**ST BENEDICT’S COLLEGE**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT** | **Information Technology**  **Practical Exam** | **DATE** | **Nov 2017** |
| **GRADE** | **11** | **MARKS** | **120** |
| **EXAMINER** | **Mrs Kench** | **MODERATOR** | **Mr Blieden and Mrs Nocton-Smith** |
| **NAME** |  | **DURATION** | **3 hours** |
| **USER NAME** |  |  |  |
| **PASSWOD** |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **COGNITIVE LEVELS** | | | | | |
| **LOW ORDER** | 30 % | **MIDDLE ORDER** | 40 % | **HIGH ORDER** | 29 % |

**SECTION A: SQL**

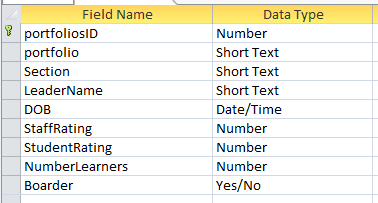
*The files required for this question can be found in the folder named* ***Section A.*** *Please complete your work in this folder****.***

A school has chosen its leaders and has assigned leaders to particular portfolio within the school. Not all leaders have been assigned to a portfolio. At the end of the first term, the leader’s ability to fulfil the requirements of their portfolio have been rated by staff and learners. In each portfolio, the number of learners who are involved in a group are listed. Most portfolios fall under a section such as sport or culture. Some of the leaders are boarders.

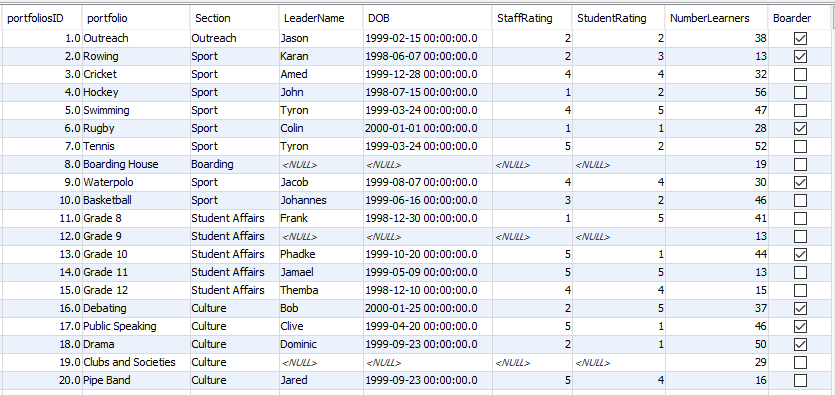
The data is stored in a table called **tblPortfolio** in a database called **LeaderDB**.

**Table design:**

**tblPortfolio**



The data is the table is show below:



**QUESTION 1 40 MARKS**

1.1 Display the portfolios and sections fields sorted alphabetically by section and then by portfolio. (3)

SELECT section, portfolio✓

FROM tblPortfolio

ORDER BY section, ✓ portfolio✓

1.2 List the leaders whose staff rating or student rating is above 3 and who have more than 20 learners in their portfolio. (5)

SELECT \*

FROM tblportfolio

WHERE (staffRating > 3✓ OR✓ StudentRating > 3) ✓ AND✓ NumberLearners > 20✓

1.3 Determine the age of each leader in years. Ensure that the age is accurate, if a person has not had a birthday their age should be one less. For example. Amed has not has his birthday and should be 17 even though he is born in 1999. Name this new field **Age**. Display only leader’s name, date of birth and **Age** (in years). Do not display an age if there is no leader for a portfolio. (7)

SELECT leaderName, DOB, ✓ TIMESTAMPDIFF✓ (YEAR,DOB, ✓Now())✓ AS Age✓

FROM tblportfolio

WHERE leaderName✓ IS NOT null✓

**ACCESS**

SELECT leaderName, DOB, DATEDIFF ('yyyy',DOB, Now()) AS Age

FROM tblportfolio

WHERE leaderName IS NOT null;

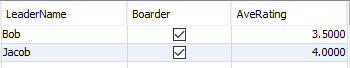
1.4 Count the number of portfolios in each section. (4)

SELECT COUNT(\*) ✓✓AS NumPortfolios

FROM tblportfolio

GROUP BY ✓Section✓

1.5 A new leader for boarding is needs to be considered. This leader will be chosen from the existing leaders. The citeria for the leader of boarding is to have a rating average of 3 or more and to be a boarder themselves. The average rating is calculated using the staff and student rating fields. Code a SQL statement to list all the possible candidates using the criteria. Sort the result according the the average in descending order. Your SQL statement should produce the following result:

 (8)

SELECT LeaderName, Boarder, ✓ (staffRating + studentRating)/2 ✓✓AS AveRating

FROM tblportfolio

WHERE (staffRating + studentRating)/2 > 3✓ AND✓ boarder = true✓

ORDER BY AveRating✓ DESC✓

**ACCESS**

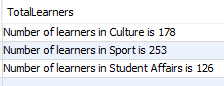
SELECT LeaderName, Boarder, (staffRating + studentRating)/2 AS AveRating

FROM tblportfolio

WHERE (staffRating + studentRating)/2 > 3 AND boarder = true

ORDER BY (staffRating + studentRating)/2 DESC;

1.6 List the sections in the school that have more than 50 learners in total. Display your answer as **Number of learners in *Section* is *xxx***. Your output should appear as follows with the new field called **TotalLearners**:

 (8)

SELECT CONCAT✓ ('Number of learners in ', section✓,' is ',SUM(NumberLearners) ✓) AS TotalLearners✓

FROM tblportfolio

GROUP✓ BY Section✓

HAVING ✓SUM(NumberLearners) > 50✓

**ACCESS**

SELECT "Number of learners in " & section & " is "&SUM (NumberLearners) AS totallearners

FROM tblPortfolio

GROUP BY section

HAVING

SUM (NumberLearners) > 50;

1.7 The sport portfolios are becoming more popular and are going to increase their number of learners by 20%. Code a query to increase the Number of learners in the sport portfolios by 20% rounded to 0 decimal places. (5)

UPDATE tblPortfolio✓

SET NumberLearners ✓= ROUND✓ (1.2\*NumberLearners,0) ✓

WHERE section = 'Sport'✓

**SECTION B OOP 80 MARKS**

**QUESTION 2 34 MARKS**

**//2.1 [4 marks]**

public class Leader {✓

private✓ String name; ✓

private int staffRate, g10Rate, g11Rate, campRate; ✓

**//2.2 [4 marks]**

public Leader✓ (String name, int staffRate, int g10Rate, int g11Rate, int campRate) {✓

this.name = name;

this.staffRate = staffRate;

this.g10Rate = g10Rate;

this.g11Rate = g11Rate;

this.campRate = campRate; ✓✓ for assigning

}

**//2.3 [4 marks]**

✓✓ for all accessors and ✓✓ for all mutators

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getStaffRate() {

return staffRate;

}

public void setStaffRate(int staffRate) {

this.staffRate = staffRate;

}

public int getG10Rate() {

return g10Rate;

}

public void setG10Rate(int g10Rate) {

this.g10Rate = g10Rate;

}

public int getG11Rate() {

return g11Rate;

}

public void setG11Rate(int g11Rate) {

this.g11Rate = g11Rate;

}

public int getCampRate() {

return campRate;

}

public void setCampRate(int campRate) {

this.campRate = campRate;

}

**//2.4 [5 marks]**

public int getPoints() {✓

int temp = 3 \* staffRate + 2 \* g11Rate + g10Rate + 4 \* campRate; ✓✓✓

return temp; ✓

}

**//2.6 [9 marks]**

public void removeVowels() {✓

String temp = "" + name.charAt(0); ✓

final String VOWELS = "aeiouAEIOU";✓

for (int i = 1; i < name.length(); i++) {✓✓

if (VOWELS.indexOf(name.charAt(i)) < 0) {✓✓

temp = temp + name.charAt(i); ✓

}

}

name = temp; ✓

}

**//2.7 [4 marks]**

public boolean lessThan3() {✓

boolean temp = staffRate < 3 | g11Rate < 3 | g10Rate < 3 | campRate < 3; ✓✓

return temp; ✓

}

**//2.5 [4 marks]**

@Override

public String toString() {✓

return✓ name + "\t" + staffRate + "\t" + g10Rate + "\t" + g11Rate + "\t" + campRate + "\t" ✓+ getPoints();✓

}

}

**QUESTION 3 40 MARKS**

**//3.1 [4 marks]**

public class LeaderArray {✓

private Leader leadArr[]✓ = new Leader[50]; ✓

private int size = 0; ✓

**//3.2 [8 marks]**

public LeaderArray() {

try {

Scanner scFile = new Scanner(new FileReader("Leaders.txt"));✓

while (scFile.hasNextLine()) {✓

String line = scFile.nextLine();✓

Scanner tokens = new Scanner(line).useDelimiter(",");✓

String name = tokens.next();

int s = tokens.nextInt();

int g11 = tokens.nextInt();

int g10 = tokens.nextInt();

int c = tokens.nextInt();✓

leadArr[size] = new Leader(name, s, g11, g10, c); ✓

size++;✓

}

scFile.close();

} catch (FileNotFoundException ex) {

System.out.println("File not found");✓

}

}

**//3.3 [5 marks]**

public String toString() {✓

String rString = "";

for (int i = 0; i < size; i++) {✓✓

rString += leadArr[i].toString() + "\n";✓

}

return rString; ✓

}

**//3.4 [9 marks]**

public void sort() {

for (int i = 0; i < size - 1; i++) {✓✓

for (int j = i + 1; j < size; j++) {✓✓

if (leadArr[i].getPoints()✓ <✓ leadArr[j].getPoints())✓

{

Leader temp = leadArr[i];

leadArr[i] = leadArr[j];

leadArr[j] = temp; ✓✓

}

}

}

}

**//3.5 [5 marks]**

public void vowels() {✓

for (int i = 0; i < size; i++) {✓✓

leadArr[i]. ✓removeVowels();✓

}

}

**//3.6 [9 marks]**

public void delete(int pos) {✓

for (int j = pos; j < size - 1; j++) {✓

leadArr[j] = leadArr[j + 1]; ✓

}

size--; ✓

}

public void findLowRatings() {

int i = 0;

while (i < size) {✓

if (leadArr[i].lessThan3() == true) {✓✓

delete(i); ✓

} else {

i++;✓

}

}

}

}

**QUESTION 4 6 MARKS**

**//4.1 [1 mark]**

public class LeaderUI {✓

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

**//4.2 [1 mark]**

LeaderArray la = new LeaderArray();✓

**//4.3 [1 mark]**

System.out.println(la); ✓

**//4.4 [1 mark]**

la.sort();

System.out.println("After Sort\n" + la); ✓

**//4.5 [1 mark]**

la.vowels();✓

System.out.println("After Vowels\n" + la);

**//4.6 [1 mark]**

la.findLowRatings();✓

System.out.println("After Low Ratings\n" + la);

}

}