```
------------Database-----------------
USE online_homestay;

------------Table Creation-----------------

CREATE TABLE dbo.Customers (
    CustID int NOT NULL IDENTITY(1,1) PRIMARY KEY,
    FirstName varchar(100) NOT NULL,
    LastName varchar(100),
    PhoneNo int NOT NULL CHECK ([PhoneNo] <= 9999999999),
    AddressLine1 varchar(100) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    ZipCode varchar(5) NOT NULL CHECK ([ZipCode] > 9999 AND [ZipCode] <= 99999),
    EmailID varchar(50) NOT NULL CHECK (EmailID LIKE '%_@__%.__%')
);
DBCC CHECKIDENT ('dbo.Customers', RESEED, 1);


CREATE TABLE host (
hostid int NOT NULL primary key,
first_name varchar(20) not NULL ,
last_name varchar(20),
phone_no int not NULL ,
address_line varchar(20) not null,
city varchar(20) not null,
state varchar(20) not null,
zipcode varchar(5) not null check (zipcode>=10000 and zipcode<=99999),
emailid varchar(20) not null check (emailid like '%@%.%')
);


CREATE TABLE payment_type (
typeID int NOT NULL IDENTITY(1,1),
payment_type varchar(50)
PRIMARY KEY (typeID),
);
DBCC CHECKIDENT ('dbo.payment_type', RESEED, 1);


CREATE TABLE payment_status (
```

```
statusID int NOT NULL IDENTITY(1,1),
status varchar(50),
PRIMARY KEY (statusID),
);
DBCC CHECKIDENT ('dbo.payment_status', RESEED, 1);
```

```
create table dbo.commission (CID int IDENTITY(1,1) not NULL primary key,
zipcode int not null constraint zipcode_digits check (len( [zipcode]) = 5),
comm_percent float not null);
DBCC CHECKIDENT ('dbo.commission', RESEED, 1);
```

```
CREATE TABLE dbo.Property (
    PropertyID int NOT NULL IDENTITY(1,1) PRIMARY KEY,
    hostID int NOT NULL,
    CID int NOT NULL,
    PropertyType varchar(100) NOT NULL,
    AddressLine1 varchar(100) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    ZipCode INT NOT NULL CHECK ([ZipCode] > 9999 AND [ZipCode] <= 99999),
    Price_per_day money NOT NULL,
    FOREIGN KEY (hostID) REFERENCES dbo.host(hostID),
    FOREIGN KEY (CID) REFERENCES dbo.commission (CID)
);
DBCC CHECKIDENT ('dbo.Property', RESEED, 1);
```

```
create table dbo.amenities (propertyID int not null primary key references
dbo.Property(PropertyID),
no_of_bedrooms int not null,
no_of_bathrooms int not null ,
area int not null,
AC int NOT NULL check (AC in (0,1)) ,
geyser int NOT NULL check (geyser in (0,1)),
parking_lot int NOT NULL check (parking_lot in (0,1)),
laundry int not null check (laundry in (0,1))
);
```

```
CREATE TABLE payment (
paymentID int NOT NULL IDENTITY(1,1),
statusID int NOT NULL FOREIGN KEY REFERENCES payment_status(statusID),
typeID int NOT NULL FOREIGN KEY REFERENCES payment_type(typeID),
final_amt MONEY NOT NULL,
PRIMARY KEY (paymentID),
);
DBCC CHECKIDENT ('dbo.payment', RESEED, 1);
```

```
create table dbo.bookings (bookingID int IDENTITY(1,1) not null primary key,
CustID int not null REFERENCES dbo.Customers(CustID) ,
propertyID int not null REFERENCES dbo.Property(PropertyID),
paymentID int not null REFERENCES dbo.payment(paymentID),
booked_from date not null,
booked_to date not null,
commission_amount money not null);
DBCC CHECKIDENT ('dbo.bookings', RESEED, 1);
```

```
CREATE TABLE Availibility (
AID int NOT NULL IDENTITY(1,1),
PropertyID int NOT NULL FOREIGN KEY REFERENCES Property,
Available_date date NOT NULL,
IsAvailable int NOT NULL CHECK (IsAvailable in (0,1)),
PRIMARY KEY (AID),
);
DBCC CHECKIDENT ('dbo.Availibility', RESEED, 1);
```

```
CREATE TABLE PriceHistory (
priceHistId int NOT NULL IDENTITY(1,1),
PropertyID int NOT NULL FOREIGN KEY REFERENCES Property,
last_changed_date date NOT NULL,
price money NOT NULL,
PRIMARY KEY (PriceHistId)
);
DBCC CHECKIDENT ('dbo.PriceHistory', RESEED, 1);
```

```
CREATE TABLE dbo.review
```

```
(
	revID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
	custID INT NOT NULL FOREIGN KEY REFERENCES dbo.Customers(CustID),
	propertyID INT NOT NULL FOREIGN KEY REFERENCES dbo.Property(PropertyID),
	review VARCHAR(50)

);
DBCC CHECKIDENT ('dbo.review', RESEED, 1);


INSERT INTO dbo.Customers
(FirstName, LastName, PhoneNo, AddressLine1, City, State, ZipCode, EmailID)
VALUES
('James', 'Smith', 2067805566, '324 17th Ave NE', 'Seattle', 'Washington', 98105,
'jamesmith@gmail.com'),
('Robert', 'Miller', 2067802222, '4456 Belmont Ave W', 'Seattle', 'Washington', 98109,
'miller_robert@yahoo.com'),
('David', 'Davis', 2067802222, '534 Whitefield Rd', 'Buffalo', 'NewYork', 14201,
'david12davis@gmail.com'),
('Hary', 'Clark', 2061111166, '324 Supreme Ave NE', 'Minneapolis', 'Minnesota', 55111,
'clark_hary3@gmail.com'),
('Maria', 'Sanders', 2117803366, '438 Market St', 'San Francisco', 'California', 94117,
'maria123@hotmail.com'),
('Maria', 'Garcia', 1677804966, '5566 17th Ave NE', 'Seattle', 'Washington', 98122,
'mariagarcia@hotmail.com'),
('Sarah', 'Johnson', 1237805432, '800 Oregon St', 'Portland', 'Oregan', 97035,
'johnson_sarah@gmail.com'),
('Ann', 'Wilson', 2069990000, '4488 18th Street SE', 'Bellevue', 'Washington', 98005,
'wilsonann@yahoo.com'),
('Jane', 'Brown', 1171115566, '327 Roosevelt Ave', 'New York', 'New York', 10005,
'janebrown@gmail.com'),
('Mike', 'Jones', 2113336666, '23 Puffer St', 'Los Angeles',' California', 95336,
'mike123jones@ymail.com');

DROP SEQUENCE SEQ_USER;
CREATE SEQUENCE SEQ_USER START WITH 1 INCREMENT BY 1;
insert into host
(hostID, first_name, last_name, phone_no, address_line, city, State, zipcode, emailid)
values
(next value for seq_user, 'John', 'yo', 1023557980, '#432 St.','Seattle', 'WA', 23790,
'john@yahoo.com'),
(next value for seq_user, 'Chris', 'to', 1123550980, '#205 St.','Seattle', 'WA', 23792,
'chris@gmail.com'),
```

```
(next value for seq_user, 'Albert', 'Stewart', 1294667980, '#405 St.','New York City', 'NY', 72021,
'albert@gmail.com'),
(next value for seq_user, 'Bruno', 'Fernandes', 1023667980, '#Old Trafford St.','Manchester', 'MA',
12090, 'bruno@gmail.com'),
(next value for seq_user, 'Paul', 'Pogba', 1071246980, '#432 St.','Venice', 'CA', 71650,
'paul@yahoo.com'),
(next value for seq_user, 'Harry', 'Maguire', 1123612976, '#Old Trafford St.','Manchester', 'MA',
29270, 'harry@yahoo.com'),
(next value for seq_user, 'Luke', 'Lu', 1896356176, '#7th Avenue','Boston', 'MA', 47820,
'luke@gmail.com'),
(next value for seq_user, 'Bobby', 'Duncan', 1975612006, '#10 th Avenue.','New York City', 'NY',
72190, 'bobby@gmail.com'),
(next value for seq_user, 'Simon', NULL, 1171856976, '#U District St.','Seattle', 'WA', 98105,
'simon@gmail.com'),
(next value for seq_user, 'Anthony', NULL, 1723576401, '#Old Trafford St.','Manchester', 'MA',
20075, 'anthony@gmail.com');


Insert into dbo.commission
(zipcode,comm_percent)
Select 98105,9 --seattle
Union
Select 10026, 14 --new york
Union
Select 94016,12 --San francisco
Union
Select 90034, 14 --LA
Union
Select 60623, 8 --chicago
Union
Select 75032, 11 -- dallas
Union
Select 85002, 6 -- phoenix
Union
Select 89125 , 13 -- Las Vegas
Union
Select 33129 , 11 -- Miami
Union
Select 70032 , 8 --New orelans


DROP PROCEDURE dbo.PriceInsert;
CREATE PROCEDURE dbo.PriceInsert
```

```
       @PropertyID int,
       @Price_per_day money
AS
BEGIN
     SET NOCOUNT ON

     INSERT INTO dbo.PriceHistory
          (PropertyID, last_changed_date, price)
     VALUES
          (
               @PropertyID,
               current_timestamp,
               @Price_per_day
          )

END;


DROP PROCEDURE dbo.CheckAvailabilty;
create procedure dbo.CheckAvailabilty
@PropertyID int,
@StartDate DATE,
@NoOfDays INT
AS
BEGIN
DECLARE @COUNT INT = 0;
WHILE(@COUNT < @NoOfDays)
BEGIN
INSERT INTO dbo.Availibility values(@PropertyID, DATEADD(DAY,  @COUNT,@StartDate), 1)
SET @COUNT +=1;
END
END;


DROP procedure dbo.PropertyInsert ;
CREATE PROCEDURE dbo.PropertyInsert
  (    @hostID int  ,
       @CID int,
       @PropertyType varchar(100) ,
       @AddressLine1 varchar(100) ,
       @City varchar(50) ,
       @State varchar(50) ,
       @ZipCode varchar(5) ,
```

```
       @Price_per_day money
    )
AS
BEGIN
     SET NOCOUNT ON
      DECLARE @ID int, @StartDate1 date, @NoOfDays1 INT;
         SET @StartDate1 = CURRENT_TIMESTAMP
         SET @NoOfDays1 = 15
     INSERT INTO dbo.Property
          (hostID, CID, PropertyType, AddressLine1, City, State, ZipCode, Price_per_day)
     VALUES
          (
               @hostID,
               @CID,
               @PropertyType,
               @AddressLine1,
               @City,
               @State,
               @ZipCode,
               @Price_per_day
       )
   SET @ID = SCOPE_IDENTITY()


EXEC dbo.PriceInsert @ID, @Price_per_day
EXEC dbo.CheckAvailabilty @ID, @StartDate1, @NoOfDays1

END ;



DECLARE @PropertyID int;
DECLARE @hostID int;
DECLARE @CID int;
DECLARE @PropertyType varchar(100);
DECLARE @AddressLine1 varchar(100);
DECLARE @City varchar(50);
DECLARE @State varchar(50);
DECLARE @ZipCode varchar(5);
DECLARE @Price_per_day money;
DECLARE @StartDate1 DATE;
DECLARE @NoOfDays1 INT;
DECLARE @ID INT;
```

```
--Initilize variable
SET @hostID = 1;
SET @CID = 3;
SET @PropertyType = 'studio';
SET @AddressLine1 = '234 Master Ave';
SET @City = 'Chicago';
SET @State = 'Illinois';
SET @ZipCode ='60623';
SET @Price_per_day = 100;
SET @ID =12;
--Execute the procedure
EXEC dbo.PropertyInsert  @hostID, @CID, @PropertyType, @AddressLine1, @City, @State, @ZipCode,
@Price_per_day ;


insert into dbo.review
(custID,propertyID,review)
Select 1,1,'Very good service'
Union
Select 5,1,'Very friendly staff'
Union
Select 9,2, 'Unhygenic Place'
Union
Select 1,2,'Average Rooms'


INSERT INTO dbo.payment_type
VALUES ('Credit Card');
INSERT INTO dbo.payment_type
VALUES ('Debit Card');
INSERT INTO dbo.payment_type
VALUES ('Paypal'),('Mobile Payments'),('Direct Deposit');


INSERT INTO dbo.payment_status
VALUES ('Confirmed'),('Pending'),('Cancelled'),('Failed');


CREATE TRIGGER dbo.updatePropertyCost
ON dbo.Property
AFTER  UPDATE
AS
begin
```

```
if update(Price_per_day)
begin
insert into dbo.PriceHistory ( PropertyID,last_changed_date, price )
select
    i.PropertyID,
    CURRENT_TIMESTAMP,
    i.Price_per_day
FROM
    inserted AS i;

       END;
END;


insert into amenities
(propertyID, no_of_bedrooms, no_of_bathrooms, area, AC, geyser, parking_lot, laundry)
values
(1,2,1,300,1,0,0,1),
(2,4,3,900,1,1,1,1);



CREATE PROCEDURE create_payment @StatusID varchar(40), @TypeID varchar(40),@final_amt int
AS
INSERT INTO payment VALUES (@StatusID, @TypeID ,@final_amt)


CREATE PROCEDURE booking_insert @Cust_id int, @PropertyID int, @PaymentID int, @booked_from date,
@booked_to date, @com_amt float
AS
INSERT INTO bookings VALUES (@Cust_id, @PropertyID, @PaymentID, @booked_from, @booked_to, @com_amt)

CREATE PROCEDURE update_availibility @PropertyID int, @StartDate date, @EndDate date
AS
UPDATE Availibility SET IsAvailable =0  WHERE PropertyID = @PropertyID AND Available_date between
@StartDate and @Enddate


CREATE PROCEDURE make_bookings @PropertyID int, @StartDate Date, @EndDate Date, @Cust_id int,
@StatusID int, @TypeID int
AS
```

```
DECLARE @final_amt float, @price_per_day float, @zipcode int, @comm_per float, @comm_amt float,
@no_of_days int, @amt float

DECLARE @payID INT
SELECT available_Date
into #date_range
from availibility
where propertyid = @propertyID and isAvailable = 1

DECLARE @payID_not_available INT
SELECT available_Date
into #date_range_not_available
from availibility
where propertyid = @PropertyID and isAvailable = 0


if @startdate in (select * from #date_range_not_available )
	BEGIN
	print 'Not Available'
	RETURN -1;
	END
else if @enddate in (select * from #date_range_not_available )
	BEGIN
	print 'Not Available'
	RETURN -1;
	END
else
	SELECT @price_per_day = price_per_day, @zipcode = zipcode
	FROM property
	WHERE propertyID = @PropertyID

	SELECT  @comm_per = comm_percent
	FROM commission
	WHERE zipcode = @zipcode

	SELECT @no_of_days =  DATEDIFF(day, @StartDate, @EndDate) + 1
	SET @amt = @no_of_days * @price_per_day
	SET @final_amt = ((@comm_per/100) * @amt) + @amt
	SET @comm_amt = (@comm_per * @amt)/100

	INSERT INTO payment VALUES (@StatusID, @TypeID ,@final_amt)
	set @payID = SCOPE_IDENTITY()
```

```
	EXEC booking_insert @Cust_id= @Cust_id, @PropertyID = @PropertyID,@PaymentID = @payID,
@booked_from= @StartDate, @booked_to =@EndDate, @com_amt = @comm_amt
	EXEC update_availibility @PropertyId = @PropertyId, @StartDate    = @StartDate, @EndDate =
@Enddate



DECLARE @StartDate Date;
DECLARE @EndDate Date;
DECLARE @PropertyID int;
DECLARE @Cust_id int;
DECLARE @StatusID int;
DECLARE @TypeID int;
SET @StartDate = '2020-03-16'
SET @EndDate = '2020-03-19'
SET @PropertyID = 3
SET @Cust_ID = 2
SET @StatusID =1
SET @TypeID = 1

EXEC make_bookings @PropertyID, @StartDate , @EndDate, @Cust_id, @StatusID, @TypeID


----------------------Creation of views----------------------

CREATE VIEW propertyreview
AS SELECT p.PropertyID, p.City, h.hostID, r.review
FROM dbo.host h, dbo.Property p, dbo.review r
WHERE h.hostID = p.hostID
AND r.propertyID = p.PropertyID;

SELECT * FROM propertyreview;


CREATE VIEW PropertyDetails AS
SELECT p.PropertyID,p.PropertyType,p.AddressLine1,p.City,p.Price_per_day,am.no_of_bedrooms,
am.no_of_bathrooms , av.Available_date, av.IsAvailable from dbo.property as p join dbo.amenities as
am
on p.PropertyID = am.PropertyID join dbo.Availibility as av on av.PropertyID = p.PropertyID group
by p.PropertyID,p.PropertyType,p.AddressLine1,p.City,p.Price_per_day,am.no_of_bedrooms,
am.no_of_bathrooms ,
av.Available_date, av.IsAvailable having av.IsAvailable = 1;
```

```
SELECT * FROM PropertyDetails;

-------------------------------------------------------------------------------------------------------------
---------------------
-------------------------------------------------------------------------------------------------------------
---------------------



DROP TABLE dbo.bookings ;
drop TABLE dbo.Availibility ;
DROP TABLE dbo.amenities ;
drop table dbo.PriceHistory ;
drop table dbo.review ;
drop table dbo.payment ;
drop table dbo.payment_status ;
drop table dbo.payment_type ;
drop table dbo.Customers ;
drop table dbo.host ;
drop table dbo.commission ;
drop table dbo.Property ;



SELECT * from dbo.Customers;
SELECT * FROM dbo.host;
SELECT * FROM dbo.Property;
SELECT * FROM dbo.Bookings;
select * from dbo.availibility;
select * from dbo.pricehistory;
select * from dbo.payment;
select * from dbo.payment_type;
Select * from dbo.payment_status;
select * from dbo.commission;
select * from dbo.amenities;
select * from dbo.review;



DROP PROCEDURE create_payment;
DROP PROCEDURE update_availibility;
DROP PROCEDURE make_bookings;
DROP PROCEDURE booking_insert;
```