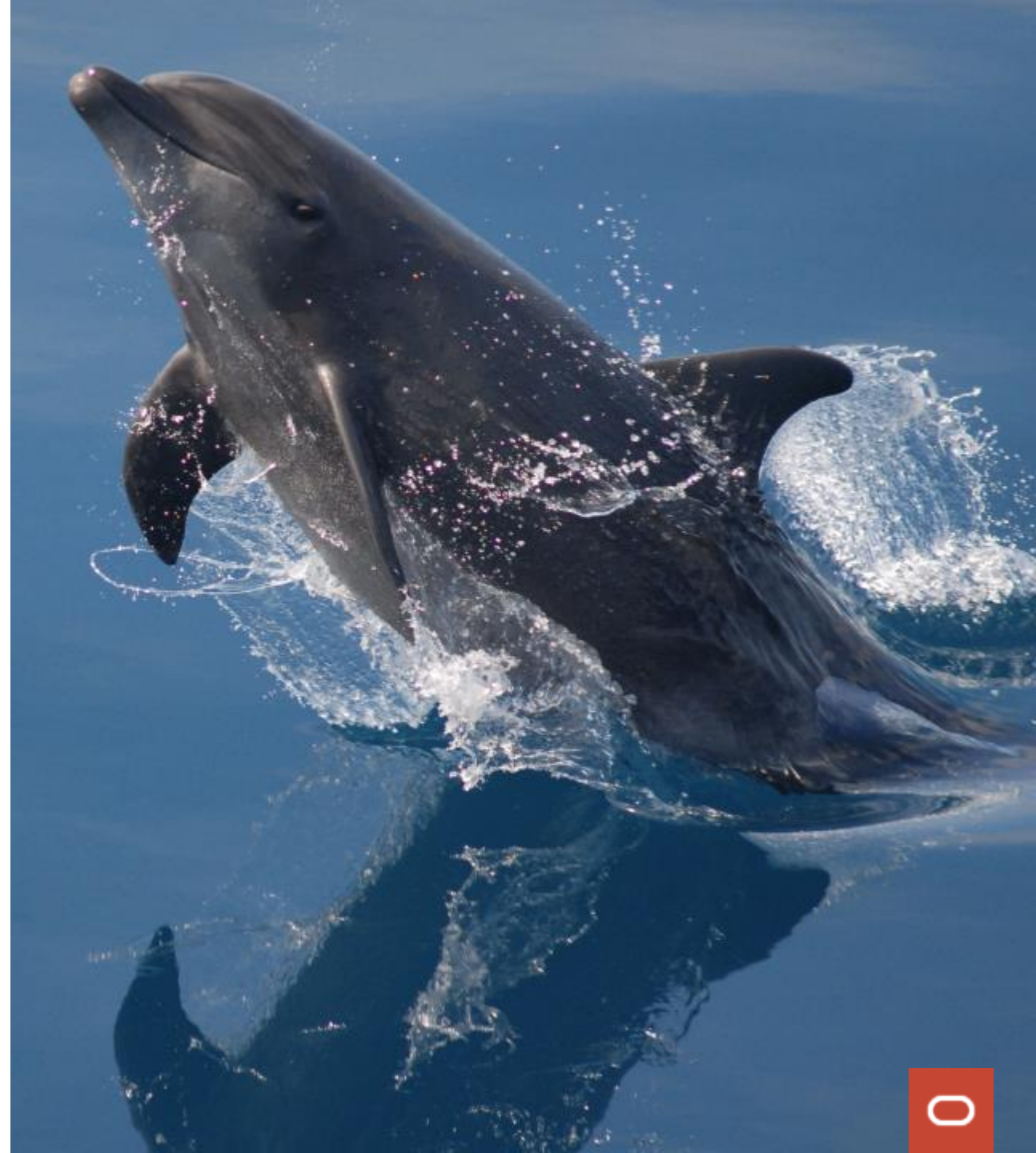ORACLE

MySQL Best Practices
In 8 Schritten zur optimierten DB!

Carsten Thalheimer

Oracle MySQL Business Unit

Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

State of Open-Source RDBMSs, 2015 – Gartner:

Open-source relational DBMSs have matured significantly and can be used to replace commercial RDBMSs at a considerable TCO saving. Information leaders, DBAs and application development management can now consider them as a standard choice for deploying applications.

Gartner Strategic assumptions for <u>2018</u>:
[...] more than 70% of new in-house applications will be developed on an OSDBMS
and [...] 50% of existing commercial RDBMS instances will have been converted or will be in process

https://www.gartner.com/doc/3033819/state-opensource-rdbmss-

ORACLE®

350 systems in ranking, December 2019

Rank			DBMS	Database Model	Score		
Dec 2019	Nov 2019	Dec 2018			Dec 2019	Nov 2019	Dec 2018
1.	1.	1.	Oracle ➕	Relational, Multi-model ℹ	1346.39	+10.33	+63.17
2.	2.	2.	MySQL ➕	Relational, Multi-model ℹ	1275.67	+9.38	+114.42
3.	3.	3.	Microsoft SQL Server ➕	Relational, Multi-model ℹ	1096.20	+14.29	+55.86
4.	4.	4.	PostgreSQL ➕	Relational, Multi-model ℹ	503.37	+12.30	+42.74
5.	5.	5.	MongoDB ➕	Document, Multi-model ℹ	421.12	+7.94	+42.50
6.	6.	6.	IBM Db2 ➕	Relational, Multi-model ℹ	171.35	-1.25	-9.40
7.	7.	⬆8.	Elasticsearch ➕	Search engine, Multi-model ℹ	150.25	+1.85	+5.55
8.	8.	⬇7.	Redis ➕	Key-value, Multi-model ℹ	146.23	+1.00	-0.59
9.	9.	9.	Microsoft Access	Relational	129.47	-0.60	-10.04
10.	10.	⬆11.	Cassandra ➕	Wide column	120.71	-2.52	-1.10

MySQL is #2 most widely used database …

ORACLE®

Stack Overflow Developer Survey
Developer Survey Results

stackoverflow

Databases

| All Respondents | Professional Developers |

Database	%	
MySQL	54.0%	
PostgreSQL	34.3%	
Microsoft SQL Server	32.8%	
SQLite	31.6%	
MongoDB	25.5%	
Redis	18.6%	
MariaDB	16.5%	
Oracle	16.5%	
Elasticsearch	14.3%	
Firebase	12.8%	
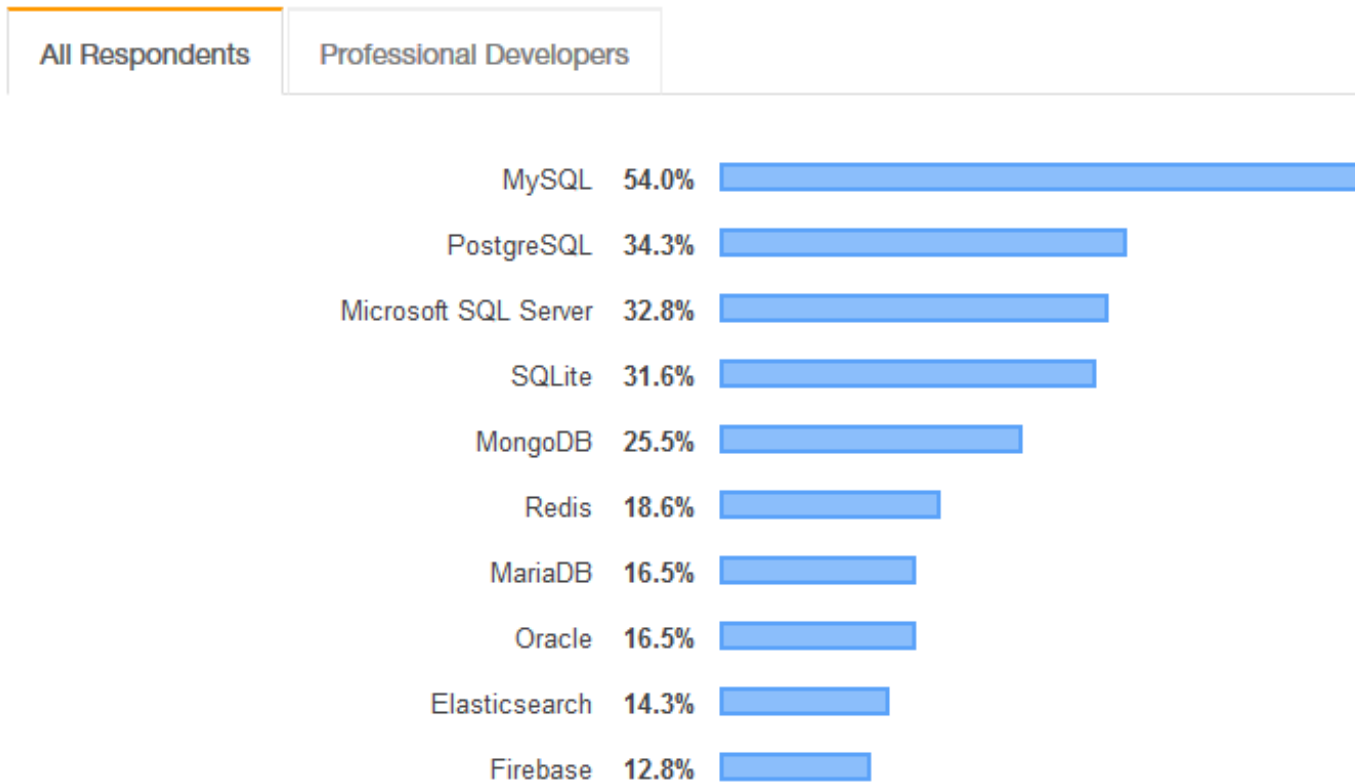
...but **MySQL is the #1** choice for development

ORACLE®

MySQL Innovation: 5.7 → 8.0

2018

2016

MySQL 8.0
- **Still GNU GPL v2**
- **Data Dictionary**

2015

MySQL 5.7 GR Plugin
- MySQL Group Replication
- MySQL X-Protocol (DMR)

- Roles
- **Unicode**

MySQL 5.7

- CTEs
- Window Functions

- 3x Better Performance
- Replication Enhancements
- Optimizer Cost Model
- JSON Support
- Improved Security
- Sys & Performance Schema
- GIS

2017

- Security
- Replication
- SysSchema
- GIS
- MySQL X

MySQL InnoDB Cluster
- MySQL Router
- MySQL Shell

- **Document Store**

ORACLE

MySQL 8.0 since 8.0.11 GA (04/18)
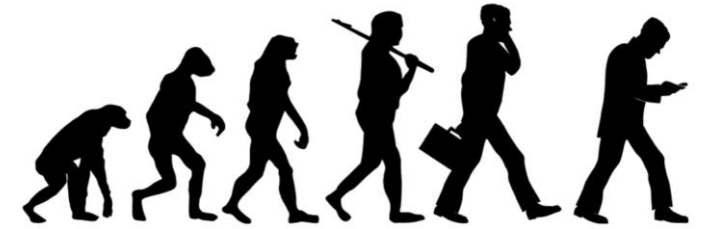
- **8.0.12** 07/18   Instant Add Columns, Query Rewrite Plugin+

- **8.0.13** 10/18   SQL Functional Indexes, Importing MongoDB Data

- **8.0.14** 01/19   Consistent Reads, Parallel read of index

- **8.0.16** 04/19   CHECK constraints, Support of TLS 1.3

- **8.0.17** 07/19   Provisioning by Cloning, Multi-valued indexes

- **8.0.18** 10/19   Hash join, MEB Page Tracking

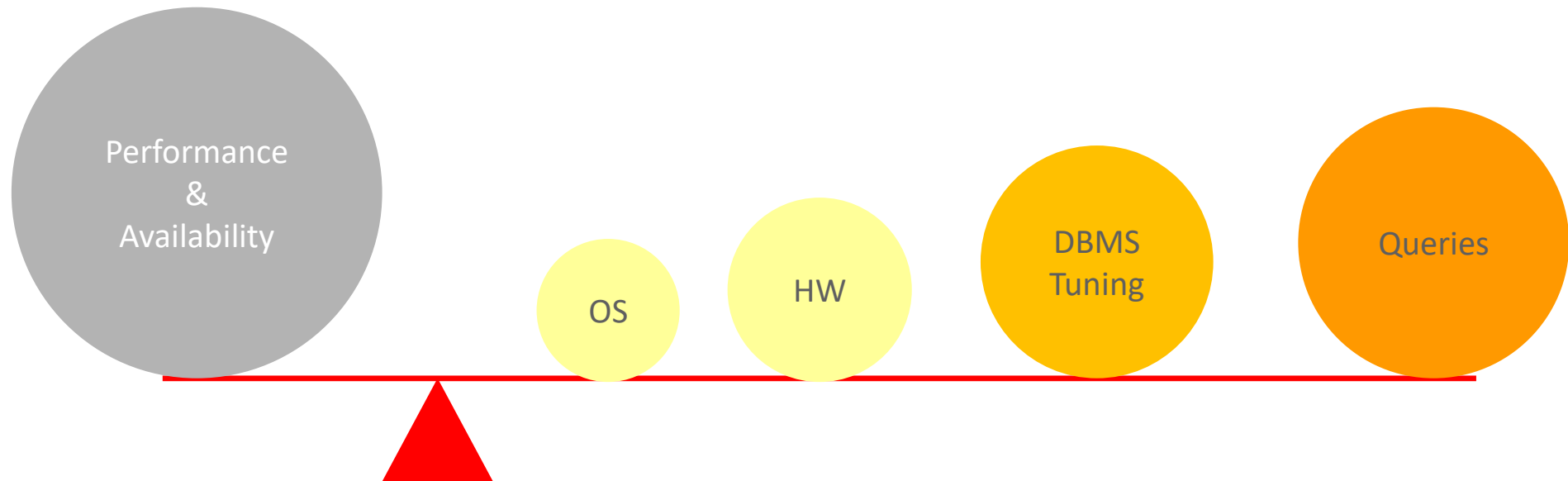- **8.0.19** 01/00   ReplicaSets, MTS: slave-preserve-commit

ORACLE®

Agenda

- MySQL Status Quo (Januar 2020)

- MySQL "Best Practices"

ORACLE®

Performance & Availability

Queries are most important …

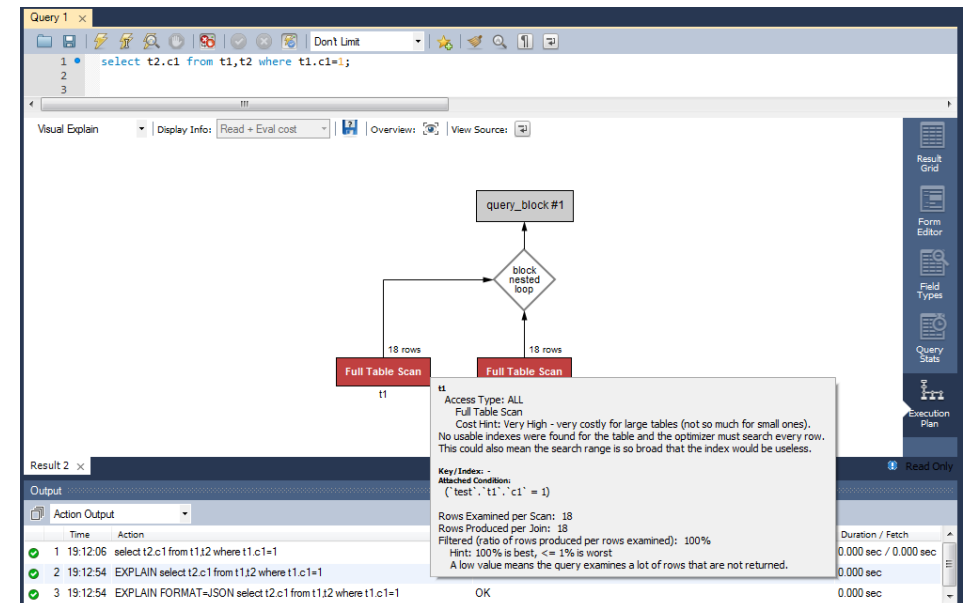*…. but not really relevant for todays presentation :-)*

Breakout: Queries

Use your brain !

- Slow Query log

- Monitoring Tools

- Performance/SYS Schema

- EXPLAIN shows the selected plan
  TRACE shows WHY the plan was selected:
  Alternative plans
  Estimated costs
  Decisions made

ORACLE®

Disclaimer:

- Dan McKinley - http://boringtechnology.club/
  Choose Boring Technology
  or: How to be old, for young people!

  KISS - an acronym for "**keep it simple, stupid**"

- Dimitri Kravchuk
  http://dimitrik.free.fr/blog/
  MySQL 8.0 Performance: Tuning for High Load

ORACLE®

MySQL Environment

- Dedicated vs. Shared OS (Standalone DB vs. DB & App)

- Bare metal vs. Virtualization vs. Container vs. Cloud

- OS - Linux versus ??? [but use a modern one]

- CPUs - More is better & Faster it better

- Memory - More is better

- Disks - You should have some

- Ext4 vs. XFS

- Network – it depends (we'll discuss later)

ORACLE®

OS tunings

- **Max open files per process/user** /etc/security/limits.conf (ulimit -n / -u)

- **Swappiness** /etc/sysctl.conf (vm.swappiness = 0)

- **OOM killer** - sudo systemctl edit mysqld.service (OOMScoreAdjust=-1000)

- **OS User for MySQL** (Tuning?)

- **[Consider using alternative malloc]**
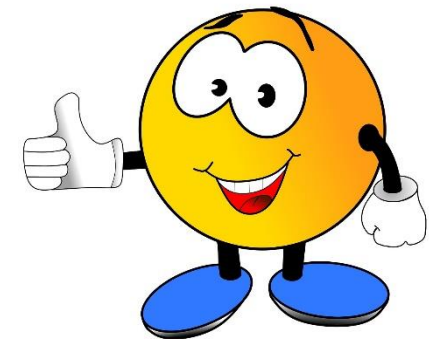 malloc-lib=/usr/lib64/libjemalloc.so

Tuning wizards?

ORACLE®

MySQL Version

- MySQL vs. *Other* DB World

- MySQL 5.7 vs. MySQL 8.0 **(utf8mb4)**

- MySQL Repo vs. Linux & MySQL

→ *rpm Installation*

→ *Manual installation (tar ball)*

→ *MySQL yum repo*

…from https://MySQL.com

ORACLE®

my.cnf vs. mysqld-auto.cnf
SET PERSIST for global variables

- All settings (next slides) belongs to the [mysqld] section

- No duplicate settings in the configuration file

- A primary key per table is mandatory (and can be forced in late MySQL 8)

- Main performance comes from good queries not Hardware/OS

- Monitor your MySQL dbs (show global status / show variables)

Important settings in my.cnf

- port=3306 (default)

Comments:

- Use a MySQL Router in front of MySQL

- Default 3306 [**3307**]

- Straight forward

- may need to be tweaked when using router:
  App -> 3306 -> Router -> xyz -> MySQLd

MySQL and Memory – Avoid OS swapping

Global	Per Connection
Server Start	As Needed
Large Values	Small Values
Allocated Once	Allocated 0-N Times

Global Memory + (Max Connections * Per Connection Buffers)

ORACLE®

Important settings in my.cnf

- port=3306 (default)

- *placeholder*
  - innodb_buffer_pool_size=xyz G

Comments:

- Default ~128M, [better **75% of memory**]

- Disks are still much slower than memory

- Online dataset and indexes kept in cache

- Will make sure you use memory for most reads

- The bigger the better (?) - avoid OS Caching

- Monitor Dirty Pages

ORACLE®

Important settings in my.cnf

- port=3306 (default)

- *placeholder*
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M

Comments:

- Default 100M (2x 50M), 2x ib_logfile [**256M** each]

- Transaction log

- Equivalent to Redo logs

ORACLE

Important settings in my.cnf

- port=3306 (default)

- *placeholder*
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT

Comments:

- Default fsync [**O_DIRECT**]

- No overhead of double buffering

- The better the fsync, the faster...

- O_DIRECT is often better for direct attached storage

ORACLE®

Important settings in my.cnf

- port=3306 (default)
- **innodb_dedicated_server**
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT

recommended if the MySQL instance shares system resources with other applications.

The information that follows describes how each variable is automatically configured.

- innodb_buffer_pool_size

Buffer pool size is configured according to the amount of memory detected on the server.

Table 15.8 Automatically Configured Buffer Pool Size

Detected Server Memory	Buffer Pool Size
Less than 1GB	128MiB (the default value)
1GB to 4GB	*detected server memory* * 0.5
Greater than 4GB	*detected server memory* * 0.75

- innodb_log_file_size

As of MySQL 8.0.14, log file size is configured according to the automatically configured buffer pool size.

Table 15.9 Automatically Configured Log File Size

Buffer Pool Size	Log File Size
Less than 8GB	512MiB
8GB to 128GB	1024MiB
Greater than 128GB	2048MiB

ORACLE

Important settings in my.cnf

- port=3306 (default)
- innodb_dedicated_server
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT
- innodb_flush_log_at_trx_commit=x

Comments:

- Default 1 (ACID compliant) [**1/2**]
- fysnc of transaction log
- 1 for durability
- 2 normally significant faster >30% for writes (but you may loose 1sec of info)
- can often be reduced (e.g. MyISAM use cases)

ORACLE®

Important settings in my.cnf

- port=3306 (default)
- innodb_dedicated_server
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT
- innodb_flush_log_at_trx_commit=x
- skip_name_resolve=x

Comments:

- Default 0 [**1**]
- DNS Server can be critical!
- Customer often have simplified user management
- GRANT Statements must use IP addresses only (!)

Important settings in my.cnf

- port=3306 (default)
- innodb_dedicated_server
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT
- innodb_flush_log_at_trx_commit=x
- skip_name_resolve=x
- disable-log-bin / sync_Binlog=x

Comments:

- Default 1 [**0**]
- Binlog is key for PITR, Replication and Cluster
- Be careful!

ORACLE®

24

Important settings in my.cnf

- port=3306 (default)

- innodb_dedicated_server
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT

- innodb_flush_log_at_trx_commit=x

- skip_name_resolve=x

- disable-log-bin / sync_Binlog=x

- max_connections= x

Comments:

- Default 151

- MySQL Thread Pool
  - Improves performance on high concurrency workloads
  - Contains configurable number of thread groups [16]
  - Threads are prioritized and statements queued

- How to Trace? SHOW GLOBAL STATUS;

- - Connections|Threads_connected|Max_used_con

- Good value?

ORACLE

Important settings in my.cnf

- port=3306 (default)
- innodb_dedicated_server
  - innodb_buffer_pool_size=xyz G
  - Innodb_log_file_size=x M
  - innodb_flush_method= O_DIRECT
- innodb_flush_log_at_trx_commit=x
- skip_name_resolve=x
- disable-log-bin / sync_Binlog=x
- max_connections= x
- [Innodb_io_capacity=a]
- [innodb_io_capacity_max=b]

Comments:

- Default 200 /2000
- Dirty pages - Clean/Data ratio
- innodb_io_capacity=500
- innodb_io_capacity_max=3000

Agenda

- MySQL Status Quo (August 2019)

- MySQL "Best Practices"

  Breakout: MySQL Shell

MySQL Shell 8.0

Prompt Themes

Display useful information

- Session data
- Status, global or session variables.
- Environment variables.
- User defined variables.

Still there ☺ :
- *MySQL [client]*
- *MySQL Workbench (8.0)*

Agenda

- MySQL Status Quo (August 2019)

- MySQL "Best Practices"

  Breakout: MySQL Shell
  Breakout: MySQL HA

ORACLE®
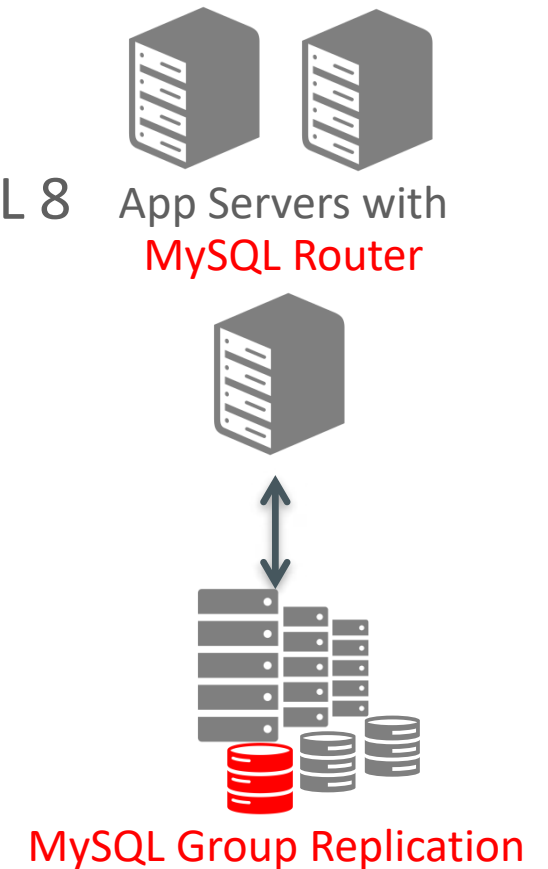
100%

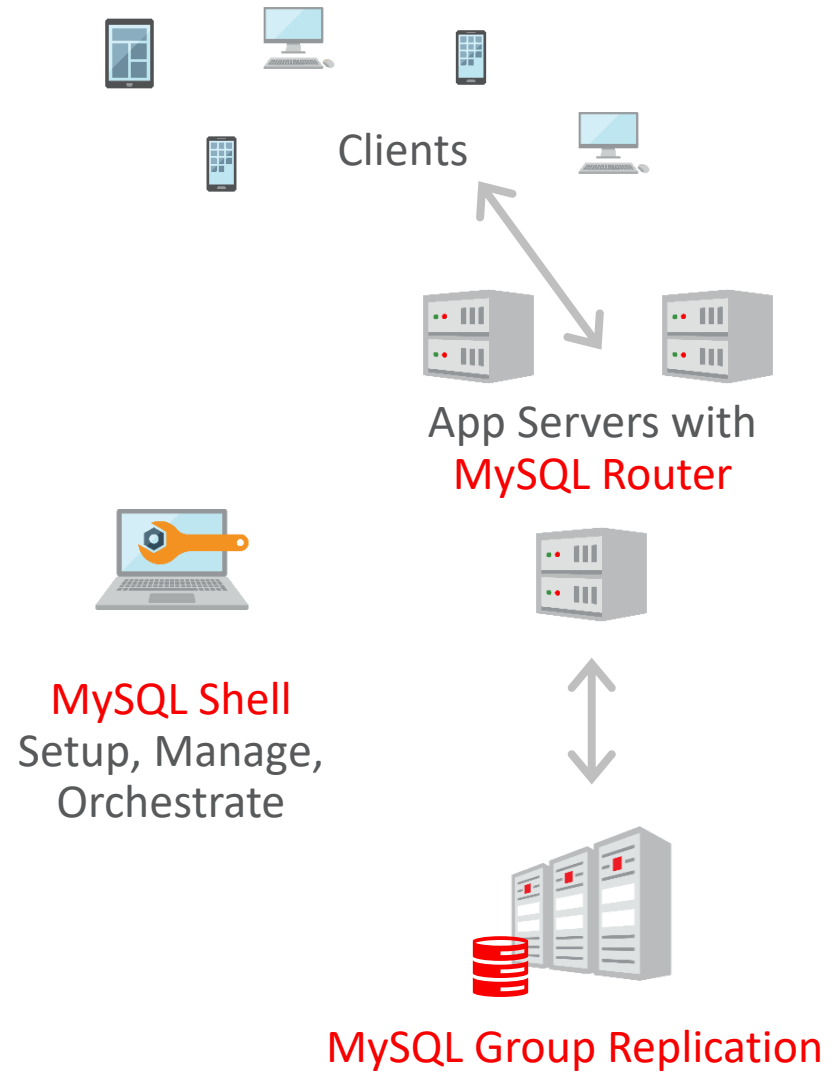Virtually all organizations require their most critical systems to be highly available

MySQL Group Replication: What Is It?

- Group Replication library
  - Implementation of Replicated Database State Machine theory
  - Provides *virtually* synchronous replication for MySQL 5.7+ & MySQL 8
  - It's Open Source
  - It's easy

- But...
  - 3x resources (two datacenter solutions?)
  - Network is key
  - GTID are mandatory
  - Primary keys are mandatory
  - MySQL Shell is *kind of* mandatory

App Servers with
MySQL Router

MySQL Group Replication

ORACLE®

MySQL InnoDB Cluster

Clients

App Servers with **MySQL Router**

MySQL Shell
Setup, Manage,
Orchestrate

MySQL Group Replication

1.) mysqlsh
-
2.) dba.deploySandboxInstance(3501)
3.) dba.deploySandboxInstance(3502)
4.) dba.deploySandboxInstance(3503)
5.) \connect root@localhost:3501
-
6.) mc = dba.createCluster("mycluster")
7.) mc.addInstance("root@localhost:3502")
8.) mc.addInstance("root@localhost:3503")
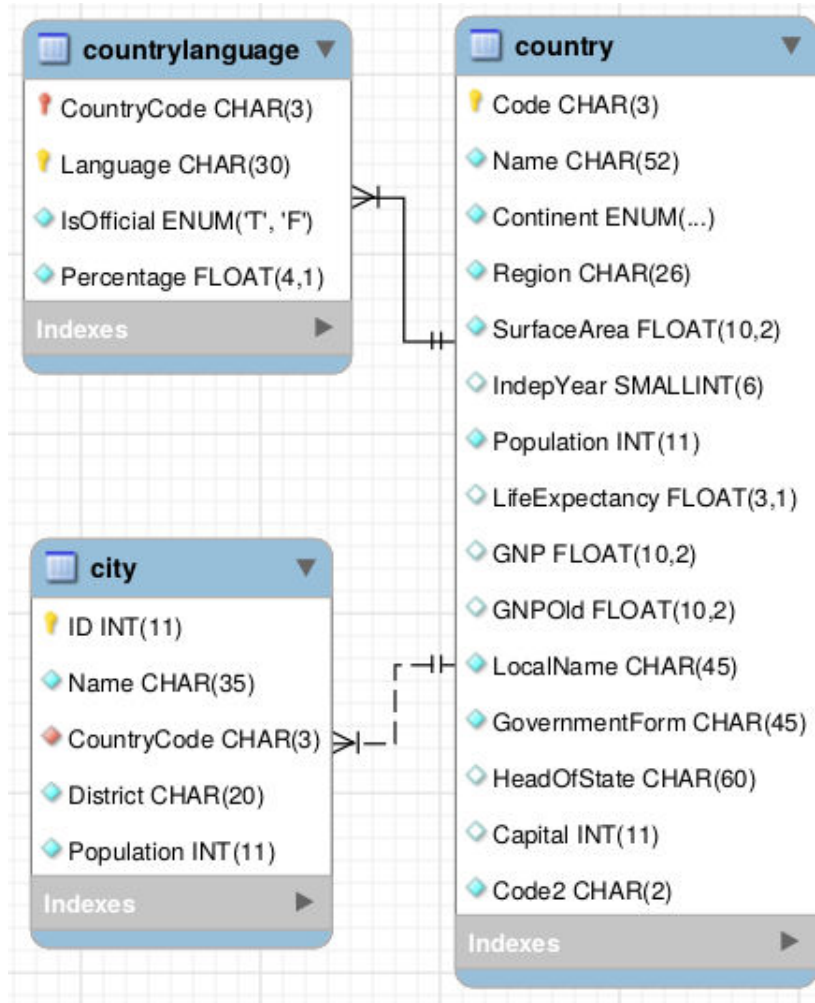-
mysqlrouter --bootstrap localhost:3501 --user=root

ORACLE®

Agenda

- MySQL Status Quo (August 2019)

- MySQL "Best Practices"

    Breakout: MySQL Shell
    Breakout: MySQL HA
    Breakout: MySQL DocumentStore

ORACLE®

How *DBAs* see data

How *Developers* see data

```json
{
    "GNP" : 249704,
    "Name" : "Belgium",
    "government" : {
        "GovernmentForm" :
            "Constitutional Monarchy, Federation",
        "HeadOfState" : "Philippe I"
    },
    "_id" : "BEL",
    "IndepYear" : 1830,
    "demographics" : {
        "Population" : 10239000,
        "LifeExpectancy" : 77.8000030517578
    },
    "geography" : {
        "Region" : "Western Europe",
        "SurfaceArea" : 30518,
        "Continent" : "Europe"
    }
}
```

MySQL Document Store demo "cheat sheet"

```
js> session.createSchema('name')
js> db.getCollections()
js> db.createCollection('myCollection')
js> db.myCollection.add({"param1":"value1", "param2":"value2"})
```

Create

```
js> db.myCollection.find()
js> db.myCollection.find().limit(1)
js> db.myCollection.find("_id = '00005af018430000000000000002'")
```

Read

Update

```
js> db.myCollection.modify("_id = '1234'").set("param","value")
```

Delete

```
js> db.myCollection.remove("_id = '1234'")
```

```
js> session.startTransaction()
js> …
js> session.rollback()
```

Transaction

ORACLE®

Agenda

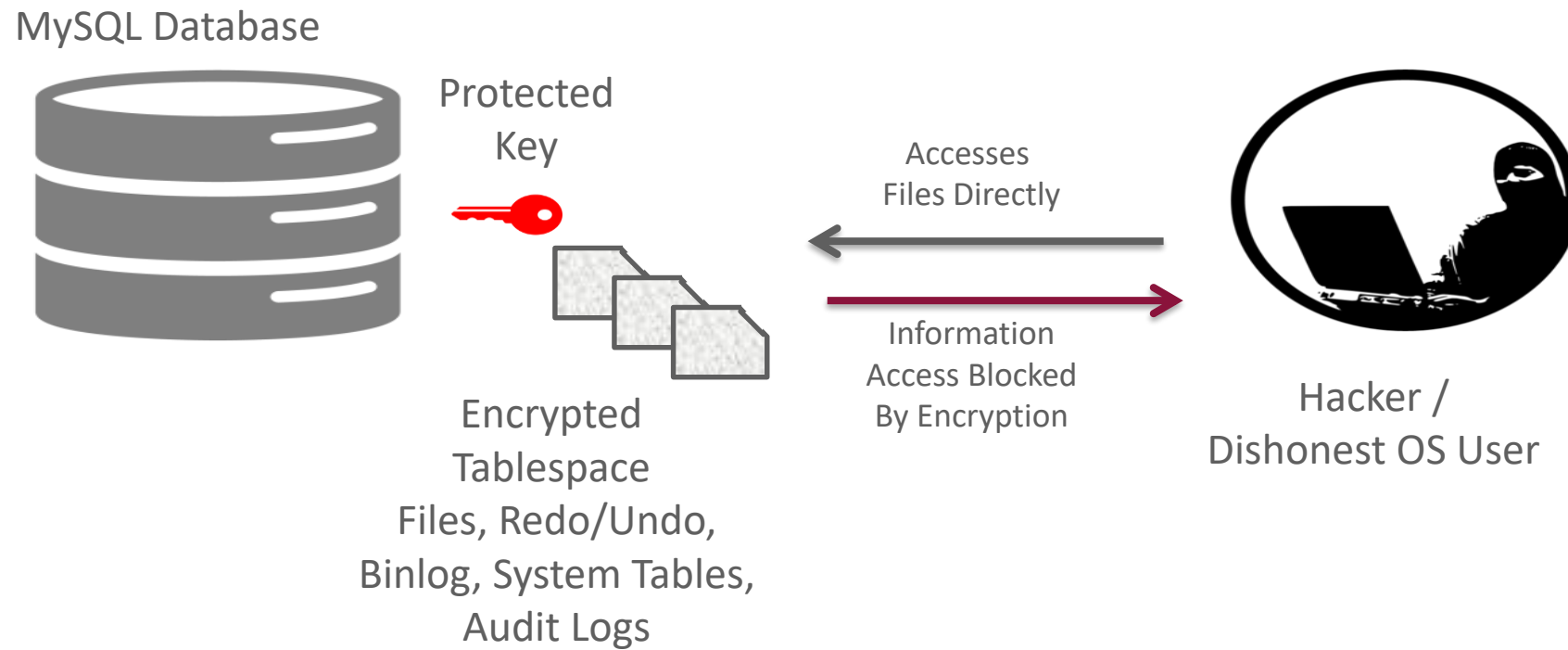- MySQL Status Quo (August 2019)

- MySQL "Best Practices"

  Breakout: MySQL Shell
  Breakout: MySQL HA
  Breakout: MySQL DocumentStore
  Breakout: MySQL Security

ORACLE®

Attack on Files

MySQL Database

Protected
Key

Encrypted
Tablespace
Files, Redo/Undo,
Binlog, System Tables,
Audit Logs

Accesses
Files Directly

Information
Access Blocked
By Encryption

Hacker /
Dishonest OS User

ORACLE®

MySQL Transparent Data Encryption Example

- 1. my.cnf config:
  - early-plugin-load=keyring_file.so
  - keyring_file_data=/var/lib/mysql-keyring/file

- 2. Create/Alter table
  - CREATE TABLE xyz (abc) encryption='y';

- Protect your key!

ORACLE®

Agenda

- MySQL Status Quo (Jan 2020)

- MySQL "Best Practices"

  Breakout: MySQL Shell
  Breakout: MySQL HA
  Breakout: MySQL DocumentStore
  Breakout: MySQL Security

ORACLE®

8 Schritte zur optimierten Datenbank?

0. OS Changes e.g. Max open files limit / Swappiness / OOM
1. Use the MySQL Yum Repository
2. my.cnf: innodb_dedicated_server
3. my.cnf: innodb_flush_log_at_trx_commit
4. my.cnf: skip_name_resolve=x
5. my.cnf: max_connections= x
6. Use a HA Solution (or Replication)
7. Consider DocumentStore
8. Consider Encryption of your data

There's more: Monitoring, Audit, User Mgmt, Authentication….

ORACLE®