

## Criterion E: Product development

### Complex techniques used:

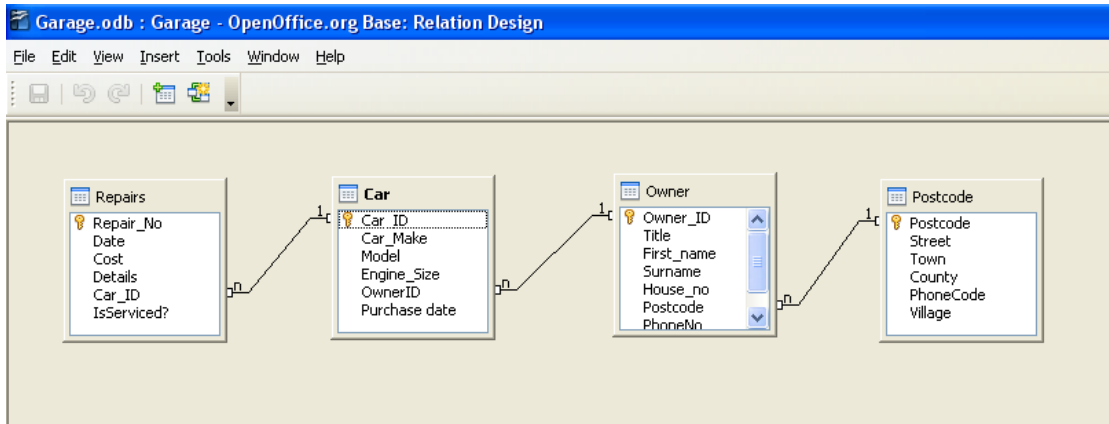
- 4 related tables – pages 2 -4
- Complex queries – pages 5 - 7
- Sub-forms – page 8
- Use of macros -pages 9 - 10

### Simple techniques used:

- Basic queries using related tables
- Forms using tables and queries
- Reports linked to the repairs table

## Organisation of the database (Document object model)

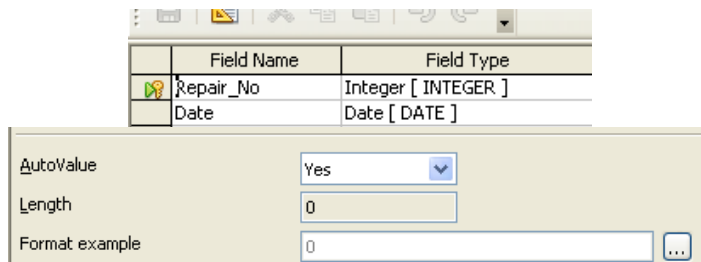
The screenshot below indicates how the database is organized:



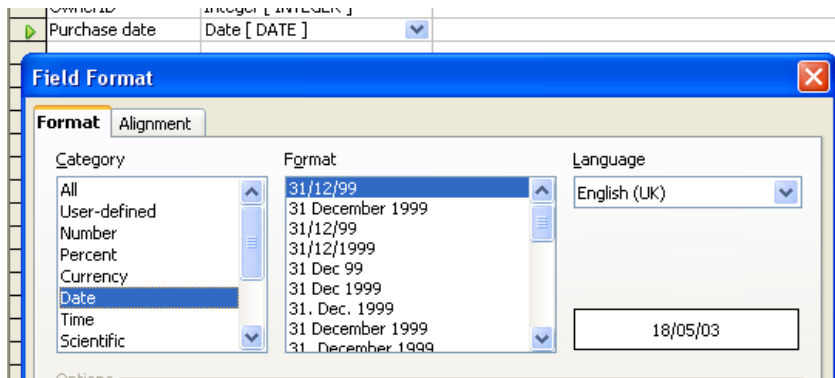
The data has been added to the tables and sensible file names have been used. The fields within the tables have been changed from the default settings to make them fit the type of data that is going to be added.

Below are some examples:

- The repair number in the repairs table is an autonumber so that as each new repair job is done it automatically creates a new record, see screenshot below



- The purchase date field in the car table has been changed to a UK style as the garage is in the UK



- The title field in the owners table has been reduced to 10 characters as 50 are not needed.

	Field Name	Field Type
🔑	Owner_ID	Integer [ INTEGER ]
▶	Title	Text [ VARCHAR ]
	First name	Text [ VARCHAR ]

Field Properties	
Entry required	No
Length	10
Default value	
Format example	@

- A default value has been added to the area phone code in the postcode table because most customers have a 01600 area code. This saves you having to type in this value to most records

	Field Name	Field Type
🔑	Postcode	Text [ VARCHAR ]
	Street	Text [ VARCHAR ]
	Town	Text [ VARCHAR ]
	County	Text [ VARCHAR ]
▶	PhoneCode	Text [ VARCHAR ]
	Village	Text [ VARCHAR ]

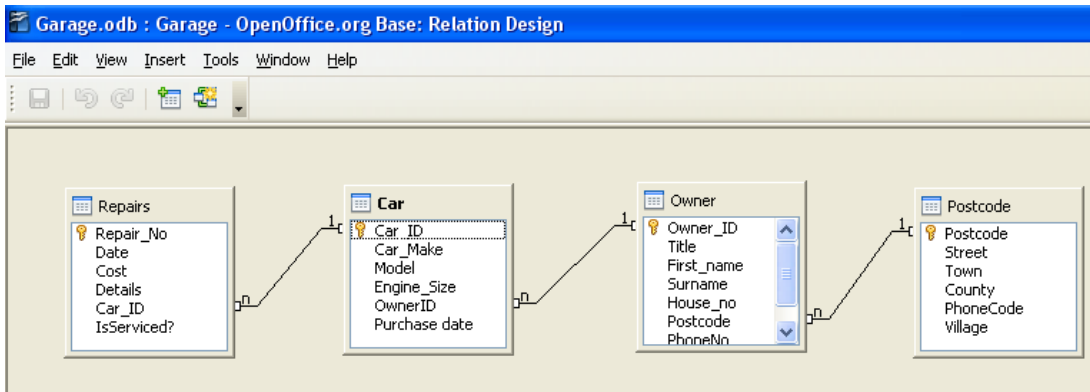
Field Properties	
Entry required	No
Length	5
Default value	01600
Format example	01600

The field type has also been changed to text so that the first zero in the area code does not get removed by the program thinking it is a number.

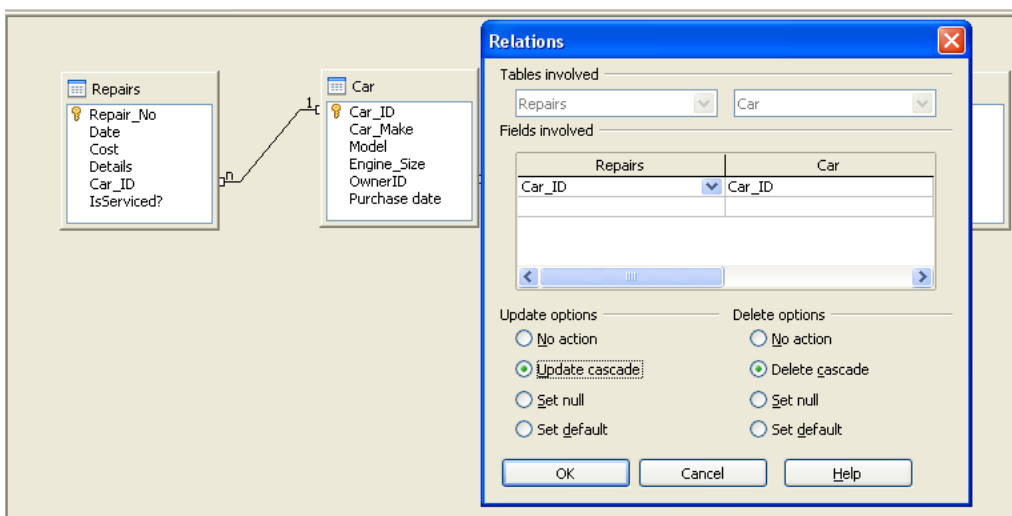
## The structure and organisation of the database

### Technique: related tables

The database consists of four related tables joined by one-to-many relationships. This is shown below:



The tables are linked and the updates are allowed to cascade through the database. This means that if a change is made to the data in one table it will automatically update the value in the linked table. This is shown in the screenshot below:



The use of related tables also means that data only has to be entered once. For example if a new customer has an address in the same postcode as an existing one, Steve only needs to enter their house number, postcode and phone number. The rest is added automatically, see the new owner form.

This means that there will not be either redundant or duplicated data within the database which may cause Steve problems in deciding where to send information.

**Technique: complex queries**

Sometimes Steve needs to know what types of cars he has when ordering spare parts as he only has a small garage and cannot keep lots of spares that he isn't using, he calls it a just-when-I-need-it system. Steve wanted a way of finding out if he had lots of the same type of car, such as Fords which would let him order a part like an air filter or headlight bulb that could be used in more than one car.

To do this the following queries were set up:

**Select\_car\_make**

The screenshot shows the 'Query Design' window in OpenOffice.org Base. The title bar reads 'Garage.odb : Search for car make - OpenOffice.org Base: Query Design'. The window contains three tables: 'Car', 'Owner', and 'Postcode'. Lines connect the 'Owner\_ID' field in the 'Car' table to the 'Owner\_ID' field in the 'Owner' table, and the 'Postcode' field in the 'Owner' table to the 'Postcode' field in the 'Postcode' table. Below the tables are two data tables for the query design. The first table shows fields from the 'Owner' table: Title, First\_name, Surname, House\_no, Street, Village, Town, and Postcode. The second table shows fields from the 'Postcode' table: Postcode, PhoneCode, and PhoneNo, and fields from the 'Car' table: Car\_ID, Car\_Make, and Model. The 'Criterion' row in the second table contains ':Select\_car\_make'. The 'Visible' row has checkboxes for each field, with the last one (Model) unchecked.

Field	Title	First_name	Surname	House_no	Street	Village	Town	Postcode
Alias	Title	First_name	Surname	House_no	Street	Village	Town	Postcode
Table	Owner	Owner	Owner	Owner	Postcode	Postcode	Postcode	Postcode
Sort		ascending	ascending					
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Field	Postcode	PhoneCode	PhoneNo	Car_ID	Car_Make	Model	
Alias	Postcode	PhoneCode	PhoneNo	Car_ID	Car_Make	Model	
Table	Postcode	Postcode	Owner	Car	Car	Car	
Sort							
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Function							
Criterion					:Select_car_make		
Or							

Steve is able to type in the car make when he opens the select\_car\_make form, see below.

The screenshot shows a 'Parameter Input' dialog box. The title bar reads 'Parameter Input'. The 'Parameters' list contains 'Select\_car\_make'. There are 'OK' and 'Cancel' buttons. Below the list is a 'Value' field which is currently empty, and a 'Next' button.

Which when he types in Ford he gets these results.

	Title	First_name	Surname	House_no	Street	Village	County	Postcode	PhoneCode	PhoneNo	Car_ID	Car_Make	Model
▶	Mr	Jonathan	Powell	Coppet Hill Vie	Penraig R	Goodrich		HR9 7ED	01600	892334	WM57YYY	Ford	Mondeo
	Mr	Jonathan	Powell	Coppet Hill Vie	Penraig R	Goodrich		HR9 7ED	01600	892334	SC51TCV	Ford	Focus
	Ms	Margaret	McDowell	34	LLangarror	Llangrove		HR9 6EF	01600	893212	Y567EDF	Ford	Ka
	Mrs	Pauline	Powell	9	Osbaston I			NP25 3RT	01600	712333	CF57UYH	Ford	Focus
	Mr	Richard	Taylor	Doward View	Llangrove	Whitchurcl		HR9 6DQ	01600	891045	RJ51ODK	Ford	Ka
	Mr	Richard	Taylor	Doward View	Llangrove	Whitchurcl		HR9 6DQ	01600	891045	BJ55CLC	Ford	Focus
	Mr	Terry	Powell	42	Osbaston I			NP25 3RT	01600	713212	WM58YUT	Ford	Mondeo

**Select\_car\_Make\_and\_type**

Steve also wants to find out how many cars of a certain type he has. In this case he is looking for the number of Ford Kas he has repaired.

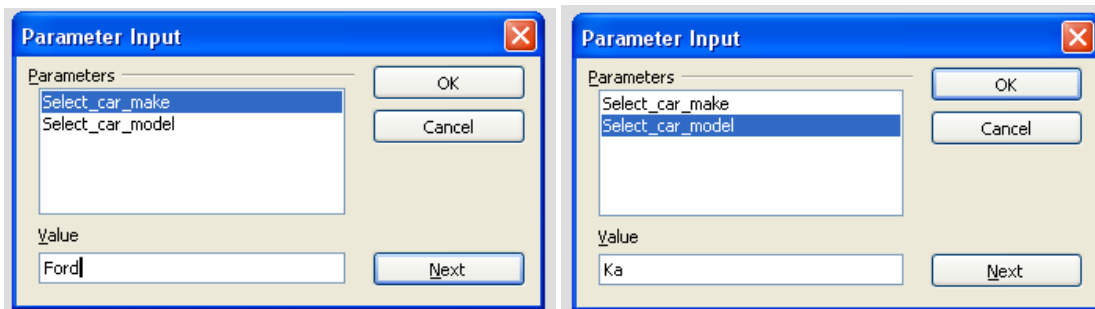
This uses the ‘Search for car make and model’ query which is used when the ‘Search for car make and model’ form is opened.

The query is exactly the same as the search for car make query except the value for a second field has to be added. See below.

r_ID	Car_Make	Model
r_ID	Car_Make	Model
r	Car	Car
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	:Select_car_make	:Select_car_mode

This is an AND query as it requires both the Car make and the Car Model to equal the required value. This is because the criteria requirements are on the same line of the query (See SQL version of query).

When the ‘Search for car make and model’ form is opened Steve has to add the car make and car model.



	Title	First_name	Surname	House_no	Street	Village	Town	Postcode	PhoneCode	PhoneNo	Car_ID	Car_Make	Mod
▶	Ms	Margaret	McDowell	34	LLangarror	Llangrove	Ross-on- <del>V</del>	HR9 6EF	01600	893212	Y567EDF	Ford	Ka
	Mr	Richard	Taylor	Doward View	Llangrove	Whitchurcl	Ross-on- <del>V</del>	HR9 6DQ	01600	891045	RJ51ODK	Ford	Ka

This means Steve can quickly check to see how many Ford Kas he has, and then look up their repair records rather than look through the whole of the database. He can also change the types of car and the models to make comparisons between different makes and models which may affect whether he decides to take on that work .

The SQL version of the 'Search for car make and model' query is below.

Table name      Choice of fields

```
SELECT "Owner"."Title" AS "Title", "Owner"."First_name" AS "First_name",
"Owner"."Surname" AS "Surname", "Owner"."House_no" AS "House_no",
"Postcode"."Street" AS "Street", "Postcode"."Village" AS "Village", "Postcode"."Town" AS
"Town", "Postcode"."Postcode" AS "Postcode", "Postcode"."PhoneCode" AS "PhoneCode",
"Owner"."PhoneNo" AS "PhoneNo", "Car"."Car_ID" AS "Car_ID", "Car"."Car_Make" AS
"Car_Make", "Car"."Model" AS "Model" FROM "Owner" AS "Owner", "Postcode" AS
"Postcode", "Car" AS "Car"
```

Joins between tables

```
WHERE "Owner"."Postcode" = "Postcode"."Postcode" AND "Car"."OwnerID" =
"Owner"."Owner_ID"
```

Requires make and model of car, both conditions must be met

```
AND "Car"."Car_Make" = :Select_car_make AND "Car"."Model" = :Select_car_model
```

Sorts the query into alphabetical order

```
ORDER BY "First_name" ASC, "Surname" ASC
```

The joins between the tables are inner joins which means that only records where the Postcode value in the Owner table is equal to the Postcode value in the Postcode table and the OwnerID value in the Car table is equal to the OwnerID value in the Owner table, where the car make is a Ford and the car model is a Ka, is 2.

If this was not the case the number of records would be equal to the number of postcodes multiplied by the number of owners multiplied by the number of cars. This is 13 for the same conditions.

**Technique: sub-forms**

The use of a sub-form allows Steve to look at the details of a car that has been to his garage more than once. For example, if he is due to have a particular car in for a service he can look to see what other work he has done and order parts, such as brake pads, in advance as he knows the job will need to be done. He can also enter the information about repairs he has carried out and see if there is a pattern, for example if a car appears to have a fault that occurs on a more regular basis than expected.

In the example below Steve can see that the Ford Focus BJ55CUC comes in every 6 months for a service (each service is required after 12000 miles), so he knows he can order a new set of front brake pads when it comes in for its 72000 mile service and also check the state of the front tyres.

**Car repair information**

Car\_ID

Car\_Make  Model

Engine\_Size

	Date	Cost	Details
▶	03/06/08	103.99	48000 mile service and new front brake pads, 4 new tyres
◀	04/12/08	112.78	60000 mile service
⚙			

Record  of  ⏪ ⏩ ⏴ ⏵ ⚙

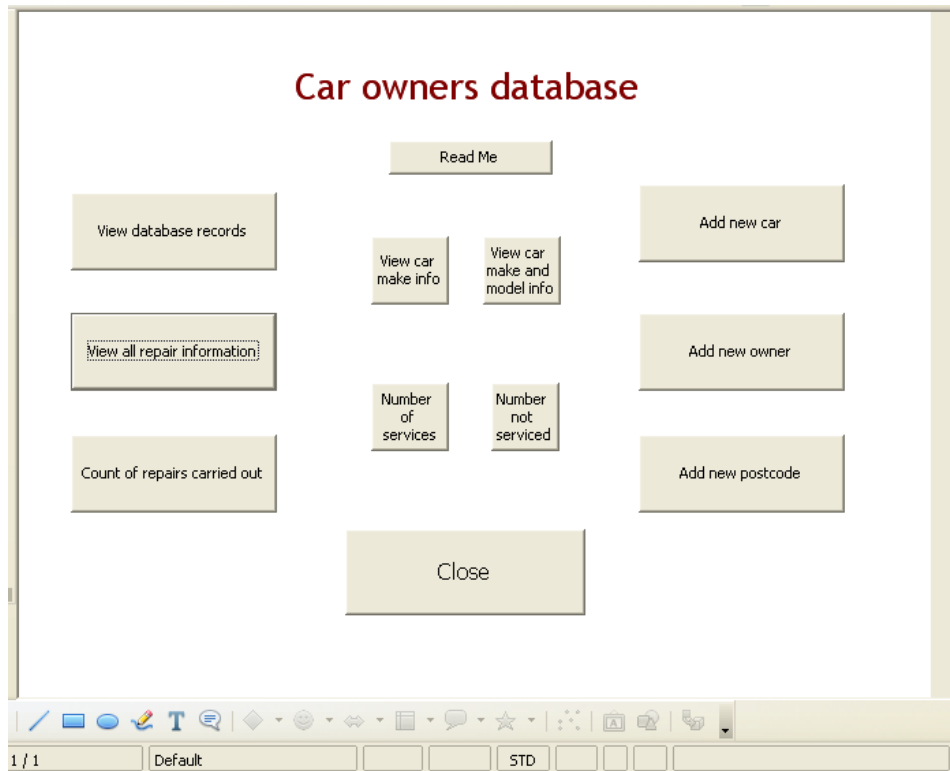
Record  of 11 \* ⏪ ⏩ ⏴ ⏵ ⚙

The form also shows that the Focus is one of 11 cars in Steve's database.



### Technique: macros

Macros have been used in this database to link the different forms rather than expect Steve to go through the menu. The main switchboard is the Welcome form that gives access to the other forms by clicking on a button.



When each button is clicked a macro assigned to it runs that opens the correct form. This makes it easier for Steve to move around the database than using the main menu. A second button on the form allows Steve to close the form and return to the Welcome Form which does not close when the other form was opened. Unfortunately in testing the clicking of the close button caused the computer to 'hang', so the macro was disabled.

The macro is based on snippets of code taken from the following open office help sites.

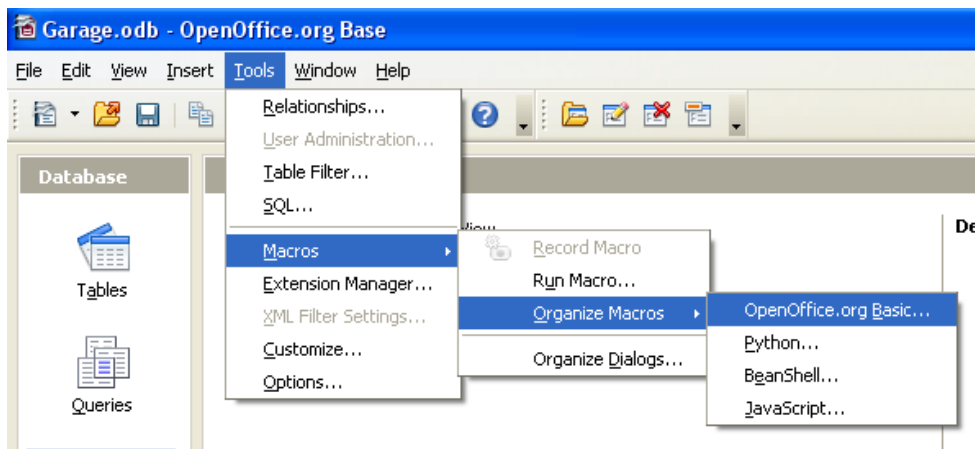
"Problem with taking input from users for queries." [The OpenOffice.org Forum](http://www.ofoforum.org/forum/viewtopic.php?t=53530). Feb 23, 2007. OpenOffice.org. 23 Dec 2008 <<http://www.ofoforum.org/forum/viewtopic.php?t=53530>>.

"Macro for button to open a Form in Base." [The OpenOffice.org Forum](http://www.ofoforum.org/forum/viewtopic.php?t=63587). Oct 02, 2007. OpenOffice.org. 24 Dec 2008 <<http://www.ofoforum.org/forum/viewtopic.php?t=63587>>.

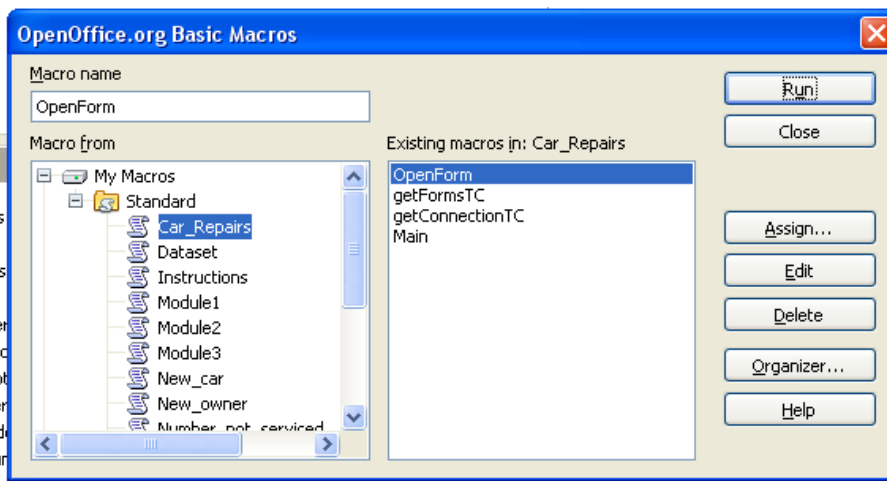
"Close form macro..." [The OpenOffice.org Forum](http://www.ofoforum.org/forum/viewtopic.php?t=60613). Aug 08, 2007. OpenOffice.org. 24 Dec 2008 <<http://www.ofoforum.org/forum/viewtopic.php?t=60613>>.

"Open form with a button form another form." [The OpenOffice.org Forum](http://www.ofoforum.org/forum/viewtopic.php?t=37999). Jun 14, 2006. OpenOffice.org. 28 Dec 2008 <<http://www.ofoforum.org/forum/viewtopic.php?t=37999>>.

This code was then pasted into the modules created in the macro organizer, see screenshot.



Once pasted into the required module the generic text from the website is edited, usually by changing the name of the form, and attached to a button on the form.



```
sub Main ( oev as variant )
```

```
sFormName = "Car_repairs" This text is edited to change which form opens otherwise no changes are needed
```

```
OpenForm( getFormsTC, getConnectionTC, sFormName )
```