Group12- Second Project Deliverable

CSI2132- Database I

Gabriel Karkaji, 6575850
Khyber Lahori, 8018423

1. a) We have used PostgreSQL as our DBMS other programming languages we have used include HTML and PHP

   b) The DDL used to create our database are:

```
CREATE TABLE Branch(
BranchNumber Serial,
NumberofEmployees Integer NOT NULL,
Country VarChar(200) NOT NULL,
Manager_SSN Integer Default 0,
PRIMARY KEY (BranchNumber, Country)
);

CREATE TABLE Employee(
SSN Integer Default 0,
HashedPassword VarChar(200) NOT NULL,
FName VarChar(200) NOT NULL,
MName VarChar(200) NOT NULL,
LName VarChar(200) NOT NULL,
Address VarChar(200) NOT NULL,
City VarChar(200) NOT NULL,
Province_State VarChar(200) NOT NULL,
Country VarChar(200) NOT NULL,
EPosition VarChar(200) NOT NULL,
Salary Integer NOT NULL,
EmailAddress VarChar(200) NOT NULL,
PhoneNumber VarChar(200) NOT NULL,
Super_SSN Integer Default 0,
BranchNumber Integer  NOT NULL,
PRIMARY KEY (SSN),
FOREIGN KEY(Super_SSN) REFERENCES Employee(SSN)
ON DELETE SET DEFAULT
ON UPDATE CASCADE,
FOREIGN KEY(BranchNumber, Country) REFERENCES Branch(BranchNumber, Country)
ON DELETE CASCADE
);

CREATE TABLE Host(
HostID Serial NOT NULL,
HashedPassword VarChar(200) NOT NULL,
```

```
FName VarChar(200) NOT NULL,
MName VarChar(200) NOT NULL,
LName VarChar(200) NOT NULL,
Address VarChar(200) NOT NULL,
City VarChar(200) NOT NULL,
Province_State VarChar(200) NOT NULL,
Country VarChar(200) Default 0,
BranchNumber integer Default 0,
PRIMARY KEY (HostID),
FOREIGN KEY(BranchNumber, Country) REFERENCES Branch(BranchNumber, Country)
ON DELETE CASCADE
);


CREATE TABLE Guest(
GuestID Serial NOT NULL,
HashedPassword VarChar(200) NOT NULL,
FName VarChar(200) NOT NULL,
MName VarChar(200) NOT NULL,
LName VarChar(200) NOT NULL,
Address VarChar(200) NOT NULL,
City VarChar(200) NOT NULL,
Province_State VarChar(200) NOT NULL,
Country VarChar(200) NOT NULL,
BranchNumber Integer NOT NULL,
EmailAddress VarChar(200) NOT NULL,
PRIMARY KEY (GuestID),
FOREIGN KEY(Country, BranchNumber) REFERENCES Branch (Country, BranchNumber)
ON DELETE CASCADE
);

CREATE TABLE Property(
PropertyID Serial NOT NULL,
Rented BIT(1) NOT NULL,
Address VarChar(200) NOT NULL,
City VarChar(200) NOT NULL,
Province_State VarChar(200) NOT NULL,
Country VarChar(200) NOT NULL,
PropertyType VarChar(200) NOT NULL,
Accommodates Integer NOT NULL,
Amenities VarChar(200) NOT NULL,
Bedroom integer NOT NULL,
Beds integer NOT NULL,
```

```
Bathroom integer NOT NULL,
HostID Integer NOT NULL,
PropManagerID Integer NOT NULL,
BranchNumber integer Default 0,
CONSTRAINT  PropertyType_Check CHECK (PropertyType = 'Apartment' or PropertyType =
'Bed and Breakfast' or PropertyType = 'Unique Home' or PropertyType = 'Vacation Home' or
PropertyType = 'Cottage'),
PRIMARY KEY (PropertyID),
FOREIGN KEY(HostID) REFERENCES Host(HostID)
ON DELETE CASCADE,
FOREIGN KEY(BranchNumber, Country) REFERENCES Branch(BranchNumber, Country)
ON DELETE CASCADE,
FOREIGN KEY(PropManagerID) REFERENCES Employee(SSN)
ON DELETE SET DEFAULT
ON UPDATE CASCADE
);

CREATE TABLE Room (
PropertyID Integer REFERENCES Property(PropertyID) ON DELETE CASCADE,
RoomID Serial NOT NULL,
RoomType Varchar(200) NOT NULL,
Accommodates Integer NOT NULL,
Amenities VarChar(200) NOT NULL,
CONSTRAINT  RoomType_Check CHECK (RoomType = 'Unique Space' or RoomType =
'Private' or RoomType = 'Shared'),
Primary Key (RoomID,PropertyID)
);

CREATE TABLE Pricing (
PricingID Serial NOT NULL,
Rules Varchar(200),
NumberOfGuests integer NOT NULL,
HomeType VarChar(200) NOT NULL,
Amenities VarChar(200) NOT NULL,
DailyCost integer NOT NULL,
HostID Integer NOT NULL,
PropertyID Integer NOT NULL,
RoomId integer default null,
CONSTRAINT  HomeType_Check CHECK (HomeType = 'Apartment' or HomeType = 'Bed and
Breakfast' or HomeType = 'Unique Home' or HomeType = 'Vacation Home' or HomeType =
'Cottage' or  HomeType = 'Unique Space' or  HomeType = 'Private' or  HomeType = 'Shared'),
Primary Key (PricingID),
FOREIGN KEY(HostID) REFERENCES Host(HostID)
```

```
ON DELETE CASCADE,
FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID)
ON DELETE CASCADE,
FOREIGN KEY(PropertyID, RoomID) REFERENCES Room(PropertyID, RoomID)
ON DELETE CASCADE
);

CREATE TABLE Rental_Agreement (
OrderID Serial NOT NULL,
TotalPrice integer default 0,
Signature VarChar(200) NOT NULL,
SigningDate Date NOT NULL,
StartDate Date NOT NULL,
EndDate Date NOT NULL,
GuestID Integer NOT NULL,
PricingID Integer NOT NULL,
OccupancyRate Integer NOT NULL,
Primary Key (OrderID),
FOREIGN KEY(GuestID) REFERENCES Guest(GuestID),
FOREIGN KEY(PricingID) REFERENCES Pricing(PricingID)
);

CREATE TABLE Payment(
PaymentID Serial NOT NULL,
PaymentAccepted BIT(1) NOT NULL,
PaymentType VarChar(200) NOT NULL,
PaymentStatus VarChar(200) NOT NULL,
TotalPrice Integer NOT NULL,
HostID Integer NOT NULL,
GuestID Integer NOT NULL,
OrderID Integer NOT NULL,
CONSTRAINT  PaymentType_Check CHECK (PaymentType = 'Credit' or PaymentType =
'Cash' or PaymentType = 'Check' or PaymentType = 'Debit'),
CONSTRAINT  PaymentStatus_Check CHECK (PaymentStatus = 'Pending' or PaymentStatus
= 'Approved' or PaymentStatus = 'Declined' or PaymentStatus = 'Completed'),
PRIMARY KEY (PaymentID),
FOREIGN KEY(HostID) REFERENCES Host(HostID)
ON DELETE CASCADE,
FOREIGN KEY(GuestID) REFERENCES Guest(GuestID),
FOREIGN KEY(OrderID) REFERENCES Rental_Agreement(OrderID)
);
```

```
CREATE TABLE Review (
ReviewID Serial NOT NULL,
Communication integer NOT NULL,
Cleanliness integer NOT NULL,
TheValue integer NOT NULL,
OverallRating integer Not NUll,
TheComment VarChar(500),
OrderID Integer NOT NULL,
Primary Key (ReviewID),
CONSTRAINT  Communication_Rating_Check CHECK (Communication >= 1 and
Communication <= 10),
CONSTRAINT  Cleanliness_Rating_Check CHECK (Cleanliness >= 1 and Cleanliness <= 10),
CONSTRAINT  TheValue_Rating_Check CHECK (TheValue >= 1 and TheValue <= 10),
FOREIGN KEY(OrderID) REFERENCES Rental_Agreement (OrderID)
ON DELETE SET DEFAULT
);

CREATE TABLE Host_Phone (
PhoneNumber varchar(200) NOT NULL ,
HostID Integer NOT NULL REFERENCES Host(HostID) ON DELETE  CASCADE,
Primary Key (HostID, PhoneNumber)
);

CREATE TABLE Guest_Phone (
PhoneNumber varchar(200) NOT NULL,
GuestID Integer NOT NULL  REFERENCES Guest(GuestID) ON DELETE CASCADE,
Primary Key (GuestID, PhoneNumber)
);

CREATE TABLE Host_Email (
Email VarChar(200) NOT NULL,
HostID Integer NOT NULL REFERENCES Host(HostID) ON DELETE CASCADE,
Primary Key(HostID, Email)
);

Create or Replace Function TotalPriceFunction()
Returns trigger as
$BODY$
Begin
```

```
update Rental_Agreement set TotalPrice = (date_part('day', age(New.enddate,
new.startdate))*(Select dailycost from pricing where pricingid = New.pricingid))where orderid =
new.orderid;
RETURN NEW;
end;
$BODY$ language plpgsql;


Create Trigger totalPriceTrigger After Insert On Rental_Agreement
For each row
execute procedure TotalPriceFunction();


Create or Replace Function overallRatingFunction()
Returns trigger as
$BODY$
Begin
Update Review set OverallRating = ((New.Communication::float + new.Cleanliness::float +
new.TheValue::float)/3) where reviewid = new.reviewid;
return new;
end;
$BODY$ language plpgsql;

Create Trigger overallRatingTrigger After Insert On Review
For each row
execute procedure overallRatingFunction();
```

c)
Step 1: install the appropriate software needed which include Apache2.4, pgAdmin,,postgresql, and php(follow lab 8a in needed)
Step 2: Into pgAdmin paste the code from the script.sql file that contains all the database information
Step 3: Make sure all the PHP files are in placed in the htdocs of Apache2.4
Step 4:Inside all the php files paste in your user information and password
Step5: on your browser enter http://127.0.0.1/pickLogin.php to start from the beginning of the app
Step 6: application has started you may now wither pick who you want to login as and continue using application

For the DB Admin, he can add Branches and Employees to the DB by inputting the necessary queries in the SQL Shell command prompt (which he can download from here: https://www.enterprisedb.com/downloads/postgresql). He will be prompted to enter the server

address, the database name, the port number, his username and password to have access to it. After all of this is done, he will be able to add any number of employees and branches, as the application grows in popularity. To create a new Branch, he will have to input this query with the appropriate values:
INSERT INTO Branch (BranchNumber,NumberofEmployees,Country,Manager_SSN) VALUES (int, int, varchar, int);
And to create a new Employee, he will have to input this query with the appropriate values:
INSERT INTO Employee (SSN,HashedPassword,FName,MName,LName,Address,City,Province_State,Country,EPosition,Salary,EmailAddress,PhoneNumber,Super_SSN,BranchNumber) VALUES (int, varchar, varchar, varchar, varchar, varchar, varchar, varchar, varchar, varchar, int, varchar, varchar, varchar, varchar);

The 10 queries that were asked to be written:

1. SELECT Gu.Fname, Gu.MName, Gu.LName,
   Pri.HomeType, Pa.PaymentType, Pa.PaymentStatus,
   R.TotalPrice, R.SigningDate
   FROM Guest as Gu NATURAL JOIN Pricing as Pri NATURAL JOIN Payment as Pa,
   Rental_Agreement as R
   where R.PricingID = Pri.PricingID
   ORDER BY Pa.PaymentType ASC, R.SigningDate DESC

2. Create VIEW GuestListView AS
   SELECT *
   FROM Guest natural join Guest_phone
   Order by BranchNumber ASC, GuestID ASC

3. Select * from Rental_Agreement Natural join Payment
   where TotalPrice = (Select min(TotalPrice) from (Select * from Payment where paymentstatus = 'Completed') as x)

4. Select Property.*, OverallRating
   from Property natural join Review
   where rented = '1'
   order by branchnumber asc, overallrating asc

5. Select * from Property where rented = '0'

6. Select * from Rental_Agreement
   where (date_part('day', startdate) <= 10 and date_part('day', enddate) >= 10)
   or (date_part('month', startdate) < date_part('month', enddate) and date_part('day', enddate) >= 10)

7. Select fname, mname, lname, ssn, country, branchnumber, salary, super_ssn
   from Employee
   where salary >= 15000
   order by super_ssn asc,
   ssn asc

8. If It's a Room:
   Select Pri.Hometype, Hst.HostID, Hst.FName, Hst.LName, Pro.address, RA.TotalPrice, Pay.paymenttype
   from Pricing as Pri, Property as Pro, Room as Ro, Host as Hst,
   Rental_Agreement as RA, Payment as Pay
   where Hst.HostID = Pri.HostID and RA.PricingID = Pri.PricingID
   and Pay.OrderID = RA.OrderID
   and Pri.PropertyId = Pro.PropertyID and Pri.HostID = Pro.HostID
   and Pro.PropertyID = Ro.PropertyID
   and Pri.RoomID is not NULL and Ro.RoomID = Pri.RoomID
   order by Pri.pricingID asc

   If It's the entire Property:
   Select Pri.Hometype, Hst.HostID, Hst.FName, Hst.LName, Pro.address, RA.TotalPrice, Pay.paymenttype
   from Pricing as Pri, Property as Pro, Host as Hst,
   Rental_Agreement as RA, Payment as Pay
   where Hst.HostID = Pri.HostID and RA.PricingID = Pri.PricingID and Pay.OrderID = RA.OrderID and Pri.PropertyId = Pro.PropertyID and Pri.HostID = Pro.HostID
   and Pri.RoomId is NULL
   order by Pri.pricingID asc

9. Update Guest_Phone
   set Phonenumber = '613-612-1193'
   where GuestID = (Select GuestID from Guest where FName = 'Burks' and LName = 'Ivana') and PhoneNumber = '1-324-981-8037'

10. CREATE FUNCTION FirstNameFirst(FName varchar, LName varchar)
    RETURNS Varchar AS
    $$
    BEGIN
        return FName || LName;
    END;
    $$
    LANGUAGE plpgsql;

Select FirstNameFirst(Data, Base)