EXAMINATION	NATIONAL SENIOR CERTIFICATE
GRADE	12
DATE	NOVEMBER 2025
SUBJECT	INFORMATION TECHNOLOGY
PAPER	1
MARK TOTAL	150
DURATION (HOURS)	3
NUMBER OF PAGES	26

SOUTH AFRICAN COMPREHENSIVE ASSESSMENT INSTITUTE

SUID-AFRIKAANSE KOMPREHENSIEWE ASSESSERINGSINSTITUUT

INSTRUCTIONS AND INFORMATION

1. The question paper consists of four sections (four questions).

   SECTION A: GENERAL PROGRAMMING SKILLS
   SECTION B: SQL AND ADO-DATABASE PROGRAMMING
   SECTION C: OBJECT-ORIENTED PROGRAMMING
   SECTION D: PROBLEM SOLVING PROGRAMMING

2. Save your work at regular intervals as a precaution against power failures.

3. Make sure that you enter your exam number as a comment on the third line in each program (**QUESTION 1** to **QUESTION 4**) unit, AS WELL AS in the class unit (**QUESTION 3**).

4. Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for the data validation.

5. Routines must be developed from first principles. You may **NOT** use the built-in features of Delphi for any of these routines:

String handling: Substring, Split, Stringlist, Comparestr, Stringreplace, Stringtointdef, Reversestring, Shl-function for movement in a string, Stringofchar	**Arrays:**	Max/min, Remove
	SQL:	Datediff
	ADO:	Locate, Sort, Filter, Search
	Text files:	LoadFromFile

6. Non-programmable pocket calculators may be used.

7. You may comment out a specific part of your code if that part doesn't compile. All commented code will be marked if there is no alternative code.

8. Notes on Question 2:

   Field names for the tables are included as comments, you may copy and paste these field names in your answers for question 2.

9. During the control session, you should check all your units, namely **Question1_U, Question2_U, clsQuestion3_U, Question3_U** and **Question4_U.**

   - Can the project open and run?
   - Is your exam number entered as a comment in the unit?

10. Two blank pages are included at the back of the paper for planning purposes.

11. The files you need to complete **QUESTION 1** to **QUESTION 4** can be found in the **'Data_Nov2025'** folder.

    Rename this folder using your **exam number. e.g. 2589911_ Nov2025.**

    It is your responsibility to ensure that all the listed files are visible in your exam data folder before the three-hour examination commences.

Question1:	**Question3:**
Indicators.jpg	clsQuestion3_u.pas
Question1_p.dpr	Question3_p.dpr
Question1_p.dproj	Question3_p.dproj
Question1_p.res	Question3_p.res
Question1_u.dfm	Question3_u.dfm
Question1_u.pas	Question3_u.pas
PollutionData.txt	
Question2:	**Question4:**
RedGreen.mdb	Question4_p.dpr
RedGreenBCK.mdb	Question4_p.dproj
Question2_p.dpr	Question4_p.res
Question2_p.dproj	Question4_u.dfm
Question2_p.res	Question4_u.pas
Question2_u.dfm	
Question2_u.pas	
DBConnection_U.pas	

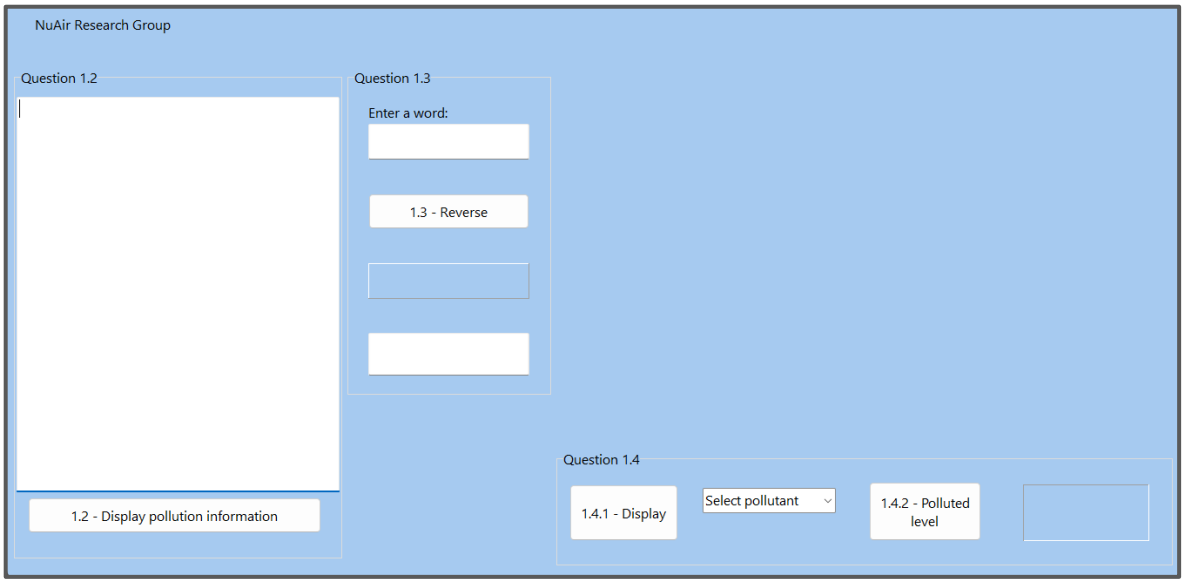
SECTION A: GENERAL PROGRAMMING SKILLS

QUESTION 1

A research team investigating pollution requires a set of small software applications to assist with their work.

- Open the incomplete **Question1_p.dpr** program in your **ExamNumber_Nov2025/Question1** folder.
- Add your exam number as a comment on the third line of the program unit.
- Write code to complete **QUESTION 1.1 to QUESTION 1.4** as instructed in the question paper.

Graphical user interface:



1.1 **[Onactivate event handler]**

Write code to do the following:

- Change the colour of the **frmQuestion1** form to '**btnFace**'.
- Change the appearance of the heading of the **lblQ1_1** label to font type 'Forte' and size 36 pt.
- Set the x-position of the **lblQ1_1** label to 200.

Example of output:

(4)

1.2 **Button [1.2 – Display pollution information]**

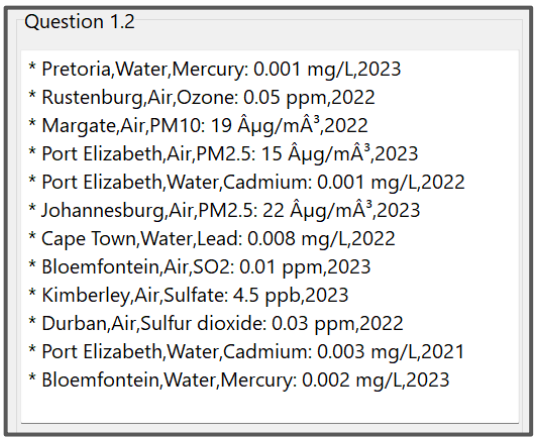
Research data on pollution is stored in a text file, called **PollutionData.txt**.

Write code to do the following:

- Open the text file.
- Add an '*' (asterisk) in front of each line in the text file.
- Display all the lines in the text file in the **memQ1_2** memo.

NOTE: You do not have to test if the file exists.

Example of output:



(7)

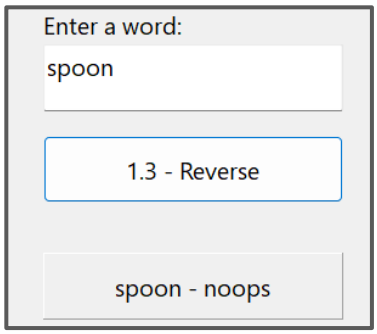
1.3 **Button [1.3 – Reverse]**

The research team enjoys playing word games.

1.3.1 Write code to do the following:

- Retrieve the word entered from the **edtQ1_3_1** edit box.
- Reverse the word, e.g. 'Holiday' changes to 'yadiloH'.
- Display both words on the **pnlQ1_3_1** panel with a - (hyphen) between the words.

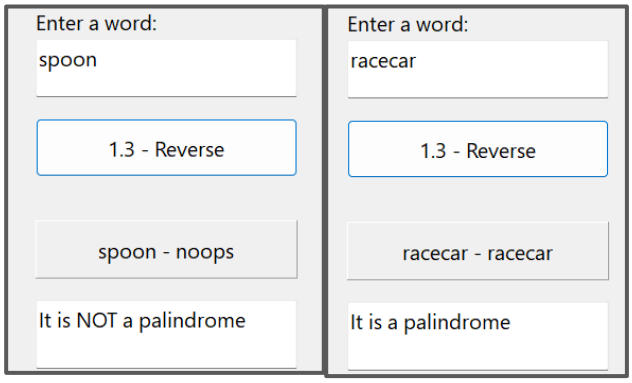
Example of output:



(8)

1.3.2
- Display the message 'It is a palindrome' in the **edtQ1_3_2** edit box if the reversed word is the same as the original word.
- Display the message 'It is NOT a palindrome' in the **edtQ1_3_2** edit box if the reversed word is not the same as the original word.

Examples of output:



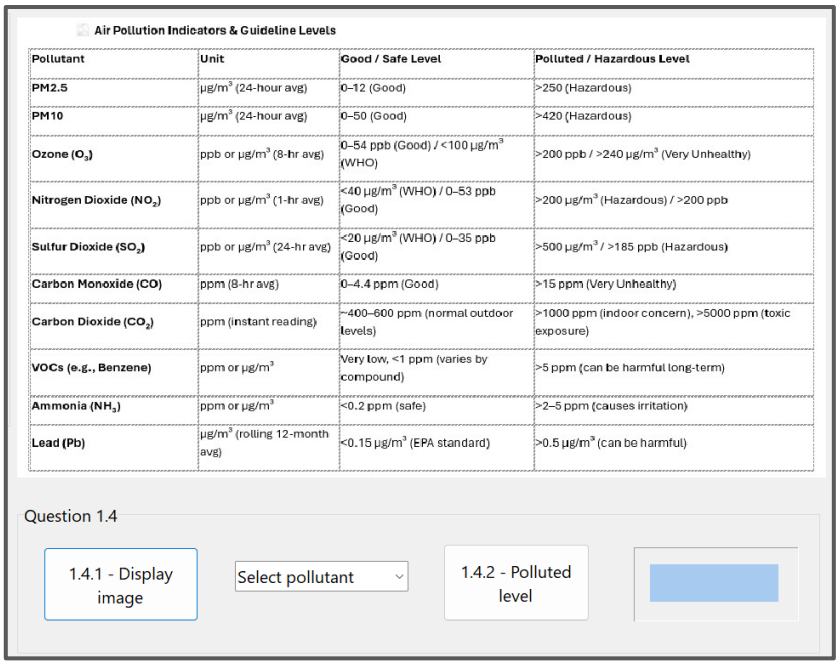
(6)

1.4 A table with pollution indicators is saved as an image file named: **'Indicators.jpg'**.

1.4.1 **Button [1.4.1 – Display]**

Write code to:
- Display the picture in the **imgQ1_4** image component.
- Ensure that the image fills the component completely.

Example of output:



(2)

1.4.2 Button [1.4.2 – Polluted level]

The extent of pollution needs to be measured.

Write code to do the following:

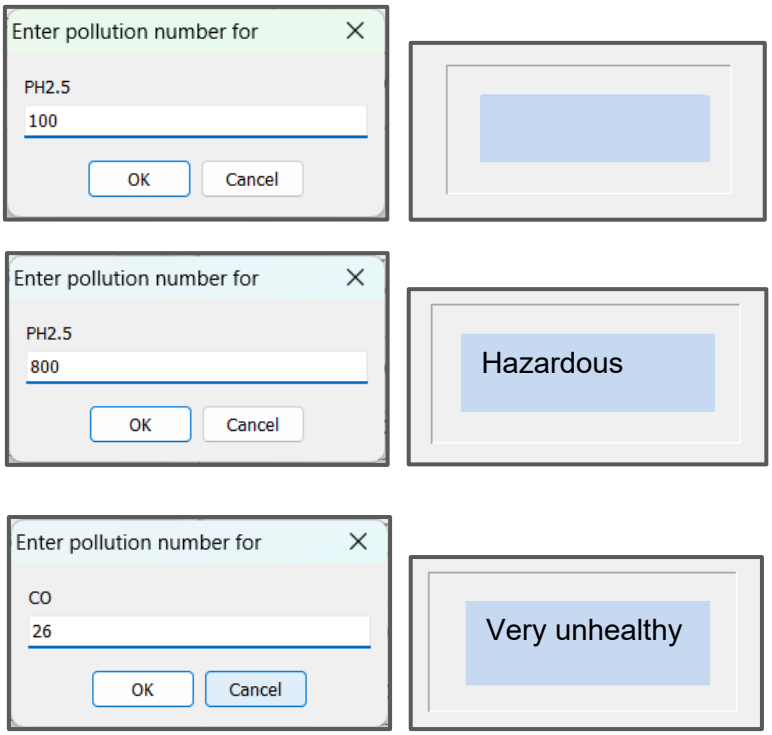
- Retrieve the selected pollutant from the **cmbQ1_4_2** combo box.
- Use a dialogue box (input box) to enter the pollution reading for the selected pollutant. Show the heading "`Enter pollution number for`" as well as the selected pollutant as part of the input box.
- Remember, a real number can be entered as a reading.
- Use the table below to determine the hazardous level.
- Display the hazardous level on the **lblQ1_4_2** label.
- Change the label's colour to red if the level is 'hazardous' or 'very unhealthy'.

NOTE: It is not necessary to reset the colour of the label.

The following table is used to determine the hazardous level.

Pollutant	Pollution reading	Hazardous level
PH2.5	>250	Hazardous
PH10	>420	Hazardous
O_3	>500	Very unhealthy
NO_3	>200	Hazardous
SO_3	>500	Hazardous
CO	>15	Very unhealthy

Examples of output:



(14)

TOTAL SECTION A: [41]

SECTION B: SQL AND ADO-DATABASE PROGRAMMING

QUESTION 2

Information on traffic offences are saved in the database **RedGreen.mdb**

The database consists of four tables:

NOTE: You will only use **one** or **two** tables for a given question at any point.

The design and content of the **RedGreen.mdb** database are attached at the end of the question paper.

- Open the incomplete **Question2_p.dpr** program in your **ExamNumber_Nov2025**/**Question2** folder.
- Add your exam number as a comment on the third line of the program unit.
- Write code to complete **QUESTION 2.1** (SQL) and **QUESTION 2.2** (ADO database management) as instructed in each section.
- The **'Restore database'** button is provided to restore the data in the database to the original content.

NOTE: Code is provided to connect the database to Delphi. The database is password protected, so you will not be able to access it directly.

The following form class variables are declared:

```
tblDrivers, tblOffences, tblGroupCodes, tblOffenceCodes: TADOTable;
```

The table names and field names are provided as commented code to copy into your answers.

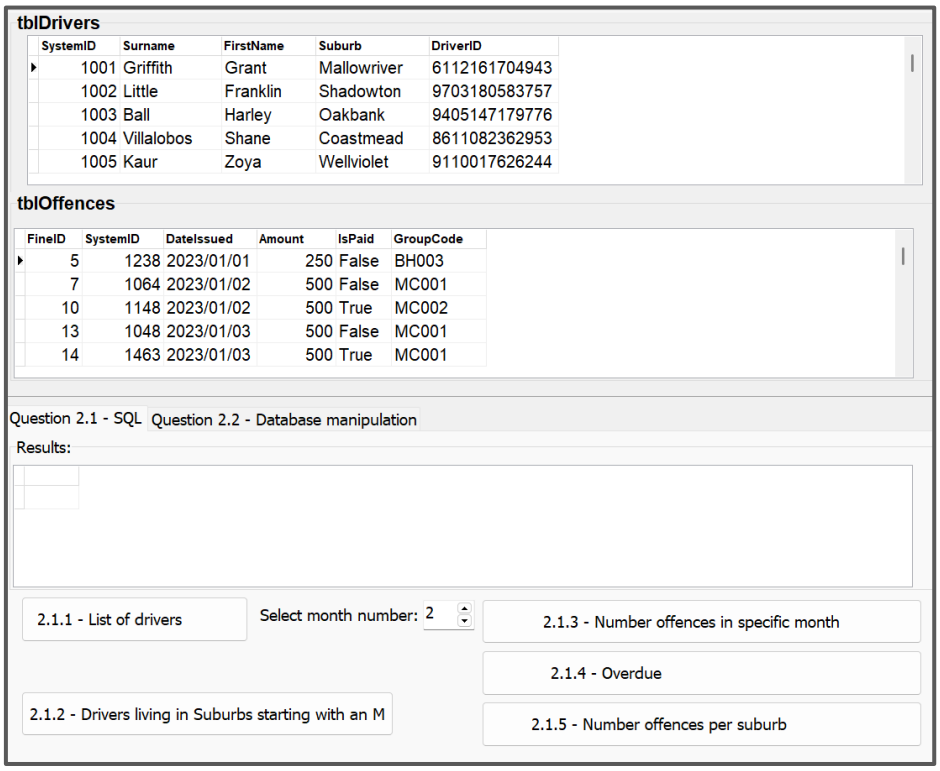
2.1 Tab sheet [Question 2.1 - SQL]

Type your SQL statement in the provided code between the ‘ ’ of the line:

```
SQLQry := ‘ ’;
```

Code has been provided to execute the SQL statements.
SQL statements MUST be used to answer question 2.1.1 to question 2.1.5.
NO marks will be allocated if Delphi code constructs are used.

Graphical user interface for QUESTION 2.1:



2.1.1 **Button [2.1.1 – List of drivers]**

Write an **SQL** statement to display all the records from the **tblDrivers** table.

An example of output for the first four records:

SystemID	Surname	FirstName	Suburb	DriverID
1001	Griffith	Grant	Mallowriver	6112161704943
1002	Little	Franklin	Shadowton	9703180583757
1003	Ball	Harley	Oakbank	9405147179776
1004	Villalobos	Shane	Coastmead	8611082362953

(2)

2.1.2 **Button [2.1.2 – Drivers living in Suburbs starting with an M]**

Write an **SQL** statement to display the **SystemID**, **Surname** and **Suburb** of drivers who live in suburbs starting with the letter M.

An example of output for the first four records:

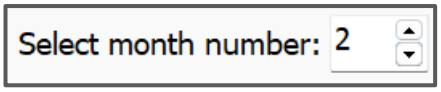
SystemID	Surname	Suburb
1001	Griffith	Mallowriver
1024	Davenport	Mallowriver
1041	Hartman	Mistwick
1105	York	Mallowwyvern

(4)

2.1.3 **Button [2.1.3 – Number of offences in specific month]**

Write an **SQL** statement to display the date issued and group code, from **tblOffences** table, for all offences that occurred in the month selected in the **spnQ2** spin edit control.

Example output:



DateIssued	GroupCode
01/02/2023	MC090
02/02/2023	BH004
02/02/2023	MC001
03/02/2023	MC001

(3)

2.1.4 **Button [2.1.4 – Overdue]**

Write an **SQL** statement that displays the **SystemID**, the **Amount** (labelled as 'Amount' in currency format), and the **DateIssued** for all fines that are unpaid and have been overdue by more than 45 days. Use the current date to determine which fines are overdue.

An example of output for the first four records:

SystemID	Amount	DateIssued
1238	R250.00	2023/01/01
1064	R500.00	2023/01/02
1048	R500.00	2023/01/03
1119	R500.00	2023/01/04

(5)

2.1.5 **Button [2.1.5 – Number offences per suburb]**

Write an **SQL** statement to count the number of offences in each suburb and display the total as 'Number of offences per suburb'.
TIP: You will use the **tblOffences** and **tblDrivers** tables as part of your SQL statement.

Example of output for the first four records:

Suburb	Number of offences per suburb
Aldlake	79
Aldmoor	76
Baybush	24
Baycliff	21

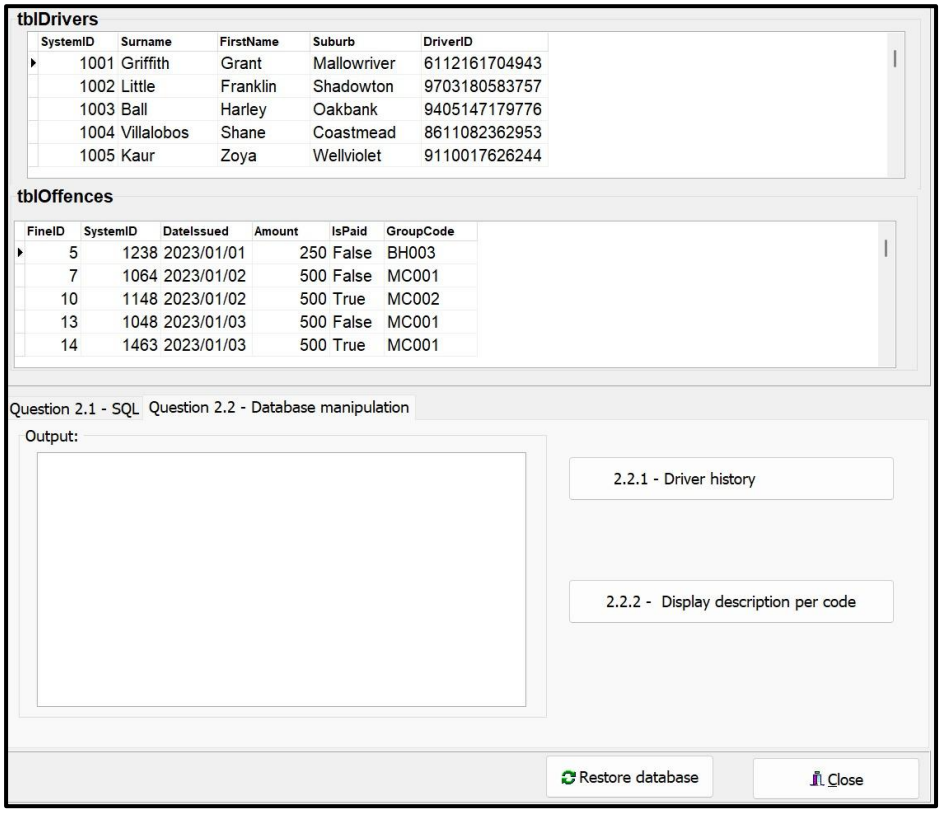
(5)

[19]

2.2 Tab sheet [Question 2.2 - DB Operations]

NOTE: NO marks will be awarded for SQL statements in QUESTION 2.2.

Graphical user interface for QUESTION 2.2.



2.2.1 Button [2.2.1 – Driver history]

Write code to do the following:

- Display the currently selected driver's surname and first name as part of a heading with the following format in the **redQ2** rich edit:

  `OUTSTANDING FINES FOR: <Surname>, <Firstname>`

- Display the **FineID**, **DateIssued**, and **Amount** of all unpaid fines for the currently selected driver in the **redQ2** rich edit, in neat columns.
- Count the number of paid fines and display the result below the list of fines, as shown in the example below.

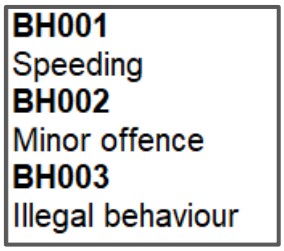
```
OUTSTANDING FINES FOR: Griffith, Grant

FineID Date Issued  Amount
4580   2024/05/06   R500.00
6575   2022/05/01   R250.00
9016   2025/06/01   R500.00

Number of PAID fines for this driver: 7
```

(11)

2.2.2 **Button [2.2.2 – Display description per code]**

Write code to display the **GroupCode** (in bold) and the **Description** fields for the first three records from the **tblGroupCodes** table in the **redQ2** rich edit.

Example of output:



(8)

[19]

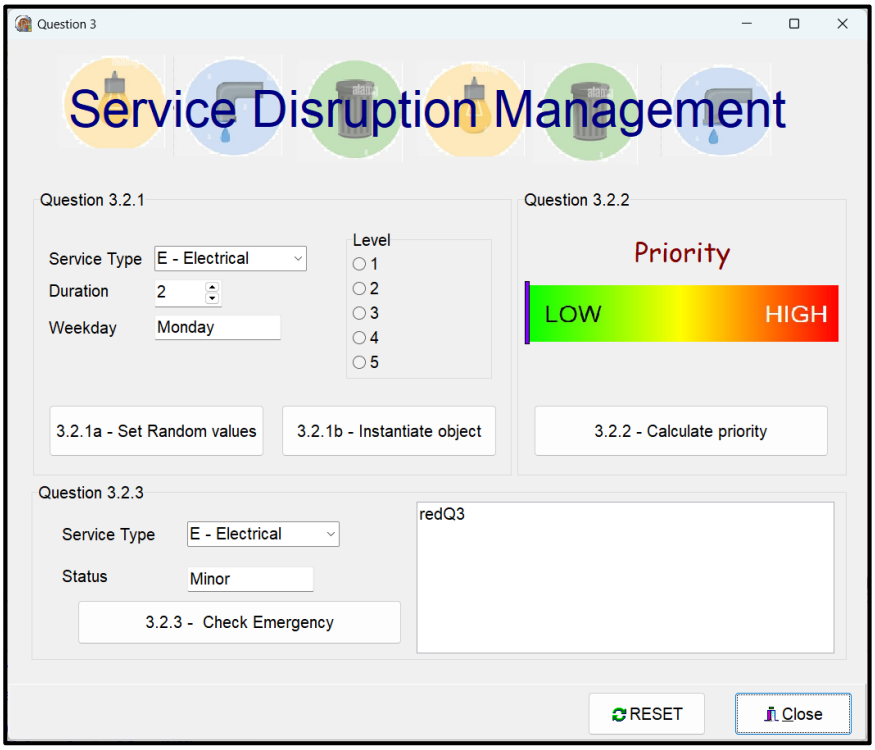
TOTAL SECTION B: [38]

SECTION C: OBJECT-ORIENTED PROGRAMMING

QUESTION 3

- Open the incomplete **Question3_p.dpr** program and **clsQuestion3_u.pas** unit in your **ExamNumber_Nov2025/Question3** folder.
- Add your exam number as a comment on the third line of the program and class unit.
- Write code to complete **QUESTION 3.1** and **QUESTION 3.2** as instructed in the question paper.

Graphical user interface:



3.1

A service disruption management system is needed to monitor and manage different types of service disruptions (electrical, water, and garbage) across a municipality. Each disruption is evaluated based on its type, duration, day of occurrence, and level of severity to calculate priority and determine appropriate status.

An incomplete object class **TDisruption** is provided in the **clsQuestion3_U.pas** unit.

You may NOT add any attributes (fields) or methods except when specifically indicated in a question.

The attributes of the disruption object have been declared as follows:

ATTRIBUTE/PROPERTY	DESCRIPTION
`fServiceType`	The type of service affected (E - Electrical, W - Water, G - Garbage)
`fDuration`	The number of full hours the disruption is expected to last
`fWeekday`	The day of the week when the disruption will occur
`fLevel`	A value from 1 to 5 indicating the severity level of the disruption
`fIsPlanned`	A Boolean value indicating whether the disruption is planned (true) or an emergency (false)

Complete the code in the object class, **TDisruption,** as described in **QUESTION 3.1.1** to **QUESTION 3.1.5** below.

3.1.1 Write code for a constructor method, **a_Create**, to receive the service type, duration, weekday, and level as parameters and do the following:

- Assign the parameter values to the respective attributes.
- Assign the value of TRUE to the **fIsPlanned** attribute. (5)

3.1.2 Write code for an accessor method called **b_getLevel** that will return the value of the **fLevel** attribute. (2)

3.1.3 Write code for a method called **c_calcPriority** that will calculate and return the priority of the disruption as an integer, based on the service type (**fServiceType**) and level (**fLeve**l) attributes.
The priority is calculated as follows, for:

- Electrical service (type 'E'): Priority = **fLevel** × 30
- Water service (type 'W'): Priority = **fLevel** × 25
- Garbage service (type 'G'): Priority = **fLevel** × 10 (6)

3.1.4 Write code for a method called **d_setEmergency** that will:

- Set the **fIsPlanned** attribute to FALSE.
- Increase the **fLevel** attribute by 1 if it is less than 5. (3)

3.1.5 A partially completed method called **e_determineStatus** has been provided. Code has been provided to set the status to 'Normal'.

Add code to complete the method using the **fIsPlanned** and **fLevel** attributes to establish the current status of the disruption according to the following rules:

STATUS	CONDITION
Emergency	If fIsPlanned is FALSE
Minor	If fIsPlanned is TRUE and fLevel is 1 or 2
Moderate	If fIsPlanned is TRUE and fLevel is 3 or 4
Major	If fIsPlanned is TRUE and fLevel is 5
Normal	All other cases

(7)

NOTE: A complete **toString** method has been provided. You will use this method later in question 3.2.3

[23]

3.2 An incomplete form class called **frmQuestion3_U** has been provided in the **Question3_U.pas** unit.

The following declarations have been provided may be used in your answer:

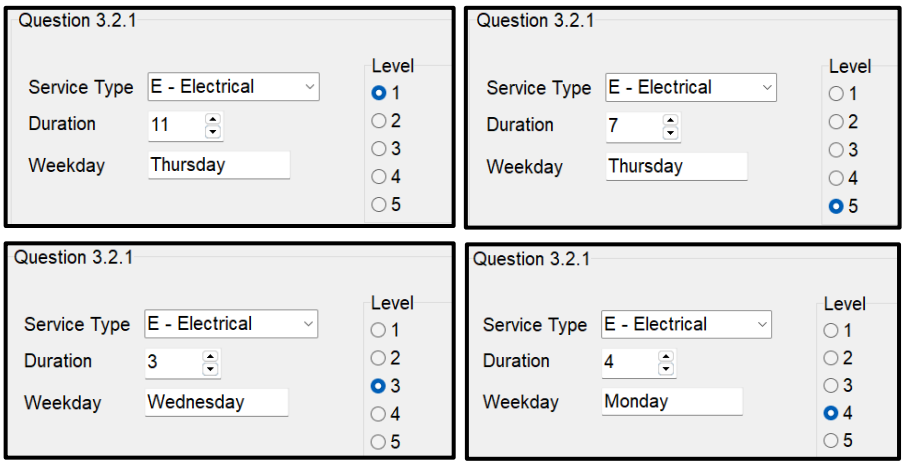
- ```
  Const
  arrWeekdays: array[1..7] of string =
  (‘Monday’,’Tuesday’,’Wednesday’,’Thursday’,
  ’Friday’,’Saturday’,’Sunday’)
  //represents all the days in a week
  ```
- ```
  objDisruption : TDisruption
  //to represent the object
  ```
- ```
  sType, sWeekday: string
  //represents the service type and day of the week of a
  disruption
  ```
- ```
  iLevel, iDuration: Integer;
  //represents the level and hours that the disruption will last
  ```

3.2.1 a) **Button [3.2.1a - Set Random values]**

Write code to do the following:

- Assign a random value between 1 and 12, both included, to the duration and display the random value in the **sedQ3_2_1** spin edit box.
- Assign a random day from the **arrWeekdays** array to the **sWeekday** variable.
- Assign a random value between 1 and 5, both included, to the **iLevel** variable and display the random value on the **rdgQ3_2_1** radiogroup.

Examples of possible output (your answers will differ because it is randomly assigned):



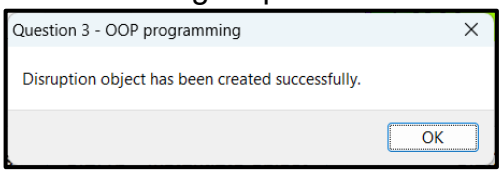
(3)

3.2.1 b) **Button [3.2.1b - Instantiate object]**

Write code to do the following:

- Retrieve the value the user selected from the combo box.
- Instantiate a **TDisruption** object, called **objDisruption**, passing the generated and selected values over as parameters.

Code is provided for the following output:



(3)

3.2.2 **Button [3.2.2 – Calculate priority]**

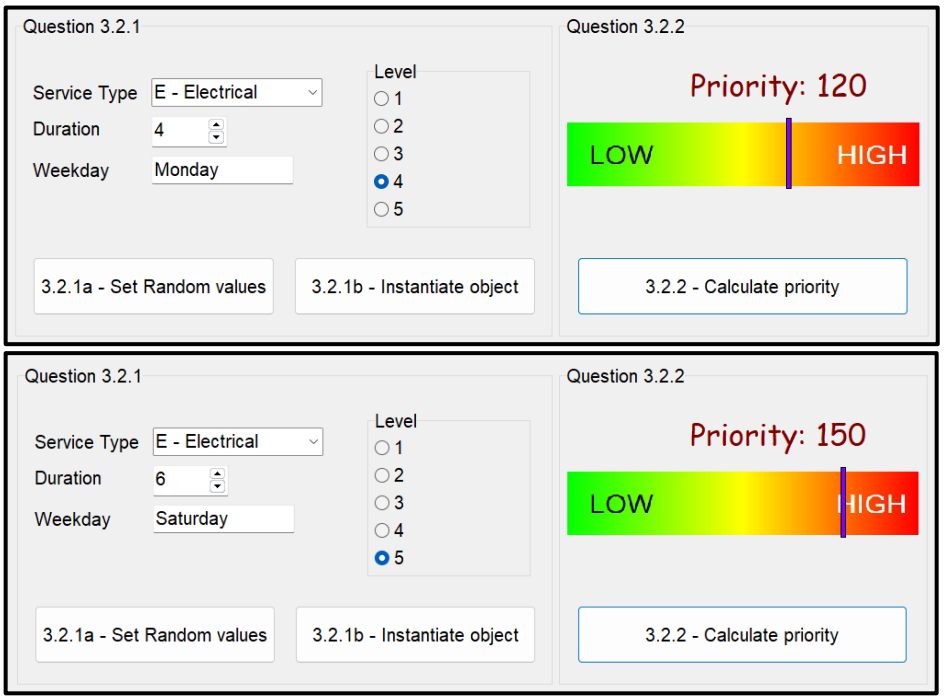
Write code to do the following:

- Call the **objDisruption** object's **c_calcPriority** method to determine the priority value.
- Move the **shpQ3_2_2** shape horizontally on the priority scale using the following pseudocode as reference:

  ```
  Left position of shpQ3_2_2 = 7 + (priority × 2)
  ```

- Display the **objDisruption** object's priority on the **lblQ3_2_2** label in the format: "Priority:" + [value].

Examples of possible output:



(4)

3.2.3 Button [3.2.3 – Check Emergency]

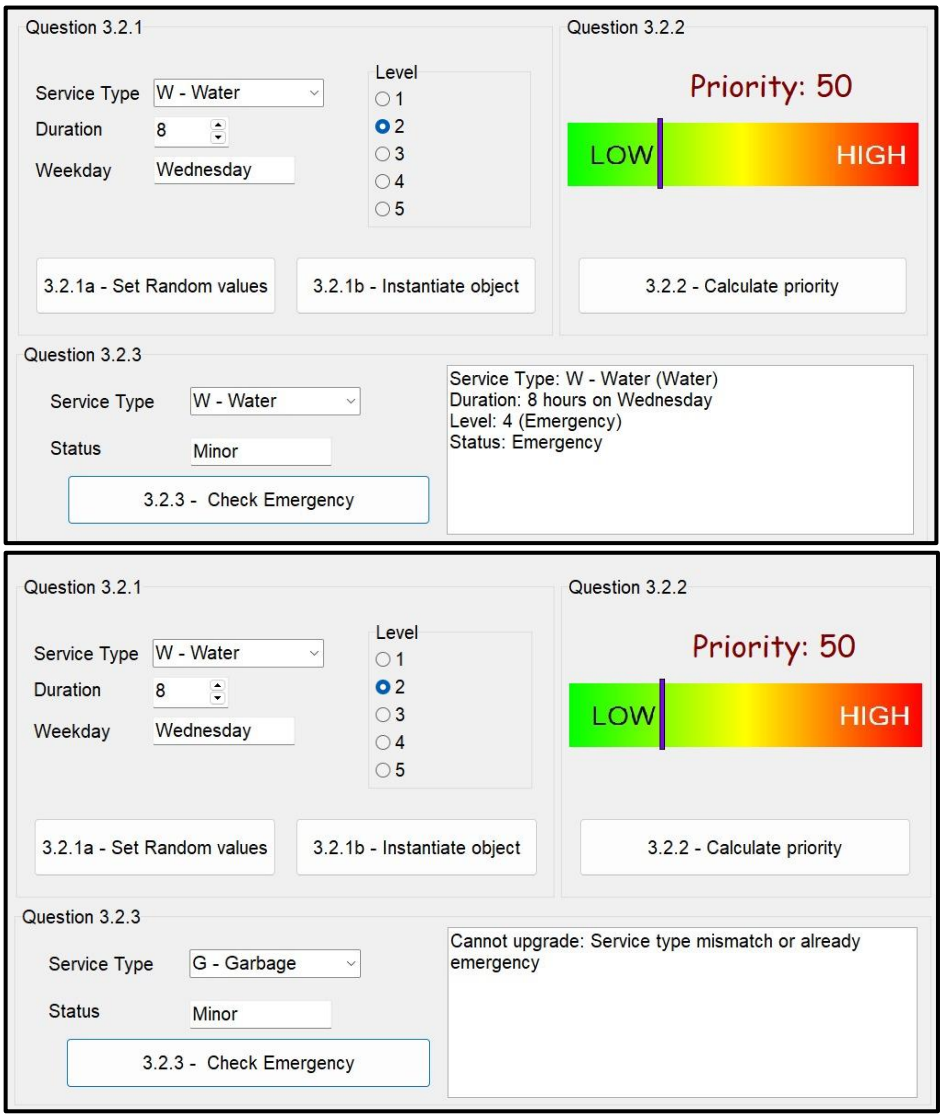
The user must select the service type from the **cmbQ3_2_3** combo box and enter the status in the **edtQ3_2_3** edit box.

Code has been provided to retrieve the service type from the **cmbQ3_2_3** combo box and the status entered in the edit box, **edtQ3_2_3**.

Write code to do the following:

- Check if the selected service type matches the **objDisruption** object's original service type (**sType**).
- Test if the entered status is not an 'Emergency'
- Do the following if both previous conditions are true:
  - Call the **d_setEmergency** method to update the disruption object
  - Display the updated object's information using **toString** method in the **redQ3** rich edit box.
  - Add a line showing the status by calling the **e_determineStatus** method.
- Display the message, "Cannot upgrade: Service type mismatch or already emergency", if either condition is false.

Examples of possible output:



(7)

[17]

TOTAL SECTION C: [40]

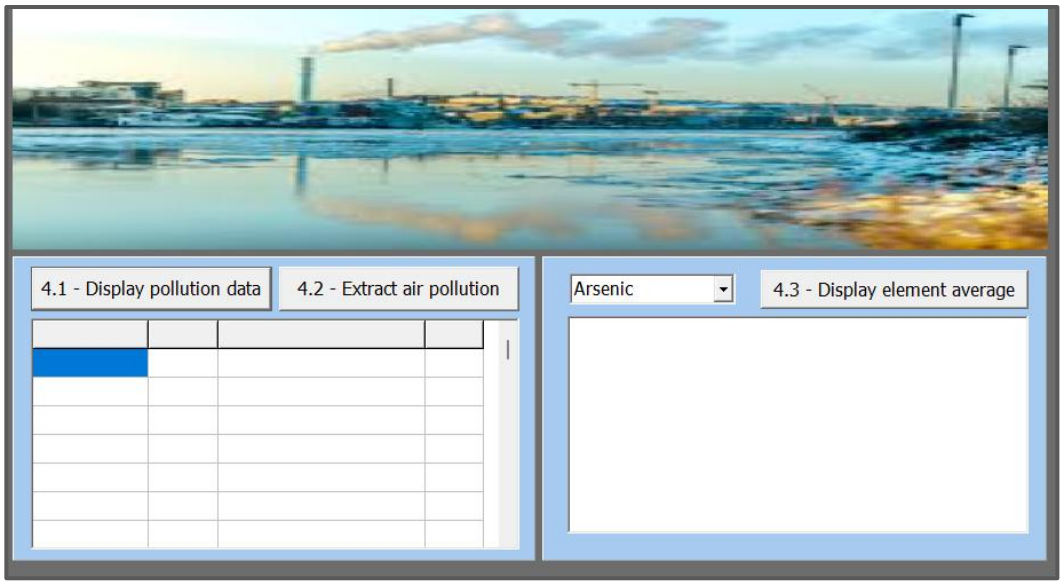
SECTION D: PROBLEM SOLVING PROGRAMMING

QUESTION 4

- Open the incomplete **Question4_p.dpr** program in your **ExamNumber_ Nov2025/Question4** folder.
- Add your exam number as a comment on the third line of the program unit.
- Add code to complete **QUESTION 4.1** – **QUESTION 4.3** as instructed in the question paper**.**
- Do not change any provided code and save your work regularly.

Riverport is a mid-sized town located near a river and surrounded by hills. It hosts several industries, including a steel factory, a chemical manufacturing plant, and a coal-fired power station. The Environmental Monitoring Agency conducted a recent study on Riverport's air and water quality.

Graphical user interface:



Study the given declarations in the program as well as the following declarations, which have been provided, that you must use in your answers.

- ```
  arrPollutionData: array[1..50,1..4] of string;
  //Represents the general pollution information
  //Currently there are (28) rows of data with some empty rows
  ```
- ```
  arrWaterData: array[1..25,1..3] of string;
  //Represents the water pollution information
  //Currently there are (10) rows of data with some empty rows
  ```
- ```
  arrAirData: array[1..25,1..3] of string;
  //Represents the air pollution information
  //Contains no data currently
  ```
- ```
  arrHeadings: array[1..4] of string;
  //Represents the headings for the string grid
  ```

4.1 **Button [4.1 - Display pollution data]**

Write code to display the data in **arrPollutionData** in the **stgQ4_1** string grid, with the headings as provided in **arrHeadings.**
Use the given **iRow** and **iCol** variables as part of your solution.

Example of output:

Location	Type	Level	Year
Port Elizabeth	Air	Ozone: 0.05 ppm	2024
Kimberley	Water	Arsenic: 0.0015 mg/L	2025
Pretoria	Air	NO2: 27 Âµg/mÂ³	2025
Durban	Water	Arsenic: 0.005 mg/L	2025
Johannesburg	Water	Arsenic: 0.010 mg/L	2025
Sasolburg	Air	Benzene: 1.5 ppb	2025
Polokwane	Air	PM2.5: 23 Âµg/mÂ³	2025

(5)

4.2 **Button [4.2 – Extract air pollution]**

The **arrAirData** array must be populated using the **arrPollutionData** array.

The type column indicates the type of pollution and only records that indicate **Air** should be transferred from the **arrPollutionData** array to the **arrAirData** array. In addition, only the latest (most recent) reading per location should be transferred.

The columns in the **arrPollutionData** array respectively represent the:

```
<Location>,<Pollution Type>,<Pollution Level>,<Year>
```

The columns in the **arrAirData** array respectively represent the:

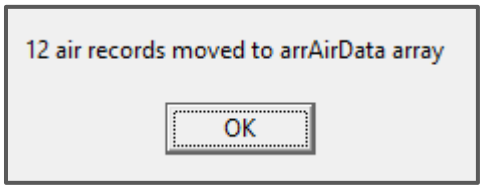
```
<Location>,< Pollution Level>,<Year>
```

The following variable is declared and may be used in your answer:

- **iCountAir** to count the number of rows added to the **arrAirData** array.

Write code to extract and store the latest data for air type pollution in the provided **arrAirData** array.
For example, Cape Town has two air pollution readings; only the data for the latest available year (2024) should be added to the **arrAirData** array.
A message should inform the user how many rows have been stored to the **arrAirData** array.
NOTE: The content of the **arrAirData** array is not to be displayed.

Example of message:



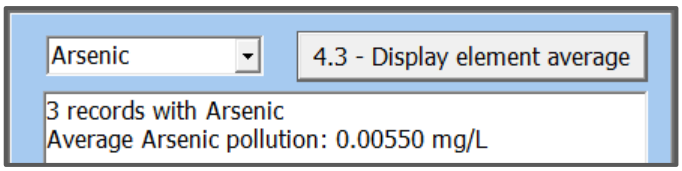
(15)

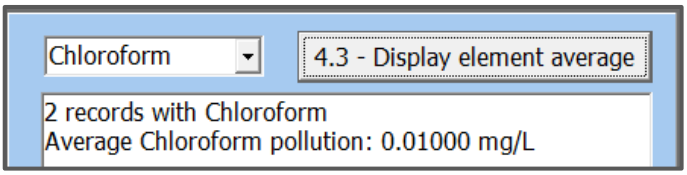
4.3 **Button [4.3 - Display element average]**

Write code to do the following:

- Retrieve the value the user selected from the combo box.
- Go through **arrWaterData** 2D-array and calculate the average pollution of the selected element.

Examples of output
NOTE: Code is provided to display output:





(11)

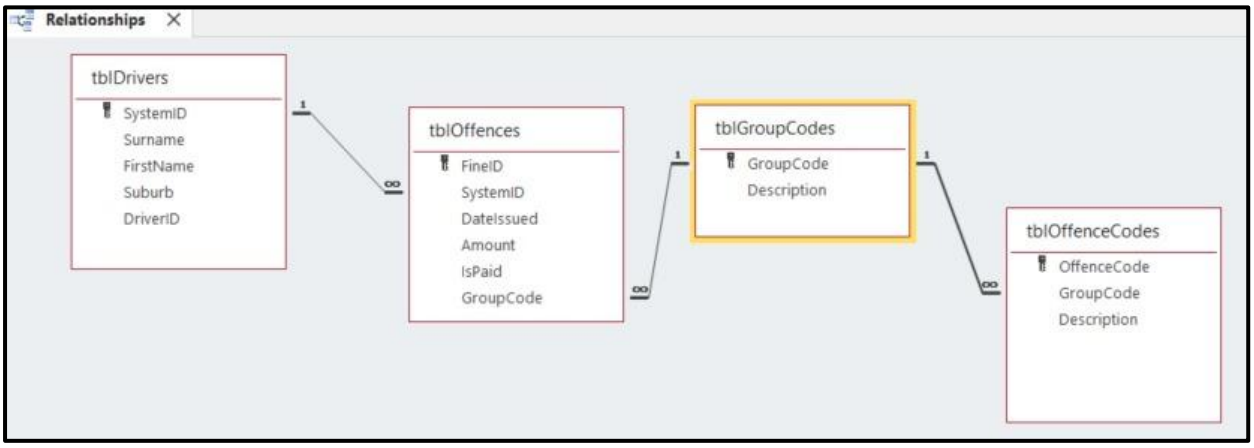
TOTAL SECTION D: [31]

GRAND TOTAL: [150]

DATABASE INFORMATION QUESTION 2:

The database **RedGreen.mdb** consists of four tables.

The following one-to-many relationship with referential integrity exists between the database tables that is needed to answer the question:



tblDrivers table: Design view

tblDrivers

Field Name	Data Type	
SystemID	AutoNumber	Uniques number for customer
Surname	Short Text	Surname of customer
FirstName	Short Text	Firstname of customer
Suburb	Short Text	Suburb were customer resides
DriverID	Short Text	National ID number

tblDrivers table: First few records

tblDrivers

SystemID	Surname	FirstName	Suburb	DriverID
1001	Griffith	Grant	Mallowriver	6112161704943
1002	Little	Franklin	Shadowton	9703180583757
1003	Ball	Harley	Oakbank	9405147179776
1004	Villalobos	Shane	Coastmead	8611082362953
1005	Kaur	Zoya	Wellviolet	9110017626244
1006	Manning	Augustine	Beldell	9302290887178
1007	Stanley	Jennifer	Greyhedge	8202177432454
1008	Terry	Manuel	Lochcliff	9710198048975
1009	Bravo	Wren	Northton	6709228577756
1010	Cantrell	Genesis	Summerton	6110174220704

tblGroupCodes table: Design view

tblGroupCodes

Field Name	Data Type
GroupCode	Short Text
Description	Short Text

tblGroupCodes table: First few records

tblGroupCodes

GroupCode	Description
BH001	Speeding
BH002	Minor offence
BH003	Illigal behaviour
BH004	Driver problems
BH005	Pedestrian
BH006	Rule ignoring
BH010	Vehicle illigal
BH020	Fatal offence
MC001	Basic vehicle
MC002	Licences
MC090	Unroadworthy

tblOffenceCodes table: Design view

tblOffenceCodes

Field Name	Data Type
OffenceCode	AutoNumber
GroupCode	Short Text
Description	Short Text

tblOffenceCodes table: First few records

tblOffenceCodes

OffenceCode	GroupCode	Description
1	BH003	Illegal entrances and exits
2	MC001	Defective/incorrectly fitted rear underrun bumper
3	MC002	Owner did not report motor vehicle permanently unfit for use
4	MC001	Parking brake inadequate on non-RWC vehicle
5	BH002	Motor vehicle with plates not applicable
6	MC001	Defective seat belt
7	BH006	Not using dipped beam with oncoming traffic
8	BH004	Operating vehicle with special licence on public road
9	BH001	Exceeding the speed limit by:11 km/h to 15 km/h
10	LI001	Failed to produce licence to court

Please turn over

tblOffences table: Design view

tblOffences

Field Name	Data Type	
FineID	AutoNumber	Unique number for every traffic fine issued
SystemID	Number	Unique number of customer
DateIssued	Date/Time	Date the traffic fine was issued
Amount	Currency	Amount of the fine
IsPaid	Yes/No	Indicate if fine is payed or not
GroupCode	Short Text	Unique code for group of offences

tblOffences table: First few records

tblOffences

FineID	SystemID	DateIssued	Amount	IsPaid	GroupCode
7	1064	2023/01/02	R500.00	☐	MC001
10	1148	2023/01/02	R500.00	☑	MC002
13	1048	2023/01/03	R500.00	☐	MC001
14	1463	2023/01/03	R500.00	☑	MC001
22	1522	2023/01/04	R99 999.99	☑	BH002
23	1128	2023/01/04	R250.00	☑	MC001
24	1119	2023/01/04	R500.00	☐	BH006
26	1279	2023/01/05	R500.00	☐	BH004
30	1241	2023/01/05	R500.00	☐	BH006
33	1383	2023/01/06	R250.00	☐	BH001

PLANNING PAGES