Rick's RoTs
Rules of Thumb for MySQL
Rick James
Agenda

- Indexing
- Optimization
- Partitioning
- Character Sets
- Galera/PXC
- Datatypes
- How To
- Hardware
- Miscellany
INDEXing

A Mini Lesson
INDEX Purpose & Design

- Index may greatly speed up `SELECT`
- Adding indexes is not a panacea
- BTree – good all around
INDEX -- 2

• Start INDEX with "=" from WHERE
• Avoid: WHERE func(col) = 'const'
  • flip: WHERE col = inverse('const')
• Hard to opt: WHERE active = 1
• Only 1 index used per SELECT
• Prefix often bad: INDEX(name(10))
• Usually wrong: UNIQUE(name(10))
INDEX -- 3

- \texttt{INDEX(a,b)} \neq \texttt{INDEX(a)}, \texttt{INDEX(b)}
- \texttt{INDEX(a,b)} handles \texttt{INDEX(a)}, \texttt{not (b)}
- \texttt{INDEX(a,b,c,d,e)} – may be excessive
- "Using index" = "Covering index" = 2x speedup
INDEX -- 4

• Index is shunned if need > ~20% of table

• Avoid **USE/FORCE/IGNORE INDEX, STRAIGHT_JOIN**
  • except in desperation
Index Q&A

1 question (hold rest until end)

Index Cookbook (includes 7 tips on efficient many:many mapping tables):
mysql.rjweb.org/doc.php/index_cookbook_mysql
Optimization

(or not)
Opt 1 -- Subqueries

• Subqueries *may* perform poorly
  • Turn into **JOIN** where possible
  • Even with 5.6's auto-key, subquery slow

• **IN ( SELECT ... )** – especially bad
• **( ... GROUP / LIMIT )** – *may* be good
Opt 2

- **OR $\Rightarrow$ UNION**
- "Using Temporary" and "Filesorit" --
  - not the end of the world
  - does not necessarily mean hitting the disk
Opt 3

- Clustered Data is 10x faster (less I/O)
  - Range by PK in InnnoDB
- 1000 qps (YMMV)
- SlowLog is best clue
- No: mix `DISTINCT` and `GROUP BY`
- On `UNION`, explicitly `ALL` or `DISTINCT`
- `JOIN` + `GROUP BY` over-counts aggregates
Opt 4

• How serious are optimizations?
  • 1K rows: Yawn
  • 1M rows: Serious
  • 1B rows: You'll need more than these RoTs
Opt 5

- < 10% improvement ⇒ don't bother
  - Except: do datatypes 'right' up front
- Normalize, but don't over-normalize
- Protect against "SQL injection"
- InnoDB transaction length:
  - if > 5 seconds, redesign
Optimization Q&A

1 question (hold rest until end)
PARTITIONing

Are you sure?
Partition - When?

- Don't use PARTITION, unless…
  - You know that it will help
  - > 1M rows
- No UNIQUE, FOREIGN KEY (maybe 8.x?)
Partition - Use Cases

- Sliding time
- 2D index needed
- Hot partition + messy indexes
- 'Transportable tablespaces'
Partition - Limits

• Only **BY RANGE**
• No **SUBPARTITION**
• No index should start with Partition key
• **AUTO_INCREMENT**
  • need not be **PRIMARY KEY**
  • must be 1st col of *some* key
  • 20-50 partitions/table (more in 8.0?)
Partition - Q&A

1 question (hold rest until end)

mariadb.com/kb/en/mariadb/partition-maintenance/
CHARACTER SETs

And COLLATION
Notation

- **UTF-8** is what the rest of the world calls it
- **utf8mb4** is the equivalent in MySQL
  - **utf8** is a subset of **utf8mb4**
- "Unicode" is related, but *not* what to use in text

- **CHARACTER SET** != **COLLATION**
- UTF-8 != Unicode
Character set

• Use `utf8mb4` for text
  • `utf8` fails to handle Emoji and some of Chinese

• Use `ascii` or `latin1`
  • for hex/ascii
    • GUID, UUID, md5, sha1
    • IP address
    • country_code, postal_code, ...
Character set debugging

- **HEX**(col)
- **LENGTH**(col) – bytes
- **CHAR_LENGTH**(col) – characters
Common corruption

For Señor, you might see

• Señor – Question marks
• Señor – Mojibake or Double-encoding
• Señor – Black diamond
• Se – Truncation
Best Practice

• Outside: Use \texttt{UTF-8} for bytes, editor, and client
• Connection:
  \begin{verbatim}
  SET NAMES utf8mb4;
  \end{verbatim}
  • or some API-specific equivalent
• Have the column/table declared
  \begin{verbatim}
  <CHARACTER SET utf8mb4
  \end{verbatim}
• HTML – starting and forms:
  \begin{verbatim}
  <meta charset=UTF-8>
  <form accept-charset="UTF-8">
  \end{verbatim}
**COLLATION**

Least realistic to most:

<table>
<thead>
<tr>
<th>Collation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>utf8mb4_bin</td>
<td>just compare bits</td>
</tr>
<tr>
<td>utf8mb4_general_ci</td>
<td>no multi-char equiv</td>
</tr>
<tr>
<td>utf8mb4_unicode_ci</td>
<td>old Unicode</td>
</tr>
<tr>
<td>utf8mb4_unicode_520_ci</td>
<td>5.20</td>
</tr>
<tr>
<td>utf8mb4_0900_ai_ci</td>
<td>9.0 (in 8.0)</td>
</tr>
</tbody>
</table>

Case folding and Accent stripping go together 😊
Index too large (767)

To work around this error, do one of

- Change 255 to 191 on the **VARCHAR** (but limit column size)
- **ALTER .. CONVERT TO utf8** (but disallow Emoji and some Chinese)
- Use a "prefix" index (ill-advised)
- Reconfigure (for 5.6.3)
- Upgrade to 5.7.7 (or later)
Character Set - Q&A

1 question (hold rest until end)

More on common troubles and solutions:
stackoverflow.com/a/38363567/1766831

Collations:
mysql.rjweb.org/utf8_collations.html
Galera / PXC

Galera and XtraDB Cluster
Galera - on Local Node

- Check for errors even after `COMMIT`
- `AUTO_INCREMENT` values not consecutive
Galera - on Local Node

- Transactions may be faster or slower, even between datacenters
- Best HA: 3 datacenters, 1+ node each

```
SET SESSION wsrep_sync_wait = 1;
before SELECT
```
Galera - on Local Node

• For Backup, testing, upgrades, alter, ...:
  • Remove node from cluster;
  • Do the task;
  • Put back in cluster; syncup is auto
HA Competition

- PXC is Galera, plus a few mods
- Group Replication (from Oracle) seems to be good competition
- Fabric (from Oracle) seems to be dead
- MHA and Orchestrator are good, but not quite in the same niche
- Traditional Replication with Dual-Masters – less desirable
Galera Q&A

1 question (hold rest until end)

If you *might* use PXC / Galera, code for it anyway

Tips for Programmers/DBAs:
[maria.db.com/kb/en/mariadb/tips-on-converting-to-galera/](maria.db.com/kb/en/mariadb/tips-on-converting-to-galera/)
Datatypes

Declaring Columns (do it right to start with)
Datatypes - 1

- **DATETIME**, not **DATE** and **TIME**
- Usually **UNSIGNED**
- Usually **NOT NULL**
Datatypes - 2

• Overlap test:

```sql
WHERE a.start < b.end
AND a.end > b.start
```

• `SEQUENCE` ⇒ `AUTO_INCREMENT`

  • See also MariaDB's sequence tables
Datatypes - Sizes

- **INT (2)** ain't what you think!
  - Learn the sizes (INT is 4 bytes, etc)
- **BIGINT** – do you really need such large
- *No: FLOAT/Doubles* for money - use **DECIMAL**
- *Never: FLOAT (m, n)*
- Eschew **VARCHAR (255)**
- **VARCHAR**, not **CHAR** (unless truly fixed len)
Datatypes - Custom

- GUID/UUID/MD5 as a key: Random, hence slow
- IP address (IPv6) - \textbf{VARBINARY (39)} or \textbf{BINARY (16)}
- \textit{No}: Credit cards, SSNs, etc – Security issues
- Lat/Ing: \textbf{DECIMAL ( , 4)}
1 question (hold rest until end)

IP ranges:
mariadb.com/kb/en/ip-range-table-performance/

Find the 10 nearest Starbucks:
mariadb.com/kb/en/latitudelongitude-indexing/
Tips on various problems
Pagination

• No: Pagination via **OFFSET** and **LIMIT**
  • Can display dup/missing rows
  • Instead, remember where "left off"
Fast INSERT

• **LOAD DATA**

• **batch INSERT**
  • 100-1000 rows per batch
    • Replication-friendly
    • Transaction per batch
Fast DELETE

- **DELETE**
  - 1000 rows per **DELETE**
  - Chunk on primary key
- Delete via **DROP PARTITION**
Date range tip

\[ dt \geq '2017-02-26' \text{ AND } dt < '2017-02-26' + \text{ INTERVAL 7 DAY} \]

- avoids end second
- avoids leapday (etc) hassle
- works for DATE, DATETIME, DATETIME(6), TIMESTAMP, etc.
Data Warehouse

• Create & maintain Summary Tables

• Do not normalize "continuous" values (dates, floats)

• With MariaDB, consider ColumnStore (nee InfoBright), TokuDB
Entity-Attribute-Value

• Don't use Key-value schema
  • Instead toss into JSON blob
How To - Q&A

1 question (hold rest until end)

14 Tough Tasks (including the above) - See the first section of this: mysql.rjweb.org/
Hardware

Only a 1-time performance fix
Hardware

• Fix the schema, not the hardware
• 10x speedup for cached data
• When timing:
  • `SQL_NO_CACHE` (to avoid Query cache)
  • Run twice (to allow for other caching)
Disk

- "Count the disk hits"
- 100 IOPs on plain drives
  - more on SSDs

- RAID
  - N times as fast (striping)
  - BBWC ⇒ "instant write"

- No: manually separating tables across filesystems
Hardware CPU / IO

• MySQL uses *only 1* CPU core per connection
  • PARTITION or UNION – still only 1
• High CPU ⇒ fix slow query
  • often need "composite" index
• High I/O ⇒ tuning / schema / index
• Linux: use XFS filesystem
Memory Allocation

- InnoDB: 70% of RAM for buffer_pool
  - lower % for tiny VMs
- Other tunables – defaults usually OK
  - You won't hit any hard limits
- Do *not* let mysqld swap!
- Query_cache_type/size = OFF/0
Hardware Q&A

1 question (hold rest until end)

Hard Limits:
mysql.rjweb.org/doc.php/limits
Miscellany

(What did not fit above)
Numbers

• 1000 qps (YMMV)
• SlowLog is best clue
• No: 1000 databases
• No: 1000 tables in a db

• Tame **MaxClients**
• **SHOW CREATE TABLE** is more descriptive than **DESCRIBE**
SELECTs

- No: `SELECT *`
  - Except debugging or into client hash
- `COUNT(*)`, not `COUNT(x)`
  - The latter checks each for not `NULL`
- No: mix `DISTINCT` and `GROUP BY`
- Aggregate counts/sums are inflated when doing both `JOIN` and `GROUP BY`
Processing

- Linux "Load Average" is of little use
- Profiling is of little use
  - nor Performance Schema

- Threads_running > 10 *may* mean trouble
- "Don't queue it, just do it."
Locked

- **SHOW PROCESSLIST** saying "Locked":
  - Look for other process that is hogging
  - or transaction that failed to **COMMIT**
    - *autocommit=0* begs you to forget to **COMMIT**
ALTER

• Usually combine multiple ALTERs to get rebuild all at once
• 5.6.5 allows for many ALTERs to be done
  ALGORITHM=INPLACE
• pt-online-schema-change
MyISAM

- MyISAM is dying; don't use it
  - Gone in 8.0

Conversion tips:
mysql.rjweb.org/doc.php/myisam2innodb
Miscellany - Q&A

1 question (hold rest until end)

Memory Allocation:
mysql.rjweb.org/doc.php/memory
Closing

Let the questions flow!
Rate My Session

Rick's RoTs – Slides / more details
mysql.rjweb.org/slides/rots.pdf
mysql.rjweb.org/doc.php/ricksrots

Rick: mysql@rjweb.org
mysql.rjweb.org/