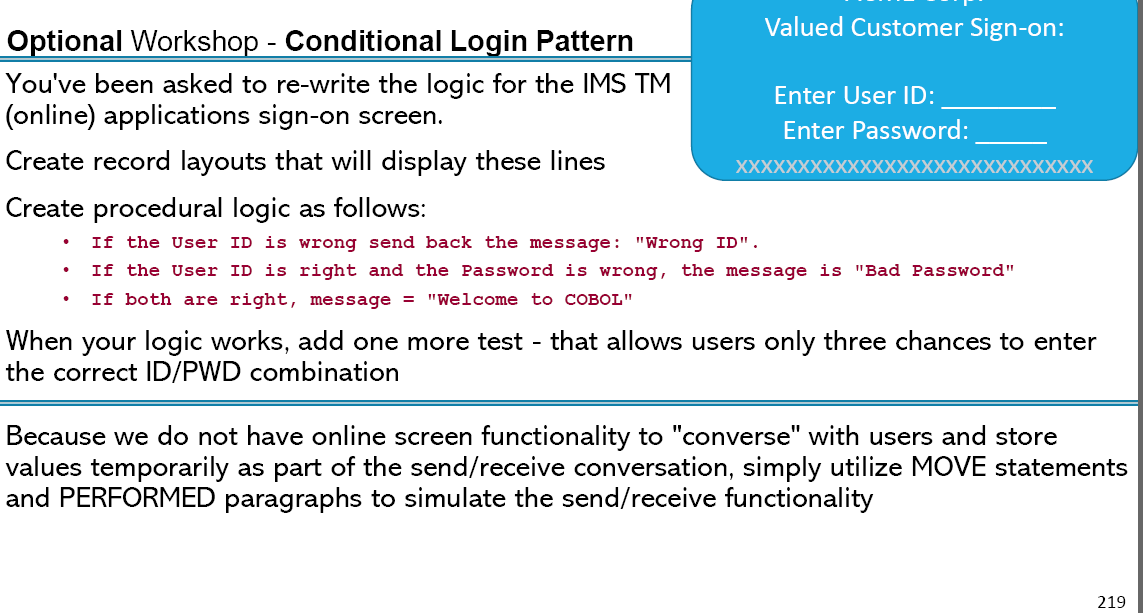
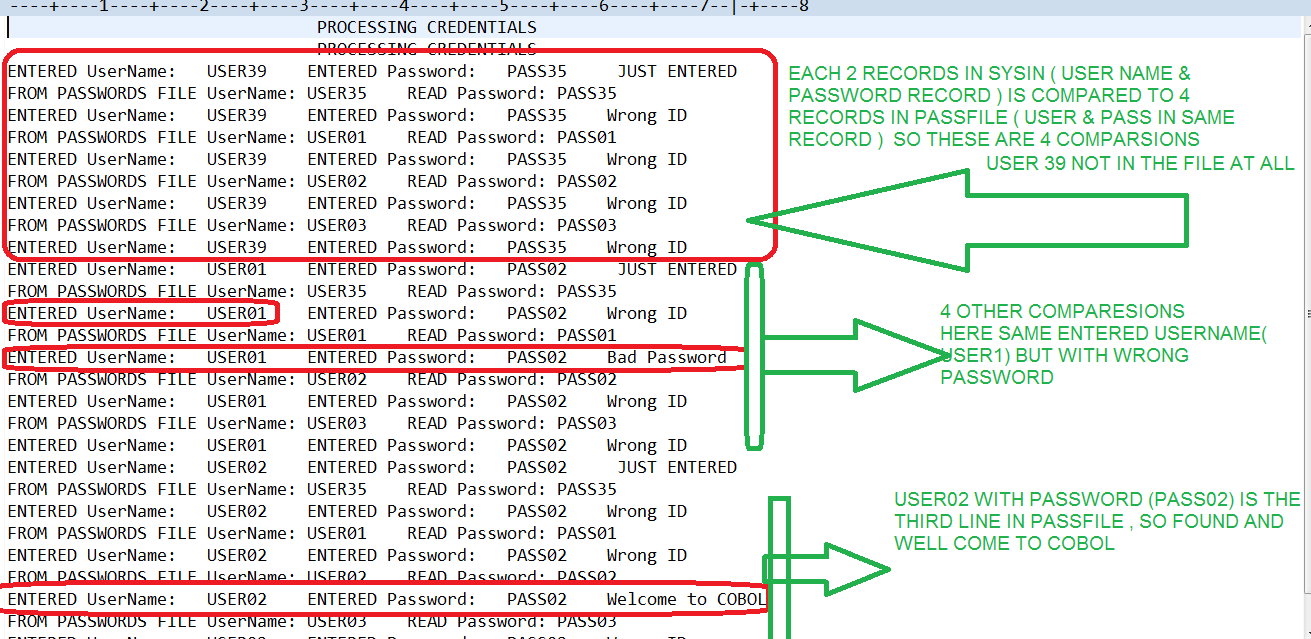
Question?



Answer :



//USER35G JOB ,

// MSGCLASS=H,MSGLEVEL=(1,1),TIME=(,4),REGION=144M,COND=(16,LT)

//\*

// SET COBPGM='USERPASS'

//\*\*\*\* Compile JCL \*\*\*\*\*\*

//STP0000 EXEC PROC=ELAXFCOC,

// CICS=,

// DB2=,

// COMP=

//COBOL.SYSPRINT DD SYSOUT=\*

//SYSLIN DD DISP=SHR,

// DSN=&SYSUID..COBOBJS.OBJ(&COBPGM.)

//COBOL.SYSLIB DD DISP=SHR,

// DSN=&SYSUID..COBOL.COPYLIB

//COBOL.SYSXMLSD DD DUMMY

//COBOL.SYSIN DD DISP=SHR,

// DSN=&SYSUID..LEARN.COBOL(&COBPGM.)

//\*\*\*\*Link/Edit Step \*\*\*\*\*\*

//LKED EXEC PROC=ELAXFLNK

//LINK.SYSLIB DD DSN=CEE.SCEELKED,

// DISP=SHR

// DD DSN=&SYSUID..LEARN.LOAD,

// DISP=SHR

//LINK.OBJ0000 DD DISP=SHR,

// DSN=&SYSUID..COBOBJS.OBJ(&COBPGM.)

//LINK.SYSLIN DD \*

INCLUDE OBJ0000

/\*

//LINK.SYSLMOD DD DISP=SHR,

// DSN=&SYSUID..LEARN.LOAD(&COBPGM.)

//\*

//\*\* Go (Run) Step. Add //DD cards when needed \*\*\*\*\*\*

//GO EXEC PROC=ELAXFGO,GO=&COBPGM.,

// LOADDSN=&SYSUID..LEARN.LOAD

//SYSIN DD DISP=SHR,

// DSN=&SYSUID..LEARN.SYSIN35

//\*\*\*\*\*\*\* ADDITIONAL RUNTIME JCL HERE \*\*\*\*\*\*

//PAYROLL DD DSN=USER35.LEARN.PAYROL01,DISP=SHR

//EMPTYFIL DD DSN=USER35.LEARN.EMPTYFIL,DISP=SHR

//PAYROL3A DD DSN=USER35.LEARN.PAYROL3A,DISP=SHR

//PAYCHECK DD SYSOUT=\*

//ACCTREC DD DSN=USER35.LEARN.ACCT.DATA,DISP=SHR

//INFILE1 DD DSN=USER35.LEARN.ACCT.DATA,DISP=SHR

//OUTFILE DD DSN=USER35.LEARN.OUTFILE,DISP=SHR

//INVALS DD DSN=USER35.LEARN.INVALS,DISP=SHR

//PASSFILE DD DSN=USER35.LEARN.PASSFILE,DISP=SHR

//FAVIN DD DSN=USER35.LEARN.FAVIN,DISP=SHR

//FAVRPT DD SYSOUT=\*

//PRTLINE DD SYSOUT=\*

//B37 DD DSN=USER35.LEARN.B37,DISP=SHR

//CLAIM DD DSN=USER35.LEARN.INSCLAIM,DISP=SHR

//\*RFPIN DD DSN=&SYSUID..LEARN.FAVRFP,DISP=SHR

//RFPIN DD \*

11111111111111111111111111111111111111111111111111111111111111111111111

22222222222222222222222222222222222222222222222222222222222222222222222

33333333333333333333333333333333333333333333333333333333333333333333333

44444444444444444444444444444444444444444444444444444444444444444444444

//PROPOSAL DD SYSOUT=\*

//CLAIMRPT DD SYSOUT=\*

AND THE PROGRAM

IDENTIFICATION DIVISION.

PROGRAM-ID. USERPASS.

\*\*\*\*\* This MODULE Workshop 8 - Create new COBOL program

\*\*\*\*\* THAT CHECK FOR USER CREDENTIALS (NAME/PASSWORD)

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT PASS-REC ASSIGN TO PASSFILE

FILE STATUS IS PASSFILE-F-STATUS.

SELECT OUT2 ASSIGN TO PRTLINE.

DATA DIVISION.

FILE SECTION.

FD PASS-REC

RECORDING MODE IS F

LABEL RECORDS ARE STANDARD

RECORD CONTAINS 80 CHARACTERS

BLOCK CONTAINS 0 RECORDS

DATA RECORD IS PASSFILE-REC.

01 PASSFILE-REC.

05 PFILE-USER PIC X(10) .

05 PFILE-PASS PIC X(10).

FD OUT2

RECORDING MODE IS F

LABEL RECORDS ARE STANDARD

RECORD CONTAINS 80 CHARACTERS

BLOCK CONTAINS 0 RECORDS

DATA RECORD IS OUTFILE-REC.

01 OUTFILE-REC.

05 FILLER PIC X(80).

WORKING-STORAGE SECTION.

01 HEADER-1.

05 FILLER PIC X(30) VALUE SPACES.

05 FILLER PIC X(30) VALUE ' PROCESSING CREDENTIALS '.

05 FILLER PIC X(20) VALUE SPACES.

01 TRAILLER-2.

05 FILLER PIC X(20) VALUE 'ENTERED UserName:'.

05 USER-NAME PIC X(10) VALUE SPACES .

05 FILLER PIC X(20) VALUE 'ENTERED Password:'.

05 PASS-WORD PIC X(10) VALUE SPACES .

05 MESSAGE-RZLT PIC X(30) .

01 TRAILLER-3.

05 FILLER PIC X(30) VALUE 'FROM PASSWORDS FILE UserName:'.

05 USER-NAME-OUT PIC X(10) VALUE SPACES .

05 FILLER PIC X(15) VALUE 'READ Password:'.

05 PASS-WORD-OUT PIC X(10) VALUE SPACES .

77 LASTREC PIC X.

88 NO-MORE-RECORDS VALUE 'Y'.

77 PASSFILE-F-STATUS PIC XX .

77 TRIAL-KTR PIC 9(1) VALUE 0 .

PROCEDURE DIVISION.

OPEN OUTPUT OUT2.

WRITE OUTFILE-REC FROM HEADER-1.

MOVE SPACES TO OUTFILE-REC.

WRITE OUTFILE-REC AFTER ADVANCING 1 LINES.

PERFORM UNTIL TRIAL-KTR = 3 OR USER-NAME IS NUMERIC

OPEN INPUT PASS-REC

\* Prime Read

PERFORM ACCEPT-User-Name

PERFORM READ-RECORD

PERFORM UNTIL LASTREC = 'Y' OR NO-MORE-RECORDS

PERFORM Check-User-Name

PERFORM WRITE-CHK-RZLT-RECORD

PERFORM READ-RECORD

END-PERFORM

CLOSE PASS-REC

MOVE SPACES TO LASTREC

COMPUTE TRIAL-KTR = TRIAL-KTR + 1

END-PERFORM

CLOSE OUT2.

STOP RUN.

ACCEPT-User-Name.

MOVE SPACES TO USER-NAME, PASS-WORD , MESSAGE-RZLT.

DISPLAY 'ENTER USER NAME OR NUMBER TO EXIT'

ACCEPT USER-NAME .

DISPLAY 'ENTER THE PASSWORD '

ACCEPT PASS-WORD .

MOVE ' JUST ENTERED ' TO MESSAGE-RZLT

DISPLAY TRAILLER-2.

WRITE OUTFILE-REC FROM TRAILLER-2 .

READ-RECORD.

READ PASS-REC

\* AT END MOVE 'Y' TO LASTREC

AT END

PERFORM END-OF-FILE

\* NOT AT END PERFORM PROCESS-RECORDS

END-READ.

Check-User-Name.

MOVE PFILE-USER TO USER-NAME-OUT

MOVE PFILE-PASS TO PASS-WORD-OUT

WRITE OUTFILE-REC FROM TRAILLER-3 .

EVALUATE TRUE ALSO TRUE

WHEN USER-NAME NOT = PFILE-USER ALSO TRUE

MOVE 'Wrong ID' TO MESSAGE-RZLT

WHEN USER-NAME = PFILE-USER ALSO PASS-WORD NOT= PFILE-PASS

MOVE 'Bad Password' TO MESSAGE-RZLT

WHEN USER-NAME = PFILE-USER ALSO PASS-WORD = PFILE-PASS

MOVE 'Welcome to COBOL' TO MESSAGE-RZLT

\* SET EXIT CONTER

MOVE 'Y' TO LASTREC

COMPUTE TRIAL-KTR = 2

WHEN OTHER

MOVE 'Other condition' TO MESSAGE-RZLT

END-EVALUATE .

WRITE-CHK-RZLT-RECORD.

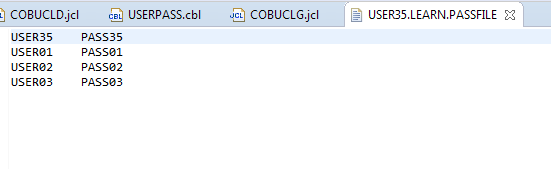
DISPLAY MESSAGE-RZLT

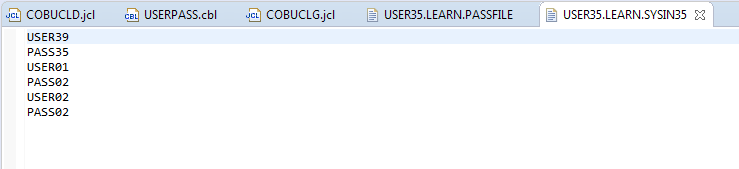
WRITE OUTFILE-REC FROM TRAILLER-2 .

END-OF-FILE.

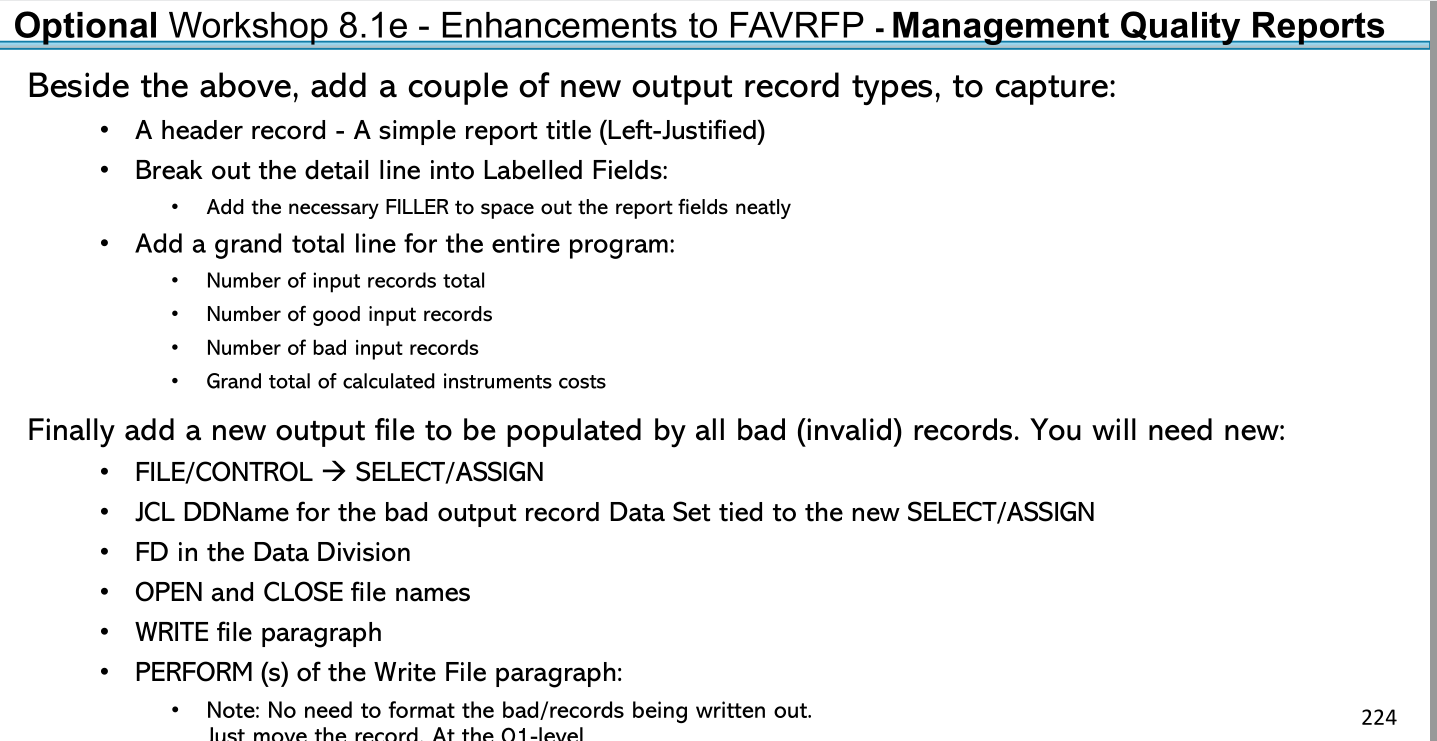
MOVE 'Y' TO LASTREC.

USED FILES



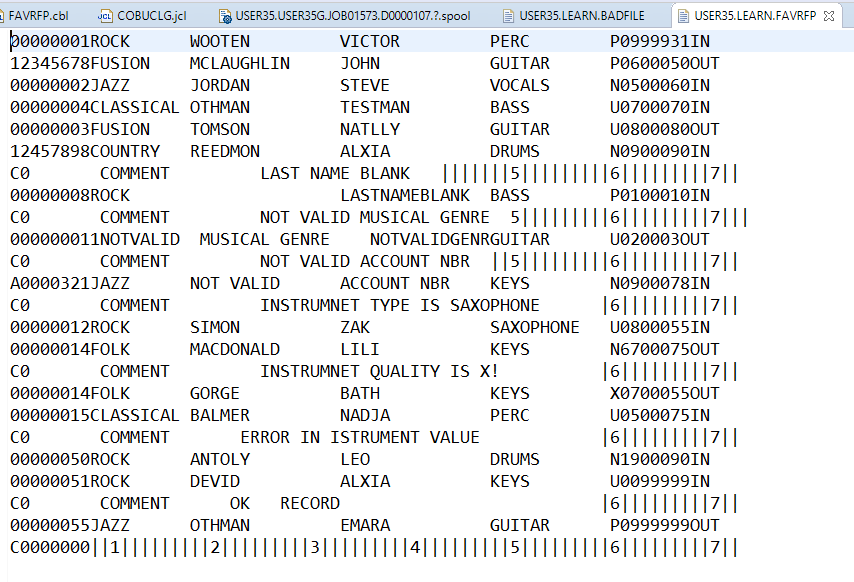


Question 8.1?

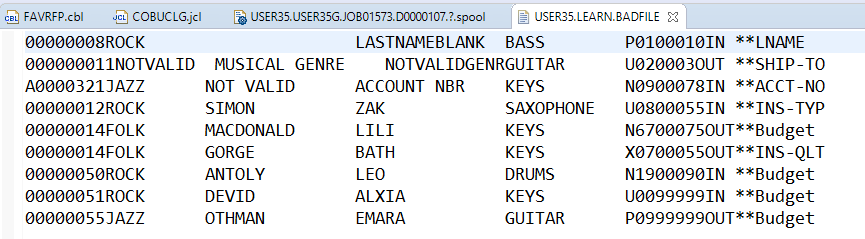


Answer :

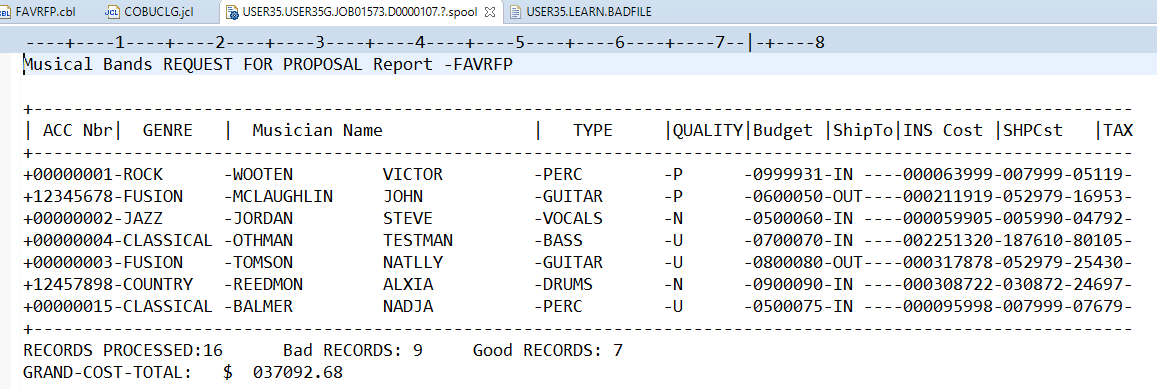
Data to be processed / in file



Identified corrupted records



Generated report



IDENTIFICATION DIVISION.

PROGRAM-ID. FAVRFP.

\*\*\*\*\* This MODULE Workshop 8.1 - Create new COBOL program

\*\*\*\*\* THAT USES A FILE READ - IF STATMENTS / EVALUATE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT RFP-INFILE ASSIGN TO RFPIN

FILE STATUS IS RFPIN-F-STATUS.

SELECT BADFILE-OUT ASSIGN TO BADFILE

FILE STATUS IS BAD-F-STATUS.

SELECT RFP-RPTOUT ASSIGN TO PROPOSAL.

DATA DIVISION.

FILE SECTION.

FD BADFILE-OUT

RECORDING MODE IS F

LABEL RECORDS ARE STANDARD

RECORD CONTAINS 80 CHARACTERS

BLOCK CONTAINS 0 RECORDS

DATA RECORD IS BAD-REC.

01 BAD-REC PIC X(80).

FD RFP-INFILE

RECORDING MODE IS F

LABEL RECORDS ARE STANDARD

RECORD CONTAINS 80 CHARACTERS

BLOCK CONTAINS 0 RECORDS

DATA RECORD IS RFPIN-REC.

01 RFPIN-REC.

05 ARTIST-ACCT-NO.

10 RECORD-TYPE-FLAG PIC X(01).

88 COMMENTED-RECORD VALUE 'C'.

10 REST-OF-ARTIST-ACCT-NO PIC X(07).

05 MUSICAL-GENRE PIC X(10).

88 ROCK VALUE 'ROCK'.

88 JAZZ VALUE 'JAZZ'.

88 FUSION VALUE 'FUSION'.

88 FOLK VALUE 'FOLK'.

88 CLASSICAL VALUE 'CLASSICAL'.

88 COUNTRY VALUE 'COUNTRY'.

05 MUSICIAN.

10 MUSICIAN-LNAME PIC X(15).

10 MUSICIAN-FNAME PIC X(15).

05 MUSICIAN-INSTRUMENT-TYPE PIC X(12).

88 DRUMS VALUE 'DRUMS'.

88 GUITAR VALUE 'GUITAR'.

88 BASS VALUE 'BASS'.

88 VOCALS VALUE 'VOCALS'.

88 KEYBOARD VALUE 'KEYS'.

88 PERCUSSION VALUE 'PERC'.

05 INSTRUMENT-QUALITY PIC X.

88 USED-FLAG VALUE 'U'.

88 NEW-FLAG VALUE 'N'.

88 PREMIUM-FLAG VALUE 'P'.

05 Max-Musician-Budget PIC 9(5)V99.

05 SHIP-TO PIC X(3).

88 IN-COUNTRY VALUE 'IN'.

88 OUT-OF-COUNTRY VALUE 'OUT'.

05 ERROR-CAUSE PIC X(9) .

FD RFP-RPTOUT

RECORDING MODE IS F

LABEL RECORDS ARE STANDARD

RECORD CONTAINS 120 CHARACTERS

BLOCK CONTAINS 0 RECORDS

DATA RECORD IS RFPOUT-REC.

01 RFPOUT-REC.

05 FILLER PIC X VALUE '|' .

05 ARTIST-ACCT-NO-O PIC X(08).

05 FILLER PIC X VALUE '|' .

05 MUSICAL-GENRE-O PIC X(10).

05 FILLER PIC X VALUE '|' .

05 MUSICIAN-O PIC X(30) .

05 FILLER PIC X VALUE '|' .

05 MUSICIAN-INSTRUMENT-TYPE-O PIC X(12).

05 FILLER PIC X VALUE '|' .

05 INSTRUMENT-QUALITY-O PIC X(7).

05 FILLER PIC X VALUE '|' .

05 Max-Musician-Budget-AMOUNT-o PIC 9(5)V99.

05 FILLER PIC X VALUE '|' .

05 SHIP-TO-o PIC X(3).

88 IN-COUNTRY-o VALUE 'IN'.

88 OUT-OF-COUNTRY-o VALUE 'OUT'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC XXX VALUE ' ' .

05 COST-P-INST-O PIC 9(7)V99.

05 FILLER PIC X VALUE '|' .

05 SHIPPING-COST-O PIC 9(4)V99.

05 FILLER PIC X VALUE '|' .

05 TAX-O PIC 9(3)V99.

05 FILLER PIC X VALUE '|' .

\*

WORKING-STORAGE SECTION.

01 HEADER-1.

05 FILLER PIC X(22) VALUE 'Musical Bands REQUEST '.

05 FILLER PIC X(27) VALUE 'FOR PROPOSAL Report -FAVRFP'.

05 FILLER PIC X(20) VALUE SPACES.

\*

01 BOARDR-LINE.

05 FILLER PIC X(30) VALUE '+-----------------------------'.

05 FILLER PIC X(30) VALUE '------------------------------'.

05 FILLER PIC X(30) VALUE '------------------------------'.

05 FILLER PIC X(30) VALUE '-----------------------------+'.

\*

01 COLUMS-HEADING.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(08) VALUE ' ACC Nbr'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(10) VALUE ' GENRE '.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(30) VALUE ' Musician Name '.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(12) VALUE ' TYPE '.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(7) VALUE 'QUALITY'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(7) VALUE 'Budget'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(6) VALUE 'ShipTo'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(9) VALUE 'INS Cost'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(9) VALUE 'SHPCst'.

05 FILLER PIC X VALUE '|' .

05 FILLER PIC X(9) VALUE 'TAXCst'.

05 FILLER PIC X VALUE '|' .

\*

01 TRAILLER-1.

05 FILLER PIC X(18) VALUE 'RECORDS PROCESSED:'.

05 REC-KTR-ALL-O PIC z9 VALUE ZEROS .

05 FILLER PIC X(18) VALUE ' Bad RECORDS:'

JUSTIFIED RIGHT.

05 REC-KTR-BAD-O PIC z9 VALUE ZEROS .

05 FILLER PIC X(18) VALUE ' Good RECORDS:'

JUSTIFIED RIGHT.

05 REC-KTR-GOOD-O PIC z9 VALUE ZEROS .

\*

01 TRAILLER-2.

05 FILLER PIC X(20) VALUE 'GRAND-COST-TOTAL:'.

05 GRAND-COST-TOTAL-O PIC $ZZ9(6).99 VALUE ZEROS .

\*

77 RFPIN-REC2 PIC X(80).

77 RFPIN-F-STATUS PIC XX.

77 BAD-F-STATUS PIC XX.

77 LASTREC PIC X VALUE ' '.

88 NO-MORE-RECORDS VALUE 'Y'.

77 ERRORROTIC-FLAG PIC X VALUE ' '.

88 THERE-IS-ERROR VALUE 'Y'.

77 REC-KTR-ALL PIC 99 VALUE ZEROS .

77 REC-KTR-BAD PIC 99 VALUE ZEROS .

77 REC-KTR-GOOD PIC 99 VALUE ZEROS .

77 GRAND-COST-TOTAL PIC 9(6)V99 VALUE ZEROS .

PROCEDURE DIVISION.

OPEN INPUT RFP-INFILE.

OPEN OUTPUT RFP-RPTOUT.

OPEN OUTPUT BADFILE-OUT.

INITIALIZE TRAILLER-1 , TRAILLER-2 .

WRITE RFPOUT-REC FROM HEADER-1.

MOVE SPACES TO RFPOUT-REC.

WRITE RFPOUT-REC AFTER ADVANCING 1 LINES.

\* Prime Read

PERFORM READ-RECORD.

PERFORM UNTIL LASTREC = 'Y' OR NO-MORE-RECORDS

PERFORM PROCESS-RCDS

PERFORM WRITE-RPT-OUT-REC

PERFORM READ-RECORD

END-PERFORM

PERFORM WRITE-LAST-REC

PERFORM CLOSE-FILES

STOP RUN.

WRITE-TABLE-BORDER-COL-HEAD.

WRITE RFPOUT-REC FROM BOARDR-LINE .

WRITE RFPOUT-REC FROM COLUMS-HEADING .

WRITE RFPOUT-REC FROM BOARDR-LINE .

READ-RECORD.

READ RFP-INFILE

AT END PERFORM END-OF-FILE

\* NOT AT END PERFORM PROCESS-RCDS

END-READ.

PROCESS-RCDS.

IF NOT COMMENTED-RECORD

PERFORM PROCSS-A-NOT-COMMENTED-RCD

END-IF.

PROCSS-A-NOT-COMMENTED-RCD .

COMPUTE REC-KTR-ALL = REC-KTR-ALL + 1 .

MOVE SPACES TO ERRORROTIC-FLAG

PERFORM CHECK-IS-VALID-REC.

IF THERE-IS-ERROR THEN

COMPUTE REC-KTR-BAD = REC-KTR-BAD + 1

WRITE BAD-REC FROM RFPIN-REC

ELSE

COMPUTE REC-KTR-GOOD = REC-KTR-GOOD + 1

COMPUTE GRAND-COST-TOTAL = GRAND-COST-TOTAL +

TAX-O + SHIPPING-COST-O + COST-P-INST-O

END-IF.

WRITE-RPT-OUT-REC.

IF REC-KTR-GOOD = 1 then

PERFORM WRITE-TABLE-BORDER-COL-HEAD

End-if.

IF REC-KTR-GOOD >= 1 AND NOT THERE-IS-ERROR AND

NOT COMMENTED-RECORD THEN

MOVE ARTIST-ACCT-NO TO ARTIST-ACCT-NO-O

MOVE MUSICAL-GENRE TO MUSICAL-GENRE-O

MOVE MUSICIAN TO MUSICIAN-O

MOVE MUSICIAN-INSTRUMENT-TYPE

TO MUSICIAN-INSTRUMENT-TYPE-O

MOVE INSTRUMENT-QUALITY TO INSTRUMENT-QUALITY-O

MOVE Max-Musician-Budget TO

Max-Musician-Budget-AMOUNT-o

MOVE SHIP-TO TO SHIP-TO-o

PERFORM MAKE-REC-CALC

WRITE RFPOUT-REC

END-IF.

CLOSE-FILES.

CLOSE RFP-INFILE.

CLOSE RFP-RPTOUT.

CLOSE BADFILE-OUT.

END-OF-FILE.

MOVE 'Y' TO LASTREC.

WRITE-LAST-REC.

\* last Record Calculations

WRITE RFPOUT-REC FROM BOARDR-LINE .

MOVE REC-KTR-ALL TO REC-KTR-ALL-O.

MOVE REC-KTR-BAD TO REC-KTR-BAD-O.

MOVE REC-KTR-GOOD TO REC-KTR-GOOD-O.

MOVE GRAND-COST-TOTAL TO GRAND-COST-TOTAL-O.

WRITE RFPOUT-REC FROM TRAILLER-1.

\* FILL TRAILLER-2

WRITE RFPOUT-REC FROM TRAILLER-2 .

CHECK-IS-VALID-REC.

\* HERE WE PUT ALL RECORDS VALIDATION RULES

MOVE SPACES TO ERRORROTIC-FLAG

MOVE SPACES TO ERROR-CAUSE

EVALUATE TRUE

WHEN NOT ARTIST-ACCT-NO IS NUMERIC

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*ACCT-NO' TO ERROR-CAUSE

WHEN MUSICIAN-LNAME IS ALPHABETIC

AND MUSICIAN-LNAME = SPACES

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*LNAME' TO ERROR-CAUSE

WHEN MUSICIAN-FNAME IS ALPHABETIC

AND MUSICIAN-FNAME = SPACES

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*FNAME' TO ERROR-CAUSE

WHEN NOT (Max-Musician-Budget >= 1000

AND Max-Musician-Budget < 9999.99)

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*Budget' TO ERROR-CAUSE

END-EVALUATE.

EVALUATE TRUE

WHEN MUSICAL-GENRE = 'ROCK' OR MUSICAL-GENRE = 'FOLK' OR

MUSICAL-GENRE = 'JAZZ' OR MUSICAL-GENRE = 'FUSION' OR

MUSICAL-GENRE = 'CLASSICAL' OR MUSICAL-GENRE = 'COUNTRY'

MOVE MUSICAL-GENRE TO MUSICAL-GENRE

WHEN OTHER

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*GENRE' TO ERROR-CAUSE

END-EVALUATE.

\*

EVALUATE TRUE

WHEN INSTRUMENT-QUALITY ='P' OR INSTRUMENT-QUALITY = 'U'

OR INSTRUMENT-QUALITY = 'N'

MOVE INSTRUMENT-QUALITY TO INSTRUMENT-QUALITY

WHEN OTHER

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*INS-QLTY' TO ERROR-CAUSE

END-EVALUATE.

\*

EVALUATE TRUE

WHEN SHIP-TO = 'IN ' or SHIP-TO = 'OUT'

MOVE SHIP-TO TO SHIP-TO

WHEN OTHER

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*SHIP-TO' TO ERROR-CAUSE

END-EVALUATE.

\*

EVALUATE TRUE

WHEN MUSICIAN-INSTRUMENT-TYPE = 'DRUMS' or

MUSICIAN-INSTRUMENT-TYPE = 'GUITAR' or

MUSICIAN-INSTRUMENT-TYPE = 'BASS' or

MUSICIAN-INSTRUMENT-TYPE = 'VOCALS' or

MUSICIAN-INSTRUMENT-TYPE = 'KEYS' or

MUSICIAN-INSTRUMENT-TYPE = 'PERC'

MOVE MUSICIAN-INSTRUMENT-TYPE TO MUSICIAN-INSTRUMENT-TYPE

WHEN OTHER

MOVE 'Y' TO ERRORROTIC-FLAG

MOVE '\*\*INS-TYP' TO ERROR-CAUSE

END-EVALUATE.

\*

MAKE-REC-CALC.

EVALUATE MUSICIAN-INSTRUMENT-TYPE

WHEN 'DRUMS'

COMPUTE COST-P-INST-O = 3087.22

WHEN 'GUITAR'

COMPUTE COST-P-INST-O = 2648.99

WHEN 'BASS'

COMPUTE COST-P-INST-O = 18761

WHEN 'VOCALS'

COMPUTE COST-P-INST-O = 599.05

WHEN 'KEYS'

COMPUTE COST-P-INST-O =3017.89

WHEN 'PERC'

COMPUTE COST-P-INST-O = 799.99

END-EVALUATE.

\*

EVALUATE SHIP-TO

WHEN 'IN '

COMPUTE SHIPPING-COST-O = COST-P-INST-O \* 0.1

WHEN 'OUT'

COMPUTE SHIPPING-COST-O = COST-P-INST-O \* 0.2

END-EVALUATE.

\* \*

EVALUATE INSTRUMENT-QUALITY

WHEN 'P'

COMPUTE COST-P-INST-O = COST-P-INST-O \* 0.8

WHEN 'U'

COMPUTE COST-P-INST-O = COST-P-INST-O \* 1.2

END-EVALUATE.

\*

COMPUTE TAX-O = COST-P-INST-O \* 0.08.