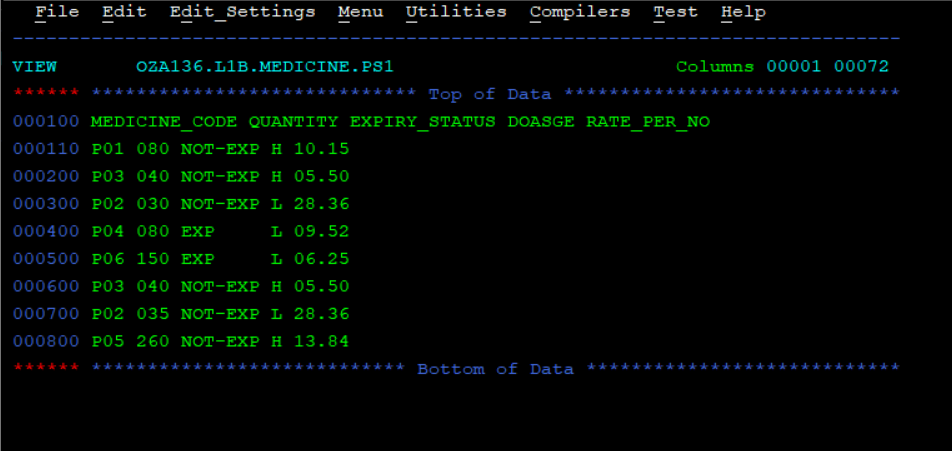
**Question No: 1  
  
PART A:**

**STEP1: Creating a PS File**

OZA136.L1B.MEDICINE.PS1:

****

**STEP2: JA11B136**

//OZA136A JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M

//STEP01 EXEC PGM=IDCAMS

//SYSIN DD \*

DEFINE CLUSTER -

(NAME(OZA136.L1B.MEDICINE.NEWORDER.KSDS) -

KEYS(3,0) -

CYLINDERS(5,10) -

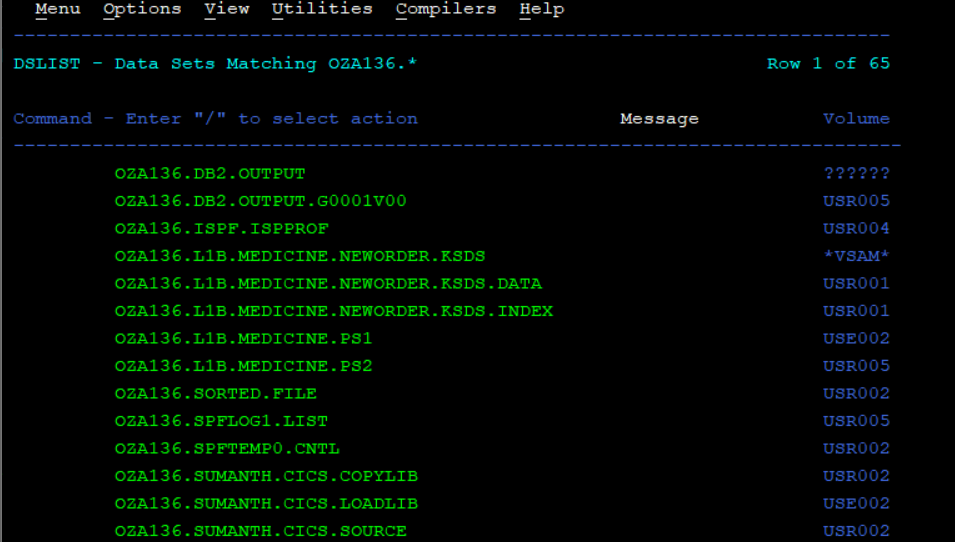
INDEXED -

RECSZ(80,80) -

CISZ(4096))

/\*

//SYSPRINT DD SYSOUT=\*



**STEP3:**

1. **JA21B136**

//OZA136B JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M

//STEP1 EXEC PGM=IEFBR14

//PS2FILE DD DSN=OZA136.L1B.MEDICINE.PS2,

// DISP=(MOD,DELETE,DELETE),

// SPACE=(CYL,(1,1),RLSE)

//STEP2 EXEC PGM=SORT

//SYSOUT DD SYSOUT=\*

//SORTIN DD DSN=OZA136.L1B.MEDICINE.PS1,DISP=SHR

//SORTOUT DD DSN=OZA136.L1B.MEDICINE.PS2,

// DISP=(NEW,CATLG,DELETE),

// SPACE=(CYL,(5,5),RLSE),

// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

//SYSIN DD \*

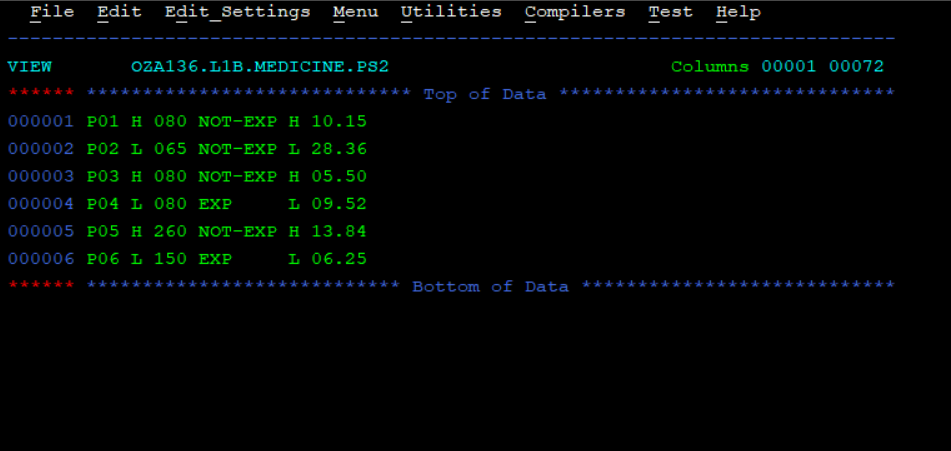
SORT FIELDS=(1,3,CH,A)

SUM FIELDS=(5,3,ZD)

OMIT COND=(1,12,CH,EQ,C'MEDICINE\_CODE')

OUTREC FIELDS=(1,3,1X,17,1,1X,5,3,1X,9,7,1X,17,1,1X,19,5)

/\*

OZA136.L1B.MEDICINE.PS2:  


1. **JA31B136**

//OZA136C JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M

//STEP1 EXEC PGM=IEFBR14

//LOWD DD DSN=OZA136.L1B.MEDICINE.LOWDOS,

// DISP=(MOD,DELETE,DELETE),

// SPACE=(CYL,(1,1),RLSE)

//HIGHD DD DSN=OZA136.L1B.MEDICINE.HIGHDOS,

// DISP=(MOD,DELETE,DELETE),

// SPACE=(CYL,(1,1),RLSE)

//STEP2 EXEC PGM=SORT

//SYSOUT DD SYSOUT=\*

//SORTIN DD DSN=OZA136.L1B.MEDICINE.PS2,DISP=SHR

//LOWO DD DSN=OZA136.L1B.MEDICINE.LOWDOS,

// DISP=(NEW,CATLG,DELETE),

// SPACE=(CYL,(1,1),RLSE),

// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

//HIGHO DD DSN=OZA136.L1B.MEDICINE.HIGHDOS,

// DISP=(NEW,CATLG,DELETE),

// SPACE=(CYL,(1,1),RLSE),

// DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

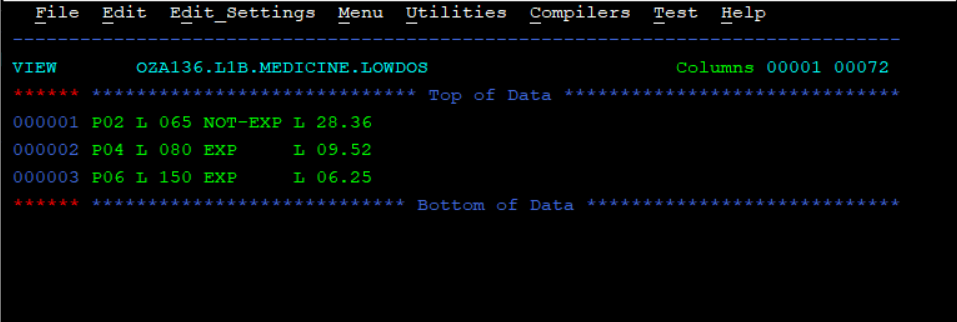
//SYSIN DD \*

SORT FIELDS=COPY

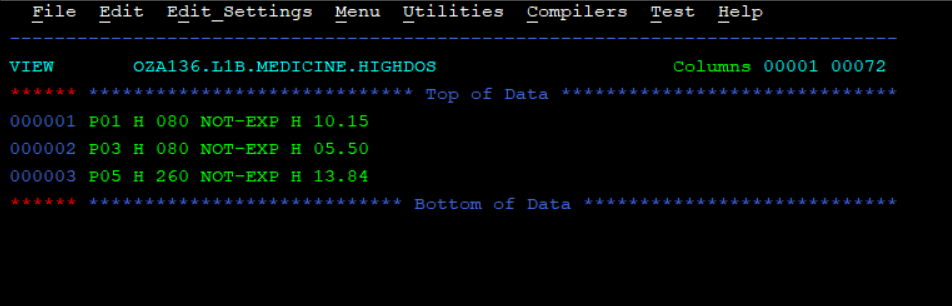
OUTFIL FNAMES=LOWO,INCLUDE=(5,1,CH,EQ,C'L')

OUTFIL FNAMES=HIGHO,INCLUDE=(5,1,CH,EQ,C'H')

/\*

OZA136.L1B.MEDICINE.LOWDOS:  


OZA136.L1B.MEDICINE.HIGHDOS:



**STEP4:**

000100 IDENTIFICATION DIVISION.

000200 PROGRAM-ID. MEDC.

000300 AUTHOR. SUMANTH.

000400

000500 ENVIRONMENT DIVISION.

000600 INPUT-OUTPUT SECTION.

000700 FILE-CONTROL.

000800 SELECT INPMEDPS ASSIGN TO INPMEDPS

000900 ORGANIZATION IS SEQUENTIAL

001000 ACCESS MODE IS SEQUENTIAL

001100 FILE STATUS IS WS-INP-FILE-STATUS.

001200

001210 SELECT OUTMEDPS ASSIGN TO OUTMEDPS

001220 ORGANIZATION IS INDEXED

001230 ACCESS MODE IS SEQUENTIAL

001231 RECORD KEY IS OUT-MEDICINE-CODE

001240 FILE STATUS IS WS-OUT-FILE-STATUS.

001241

001242 SELECT ERRMEDPS ASSIGN TO ERRMEDPS

001243 ORGANIZATION IS SEQUENTIAL

001244 ACCESS MODE IS SEQUENTIAL

001246 FILE STATUS IS WS-ERR-FILE-STATUS.

001250

001300 DATA DIVISION.   
001400 FILE SECTION.

001500 FD INPMEDPS

001600 RECORDING MODE IS F

001700 RECORD CONTAINS 80 CHARACTERS.

001800 01 INP-RECORD.

001900 05 INP-MEDICINE-CODE PIC X(03).

002000 05 FILLER PIC X(01).

002010 05 INP-DOSAGE PIC X(01).

002100 05 FILLER PIC X(01).

002110 05 INP-QUANTITY PIC 9(03).

002200 05 FILLER PIC X(01).

002210 05 INP-EXPIRY PIC X(07).

002300 05 FILLER PIC X(01).

002310 05 INP-STATUS PIC X(01).

002400 05 FILLER PIC X(01).

002410 05 INP-RATE-PER-NO PIC 9(02)V9(02).

002500 05 FILLER PIC X(55).

002600

002700 FD OUTMEDPS

002800 RECORDING MODE IS F

002900 RECORD CONTAINS 80 CHARACTERS.

003000 01 OUT-RECORD.

003100 05 OUT-MEDICINE-CODE PIC X(03).

003200 05 FILLER PIC X(01).

003210 05 OUT-QUANTITY PIC X(03).   
003300 05 FILLER PIC X(01).

003310 05 OUT-EXPIRY-STATUS PIC X(07).

003400 05 FILLER PIC X(01).

003410 05 OUT-ORDER-STATUS PIC X(07).

003500 05 FILLER PIC X(01).

003510 05 OUT-RATE-PER-NO PIC 9(02)V9(02).

003600 05 FILLER PIC X(01).

003610 05 OUT-DISC-RATE-PER-NO PIC 9(02)V9(02).

003700 05 FILLER PIC X(01).

003710 05 OUT-TOT-PRICE PIC 9(05)V9(02).

003800 05 FILLER PIC X(01).

003900

004000 FD ERRMEDPS

004100 RECORDING MODE IS F

004200 RECORD CONTAINS 80 CHARACTERS.

004300 01 ERR-RECORD PIC X(80).

004400

004500 WORKING-STORAGE SECTION.

004600 01 WS-INP-FILE-STATUS PIC XX VALUE SPACES.

004700 01 WS-OUT-FILE-STATUS PIC XX VALUE SPACES.

004800 01 WS-ERR-FILE-STATUS PIC XX VALUE SPACES.

004900 01 WS-EOF PIC X VALUE 'N'.

005000 88 EOF-Y VALUE 'Y'.

005100 88 EOF-N VALUE 'N'.

005200 01 WS-VALID-RECORD PIC X VALUE 'Y'.   
005300 88 VALID-RECORD VALUE 'Y'.

005400 88 INVALID-RECORD VALUE 'N'.

005500 01 WS-NEW-QUANTITY PIC 9(03) VALUE 0.

005600 01 WS-DISC-RATE-PER-NO PIC 9(02)V9(02) VALUE 0.

005700 01 WS-TOT-PRICE PIC 9(05)V9(02) VALUE 0.

005800 01 WS-RATE-PER-NO-CHECK.

005900 05 WS-RATE-BEFORE-DEC PIC 9(02).

006000 05 WS-RATE-DECIMAL PIC X(01).

006100 05 WS-RATE-AFTER-DEC PIC 9(02).

006200 01 WS-RATE-PER-NO-STR PIC X(05).

006300

006400 PROCEDURE DIVISION.

006500 MAIN-PARA.

006600 PERFORM INIT

006700 PERFORM PROCESS-RECORDS

006800 UNTIL EOF-Y

006900 PERFORM TERMINAT

007000 STOP RUN.

007100

007200 INIT.

007300 OPEN INPUT INPMEDPS

007400 IF WS-INP-FILE-STATUS NOT = '00'

007500 DISPLAY 'ERROR OPENING INPUT FILE: ' WS-INP-FILE-STATUS

007600 MOVE 'Y' TO WS-EOF

007700 END-IF   
007800 OPEN OUTPUT OUTMEDPS

007900 IF WS-OUT-FILE-STATUS NOT = '00'

008000 DISPLAY 'ERROR OPENING OUTPUT KSDS: ' WS-OUT-FILE-STATUS

008100 MOVE 'Y' TO WS-EOF

008200 END-IF

008300 OPEN OUTPUT ERRMEDPS

008400 IF WS-ERR-FILE-STATUS NOT = '00'

008500 DISPLAY 'ERROR OPENING ERROR FILE: ' WS-ERR-FILE-STATUS

008600 MOVE 'Y' TO WS-EOF

008700 END-IF

008800 PERFORM READ-RECORD.

008900

009000 READ-RECORD.

009100 READ INPMEDPS

009200 AT END

009300 MOVE 'Y' TO WS-EOF

009400 NOT AT END

009500 CONTINUE

009600 END-READ

009700 IF WS-INP-FILE-STATUS NOT = '00' AND

009800 WS-INP-FILE-STATUS NOT = '10'

009900 DISPLAY 'ERROR READING INPUT FILE: ' WS-INP-FILE-STATUS

010000 MOVE 'Y' TO WS-EOF

010100 END-IF.

010200   
010300 PROCESS-RECORDS.

010400 MOVE 'Y' TO WS-VALID-RECORD

010500 PERFORM VALIDATE-RECORD

010600 IF INVALID-RECORD

010700 PERFORM WRITE-ERROR-RECORD

010800 END-IF

010900 IF VALID-RECORD

011000 PERFORM PROCESS-VALID-RECORD

011100 END-IF

011200 PERFORM READ-RECORD.

011300

011400 VALIDATE-RECORD.

011500 PERFORM CHECK-MEDICINE-CODE

011600 IF VALID-RECORD

011700 PERFORM CHECK-DOSAGE

011800 END-IF

011900 IF VALID-RECORD

012000 PERFORM CHECK-QUANTITY

012100 END-IF

012200 IF VALID-RECORD

012300 PERFORM CHECK-EXPIRY-STATUS

012400 END-IF

012500 IF VALID-RECORD

012600 PERFORM CHECK-RATE-PER-NO

012700 END-IF.   
012800

012900 CHECK-MEDICINE-CODE.

013000 IF INP-MEDICINE-CODE = SPACES

013100 MOVE 'N' TO WS-VALID-RECORD

013200 DISPLAY 'INVALID MEDICINE CODE: BLANK FOR RECORD'

013300 END-IF.

013400

013500 CHECK-DOSAGE.

013600 IF INP-DOSAGE NOT = 'L' AND INP-DOSAGE NOT = 'H'

013700 MOVE 'N' TO WS-VALID-RECORD

013800 DISPLAY 'INVALID DOSAGE: ' INP-DOSAGE

013900 END-IF.

014000

014010 CHECK-QUANTITY.

014020 IF INP-QUANTITY NOT NUMERIC

014030 MOVE 'N' TO WS-VALID-RECORD

014040 DISPLAY 'INVALID QUANTITY: ' INP-QUANTITY

014050 END-IF.

014060

014061 CHECK-EXPIRY-STATUS.

014062 IF INP-EXPIRY NOT = 'EXP' AND

014063 INP-EXPIRY NOT = 'NOT-EXP'

014064 MOVE 'N' TO WS-VALID-RECORD

014065 DISPLAY 'INVALID EXPIRY STATUS: ' INP-EXPIRY

014066

014070 CHECK-RATE-PER-NO.

014071 MOVE INP-RATE-PER-NO TO WS-RATE-PER-NO-STR

014072 MOVE WS-RATE-PER-NO-STR TO WS-RATE-PER-NO-CHECK

014080 IF WS-RATE-BEFOR-DEC NOT NUMERIC OR

014081 WS-RATE-AFTER-DEC NOT NUMERIC

014090 MOVE 'N' TO WS-VALID-RECORD   
014091 DISPLAY 'INVALID RATE-PER-NO FORMAT: ' WS-RATE-PER-NO-STR

014092 END-IF.

014093 IF WS-RATE-DECIMAL NOT = '.'

014100 MOVE 'N' TO WS-VALID-RECORD

014200 DISPLAY 'INVALID RATE-PER-NO FORMAT: ' WS-RATE-PER-NO-STR

014300 END-IF.

014400

014500 WRITE-ERROR-RECORD.

014600 MOVE INP-RECORD TO ERR-RECORD

014700 WRITE ERR-RECORD

014800 IF WS-ERR-FILE-STATUS NOT = '00'

014900 DISPLAY 'ERROR WRIRTING TO ERRORFILE:' WS-ERR-FILE-STATUS

015000 MOVE 'Y' TO WS-EOF

015100 END-IF.

015200

015300 PROCESS-VALID-RECORD.

015400 PERFORM APPLY-QUANTITY-RULES

015500 IF WS-NEW-QUANTITY NOT = 0 OR INP-EXPIRY = 'EXP'

015600 PERFORM CALL-SUBPROGRAM

015700 PERFORM WRITE-OUTPUT-RECORD

015800 END-IF.

015900

016000 APPLY-QUANTITY-RULES.

016100 MOVE 0 TO WS-NEW-QUANTITY

016200 IF INP-QUANTITY < 100 AND INP-EXPIRY = 'NOT-EXP'   
016300 MOVE 250 TO WS-NEW-QUANTITY

016400 MOVE 'REORDER' TO OUT-ORDER-STATUS

016500 END-IF

016600 IF INP-QUANTITY < 100 AND INP-EXPIRY = 'EXP'

016700 MOVE 50 TO WS-NEW-QUANTITY

016800 MOVE 'REPLACE' TO OUT-ORDER-STATUS

016900 END-IF

017000 IF INP-QUANTITY > 100 AND INP-EXPIRY = 'EXP'

017100 MOVE 0 TO WS-NEW-QUANTITY

017200 MOVE 'STPSELL' TO OUT-ORDER-STATUS

017300 END-IF

017400

017500 CALL-SUBPROGRAM.

017600 CALL 'MEDSUB' USING INP-RATE-PER-NO WS-NEW-QUANTITY

017700 WS-DISC-RATE-PER-NO WS-TOT-PRICE.

017800

017900 WRITE-OUTPUT-RECORD.

018000 MOVE INP-MEDICINE-CODE TO OUT-MEDICINE-CODE

018100 MOVE WS-NEW-QUANTITY TO OUT-QUNATITY

018200 MOVE INP-EXPIRY TO OUT-EXPIRY-STATUS

018300 MOVE INP-RATE-PER-NO TO OUT-DISC-RATE-PER-NO

018400 MOVE WS-TOT-PRICE TO OUT-TOT-PRICE

018500 WRITE OUT-RECORD

018600 IF WS-OUT-FILE-STATUS NOT = '00'

018700 DISPLAY 'ERROR WRITING TO KSDS: ' WS-OUT-FILE-STATUS   
018800 MOVE 'Y' TO WS-EOF

018900 END-IF.

019000

019100 TERMINAT.

019200 CLOSE INPMEDPS.

019300 CLOSE OUTMEDPS.

019400 CLOSE ERRMEDPS.   
  
**Subprogram:**

000100 IDENTIFICATION DIVISION.

000200 PROGRAM-ID. MEDSUB.

000300 AUTHOR. SUMANTH.

000400

000500 DATA DIVISION.

000600 LINKAGE SECTION.

000700 01 LS-RATE-PER-NO PIC 9(02)V9(02).

000800 01 LS-NEW-QUANTITY PIC 9(03).

000900 01 LS-DISC-RATE-PER-NO PIC 9(02)V9(02).

001000 01 LS-TOT-PRICE PIC 9(05)V9(02).

001100

001200 PROCEDURE DIVISION USING LS-RATE-PER-NO LS-NEW-QUANTITY

001300 LS-DISC-RATE-PER-NO LS-TOT-PRICE.

001400

001500 MAIN-PARA.

001600 PERFORM CAL-DISCOUNT

001700 PERFORM CAL-TOTAL-PRICE

001800 GOBACK.

001900

002000 CAL-DISCOUNT.

002100 COMPUTE LS-DISC-RATE-PER-NO = LS-RATE-PER-NO -

002200 (S-RATE-PER-NO \* 20 / 100).

002300

002400 CAL-TOTAL-PRICE   
002500 LS-TOT-PRICE ROUNDED = LS-NEW-QUANTITY \*

002600 LS-DISC-RATE-PER-NO.   
  


**PART B:  
  
DB2 QUERY:**

CREATE TABLE STUDENT (

STUDENT\_NUMBER CHAR(5) NOT NULL PRIMARY KEY,

STUDENT\_NAME CHAR(25),

STUDENT\_TOTAL\_MARKS DECIMAL(3,0),

ATHLETIC CHAR(1)

) IN OZA136DB.OZA136TS;

CREATE UNIQUE INDEX STU ON STUDENT(STUDENT\_NUMBER);

INSERT INTO STUDENT VALUES

('S0001', 'JOHN DOE', 200, 'Y');

('S0002', 'JANE SMITH', 250, 'N');

('S0003', 'BOB JONES', 350, 'Y');

('S0004', 'ALICE LEE', 180, 'Y');

('S0005', 'MIKE BROWN', 275, 'Y');

SELECT \* FROM STUDENT;   
  
**COBOL Program:**

000100 IDENTIFICATION DIVISION.

000200 PROGRAM-ID. STUDENT.

000300 AUTHOR. SUMANTH.

000310

000400 ENVIRONMENT DIVISION.

000500 INPUT-OUTPUT SECTION.

000600 FILE-CONTROL.

000700 SELECT REPORT-FILE ASSIGN TO STUDE

000800 ORGANIZATION IS SEQUENTIAL.

000900

001000 DATA DIVISION.

001100 FILE SECTION.

001200 FD REPORT-FILE

001300 RECORDING MODE IS F

001400 RECORD CONTAINS 80 CHARACTERS

001500 BLOCK CONTAINS 0 RECORDS.

001600 01 REPORT-RECORD PIC X(80).

001700

001800 WORKING-STORAGE SECTION.

001900 01 WS-BLANK-LINE PIC X(70) VALUE SPACES.

002000 01 HEADER-1 PIC X(35) VALUE

002100 ' STUDENT MODERATION REPORT'.

002200 01 HEADER-2 PIC X(35) VALUE

002300 ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'.   
002400 01 HEADER-3 PIC X(70) VALUE ALL '\*'.

002500 01 HEADER-4 PIC X(50) VALUE

002600 'STUDENT NUMBER STUDENT NAME TOTAL MARKS'.

002700 01 HEADER-5 PIC X(70) VALUE ALL '\*'.

002800 01 DETAIL-LINE.

002900 05 DET-STUD-NUM PIC X(05).

003000 05 FILLER PIC X(11) VALUE SPACES.

003100 05 DET-STUD-NAME PIC X(12).

003200 05 FILLER PIC X(08) VALUE SPACES.

003300 05 DET-TOTAL-MARKS PIC ZZ9.

003400 05 FILLER PIC X(34) VALUE SPACES.

003500 01 TOTAL-LINE.

003600 05 FILLER PIC X(09) VALUE SPACES.

003700 05 FILLER PIC X(50) VALUE

003800 'TOTAL NUMBER OF STUDENT RECEIVED MODERATION : '.

003900 05 TOT-MOD-COUNT PIC ZZ9.

004000 05 FILLER PIC X(12) VALUE SPACES.

004100

004200 01 WS-MOD-COUNT PIC 9(03) VALUE 0.

004300

004400 EXEC SQL

004500 INCLUDE STUDENT

004600 END-EXEC.

004700

004800 EXEC SQL   
004900 INCLUDE SQLCA

005000 END-EXEC.

005100

005200 EXEC SQL

005300 DECLARE STUD-CSR CURSOR FOR

005400 SELECT STUDENT\_NUMBER,

005500 STUDENT\_NAME,

005600 STUDENT\_TOTAL\_MARKS,

005700 ATHLETIC

005800 FROM STUDENT

005900 FOR UPDATE OF STUDENT\_TOTAL\_MARKS

006000 END-EXEC.

006100

006200 PROCEDURE DIVISION.

006300 MAIN-PARA.

006400 PERFORM INIT-PARA.

006500 PERFORM OPEN-CURSOR-PARA.

006600 PERFORM FETCH-AND-UPDATE-PARA

006700 UNTIL SQLCODE NOT = 0.

006800 PERFORM CLOSE-CURSOR-PARA.

006900 PERFORM END-PARA.

007000

007100 INIT-PARA.

007200 OPEN OUTPUT REPORT-FILE.

007300 WRITE REPORT-RECORD FROM WS-BLANK-LINE.   
007400 WRITE REPORT-RECORD FROM HEADER-1

007500 WRITE REPORT-RECORD FROM HEADER-2

007600 WRITE REPORT-RECORD FROM HEADER-3.

007700 WRITE REPORT-RECORD FROM HEADER-4.

007800 WRITE REPORT-RECORD FROM HEADER-5.

007900

008000 OPEN-CURSOR-PARA.

008100 EXEC SQL

008200 OPEN STUD-CSR

008300 END-EXEC.

008400

008500 FETCH-AND-UPDATE-PARA.

008600 EXEC SQL

008700 FETCH STUD-CSR

008800 INTO :STUDENT-NUMBER,

008900 :STUDENT-NAME,

009000 :STUDENT-TOTAL-MARKS,

009100 :ATHLETIC

009200 END-EXEC.

009300 EVALUATE SQLCODE

009400 WHEN 0

009500 PERFORM PROCESS-STUDENT-RECORD

009600 WHEN 100

009700 CONTINUE

009800 WHEN OTHER   
009900 DISPLAY 'SQL ERROR: ' SQLCODE

010000 END-EVALUATE.

010100

010200 PROCESS-STUDENT-RECORD.

010300 IF ATHLETIC = 'Y'

010400 AND STUDENT-TOTAL-MARKS >= 200

010500 AND STUDENT-TOTAL-MARKS <= 300

010600 COMPUTE STUDENT-TOTAL-MARKS = STUDENT-TOTAL-MARKS \* 1.10

010700 EXEC SQL

010800 UPDATE STUDENT

010900 SET STUDENT\_TOTAL\_MARKS = :STUDENT-TOTAL-MARKS

011000 WHERE CURRENT OF STUD-CSR

011100 END-EXEC

011200 MOVE STUDENT-NUMBER TO DET-STUD-NUM

011300 MOVE STUDENT-NAME TO DET-STUD-NAME

011400 MOVE STUDENT-TOTAL-MARKS TO DET-TOTAL-MARKS

011500 WRITE REPORT-RECORD FROM DETAIL-LINE

011600 ADD 1 TO WS-MOD-COUNT

011700 END-IF.

011800

011900 CLOSE-CURSOR-PARA.

012000 EXEC SQL

012100 CLOSE STUD-CSR

012200 END-EXEC.

012300   
012400 END-PARA.

012500 WRITE REPORT-RECORD FROM WS-BLANK-LINE.

012600 MOVE WS-MOD-COUNT TO TOT-MOD-COUNT.

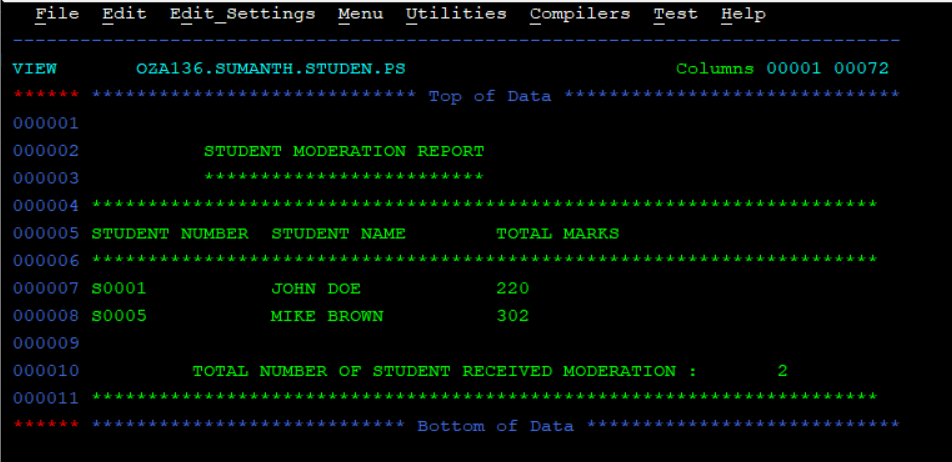
012700 WRITE REPORT-RECORD FROM TOTAL-LINE.

012800 WRITE REPORT-RECORD FROM HEADER-3.

012900 CLOSE REPORT-FILE.

013000 STOP RUN.

**Output:**

****