

RENTAL CAR APPLICATION WITH TEMPORAL DATABASE

**A Report Submitted to the
Dokuz Eylül University, Department of Computer Engineering
In Partial Fulfillment of the Requirements for
CME4413 Topics in Databases**

**by
Mehmet UYĞUT
2015510064**

**Ömer Selim ATİLA
2015510010**

January, 2021

İZMİR

RENT A CAR SYSTEM WITH TEMPORAL DATABASE

1. Introduction

1.1 What is Temporal Databases?

A temporal database stores data relating to time instances. It offers temporal data types and stores information relating to past, present and future time. Temporal databases could be uni-temporal, bi-temporal or tri-temporal.

Uni-Temporal: A uni-temporal database has one axis of time, either the validity range or the system time range.

Bi-Temporal: A bi-temporal database has two axis of time these are valid time and transaction time or decision time.

Tri-Temporal: A tri-temporal database has three axes of time these are valid time, transaction time and decision time.

2. Rent a Car System with Temporal Database

In this project, main goal is creating a rent a car database with temporal database. For this situation we have three different tables which names are customer cars and contracts. Customer and cars will be created as temporal tables. So that cars and customer information will be hold in the history tables if any changes happens and user can see the snapshots from history.

2.1 Tools

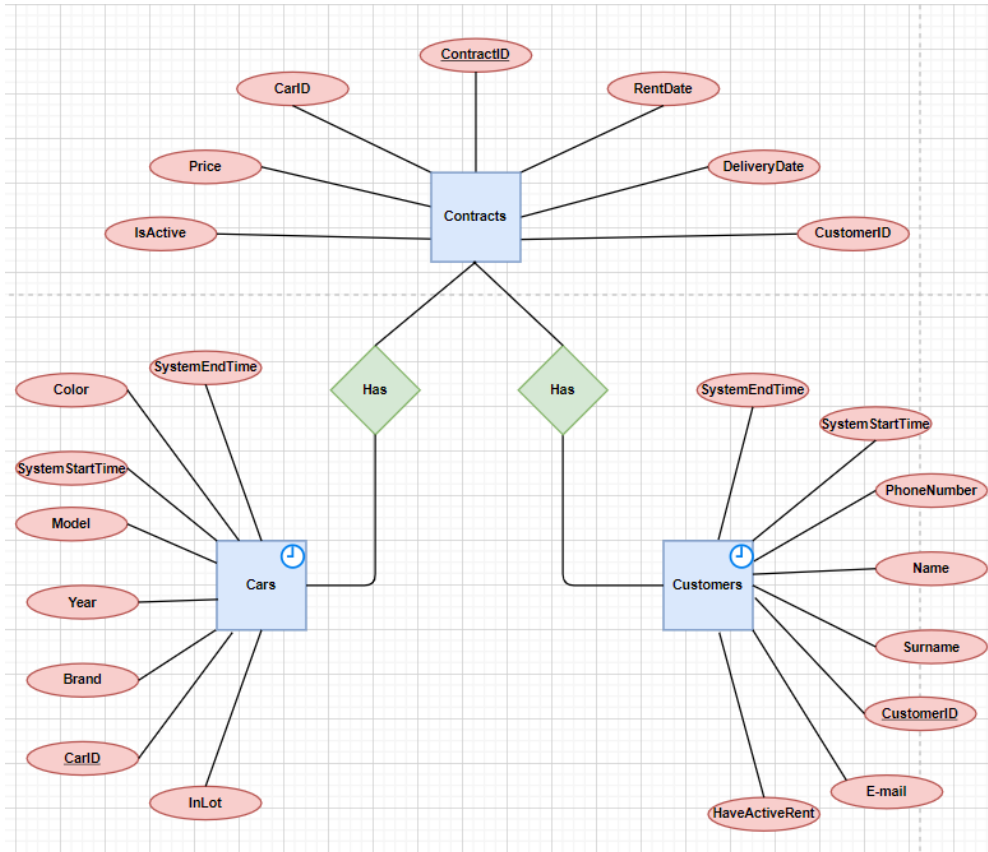
This project will be a form application because in general temporal databases help for the administration and we don't need any user screen. For the form application we use C# .net framework form application which is a great way to create a desktop application and for the database we use Microsoft SQL servers. Microsoft SQL server has a support for temporal tables and easy to connect with C# .net framework form application. So that MsSQL was great choice for us.

2.2 Connection Between MsSQL and C# .net Framework Form Application

System.Data.SqlClient library used for the connection between MsSQL and C# form application. You can see the connection line between them below.

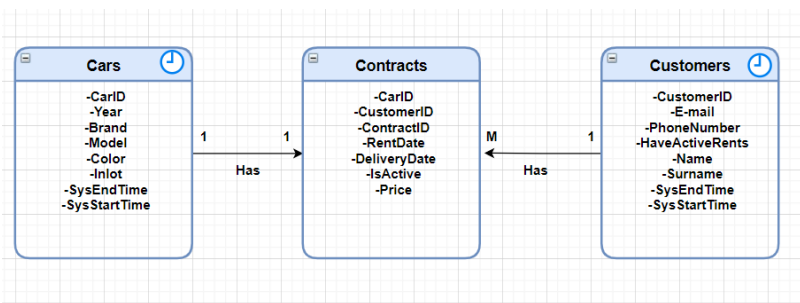
```
static SqlConnection con = new SqlConnection("Data Source=DESKTOP-M9RRIT7;Initial Catalog=rentAcar;Integrated Security=True");
```

2.3 ER Diagram for Rent a Car Project



In ER diagrams if there is a clock on upper right corner of table means that the table is temporal a temporal table and has own history table. The underlined words CustomerID, CarID and ContractID indicate that they are primary keys for their tables.

2.4 UML Diagram for Rent a Car Project



In ER diagrams if there is a clock on upper right corner of table means that the table is temporal a temporal table and has own history table. As you can see in the chart cars and contracts has 1 to 1 relation which means 1 car only has 1 contracts at the same time and Customers between contracts has 1 to M relationship which means 1 customer can have many contracts at the same time.

2.5 Screens and Their Description from Project

In this section, we show the screenshots from the project and explain how to use the application.

2.5.1 Car Status and Adding New Car Screen

CarId	Year	Make	Model	Color	InLot
1	2010	Renault	Megane	red	<input type="checkbox"/>
2	2011	Fiat	Punto	Gray	<input type="checkbox"/>
3	2015	Audi	A3	Red	<input checked="" type="checkbox"/>

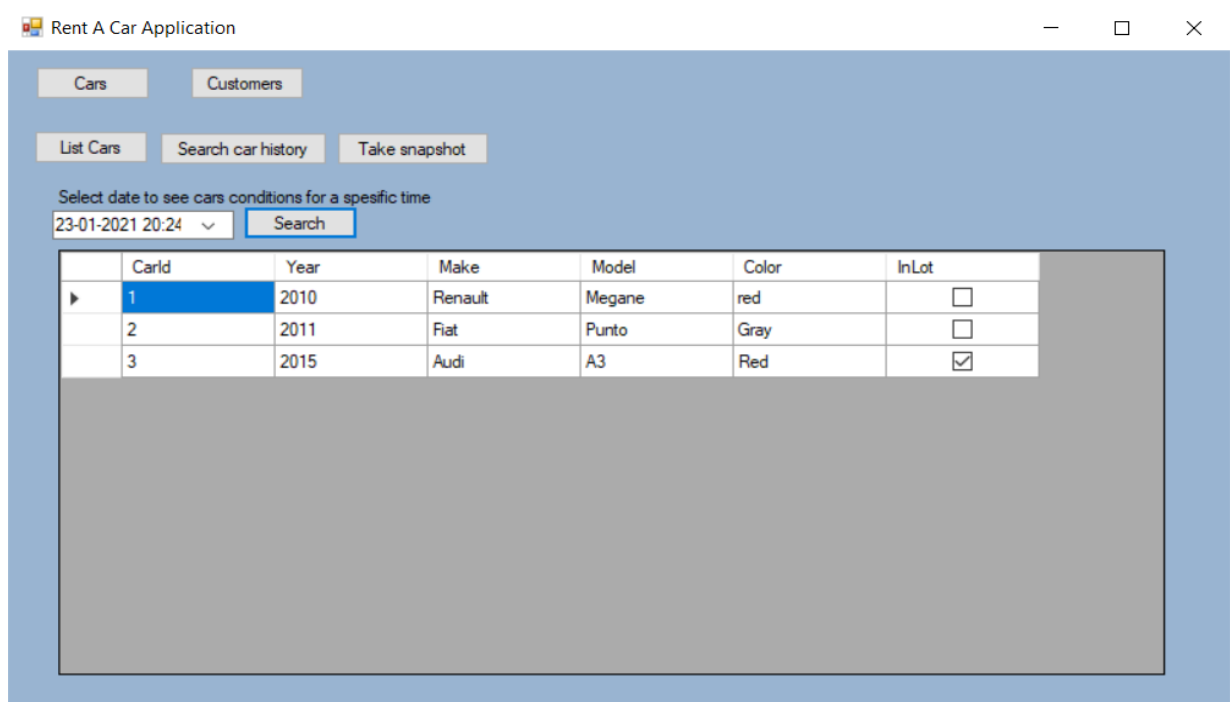
This Screen reaches by clicking List Cars button on the left upper corner. As you can see, all the cars listed in the middle of screen. User can see whether the cars have a contract or not from the inlot section. If user wants to see only cars that are currently available, user can click the "show only available cars" tab on the top left. Finally, if user wants to add a new car, user can fill the information on the left side and press the "add new car" button.

2.5.2 Search Car History

CarId	Year	Make	Model	Color	InLot
1	2010	Renault	Megane	red	<input checked="" type="checkbox"/>
1	2010	Renault	Megane	red	<input type="checkbox"/>
2	2011	Fiat	Punto	Gray	<input checked="" type="checkbox"/>
2	2011	Fiat	Punto	Gray	<input type="checkbox"/>
3	2015	Audi	A3	Red	<input checked="" type="checkbox"/>

This screen opens when user clicks on “search car history” button. In this section you can see the changes in cars at certain dates in the past. As you can see in the example, the first car appears to be inlot first, then not inlot in between 23-01-2021 and 24-01-2021, which means the car was rented between this dates. The queries on this page have been created with temporal queries.

2.5.3 Take Snapshot



The screenshot shows a web application window titled "Rent A Car Application". It features a navigation bar with "Cars" and "Customers" tabs. Below this, there are three buttons: "List Cars", "Search car history", and "Take snapshot". A label "Select date to see cars conditions for a spesific time" is followed by a date-time picker showing "23-01-2021 20:24" and a "Search" button. Below the search controls is a table with the following data:

	CarId	Year	Make	Model	Color	InLot
▶	1	2010	Renault	Megane	red	<input type="checkbox"/>
	2	2011	Fiat	Punto	Gray	<input type="checkbox"/>
	3	2015	Audi	A3	Red	<input checked="" type="checkbox"/>

The table is followed by a large gray rectangular area, likely a placeholder for a chart or additional data.

This section is accessed by clicking the “take snapshot” button. After clicking “Take snapshot”, user can enter a date and time to instantly see the status of all cars on that date and time. Also this screens queries use temporal queries to show the history.

2.5.4 Customer Status and Adding or Updating New Customer Screen

Rent A Car Application

Cars Customers

Customers Contracts Customer history

☐ show only have active renting customers

Enter Name
[Text Box]

Enter Surname
[Text Box]

Enter Phone
[Text Box]

Enter Email
[Text Box]

Add new

Update User

Refresh

	CustomerId	Name	Surname	PhoneNumber	Email	haveActiveRent
▶	2	ömer	atila	5451234545	ömer@gmail.com	<input checked="" type="checkbox"/>
	3	mehmet	uygut	5346546565	mehmet@gmail.c...	<input checked="" type="checkbox"/>

The user reaches this screen after pressing the customers button at the top. By filling the information on this screen, a new user can be added or the user's information can be updated. After the update is done, pressing the refresh button will show the updated information on the screen. Also click the "show only have active rent" button to view customers who currently have a contract.

2.5.5 Customer History

Rent A Car Application

Cars Customers

Customers Contracts Customer history

omer atila

	ContractsId	CarId	CustomerId	Price	RentDate	DeliveryDate	IsActive
▶	1	1	2	1500	1/23/2021 5:20 ...	1/28/2021 5:20 ...	<input checked="" type="checkbox"/>

User can access this screen by clicking the “customer history” button. In this section, user can select the customer's name from the dropdown textbox on the left and access all the contracts customer has made so far.

2.5.6 Contracts

The screenshot shows a web application window titled "Rent A Car Application". At the top, there are two tabs: "Cars" and "Customers". Below these, there are three sub-tabs: "Customers", "Contracts" (which is selected), and "Customer history". On the left side, there is a form for creating a new contract. It includes a "Select Customer" dropdown menu, a "Select Car" dropdown menu showing "3 Audi A3 Red", an "Enter end date" field with a date picker showing "23-01-2021 20:25", an "Enter price" text input field, and a "Submit" button. In the center, there is a table displaying a list of contracts. The table has columns for "ContractsId", "CarId", "CustomerId", "Price", "RentDate", "DeliveryDate", and "IsActive". Two contracts are listed: Contract 1 with CarId 1, CustomerId 2, Price 1500, RentDate 1/23/2021 5:20 PM, DeliveryDate 1/28/2021 5:20 PM, and IsActive checked; and Contract 2 with CarId 2, CustomerId 3, Price 1000, RentDate 1/23/2021 5:21 PM, DeliveryDate 2/3/2021 5:20 PM, and IsActive checked. Below the table, there is a checkbox labeled "Show only active contracts" which is checked. At the bottom right, there is a button labeled "finish contract".

	ContractsId	CarId	CustomerId	Price	RentDate	DeliveryDate	IsActive
▶	1	1	2	1500	1/23/2021 5:20 PM	1/28/2021 5:20 PM	<input checked="" type="checkbox"/>
	2	2	3	1000	1/23/2021 5:21 PM	2/3/2021 5:20 PM	<input checked="" type="checkbox"/>

The user can access this section by pressing the "contracts" button. In this section, all contracts appear in the middle of the screen. At the same time, you can create a new contract by entering the information on the left. Finally, by clicking the "show only active contracts" button, the active contracts can be executed and the contract termination operations can be performed by clicking the "finish contract" button.

2.6 Temporal Query Examples

2.6.1 Creating Temporal Table

```
CREATE TABLE [dbo].[CarInventory](
    [CarId] [int] IDENTITY(1,1) NOT NULL,
    [Year] [int] NULL,
    [Make] [varchar](40) NULL,
    [Model] [varchar](40) NULL,
    [Color] [varchar](10) NULL,
    [InLot] [bit] NOT NULL,
    [SysStartTime] [datetime2](7) GENERATED ALWAYS AS ROW START NOT NULL,
    [SysEndTime] [datetime2](7) GENERATED ALWAYS AS ROW END NOT NULL,
PRIMARY KEY CLUSTERED
(
    [CarId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
    PERIOD FOR SYSTEM_TIME ([SysStartTime], [SysEndTime])
) ON [PRIMARY]
WITH
(
SYSTEM_VERSIONING = ON ( HISTORY_TABLE = [dbo].[CarInventoryHistory] )
)
GO
```

2.6.2 Temporal Query for Snapshot

```
private void ok_Click(object sender, EventArgs e)
{
    str = "SELECT [CarId],[Year],[Make] ,[Model],[Color],[InLot] FROM CarInventory FOR SYSTEM_TIME as of ";
    str += "'" + dateTimePicker1.Value.ToString("yyyy-MM-dd HH:mm") + ":59.9999999 ' order by CarId ";
    updateDataGrid();
}
```

In this query, it search the state of the cars at a specific time in the past. For this situation temporal queries has been used.

3. CONCLUSION

As a result, Rental Car system created successfully with temporal database elements. Past searches were carried out very simply thanks to the temporal databases so that, the system has become useful for people who works in car rental.