Table of Contents

Preface 1

Chapter 1: Introduction 5

Why MySQL 8? 6

Why is MySQL 8.0 the next generation? 6

Why it is so important to have a good MySQL architecture design 7

Summary 8

Chapter 2: MySQL 8's New Features 9

Global data dictionary 9

MySQL 8's support roles and history 12

MySQL 8 supports the creation and management of resource groups and permissions 13

InnoDB enhancements 17

JSON enhancements functionalities 18

Invisible indexes from MySQL optimizer 21

Features deprecated in MySQL 8.0 24

A quick look at the features removed from MySQL 8.0 24

Summary 26

Chapter 3: Indexing Your Data for High Performance 27

How does MySQL optimizer think in MySQL 8.0? 28

What kind of data type should I consider indexing first and why? 30

Why should I have a good index strategy? 32

What impact does an index have on MySQL performance? 33

How to display and analyze a table structure 35

How to efficiently read MySQL query execution plans 41

How to effectively read the EXPLAIN results 41

id (JSON name: select_id) 44

select_type (JSON name: none) 44

table (JSON name: table_name) 45

partitions (JSON name: partitions) 45

type (JSON name: access_type) 45

EXPLAIN extra information 46

How to know when to create an index 48

Multiple column index versus multiple indexes 49

How to organize your columns in an index for good performance 54

Case study 1 – how to use the EXPLAIN plan tool in MySQL 8.0 57

EXPLAIN options 58

Case study 2 – how to display and analyze a table structure versus the EXPLAIN plan tool 60
Case study 3 – how to organize your columns in an index efficiently 62
Creating a missing index 65
Tips and techniques 67
The five general rules for your indexes 67
Tip 2 – the five general rules to optimize your queries 67
Tip 3 – understand your material resources 68
Tip 4 – the configuration is not the only thing to take into consideration 69
Tip 5 – recommendations on the performance of the MySQL architecture 70
A technical case study 70
Summary 72
Chapter 4: Advanced Data Techniques for Large Queries 73
The most important variables are full-scan indicators 74
Partitioning a table 76
An overview of partitioning in MySQL 8.0 76
Available partitioning type 77
Horizontally partitioning your data 77
Managing partitions 77
RANGE partitioning 78
LIST partitioning 80
HASH partitioning 82
KEY partitioning 83
Using partitions 84
Partition pruning 84
Getting rid of unused and duplicate indexes 86
Unused indexes 86
Duplicate indexes 87
Bonus – potentially missing indexes 88
The most important query optimizations 90
Optimizing a query with the WHERE clause 90
Optimizing a query with a GROUP BY clause 94
Optimizing a query with the ORDER BY clause 96
Temporary tables 98
Case study 1 – an example of how to optimize a complex query 101
Case study 2 – how to optimize sort indexes 105
Tips and techniques 107
Partitions 107
Optimization 108
Techniques 108
A typical use case: time series data 108
Example of a mass DELETE 111
Summary 112
Chapter 5: MySQL Data Dictionary in MySQL 8.0 115

MySQL data dictionary structure in MySQL 8.0 115
Dictionary object cache 117
Transactional storage of the data dictionary 118
Applications of the data dictionary 119
Removal of file-based storage metadata 119
Serialized Dictionary Information (SDI) 120
Limitations of the data dictionary 121
Tips and techniques 121
Summary 122
Chapter 6: MySQL Server Settings 123
Getting started with the most significant variables 123
MySQL server optimization 125
Control the types of data change operations 125
Enabling the adaptive hash indexing function 125
Set a limit on the number of concurrent threads 126
Controlling the amount of InnoDB preloading 126
Increasing the number of background threads 126
Controlling InnoDB input/output performance in the background 127
Taking advantage of multicore processors 127
Preventing punctual operations 127
Configuring the number and size of instances 127
The InnoDB buffer pool 128
The thread cache 129
Case study 1 – when MySQL uses more than 100% of a CPU 130
How to detect high usage of the MySQL processor 131
Correcting the use of the MySQL CPU 132
How to prevent MySQL from using high CPUs 132
Case study 2 – when MySQL swaps on disk 132
Tips and techniques 133
Summary 134
Chapter 7: Group Replication in MySQL 8.0 135
High availability and requirements 136
Scaling 136
Replication 137
Group replication 138
Use cases for group replication 140
Elastic replication 140
Highly available shards 141
Alternative to master – slave replication 142
Autonomic systems 142
An overview of MySQL's database replication 142
Asynchronous replication 144
Semi-synchronous replication 144
Delayed replication 145

- Global transaction identifier-based replication 145
- Multi-source replication 146

MySQL's group replication architecture 147

- Group 148
- Writeset 148
- How group communication works 149
- Certification process 151
  - Total order delivery 151
  - Detecting failure 152
    - Network partitioning 153
  - Traditional locking versus optimistic locking 154
    - Distributed first commit wins rule 154
    - Drawbacks of optimistic locking 155
- Modes of group replication 155
  - Single primary mode 156
  - Multi-primary 157
- Group replication requirements 157
- Configuring the server 158
- Configuring group replication 158
- Monitoring group replication 166
  - Replication_group_members 166
    - replication_group_member_stats 166
  - Replication_connection_status 167
  - Replication_applier_status 167
    - Server state 167

Limitations of group replication 168

Group replication security 169

- IP address whitelist 169
- SSL 170
- VPN 171

Operations on an online group 171

- Changing the group mode 172
- Tuning recovery 172
- Combining group replication versions 173
- Performance tuning 174
- Message compression 174
- Flow control 175

Summary 178

Chapter 8: InnoDB Cluster in MySQL 8.0 181

What is InnoDB cluster? 182

- InnoDB cluster requirements 184
  - Installing MySQL Shell 184
    - How to use MySQL Shell 185
- Installing an InnoDB cluster 188
  - MySQL InnoDB cluster for a sandbox environment 188
  - InnoDB cluster in a production environment 195
  - Configuring the router configuration 199

Managing clusters 201
Getting details of a cluster 202
Removing instances from a cluster 202
Adding instances to a cluster 202
Restoring a cluster after quorum loss 202
Rebooting a cluster after a major outage 202
Rescanning a cluster 202
Checking instance states 203
Dissolving an InnoDB cluster 203
InnoDB cluster limitations 203
Storage engines 204
Setting a storage engine 205
MyISAM storage engine 205
The MEMORY storage engine 206
The CSV storage engine 206
The ARCHIVE storage engine 207
The BLACKHOLE storage engine 207
The MERGE storage engine 207
The FEDERATED storage engine 208
InnoDB engine 208
Migrating from master-slave replication to MySQL InnoDB cluster 209
Summary 215
Chapter 9: Monitoring Your Large Distributed Databases 217
MONyog 218
Pros 219
Cons 220
Conclusion 221
Datadog 222
Getting started 222
Pros 224
Cons 225
Conclusion 225
Navicat 225
Pros 227
Cons 228
Conclusion 228
Comparison between monitoring tools 229
The price 229
Pros 229
Cons 230
Top clients 230
Tips and techniques 231
Summary 232
Chapter 10: Authentication and Security Management with MySQL 8.0 233
MySQL 8.0 security features 233

Privileges provided by MySQL 8.0 234
Where are privileges stored in MySQL 8.0? 235
The differences between dynamic and static privileges 236
Creating roles and users in MySQL 8.0 237
Displaying assigned roles using SHOW GRANTS 239
Troubleshooting connection problems 240
Tips and techniques 241
Restricting or disabling remote access to the server 241
Disabling LOCAL INFILE usage 241
Changing the username and password for root 242
Summary 242
Chapter 11: Advanced MySQL Performance Tips and Techniques 243
Tips/best practices 243
Optimizing your queries for the query cache 244
EXPLAIN your SELECT queries 245
LIMIT 1 when getting a unique row 247
Indexing for search fields 248
Indexing strategy for Joins 249
Avoiding SELECT * and COUNT * 250
Almost always have an ID field 253
Using ENUM over VARCHAR 255
Using prepared statements if and when possible 256
Splitting the big DELETE or INSERT queries 257
Avoiding the delete trigger 259
Techniques 260
Can MySQL perform queries on billions of rows? 260
Is InnoDB the right choice for multi-billion rows? 261
How big can a MySQL database get before the performance starts to degrade? 261
Why MySQL could be slow with large tables 261
Is MySQL the best solution for handling blobs? 261
Summary 262
Other Books You May Enjoy 263
Index 267