



Tshwane University  
of Technology

*We empower people*

Student Number

Surname & Initials

PC Number

**YEAR:** 2020

**SEMESTER:** 1

**ASSESSMENT:** FINAL SUMMATIVE ASSESSMENT 1

**SUBJECT NAME:**

TECHNICAL PROGRAMMING II

**SUBJECT CODE:**

TPG201t

**QUALIFICATION(S):**

NDIP SOFTWARE DEVELOPMENT  
NDIP TECHNICAL APPLICATIONS  
NDIP INTELLIGENT INDUSTRIAL SYSTEMS

**PAPER DESCRIPTION:** CLOSED BOOK

**DURATION:** 3 HOURS

**PAPER:** ONLY

#### SPECIAL REQUIREMENTS

☐ **NONE**

☐ **NON-PROGRAMMABLE POCKET CALCULATOR**

☐ **SCIENTIFIC CALCULATOR**

☐ **COMPUTER ANSWER SHEET**

☐ **GRAPH PAPER**

☐ **DRAWING INSTRUMENTS**

**OTHER:**

**INSTRUCTIONS TO CANDIDATES:** ANSWER ALL QUESTIONS

Answer questions on EC.  
Ensure that you copy and paste your code in time.  
**NO USB drives are allowed**

**TOTAL NUMBER OF PAGES INCLUDING COVER PAGE:** 30

**TOTAL NUMBER OF ANNEXURES:** 0

**EXAMINER:** A.A.K. Buitendag

**FULL MARKS:** 90

**MODERATOR:** H. Jeske

**TOTAL MARKS:** 91

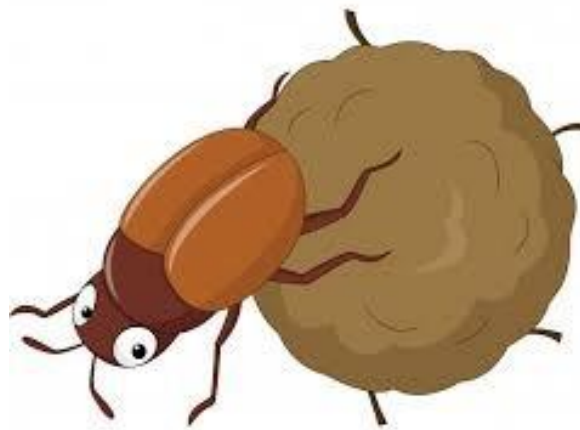
**STUDENT TOTAL:** \_\_\_\_

**STUDENT %:** \_\_\_\_

**TECHNICAL PROGRAMMING II**

TPG201T

**FINAL SUMMATIVE ASSESSMENT 1**



**S1 2020**

**This question paper consists of 30 pages.**

<p><b>TIME: 3 HOURS</b> <b>MARKS: 91</b></p>
--

---

## **INSTRUCTIONS AND INFORMATION**

1. Answer ALL the questions.
  2. Read ALL the questions carefully.
  3. This is a CLOSED book practical assessment. You may not be in possession of any memory device (e.g. USB memory stick, memory card, cell phone or any soft copy code) or any textbooks or notes. Should it be found in your possession, a disciplinary hearing will be opened against you.
  4. You are allowed to reference the Embarcadero RAD studio help files only. You must save your work regularly. Use the SAVE ALL option. No marks will be assigned if your work is lost due to incorrect saving methods.
  5. No extra time will be given if work is not saved and there is a power failure or any other kind of problem.
  6. Save your work on the D:\MyDocuments drive-folder of the computer that you are working on.
  7. Insert the following comments in your program: Student Surname and Initials, Student Number, PC Number.
  8. You are not allowed to access any other electronic resources over the network or access any internet related resource.
-

## SCENARIO

You are going to develop an application that are focussed on some operations regarding the Kruger National Park.

All applicable data files are stored within the subfolder:

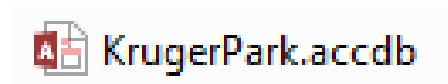


TAKE NOTE: Open this folder and study the given files.

## DB Description

The applicable database for the question has been provided in the folder AppData which is a subfolder within your project folder.

The applicable DB (in Access) is named: **(KrugerPark.accdb)**



Study the following database schemas and excerpts of the data from the various applicable tables. Not all records are shown

### The Camps table

Camps	
Field Name	Data Type
CampCode	Short Text
CampDescription	Short Text
CampType	Short Text

Some sample data from the table

Camps		
CampCode	CampDescription	CampType
KPBAL	Balule Satellite Camp	SL
KPBAT	Bateleur Bushveld Camp	BC
KPBBL	Boulders Bush Lodge	BL
KPBED	Berg-en-Dal Rest Camp	MC
KPBIY	Biyamiti Bushveld Camp	BC
KPCRB	Crocodile Bridge Rest Camp	MC
KPHHT	Hoyo Hoyo Tsonga lodge (Luxury Lodge)	BL
KPIMS	Imbali safari lodge (Luxury Lodge)	BL
KPLET	Letaba Rest Camp	MC
KPLOE	Loerie Lodge	BC

Notes on some fields:

- The CampType field is a FK field to the CampTypes table, relating to the CTypeCode field.

### ***The CampTypes Table***

CampTypes	
Field Name	Data Type
CTypeCode	Short Text
CampType	Short Text

Some sample data from the table

CampTypes	
CTypeCode	CampType
BC	Bushveld Camp
BL	Bush Lodge
CS	Camp Site
MC	Main camp
SL	Satellite camp

Notes on the table

- The CTypeCode is the PK field

### ***The AccommodationTypes table***

AccommodationTypes	
Field Name	Data Type
AccomCode	Short Text
AccomDesc	Short Text

Some sample data from the table

AccommodationTypes	
AccomCode	AccomDesc
CK6PZ	Camp Site
CO4	Cottage
CO4D	Cottage
CTT2	Safari Tent
CTT2Z	Safari Tent
CTT4	Safari Tent
CTT4U	Safari Tent
CTT4UZ	Safari Tent
CTT4Z	Safari Tent
EA3U	Hut
EB2	Hut
EB3	Hut
EB5	Hut
EB6	Hut
EH1	Hut
EH2	Hut
EH3	Hut
EH4	Hut
ES2	Hut
FA6	Family Cottage
FA6C	Family Cottage

This table comprise of an accommodation type code and a description of the applicable accommodation type.

### The Clients table

Field Name	Data Type
ClientNr	Short Text
Surname	Short Text
Initials	Short Text
SAIDNum	Short Text
JoinDate	Date/Time
AddressLine1	Short Text
AddressLine2	Short Text
CityTown	Short Text
PostalCode	Short Text
LoyaltyPoints	Number
WildCardNr	Short Text

This table contain data from the clients (i.e. visitors) that has visited the park in the past.

Some sample data from the table

ClientNr	Surname	Initials	SAIDNum	JoinDate	AddressLine1	AddressLine2	CityTown	PostalCode	LoyaltyPoints	WildCardNr
1000008	MPHANAMA	AG	5710265187087	2011/04/16	PO BOX 3227		THOHOYANDO	950	146	W0000431
1000016	MASIPA	GN	5901045191081	2009/02/20	PO BOX 1116		RAMOKGOPA	811	532	W0003203
1000027	BERNS	EC	8112070061085	2010/11/01	PO BOX 4939		EMPANGENI	3880	299	W0002487
1000039	RAMOSHABA	ME	6910265413082	2011/04/01	PO BOX 2143		LENYENYE	857	947	
1000041	OOSTHUIZEN	JH	6204115215008	2010/12/28	PRIVATE BAG X3		INDUSTRIELE G	6230	484	W0004123
1000049	MNGOMEZULU	SA	6006115732084	2009/06/05	159 MURRAY RD		HAYFIELDS	3201	215	
1000061	MOSERY	ENB	6001170680088	2009/06/16	PO BOX 2212		DURBAN	4000	73	W0002633
1000068	MASEKWAMEN	TI	5505225760084	2009/03/29	PO BOX 1522		HAMMANSKR	400	1554	
1000072	KGANAKGA	MHL	6411025289086	2010/09/07	186 LEOPARD ROCK HENDRIN		RIDGEWAY	2091	255	
1000081	GREYVENSTEIN	M	6405090011086	2011/03/10	VILLA ZENOBIA 30	VEDA AVENUE	PRETORIA	1	1779	
1000082	HOOSEN	R	7601285258081	2011/02/05	BLOCK 17 / FLAT 10	259 HIMALAYA	LAUDIUM	37	592	
1000092	SEEPE	RC	6808180552085	2010/09/05	PO BOX 49		SEKHUKHUNE	1124	86	
1000105	MUDAU	MC	5808085194083	2009/08/30	PRIVATE BAG X2004		OBBERHOLZER	2502	297	
1000119	DANISORUS	MT	7502275402087	2010/12/30	PRIVATE BAG X06		VEREENIGING	1930	529	
1000121	MAMBURU	TJ	7006155271083	2010/04/24	PRIVATE BAG X2	ODENDALRUS	ODENDAALSR	9480	204	
1000144	MBOYI	TV	5806270999084	2009/08/29	PO BOX 503		THOHOYANDO	950	523	
1000165	BOTHA	A	7104230050083	2010/03/15	PRIVATE BAG X6106		KIMBERLEY	8300	1988	W0000267

Notes on the table

- The ClientNr is the PK field
- The WildCardNr is a FK field to the ClientWildCards table (Note : Not all clients have wildcards)

### The ClientWildCards table

Field Name	Data Type
WildCardNr	Short Text
LastReviewalDate	Date/Time
WildCardType	Short Text
AmountPaid	Currency
MaxPax	Number

This table relate to the clients that have wildcards which is a type of LoyaltyCard and that promote conservation of animals

Some sample data from the table

WildCardNr	LastReviewalDat	WildCardType	AmountPaic	MaxPax
W0000001	2014/03/01	AllPark	R 1 140.00	6
W0000002	2017/12/08	CapeNatu	R 1 140.00	1
W0000003	2014/09/17	MsiCl	R 960.00	6
W0000004	2010/08/30	MsiCl	R 960.00	7
W0000005	2013/07/14	MsiCl	R 960.00	1
W0000006	2017/02/15	AllPark	R 1 140.00	5
W0000007	2012/12/26	CapeNatu	R 1 140.00	2
W0000008	2016/03/31	Internat	R 960.00	4
W0000009	2015/07/27	SwaziLan	R 810.00	5
W0000010	2010/01/01	EKZNWC	R 1 050.00	4
W0000011	2015/01/20	SANPark	R 960.00	5
W0000012	2017/07/20	EKZNWC	R 1 050.00	5
W0000013	2010/10/18	SANPark	R 960.00	6

Notes on the table

- The WildCardNr is the PK field
- A WildCard is valid only for a year and must then be renewed
- The WildCardType is a FK field to the WildCard table

### The WildCard table

Field Name	Data Type
Code	Short Text
Fullname	Short Text
SingleRate	Currency
CoupleRate	Currency
FamilyRate	Currency
Description	Short Text

Some sample data from the table

Code	Fullname	SingleRate	CoupleRate	FamilyRate	Description
AllPark	All Parks Cluster	R 565.00	R 930.00	R 1 140.00	Access to 80+ Parks and Reserves around Southern Africa, which are included in the SANParks
CapeNatu	CapeNature cluster	R 490.00	R 800.00	R 960.00	Access to 24 of Cape Nature's Parks and Reserves in the Western Cape
EKZNWC	EKZN Wildlife Cluster	R 540.00	R 875.00	R 1 050.00	Access to 25 of KZN Wildlife's Parks and Reserves in KwaZulu-Natal
Internat	International All Parks Cluste	R 2 430.00	R 3 800.00	R 4 545.00	Access to 80+ Parks and Reserves around Southern Africa, which are included
MsiCl	Msimsi Cluster	R 490.00	R 805.00	R 955.00	Access to all 6 of Msimsi's Resorts and Reserves near Durban and Pietermaritzburg
SANPark	SANParks Cluster	R 540.00	R 880.00	R 1 055.00	Access to all 21 of SANParks National Parks in South Africa
SwaziLan	Swaziland's Big Game Parks Clu	R 405.00	R 665.00	R 810.00	Access to Big Game Parks of Swaziland's 3 Parks in Swaziland

Notes on the table

- The Code field is the PK field of the table and the Fullname is the name of the type of wildcard
- A WildCard is valid only for a year and must then be renewed
- The WildCardType is a FK field to the WildCard table

**The Rangers table<sup>1</sup>**

Field Name	Data Type	Description (Optional)
RangerID	Number	A unique ID for each Ranger
Name	Short Text	The name of the ranger
Surname	Short Text	The surname of the ranger
Rank	Short Text	The rank of the ranger
DateAppointed	Date/Time	The date the ranger was appointed
ReportCamp	Short Text	The main camp that the ranger reside

This table contain data relating to the different rangers in the park. Each ranger is assigned to a base camp.

Some sample data from the table

RangerID	Name	Surname	Rank	DateAppoin	ReportCamp
1	Jada	Harrison	Park	2002/10/18	KPLET
2	Kenyon	Carney	Park	2007/04/30	KPLET
3	Dylan	Pollard	Field	2005/05/13	KPOLI
4	Sylvester	Walls	Senior	2003/11/01	KPORP
5	Urielle	Wynn	Park	2007/12/01	KPSKU
6	Giselle	Head	Field	2003/06/27	KPSKU
7	Amos	Roach	Recruit	2005/06/26	KPPRE
8	Tobias	Paul	Park	2004/09/19	KPPRE
9	Odessa	Head	Section	2003/06/07	KPLSR
10	Charles	Buckner	Section	2008/08/07	KPLSR
11	Ariel	Hooper	Recruit	2017/01/01	KPBED
12	Hammett	Gates	Field	2005/07/02	KPBED

Notes on the table

- The RangerID is the PK field
- The ReportCamp field is a FK field to the Camps Table

**The Sightings table**

Field Name	Data Type	Description (Optional)
SightingID	Number	A unique ID given to each sighting
SightingDate	Date/Time	The date of the sighting
Animal	Short Text	The animal that was sighted
NumAnimals	Number	The number of animals that were sighted
Young	Yes/No	If there were any young sighted
RangerID	Number	The ID of the ranger that sighted the animals

Some sample data from the table

<sup>1</sup> The Sightings and the adapted Rangers table is taken from the DBE NSC IT Paper 1 exam of 2009. – Recognition Department of Basic Education



SightingID	SightingDate	Animal	NumAnimal	Young	RangerID
101	2018/05/01	Giraffe	2	True	3
102	2018/04/03	Impala	23	False	5
103	2018/01/13	Lion	13	False	7
105	2018/05/31	Aardvark	1	True	8
106	2018/04/01	Elephant	1	False	3
107	2018/06/19	Lion	23	False	13
108	2018/04/19	Elephant	32	False	1
109	2018/05/06	Rhino	13	False	2
110	2018/06/21	Rhino	13	False	3
111	2018/05/27	Elephant	10	False	15
112	2018/03/23	Aardvark	7	True	19
113	2018/03/24	Giraffe	14	True	20
114	2018/02/10	Aardvark	25	True	14
115	2018/03/30	Cheetah	14	False	18

Notes on the table

- The SightingsID is the PK field (it is incremental)
- The RangerID field is a FK field to the Rangers table

### The TarrifRates table

Field Name	Data Type
AccommodationType	Short Text
CampCode	Short Text
BaseRate	Currency
NrUnits	Number
BaseRatePAX	Number
MaxPAX	Number
AddAdult	Currency
AddChild	Currency

This table contain the data of the different rates per night for the different accommodation types at the various camps.

AccommodationType	CampCode	BaseRate	NrUnits	BaseRatePAX	MaxPAX	AddAdult	AddChild	Click to Add
BA3	KPBED	R 1 150.00	15	2	3	R 260.00	R 130.00	
BA3U	KPBED	R 1 235.00	12	2	3	R 260.00	R 130.00	
BA3UZ	KPBED	R 1 235.00	6	2	3	R 260.00	R 130.00	
BBD2V	KPOLI	R 1 650.00	12	2	2	R 260.00	R 130.00	
BD2	KPLET	R 1 120.00	13	2	2	R 260.00	R 130.00	
BD2	KPLSR	R 1 400.00	4	2	2	R 260.00	R 130.00	
BD2	KPOLI	R 1 120.00	10	2	2	R 260.00	R 130.00	
BD2	KPORP	R 1 260.00	10	2	2	R 260.00	R 130.00	
BD2	KPPRE	R 1 200.00	22	2	2	R 260.00	R 130.00	
BD2	KPSAT	R 1 455.00	24	2	2	R 260.00	R 130.00	
BD2	KPSKU	R 1 400.00	10	2	2	R 260.00	R 130.00	
BD2	KPSRC	R 1 070.00	12	2	2	R 260.00	R 130.00	
BD2D	KPPRE	R 1 200.00	10	2	2	R 260.00	R 130.00	
BD2D	KPSRC	R 1 100.00	10	2	2	R 260.00	R 130.00	
BD2E	KPSKU	R 1 400.00	4	2	2	R 260.00	R 130.00	
BD2EC	KPSAT	R 1 455.00	4	2	2	R 260.00	R 130.00	
BD2M	KPPRE	R 1 250.00	12	2	2	R 260.00	R 130.00	
BD2MZ	KPPRE	R 1 250.00	1	2	2	R 260.00	R 130.00	
BD2N	KPSAT	R 1 455.00	16	2	2	R 260.00	R 130.00	

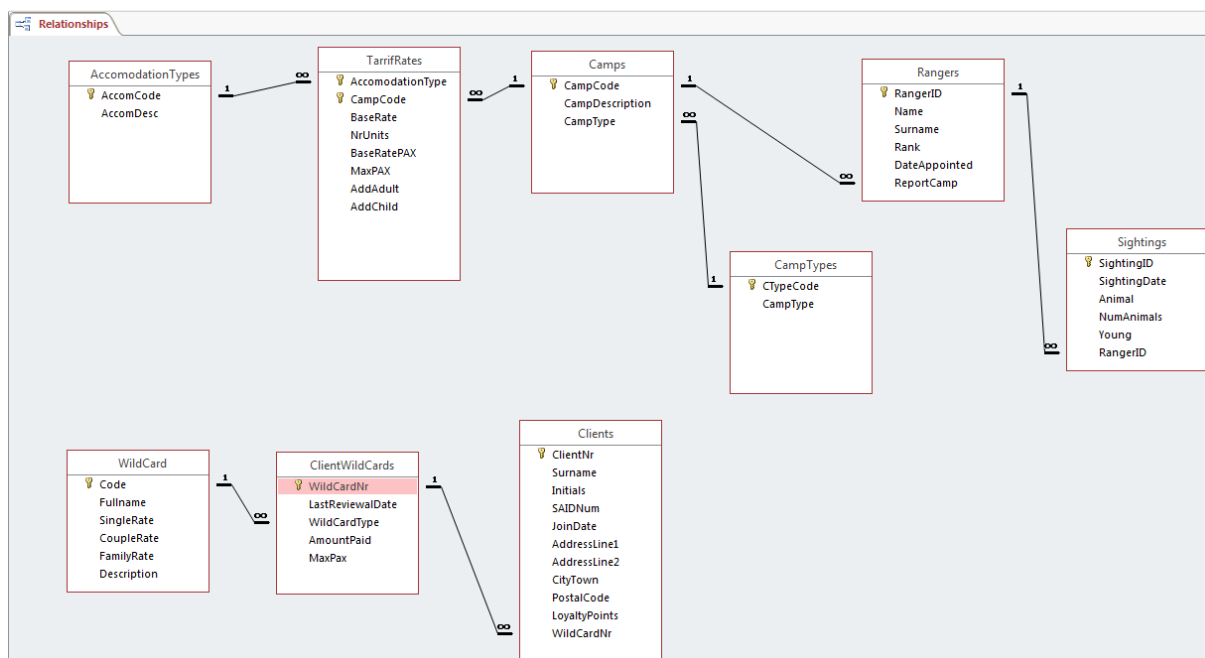
## Notes on the table

- The table has a composite primary key comprising of the AccommodationType and CampCode fields
- The baserate the rate per night for one of the types of accommodation, the BaseRatePAX is the number of people included in the base rate.
- The MaxPax field indicates the maximum number of persons that may be accommodated in the accommodation

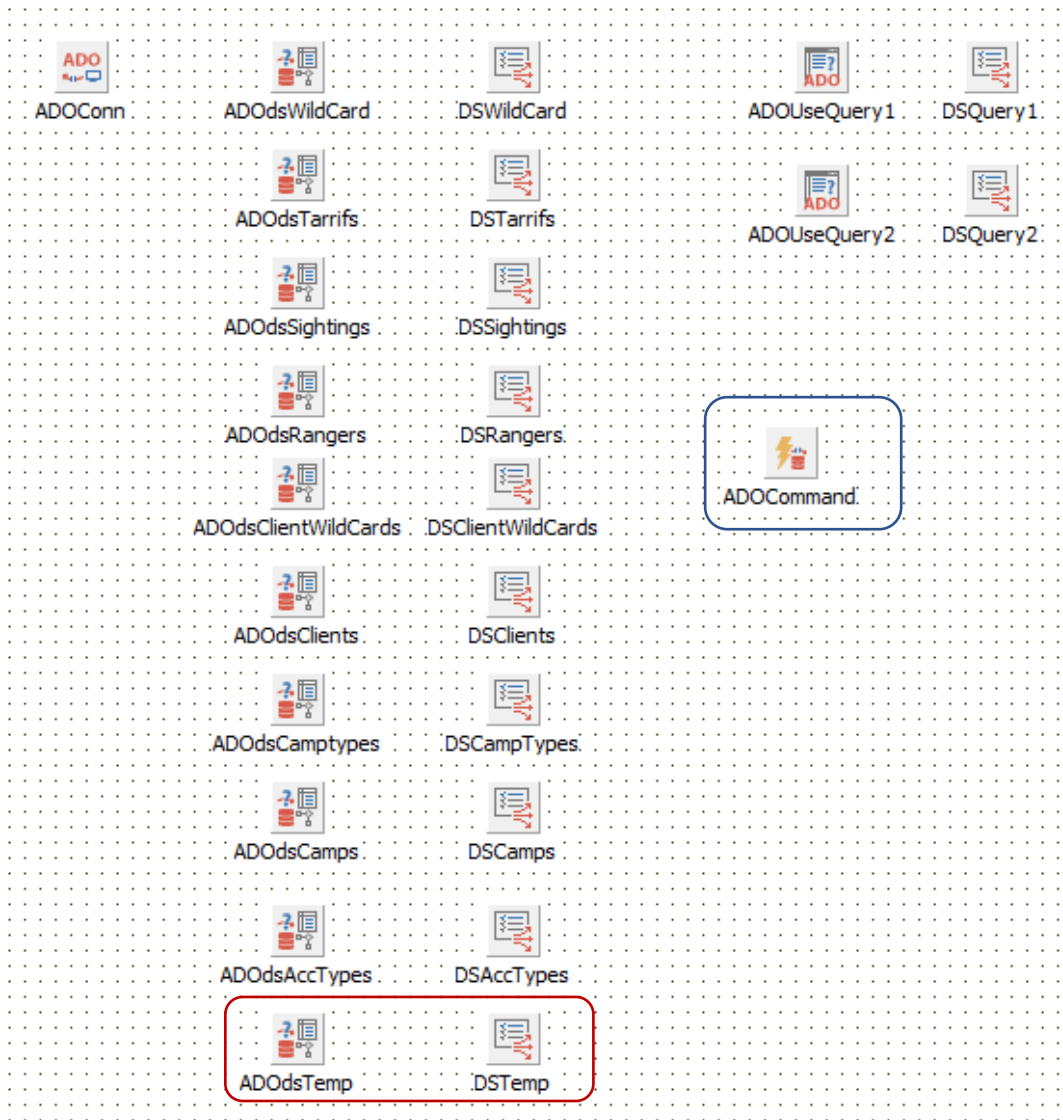
If we evaluate the first record of the table we could state the following;

At the Berg-en-Dal Rest Camp (KPBED) there are 15 BA3 (which is a Bungalow) units available. Each such unit costs R1150 per night for 2 persons (BaseRatePAX) and additional one person could be added which is either a child or adult for a R130 or R260 respectively

**The relationship diagram is presented below:**



For the application that you are going to develop the following DataModule has been made and included as part of the Project.



***You will have to set the Connection Settings using the object inspector to point to the applicable file KrugerPark.accdb.***

The ADOdsTemp dataset is used for temporary processing of queries and records.

The ADOUseQuery1 and ADOUseQuery2 query components are used to execute queries for retrieval and data modification purposes

The ADOCommand component is included for use with DML related queries.

Please familiarize yourself with the command texts for each of the other DataSet components.

## The .h file

`frmExamSectionAUnit.h`

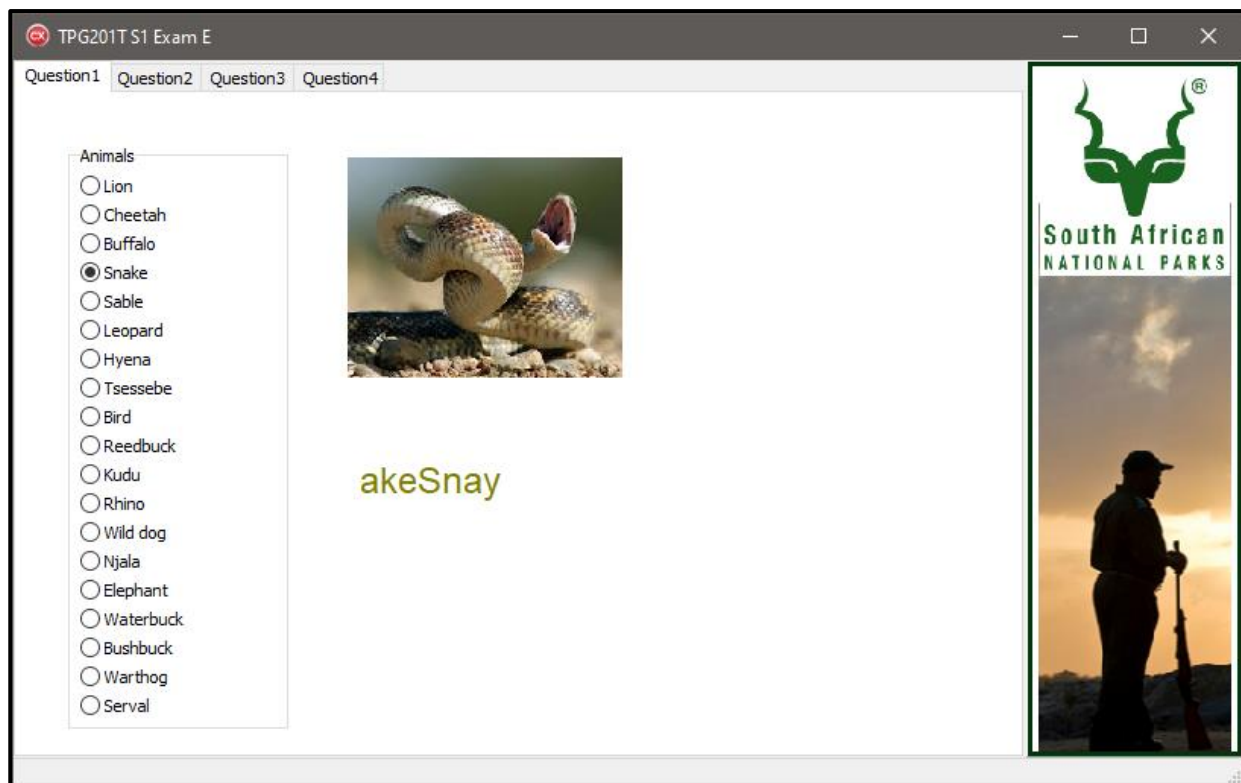
It is imperative that you study the given field declarations (and or other method declarations) as part of the form class.

YOU must use these as part of your application code.

```
private:    // User declarations
    AnsiString RangerName , RangerSurname;
    int rangerID;
    bool isVowel(char Let);
    AnsiString OneListLine;
    TStringList *lstRawData;
```

## SECTION B Questions 1 & 2

### QUESTION 1 – tbsQuestion1



- 1.1 Write the code for the on Show event of the Question 1 tabsheet that will change the appearance of the lblPigLatinAnimal label as follows: (2)
- change the font to appear as 18pt
  - the colour to be clOlive and;
  - the displayfont as Comic Sans

- 1.2 Complete the code for the isVowel form class method that will return a true if the passed character is a vowel (i.e. A,E,I, O, U) and a false if not. (5)
- 1.3 Complete the code for the onClick event of the rdgAnimals radiogroup that will load and display the corresponding file image from the AppData on the imgAnimal component. (12)

Also write applicable code as part of the event handler to determine and display the Piglatin name of the animal on the lblPigLatinAnimal label.

In order to create the corresponding Piglatin animal name the following rules apply:

- If the animal name starts with a consonant and a vowel, put the first letter of the word at the end of the word and add "ay."

Example: Kudu = uduk + ay = uduKay

- If a animal name starts with two consonants move the two consonants to the end of the word and add "ay."


Example: Cheetha = eethaCh + ay = eethaChay

- If the animal name starts with a vowel add the word "way" at the end of the word.

Example: Elephant = Elephant +way = Elephantway

Animals


- ☐ Lion
- ☒ Cheetah
- ☐ Buffalo
- ☐ Snake
- ☐ Sable
- ☐ Leopard
- ☐ Hyena
- ☐ Tsessebe
- ☐ Bird
- ☐ Reedbuck
- ☐ Kudu
- ☐ Rhino



eetahChay

Animals

- ☐ Lion
- ☐ Cheetah
- ☐ Buffalo
- ☐ Snake
- ☐ Sable
- ☐ Leopard
- ☐ Hyena
- ☐ Tsessebe
- ☐ Bird
- ☐ Reedbuck
- ☐ Kudu
- ☐ Rhino
- ☐ Wild dog
- ☐ Njala
- ☒ Elephant



Elephantway

(19)

## QUESTION 2 – tbsQuestion2

TPG201T S1 Exam E

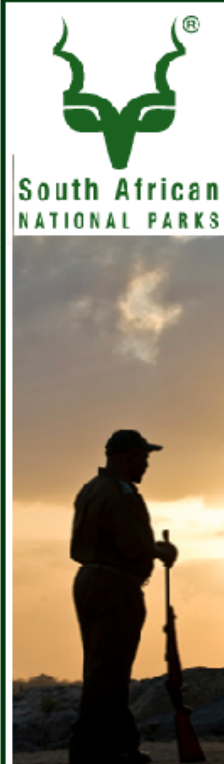
Question1 Question2 Question3 Question4

Load Data

Lion ☐ Filter on animal keyword

101	2018/05/01	Giraffe	2	True	3
102	2018/04/03	Impala	23	False	5
103	2018/01/13	Lion	13	False	7
105	2018/05/31	Aardvark	1	True	8
106	2018/04/01	Elephant	1	False	3
107	2018/06/19	Lion	23	False	13
108	2018/04/19	Elephant	32	False	1
109	2018/05/06	Rhino	13	False	2
110	2018/06/21	Rhino	13	False	3
111	2018/05/27	Elephant	10	False	15
112	2018/03/23	Aardvark	7	True	19
113	2018/03/24	Giraffe	14	True	20
114	2018/02/10	Aardvark	25	True	14
115	2018/03/30	Cheetah	14	False	18
116	2018/06/30	Giraffe	7	False	20
117	2018/05/29	Elephant	17	False	14
118	2018/01/17	Impala	8	False	18
119	2018/06/12	Lion	8	True	7
120	2018/01/20	Giraffe	15	False	3
121	2018/03/08	Impala	8	True	20
122	2018/01/30	Kudu	34	False	11

Records processed - 99



The code for the on click event of the [Load Data] button that will load the textfile for display and processing purposes in the redOutput richeditbox has been provided.



SightingsListTab  
Del.txt

```
101 2018/05/01  Giraffe 2    True   3
102 2018/04/03  Impala 23   False  5
103 2018/01/13  Lion   13   False  7
105 2018/05/31  Aardvark 1    True   8
106 2018/04/01  Elephant 1    False  3
107 2018/06/19  Lion   23   False  13
108 2018/04/19  Elephant 32   False  1
109 2018/05/06  Rhino  13   False  2
110 2018/06/21  Rhino  13   False  3
111 2018/05/27  Elephant 10   False  15
112 2018/03/23  Aardvark 7    True   19
```

The textfile contains the data of the various sighting records (with the same fields as part of the) corresponding database table. The fields of the textfile is tab delimited.

Load Data

Lion ☐ Filter on animal keyword

101	2018/05/01	Giraffe	2	True	3
102	2018/04/03	Impala	23	False	5
103	2018/01/13	Lion	13	False	7
105	2018/05/31	Aardvark	1	True	8
106	2018/04/01	Elephant	1	False	3
107	2018/06/19	Lion	23	False	13
108	2018/04/19	Elephant	32	False	1
109	2018/05/06	Rhino	13	False	2
110	2018/06/21	Rhino	13	False	3
111	2018/05/27	Elephant	10	False	15
112	2018/03/23	Aardvark	7	True	19
113	2018/03/24	Giraffe	14	True	20
114	2018/02/10	Aardvark	25	True	14
115	2018/03/30	Cheetah	14	False	18
116	2018/06/30	Giraffe	7	False	20
117	2018/05/29	Elephant	17	False	14
118	2018/01/17	Impala	8	False	18
119	2018/06/12	Lion	8	True	7
120	2018/01/20	Giraffe	15	False	3
121	2018/03/08	Impala	8	True	20
122	2018/01/30	Kudu	34	False	11

Records processed - 99

A label named `lblRecords` should indicate the number of records loaded in the richeditbox.

The code for the function `getFieldByNr` has been provided

```
AnsiString getFieldByNr(AnsiString aLine, int fieldNr, char delimiter)
```

This function splits `aLine` into parts and return the `fieldNr`'th field when the delimiter is used.

- 2.1 **Write the code for the onClick event of the `ckbAnimalKeyword` checkbox** that will when checked filter the data for display purposes in the `redOutput` Richeditbox according to the keyword that was selected in the combobox.

In order to do so, first assign the content of the richeditbox (i.e. which contains the data from the textfile to the `lstRawData` list. The use the `lstRawData` list for further processing

Load Data

Lion ☒ Filter on animal keyword

103	2018/01/13	Lion	13	False	7
107	2018/06/19	Lion	23	False	13
119	2018/06/12	Lion	8	True	7
126	2018/04/05	Lion	24	True	7
127	2018/03/06	Lion	20	False	10
132	2018/02/19	Lion	2	True	4
136	2018/03/26	Lion	19	True	11
139	2018/03/16	Lion	2	True	12
164	2018/05/06	Lion	34	False	12
174	2018/07/09	Lion	9	True	16
176	2018/03/13	Lion	10	False	16
186	2018/07/01	Lion	12	True	11
187	2018/01/28	Lion	26	False	8

Records processed - 14 Total number of Lion = 202

In the example screenshot the user selected the keyword `Lion` in the combobox. All the “records” of the sightings where Lions were seen are displayed.

In the example below the user selected Elephant as a keyword and all records where the 3<sup>rd</sup> field is the word Elephant are listed.

Take note: The current number of filtered records are also displayed as well as the total of the sightings of the particular animal as indicated in the 4<sup>th</sup> column

Load Data				
Elephant <input type="checkbox"/> Filter on animal keyword				
106	2018/04/01	Elephant 1	False	3
108	2018/04/19	Elephant 32	False	1
111	2018/05/27	Elephant 10	False	15
117	2018/05/29	Elephant 17	False	14
133	2018/01/25	Elephant 22	False	15
154	2018/01/26	Elephant 34	True	8
155	2018/01/06	Elephant 19	False	6
159	2018/02/08	Elephant 23	False	3
166	2018/07/04	Elephant 16	False	1
168	2018/03/18	Elephant 18	False	8
189	2018/01/04	Elephant 3	False	1
190	2018/06/06	Elephant 27	True	9
195	2018/01/03	Elephant 14	False	4
106	2018/04/01	Elephant 1	False	3
108	2018/04/19	Elephant 32	False	1
111	2018/05/27	Elephant 10	False	15
117	2018/05/29	Elephant 17	False	14
133	2018/01/25	Elephant 22	False	15
154	2018/01/26	Elephant 34	True	8
155	2018/01/06	Elephant 19	False	6
159	2018/02/08	Elephant 23	False	3

Records processed - 92 Total number of Elephant = 1652

**TOTAL Section B**

(17)  
[36]



**SECTION C – Questions 3 & 4****QUESTION 3 – tbsQuestion3**

TPG201T S1 Exam E

Question1 Question2 Question3 Question4

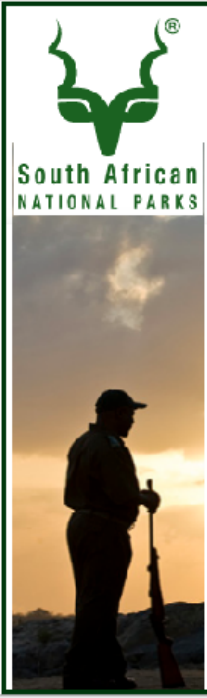
## Rangers

RangerID 1  
Name Jada Surname Harrison

Navigation icons: Previous, Next, Add, Remove, Up, Down, Confirm, Cancel, Refresh

## Sightings

SightingID	SightingDate	Animal	NumAnimals	Young	RangerID
235	2020/08/14	Lion	2	True	1
236	2020/08/17	Sable	2	True	1
108	2018/04/19	Elephant	32	False	1
124	2018/07/17	Cheetah	22	True	1
130	2018/05/05	Impala	29	False	1
131	2018/01/11	Kudu	6	False	1
141	2018/02/26	Aardvark	11	True	1
166	2018/07/04	Elephant	16	False	1
172	2018/01/26	Rhino	7	True	1



This question pertains to the use of the database and the applicable datamodule and other data access components.

Complete the code for the onShow event of the tabsheet that will set up and accomplish the following by writing code to:

- 3.1 Set up the properties of the various DBText controls to link and display the corresponding fields as part of the Rangers table. (3)

**Rangers**

RangerID 1  
Name Jada Surname Harrison

- 3.2 Dynamically instantiate a dbNavigator that will allow the user to browse the various records of the Rangers dataset (5)

**Rangers**

RangerID 1  
Name Jada Surname Harrison

Navigation icons: Previous, Next, Add, Remove, Up, Down, Confirm, Cancel, Refresh

Create a master detail relationship using the dsTemp datasource and dataset with an applicable query that will display the corresponding sightings records for the particular master ranger record in the DBGrid.

(7)

### Rangers

**RangerID** 4  
**Name** Sylvester **Surname** Walls

Navigation buttons: [Previous], [Previous], [Next], [Next], [Add], [Delete], [Refresh], [Save], [Cancel], [Reset]

### Sightings

SightingID	SightingDate	Animal	NumAnimals	Young	RangerID
123	2018/02/26	Cheetah	25	True	4
132	2018/02/19	Lion	2	True	4
158	2018/05/04	Giraffe	24	False	4
163	2018/06/14	Impala	34	True	4
191	2018/06/11	Impala	22	True	4
195	2018/01/03	Elephant	14	False	4
197	2018/07/27	Cheetah	2	True	4
198	2018/04/02	Impala	21	True	4

In the sample screenshot the current record is for Ranger 4 and all the corresponding sightings registered by the ranger is shown in the DBGrid.

(15)

#### QUESTION 4 – tbsQuestion4

TPG201T S1 Exam E

Question1 Question2 Question3 Question4

### Update Sightings

SightingID	SightingDate	Animal	NumAni
211	2020/02/24	Lion	
212	2020/02/24	Lion	
213	2020/02/24	Giraffe	
214	2020/02/24	Elephant	
215	2020/02/24	Elephant	
216	2020/02/24	Lion	
217	2020/02/24	Cheetah	
218	2020/02/24	Buffalo	
219	2020/02/24	Snake	
220	2020/02/24	Sable	
221	2020/02/24	Leopard	
222	2020/02/24	Hyena	
223	2020/02/24	Tsessebe	
224	2020/02/24	Bird	

Create New SightingId

Sighting ID **lblSightingID**

Animal

Number of animals

Young

Ranger id

Save Sighting

This question pertains to the use of the database and the applicable datamodule and other data access components.

This tabsheet will allow a user to browse the various sightings and add a new sighting record.

- 4.1 Complete the code for the on click event of the [Create New SightingID] button that will create a new sighting ID based on the highest value of the SightingID field in the table and incrementing the value with one. Implement an SQL query to determine the highest value and display the new sighting ID on the lblSightingID label. (5)

### Update Sightings

SightingID	SightingDate	Animal	NumAni
211	2020/02/24	Lion	
212	2020/02/24	Lion	
213	2020/02/24	Giraffe	
214	2020/02/24	Elephant	
215	2020/02/24	Elephant	
216	2020/02/24	Lion	
217	2020/02/24	Cheetah	
218	2020/02/24	Buffalo	
219	2020/02/24	Snake	

Sighting ID 237

Animal Cheetah

Number of animals 0

In the sample screenshot above the last (i.e. highest SightingID was **236**) thus the new sighting ID would be **237**.

- 4.2 Complete the code for the on click event of the [Save Sighting] button that will add a new record based on the sighting data entered by the user using the various controls on the right hand side of the tab sheet. (15)

Your event handler must **implement a parameterised query, and the ADOCommand** component. Take note of each of the controls and in particular the way in which the data of the ranger are presented in the combobox. The sighting date is set to the current date.

The ranger code is followed by the surname of the ranger.

When the record has been added an appropriate message must be displayed to the user indicating that the record has been saved.

## Update Sightings

SightingID	SightingDate	Animal	NumAni
211	2020/02/24	Lion	
212	2020/02/24	Lion	
213	2020/02/24	Giraffe	
214	2020/02/24	Elephant	
215	2020/02/24	Elephant	
216	2020/02/24	Lion	
217	2020/02/24	Cheetah	
218	2020/02/24	Buffalo	
219	2020/02/24	Snake	
220	2020/02/24	Sable	
221	2020/02/24	Leopard	
222	2020/02/24	Hyena	
223	2020/02/24	Tsessebe	
224	2020/02/24	Bird	

Create New SightingId

Sighting ID **237**

Animal

Number of animals

Young

Ranger id

Save Sighting

Projectexamseca

Sighting recorded

OK

## Update Sightings

SightingID	SightingDate	Animal	NumAnimals	Young	f
228	2020/02/24	Wild dog	8	False	
229	2020/02/24	Njala	9	True	
230	2020/02/24	Elephant	2	False	
231	2020/02/24	Waterbuck	6	True	
232	2020/02/24	Bushbuck	9	False	
233	2020/02/24	Warthog	8	False	
234	2020/02/24	Serval	2	True	
235	2020/08/14	Lion	2	True	
236	2020/08/17	Sable	2	True	
237	2020/08/17	Hyena	3	True	
101	2018/05/01	Giraffe	2	True	
102	2018/04/03	Impala	23	False	
103	2018/01/13	Lion	13	False	
105	2018/05/31	Aardvark	1	True	

Create New SightingId

Sighting ID **237**

Animal

Number of animals

Young

Ranger id

Save Sighting

TOTAL SECTION C

(20)  
[35]MAXIMUM MARKS  
PAPER TOTAL[90]  
[91]

**Marking Guideline Section B**

```
//-----
#include <vcl.h>
#include "jpeg.hpp"
#include "DMUnit.h"
#include "DateUtils.hpp"
#pragma hdrstop

#include "frmExamSectionAUnit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TfrmExamFinSumAsses1 *frmExamFinSumAsses1;
//-----
__fastcall TfrmExamFinSumAsses1::TfrmExamFinSumAsses1(TComponent* Owner)
    : TForm(Owner)
{
}

//-----
void __fastcall TfrmExamFinSumAsses1::btnLoadClick(TObject *Sender)
{
    // GIVEN Code
    redOutput->Lines->LoadFromFile("../\\..\\AppData\\SightingsListTabDel.txt");
    lblRecords->Caption = " Records processed - " + IntToStr(redOutput->Lines->Count);
}

//-----
// GetFieldByNr - GIVEN
AnsiString getFieldByNr(AnsiString aLine, int fieldNr, char delimiter)
{
    aLine += delimiter;
    int pos;
    AnsiString field;
    for (int i = 1; i <= fieldNr; i++) {
        pos = aLine.Pos(delimiter);
        field = aLine.SubString(1, pos - 1);
        aLine.Delete(1, pos);
    }

    return field;
}

//-----

void __fastcall TfrmExamFinSumAsses1::tbsQuestion1Show(TObject *Sender)
{
    // Question 1.1
    // (2) For every inncorrect setting -1

    lblPigLatinAnimal->Font->Size = 18;
    lblPigLatinAnimal->Font->Name = "Comic Sans";
    lblPigLatinAnimal->Font->Color = clOlive;
}

//-----
```

```
bool TfrmExamFinSumAsses1::isVowel(char Let)
```

```
{ // Question 1.2
  // (5)
```

```
    bool flag = false;
    switch ( toupper(Let) )  {
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U':
            { flag = true;
              break;
            }
    }
```

```
    default: flag = false;
```

```
    }
    return flag;
```

```
}
```

```
//-----
```

```
void __fastcall TfrmExamFinSumAsses1::rdgAnimalsClick(TObject *Sender)
```

```
{ // Question 1.3
```

```
  // (12)
```

```
  AnsiString Animal , PiglatinAnimal;
```

```
  imgAnimal->Stretch = true;
```

```
  imgAnimal->Proportional = true;
```

```
  Animal = rdgAnimals->Items->Strings[rdgAnimals->ItemIndex];    //1
```

```
  imgAnimal->Picture->LoadFromFile("../..\AppData\" + Animal + ".jpg"); //2
```

```
  char Let1 , Let2;
```

```
  Let1 = Animal[1];    //1
```

```
  Let2=  Animal[2];    //1
```

```
  if (!isVowel(Let1) && isVowel(Let2)) {    //1
      PiglatinAnimal = Animal.Delete(1,1);    //1
      PiglatinAnimal = PiglatinAnimal + Let1 + "ay";    //1
  }
```

```
  if (!isVowel(Let1) && !isVowel(Let2)) {    //1
      PiglatinAnimal = Animal.Delete(1,2);    //1
      PiglatinAnimal = PiglatinAnimal + Let1 + Let2 + "ay";
  }
```

```
  if (isVowel(Let1)) {
      PiglatinAnimal = Animal + "way";    //1
  }
```

```
  lblPigLatinAnimal->Caption = PiglatinAnimal;    //1
```

```
/*
```

```
1. If a word starts with a consonant and a vowel,
   put the first letter of the word at the end of the word and add "ay."
   Example: Happy = appyh + ay = appyhay
```

```
2. If a word starts with two consonants move the
   two consonants to the end of the word and add "ay."
   Example: Child = Ildch + ay = Ildchay
```

3. If a word starts with a vowel add the word "way" at the end of the word.

Example: Awesome = Awesome +way = Awesomeway

```

*/
}
//-----

void __fastcall TfrmExamFinSumAsses1::ckbAnimalKeywordClick(TObject *Sender)
{
    // Question 2.1
    // (17)
    lstRawData->AddStrings(redOutput->Lines);    // 1

    redOutput->Lines->Clear();    // 1
    AnsiString oneField;

    int K = 0;

    int AnimalSum = 0;    // 1

    if (ckbAnimalKeyword->Checked == true) {    // 1
        for (int i = 1; i < lstRawData->Count ; i++)    // 1
        {
            OneListLine = lstRawData->Strings[i];    // 1

            oneField = getFieldByNr(OneListLine,3,'\t');    // 1
            oneField = oneField.UpperCase();    // 1

            if (oneField.Pos(cmbKeywordFilter->Text.UpperCase()) > 0) {    // 2
                K++;
                AnimalSum = AnimalSum + getFieldByNr(OneListLine,4,'\t').ToInt();    // 2
                OneListLine = lstRawData->Strings[i];    // 1
                redOutput->Lines->Add(OneListLine);    // 1
            }

            lblRecords->Caption = " Records processed - " + IntToStr(K+1) +
                " Total number of " + cmbKeywordFilter->Text +
                " = " + IntToStr(AnimalSum);    // 2

            lblRecords->Repaint();    // 1
        }
    }
    else
    {
        btnLoadClick(this);
    }
}

//-----

void __fastcall TfrmExamFinSumAsses1::tbsQuestion3Show(TObject *Sender)
{
    // Question 3.1
    // Set up controls //3

    DMod->ADODsRangers->Active = true;

    DBTextRangerID->DataSource = DMod->DSRangers;
    DBTextRangerName->DataSource = DMod->DSRangers;
    DBTextRangerSurname->DataSource = DMod->DSRangers;

    DBTextRangerID->DataField = "RangerID";
    DBTextRangerName->DataField = "Name";
    DBTextRangerSurname->DataField = "Surname";

    // Question 3.2
    // Create Dynamic navigator // 5

```

```

TDBNavigator *RangerNav = new TDBNavigator(this);
RangerNav->Parent = tbsQuestion3;
RangerNav->Top = 105;
RangerNav->Left = 15;
RangerNav->DataSource = DMod->DSRangers;

// Question 3.3
// set up MD relationship // 7
DMod->ADODsTemp->CommandText = "SELECT * FROM Sightings";
DMod->ADODsTemp->DataSource = DMod->DSRangers;

DMod->ADODsTemp->MasterFields = "RangerID";
DMod->ADODsTemp->IndexFieldNames = "RangerID";

DMod->ADODsTemp->Open();
DMod->DSTemp->DataSet = DMod->ADODsTemp;

DBGridSightings->DataSource = DMod->DSTemp;
}

//-----
void __fastcall TfrmExamFinSumAsses1::btnCreateNewSightingIdClick(TObject *Sender)
{
    // Question 4.1
    // (5)

    DMod->ADOUseQuery1->Close();
    DMod->ADOUseQuery1->Connection = DMod->ADODConn;
    DMod->ADOUseQuery1->SQL->Text =
        "SELECT MAX(sightingID) as lastVal FROM sightings";
    DMod->ADOUseQuery1->Open();

    int lastID = DMod->ADOUseQuery1->FieldByName("lastVal")->AsInteger;

    lastID++; // increment it
    lblSightingID->Caption = IntToStr(lastID);
}
//-----

void __fastcall TfrmExamFinSumAsses1::btnSaveSightingClick(TObject *Sender)
{
    // Question 4.2
    // (15)
    DMod->ADODCommand->Connection = DMod->ADODConn;
    AnsiString sql = "INSERT INTO sightings ";
    sql += " (sightingid, sightingdate, Animal, numAnimals, young, rangerID) ";
    sql += " VALUES (:sightid, :dte, :an, :num,:yng, :id)"; // 3

    DMod->ADODCommand->CommandText = sql;
    DMod->ADODCommand->Parameters->ParamByName("sightid")->Value =
        lblSightingID->Caption.ToInt(); // 1
    DMod->ADODCommand->Parameters->ParamByName("dte")->Value = Date(); // 1
    DMod->ADODCommand->Parameters->ParamByName("an")->Value =
        cmbAnimal->Text; // 1
    DMod->ADODCommand->Parameters->ParamByName("num")->Value = sedNumAnimals->Value; // 1

    bool young = false; // 2
    if (cmbYesNo->Text == "Yes") {
        young = true;
    }

    DMod->ADODCommand->Parameters->ParamByName("yng")->Value = young; // 1
    DMod->ADODCommand->Parameters->ParamByName("id")->Value =
        getFieldByNr(cmbRangerDet->Text,1,',').ToInt(); // 2
}

```



```

        DMod->ADOCCommand->Execute();          // 1
        ShowMessage("Sighting recorded");      // 1
        DMod->ADODsSightings->Close();
        DMod->ADODsSightings->Open();

        DMod->ADODsSightings->Refresh();        // 1
    }
//-----

//-----

```

```

void __fastcall TfrmExamFinSumAsses1::FormShow(TObject *Sender)
{
    // Given code do not delete

    FormatSettings.DecimalSeparator = '.';
    FormatSettings.ShortDateFormat = "yyyy/mm/dd";
    FormatSettings.DateSeparator = '/';
    FormatSettings.TimeSeparator = ':';

    AnsiString DBsource = GetCurrentDir();
    DBsource.Delete(DBsource.LastDelimiter("\\"), 40);
    DBsource.Delete(DBsource.LastDelimiter("\\"), 40);
    AnsiString conn = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=";
    conn += DBsource + "\\AppData\\KrugerPark.accdb";
    conn += ";Mode=ReadWrite;Persist Security Info=False";

    DMod->ADOCConn->LoginPrompt = false;

    DMod->ADOCConn->Connected = false;
    DMod->ADOCConn->ConnectionString = conn;
    try
    {
        DMod->ADOCConn->Connected = true;
        DMod->ADODsTariffs->Open();
        DMod->ADODsCamptypes->Open();
        DMod->ADODsCamps->Open();
    }
    catch (EDatabaseError *E)
    {
        ShowMessage("Can't connect!");
    }

    pgctrlMain->ActivePageIndex = 0;

    lstRawData = new TStringList;
}
//-----

```

```

void __fastcall TfrmExamFinSumAsses1::tbsQuestion4Show(TObject *Sender)
{
    // Given code

    DMod->ADODsSightings->Active = true;
    grdSightings->DataSource = DMod->DSSightings;

    AnsiString SQLString = "Select str(RangerID) + ',' + Surname as RDet from
Rangers";
}

```

```
DMod->ADOUseQuery1->Close();
DMod->ADOUseQuery1->SQL->Clear();
DMod->ADOUseQuery1->SQL->Add(SQLString);
DMod->ADOUseQuery1->Open();

cmbRangerDet->Items->Clear();
for (DMod->ADOUseQuery1->First(); !DMod->ADOUseQuery1->Eof; DMod->ADOUseQuery1->Next()) {
    cmbRangerDet->Items->Add(DMod->ADOUseQuery1->FieldByName("RDet") ->AsAnsiString);
}

cmbRangerDet->ItemIndex = 0;
}
//-----
```