

Item 273

git_comments:

git_commits:

1. **summary:** PHOENIX-3925 Disallow usage of ON DUPLICATE KEY clause on tables with global secondary indexes
message: PHOENIX-3925 Disallow usage of ON DUPLICATE KEY clause on tables with global secondary indexes

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Disallow usage of ON DUPLICATE KEY clause on tables with global secondary indexes
description: For the reasons of service protection, rather than just documenting that you shouldn't use the ON DUPLICATE KEY clause on tables with global secondary indexes, we should instead throw an exception if this is attempted. See reasons listed here for the reason:
https://phoenix.apache.org/atomic_upsert.html#Limitations

jira_issues_comments:

1. **body:** We should probably also make the language on the website stronger. Right now it says it's "supported" but "not recommended", as opposed to the other limitations which say the stronger "may not". For 4.11 and up Phoenix will enforce, but it would be good for users of older versions using the docs to get more clarity too.
label: documentation
2. **body:** Agreed. We'll update the docs when 4.11 is released.
label: documentation
3. Please review, [~gjacobby].
4. +1, thanks [~jamestaylor].
5. FAILURE: Integrated in Jenkins build Phoenix-master #1651 (See [<https://builds.apache.org/job/Phoenix-master/1651/>]) PHOENIX-3925 Disallow usage of ON DUPLICATE KEY clause on tables with (jamestaylor: rev 9b402043896fdeb78a236542bddf88e4a7f300e7) * (edit) phoenix-core/src/main/java/org/apache/phoenix/exception/SQLExceptionCode.java * (edit) phoenix-core/src/it/java/org/apache/phoenix/end2end/OnDuplicateKeyIT.java * (edit) phoenix-core/src/test/java/org/apache/phoenix/compile/QueryCompilerTest.java * (edit) phoenix-core/src/main/java/org/apache/phoenix/compile/UpsertCompiler.java * (edit) phoenix-core/src/main/java/org/apache/phoenix/util/SchemaUtil.java
6. Updated website documentation to this: "Global indexes on columns being atomically updated are not supported, as potentially a separate RPC across the wire would be made while the row is under lock to maintain the secondary index."