



MySQL 5.7

Neue Funktionen

DOAG 2016, Webinar

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com



www.fromdual.com

About FromDual GmbH

Support



Beratung



remote-DBA

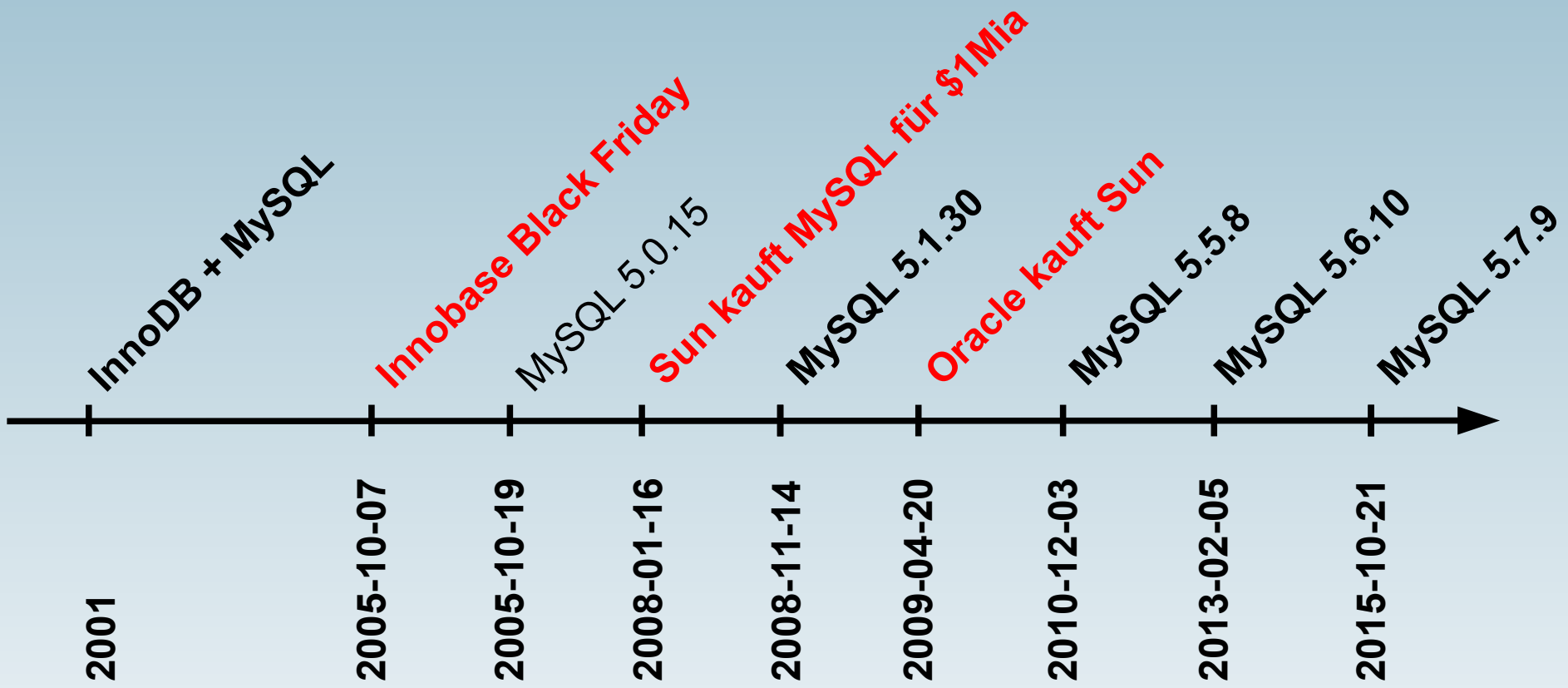


Schulung





MySQL (Release) Geschichte



Welchen Release?

- MySQL: ~~5.0~~, ~~5.1~~, ~~5.5~~, 5.6, 5.7, 8.0
- Heute (2016) sollten wir auf MySQL 5.6 sein und über Upgrade auf 5.7 nachdenken!!!
- Neue Projekte mit MySQL 5.7!

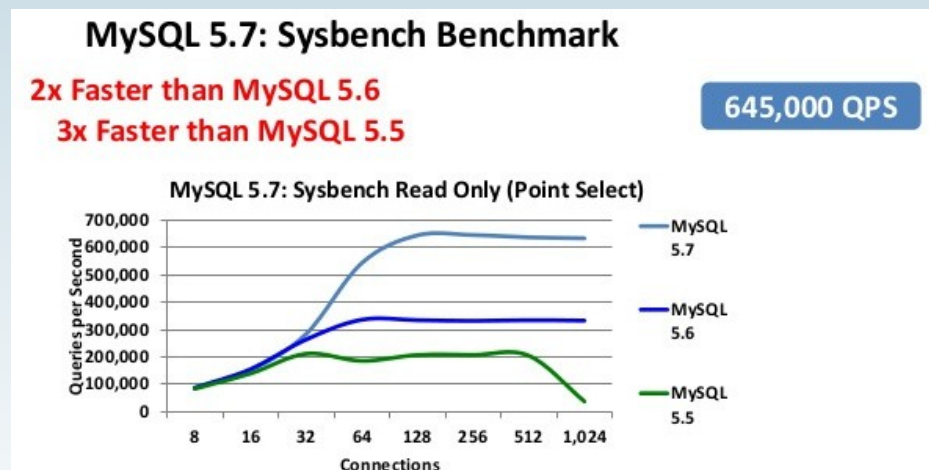
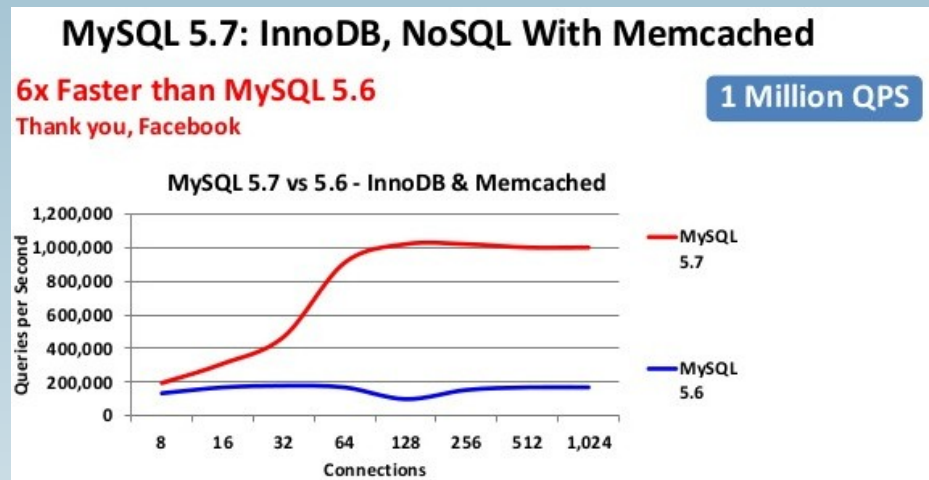
ORACLE LIFETIME SUPPORT FOR MYSQL RELEASES

RELEASE	GA DATE	SUPPORT END	EXT. SUPPORT END
MySQL Database 5.0	Oct 2005	Dec 2011	n.a.
MySQL Database 5.1	Dec 2008	Dec 2013	n.a.
MySQL Database 5.5	Dec 2010	Dec 2015	Dec 2018
MySQL Database 5.6	Feb 2013	Feb 2018	Feb 2021
MySQL Database 5.7	Oct 2015	Oct 2020	Oct 2023
MySQL Database 8.0	expected 2017	unknown	unknown

Warum MySQL 5.7?

- **Skalierbarkeit und Performance**
- **Security-Verbesserungen**
- **Neue Funktionalität**
 - **JSON-Unterstützung**
 - **GIS Funktionalität (geographische Informationen)**
 - **Generierte Spalten (generated columns)**
 - **Optimizer-Verbesserungen**
 - **Multi-source Replikation**
- **Betriebliche Aspekte**
- **Altlasten aufräumen**

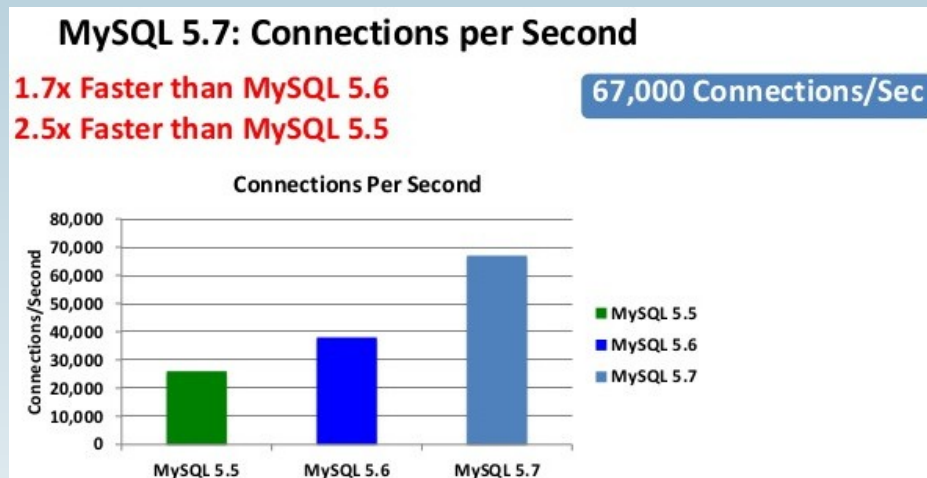
Skalierbarkeit



- Wer hat 16 – 32 oder mehr parallele Abfragen?
- 100 – 200k Qps?
- Wer verwendet InnoDB mit Memcached?
- Wer macht „Point Select“ Abfragen?
- Wer hier ist von FB oder Ähnlichen?
- Einzel-Abfrage Performanz (Latenz) wird schlechter!!! :-)

Performance

- TCP Timeout 120s?
 - 400 – 800 Verb/s
 - 32'000 Ports
- SSL neu Standard!
 - Einfluss auf Verbindungsrate?



Security Verbesserungen

- **MySQL ist neu sicher „by default“!**
 - **Wir alle wissen: Das bedeutet weniger KISS!**
- **SSL Standard (langsamer und teurer)**
- **Tablespace Verschlüsselung (sinnvoll?)**
- **Password Ablauf Regeln**
- **Password Stärken Regeln**
 - **Gelöschte Spalte in `mysql.user` Tabelle :-(
• **Alte Admin-Applikationen funktionieren ev. nicht mehr?****
- **`sql_mode` ist strikter als früher (gut!)**

InnoDB Tablespace Verschlüsselung

```
SELECT plugin_name, plugin_status FROM information_schema.plugins
WHERE plugin_name='keyring_file';
```

```
+-----+-----+
| PLUGIN_NAME | PLUGIN_STATUS |
+-----+-----+
| keyring_file | ACTIVE        |
+-----+-----+
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_file_per_table | ON |
+-----+-----+
```

```
SHOW GLOBAL VARIABLES LIKE 'keyring%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| keyring_file_data | /usr/local/mysql/keyring/keyring |
+-----+-----+
```

```
ERROR 3185 (HY000): Can't find master key from keyring,
please check keyring plugin is loaded.
```

InnoDB Tablespace Verschlüsselung

```
strings test.ibd
infimum
supremum
secret dataV
secret dataV
secret dataV
secret dataV
```

```
ALTER TABLE test ENCRYPTION = 'y';
```

```
strings test.ibd
VBw1?"sh
Pk- {
`dIO
...
```

Generierte Spalten

```
CREATE TABLE product (
  id          INT UNSIGNED NOT NULL AUTO_INCREMENT
, product_id VARCHAR(40)
, product_id_norm INT AS (SUBSTR(product_id,3)) VIRTUAL
, PRIMARY KEY (id)
, INDEX (product_id_norm)
);
```

```
SELECT * FROM product WHERE product_id_norm = 1;
```

```
+----+-----+-----+
| id | product_id | product_id_norm |
+----+-----+-----+
|  1 | 550000001 |          1      |
+----+-----+-----+
```

```
+-----+-----+-----+-----+-----+
| table  | type  | possible_keys | key          | ref  |
+-----+-----+-----+-----+-----+
| product | ref  | product_id_norm | product_id_norm | const |
+-----+-----+-----+-----+-----+
```

JSON Unterstützung

- JSON Datentyp
 - Automatische Validierung von JSON Dokumenten
 - Optimiertes Speicherformat für schnellen Lesezugriff
 - Grösse bis `max_allowed_packet`

```
CREATE TABLE t1 (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY
, jdoc JSON);

INSERT INTO t1 VALUES(NULL, '{"key1": "value1", "key2": "value2"}');
```

```
SELECT * FROM t1;
```

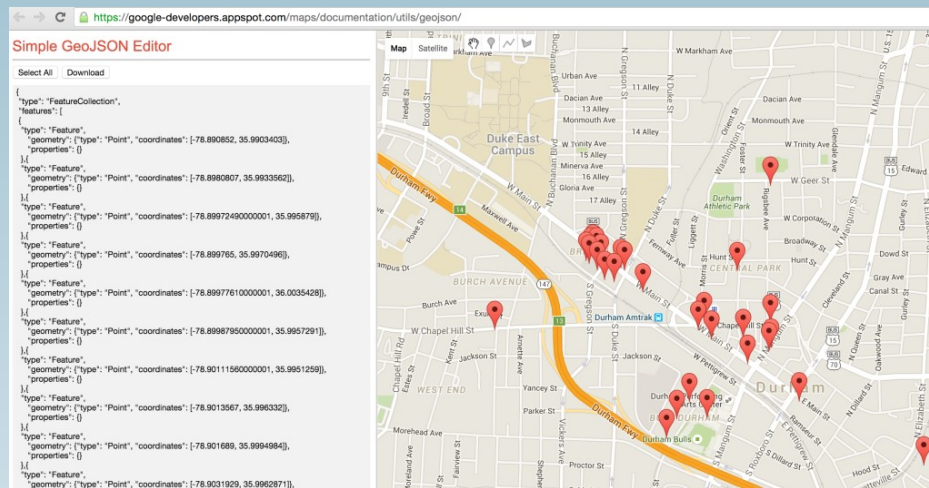
id	jdoc
1	<code>{"key1": "value1", "key2": "value2"}</code>

JSON Funktionen

- **JSON_xxx**
 - APPEND, ARRAY, ARRAY_APPEND, ARRAY_INSERT, CONTAINS, DEPTH, EXTRACT, INSERT, KEYS, LENGTH, MERGE, OBJECT, QUOTE, REMOVE, REPLACE, SEARCH, SET, TYPE, UNQUOTE, VALID
- Indexierung: Erfolgt über generierte Spalten

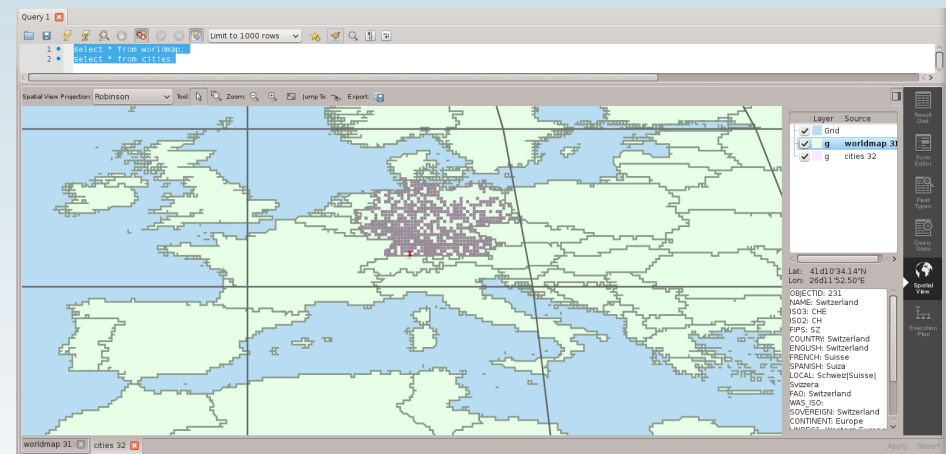
```
CREATE TABLE jsongen (  
  c JSON  
  , g INT GENERATED ALWAYS AS (JSON_EXTRACT(c, '$.id'))  
  , INDEX (g)  
);
```

GIS (spatial) Indices



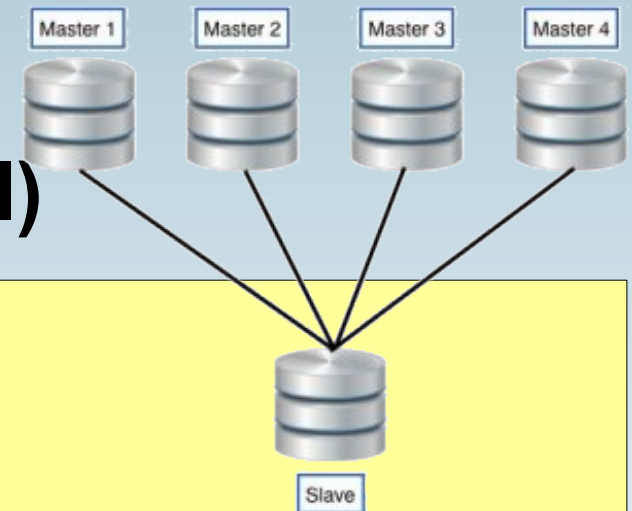
```
CREATE TABLE gis (
  id INT UNSIGNED NOT NULL
  , g GEOMETRY NOT NULL
  , PRIMARY KEY (id)
  , SPATIAL INDEX(g)
) ENGINE = InnoDB;
```

- Spatial Viewer in der MySQL Workbench
- Kunden in Deutschland in weniger als 1 Stunde!
- In Kombination mit JSON



Multi-source Replikation

- Fan-out Replikation (können wir schon lange!)
- Fan-in Replikation: viele Master eine Slave
 - Für Sammeln/Aggregieren
 - Replikations-Kanal (Channel)



```
master_info_repository      = TABLE
relay_log_info_repository  = TABLE
```

```
CHANGE MASTER
```

```
TO MASTER_HOST='master1', MASTER_USER='replication'
, MASTER_PASSWORD='secret'
, MASTER_LOG_FILE='master1-bin.000042', MASTER_LOG_POS=466918
FOR CHANNEL 'master1';
```

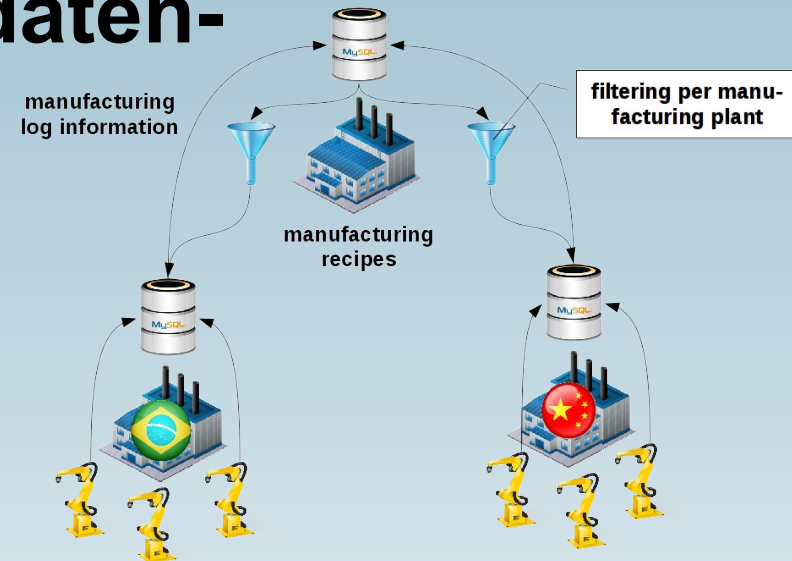
```
START SLAVE FOR CHANNEL 'master1';
```

Multi-Source Replikation Anwendungsfälle

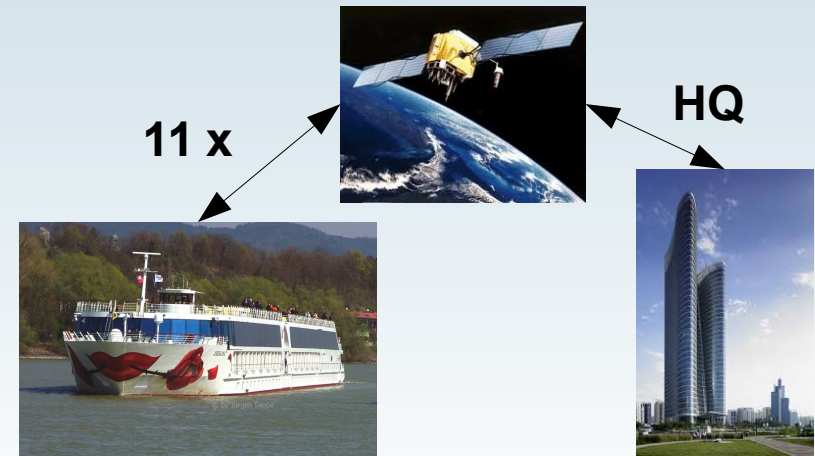


www.fromdual.com

- **Weltweite Produktionsdaten-Verteilung**



- **Flotten-Management**



Flexible (allg.) Tablespaces

- **Hoster/SaaS: multi-mandantenfähige Anwendungen**
 - **10'000 Kunden mit 200 Tabellen/Schema**
 - **`innodb_file_per_table = 0` oder `1` ???**
 - **2 Mio Tabellen!!!**
- **File Handles**
 - **Linux ist nicht glücklich**
 - **MySQL ist auch nicht glücklich**

Flexible (allg.) Tablespaces

```

CREATE TABLESPACE customer0001
  ADD DATAFILE 'customer0001.ibd' ENGINE = InnoDB;

ALTER TABLE customer0001.invoices TABLESPACE = customer0001;

SELECT ts.space AS ts_id, ts.name AS ts_name, d.path
  , SUBSTRING_INDEX(t.name, '/', 1) AS t_schema
  , SUBSTRING_INDEX(t.name, '/', -1) AS t_name
FROM I_S.innodb_sys_tablespaces AS ts
JOIN I_S.innodb_sys_tables AS t ON t.space = ts.space
JOIN I_S.innodb_sys_datafiles AS d ON t.space = d.space
WHERE ts.space_type = 'General';

```

ts_id	ts_name	t_schema	t_name	path
30	my_ts	mysql	test2	./my_ts.ibd
186	customer0001	test	jsongen	./customer0001.ibd

Optimizer Verbesserungen

```
EXPLAIN FOR CONNECTION 42;
```

- **Neue Optimizer Hints:**

```
SELECT /*+ hint(<table> <index>) */ * FROM table;
```

- **NO_RANGE_OPTIMIZATION, NO_MRR, MAX_EXECUTION_TIME, NO_ICP**
- **Neues Kostenmodell:**
 - **mysql.engine_cost, mysql.server_cost**

sys Schema

- In MySQL 5.7 mit dabei (`mysql_upgrade`)
- Einfacher Zugriff auf `PERFORMANCE_SCHEMA`
- Abgedeckte Themen:
 - `host_*` → Aktivitäten gruppiert nach Host
 - `innodb_*` → InnoDB Informationen
 - `io_*` → I/O Konsumenten gruppiert nach File, Bytes, Latenz
 - `memory_*` → Speicherverbrauch gruppiert nach Host, Thread, User, Type
 - `schema_*` → Verschiedene Informationen über Schemata
 - `statement_*` → Statistiken über Abfragen
 - `user_*` → Information pro User
 - `waits_*` → Wait Event Informationen
- Use cases: <http://fromdual.com/mysql-performance-schema-hints>

Betriebliche Aspekte

- InnoDB Buffer Pool online ändern:
 - `SET GLOBAL innodb_buffer_pool_size = 16 * CAST(POW(1024, 3) AS SIGNED);`
- `mysql_install_db` ist obsolet:
 - `mysqld --initialize / --initialize-insecure`
- Implizites Kreieren eines Users ist nicht mehr zulässig:
 - `GRANT ALL ON *.* to 'nonexist'@'localhost';`
`ERROR 1133 (42000): Can't find any matching row in the user table`
 - → Zuerst: `CREATE USER ...` dann `GRANT ...`
- Upgrade einer Master/Slave Replikation von 5.5 direkt nach 5.7 ist NICHT möglich (Server UUID)!
- Fehlerausgabe
 - Syslog native Unterstützung
 - `SET GLOBAL log_verbosity = 1 - 3` (Fehler, Warnungen, Mitteilungen)
- `super_read_only = 1`
- `SET GLOBAL offline_mode=1`
- `disabled_storage_engines="MyISAM"`

Veraltet und entfernt

- **Veraltete Funktionalität**
 - `innodb_install_db` → `mysqld --initialize`
 - Alte Variablen wie `innodb_file_format`, `innodb_large_prefix`
 - `GRANT USER` um User anzulegen → `CREATE USER`
 - Implizites `GROUP BY` Sortieren → `ORDER BY`
 - `--log-warnings` → `log_error_verbosity`
 - etc.
- **Entfernte Funktionalität**
 - Alte Passwort-Hashes
 - `YEAR(2)` → Kovertieren in `YEAR(4)` (kann teuer sein!)
 - `storage_engine` → `default_storage_engine`
 - `thread_concurrency`
 - `--key-buffer` → `--key-buffer-size`
 - `innodb_additional_mem_pool_size`
 - `INSERT DELAYED`
 - InnoDB Monitor und InnoDB Lock Monitor Tabellen
 - MySQL Utilities: `mysqlhotcopy`, etc.
- **Upgrade sorgfältig testen!!!**

Q & A

```
mysql> SHUTDOWN;  
Query OK, 0 rows affected (0.00 sec)
```



Fragen ?

Diskussion?

Slides are on-line:

www.fromdual.com/presentations