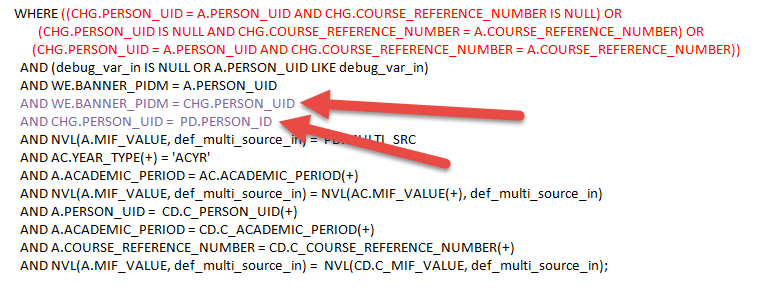
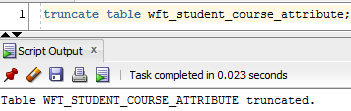
Proving out that these joins are illogical and are causing records to be deleted from the table wft\_student\_course\_attribute, and those records are NOT re-added in cases where they should be.

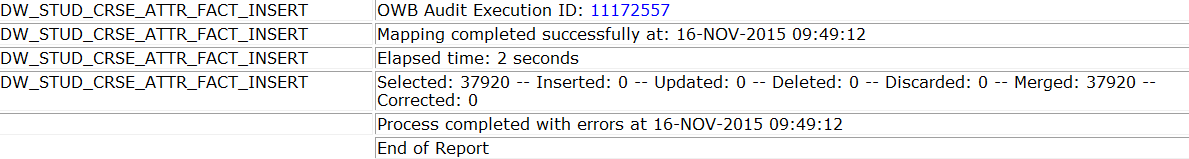
My suspected illogical joins are within the refresh cursor within edw\_student\_extr.f\_get\_student\_course\_attr (the purple joins directly contradict the fact that in the red OR clause, we have shown that person\_uid can be null)



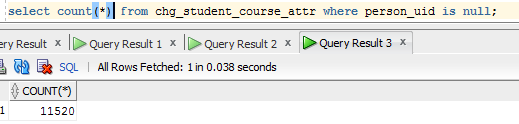
**Truncated Fact table:**



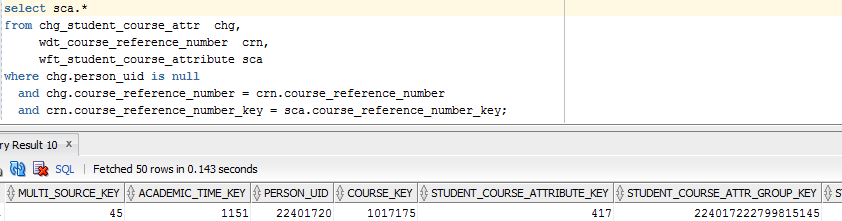
**Loaded up the student course attribute star:**



**Proving out that records with NULL person\_uid exist in the change table:**

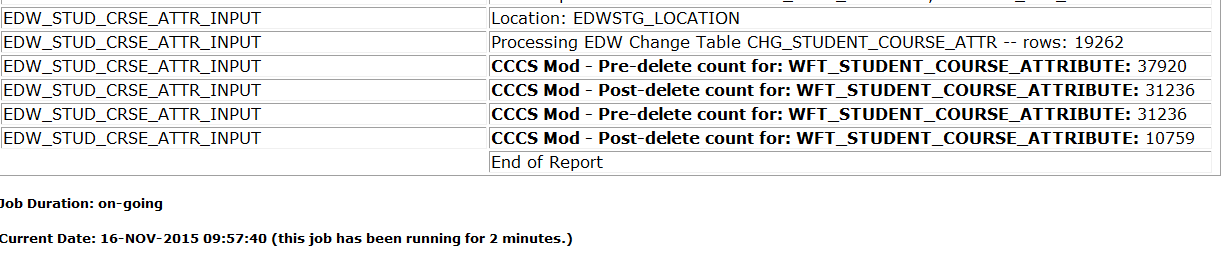


**We’ll pick on one of these records which have an entry in the fact table:**

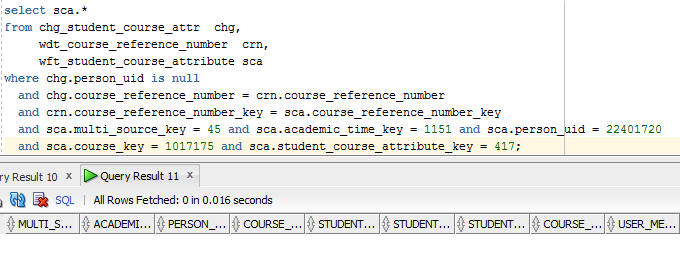


**Now I kick off the student course attribute refresh.**

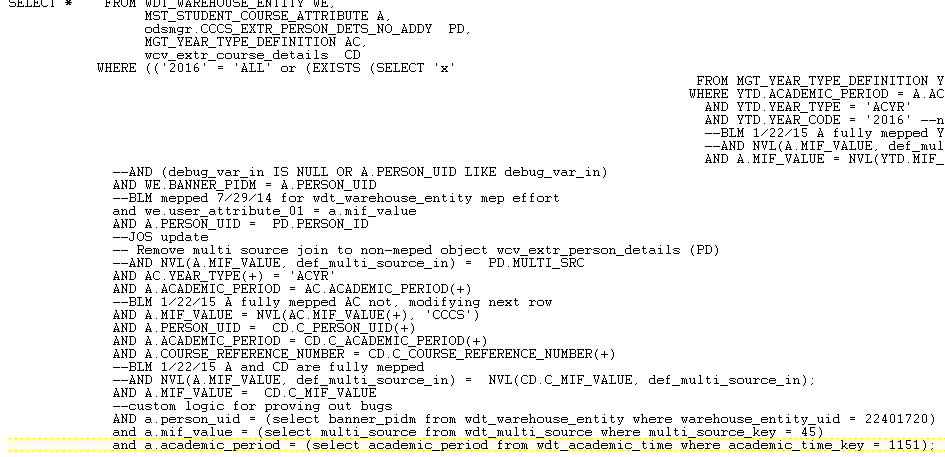
**The process is currently doing an insert to the INPUT table, but the deletes have already happened, we know this as we added custom counts to the control report which display after the delete statements execute:**

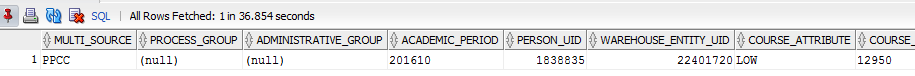


**After the deletes were complete, but before the fact inserts, I searched for the record posted above to confirm it was deleted:**

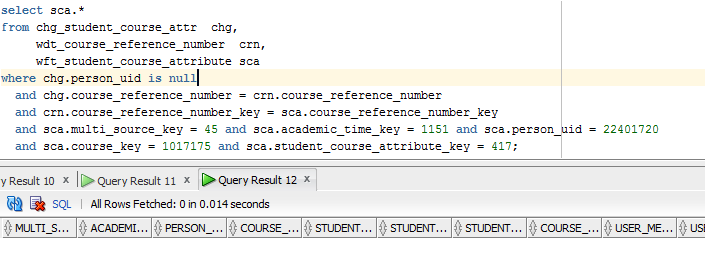


**Ok, we’ve confirmed it’s gone. Let’s confirm that that record would still be loaded if we re-loaded the star (as opoposed to refreshing). Singling out the table function (for the load) for this single record:**



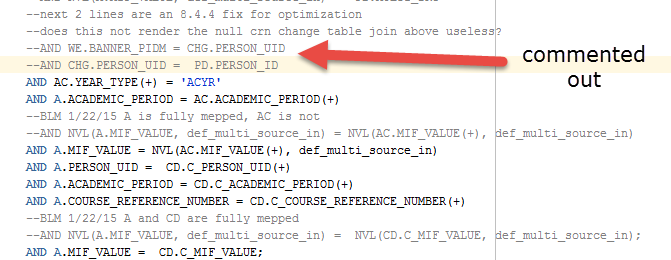


**Now that we’ve proven the record exists in the ODS, should be in the EDW, and has a course reference number with a change table record with a NULL person UID, we know logically the refresh should insert the record again. With the refresh mapping completed, we’ll look for the record again to see if it inserted:**

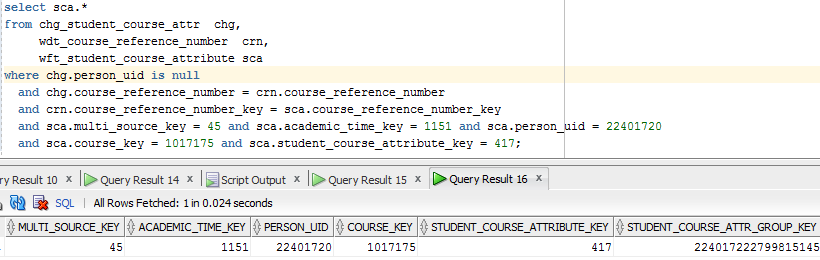


**Nope, it’s gone.**

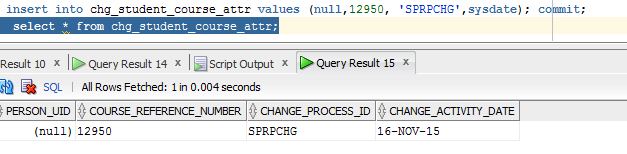
**NOW – I commented out those 2 joins (which were in red atop this doc) in the refresh logic and tried the entire process again. The next screen shot is the refresh block with the illogical joins commented out:**



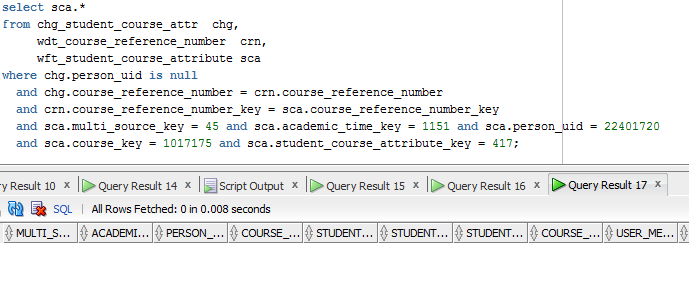
**Loaded the entire star – is that record there now?:**



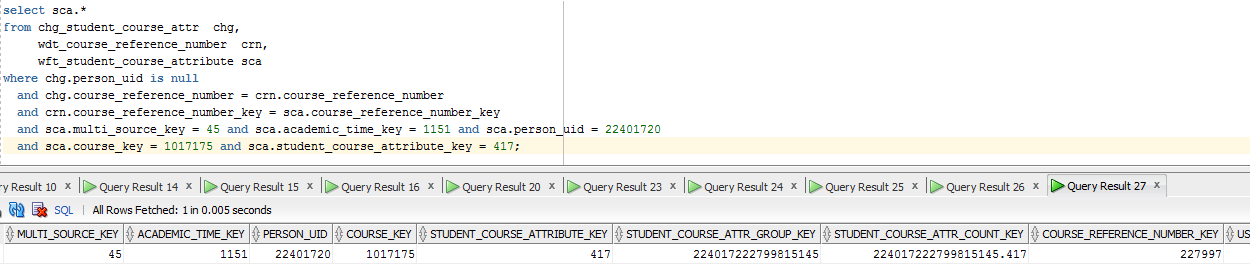
**Ok – it’s there. Made sure that record was in the change table again (I manually put it back in there):**



**Ran the refresh – the record should be deleted after the INPUT table is populated:**



**Yes, it deleted. Now the fact table insert should add it back since we removed the illogical joins:**



**Yup! It’s back. Those joins are bad.**

Bottom line: Those joins are illogical and are causing fact records to be deleted and they are not re-added when they should be. I believe this was done to improve efficiency of the refresh due to audit trail notes (though I disagree that, even when logical, that they would actually improve performance). Rather than add an illogical join that causes you to lose data, my suggestion if people have poor performance here would be to add the academic period to the change table chg\_student\_course\_attribute. We did this to the student course star and it worked flawlessly; doesn’t take much effort, you just:

1. Add academic\_period varchar2(6) to chg\_student\_course\_attribute
2. Modify odsmgr.maintain\_edw\_change\_records to change all references on said change table to insert a term, and add the correct joins
3. Modify the change table condition in edw\_student\_extr.f\_get\_student\_course\_attr to join on term
4. Modify edwstg.maintain\_edw\_error\_records procedure, add a CASE block for this specific change table to adjust the join logic
5. Modify edw\_general\_extr.p\_delete\_fact\_records, add a block to the CASE statement to join on term for this change table