

# [OBGM-626] Deployment with newly created database fails due database migration failure Created: 07/21/23 Updated:

07/22/23

<b>Status:</b>	In Progress
<b>Project:</b>	<a href="#">OpenBoxes Grails Migration</a>
<b>Components:</b>	None
<b>Fix versions:</b>	<a href="#">MVP1</a>

<b>Type:</b>	Bug	<b>Priority:</b>	Must
<b>Reporter:</b>	<a href="#">Justin Miranda</a>	<b>Assignee:</b>	<a href="#">Justin Miranda</a>
<b>Resolution:</b>	Unresolved	<b>Votes:</b>	0
<b>Labels:</b>	None		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Attachments:</b>	 <a href="#">azure-openboxes.log</a>
<b>Sprint:</b>	Sprint 121 - Intuitive Cicada
<b>Epic Link:</b>	<a href="#">Database Migration</a>

## Description

This happens when deploying to Azure, but I assume the same will happen on any new database

## Stacktrace

```
Jul 21 21:05:31 OB-VM java[7293]: 2023-07-21 21:05:31,978 ERROR [main ] liquibase.changelog.ChangeSet : Change Set
views/changelog.xml::1580848680306-7::jmiranda failed.
Error: Unknown column 'order.order_type_id' in 'where clause' [Failed SQL: (1054) CREATE OR REPLACE VIEW
on_order_order_item_summary AS
Jul 21 21:05:31 OB-VM java[7293]: (
Jul 21 21:05:31 OB-VM java[7293]: SELECT a.product_id,
Jul 21 21:05:31 OB-VM java[7293]:         a.destination_id,
Jul 21 21:05:31 OB-VM java[7293]:         ifnull(sum(case
Jul 21 21:05:31 OB-VM java[7293]:             when a.quantity_ordered - a.quantity_shipped > 0
Jul 21 21:05:31 OB-VM java[7293]:                 then a.quantity_ordered - a.quantity_shipped
Jul 21 21:05:31 OB-VM java[7293]:             else 0 end), 0) as quantity_ordered_not_shipped,
Jul 21 21:05:31 OB-VM java[7293]:         null as
quantity_shipped_not_received
Jul 21 21:05:31 OB-VM java[7293]: FROM (
Jul 21 21:05:31 OB-VM java[7293]:         SELECT product.id as product_id,
Jul 21 21:05:31 OB-VM java[7293]:                order_item.id,
Jul 21 21:05:31 OB-VM java[7293]:                `order`.destination_id,
Jul 21 21:05:31 OB-VM java[7293]:                sum(distinct case
Jul 21 21:05:31 OB-VM java[7293]:                    when `order`.status != 'PENDING'
Jul 21 21:05:31 OB-VM java[7293]:                        then order_item.quantity * order_item.quantity_per_uom
Jul 21 21:05:31 OB-VM java[7293]:                    else 0 end) as quantity_ordered,
Jul 21 21:05:31 OB-VM java[7293]:                sum(case
Jul 21 21:05:31 OB-VM java[7293]:                    when shipment_item.quantity and shipment_current_status in ('SHIPPED'
```

## Full log file

[azure-openboxes.log](#)

## Comments

Comment by [Justin Miranda](#) [07/21/23]

This is related to an issue on the devops repository <https://github.com/openboxes/openboxes-devops/issues/61>

We need to make sure the base install changelogs are in sync with what's in development (develop) branch. Or we could just regenerate the changelog-create-schema.groovy file using the DBM (Liquibase) plugin. <https://pihemr.atlassian.net/browse/OBGM-199>

In the meantime, we could probably trick the Azure deployment script to run the upgrade changelogs (i.e. execute all of the existing database migrations in 0.5.x - 0.9.x) by adding some bogus lines to the DATABASECHANGELOG table so that hasExecutedAnyPreviousChangesets resolves to true.

```
// Check if the executed changelog versions include one of the previous versions
// and if so, then we need to keep running the old updates to catch up to 0.9.x
```

```
boolean hasExecutedAnyPreviousChangesets =
    executedChangelogVersions.any { previousChangelogVersions.contains(it.version) }

// FIXME Remove !hasExecutedAnyPreviousChangesets once this goes to production
//If nothing has been created yet, let's create all new database objects with the install scripts
List<ChangeSet> hasExecutedAnyChangesets = database.getRanChangeSetList()
if (!hasExecutedAnyChangesets || !hasExecutedAnyPreviousChangesets) {
    log.info("Running install changelog ...")
    liquibase = new Liquibase("install/changelog.xml", new ClassLoaderResourceAccessor(), database)
    liquibase.update(null as Contexts, new LabelExpression());
}

if (hasExecutedAnyPreviousChangesets) {
    log.info("Running upgrade changelog ...")
    liquibase = new Liquibase("upgrade/changelog.xml", new ClassLoaderResourceAccessor(), database)
    liquibase.update(null as Contexts, new LabelExpression())
}
```

The only bummer with that approach would be the time it takes to get the application up and running.

So the sync approach seems like the best option.

Generated at Sat Jul 22 04:00:57 UTC 2023 by Justin Miranda using Jira 1001.0.0-SNAPSHOT#100232-  
sha1:4329f547fe865132f2e3fd6e3684d079c80e4f13.