**SENECA COLLEGE OF APPLIED ARTS AND TECHNOLOGY**

**SCHOOL OF SOFTWARE DESIGN AND DATA SCIENCE**

**FINAL EXAMINATION** 

| TERM | COURSE NAME | COURSE CODE | VERSION |
| --- | --- | --- | --- |
| SUMMER  2021 | IBMi Business Computing | BCI433 | VA |

| Name | Davinder Verma |
| --- | --- |
| Student Number | 121802201 |
| IBMi User ID | DC433C37 |

TIME ALLOWED: 1 hour 30 minutes

PERCENTAGE: 30%

TOTAL MARKS: 30

PROFESSOR: Lydia Li

SPECIAL INSTRUCTIONS:

Write your answers in this document and Email your answers to lydia.li@senecacollege.ca

This exam includes a *cover page*, plus 5 pages of questions.

|  |
| --- |
| SENECA’S ACADEMIC INTEGRITY POLICY |
| As a Seneca student, you must conduct yourself in an honest and trustworthy manner in all aspects of your academic career. A dishonest attempt to obtain an academic advantage is considered an offense, and will not be tolerated by the College. |

**PART A------Short Questions (10 Marks)**

One mark for each

1. What type of variable is set to -1 when a null value is returned in a RPGLE program?

**Ans. Indicator variable is set to -1.**

2. You have a working RPGLE payroll application program and have decided to use an embedded SQL statement to find out how many records are in the SHIFTWEEK file. You insert the correct code in the appropriate spot, but your program does not compile. Why?

**Ans. The program name should be SQLRPGLE instead of rpgle. SQLRPGLE is the filename for the embedded sql statements.**

3. Find out how many rows are in the CUSTOMER table stored in BCI433LIB with embedded SQL in an SQLRPGLE program. Store your answer in CUSTTOT.

**EXEC SQL**

**SELECT COUNT(\*) INTO :CUSTTOT**

**FROM BCI433LIB/CUSTOMER**;

4. How we can find out the user, server, current date and timestamp in an embedded SQL statement?

**EXEC SQL**

**SELECT USER, CURRENT SERVER, CURRENT TIMESTAMP**

**INTO :USER, :SERVER, :TIMESTAMP**

**FROM SYSTEM/SYSDUMMY1;**

5. How do SQLSTATE and SQLCODE inform you that no row was found?

**AND. SQLCODE = +100, SQLSTATE = 02000**

6. The command that will ensure your overflow line is on line 25 for a printer file called REPORT is:

**OFLIND(\*IN25)**

7. The command that will combine modules called RUNPROG1 and PROG1 in a program called FINAL is:

**CRTPGM FINAL MODULE(RUNPROG1 PROG1)**

8. What type of field replaces the function %EOF when processing rows from a cursor?

**IF SQLCODE <> 0 OR SQLWNO = ‘W’;**

9. The CLLE command that will remove all the spooled files from the output queue DM444A01 is:

**CLROUTQ DM444A01**

10. Which command can you check your journal report?

**DSPJRN**

**PART B --------PROGRAMMING (20 MARKS)**

Complete the code using embedded SQL with cursor.

**STUDENTS21.PF:**

**Table

Description automatically generated**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**STGRADES21.PF:**

**Table

Description automatically generated**

**PROGRAM21.PF:**

**Table

Description automatically generated**

The report prints fields from 3 data files: STUDENTS21, STGRADES21, and PROGRAM21, all linked by Student ID. Sorted by Program, Student ID.

For each student print details, calculate and print his/her GPA for all courses in STGRADE21 table. If GPA < 2, flag "at risk" - print “\*” by turning \*IN50 on.

Break when program changes and print totals and average line.

No need to check for NULL values – they are not allowed. No need to create PRTF - it is provided.

Complete the code using embedded SQL with cursor.

**REPORT:**

**Text

Description automatically generated**

**STUDENTREP.PRTF:**

A R REPORTNAME

A 2 43'Students Academic Performance Repo-

A rt 2021'

A R COLHDR SPACEB(2)

A 2'Program'

A 21'Student ID'

A 34'Full Name'

A 86'Age'

A 100'GPA'

A 112'At Risk'

A R DETAILS

A STUDENT# R 21REFFLD(STUDENTR/STUDENT# +

A IKARASIK/STUDENTS21)

A SPACEB(1)

A EDTCDE(M)

A FULLNAME 50 34

A AGE 2 0 86

A GPA 3 2 100

A EDTCDE(2)

A 50 114'\*'

A R TOTALS SPACEB(2)

A 43'Total records:'

A TOTALRECRD 3 0 57EDTCDE(1)

A 73'Average Age:'

A AVERAGEAGE 3 0 85EDTCDE(1)

A 97'Total at risk:'

A TOTALRISK 3 0 112EDTCDE(1)

A R PROGNAME SPACEB(2)

A PROGRAM 20 3

**STUDENTSQL.SQLRPGLE:**

 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DCL-F STUDENTREP PRINTER OFLIND(\*IN01) ;

DCL-F STUDENTS21 KEYED USAGE(\*INPUT) RENAME(STUDENTS21: STUDENTFILE)

DCL-F STGRADES21 KEYED USAGE(\*INPUT) RENAME(STGRADES21: GRADEFILE)

DCL-F PROGRAM21 KEYED USAGE(\*INPUT) RENAME(PROGRAM21: PROGRAMFILE)

DCL-S EndOfFile IND;

DCL-S TotalRecords PACKED(5:0);

DCL-DS STUDENTRECORD;

StID PACKED(3:0);

PROGRAM CHAR(31);

GPA PACKED(3:2);

RISKCOUNT PACKED(1:0);

END-DS STUDENTRECORD;

EXSR PREPAREFILE;

WRITE REPORTNAME;

WRITE COLHDR;

WRITE PROGNAME;

EXSR GETROW;

WRITE DETAILS;

DOW NOT EndOfFile;

IF \*IN01 = \*ON;

WRITE REPORTNAME;

WRITE COLHDR;

WRITE PROGNAME;

\*IN01 = \*OFF;

ENDIF;

IF GPA < 2;

\*IN50 = \*ON;

RISKCOUNT = RISKCOUNT + 1;

ENDIF;

ENDDO;

EXSR WRAPUP;

WRITE TOTALS;

\*INLR = \*ON;

RETURN;

//SUBROUTINES

BEGSR PREPAREFILE;

EXEC SQL

DECLARE STUDENTCURSOR CURSOR

FOR

SELECT STUDENT#, TRIM(FNAME) || ' ' || LNAME AS FULLNAME, SELECT DATEDIFF(BIRTHDATE)/8766.0 AS AgeYearsDecimal

,CONVERT(int,ROUND(DATEDIFF(hour,@dob,GETDATE())/8766.0,0)) AS AgeYearsIntRound

,DATEDIFF(hour,@dob,GETDATE())/8766 AS Age

FROM BCI433LIB/STUDENTS21

WHERE STUDENT# = :STID

UNION ALL

SELECT SGCOURSE, SGARDE

FROM BCI433LIB/STGRADES21

WHERE SGRID = :STDID

UNION ALL

SELECT PRDESCRP

FROM BCI433LIB/PROGRAM21

WHERE PRSTID = :STDID

FOR READ ONLY;

EXEC SQL

OPEN STUDENTCURSOR;

IF(SQLCODE <> 0) OR (SQLWN0 = 'W');

ENDOFFLIE = \*ON;

ENDIF;

ENDSR;

BEGSR GETROW;

EXEC SQLFETCH NEXT

FROM STUDENTCURSOR

INTO :STUDENTRECORD;

IF(SQLCODE <> 0) OR (SQLWN0 = 'W');

ENDOFFLIE = \*ON;

ENDIF;

ENDSR;

BEGSR WRAPUP;

EXEC SQL

CLOSE STUDENTCURSOR;

IF(SQLCODE <> 0) OR (SQLWN0 = 'W');

ENDOFFLIE = \*ON;

ENDIF;

EXEC SQL

SELECT COUNT(SGRADE) INTO: GPA

FROM STGRADES21

WHERE SGRID = :STDID;

EXEC SQL

SELECT COUNT(\*) INTO: TOTALRECRD

FROM STUDENTCURSOR;

EXEC SQL

SELECT AVG(Age) INTO: AVERAGEAGE

FROM STUDENTCURSOR;

EXEC SQL

SELECT COUNT(RISKCOUNT) INTO: TOTALRISK

FROM STUDENTCURSOR;

ENDSR;