



MySQL für Oracle DBA's

DOAG SIG 2015, Stuttgart

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com



FromDual GmbH

Support



Beratung



remote-DBA



Schulung



Inhalt

MySQL für Oracle DBA's

- **Geschichte**
- **Subskription**
- **Installation**
- **Storage Engines und InnoDB**
- **Konfiguration**
- **Backup, Restore, Recovery**
- **Monitoring**
- **Upgrade**
- **Tuning**
- **Security**
- **Hochverfügbarkeit**



Geschichte

- **Anfang in den 1990ern**
 - Als SQL Layer für C-ISAM Dateien
- **Ersatz von mSQL im LAMP Stack, um 2000**
- **Verschiedene Storage Engines in MySQL**
 - Innobase OY: InnoDB, ca. 2003
- **InnoDB Black Friday**
 - Oracle kauft Innobase OY, Nov 2005
- **Sun Microsystems kauft MySQL für USD 1 Mia, Apr 2008**
- **Oracle kauf Sun für USD 6 Mia, 2009**
 - MySQL und InnoDB jetzt vereint!
- **Kontinuität: 2010 – 2015 MySQL 5.6 und 5.7**
 - Verbesserung der Skalierbarkeit
 - Bessere Replikation
 - Ausmerzen von Kinderkrankheiten

Wer ist es?

- Wer/was seid Ihr?
- Was habt Ihr im Einsatz?

257 systems in ranking, March 2015

Rank			DBMS	Database Model	Score		
Mar 2015	Feb 2015	Mar 2014			Mar 2015	Feb 2015	Mar 2014
1.	1.	1.	Oracle	Relational DBMS	1469.09	+29.37	-22.71
2.	2.	2.	MySQL	Relational DBMS	1261.09	-11.36	-29.12
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1164.80	-12.68	-40.48
4.	4.	↑ 5.	MongoDB 	Document store	275.01	+7.77	+75.03
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	264.44	+2.10	+29.38
6.	6.	6.	DB2	Relational DBMS	198.85	-3.57	+11.52
7.	7.	7.	Microsoft Access	Relational DBMS	141.69	+1.15	-4.79
8.	8.	↑ 10.	Cassandra 	Wide column store	107.31	+0.23	+29.22
9.	9.	↓ 8.	SQLite	Relational DBMS	101.71	+2.14	+8.73
10.	10.	↑ 13.	Redis	Key-value store	97.05	-2.16	+43.59

<http://db-engines.com/en/ranking>

Branches und Forks

- **MySQL ist Open Source (GPL)**
 - d. h. jede/r darf selber
 - → Branches und Forks
 - Wer setzt MySQL Branch/Fork ein?
- **NICHT primäre kostenfrei sonder Wahlfreiheit!**
 - Echte Konkurrenz auf MySQL-Markt!
 - Wer setzt MySQL „free of cost“ ein?
- **Nicht Lizenz sondern eine Subskription (Abo) für Support + Tools**
 - Ausnahme: OEM, VAR, ISV

MySQL Subskriptionen

- **MySQL Community Edition (CE)**
 - Open Source MySQL Server
 - + hilf dir selber
- **MySQL Standard Edition (SE)**
 - Open Source oder Commercial MySQL Server
 - + MySQL Support
- **MySQL Enterprise Edition (EE)**
 - Standard Edition (SE)
 - + Enterprise Tools (Monitor, Backup, Audit, ...)
- **MySQL Commercial – OEM, ISV, VAR (Lizenz)**

Installation

- **Wichtiger Grundsatz bei MySQL: KISS!**
- **Oracle: OUI (MySQL zum Glück nicht!)**
- **MySQL:**
 - **Pakete der Distribution (RPM, DEB)**
 - **Binary-Tar-Ball (.tar.gz)**
 - **Source zum selber kompilieren**
 - **Win-Installer**
- **Repository direkt vom Hersteller**
 - <http://www.mysql.com/downloads>
 - <http://dev.mysql.com/downloads/repo/yum>



Erstellen der Datenbank

- **Distribution legt eine Datenbank an**
 - `/var/lib/mysql`
- **Von Hand:**
 - `mysql_install_db --user=mysql --datadir=...`
 - 5.7 soll sich das ändern...
- **Multi-Instanzen Setups vs. Lamp-Silos**
 - `myEnv`



Konfiguration

- **MySQL Konfigurationsdatei:**
 - `my.cnf` oder `my.ini` (Win)
 - Liegt unter: `/etc/` oder `/etc/mysql`
- **MySQL: kein SPFILE**
- **Änderung in `my.cnf` → Restart oder**
- **`SET GLOBAL variable = wert;`**
- **→ von Hand in `my.cnf` schreiben!**
- **Gutes Template:**
 - <http://fromdual.com/mysql-configuration-file-sample>



Starten/stoppen von MySQL

- Automatisch: in Boot-Prozess eingebunden

- Von Hand:

```
service mysql [start | stop]
```

- oder

- `mysqld_safe` → Angel-Prozess
- Neu: bei Systemd oder Upstart obsolet.

- Prüfen:

```
ps -ef | grep mysqld
```

- Killen von Hand (nett, dann hart):

```
kill -TERM <pid>; kill -KILL <pid>
```

- **Obacht bei NICHT crash-safem Storage Engines (MyISAM)!**



Prozess vs. Thread

- MySQL: Threads
- Oracle: Prozess (+ neu auch Threads)
 - RECO, PMON, SMON, DBW0, LGWR, ARC0, ...

```
shell> ps -eLf | grep mysqld
UID          PID    PPID    LWP   CMD
mysql        6411     1    6411  mysqld_safe
mysql        7081    6411    7081  mysqld
mysql        7081    6411    7083  mysqld
mysql        7081    6411    7084  mysqld
mysql        7081    6411    7085  mysqld
```

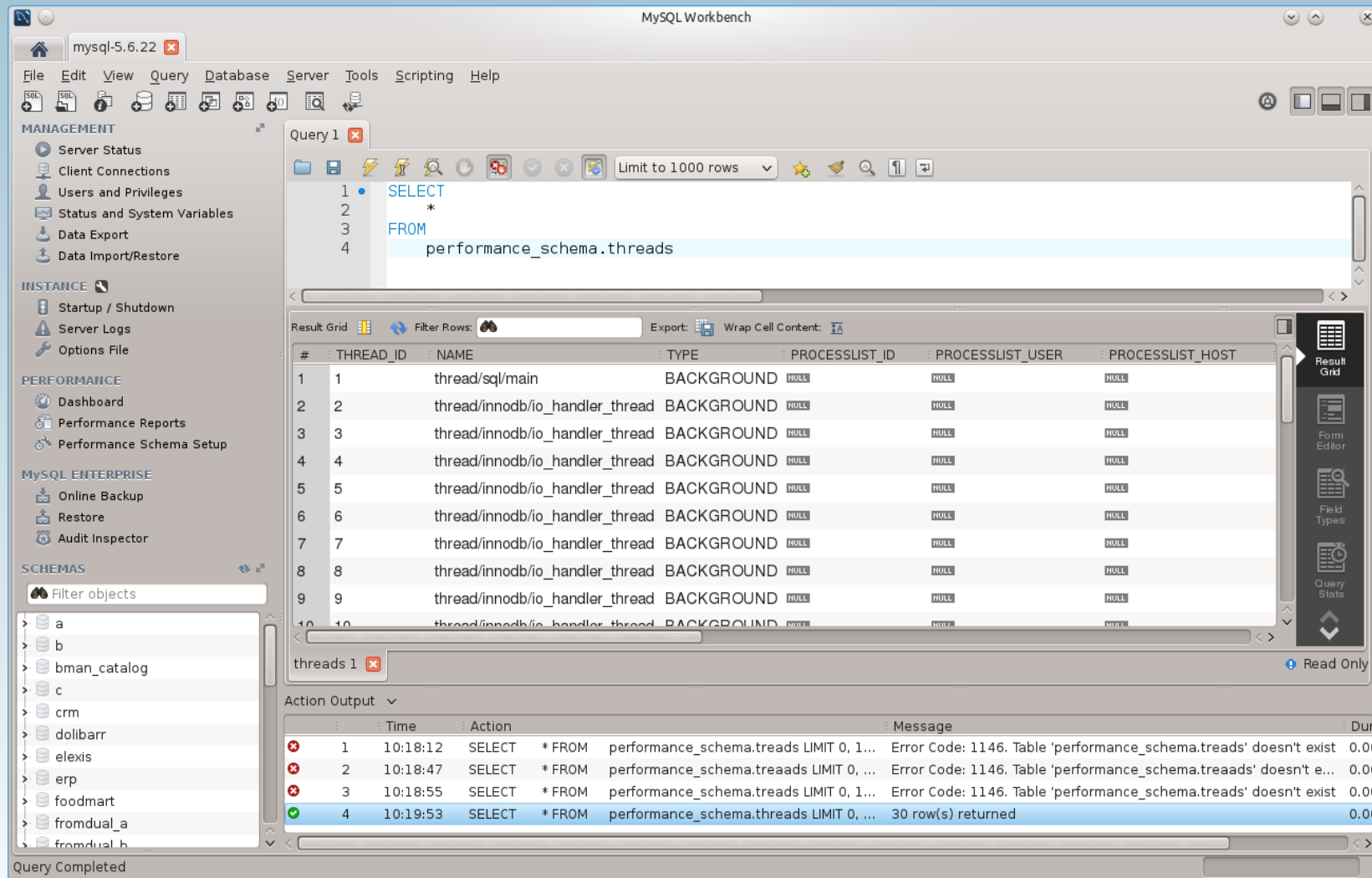
- Welche?

```
SELECT * FROM performance_schema.threads;
```

Zugriff auf MySQL

- CLI vs. GUI
- CLI: `mysql` (Oracle: `SQL*Plus`)
 - Nicht so mächtig wie `SQL*Plus`
 - `shell> mysql --user=... --password=... --host=... --port=... database`
 - `mysql> help`
- GUI: MySQL Enterprise Workbench (WB)
 - Administration (Backup, Monitoring, ...)
 - Entwicklung (SQL, Migration)
 - Designer (ER-Diagramme)

Query Browser



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```

1 SELECT
2 *
3 FROM
4 performance_schema.threads

```

The result grid shows a table with the following columns: #, THREAD_ID, NAME, TYPE, PROCESSLIST_ID, PROCESSLIST_USER, and PROCESSLIST_HOST. The table contains 10 rows of background threads.

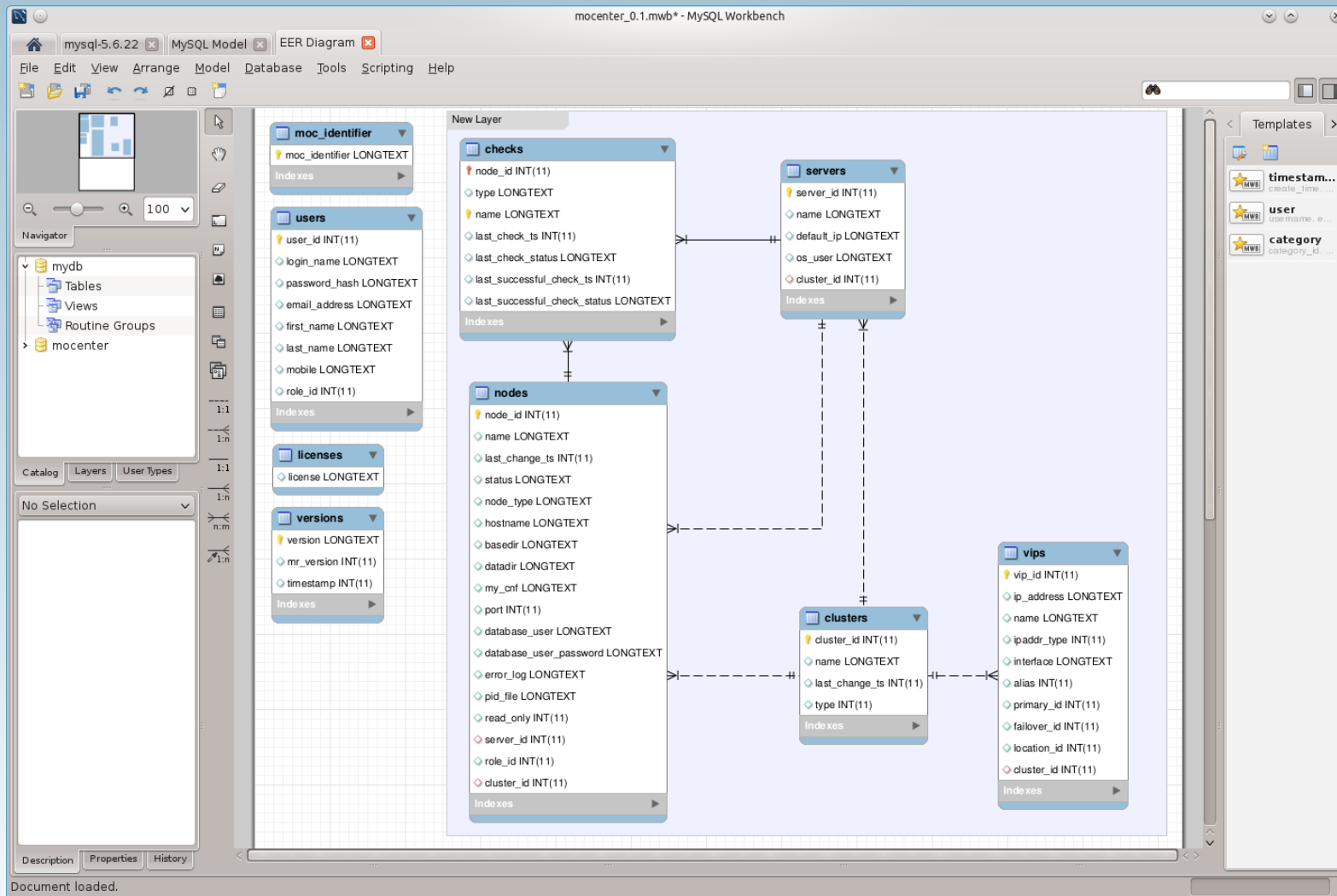
#	THREAD_ID	NAME	TYPE	PROCESSLIST_ID	PROCESSLIST_USER	PROCESSLIST_HOST
1	1	thread/sql/main	BACKGROUND	NULL	NULL	NULL
2	2	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
3	3	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
4	4	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
5	5	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
6	6	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
7	7	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
8	8	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
9	9	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL
10	10	thread/innodb/io_handler_thread	BACKGROUND	NULL	NULL	NULL

The Action Output pane shows the following log entries:

#	Time	Action	Message	Dur
1	10:18:12	SELECT * FROM performance_schema.treads LIMIT 0, 1...	Error Code: 1146. Table 'performance_schema.treads' doesn't exist	0.00
2	10:18:47	SELECT * FROM performance_schema.treaads LIMIT 0, ...	Error Code: 1146. Table 'performance_schema.treaads' doesn't e...	0.00
3	10:18:55	SELECT * FROM performance_schema.treads LIMIT 0, 1...	Error Code: 1146. Table 'performance_schema.treads' doesn't exist	0.00
4	10:19:53	SELECT * FROM performance_schema.threads LIMIT 0, ...	30 row(s) returned	0.00

The status bar at the bottom indicates "Query Completed".

ER-Diagrammer



Schema – Database – Instanz

- Database und Schema sind synonym
- Instanz = mysqld Prozess

```
mysql> show databases;
```

```
mysql> show schemas;
```

= Directory unter `$datadir (/var/lib/mysql)`

- DB Objekte gehören KEINEM User
 - → DB User



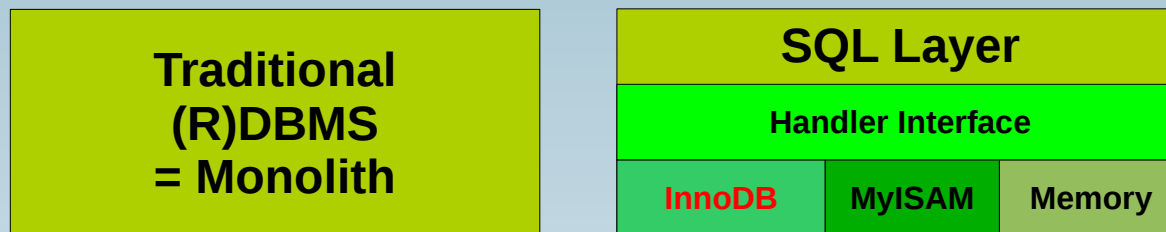
Wichtigste Befehle

- **SHOW DATABASES;**
- **use mysql;**
- **SHOW TABLES;**
- **SHOW [FULL] PROCESSLIST;**
- **SELECT user, host FROM mysql.user;**
- **SHOW GRANTS FOR user@host;**
- **SHOW GLOBAL VARIABLES;**
- **SHOW GLOBAL STATUS;**
- **SHOW ENGINES;**



Storage Engines

- MySQL Architektur ist speziell:



- Verschiedene Storage Engines (DB Engine): **InnoDB**, MyISAM, MEMORY

```
SELECT * FROM information_schema.tables;
```

- Früher MyISAM (≤ 5.1)
- heute: InnoDB (≥ 5.5)

```
SHOW CREATE TABLE test\G
CREATE TABLE `test` (
...
) ENGINE=InnoDB;
```

InnoDB Storage Engine

- **InnoDB Buffer Pool → Database Buffer Cache**
 - `innodb_buffer_pool_size`
- **InnoDB Log File (`ib_logfile?`) → Redo Logs**
 - `innodb_log_file_size`
- **InnoDB System Tablespace (`ibdata1`)**
- **InnoDB TS Konzept (`innodb_file_per_table`):**
 - alles im System TS (default bis 5.5)
 - 1 Tablespace pro Tabelle (default ab 5.6)
 - mehrere TS (ab 5.7)

Wichtigste Tuning-Parameter

- `innodb_buffer_pool_size` (default zu klein)
- `innodb_log_file_size` (defaults in 5.6 besser)
- `innodb_flush_log_at_trx_commit` (default langsam aber sicher)
- `sync_binlog` (default schnell aber unsicher)
- `key_buffer_size` (default zu klein für MyISAM DB)
- `table_open_cache` (default besser – gut mit 5.6)
- `table_definition_cache` (default besser – gut mit 5.6)
- `query_cache_size` (default off seit 5.6!) :-)

Backup / Restore

- Backup: logisch vs. physisch
- Physisches Backup mit MySQL Enterprise Backup (MEB)
- Backup:
 - logisch: `mysqldump > full_dump.sql`
 - physisch: `mysqlbackup backup-and-apply-log`
- Restore:
 - logisch: `mysql < full_dump.sql`
 - physisch: `mysqlbackup copy-back`

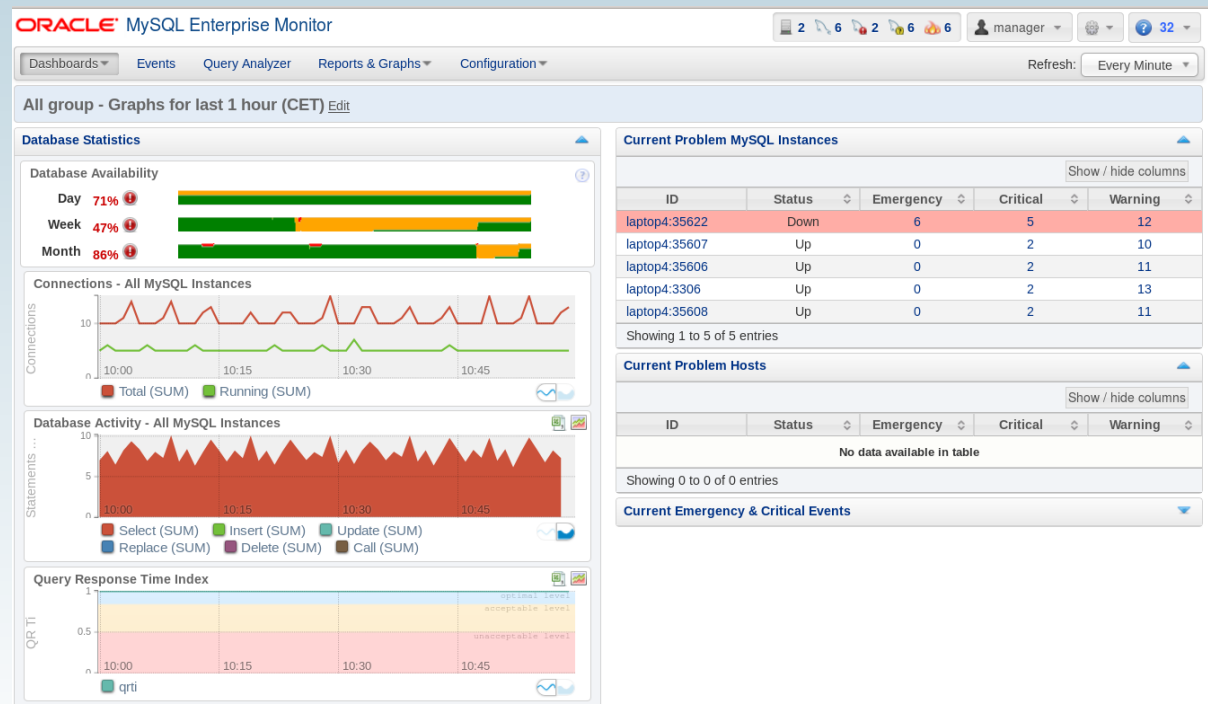


Point-in-Time-Recovery

- **Binary-Log für Point-in-Time-Recovery**
- **Binary-Log einschalten**
 - `log_bin = binary-log`
 - erfordert DB Neustart
 - ~ Oracle Archive Log (aber nur ähnlich!)
- **Bei Backup: Binary-Log und Position merken**
- **PiTR:**
 - `mysqlbinlog --start-position=45678
--stop-datetime=... binary-log.000042 ...`

Monitoring von MySQL

- MySQL Enterprise Monitor (MEM)
- Oracle Cloud Control (OEM/OMS) mit MySQL Plug-in
- 3rd Party Tools



Logging

- **Error Log**
 - Fehler und Warnungen (`log_warnings = 2`)
- **Slow Query Log**
 - „langsame“ Abfragen
 - `slow_query_log = ON`
 - `long_query_time = 1.25`
- **General Query Log**
 - „alle“ Abfragen
 - `general_query_log = ON`

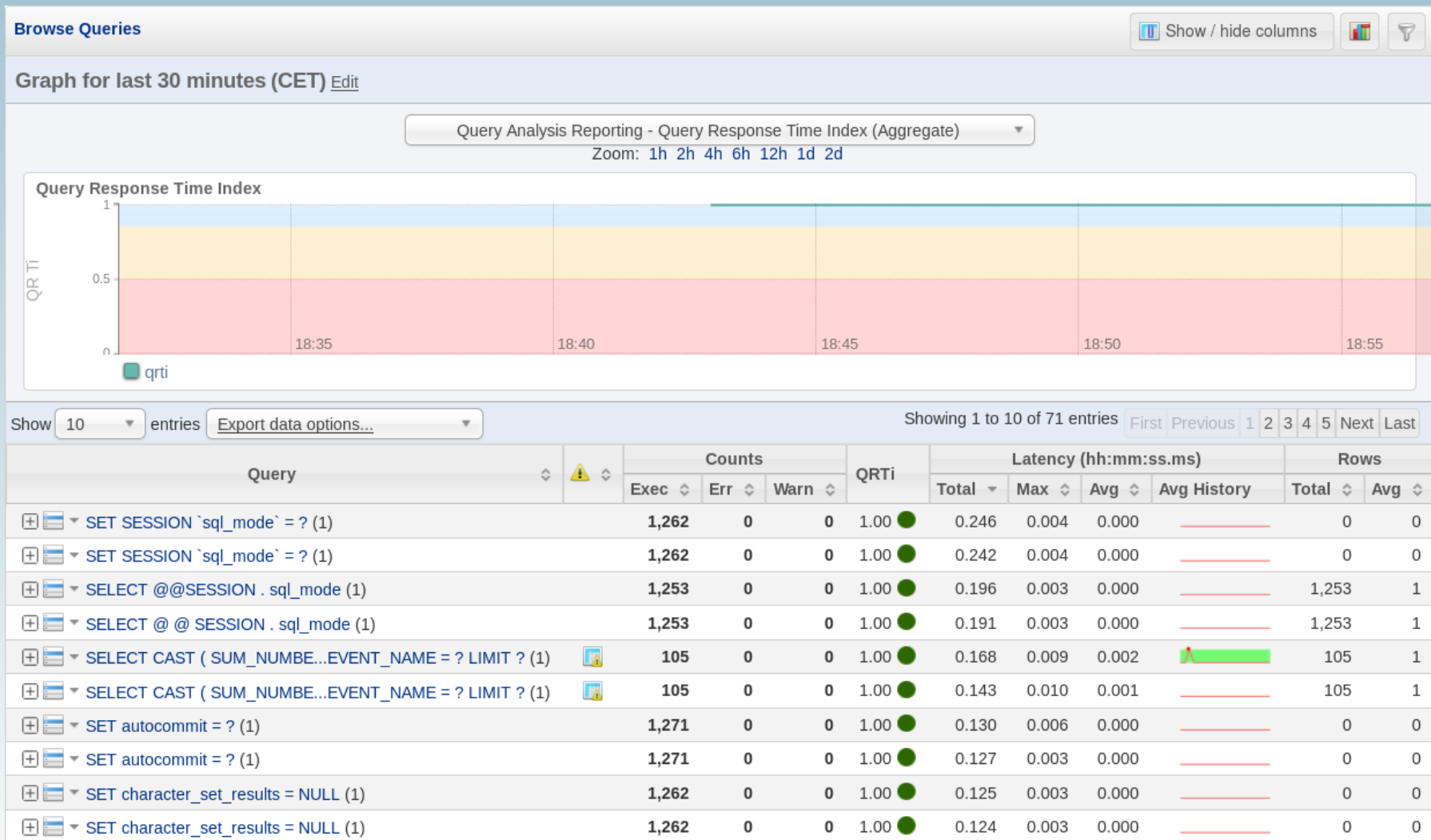
MySQL Upgrade

- **2 Varianten:**
 - **Dump/Restore (5 TB DWH?)**
 - **Binary-Upgrade**
- **Vorgehen (ca. 15 min):**
 - **Vorgängig testen und Backup!**
 - **DB stoppen**
 - **alte Pakete deinstallieren**
 - **neue Pakete installieren**
 - **DB starten**
 - **mysql_upgrade**

MySQL Tuning

- **Slow Query Log**
 - alle „langsamen“ Abfragen
- **General Query Log**
 - ALLE Abfragen
- **Query Analyzer (MEM)**
- **EXPLAIN** → **Query Execution Plan (QEP)**

Graphisch: Query Analyzer



Security

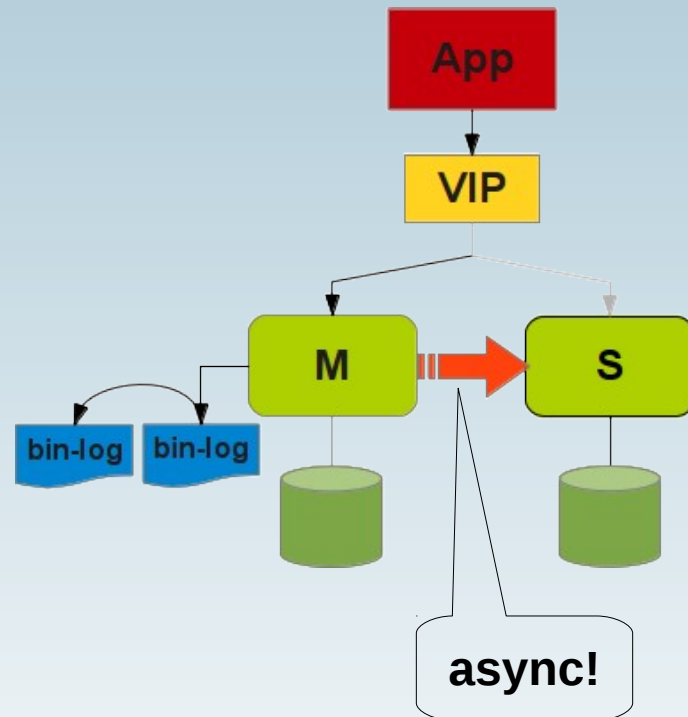
- Oft mit O/S User root (Devops-Konzept), geht auch ohne!
- User: `user@host`
- `localhost` (Socket) \neq `127.0.0.1` (TCP/IP)
- Keine Rollen!
 - Named User in DB „altes“ Konzept!
 - Heute: Applikations-User, Rollen in Applikation
- Härten von MySQL nach der Installation:
`mysql_secure_installation`
- MySQL 5.7 „secure by default“
- Enterprise-Features
 - MySQL Enterprise Audit Plugin (Oracle Audit Vault)
 - PAM-Authentication Plugin (OpenLDAP, AD)
 - MySQL Enterprise Encryption



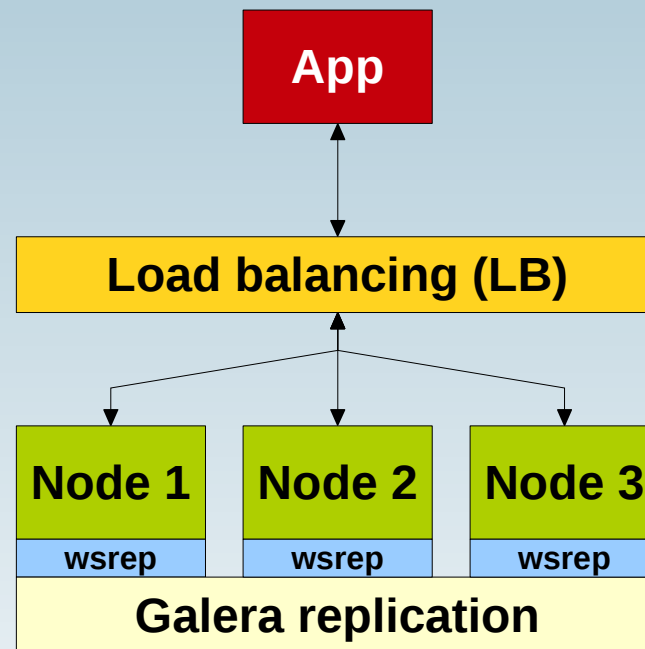
Hochverfügbarkeit

- **Master/Slave Replikation**
 - Data-Guard
- **aktiv/passiv Failover-Cluster**
 - dito
- **Galera Cluster für MySQL**
 - Oracle RAC
- **MySQL Fabric**
 - Skalierung/Sharding
- **NDB-Cluster**

MySQL Replication

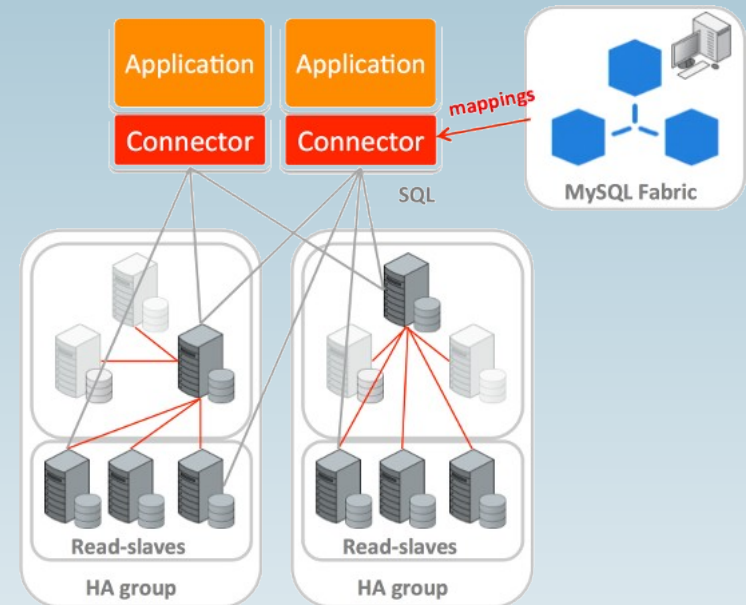


Galera Cluster



MySQL Fabric

- Sharding und Availability, GA
- Seit MySQL 5.6.10 (GTID)
- MySQL Utilities ($\geq 1.4.4$)
- Fabric aware Connector
 - Connector/J
 - Python
 - PHP (`mysqlnd_ms`)
- Applikation muss mit Architektur umgehen können!
- Noch nicht ganz perfekt...



Wir suchen noch:



**MySQL Datenbank Enthusiast/in für
Support / remote-DBA / Beratung**

Q & A



Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig:**
 - **Beratung**
 - **Remote-DBA**
 - **Support für MySQL, Galera, Percona Server und MariaDB**
 - **Schulung**

www.fromdual.com/presentations