

```

1 ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY';
2 SET feedback OFF
3 SET echo OFF
4 -----
5 --
6 --          SENTRY_STATS_VALIDATION.SQL
7 --
8 -- FUNCTION: Run some integrity checks prior to inserting into UNLS
9 --
10 -- MODIFICATION HISTORY:
11 --
12 -- PROJECT# TASK   DATE      TECH ACTION
13 -- -----
14 -- PLI32502.31220 20-JUN-2016 JPC  INITIAL CREATION
15 --
16 -- -----
17 SET SERVEROUT ON
18 prompt -----
19
20
21 prompt Beginning VALIDATION Process
22 DECLARE
23  -----
24  WS_POLNUM SUNL.SUNL_WC_POLNUMBER%TYPE;
25  WS_POLEFF SUNL.SUNL_WC_EFFDATE%TYPE;
26  WS_RPTDTE SUNL.SUNL_CL_REPORT_DATE%TYPE;
27  WS_CLSCDE SUNL.SUNL_CL_RVHS_CLASS_CODE%TYPE;
28  WS_REPIND SUNL.SUNL_CL_PREV_REP_IND%TYPE;
29  WS_CVGCDE SUNL.SUNL_CL_LOSS_COVG_CODE%TYPE;
30  WS_NOCLMS SUNL.SUNL_CL_NUMBER_OF_CLAIMS%TYPE;
31  WS_ACCDTE SUNL.SUNL_CL_ACCIDENT_DATE%TYPE;
32  WS_CLMNUM SUNL.SUNL_CL_CLAIM_NUMBER%TYPE;
33  WS_CLSTTS SUNL.SUNL_CL_CLAIM_STATUS%TYPE;
34  WS_INJCDE SUNL.SUNL_CL_INJURY_CODE%TYPE;
35  WS_CATNUM SUNL.SUNL_CL_CAT_NUMBER%TYPE;
36  WS_INDAMT SUNL.SUNL_CL_INDEMNITY_AMOUNT%TYPE;
37  WS_MEDAMT SUNL.SUNL_CL_MEDICAL_AMOUNT%TYPE;
38  WS_FEINNO SUNL.SUNL_CL_SSN%TYPE;
39  WS_UPDTYP SUNL.SUNL_CL_UPDATE_TYPE%TYPE;
40  WS_LSSACT SUNL.SUNL_CL_LOSS_ACT%TYPE;
41  WS_LSSTYP SUNL.SUNL_CL_LOSS_TYPE%TYPE;
42  WS_LSSREC SUNL.SUNL_CL_LOSS_RECOVERY%TYPE;
43  WS_LSSCOV SUNL.SUNL_CL_LOSS_COVERAGE%TYPE;
44  WS_LSSSTL SUNL.SUNL_CL_LOSS_SETTLEMENT%TYPE;
45  WS_STATNO SUNL.SUNL_STAT_GB_STATE_NO%TYPE;
46  WS_HMOTYP SUNL.SUNL_CL_MANG_CARE_ORG_TYPE%TYPE;
47  WS_POBINO SUNL.SUNL_CL_INJ_DESC_CODE_PART%TYPE;
48  WS_NOSINO SUNL.SUNL_CL_INJ_DESC_CODE_NAT%TYPE;
49  WS_COLTNO SUNL.SUNL_CL_INJ_DESC_CODE_CLAUSE%TYPE;
50  WS_OCCDSC SUNL.SUNL_CL_OCCUPATION_DESC%TYPE;
51  WS_REHIND SUNL.SUNL_CL_VOCATION_REHAB_IND%TYPE;
52  WS_LUMPSM SUNL.SUNL_CL_LUMP_SUM_IND%TYPE;
53  WS_FRDIND SUNL.SUNL_CL_FRAUD_CLAIM_IND%TYPE;
54  WS_DEDCDE SUNL.SUNL_CL_DEDUCT_CODE%TYPE;
55  WS_PAIDWI SUNL.SUNL_CL_PAID_INDEMNITY_AMT%TYPE;
56  WS_PAIDWM SUNL.SUNL_CL_PAID_MEDICAL_AMT%TYPE;

```

```

57 WS_PDALAE SUNL.SUNL_CL_ALAE_PAID_AMT%TYPE;
58 WS_INALAE SUNL.SUNL_CL_ALAE_INCURRED_AMT%TYPE;
59 WS_WKLYWG SUNL.SUNL_CL_AVG_WKLY_WAGE%TYPE;
60 WS_INREHB SUNL.SUNL_CL_INCUR_VOC_REHAB%TYPE;
61 WS_GRSINC SUNL.SUNL_CL_CA_GROSS_INCUR%TYPE;
62 WS_DISABL SUNL.SUNL_CL_PERM_DISABL_RATING%TYPE;
63 WS_DEDAMT SUNL.SUNL_CL_DEDUCT_AMT%TYPE;
64 WS_DESDED SUNL.SUNL_CL_DEDUCT_AMT%TYPE;
65 WS_HIRDTE SUNL.SUNL_US_HIREDATE%TYPE;
66 WS_LAWIND VARCHAR2(1);
67 WS_RPTNUM VARCHAR2(1);
68 WS_ENDDTE DATE := '30-SEP-2017';
69 WS_RECUPD DATE;
70 WS_OCCDTE DATE;
71 WS_SUNLID NUMBER;
72 WS_FULLPOLICY VARCHAR2(12);
73 WS_GOOD NUMBER := 0;
74 WS_BAD NUMBER := 0;
75 WS_SKIPPED NUMBER := 0;
76 WS_MONTHLY NUMBER := 0;
77 WS_LENGTH NUMBER(2);
78 WS_CONTROL VARCHAR2(1);
79 --
80 V_ENDDATE VARCHAR2(3);
81 V_RPM NUMBER := 0;
82 V_POL_COUNT NUMBER;
83 V_STATUS VARCHAR2(2);
84 V_SECTN VARCHAR2(8);
85 V_POLDAT VARCHAR2(2);
86 --
87 RD_UNPLID SUNL.SUNL_CL_NO%TYPE;
88 RD_CLAIM BO_RECOVERABLE_DEDUCTIBLE.CLAIM_NUMBER%TYPE;
89 RD_AMT BO_RECOVERABLE_DEDUCTIBLE.DEDUCTIBLE_AMT%TYPE;
90 RD_RPTAMT SUNL.SUNL_CL_DEDUCT_AMT%TYPE;
91
92 --
93 UPDATE_SWITCH BOOLEAN := FALSE;
94 ANOMALY exception;
95 --
96 CURSOR SENTRY_DATA
97 IS
98 SELECT DISTINCT
99     SUNL_CL_NO AS SUNLID,
100     SUBSTR(SUNL_WC_POLNUMBER,4,5) AS POLNUMBER,
101     SUNL_WC_EFFDATE AS POLEFF,
102     SUNL_CL_REPORT_DATE AS REPRTDTE,
103     SUNL_CL_RVHS_CLASS_CODE AS CLASSCODE,
104     SUNL_CL_PREV_REP_IND AS PREVREP,
105     SUNL_CL_LOSS_COVG_CODE AS LOSSCOVGCODE,
106     SUNL_CL_NUMBER_OF_CLAIMS AS MANYCLAIMS,
107     SUNL_CL_ACCIDENT_DATE AS ACCDATE,
108     NVL(SUNL_CL_CLAIM_NUMBER,SUNL_SENTRY_CLAIM_NUMBER) AS CLAIMNO,
109     SUNL_CL_CLAIM_STATUS AS STATUS,
110     SUNL_CL_INJURY_CODE AS INJCODE,
111     SUNL_CL_CAT_NUMBER AS CATNUM,
112     SUNL_CL_INDEMNITY_AMOUNT AS INDAMT,

```

```

113     SUNL_CL_MEDICAL_AMOUNT                AS MEDAMT,
114     SUNL_CL_SSN                            AS FEINNO,
115     SUNL_CL_UPDATE_TYPE                    AS UPDTYPE,
116     SUNL_CL_LOSS_ACT                       AS LOSSACT,
117     SUNL_CL_LOSS_TYPE                      AS LOSSTYPE,
118     SUNL_CL_LOSS_RECOVERY                  AS LOSSRECOV,
119     SUNL_CL_LOSS_COVERAGE                  AS LOSSCOVG,
120     SUNL_CL_LOSS_SETTLEMENT                AS LOSSSETTLE,
121     SUNL_STAT_GB_STATE_NO                  AS STATENO,
122     SUNL_CL_MANG_CARE_ORG_TYPE              AS MANGCARE,
123     SUNL_CL_INJ_DESC_CODE_PART              AS POBICODE,
124     SUNL_CL_INJ_DESC_CODE_NAT              AS NOSICODE,
125     SUNL_CL_INJ_DESC_CODE_CLAUSE           AS COLTCODE,
126     SUNL_CL_OCCUPATION_DESC                AS OCCDESC,
127     SUNL_CL_VOCATION_REHAB_IND              AS VOCREHAB,
128     SUNL_CL_LUMP_SUM_IND                   AS LUMPSUM,
129     SUNL_CL_FRAUD_CLAIM_IND                AS FRAUDIND,
130     SUNL_CL_DEDUCT_CODE                    AS DEDUCTCODE,
131     SUNL_CL_PAID_INDEMNITY_AMT              AS PAIDIND,
132     SUNL_CL_PAID_MEDICAL_AMT               AS PAIDMED,
133     SUNL_CL_ALAE_PAID_AMT                  AS PAIDALAE,
134     SUNL_CL_ALAE_INCURRED_AMT              AS INCURALAE,
135     SUNL_CL_AVG_WKLY_WAGE                  AS WEEKLYWAGE,
136     SUNL_CL_INCUR_VOC_REHAB               AS INCVOCREHAB,
137     SUNL_CL_CA_GROSS_INCUR                 AS GROSSINCUR,
138     SUNL_CL_PERM_DISABL_RATING              AS DISABLERATE,
139     SUNL_CL_DEDUCT_AMT                     AS DEDUCTAMT,
140     SUNL_US_HIREDATE                       AS HIREDATE,
141     --
142     SUNL_WC_POLNUMBER AS FULLPOLY,
143     ALREADY_REPORTED AS SEND_IT
144 FROM
145     SUNL
146 WHERE
147     SUBSTR(SUNL_CL_REPORT_DATE,4,3)        = V_ENDDATE
148 AND TO_CHAR(SUNL_CL_REPORT_DATE,'YYYYMMDD') <=
149     TO_CHAR(WS_ENDDTE,'YYYYMMDD')
150 AND NVL(ALREADY_REPORTED,'N') = 'N'
151 ORDER BY
152     SUNL_WC_EFFDATE,
153     SUBSTR(SUNL_WC_POLNUMBER,4,6);
154 -----
155 BEGIN
156 -----
157 SELECT
158     -- PPDT_GB_CUR_DATE
159     '30-SEP-2017'
160 INTO
161     WS_ENDDTE
162 FROM
163     PPDT
164 WHERE
165     PPDT_GB_KEY_VAL = 'DATEDISP2';
166 --
167 V_ENDDATE := SUBSTR(WS_ENDDTE,4,3);

```

```

168  --
169  FOR S1REC IN SENTRY_DATA
170  LOOP
171      WS_SUNLID := S1REC.SUNLID;
172      WS_POLNUM := S1REC.POLNUMBER;
173      WS_POLEFF := S1REC.POLEFF;
174      WS_RPTDTE := S1REC.REPRTDTE;
175      WS_CLSCDE := S1REC.CLASSCDE;
176      WS_REPIND := S1REC.PREVREP;
177      WS_CVGCDE := S1REC.LOSSCOVGCODE;
178      WS_NOCLMS := S1REC.MANYCLAIMS;
179      WS_ACCDTE := S1REC.ACCDATE;
180      WS_CLMNUM := S1REC.CLAIMNO;
181      WS_CLSTTS := S1REC.STATUS;
182      WS_INJCDE := S1REC.INJCODE;
183      WS_CATNUM := S1REC.CATNUM;
184      WS_INDAMT := S1REC.INDAMT;
185      WS_MEDAMT := S1REC.MEDAMT;
186      WS_FEINNO := S1REC.FEINNO;
187      WS_UPDTYP := S1REC.UPDTYPE;
188      WS_LSSACT := S1REC.LOSSACT;
189      WS_LSSTYP := S1REC.LOSSTYPE;
190      WS_LSSREC := S1REC.LOSSRECOV;
191      WS_LSSCOV := S1REC.LOSSCOVG;
192      WS_LSSSTL := S1REC.LOSSSETTLE;
193      WS_STATNO := S1REC.STATENO;
194      WS_HMOTYP := S1REC.MANGCARE;
195      WS_POBINO := S1REC.POBICODE;
196      WS_NOSINO := S1REC.NOSICODE;
197      WS_COLTNO := S1REC.COLTCODE;
198      WS_OCCDSC := S1REC.OCCDESC;
199      WS_REHIND := S1REC.VOCREHAB;
200      WS_LUMPSM := S1REC.LUMPSUM;
201      WS_FRDIND := S1REC.FRAUDIND;
202      WS_DEDCDE := S1REC.DEDUCTCODE;
203      WS_PAIDWI := S1REC.PAIDIND;
204      WS_PAIDWM := S1REC.PAIDMED;
205      WS_PDALAE := S1REC.PAIDALAE;
206      WS_INALAE := S1REC.INCURALAE;
207      WS_WKLYWG := S1REC.WEEKLYWAGE;
208      WS_INREHB := S1REC.INCVOCREHAB;
209      WS_GRSINC := S1REC.GROSSINCUR;
210      WS_DISABL := S1REC.DISABLERATE;
211      WS_DEDAMT := S1REC.DEDUCTAMT;
212      WS_HIRDTE := S1REC.HIREDATE;
213      ws_CONTROL := S1REC.SEND_IT;
214      --
215      WS_MONTHLY := WS_MONTHLY + 1;
216      V_SECTN := 'ALPHA';
217
218
219
220  --
221  IF WS_CONTROL = 'F' THEN
222      WS_BAD := WS_BAD + 1;
223  ELSIF WS_CONTROL = 'Y' THEN

```

```

224     WS_SKIPPED := WS_SKIPPED + 1;
225 END IF;
226 --
227 V_SECTN      := 'BRAVO';
228
229 IF SUBSTR(WS_CLMNUM,1,3) <> '55C' THEN
230     IF ws_length > 9 THEN
231         DBMS_OUTPUT.PUT_LINE('FIXING CLAIM NUMBER1 '||WS_CLMNUM);
232         UPDATE
233             UNIT_STATS_LOSS
234         SET
235             UNLS_CL_CLAIM_NUMBER = SUBSTR(WS_CLMNUM,1,7)
236         WHERE
237             UNLS_CL_NO = WS_SUNLID;
238         COMMIT;
239         ws_good := ws_good+1;
240     END IF;
241 END IF;
242
243 --
244 V_SECTN      := 'CHARLIE';
245
246 IF SUBSTR(WS_CLMNUM,1,1) = '0' THEN
247 --     DBMS_OUTPUT.PUT_LINE('FIXING CLAIM NUMBER2 '||WS_CLMNUM);
248     UPDATE
249         SENTRY_UNITSTAT_LOSS
250     SET
251         SUNL_CL_CLAIM_NUMBER = SUBSTR(WS_CLMNUM,2,6)
252     where SUNL_CL_NO = WS_SUNLID;
253     COMMIT;
254 END IF;
255
256 --
257 V_SECTN      := 'DELTA';
258 IF WS_CONTROL = 'N' THEN
259     WS_FULLPOLICY := S1REC.FULLPOLY;
260
261 --
262
263 IF SUBSTR(WS_POLNUM,1,1) NOT IN ('0','1','2','3','4','5','6','7','8','9')
264 THEN
265     DBMS_OUTPUT.PUT_LINE('-----');
266     DBMS_OUTPUT.PUT_LINE('BAD POLICY NUMBER A '||WS_FULLPOLICY);
267     UPDATE_SWITCH := TRUE;
268 END IF;
269 --
270 V_SECTN      := 'ECHO';
271 IF LENGTH(WS_FULLPOLICY) < 10 THEN
272     WS_FULLPOLICY := WS_FULLPOLICY||SUBSTR(WS_POLEFF,10,2);
273     UPDATE
274         SENTRY_UNITSTAT_LOSS
275     SET
276         SUNL_WC_POLNUMBER = WS_FULLPOLICY
277     WHERE
278         SUNL_CL_NO = WS_SUNLID;
279     COMMIT;

```

```

280     ELSE
281         IF LENGTH(WS_FULLPOLICY) > 10 THEN
282             DBMS_OUTPUT.PUT_LINE('-----');
283             DBMS_OUTPUT.PUT_LINE('BAD POLICY NUMBER B '||WS_FULLPOLICY);
284             UPDATE_SWITCH := TRUE;
285         END IF;
286     END IF;
287     --
288 --
289
290
291 --
292
293     V_SECTN := 'FOXTROT';
294
295
296 V_POLDAT := F_VERIFY_RECORD_DATA(WS_FULLPOLICY,WS_POLEFF);
297
298 if substr(V_POLDAT,1,1) = 'N' then
299     DBMS_OUTPUT.PUT_LINE('-----');
300     DBMS_OUTPUT.PUT_LINE('BAD POLICY NUMBER C '||WS_FULLPOLICY||' '||
301     WS_POLEFF||' '||WS_CLMNUM);
302     DBMS_OUTPUT.PUT_LINE(V_POLDAT);
303     UPDATE_SWITCH := TRUE;
304 end if;
305
306
307 if substr(V_POLDAT,1,1) = 'M' then
308     DBMS_OUTPUT.PUT_LINE('-----');
309     DBMS_OUTPUT.PUT_LINE('MISSING POLICY SUFFIX F '||WS_FULLPOLICY||' '||
310     WS_POLEFF||' '||WS_CLMNUM);
311     UPDATE_SWITCH := TRUE;
312 end if;
313
314
315
316 if substr(V_POLDAT,1,1) = 'Y' and substr(V_POLDAT,2,1) = 'N' then
317     DBMS_OUTPUT.PUT_LINE('-----');
318     DBMS_OUTPUT.PUT_LINE('BAD EFFECTIVE DATE '||WS_FULLPOLICY||' '||
319     WS_POLEFF||' '||WS_CLMNUM);
320     UPDATE_SWITCH := TRUE;
321 end if;
322
323
324 -- SELECT
325 --   COUNT(UNPL_WC_NO)
326 -- INTO
327 --   V_POL_COUNT
328 -- FROM
329 --   UNPL
330 -- WHERE
331 --   SUBSTR(UNPL.UNPL_WC_NCCI_POL_NUMBER,1,8)||SUBSTR(
332 --   UNPL.UNPL_WC_POLY_EFF_DATE,10,2) = WS_FULLPOLICY
333 --   AND UNPL.UNPL_WC_POLY_EFF_DATE = WS_POLEFF;
334 --
335 -- IF V_POL_COUNT < 1 THEN

```

```

335 --      DBMS_OUTPUT.PUT_LINE('-----');
336 --      DBMS_OUTPUT.PUT_LINE('BAD POLICY NUMBER C '||WS_FULLPOLICY||' '||
337 --      WS_POLEFF||' '||WS_CLMNUM);
338 --      UPDATE_SWITCH := TRUE;
339 --  END IF;
340 --
341 V_SECTN := 'GOLF';
342 BEGIN
343     SELECT
344         'OK'
345     INTO
346         V_STATUS
347     FROM
348         SUNL
349     WHERE
350         WS_CLSTTS IN ('O','C','0','1','2')
351         AND SUNL_CL_NO = WS_SUNLID;
352 EXCEPTION
353 WHEN NO_DATA_FOUND THEN
354     V_STATUS := 'ZZ';
355 END;
356 --
357 V_SECTN := 'HOTEL';
358 IF V_STATUS != 'OK' THEN
359     DBMS_OUTPUT.PUT_LINE('-----');
360     DBMS_OUTPUT.PUT_LINE('BAD CLAIM STATUS D '||WS_FULLPOLICY||' '||
361     WS_CLMNUM||' status: '||WS_CLSTTS);
362     UPDATE_SWITCH := TRUE;
363 END IF;
364 V_STATUS := NULL;
365 --
366 SELECT
367     LENGTH(WS_INJCDE)
368 INTO
369     ws_length
370 FROM
371     dual;
372 IF ws_length > 1 THEN
373     DBMS_OUTPUT.PUT_LINE('-----');
374     DBMS_OUTPUT.PUT_LINE('BAD INJURY CODE E '||WS_FULLPOLICY);
375     UPDATE_SWITCH := TRUE;
376 END IF;
377 ---
378 BEGIN
379     V_SECTN := 'INDIAN';
380     SELECT
381         CLAIM_NUMBER,
382         DEDUCTIBLE_AMT,
383         SUNL_CL_DEDUCT_AMT
384     INTO
385         RD_CLAIM,
386         RD_AMT,
387         RD_RPTAMT
388     FROM
389         BO_RECOVERABLE_DEDUCTIBLE BORD,
390         SUNL

```



```

391 WHERE
392     SUBSTR(CLAIM_NUMBER,2,7) = RTRIM(SUNL_CL_CLAIM_NUMBER)
393 AND SUBSTR(CLAIM_NUMBER,1,1) = 'W'
394 AND SUNL_CL_NO = WS_SUNLID;
395 --
396 IF RD_AMT <> RD_RPTAMT THEN
397
398     DBMS_OUTPUT.PUT_LINE('-----');
399     DBMS_OUTPUT.PUT_LINE('ID# CLAIM# DEDUCT REPORTED');
400     DBMS_OUTPUT.PUT_LINE(WS_SUNLID||' '||RD_CLAIM||' '||RD_AMT||' '||RD_RPTAMT);
401
402     update SUNL
403     set SUNL_CL_DEDUCT_AMT = RD_AMT
404     where SUNL_CL_NO = WS_SUNLID;
405     commit;
406
407
408     END IF;
409     --
410 EXCEPTION
411 WHEN NO_DATA_FOUND THEN
412     NULL;
413 END;
414 ---
415 V_SECTN := 'JESTER';
416 -- *****
417 IF UPDATE_SWITCH THEN
418 -- DBMS_OUTPUT.PUT_LINE('DO NOT REPORT POLICY '||WS_FULLPOLICY);
419 -- DBMS_OUTPUT.PUT_LINE('-----');
420 --
421 UPDATE
422     SUNL
423 SET
424     ALREADY_REPORTED = 'F'
425 WHERE
426     SUNL_WC_POLNUMBER = WS_FULLPOLICY
427 AND SUNL_CL_NO = WS_SUNLID;
428 COMMIT;
429 --
430 UPDATE_SWITCH := FALSE;
431 WS_BAD := WS_BAD + 1;
432 ELSE
433
434     WS_GOOD := WS_GOOD + 1;
435 END IF;
436 END IF;
437 --
438
439
440
441
442 END LOOP;
443 DBMS_OUTPUT.PUT_LINE('_____');
444 DBMS_OUTPUT.PUT_LINE('');
445 DBMS_OUTPUT.PUT_LINE('');
446 DBMS_OUTPUT.PUT_LINE(' '||WS_MONTHLY||' TOTAL RECORDS for '||WS_ENDDTE);

```



```
447 DBMS_OUTPUT.PUT_LINE(' Passed '||WS_GOOD||' Records');
448 DBMS_OUTPUT.PUT_LINE(' Failed '||WS_BAD||' Records');
449 DBMS_OUTPUT.PUT_LINE(' Skipped '||WS_SKIPPED||' Records');
450 -----
451 EXCEPTION
452 WHEN VALUE_ERROR THEN
453   DBMS_OUTPUT.PUT_LINE('VALUE_ERROR POL#'||WS_POLNUM||' SUNL ID: '||
454   '||SQLCODE);
455 WHEN INVALID_NUMBER THEN
456   DBMS_OUTPUT.PUT_LINE('INVALID_NUMBER POL#'||WS_POLNUM||' SUNL ID: '||
457   WS_SUNLID||' '||SQLCODE);
458 WHEN ROWTYPE_MISMATCH THEN
459   DBMS_OUTPUT.PUT_LINE('ROWTYPE_MISMATCH POL#'||WS_POLNUM||' SUNL ID: '||
460   WS_SUNLID||' '||SQLCODE);
461 WHEN TOO_MANY_ROWS THEN
462   DBMS_OUTPUT.PUT_LINE('TOO_MANY_ROWS POL#'||WS_POLNUM||' SUNL ID:
463   '||WS_SUNLID
464   ||' '||SQLCODE);
465 WHEN NO_DATA_FOUND THEN
466   DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND POL#'||WS_POLNUM||' SUNL ID: '||WS_SUNLID
467   ||' '||V_SECTN);
468 WHEN OTHERS THEN
469   DBMS_OUTPUT.PUT_LINE('FAILURE ON POL#'||WS_POLNUM||' SUNL ID: '||WS_SUNLID||
470   ' '||SQLCODE);
471 END;
472 /
```