INTRODUCTION

Implementation of the operational application database for the car rental chain. Satisfied relationships between stakeholders were given as data in the first task. These are:

- For each branch of the chain we keep its details.
- The clients of each branch are divided into 3 categories:
  - foreign
  - organizations (companies),
  - individuals, (ie not companies), where for each customer we pass certain elements.
- For each branch we pass respectively the details of the employees who work in it.
- For each branch we pass the details of the cars that are for rent.
- Each customer is connected to a vehicle, an employee and a branch.
- Each employee is connected to a vehicle, a customer and a branch.
- The customer chooses a date to rent a vehicle and the vehicle to rent as well as the time period to rent.

Based on these specifications we were given a solution to the work that satisfies this base system.

We used Microsoft SQL Management Studio to implement the database. Through this tool and utilizing our knowledge of the SQL language we created our database, ie the tables with the relationships that surround them. The choice of this program was made solely on the grounds of facilitating the design of our database. In addition to the base, we had to create a User Interface in order to facilitate the user to utilize our application. We used Microsoft Access to create our application. So, we created menus, through which the user can operate the database without knowing any programming language. For this purpose, the home page gives the options Insert, Update, Delete, DB Info, Views and Queries. In addition, it has been possible through various dropdown lists to choose exactly each time which action we want to perform, thus making the application easier for the user. We also provide for cases where something will go wrong, right or wrong, but which can give wrong results on our basis. Therefore, we ensure that we do not have negative values in dates, PCs, but also do not leave gaps in certain fields which are considered necessary for the correct completion of each set and therefore of the tables of our database.

Base Design

Limitations:
For the correct design of the database and its proper operation we had to put some restrictions regarding either what the user enters, or the values that the fields can get in the forms that we fill in our application.

Entity Integrity : Where each record must be uniquely identified by an Id key.

Keys : We have defined integers everywhere except for the losses that a compound key has.

Referential integrity : It is forbidden to delete a record that refers to FK in another record because it will be orphaned.
Here now to delete a registration such as the customer must first delete all his rentals.

Price domain integrity: Here we specify prices and a specific format of the car to be car or minivan or ATV.

Columns: It is the validity of the data entries in a specific column.
We can enhance it by restricting the formula using data type, restricting the format using control constraints and rules, or limiting the range of possible values using FOREIGN KEY constraints, control constraints, DEFAULT constraints, NOT NULL definitions and rules.
user defined:

For example you must endate> startdate, or in order to rent the payment must have been made
Still in the rentals I will pass 3 restrictions

1. The payment
2. The car must be available
3. Its kilometers must not have exceeded kilometers of service
4. The enddate should not exceed the enddate of security

Indexes:
For each relationship we have a B-Tree index in the Primary Key, which can be a Foreign Key that refers to another relationship, to be used as a Foreign Key by another relationship or combination of them.
The default questions we have provided in our database are as follows:
Insert: The user can insert a bunch into one of the defined tables. The base takes care of creating the corresponding array in any other array needed.
Update: The user can update its database, change some of the data in the tables without unnecessary moves.
Delete: The user can delete some blocks from one of the tables that he is editing. The base will again take care to fill the gap created and delete the corresponding connections.
DB Info: This option gives the user access to the database data, that is, the tables with all the data for each table, that is, each relation.

3)

Views: This is a way of viewing the tables we choose. Here we have Updateable and Non-Updateable view.

Updateable View:

```
SELECT         B.BranchName, SUM (CASE WHEN C.Type = 'ATV' THEN 1 ELSE 0 END)
     AS ATVs, SUM (CASE WHEN C.Type = 'Car' THEN 1 ELSE 0 END) AS Cars,
                SUM (CASE WHEN C.Type = 'Mini Van' THEN 1 ELSE 0 END) AS [Mini
Vans], SUM (CASE WHEN C.Type = 'Motocycle' THEN 1 ELSE 0 END) AS Motorcycles,
                SUM (CASE WHEN C.Type = 'Truck' THEN 1 ELSE 0 END) AS Trucks
               FROM            dbo.tbl_Branches AS B LEFT OUTER JOIN
                                dbo.tbl_Cars AS C ON C.BranchID = B.ID
                               GROUP BY B.BranchName
                                        '
                                        '
                                        '
```

```
         SELECT         dbo.tbl_Reservations.ID, dbo.tbl_Reservations.CarID,
     dbo.tbl_Reservations.ClientID, dbo.tbl_Reservates.Resbations. StartDate,
                 dbo.tbl_Reservations.EndDate, dbo.tbl_Reservations.Payment
             FROM            dbo.tbl_Reservations LEFT OUTER JOIN
                dbo.tbl_Rentals ON dbo.tbl_Reservations.ID = dbo.tbl_Rentals.ID
                    WHERE         (dbo.tbl_Rentals.ID IS NULL)

                                        '
                                        '
                                        '
```

```
          SELECT         LISTA_PELATON.ID, COUNT (*) [NumberOfRentals]
              FROM            (SELECT        A. AS Name, B.AFM
                   FROM           DBO.tbl_Individuals AS A INNER JOIN
                                     DBO.tbl_Clients AS B ON A.ID = B.ID
                                        EXCEPT
                    SELECT        A.ID, A.LastName AS Name, B.AFM
                   FROM           DBO.tbl_Individuals AS A INNER JOIN
```

```
DBO.tbl_Clients AS B ON A.ID = B.ID INNER JOIN
DBO.tbl_Organizations AS C ON A.ID = C.ID) AS
LISTA_PELATON INNER JOIN
dbo.tbl_Reservations AS RES ON RES.ClientID = LISTA_PELATON.ID
INNER JOIN
DBO.tbl_Rentals AS REN ON RES.ID = REN.ID
GROUP BY LISTA_PELATON.ID
```

4) In addition, in the design of our database, we were asked to provide for 2 triggers of our choice. This means that when a specific event occurs, then they are triggered and protect our data and the proper functioning of our database. These are explained below:

Trigger1:

This trigger checks the reservations table every time a reservation is made, as soon as the car is given to the customer puts 0 in the flag of the booked car. the date on which this event occurred.

```
CREATE TRIGGER trgCarNotAvailable on DBO.tbl_Rentals FOR INSERT
AS
DECLARE @ResID INT
SELECT @ ResID = I.ID FROM INSERTED I;

UPDATE DBO.tbl_Cars
SET Available = 0
WHERE ID = (SELECT CARID FROM tbl_Reservations WHERE ID = @ ResID)
GO
```

Trigger2:

```
CREATE TRIGGER trgCarAvailable on DBO.tbl_Rentals FOR UPDATE
AS
DECLARE @LesART
INT
D FROM INSERTED I;
SELECT @ Kilometers = I.Kilometers FROM INSERTED I;

UPDATE DBO.tbl_Cars
SET Available = 1,
KILOMETERS = @KILOMETERS
WHERE ID = (SELECT CARID FROM tbl_Reservations WHERE ID = @ ResID)
GO
```

In this trigger the opposite actually happens as soon as the car is returned by the customer he informs the flag by putting the price 1 and at the same time updates the mileage of the vehicle with the mileage done during the rental.

5) Queries: In this option we give the user the ability to ask certain questions at its base. These questions are predefined by the developer and it is about joining tables, finding maximum, average and others. Specifically:
finding client from ID

```
sqltxt = "SELECT C. [ID], C. [EntryDate], C. [AFM], C. [AddressStreet], C. [AddressNumber],
C. [AddressTK], C. [AddressCity ], C. [LandLine], C. [Mobile], C. [Email], "
sqltxt = sqltxt +" I. [ID] AS IND_ID, I. [Occupation], I. [BirthDate], I. [LastName ], I.
[FirstName], I. [FatherName], I. [ADT], "
sqltxt = sqltxt +" O. [ID] AS ORG_ID, O. [Name], O. [Type], O. [EstablishmentDate ], O.
[RegistrationNum] "
sqltxt = sqltxt +" FROM TBL_CLIENTS AS C "
sqltxt = sqltxt +" LEFT OUTER JOIN TBL_INDIVIDUALS AS I ON C.ID = I.ID "
sqltxt = sqltxt +" LEFT OUTER JOIN T C.ID = O.ID "
sqltxt = sqltxt +" WHERE C.ID = "+ CStr (ClientID)
```

delete client

```
sqltxt =" DELETE FROM DBO.TBL_CLIENTS "
sqltxt = sqltxt +" WHERE [ID] = "+ CStr (frmClientID )
```

private input

```
sqltxt = "INSERT INTO [dbo]. [tbl_Clients] ([EntryDate], [AFM], [AddressStreet],
[AddressNumber], [AddressTK], [AddressCity], [LandLine], [Mobile], [Email ]) "
sqltxt = sqltxt +" OUTPUT (INSERTED.ID) "
sqltxt = sqltxt +" VALUES ('"+ Format (Date," YYYY -MM-DD ") +" '"
sqltxt = sqltxt +", N' "+ CStr (Me.txtIndAFM) +" '"
sqltxt = sqltxt +", N' "+ CStr (Me.txtAddress) +" '"
sqltxt = sqltxt + ", N '" + CStr (Me.txtAddressNum) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtTK) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtCity) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtPhoneNum) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtMobNum) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtEmail) + "')"

sqltxt = "INSERT INTO [dbo]. [tbl_Individuals] ([ID], [Occupation], [BirthDate], [LastName],
[FirstName], [FatherName], [ADT]) "
sqltxt = sqltxt +" VALUES ("+ CStr (ClientID)
sqltxt = sqltxt +", N '"+ CStr (Me.txtIndOccupation) +"' "
sqltxt = sqltxt + ", '" + Format (CStr (Me.txtIndBirthDate), "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtIndLastName) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtIndFirstName) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtIndFatherName) + "'
sqltxt = sqltxt +", N '"+ CStr (Me. txtIndADT) + "')"
```

private update

```
sqltxt = "Update [dbo]. [tb l_Clients] "
sqltxt = sqltxt +" SET [AFM] = "+" N '"+ CStr (Me.txtIndAFM) +"' "
sqltxt = sqltxt +", [AddressStreet] = N '"+ CStr (Me.txtAddress) + "'"
sqltxt = sqltxt + ", [AddressNumber] = N'" + CStr (Me.txtAddressNum) + "'"
sqltxt = sqltxt + ", [AddressTK] = N'" + CStr (Me.txtTK) + " '"
sqltxt = sqltxt +", [AddressCity] = N' "+ CStr (Me.txtCity) +" '"
sqltxt = sqltxt +", [LandLine] = N' "+ CStr (Me.txtPhoneNum) +" '"
sqltxt = sqltxt + ", [Mobile] = N '" + CStr (Me.txtMobNum) + "'"
sqltxt = sqltxt + ", [Email] = N '" + CStr (Me.txtEmail) + "'"
sqltxt = sqltxt + "WHERE ID =" + CStr (frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
'rst.Close

sqltxt = sqltxt + "Update [dbo]. [tbl_Individuals]"
sqltxt = sqltxt + "SET [Occupation] = N '" + CStr (Me.txtIndOccupation) + "'"
sqltxt = sqltxt + ", [BirthDate] = '" + Format (CStr (Me.txtIndBirthDate),
"YYYY-MM-DD") + "'
sqltxt = sqltxt +", [LastName] = N' "+ CStr (Me.txtIndLastName) +" '"
sqltxt = sqltxt + ", [FirstName] = N '" + CStr (Me.txtIndFirstName) + "'"
sqltxt = sqltxt + ", [FatherName] = N '" + CStr (Me.txtIndFatherName) + "'"
sqltxt = sqltxt + ", [ADT] = N '"+ CStr (Me.txtIndADT) +"' "
sqltxt = sqltxt +" WHERE ID = "+ CStr (frmClientID)
```

import company

```
sqltxt =" INSERT INTO [dbo]. [Tbl_Clients] ([EntryDate] , [AFM], [AddressStreet],
[AddressNumber], [AddressTK], [AddressCity], [LandLine], [Mobile], [Email]) "
sqltxt = sqltxt +" OUTPUT (INSERTED.ID) "
sqltxt = sqltxt + "VALUES ('" + Format (Date, "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtOrgAFM) + "'"
sqltxt = sqltxt + ", N ' "+ CStr (Me.txtAddress) +" '"
sqltxt = sqltxt +", N' "+ CStr (Me.txtAddressNum) +" '"
sqltxt = sqltxt +", N' "+ CStr (Me.txtTK) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtCity) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtPhoneNum) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtMobNum) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtEmail) + "')"
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
ClientID = rst.Fields.Item (0) .Value
```

```
rst.Close

sqltxt = "INSERT INTO [dbo]. [tbl_Organizations] ([ID], [Name], [Type],
[EstablishmentDate], [RegistrationNum])"
sqltxt = sqltxt + "VALUES (" + CStr (ClientID)
sqltxt = sqltxt + ", N '" + CStr (Me.txtOrgFirmName) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtOrgFirmType) + "'"
sqltxt = sqltxt + ", '" + Format (CStr ( Me.txtOrgFoundedDate), "YYYY-MM-DD") + "'
sqltxt = sqltxt +", N' "+ CStr (Me.txtOrgRegNum) +" ') "
rst.Open sqltxt, con, adOpenKeyset, adLockReadOn

sqltxt = "INSERT INTO [dbo]. [tbl_Individuals] ([ID], [Occupation], [BirthDate],
[LastName], [FirstName], [FatherName], [ADT])"
sqltxt = sqltxt + "VALUES (" + CStr (ClientID)
sqltxt = sqltxt + ", N'ÕĐÁËËÇËÏÓ '"
sqltxt = sqltxt + ",'" + Format (CStr (Me.txtDriverBirthDate), "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtDriverLastName) + "'"
sqltxt = sqltxt + ", N '" + CStr (Me.txtDriverFirstName) + "'
sqltxt = sqltxt +", N '"+ CStr (Me. txtDriverFatherName) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtDriverADT) + "')"
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
```

company update

```
sqltxt = "Update [dbo_C]. "
sqltxt = sqltxt +" SET [AFM] = "+" N '"+ CStr (Me.txtOrgAFM) +"' "
sqltxt = sqltxt +", [AddressStreet] = N '"+ CStr (Me.txtAddress) +" '"
sqltxt = sqltxt +", [AddressNumber] = N' "+ CStr (Me.txtAddressNum) +" '"
sqltxt = sqltxt +", [AddressTK] = N' "+ CStr (Me.txtTK) +" '"
sqltxt = sqltxt + ", [AddressCity] = N '" + CStr (Me.txtCity) + "'"
sqltxt = sqltxt + ", [LandLine] = N '"+ CStr (Me.txtPhoneNum) +"' "
sqltxt = sqltxt +", [Mobile] = N '"+ CStr (Me.txtMobNum) +"' "
sqltxt = sqltxt +", [Email] = N '"+ CStr (Me.txtEmail) +"' "
sqltxt = sqltxt +" WHERE ID = "+ CStr (frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
'rst.Close

sqltxt = "Update [dbo]. [tbl_Organizations] SET"
sqltxt = sqltxt + "[Name] = N '" + CStr (Me.txtOrgFirmName) + "'"
sqltxt = sqltxt + ", [Type] = N '" + CStr (Me.txtOrgFirmType) + "'"
sqltxt = sqltxt + ", [EstablishmentDate] ='" + CStr (Format (Me.txtOrgFoundedDate,
"YYYY-MM-DD")) "'" "
sqltxt = sqltxt +", [RegistrationNum] = N '"+ CStr (Me.txtOrgRegNum) +"' "
sqltxt = sqltxt +" WHERE ID = "+ CStr (frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
```

```
'rst.Close

sqltxt = sqltxt + "Update [dbo]. [tbl_Individuals]"
sqltxt = sqltxt + "SET [Occupation] = ''"
sqltxt = sqltxt + ", [BirthDate] = '" + Format (CStr (Me.txtDriverBirthDate), " YYYY-MM-DD ") +" '"
sqltxt = sqltxt +", [LastName] = N' "+ CStr (Me.txtDriverLastName) +" '"
sqltxt = sqltxt +", [FirstName] = N' "+ CStr (Me.txtDriverFirstName) + "'"
sqltxt = sqltxt + ", [FatherName] = N'" + CStr (Me.txtDriverFatherName) + "'"
sqltxt = sqltxt + ", [ADT] = N'" + CStr (Me .txtDriverADT) + "'"
sqltxt = sqltxt + "WHERE ID =" + CStr (frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
```

find client with name, surname, surname

```
Case 1
sqltID = "SE .AFM, B.NAME "
sqltxt = sqltxt +" FROM [CarRental]. [Dbo]. [Tbl_Clients] AS A "
sqltxt = sqltxt +" INNER JOIN [CarRental]. [Dbo]. [Tbl_Organizations] AS B ON A. ID = B.ID "
sqltxt = sqltxt +" WHERE A.AFM LIKE N '% "+ CStr (txtSearchCriteria) +"%' "
sqltxt = sqltxt +" UNION "
sqltxt = sqltxt +" SELECT A.ID, A.AFM , B.LASTNAME AS NAME "
sqltxt = sqltxt +" FROM [C arRental]. [dbo]. [tbl_Clients] AS A "
sqltxt = sqltxt +" INNER JOIN [CarRental]. [dbo]. [tbl_Individuals] AS B ON A.ID = B.ID "
sqltxt = sqltxt +" WHERE A. AFM LIKE N '% "+ CStr (txtSearchCriteria) +"%' "
sqltxt = sqltxt +" AND B.ID NOT IN (SELECT ID FROM TBL_ORGANIZATIONS) "
Case 2
sqltxt =" SELECT A.ID, A.NAME, B. AFM "
sqltxt = sqltxt +" FROM [CarRental]. [Dbo]. [Tbl_Organizations] AS A "
sqltxt = sqltxt +" INNER JOIN [CarRental]. [Dbo]. [Tbl_Clients] AS B ON A.ID = B.ID "
sqltxt = sqltxt +" WHERE A.NAME LIKE N '% "+ CStr (txtSearchCriteria) +"%' "
Case 3
sqltxt =" SELECT A.ID, A.LASTNAME, B.AFM "
sqltxt = sqltxt +" FROM [ CarRental]. [Dbo]. [Tbl_Individuals] AS A "
sqltxt = sqltxt +" INNER JOIN [CarRental]. [Dbo]. [Tbl_Clients] AS B ON A.ID = B.ID "
sqltxt = sqltxt +" WHERE A. LASTNAME LIKE N '% "+ CStr (txtSearchCriteria) +"%' "
sqltxt = sqltxt +" AND A.ID NOT IN (SELECT ID FROM TBL_ORGANIZATIONS) "
End Select
```

import rental

```
sqltxt =" INSERT INTO [dbo]. [Tbl_ ([ID], [DelivererEmpID]) "
sqltxt = sqltxt + "Values (" + CStr (frmReservationID)
sqltxt = sqltxt + "," + CStr (Me.comboEmp.Value) + ")"
Set con = New ADODB.Connection
```

rental update

```
sqltxt = "UPDATE [dbo]. [tbl_Rentals ] "
sqltxt = sqltxt +" SET [ReceiverEmpID] = "+ CStr (Me.comboEmpReturn.Value)
sqltxt = sqltxt +", [CarConditionGrade] = '"+ CStr (Me.txtCarCondition) +"' "
sqltxt = sqltxt +" , [Ontime] = "+ CStr (Me.Frame1.Value)
sqltxt = sqltxt +", [Kilometers] = "+ CStr (txtKillos)
sqltxt = sqltxt +" WHERE [ID] = "+ CStr (frmReservationID)
```

delete reservation

```
sqltxt = "DELETE FROM DBO.TBL_RESERVATIONS"
sqltxt = sqltxt + "WHERE ID =" + CStr (ResID)
```

reservation

```
sqltxt = "SELECT A. *, B.ID AS RentalID, B. [DelivererEmpID], B. [ReceiverEmpID], B.
[CarConditionGrade], B. [Ontime], B.Kilometers, "
sqltxt = sqltxt +" C.Model, C.Maker, C.LicensePlate, C.BranchID, "
sqltxt = sqltxt +" I.LastName, I. FirstName, "
sqltxt = sqltxt +" O.Name, "
sqltxt = sqltxt +" P.Landline, P.Mobile "
sqltxt = sqltxt +" FROM [CarRental]. [Dbo] .tbl_R eservations AS A "
sqltxt = sqltxt +" INNER JOIN [CarRental]. [dbo] .tbl_CARS AS C ON A.CarID = C.ID "
sqltxt = sqltxt +" INNER JOIN [CarRental]. [dbo] .tbl_Clients AS P ON A.ClientID = P.ID "
sqltxt = sqltxt +" LEFT OUTER JOIN [CarRental]. [Dbo] .tbl_Rentals AS B ON A.ID = B.ID "
sqltxt = sqltxt +" LEFT OUTER JOIN [CarRental]. [Dbo ] .tbl_Individuals AS I ON A.ClientID
= I.ID "
sqltxt = sqltxt +" LEFT OUTER JOIN [CarRental]. [dbo] .tbl_Organizations AS O ON
A.ClientID = O.ID "
sqltxt = sqltxt +" WHERE A. ID = "+ CStr (RentalID)
```

import reservation

```
sqltxt =" INSERT INTO [dbo]. [Vw_UpcomingRentals] ([CarID], [ClientID], [ReservationDate],
[StartDate], [EndDate]) "
sqltxt = sqltxt +" OUTPUT INSERTED.ID "
sqltxt = sqltxt +" VALUES ("+ CStr (CarID) +", "+ CStr (ClientID) +", "
sqltxt = sqltxt +" '"+ CStr (DateReserve) +"', '"+ CStr (DateFrom) + "','" + CStr (DateTo) + "')"
```

reservation update

```
sqltxt = "UPDATE DBO.VW_UPCOMINGRENTALS"
sqltxt = sqltxt + "SET PAYMENT ='" + Format (Payment, "YYYY-MM-DD ") +" '"
sqltxt = sqltxt + "WHERE ID =" + CStr (ResID)
```

store staff

```
sqltxt = "SELECT [ID], [LastName], [FirstName] FROM TBL_EMPLOYEES"
sqltxt = sqltxt + "WHERE BRANCHID =" + Me.ComboBranchReturn.Val
sq + "ORDER BY [LastName], [FirstName]"
```

import

```
sqltxt = "INSERT INTO [dbo]. [Tbl_Cars]"
sqltxt = sqltxt + "OUTPUT (INSERTED.ID)"
sqltxt = sqltxt + "([BranchID], [Maker], [Model], [Type], [Cubism], [Horsepower],
[LicensePlate], [YearOfPurchase], [Kilometers] "
sqltxt = sqltxt +", [LastServiceDate], [LastServiceKms], [NextServiceKms],
[InsuranceEndDate], [InsuranceCompany], [InsuranceAgreementNum], [Available]) "
sqltxt = sqltxt +" Values ("
sqltxt = sqltxt +" [BranchID] = "+ CStr (Me.comboBranches.Value)
sqltxt = sqltxt +", [Maker] = N '"+ CStr (Me.txtMaker) +"' "
sqltxt = sqltxt +", [Model] = N '"+ CStr (Me.txtModel) +"' "
sqltxt = sqltxt +", [Type ] = N '"+ CStr (Me.txtType) +"' "
sqltxt = sqltxt +", [Cubism] = "+ CStr (Me.txtCubism)
sqltxt = sqltxt +", [Horsepower] = "+ CStr (Me. txtHorsePower)
sqltxt = sqltxt + ", [LicensePlate] = N '" + CStr (Me.txtLicensePlate) + "'"
sqltxt = sqltxt + ", [YearOfPurchase] = '" + Format (Me.txtYearOfPurchase, "YYYY-MM-DD")
+ "'"
sqltxt = sqltxt + ", [Kilometers] =" + CStr (Me.txtKilometers)
sqltxt = sqltxt + ", [LastServiceDate] ='" + Format (Me.txtLastServiceDate, "YYYY-MM-DD") +
"'"
sqltxt = sqltxt + ", [LastServiceKms] =" + CStr (Me.txtLastServiceKms)
sqltxt = sqltxt + ", [NextServiceKms] =" + CStr (Me.txtNextServiceKms)
sqltxt = sqltxt + ", [InsuranceEnd "+ Format (Me.txtInsuranceEndDate," YYYY-MM-DD ") +" '"
sqltxt = sqltxt +", [InsuranceCompany] = N' "+ CStr (Me.txtInsuranceCompany) +" '"
sqltxt = sqltxt +", [InsuranceAgreementNum] = N '"+ CStr (Me.txtInsuranceAgreementNum)
+"' "
```

car update

```
sqltxt =" UPDATE DBO.TBL_CARS SET "
sqltxt = sqltxt +" [BranchID] = "+ CStr (Me.comboBranches.Value)
sql sqltxt + ", [Maker] = N '" + CStr (Me.txtMaker) + "'"
sqltxt = sqltxt + ", [Model] = N '" + CStr (Me.txtModel) + "'"
sqltxt = sqltxt + ", [Type] = N '" + CStr (Me .txtType) + "'"
sqltxt = sqltxt + ", [Cubism] =" + CStr (Me.txtCubism)
sqltxt = sqltxt + ", [Horsepower] =" + CStr (Me.txtHorsePower)
sqltxt = sqltxt + ", [ LicensePlate] = N '"+ CStr (Me.txtLicensePlate) +"' "
sqltxt = sqltxt +", [YearOfPurchase] = '"+ Format (Me.txtYearOfPurchase," YYYY-MM-DD ") +"' "
sqltxt = sqltxt + ", [Kilometers] =" + CStr (Me.txtKilometers)
sqltxt = sqltxt + ", [LastServiceDate] = '" + Format (Me.txtLastServiceDate, "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ", [LastServiceKms] =" + CStr (Me.txtLastServiceKms)
sqltxt = sqltxt + ", [NextServiceKms] =" + CStr (Me.txtNextServiceKms)
sqltxt = sqltxt + ", [InsuranceEndDate] = '" + Format (Me. txtInsuranceEndDate, "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ", [InsuranceCompany] = N'" + CStr (Me.txtInsuranceCompany) + "'"
sqltxt = sqltxt + ", [InsuranceAgreementNum] = N' "+ CStr (Me.txtInsuranceAgreementNum) +" '"
' sqltxt = sqltxt +", [Available] = "+ CStr (Me.)
Sqltxt = sqltxt +" WHERE [ID] = "+ CStr (frmCarID)
```

delete car

```
sqltxt = DELETE FROM DBO .TBL_CARS "
sqltxt = sqltxt +" WHERE [CARID] = "+ CStr (frmCarID)
```

delete damage

```
sqltxt =" DELETE FROM DBO.TBL_DAMAGES "
sqltxt = sqltxt +" WHERE [CARID] = "+ CStr (frmCI
) "AND [KILOMETERS] =" + CStr (listDamages.Column (1, listDamages.ListIndex + 1))
```

import update damage

```
If frmMode = "New" Then
sqltxt = "INSERT INTO [dbo]. [Tbl_Damages] (CarID, Kilometers, EntryDate, Description) "
sqltxt = sqltxt +" VALUES ("+ CStr (client client from ID
```

```
sqltxt =" SELECT C. [ID], C. [EntryDate], C. [AFM], C. [AddressStreet], C. [AddressNumber], C. [AddressTK], C. [AddressCity], C. [LandLine], C. [Mobile], C. [Email], "
sqltxt = sqltxt +" I. [ID] AS IND_ID, I. [Occupation], I. [BirthDate], I. [LastName], I. [FirstName], I. [FatherName], I. [ADT], "
sqltxt = sqltxt +" O. [ID] AS ORG_ID, O. [Name], O. [Type], O. [EstablishmentDate], O. [RegistrationNum] "
sqltxt = sqltxt +" FROM TBL_CLIENTS AS C "
sqltxt = sqltxt +" LEFT OUTER JOIN TBL_INDIVIDUALS AS I ON C.ID = I. ID "
sqltxt = sqltxt +" LEFT OUTER JOIN TBL_ORGANIZATIONS AS O ON C.ID = O.ID "
sqltxt = sqltxt +" WHERE C.ID = "+ CStr (ClientID)
```

delete client

```
sqltxt =" DELETE FROM DBO.TBL_CLIENTS "
sqltxt = sqltxt +" WHERE [ID] = "+ CStr (frmClientID )
```

private input

```
sqltxt = "INSERT INTO [dbo]. [tbl_Clients] ([EntryDate], [AFM], [AddressStreet],
[AddressNumber], [AddressTK], [AddressCity], [LandLine], [Mobile], [Email ]) "
sqltxt = sqltxt +" OUTPUT (INSERTED.ID) "
sqltxt = sqltxt +" VALUES ('"+ Format (Date," YYYY-MM-DD ") +"' "
sqltxt = sqltxt +", N '" + CStr (Me.txtIndAFM) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtAddress) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtAddressNum) + "' "
sqltxt = sqltxt +", N '"+ CStr (Me.txtTK) +"' "
sqltxt = sqltxt +", N '"+ CStr (Me.txtCity) +"' "
sqltxt = sqltxt +", N '" + CStr (Me.txtPhoneNum) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtMobNum) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtEmail) + "' ) "

sqltxt =" INSERT INTO [dbo]. [tbl_Individuals] ([ID], [Occupation], [BirthDate], [LastName],
[FirstName], [FatherName], [ADT]) "
sqltxt = sqltxt +" VALUES (" + CStr (ClientID)
sqltxt = sqltxt + ", N '" + CStr (Me.txtIndOccupation) + "'"
sqltxt = sqltxt + ", '" + Format (CStr (Me.txtIndBirthDate)), "YYYY-MM-DD" ) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtIndLastName) + "'"
sqltxt = sqltxt + ", N'" + CStr (Me.txtIndFirstName) + "'"
sqltxt = sqltxt + " , N '"+ CStr (Me.txtIndFatherName) +"' "
sqltxt = sqltxt +", N '"+ CStr (Me.txtIndADT) +"') "
```

private update

```
sqltxt =" Update [dbo]. [Tbl_Clients] "
sqltxt = sqltxt + "SET [AFM] =" + "N '" + CStr (Me.txtIndAFM) + "'"
sqltxt = sqltxt + ", [AddressStreet] = N '" + CStr (Me.txtAddress) + "' "
sqltxt = sqltxt +", [AddressNumber] = N '"+ CStr (Me.txtAddressNum) +"' "
sqltxt = sqltxt +", [AddressTK] = N '"+ CStr (Me.txtTK) +"' "
sqltxt = sqltxt + ", [AddressCity] = N '" + CStr (Me.txtCity) + "'"
sqltxt = sqltxt + ", [LandLine] = N '" + CStr (Me.txtPhoneNum) + "'"
sqltxt = sqltxt + ", [Mobile] = N '" + CStr (Me.txtMobNum) + "'"
sqltxt = sqltxt + ", [Email] = N '" + CStr (Me.txtEmail) + "'"
sqltxt = sqltxt + " WHERE ID = "+ CSt r (frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
'rst.Close

sqltxt = sqltxt + "Update [dbo]. [tbl_Individuals]"
sqltxt = sqltxt + "SET [Occupation] = N '" + CStr (Me.txtIndOccupation) + "'"
sqltxt = sqltxt + ", [BirthDate] = '" + Format (CStr (Me.txtIndBirthDate),
"YYYY-MM-DD") + "'
sqltxt = sqltxt +", [LastName] = N' "+ CStr (Me.txtIndLastName) +" '"
sqltxt = sqltxt + ", [FirstName] = N '" + CStr (Me.txtIndFirstName) + "'"
sqltxt = sqltxt + ", [FatherName] = N '" + CStr (Me.txtIndFatherName) + "'"
sqltxt = sqltxt + ", [ADT] = N '"+ CStr (Me.txtIndADT) +"' "
sqltxt = sqltxt +" WHERE ID = "+ CStr (frmClientID)
```

import company

```
sqltxt =" INSERT INTO [dbo]. [Tbl_Clients] ([EntryDate] , [AFM], [AddressStreet],
[AddressNumber], [AddressTK], [AddressCity], [LandLine], [Mobile], [Email]) "
sqltxt = sqltxt +" OUTPUT (INSERTED.ID) "
sqltxt = sqltxt + " VALUES ('" + Format(Date, "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtOrgAFM) + "'"
sqltxt = sqltxt + ",N' " + CStr(Me.txtAddress) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtAddressNum) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtTK) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtCity) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtPhoneNum) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtMobNum) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtEmail) + "')"
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
ClientID = rst.Fields.Item(0).Value
rst.Close

sqltxt = " INSERT INTO
[dbo].[tbl_Organizations]([ID],[Name],[Type],[EstablishmentDate],[RegistrationNum])"
sqltxt = sqltxt + " VALUES (" + CStr(ClientID)
sqltxt = sqltxt + ",N'" + CStr(Me.txtOrgFirmName) + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtOrgFirmType) + "'"
sqltxt = sqltxt + ",'" + Format(CStr(Me.txtOrgFoundedDate), "YYYY-MM-DD") + "'"
sqltxt = sqltxt + ",N'" + CStr(Me.txtOrgRegNum) + "')"
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly

sqltxt = "INSERT INTO
[dbo].[tbl_Individuals]([ID],[Occupation],[BirthDate],[LastName],[FirstName],[FatherName]
,[ADT])"
sqltxt = sqltxt + " VALUES(" + CStr(ClientID)
sqltxt = sqltxt + " ,N'ÕÐÁËËÇËÏÓ'"
sqltxt = sqltxt + " ,'" + Format(CStr(Me.txtDriverBirthDate), "YYYY-MM-DD") + "'"
sqltxt = sqltxt + " ,N'" + CStr(Me.txtDriverLastName) + "'"
sqltxt = sqltxt + " ,N'" + CStr(Me.txtDriverFirstName) + "'"
sqltxt = sqltxt + " ,N'" + CStr(Me.txtDriverFatherName) + "'"
sqltxt = sqltxt + " ,N'" + CStr(Me.txtDriverADT) + "')"
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
```

ενημέρωση εταιρείας

```
sqltxt = " Update [dbo].[tbl_Clients]"
sqltxt = sqltxt + " SET [AFM] = " + "N'" + CStr(Me.txtOrgAFM) + "'"
sqltxt = sqltxt + ",[AddressStreet] = N'" + CStr(Me.txtAddress) + "'"
sqltxt = sqltxt + ",[AddressNumber] = N'" + CStr(Me.txtAddressNum) + "'"
sqltxt = sqltxt + ",[AddressTK] = N'" + CStr(Me.txtTK) + "'"
sqltxt = sqltxt + ",[AddressCity] = N'" + CStr(Me.txtCity) + "'"
sqltxt = sqltxt + ",[LandLine] = N'" + CStr(Me.txtPhoneNum) + "'"
sqltxt = sqltxt + ",[Mobile] = N'" + CStr(Me.txtMobNum) + "'"
sqltxt = sqltxt + ",[Email] = N'" + CStr(Me.txtEmail) + "'"
sqltxt = sqltxt + " WHERE ID=" + CStr(frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
'rst.Close

sqltxt = "Update [dbo].[tbl_Organizations] SET"
sqltxt = sqltxt + " [Name] = N'" + CStr(Me.txtOrgFirmName) + "'"
sqltxt = sqltxt + ",[Type] = N'" + CStr(Me.txtOrgFirmType) + "'"
sqltxt = sqltxt + " ,[EstablishmentDate] = '" + CStr(Format(Me.txtOrgFoundedDate,
"YYYY-MM-DD")) + "'"
sqltxt = sqltxt + ",[RegistrationNum] = N'" + CStr(Me.txtOrgRegNum) + "'"
sqltxt = sqltxt + " WHERE ID=" + CStr(frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
'rst.Close

sqltxt = sqltxt + "Update [dbo].[tbl_Individuals]"
sqltxt = sqltxt + " SET [Occupation] =' ' "
sqltxt = sqltxt + ",[BirthDate] = '" + Format(CStr(Me.txtDriverBirthDate),
"YYYY-MM-DD") + "'"
sqltxt = sqltxt + ",[LastName] = N'" + CStr(Me.txtDriverLastName) + "'"
sqltxt = sqltxt + ",[FirstName] = N'" + CStr(Me.txtDriverFirstName) + "'"
sqltxt = sqltxt + ",[FatherName] = N'" + CStr(Me.txtDriverFatherName) + "'"
sqltxt = sqltxt + ",[ADT] = N'" + CStr(Me.txtDriverADT) + "'"
sqltxt = sqltxt + " WHERE ID=" + CStr(frmClientID)
rst.Open sqltxt, con, adOpenKeyset, adLockReadOnly
```

ευρεση πελάτη με αφμ, επωνυμία, επώνυμο

```
Case 1
sqltxt = " SELECT A.ID,A.AFM, B.NAME"
sqltxt = sqltxt + " FROM [CarRental].[dbo].[tbl_Clients] AS A"
sqltxt = sqltxt + " INNER JOIN [CarRental].[dbo].[tbl_Organizations] AS B ON
A.ID=B.ID"
```

```
sqltxt = sqltxt + " WHERE A.AFM LIKE N'%" + CStr(txtSearchCriteria) + "%'"
sqltxt = sqltxt + " UNION"
sqltxt = sqltxt + " SELECT A.ID,A.AFM, B.LASTNAME AS NAME"
sqltxt = sqltxt + " FROM [CarRental].[dbo].[tbl_Clients] AS A"
sqltxt = sqltxt + " INNER JOIN [CarRental].[dbo].[tbl_Individuals] AS B ON A.ID=B.ID"
sqltxt = sqltxt + " WHERE A.AFM LIKE N'%" + CStr(txtSearchCriteria) + "%'"
sqltxt = sqltxt + " AND B.ID NOT IN (SELECT ID FROM TBL_ORGANIZATIONS)"
Case 2
sqltxt = " SELECT A.ID, A.NAME, B.AFM"
sqltxt = sqltxt + " FROM [CarRental].[dbo].[tbl_Organizations] AS A"
sqltxt = sqltxt + " INNER JOIN [CarRental].[dbo].[tbl_Clients] AS B ON A.ID=B.ID"
sqltxt = sqltxt + " WHERE A.NAME LIKE N'%" + CStr(txtSearchCriteria) + "%'"
Case 3
sqltxt = " SELECT A.ID, A.LASTNAME, B.AFM"
sqltxt = sqltxt + " FROM [CarRental].[dbo].[tbl_Individuals] AS A"
sqltxt = sqltxt + " INNER JOIN [CarRental].[dbo].[tbl_Clients] AS B ON A.ID=B.ID"
sqltxt = sqltxt + " WHERE A.LASTNAME LIKE N'%" + CStr(txtSearchCriteria) + "%'"
sqltxt = sqltxt + " AND A.ID NOT IN (SELECT ID FROM TBL_ORGANIZATIONS)"
End Select
```

εισαγωγή ενοικίασης

```
sqltxt = "INSERT INTO [dbo].[tbl_Rentals] ([ID],[DelivererEmpID])"
sqltxt = sqltxt + " Values (" + CStr(frmReservationID)
sqltxt = sqltxt + " ," + CStr(Me.comboEmp.Value) + ")"
Set con = New ADODB.Connection
```

ενημέρωση ενοικίασης

```
sqltxt = "UPDATE [dbo].[tbl_Rentals]"
sqltxt = sqltxt + " SET [ReceiverEmpID] = " + CStr(Me.comboEmpReturn.Value)
sqltxt = sqltxt + " ,[CarConditionGrade] = '" + CStr(Me.txtCarCondition) + "'"
sqltxt = sqltxt + " ,[Ontime] = " + CStr(Me.Frame1.Value)
sqltxt = sqltxt + " ,[Kilometers] = " + CStr(txtKillos)
sqltxt = sqltxt + " WHERE [ID] = " + CStr(frmReservationID)
```

διαγραφή κράτησης

```
sqltxt = "DELETE FROM DBO.TBL_RESERVATIONS"
sqltxt = sqltxt + " WHERE ID=" + CStr(ResID)
```

ευρεση κράτησης

```
sqltxt = " SELECT A.*, B.ID AS RentalID,
B.[DelivererEmpID],B.[ReceiverEmpID],B.[CarConditionGrade],B.[Ontime],B.Kilometers,"
sqltxt = sqltxt + " C.Model, C.Maker, C.LicensePlate,C.BranchID,"
sqltxt = sqltxt + " I.LastName, I.FirstName,"
sqltxt = sqltxt + " O.Name,"
sqltxt = sqltxt + " P.Landline, P.Mobile"
sqltxt = sqltxt + " FROM [CarRental].[dbo].tbl_Reservations AS A "
sqltxt = sqltxt + " INNER JOIN [CarRental].[dbo].tbl_CARS AS C ON A.CarID=C.ID"
sqltxt = sqltxt + " INNER JOIN [CarRental].[dbo].tbl_Clients AS P ON A.ClientID=P.ID"
sqltxt = sqltxt + " LEFT OUTER JOIN [CarRental].[dbo].tbl_Rentals AS B ON A.ID=B.ID"
sqltxt = sqltxt + " LEFT OUTER JOIN [CarRental].[dbo].tbl_Individuals AS I ON A.ClientID=I.ID"
sqltxt = sqltxt + " LEFT OUTER JOIN [CarRental].[dbo].tbl_Organizations AS O ON A.ClientID=O.ID"
sqltxt = sqltxt + " WHERE A.ID=" + CStr(RentalID)
```

εισαγωγή κράτησης

```
sqltxt = "INSERT INTO [dbo].[vw_UpcomingRentals] ([CarID],[ClientID],[ReservationDate],[StartDate],[EndDate])"
sqltxt = sqltxt + " OUTPUT INSERTED.ID"
sqltxt = sqltxt + " VALUES (" + CStr(CarID) + "," + CStr(ClientID) + ","
sqltxt = sqltxt + "'" + CStr(DateReserve) + "','" + CStr(DateFrom) + "','" + CStr(DateTo) + "')"
```

ενημέρωση κράτησης

```
sqltxt = "UPDATE DBO.VW_UPCOMINGRENTALS"
sqltxt = sqltxt + " SET PAYMENT='" + Format(Payment, "YYYY-MM-DD") + "'"
sqltxt = sqltxt + " WHERE ID=" + CStr(ResID)
```

υπάλληλοι σε κατάστημα

```
sqltxt = "SELECT [ID],[LastName],[FirstName] FROM TBL_EMPLOYEES"
sqltxt = sqltxt + " WHERE BRANCHID=" + Me.ComboBranchReturn.Value
sqltxt = sqltxt + " ORDER BY [LastName],[FirstName]"
```

εισαγωγή΄αυτοκινήτου

```
sqltxt = "INSERT INTO [dbo].[tbl_Cars]"
sqltxt = sqltxt + " OUTPUT(INSERTED.ID)"
sqltxt = sqltxt + " ([BranchID],[Maker],[Model],[Type],[Cubism],[Horsepower],[LicensePlate],[YearOfPurchase],[Kilometers]"
sqltxt = sqltxt + " ,[LastServiceDate],[LastServiceKms],[NextServiceKms],[InsuranceEndDate],[InsuranceCompany],[InsuranceAgreementNum],[Available])"
sqltxt = sqltxt + " Values ("
sqltxt = sqltxt + "[BranchID] =" + CStr(Me.comboBranches.Value)
sqltxt = sqltxt + ",[Maker] =N'" + CStr(Me.txtMaker) + "'"
sqltxt = sqltxt + ",[Model] =N'" + CStr(Me.txtModel) + "'"
sqltxt = sqltxt + ",[Type] =N'" + CStr(Me.txtType) + "'"
sqltxt = sqltxt + ",[Cubism] =" + CStr(Me.txtCubism)
sqltxt = sqltxt + ",[Horsepower] =" + CStr(Me.txtHorsePower)
sqltxt = sqltxt + ",[LicensePlate] =N'" + CStr(Me.txtLicensePlate) + "'"
sqltxt = sqltxt + ",[YearOfPurchase] ='" + Format(Me.txtYearOfPurchase, "YYYY-MM-DD") +
"'"
sqltxt = sqltxt + ",[Kilometers] =" + CStr(Me.txtKilometers)
sqltxt = sqltxt + ",[LastServiceDate] ='" + Format(Me.txtLastServiceDate, "YYYY-MM-DD") +
"'"
sqltxt = sqltxt + ",[LastServiceKms] =" + CStr(Me.txtLastServiceKms)
sqltxt = sqltxt + ",[NextServiceKms] =" + CStr(Me.txtNextServiceKms)
sqltxt = sqltxt + ",[InsuranceEndDate] ='" + Format(Me.txtInsuranceEndDate,
"YYYY-MM-DD") + "'"
sqltxt = sqltxt + ",[InsuranceCompany] =N'" + CStr(Me.txtInsuranceCompany) + "'"
sqltxt = sqltxt + ",[InsuranceAgreementNum] =N'" + CStr(Me.txtInsuranceAgreementNum) +
"'"
```

ενημέρωση αυτοκινήτου

```
sqltxt = "UPDATE DBO.TBL_CARS SET "
sqltxt = sqltxt + "[BranchID] =" + CStr(Me.comboBranches.Value)
sqltxt = sqltxt + ",[Maker] =N'" + CStr(Me.txtMaker) + "'"
sqltxt = sqltxt + ",[Model] =N'" + CStr(Me.txtModel) + "'"
sqltxt = sqltxt + ",[Type] =N'" + CStr(Me.txtType) + "'"
sqltxt = sqltxt + ",[Cubism] =" + CStr(Me.txtCubism)
sqltxt = sqltxt + ",[Horsepower] =" + CStr(Me.txtHorsePower)
sqltxt = sqltxt + ",[LicensePlate] =N'" + CStr(Me.txtLicensePlate) + "'"
sqltxt = sqltxt + ",[YearOfPurchase] ='" + Format(Me.txtYearOfPurchase, "YYYY-MM-DD") +
"'"
sqltxt = sqltxt + ",[Kilometers] =" + CStr(Me.txtKilometers)
sqltxt = sqltxt + ",[LastServiceDate] ='" + Format(Me.txtLastServiceDate, "YYYY-MM-DD") +
"'"
sqltxt = sqltxt + ",[LastServiceKms] =" + CStr(Me.txtLastServiceKms)
sqltxt = sqltxt + ",[NextServiceKms] =" + CStr(Me.txtNextServiceKms)
sqltxt = sqltxt + ",[InsuranceEndDate] ='" + Format(Me.txtInsuranceEndDate,
"YYYY-MM-DD") + "'"
sqltxt = sqltxt + ",[InsuranceCompany] =N'" + CStr(Me.txtInsuranceCompany) + "'"
sqltxt = sqltxt + ",[InsuranceAgreementNum] =N'" + CStr(Me.txtInsuranceAgreementNum) +
"'"
'sqltxt = sqltxt + ",[Available] =" + CStr(Me.)
sqltxt = sqltxt + " WHERE [ID]=" + CStr(frmCarID)
```

διαγραφή αυτοκινήτου

```
sqltxt = "DELETE FROM DBO.TBL_CARS"
sqltxt = sqltxt + " WHERE [CARID]=" + CStr(frmCarID)
```

διαγραφή ζημιάς

```
sqltxt = "DELETE FROM DBO.TBL_DAMAGES"
sqltxt = sqltxt + " WHERE [CARID]=" + CStr(frmCarID)
sqltxt = sqltxt + " AND [KILOMETERS]=" + CStr(listDamages.Column(1,
listDamages.ListIndex + 1))
```

εισαγωή ενημέρωση ζημιάς

```
If frmMode = "New" Then
sqltxt = "INSERT INTO [dbo].[tbl_Damages] (CarID, Kilometers, EntryDate,
Description)"
sqltxt = sqltxt + " VALUES (" + CStr(frmCarID) + "," + CStr(txtKilometers) + ","
sqltxt = sqltxt + "'" + Format(txtEntryDate, "yyyy-mm-dd") + "',N'" + txtDescription +
"')"
Else
sqltxt = "UPDATE [dbo].[tbl_Damages]"
sqltxt = sqltxt + " SET [EntryDate] = '" + Format(txtEntryDate, "yyyy-mm-dd") + "'"
sqltxt = sqltxt + " ,[Description] = N'" + txtDescription + "'"
sqltxt = sqltxt + " WHERE [CarId] = " + CStr(txtCarID) + " And [Kilometers] = " +
CStr(txtKilometers)
End If
```

```
frmCarID) + "," + CStr(txtKilometers) + ","
sqltxt = sqltxt + "'" + Format(txtEntryDate, "yyyy-mm-dd") + "',N'" + txtDescription +
"')"
Else
sqltxt = "UPDATE [dbo].[tbl_Damages]"
sqltxt = sqltxt + " SET [EntryDate] = '" + Format(txtEntryDate, "yyyy-mm-dd") + "'"
sqltxt = sqltxt + " ,[Description] = N'" + txtDescription + "'"
sqltxt = sqltxt + " WHERE [CarId] = " + CStr(txtCarID) + " And [Kilometers] = " +
CStr(txtKilometers)
End If
```