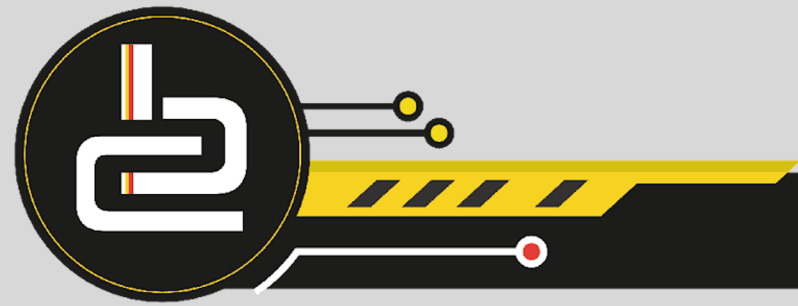




Delphi - Database

V. Pretorius



Lesson objectives

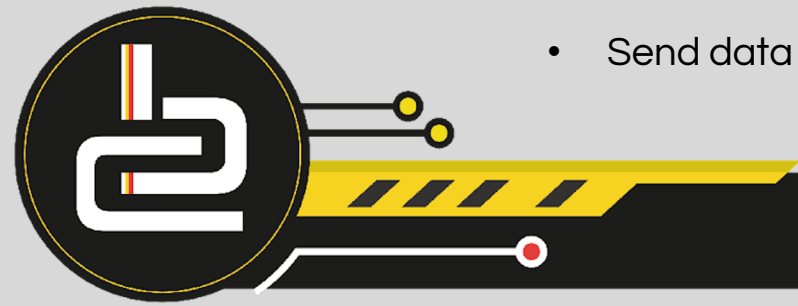
- Work with fields in the database
- Glossary
- Components for the manipulation of data
- Processing data
- Crud operations
- Working with text files and the database



Work With Fields Inside The Database

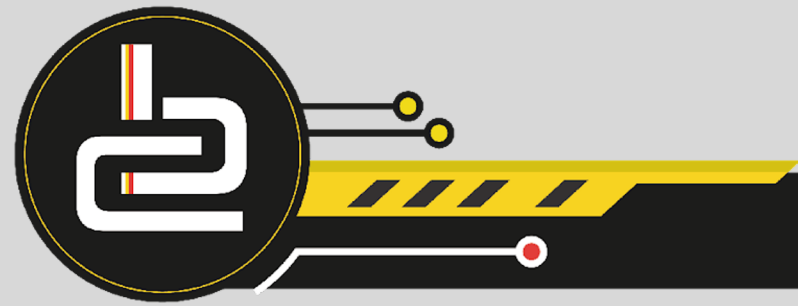
In this section we will concentrate on the following.

- How can we use data from the database in Delphi.
- Count the number of records in the table.
- Seek for a specific record in the table.
- Add and delete data.
- Filter records according to given criteria.
- Edit record data from Delphi.
- Sort the data
- Data validation
- Send data between text files and databases.



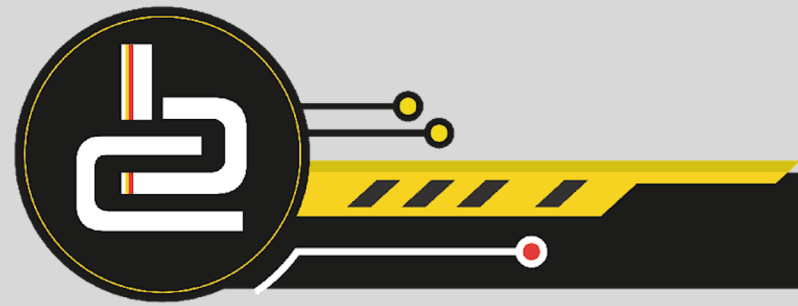
New Concepts

- The following concepts and components are very important.
- It will count in your favour if you learn these concept early on and know how to apply them.



Glossary

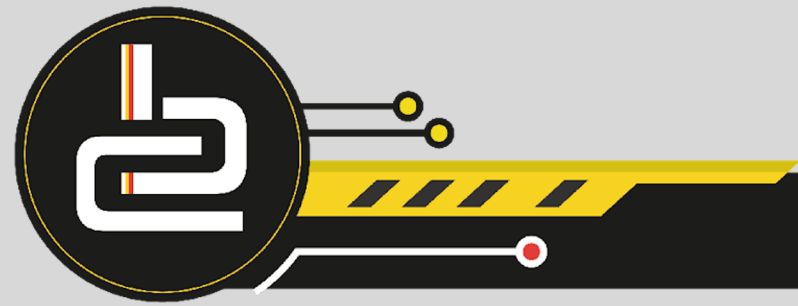
- **FIRST:** Move the file marker to the beginning of the table.
- **NEXT:** The file marker will move to the next record.
- **LAST:** Move the file marker to the last record in the table.
- **INSERT:** Insert a empty record in die table.
- **EDIT:** The user can edit data of the active record.
- **POST:** Use this with the INSERT button to add a record in the open space. When it is used with EDIT the suggested changes will be made to the table.
- **CANCEL:** Cancel the suggested changes.
- **ABORT:** The update of the post will not take place until the mistake is corrected.
- **DELETE:** Delete the active record in the table.
- **BOF:** Test if the current position in the table is indeed the beginning of the table.
- **EOF:** Test if the current position in the table is indeed the end of the table.



Database

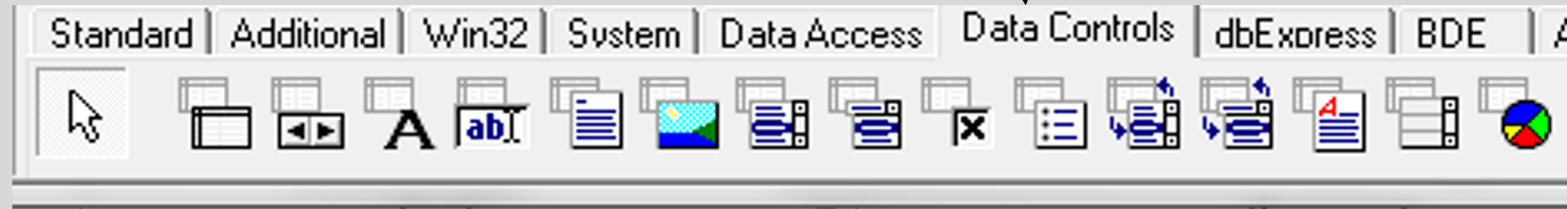
- In this section we will make use of the LM Cruises database.
- The following is an extract of the table that we will work with.

Passenger & Cabin Details								
Passenger Name	Passenger Address	Phone	Sail Date	Cabi	Slee	Deck	Paid?	
René Eitzen	56 3rd Av, Linden	011-673 5783	2008/03/03	A105	6	Atlantic	n	
Chantal Prinsloo	21 Letitia St, Ridge	012-464 2795	2008/01/14	A148	6	Atlantic	y	
Dominique Prinsloo	21 Letitia St, Ridge	012-464 2795	2008/01/14	A148	6	Atlantic	y	
Nathalie Prinsloo	21 Letitia St, Ridge	012-464 2795	2008/01/14	A148	6	Atlantic	y	
Cherice Anne Christi	114 Matie Av, Lone H	013-971 6263	2008/01/14	C104	2	Caribbean	y	
Lebogang Selebogo	1 Denton Place, Melr	031-326 3095	2007/12/08	C160	2	Caribbean	n	
Gladys Xaba	271 Republic Rd, Win	031-781 3565	2007/11/17	C166	2	Caribbean	y	
Lynray Jefferson Bar	367b Main St, Durban	031-828 9941	2007/12/23	C69	2	Caribbean	n	
Waleed Asad	98a Sandton Dr, Sand	011-486 5517	2007/12/08	L12	2	Lounge	y	
Christopher David Ma	29 Coertze St, Vorna	011-576 1588	2007/12/23	L133	2	Lounge	y	

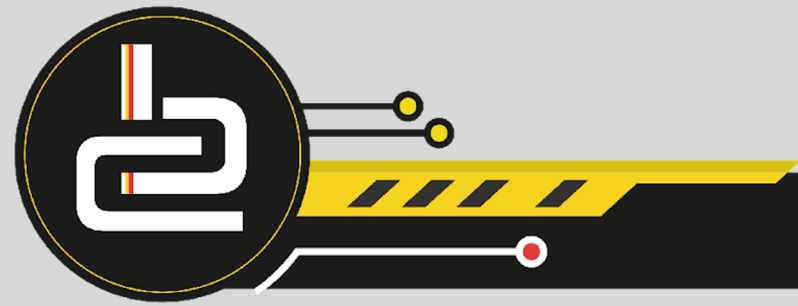


Components for the Manipulation of Data

- These components can be found on the Data Controls page.



- Lets look at all these components.



Components for the Manipulation of Data

We already explained the working of the following components in the previous module:

- dbGrid



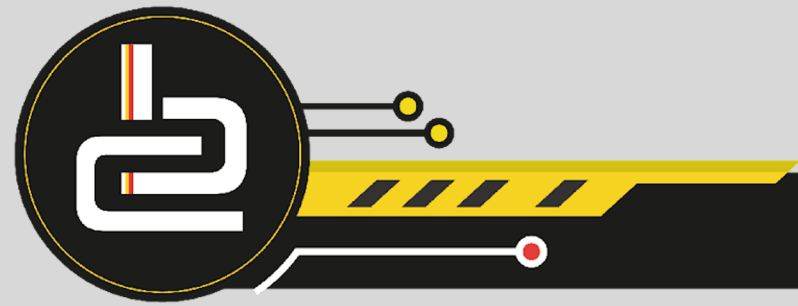
- dbNavigator



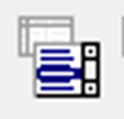




- dbText

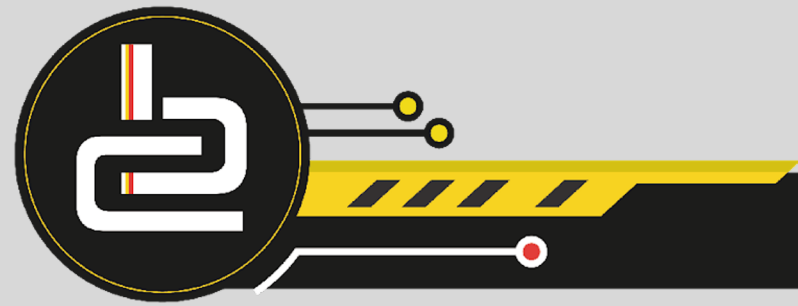


- dbEdit



Components for the Manipulation of Data

- dbListBox  displays a list of the table
- dbComboBox  Same as the dbListBox, but it will let you make changes to the data.
- dbCheckBox  This will display the current Boolean data field and allow you to change it.
- dbLookupListBox  It shows a limited list of valid choices from another table.
- dbLookupComboBox  Here is a list of valid values from another table

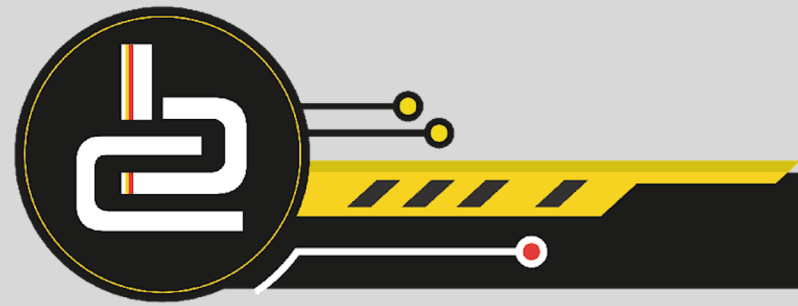


Processing of Data

- It is very important to remember the following:
- First, you need to open the table – `tblBusiness.Open`
- Put the pointer in the first position – `tblBusiness.First`
- Use a loop structure to go through the data – `While Not tblBusiness.EOF`
- Read the data from the table and do the necessary changes.
- Move to the next record – `tblBusiness.Next`

Example: Processing of data in the table

```
procedure TForm1.btnPaidClick(Sender: TObject);
var iCountPaid : Integer;
begin
    iCountPaid := 0;
    tblPassenger.Open;
    tblPassenger.First;
    while not tblPassenger.Eof do
    begin
        If  tblPassenger[ 'Paid?' ] = True then Inc(iCountPaid);
        tblPassenger.Next;
    end;
    redOutput.Lines.Add('LM Cruises');
    redOutput.Lines.Add('');
    redOutput.Lines.Add('The number of passengers who paid ' +
                        IntToStr(iCountPaid));
end;
```

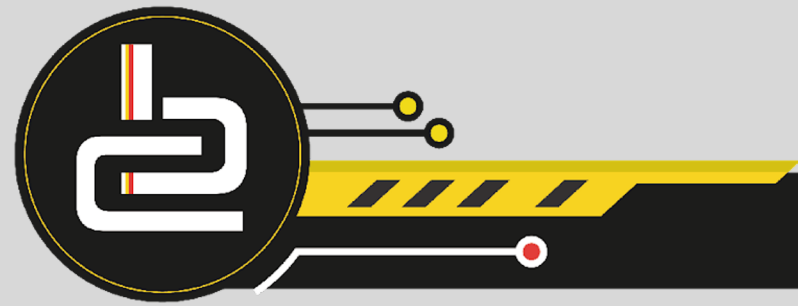


Make Use of Variable to get Data

At any time you can get data from the table and put it in a variable to use it.

There are several ways to do this, you can decide which one you want to use:

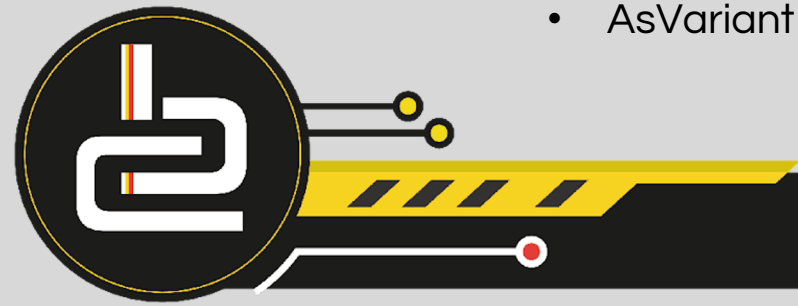
- `sName := tblBusiness.FieldName('Name').AsString;` or
- `sName := tblBusiness.FieldValues['Name'];` or
- `sName := tblBusiness['Name'];`



Make Use of Variables to get Data

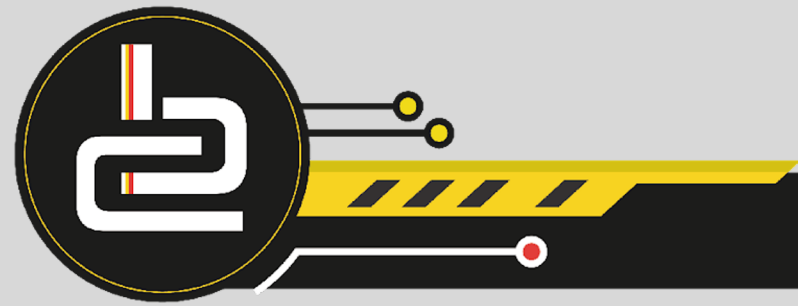
Once you accessed the values within the field obtained through the EDIT feature (method to update data in the table), you can use the AS-feature to work with the data type

- AsBoolean
- AsDateTime
- AsFloat
- AsInteger
- AsString
- AsVariant



Add a record

- When we want to add data in the table, there are a variety of methods you can use.
- We will look at three different methods to add data to the table.
- Note: The data to be inserted can also be done through edits etc...
- In the following slides we will see how to add **Pieter Nel** in the table.

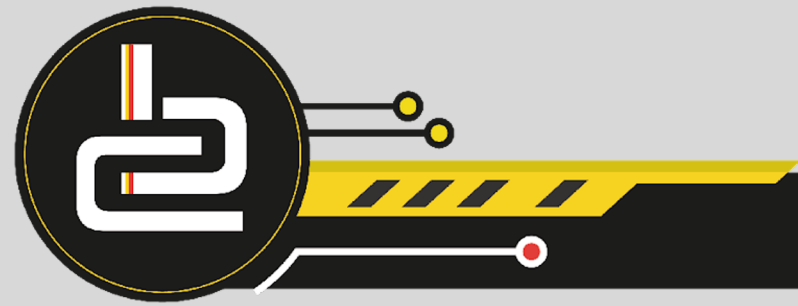


Add a record– method 1

Use a WITH statement:

```
With tblBusiness do  
begin  
Insert;  
        FieldByName('Name').Value := 'Pieter';  
        FieldByName('Surname').Value := 'Nel';  
  
Post;  
end; // with statement
```

NOTE: Order of fields in INSERT matter!



Add a record– method 2 and 3

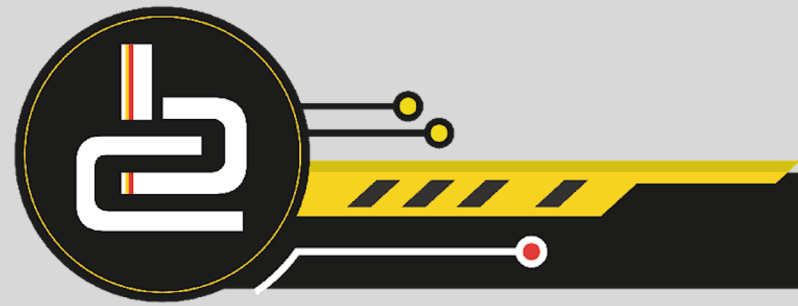
Use the InsertRecord statement:

```
tblBusiness.InsertRecord('Pieter', 'Nel');
```

NB: You need to add all the fields in the correct order

Read directly into the field:

```
tblBusiness.Insert;  
    tblBusiness['Name'] := 'Pieter';  
    tblBusiness['Surname'] := 'Nel';  
tblBusiness.Post;
```

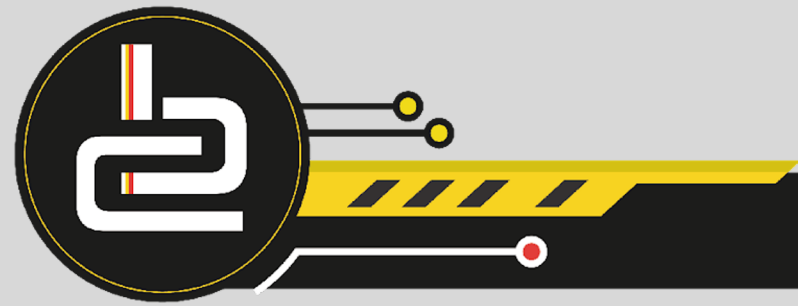


Delete a Record From the Table

- The DELETE method will delete the record where the file marker is at the moment and then move the file cursor to the next record.
- It is therefore very important to make sure that the correct record will be deleted.
It is therefore wise to use the Locate statement with the DELETE.
- Use a dialog box just to make sure that the correct record will be deleted.

Example of how to use the dialog box:

```
If MessageDlg('Are you sure you want to delete the name' + tblBusiness['Name'] + '?', mtWarning, [mbOK, mbCancel], 0) = mrOK then  
    tblBusiness.Delete;
```

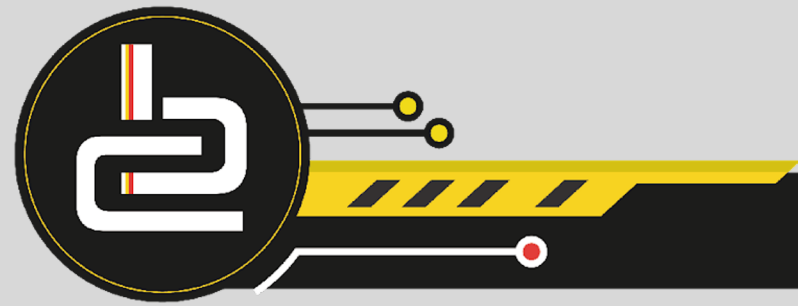


Remove More than One Record at a Time

- If you have more than one record that you want to remove you can use the following method:
In the following code all the passengers who have not paid will be removed from the table.

```
tblBusiness.Open;  
tblBusiness.First;
```

```
While Not tblBusiness.EoF do  
Begin  
    if tblBusiness['Paid'] = 'n' then  
        tblBusiness.Delete  
    else  
        tblBusiness.Next;  
End;
```

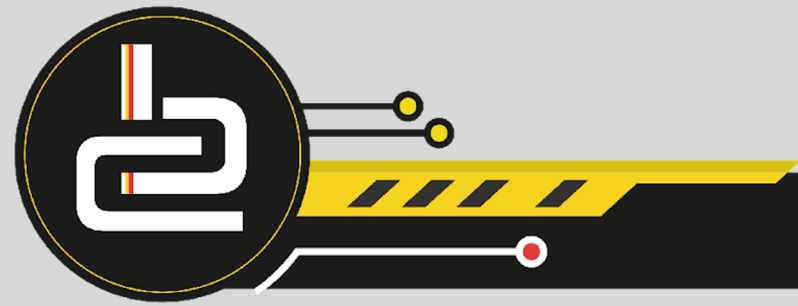


Filter Records According to Certain Criteria

When you use this statement only the data that meet the criteria will show in the DBGrid.

It is very important to take into account that all future calculations will be done on the filtered data, **unless** you remove the filter.

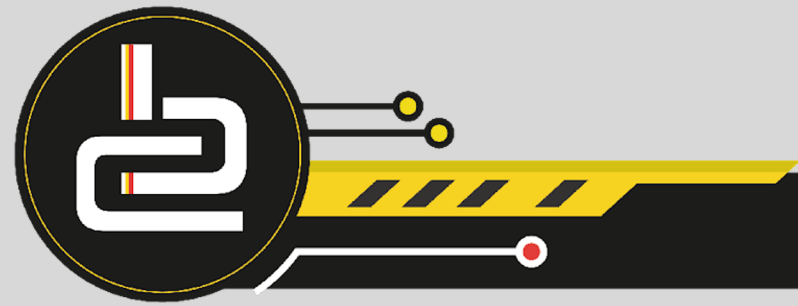
```
tblPassenger.Filter := 'Deck = "Caribbean";  
tblPassenger.filtered := True;
```



Filter Records According to Certain Criteria

There are different ways to do the same thing:

- `tblPassenger.Filter := 'Deck = '+QuotedStr('Caribbean');`
- `tblPassenger.Filter := 'Deck = "Caribbean"';`
- `tblPassenger.Filter := 'Deck = '+ QuotedStr(Edit1.text);`
- `tblPassenger.Filter := 'Deck = '+ QuotedStr(sName);`



Count the Number of Records in a Table

You can count all the records in a table using the following code.

This statement will return an integer value.

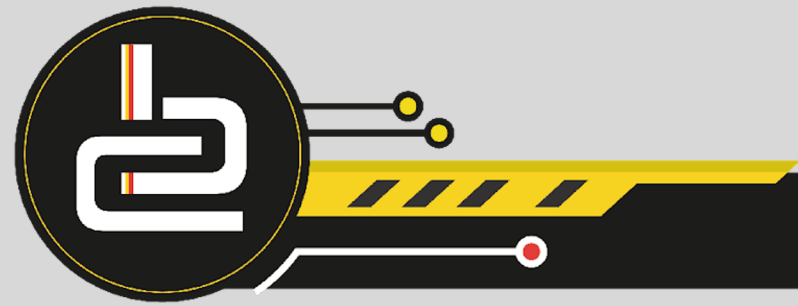
- `iCount := tblBusiness.RecordCount;`

If you want to output this, you will do it as follows:

- `lblOutput.Caption := 'The number of record in the table is: ' + IntToStr(iCount);`

OR

- `lblOutput.Caption := 'The number of record in the table is: ' + IntToStr(tblBusiness.RecordCount);`



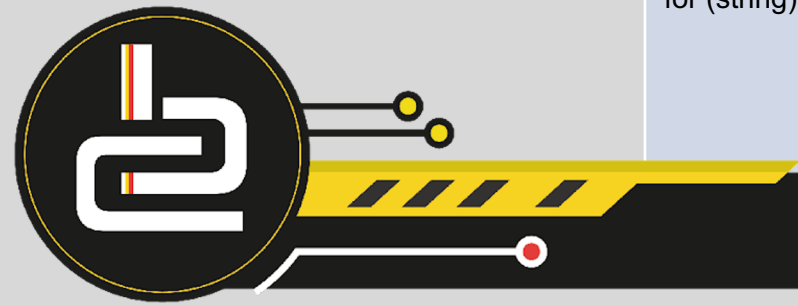
Search for a Specific Record in the Table

```
If tblBusiness.Locate('Name','Pieter',[ ]) then
    ShowMessage('The passenger is on deck' + tblBusiness['Deck']
else
    ShowMessage('No such person');
```

- This code is only for your PAT and will not be asked in the examination.

If you want to use the locate, the following will be very helpful:

First parameter	Second parameter	Third parameter options
The field you are searching for (string)	The value of the field (string)	[loCaseInsensitive,loPartialKey] loCaseInsensitive -> ignore lower&uppercase LoPartialKey -> Search for files that begin with the same value

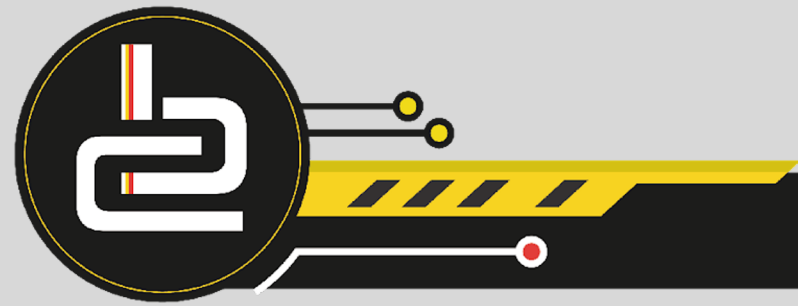


Change Data Within the Table

We use the EDIT and the POST statements to change the data in the table.

```
tblBusiness.Open;  
tblBusiness.First;
```

```
While NOT tblBusiness.EoF do  
Begin  
  If tblBusiness ['Deck'] = 'Caribbean'  
  then begin  
    tblBusiness.Edit;  
    tblBusiness ['Deck'] := 'Lounge';  
    tblBusiness.Post;  
    end;  
  tblBusiness.Next;  
End;
```



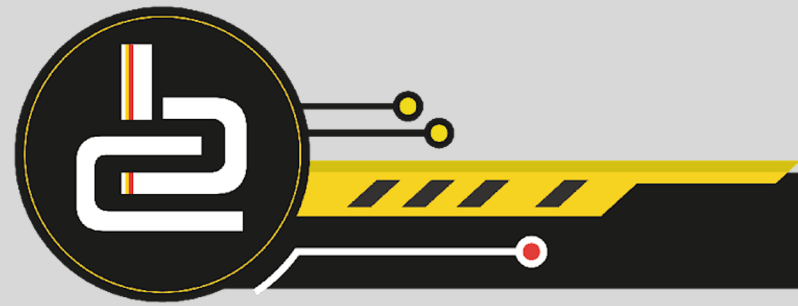
Sort the Records

It is possible to sort the data according to a field:

- By default the data will be displayed in ascending order.
`tblBusiness.Sort := 'Name';`
- If you want the data to appear in descending order
`tbl.Business.Sort := 'Name desc';`

It is also possible to use more than one field during sorting

- `tblBusiness.Sort := 'Name asc, Surname desc';`



Reading Data from a Text File

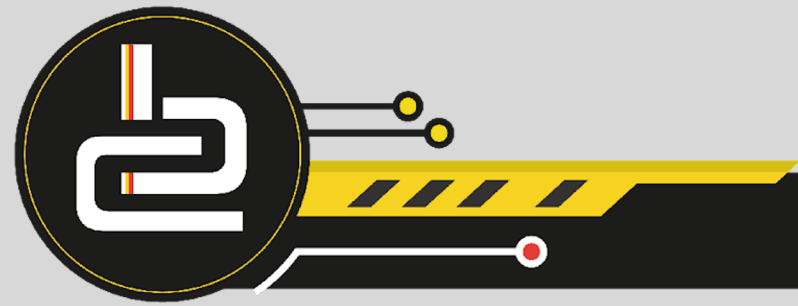
Data can be inserted in a database via a text file by using the following code:

```
If fileExists('Data.txt') = False then
begin
  ShowMessage('File does not exist');
  Application.Terminate;
end;

AssignFile(MyFile, 'Data.txt');
Reset(MyFile);
tblInfo.Open;
tblInfo.Last;

While Not Eof(MyFile) do
begin
  Readln(MyFile, sNumber);
  Readln(MyFile, sName);
  Readln(MyFile, sSurname);
  Readln(MyFile, sMark);

  tblInfo.Insert;
  tblInfo['StudentNumber'] := sNumber;
  tblInfo['Name'] := sName;
  tblInfo['Surname'] := sSurname;
  tblInfo['TestMark'] := IntToStr(sMark);
  tblInfo.Post;
end; //while
```

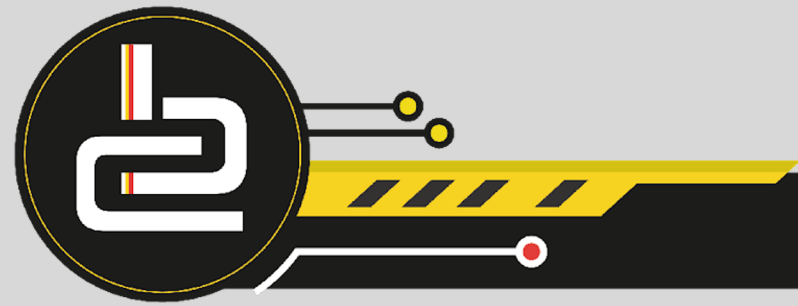


Write Data to a Text File

Data can be written from the database to a text file using the following code:

```
AssignFile(MyFile, 'Data.txt');
Append(MyFile);
tblInfo.Open;
tblInfo.First;

while Not tblInfo.Eof do
begin
    Writeln(MyFile, tblInfo.FieldName('Studentnr').AsString);
    Writeln(MyFile, tblInfo.FieldName('Name').AsString);
    Writeln(MyFile, tblInfo.FieldName('Surname').AsString);
    Writeln(MyFile, tblInfo.FieldName('Marks').AsString);
    tblInfo.Next;
end; //while
CloseFile(MyFile);
```



Writing Data Between Database and Text File

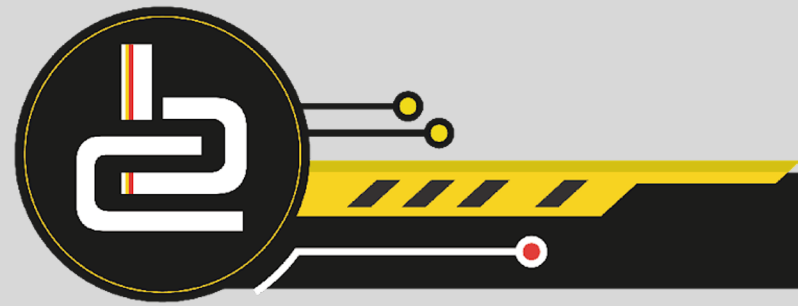
Look at the two previous slides, where we wrote data between the database and the text file.

NOTE:

In the first slide we made use of variables.

In the second slide we made use of the FieldByName feature.

Both methods work for every case, you can decide which one is the easiest for you and make use of it.



Validation

Validation of data is probably one of the most important parts of programming and should be part of all your programs.

There are several methods you can use to validate data:

- One of the methods you are probably already familiar with the TRY / EXCEPT
- You can also use an IF in conjunction with the abort statement.
- Make sure you use the correct components, for example, use a dbListBox where limited choices are available or a dbCheckBox where only Yes / No should be selected.
- Make sure you ALWAYS have a message with validation, so it can help the user to fix the error.

