

```

1
2  -----
3
4  SENTRY_STATS_XFER.SQL
5
6  FUNCTION: USE TABLE SENTRY_UNITSTATS_LOSS (SUNL) TO
7  LOAD TABLE UNIT_STATS_LOSS (UNLS)
8
9
10 MODIFICATION HISTORY:
11
12 PROJECT# TASK DATE TECH ACTION
13 -----
14 PLI32502.31220 02-FEB-2016 JPC INITIAL CREATION
15
16 -----
17 DECLARE
18 WS_POLNUM SUNL.SUNL_WC_POLNUMBER%TYPE;
19 WS_POLEFF SUNL.SUNL_WC_EFFDATE%TYPE;
20 WS_RPTDTE SUNL.SUNL_CL_REPORT_DATE%TYPE;
21 WS_CLSCDE SUNL.SUNL_CL_RVHS_CLASS_CODE%TYPE;
22 WS_REPIND SUNL.SUNL_CL_PREV_REP_IND%TYPE;
23 WS_CVGCDE SUNL.SUNL_CL_LOSS_COVG_CODE%TYPE;
24 WS_NOCLMS SUNL.SUNL_CL_NUMBER_OF_CLAIMS%TYPE;
25 WS_ACCDTE SUNL.SUNL_CL_ACCIDENT_DATE%TYPE;
26 WS_CLMNUM SUNL.SUNL_CL_CLAIM_NUMBER%TYPE;
27 WS_CLSTTS SUNL.SUNL_CL_CLAIM_STATUS%TYPE;
28 WS_INJCDE SUNL.SUNL_CL_INJURY_CODE%TYPE;
29 WS_CATNUM SUNL.SUNL_CL_CAT_NUMBER%TYPE;
30 WS_INDAMT SUNL.SUNL_CL_INDEMNITY_AMOUNT%TYPE;
31 WS_MEDAMT SUNL.SUNL_CL_MEDICAL_AMOUNT%TYPE;
32 WS_FEINNO SUNL.SUNL_CL_SSN%TYPE;
33 WS_UPDTYP SUNL.SUNL_CL_UPDATE_TYPE%TYPE;
34 WS_LSSACT SUNL.SUNL_CL_LOSS_ACT%TYPE;
35 WS_LSSTYP SUNL.SUNL_CL_LOSS_TYPE%TYPE;
36 WS_LSSREC SUNL.SUNL_CL_LOSS_RECOVERY%TYPE;
37 WS_LSSCOV SUNL.SUNL_CL_LOSS_COVERAGE%TYPE;
38 WS_LSSSTL SUNL.SUNL_CL_LOSS_SETTLEMENT%TYPE;
39 WS_STATNO SUNL.SUNL_STAT_GB_STATE_NO%TYPE;
40 WS_HMOTYP SUNL.SUNL_CL_MANG_CARE_ORG_TYPE%TYPE;
41 WS_POBINO SUNL.SUNL_CL_INJ_DESC_CODE_PART%TYPE;
42 WS_NOSINO SUNL.SUNL_CL_INJ_DESC_CODE_NAT%TYPE;
43 WS_COLTNO SUNL.SUNL_CL_INJ_DESC_CODE_CLAUSE%TYPE;
44 WS_OCCDSC SUNL.SUNL_CL_OCCUPATION_DESC%TYPE;
45 WS_REHIND SUNL.SUNL_CL_VOCATION_REHAB_IND%TYPE;
46 WS_LUMPSM SUNL.SUNL_CL_LUMP_SUM_IND%TYPE;
47 WS_FRDIND SUNL.SUNL_CL_FRAUD_CLAIM_IND%TYPE;
48 WS_DEDCDE SUNL.SUNL_CL_DEDUCT_CODE%TYPE;
49 WS_PAIDWI SUNL.SUNL_CL_PAID_INDEMNITY_AMT%TYPE;
50 WS_PAIDWM SUNL.SUNL_CL_PAID_MEDICAL_AMT%TYPE;
51 WS_PDALAE SUNL.SUNL_CL_ALAE_PAID_AMT%TYPE;
52 WS_INALAE SUNL.SUNL_CL_ALAE_INCURRED_AMT%TYPE;
53 WS_WKLYWG SUNL.SUNL_CL_AVG_WKLY_WAGE%TYPE;
54 WS_INREHB SUNL.SUNL_CL_INCUR_VOC_REHAB%TYPE;
55 WS_GRSINC SUNL.SUNL_CL_CA_GROSS_INCUR%TYPE;
56 WS_DISABL SUNL.SUNL_CL_PERM_DISABL_RATING%TYPE;

```

```

57 WS_DEDAMT SUNL.SUNL_CL_DEDUCT_AMT%TYPE;
58 WS_DESDED SUNL.SUNL_CL_DEDUCT_AMT%TYPE;
59 WS_HIRDTE SUNL.SUNL_US_HIREDATE%TYPE;
60 --
61 WS_ATTCLM UNLS.UNLS_CL_CLAIMANT_ATT_FEE%TYPE;
62 WS_ATTEMP UNLS.UNLS_CL_EMPLOYER_ATT_FEE%TYPE;
63 WS_UNLSNO UNLS.UNLS_CL_NO%TYPE;
64 WS_UNPLNO UNPL.UNPL_WC_NO%TYPE;
65 WS_LAWIND VARCHAR2(1);
66 WS_RPTNUM VARCHAR2(1);
67 WS_ENDDTE DATE := NULL;
68 WS_RECUPD DATE;
69 WS_OCCDTE DATE;
70 WS_SUNLID NUMBER;
71 V_ENDDATE VARCHAR2(3);
72 V_RPM NUMBER := 0;
73 --
74 WS_SECTION NUMBER;
75 BADFUNCTION EXCEPTION;
76 --
77 CURSOR SENTRY_DATA
78 IS
79 SELECT DISTINCT
80     SUNL_CL_NO                                AS SUNLID,
81     SUBSTR(SUNL_WC_POLNUMBER,4,5)             AS POLNUMBER,
82     SUNL_WC_EFFDATE                           AS POLEFF,
83     SUNL_CL_REPORT_DATE                       AS REPRTDTE,
84     SUNL_CL_RVHS_CLASS_CODE                   AS CLASSCODE,
85     NVL(SUNL_CL_PREV_REP_IND,0)               AS PREVREP,
86     SUNL_CL_LOSS_COVG_CODE                   AS LOSSCOVGCODE,
87     SUNL_CL_NUMBER_OF_CLAIMS                 AS MANYCLAIMS,
88     SUNL_CL_ACCIDENT_DATE                    AS ACCDATE,
89     rtrim(NVL(SUNL_CL_CLAIM_NUMBER,SUNL_SENTRY_CLAIM_NUMBER)) AS
        CLAIMNO,
90     decode(SUNL_CL_CLAIM_STATUS,'0','O','1','C') AS STATUS,
91     SUNL_CL_INJURY_CODE                      AS INJCODE,
92     SUNL_CL_CAT_NUMBER                      AS CATNUM,
93     SUNL_CL_INDEMNITY_AMOUNT                AS INDAMT,
94     SUNL_CL_MEDICAL_AMOUNT                  AS MEDAMT,
95     SUNL_CL_SSN                             AS FEINNO,
96     SUNL_CL_UPDATE_TYPE                     AS UPDTYPE,
97     SUNL_CL_LOSS_ACT                        AS LOSSACT,
98     SUNL_CL_LOSS_TYPE                       AS LOSSTYPE,
99     SUNL_CL_LOSS_RECOVERY                   AS LOSSRECOV,
100    SUNL_CL_LOSS_COVERAGE                   AS LOSSCOVG,
101    SUNL_CL_LOSS_SETTLEMENT                 AS LOSSSETTLE,
102    SUNL_STAT_GB_STATE_NO                   AS STATENO,
103    SUNL_CL_MANG_CARE_ORG_TYPE               AS MANGCARE,
104    SUNL_CL_INJ_DESC_CODE_PART              AS POBICODE,
105    SUNL_CL_INJ_DESC_CODE_NAT              AS NOSICODE,
106    SUNL_CL_INJ_DESC_CODE_CLAUSE            AS COLTCODE,
107    replace(SUNL_CL_OCCUPATION_DESC,'&','/') AS OCCDESC,
108    SUNL_CL_VOCATION_REHAB_IND              AS VOCREHAB,
109    SUNL_CL_LUMP_SUM_IND                    AS LUMPSUM,
110    SUNL_CL_FRAUD_CLAIM_IND                AS FRAUDIND,
111    SUNL_CL_DEDUCT_CODE                     AS DEDUCTCODE,

```

```

112     SUNL_CL_PAID_INDEMNITY_AMT                AS PAIDIND,
113     SUNL_CL_PAID_MEDICAL_AMT                  AS PAIDMED,
114     SUNL_CL_ALAE_PAID_AMT                      AS PAIDALAE,
115     SUNL_CL_ALAE_INCURRED_AMT                  AS INCURALAE,
116     SUNL_CL_AVG_WKLY_WAGE                      AS WEEKLYWAGE,
117     SUNL_CL_INCUR_VOC_REHAB                   AS INCVOCREHAB,
118     SUNL_CL_CA_GROSS_INCUR                     AS GROSSINCUR,
119     SUNL_CL_PERM_DISABL_RATING                  AS DISABLERATE,
120     SUNL_CL_DEDUCT_AMT                        AS DEDUCTAMT,
121     SUNL_US_HIREDATE                          AS HIREDATE
122 FROM
123     SUNL
124 WHERE
125     SUBSTR(SUNL_CL_REPORT_DATE,4,3)            = V_ENDDATE
126 AND TO_CHAR(SUNL_CL_REPORT_DATE,'YYYYMMDD') <=
    TO_CHAR(to_date(WS_ENDDTE),'YYYYMMDD')
127 AND NVL(ALREADY_REPORTED,'N')                = 'N'
128 ORDER BY
129     SUNL_WC_EFFDATE,
130     SUBSTR(SUNL_WC_POLNUMBER,4,5);
131 -- *****
132 -- GET_UNLS_RPT_SEQ
133 -- *****
134 FUNCTION GET_UNLS_RPT_SEQ(
135     WS_END_DATE          DATE,
136     WS_RVHS_GB_POLY_EFF_DATE DATE)
137 RETURN VARCHAR2
138 IS
139     WS_RPT_SEQ VARCHAR2(1);
140 BEGIN
141     -- ASSIGNS REPORT SEQUENCE NUMBER
142 SELECT
143     DECODE(TO_CHAR(MONTHS_BETWEEN (WS_END_DATE, UPPER(LAST_DAY(
144     WS_RVHS_GB_POLY_EFF_DATE))))), '18','1', '30','2', '42','3', '54','4',
145     '66','5', '78','6', '90','7', '102','8', '114','9', '126','A', '139','B'
146     , 'X') AS RPT_SEQ
147 INTO
148     WS_RPT_SEQ
149 FROM
150     DUAL;
151 --
152 RETURN WS_RPT_SEQ;
153 EXCEPTION
154 WHEN NO_DATA_FOUND THEN
155     DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND IN GET_UNLS_RPT_SEQ USING
    '||WS_END_DATE||' AND '||WS_RVHS_GB_POLY_EFF_DATE);
156 WHEN OTHERS THEN
157     DBMS_OUTPUT.PUT_LINE('FAILURE IN GET_UNLS_RPT_SEQ '||SQLCODE);
158 END GET_UNLS_RPT_SEQ;
159 -- *****\--
160 -- GET_UNLS_ID
161 -- *****
162 FUNCTION GET_UNLS_ID
163     --(WS_UNLS_CL_NO    NUMBER)
164 RETURN NUMBER
165 IS

```

```

166  WS_UNLS_CL_NO NUMBER;
167  BEGIN
168  -- ASSIGNS UNLS ID NUMBER
169  SELECT
170  UNIT_STATS_LOSS_SEQUENCE.NEXTVAL
171  INTO
172  WS_UNLS_CL_NO
173  FROM
174  DUAL;
175  RETURN WS_UNLS_CL_NO;
176  EXCEPTION
177  WHEN OTHERS THEN
178  DBMS_OUTPUT.PUT_LINE('FAILURE IN GET_UNLS_ID '||SQLCODE);
179  END GET_UNLS_ID;
180  -- *****
181  -- GET_UNPL_ID
182  -- *****
183  FUNCTION GET_UNPL_ID(
184  WS_RVHS_POL_NO          VARCHAR2,
185  WS_RVHS_GB_POLY_EFF_MMDDYYYY DATE)
186  RETURN NUMBER
187  IS
188  WS_UNLS_UNPL_WC_NO NUMBER;
189  BEGIN
190  -- FINDS CORRECT UNPL ID NUMBER
191  SELECT
192  UNPL_WC_NO
193  INTO
194  WS_UNLS_UNPL_WC_NO
195  FROM
196  UNPL
197  WHERE
198  UNPL_WC_POLY_NUMBER = RTRIM(WS_RVHS_POL_NO)
199  AND UNPL_WC_POLY_EFF_DATE =
200  TO_DATE(WS_RVHS_GB_POLY_EFF_MMDDYYYY,'DD-MON-YYYY')
201  AND UNPL_WC_CORRECTION_NO = 0 ;
202  RETURN WS_UNLS_UNPL_WC_NO;
203  EXCEPTION
204  WHEN NO_DATA_FOUND THEN
205  DBMS_OUTPUT.PUT_LINE('NO DATA FOUND IN GET_UNPL_ID POL
206  #'||WS_RVHS_POL_NO||' EFF: '||WS_RVHS_GB_POLY_EFF_MMDDYYYY);
207  RAISE BADFUNCTION;
208
209  -- USE THIS FOR TESTING
210  --RETURN 0;
211
212  WHEN OTHERS THEN
213  DBMS_OUTPUT.PUT_LINE('FAILURE IN GET_UNPL_ID '||WS_RVHS_POL_NO||' '||
214  WS_RVHS_GB_POLY_EFF_MMDDYYYY||' '||SQLCODE);
215  END GET_UNPL_ID;
216  -- *****
217  -- GET_ALPHA_STATE
218  -- *****
219  FUNCTION GET_ALPHA_STATE(
220  WS_STATE_NO VARCHAR2)

```

```

220 RETURN VARCHAR2
221 IS
222 WS_ALPHA VARCHAR2(2);
223 BEGIN
224 SELECT
225 STAT_GB_ALPHA_STATE
226 INTO
227 WS_ALPHA
228 FROM
229 STAT
230 WHERE
231 STAT_GB_STATE_NO = WS_STATE_NO;
232 RETURN WS_ALPHA;
233 EXCEPTION
234 WHEN OTHERS THEN
235 DBMS_OUTPUT.PUT_LINE('FAILURE IN GET_ALPHA_STATE '||SQLCODE);
236 END GET_ALPHA_STATE;
237
238 -- *****
239 -- MAIN PROCESS
240 -- *****
241 BEGIN
242
243 SELECT
244 PPDT_GB_CUR_DATE
245 INTO
246 WS_ENDDTE
247 FROM
248 PPDT
249 WHERE
250 PPDT_GB_KEY_VAL = 'DATEDISP2';
251
252
253
254 WS_SECTION := 1;
255 -- V_ENDDATE GETS THE REPORTING MONTH
256 V_ENDDATE := SUBSTR(WS_ENDDTE,4,3);
257 --
258 FOR S1REC IN SENTRY_DATA
259 LOOP
260 WS_SUNLID := S1REC.SUNLID;
261 WS_POLNUM := S1REC.POLNUMBER;
262 WS_POLEFF := S1REC.POLEFF;
263 WS_RPTDTE := S1REC.REPRTDTE;
264 WS_CLSCDE := S1REC.CLASSCDE;
265 WS_REPIND := S1REC.PREVREP;
266 WS_CVGCDE := S1REC.LOSSCOVGCDE;
267 WS_NOCLMS := S1REC.MANYCLAIMS;
268 WS_ACCDTE := S1REC.ACCDATE;
269 WS_CLMNUM := S1REC.CLAIMNO;
270 WS_CLSTTS := S1REC.STATUS;
271 WS_INJCDE := S1REC.INJCODE;
272 WS_CATNUM := S1REC.CATNUM;
273 WS_INDAMT := S1REC.INDAMT;
274 WS_MEDAMT := S1REC.MEDAMT;
275 WS_FEINNO := S1REC.FEINNO;

```

```

276 WS_UPDTYP := S1REC.UPDTYPE;
277 WS_LSSACT := S1REC.LOSSACT;
278 WS_LSSTYP := S1REC.LOSSTYPE;
279 WS_LSSREC := S1REC.LOSSRECOV;
280 WS_LSSCOV := S1REC.LOSSCOVG;
281 WS_LSSSTL := S1REC.LOSSSETTLE;
282 WS_STATNO := S1REC.STATENO;
283 WS_HMOTYP := S1REC.MANGCARE;
284 WS_POBINO := S1REC.POBICODE;
285 WS_NOSINO := S1REC.NOSICODE;
286 WS_COLTNO := S1REC.COLTCODE;
287 WS_OCCDSC := S1REC.OCCDESC;
288 WS_REHIND := S1REC.VOCREHAB;
289 WS_LUMPSM := S1REC.LUMPSUM;
290 WS_FRDIND := S1REC.FRAUDIND;
291 WS_DEDCDE := S1REC.DEDUCTCODE;
292 WS_PAIDWI := S1REC.PAIDIND;
293 WS_PAIDWM := S1REC.PAIDMED;
294 WS_PDALAE := S1REC.PAIDALAE;
295 WS_INALAE := S1REC.INCURALAE;
296 WS_WKLYWG := S1REC.WEEKLYWAGE;
297 WS_INREHB := S1REC.INCVOCREHAB;
298 WS_GRSINC := S1REC.GROSSINCUR;
299 WS_DISABL := S1REC.DISABLERATE;
300 WS_DEDAMT := S1REC.DEDUCTAMT;
301 WS_HIRDTE := S1REC.HIREDATE;
302 --
303 WS_SECTION := 2;
304 WS_RECUPD := SYSDATE;
305 --
306 WS_SECTION := 3;
307 WS_ENDDTE := WS_RPTDTE;
308 --
309 BEGIN
310 WS_SECTION := 4;
311 WS_UNPLNO := GET_UNPL_ID(WS_POLNUM,WS_POLEFF);
312 EXCEPTION
313     WHEN NO_DATA_FOUND THEN
314         DBMS_OUTPUT.PUT_LINE('UNPL_ID FOR '||WS_POLNUM||' FAILED');
315     WHEN OTHERS THEN
316         DBMS_OUTPUT.PUT_LINE('UNPL_ID FOR '||WS_POLNUM||' WENT PEAR-SHAPED');
317 END;
318 --
319 BEGIN
320 WS_SECTION := 5;
321 WS_UNLSNO := GET_UNLS_ID;
322 EXCEPTION
323     WHEN NO_DATA_FOUND THEN
324         DBMS_OUTPUT.PUT_LINE('UNLS_ID FOR '||WS_POLNUM||' FAILED');
325     WHEN OTHERS THEN
326         DBMS_OUTPUT.PUT_LINE('UNLS_ID FOR '||WS_POLNUM||' WENT PEAR-SHAPED');
327 END;
328 --
329 BEGIN
330 WS_SECTION := 6;
331 WS_RPTNUM := GET_UNLS_RPT_SEQ(WS_ENDDTE,WS_POLEFF);

```

```

332 EXCEPTION
333     WHEN NO_DATA_FOUND THEN
334         DBMS_OUTPUT.PUT_LINE('REPORT SEQ. FOR '||WS_POLNUM||' FAILED');
335     WHEN OTHERS THEN
336         DBMS_OUTPUT.PUT_LINE('REPORT SEQ. FOR '||WS_POLNUM||' WENT
            PEAR-SHAPED');
337 END;
338 --
339 WS_SECTION := 62;
340 WS_LAWIND := NULL;
341 WS_ATTCLM := 0;
342 WS_ATTEMP := 0;
343 --
344 WS_SECTION := 10;
345
346 INSERT
347 INTO
348     OPS$JAMES.UNIT_STATS_LOSS
349     (
350         UNLS_CL_NO,
351         UNLS_UNPL_WC_NO ,
352         UNLS_CL_REPORT_DATE ,
353         UNLS_CL_REPORT_NO ,
354         UNLS_CL_RVHS_CLASS_CODE ,
355         UNLS_CL_PREV_REP_IND ,
356         UNLS_CL_LOSS_COVG_CODE ,
357         UNLS_CL_NUMBER_OF_CLAIMS ,
358         UNLS_CL_ACCIDENT_DATE ,
359         UNLS_CL_CLAIM_NUMBER ,
360         UNLS_CL_CLAIM_STATUS ,
361         UNLS_CL_INJURY_CODE ,
362         UNLS_CL_CAT_NUMBER ,
363         UNLS_CL_INDEMNITY_AMOUNT ,
364         UNLS_CL_MEDICAL_AMOUNT ,
365         UNLS_CL_SSN ,
366         UNLS_CL_UPDATE_TYPE ,
367         UNLS_CL_LOSS_ACT ,
368         UNLS_CL_LOSS_TYPE ,
369         UNLS_CL_LOSS_RECOVERY ,
370         UNLS_CL_LOSS_COVERAGE ,
371         UNLS_CL_LOSS_SETTLEMENT ,
372         UNLS_STAT_GB_STATE_NO ,
373         UNLS_CL_MANG_CARE_ORG_TYPE ,
374         UNLS_CL_INJ_DESC_CODE_PART ,
375         UNLS_CL_INJ_DESC_CODE_NAT ,
376         UNLS_CL_INJ_DESC_CODE_CLAUSE ,
377         UNLS_CL_OCCUPATION_DESC ,
378         UNLS_CL_VOCATION_REHAB_IND ,
379         UNLS_CL_LUMP_SUM_IND ,
380         UNLS_CL_FRAUD_CLAIM_IND ,
381         UNLS_CL_DEDUCT_CODE ,
382         UNLS_CL_PAID_INDEMNITY_AMT ,
383         UNLS_CL_PAID_MEDICAL_AMT ,
384         UNLS_CL_CLAIMANT_ATT_FEE ,
385         UNLS_CL_EMPLOYER_ATT_FEE ,
386         UNLS_CL_ALAE_PAID_AMT ,

```



```

387 UNLS_CL_ALAE_INCURRED_AMT ,
388 UNLS_CL_RECORD_UPDATE ,
389 UNLS_CL_AVG_WKLY_WAGE ,
390 UNLS_CL_INCUR_VOC_REHAB ,
391 UNLS_CL_CA_GROSS_INCUR ,
392 UNLS_CL_PERM_DISABL_RATING ,
393 UNLS_CL_DEDUCT_AMT ,
394 UNLS_US_HIREDATE ,
395 UNLS_UNIT_XMIT
396 )
397 VALUES
398 (
399     WS_UNLSNO,
400     WS_UNPLNO,
401     WS_RPTDTE,
402     DECODE(WS_RPTNUM,'A',10,'X',0,WS_RPTNUM),
403     lpad(WS_CLSCDE,4,'0'),
404     WS_REPIND,
405     WS_CVGCDE,
406     WS_NOCLMS,
407     WS_ACCDTE,
408     DECODE(SUBSTR(WS_CLMNUM,1,1),0,SUBSTR(WS_CLMNUM,2,6),WS_CLMNUM),
409     WS_CLSTTS,
410     WS_INJCDE,
411     WS_CATNUM,
412     WS_INDAMT,
413     WS_MEDAMT,
414     WS_FEINNO,
415     WS_UPDTYP,
416     WS_LSSACT,
417     WS_LSSTYP,
418     WS_LSSREC,
419     WS_LSSCOV,
420     WS_LSSSTL,
421     lpad(WS_STATNO,2,'0'),
422     WS_HMOTYP,
423     WS_POBINO,
424     WS_NOSINO,
425     WS_COLTNO,
426     WS_OCCDSC,
427     WS_REHIND,
428     WS_LUMPSM,
429     WS_FRDIND,
430     WS_DEDCDE,
431     WS_PAIDWI,
432     WS_PAIDWM,
433     WS_ATTCLM,
434     WS_ATTEMP,
435     WS_PDALAE,
436     WS_INALAE,
437     WS_RECUPD,
438     WS_WKLYWG,
439     WS_INREHB,
440     WS_GRSINC,
441     WS_DISABL,
442     WS_DEDAMT,

```



```

443     WS_HIRDTE,
444     'Y'
445 );
446 --
447     WS_SECTION := 11;
448     UPDATE
449     SUNL
450     SET
451     ALREADY_REPORTED = 'Y'
452     WHERE
453     SUNL_CL_NO = WS_SUNLID;
454     --
455     V_RPM := V_RPM + 1;
456     COMMIT;
457     --
458     END LOOP;
459     --
460     IF SQLCODE = 0 THEN
461         DBMS_OUTPUT.PUT_LINE('PROCESSED '||V_RPM||' RECORDS');
462     END IF;
463     --
464
465     EXCEPTION
466     WHEN BADFUNCTION THEN
467         DBMS_OUTPUT.PUT_LINE('FUNCTION RETURNED W/O VALUE '||WS_SECTION||' POL#'||
468         WS_POLNUM||' '||SQLCODE);
469         DBMS_OUTPUT.PUT_LINE('PROCESSED '||V_RPM||' RECORDS');
470     WHEN VALUE_ERROR THEN
471         DBMS_OUTPUT.PUT_LINE('VALUE TYPE MISMATCH '||WS_SECTION||'
472         POL#'||WS_POLNUM||' '
473         ||SQLCODE);
474         DBMS_OUTPUT.PUT_LINE('PROCESSED '||V_RPM||' RECORDS');
475     WHEN NO_DATA_FOUND THEN
476         DBMS_OUTPUT.PUT_LINE('NO DATA FOUND IN SENTRY_STATS_XFER SECTION
477         '||WS_SECTION||' POL#'||
478         WS_POLNUM||' ERROR '||SQLCODE);
479         DBMS_OUTPUT.PUT_LINE('PROCESSED '||V_RPM||' RECORDS');
480     WHEN TOO_MANY_ROWS THEN
481         DBMS_OUTPUT.PUT_LINE('TOO MANY ROWS IN SENTRY_STATS_XFER SECTION
482         '||WS_SECTION||' POL#'||
483         WS_POLNUM||' ERROR '||SQLCODE);
484         DBMS_OUTPUT.PUT_LINE('PROCESSED '||V_RPM||' RECORDS');
485     WHEN OTHERS THEN
486         DBMS_OUTPUT.PUT_LINE('FAILURE IN SENTRY_STATS_XFER SECTION '||WS_SECTION||'
487         POL#'||
488         WS_POLNUM||' '||SQLCODE);
489         DBMS_OUTPUT.PUT_LINE('PROCESSED '||V_RPM||' RECORDS');
490     END;
491 /

```