MySQL Query Tuning

DOAG K & A 2017, Nürnberg

Oli Sennhauser

Senior MySQL Berater, FromDual GmbH

oli.sennhauser@fromdual.com

Über FromDual GmbH

www.fromdual.com

Beratung

Schulung

remote-DBA

Support

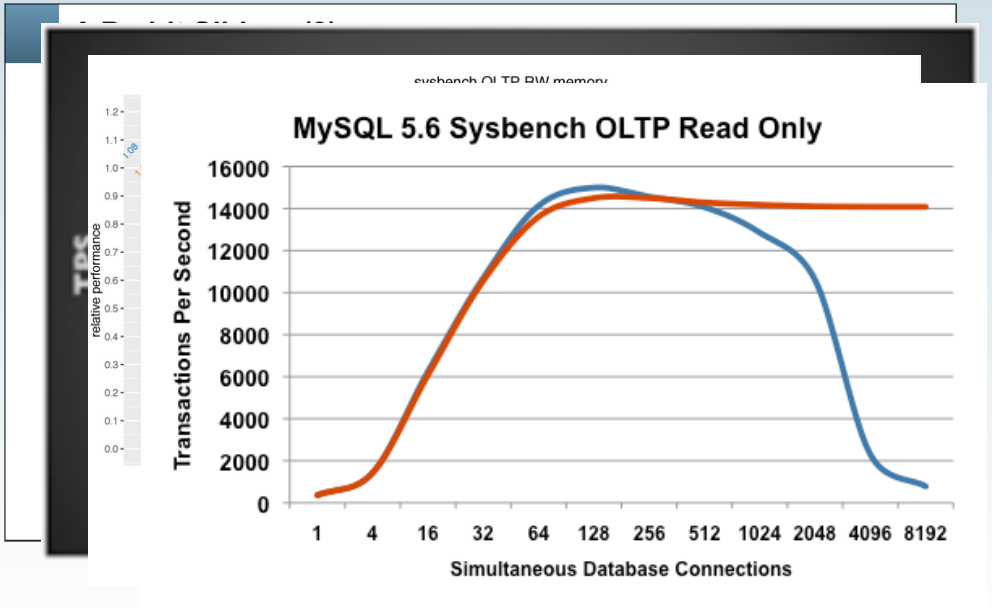
Inhalt

SQL Query Tuning

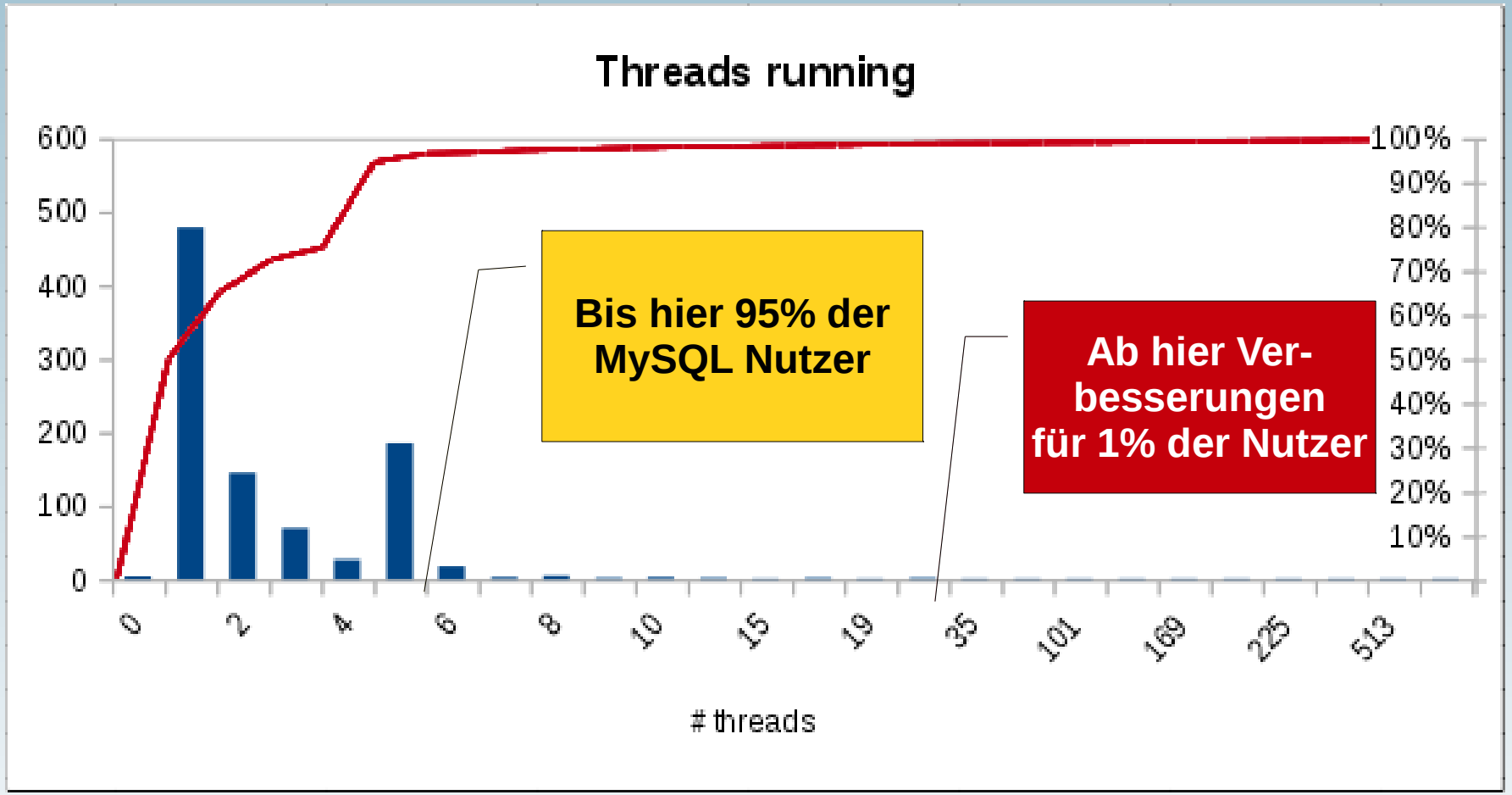
- Performance - Verständnis
- Slow Query Log
- Welche Queries zuerst?
- Query Execution Plan (QEP)
- Vorgehen
- Beispiele

Performance - Wasndas?

- 1 Wort verschiedene Vorstellungen:
  - Latenz (Zeit) vs. Durchsatz (Dingens/Zeit)
- Softwarehersteller → Durchsatz
  - Jeder neue MySQL Release: mehr Durchsatz
  - Aber tendenziell schlechtere Latenz! :-(



Typischer MySQL User



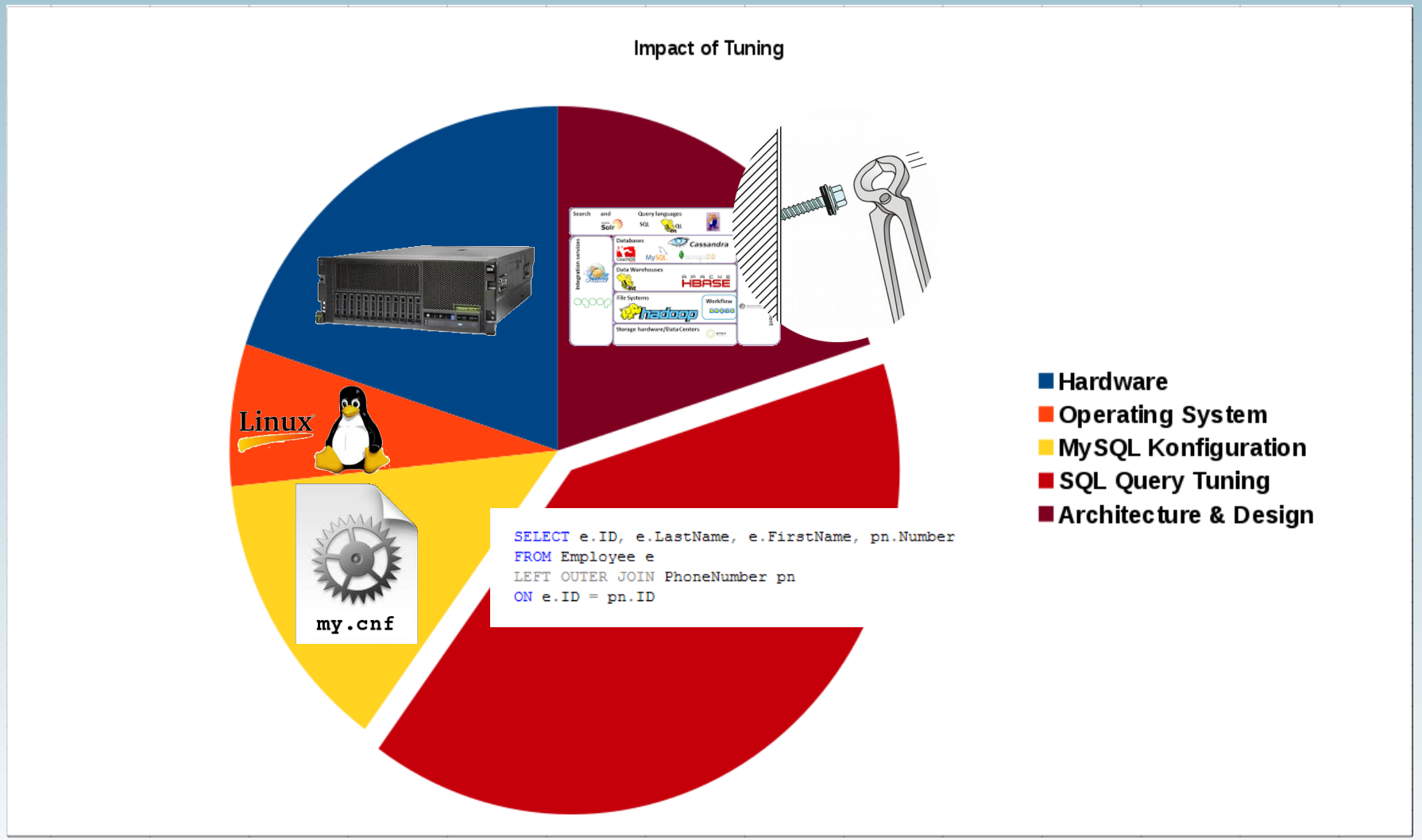
- **Wir sind NICHT: { Alibaba | Google | Xing | Twitter | LinkedIn | Facebook | Booking.com | … }!!!**

Nebel!

- **Will uns da jemand vom rechten Weg abbringen?**

- **Wir haben typischerweise ein Latenzproblem und KEIN Durchsatzproblem!!!**

Performance - Angriffsvektoren



Query Performance Tuning

- Königsdisziplin
  - Bis 100 fache schnellere Antwortzeit (Latenz)!
- Oft „Low hanging fruits“
- Primär Latenz-Tuning
  - Sekundär auch mehr Durchsatz

Slow Query Log I


```
#
# my.cnf
#

[mysqld]

slow_query_log                  = OFF
log_output                      = FILE
slow_query_log_file             = /var/log/mysqld/slow.log
log_timestamps                  = UTC

long_query_time                 = 0.35   # default 10.0s

log_slow_admin_statements       = OFF
log_slow_slave_statements       = OFF
log_queries_not_using_indexes   = OFF

min_examined_row_limit                  = 0
log_throttle_queries_not_using_indexes = 0
```

```
SET GLOBAL slow_query_log = ON;
```

Slow Query Log II

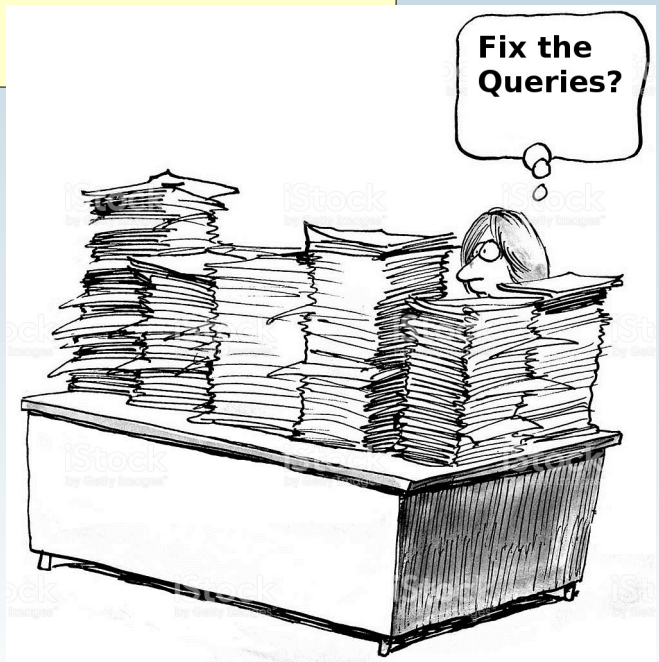

```
/home/mysql/product/mysql-5.7/bin/mysqld, Version: 5.7.20 (MySQL Community Server (GPL)).
Tcp port: 3306  Unix socket: /var/run/mysqld/mysql.sock
Time                 Id Command    Argument
# Time: 160901  9:40:27
# User@Host: root[root] @ localhost [127.0.0.1]
# Query_time: 0.816336  Lock_time: 0.000318 Rows_sent: 22  Rows_examined: 485379
use test;
SET timestamp=1472737227;
SELECT a.*, ig.descr as cb_ip_groups_descr, ig.ip as cb_ip_groups_ip, ng.descr ...
# User@Host: test_db_user[test_db_user] @  [10.30.4.153]
# Query_time: 0.622558  Lock_time: 0.000234 Rows_sent: 1  Rows_examined: 1977
SET timestamp=1390381200;
SELECT object_2.o_id AS o_id,o_type FROM `object_2` WHERE ( atr_lang = 'de' and ...
# User@Host: test_db_user[test_db_user] @  [10.30.4.153]
# Query_time: 0.377936  Lock_time: 0.000260 Rows_sent: 1  Rows_examined: 2814
SET timestamp=1390381200;
SELECT object_2.o_id AS o_id,o_type FROM `object_2` WHERE ( atr_lang = 'de' and ...
# User@Host: test2[test2] @  [10.30.4.153]
# Query_time: 0.015225  Lock_time: 0.000039 Rows_sent: 1  Rows_examined: 1
use test2;
SET timestamp=1390381200;
SELECT accommodationId, defOrder FROM `accommodationSearchTables`.`_9a6dfffebb5c51` LIMIT 1;
# User@Host: test2[test2] @  [10.30.4.153]
# Query_time: 0.928405  Lock_time: 0.000041 Rows_sent: 9  Rows_examined: 9
SET timestamp=1390381200;
SELECT kurztext FROM haus_suche_de hs WHERE hs.hausnr = 361220;
# User@Host: test2[test2] @  [10.30.4.153]
# Query_time: 1.385471  Lock_time: 0.000080 Rows_sent: 0  Rows_examined: 28356
SET timestamp=1390381200;
SELECT t1.tablename from `accommodationSearch`.`tables` t1 LEFT JOIN `accommodati...
```

MySQLdumpSlow I

- 10 Gb Slow Query Log → Entwickler: fixen!
- Profile des Slow Query Logs:

```
shell> mysqldumpslow --s t slow.log > slow.log.profile

shell> mysqldumpslow --help !!!
```

- In Perl (Windows: :-( )



MySQLdumpSlow II


```
Count: 19752  Time=5.20s (102797s)  Lock=0.00s (5s)  Rows=0.2 (3577), root[root]@localhost
  SELECT cdr.* ...
  FROM cdr as cdr  force index (calldate)
  STRAIGHT_JOIN cdr_next as cdr_next on (cdr_next.cdr_ID = cdr.ID)
  WHERE (cdr.calldate>'S' and cdr.id>N and cdr.id<=N) AND ((( (cdr.mos_min_mult10 / N) ...

Count: 63206  Time=1.02s (64696s)  Lock=0.00s (24s)  Rows=22.0 (1390532), root[root]@localhost
  SELECT a.* ...
  from alerts_sended as1 ...
  WHERE (a.disable is null or not a.disable) and
  (TRIM(coalesce(a.email,'S'))<>'S' ...

Count: 6572  Time=5.17s (33955s)  Lock=0.00s (1s)  Rows=0.5 (3184), root[root]@localhost
  SELECT cdr.*, ...
  FROM cdr as cdr  force index (calldate)
  STRAIGHT_JOIN cdr_next as cdr_next on (cdr_next.cdr_ID = cdr.ID)
  WHERE (cdr.calldate>'S' and cdr.id>N and cdr.id<=N) AND ((( (cdr.mos_min_mult10 / N) ...

Count: 2193  Time=5.30s (11615s)  Lock=0.00s (0s)  Rows=21.4 (46911), root[root]@localhost
  SELECT cdr.*, ...

Count: 1026  Time=3.69s (3789s)  Lock=0.00s (0s)  Rows=1644.2 (1686978), root[root]@localhost
  select sipcallerip,connect_duration as connect_duration,delay_sum as delay_sum, ...
```

Performance Schema

- Query dazu:
  - https://fromdual.com/mysql-performance-schema-hints#top-long-running-queries

```
+----------------------------------+---------+---------+-------------+-------------+-----+
| digest_text                      | exec_ms | lock_ms | first_seen  | last_seen   | cnt |
+----------------------------------+---------+---------+-------------+-------------+-----+
| INSERT INTO `test` SELECT ? , ... | 50493.5 |    26.3 | 12 16:41:35 | 12 16:42:04 |  20 |
| SELECT LEFT ( `digest_text` , ... | 14434.6 |    25.8 | 12 16:48:44 | 12 17:07:15 |   6 |
| SELECT * FROM `test`             |  7483.0 |     0.2 | 12 16:41:16 | 12 16:42:34 |   2 |
| SHOW ENGINE INNODB STATUS        |  1912.4 |     0.0 | 12 16:37:19 | 12 17:07:36 | 687 |
| SHOW GLOBAL VARIABLES            |  1091.1 |    68.8 | 12 16:37:19 | 12 17:07:36 | 687 |
| SHOW GLOBAL STATUS               |   638.7 |    40.8 | 12 16:37:19 | 12 17:07:36 | 687 |
| SELECT LEFT ( `digest_text` , ... |   356.2 |    42.4 | 12 16:42:38 | 12 16:45:00 |   6 |
| SELECT `digest_text` , SUM ( ... |   325.3 |     0.4 | 12 16:40:44 | 12 16:42:18 |   3 |
| SELECT `DIGEST_TEXT` , ( `TIME... |   163.2 |     1.0 | 12 16:37:44 | 12 16:39:22 |   9 |
| SELECT LOWER ( REPLACE ( trx_s... |   133.9 |    80.2 | 12 16:37:19 | 12 17:07:36 | 687 |
+----------------------------------+---------+---------+-------------+-------------+-----+
```

Query und was nun?

- Query Execution Plan (QEP)
  - Zeigt uns, was MySQL zu tun gedenkt:
- Operationen, Reihenfolge und Kosten
- 3 verschiedene Ausgabeformate
  - tabellarisch
  - JSON Format
  - graphisch

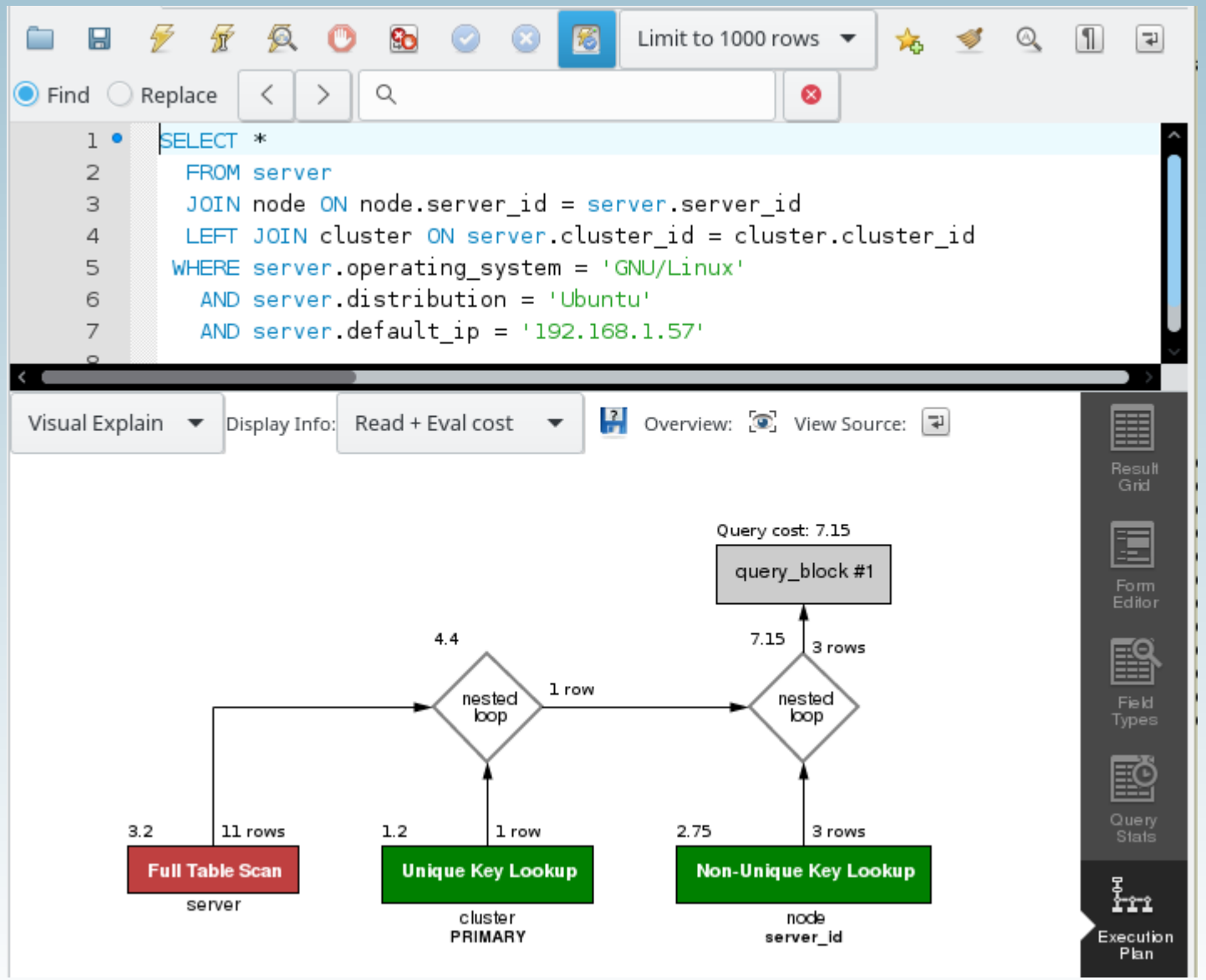
Tabellarischer QEP


```
EXPLAIN
SELECT o.brochure_code, o.service_code, c.category_code, os.traveldate, ...
  FROM object AS o
  JOIN object_category AS c ON o.object_key = c.object_key
  JOIN object_searchtable_de_DE AS os ON c.object_category_id = os.object_category_id
 WHERE o.brochure_code IN ( 'FHD11', 'FHF11', 'FHSK11', 'FHSUED11' )
   AND WEEKDAY(os.traveldate) = 5
   AND o.suppliernumber NOT IN (1000173,1000535,1004289, ...)
 ORDER BY o.brochure_code, o.service_code, c.category_code, os.traveldate;

+-------------+-------+--------+-----------------------------+----------+---------+...
| select_type | table | type   | possible_keys               | key      | key_len |...
+-------------+-------+--------+-----------------------------+----------+---------+...
| SIMPLE      | o     | ALL    | PRIMARY,suppliernumber, ... | NULL     | NULL    |...
| SIMPLE      | c     | ref    | PRIMARY,FKIndex1            | FKIndex1 | 66      |...
| SIMPLE      | os    | eq_ref | PRIMARY,is_valid            | PRIMARY  | 3       |...
+-------------+-------+--------+-----------------------------+----------+---------+...
            ...+---------------------+------+-------------------------------------------+
            ...| ref                 | rows | Extra                                     |
            ...+---------------------+------+-------------------------------------------+
            ...| NULL                | 9303 | Using where; Using temporary; Using filesort |
            ...| o.object_key        |    1 |                                           |
            ...| c.object_category_id |    1 | Using where                               |
            ...+---------------------+------+-------------------------------------------+
```

QEP im JSON Format


```
EXPLAIN FORMAT = JSON SELECT * FROM test WHERE ts = '2016-05-25'\G

{ "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "106648.00"
    },
    "table": {
      "table_name": "test",
      "access_type": "ALL",
      "rows_examined_per_scan": 522500,
      "rows_produced_per_join": 52250,
      "filtered": "10.00",
      "cost_info": {
        "read_cost": "96198.00",
        "eval_cost": "10450.00",
        "prefix_cost": "106648.00",
        "data_read_per_join": "3M"
      },
      "used_columns": [
        "id",
        "data",
        "ts"
      ],
      "attached_condition": "(`test`.`test`.`ts` = '2016-05-25 00:00:00')"
}  }  }
```

Graphischer QEP



`EXPLAIN` Typen

schnell

Type	Bedeutung
const	Höchstens eine Zeile passt, gegen eine Konstante
eq_ref	Join auf PK (Child→Parent), Zeilenanzahl bleibt gleich oder wird kleiner
ref	Join auf FK (Parent→Child), Zeilenanzahl wird möglicherweise/ wahrscheinlich grösser (kritisch für Optimizer!)
index_merge	Mehere Indexresultate werden gemerged
xxx_subquery	Subqueries werden ausgeführt
range	Index Range Scan
index	Index Full Scan
ALL	Full Table Scan

langsam

`EXPLAIN` Extras

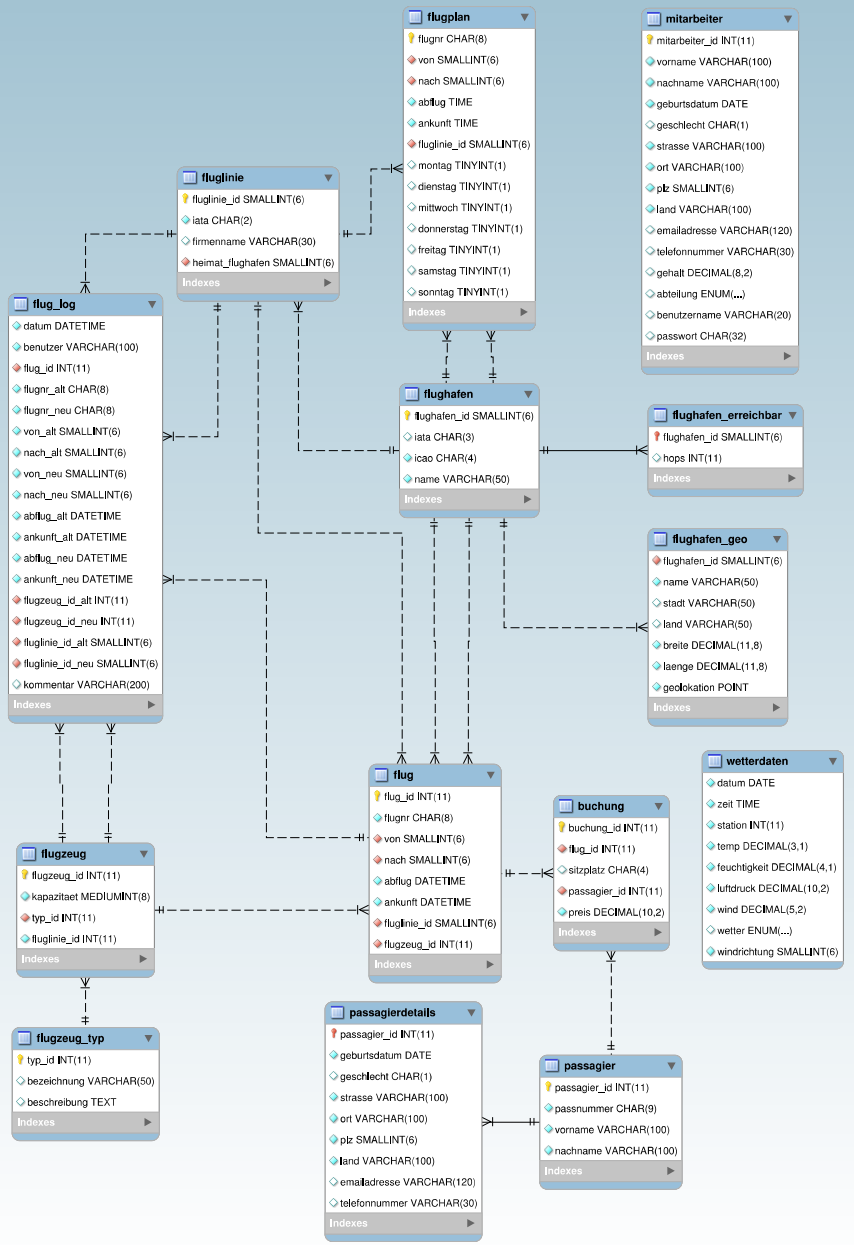
Extra	Bedeutung
Using filesort	Extras Schritt um Sortierordnung zu ermitteln
Using index	Covering Index wird verwendet
Using index condition	Es wird gefiltert bevor die Row geholt wird
Using join buffer (...)	Rows werden in Batches unter Verwendung des Join Buffers verarbeitet (oft schlechtes Zeichen, Index fehlt!).
Using MRR	Die Multi-Range Read Optimierung wird verwendet
Using temporary	Eine temporäre Tabelle wird erstellt werden
Using where	Rows werden über die `WHERE` Klausel rausgefiltert

Vorgehen

- Was braucht man fürs Query Tuning?
  - Das Query! :-)
  - Den Query Execution Plan (QEP)!
  - UND die Query Laufzeit (Ziel)!
  - Tabellenstruktur (`SHOW CREATE TABLE\G`)!
  - Das Datenmodell?
  - Kenntnisse über die Datenverteilung?
  - Realistische Daten(-verteilung)?
  - Realistische Menge von Daten (3 Rows :-( )?
  - Im Idealfall echte Daten?
    - Wenn nicht verfügbar: selber Datenmenge generieren!
- Analysieren und richtige Indices setzen…
  - Wenn Ihr die Queries in cm und nicht mehr in Zeilen messt, dann ist etwas an Eurem Schema Design faul!
  - Fangt mit den einfachen Abfragen an, bis Ihr Übung habt!

Datenmodell

www.fromdual.com



Beispiel 1

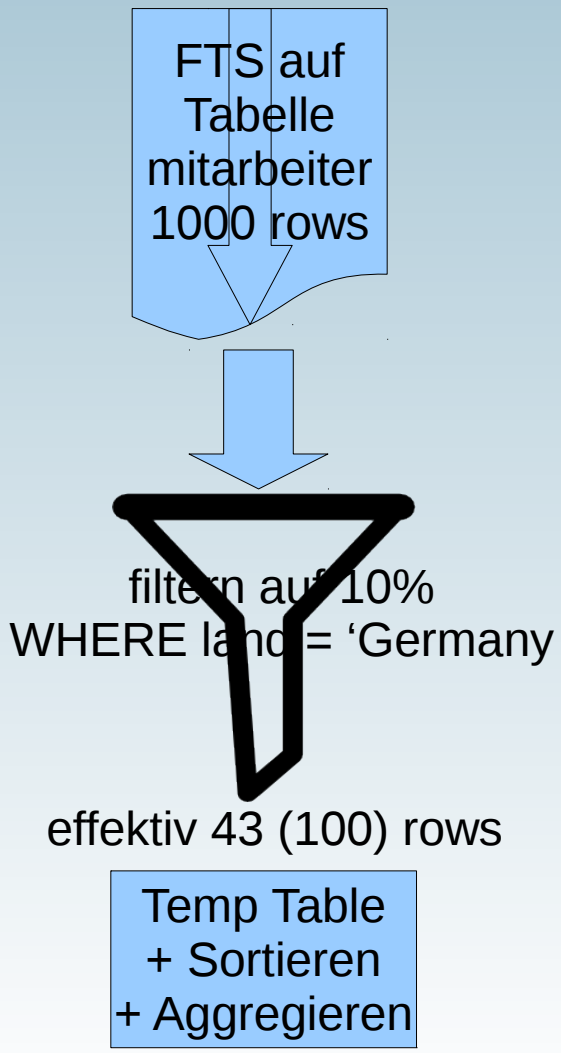

```
EXPLAIN
SELECT LEFT(plz, 1) AS plz_kreis, COUNT(*) AS cnt
  FROM mitarbeiter
 WHERE land = 'Germany'
 GROUP BY plz_kreis
 ORDER BY cnt DESC
;

+-------------+-------------+------+---------------+------+---------+...
| select_type | table       | type | possible_keys | key  | key_len |...
+-------------+-------------+------+---------------+------+---------+...
| SIMPLE      | mitarbeiter | ALL  | NULL          | NULL | NULL    |...
+-------------+-------------+------+---------------+------+---------+...
          ...+------+------+----------+-----------------------------------------------+
          ...| ref  | rows | filtered | Extra                                         |
          ...+------+------+----------+-----------------------------------------------+
          ...| NULL | 1000 |    10.00 | Using where; Using temporary; Using filesort  |
          ...+------+------+----------+-----------------------------------------------+

SHOW CREATE TABLE mitarbeiter\G
CREATE TABLE `mitarbeiter` (
  `mitarbeiter_id` int(11) NOT NULL AUTO_INCREMENT,
  ...
  PRIMARY KEY (`mitarbeiter_id`),
  UNIQUE KEY `benutzer_unq` (`benutzername`)
)
```

Lösung 1

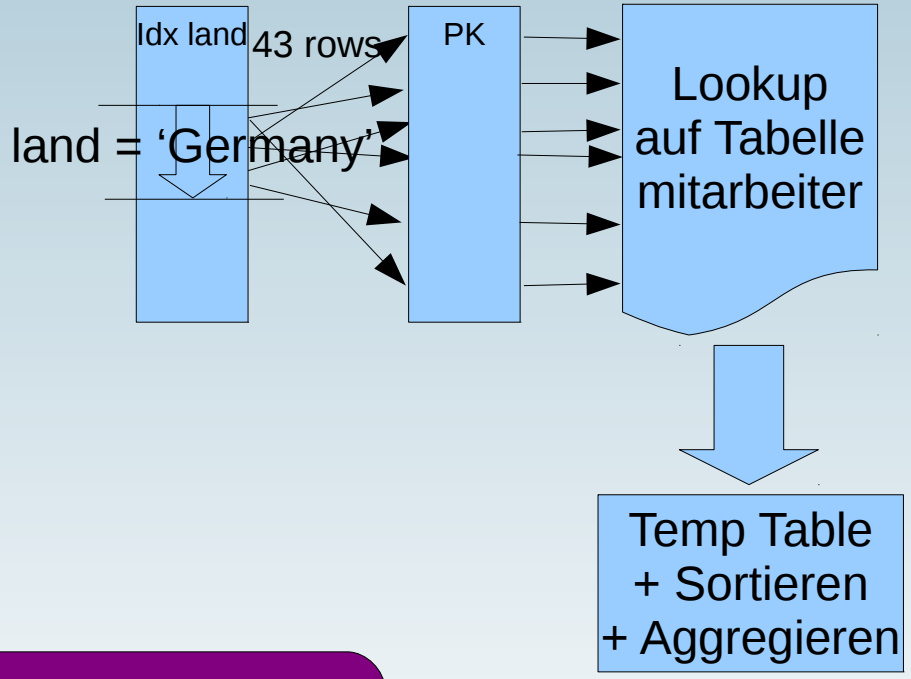

```
ALTER TABLE mitarbeiter ADD INDEX (land);

EXPLAIN SELECT … WHERE land = 'Germany' …;

+-------------+-------------+-------+---------------+-------+---------+...
| select_type | table       | type  | possible_keys | key   | key_len |...
+-------------+-------------+-------+---------------+-------+---------+...
| SIMPLE      | mitarbeiter | ref   | land          | land  | 102     |...
+-------------+-------------+-------+---------------+-------+---------+...
...+-------+------+----------+-----------------------------------------------------+
...| ref   | rows | filtered | Extra                                               |
...+-------+------+----------+-----------------------------------------------------+
...| const |   43 |   100.00 | Using index condition; Using temporary; Using filesort |
...+-------+------+----------+-----------------------------------------------------+

Antwortzeit: 57 → 45 ms (21%) :-)
```

Erklärung 1

- **Vorher**



- **Nachher**



1000 seq
vs
43 + 43 seq + 43 rnd

Beispiel 2

Welche Fluglinie hat Flugzeuge vom Typ A380?

```
EXPLAIN
SELECT ft.bezeichnung, fl.firmenname, COUNT(*)
  FROM flugzeug_typ AS ft
  JOIN flugzeug AS fz ON ft.typ_id = fz.typ_id
  JOIN fluglinie AS fl ON fl.fluglinie_id = fz.fluglinie_id
 WHERE ft.bezeichnung LIKE 'Airbus A38%'
GROUP BY ft.bezeichnung, fl.firmenname
;
```

Laufzeit: 0.130 Sekunden

```
+-------------+-------+--------+----------------------+---------+---------+...
| select_type | table | type   | possible_keys        | key     | key_len |...
+-------------+-------+--------+----------------------+---------+---------+...
| SIMPLE      | fz    | ALL    | NULL                 | NULL    | NULL    |...
| SIMPLE      | fl    | eq_ref | PRIMARY              | PRIMARY | 2       |...
| SIMPLE      | ft    | eq_ref | PRIMARY,idx_fulltext | PRIMARY | 4       |...
+-------------+-------+--------+----------------------+---------+---------+...

...+-----------------+------+----------+----------------------------------+
...| ref             | rows | filtered | Extra                            |
...+-----------------+------+----------+----------------------------------+
...| NULL            | 5583 |   100.00 | Using temporary; Using filesort  |
...| fz.fluglinie_id |    1 |   100.00 | Using where                      |
...| fz.typ_id       |    1 |    11.11 | Using where                      |
...+-----------------+------+----------+----------------------------------+
```

Weitere Infos 2.1


```
SHOW CREATE TABLE flugzeug_typ\G

CREATE TABLE `flugzeug_typ` (
  `typ_id` int(11) NOT NULL AUTO_INCREMENT,
  ...
  PRIMARY KEY (`typ_id`),
)
```

Lösung 2.1


```
ALTER TABLE flugzeug_typ ADD INDEX (bezeichnung);

Laufzeit: 0.025 Sekunden (6 mal schneller)

+-------------+-------+--------+--------------------+-------------+---------+...
| select_type | table | type   | possible_keys      | key         | key_len |...
+-------------+-------+--------+--------------------+-------------+---------+...
| SIMPLE      | ft    | range  | PRIMARY,bezeichnung | bezeichnung | 53      |...
| SIMPLE      | fz    | ALL    | NULL               | NULL        | NULL    |...
| SIMPLE      | fl    | eq_ref | PRIMARY            | PRIMARY     | 2       |...
+-------------+-------+--------+--------------------+-------------+---------+...
  ...+-----------------+------+----------+-----------------------------------------+
  ...| ref             | rows | filtered | Extra                                   |
  ...+-----------------+------+----------+-----------------------------------------+
  ...| NULL            |    1 |   100.00 | Using index condition; Using temporary... |
  ...| NULL            | 5583 |    10.00 | Using where; Using join buffer (Block ... |
  ...| fz.fluglinie_id |    1 |   100.00 | Using where                             |
  ...+-----------------+------+----------+-----------------------------------------+
```

Weitere Infos 2.2


```
SHOW CREATE TABLE flugzeug\G

CREATE TABLE `flugzeug` (
  `flugzeug_id` int(11) NOT NULL AUTO_INCREMENT,
  ...
  `typ_id` int(11) NOT NULL,
  `fluglinie_id` int(11) NOT NULL,
  PRIMARY KEY (`flugzeug_id`),
)
```

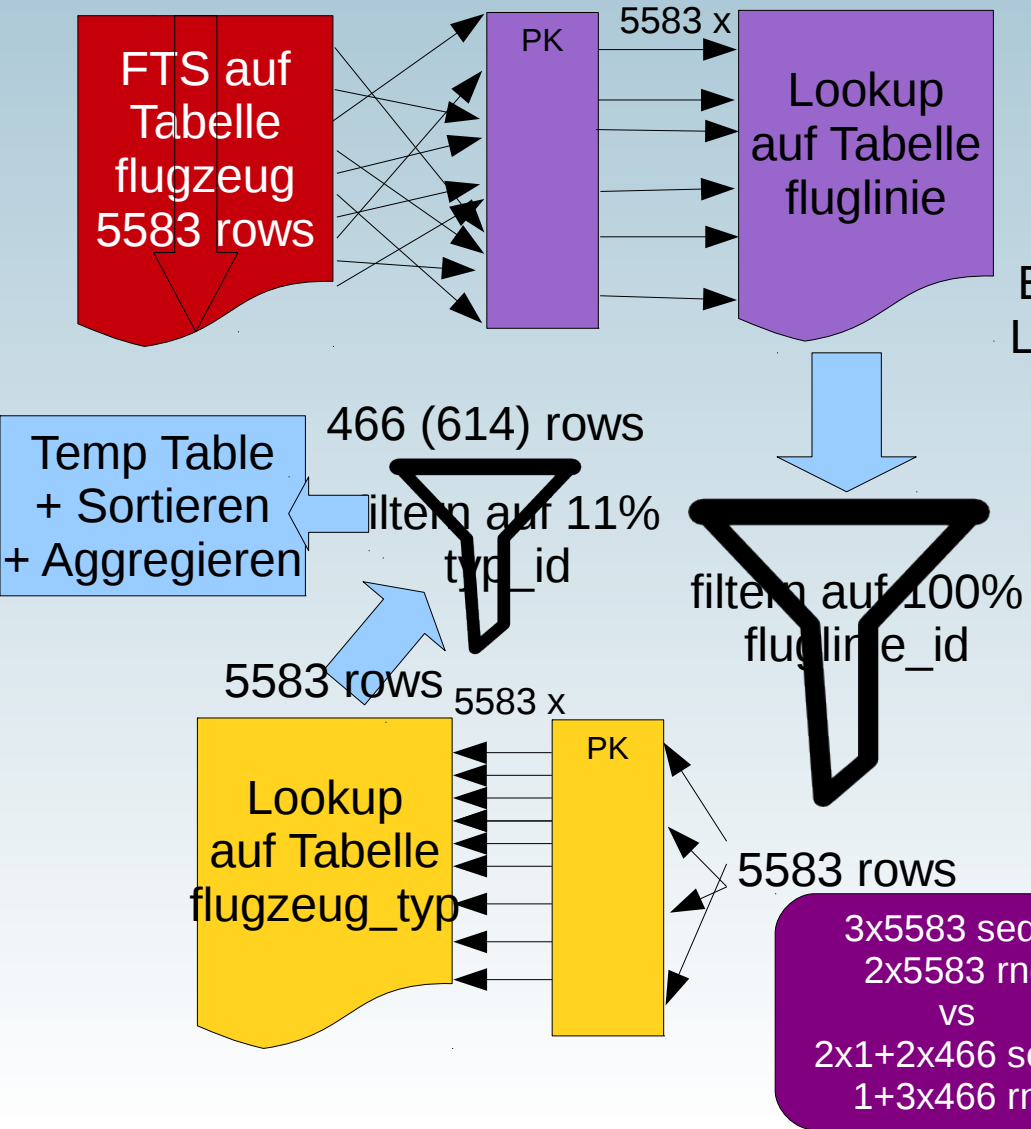
Lösung 2.2


```
ALTER TABLE flugzeug ADD INDEX (typ_id);

Laufzeit: 0.015 Sekunden (1.7 mal schneller (total 8.7 mal schneller))

+-------------+-------+--------+--------------------+------------+---------+...
| select_type | table | type   | possible_keys      | key        | key_len |...
+-------------+-------+--------+--------------------+------------+---------+...
| SIMPLE      | ft    | range  | PRIMARY,bezeichnung | bezeichnung | 53      |...
| SIMPLE      | fz    | ref    | typ_id             | typ_id     | 4       |...
| SIMPLE      | fl    | eq_ref | PRIMARY            | PRIMARY    | 2       |...
+-------------+-------+--------+--------------------+------------+---------+...
...+----------------+------+----------+----------------------------------------------+
...| ref            | rows | filtered | Extra                                        |
...+----------------+------+----------+----------------------------------------------+
...| NULL           |    1 |   100.00 | Using index condition; Using temporary; ... |
...| ft.typ_id      |  429 |   100.00 | NULL                                         |
...| fz.fluglinie_id |    1 |   100.00 | Using where                                  |
...+----------------+------+----------+----------------------------------------------+
```

Erklärung 2

www.fromdual.com

- Vorher



- Nachher

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