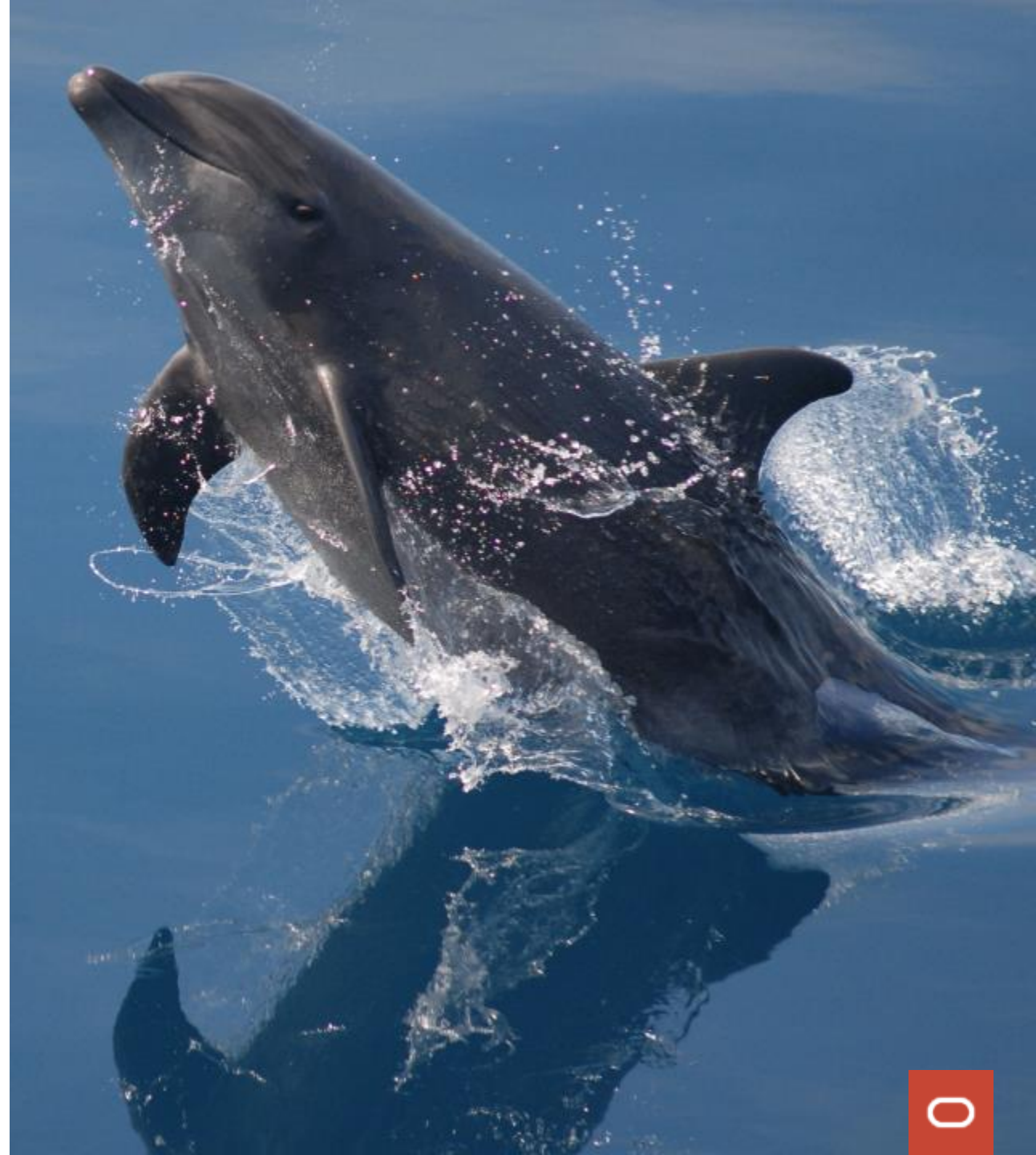


ORACLE



# 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 2018:*

*[...] 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->

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	1346.39	+10.33	+63.17
2.	2.	2.	MySQL	Relational, Multi-model	1275.67	+9.38	+114.42
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1096.20	+14.29	+55.86
4.	4.	4.	PostgreSQL	Relational, Multi-model	503.37	+12.30	+42.74
5.	5.	5.	MongoDB	Document, Multi-model	421.12	+7.94	+42.50
6.	6.	6.	IBM Db2	Relational, Multi-model	171.35	-1.25	-9.40
7.	7.	8.	Elasticsearch	Search engine, Multi-model	150.25	+1.85	+5.55
8.	8.	7.	Redis	Key-value, Multi-model	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

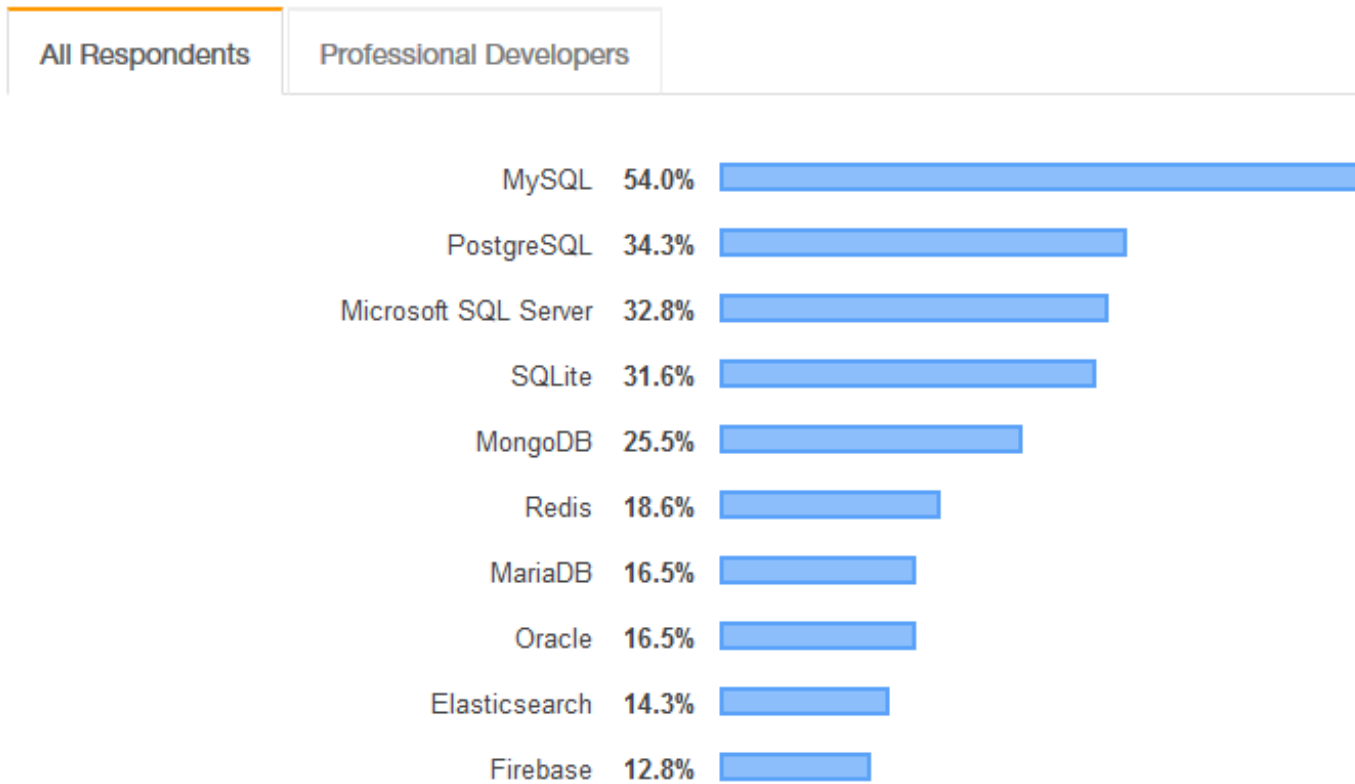
MySQL is #2 most widely used database ...

# Stack Overflow Developer Survey

## Developer Survey Results



### Databases



...but **MySQL** is the **#1** choice for development

<https://insights.stackoverflow.com/survey/2019/#technology--databases>

# MySQL Innovation: 5.7 → 8.0

2015

## MySQL 5.7

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

2016

## MySQL 5.7 GR Plugin

- MySQL Group Replication
- MySQL X-Protocol (DMR)

2017

## MySQL InnoDB Cluster

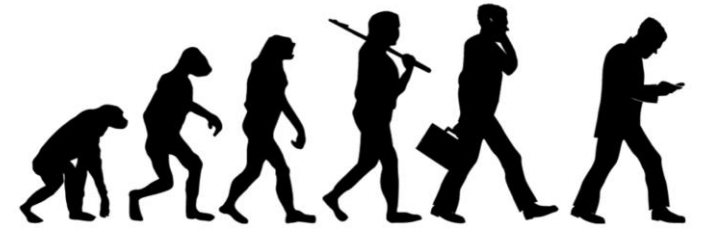
- MySQL Router
- MySQL Shell

2018

## MySQL 8.0

- Still GNU GPL v2
- Data Dictionary
- Roles
- Unicode
- CTEs
- Window Functions
- Security
- Replication
- SysSchema
- GIS
- MySQL X
- Document Store

# 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

# Agenda

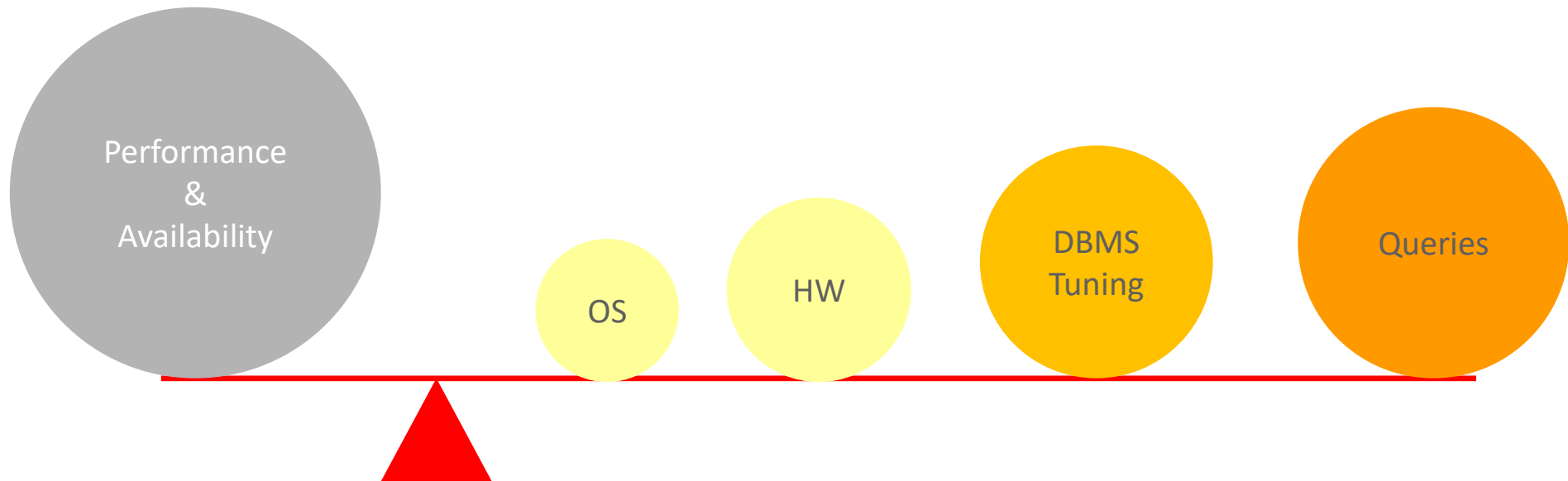
- MySQL Status Quo (Januar 2020)
- **MySQL “Best Practices”**



# Performance & Availability

*Queries are most important ...*

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



# Breakout: Queries

- 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*

*Use your brain !*



Query 1 x

```
1 select t2.c1 from t1,t2 where t1.c1=1;
```

Visual Explain | Display Info: Read + Eval cost | Overview: | View Sources

query\_block #1

block nested loop

Full Table Scan (18 rows) t1

Full Table Scan (18 rows) t2

Access Type: ALL  
Full Table Scan  
Cost Hint: Very High - very costly for large tables (not so much for small ones).  
No usable indexes were found for the table and the optimizer must search every row.  
This could also mean the search range is so broad that the index would be useless.

Key/Index: --  
Attached Condition:  
('test', 't1', 'c1' = 1)

Rows Examined per Scan: 18  
Rows Produced per Join: 18  
Filtered (ratio of rows produced per rows examined): 100%  
Hint: 100% is best, <= 1% is worst  
A low value means the query examines a lot of rows that are not returned.

Result 2 x

Time	Action	Duration / Fetch
1 19:12:06	select t2.c1 from t1,t2 where t1.c1=1	0.000 sec / 0.000 sec
2 19:12:54	EXPLAIN select t2.c1 from t1,t2 where t1.c1=1	0.000 sec
3 19:12:54	EXPLAIN FORMAT=JSON select t2.c1 from t1,t2 where t1.c1=1	0.000 sec

## 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](#)



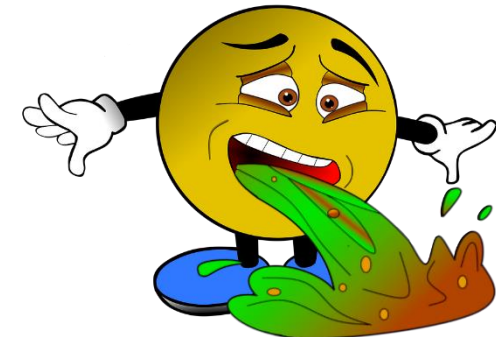
# MySQL Environment

- 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)

## 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?*



# MySQL Version

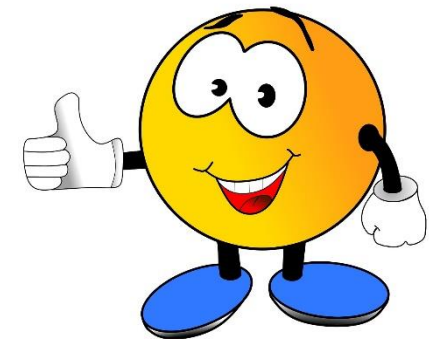
- MySQL vs. *Other* DB World
- MySQL 5.7 vs. MySQL 8.0 (**utf8mb4**)
- MySQL Repo vs. Linux & MySQL

→ *rpm Installation*

→ *Manual installation (tar ball)*

→ *MySQL yum repo*

*...from <https://MySQL.com>*



# 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)

# 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

# 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

# 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

# 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

https://dev.mysql.com/doc/refman/8.0/en/innodb-dedicated-server.html

example, consider enabling if you run MySQL server in a Docker container or dedicated VM and only running MySQL 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	<i>detected server memory</i> * 0.5
Greater than 4GB	<i>detected server memory</i> * 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

# 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**]
- fsync of transaction log
- 1 for durability
- 2 normally significant faster >30% for writes (but you may lose 1sec of info)
- can often be reduced (e.g. MyISAM use cases)

# 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!



# 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?

# 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

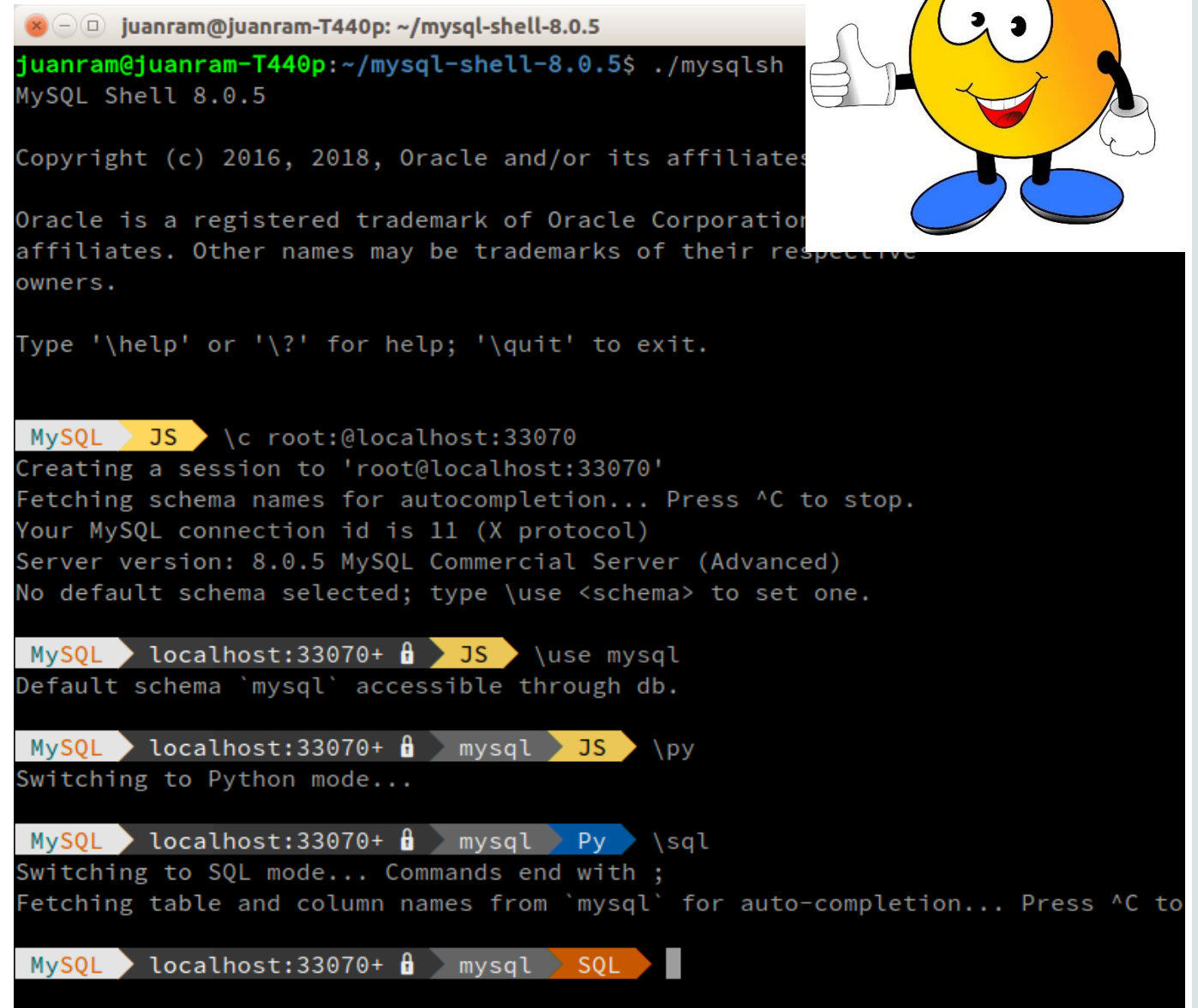
## Prompt Themes

### Display useful information

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

Still there 😊 :

- *MySQL [client]*
- *MySQL Workbench (8.0)*



```
juanram@juanram-T440p: ~/mysql-shell-8.0.5
juanram@juanram-T440p:~/mysql-shell-8.0.5$ ./mysqlsh
MySQL Shell 8.0.5

Copyright (c) 2016, 2018, Oracle and/or its affiliates

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type '\help' or '\?' for help; '\quit' to exit.

MySQL JS \c root:@localhost:33070
Creating a session to 'root@localhost:33070'
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 11 (X protocol)
Server version: 8.0.5 MySQL Commercial Server (Advanced)
No default schema selected; type \use <schema> to set one.

MySQL localhost:33070+ JS \use mysql
Default schema 'mysql' accessible through db.

MySQL localhost:33070+ mysql JS \py
Switching to Python mode...

MySQL localhost:33070+ mysql Py \sql
Switching to SQL mode... Commands end with ;
Fetching table and column names from 'mysql' for auto-completion... Press ^C to

MySQL localhost:33070+ mysql SQL
```

# Agenda

- MySQL Status Quo (August 2019)
- MySQL “Best Practices”

Breakout: MySQL Shell

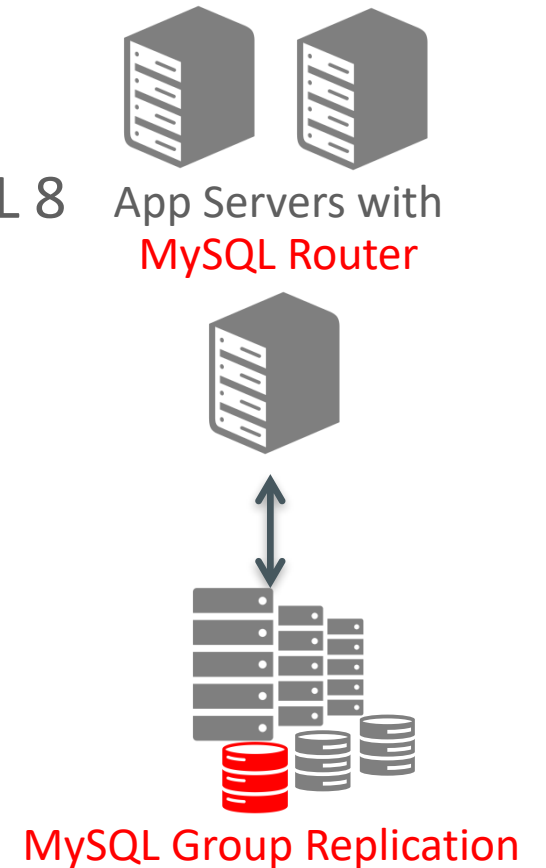
Breakout: MySQL HA

# 100%

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

# MySQL Group Replication: **What Is It?**

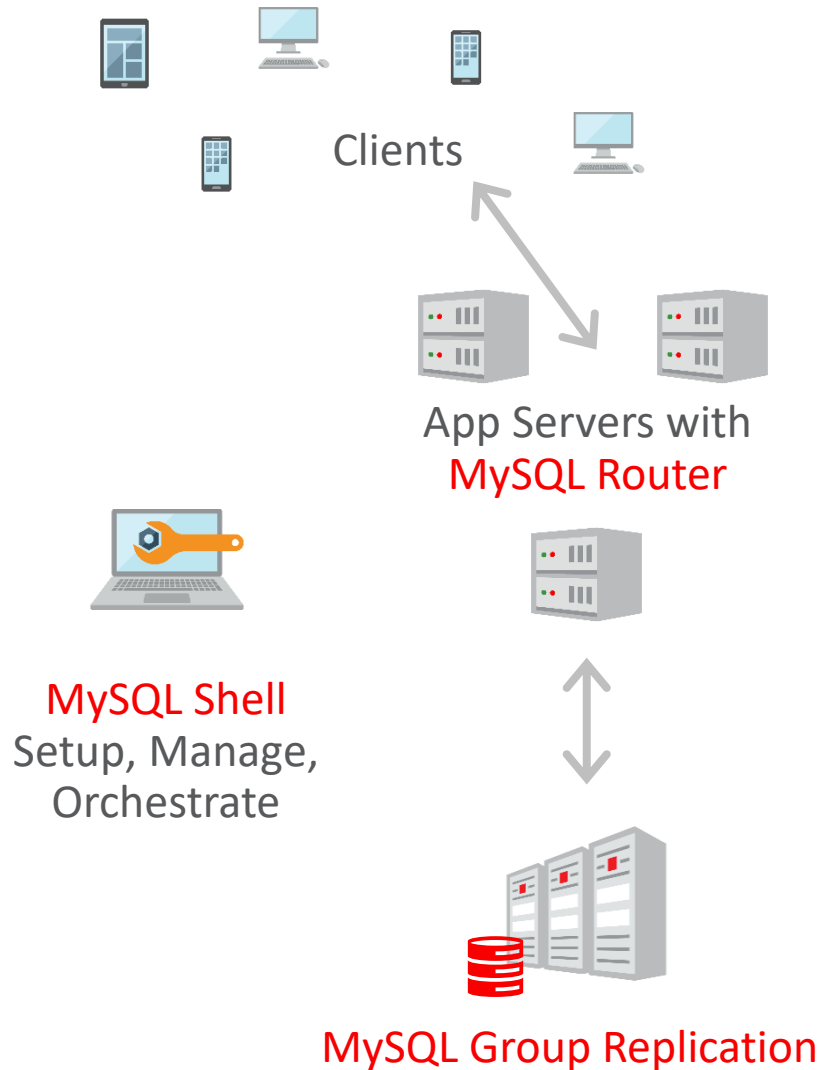
- Group Replication library
  - Implementation of Replicated Database State Machine theory
  - Provides *virtually* synchronous replication for MySQL 5.7+ & MySQL 8
  - It's Open Source
  - It's easy
- But...
  - 3x resources (two datacenter solutions?)
  - Network is key
  - GTID are mandatory
  - Primary keys are mandatory
  - MySQL Shell is *kind of* mandatory





# InnoDB Cluster

- 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





# Agenda

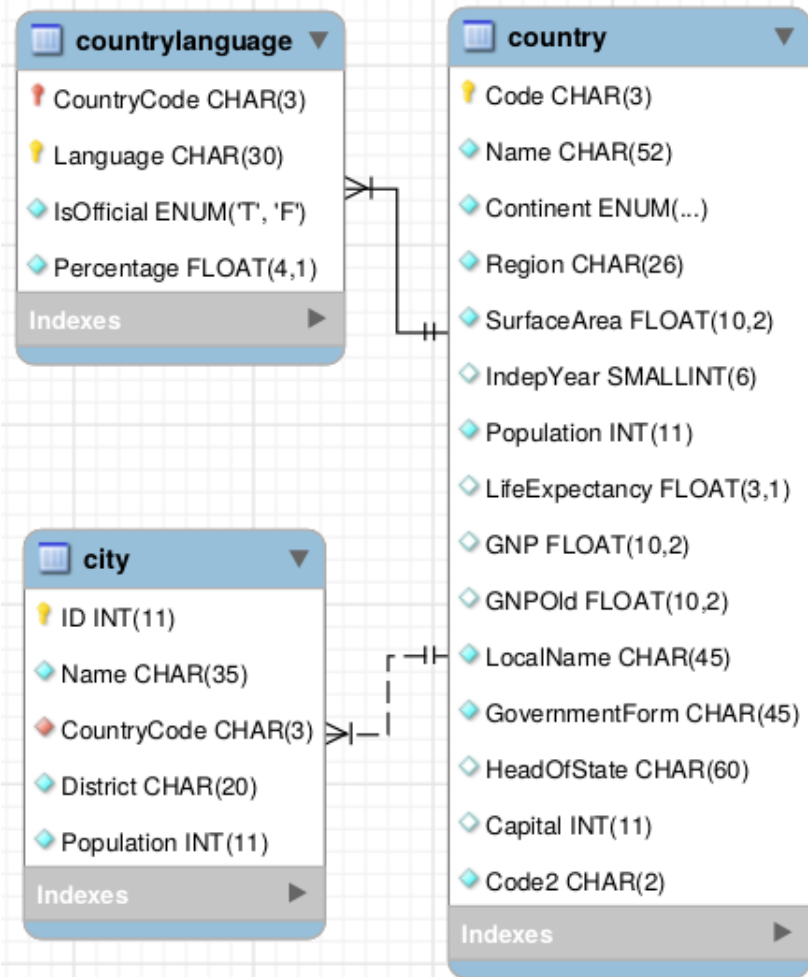
- MySQL Status Quo (August 2019)
- MySQL “Best Practices”

Breakout: MySQL Shell

Breakout: MySQL HA

**Breakout: MySQL DocumentStore**

# How *DBAs* see data



# How *Developers* see data

```
{
  "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 = '00005af0184300000000000000000002'")
```

Read

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

Update

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

Delete

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

Transaction

# Agenda

- MySQL Status Quo (August 2019)
- MySQL “Best Practices”

Breakout: MySQL Shell

Breakout: MySQL HA

Breakout: MySQL DocumentStore

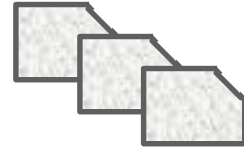
**Breakout: MySQL Security**

# 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

# 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!

# Agenda

- MySQL Status Quo (Jan 2020)
- MySQL “Best Practices”

Breakout: MySQL Shell

Breakout: MySQL HA

Breakout: MySQL DocumentStore

Breakout: MySQL Security



# 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....



감사합니다 Natick  
Grazie Danke Ευχαριστίες Dalu Obrigado  
Thank You Köszönöm  
Спасибо Dank Gracias  
谢谢 Merci Seé  
ありがとう

