Todd's MySQL Blog

*Tiny tidbits of trivia from Todd*

# Exam Cram: General MySQL Syntax for Developers (Section 2)

10/23/2013 MySQL certification, MySQL 5.6

The *General MySQL Syntax* section of the MySQL 5.6 Developer certification exam is a bit meatier than the *MySQL Architecture* section covered in my last post, but it's still likely to be very familiar to experienced MySQL developers (or DBAs):

- *Explain MySQL implementation of identifiers including case sensitivity, qualified names, aliases and use of reserved words*
- *Identify MySQL data type properties and appropriate usage*
- *Recognize and use common functions and expressions for all MySQL data types*
- *Identify and use comment syntax*
- *Describe and utilize prepared statements*
- *Describe transactions and transaction isolation levels and the impact they have on database behavior*

## MySQL Identifiers

- MySQL treats some words differently, and there's a list of reserved words found in the manual. Not a chance I'll commit this list to memory.
- You can quote these reserved words for use as identifiers, but it's better to avoid them entirely – it *will* cause problems later.
- There's a handful of simple rules for valid identifiers; it's worth committing them to memory. I generally boil them down to:
  - 64 characters maximum
  - If using ASCII symbols other than $ or _, quote it.
  - If using whitespace or control characters, quote it.

  - Can't have ~~leading or~~ trailing whitespace characters (**UPDATE**: only trailing whitespace characters are restricted).
  - If using all numbers for an identifier, quote it.
- Database and table names are represented by files on disk; identifier name case-sensitivity for these objects is tied to the behavior of the filesystem
  - The filesystem-dependent behavior can be overridden with lower_case_table_names – but be careful to prevent stranding pre-existing objects with upper-case characters in the identifiers.
- Know how to alias objects in SQL statements and their scope. Review the documentation dedicated to problems with aliases.

## Data type properties

Picking the right data type for application needs is a fundamental step towards a successful MySQL implementation. The entire data types section of the manual is a masterpiece (thanks to the awesome Docs team!), and should be read in full, but these are my highlights that I'm hitting in my certification exam prep work:

- Review the addition of fractional-section support to temporal data types in MySQL 5.6.
- Become familiar with data type storage requirements.
- Know the default values for various data types.
- Understand the various attributes for numeric data types.
- Demonstrate the automatic initialization and update properties of the TIMESTAMP and DATETIME data type.
- Know the range limitations of various data types.

## Expressions and functions

Chapter 12 of the MySQL reference manual is key, here. If you want a single reference of all operators and functions, this page is great. It's hard to narrow this down, honestly – yet at the same time, I'm unlikely to remember functions like CRC32() or MAKE_SET(). I'll focus on:

- Comparison operators
- The := assignment operator
- Time and date functions
- FULLTEXT search functions
- Certain information functions:
  - CONNECTION_ID()
  - CURRENT_USER(), USER() – and how they differ
  - FOUND_ROWS()
  - LAST_INSERT_ID()
  - ROW_COUNT()
- Aggregate functions

I didn't include string functions, even though there are a number of fundamental functions there. Chances are, if you've used MySQL for a while as a developer, you're sufficiently comfortable with these functions.

## Comments

There's not much to be said about comments beyond what this manual page says. Be familiar with all the different comment syntax, and particularly those which trigger execution solely in MySQL or on certain versions.

## Prepared Statements

The manual page for prepared statements is here. Know how to create, execute and deallocate prepared statements. Make sure you understand the security aspects of prepared statement usage, and that prepared statements are session-specific. There's some useful examples of prepared statements on the referenced manual page, including how to use prepared statements to dynamically define and execute arbitrary SQL. This can be useful particularly in the context of stored routines (discussed later).

## Transactions

There's a bit to understand about transactions in MySQL. Start with the basics here, noting how START TRANSACTION, BEGIN and autocommit variable state interact. You should know which storage engines offer transaction support (and how to evaluate this for any storage engine), which statements are not transactional (cannot be rolled back), and which statements cause an implicit commit of pending transactions. Understanding the explicit table lock syntax here is helpful. I'm skipping manual sections on savepoints and XA transactions, under the hope that I really only need to know that such support exists.

## Practice Questions

Just as before, I make no assertion that the practice questions provided here are representative of what candidates will encounter on the exam. In fact, they mostly represent what I consider to be interesting – but likely edge-case or tricky – behavior encountered while doing my own exam preparation. I wouldn't focus on how many questions you get right – I think this is more a trivia contest that keeps me (and hopefully you) interested and looking to delve a little deeper into the topics, or re-read the relevant documentation with additional context. I hope you find them useful.

## *15 thoughts on “Exam Cram: General MySQL Syntax for Developers (Section 2)”*

Valeriy Kravchuk10/26/2013 at 4:24 AM

This statement: “Can’t have leading or trailing whitespace characters.” is not true about leading while spaces. Check this:

```
mysql> create table ` 1 2 3 ` (c1 int);
ERROR 1103 (42000): Incorrect table name ‘ 1 2 3 ‘
mysql> create table ` 0 1 2 3` (c1 int);
Query OK, 0 rows affected (1.00 sec)

mysql> show create table ` 0 1 2 3`\G
*************************** 1. row ***************************
Table: 0 1 2 3
Create Table: CREATE TABLE ` 0 1 2 3` (
`c1` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```

Manual (http://dev.mysql.com/doc/refman/5.6/en/identifiers.html) also says nothing about leading white spaces.

So, is this intentional to give misleading information? Do you base questions in real exam on the same wrong assumptions in a hope to get less users pass it?

★ Todd Farmer10/26/2013 at 3:03 PM

Valeriy,

Wow, that’s a mighty uncharitable assessment. I wonder if you’re joking; either way, such responses seem likely to suppress community initiatives instead of encourage them. And knowing you, that can hardly be your intent.

Regardless, you are completely accurate in your correction that identifiers only restrict trailing whitespace. There’s an extensive story on how I came to conclude that leading whitespace was prohibited, but at its core, I made a mistake. Thanks for your correction.

As I’ve noted, these are my exam prep notes, not an official study guide, and I’m publishing them in the hopes that others will find them useful in their own study. I’ve taken a lot of care to ensure that the practice questions are accurate, but I’m not referencing actual exam questions (which have separate technical and language review processes) in my study, and I’ve already highlighted that my practice exams don't comply with Oracle standards for exam questions (open-ended multiple choice are prohibited, for example). I also noted that I focus on content that I find interesting, curious or new in my study – and that means some questions would be considered "trick questions" or focus on corner cases. My hope is that people will also find this interesting and will inspire people to look into these topics more fully (and yes, pass the exam).

Anyhow, a lot of work goes into this. I'm sorry you see it as detrimental – or even subversive – to exam candidates, but I appreciate your correction all the same.

Valeriy Kravchuk10/27/2013 at 4:00 AM

Well, I've got a question from one of my Facebook friends about this specific question as a tricky one, and surely we both failed to give proper answer… Then I tried to figure out why I failed and found this small mistake.

So, your goal (that I honestly share) to encourage users to study specific topics and corner cases in details is achieved.

Nice to know that real exam questions are different and checked more thoroughly, but it's not only my feeling that they are going to be tricky and even "evil", and may be based on small notes in the manual that may just document some gotchas (or even known deficiencies in MySQL 5.6) more than good practices or common use cases for new features.

I wanted to share the concerns users have after reading your exam prep notes and sample questions, they are not mine (I do not care much yet about the exams, maybe next month)… But surely I felt bad failing the test in public 🙂

★ Todd Farmer10/28/2013 at 8:46 AM

Valeriy,

The goal of the certification exams is to identify those capable of performing at a high level as a MySQL developer or DBA, not who know the dark and dusty corners of the manual. The whole point of the beta period is to screen questions – which are the best possible indicators successful candidates? If a question deals in minutiae such that qualified and unqualified candidates alike have to guess, we'll see that the question isn't an effective predictor of overall success, and it will be eliminated (both from use in calculating beta participant scores as well as the later GA exams). Likewise, if the question is so remarkably simple that everybody gets it right, it's not useful to us.

I can hardly keep you from speculating that the exam has been crafted to trick candidates and cause as many to fail as possible. I can only point out this doesn't align with the facts, and that the beta period is instrumental in field-testing our (already-tested) questions, and identifying which are and are not good indicators of successful candidates.

If it makes you feel any better, I surely would have failed many of my own practice exam questions before writing them, which is exactly why I wrote them. I find I want to spend more time studying subjects where I've seen there's aspects I've missed (even if they are minutiae). Why would I – or anybody who's been involved with MySQL for several years – want to study identifiers (for example)? For me, I'm motivated when I see clear examples of where there are gaps in my own knowledge.

Valeriy Kravchuk10/29/2013 at 1:26 AM

Thank you for clarifications on what we can expect from exam questions, especially during beta period. I agree with you, this approach is reasonable. I'd probably do the same and add notably more weird and specific questions – this is something with a mindset of many support engineers…

I just wanted to share some FUD community users may get after reading your posts, based on real experience of one of my Facebook friends. Only single case for now, but I had to spend a lot of time explaining there are no evil intentions.

Clear mistakes in "training materials", even unintentional, do NOT help to overcome this FUD, even if they help some of users with a very special mindset (like we have) to prepare to these exams better.

Jonatas12/15/2013 at 3:21 PM

Hi Todd,

Did you take the exam already? What's your thoughts?

I have a question: did the test specifies a subversion of the 5.6? Example, the fractional seconds are included only in MySQL 5.6.4 and up…
If not, what version I should take as reference?

Regards,
Jonatas

★ Todd Farmer12/17/2013 at 10:20 AM

Hi Jonatas,

I did take the exam, and to be honest, I found it difficult. I had planned a pretty comprehensive review in advance, but what I've blogged about is about all that I did. I relied pretty heavily on my (pretty significant) exposure to 5.6 instead of explicitly studying, and I'm not sure whether that will prove sufficient.

With respect to the specific version, I would consider behavior of 5.6.10 (first GA version) specifically. Many features were introduced in various development milestone releases prior to GA, but it all comes together with 5.6.10, and that's what I would use as the reference.

Todd

Jonatas12/18/2013 at 9:20 AM

Hi Todd,

I did the exam today.. wow, was really difficult!
The tips and the tests you wrote helped in several questions, thanks!
Did you receive an email from Oracle/Vue after doing the test? I didn't.
Now lets wait the 11 weeks for the score.

Regards,
Jonatas

★Todd Farmer12/23/2013 at 9:10 AM

Hi Jonatas,

Glad the exam prep helped in some small measure. I know the feeling – I also found it quite challenging, and wished that I had found more time to prepare. I did not get an email (I think the end-of-exam notice lies).

Jonatas02/13/2014 at 12:24 PM

Hi dude,

Any news about the status (pass/fail) of the your exam?

Regards.

★Todd Farmer03/21/2014 at 7:49 AM

Hi Jonatas!

Just got official word yesterday – I passed!

Jonatas03/21/2014 at 8:10 AM

Awesome! Congratulations for us!

David Michel03/17/2014 at 12:44 PM

Hi Todd, are you going to add any additional post regarding the 5.6 Developer cert. I find your blog extremely helpful.

★Todd Farmer03/21/2014 at 7:47 AM

Hi David,

Good timing! I actually received my results yesterday – knowing that I actually passed both exams gives me a little more comfort that the effort to produce study guide posts will be of some benefit. I've got a few other projects I'm busy on now, but aim to get back to the study guide posts in April.

Erico Fusco04/12/2014 at 12:38 AM

Hey Todd,

Thanks a lot for the info. I'm planning to take the test soon. Your topic was definitely helpful.

Cheers!

Erico