# **Optimizing MySQL: The Basics of Query Optimization**

In this blog, we will walk you through the basics of query optimization in MySQL. Have a read!

MySQL is a very interesting beast to deal with – both junior developers and expert database administrators working with the RDBMS know that the world of MySQL offers a breadth of opportunities not found anywhere else: people can choose a storage engine to use from a list of powerful storage engines, the RDBMS can be heavily optimized to serve very specific needs, MySQL has multiple specific types of tools and operations in place that can be used to optimize query performance, etc.

One of the most frequent causes of headaches for database administrators for ages are queries and their optimization for speed. As old as this question is, it’s still very hot. [Have a skim through database-related questions posted on StackOverflow within the present week](https://stackoverflow.com/questions/tagged/query-optimization) – do you see any patterns? “Slow query…”, “Performance Improvement…“, “Indexing strategy for..”, “Preventing toxic selects”, “Database not using index.” All of those questions mean one thing – issues concerning database performance are still very important, and people want to know how best to work with their queries to achieve the best mix of performance, availability, security, and capacity.

In the upcoming series, we’re focusing on the optimization of MySQL – we’re going to walk you through storage engine and query optimization, we’re going to tell you a couple of secrets situated around MySQL optimization settings, how best to measure performance, amongst other things. We’re starting small – in this blog, we’re walking you through the basics of the basics and answering the age-old question – ****why are my queries slow****?!

# **Why Your Queries Are Slow**

MySQL offers a couple of types of queries we can choose from:

* INSERT queries that insert data into our database;
* SELECT queries that read through data;
* UPDATE queries that update our data;
* DELETE queries that delete data within our database;
* ALTER queries that change (alter, hence the name) data within our database.

The main reasons those queries are slow are as follows:

* INSERT queries are slow most likely because there is a lot of data to be inserted into the database or (and) because the database isn’t optimized properly.
* SELECT queries are slow mainly because they read through a lot of unnecessary data. The less data there is to scan through, the faster our SELECT queries will become. That’s it – there’s no rocket science involved.
* UPDATE queries are usually slow because we either have an index on the column that is being updated or we are executing a huge update query without familiarizing ourselves with the internals of storage engines – *InnoDB usually locks the table when UPDATEs are being performed and unlocks it after the fact.*
* DELETE queries might be slow because of an index on the column that we’re deleting the data from or because we’re deleting massive amounts of data without knowing the internals of storage engines as well. However, unlike UPDATE queries, DELETE queries have a couple of tricks up their sleeve – they can be massively sped up if we use the TRUNCATE query to remove all of the rows at once, then reimport only the necessary data.
* ALTER queries are usually slow because of improperly optimized operations that run internally:
  + ALTER queries first make a copy of the table.
  + Once the copy of the table is made, the data existing in the original table is copied into it.
  + All of the necessary changes are made.
  + The original table is swapped with the copy.

All of those operations consume time – time that must be accounted for.

Some of such operations can be optimized for performance by properly adjusting the buffer pool used by the InnoDB storage engine, others can be optimized by adding or removing the presence of indexes and partitions, and all of them can be optimized by understanding the internals of queries available within MySQL. For now, though, understand one thing – ****queries are processes comprised of tasks, and to optimize the performance of those processes, we must optimize the tasks within those processes.****

We hope that you’ve enjoyed reading this article and that it helped you grasp the basics of query optimization in the database management system of MySQL – stay tuned to learn how to optimize tasks within processes, [read our blog to learn about database management systems in more detail](https://www.red-gate.com/simple-talk/databases/mysql/), and until next time.