**First pass thru this document (steps 1-9) was completed in January 2018 by Gerry Mulligan and Larry Scott.**

**The ERS implementation priority was changed from UCSD to UCSB. This document (as is) was cloned and used as a starting point for the UCSB migration document. When ERS UCSD is implemented, compare this document to the ERS UCSB document to ensure completeness.**

**TO DO: Notice steps 7B, 7C, 7D deal with off quarter reporting. UCSD does not have off quarter reporting periods. 7B, 7C, and 7D should be adjusted accordingly.**

**Version Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.1 | Feb 6 2018 | G Mulligan | Initial document – Copied from UCLA migration document and updated for DB2 |
|  |  |  | **Values in Step 7K need to be double-checked** |
| 1.2 | Feb 7, 2018 | LScott | Add step 7L - ‘Over The Cap’ and ‘Salary Cap’ to ERSEarnings table |

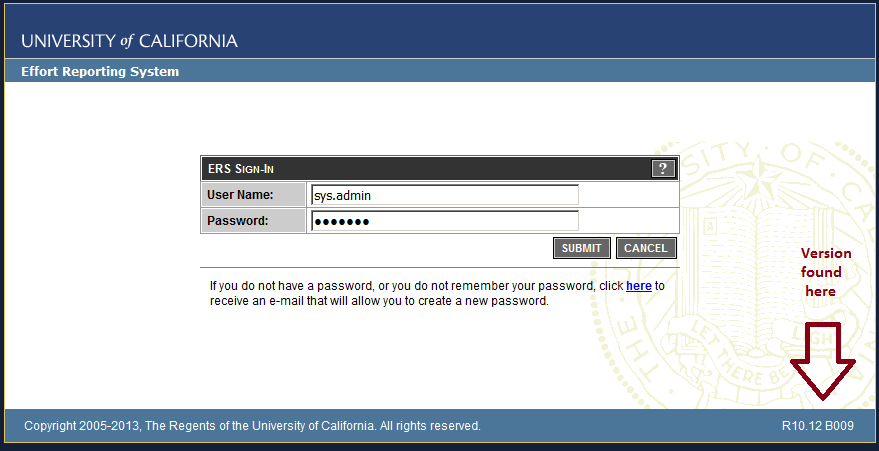
This document provides step-by-step instructions to upgrade a DB2 ERS 10.12 B009 database to ERS 11.0 format. Due to dependencies, this document must be executed in the order presented. See related documents for IBM DB2 databases and Oracle databases.

**Recommendation:**  Execute these database SQL commands individually and verify that each statement completes successfully before proceeding to the next statement. This is a longer, but more thorough approach. If statements are executed in blocks, it is easy to miss an error which may have a cascading (bad) effect as more SQL statements are executed.

**Step 1: UCSD Database Upgrade Prerequisite**

Before starting this database upgrade process, make sure your ERS instance is running at release level

ERS 10.12 B009 before proceeding with the following steps.



If needed, the UCOP ERS program support staff can provide a custom ERS upgrade script (eliminating the need to perform all interim releases) to get your database to ERS 10.12 B009 format.

**Modified script, Converted script (DB2 - UCSD) - queried all keys on the table named in each statement below. See p:\PROJECTS\ers\sql use in dbvisualizer\** **Select all from SYSCAT.TABCONST 05.txt**

select distinct TC.tabname, TC.constname, TC.type, SKCU.colname

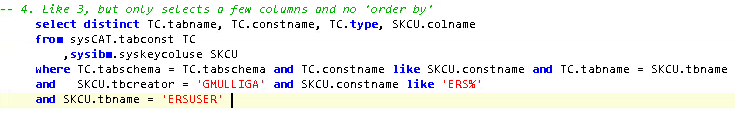
from sysCAT.tabconst TC

,sysibm.syskeycoluse SKCU

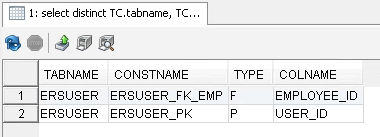
where TC.tabschema = TC.tabschema and TC.constname like SKCU.constname and TC.tabname = SKCU.tbname

and SKCU.tbcreator = 'GMULLIGA' and SKCU.constname like 'ERS%'

and SKCU.tbname = 'ERSUSER'



**For example, running the above sql for the first ALTER statement returns this:**



**So I changed the constraint name in the first ALTER to ERSUser\_FK\_EMP**

Original SQL Server: ALTER TABLE ERSUser DROP CONSTRAINT ERSUser\_FK;

Modified for Db2: ALTER TABLE ERSUSER DROP CONSTRAINT ERSUser\_FK\_EMP;

**Step 2: Employee Update from char(9) to varchar(11)**

-- ALTER TABLE ERSUser DROP CONSTRAINT ERSUser\_FK;

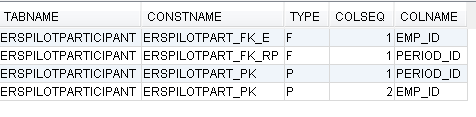
ALTER TABLE ERSUSER DROP CONSTRAINT ERSUser\_FK\_EMP;



-- ALTER TABLE ERSPILOTPARTICIPANT DROP CONSTRAINT ERSPilotPart\_FK\_Emp;

ALTER TABLE ERSPILOTPARTICIPANT DROP CONSTRAINT ERSPilotPart\_FK\_E;

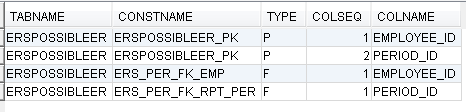
ERSPILOTPARTICIPANT also has a foreign key ERSPILOTPART\_FK\_RP on PERIOD\_ID



-- ALTER TABLE ERSPossibleER DROP CONSTRAINT ERSPossibleER\_FK;

ALTER TABLE ERSPOSSIBLEER DROP CONSTRAINT ERS\_PER\_FK\_EMP;

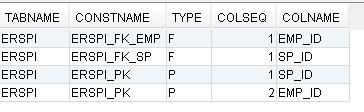
ERSPOSSIBLEER also has a foreign key ERS\_PER\_FK\_RPT\_PER on PERIOD\_ID



-- ALTER TABLE ERSPI DROP CONSTRAINT ERSPI\_FK\_EMP;

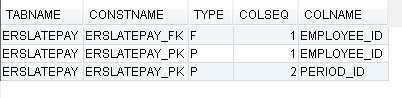
ALTER TABLE ERSPI DROP CONSTRAINT ERSPI\_FK\_EMP;

ERSPI also has a foreign key ERSPI\_FK\_SP on SP\_ID



-- ALTER TABLE ERSLatePay DROP CONSTRAINT ERSLatePay\_FK;

ALTER TABLE ERSLATEPAY DROP CONSTRAINT ERSLatePay\_FK;

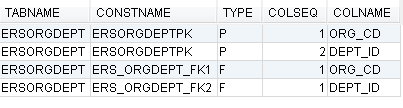


Dept&&

-- ALTER TABLE ERSOrgDept DROP CONSTRAINT ERSOrgDept\_FK1

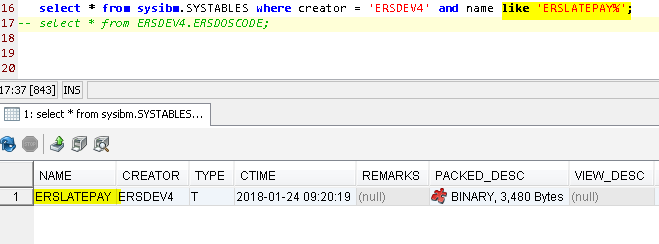
ALTER TABLE ERSORGDEPT DROP CONSTRAINT ERS\_ORGDEPT\_FK1

ERSORGDEPT also has a foreign key ERS\_ORGDEPT\_FK2 on DEPT\_ID



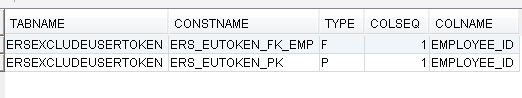
ALTER TABLE ERSLatePay\_Temp DROP CONSTRAINT ERSLatePay\_TEMP\_FK; -- (might not exist)

Does not exist in UCSB



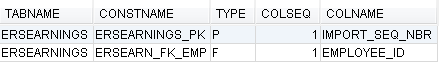
-- ALTER TABLE ERSExcludeUserToken DROP CONSTRAINT ERSExcludeUserToken\_FK;

ALTER TABLE ERSEXCLUDEUSERTOKEN DROP CONSTRAINT ERS\_EUTOKEN\_FK\_EMP;



-- ALTER TABLE ERSEarnings DROP CONSTRAINT ERSEarnings\_FK;

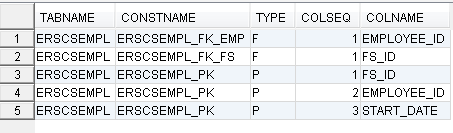
ALTER TABLE ERSEARNINGS DROP CONSTRAINT ERSEARN\_FK\_EMP;



-- ALTER TABLE ERSCSEmpl DROP CONSTRAINT ERSCSEmpl\_FK1;

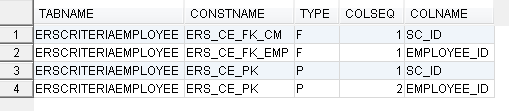
ALTER TABLE ERSCSEMPL DROP CONSTRAINT ERSCSEMPL\_FK\_EMP;

ERSCSEMPL also has a foreign key ERSCSEMPL\_FK\_FS on FS\_ID



-- ALTER TABLE ERSCriteriaEmployee DROP CONSTRAINT ERSCriteriaEmployee\_FK;

ALTER TABLE ERSCRITERIAEMPLOYEE DROP CONSTRAINT ERS\_CE\_FK\_EMP;

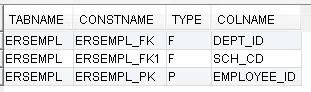


-- ALTER TABLE ERSEmpl DROP CONSTRAINT ERSEmpl\_PK;

ALTER TABLE ERSEMPL DROP CONSTRAINT ERSEmpl\_PK;

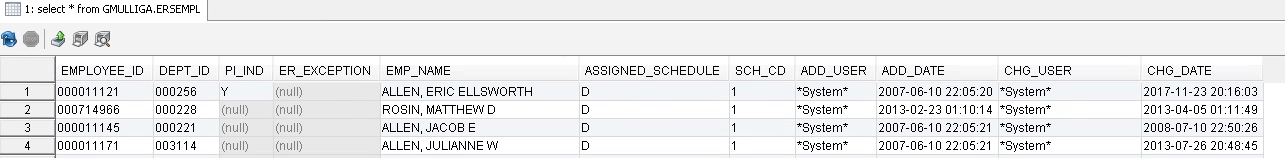
ERSEMPL also has a foreign key ERSEMPL\_FK on DEPT\_ID

ERSEMPL also has a foreign key ERSEMPL\_FK1 on SCH\_ID



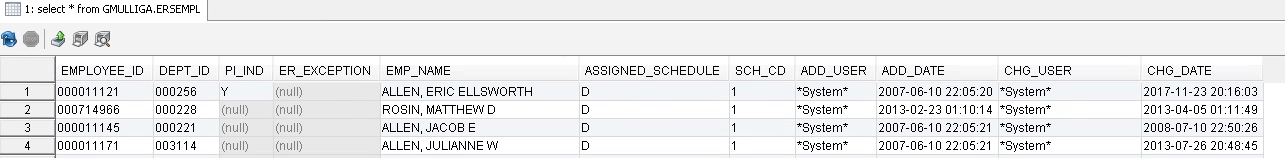
-- ALTER TABLE ERSEmpl ADD PPS\_EMPLOYEE\_ID CHAR(9);

ALTER TABLE ERSEMPL ADD PPS\_EMPLOYEE\_ID CHAR(9);



-- UPDATE ERSEmpl set pps\_employee\_id = employee\_id;

UPDATE ERSEMPL set pps\_employee\_id = employee\_id;



-- ALTER TABLE ERSEmpl ALTER COLUMN PPS\_EMPLOYEE\_ID CHAR(9) NOT NULL;

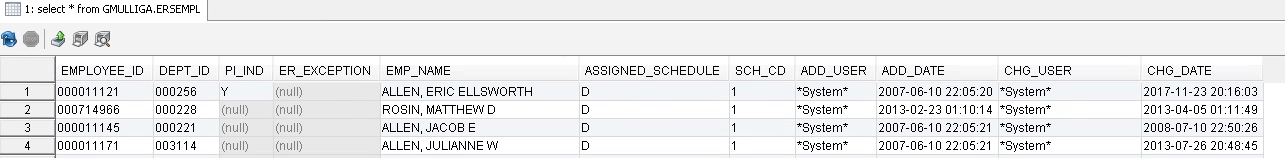
ALTER TABLE ERSEMPL ALTER COLUMN PPS\_EMPLOYEE\_ID SET DATA TYPE CHAR(9) NOT NULL;

- first had -104, then added 'SET DATA TYPE' but it will still fail cause of NOT NULL.

Must run a second command to SET NOT NULL like this:

ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

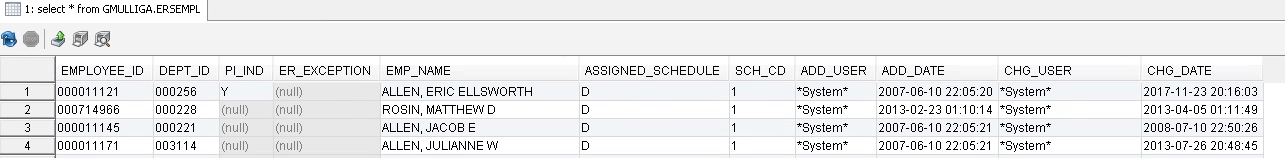
ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;



-- ALTER TABLE ERSEmpl ALTER COLUMN EMP\_NAME varchar(120);

ALTER TABLE ERSEMPL ALTER COLUMN EMP\_NAME SET DATA TYPE varchar(120);

- first had -104, then added 'SET DATA TYPE' to make it work.



-- ALTER TABLE ERSEmpl ALTER COLUMN EMPLOYEE\_ID varchar(11) not null;

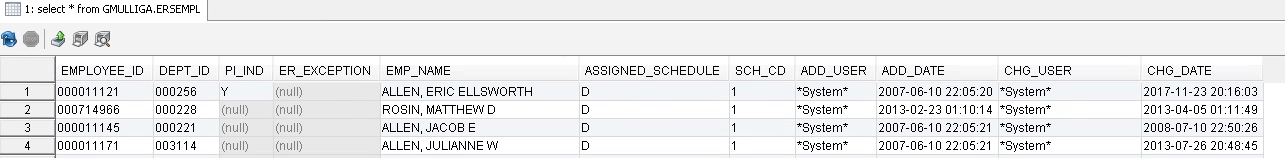
ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11) not null;

- first had -104, then added 'SET DATA TYPE' but it will still fail cause of NOT NULL.

Must run a second command to SET NOT NULL like this:

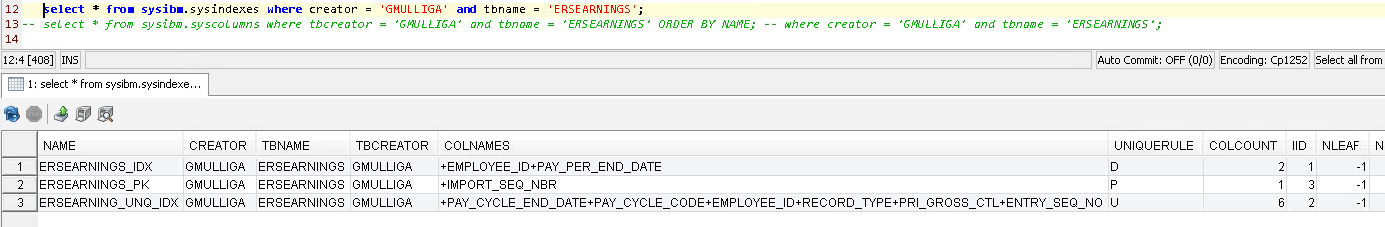
ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;



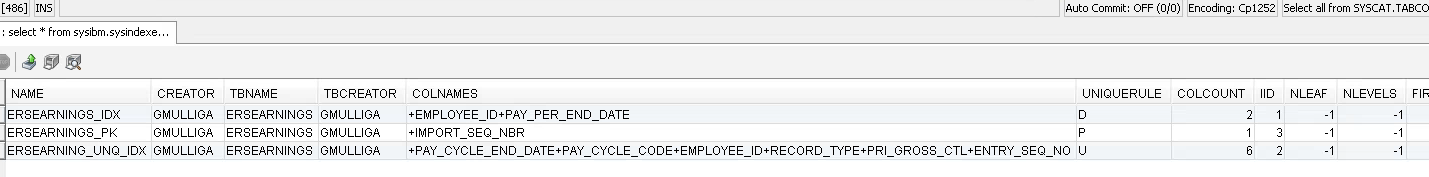
-- DROP INDEX ERSEarnings.ERSEARNINGS\_IDX

DROP INDEX ERSEARNINGS\_IDX

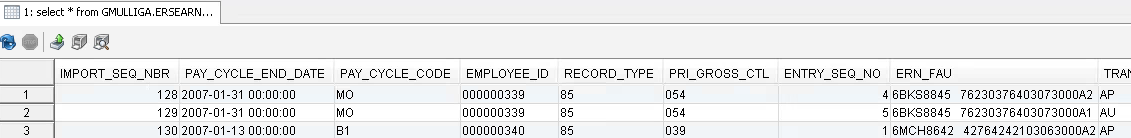


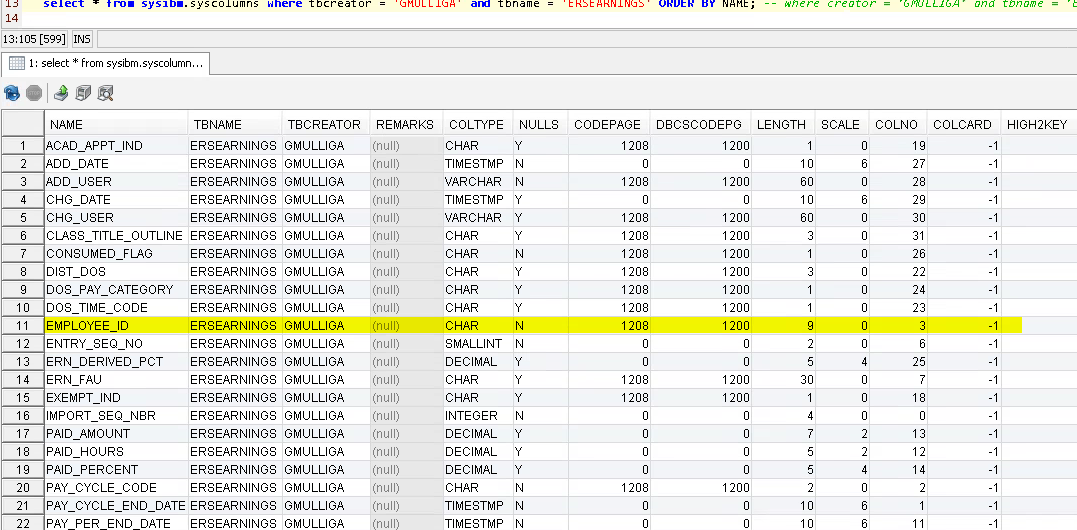
-- DROP INDEX ERSEarnings. ERSEARNINGS\_UNIQUE\_IDX

DROP INDEX ERSEARNING\_UNQ\_IDX



-- ALTER TABLE ERSEarnings ALTER COLUMN Employee\_ID varchar(11);

ALTER TABLE ERSEARNINGS ALTER COLUMN Employee\_ID SET DATA TYPE varchar(11);

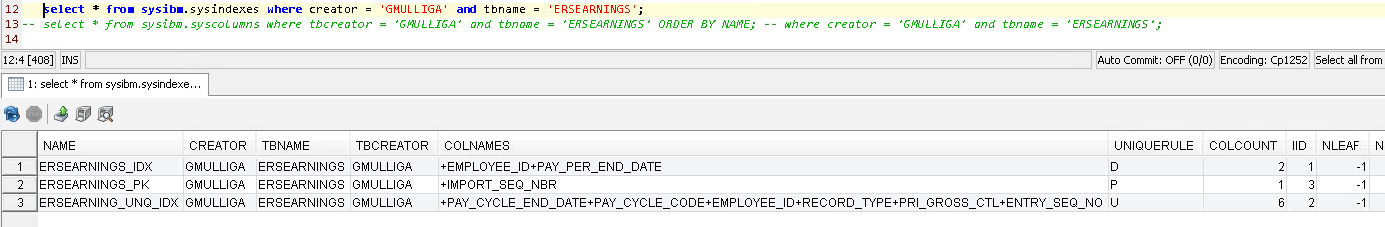


--CREATE NONCLUSTERED INDEX ERSEARNINGS\_IDX

-- ON ERSEarnings (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);

CREATE INDEX ERSEARNINGS\_IDX

ON ERSEARNINGS (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);



1/29/2018:

1. We ran this:

-- CREATE INDEX ERSEARNINGS\_IDX

ON ERSEARNINGS (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);

At first it failed with SQLCODE -668, SQLSTATE 50716 running

2. Larry remembered he had previously had to run REORG before it would let him create indexes. So

we ran this:

CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSEARNINGS');

It ran for 3-4 minutes and returned successfully.

3. We ran:

CREATE INDEX ERSEARNINGS\_IDX

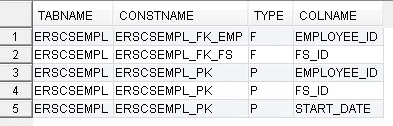
ON ERSEARNINGS (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);

That ran for a few minutes, then finished successfully

We resumed here on Tuesday 1/30:

-- ALTER TABLE ERSCSEmpl DROP CONSTRAINT ERSCSEmpl\_PK

ALTER TABLE ERSCSEMPL DROP CONSTRAINT ERSCSEmpl\_PK



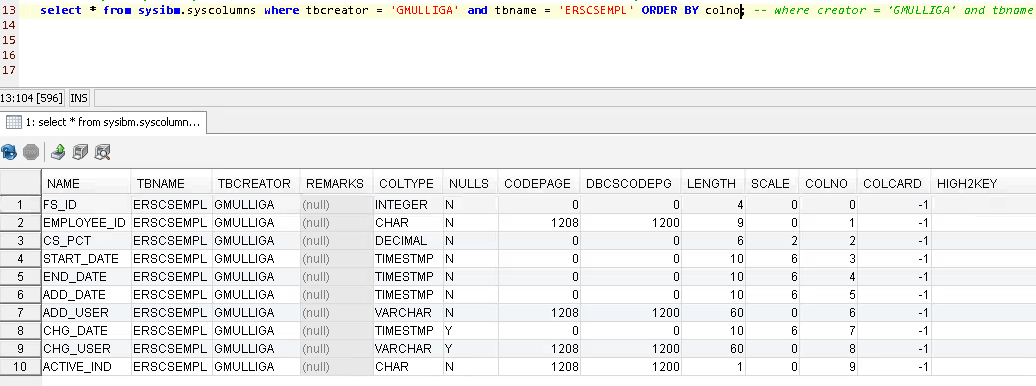
-- ALTER TABLE ERSCSEmpl ALTER COLUMN EMPLOYEE\_ID varchar(11) NOT NULL;

*GM: This threw a 07:17:30 [ALTER - 0 row(s), 0.000 secs] [Error Code: -104, SQL State: 42601] DB2 SQL error: SQLCODE: -104, SQLSTATE: 42601, SQLERRMC: ALTER TABLE ERSCSEMPL ALTER COLUM;BEGIN-OF-STATEMENT;<create\_variable>*

*Fixed by separating into two statements:*

ALTER TABLE ERSCSEMPL ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSCSEMPL ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;



-- ALTER TABLE ERSCSEmpl ADD CONSTRAINT ERSCSEmpl\_PK

-- PRIMARY KEY NONCLUSTERED (FS\_ID ASC, Employee\_ID ASC, Start\_Date ASC)

ALTER TABLE ERSCSEMPL ADD CONSTRAINT ERSCSEmpl\_PK

PRIMARY KEY NONCLUSTERED (FS\_ID ASC, Employee\_ID ASC, Start\_Date ASC)

*GM: failed with 7:20:26 [ALTER - 0 row(s), 0.000 secs] [Error Code: -104, SQL State: 42601] DB2 SQL error: SQLCODE: -104, SQLSTATE: 42601, SQLERRMC: NONCLUSTERED;SEmpl\_PK*

*42601 is often a syntax error. Earlier we fixed a SQLCODE: -104, SQLSTATE: 42601 by splitting the operation into two* commands*, the second being SET NOT NULL*

*ERSCSEMPL\_PK still exists in my GMULLIGA data base, so maybe that's the error.*

***Fixed by reorg'ing it first!***

***Here's what worked***

ALTER TABLE ERSCSEMPL ADD CONSTRAINT ERSCSEMPL\_PK

PRIMARY KEY (FS\_ID, EMPLOYEE\_ID, START\_DATE)

<https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dbobj.doc/doc/c0061098.html>

Index architectures are classified as clustered or non-clustered.

Clustered indexes are indexes whose order of the rows in the data pages corresponds to the order of the rows in the index.

This order is why only one clustered index can exist in any table, whereas, many non-clustered indexes can exist in the table.

In some database systems, the leaf node of the clustered index corresponds to the actual data, not a pointer to data that is found elsewhere.

Both clustered and non-clustered indexes contain only keys and record identifiers in the index structure.

The record identifiers always point to rows in the data pages. With clustered indexes, the database manager attempts

 keep the data in the data pages in the same order as the corresponding keys in the index pages.

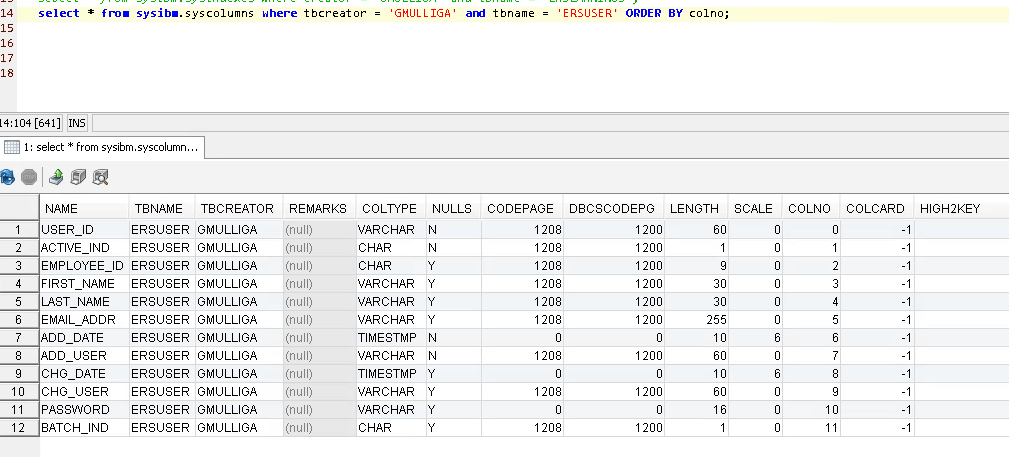
 Thus the database manager attempts to insert rows with similar keys onto the same pages. If the table is reorganized,

 data is inserted into the data pages in the order of the index keys. The database manager does not maintain any order

 of the data when compared to the order of the corresponding keys of a non-clustered index.

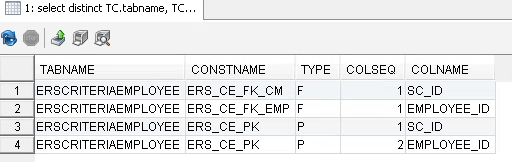
-- ALTER TABLE ERSUser ALTER COLUMN EMPLOYEE\_ID varchar(11);

ALTER TABLE ERSUSER ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);



-- ALTER TABLE ERSCriteriaEmployee DROP CONSTRAINT ERSCriteriaEmployee\_PK

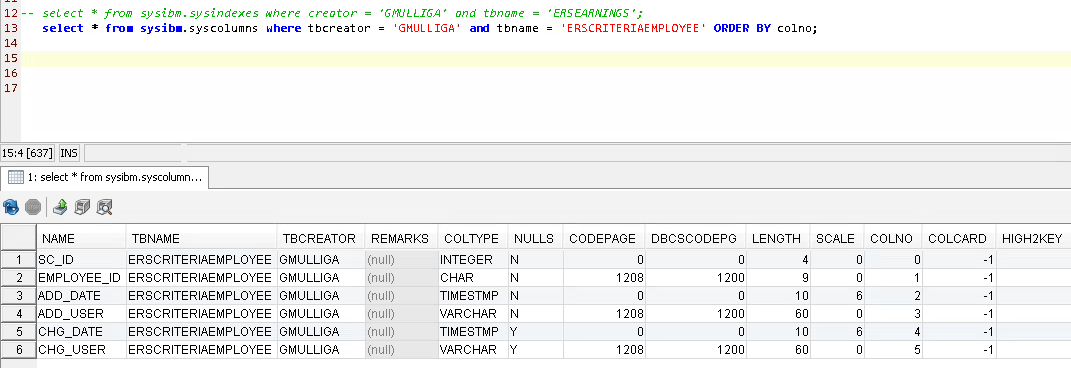
ALTER TABLE ERSCRITERIAEMPLOYEE DROP CONSTRAINT ERS\_CE\_PK



-- ALTER TABLE ERSCriteriaEmployee ALTER COLUMN EMPLOYEE\_ID varchar(11) NOT NULL;

ALTER TABLE ERSCRITERIAEMPLOYEE ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSCRITERIAEMPLOYEE ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;

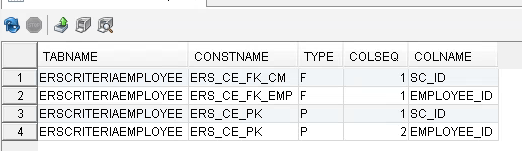


-- ALTER TABLE ERSCriteriaEmployee ADD CONSTRAINT ERSCriteriaEmployee\_PK

-- PRIMARY KEY NONCLUSTERED (SC\_ID ASC, Employee\_ID)

ALTER TABLE ERSCRITERIAEMPLOYEE ADD CONSTRAINT ERS\_CE\_PK

PRIMARY KEY NONCLUSTERED (SC\_ID ASC, Employee\_ID)



*GM: ran it as shown above:*

*failed with 07:50:18 [ALTER - 0 row(s), 0.000 secs] [Error Code: -104, SQL State: 42601] DB2 SQL error: SQLCODE: -104, SQLSTATE: 42601, SQLERRMC: NONCLUSTERED;yee\_PK*

*PRIMARY KEY;<space>42601 is often a syntax error. Earlier we fixed a SQLCODE: -104, SQLSTATE: 42601 by splitting the operation into two* commands*, the second being SET NOT NULL*

|  |  |
| --- | --- |
| *42601* | *A character, token, or clause is invalid or missing.* |
|  |  |

*Then we deleted the NONCLUSTERED and the ASC args, reran it, and got a different error*

*[Error Code: -668, SQL State: 57016] DB2 SQL error: SQLCODE: -668, SQLSTATE: 57016, SQLERRMC: 7;ERSDEV2.ERSCRITERIAEMPLOYEE*

n DB2, you may face this error. This error means that the access to the table is restricted. Do not be worry if you see this error because it may happen after the ALTER TABLE DROP COLUMN command.

In order to fix this error, you need to run the REORG command as follows:

REORG TABLE [YOUR\_TABLE\_NAME]

After running this command, you will be able to access the table.

So we ran the reorg

CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSCRITERIAEMPLOYEE');

Then reran:

ALTER TABLE ERSCRITERIAEMPLOYEE ADD CONSTRAINT ERS\_CE\_PK

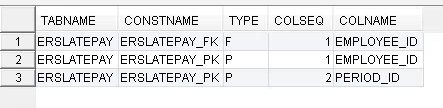
PRIMARY KEY (SC\_ID , Employee\_ID)

**Result? It worked.**

Existing PK is being dropped because it contains employee id.

-- ALTER TABLE ERSLatePay DROP CONSTRAINT ERSLatePay\_PK

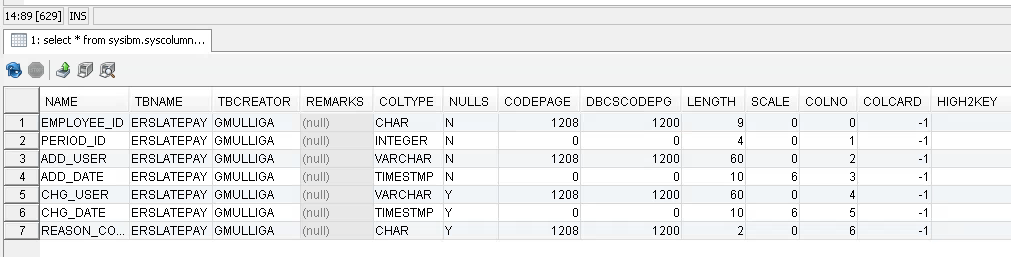
ALTER TABLE ERSLATEPAY DROP CONSTRAINT ERSLatePay\_PK



-- ALTER TABLE ERSLatePay ALTER COLUMN EMPLOYEE\_ID varchar(11) NOT NULL;

ALTER TABLE ERSLATEPAY ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSLATEPAY ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;



**Step 2 (continued): Employee Update from char(9) to varchar(11)**

-- ALTER TABLE ERSLatePay ADD CONSTRAINT ERSLatePay\_PK PRIMARY KEY(Employee\_ID,Period\_ID)

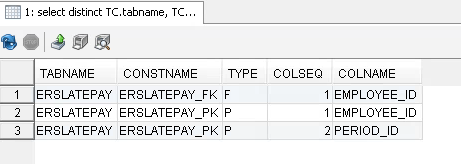
ALTER TABLE ERSLATEPAY ADD CONSTRAINT ERSLATEPAY\_PK PRIMARY KEY(Employee\_ID,Period\_ID)

*GM: failed with 57016 - needs reorg*

*[Error Code: -668, SQL State: 57016] DB2 SQL error: SQLCODE: -668, SQLSTATE: 57016, SQLERRMC: 7;ERSDEV2.ERSCRITERIAEMPLOYEE*

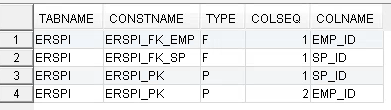
*After the reorg, reran it and it worked.*

|  |  |
| --- | --- |
|  |  |



-- ALTER TABLE ERSPI DROP CONSTRAINT ERSPI\_PK

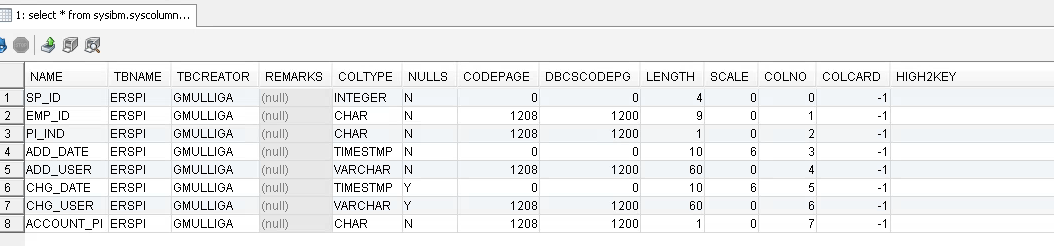
ALTER TABLE ERSPI DROP CONSTRAINT ERSPI\_PK



-- ALTER TABLE ERSPI ALTER COLUMN EMP\_ID varchar(11) NOT NULL;

ALTER TABLE ERSPI ALTER COLUMN EMP\_ID SET DATA TYPE varchar(11) ;

ALTER TABLE ERSPI ALTER COLUMN EMP\_ID SET NOT NULL;



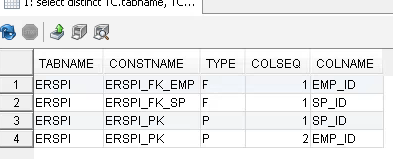
**resumed here on Wednesday 1/31:**

-- ALTER TABLE ERSPI ADD CONSTRAINT ERSPI\_PK PRIMARY KEY (SP\_ID, Emp\_ID)

ALTER TABLE ERSPI ADD CONSTRAINT ERSPI\_PK PRIMARY KEY (SP\_ID, Emp\_ID)

*GM: This 'ALTER TABLE XXX ADD CONSTRAINT YYY' fails for a different reason than the previous one:*

|  |  |
| --- | --- |
| *42889* | *The table already has a primary key.* |



-- ALTER TABLE ERSExcludeUserToken DROP CONSTRAINT ERSExcludeUserToken\_PK

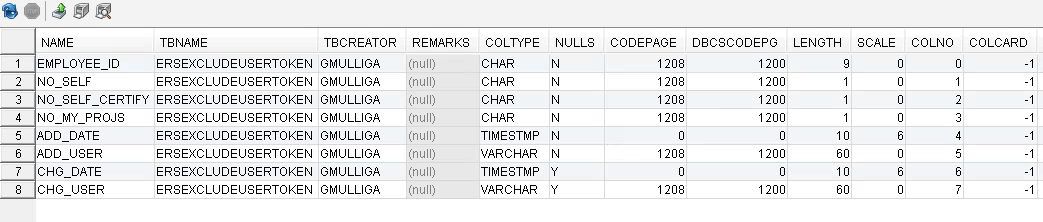
ALTER TABLE ERSEXCLUDEUSERTOKEN DROP CONSTRAINT ERS\_EUToken\_PK



-- ALTER TABLE ERSExcludeUserToken ALTER COLUMN EMPLOYEE\_ID varchar(11) not null;

ALTER TABLE ERSEXCLUDEUSERTOKEN ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSEXCLUDEUSERTOKEN ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;



-- ALTER TABLE ERSExcludeUserToken ADD CONSTRAINT ERSExcludeUserToken\_PK

-- PRIMARY KEY NONCLUSTERED (Employee\_ID ASC)

ALTER TABLE ERSEXCLUDEUSERTOKEN ADD CONSTRAINT ERS\_ERSEUTOKEN\_PK

PRIMARY KEY NONCLUSTERED (Employee\_ID ASC)

|  |  |
| --- | --- |
| *42601* | *A character, token, or clause is invalid or missing.* |
|  |  |

1. Removed NONCLUSTERED and ran again: received 42601 again

2. Removed both NONCLUSTERED and ACS ran again: received 57016

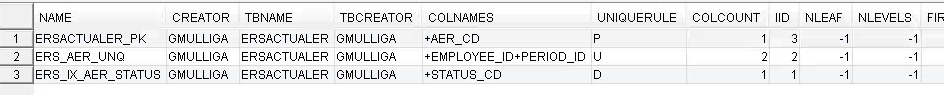
3. Ran Reorg.

4. Ran as in 2 above: WORKED!



-- DROP INDEX ERSActualER.ERS\_AER\_UNIQ

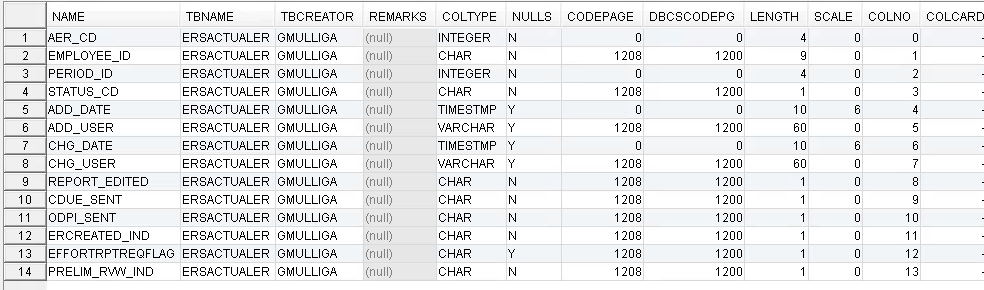
DROP INDEX ERS\_AER\_UNQ



-- ALTER TABLE ERSActualER ALTER COLUMN Employee\_ID varchar(11) not null;

ALTER TABLE ERSACTUALER ALTER COLUMN Employee\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSACTUALER ALTER COLUMN Employee\_ID SET NOT NULL;



-- CREATE UNIQUE NONCLUSTERED INDEX ERS\_AER\_UNIQ ON ERSActualER( employee\_id, period\_id );

CREATE UNIQUE NONCLUSTERED INDEX ERS\_AER\_UNQ ON ERSACTUALER( employee\_id, period\_id );

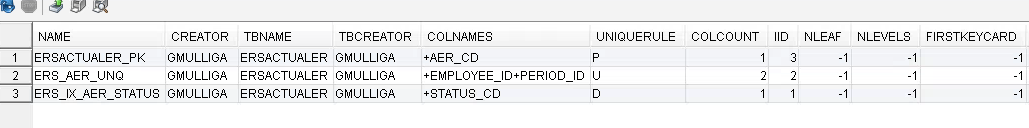
*GM: As constituted above, it failed with:*

|  |  |
| --- | --- |
| *42601* | *A character, token, or clause is invalid or missing.* |

***But it worked after removing the NONCLUSTERED:***

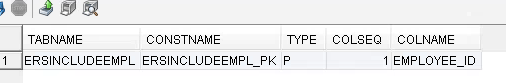
***CREATE UNIQUE INDEX ERS\_AER\_UNQ ON ERSACTUALER( employee\_id, period\_id );***

*NONCLUSTERED is the default in DB2. Its presence will cause the command to fail so remove NONCLUSTERED.*



-- ALTER TABLE ERSIncludeEmpl DROP CONSTRAINT ERSIncludeEmpl\_PK

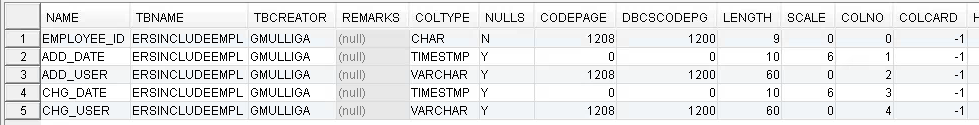
ALTER TABLE ERSINCLUDEEMPL DROP CONSTRAINT ERSINCLUDEEMPL\_PK



- ALTER TABLE ERSIncludeEmpl ALTER COLUMN Employee\_ID varchar(11) not null;

ALTER TABLE ERSINCLUDEEMPL ALTER COLUMN Employee\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSINCLUDEEMPL ALTER COLUMN Employee\_ID SET not null;



-- ALTER TABLE ERSIncludeEmpl ADD CONSTRAINT ERSIncludeEmpl\_PK

-- PRIMARY KEY NONCLUSTERED (Employee\_ID ASC)

1. Ran the following; rec'd 42601:

ALTER TABLE ERSIncludeEmpl ADD CONSTRAINT ERSIncludeEmpl\_PK

PRIMARY KEY NONCLUSTERED (Employee\_ID ASC)

2. Removed NONCLUSTERED and ran again: received 42601 again

ALTER TABLE ERSIncludeEmpl ADD CONSTRAINT ERSIncludeEmpl\_PK

PRIMARY KEY (Employee\_ID ASC)

3. Removed both NONCLUSTERED and ACS ran again: received 57016

ALTER TABLE ERSIncludeEmpl ADD CONSTRAINT ERSIncludeEmpl\_PK

PRIMARY KEY (Employee\_ID)

4. Ran Reorg.

5. Ran as in 2 above: WORKED!

-- ALTER TABLE ERSPIEmplTank DROP CONSTRAINT ERSPIEmplTank\_PK

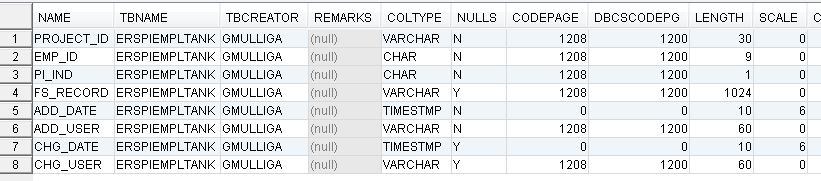
ALTER TABLE ERSPIEMPLTANK DROP CONSTRAINT ERSPIEmplTank\_PK



-- ALTER TABLE ERSPIEmplTank ALTER COLUMN Emp\_ID varchar(11) not null;

ALTER TABLE ERSPIEMPLTANK ALTER COLUMN Emp\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSPIEMPLTANK ALTER COLUMN Emp\_ID SET not null;



-- ALTER TABLE ERSPIEmplTank ADD CONSTRAINT ERSPIEmplTank\_PK PRIMARY KEY NONCLUSTERED (Project\_ID ASC, Emp\_ID)

ALTER TABLE ERSPIEmplTank ADD CONSTRAINT ERSPIEmplTank\_PK PRIMARY KEY NONCLUSTERED (Project\_ID ASC, Emp\_ID)

1. Ran as shown above and rec'd 42601.

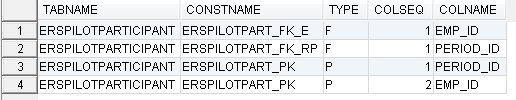
2. Removed ASC, reran, and still rec'd 42601.

3. Removed NONCLUSTERED, reran, and got 57016.  
4. Regorg'd and reran as shown in 3, and IT WORKED!



-- ALTER TABLE ERSPilotParticipant DROP CONSTRAINT ERSPilotPart\_PK

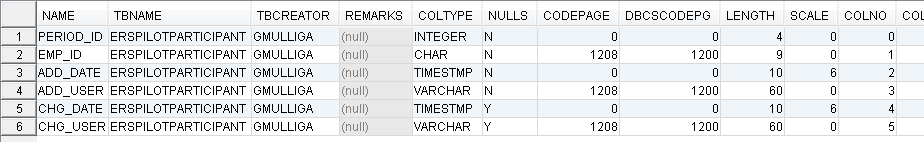
ALTER TABLE ERSPILOTPARTICIPANT DROP CONSTRAINT ERSPilotPart\_PK



-- ALTER TABLE ERSPilotParticipant ALTER COLUMN emp\_id varchar(11) not null;

ALTER TABLE ERSPILOTPARTICIPANT ALTER COLUMN emp\_id SET DATA TYPE varchar(11);

ALTER TABLE ERSPILOTPARTICIPANT ALTER COLUMN emp\_id SET not null;



-- ALTER TABLE ERSPilotParticipant ADD CONSTRAINT ERSPilotPart\_PK

-- PRIMARY KEY NONCLUSTERED (period\_id, Emp\_ID)

ALTER TABLE ERSPILOTPARTICIPANT ADD CONSTRAINT ERSPilotPart\_PK

PRIMARY KEY NONCLUSTERED (period\_id, Emp\_ID)

GM: failed with 42601

|  |  |
| --- | --- |
| *42601* | *A character, token, or clause is invalid or missing.* |
|  |  |

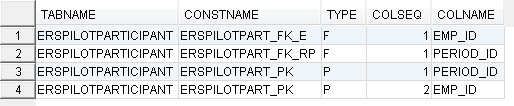
1. Removed NONCLUSTERED and reran as shown next:

ALTER TABLE ERSPILOTPARTICIPANT ADD CONSTRAINT ERSPilotPart\_PK

PRIMARY KEY (period\_id, Emp\_ID)

Rec'd 57016

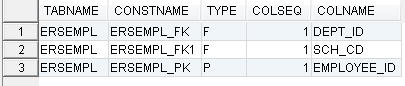
2. Reorg'd, and reran as shown in 1, and it worked.



**resumed here on Thursday 2/1**

-- ALTER TABLE ERSEMPL ADD CONSTRAINT ERSEmpl\_PK PRIMARY KEY(Employee\_ID);

ALTER TABLE ERSEmpl ADD CONSTRAINT ERSEmpl\_PK PRIMARY KEY(Employee\_ID);

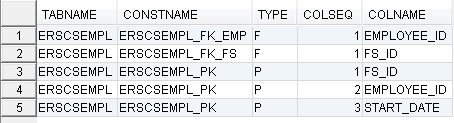


-- ALTER TABLE ERSCSEmpl ADD CONSTRAINT ERSCSEmpl\_fk\_emp

-- FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSCSEMPL ADD CONSTRAINT ERSCSEmpl\_fk\_emp

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

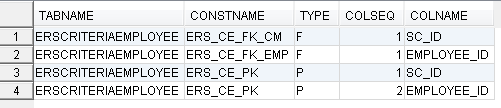


-- ALTER TABLE ERSCriteriaEmployee ADD CONSTRAINT ERS\_CE\_FK\_EMP

-- FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSCRITERIAEMPLOYEE ADD CONSTRAINT ERS\_CE\_FK\_EMP

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);



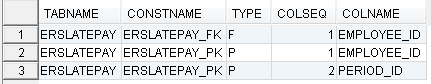
**Step 2 (continued): Employee Update from char(9) to varchar(11)**

-- ALTER TABLE ERSLatePay ADD CONSTRAINT ERSLatePay\_FK

-- FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSLATEPAY ADD CONSTRAINT ERSLatePay\_FK

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

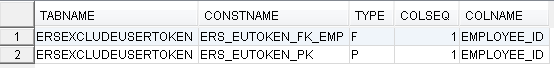


-- ALTER TABLE ERSExcludeUserToken ADD CONSTRAINT ERS\_EUTOKEN\_FK\_EMP

-- FOREIGN KEY (EMPLOYEE\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSEXCLUDEUSERTOKEN ADD CONSTRAINT ERS\_EUTOKEN\_FK\_EMP

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES ERSEmpl (Employee\_ID);

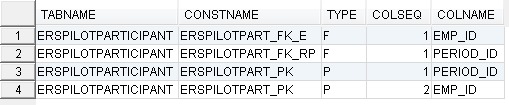


-- ALTER TABLE ERSPilotParticipant ADD CONSTRAINT ERSPilotPart\_FK\_E

-- FOREIGN KEY (EMP\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSPILOTPARTICIPANT ADD CONSTRAINT ERSPilotPart\_FK\_E

FOREIGN KEY (EMP\_ID) REFERENCES ERSEmpl (Employee\_ID);

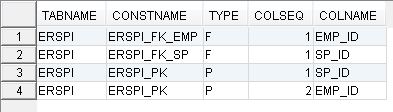


-- ALTER TABLE ERSPI ADD CONSTRAINT ERSPI\_FK\_EMP FOREIGN KEY (EMP\_ID)

-- REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSPI ADD CONSTRAINT ERSPI\_FK\_EMP FOREIGN KEY (EMP\_ID)

REFERENCES ERSEmpl (Employee\_ID);

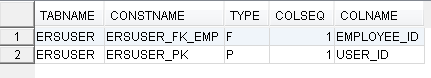


-- ALTER TABLE ERSUser ADD CONSTRAINT ERSUser\_fk\_emp

-- FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSUSER ADD CONSTRAINT ERSUser\_fk\_emp

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

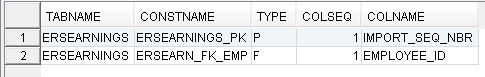


-- ALTER TABLE ERSEarnings ADD CONSTRAINT ERSEarn\_FK\_emp

-- FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSEARNINGS ADD CONSTRAINT ERSEarn\_FK\_emp

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

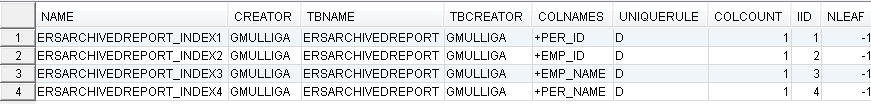


**Step3: Supplemental Employee Updates**

-- DROP INDEX ERSArchivedReport.ersar\_eid

DROP INDEX ERSARCHIVEDREPORT\_INDEX2

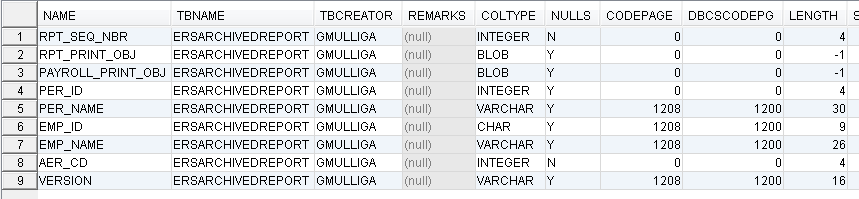
*Gm: The original index name contained 'eid', which may stand for Employee ID, so I selected ERSARCHIVEDREPORT\_INDEX2.*



-- ALTER TABLE ERSArchivedReport ALTER COLUMN EMP\_ID varchar(11) not null;

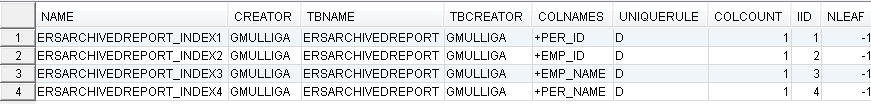
ALTER TABLE ERSARCHIVEDREPORT ALTER COLUMN EMP\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSARCHIVEDREPORT ALTER COLUMN EMP\_ID SET NOT NULL;



-- CREATE INDEX ersar\_eid ON ERSArchivedReport( emp\_id )

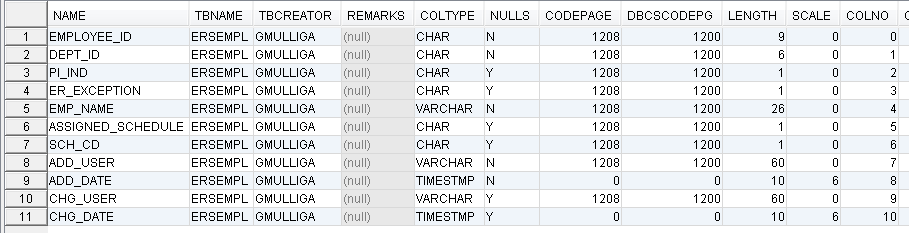
CREATE INDEX *ERSARCHIVEDREPORT\_INDEX2* ON ERSArchivedReport( emp\_id )



**Step 4: Clean Employee IDs**

-- update ersempl set employee\_id=ltrim(employee\_id)

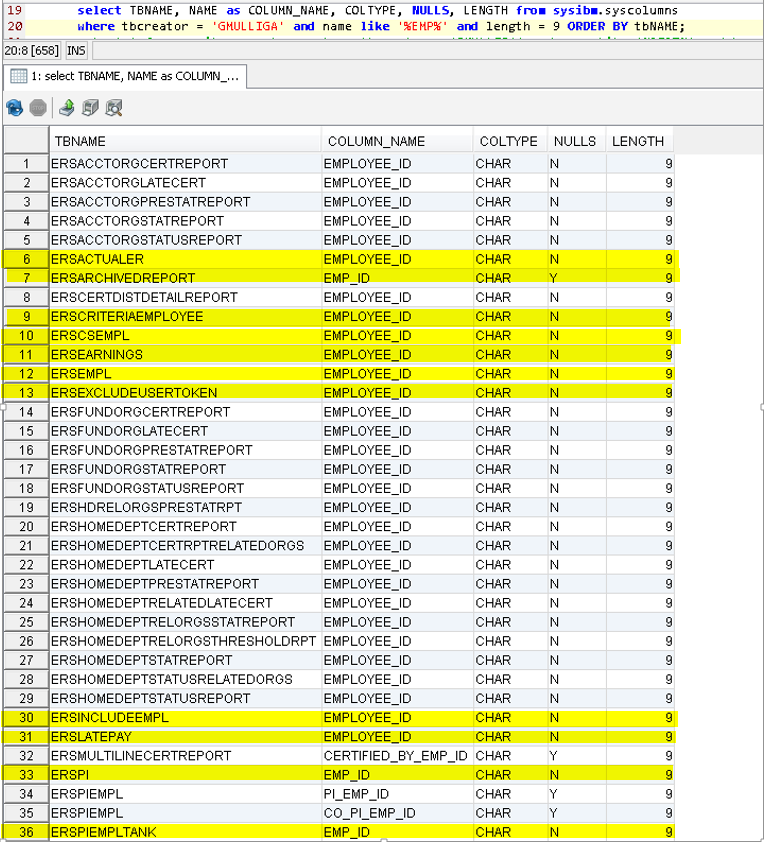
update ERSEMPL set employee\_id=trim(employee\_id) (Note this is trim and that's all we need)

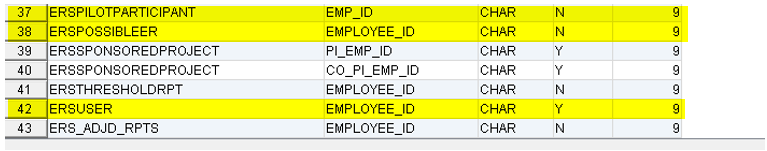


Any other tables ? TBD for GM

GM: Since many of the changes are to convert empl\_id from 9 char to 11 varchar, here are all the tables containing cols like '%EMP%' and also of length = 9.

a. GMULLIGA - after loading & before running migration script. Tables known to be in this script's commands statements are highlighted

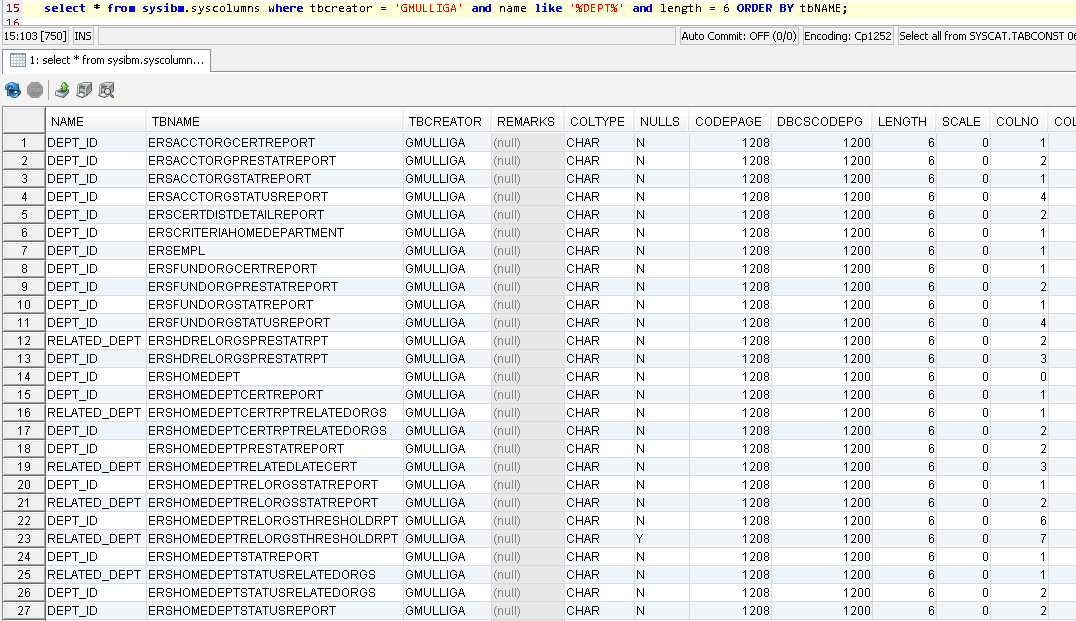


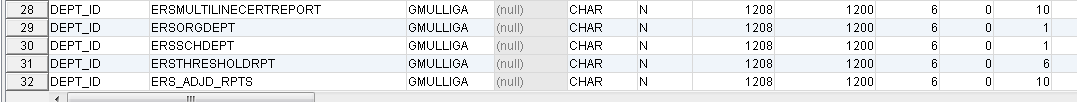


b. ERSDEV2 - After migration script. Many tables that weren't operated on by the script are now gone from the list. Need to ask Slava to load ERSDEV2 again and see if they reappear in the list.

**Step 5: Home Department Updates**

Home Department Update char(6) to varchar(10)

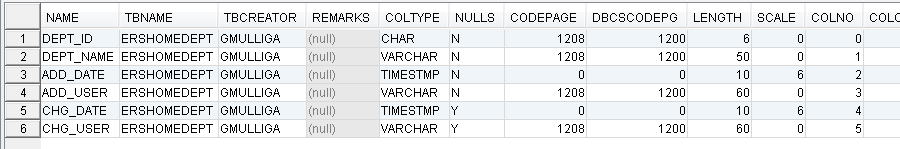




**DB2: Execute DB2 Scripts in this order (SQL Server script below is more thorough)**

-- ALTER TABLE ERSHOMEDEPT DROP CONSTRAINT ERSHomeDept\_PK;

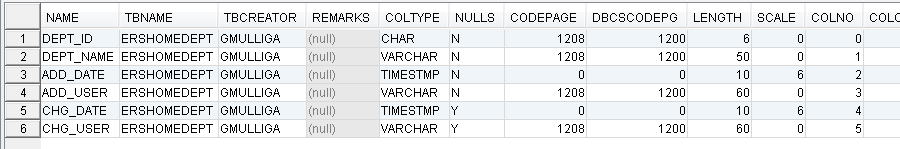
ALTER TABLE ERSHOMEDEPT DROP CONSTRAINT ERSHomeDept\_PK;





-- ALTER TABLE ERSHOMEDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSHOMEDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);



-- ALTER TABLE ERSHOMEDEPT ADD CONSTRAINT ERSHomeDept\_PK PRIMARY KEY(DEPT\_ID);

ALTER TABLE ERSHOMEDEPT ADD CONSTRAINT ERSHomeDept\_PK PRIMARY KEY(DEPT\_ID);



CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSHOMEDEPT ');

1/29/2018:

1. We ran this:

-- CREATE INDEX ERSEARNINGS\_IDX ON ERSEARNINGS (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);

It failed with SQLCODE -668, SQLSTATE 50716 running

2. Larry remembered he had previously had to run REORG before it would let him create indexes. So

we ran this:

CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSEARNINGS');

It ran for 3-4 minutes and returned successfully.

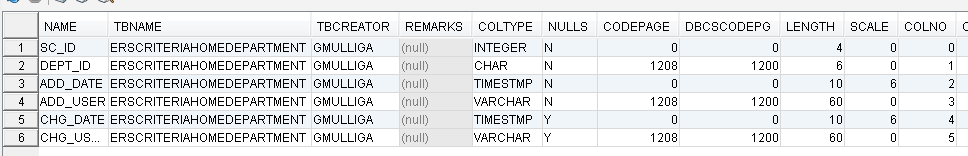
3. We ran:

CREATE INDEX ERSEARNINGS\_IDX ON ERSEARNINGS (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);

That ran for a few minutes, then finished successfully

-- ALTER TABLE ERSCriteriaHomeDepartment ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);



CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSCriteriaHomeDepartment ');

-- ALTER TABLE ERSSchDept ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

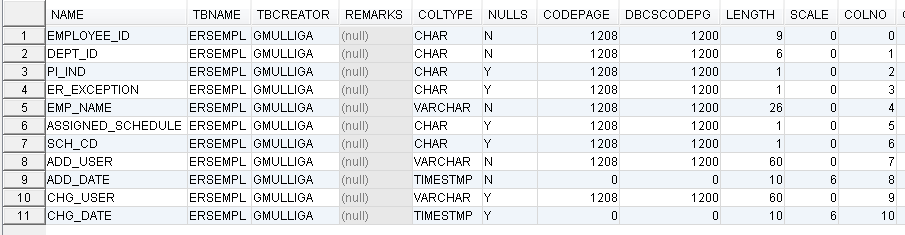
ALTER TABLE ERSSCHDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);



CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSSchDept ');

-- ALTER TABLE ERSEmpl ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

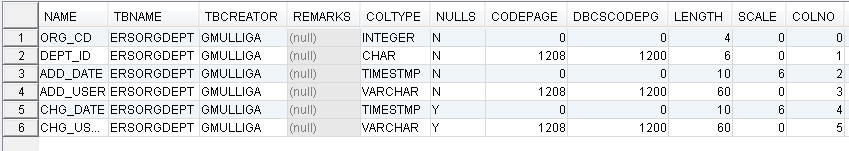
ALTER TABLE ERSEMPL ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);



CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSEmpl ');

-- ALTER TABLE ERSORGDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSORGDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

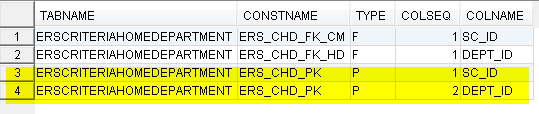


CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSORGDEPT ');

**MS SQL SERVER: Execute Microsoft SQL Server Scripts in this order**

-- ALTER TABLE ERSCriteriaHomeDepartment DROP CONSTRAINT ERSCriteriaHomeDepartment\_PK

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT DROP CONSTRAINT ERS\_CHD\_PK

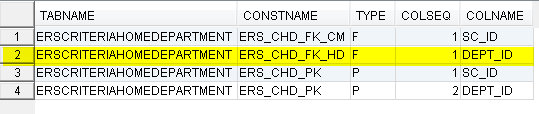


GM: the PK has two columns, SC\_ID and DEPT\_ID. Both should be dropped before DEPT\_ID can be expanded Dropping the constraint will drop the composite key.

-- ALTER TABLE ERSCriteriaHomeDepartment DROP CONSTRAINT ERSCriteriaHomeDepartment\_FK

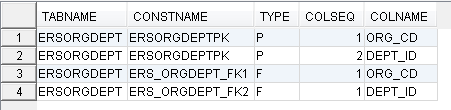
ALTER TABLE ERSCRITERIAHOMEDEPARTMENT DROP CONSTRAINT ERS\_CHD\_FK\_HD

GM: Yes, drop the constraint on DEPT\_ID and add it back later.



-- ALTER TABLE ERSOrgDept DROP CONSTRAINT ERSOrgDept\_PK

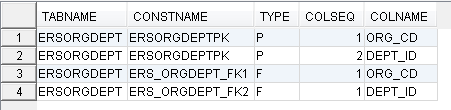
ALTER TABLE ERSORGDEPT DROP CONSTRAINT ERSORGDEPTPK



GM: The existing PK is named ERSORGDEPTPK

-- ALTER TABLE ERSOrgDept DROP CONSTRAINT ERSOrgDept\_FK1

ALTER TABLE ERSORGDEPT DROP CONSTRAINT ERS\_OrgDept\_FK2

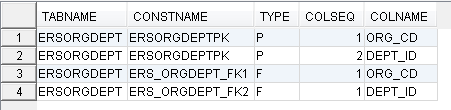


GM: there are two FKs, one on ORG\_CD and one on DEPT\_ID. This one is on DEPT\_ID, constraint is named ERS\_ORGDEPT\_FK2. But FK2 was already dropped on approx. p 9, so this is redundant and no need to run. Skip.

Also UCSD's ERS\_ORGDEPT\_FK1 is on ORG\_CD so maybe it doesn’t need to be dropped since UCPath may not be changing ORG\_CD.

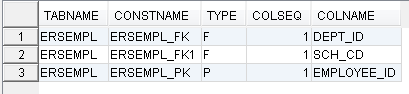
TBD Don’t run this page. It was already deleted earlier. Additionally, we only want to delete FK2 (DEPT) not FK1 (ORG), so removed the FK1 drops

--ALTER TABLE ERSOrgDept DROP CONSTRAINT ERSOrgDept\_FK1 -- Already removed in Step2 but ORG\_CD isn’t changing for Path so ORG\_CD constraints should stay.



-- ALTER TABLE ERSEmpl DROP CONSTRAINT ERSEmpl\_FK

ALTER TABLE ERSEMPL DROP CONSTRAINT ERSEmpl\_FK

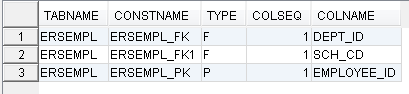


GM: likely dropped on p.15. But keep this page since needs to be dropped.

GM: FK1 is on SCH\_CD and Path is not changing SCH\_CD so I think it can stay. Don’t delete.

-- ALTER TABLE ERSEmpl DROP CONSTRAINT ERSEmpl\_FK1

ALTER TABLE ERSEMPL DROP CONSTRAINT ERSEmpl\_FK1



-- ALTER TABLE ERSHOMEDEPT DROP CONSTRAINT ERSHomeDept\_PK;

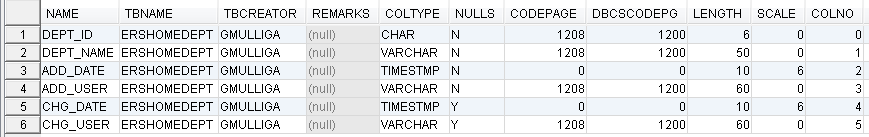
ALTER TABLE ERSHOMEDEPT DROP CONSTRAINT ERSHomeDept\_PK;



-- ALTER TABLE ERSHOMEDEPT ALTER COLUMN DEPT\_ID varchar(10) NOT NULL;

ALTER TABLE ERSHOMEDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSHOMEDEPT ALTER COLUMN DEPT\_ID SET NOT NULL;



-- ALTER TABLE ERSHOMEDEPT ADD CONSTRAINT ERSHomeDept\_PK PRIMARY KEY(DEPT\_ID);

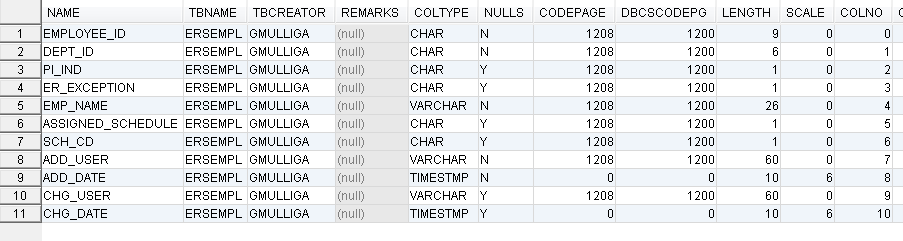
ALTER TABLE ERSHOMEDEPT ADD CONSTRAINT ERSHomeDept\_PK PRIMARY KEY(DEPT\_ID);



-- ALTER TABLE ERSEmpl ALTER COLUMN DEPT\_ID varchar(10) NOT NULL;

ALTER TABLE ERSEMPL ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSEMPL ALTER COLUMN DEPT\_ID SET NOT NULL;

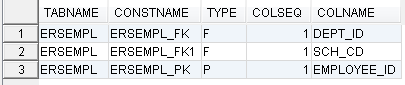


-- ALTER TABLE [dbo].[ERSEmpl] WITH CHECK ADD CONSTRAINT [ERSEmpl\_FK]

-- FOREIGN KEY([Dept\_ID]) REFERENCES [dbo].[ERSHomeDept] ([Dept\_ID])

ALTER TABLE ERSEMPL ADD CONSTRAINT ERSEmpl\_FK

FOREIGN KEY (Dept\_ID) REFERENCES ERSHomeDept (Dept\_ID)



GM: this may not be needed - removed the 'WITH CHECK' from the declaration. See below.

-- ALTER TABLE [dbo].[ERSEmpl] CHECK CONSTRAINT [ERSEmpl\_FK]

ALTER TABLE [dbo].[ERSEmpl] CHECK CONSTRAINT [ERSEmpl\_FK]

GM: This and the constraint declaration before it look like code that DB Visualizer generated to recreate the constraint if needed. My research findings:

A "CHECK CONSTRAINT" is a restriction placed on the allowable values that a column can contain, like [1-5], or ['This', 'That']. If the Check Constraint is turned on, only the values in the brackets [ ] will be accepted. When the constraint is declared with "NOCHECK" the validation is turned off. In that case an additional statement is needed to turn it on like this:

ALTER TABLE [dbo].[ERSEmpl] CHECK CONSTRAINT [ERSEmpl\_FK]

But this constraint was declared with CHECK, not NOCHECK. It is already turned on. This "CHECK CONSTRAINT" statement seems redundant. Trying to run it returns 42601.

Another q: The link below shows an example where a range of values [FK\_T2\_T1] is declared with NOCHECK. The declaration on the previous page does not specify a range of values in a [ ] clause. Apparently that means that all values will be allowed.

In sum, since the preceding statement declares it with CHECK, there seems to be no need to run this "CHECK CONSTRAINT" statement.

See <https://dba.stackexchange.com/questions/24297/alter-table-check-constraint1>

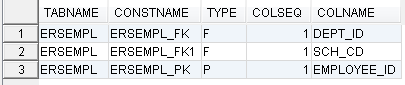
**Step 5 (continued): Home Department Updates**

GM: don't run this; see below.

-- ALTER TABLE [dbo].[ERSEmpl] WITH CHECK ADD CONSTRAINT [ERSEmpl\_FK1]

-- FOREIGN KEY([Sch\_CD]) REFERENCES [dbo].[ERSSchType] ([Sch\_CD])

GM: ERSEMPL\_FK1 is on SCH\_CD. Path isn’t changing SCH\_CD so this constraint should not have been dropped earlier and therefore not now be added back.



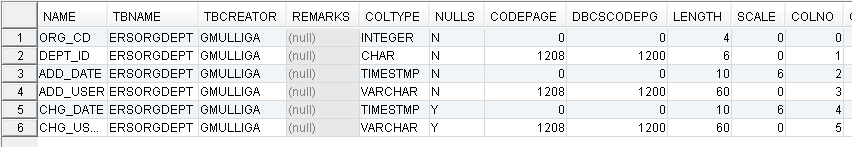
ALTER TABLE [dbo].[ERSEmpl] CHECK CONSTRAINT [ERSEmpl\_FK1]

GM: Since the "ALTER TABLE … WITH CHECK … ADD CONSTRAINT" will not be run, the "ALTER TABLE … CHECK CONSTRAINT" won't be run either. FYI the "ALTER TABLE … CHECK CONSTRAINT" statement is not needed when the constraint is added WITH CHECK. See "restriction placed on the allowable values" in this document for more information.

-- ALTER TABLE ERSOrgDept ALTER COLUMN DEPT\_ID varchar(10) NOT NULL

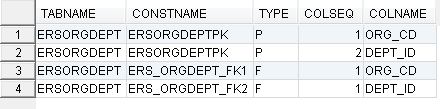
ALTER TABLE ERSORGDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSORGDEPT ALTER COLUMN DEPT\_ID SET NOT NULL



-- ALTER TABLE ERSOrgDept ADD CONSTRAINT ERSOrgDept\_PK PRIMARY KEY(Org\_CD, Dept\_ID)

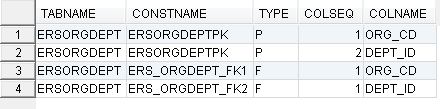
ALTER TABLE ERSORGDEPT ADD CONSTRAINT ERSORGDEPTPK PRIMARY KEY(Org\_CD, Dept\_ID)



-- ALTER TABLE ERSOrgDept WITH CHECK ADD CONSTRAINT [ERSOrgDept\_FK]

-- FOREIGN KEY([Org\_CD]) REFERENCES ERSOrganization ([Org\_CD])

GM: Don't run. It looks like this should be FK1 since its on ORG\_CD. I think since Path isn’t changing ORG\_CD, this constraint should not have been dropped earlier and therefore doesn’t need to be added back now.



GM: Don't run the following CHECK CONSTRAINT. See "restriction placed on the allowable values" in this document for more information.

-- ALTER TABLE ERSOrgDept CHECK CONSTRAINT ERSOrgDept\_FK

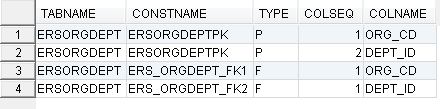
ALTER TABLE ERSOrgDept CHECK CONSTRAINT ERSOrgDept\_FK

-- ALTER TABLE ERSOrgDept WITH CHECK ADD CONSTRAINT [ERSOrgDept\_FK1]

-- FOREIGN KEY([Dept\_ID]) REFERENCES ERSHomeDept([Dept\_ID])

ALTER TABLE ERSOrgDept ADD CONSTRAINT ERSORGDEPT\_FK2

FOREIGN KEY (Dept\_ID) REFERENCES ERSHomeDept (Dept\_ID)



GM: This is FK2 since it's on DEPT\_ID

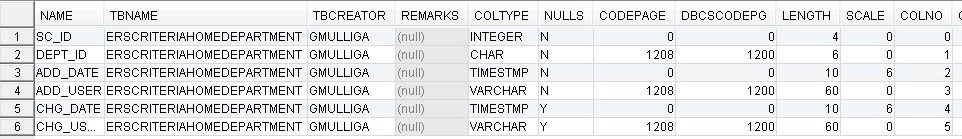
GM: Don't run the following CHECK CONSTRAINT. See "restriction placed on the allowable values" in this document for more information.

-- ALTER TABLE ERSOrgDept CHECK CONSTRAINT ERSOrgDept\_FK1

ALTER TABLE ERSOrgDept CHECK CONSTRAINT ERSORGDEPT\_FK2

-- ALTER TABLE ERSCriteriaHomeDepartment ALTER COLUMN DEPT\_ID varchar(10)

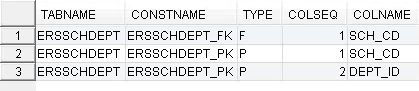
ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10)



GM: yes, drop this constraint since its composite key contains DEPT\_ID

-- ALTER TABLE ERSSchDept DROP CONSTRAINT ERSSchDept\_PK

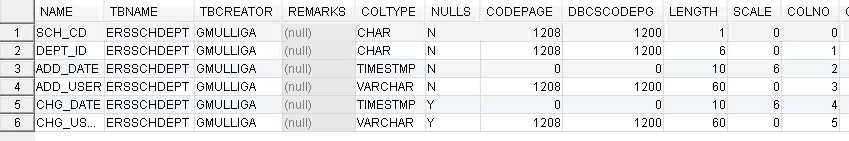
ALTER TABLE ERSSCHDEPT DROP CONSTRAINT ERSSchDept\_PK



-- ALTER TABLE ERSSchDept ALTER COLUMN DEPT\_ID varchar(10) not null;

ALTER TABLE ERSSCHDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

ALTER TABLE ERSSCHDEPT ALTER COLUMN DEPT\_ID SET NOT NULL;

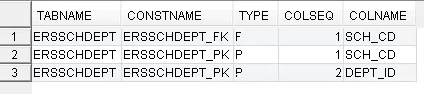


-- ALTER TABLE ERSSchDept ADD CONSTRAINT ERSSchDept\_PK

-- PRIMARY KEY NONCLUSTERED (Sch\_CD ASC, Dept\_ID ASC)

ALTER TABLE ERSSchDept ADD CONSTRAINT ERSSchDept\_PK

PRIMARY KEY (Sch\_CD, Dept\_ID)

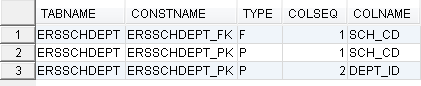


ALTER TABLE ERSSCHDEPT ADD CONSTRAINT ERSSchDept\_PK

PRIMARY KEY NONCLUSTERED (Sch\_CD ASC, Dept\_ID ASC)

ALTER TABLE ERSSCHDEPT ADD CONSTRAINT ERSSchDept\_PK

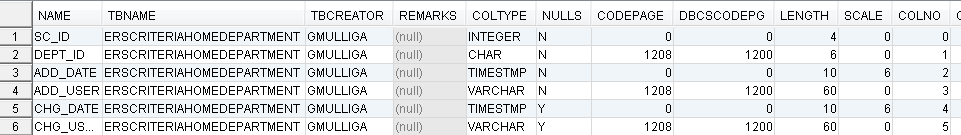
PRIMARY KEY (Sch\_CD, Dept\_ID)



-- ALTER TABLE ERSCriteriaHomeDepartment ALTER COLUMN DEPT\_ID varchar(10) NOT NULL;

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10)

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ALTER COLUMN DEPT\_ID SET NOT NULL;

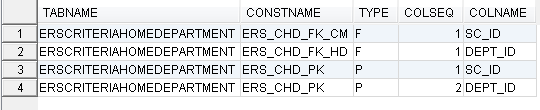


-- ALTER TABLE ERSCriteriaHomeDepartment ADD CONSTRAINT [ERSCriteriaHomeDepartment\_PK]

-- PRIMARY KEY NONCLUSTERED ([SC\_ID] ASC, [Dept\_ID] ASC)

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ADD CONSTRAINT ERS\_CHD\_PK

PRIMARY KEY (SC\_ID, DEPT\_ID)



-- ALTER TABLE ERSCriteriaHomeDepartment WITH CHECK ADD CONSTRAINT

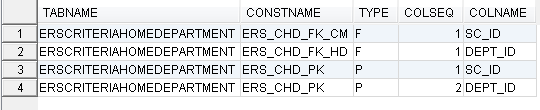
-- [ERSCriteriaHomeDepartment\_FK] FOREIGN KEY([Dept\_ID])

-- REFERENCES [dbo].[ERSHomeDept] ([Dept\_ID])

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ADD CONSTRAINT

ERS\_CHD\_FK\_HD FOREIGN KEY (DEPT\_ID)

REFERENCES ERSHOMEDEPT (DEPT\_ID)



GM: Don't run the following CHECK CONSTRAINT. See "restriction placed on the allowable values" in this document for more information.

-- ALTER TABLE ERSCriteriaHomeDepartment CHECK CONSTRAINT [ERSCriteriaHomeDepartment\_FK

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT CHECK CONSTRAINT [ERS\_CHD\_FK\_HD]

**Step 5 (continued): Home Department Updates**

GM: Find out if Path is changing SC\_ID and therefore if this is needed.

-- ALTER TABLE ERSCriteriaHomeDepartment WITH CHECK ADD CONSTRAINT

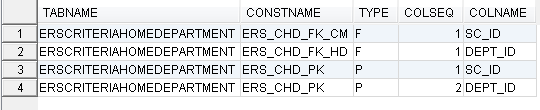
-- [ERSCriteriaHomeDepartment\_FK1] FOREIGN KEY([SC\_ID])

-- REFERENCES [dbo].[ERSCriteriaMaster] ([SC\_ID])

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ADD CONSTRAINT

ERS\_CHD\_FK\_CM FOREIGN KEY(SC\_ID)

REFERENCES ERSCRITERIAMASTER (SC\_ID)



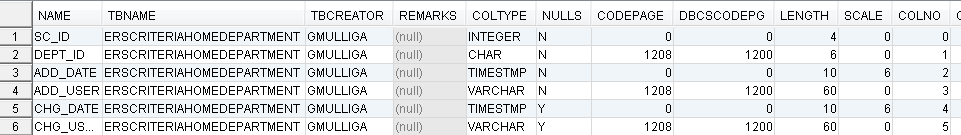
GM: Don't run the following CHECK CONSTRAINT. See "restriction placed on the allowable values" in this document for more information.

-- ALTER TABLE ERSCriteriaHomeDepartment CHECK CONSTRAINT [ERSCriteriaHomeDepartment\_FK1]

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT CHECK CONSTRAINT [ERS\_CHD\_FK\_CM]

-- UPDATE ERSHOMEDEPT set DEPT\_ID=ltrim(DEPT\_ID)

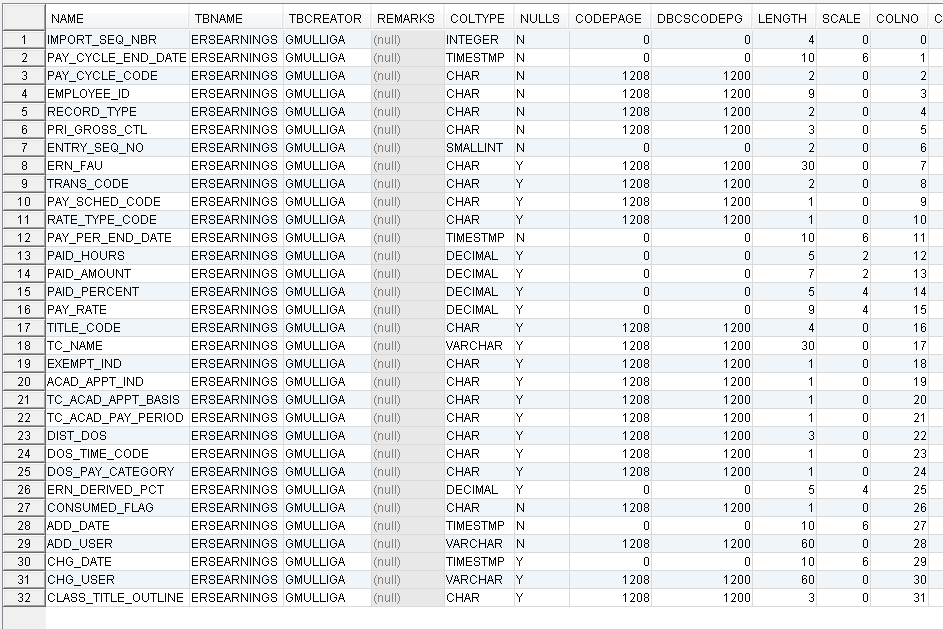
UPDATE ERSHOMEDEPT set DEPT\_ID=TRIM(DEPT\_ID)

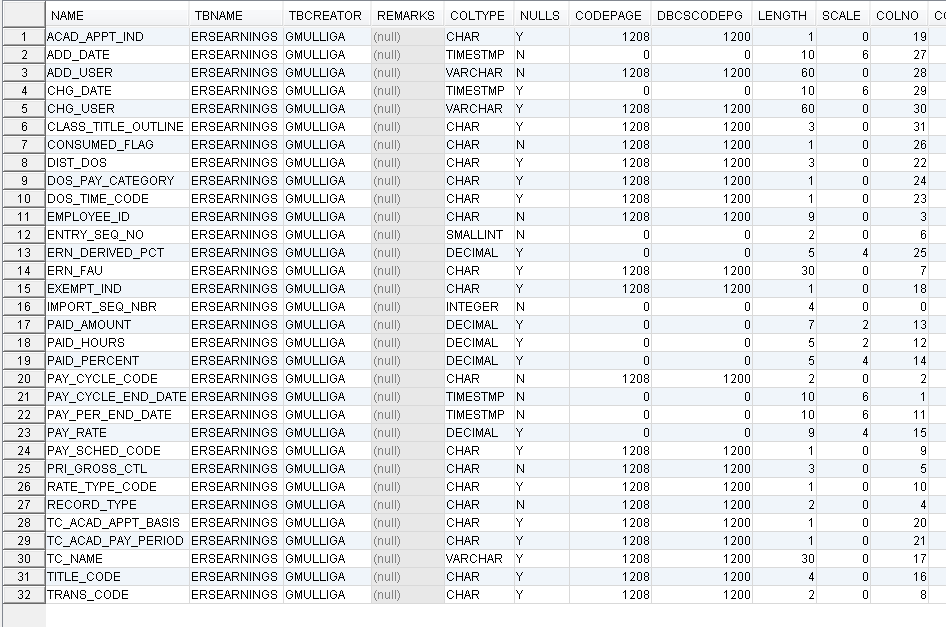


**Step 6: Add new Labor Ledger columns to ERSEARNINGS table**.

GM: Before adding new columns, see existing columns below, first in column number sequence, then in column name sequence

1. **Before**. Ordered by column number sequence



2. **Before**. Ordered by column name.

**-- Run the following command in DB2 environments only (used several times above to fix 57016 errors)**

-- CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ersearnings');

CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERSEARNINGS');

-- **The following 12 ALTER DROP commands are not necessary if columns do not exist in ERSEarnings**

alter table ersearnings drop column RUN\_ID

GM: no such column so do not run

alter table ersearnings drop column EMPL\_RCD

GM: no such column so do not run

alter table ersearnings drop column EFFDT

GM: no such column so do not run

alter table ersearnings drop column EFF\_SEQ

GM: no such column so do not run

alter table ersearnings drop column OFF\_CYCLE

GM: no such column so do not run

alter table ersearnings drop column RUN\_ID\_EARN

GM: no such column so do not run

alter table ersearnings drop column ADDL\_SEQ

GM: no such column so do not run

alter table ersearnings drop column JOURNAL\_ID

GM: no such column so do not run

alter table ersearnings drop column JOURNAL\_LINE

GM: no such column so do not run

alter table ersearnings drop column JOURNAL\_LINE\_REF

GM: no such column so do not run

alter table ersearnings drop column BUSINESS\_UNIT

GM: no such column so do not run

alter table ersearnings drop column PAY\_CAT

GM: no such column so do not run

alter table ersearnings drop column TIME\_CODE

GM: no such column so do not run

GM: all the following columns don’t exist so run the statements to add them. These had 'drop' statements.

-- alter table ersearnings add Run\_Id varchar(10)

alter table ERSEARNINGS add Run\_Id varchar(10)

-- alter table ersearnings add Empl\_Rcd integer

alter table ERSEARNINGS add Empl\_Rcd integer

-- alter table ersearnings add Eff\_Date datetime -- DB2 should be TIMESTAMP

alter table ERSEARNINGS add Eff\_Date TIMESTAMP

-- alter table ersearnings add Eff\_Seq integer

alter table ERSEARNINGS add Eff\_Seq integer

-- alter table ersearnings add Off\_Cycle varchar(1)

alter table ERSEARNINGS add Off\_Cycle varchar(1)

-- alter table ersearnings add Run\_Id\_Earn varchar(10)

alter table ERSEARNINGS add Run\_Id\_Earn varchar(10)

-- alter table ersearnings add Addl\_Seq varchar(10)

alter table ERSEARNINGS add Addl\_Seq varchar(10)

-- alter table ersearnings add Journal\_Id varchar(10)

alter table ERSEARNINGS add Journal\_Id varchar(10)

-- alter table ersearnings add Journal\_Line varchar(10)

alter table ERSEARNINGS add Journal\_Line varchar(10)

-- alter table ersearnings add Journal\_Line\_Ref varchar(10)

alter table ERSEARNINGS add Journal\_Line\_Ref varchar(10)

-- alter table ersearnings add Business\_Unit varchar(5)

alter table ERSEARNINGS add Business\_Unit varchar(5)

-- alter table ersearnings add Pay\_Cat varchar(2)

alter table ERSEARNINGS add Pay\_Cat varchar(2)

-- alter table ersearnings add Time\_Code varchar(2)

alter table ERSEARNINGS add Time\_Code varchar(2)

GM: These did not have 'drop' statements above but still need to be run also to add the columns.

-- alter table ersearnings add Restatement varchar(1)

alter table ERSEARNINGS add Restatement varchar(1)

-- alter table ersearnings add Interface varchar(1)

alter table ERSEARNINGS add Interface varchar(1)

-- Alter TABLE ERSEARNINGS ALTER COLUMN DOS\_TIME\_CODE varchar(2);

Alter TABLE ERSEARNINGS ALTER COLUMN DOS\_TIME\_CODE SET DATA TYPE varchar(2);

GM: change from 1 to 2

-- Alter TABLE ERSEARNINGS ALTER COLUMN DOS\_PAY\_CATEGORY varchar(2);

Alter TABLE ERSEARNINGS ALTER COLUMN DOS\_PAY\_CATEGORY SET DATA TYPE varchar(2);

GM: change from 1 to 2

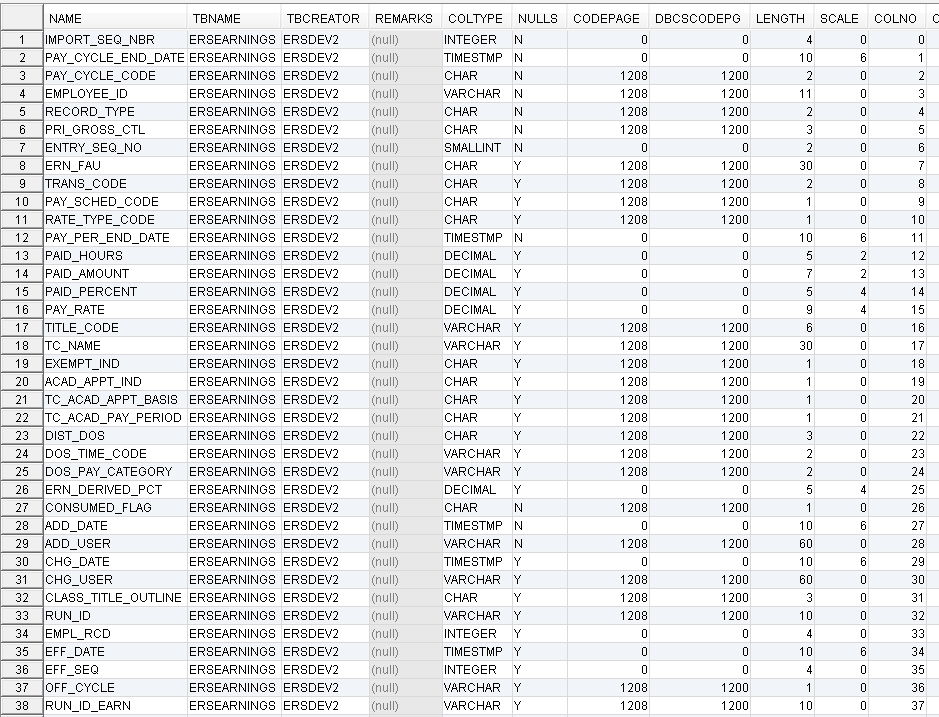
-- Alter TABLE ERSEARNINGS ALTER COLUMN TITLE\_CODE varchar(6)

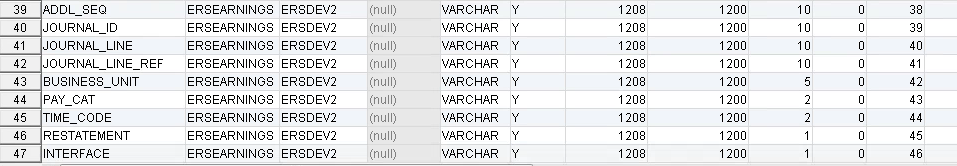
Alter TABLE ERSEARNINGS ALTER COLUMN TITLE\_CODE SET DATA TYPE varchar(6)

GM: change from 4 to 6

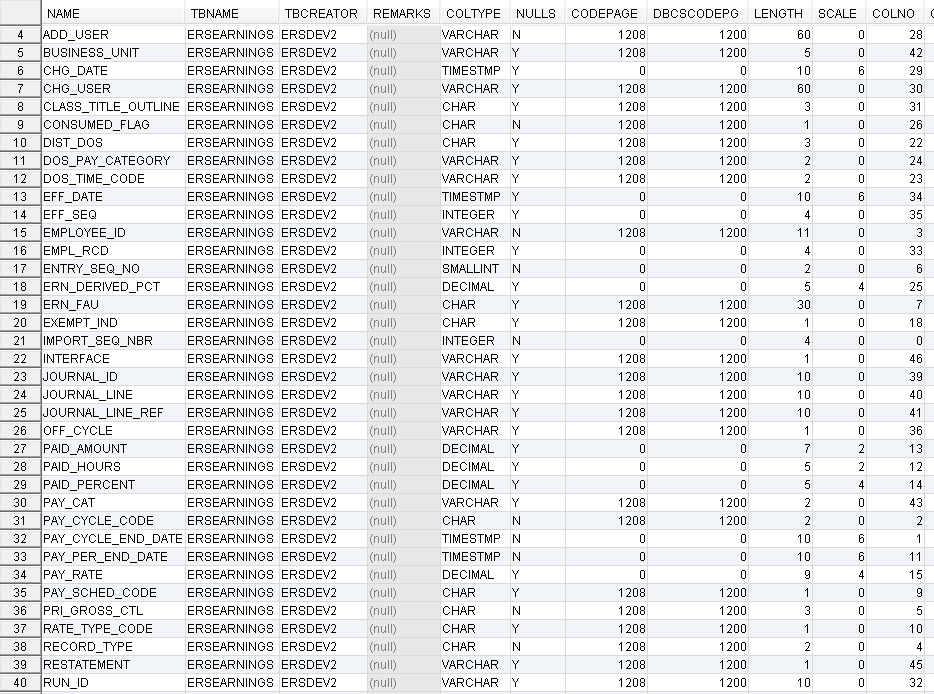
GM: **After** adding the new columns and changing the lengths of some existing columns, here is what the ERSEARNINGS table looked like, first in column number sequence, then in column name sequence.

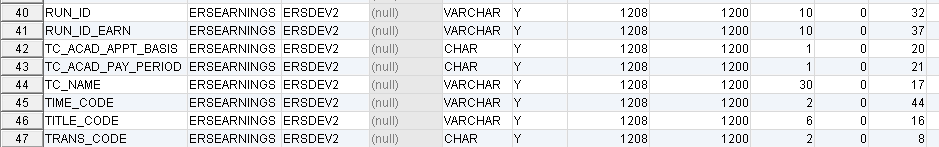
1. **After**. Ordered by column number sequence





2. **After**. Ordered by column name sequence





**Step 6 (continued): Add new Labor Ledger (I-129) columns to ERSEARNINGS table**.

-- Initialize restatement and interface columns

-- perform the following update one year at a time (otherwise transaction log will fill and query will fail)

GM: No rows for 2003-2006. Start with 2007

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2003

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2003

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2004

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2004

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2005

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2005

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2006

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2006

GM: UCSD has data starting only in 2007, not for 2003-2006.

a. I tried to run the 2007 command below, and got -668, 57016.

b. Tried to run reorg. Got 15:51:31 [CALL - 0 row(s), 0.000 secs] [Error Code: -289, SQL State: 57011] DB2 SQL error: SQLCODE: -289, SQLSTATE: 57011, SQLERRMC: ERSTS802

... 1 statement(s) executed, 0 row(s) affected, exec/fetch time: 0.000/0.000 sec [0 successful, 0 warnings, 1 errors]

c. Slava fixed it by extending the tablespace.

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2007

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2007

GM: 2007 is done. Skipped the rest for now since we won't be testing with UCSD.

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2008

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2008

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2009

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2009

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2010

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2010

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2011

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2011

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2012

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2012

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2013

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2013

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2014

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2014

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2015

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2015

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2016

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2016

-- update ersearnings set restatement='N', interface='P' where year(Pay\_Cycle\_End\_Date) = 2017

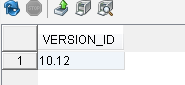
update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2017

update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) > 2017

**Step 6B: Set database version to 11.0**

-- update ersversion set version\_id='11.0'

update ERSVERSION set version\_id='11.0'

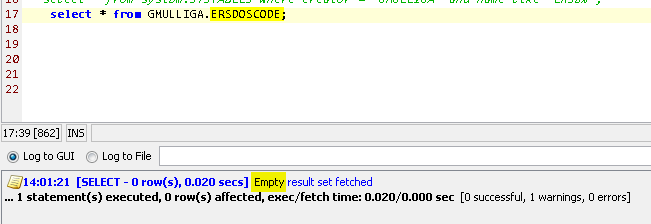


**Step 7: ERSCONSTANTS update**... + ERSEarnings (set default INTERFACE to 'P' PAR)

**Step 7A: Removed because reddundant**

**Step 7B: Update ERS DOS Codes for Off Quarter processing**

GM: the ERSDOSCODE table was empty following the DB load.



-- insert into ERSDosCode values ('9AC','O',getdate(),'\*INSTALL\*',null,null)

insert into ERSDOSCODE values ('9AC','O', current timestamp,'\*INSTALL\*',null,null)

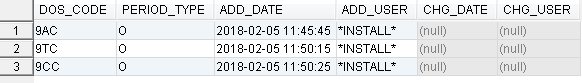
-- insert into ERSDosCode values ('9TC','O',getdate(),'\*INSTALL\*',null,null)

insert into ERSDOSCODE values ('9TC','O', current timestamp,'\*INSTALL\*',null,null)

-- insert into ERSDosCode values ('9CC','O',getdate(),'\*INSTALL\*',null,null)

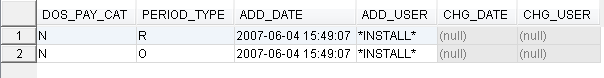
insert into ERSDOSCODE values ('9CC','O', current timestamp,'\*INSTALL\*',null,null)

GM: after inserting the above 3 rows:



**Step 7C: Update ERS DOS Codes for Off Quarter processing**

GM: the ERSPAYCAT table was empty following the DB load.



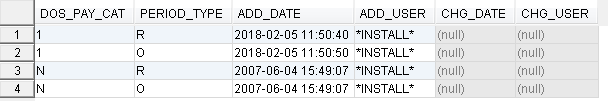
-- insert into ERSPayCat values('1','R',getdate(),'\*INSTALL\*',null,null) -- regular reporting period

insert into ERSPAYCAT values('1','R', current timestamp,'\*INSTALL\*',null,null) -- regular reporting period

-- insert into ERSPayCat values('1','O',getdate(),'\*INSTALL\*',null,null) -- off quarter reporting period

insert into ERSPAYCAT values('1','O', current timestamp,'\*INSTALL\*',null,null) -- off quarter reporting period

GM: after inserting the above 2 rows:

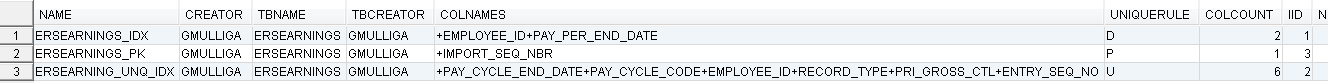


**Step 7D: Create Unique PAR Earnings and I-129 Earnings unique indexes for duplicate earning check**

**I-129**

-- DROP INDEX ERSEarnings. ERS\_EARNINGS\_I129\_UNIQUE

DROP INDEX ERSEARNINGS. ERS\_EARNINGS\_I129\_UNIQUE



GM: Although there is no index on the I-129 fields below, earlier in the document we dropped

ERSEARNING\_UNQ\_IDX, which was on these columns, and never added it back:

PAY\_CYCLE\_END\_DATE

PAY\_CYCLE\_CODE

EMPLOYEE-ID

RECORD\_TYPE

PRI\_GROSS\_CTL

ENTRY\_SEQ\_NO

GM: Should the above index have been dropped? Is it changing for Path?

GM: Now we want to create an I-129 index on a different set of fields from ERSEARNING\_UNQ\_IDX. Here are the I-129 index's fields

Interface

Empl\_Rcd

Eff\_date

Eff\_Seq

Run\_Id

Run\_Id\_Earn

Business\_Unit

Off\_Cycle

Dist\_Dos

Journal\_Id

Journal\_Line

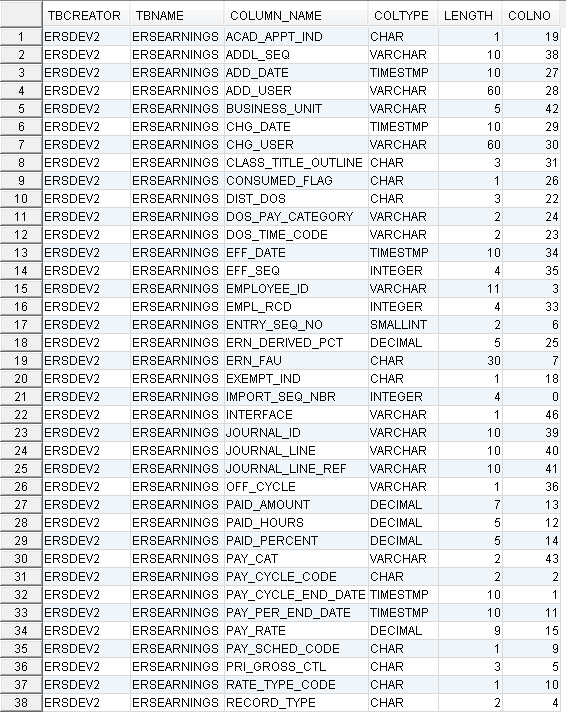
Journal\_Line\_Ref

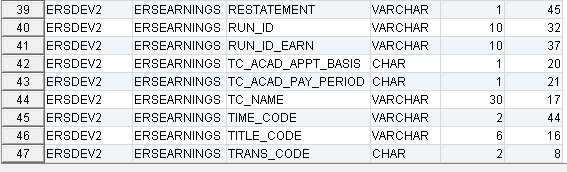
Addl\_Seq

-- CREATE NONCLUSTERED INDEX ERS\_EARNINGS\_I129\_UNIQUE ON ERSEARNINGS ( Interface, Empl\_Rcd, Eff\_date, Eff\_Seq, Run\_Id, Run\_Id\_Earn, Business\_Unit, -- Off\_Cycle, Dist\_Dos, Journal\_Id, Journal\_Line, Journal\_Line\_Ref, Addl\_Seq)

**GM: must remove keyword NONCLUSTERED before it will run** **successfully**.

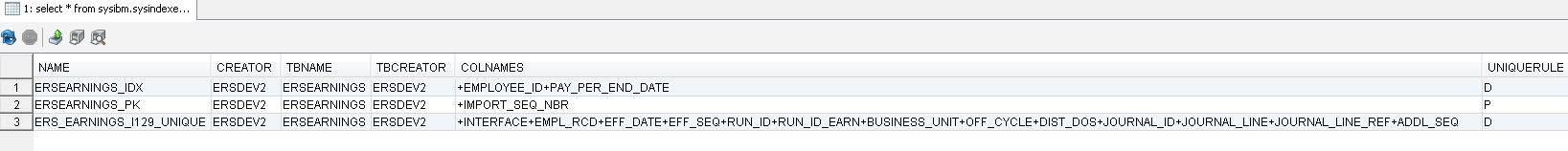
CREATE INDEX ERS\_EARNINGS\_I129\_UNIQUE ON ERSEARNINGS (Interface, Empl\_Rcd, Eff\_date, Eff\_Seq, Run\_Id, Run\_Id\_Earn, Business\_Unit, Off\_Cycle, Dist\_Dos, Journal\_Id, Journal\_Line, Journal\_Line\_Ref, Addl\_Seq)





\*\* note that the UNIQUE keyword has been purposely omitted to eliminate potential problems with historical earning uniqueness

**GM: after creating the new I-129 index, here are all indexes on ERSEARNINGS:**

****

**PAR**

-- DROP INDEX ERSEarnings. ERS\_EARNINGS\_PAR\_UNIQUE

GM: Although there is no index on the I-129 fields below, earlier in the document we dropped

ERSEARNING\_UNQ\_IDX, which was on these columns, and never added it back:

PAY\_CYCLE\_END\_DATE

PAY\_CYCLE\_CODE

EMPLOYEE-ID

RECORD\_TYPE

PRI\_GROSS\_CTL

ENTRY\_SEQ\_NO

GM: Should the above index have been dropped? Is it changing for Path?

GM: Now we want to create a ERS\_EARNINGS\_PAR\_UNIQUE index on a different set of fields from ERSEARNING\_UNQ\_IDX. Here are the PAR index's fields:

Interface

Pay\_Cycle\_End\_Date

Pay\_Cycle\_Code

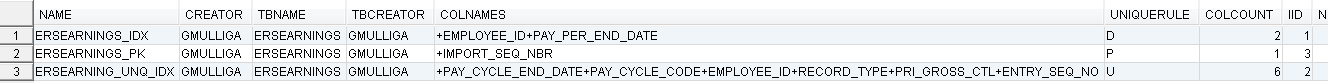
Record\_Type

Pri\_Gross\_Ctl

Entry\_Seq\_No

GM: see below. There's no index named ERS\_EARNINGS\_PAR\_UNIQUE but there is one named

ERSEARNING\_UNQ\_IDX

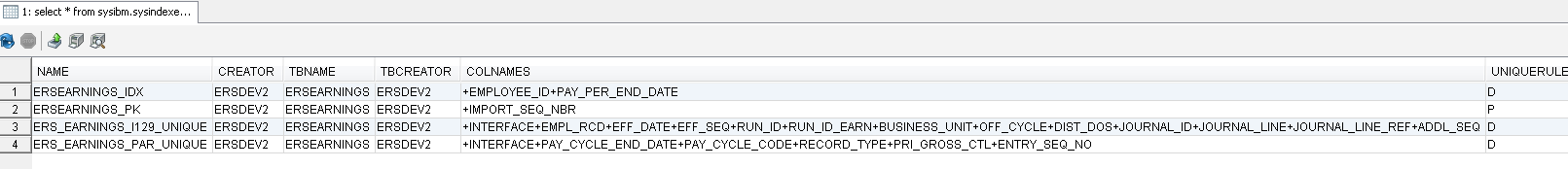


-- CREATE NONCLUSTERED INDEX CREATE INDEX ERS\_EARNINGS\_PAR\_UNIQUE ON ERSEARNINGS(Interface, Pay\_Cycle\_End\_Date, Pay\_Cycle\_Code, Record\_Type, Pri\_Gross\_Ctl, Entry\_Seq\_No)

CREATE INDEX CREATE INDEX ERS\_EARNINGS\_PAR\_UNIQUE ON ERSEARNINGS(Interface, Pay\_Cycle\_End\_Date, Pay\_Cycle\_Code, Record\_Type, Pri\_Gross\_Ctl, Entry\_Seq\_No)

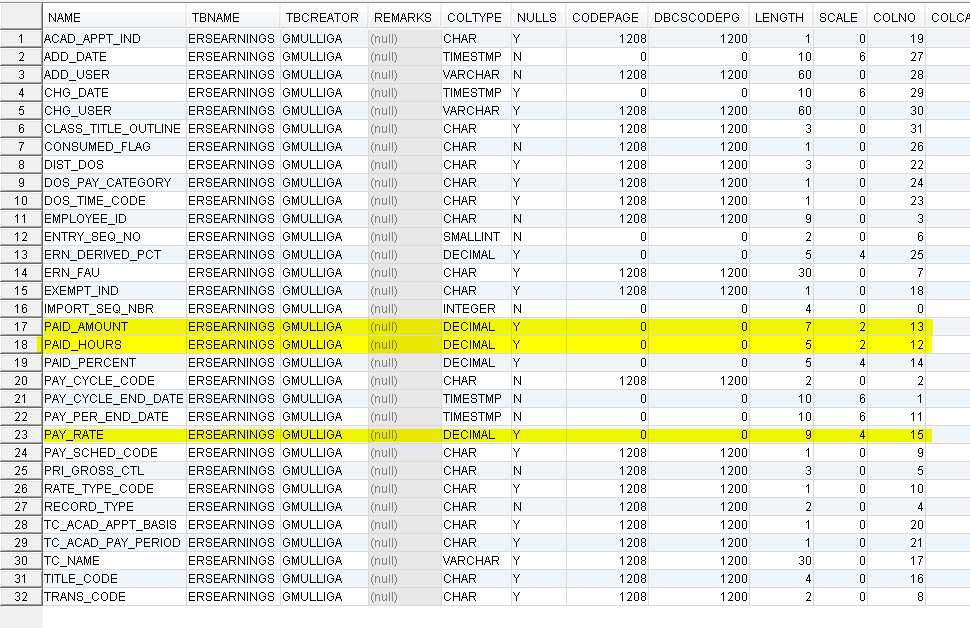
\*\* note that the UNIQUE keyword has been purposely omitted to eliminate potential problems with historical earning uniqueness

**GM: after creating the new I-129 index, here are all indexes on ERSEARNINGS:**

****

**Step 7E: Expand ERSEarnings Paid\_Amount, Paid\_Hours, and Pay\_Rate columns**

GM: before changing these 3 columns, they looked like this:



GM: now change the 3 columns with these commands:

-- ALTER TABLE ERSEarnings ALTER COLUMN Paid\_Hours decimal(6,2) NOT NULL;

ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Hours SET DATA TYPE decimal(6,2)

ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Hours SET NOT NULL;

-- ALTER TABLE ERSEarnings ALTER COLUMN Paid\_Amount decimal(12,2) NOT NULL;

ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Amount SET DATA TYPE decimal(12,2);

ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Amount SET NOT NULL;

**&&& Skip for now**

-- ALTER TABLE ERSEarnings ALTER COLUMN Pay\_Rate decimal(14,4) NOT NULL;

ALTER TABLE ERSEARNINGS ALTER COLUMN Pay\_Rate SET DATA TYPE decimal(14,4);

ALTER TABLE ERSEARNINGS ALTER COLUMN Pay\_Rate SET NOT NULL;

a. The above failed with [Error Code: -20054, SQL State: 55019] DB2 SQL error: SQLCODE: -20054, SQLSTATE: 55019, SQLERRMC: ERSDEV2.ERSEARNINGS

**Step 7F: Create six digit UCPATH Department ID’s in ERS Scheduler.**

**GM: skip this. Different from UCLA. UCLA's scheme is more complicated. UCSD doesn’t use ERSSchDeptBU.**

**ERSSCHDEPT is empty.**

-- select \* into ERSSchDeptBU from ERSSchDept

CREATE TABLE ERSDEV2.ERSSCHDEPTBU AS (

SELECT \*

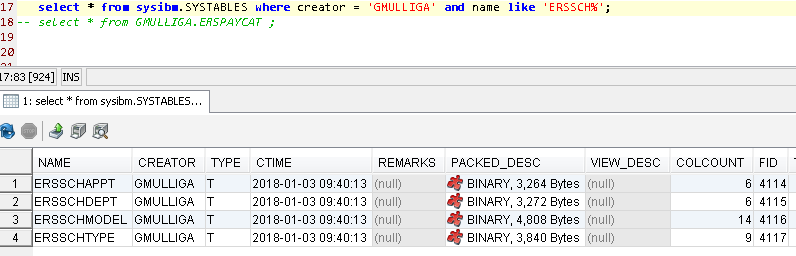
FROM ERSDEV2.ERSSCHDEPT

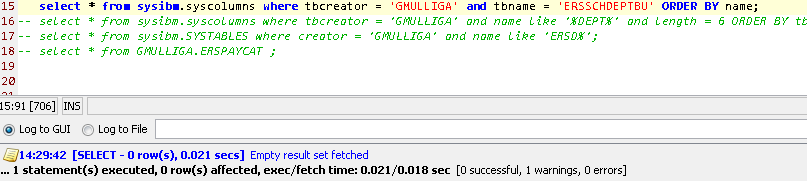
) WITH NO DATA

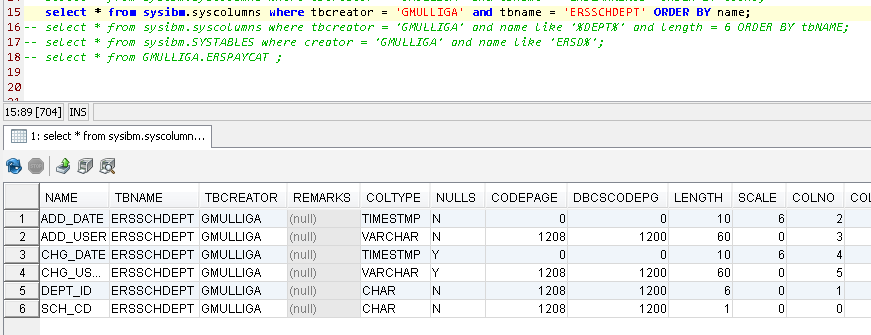
INSERT INTO ERSSchDeptBU

(SELECT \* FROM ERSSchDept)

GM: Target table ERSSCHDEPTBU doesn’t exist, but there is a source table named ERSSCHDEPT. Is ERSSCHDEPTBU being newly added by this script or other means?



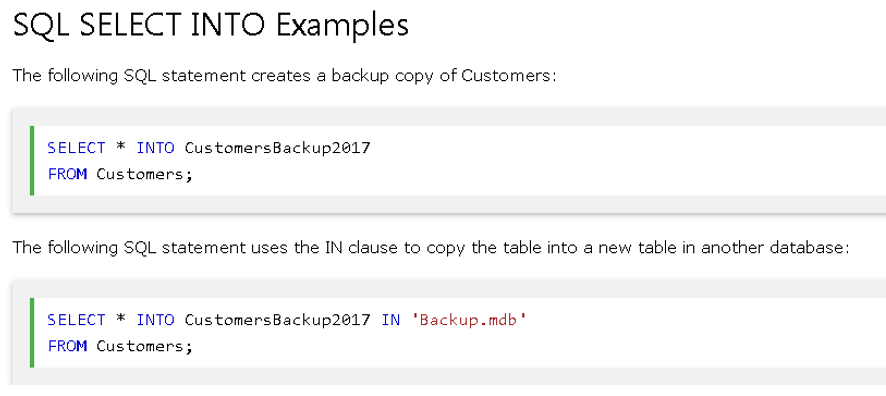




BTW, select \* into ERSSchDeptBU from ERSSchDept

is a legal SQL pattern:

select \* into T1 from T2



Here are the patterns I've seen:

INSERT INTO PB7DA.PPPBRR

(SELECT \* FROM PBUDA.PPPBRR

WHERE BRR\_RETIRE\_DESCRIP LIKE '%u%');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Bottom of Da

**Step 7G: Add Non-Academic Appointment Indicators (3,B,C)**

**GM: skip this. Different from UCLA. UCLA's scheme is more complicated. UCSD doesn’t use ERSPERSPGMCODE.**

-- insert into ERSPersPgmCode values('4','3',getdate(), 'sys.admin',getdate(),'sys.admin')

insert into ERSPERSPGMCODE values('4','3', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')

GM: The above is inserting 6 cols? Don;'t know what 6 cols will be pop'd in a table with more cols. Here's the table:

No, IT ONLY HAS 6.























-- insert into ERSPersPgmCode values('4','B',getdate(), 'sys.admin',getdate(),'sys.admin')

insert into ERSPersPgmCode values('4','B',getdate(), 'sys.admin',getdate(),'sys.admin')

GM: The above is inserting 6 cols. Don;'t know what 6 cols will be pop'd in a table with more cols.

-- insert into ERSPersPgmCode values('4','C',getdate(), 'sys.admin',getdate(),'sys.admin')

insert into ERSPersPgmCode values('4','C',getdate(), 'sys.admin',getdate(),'sys.admin')

GM: The above is inserting 6 cols. Don;'t know what 6 cols will be pop'd in a table with more cols.

**\*\* NOTE – 4 might not be the ‘non-academic’ identifier for hosted campuses**

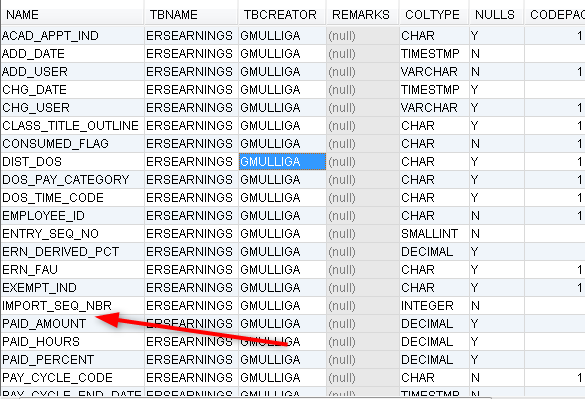
**Step 7H: Update Journal\_Line\_Ref for I-129 records (set to “PAYROLL”)**

**&&& skip for now, needs reorg and cant do reorg cause of 57011**

-- update ERSEarnings set Journal\_Line\_Ref ='PAYROLL' where interface='L'

update ERSEARNINGS set Journal\_Line\_Ref ='PAYROLL' where interface='L'

GM: the ERSEARNINGS table doesn't have a column JOURNAL\_LINE\_REF. Here's the tabhle:



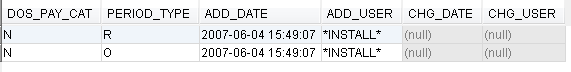
**Step 7I: Add UCPATH ‘Leave’ (sick, sebatical,….) as effort bearing to ERSPayCat table.**

**UCPATH PayCat ‘2’ is not used by PPS so no PPS impact.**

**PPS uses ‘L’ witch is not effort bearing**

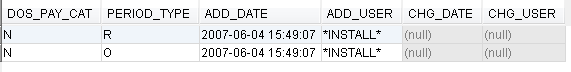
-- insert into ERSPayCat values('2','R',getdate(), 'sys.admin',getdate(),'sys.admin')

insert into ERSPAYCAT values('2','R', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')



-- insert into ERSPayCat values('2','O',getdate(), 'sys.admin',getdate(),'sys.admin')

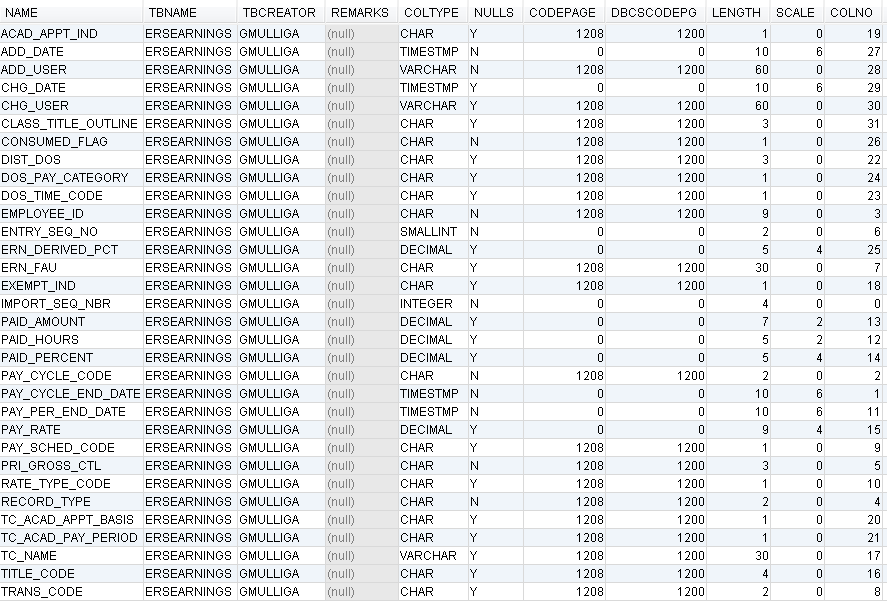
insert into ERSPAYCAT values('2','O', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')



**Step 7J: Add ‘Over The Cap’ to ERSEarnings table.**

**--** alter table ERSEarnings add Over\_The\_Cap varchar(10)

alter table ERSEARNINGS add Over\_The\_Cap varchar(10)

****

**Step 7K: ERSDOSCODE table and ERSPAYCAT table are now indexed using ‘Interface’**

**(same codes in PPS and I-129 are used, but identify ‘Effort’ differently)**

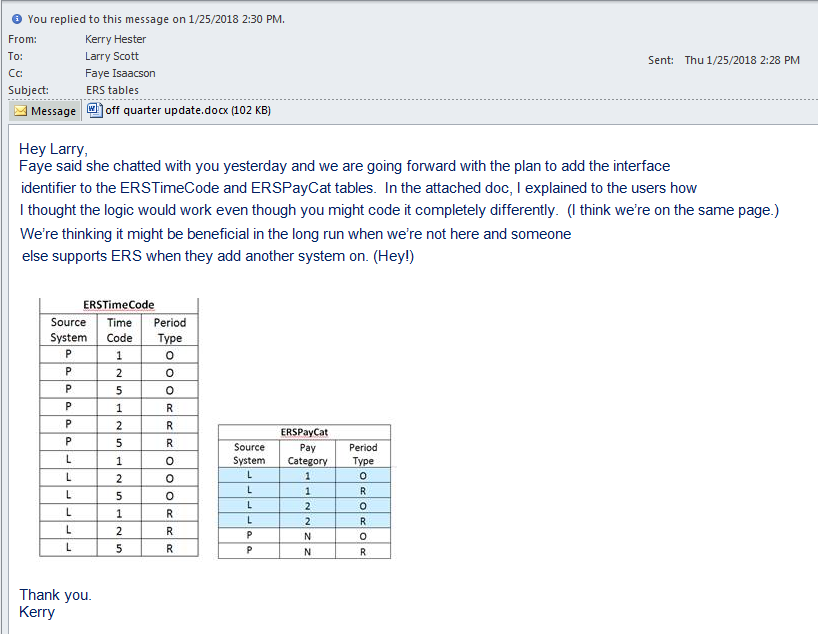
-- alter table **ERSTIMECODE** add Interface varchar(1)

alter table **ERSTIMECODE** add Interface varchar(1)

-- alter table **ERSPAYCAT** add Interface varchar(1)

alter table **ERSPAYCAT** add Interface varchar(1)

**This needs to be confirmed, but setup these tables as shown below….**



-- MS SQL Commands used to build the above tables (primary key violations)

**(GM: watch these tics, the wrong tics can cause -206**

Update ERSTimeCode set interface = 'P'

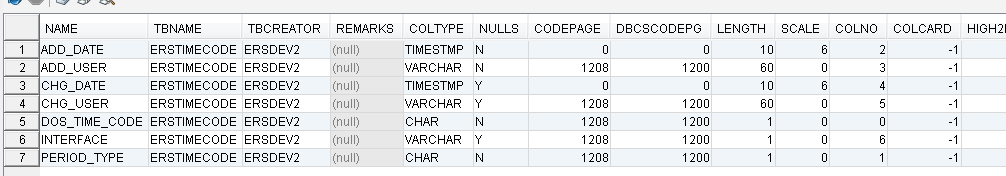
ALTER TABLE ERSTimeCode DROP CONSTRAINT ERSTimeCode\_PK

**Gm: NO NEED to set varchar. It was done earlier**

ALTER TABLE ERSTimeCode ALTER COLUMN Interface Varchar(1) NOT NULL

**GM: still need to set not null**

ALTER TABLE ERSTimeCode ALTER COLUMN Interface SET NOT NULL



-- ALTER TABLE ERSTimeCode ADD CONSTRAINT ERSTimeCode\_PK

-- PRIMARY KEY (DOS\_Time\_Code, Period\_Type, Interface)

ALTER TABLE ERSTimeCode ADD CONSTRAINT ERSTimeCode\_PK

PRIMARY KEY (DOS\_Time\_Code, Period\_Type, Interface)

insert into ERSTimeCode values ('1','O', CURRENT TIMESTAMP, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('2','O', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('5','O', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('1','R', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('2','R', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('5','R', current timestamp,'\*UPGRADE\*',null,null,'L')

-- Update ERSPayCat set interface = ‘P’

Update ERSPayCat set interface = 'P'

-- ALTER TABLE ERSPayCat DROP CONSTRAINT ERSPayCat\_PK

ALTER TABLE ERSPayCat DROP CONSTRAINT ERSPayCat\_PK

- ALTER TABLE ERSPayCat ALTER COLUMN Interface Varchar(1) NOT NULL

ALTER TABLE ERSPayCat ALTER COLUMN Interface SET DATA TYPE Varchar(1)

ALTER TABLE ERSPayCat ALTER COLUMN Interface SET NOT NULL

--ALTER TABLE ERSPayCat ADD CONSTRAINT ERSPayCat\_PK

-- PRIMARY KEY (DOS\_Pay\_Cat, Period\_Type, Interface)

ALTER TABLE ERSPayCat ADD CONSTRAINT ERSPayCat\_PK

PRIMARY KEY (DOS\_Pay\_Cat, Period\_Type, Interface)

-- Update ERSPayCat set interface = 'L' where DOS\_Pay\_Cat='1'

Update ERSPayCat set interface = 'L' where DOS\_Pay\_Cat='1'

-- insert into ERSPayCat values ('2','O',getdate(),'\*UPGRADE\*',null,null,'L')

insert into ERSPayCat values ('2','O', CURRENT TIMESTAMP,'\*UPGRADE\*',null,null,'L')

-- insert into ERSPayCat values ('2','R',getdate(),'\*UPGRADE\*',null,null,'L')

insert into ERSPayCat values ('2','R', CURRENT TIMESTAMP,'\*UPGRADE\*',null,null,'L')

**Step 7L: Add ‘Over The Cap’ and ‘ Salary Cap’**

alter table ersearnings add Over\_The\_Cap varchar(10) NULL

alter table ersearnings add Cap\_Rate Decimal (14, 4) NULL

update ersearnings set cap\_rate=0

**Step 8: Set system generate key values for ERSEarnings table inserts by PAR & I-129 loads**

**GM: skip for now, ERSEARNINGS needs a reorg and can't 'cause -668**

--select max(Import\_Seq\_Nbr) from ERSEarnings -- > example 113xxxxx

select max(Import\_Seq\_Nbr) from ERSEarnings

**GM: skip for now, ERSEARNINGS needs a reorg and can't 'cause -668**

--ALTER TABLE ERSEarnings ALTER COLUMN Import\_Seq\_Nbr RESTART WITH 11400000

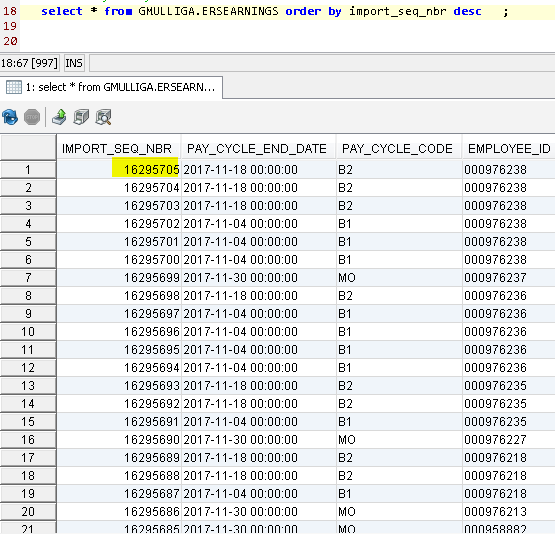
ALTER TABLE ERSEarnings ALTER COLUMN Import\_Seq\_Nbr RESTART WITH 17000000

GM: why are the above two lines commented and are they needed?

A: Yes they are needed. Inserts into EARNINGS table automatically increment IMPORT\_SEQ\_NBR. Must guarantee unique values. Must start with a number bigger than any existing numbers, so 17,000,000

GM: currently the max sequence number is 16295705. What would the RESTART command, assuming it will be run, do to the table?

A: make it 17,000000



**Step 9: Set ERS sys.admin password allowing user to login to ERS**

(DO THIS FROM ERS LOGIN SCREEN – S.B. PERFORMED BY .WAR FILE INSTALLER)

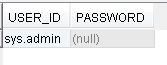
-- update ~~ersdb\_20150409\_data\_for\_DEV\_and\_QA\_use.dbo.~~ ERSUser

Update ERSUSER (incomplete; see below)

**!!! GM: don't run. See 'DO THIS' above**

-- set Password = (select password from ersdb\_20150116\_original.dbo.ERSUser where User\_ID = 'sys.admin')

Update ERSUSER set Password = (select password from ERSUser where User\_ID = 'sys.admin')



**GM: skip all 10.x for now**

**Step 10: Archive Migration from DTO format to PDF format**

This code is contained only in ERS 10.12 B010

The archive code in ERS 10.12 B010 now builds archives in PDF format, not DTO format

All archive code in releases prior to ERS 10.12 B010 build archives in DTO (data) format

The ERS 10.12 B010 batch program to be run is **InterfaceArchiveReportMigation10to11.java**

**Overview**

ERS 10.X created archives in DTO format. ERS 11.0 creates archives in PDF format

This step changes the ERS 10.x archives from a **DTO** (data transfer object) to a **PDF** format

(necessary to solve the issue of archiving from one version of ERS and restoring to another)

The ERS Archive tables are **ERSARCHIVEDREPORT** and **ERSARCHIVEDFS**.

All archive data is contained in ERSARCHIVEDREPORT and ERSARCHIVEDFS

**\*\*\* IMPORTANT\*\*\***

**Make sure you run this process in a QA environment and verify the archives in a QA environment.**

**It would be a good precaution, to make a long term backup of the production database.**

**Process:**

These are the steps necessary to convert ERS DTO archives to PDF archives:

**10.1:** Check for data in database table **ERSARCHIVEDREPORT.**

If this table is empty, skip this step. No archive migration is necessary.

**10.2:** Check the year of the archives

<2018 is DTO format. 2018+ is PDF format.

This will tell you if the archives are in a DTO format or PDF format.

If DTO format, migration is necessary. If PDF format, no migration is necessary.

Run this SQL: **select distinct year(archived\_date) from ERSARCHIVEDREPORT**

If you only see year 2018 (or greater), your archives are in PDF format and skip the migration.

If you see year less than 2018, the migration required.

**10.3:** Preparing for migration

Migration is performed in a non-production environment to minimize ERS production downtime.

Archive migration may run for days.

Load the ERS 10.12 B010 war file to your QA environment

Copy your production ERS database to your QA environment

Verify your QA ERS database connection and ERS code (ERS signon screen display version 10.12 B010)

**10.4:** Run the migration program

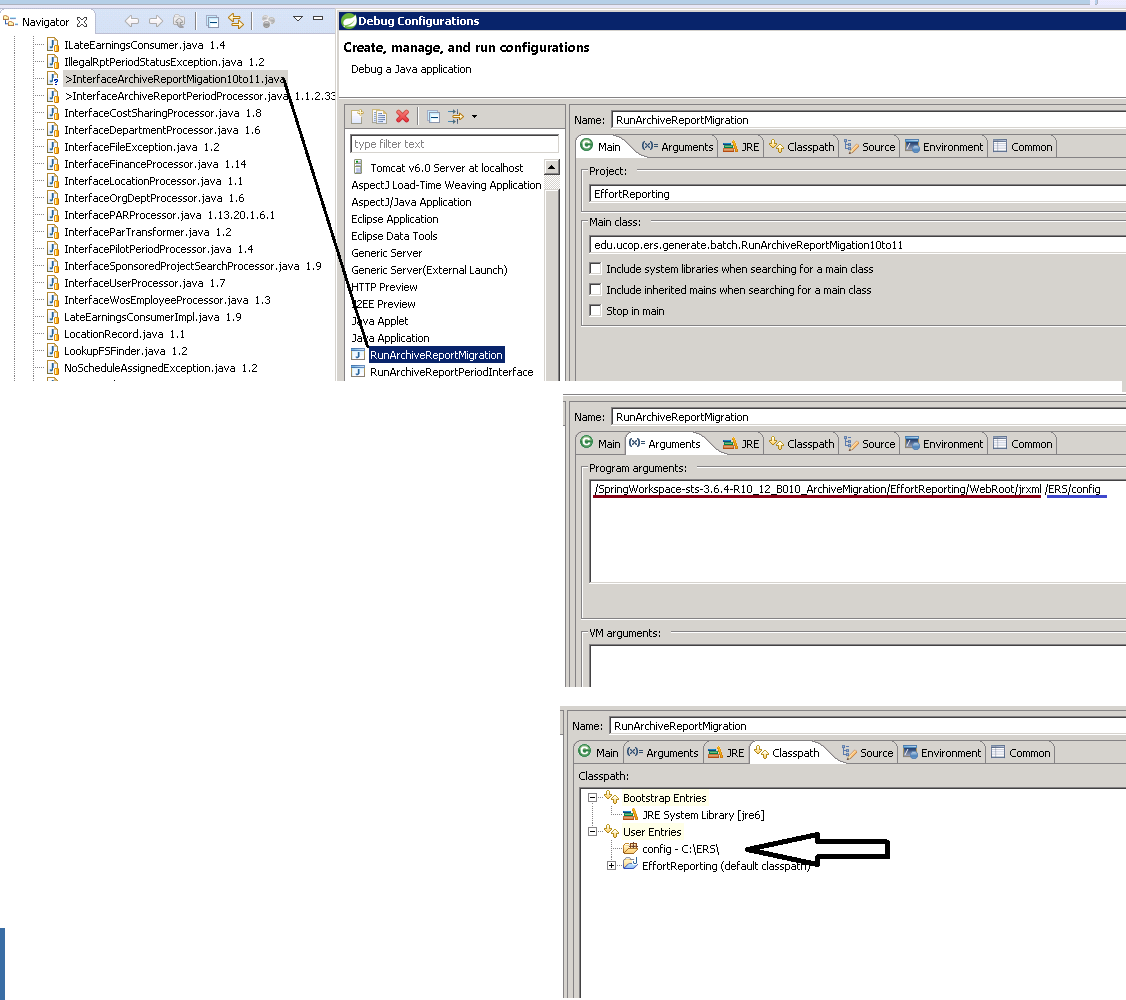
In you QA environment, run ERS batch program InterfaceArchiveReportMigation10to11.java

It is necessary to set your ERS config directory in the classpath (shown in screenshot)

It also is necessary to pass two parameters to the archive migration process.

1: The path to the ERS Jasper report objects (red underline) . Explode you ers.war file to a temp directory

2: The path to your ExternalizedString.xml file (blue underline)



**10.5: Replace production DTO archives with QA PDF Archives**

Tables **ERSARCHIVEDREPORT**and **ERSARCHIVEDFS** in QA environment **REPLACE** the same tables in production ERS environment. Copy/Replace these tables from QA to PROD

**Archive migration program execution comments**

The migration program can easily consume all memory on your server causing the process to crash.

To prevent ‘out-of-memory errors’, the migration program processes 100,000 archived effort reports at a time and then stops to releases memory. The migration process commits updates in 5000 archive increments.

As the archive migration run, the DTO pdf is read from table ERSAarchivedReport and converted to PDF (using the JasperReport objects and the campuses customizations in ErternalizedStrings.xml). Once the PDF is built, the DTO in ERSArchivedReports is overlaid with the PDF . The same process is used for the archive payroll report. The process continues until all ERSArchivedReport rows are processed or 100,000 rows are processed.

Larger campuses may have millions of rows in the ERSArchivedReport table, making it necessary to run the migration process 10+ times. Each iteration of the migration process will likely take a few days.

Run this SQL:

**select year(archived\_date) , count(\*) from ERSARCHIVEDREPORT group by year(archived\_date)**

If you see any archive dates < 2018, run the process again

Once all archive dates < 2018 are cleared, the ERS Archive Migration is complete.

**Performance:**

When this process was run for UCSB in QA (Jan 2018), it took 15 hours to migrate 100,000 archived effort reports.

**Step 11:** **Closing comments**

It is highly recommended that you long term archive a copy of the ERS 10.12 production database

If any issues arise after the UCPATH implementation, it may be necessary to load the ERS 10.12 system in a QA environment to retrieve or repair ERS data.

**Done with ERS 11.0 Upgrade !**

**Appendix 1**

Programming Note 1:

Any changes to the ERSEarnings table need to be replicated in ERS 11.0 Java class

DrillDownQuery.java, which is used by DrillDownBuilder.java to construct the payroll report.

See ERSMappings folder, files payrollReport2.png and payrollReport3.png

Programming Note 2:

**The following methods should be part of BaseEmployee.java but lost when employee.hbm changed.**

Methods were lost when employee.hmb is compiled by hibernate synchronizer

Methods should not be part off employee.hbm

I manually added it back to BaseEmployee.java.

It is called as part of a JUnit test

public void addToUsers (edu.ucop.ers.domain.User user) {

if (null == getUsers()) setUsers(new java.util.HashSet<edu.ucop.ers.domain.User>());

getUsers().add(user);

}

public void addToEarnings (edu.ucop.ers.domain.Earning earning) {

if (null == getEarnings()) setEarnings(new java.util.HashSet<edu.ucop.ers.domain.Earning>());

getEarnings().add(earning);

}

public void addToReportHeaders (edu.ucop.ers.domain.ReportHeader reportHeader) {

if (null == getReportHeaders()) setReportHeaders(

new java.util.HashSet<edu.ucop.ers.domain.ReportHeader>());

getReportHeaders().add(reportHeader);

}

public void addToCostSharingCommitments (edu.ucop.ers.domain.CostSharingCommitment

costSharingCommitment) {

if (null == getCostSharingCommitments()) setCostSharingCommitments(new

java.util.HashSet<edu.ucop.ers.domain.CostSharingCommitment>());

getCostSharingCommitments().add(costSharingCommitment);

}

public void addToPiProjects (edu.ucop.ers.domain.PrincipalInvestigator principalInvestigator) {

if (null == getPiProjects()) setPiProjects(new

java.util.HashSet<edu.ucop.ers.domain.PrincipalInvestigator>());

getPiProjects().add(principalInvestigator);

}

Programming Note 3: LaborLedger Date Format

Labor Ledger record date format sometimes changes requiring format change to

ERSCONSTANTS.java - public static final String DATE\_FORMAT\_PAR = "MMddyyyy";

Programming Note 4: EarningMatcher.java - method MatchStatus

Checks PayCategory, DOS Code, amd Time Code to determine if an earning should be certified.

This check should be skipped for ERS 11.0

Notice that column 'interface' (P=PAR, L=LaborLedger) has been added to ERSEarnings table.

and code to populate in InterfaceLaborLedgerProcessor, and InterfacePARProcessor

\*\* method MatchStatus determines if a LaborLedger earning should be certified only if derived% >0. This may not be sufficient.

Configuation Note 1: ERSConfig.xml - used in ConfigLoader.java

// which payroll system being used (PPS or PeopleSoft). Default to PeopleSoft

String payrollSystemPps = (String) cfg.getProperty(

ERSConstants.***CFG\_OPTIONS\_PAYROLL\_SYSTEM\_PPS***);

**if** ( payrollSystemPps != **null** ) {

ctx.setAttribute( ERSConstants.***PAYROLL\_SYSTEM\_PPS***,

**new** Boolean( bypass11Over12Check ) );

} **else** {

ctx.setAttribute( ERSConstants.***PAYROLL\_SYSTEM\_PPS***,

ERSConstants.***FALSE*** );

}

see ERSConfig.xml

<bypass-11over12-appointment-check>true</bypass-11over12-appointment-check>

JUNIT Note: (JUnits run correctly in EffortReporting R10\_11\_B001, but not EffortReporting R11\_Support)

1: Latheef and Bascar's work

RunUcpathEmployeeConversionTest.java

UcpathEmployeeConversionProcessorTest.java

2: See step 1: Some JUnit required methods are missing from employee.hbm

Methods were lost when employee.hmb is compiled by hibernate synchronizer

Methods should not be part off employee.hbm

3: JUnit TestFundingSrcUniquenessTest is looking for file Z:/interfaces/Test

Files/UCD/fin\_systems\_interface.31-MAR-2005

File can not be found so JUnit is commented out.

4: JUnit class UcpathEmployeeConversionProcessorTest method testAddEmployee does not run....

has bugs, commented out

5: JUnit RunUcpathEmployeeConversionTest fails - commented out

6: JUnit SearchDeadlockFinder pukes all-over itself

see class: AbstractSavableSearchImpl method:addToCachedSelectedItemsSet( List moreKeys ) {

error is: java.lang.ClassCastException: java.lang.NoSuchMethodException: Unknown property 'dummy'

The following methods have been commented out

- testSaveAndDelete()

- testForDeadlock()

- testSaveAndDeleteAdHoc()

7: JUint class SearchManagerImpl method testCopySearch - added try / catch block to force success

8; JUnit: InterfacePARProcessorTest

Notice junit test class "extends HibernateUnitTester" This sets up the database

\* insert\_base\_test\_fixture\_data.sql

\* insert\_standard\_test\_fixture\_data.sql

9: JUnit: InterfaceLaborLedgerTest - needs to be developed

**Appendix 2:**

These steps were run at UCLA to remove Effort Reports and Earnings prior to 2014.

This shunk the size of the UCLA ERS earnings table from 35M earnings to 12M earnings.

Database size was reduced approximately by half.

This was needed for UCLA database to ‘fit’ on UCOP ERS DEV DB server

delete from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ersearnings

where year(pay\_per\_end\_date) < 2014

and not import\_seq\_nbr in

(select import\_seq\_nbr from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ERSEarningsVersion V)

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  1

- A cursor with the name 'C1' does not exist. [SQLSTATE 34000] (Error 16916) – This is OK, do not stop.

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  2

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  3

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  4

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  5

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  6

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  7

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  8

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  9

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  10

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  11

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  12

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  13

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  14

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  15

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  16

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  17

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  18

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  19

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  20

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  21

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  22

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  23

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  24

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  25

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  26

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  27

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  28

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  29

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  30

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  31

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  32

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  33

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  34

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  35

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  36

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  37

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  38

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  39

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  40

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  41

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  42

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  43

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  44

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  45

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  46

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  47

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  48

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  49

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  50

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  51

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  52

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  53

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  54

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  55

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  56

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  57

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  58

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  59

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  60

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  61

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  62

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  63

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  64

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  65

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  66

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  67

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  68

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  69

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  70

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  71

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  72

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  73

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  74

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  75

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  76

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  77

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  78

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  79

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  80

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  81

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  82

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  83

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  84

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  85

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  86

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  87

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  88

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  89

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  90

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  91

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  92

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  93

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  94

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  95

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  96

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  97

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  98

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  99

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  100

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  101

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  102

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  103

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  104

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  105

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  106

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  107

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  108

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  109

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  110

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  111

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  112

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  113

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  114

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  115

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  116

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  117

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  118

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  119

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  120

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  121

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  122

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  123

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  124

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  125

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  126

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  127

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  128

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  129

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  130

delete from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ersearnings

where year(pay\_per\_end\_date) < 2014

and not import\_seq\_nbr in

(select import\_seq\_nbr from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ERSEarningsVersion V)

**Appendix 1**

Programming Note 1:

Any changes to the ERSEarnings table need to be replicated in ERS 11.0 Java class

DrillDownQuery.java, which is used by DrillDownBuilder.java to construct the payroll report.

See ERSMappings folder, files payrollReport2.png and payrollReport3.png

Programming Note 2:

**The following methods should be part of BaseEmployee.java but lost when employee.hbm changed.**

Methods were lost when employee.hmb is compiled by hibernate synchronizer

Methods should not be part off employee.hbm

I manually added it back to BaseEmployee.java.

It is called as part of a JUnit test

public void addToUsers (edu.ucop.ers.domain.User user) {

if (null == getUsers()) setUsers(new java.util.HashSet<edu.ucop.ers.domain.User>());

getUsers().add(user);

}

public void addToEarnings (edu.ucop.ers.domain.Earning earning) {

if (null == getEarnings()) setEarnings(new java.util.HashSet<edu.ucop.ers.domain.Earning>());

getEarnings().add(earning);

}

public void addToReportHeaders (edu.ucop.ers.domain.ReportHeader reportHeader) {

if (null == getReportHeaders()) setReportHeaders(

new java.util.HashSet<edu.ucop.ers.domain.ReportHeader>());

getReportHeaders().add(reportHeader);

}

public void addToCostSharingCommitments (edu.ucop.ers.domain.CostSharingCommitment

costSharingCommitment) {

if (null == getCostSharingCommitments()) setCostSharingCommitments(new

java.util.HashSet<edu.ucop.ers.domain.CostSharingCommitment>());

getCostSharingCommitments().add(costSharingCommitment);

}

public void addToPiProjects (edu.ucop.ers.domain.PrincipalInvestigator principalInvestigator) {

if (null == getPiProjects()) setPiProjects(new

java.util.HashSet<edu.ucop.ers.domain.PrincipalInvestigator>());

getPiProjects().add(principalInvestigator);

}

Programming Note 3: LaborLedger Date Format

Labor Ledger record date format sometimes changes requiring format change to

ERSCONSTANTS.java - public static final String DATE\_FORMAT\_PAR = "MMddyyyy";

Programming Note 4: EarningMatcher.java - method MatchStatus

Checks PayCategory, DOS Code, amd Time Code to determine if an earning should be certified.

This check should be skipped for ERS 11.0

Notice that column 'interface' (P=PAR, L=LaborLedger) has been added to ERSEarnings table.

and code to populate in InterfaceLaborLedgerProcessor, and InterfacePARProcessor

\*\* method MatchStatus determines if a LaborLedger earning should be certified only if derived% >0. This may not be sufficient.

Configuation Note 1: ERSConfig.xml - used in ConfigLoader.java

// which payroll system being used (PPS or PeopleSoft). Default to PeopleSoft

String payrollSystemPps = (String) cfg.getProperty(

ERSConstants.***CFG\_OPTIONS\_PAYROLL\_SYSTEM\_PPS***);

**if** ( payrollSystemPps != **null** ) {

ctx.setAttribute( ERSConstants.***PAYROLL\_SYSTEM\_PPS***,

**new** Boolean( bypass11Over12Check ) );

} **else** {

ctx.setAttribute( ERSConstants.***PAYROLL\_SYSTEM\_PPS***,

ERSConstants.***FALSE*** );

}

see ERSConfig.xml

<bypass-11over12-appointment-check>true</bypass-11over12-appointment-check>

JUNIT Note: (JUnits run correctly in EffortReporting R10\_11\_B001, but not EffortReporting R11\_Support)

1: Latheef and Bascar's work

RunUcpathEmployeeConversionTest.java

UcpathEmployeeConversionProcessorTest.java

2: See step 1: Some JUnit required methods are missing from employee.hbm

Methods were lost when employee.hmb is compiled by hibernate synchronizer

Methods should not be part off employee.hbm

3: JUnit TestFundingSrcUniquenessTest is looking for file Z:/interfaces/Test

Files/UCD/fin\_systems\_interface.31-MAR-2005

File can not be found so JUnit is commented out.

4: JUnit class UcpathEmployeeConversionProcessorTest method testAddEmployee does not run....

has bugs, commented out

5: JUnit RunUcpathEmployeeConversionTest fails - commented out

6: JUnit SearchDeadlockFinder pukes all-over itself

see class: AbstractSavableSearchImpl method:addToCachedSelectedItemsSet( List moreKeys ) {

error is: java.lang.ClassCastException: java.lang.NoSuchMethodException: Unknown property 'dummy'

The following methods have been commented out

- testSaveAndDelete()

- testForDeadlock()

- testSaveAndDeleteAdHoc()

7: JUint class SearchManagerImpl method testCopySearch - added try / catch block to force success

8; JUnit: InterfacePARProcessorTest

Notice junit test class "extends HibernateUnitTester" This sets up the database

\* insert\_base\_test\_fixture\_data.sql

\* insert\_standard\_test\_fixture\_data.sql

9: JUnit: InterfaceLaborLedgerTest - needs to be developed

**Appendix 2:**

These steps were run at UCLA to remove Effort Reports and Earnings prior to 2014.

This shunk the size of the UCLA ERS earnings table from 35M earnings to 12M earnings.

Database size was reduced approximately by half.

This was needed for UCLA database to ‘fit’ on UCOP ERS DEV DB server

delete from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ersearnings

where year(pay\_per\_end\_date) < 2014

and not import\_seq\_nbr in

(select import\_seq\_nbr from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ERSEarningsVersion V)

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  1

- A cursor with the name 'C1' does not exist. [SQLSTATE 34000] (Error 16916) – This is OK, do not stop.

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  2

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  3

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  4

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  5

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  6

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  7

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  8

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  9

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  10

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  11

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  12

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  13

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  14

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  15

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  16

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  17

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  18

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  19

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  20

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  21

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  22

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  23

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  24

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  25

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  26

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  27

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  28

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  29

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  30

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  31

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  32

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  33

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  34

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  35

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  36

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  37

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  38

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  39

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  40

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  41

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  42

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  43

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  44

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  45

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  46

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  47

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  48

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  49

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  50

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  51

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  52

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  53

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  54

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  55

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  56

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  57

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  58

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  59

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  60

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  61

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  62

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  63

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  64

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  65

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  66

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  67

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  68

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  69

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  70

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  71

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  72

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  73

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  74

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  75

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  76

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  77

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  78

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  79

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  80

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  81

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  82

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  83

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  84

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  85

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  86

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  87

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  88

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  89

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  90

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  91

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  92

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  93

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  94

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  95

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  96

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  97

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  98

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  99

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  100

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  101

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  102

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  103

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  104

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  105

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  106

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  107

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  108

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  109

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  110

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  111

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  112

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  113

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  114

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  115

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  116

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  117

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  118

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  119

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  120

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  121

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  122

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  123

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  124

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  125

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  126

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  127

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  128

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  129

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  130

delete from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ersearnings

where year(pay\_per\_end\_date) < 2014

and not import\_seq\_nbr in

(select import\_seq\_nbr from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ERSEarningsVersion V)

**Appendix 4 - Common DB2 Error Codes**

SQLCODE

-205: column-name IS NOT A COLUMN OF TABLE table-name

-289: see 57011

-440: see 42884

-601: name already in use

-668: Access problem. Run reorg, then try again.   
-20054 see 55019

SQLSTATE

42601: A character, token, or clause is invalid or missing

42710: name already in use

42703: column-name IS NOT A COLUMN OF TABLE table-name

42704: object not defined

42884: It happened trying to run this command:

insert into ERSDOSCODE values ('9AC','O',getdate(),'\*INSTALL\*',null,null)

Fixed by removing the 'getdate()' and replacing it with 'CURRENT TIMESTAMP' in the above command

42889: The table already has a primary key

57011: Failed during reorg. Tablespace ran out of space. Slava fixed by extending **ERSTS801** tablespace

57016: Access problem. Run reorg, then try again.

55019: It happened trying to run this command:

ALTER TABLE ERSEARNINGS ALTER COLUMN Pay\_Rate SET NOT NULL;

Thought it needed re-org'ing, so ran the reorg command, but got a 57011. Slava had just extended the TS for ERSEARNINGS the day before. I opened a new ticket for him to fix it again.