



# MySQL für Oracle DBA's

**DOAG Regionaltreffen Freiburg/Südbaden**

**26. Januar 2016, Freiburg**

**Oli Sennhauser**

Senior MySQL Consultant, FromDual GmbH

**[oli.sennhauser@fromdual.com](mailto:oli.sennhauser@fromdual.com)**



# Über FromDual GmbH

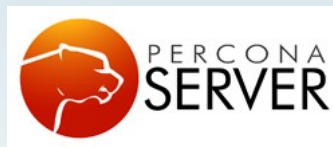
## Support



## Beratung



## remote-DBA



## Schulung



# Inhalt

## MySQL für Oracle DBA's

- **Geschichte**
- **Branches und Forks**
- **Installation**
- **Zugriff auf MySQL**
- **User und Schema**
- **Storage Engines**
- **InnoDB Storage Engine**
- **Performance Tuning**
- **Monitoring**
- **Logging**
- **Hochverfügbarkeit**
- **MySQL in der Cloud**

# 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 7.4 Mia, 2009**
  - MySQL und InnoDB jetzt wieder vereint!
- **Kontinuität: 2010 – 2016 MySQL 5.5, 5.6 und 5.7**
  - Verbesserung der Skalierbarkeit
  - Robustere Replikation
  - Ausmerzen von Kinderkrankheiten

# Wo steht MySQL heute?

292 systems in ranking, January 2016

Rank			DBMS	Database Model	Score		
Jan 2016	Dec 2015	Jan 2015			Jan 2016	Dec 2015	Jan 2015
1.	1.	1.	Oracle	Relational DBMS	1496.08	-1.47	+56.92
2.	2.	2.	MySQL	Relational DBMS	1299.26	+0.72	+21.75
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1144.06	+20.90	-54.55
4.	4.	↑ 5.	MongoDB +	Document store	306.03	+4.64	+55.13
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	282.40	+2.31	+27.91
6.	6.	6.	DB2	Relational DBMS	196.37	+0.24	-3.76
7.	7.	7.	Microsoft Access	Relational DBMS	134.04	-6.17	-5.10
8.	8.	8.	Cassandra +	Wide column store	130.95	+0.11	+32.20
9.	9.	9.	SQLite	Relational DBMS	103.74	+2.89	+7.54
10.	10.	10.	Redis +	Key-value store	101.16	+0.62	+6.92

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

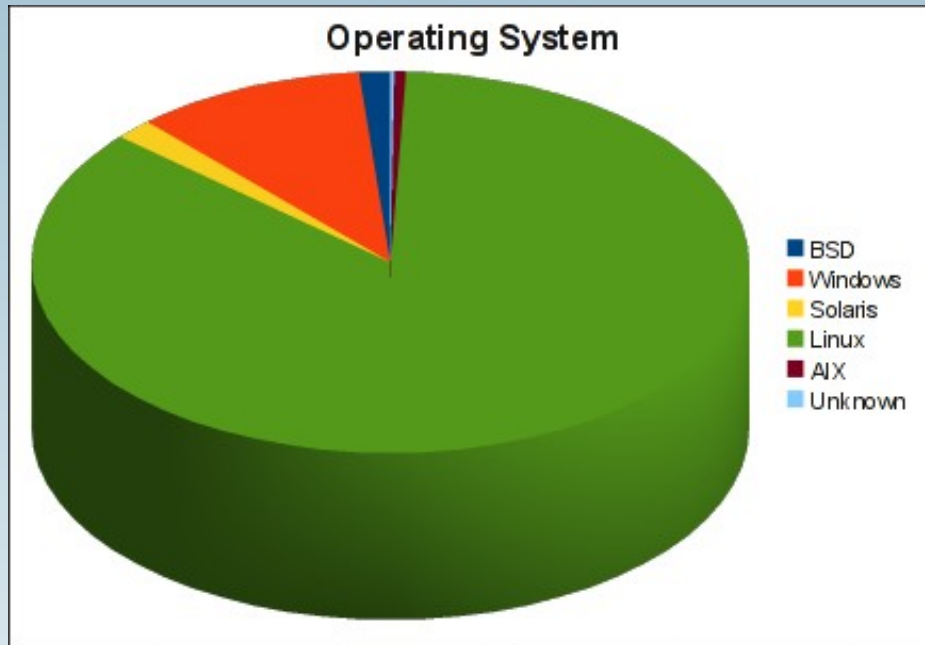
# Branches und Forks

- **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 wegen „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)**

# MySQL Plattform



- **85.7% Linux**
- **10.5% Windows**
- **1.7% Solaris**
- **1.4% BSD**
- **0.7% Others**

- **„Exotische“ Plattformen führen wahrscheinlich eher zu Problemen!**



# Installation

- **Wichtiger Grundsatz bei MySQL: KISS!**
  - Warum einfach machen, wenn es auch kompliziert geht?
- **Oracle: OUI (MySQL zum Glück nicht!)**
- **MySQL:**
  - Pakete der Hersteller (RPM, DEB)
  - Pakete der Distribution (RPM, DEB)
  - Binary-Tar-Ball (.tar.gz)
  - Source zum selber kompilieren
- **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=...`
  - **Neu in 5.7:**
  - `mysqld --initialize --user=mysql --datadir=...`
- **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)!**
- **Achtung bei Distrio-Start/Stop-Skripten (Debian)!**

# 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)

# Connections / Connectors

- **Verbindung**
  - In MySQL billig: oft KEIN Connection-Pooling
  - 1 Verbindung = 1 Thread → 1 Query → 1 Core
  - Thread Pool (1000e von Verbindungen)
- **Connectors:**
  - JDBC/ODBC
  - PHP, Perl, Python, Ruby, .NET

# User und Schema

- **User**
  - `'oli'@'localhost'` → **Unix Socket**
  - `'oli'@'127.0.0.1'` → **TCP von IP**
  - `'oli'@'%'` → **TCP von überall her**
- **MySQL Port: 3306**
- **Privilegien**
  - **Global: \*.\***, pro Schema , pro Tabelle, pro Spalte
- **Schema (= Database)**
  - **Objekte unabhängig vom User (→ gehört System)**

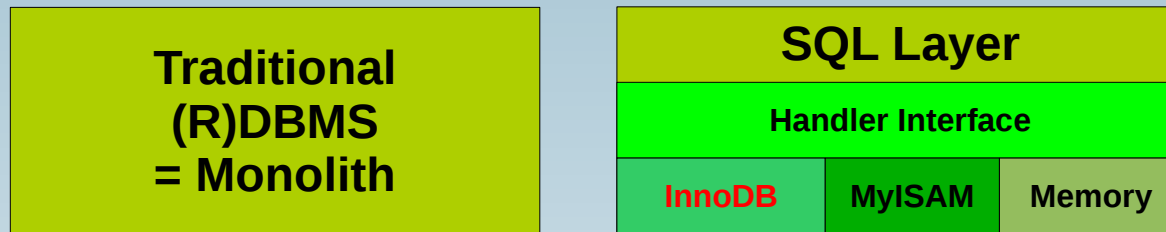


# Wichtigste Befehle

- **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 ( $\leq 5.1$ )
- heute: InnoDB ( $\geq 5.5$ )

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

# InnoDB Storage Engine

- Transaktionen (ACID)
- Isolation Level (default repeatable-read)
- InnoDB Buffer Pool → Database Buffer Cache
  - `innodb_buffer_pool_size`
  - 16k Pages
- 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)
- InnoDB: PK geclusterte Tabellen → IOT

# Performance Tuning

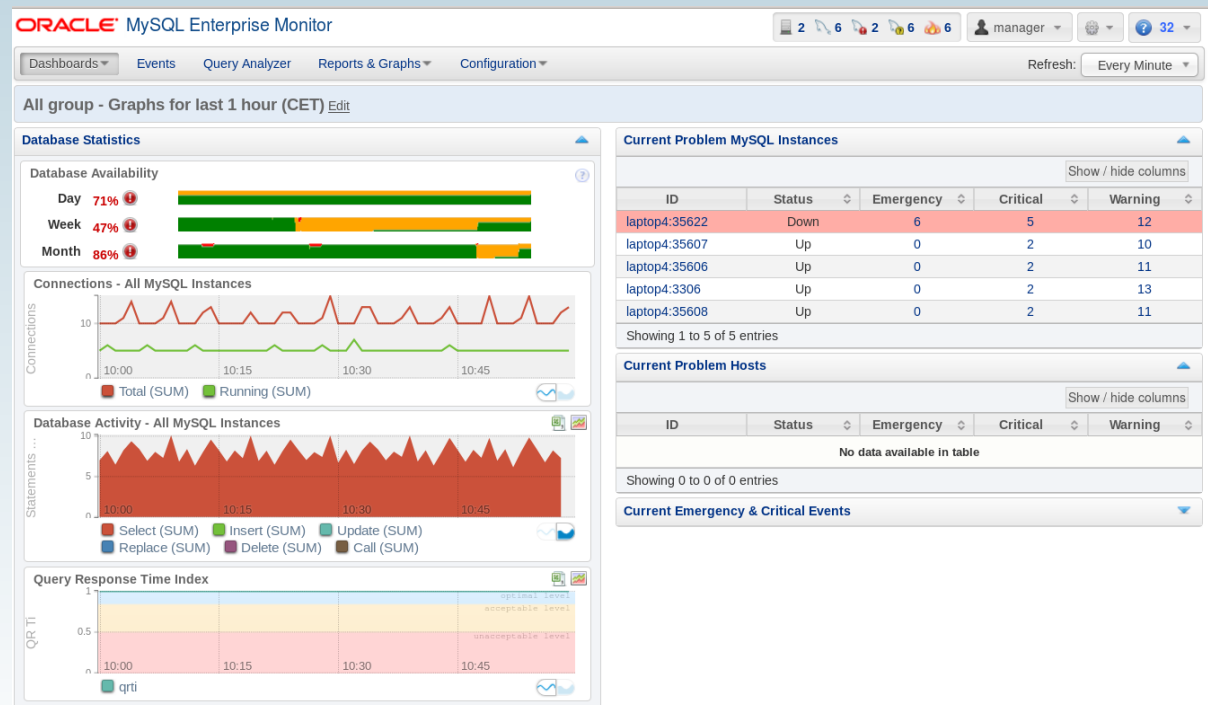
- `mysql> SHOW GLOBAL STATUS;`
- `PERFORMANCE_SCHEMA`
- **Slow Query Log**
  - `slow_query_log = 1`
  - `long_query_time = 0.5`
  - `shell> mysqldumpslow -s t slow.log > profile`
- **Query Execution Plan:**
  - `mysql> EXPLAIN SELECT * FROM test;`

# 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!) :-)

# Monitoring von MySQL

- MySQL Enterprise Monitor (MEM)
- Oracle Cloud Control (OEM/OMS) mit MySQL Plug-in
- 3<sup>rd</sup> 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`

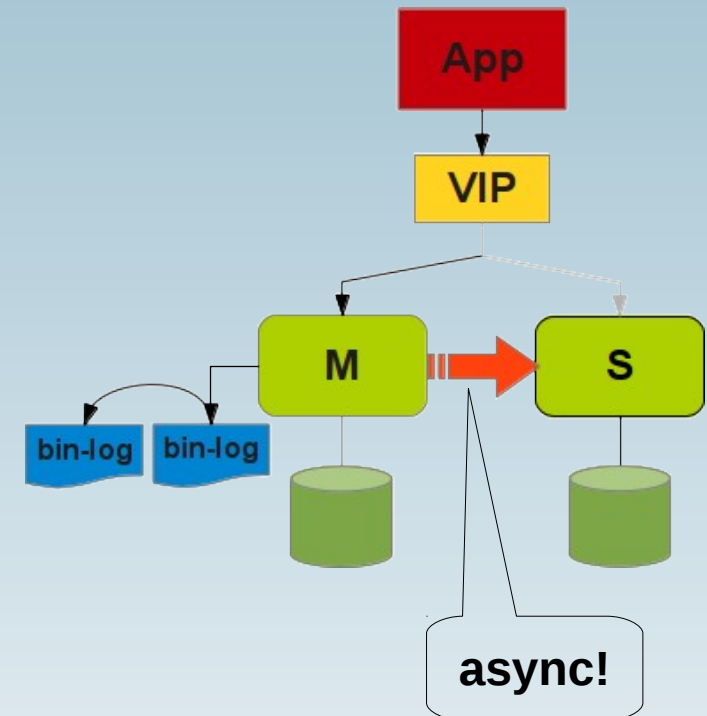
# Hochverfügbarkeit

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



# MySQL Replikation

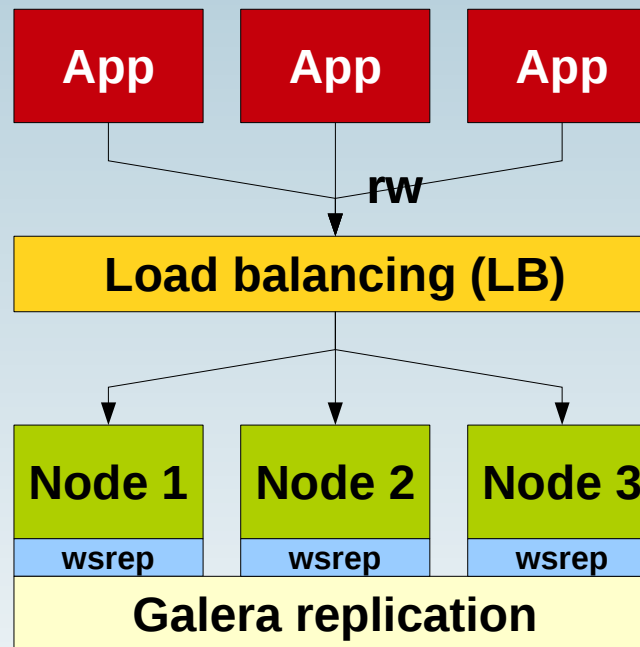
- MySQL Master/Slave Replikation
  - Entspricht Oracle Data Guard
- Einsatz-Szenarien
  - Hochverfügbarkeits-Anforderungen mittel bis hoch
  - Failover typischerweise < 1 min  
→ Failover-Slave
  - Reporting-Slave
  - Backup-Slave
  - Massives Read-Scaleout
- Set-up ziemlich einfach
  - Know-How ist weit verbreitet
  - Man muss trotzdem wissen, was man tut!
  - One-shot failover
  - Google weiss Rat!



# RAC: Galera Cluster

## Oracle Real Application Cluster (RAC)

- MySQL: Galera Cluster



- Shared-Nothing Architektur

# MySQL in der Cloud

- **Ja geht!**
- **Sowohl in:**
  - **Private Cloud**
  - **Public Cloud**
- **Rechenzentrums-Übergreifend**
- **Kontinent-Übergreifend**
- **Aber auf Kosten von:**
  - **Performance (Latenz, Noisy Neighbours)**
  - **Preis (dafür Flexibilität)**
  - **Vertraulichkeit**

# Q & A



[www.fromdual.com](http://www.fromdual.com)



**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](http://www.fromdual.com/presentations)**