STORED PROCEDURES:

1) Stored Procedure “getremainingamount”:

Final Queries.sql...F2SDD96\panch (64))

```sql
CREATE PROCEDURE getremainingamount
@customerid int,
@remaining_payment int output
as
BEGIN

WITH

t1 AS (
Select VehicleID as VehicleID,CustomerID as CustomerID,SUM(Amount) as total_pay
from PaymentInfo
GROUP BY VehicleID ,CustomerID
HAVING CustomerID = @customerid),

t2 AS (

Select t1.VehicleID as vid,t1.CustomerID as cid,t1.total_pay as total_pay,v.VehicleType as vt ,v.VehicleModel as vm,v.Brand as brand,v.Co
FROM t1
JOIN Vehicle v
ON t1.VehicleID = v.VehicleID AND t1.CustomerID = @customerid)

SELECT @remaining_payment =  (t2.Price-t2.total_pay)
FROM t2;
END

DECLARE @remp int
EXECUTE getremainingamount 16828,@remaining_payment = @remp OUT
SELECT @remp as RemainingPayment
```

100 %

Results | Messages

	RemainingPayment
1	2500

Query executed successfully. | DESKTOP-F2SDD96 (14.0 RTM) | DESKTOP-F2SDD96\panch ... | VehicleVendingMachine | 00:00:00 | 1 rows

2) Stored Procedure “ServiceCenter”:

Final Queries.sql...F2SDD96\panch (64))

```sql
SELECT @remp as RemainingPayment

---------------Stored Procedure to check if vehicle is serviced at service center or vending maching------------
create procedure ServiceCenter @VehicleID int,@message varchar(100) output
as
begin
if not exists (select VehicleID from Vehicle where VehicleID=@VehicleID)
set @message='VehicleID is invalid'
else if exists (select CustomerID,s.ServiceCenterID,Brand,v.VehicleModel from Vehicle v join [Service] s on v.VehicleID=s.VehicleID
where s.ServiceCenterID IS NOT NULL AND s.VehicleID=@VehicleID)
set @message='Vehicle serviced from Service Center'

else set @message='Vehicle serviced from Vending machine'
end

declare @result varchar(100)
exec ServiceCenter 1327724,@result output
print @result

-----------------Stored Procedure to check for Vehicle Store Pickup or Vehicle Delivery----------------------
CREATE procedure DeliveryStatus  @CustomerID int, @VehicleID int output, @CustomerName varchar(100) output,
@Delivery_Status varchar(50) output
as
begin
set @VehicleID= (select v.VehicleID from Vehicle v join Customer c on v.CustomerID=c.CustomerID where v.CustomerID=@CustomerID)
set @CustomerName= (select c.CustomerName from Customer c join Vehicle v on v.CustomerID= c.CustomerID where v.CustomerID=@CustomerID)
set @Delivery_Status= (select Case when VehicleID in (Select d.VehicleID from Delivery d) then 'Delivery'
    else 'Pickup' end from Vehicle v join Customer c on v.CustomerID=c.CustomerID where v.CustomerID=@CustomerID )
end
```

100 %

Messages

```
Vehicle serviced from Service Center

Completion time: 2019-11-19T20:08:16.3811627-05:00
```

100 %

Query executed successfully. | DESKTOP-F2SDD96 (14.0 RTM) | DESKTOP-F2SDD96\panch ... | VehicleVendingMachine | 00:00:00 | 0 rows

3) Stored Procedure “DeliveryStatus”:

Final Queries.sql...F2SDD96\panch (64))

```sql
declare @result varchar(100)
exec ServiceCenter 1327724,@result output
print @result

----------------Stored Procedure to check for Vehicle Store Pickup or Vehicle Delivery----------------------
CREATE procedure DeliveryStatus  @CustomerID int, @VehicleID int output, @CustomerName varchar(100) output,
@Delivery_Status varchar(50) output
as
begin
set @VehicleID= (select v.VehicleID from Vehicle v join Customer c on v.CustomerID=c.CustomerID where v.CustomerID=@CustomerID)
set @CustomerName= (select c.CustomerName from Customer c join Vehicle v on v.CustomerID= c.CustomerID where v.CustomerID=@CustomerID)
set @Delivery_Status= (select Case when VehicleID in (Select d.VehicleID from Delivery d) then 'Delivery'
    else 'Pickup' end from Vehicle v join Customer c on v.CustomerID=c.CustomerID where v.CustomerID=@CustomerID )
end

declare @VehicleID int, @CustomerName varchar(50), @Delivery_Status varchar(50)
exec DeliveryStatus 53044, @VehicleID output, @CustomerName output, @Delivery_Status output
select @VehicleID as VehicleID, @CustomerName as Customer_Name, @Delivery_Status as Delivery_Status


-------------------------------Stored Procedure to retrieve Safety Rating for Vehicles--------------------------------
create procedure VehicleRating @VehicleModel varchar(50),@Rating float output, @VehicleName varchar(50) output
as
begin
set @Rating=(select case when VehicleModel = 'A6 Premium' then 3.5
                         when VehicleModel = 'Acadia Denali Sport Utility' then 5
                         when VehicleModel = 'Beetle 1.8T Dune' then 4
                         when VehicleModel = 'Camaro Z/28' then 3
                         when VehicleModel = 'Colorado Crew Cab' then 5
```

100 %

Results | Messages

	VehicleID	Customer_Name	Delivery_Status
1	1219315	James Butt	Pickup

Query executed successfully. | DESKTOP-F2SDD96 (14.0 RTM) | DESKTOP-F2SDD96\panch ... | VehicleVendingMachine | 00:00:00 | 1 rows

4) Stored Procedure “VehicleRating”:

Final Queries.sql...F2SDD96\panch (64))

```sql
--------------------------------Stored Procedure to retrieve Safety Rating for Vehicles--------------------------------
create procedure VehicleRating @VehicleModel varchar(50),@Rating float output, @VehicleName varchar(50) output
as
begin
set @Rating=(select case when VehicleModel = 'A6 Premium' then 3.5
                         when VehicleModel = 'Acadia Denali Sport Utility' then 5
                         when VehicleModel = 'Beetle 1.8T Dune' then 4
                         when VehicleModel = 'Camaro Z/28' then 3
                         when VehicleModel = 'Colorado Crew Cab' then 5
                         when VehicleModel = 'Durango SXT' then 2.5
                         when VehicleModel = 'F350 Super Duty' then 4.5
                         when VehicleModel = 'Forte5 LX' then 4
                         when VehicleModel = 'GLK-Class 350' then 5
                         when VehicleModel = 'LX Sport Utility' then  5
                         when VehicleModel = 'Malibu LT Sedan' then 4
                         when VehicleModel = 'Maxima S' then  3.5
                         when VehicleModel = 'RC 350 Coupe 2D' then 3.5
                         when VehicleModel = 'Wrangler Sport' then 3
                         when VehicleModel = 'XE 25t' then 5
                         end from Vehicle where VehicleModel like '%' +@VehicleModel+ '%'
                         )
set @VehicleName= (select VehicleModel from Vehicle where VehicleModel like '%' +@VehicleModel+ '%')
end

declare @VehicleRating float, @VehicleName varchar(50)
exec VehicleRating 'A6', @VehicleRating output, @VehicleName output
select @VehicleName as Model, @VehicleRating as [Safety Rating]
```

100 %

Results | Messages

	Model	Safety Rating
1	A6 Premium	3.5

Query executed successfully. | DESKTOP-F2SDD96 (14.0 RTM) | DESKTOP-F2SDD96\panch ... | VehicleVendingMachine | 00:00:00 | 1 rows

FUNCTIONS:

1) Function “GetVehicle”:

Final Queries.sql...F2SDD96\panch (64))

```sql
---------------Function to return Available cars based on Car type--------------------------------
CREATE function GetVehicle(@Vehicle_Type varchar(200))
returns @Table table (VehicleID int,Brand varchar(50),Mileage float,Color varchar(50),VehicleModel varchar(100),Price float)

As
Begin
 insert into @Table
 select VehicleID,Brand,Mileage,Color,VehicleModel,Price from Vehicle where VehicleType=@Vehicle_Type
return
End


select * from dbo.GetVehicle('Sedan') order by Price

---------------Function to return Part Details for a Particular Vehicle--------------------------------
create function GetPartDetails(@Part_no int)
returns @Table table (Part_No int,VehicleID int,Vehicle varchar(100),Description varchar(200),WarrantyYears int)
as
```

100 %

Results | Messages

	VehicleID	Brand	Mileage	Color	VehicleModel	Price
1	1330631	Jaguar	21	White	XE 25t	19300
2	1284360	Audi	19	Grey	A6 Premium	25600
3	1329125	Chevrolet	24	Blue	Malibu LT Sedan	38400
4	1324170	Nissan	23	Burgundy	Maxima S	46600

Query executed successfully. | DESKTOP-F2SDD96 (14.0 RTM) | DESKTOP-F2SDD96\panch ... | VehicleVendingMachine | 00:00:00 | 4 rows

2) Function “GetPartDetails”:

Final Queries.sql...F2SDD96\panch (64))

```
---------------Function to return Part Details for a Particular Vehicle---------------------------------
create function GetPartDetails(@Part_no int)
returns @Table table (Part_No int,VehicleID int,Vehicle varchar(100),Description varchar(200),WarrantyYears int)
as
begin
insert into @Table
select a.Part_no,v.VehicleID,v.Brand+' '+v.VehicleModel as Vehicle,a.[Description],w.Warranty_Period from Accessories a join Vehicle v on
v.VehicleID=a.VehicleID join Part_Warranty w on a.Part_no=w.Part_no where a.Part_no=@Part_no
return
end

select * from dbo.GetPartDetails(361308)

/*******************************Views*******************************/

---------------------------------Vehicle Count of each Customer---------------------------
Create VIEW VehicleCountbyCustomer
as
select v.CustomerID,v.VehicleModel, COUNT(v.VehicleModel)  OVER (PARTITION BY v.CustomerID Order by v.VehicleModel desc) as [count]
```

100 %

Results | Messages

	Part_No	VehicleID	Vehicle	Description	WarrantyYears
1	361308	1322951	Dodge Durango SXT	Front Left Interior Door Handle	2

Query executed successfully. | DESKTOP-F2SDD96 (14.0 RTM) | DESKTOP-F2SDD96\panch ... | VehicleVendingMachine | 00:00:00 | 1 rows

Views

1) Count of Vehicle by Customer

Final Queries.sql -...ARPIT-PC\gulat (56)) | Views.sql - not connected

```
end

select * from dbo.GetPartDetails(361308)

/*********************************Views*********************************/

----------------------------------Vehicle Count of each Customer---------------------------
Create VIEW VehicleCountbyCustomer
as
select v.CustomerID,v.VehicleModel, COUNT(v.VehicleModel)  OVER (PARTITION BY v.CustomerID Order by v.Ve
from Vehicle v


select *
from VehicleCountbyCustomer
```

109 %

Results | Messages

	CustomerID	VehicleModel	count
1	16828	GLK-Class 350	1
2	18250	XE 25t	1
3	32101	Beetle 1.8T Dune	1
4	42939	F350 Super Duty	1
5	42939	Durango SXT	2
6	42977	Forte5 LX	1
7	44879	A6 Premium	1
8	53044	LX Sport Utility	1
9	53898	RC 350 Coupe 2D	1
10	57870	Maxima S	1
11	66671	Colorado Crew Cab	1
12	89824	Malibu LT Sedan	1
13	89824	Camaro Z/28	2
14	93429	Acadia Denali Sport Utility	1
15	98598	Wrangler Sport	1

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 15 rows

2) Vehicle Warranty End Date of each Vehicle

Final Queries.sql -...ARPIT-PC\gulat (56)) | Views.sql - not connected

```sql
select *
from VehicleCountbyCustomer

--------------------------Vehicle Warranty End Date of each Vehicle------------------------

create view WarrantyEnd as
select w.VehicleID, v.ManufacturedYear, w.Date_of_Purchase, (DATEADD(yyyy,Warranty_Period,Date_of_Purcha
from Vehicle v join Vehicle_Warranty w
on v.VehicleID=w.VehicleID


Select * from WarrantyEnd

-------------------------Vehicle Count by Fuel Type------------------
create view VehicleCountByFuelType as
```

109 %

Results | Messages

	VehicleID	ManufacturedYear	Date_of_Purchase	WarrantyEndDate
1	1284360	2016	2017-01-25	2018-01-25
2	1315301	2014	2015-11-18	2017-11-18
3	1219315	2017	2020-06-25	2022-06-25
4	1324764	2018	2019-02-07	2020-02-07
5	1327724	2013	2014-06-09	2017-06-09
6	1329125	2018	2018-12-01	2019-12-01
7	1268666	2016	2017-06-04	2018-06-04
8	1322951	2018	2019-10-08	2021-10-08
9	1328639	2017	2017-07-28	2019-07-28
10	1324170	2014	2016-12-12	2017-12-12
11	1321233	2016	2018-09-30	2020-09-30
12	1333281	2014	2016-11-10	2018-11-10
13	1326245	2016	2016-04-01	2018-04-01
14	1330631	2018	2020-07-18	2021-07-18
15	1290071	2015	2016-02-06	2018-02-06

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 15 rows

3) Vehicle Count based on Fuel Type

Final Queries.sql -...ARPIT-PC\gulat (56)) | Views.sql - not connected

```sql
create view WarrantyEnd as
select w.VehicleID, v.ManufacturedYear, w.Date_of_Purchase, (DATEADD(yyyy,Warranty_Period,Date_of_Purcha
from Vehicle v join Vehicle_Warranty w
on v.VehicleID=w.VehicleID


Select * from WarrantyEnd

-------------------------Vehicle Count by Fuel Type------------------
create view VehicleCountByFuelType as
select FuelType,count(FuelType) as VehicleCount from Vehicle
group by FuelType

select * from VehicleCountByFuelType
```

109 %

Results | Messages

	FuelType	VehicleCount
1	Diesel	5
2	Petrol	10

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 2 rows

Triggers

1) Trigger to track payment made by Customers

Final Queries.sql -...ARPIT-PC\gulat (56)) | Views.sql - not connected

```sql
/*******************************Triggers*******************************/

---------------------------Trigger to track payment made by customer---------------------------
create TRIGGER tr_new_payment
ON PaymentInfo
FOR INSERT
AS
BEGIN

DECLARE @customer_id int, @amount int, @mop varchar(50)
SELECT @customer_id =  CustomerID, @amount = Amount, @mop = ModeOfPayment FROM inserted


INSERT INTO LogInfo
VALUES('customer with id = ' + CAST(@customer_id as nvarchar(5)) + ' has made a payment of ' + CAST(@amo
```

109 %

Results | Messages

	PaymentID	CustomerID	VehicleID	ModeOfPayment	Amount
1	13115111	66671	1321233	Credit/Debit Card	3200
2	15138381	42939	1315301	Internet Banking	12000
3	17023639	53044	1219315	Internet Banking	5000
4	30598045	89824	1290071	Cheque	15000
5	34978835	89824	1290071	Internet Banking	11600
6	37574127	16828	1327724	Credit/Debit Card	6400
7	39139063	18250	1330631	Cheque	5600
8	42396204	18250	1330631	Cheque	12000
9	44217061	89824	1290071	Internet Banking	20000
10	54704707	66671	1321233	Credit/Debit Card	8000

	id	message
1	1	customer with id = 66671 has made a payment of 5...

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 25 rows

2) Trigger to Track Accessory addition for vehicles

Final Queries.sql -...ARPIT-PC\gulat (56)) | Views.sql - not connected

```sql
DECLARE @part_no int, @vehicle_id int
SELECT @part_no =  Part_No, @vehicle_id = VehicleID FROM inserted


INSERT INTO Part_Warranty
VALUES(@part_no,@vehicle_id,GETDATE(),2,'Carburetor')


end

select * from Part_Warranty
select * from Accessories

INSERT INTO Accessories values (1112,1290071,50,'Motor Seasons', 'Bumper')

/*********************************Column Data Encryption*********************************/
```

109 %

Results | Messages

	Part_no	VehicleID	Date_of_Purchase	Warranty_Period	Description
1	13242	1219315	2017-08-23	3	Front Seat Cover with Custom Fit Poly-Cotton
2	11113	1324764	2018-07-12	2	Front Door Hinge
3	56107	1328639	2017-02-08	1	Right Door Mirror Glass
4	36019	1329125	2018-05-03	3	Standard Ignition Parking Aid Sensor
5	34794	1333281	2015-12-13	2	Steering Wheel
6	74016	1327724	2014-08-26	3	Accelerator Pedal Bushing
7	84738	1284360	2016-04-18	2	Sunroof Motor
8	76950	1290071	2015-05-20	1	Windshield Wiper Control Knob

	Part_no	VehicleID	Price	Manufacturer	Description
1	1111	1330631	50	Four Seasons	Windshield
2	1112	1290071	50	Motor Seasons	Bumper
3	6434	1330631	27	RhinoPac	Trunk Lid Lift Support
4	11113	1324764	69.99	Rugged Ridge	Front Door Hinge
5	13242	1219315	131	Rugged Ridge	Front Seat Cover wi...

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 25 rows

Computed Data Encryption

Column Data Encryption on Credit Score of Customer

Final Queries.sql -...ARPIT-PC\gulat (56)) | Views.sql - not connected

```sql
CREATE CERTIFICATE Customercert
WITH SUBJECT = 'Customer Credit Score'

CREATE SYMMETRIC KEY Credit_Score_Key
WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE Customercert;

OPEN SYMMETRIC KEY Credit_Score_Key
DECRYPTION BY CERTIFICATE Customercert;

select * from Finance
-- Encrypt credit score and add in Credit Score Encrypted
UPDATE Finance
SET Credit_Score_Encrypted = EncryptByKey(Key_GUID('Credit_Score_Key'),CONVERT(nvarchar,Credit_Score))

-- Decrypt credit score and show as Decrypted Score
OPEN SYMMETRIC KEY Credit_Score_Key
DECRYPTION BY CERTIFICATE Customercert;
select CustomerID, Annual_Income,Credit_Score_Encrypted, CONVERT(nvarchar, DecryptByKey(Credit_Score_Enc
```

109 %

Results | Messages

	CustomerID	Annual_Income	Credit_Score_Encrypted	Decrypted Credit Score
1	53044	80000	0x0084D789B71A8E49A2BF6730A1F70F4002000000F5A4F10...	650
2	53898	125000	0x0084D789B71A8E49A2BF6730A1F70F4002000000A5F9630...	745
3	93429	76000	0x0084D789B71A8E49A2BF6730A1F70F4002000000DB90EA...	670
4	89824	155000	0x0084D789B71A8E49A2BF6730A1F70F40020000008CC927C...	780
5	98598	60000	0x0084D789B71A8E49A2BF6730A1F70F400200000044B1CDC...	680
6	16828	150000	0x0084D789B71A8E49A2BF6730A1F70F40020000009FC9A0B...	760
7	42939	180000	0x0084D789B71A8E49A2BF6730A1F70F400200000099D34B1...	800
8	18250	200000	0x0084D789B71A8E49A2BF6730A1F70F40020000005C0C617...	820
9	57870	80000	0x0084D789B71A8E49A2BF6730A1F70F40020000009C68CFE...	650
10	66671	140000	0x0084D789B71A8E49A2BF6730A1F70F4002000000617EEF0...	770
11	42939	200000	0x0084D789B71A8E49A2BF6730A1F70F4002000000D4FECA...	840

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 13 rows

Computed Columns

Compute Discount and Effective Price Values in Vehicle Table

Final Queries.sql -...ARPIT-PC\gulat (56))* | Views.sql - not connected

```sql
/*********************************Computed Columns*********************************/

------------------------Compute Discount and Effective Price Values in Vehicle Table--------------------
Create View DiscountedPrice as
select v.VehicleID, v.VehicleType, concat(v.Brand, v.VehicleModel) as Vehicle, v.Price, v.FuelType,
(case when (v.Price<20000) then '10%'
        when (v.Price>20000 and v.Price<30000) then '15%'
        else '20%'
        end) as Discount,
(case when (v.Price<20000) then (v.Price-(v.Price*10/100))
when (v.Price>20000 and v.Price<30000) then (v.Price-(v.Price*15/100))
else (v.Price-(v.Price*20/100)) end) as Effective_Price
from Vehicle v
select * from DiscountedPrice
```

109 %

Results | Messages

	CustomerID	Annual_Income	Credit_Score_Encrypted	Decrypted Credit Score
1	53044	80000	0x0084D789B71A8E49A2BF6730A1F70F4002000000F5A4F10...	650
2	53898	125000	0x0084D789B71A8E49A2BF6730A1F70F4002000000A5F9630...	745
3	93429	76000	0x0084D789B71A8E49A2BF6730A1F70F4002000000DB90EA...	670
4	89824	155000	0x0084D789B71A8E49A2BF6730A1F70F40020000008CC927C...	780
5	98598	60000	0x0084D789B71A8E49A2BF6730A1F70F400200000044B1CDC...	680
6	16828	150000	0x0084D789B71A8E49A2BF6730A1F70F40020000009FC9A0B...	760
7	42939	180000	0x0084D789B71A8E49A2BF6730A1F70F400200000099D34B1...	800
8	18250	200000	0x0084D789B71A8E49A2BF6730A1F70F40020000005C0C617...	820
9	57870	80000	0x0084D789B71A8E49A2BF6730A1F70F40020000009C68CFE...	650
10	66671	140000	0x0084D789B71A8E49A2BF6730A1F70F4002000000617EEF0...	770
11	42939	200000	0x0084D789B71A8E49A2BF6730A1F70F4002000000D4FECA...	840

Query executed successfully. | ARPIT-PC (14.0 RTM) | ARPIT-PC\gulat (56) | VehicleVendingMachine | 00:00:00 | 13 rows