

CAR RENTAL SYSTEM

SOFTWARE ENGINEERING

SIS1012Y

21 JULY 2022

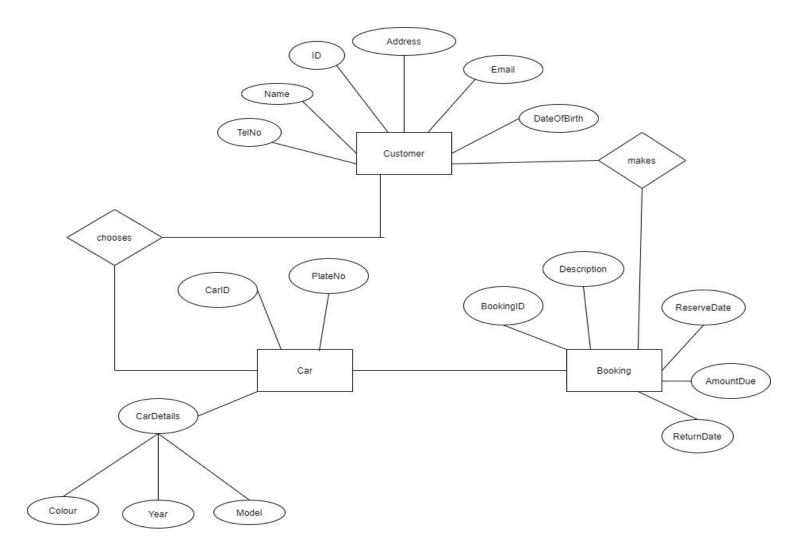
Students:

Chumbit Khilesh	ID:2117855
Veerapen Chetty Sarah	ID:2116427
Beerachee Deesha	ID:2117144
Peeharry Geerna	ID:2116307
Bholah Mohammad Yassir	ID:2114519

Submit to:

Mrs Soulakshmee Ghurbhurrun Nagowah

1. CREATE AN ER DIAGRAM FOR YOUR SYSTEM



2. NORMALISATION OF TABLES

Assumptions:

- 1. A customer has a unique CustomerID.
- 2. A customer can choose many cars

0NF

CarRental(CustomerID,Name,Address,Email,TelNo,DateofBirth,CarID,PlateNo,Colour,Year,Model,SeatingCapacity,Condition,BookingID,Descprition,ReserveDate,ReturnDate,AmountDue)

<u>1NF:</u>

CarRental(<u>CustomerID</u>, Name, Address, Email, TelNo, DateofBirth, BookingID, Description, Reser veDate, ReturnDate, AmountDue(CarID, PlateNo, Colour, Year, Model, Seating Capacity, Condition))

FUNCTIONAL DEPENDENCIES

CustomerID--->

Name, Address, Email, TelNo, Date Of Birth, Booking ID, Description, Reserve Date, Return Date, Amount Due and Company of the Company of th

CarID ---> PlateNo, Colour, Year, Model, SeatingCapacity, Condition

2NF:

 $\label{lem:customer} \textbf{Customer}(\underline{\textbf{Customer}ID}, \textbf{Name}, \textbf{Address}, \textbf{Email}, \textbf{TelNO}, \textbf{DateOfBirth}, \textbf{BookingID}, \textbf{Description}, \textbf{ReserveDate}, \textbf{ReturnDate}, \textbf{AmountDue})$

Car(CarID, PlateNo, Colour, Year, Model, Seating Capacity, Condition)

CustomerCar (CustomerID,CarID)

PARTIAL DEPENDENCIES

BookingID ---> Description, ReserveDate, ReturnDate, AmountDue

3NF:

Customer(CustomerID,Name,Address,Email,TelNO,DateOfBirth,BookingID)

Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue)

Car (CarID, PlateNo, Colour, Year, Model, SeatingCapacity, Condition)

CustomerCar (CustomerID, CarID)

3. IMPLEMENT THE DATABASE USING SQL SERVER

CUSTOMER TABLE

CustomerID	Name	Address	Email	TelNo	DateOfBirth
C001	Selena	Triolet	Selenagomez@gmail.com	57810028	10/03/2001
	Gomez				
C002	James	Montagne	Jamesnicklon@gmail.com	58266754	02/02/1990
	Nicklon	Blanche			
C003	Ariana	Curepipe	Arianagrande@gmail.com	57774002	29/01/1995
	Grande				
C004	Iana	Port	Ianabugs@gmail.com	57774002	29/01/1995
	Bugs	Louis			
C005	Jean	Vacoas	Jeanlouis@gmail.com	59740002	09/05/1975
	Louis				
C006	Tania	Quatre	Taniabeckham@gmail.com	57442905	07/02/1999
	Beckham	Bornes			

BOOKING TABLE

BookingID	Description	ReserveDate	ReturnDate	AmountDue
B001	Online Payment	17/01/2006	27/01/2006	2500
B002	Pay when	05/02/2009	25/02/2009	3600
	fetching car			
B003	Pay when	06/06/2015	17/06/2015	3000
	fetching car			
B004	Online Payment	18/07/2020	30/07/2020	4200
B005	Online Payment	01/12/2021	28/12/2021	6900
B006	Pay when	15/04/2022	20/04/2022	2000
	fetching car			

CAR TABLE

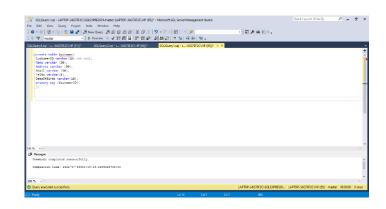
CusromerID	CarID	PlateNo	Colour	Year	Model	SeatingCapacity	Condition
C001	CR1	G1115	Black	2020	Mercedes	4	Very
							Good
C002	CR2	J1132	Brown	2019	BMW	5	Very
							Good
C003	CR3	CR3	Red	2017	Minicooper	5	Very
							Good
C004	CR4	CR4	Blue	2021	Audi	5	Excellent
C005	CR5	CR5	White	2005	Toyota	4	Good
C006	CR6	CR6	Silver	2016	Range Rover	6	Very
							Good

4. IMPLEMENT THE DATABASE USING SQL SERVER

SQL codes for creating tables:

Customer

```
create table Customer(
CustomerID varchar(10) not null,
Name varchar (30),
Address varchar (50),
Email varchar (50),
TelNo varchar(8),
DateOfBirth varchar(10),
primary key (CustomerID)
);
```



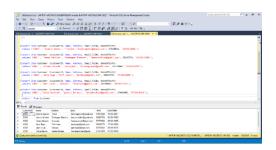
```
insert into Customer (CustomerID, Name, Address, Email, TelNo, DateOfBirth)
values ('C001', 'Selena Gomez', 'Triolet', 'Selenagomez@gmail.com', 57810028, '10/03/2001');
insert into Customer (CustomerID, Name, Address, Email, TelNo, DateOfBirth)
values('C002', 'James Nicklon', 'Montagne Blanche', 'Jamesnicklon@gmail.com', 58266754, '02/02/1990');
insert into Customer (CustomerID, Name, Address, Email, TelNo, DateOfBirth)
values('C003', 'Ariana Grande', 'Curepipe', 'Arianagrande@gmail.com', 57774002, '29/01/1995');
insert into Customer (CustomerID, Name, Address, Email, TelNo, DateOfBirth)
values ('C004', 'Jana Bugs', 'Port Louis', 'Ianabugs@gmail.com', 58159774, '17/04/2000');
insert into Customer (CustomerID, Name, Address, Email, TelNo, DateOfBirth)
values('C005', 'Jean Louis', 'Vacoas', 'Jeanlouis@gmail.com', 59740002, '09/05/1975');
```

insert into Customer (CustomerID, Name, Address, Email,TelNo, DateOfBirth) values('C006','Tania Beckham', 'Quatre Bornes', 'Taniabeckham@gmail.com', 57442905, '07/02/1999');



To view Customer table

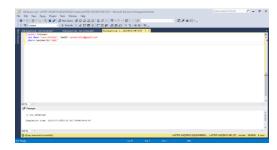
select * from Customer



James Nicklon to Lana Nicklon & Jamesnicklon@gmail.com to Lananicklon@gmail.com

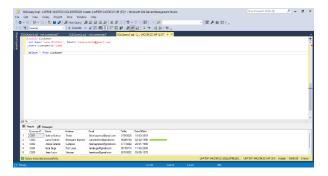
update Customer

set Name='Lana Nicklon', Email='Lananicklon@gmail.com' where CustomerID='C002'



To view table for update

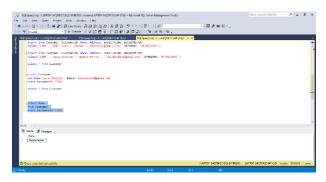
select * from Customer



select Name

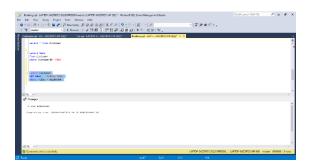
from Customer

where CustomerID='C001'



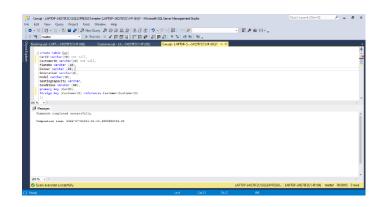
update Customer

set Name = 'Andrew Tate' where TelNo = 58266754;



Car

```
create table Car(
CarID varchar(10) not null,
CustomerID varchar(10) not null,
PlateNo varchar (10),
Colour varchar (20),
Generation varchar(4),
Model varchar(30),
SeatingCapacity varchar,
Condition varchar (50),
CustomerID varchar (10),
primary key (CarID),
foreign key (CustomerID)
references Customer(CustomerID)
);
```



insert into

Car(CustomerID,CarID,PlateNo,Colour,Generation,Model,SeatingCapacity,Condition) values('C001','CR1', 'G1115', 'Black', 2020, 'Mercedes', 4, 'Very Good');

insert into

Car(CustomerID,CarID,PlateNo,Colour,Generation,Model,SeatingCapacity,Condition) values('C002','CR2', 'J1132', 'Brown', 2019, 'BMW', 5, 'Very Good');

insert into

Car(CustomerID,CarID,PlateNo,Colour,Generation,Model,SeatingCapacity,Condition) values('C003','CR3', 'A2810', 'Red', 2017, 'Minicooper', 5, 'Very Good');

insert into

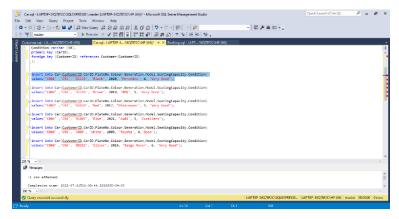
Car(CustomerID, CarID, PlateNo, Colour, Generation, Model, Seating Capacity, Condition) values ('C004', 'CR4', 'R1007', 'Blue', 2021, 'Audi', 5, 'Excellent');

insert into

Car(CustomerID, CarID, PlateNo, Colour, Generation, Model, Seating Capacity, Condition) values ('C005', 'CR5', 'K005', 'White', 2005, 'Toyota', 4, 'Good');

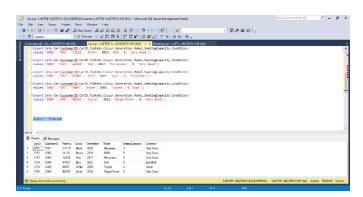
insert into

Car(CustomerID, CarID, PlateNo, Colour, Generation, Model, Seating Capacity, Condition) values ('C006', 'CR6', 'B0252', 'Silver', 2016, 'Range Rover', 6, 'Very Good');



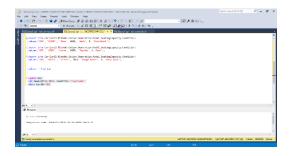
To view Car table

select * from Car



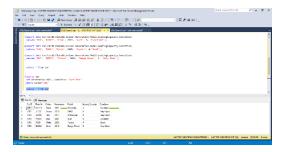
Generation from 2020 to 2021 and Condition to 'Excellent' from 'Very Good' update Car

set Generation=2021, Condition='Excellent' where CarID='CR1'



To view updated table

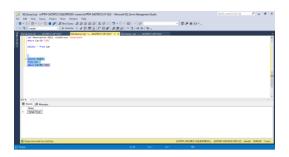
select * from Car



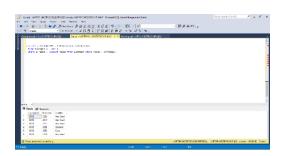
select Model

from Car

where CarID='CR6'



```
select\ a. Customer ID\ ,\ b. Generation,\ b. Condition from\ Customer\ a\ ,\ Car\ b where\ a. TelNo=(select\ TelNo\ from\ Customer\ where\ TelNo=57774002);
```



Booking

```
create table Booking(
BookingID varchar(10) not null,
Description varchar(50),
ReserveDate varchar(10),
ReturnDate varchar(10),
AmountDue float,
primary key (BookingID),
);
```

insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue) values ('B001', 'Online Payment','17/01/2006', '27/01/2006', 2500);

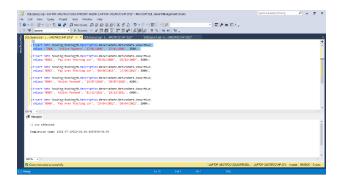
insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue) values('B002', 'Pay when fetching car', '05/02/2009', '25/02/2009', 3600);

insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue) values('B003', 'Pay when fetching car', '06/06/2015', '17/06/2015', 3000);

insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue)
values('B004', 'Online Payment', '18/07/2020', '30/07/2020', 4200);

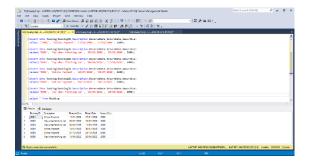
insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue)
values('B005', 'Online Payment', '01/12/2021', '28/12/2021', 6900);

insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue) values('B006', 'Pay when fetching car', '15/04/2022', '20/04/2022', 2000);



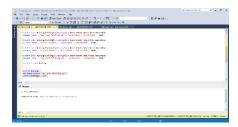
To view Booking table

select * from Booking



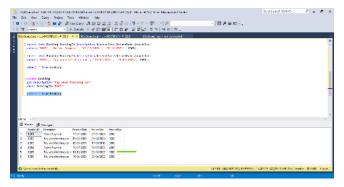
update Booking

set Description='Pay when fetching car' where BookingID='B005'



To view updated table

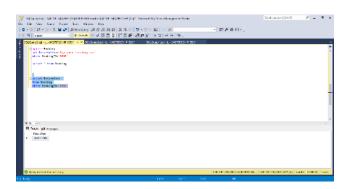
select * from Booking



select ReturnDate

from Booking

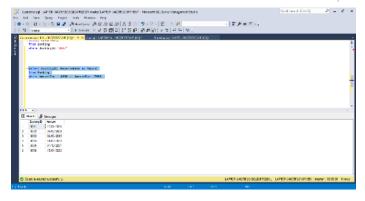
where BookingID='B002'



select BookingID, ReserveDate as Amount

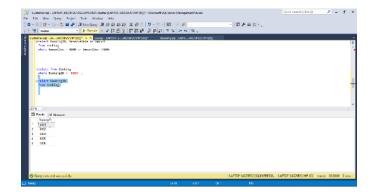
from Booking

where AmountDue > 4000 or AmountDue < 7000;

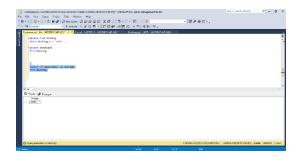


```
delete from Booking
where BookingID = 'B003';
```

To view deleted row select BookingID from Booking



select AVG(AmountDue) as Average
from Booking;



5. STORED PROCEDURES

Procedure 1:

CREATE PROC sp_carDetails @CustomerID varchar(10),@Name varchar (30),@Address varchar (50),@Email varchar (50),@TelNo varchar(8), @DoB varchar(10)

AS

BEGIN

BEGIN TRY

INSERT INTO Customer

VALUES (@CustomerID, @Name, @Address, @Email, @TelNo, @DoB)

PRINT 'Insert was successful!'

END TRY

BEGIN CATCH

SELECT ERROR_MESSAGE() AS ErrorMessage;

END CATCH

END

EXEC sp_carDetails @CustomerID= 'C023', @Name= 'Geerna Peeharry', @Address= 'Midlands', @Email= 'babyshark@gmail.com',

@TelNo= 58339245 ,@DoB= '29/02/2000'

```
SCACHT FMC Sp_cartetail @CustomerID varchar(10).@Hame varchar (30).@Address varchar (50).@Email varchar (50).@Telleo varchar(8),

AS 100 Control of the special control of the special
```

Procedure 2:

CREATE PROC sp_carDetails_02 @CarID varchar(10),@PlateNo varchar (10),@Colour varchar (20),@Generation varchar(4), @Model varchar(30),@SeatingCapacity varchar,@Condition varchar (50)

AS

BEGIN

BEGIN TRY

INSERT INTO Car

VALUES (@CarID,@PlateNo, @Colour, @Generation, @Model, @SeatingCapacity, @Condition)

PRINT 'Insert was successful!'

END TRY

BEGIN CATCH

SELECT ERROR_MESSAGE() AS ErrorMessage;

END CATCH

END

EXEC sp_carDetails_02 @CarID= 'CR7', @PlateNo= 'B0302', @Colour= 'Black', @Generation= '2022', @Model= 'Mitsubishi',

@SeatingCapacity= 5, @Condition= 'Excellent'

```
SOLOWING Set 14. SCHEMY-MONE (SS)* 9 × 1

| CERTE FROC sp_carbetails_00 & ScarD varcher(10)_@Platelso varcher(10)_@Center Froc sp_carbetails_00 & ScarD varcher(20)_@Platelso varcher(10)_@Center varcher(10)_
```

Procedure 3:

CREATE PROC sp_carDetails_03 @BookingID varchar(10),@Description varchar(50),@ReserveDate varchar(10),@ReturnDate varchar(10), @AmountDue float

AS

BEGIN

BEGIN TRY

INSERT INTO Booking

 $VALUES\ (@BookingID,\ @Description\ , @ReserveDate\ ,\ @ReturnDate\ ,$ @AmountDue)

PRINT 'Insert was successful!'

END TRY

BEGIN CATCH

SELECT ERROR_MESSAGE() AS ErrorMessage;

END CATCH

END

EXEC sp_carDetails_03 @BookingID= 'B007', @Description= 'Online Payment', @ReserveDate= '18/07/2020',

@ReturnDate= '23/07/2020', @AmountDue= 2000



Procedure 4:

CREATE PROCEDURE sp_disp_booking

AS

BEGIN

SELECT CR.PlateNO, C.Name, C.CustomerID

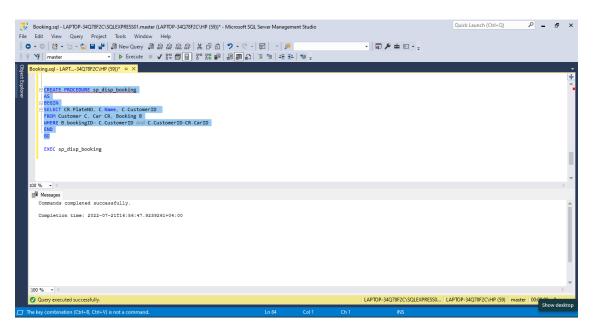
FROM Customer C, Car CR, Booking B

WHERE B.bookingID= C.CustomerID And C.CustomerID=CR.CarID

END

GO

EXEC sp_disp_booking



6. TRIGGERS

Write a trigger procedure that ensures that all deleted records from the 'booking' table are kept in the table 'booking_old' with the following fields: ReserveDate, ReturnDate, AmountDue

```
CREATE TABLE booking_old (
   ReserveDate varchar(10),
   ReturnDate varchar(10),
  AmountDue float,
 );
Trigger 1: To delete
CREATE TRIGGER old_record
ON Booking
AFTER DELETE
AS
 INSERT INTO booking_old (ReserveDate, ReturnDate, AmountDue)
       select ReserveDate, ReturnDate, AmountDue FROM deleted
--TESTING
SELECT * FROM booking_old
DELETE FROM Booking WHERE BookingID= 'B001';
select * from Booking
```

Trigger 2: Check name

口 Person ・ P 日 m B B B B O O D III の L

```
CREATE TRIGGER check_data_customer
ON Customer
INSTEAD OF INSERT
AS
DECLARE @customerName varchar(30)
SET @customerName = ''
BEGIN
PRINT 'Name cannot be empty , please enter a name'
RETURN
END
IF EXISTS (SELECT * FROM Customer, INSERTED WHERE Customer.Name =INSERTED.Name)
BEGIN
PRINT 'Name already exists'
RETURN
END
--testing--
INSERT INTO Customer (Name)
VALUES('')
SELECT * FROM Customer
```

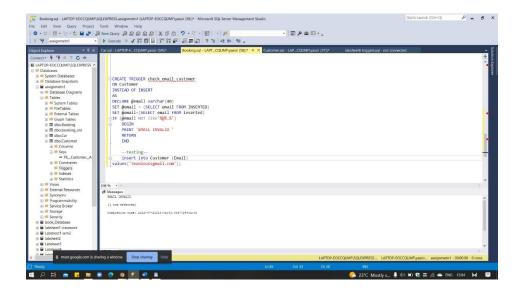
Trigger 3: Check email

```
CREATE TRIGGER check_email_customer
ON Customer
INSTEAD OF INSERT
AS

DECLARE @email varchar(40)
SET @email = (SELECT email FROM INSERTED)
SET @email=(SELECT email FROM inserted)
IF (@email NOT like'%@%.%')

BEGIN
PRINT 'EMAIL INVALID '
RETURN
END

--Testing--
insert into Customer (Email)
values('Jeanlouisgmail.com');
```



Trigger 4: checkCondition

SELECT* from Car

```
CREATE TRIGGER checkCondition
ON Car
INSTEAD OF INSERT
AS
DECLARE @condition varchar(50)
DECLARE @ID varchar(10)
SET @condition =(SELECT Condition FROM inserted)
SET @ID =(SELECT CarID FROM inserted)
IF (@ID='CR1')
  BEGIN
  SET @condition = 'Good'
   RETURN
   END
  UPDATE Car
   set Condition=@condition
  WHERE CarID = @ID
   --testing--
```

Trigger 5: check_AmountDue

```
create trigger check_AmountDue
on Booking
instead of insert
as
declare @Amount real
declare @date date
set @Amount = (select AmountDue from inserted)
if(@Amount > 10000)
begin
print 'Too expensive, give discount'
return
end
insert into Booking select * from inserted
set @Amount=(select AmountDue from inserted)
set @date = (select ReserveDate from inserted)
update Booking set AmountDue = (AmountDue - (AmountDue * 0.75)) where
ReserveDate='05/02/2009';
insert into Booking select * from inserted
```

insert into Booking values ('B006', 'Pay online', '05/02/2009', '25/02/2009', 31600)

Trigger 6: display_CustomerData

```
CREATE TRIGGER display_CustomerData
ON Customer
                                                                                   - a ≠ ± m -
AFTER INSERT
AS
DECLARE @customerID varchar(10)
DECLARE @name varchar(30)
DECLARE @address varchar(50)
DECLARE @email varchar(50)
DECLARE @telNo varchar(8)
DECLARE @dateOfBirth varchar(10)
SET @customerID =(SELECT CustomerID FROM inserted)
SELECT @name = Name, @address = Address ,@email = Email ,@telNo = TelNo, @dateOfBirth =
DateOfBirth
FROM Customer
WHERE CustomerID=@customerID
PRINT 'Booking owner have the name'+ CAST(@name AS varchar(30)) +'who can be contacted
on'+ CAST(@email AS varchar(30))
+ 'who have these details as well' + CAST(@address AS varchar(50))+CAST(@telNo AS
varchar(8)) + CAST(@dateOfBirth AS varchar(10))
--TESTING
INSERT INTO Customer VALUES('C001' , 'Selena Gomez', 'Triolet', 'Selenagomez@gmail.com',
57810028, '10/03/2001');
INSERT INTO Customer VALUES (86455, 'Lit44 Women', 220, '1980-02-15');
SELECT *FROM Customer;
 - Checking for storting Jetter (*)
GTT SOutheaville - (GELECT East) FROM DESCRIPE)
IT (SOutheaville - (GELECT East)
```

Trigger 7: Check_CustID

CREATE TRIGGER Check_CustID

ON Customer

INSTEAD OF INSERT

AS

DECLARE @CustomerID VARCHAR(10)

-- Checking for starting letter 'C'

SET @CustomerID = (SELECT Email FROM INSERTED)

IF (@CustomerID NOT LIKE 'C%')

BEGIN

PRINT 'Incorrect syntax. ID should start with a C'

END

ELSE

PRINT 'Correct ID syntax'

--Testing

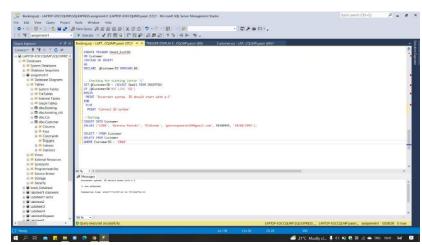
INSERT INTO Customer

VALUES ('C018', 'Geerona Peeruhi', 'Midlands', 'geeronapeeruhi096gmail.com', 58340945, '29/01/1995');

SELECT * FROM Customer

DELETE FROM Customer

WHERE CustomerID = 'C018'



Trigger 8: Check_BookingID

CREATE TRIGGER Check_BookingID

ON Booking

INSTEAD OF INSERT

AS

DECLARE @BookingID VARCHAR(10)

SET @BookingID = (SELECT BookingID FROM INSERTED)

IF (@BookingID NOT LIKE 'B%')

BEGIN

PRINT 'Incorrect syntax. ID should start with an B'

END

ELSE

PRINT 'Correct ID syntax'

insert into Booking(BookingID,Description,ReserveDate,ReturnDate,AmountDue) values ('001', 'Online Payment','17/01/2006', '27/01/2006', 2500);

