image is overlaid with a teal geometric pattern.

Connections

The more, the merrier

Connections

- Each client **connection** is associated to a thread
- Thread handles **authentication and the request**
- **Manager thread** handles creation of the thread



Connections

Basic parameters and configuration

- `max_connections`
 - The maximum permitted number of simultaneous client connections. By default, this is 151.
- `Max_used_connections`
 - The maximum number of connections that have been in use simultaneously since the server started.
- `Connection_errors_max_connections`
 - The number of connections refused because the server `max_connections` limit was reached.

Connections

Exercise 1 – UNDERSTAND HOW TO CONFIGURE CONNECTIONS

1. **mysql>** select @@max_connections;
2. **mysql>** show global status like 'Max_used_connections';
3. **shell\$** mysqlslap --auto-generate-sql --concurrency=100 --iterations=10
4. **mysql>** show global status like 'Max_used_connections';

Connections

Exercise 1 - ADD MORE LOAD!

```
shell$ mysqlslap --auto-generate-sql --  
concurrency=170 --iterations=10
```

```
mysqlslap: Error when connecting to server: 1040 Too many connections
```

WHAT'S WRONG?

```
mysql> show global status like 'Connection_errors%';
```


Connections

Exercise 1 – ALLOW MORE CONNECTIONS

1. **mysql>** SET GLOBAL max_connections=200;
2. **mysql>** select @@max_connections;
3. **shell\$** mysqlslap --auto-generate-sql --
concurrency=170 --iterations=10

Make configuration changes persistent: add them to `/etc/my.cnf`

Connections

Connection pooling

- Connection pooling is a technique of creating and managing a pool of connections that are ready for use by any [thread](#) that needs them
- Connection pooling can greatly increase the performance of your application, while reducing overall resource usage
 - Connector/J
 - Connector/Python
 - Connector/Net
 - Connector/ODBC

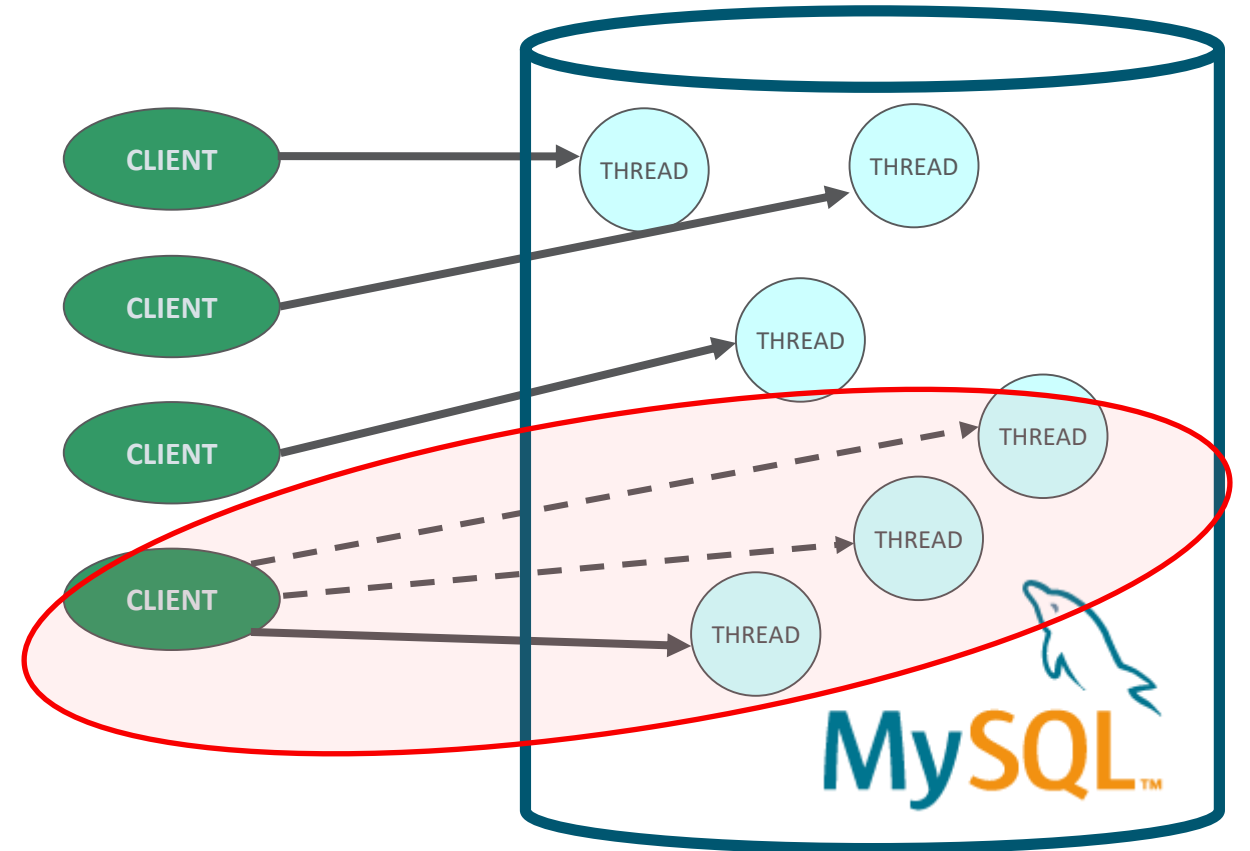
A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking at a tablet. The background is a bokeh of city lights at night. The image is overlaid with a teal geometric pattern.

Threads model

Reuse for good

Threads model

- Manager threads create a new thread when necessary but **try to avoid doing so** by consulting the thread cache
- When a connection ends, its **thread is returned to the thread cache** if the cache is not full.



Threads model

Exercise 2 – UNDERSTAND THE ROLE OF THREAD CACHE

- **mysql>** `select @@thread_cache_size;`
- **mysql>** `show global status like 'Threads_cached';`
- **mysql>** `show global status like 'Threads_created';`

Threads model

Exercise 2 – CHECK THREADS CREATION

- **shell\$** mysqlslap --auto-generate-sql --concurrency=100 --iterations=10
- **mysql>** show global status like 'Threads_created';
- **shell\$** mysqlslap --auto-generate-sql --concurrency=100 --iterations=10
- **mysql>** show global status like 'Threads_created';

Threads model

Exercise 2 – INCREASE THREAD CACHE

- **mysql>** SET GLOBAL thread_cache_size=100;
- **shell\$** mysqlslap --auto-generate-sql --concurrency=100 --iterations=10
- **mysql>** show global status like 'Threads_created';

Threads model

MySQL Enterprise Thread pool

- The default thread-handling model in MySQL Server executes statements using **one thread per client connection**
- As **more clients connect** to the server and execute statements, overall performance degrades
- The thread pool addresses several **problems of the one thread per connection model**

A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking at a tablet. The background is a bokeh of lights, suggesting an indoor setting at night. The image is overlaid with a teal geometric pattern.

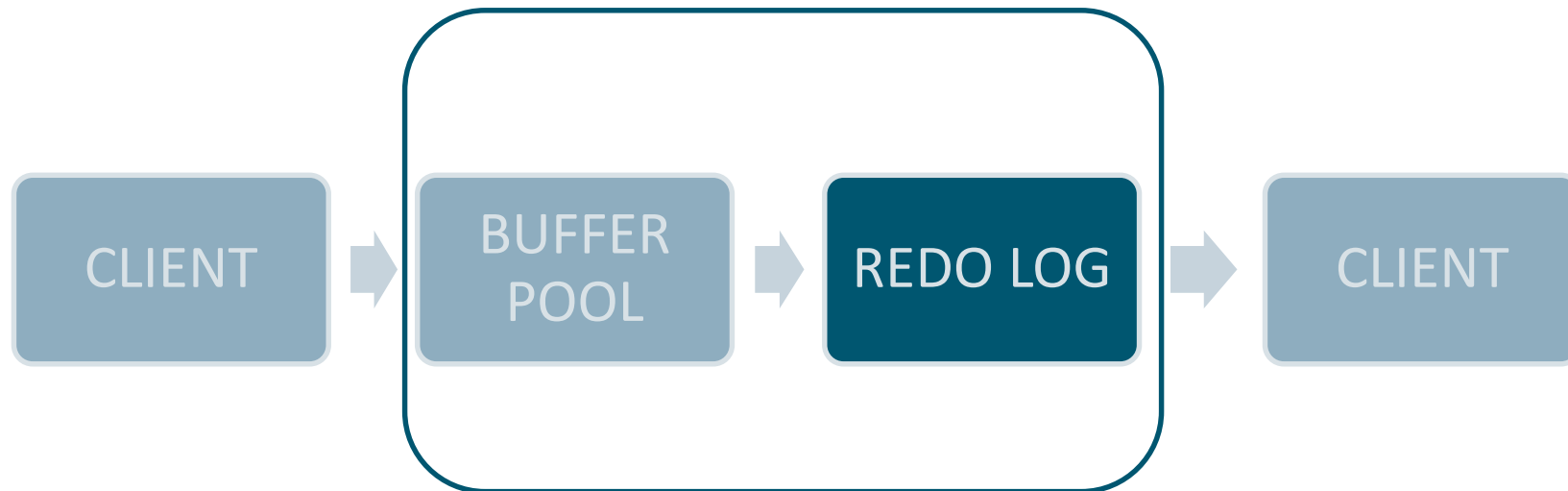
REDO log

Sustainable IO throughput

REDO log

What is it?

- The redo log is a disk-based data structure used during crash recovery
- Redo log encodes requests to change InnoDB table data
- Redo log is flushed before a transaction is committed
- MySQL writes to the redo log files in a circular fashion



REDO log

Configuration

- `innodb_log_buffer_size`
 - The size in bytes of the buffer that InnoDB uses to write to the log files on disk
- `innodb_log_file_size`
 - The size in bytes of each log file in a log group
- `innodb_log_files_in_group`
 - The number of log files in the log group
- `innodb_flush_log_at_trx_commit`
 - ACID compliance for commit operations vs higher performance

REDO log

Exercise 3 – SHOW REDO LOGGING IO IMPACT

- **shell\$** `mysqlslap --auto-generate-sql --concurrency=100 --iterations=100`
- **shell\$** `iostat -xdh /dev/sda 2`
 - Execute on other console session as mysqlslap is running

%util

Percentage of CPU time during which I/O requests were issued to the device (bandwidth utilization for the device).

w/s

The number of write requests that were issued to the device per second.

REDO log

Exercise 3 – REDUCE IO PRESSURE

- **mysql>** set global innodb_flush_log_at_trx_commit=0;
- **shell\$** mysqlslap --auto-generate-sql --concurrency=100 --iterations=100
- **shell\$** iostat -xdh /dev/sda 2
 - Execute on other console session as mysqlslap is running

With `innodb_flush_log_at_trx_commit = 0`
you can lose up to a second of transactions with any **mysqld** process crash.
Use with test instances, restores, replication slaves...

REDO log

Exercise 3 – MEASURE LATENCY

- **mysql>** show global status like 'Innodb_os_log_written';
- SYS schema: the [host summary by file io type](#) view

```
mysql> select * from sys.host_summary_by_file_io_type;
```

host	event_name	total	total_latency	max_latency
background	wait/io/file/innodb/innodb_data_file	1607	332.96 ms	39.41 ms
background	wait/io/file/innodb/innodb_log_file	171	237.56 ms	17.46 ms
background	wait/io/file/sql/FRM	2135	1.98 ms	67.37 us
background	wait/io/file/sql/casetest	15	926.63 us	833.10 us
background	wait/io/file/sql/file_parser	204	296.84 us	3.58 us
background	wait/io/file/myisam/kfile	33	51.20 us	5.19 us
background	wait/io/file/sql/pid	3	39.02 us	27.48 us

A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking intently at a tablet computer. The background is a blurred cityscape at night with bokeh lights. The image is overlaid with a teal geometric pattern.

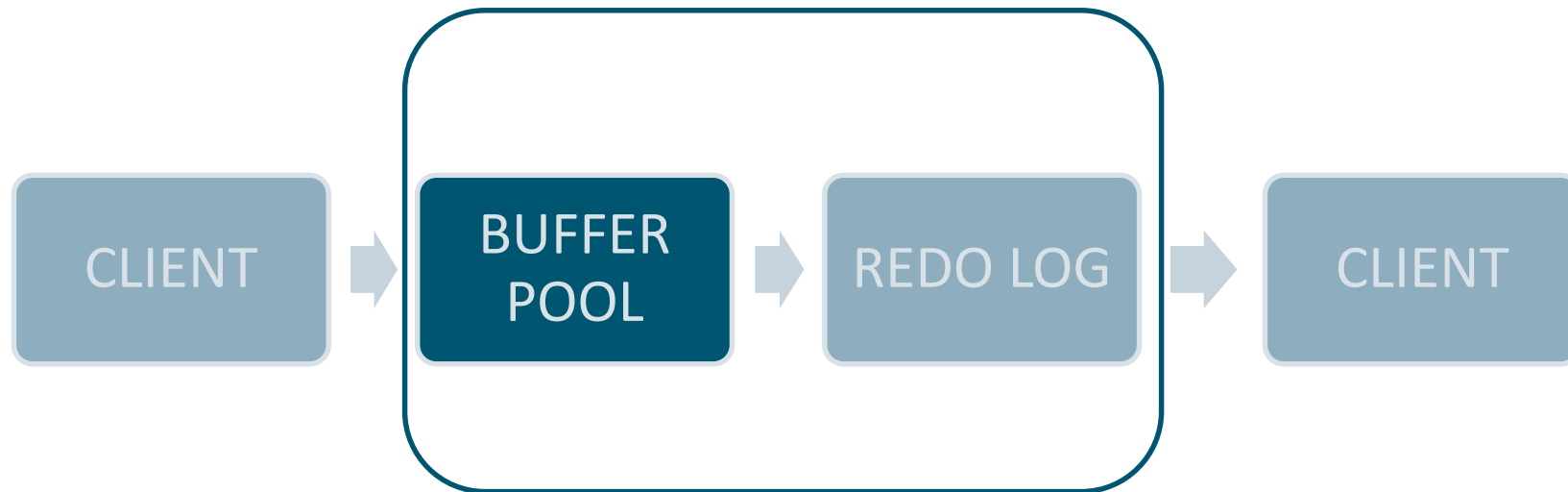
InnoDB Buffer Pool

In memory it's faster

InnoDB Buffer Pool

What is it?

- InnoDB maintains a storage area called the buffer pool for caching data and indexes in memory
- Used to keep frequently accessed data in memory



InnoDB Buffer Pool

How to Monitor the Buffer Pool

```
mysql> show global status like  
'innodb_buffer_pool_pages_total';
```

```
mysql> show global status like  
'innodb_buffer_pool_pages_free';
```

```
mysql> show global status like 'Innodb_buffer_pool_reads';
```

```
mysql> show global status like  
'Innodb_buffer_pool_read_requests';
```

```
mysql> select * from  
information_schema.INNODB_BUFFER_POOL_STATS\G;
```

```
mysql> select * from  
sys.schema_table_statistics_with_buffer\G;
```

InnoDB Buffer Pool

Exercise 4 – DIMENSION BUFFER POOL CORRECTLY

```
shell$ mysqlslap --no-drop --create-schema=employees  
--query="select * from employees.employees;" --  
concurrency=1 --iterations=100
```

```
mysql> select * from  
information_schema.INNODB_BUFFER_POOL_STATS\G;
```

InnoDB Buffer Pool

Exercise 4 – SHRINK BUFFER POOL

EDIT /ETC/MY.CNF AND RESTART THE SERVER

1. `innodb_buffer_pool_chunk_size=1048576`
2. `innodb_buffer_pool_size=5242880`
3. **shell**\$ `sudo service mysld restart`

Run `mysqlslap` and monitor the stats of InnoDB Buffer Pool on a different console

- **shell**\$ `mysqlslap --no-drop --create-schema=employees --query="select * from employees.employees;" --concurrency=1 --iterations=100`
- **mysql**> `select * from information_schema.INNODB_BUFFER_POOL_STATS\G;`
 - Monitor on other console

A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking intently at a tablet device. The background is a blurred bokeh of lights, suggesting an indoor setting at night. The overall color palette is dominated by teal and blue tones.

Execution plan

Don't waste your time

Execution plan

- Different aspects to consider to speed up a query
 - Concurrency, caching, locking, IO...and execution plan
- Tools to investigate a slow query
 - SHOW ENGINE INNODB STATUS
 - SHOW PROCESSLIST
 - EXPLAIN
 - SLOW QUERY LOG
 - SYS SCHEMA

Execution plan

EXERCISE 5 – HAVE A BAD EXECUTION PLAN?

- **mysql>** explain select * from employees where
adddate(hire_date,INTERVAL 18 YEAR)>=NOW();
- **mysql>** create index hire_date_idx on
employees(hire_date);

Simple index does not improve execution plan and full scan is performed

Execution plan

- EXERCISE 5 –FUNCTION GENERATED INDEX
- **mysql**> ALTER TABLE employees ADD COLUMN past_date date GENERATED ALWAYS AS (adddate(hire_date,INTERVAL 18 YEAR)) VIRTUAL;
- **mysql**> CREATE INDEX past_date_idx on employees (past_date);
- **mysql**> explain select * from employees where adddate(hire_date,INTERVAL 18 YEAR)>=NOW();

Wrapping up

1. Use connection pool
2. Tune thread cache / use Thread Pooling plugin (Enterprise version)
3. Configure REDO logging
4. Configure buffer pool to maximize memory usage
5. Analyze execution plan for queries

Thank You

- MySQL Documentation
dev.mysql.com
- Support Blog
dev.mysql.com/support/blogs
- Stars Bar
oracle.com/goto/starsbar

Q&A

Recommended Conference Sessions

CON7309: **MySQL 8.0: What's New in the Optimizer**

Wednesday, Oct 04, 12:00 p.m. - 12:45 p.m.
Marriott Marquis (Yerba Buena Level) - Salon 14

CON7310: **MySQL 8.0: What's New in InnoDB**

Wednesday, Oct 04, 4:30 p.m. - 5:15 p.m.
Marriott Marquis (Yerba Buena Level) - Salon 14

CON7313: **MySQL 8.0: Performance, Scalability, and Benchmarks**

Wednesday, Oct 04, 5:30 p.m. - 6:15 p.m.
Marriott Marquis (Yerba Buena Level) - Salon 14





Oracle Support Stars Bar

- Ask the Experts your toughest product questions.
- View My Oracle Support and Product Demonstrations.
- Learn what's new and more!

oracle.com/goto/starsbar

Keep Learning with Oracle University



200,000
Students trained per
year



1300+
Training Courses



2000+ Courses:
Cloud
Technology
Applications
Industries



500+
Education
Partnerships



**450+ Training On
Demand courses**
**20 Learning
Subscriptions**



2 Million
Oracle Certified
Professionals

ORACLE®

Support

oracle.com/goto/openworldservices

Integrated Cloud

Applications & Platform Services

ORACLE®