

Database - Bug #8030

unique index violation when creating a record in two sessions

11/11/2023 03:30 AM - Constantin Asofiei

Status:	Test	Start date:	
Priority:	Urgent	Due date:	
Assignee:	Dănuț Filimon	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#2 - 11/11/2023 03:43 AM - Constantin Asofiei

There is a case for #7669, which I don't have yet a complete standalone recreate; it looks like this, ran from two clients concurrently:

```
do transaction:
  create book.
  assign
    book-id = 1234
    isbn = "abdef"
  no-error.
end.
```

The problem is that it fails with a SQL unique constraint violation abend, like this, in MariaDB:

```
2774 java.sql.SQLIntegrityConstraintViolationException: (conn=5249) Duplicate entry '...' for key 'idx___...
.'
2775 at org.mariadb.jdbc.export.ExceptionFactory.createException(ExceptionFactory.java:290)
2776 at org.mariadb.jdbc.export.ExceptionFactory.create(ExceptionFactory.java:370)
2777 at org.mariadb.jdbc.message.ClientMessage.readPacket(ClientMessage.java:134)
2778 at org.mariadb.jdbc.client.impl.StandardClient.readPacket(StandardClient.java:855)
2779 at org.mariadb.jdbc.client.impl.StandardClient.readResults(StandardClient.java:794)
2780 at org.mariadb.jdbc.client.impl.StandardClient.readResponse(StandardClient.java:713)
2781 at org.mariadb.jdbc.client.impl.StandardClient.execute(StandardClient.java:637)
2782 at org.mariadb.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:95)
2783 at org.mariadb.jdbc.ClientPreparedStatement.execute(ClientPreparedStatement.java:268)
2784 at com.mchange.v2.c3p0.impl.NewProxyPreparedStatement.execute(NewProxyPreparedStatement.java:6
7)
2785 at com.goldencode.p2j.persist.orm.SQLExecutor.execute(SQLExecutor.java:139)
2786 at com.goldencode.p2j.persist.orm.Persister.insert(Persister.java:927)
2787 at com.goldencode.p2j.persist.orm.Persister.insert(Persister.java:427)
2788 at com.goldencode.p2j.persist.orm.Session.save(Session.java:781)
2789 at com.goldencode.p2j.persist.orm.Validation.flush(Validation.java:608)
2790 at com.goldencode.p2j.persist.orm.Validation.validateMaybeFlush(Validation.java:345)
2791 at com.goldencode.p2j.persist.RecordBuffer.validateMaybeFlush(RecordBuffer.java:10790)
2792 at com.goldencode.p2j.persist.RecordBuffer.flush(RecordBuffer.java:6700)
2793 at com.goldencode.p2j.persist.RecordBuffer.validate(RecordBuffer.java:6232)
2794 at com.goldencode.p2j.persist.TxWrapper$WorkArea.validate(TxWrapper.java:907)
2795 at com.goldencode.p2j.util.TransactionManager.lambda$processValidate$4(TransactionManager.java
:7176)
2796 at com.goldencode.p2j.jmx.NanoTimer.timer(NanoTimer.java:131)
2797 at com.goldencode.p2j.util.TransactionManager.processValidate(TransactionManager.java:7154)
2798 at com.goldencode.p2j.util.TransactionManager.access$10700(TransactionManager.java:711)
2799 at com.goldencode.p2j.util.TransactionManager$TransactionHelper.processValidate(TransactionMan
ager.java:10038)
2800 at com.goldencode.p2j.util.BlockManager.lambda$processBody$0(BlockManager.java:8927)
2801 at com.goldencode.p2j.util.BlockManager.executeStateManaged(BlockManager.java:11949)
```

```
2802         at com.goldencode.p2j.util.BlockManager.processBody (BlockManager.java:8927)
2803         at com.goldencode.p2j.util.BlockManager.doBlockWorker (BlockManager.java:10109)
2804         at com.goldencode.p2j.util.BlockManager.doBlock (BlockManager.java:1448)
```

Tomasz: is this something you noticed with the #6667 work? The issue is that this record can be created by multiple clients at the same time, and FWD somehow misses the new record.

#3 - 11/11/2023 02:51 PM - Constantin Asofiei

OK, this message seems to be a red herring: is just logged when this test is executed twice in a row:

```
do transaction on error undo, leave:
  create book.
  assign
    book-id = 1234
    isbn = "abdef"
  no-error.
end.
```

The logging happens in `orm.Persister.insert`:

```
catch (SQLException exc)
{
  if (LOG.isLoggable(Level.WARNING))
  {
    LOG.log(Level.WARNING, "Error during insert", exc);
  }

  throw new PersistenceException(exc);
}
```

I guess the logging can be reduced to FINE?

Anyway, there is another issue (which seems real) for MariaDB; this code in `orm.Validation.flush()` which catches the SQL exception:

```
catch (PersistenceException exc)
{
  // validation failed, or we had a different error; roll back to savepoint
  // TODO: is this always safe (or even possible) in the event of a non-unique constraint
  //       related database error?
  if (sp != null)
  {
    session.rollbackSavepoint(sp);
  }
  else if (fullTx)
  {
    session.rollback();
  }

  // go up the exception chain and look for a SQL exception:
  Throwable cause = exc.getCause();
  while (cause != null &&
        !(cause instanceof SQLException) &&
        cause != cause.getCause())
  {
```

```

        cause = cause.getCause();
    }
    if (cause instanceof SQLException)
    {
        SQLException sqle = (SQLException) cause;
        if ("23505".equals(sqle.getSQLState())) // UNIQUE VIOLATION - unique_violation
        {
            // if the right index is found, this method will not return normally, instead it will
            // throw a specific exception
            validateUniqueByQuery(null);
        }
    }

    // if the exception could not be identified as a SQL UNIQUE VIOLATION exception or the
    // generateUniqueIndexesCondition() method could not identify the right index to compose the
    // P4GL error message, we let the caller to handle the exception
    throw new ValidationException(exc.getMessage(), exc);
}

```

assumes:

```
if ("23505".equals(sqle.getSQLState())) // UNIQUE VIOLATION - unique_violation
```

This is not true for MariaDB - SQLState is 23000. I guess this code needs to be dialect-specific.

SQLState is also used in `RecordBuffer.reportValidationException` - these need to be reviewed and made dialect-specific.

#4 - 11/13/2023 03:12 AM - Tomasz Domin

Constantin Asofiei wrote:

There is a case for #7669, which I don't have yet a complete standalone recreate; it looks like this, ran from two clients concurrently:
[...]

The problem is that it fails with a SQL unique constraint violation abend, like this, in MariaDB:
[...]

Tomasz: is this something you noticed with the #6667 work? The issue is that this record can be created by multiple clients at the same time, and FWD somehow misses the new record.

No, there was nothing like that for #6667.

#5 - 11/13/2023 03:45 AM - Alexandru Lungu

Constantin, just a further notice on [#8030-3](#). In H2, the first SQLException is an internal H2 exception that has another code (I don't recall if it was 23000), **but** it is caused by another SQLException that has that internal state set on 23505. Check if this is also true for MariaDB.

My point is that we may need to continue "unwrapping" the cause of the exception until we find one that has "23505". This change is on another branch we have (i.e. 7323a), but it has way more other changes regarding validation of `_temp`. If you can test for MariaDB that continuing to unwrap the cause is fine **and** this is not a very high priority, we can wait for 7323a merge. My intention is to have it tested and merged by the end of this/next week.

#6 - 11/13/2023 03:46 AM - Alexandru Lungu

I added Radu as a watcher; he is the author of 7323a.

#7 - 11/13/2023 04:37 AM - Constantin Asofiei

The `java.sql.SQLIntegrityConstraintViolationException.cause` field is null for MariaDB.

#8 - 11/13/2023 10:38 AM - Alexandru Lungu

- Assignee set to *Dănuț Filimon*
- Status changed from *New* to *WIP*

Constantin Asofiei wrote:

The `java.sql.SQLIntegrityConstraintViolationException.cause` field is null for MariaDB.

Danut, please attempt to fix this using a dialect specific handler:

- PG: first exception with 23505 is enough
- Maria: first exception with 23000 is enough
- H2: unwrap until you find 23505

Created 8030a.

#9 - 11/14/2023 04:00 AM - Dănuț Filimon

I was not able to reproduce the scenario with **trunk/rev.14828**, especially make it to reach the catch from `orm.Persistor.insert`. I ran the test with 2 parallel clients and I only obtained a `ValidationException` in 2 out of 20 runs.

#10 - 11/14/2023 04:02 AM - Constantin Asofiei

Dănuț Filimon wrote:

I was not able to reproduce the scenario with **trunk/rev.14828**, especially make it to reach the catch from `orm.Persister.insert`. I ran the test with 2 parallel clients and I only obtained a `ValidationException` in 2 out of 20 runs.

Don't use parallel clients, run the same client twice. Also you need to use MariaDB for this.

#11 - 11/14/2023 04:05 AM - Dănuț Filimon

Constantin Asofiei wrote:

Don't use parallel clients, run the same client twice. Also you need to use MariaDB for this.

I am using MariaDB. The outcome is the same even when running the client twice/multiple times, but without the `ValidationException` part.

#12 - 11/14/2023 04:06 AM - Constantin Asofiei

Dănuț Filimon wrote:

Constantin Asofiei wrote:

Don't use parallel clients, run the same client twice. Also you need to use MariaDB for this.

I am using MariaDB. The outcome is the same even when running the client twice/multiple times, but without the `ValidationException` part.

Please try also with 7156b.

#13 - 11/14/2023 04:35 AM - Alexandru Lungu

Danut, also make sure you use the right schema (having unique indexes on `book-id / isbn`).

#14 - 11/14/2023 04:37 AM - Dănuț Filimon

Alexandru Lungu wrote:

Danut, also make sure you use the right schema (having unique indexes on book-id / isbn).

I am checking out 7156b right now and I already have the schema from #7803.

#15 - 11/14/2023 04:53 AM - Dănuț Filimon

I retested with **7156b/rev.14846** but I couldn't reproduce the error with this branch too.

#16 - 11/27/2023 04:43 AM - Alexandru Lungu

- *Priority changed from Normal to Urgent*

Danut, can you re-attempt to recreate this and fix? Please do some debugging to see where you are failing into recreating it. It is clear that the validation fails once you try to insert two records violating an index.

We need this to be fixed asap so we can move on with #7669.

#17 - 11/27/2023 05:56 AM - Dănuț Filimon

I managed to reproduce the issue after fixing a configuration issue in build_db.xml. Working on it right now.

#18 - 11/27/2023 09:04 AM - Dănuț Filimon

- *% Done changed from 0 to 50*

I committed the fix for [#8030-3](#) in **8030a/rev.14829**, currently investigating RecordBuffer.reportValidationException.

#19 - 11/27/2023 09:20 AM - Dănuț Filimon

Alexandru, in what case is required to unwrap the cause for P2JH2Dialect? I'll need a bit more context to adjust the method if necessary.

#20 - 11/27/2023 09:23 AM - Alexandru Lungu

I can't recall exactly. We have some changes in [#7323](#) that was doing the cause unwrapping for FWD-H2 **exactly** because it was needed. We either get it now through your 8030a, or later through 7323a. However, I am inclined to do it now as this task is especially made to fix this. Just consider that P2JH2 dialect needs deeper searches.

#21 - 11/27/2023 10:15 AM - Alexandru Lungu

Danut, please move this into Review once you finish the implementation so we can move on with the review part. There shouldn't be any consistent testing in my opinion; this is not quite a large functional change.

#22 - 11/28/2023 03:11 AM - Dănuț Filimon

- *Status changed from WIP to Review*

- *% Done changed from 50 to 100*

Committed 8030a/rev.14830. Modified RecordBuffer.reportValidationException.

Alexandru, the example provided did reach the method, but there was only a ValidationException with a null cause. I eventually modified the code (only for **testing** purposes) to have a SQLException with the **23505** SQLState as the cause.

#23 - 11/28/2023 11:16 AM - Eric Faulhaber

Dănuț Filimon wrote:

Committed 8030a/rev.14830. Modified RecordBuffer.reportValidationException.

Alexandru, the example provided did reach the method, but there was only a ValidationException with a null cause. I eventually modified the code (only for **testing** purposes) to have a SQLException with the **23505** SQLState as the cause.

What is the status of this issue specifically for the MariaDB dialect? We need to get a fix into trunk (and ultimately 7156b) ASAP.

#24 - 11/28/2023 11:21 AM - Alexandru Lungu

AFAIK, everything is done on 8030a (including MariaDB - I discussed with Danut and the issue in [#8030-1](#) is no longer visible on MariaDB). The fix is final.

Note: it takes me ~15 min to check-out, so maybe you can do a quick review and merge?

#25 - 11/28/2023 11:31 AM - Eric Faulhaber

Yes, please do.

#26 - 11/28/2023 12:15 PM - Eric Faulhaber

Eric Faulhaber wrote:

Yes, please do.

Sorry, I misread your note. I'll review...

#27 - 11/29/2023 12:36 AM - Eric Faulhaber

Code review 8030a/14829-14830:

The logic changes look fine to me. For clarity, I did change the name of the method Dialect.handleUniqueConstraintViolation to isUniqueConstraintViolation, since technically it is only identifying, not handling. This minor change was committed as 8030a/14831.

I then rebased 8030a to trunk rev 14848, merged 8030a to trunk, and committed on behalf of Dănuț. Trunk is now at rev 14849.

#28 - 11/29/2023 03:31 AM - Alexandru Lungu

- Status changed from Review to Test