

# MySQL implicitly coerces to wrong collation in view

Asked 10 years, 6 months ago   Modified 5 years, 9 months ago   Viewed 13k times



In MySQL 5.6, a view appears to be implicitly coercing `utf8_general_ci` into a `latin1_swedish_ci` instead of the expected `latin1_general_cs`.

11

My setup:



database variables:



```
mysql> show variables like 'col%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| collation_connection | latin1_general_cs |
| collation_database   | latin1_general_cs |
| collation_server     | latin1_general_cs |
+-----+-----+

mysql> show variables like 'char%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | latin1 |
| character_set_connection | latin1 |
| character_set_database | latin1 |
| character_set_filesystem | binary |
| character_set_results | latin1 |
| character_set_server | latin1 |
| character_set_system | utf8 |
| character_sets_dir | /usr/share/mysqlCharsets/ |
+-----+-----+
```

Here's my database and table :

```
CREATE DATABASE `example` /*!40100 DEFAULT CHARACTER SET latin1 COLLATE latin1_general_cs */;

CREATE TABLE `example` (
  `username` varchar(20) COLLATE latin1_general_cs NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;

insert into example values ('user_a');
```

And my view:

```
create or replace view example_view as
select username
```

```
from example
where substring_index(user(), '@', 1) = example.username;
```

## My problem:

When selecting from that view, I get the error:

```
mysql> select * from example_view;
ERROR 1267 (HY000): Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and
(latin1_general_cs,IMPLICIT) for operation '='
```

When I run the select statement *directly*, it works.

As far as I can tell, NOTHING is set to use `latin1_swedish_ci`. The server, database, table, and column are all set to `latin1_general_cs`.

Here's what MySQL thinks the collation is for each part:

```
mysql> select COLLATION(username) as username,
-> COLLATION(user()) as user_func,
-> COLLATION(substring_index(user(), '@', 1)) as substr_func
-> from example;
+-----+-----+-----+
| username          | user_func          | substr_func        |
+-----+-----+-----+
| latin1_general_cs | utf8_general_ci   | utf8_general_ci   |
+-----+-----+-----+
```

So MySQL is trying to convert from `utf8_general_ci` to match the `latin1_general_cs`. But somehow when in the context of a view it decides to use `latin1_swedish_ci` instead.

I'm aware I can just use `convert()` but I'd like to avoid that (partly out of curiosity, partly because lots of `converts()` are going to make for ugly queries).

## My Questions:

Why is MySQL converting to `latin1_swedish_ci` instead of `latin1_general_cs`? How do I fix that, other than using `convert()` explicitly in the query?

mysql collation

Share

edited Jun 3, 2014 at 19:44

Improve this question

Follow

asked Jun 2, 2014 at 23:02



AndyMan1

113 ● 1 ● 1 ● 7



**Q: Why is MySQL converting to `latin1_swedish_ci` instead of `latin1_general_cs` ?**

12



Every character set has a *default* collation. You can use the `SHOW COLLATION` statement to see this. An excerpt from the output shows that `latin1_swedish_ci` is the default collation for the `latin1` character set:



+150



Collation	Charset	Id	Default	Compiled	Sortlen
latin1_german1_ci	latin1	5		Yes	1
latin1_swedish_ci	latin1	8	Yes	Yes	1
latin1_bin	latin1	47		Yes	1
latin1_general_ci	latin1	48		Yes	1
latin1_general_cs	latin1	49		Yes	1

We already know that every table has a default character set and default collation. With the view definition, MySQL is actually creating a table when the query runs.

In the MySQL vernacular, it's called a "derived table".

(As an aside, MySQL does allow some views can be defined with `ALGORITHM=MERGE` instead of the typical and familiar `ALGORITHM=TEMPTABLE` . With the MERGE algorithm, we get view handling behavior that's more like the behavior of other relational databases, like Oracle and SQL Server.)

When MySQL creates the derived table, it assigns a character set along with its the default collation.

That's where the `latin1_swedish_ci` is coming from... the default collation for `latin1`.

---

**Q2: How do I fix that, other than using `CONVERT()` explicitly in the query?**

You can try specifying a collation without the `CONVERT()` function:

```
CREATE VIEW example_view
AS
SELECT username COLLATE latin1_general_cs
FROM example
WHERE SUBSTRING_INDEX(USER(), '@', 1) COLLATE latin1_general_cs =
example.username;
```

(If your client character set is `utf8`, then you're likely to encounter an error with that syntax, if you don't also have the `CONVERT(... USING ...)` . You can use `COLLATE` in

conjunction with the `CONVERT()` function.

```
CONVERT(USER() USING latin1) COLLATE latin1_general_cs
```

NOTE: I don't have any practical experience with stored views; we use inline views all over the place. But we *never* create stored views, because stored views cause a myriad of problems, *way bigger* and *way more* problems than whatever problems the view definition was a solution for.

Share Improve this answer Follow

answered Apr 29, 2015 at 3:34



spencer7593

108k ● 15 ● 117 ● 145

- 1 Your CREATE VIEW statement fixed my "Illegal mix of collations" problem when selecting from a view. The phpmyadmin SQL editor flagged COLLATE as bad syntax in my ALTER VIEW statement, but it ran fine. Strange that when creating a view, MySQL defaults to a collation that's different from the default collation of the database and server. It took me hours of research and experimentation to fix this. – [dcorsello](#) Dec 22, 2021 at 23:52 ✎



1



Not sure what exactly you are asking for, but just to avoid the error message you can:

<http://sqlfiddle.com/#!9/2697e/3>

```
create or replace view example_view as
select username
from example
where substring_index(user(), '@', 1) = example.username COLLATE
latin1_general_cs;
```

<http://sqlfiddle.com/#!9/bf88d/1>

Share

edited Apr 24, 2015 at 16:34

answered Apr 24, 2015 at 13:57

Improve this answer

Follow



Alex

17.3k ● 1 ● 30 ● 52

Can you explain why the collation on the comparison defaults to `latin1_general_ci` rather than inheriting the collation of the column being compared (i.e. `example.username`) or the collation of `user()`? – [Conspicuous Compiler](#) Apr 24, 2015 at 16:27 ✎

(It looks the coercion is explained by [dev.mysql.com/doc/refman/5.6/en/](http://dev.mysql.com/doc/refman/5.6/en/), especially the last paragraph.) – [Conspicuous Compiler](#) Apr 24, 2015 at 16:33 ✎

No I can't. And I still don't understand why CHARSET and COLLATION are 2 different things in mysql. I would prefer just use the CHARSET. You can check 2nd fiddle to see how mysql settings impact the result of the same query. But I have no explanation why it goes this way. – [Alex](#) Apr 24, 2015 at 16:36

- 1 Roughly speaking, the **character set** defines the encoding of valid characters, how each "character" is represented in binary, as a series of bits. And the **collation** defines the semantics for comparison and ordering of the represented characters. For example, whether or not a lowercase 'a' is considered to be equal to uppercase 'A'. With european/slavic languages, we've got to handle umlauts, accent grave, et al. and whether an "umlaut u" is considered to be equal to a regular "u", etc. It's two separate things because comparison/ordering is separate from the binary representation – [spencer7593](#) Apr 30, 2015 at 2:53
- 



I had similar issues, but I changed `my.ini`:

0

```
ProgramData\MySQL\MySQL Server 5.7\my.ini
```



In the file, under `[mysqld]` part I put these lines (which I needed):



```
character-set-server=utf8
```

```
collation-server=utf8_hungarian_ci
```

Share

Improve this answer

Follow

edited Mar 5, 2019 at 7:20



[adiga](#)

35.2k ● 9 ● 65 ● 87

answered Mar 5, 2019 at 7:08



[Zeghra](#)

477 ● 5 ● 14