

FAQ - Why am I receiving "Cannot insert duplicate key row..." error?

Question

Why am I receiving "Cannot insert duplicate key row..." error?

Answer

This error can be encountered when adding or updating a timesheet row to have the same identifying values as another row within one edit session (without saving in between). The "new" row on the timesheet is compared to an "old" existing row on the database and gets this error.

For example: You have a row on your timesheet charging to Project A, and you modify its value to Project B. Then - without hitting Save - you either add a new row or change another row to have a value of Project A. When you Save, the system may attempt to process the second record before the first record has been updated. This is why the system may think you have two Project A entries on your timesheet. Note that "Project A" refers to the entire set of key values for a timesheet row, which may include more fields such as project, task, pay code, project type, labor category, location, etc.

Encountering this message does not impact the integrity of the data saved in your system, but simply prevents the saving of that set of updates on the screen.

In order to avoid this situation, save the first update to the line with Project B, then make the update to the line with Project A and save that change. This sequence of events should not trigger the Unique Constraint error.

If the Unique Constraint error is triggered, there are a few options to resolve:

- Try to delete one of the rows and Save.
- Restore the two rows to their original value and Save.
- Log out, close the browser, then log in again to edit the timesheet; it should retain its original entries because the duplicates were not saved.

Other possible causes of this error include:

1. Swapping project or task entries on the timesheet.
2. Modifying an assignment, which causes timesheet changes during rerating (the rerate can leave some time outside of an assignment, and thus can cause it to default to a new value for labor category).
3. Modifying the list of master or project labor categories, such that the update results in a new first entry in the list.
 - a. The first labor category in the list is sometimes chosen in situations where a labor category is required but none is provided on an assignment and there is no default labor category (or the default is not on the list of valid project labor categories).
 - b. Modifying the list may change the labor category to be the same as another row on the timesheet, thus causing the duplicate. *The default Labor Category changes + there is time using the current default value + the new default Labor Category is already on the timesheet = duplicate row error.*
4. Changing the **Admin > Properties** setting to *Hide Labor Category On Timesheet Edit*, after two rows with different labor categories have already been saved.
 - a. In these cases, the next time the timesheet is saved (and the system attempts to rerate the entries on the timesheet) the labor category defaulting logic kicks in and both rows may now pick up the current labor category (from the assignment or default labor category).
 - b. The key here is that when the labor category is shown on the timesheet, the system requires that one be selected instead of attempting to default a value. However, when labor category is hidden on the timesheet, the system will attempt to derive a default value and then attempt to stamp multiple rows with the same labor category.
5. Updating time via a time import or an assignment import.
 - a. For example, situations in which prepopulated holidays are set up and then timesheets with holiday rows in the file are imported.

Sample error:

SQL Exception – Cannot insert duplicate key row in object 'person_time_data' with unique index 'idx_person_time_data1'.

There was a SQLException while processing /unanet/action/projects/assignment/save.

SQLException caught in com.unanet.servlet.ActionServlet.

```
java.sql.SQLException: Cannot insert duplicate key row in object 'person_time_data' with unique index
'idx_person_time_data1'.
    at net.sourceforge.jtds.jdbc.SQLDiagnostic.addDiagnostic(SQLDiagnostic.java:368)
    at net.sourceforge.jtds.jdbc.TdsCore.tdsErrorToken(TdsCore.java:2816)
    at net.sourceforge.jtds.jdbc.TdsCore.nextToken(TdsCore.java:2254)
    at net.sourceforge.jtds.jdbc.TdsCore.getMoreResults(TdsCore.java:631)
    at net.sourceforge.jtds.jdbc.JtdsStatement.processResults(JtdsStatement.java:584)
    at net.sourceforge.jtds.jdbc.JtdsStatement.executeSQL(JtdsStatement.java:546)
    at net.sourceforge.jtds.jdbc.JtdsPreparedStatement.executeUpdate(JtdsPreparedStatement.java:505)
    at com.unanet.data.factory.TimeslipFactory.update(TimeslipFactory.java:649)
    at com.unanet.data.factory.TimeslipFactory.updateTimeslips(TimeslipFactory.java:1123)
    at com.unanet.data.factory.TimeslipFactory.updateOverrides(TimeslipFactory.java:1092)
    at com.unanet.data.factory.ProjectAssignmentFactory.update(ProjectAssignmentFactory.java:812)
    at com.unanet.data.factory.ProjectAssignmentFactory.save(ProjectAssignmentFactory.java:632)
    at com.unanet.page.projects.AssignmentSaveAction.service(AssignmentSaveAction.java:96)
    at com.unanet.page.Action.service(Action.java:300)
    at com.unanet.servlet.ActionServlet.doService(ActionServlet.java:139)
    at com.unanet.servlet.ActionServlet.doPost(ActionServlet.java:25)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
    at jrun.servlet.FilterChain.doFilter(FilterChain.java:86)
    at com.unanet.servlet.GzipFilter.doFilter(GzipFilter.java:44)
    at jrun.servlet.FilterChain.doFilter(FilterChain.java:94)
    at com.unanet.servlet.AccessLogFilter.doFilter(AccessLogFilter.java:82)
    at jrun.servlet.FilterChain.doFilter(FilterChain.java:94)
    at jrun.servlet.FilterChain.service(FilterChain.java:101)
    at jrun.servlet.ServletInvoker.invoke(ServletInvoker.java:91)
    at jrun.servlet.JRunInvokerChain.invokeNext(JRunInvokerChain.java:42)
    at jrun.servlet.JRunRequestDispatcher.invoke(JRunRequestDispatcher.java:259)
    at jrun.servlet.ServletEngineService.dispatch(ServletEngineService.java:541)
    at jrun.servlet.jrpp.JRunProxyService.invokeRunnable(JRunProxyService.java:204)
    at jrunx.scheduler.ThreadPool$DownstreamMetrics.invokeRunnable(ThreadPool.java:320)
    at jrunx.scheduler.ThreadPool$ThreadThrottle.invokeRunnable(ThreadPool.java:428)
    at jrunx.scheduler.ThreadPool$UpstreamMetrics.invokeRunnable(ThreadPool.java:266)
    at jrunx.scheduler.WorkerThread.run(WorkerThread.java:66)
```