**AYSAN DATABASE PROJECT REPORT**

**Description:**

After establishing a database system for the AYSAN Car Rentals company, every customer will have the ability to view the availability of specific car models they wish to rent. Initially, we will utilize MSSQL to implement this database. Once we identify the relevant entities and their relationships, we will construct the database system. Subsequently, we will seamlessly integrate this database with the AYSAN Car Rentals website.

The database system is structured around various company-product relationships, featuring a user system enabling product purchases from AYSAN company.

**Scope:**

The Aysan Company can track rentals of their cars. They can see the details of the car rentals.

The database system includes complete lists of products and materials as per our requirements.

All product and material insert & update operations are handled.

* CUSTOMER: this entity includes customer info in detail
* CUSTOMER\_ADRESS: this entity includes customer address.
* CUSTOMER\_ORDER: this entity keeps order details.
* ORDER\_LINE: this entity keeps important and unique info of order.
* PAYMENT: this entity keeps details of payment.
* PAYMENT\_LINE: this entity keeps important and unique information of payment.
* REPORTS: this entity keeps customer requests and complaint.
* REPORTS\_LINE: this entity keeps important and unique information of report details.
* CAR: this entity includes car info in detail.
* CAR\_ORDER\_LINE: this entity keeps important information of cars.

Newly Added Tables:

-Customer\_Adresses

- Order\_line

-Payment\_line

-Car\_Order\_line

\_Reports\_line

Deleted Tables:

-Employee

**Data & Requirement Analysis:**

* **Ensure all data is properly formatted.**
* **Guarantee uniqueness of all data entries.**
* **Implement a user-friendly "sign-up" operation in the database system.**
* **Store comprehensive user information in the database.**
* **Manage and store personal details securely.**
* **Record information about company products within the database.**
* **Organize and store data related to company materials.**
* **Implement a robust system to store and manage orders.**
* **Develop the database using MSSQL.**

**Tables:**

* Customer :

🡪Columns and Data Types : Customer\_ID (int) , FirstName (nvarchar(40)) ,LastName (nvarchar(40)) ,PhoneNumber (nvarchar(11)) ,mail (nvarchar(40)), Gender (varchar(10))

🡪Definition : This table is created to store customer informations.

🡪

\*Customer\_ID is a primary key.

🡪

\*Customer\_ID and Phone\_number are unique keys.

\*Gender has constraint.

* Customer\_Address :

🡪Columns and Data Types : Province(nvarchar(40), District(nvarchar(40), Street(nvarchar(40), Aparment\_number(nvarchar(40)

🡪Definition: This table is created to store Customer Adresses.

🡪

\*Customer\_ID: Foreign key references customer.

* Customer\_Order :

🡪Columns and Data Types :Order\_ID(int), Start\_Date(smalldatetime), ReturnDate(smalldatetime)

🡪Definition : This table is created to store customer order.

🡪

\*Order\_ID: Primary key and identity

🡪

StartDate: Default variable

•Order\_Line :

🡪Columns and Data Types: Customer\_ID (int), Order\_ID(int)

🡪Definition : This table is created to store customer and order relation.

🡪

\* Customer\_ID and Order\_ID are foreign keys.

🡪Trigger : AfterInsertOrderLine

* Car :

🡪Columns and Data Types: LicencePlat(nvarchar(40), Color(nvarchar(10), brand(nvarchar(30), fueltype(nvarchar(20), price(int), availableStartDate(smalldatetime), availableEndDateTime(smalldatetime)

🡪Definition : This table is created to store information of cars.

🡪

\*Primary key: Licence\_plat

\*Nonclustered index: idx\_PriceNonClustered.

• Payment :

🡪Columns and Data Types: Bill\_ID(int), OrderDate(smalldatetime), orderamount(int), paymentType(nvarchar(20), taxAmount(int)

🡪Definition : This table is created to store payment information.

🡪

\*Bill\_ID is a primary key.

\*Tax\_amount is a computed Column.

• Payment\_Line :

🡪Columns and Data Types: Order\_ID(int), Bill\_ID(int), Customer\_ID(int)

🡪Definition : This table is created to store important payment informations.

🡪

\*Order\_ID, Bill\_ID and Customer\_ID are foreign keys

• Car\_Order\_Line :

🡪Columns and Data Types: Order\_ID(int), Licence\_Plat(nvarchar(40)), Car\_type(nvarchar(25))

🡪Definition : This table is created to store important car informations.

🡪

\*Order\_ID and Licence\_Plat are foreign keys

🡪Trigger : UpdateCarStatus

• Reports :

🡪Columns and Data Types: Reports\_ID(int), ReportsContext(nvarchar(200)), Reports\_date(smalldatetime)

🡪Definition : This table is created to store Customer request and complaints.

🡪

\*Reports\_ID is a identity keys

\* Reports\_date is a default variable

• Reports\_Line :

🡪Columns and Data Types: Reports\_ID(int), Customer\_ID(int)

🡪Definition : This table is created to store important Customer reports informations.

🡪

\*Reports\_ID is a foreign keys

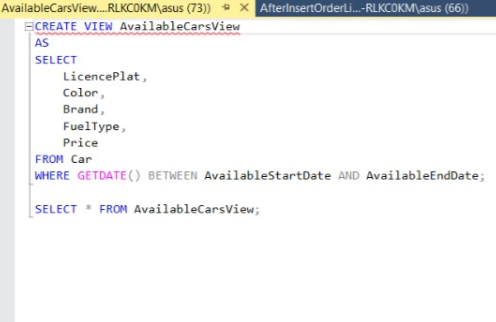
**Views:**

\*AvailableCarsView

Definition: This is the view of all available cars

Screenshots:

Code:



Output:

A screenshot of a computer

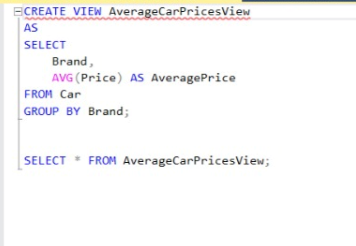
Description automatically generated

\*AverageCarPricesView

Definition: This is the view of average car prices related to brands

Screenshots:

Input:



Output:

A screenshot of a computer

Description automatically generated

\* CarCountByFuelTypeView

Definition:this is to view of Car number of same fuel type

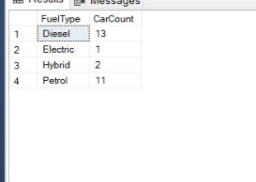
Screenshots:

Code:

A screenshot of a computer program

Description automatically generated

Output:



\*CustomerCountByProvinceView

Definition: this is to view of Customer number of same province

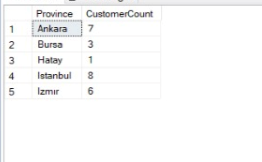
Screenshots:

Code:

A screenshot of a computer

Description automatically generated

Output:



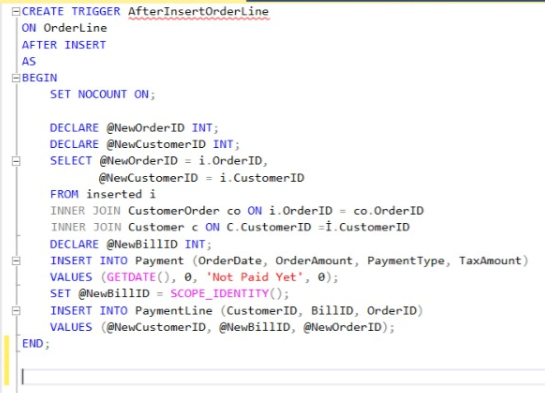
**Triggers:**

1. AfterInsertOrderLine

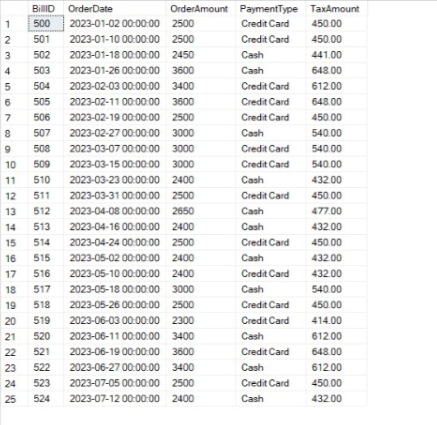
Definition: The purpose of this trigger is to automatically fill in the payment section after placing an order.

Screenshots:

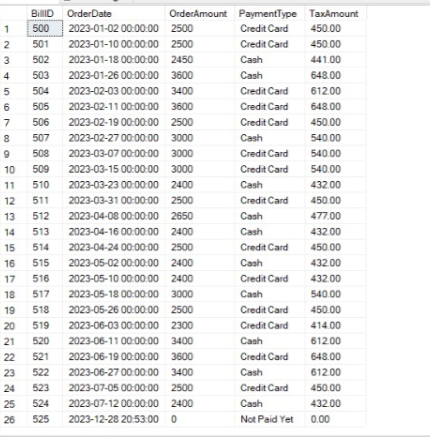
Code:



Before Data:



After Data:



1. **A tr\_UpdateCarStatus**

Definition: The purpose of this trigger is to update the available dates of the vehicle to null after the vehicle order is placed.

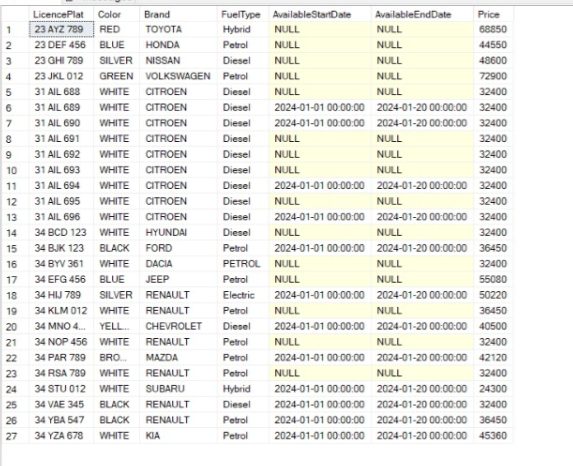
Screenshots:

Code:

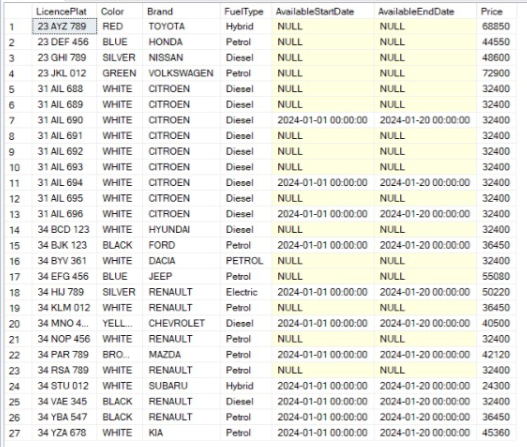
A screenshot of a computer program

Description automatically generated

Before Data:



After Data:



**Stored Procedures:**

1. **A DecreaseCarPrices**

Definition: This procedure allows us to reduce car prices by 10% as each year passes

Screenshots:

Code:

A screenshot of a computer program

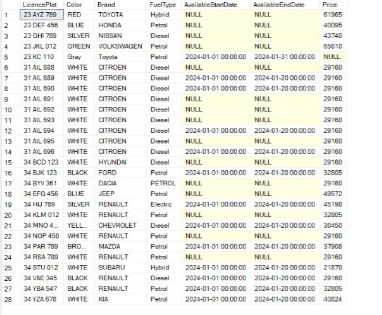
Description automatically generated

Before Data:

A screenshot of a computer

Description automatically generated

After Data:



1. **A DeleteCustomerOrder**

Definition: This procedure allows us to delete the user order from the system

Screenshots:

Code:

A computer screen shot of a code

Description automatically generated

Data:

A screenshot of a computer

Description automatically generated

1. **A DeleteCustomer**

Definition: : This procedure allows us to delete customers from the system.

Screenshots:

Code:

A screenshot of a computer program

Description automatically generated

Before Data:

A screenshot of a computer

Description automatically generated

After Data:

A screenshot of a computer

Description automatically generated

1. **A GetCustomerOrderData**

Definition: This procedure allows us to retrieve all customer information.

Screenshots:

Code:

A screenshot of a computer

Description automatically generated

Data:

A screenshot of a computer

Description automatically generated

1. **A UpdateCustomer**

Definition: This procedure allows us to update customers

Screenshots:

Code:

A screenshot of a computer program

Description automatically generated

Before Data:

A screenshot of a computer

Description automatically generated

After Data:

A screenshot of a computer

Description automatically generated

1. **A UpdatePayment**

Definition: This procedure allows us to update the payment

Screenshots:

Code:

A screenshot of a computer

Description automatically generated

Before Data:

A screenshot of a computer

Description automatically generated

After Data:

A screenshot of a credit card

Description automatically generated

1. **A AddCustomerOrder**

Definition: This procedure allows us to add a sales order

Screenshots:

Code:

A screenshot of a computer

Description automatically generated

Before Data:

A screenshot of a computer

Description automatically generated

After Data:

A screenshot of a computer

Description automatically generated

1. **A CalculateMaintenanceCost**

Definition: This procedure allows us to calculate the annual maintenance cost

Screenshots:

Code:

A screenshot of a computer

Description automatically generated

Data:



1. **A CalculateTotalTaxForAllPayments**

Definition: This procedure allows us to calculate the installment prices of all payments

Screenshots:

Code:

A computer screen shot of a computer code

Description automatically generated

Data:



1. **A AddCar**

Definition: : This procedure allows us to add vehicles by adding license plates to them.

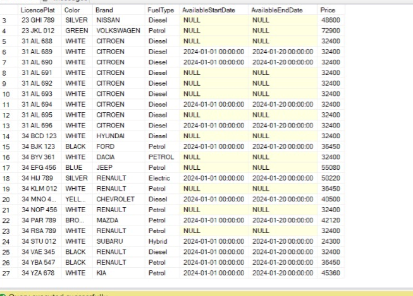
Screenshots:

Code:

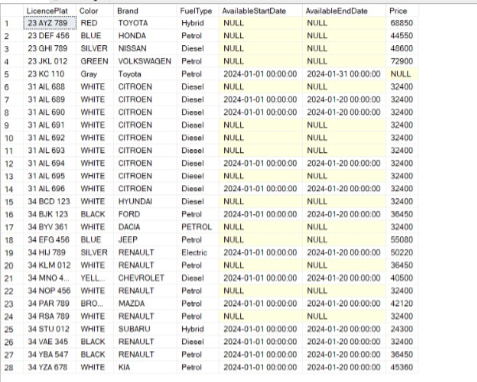
A screenshot of a computer

Description automatically generated

Before Data:



After Data:



A close-up of a number

Description automatically generated

**Developers:**

Hasan Özeren – 150121036 ( Gr Rep)

Umut Bayar – 150120043

Ahmet Arda Nalbant – 150121004

Berat Demirhan – 150119866