Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Server version name: Apache Tomcat/7.0.109

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Server built: Apr 22 2021 18:43:38 UTC

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Server version number: 7.0.109.0

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: OS Name: Linux

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: OS Version: 3.10.0-1160.49.1.el7.x86\_64

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Architecture: amd64

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Java Home: /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.261-2.6.22.2.el7\_8.x86\_64/jre

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: JVM Version: 1.7.0\_261-mockbuild\_2020\_04\_29\_08\_59-b00

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: JVM Vendor: Oracle Corporation

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: CATALINA\_BASE: /opt/tomcat

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: CATALINA\_HOME: /opt/tomcat

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Djava.awt.headless=true

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Djava.security.egd=file:/dev/./urandom

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Djdk.tls.ephemeralDHKeySize=2048

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Dorg.apache.catalina.security.SecurityListener.UMASK=0027

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Xms1024m

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Xmx1024m

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -XX:MaxPermSize=512m

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -XX:+UseParallelGC

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Dignore.endorsed.dirs=

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Dcatalina.base=/opt/tomcat

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Dcatalina.home=/opt/tomcat

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.VersionLoggerListener log

INFO: Command line argument: -Djava.io.tmpdir=/opt/tomcat/temp

Dec 24, 2021 2:41:52 PM org.apache.catalina.core.AprLifecycleListener lifecycleEvent

INFO: The Apache Tomcat Native library which allows using OpenSSL was not found on the java.library.path: [/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]

Dec 24, 2021 2:41:52 PM org.apache.coyote.AbstractProtocol init

INFO: Initializing ProtocolHandler ["http-bio-10.0.0.2-8085"]

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.Catalina load

INFO: Initialization processed in 964 ms

Dec 24, 2021 2:41:52 PM org.apache.catalina.core.StandardService startInternal

INFO: Starting service [Catalina]

Dec 24, 2021 2:41:52 PM org.apache.catalina.core.StandardEngine startInternal

INFO: Starting Servlet Engine: Apache Tomcat/7.0.109

Dec 24, 2021 2:41:52 PM org.apache.catalina.startup.HostConfig deployWAR

INFO: Deploying web application archive [/opt/tomcat/webapps/openboxes.war]

Dec 24, 2021 2:42:07 PM org.apache.catalina.startup.TldConfig execute

INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.

Using configuration locations [classpath:openboxes-config.properties, classpath:openboxes-config.groovy, [file:/opt/tomcat/.grails/openboxes-config.properties](file:///C:\opt\tomcat\.grails\openboxes-config.properties), [file:/opt/tomcat/.grails/openboxes-config.groovy](file:///C:\opt\tomcat\.grails\openboxes-config.groovy)] [production]

Unable to load specified config location classpath:openboxes-config.properties : class path resource [openboxes-config.properties] cannot be opened because it does not exist

Unable to load specified config location classpath:openboxes-config.groovy : class path resource [openboxes-config.groovy] cannot be opened because it does not exist

Unable to load specified config location [file:/opt/tomcat/.grails/openboxes-config.groovy](file:///C:\opt\tomcat\.grails\openboxes-config.groovy) : /opt/tomcat/.grails/openboxes-config.groovy (No such file or directory)

log4j:ERROR setFile(null,true) call failed.

java.io.FileNotFoundException: stacktrace.log (Permission denied)

at java.io.FileOutputStream.open(Native Method)

at java.io.FileOutputStream.<init>(FileOutputStream.java:221)

at java.io.FileOutputStream.<init>(FileOutputStream.java:142)

at org.apache.log4j.FileAppender.setFile(FileAppender.java:294)

at org.apache.log4j.FileAppender.activateOptions(FileAppender.java:165)

at org.apache.log4j.spi.OptionHandler$activateOptions.call(Unknown Source)

at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:40)

at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:116)

at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:120)

at org.codehaus.groovy.grails.plugins.logging.Log4jConfig.createFullstackTraceAppender(Log4jConfig.groovy:177)

at org.codehaus.groovy.grails.plugins.logging.Log4jConfig.this$2$createFullstackTraceAppender(Log4jConfig.groovy)

at org.codehaus.groovy.grails.plugins.logging.Log4jConfig$this$2$createFullstackTraceAppender.callCurrent(Unknown Source)

at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCallCurrent(CallSiteArray.java:44)

at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:141)

at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:145)

at org.codehaus.groovy.grails.plugins.logging.Log4jConfig.configure(Log4jConfig.groovy:145)

at org.codehaus.groovy.grails.web.util.Log4jConfigListener.contextInitialized(Log4jConfigListener.java:62)

at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:5128)

at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5653)

at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:183)

at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:1007)

at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:983)

at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:639)

at org.apache.catalina.startup.HostConfig.deployWAR(HostConfig.java:1128)

at org.apache.catalina.startup.HostConfig$DeployWar.run(HostConfig.java:2020)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:473)

at java.util.concurrent.FutureTask.run(FutureTask.java:262)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1152)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)

at java.lang.Thread.run(Thread.java:748)

2021-12-24 14:42:10,725 [localhost-startStop-1] INFO context.ContextLoader - Root WebApplicationContext: initialization started

2021-12-24 14:42:10,743 [localhost-startStop-1] INFO support.XmlWebApplicationContext - Refreshing Root WebApplicationContext: startup date [Fri Dec 24 14:42:10 CET 2021]; root of context hierarchy

2021-12-24 14:42:11,053 [localhost-startStop-1] INFO xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext.xml]

2021-12-24 14:42:11,788 [localhost-startStop-1] INFO support.DefaultListableBeanFactory - Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@1f5cec5a: defining beans [grailsApplication,pluginManager,grailsConfigurator,grailsResourceLoader,grailsResourceHolder,characterEncodingFilter]; root of factory hierarchy

2021-12-24 14:42:24,683 [localhost-startStop-1] INFO cfg.Environment - Hibernate 3.3.1.GA

2021-12-24 14:42:24,684 [localhost-startStop-1] INFO cfg.Environment - hibernate.properties not found

2021-12-24 14:42:24,686 [localhost-startStop-1] INFO cfg.Environment - Bytecode provider name : javassist

2021-12-24 14:42:24,688 [localhost-startStop-1] INFO cfg.Environment - using JDK 1.4 java.sql.Timestamp handling

2021-12-24 14:42:30,796 [localhost-startStop-1] INFO annotation.ClassPathBeanDefinitionScanner - JSR-250 'javax.annotation.ManagedBean' found and supported for component scanning

2021-12-24 14:42:35,093 [MLog-Init-Reporter] INFO log.MLog - MLog clients using slf4j logging.

2021-12-24 14:42:39,536 [localhost-startStop-1] INFO c3p0.C3P0Registry - Initializing c3p0-0.9.5.3 [built 27-January-2019 00:11:37 -0800; debug? true; trace: 10]

2021-12-24 14:42:42,410 [localhost-startStop-1] INFO impl.AbstractPoolBackedDataSource - Initializing c3p0 pool... com.mchange.v2.c3p0.ComboPooledDataSource [ acquireIncrement -> 5, acquireRetryAttempts -> 30, acquireRetryDelay -> 1000, autoCommitOnClose -> false, automaticTestTable -> null, breakAfterAcquireFailure -> false, checkoutTimeout -> 0, connectionCustomizerClassName -> null, connectionTesterClassName -> com.mchange.v2.c3p0.impl.DefaultConnectionTester, contextClassLoaderSource -> caller, dataSourceName -> 1n9yufmal1to4g411ibeir7|5f4ec71b, debugUnreturnedConnectionStackTraces -> false, description -> null, driverClass -> com.mysql.jdbc.Driver, extensions -> {}, factoryClassLocation -> null, forceIgnoreUnresolvedTransactions -> false, forceSynchronousCheckins -> false, forceUseNamedDriverClass -> false, identityToken -> 1n9yufmal1to4g411ibeir7|5f4ec71b, idleConnectionTestPeriod -> 7200, initialPoolSize -> 10, jdbcUrl -> jdbc:mysql://localhost:3306/openboxes?useSSL=false, maxAdministrativeTaskTime -> 0, maxConnectionAge -> 14400, maxIdleTime -> 0, maxIdleTimeExcessConnections -> 1800, maxPoolSize -> 100, maxStatements -> 180, maxStatementsPerConnection -> 0, minPoolSize -> 5, numHelperThreads -> 3, preferredTestQuery -> SELECT 1, privilegeSpawnedThreads -> false, properties -> {user=\*\*\*\*\*\*, password=\*\*\*\*\*\*}, propertyCycle -> 0, statementCacheNumDeferredCloseThreads -> 1, testConnectionOnCheckin -> false, testConnectionOnCheckout -> false, unreturnedConnectionTimeout -> 0, userOverrides -> {}, usesTraditionalReflectiveProxies -> false ]

2021-12-24 14:42:43,856 [localhost-startStop-1] INFO annotations.Version - Hibernate Annotations 3.4.0.GA

2021-12-24 14:42:43,895 [localhost-startStop-1] INFO common.Version - Hibernate Commons Annotations 3.1.0.GA

2021-12-24 14:42:45,324 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [shipmentItemCount] of domain class org.pih.warehouse.shipping.Shipment will not be checked during validation.

2021-12-24 14:42:45,720 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [color] of domain class org.pih.warehouse.product.Product will not be checked during validation.

2021-12-24 14:42:45,794 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [monthRequested] of domain class org.pih.warehouse.requisition.Requisition will not be checked during validation.

2021-12-24 14:42:45,794 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [statusSortOrder] of domain class org.pih.warehouse.requisition.Requisition will not be checked during validation.

2021-12-24 14:42:46,372 [localhost-startStop-1] INFO validator.Version - Hibernate Validator 3.1.0.GA

2021-12-24 14:42:46,832 [localhost-startStop-1] INFO search.HibernateSearchEventListenerRegister - Unable to find org.hibernate.search.event.FullTextIndexEventListener on the classpath. Hibernate Search is not enabled.

2021-12-24 14:42:47,356 [localhost-startStop-1] WARN mapping.RootClass - composite-id class does not override equals(): org.pih.warehouse.invoice.InvoiceList

2021-12-24 14:42:47,356 [localhost-startStop-1] WARN mapping.RootClass - composite-id class does not override hashCode(): org.pih.warehouse.invoice.InvoiceList

2021-12-24 14:42:47,703 [localhost-startStop-1] INFO connection.ConnectionProviderFactory - Initializing connection provider: org.springframework.orm.hibernate3.LocalDataSourceConnectionProvider

2021-12-24 14:42:47,703 [localhost-startStop-1] INFO cfg.SettingsFactory - RDBMS: MySQL, version: 5.5.68-MariaDB

2021-12-24 14:42:47,703 [localhost-startStop-1] INFO cfg.SettingsFactory - JDBC driver: MySQL Connector Java, version: mysql-connector-java-5.1.47 ( Revision: fe1903b1ecb4a96a917f7ed3190d80c049b1de29 )

2021-12-24 14:42:47,705 [localhost-startStop-1] INFO dialect.Dialect - Using dialect: org.hibernate.dialect.MySQL5InnoDBDialect

2021-12-24 14:42:47,707 [localhost-startStop-1] INFO transaction.TransactionFactoryFactory - Transaction strategy: org.springframework.orm.hibernate3.SpringTransactionFactory

2021-12-24 14:42:47,708 [localhost-startStop-1] INFO transaction.TransactionManagerLookupFactory - No TransactionManagerLookup configured (in JTA environment, use of read-write or transactional second-level cache is not recommended)

2021-12-24 14:42:47,708 [localhost-startStop-1] INFO cfg.SettingsFactory - Automatic flush during beforeCompletion(): disabled

2021-12-24 14:42:47,708 [localhost-startStop-1] INFO cfg.SettingsFactory - Automatic session close at end of transaction: disabled

2021-12-24 14:42:47,708 [localhost-startStop-1] INFO cfg.SettingsFactory - JDBC batch size: 15

2021-12-24 14:42:47,708 [localhost-startStop-1] INFO cfg.SettingsFactory - JDBC batch updates for versioned data: disabled

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Scrollable result sets: enabled

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - JDBC3 getGeneratedKeys(): enabled

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - JDBC result set fetch size: 25

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Connection release mode: auto

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Maximum outer join fetch depth: 2

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Default batch fetch size: 25

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Generate SQL with comments: disabled

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Order SQL updates by primary key: enabled

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Order SQL inserts for batching: enabled

2021-12-24 14:42:47,709 [localhost-startStop-1] INFO cfg.SettingsFactory - Query translator: org.hibernate.hql.ast.ASTQueryTranslatorFactory

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO ast.ASTQueryTranslatorFactory - Using ASTQueryTranslatorFactory

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - Query language substitutions: {}

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - JPA-QL strict compliance: disabled

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - Second-level cache: disabled

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - Query cache: disabled

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - Cache region factory : org.hibernate.cache.impl.NoCachingRegionFactory

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - Optimize cache for minimal puts: enabled

2021-12-24 14:42:47,711 [localhost-startStop-1] INFO cfg.SettingsFactory - Structured second-level cache entries: disabled

2021-12-24 14:42:47,714 [localhost-startStop-1] INFO cfg.SettingsFactory - Statistics: disabled

2021-12-24 14:42:47,714 [localhost-startStop-1] INFO cfg.SettingsFactory - Deleted entity synthetic identifier rollback: disabled

2021-12-24 14:42:47,726 [localhost-startStop-1] INFO cfg.SettingsFactory - Default entity-mode: pojo

2021-12-24 14:42:47,726 [localhost-startStop-1] INFO cfg.SettingsFactory - Named query checking : enabled

2021-12-24 14:42:47,789 [localhost-startStop-1] INFO impl.SessionFactoryImpl - building session factory

2021-12-24 14:42:54,617 [localhost-startStop-1] INFO util.NamingHelper - JNDI InitialContext properties:{}

2021-12-24 14:42:58,917 [localhost-startStop-1] INFO config.PropertiesFactoryBean - Loading properties file from class path resource [gsp/views.properties]

2021-12-24 14:43:02,556 [localhost-startStop-1] INFO ehcache.EhCacheManagerFactoryBean - Initializing EHCache CacheManager

2021-12-24 14:43:04,368 [localhost-startStop-1] INFO impl.StdSchedulerFactory - Using default implementation for ThreadExecutor

2021-12-24 14:43:04,370 [localhost-startStop-1] INFO simpl.SimpleThreadPool - Job execution threads will use class loader of thread: localhost-startStop-1

2021-12-24 14:43:04,379 [localhost-startStop-1] INFO core.SchedulerSignalerImpl - Initialized Scheduler Signaller of type: class org.quartz.core.SchedulerSignalerImpl

2021-12-24 14:43:04,379 [localhost-startStop-1] INFO core.QuartzScheduler - Quartz Scheduler v.2.1.6 created.

2021-12-24 14:43:04,379 [localhost-startStop-1] INFO simpl.RAMJobStore - RAMJobStore initialized.

2021-12-24 14:43:04,380 [localhost-startStop-1] INFO core.QuartzScheduler - Scheduler meta-data: Quartz Scheduler (v2.1.6) 'DefaultPluginScheduler' with instanceId 'NON\_CLUSTERED'

Scheduler class: 'org.quartz.core.QuartzScheduler' - running locally.

NOT STARTED.

Currently in standby mode.

Number of jobs executed: 0

Using thread pool 'org.quartz.simpl.SimpleThreadPool' - with 10 threads.

Using job-store 'org.quartz.simpl.RAMJobStore' - which does not support persistence. and is not clustered.

2021-12-24 14:43:04,380 [localhost-startStop-1] INFO impl.StdSchedulerFactory - Quartz scheduler 'DefaultPluginScheduler' initialized from an externally provided properties instance.

2021-12-24 14:43:04,380 [localhost-startStop-1] INFO impl.StdSchedulerFactory - Quartz scheduler version: 2.1.6

2021-12-24 14:43:04,991 [localhost-startStop-1] INFO core.QuartzScheduler - JobFactory set to: grails.plugin.quartz2.GrailsJobFactory@6dc43ee5

2021-12-24 14:43:09,690 [localhost-startStop-1] INFO support.DefaultLifecycleProcessor - Starting beans in phase 2147483647

Starting Quartz Scheduler in QuartzFactoryBean

2021-12-24 14:43:09,691 [localhost-startStop-1] INFO core.QuartzScheduler - Scheduler DefaultPluginScheduler\_$\_NON\_CLUSTERED started.

2021-12-24 14:43:11,407 [localhost-startStop-1] INFO support.DefaultListableBeanFactory - Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@32ff80ca: defining beans [org.grails.plugin.zippedresources.ZipResourceMapper,org.grails.plugin.resource.CSSRewriterResourceMapper,org.grails.plugin.cachedresources.HashAndCacheResourceMapper,org.grails.plugin.resource.BundleResourceMapperInstance,org.grails.plugin.resource.CSSRewriterResourceMapperInstance,org.grails.plugin.resource.BundleResourceMapper,org.grails.plugin.cachedresources.HashAndCacheResourceMapperInstance,org.grails.plugin.zippedresources.ZipResourceMapperInstance,org.grails.plugin.resource.CSSPreprocessorResourceMapperInstance,org.grails.plugin.resource.CSSPreprocessorResourceMapper]; parent: org.codehaus.groovy.grails.commons.spring.ReloadAwareAutowireCapableBeanFactory@3e5e369a

2021-12-24 14:43:45,068 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [shipmentItemCount] of domain class org.pih.warehouse.shipping.Shipment will not be checked during validation.

2021-12-24 14:43:45,072 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [color] of domain class org.pih.warehouse.product.Product will not be checked during validation.

2021-12-24 14:43:45,082 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [monthRequested] of domain class org.pih.warehouse.requisition.Requisition will not be checked during validation.

2021-12-24 14:43:45,082 [localhost-startStop-1] WARN commons.GrailsDomainConfigurationUtil - Derived properties may not be constrained. Property [statusSortOrder] of domain class org.pih.warehouse.requisition.Requisition will not be checked during validation.

2021-12-24 14:43:45,282 [localhost-startStop-1] INFO bootstrap.BootStrap - Running liquibase changelog(s) ...

2021-12-24 14:43:45,655 [localhost-startStop-1] INFO bootstrap.BootStrap - Liquibase running: false

2021-12-24 14:43:45,658 [localhost-startStop-1] INFO bootstrap.BootStrap - Setting default schema to openboxes

2021-12-24 14:43:45,658 [localhost-startStop-1] INFO bootstrap.BootStrap - Product Version: 5.5.68-MariaDB

2021-12-24 14:43:45,658 [localhost-startStop-1] INFO bootstrap.BootStrap - Database Version: 5.5

2021-12-24 14:43:45,659 [localhost-startStop-1] INFO liquibase - Reading from `DATABASECHANGELOG`

2021-12-24 14:43:46,049 [localhost-startStop-1] INFO liquibase - Lock Database

2021-12-24 14:43:46,062 [localhost-startStop-1] INFO liquibase - Successfully acquired change log lock

2021-12-24 14:43:54,704 [localhost-startStop-1] INFO liquibase - Changeset views/changelog.xml::1580848680306-1::jmiranda::(MD5Sum: fdb1b3291b9ec6e2d296943668e5541)

2021-12-24 14:43:54,856 [localhost-startStop-1] INFO liquibase - Release Database Lock

2021-12-24 14:43:54,857 [localhost-startStop-1] INFO liquibase - Successfully released change log lock

2021-12-24 14:43:54,859 [localhost-startStop-1] ERROR context.ContextLoader - Context initialization failed

org.springframework.beans.factory.access.BootstrapException: Error executing bootstraps; nested exception is org.codehaus.groovy.runtime.InvokerInvocationException: liquibase.exception.MigrationFailedException: Migration failed for change set views/changelog.xml::1580848680306-1::jmiranda:

Reason: liquibase.exception.JDBCException: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2:

Caused By: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2:

Caused By: View's SELECT contains a subquery in the FROM clause

at org.codehaus.groovy.grails.web.context.GrailsContextLoader.createWebApplicationContext(GrailsContextLoader.java:87)

at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:197)

at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:47)

at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:5128)

at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5653)

at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:183)

at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:1007)

at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:983)

at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:639)

at org.apache.catalina.startup.HostConfig.deployWAR(HostConfig.java:1128)

at org.apache.catalina.startup.HostConfig$DeployWar.run(HostConfig.java:2020)

at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:473)

at java.util.concurrent.FutureTask.run(FutureTask.java:262)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1152)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)

at java.lang.Thread.run(Thread.java:748)

Caused by: org.codehaus.groovy.runtime.InvokerInvocationException: liquibase.exception.MigrationFailedException: Migration failed for change set views/changelog.xml::1580848680306-1::jmiranda:

Reason: liquibase.exception.JDBCException: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2:

Caused By: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2:

Caused By: View's SELECT contains a subquery in the FROM clause

at grails.util.Environment.evaluateEnvironmentSpecificBlock(Environment.java:251)

at grails.util.Environment.executeForEnvironment(Environment.java:244)

at grails.util.Environment.executeForCurrentEnvironment(Environment.java:220)

... 5 more

Caused by: liquibase.exception.MigrationFailedException: Migration failed for change set views/changelog.xml::1580848680306-1::jmiranda:

Reason: liquibase.exception.JDBCException: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2:

Caused By: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2:

Caused By: View's SELECT contains a subquery in the FROM clause

at liquibase.ChangeSet.execute(ChangeSet.java:238)

at liquibase.parser.visitor.UpdateVisitor.visit(UpdateVisitor.java:26)

at liquibase.parser.ChangeLogIterator.run(ChangeLogIterator.java:41)

at liquibase.Liquibase.update(Liquibase.java:112)

at liquibase.Liquibase$update.call(Unknown Source)

at BootStrap$\_closure1.doCall(BootStrap.groovy:469)

... 8 more

Caused by: liquibase.exception.JDBCException: Error executing SQL CREATE OR REPLACE VIEW product\_demand AS

SELECT

-- request data

request\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

-- request item data

request\_item\_id,

product\_id,

product\_code,

product\_name,

quantity\_requested,

quantity\_canceled,

quantity\_approved,

quantity\_modified,

quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

request\_item\_status,

reason\_code\_classification,

-- pick list item data

picklist\_reason\_code,

-- reason code

COALESCE(cancel\_reason\_code, pick\_reason\_code, picklist\_reason\_code) AS reason\_code,

-- request item status

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN 'ITEM\_SUBSTITUTED'

WHEN quantity\_requested = quantity\_canceled AND quantity\_picked = 0

THEN 'ITEM\_CANCELED'

WHEN quantity\_requested = quantity\_approved AND quantity\_picked != quantity\_approved

THEN 'PICK\_MODIFIED'

WHEN quantity\_requested != quantity\_approved AND quantity\_picked = quantity\_approved

THEN 'ITEM\_MODIFIED'

WHEN quantity\_picked > 0 THEN 'ITEM\_ISSUED'

ELSE NULL

END as request\_item\_change\_status,

-- quantity demand

CASE

WHEN request\_item\_status = 'SUBSTITUTION'

THEN quantity\_requested

WHEN quantity\_picked > quantity\_requested

THEN quantity\_picked

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN (NULL, 'INSUFFICIENT\_QUANTITY\_AVAILABLE')

THEN quantity\_requested

WHEN quantity\_picked < quantity\_requested AND

reason\_code\_classification IN ('CLINICAL\_JUDGMENT')

THEN quantity\_picked

ELSE quantity\_requested

END as quantity\_demand

FROM (

SELECT request\_id,

request\_item\_id,

request\_status,

request\_number,

date\_created,

date\_requested,

date\_issued,

origin\_id,

origin\_name,

destination\_id,

destination\_name,

product\_id,

product\_code,

product\_name,

quantity\_requested,

IFNULL(quantity\_canceled, 0) AS quantity\_canceled,

IFNULL(quantity\_approved, 0) AS quantity\_approved,

IFNULL(quantity\_modified, 0) AS quantity\_modified,

IFNULL(quantity\_picked, 0) AS quantity\_picked,

cancel\_reason\_code,

cancel\_comments,

pick\_reason\_code,

reason\_code\_classification,

request\_item\_status,

picklist\_reason\_code

FROM (

SELECT requisition.id as request\_id,

requisition\_item.id as request\_item\_id,

requisition.status AS request\_status,

requisition.request\_number,

requisition.date\_created as date\_created,

requisition.date\_requested AS date\_requested,

requisition\_transaction.date\_issued AS date\_issued,

origin.id AS origin\_id,

origin.name AS origin\_name,

destination.id AS destination\_id,

destination.name AS destination\_name,

child\_request\_item.status as request\_item\_status,

product.id AS product\_id,

product.product\_code,

product.name AS product\_name,

cancel\_reason\_code,

-- FIXME converting from boolean to string does not strike me as a good idea here

CASE

WHEN insufficient\_quantity\_available > 0

THEN 'INSUFFICIENT\_QUANTITY\_AVAILABLE'

WHEN clinical\_judgment > 0 THEN 'CLINICAL\_JUDGMENT'

ELSE NULL

END as reason\_code\_classification,

insufficient\_quantity\_available,

clinical\_judgment,

cancel\_comments,

pick\_reason\_code,

quantity AS quantity\_requested,

quantity\_canceled,

quantity\_approved,

-- quantity changed

(

SELECT SUM(quantity)

FROM requisition\_item child

WHERE child.parent\_requisition\_item\_id = requisition\_item.id

AND requisition\_item\_type = 'QUANTITY\_CHANGE'

) AS quantity\_modified,

-- quantity picked

picked\_items.quantity\_picked,

-- picklist item reason code

picked\_items.picklist\_reason\_code as picklist\_reason\_code

FROM requisition\_item

JOIN product ON product.id = requisition\_item.product\_id

JOIN requisition ON requisition.id = requisition\_item.requisition\_id

JOIN location destination ON destination.id = requisition.destination\_id

JOIN location origin ON origin.id = requisition.origin\_id

JOIN location\_type on origin.location\_type\_id = location\_type.id

-- Used to get issue date

LEFT OUTER JOIN (

SELECT requisition.id as requisition\_id,

transaction.transaction\_date as date\_issued

FROM requisition

JOIN shipment on shipment.requisition\_id = requisition.id

JOIN transaction on shipment.id = transaction.outgoing\_shipment\_id

) as requisition\_transaction

on requisition.id = requisition\_transaction.requisition\_id

-- Used to get quantity picked

LEFT OUTER JOIN (

SELECT ifnull(requisition\_item.parent\_requisition\_item\_id,

requisition\_item.id) as requisition\_item\_id,

SUM(picklist\_item.quantity) as quantity\_picked,

picklist\_item.reason\_code as picklist\_reason\_code

FROM picklist\_item

join requisition\_item

on picklist\_item.requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id, picklist\_item.reason\_code

) as picked\_items ON (picked\_items.requisition\_item\_id = requisition\_item.id)

-- Used to determine whether a requisition item was changed or substituted

LEFT OUTER JOIN (

SELECT requisition\_item.id as requisition\_item\_id,

GROUP\_CONCAT(distinct (child.product\_id)) as products,

IFNULL(GROUP\_CONCAT(distinct (child.requisition\_item\_type)),

requisition\_item.requisition\_item\_type) as status

FROM requisition\_item

LEFT OUTER JOIN requisition\_item child

on child.parent\_requisition\_item\_id = requisition\_item.id

GROUP BY requisition\_item.id

) as child\_request\_item ON requisition\_item.id =

child\_request\_item.requisition\_item\_id

-- Used to classify whether a request item was changed or canceled due to reason code related to insufficient quantity available

LEFT OUTER JOIN (

select parent\_item.id as requisition\_item\_id,

-- check all reason codes for insufficient quantity available

sum(if(parent\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.cancel\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_item.pick\_reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or parent\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER')

or child\_picklist\_item.reason\_code in

('STOCKOUT', 'LOW\_STOCK', 'COULD\_NOT\_LOCATE', 'BACKORDER'), true,

false)) as insufficient\_quantity\_available,

-- otherwise the reason code is probably related to a clinical judgment

sum((coalesce(parent\_item.cancel\_reason\_code,

child\_item.cancel\_reason\_code,

parent\_item.pick\_reason\_code,

child\_item.pick\_reason\_code,

parent\_picklist\_item.reason\_code,

child\_picklist\_item.reason\_code) !=

'')) as clinical\_judgment

from requisition\_item parent\_item

left outer join requisition\_item child\_item

on child\_item.parent\_requisition\_item\_id = parent\_item.id

left outer join picklist\_item parent\_picklist\_item

on parent\_item.id = parent\_picklist\_item.requisition\_item\_id

left outer join picklist\_item child\_picklist\_item

on child\_item.id = child\_picklist\_item.requisition\_item\_id

group by parent\_item.id

) as reason\_codes on requisition\_item.id = reason\_codes.requisition\_item\_id

WHERE requisition\_item.requisition\_item\_type = 'ORIGINAL'

AND location\_type.location\_type\_code = 'DEPOT'

AND requisition.status = 'ISSUED'

) AS tmp1

) AS tmp2

at liquibase.database.template.JdbcTemplate.execute(JdbcTemplate.java:55)

at liquibase.database.template.JdbcTemplate.execute(JdbcTemplate.java:86)

at liquibase.change.AbstractChange.execute(AbstractChange.java:247)

at liquibase.change.AbstractChange.executeStatements(AbstractChange.java:93)

at liquibase.ChangeSet.execute(ChangeSet.java:214)

... 13 more

Caused by: java.sql.SQLException: View's SELECT contains a subquery in the FROM clause

at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:965)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:3978)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:3914)

at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:2530)

at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2683)

at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2491)

at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2449)

at com.mysql.jdbc.StatementImpl.executeInternal(StatementImpl.java:845)

at com.mysql.jdbc.StatementImpl.execute(StatementImpl.java:745)

at com.mchange.v2.c3p0.impl.NewProxyStatement.execute(NewProxyStatement.java:75)

at liquibase.database.template.JdbcTemplate$1ExecuteStatementCallback.doInStatement(JdbcTemplate.java:78)

at liquibase.database.template.JdbcTemplate.execute(JdbcTemplate.java:48)

... 17 more

Dec 24, 2021 2:43:54 PM org.apache.catalina.core.StandardContext startInternal

SEVERE: One or more listeners failed to start. Full details will be found in the appropriate container log file

Dec 24, 2021 2:43:54 PM org.apache.catalina.core.StandardContext startInternal

SEVERE: Context [/openboxes] startup failed due to previous errors

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesJdbc

SEVERE: The web application [/openboxes] registered the JDBC driver [com.mysql.jdbc.Driver] but failed to unregister it when the web application was stopped. To prevent a memory leak, the JDBC Driver has been forcibly unregistered.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [C3P0PooledConnectionPoolManager[identityToken->1n9yufmal1to4g411ibeir7|5f4ec71b]-AdminTaskTimer] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [C3P0PooledConnectionPoolManager[identityToken->1n9yufmal1to4g411ibeir7|5f4ec71b]-HelperThread-#0] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [C3P0PooledConnectionPoolManager[identityToken->1n9yufmal1to4g411ibeir7|5f4ec71b]-HelperThread-#1] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [C3P0PooledConnectionPoolManager[identityToken->1n9yufmal1to4g411ibeir7|5f4ec71b]-HelperThread-#2] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [C3P0PooledConnectionPoolManager[identityToken->1n9yufmal1to4g411ibeir7|5f4ec71b]-DeferredStatementDestroyerThread-#0] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Abandoned connection cleanup thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [net.sf.ehcache.CacheManager@2f2addc8] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store dashboardCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store messageCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store dashboardTotalStockValueCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store dashboardProductSummaryCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store fastMoversCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store quantityOnHandCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store binLocationSummaryCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store inventorySnapshotCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store megamenuCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store selectTagsCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-1] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-2] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-3] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-4] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-5] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-6] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-7] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-8] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-9] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_Worker-10] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store binLocationReportCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store selectCatalogsCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store inventoryBrowserCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store selectTagCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store selectCategoryCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Store dashboardGenericProductSummaryCache Spool Thread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [Timer-0] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.loader.WebappClassLoaderBase clearReferencesThreads

SEVERE: The web application [/openboxes] appears to have started a thread named [DefaultPluginScheduler\_QuartzSchedulerThread] but has failed to stop it. This is very likely to create a memory leak.

Dec 24, 2021 2:43:54 PM org.apache.catalina.startup.HostConfig deployWAR

INFO: Deployment of web application archive [/opt/tomcat/webapps/openboxes.war] has finished in [122,589] ms

Dec 24, 2021 2:43:54 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deploying web application directory [/opt/tomcat/webapps/ROOT]

Dec 24, 2021 2:43:54 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deployment of web application directory [/opt/tomcat/webapps/ROOT] has finished in [35] ms

Dec 24, 2021 2:43:54 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deploying web application directory [/opt/tomcat/webapps/docs]

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deployment of web application directory [/opt/tomcat/webapps/docs] has finished in [75] ms

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deploying web application directory [/opt/tomcat/webapps/examples]

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deployment of web application directory [/opt/tomcat/webapps/examples] has finished in [304] ms

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deploying web application directory [/opt/tomcat/webapps/host-manager]

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deployment of web application directory [/opt/tomcat/webapps/host-manager] has finished in [246] ms

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deploying web application directory [/opt/tomcat/webapps/manager]

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deployment of web application directory [/opt/tomcat/webapps/manager] has finished in [35] ms

Dec 24, 2021 2:43:55 PM org.apache.coyote.AbstractProtocol start

INFO: Starting ProtocolHandler ["http-bio-10.0.0.2-8085"]

Dec 24, 2021 2:43:55 PM org.apache.catalina.startup.Catalina start

INFO: Server startup in 123386 ms

Dec 24, 2021 2:43:58 PM org.apache.catalina.loader.WebappClassLoaderBase findResourceInternal

INFO: Illegal access: this web application instance has been stopped already. Could not load []. The following stack trace is thrown for debugging purposes as well as to attempt to terminate the thread which caused the illegal access.

Dec 24, 2021 2:44:00 PM org.apache.catalina.loader.WebappClassLoaderBase loadClass

INFO: Illegal access: this web application instance has been stopped already. Could not load [org.quartz.ExecuteInJTATransaction]. The following stack trace is thrown for debugging purposes as well as to attempt to terminate the thread which caused the illegal access.

java.lang.IllegalStateException

at org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1836)

at org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1795)

at org.quartz.ee.jta.JTAAnnotationAwareJobRunShellFactory.createJobRunShell(JTAAnnotationAwareJobRunShellFactory.java:100)

at org.quartz.core.QuartzSchedulerThread.run(QuartzSchedulerThread.java:386)

Exception in thread "DefaultPluginScheduler\_QuartzSchedulerThread" java.lang.NoClassDefFoundError: org/quartz/ExecuteInJTATransaction

at org.quartz.ee.jta.JTAAnnotationAwareJobRunShellFactory.createJobRunShell(JTAAnnotationAwareJobRunShellFactory.java:100)

at org.quartz.core.QuartzSchedulerThread.run(QuartzSchedulerThread.java:386)

Caused by: java.lang.ClassNotFoundException: org.quartz.ExecuteInJTATransaction

at org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1951)

at org.apache.catalina.loader.WebappClassLoaderBase.loadClass(WebappClassLoaderBase.java:1795)

... 2 more