Group Number: **Section-4_Team-3**

Course: Database Management System (IT214)
Instructor: Minal Bhise
Teaching Assistant: Vaishnavi Mam

Name:

Yash Shah	201901210
Dev Sanghvi	201901231
Darshil Shah	201901232
Anuj Jhawar	201901254

Travel Agency Management:

Section1 : Final version of SRS

1. Introduction

1.1 Purpose

- In today's busy and fast world every person needs a break from their hectic schedule and wants to go out from their home and spend some time with their family and friends and freshen themselves.
- The best way to fulfill their desires of roaming away from home is traveling and touring to their desired destinations.
- Tourism is the act and process of spending time away from home in pursuit of recreation and relaxation.
- The tourism industry has generated a lot of revenue in recent years. So now it has become one of the main income sources for many countries.
- Tourism also helps in understanding the various cultural, geographical, and historical cultures of each country. It opens a whole new window to the world for traveling.
- Tourism also provides jobs for people. People are needed to provide tourists all that food, drink, transportation, Photography, and lodging.
- To make all these things related to tourism, easy and hassle-free, the user needs a document or application which can help them to plan their trip with efficiency.
- Implementing effective marketing strategies like providing deals and discounts on kinds of stuff or providing sanitization and cab services from the hotel management itself to promote the hotel's services.

1.2 Intended Audience

- Users who want to plan a trip can use various facilities given by various hotel agencies.
- Hotel agencies who want to improve the services provided by them to attract more and more customers can compare their strategies and deals with their rivals and can improve their deals.
- Developers who want to make an application related to the tourism industry.

1.3 Product Scope

- The foremost purpose of the product is to provide services to the customer as well as the hotel agency.
- It will also help the users to maintain the booking of cabs, bus tickets, railway reservations, flights, hotels all in a single place.
- User-friendly booking options:- As far as the travel industry is concerned, if a user faces any difficulties in booking their travel destinations, modes of transport, accommodation, etc they will approach other Travelling Management Systems. To avoid losing business or customers, building a user-friendly booking tool is most important.
- It will provide users with a wide range of different facilities to choose from according to their choices and budget.
- Store user profile's data:- Tourism Management System is an integrated software developed for tour operating companies. The main aim of this project is to help tourism companies to manage their customers, vehicles, and agents. It makes all operations of the tour company easy and accurate. The standalone platform makes tourism management easy by handling agency requests and providing services for the customers located in different parts of the various cities. Different modules have been incorporated in this project to handle different parts and sectors of the tour management field.
- Wide Inventory:- Maintaining a large number of hotel, car, bus vendors, agents in travel management systems are most beneficial. If you don't have enough availability of low-cost accommodation, it may tend towards loss of great market exposure eventually resulting in loss of business.
- Updates on Availability and Non-Availability of Resources:- If any Air/Hotel/Car service is not operational due to any reason, the user expects travel management to send a clear message that this Service is Out of Policy.
- If any kind of problem has occurred during the trip then the user wants constant support from the place where he has booked/planned his trip which can be provided by us.

1.4 Description

The tourism industry mainly works on trust. Whenever a person books his or her trip to any location via the internet then he/she mainly worries whether the tickets they have purchased have been confirmed or the trip to which they are going by the help of the tourism agency whether it would be successful and enjoyable or would turn into ashes. Considering the COVID pandemic

situation the customer is always worried whether the stay or the hotel where they would be staying is perfectly safe or whether the food that they are being served is hygienic. So with the help of our database, we would like to assure the customers by adding some attributes which ensure whether the hotels they are booking or the flights/trains/buses by which they are traveling would be safe or not by giving an option of ratings of that particular service. We would also think of the host company which is providing the services to the customer that has planned for the trip whether they have completed the payments or not. We would also provide the information of the different places like what are the main points of attraction, etc.

Our database will be equipped with tables like booking_information(As by name suggests it will contain the booking information of customer),customer_information(As the name suggests it will hold the customer information given while making the booking),hotel_information(This table will contain information related to hotel), place_information(It will contain the information regarding the place), etc. Our database will also have attributes like booking_id, age, name, place, payment, place_name,place_locations, etc. Our attributes will vary from character varying to an integer to boolean values.

1.5 References

- https://pdfcoffee.com/tourism-management-system-pdf-free.html
- https://www.neovasolutions.com/2020/05/09/key-features-in-a-travel-management-system/

2 Fact-Finding Phase

2.1 Background Readings:

Existing Systems:

- In the present system, users/customers have to approach various agencies to find details of places and to book tickets which requires a lot of time and effort.
- A customer may not get desired information from these agencies and it has a high chance of being misguided.
- So, therefore, the existing systems are very time-consuming and it becomes difficult for the customers to plan their journeys.
- If more than one customer or manager accesses the applications using file systems, then there is a high probability of system failure or the operations that are being accessed by the user not being completed which can lead to negative impacts on the user.

- Many times handwritten books or registers are being used. For which there is a high probability that data might be lost. There are chances that other people except the manager manipulate the data without the manager's consent.
- If a system failure or system crash is encountered then there is a high chance of all the data being wiped out and cannot be recovered as we are not using a perfect database system to maintain the records.
- At the time of submission of the reports monthly and yearly it's very difficult to keep track of such a huge record set and manually searching it, compiling it and at the end.
- Manual report generation is less efficient and may cause inaccuracy in outputs.

Proposed Systems:

- The proposed system would be a web-based application that would be more user-friendly as the user would not have to give a lot of time to the application and the features that would be present inside the application would do the work on their own.
- So here there would be more software interactions rather than the customer trying too hard to plan his/her trip.
- The work of the customer would be reduced to just some clicks from his/her mouse.
- The system allows easy access to relevant information like hotel name, room number, vehicle number, date of arriving and departure, payment options, etc., and makes necessary travel arrangements.
- By using a Database Management System, we can wipe out the possibility of data inconsistency and erasure of whole data and if a system crash is encountered while using a database management system we can recover the whole data which is being wiped out temporarily.
- The new system can calculate any amount of data and to any complexity which makes the system successful and the needs of the consumer can be fulfilled.
- By the use of SQL, the user can perform any operations like retrieval of data, updating of data, removal of data, and many other such operations very easily.
- User interface is an important feature of the proposed system which provides jobs to people without any specialization or special training.
- The Database can also be adopted by various Travel agencies because of above mentioned features with uniqueness.

Make my Trip:

Make my Trip is one of the most known platforms for planning a trip. It has a rating of 4.4 on the play store which shows that it is one of the reliable platforms available currently. After going through their site we found out various things which will help us develop a suitable database.

Working and flow:

- When we visited the website. We were asked to register.
- For registration, we can do it either with an email id or mobile number.
- Then my mobile number and email id were verified.
- We were asked to complete our basic profile which includes
- For planning a tour we were asked to choose a start and end destination.
- We were also asked to choose the dates when we would like to proceed with the trip.
- Additional and useful features were given by which we can plan our activities depending upon our interest like romantic sites, outdoor activities, indoor activities, Culture activities of destination places, adventures of beautiful wildlife, Museums, Historic Sites, Visiting the beautiful and ancient temples.
- Depending upon the preference of our activities, We were given a day-to-day plan. We were asked to choose an agent for our trip.
- We were asked to choose a preferred contact method.
- After that, we were asked to pay.

Features:

If we go back 10 years we only had options of a touring agency. Now there are many competitors.

- So many websites like making my trip, trivago, etc all are giving various features like paying at the hotel and no cancellation fees.
- Features like booking of hotels, cabs, trains all are given to make our trip less stressful and all this convenience adds satisfaction while traveling.
- We were given the option to customize our trips based on our preferences.
- Discounts were available for first-time users and credit card, debit cardholders.

  Flaws:
    - It is not a multilingual platform. Someone not familiar with English can find it hard to plan a trip.

Goibibo:

Goibibo is India's leading online travel booking brand providing a range of choices for hotels, flights, trains, bus and cars for travelers. This is the most trusted user experience platform, be it in terms of quickest search and booking, fastest payments, settlement, or refund processes.

Working and Flow:

- First of all, We have to register using a mobile number, name, and email id. Once registered, we have to log in to goibibo.
- Goibibo provides bookings for Flights, Hotels, Trains, Cabs, and Buses. There can be three ways to travel like One Way, Round Trip, Multicity. So, select accordingly.
- Now, we have to select our location and the destination where we want to go.
- There are many different destinations to choose from like Beach Vacations, Mountains Calling, Indian Pilgrimages, Party Destinations.
- You can also see the different ratings of Hotels given by users. And all the hotels are certified by the Government of India.
- Now for payment, we can have different options to pay like debit cards, credit cards, Paytm, and many more.
- The user gets a flat 12% off on the first flight booking and 50 Rs of cash bonus on goibibo wallet.
- Users can see their Booking History under the “My trips” section.
- Users get tribe coins on every trip and they can use them to redeem some offers.

Features:

- Cancellation problems seem to be a huge problem across the globe. So, Goibibo offers a free cancellation facility, if the user cancels their booking at least 4 hours before departure.
- It is very difficult for any customer to choose between different hotels and destinations. Goibibo shows photos, ratings, reviews of Hotels given by experienced customers. So, the user can choose wisely.

Online Tourism Management:

The online tourism project observed from an online tourism database project is a well defined project where the project maker has worked to make an application for travelling companies containing the most basic database to the application to plan a whole trip.

Working and Flow:

- Initially when starting the application a home page is not shown and directly a page to register or log in is shown.

- So we need to register or log in (if already registered) to the application to see the different features which can be performed.
- Then if successfully registered we were asked to log in using the id and password generated during registration.
- Now all the operations that can be performed in the application can be seen.
- Now the task which we need to do must be done by selecting any of the options like booking air tickets or train tickets, booking hotels or managing tour packages,etc.
- Now primarily to plan a whole trip we need to initially click on manage tour package where it asks for the destination where we want to go.
- Then we can see the different packages with different specifications for a particular destination.
- Then the platform asks us to book tickets for our destination.
- It shows different options varying by the prices of different modes of transport.
- Now we need to select one of the transportation mode and book the tickets.
- Then it displays the booking details which needs to be verified by us.
- Then the portal for performing payment needs to be done.
- It gives us options like paying using debit card or credit card or net banking.
- If we then perform the payment successfully then the payment successful message is displayed and the details of the whole tour are also displayed.
- It also gives the option of cancellation where we can travel or tickets.
- In the penultimate step, the software gives a portal for the feedback of the trip if done else the feedback of the platform.
- Lastly we need to log out from the platform to exit the application.

Flaws:

- Initially it does not show us the home page as the person who just wants to see the different features provided by the platform could get irritated as every time he enters the platform he needs to log in even to just visit the application.
- It does not provide many offers and schemes to attract the customers.

References:

- https://www.goibibo.com/
- https://www.makemytrip.com/
- https://pdfcoffee.com/tourism-management-srs--pdf-free.html

2.2 Interviews

1. Interview Plan

System: Happy Tourism

Project Reference: DBMS/Sec4/Group3

Interviewee:

1) Hritvik Prajapati (Role Play) **Designation**: Traveller

Interviewer:

1) Yash Shah **Designation:** Business Development Executive

2) Darshil Shah **Designation**: Developer

Date: 7/10/2021 **Time:** 16:00

Duration: 30 minutes **Place:** Google Meet (Virtual Interview)

Purpose of Interview:

- Preliminary meeting to identify problems and requirements regarding the Tourism Management Database from the customer's perspective.

Agenda:

- Features that must be included in the database.
- Current security procedures.
- Different offers that can attract a traveller.

Documents to be brought to the interview:

- Rough plan of the database.

Interview Summary:

System: Happy Tourism

Project Reference: DBMS/Sec4/Group3

Interviewee:

1) Hritvik Prajapati (Role Play) **Designation**: Traveller

Interviewer:

1) Yash Shah **Designation:** Business Development Executive

2) Darshil Shah **Designation**: Developer

Date: 7/10/2021 **Time:** 16:00

Duration: 30 minutes **Place:** Google Meet (Virtual Interview)

Purpose of Interview

- We had a discussion regarding the different requirements that are needed by us as database developers from a user/customer point of view.
- Providing user Reliability, ease of access, and data security.
- Different features like hotel booking, ticket bookings, and ratings of different places and hotels and airlines must be provided.
- Difficult for uneducated people to book tickets online.
- Offers are provided at some occasions/festivals only in other tourism databases but travelers who travel the whole year can also be provided with some more offers in the time which is not around the festivals.
- Providing appropriate filters and hence tries to build a user-friendly system.
- Users have to log in every time they enter the application, so remembering the details of the user for the utmost one month can make the system more friendly to the user.

2. Interview Plan

System: Happy Tourism

Project Reference: DBMS/Sec4/Group3

Interviewee:

1) Mayank Tenzing (Role Play) **Designation:** Hotel Manager

2) Akash Chopra (Role Play) **Designation:** Airline Company Manager

Interviewer: 1) Dev Sanghvi **Designation:** Developer

2) Anuj Jhawar **Designation:** Developer

Date: 7/10/2021 **Time:** 19:30

Duration: 30 minutes **Place:** Google Meet (Virtual Interview)

Purpose of Interview:

- Preliminary meeting to identify problems and requirements regarding the database from a manager of a hotel and an airlines company.

Agenda:

- Features that must be included in the database.
- Current security procedures.

Documents to be brought to the interview:

- Rough plan of the database.

Interview Summary:

System: Happy Tourism

Project Reference: DBMS/Sec4/Group3

Interviewee:

1) Mayank Tenzing (Role Play) **Designation:** Hotel Manager

2) Akash Chopra (**Role Play**) **Designation:** Airline Company Manager

Interviewer: 1) Dev Sanghvi **Designation:** Developer

2) Anuj Jhawar **Designation:** Developer

Date: 7/10/2021 **Time:** 19:30

Duration: 30 minutes **Place:** Google Meet (Virtual Interview)

Purpose of Interview:

- We had a discussion regarding the different requirements that are needed by us as database developers from a manager's point of view.
- Creating suitable relationships among tables which would make accessibility and retrieving records easy.
- Whenever a manager of a company wants to know the growth of his/her company in that particular month/quarter/year then if the data of the tables and the relationships between tables are perfectly balanced then it would be very easy to perform the checks.
- Verifying details of a customer in a hotel/airline is a bit difficult, so the details given by customers online should be the same as the Aadhar card or any identity card.
- Whenever a customer after using the services provided by the hotel/airlines gives us feedback about the services, that feedback should be immediately shared with the hotel/airline company so that they can improve the services if the feedback is not good for the company.

3. Interview Plan

System: Happy Tourism

Project Reference: DBMS/Sec4/Group3

Interviewee:

1) Rohit Kakaria (Role Play) **Designation:** Software Developer

Interviewer: 1) Yash Shah **Designation:** Developer

2) Anuj Jhawar **Designation:** Developer

Date: 7/10/2021 **Time:** 21:00

Duration: 30 minutes **Place:** Google Meet (Virtual Interview)

Purpose of Interview:

- Introductory meeting to identify problems and requirements regarding the database and the software to be made by the developer.

Agenda:

- Documents and information of customers that is required for the bookings
- Features that must be included in the database.
- Current security procedures.

Documents to be brought to the interview:

- Rough plan of the database.

Interview Summary:

System: Happy Tourism

Project Reference: DBMS/Sec4/Group3

Interviewee:

1) Rohit Kakaria (Role Play) **Designation:** Software Developer

Interviewer: 1) Yash Shah **Designation:** Developer

2) Anuj Jhawar **Designation:** Developer

Date: 7/10/2021 **Time:** 21:00

Duration: 30 minutes **Place:** Google Meet (Virtual Interview)

Purpose of Interview:

- We had a discussion regarding the different requirements that are needed by us as database developers from a web developer or an application developer's point of view.
- Password should be first encrypted then only it should be inserted in our database.
- The data inconsistency and the redundancy should be as low as possible in the database to make the application more error-free and for better analysis of outputs.
- If a user tries to make more than one account then he/she should be prohibited from doing so by taking their phone number and making it a primary key. So by doing so there is no chance of having different accounts with the same phone numbers.
- A similar thing that is stated above can also be performed with a login id. No two users can have the same login id.

2.3 Questionnaire

In this part, to know the experiences of different users and consumers we have made a questionnaire containing different questions which will help us to know the requirements of people from different sections of occupation and the flaws of the current system. We have used different types of questions like multiple choice, Yes and No, short answers, and the linear scale.

1. What occupation do you belong to? *
a) Student
b) Teaching Assistants (TAs)
c) Professor
d) Other

2. How frequently do you travel? *
a) Once in a week
b) Once in a month
c) Once in a quarter
d) Once in a year
e) Not so frequently

3. Which platform do you use more frequently?
a) MakemyTrip
b) GoIbibo
c) Trivago
d) Airbnb
e) Other:

4. How much do you rate your preferred platform?

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

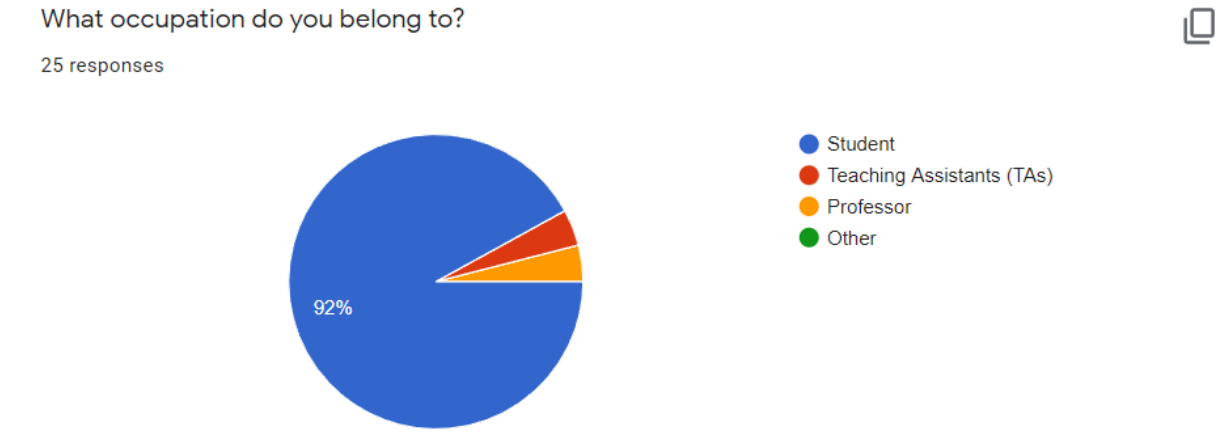
5. Which facilities do you think are not present in your preferred platform? And can you please suggest to us which facilities should be added? *
_____
_____

6. Do you feel that your journey comes out stress-free? *
a) Yes
b) No

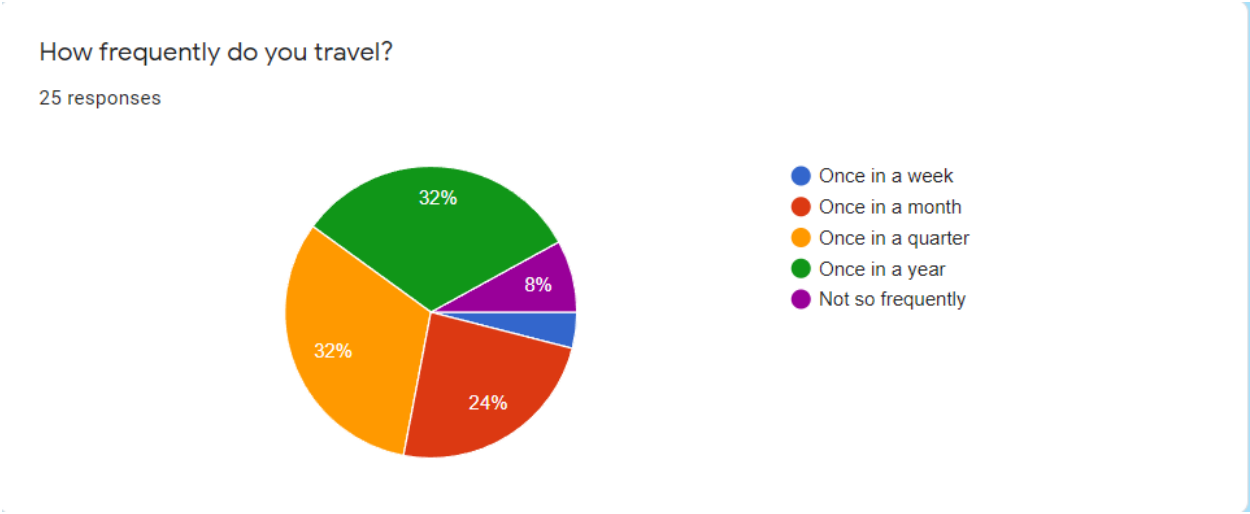
7. If No, please describe the dissatisfactions that you felt?
_____

_____

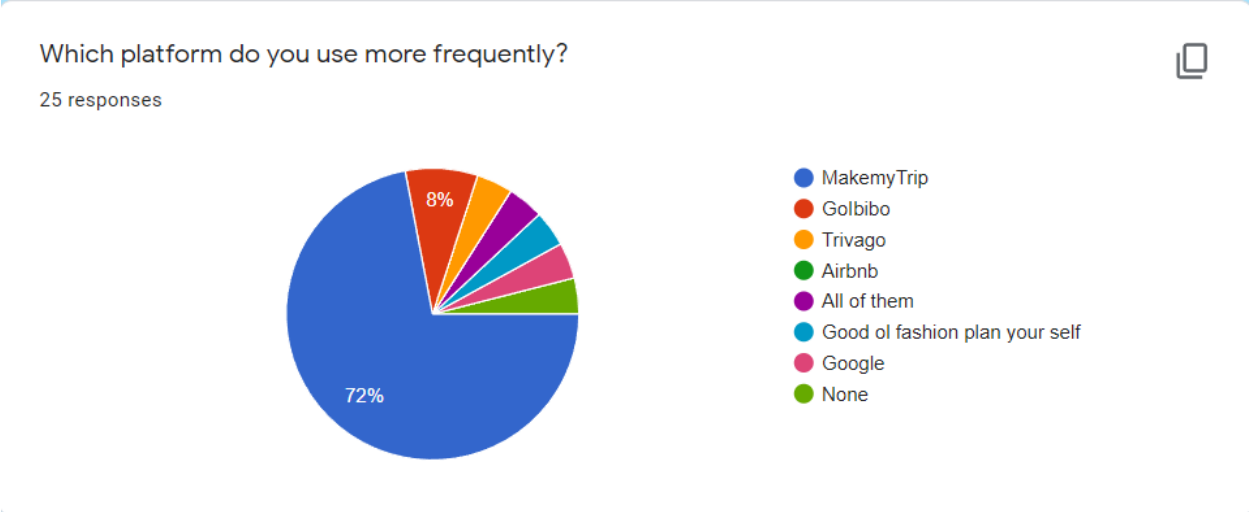
Summary



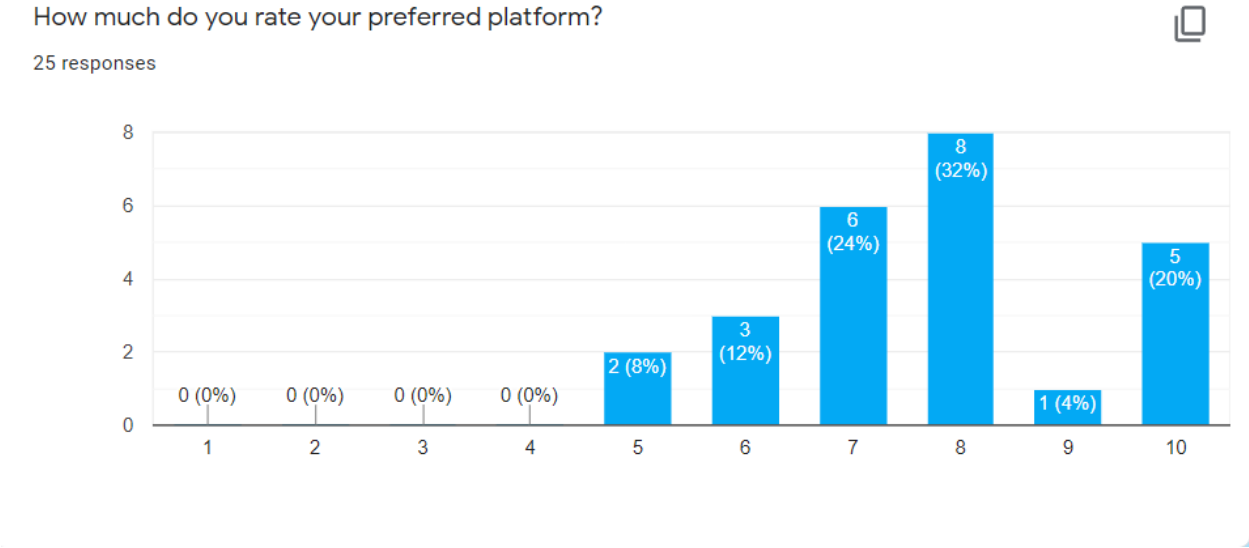
- The survey was being shared with other people and the main idea behind involving the above question is to know what is the type of participants that are answering the questions.



- The above question collects the information about the frequency of travel that is done by the user as it helps in making the database more detailed as the people who travel more can give us detailed feedback and they can be given more and more schemes and offers to stick them to our particular platform and a new user or the user who uses the application less frequently can be given more and more offers to make them greedy to travel with our platform only.



- This question was asked to identify the choices of people. Which will help us to understand which platform is widely used so that we can take a look at the platform for designing our databases.
- Various choices were given to the people, From which we found that the majority of people are using MakemyTrip.



- This question was particularly asked to find whether people are happy with the currently available platforms or not.
- We found that people are neither unsatisfied nor happy with the current platform. We found that we need to add some functionality to our database.

Which facilities do you think are not present in your preferred platform? And can you please suggest us which facilities should be added?

25 responses

NA

Security

No, i am satisfied

If we can track our whole journey then it would be much better.

Multilingual and multi-currency,support should be included a t no additional charge,clear pricing without hidden fees.

Finding good hotels with desired facilities and sometimes high charges of staying

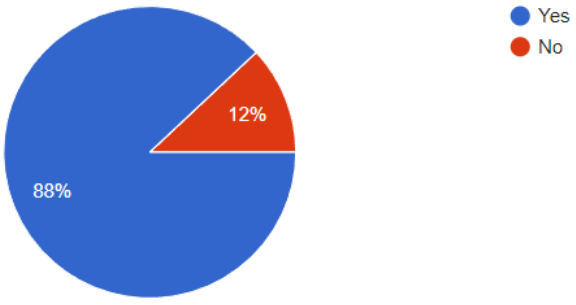
None

Ability to automatically choose flights

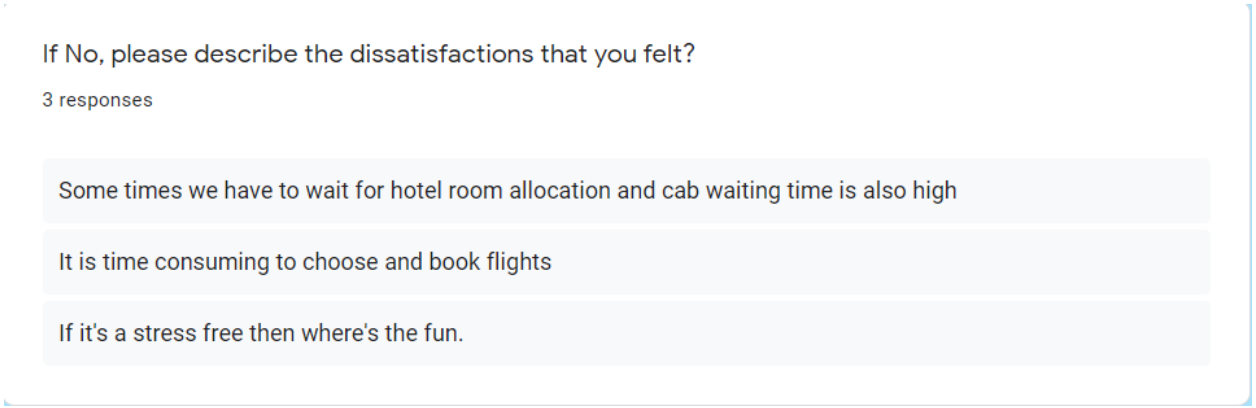
- The motive for asking this question was to identify the problems faced by the users so that one can work upon them and try to build a system more user-friendly.

Do you feel that your journey comes out stress-free?

25 responses



- This question becomes important from the customers' point of view as it gives us an idea of how the user is stressed more on planning everything, coordinating travel arrangements rather than enjoying their trips.



- This question is important from the customer's point of view because we can get to know where we have to improve or what changes we can make to satisfy the customers.

➔ Finally, we come to a conclusion from the above questionnaire that some flaws that the normal users feel need to be looked upon and encountered.
➔ The database system that we built must be such that it gives the output as fast as it can be and the risk of data inconsistency should also be reduced and the crashes should be made as little as possible so that the reputation of the system will be met up.

2.4 Observations

- Whenever a particular customer wants to travel to a particular destination then in these difficult times of COVID - 19 pandemic he/she is mainly concerned with what are the number of days that he/she needs to be quarantined.
- We have observed that nowadays data of many people get leaked so to ensure the privacy of customers, we will encrypt the password before inserting it into our database.
- Users have to log in every time they enter the application, so remembering the details of the user for the utmost one month can make the system more friendly to the user.
- Providing premium facilities like giving discounts on booking tickets, providing free breakfasts, etc.
- Sometimes customers find it difficult to make payment because of less payment options.

3 Fact-Finding Chart

Objectives	Technique	Subjects	Time Commitment
To get an overview of the current tourism trends for the Tourism Database Management System	Background Reading	Articles and Websites	2 days
To gain information regarding the number of days that are needed to be quarantined at a particular destination.	Observation	Personal Observations	2-3 hours
To get an idea of what features we could include on top of existing apps by understanding how current existing tourism agencies work.	Background Reading	Websites	5 hours
	Interview	Customer/Manager/ Developer	1 hour
To establish what records and resources are kept.	Interview	Customer/Manager/ Developer	2-3 hours
Functionalities that customers want in the interface of the system	Questionnaire(Using Google Form)	Public Survey	2 days

4 List Requirements

- Safety Requirements:- For safety requirements, the system shall save the data if the server goes down or any other harmful things happen to the server for the sake of safety.

- Security Requirements:- For security purposes, the application will be password protected or we can have an OTP system,i.e. when the user enters the application, the One Time Password (OTP) will be sent to the respective user.

- Authentication: The user should be strictly advised that he/she should not share the login details and the same should be applied with the manager and developer. The planning and the logical structure of the database should not be shared with anyone except the trustful. Similarly, the applications made using the database must be accessible using trusted platforms only and not by any third-party apps.

- Reliability: The complexity of the system should be kept in such an order that whenever we need to manipulate any entity or table data it should be done easily. If we want to insert new data/attributes to a table then it should not create data redundancy or inconsistency in the database. So the system needs to be made such that it can be maintained whenever needed.

- Portability: The application will be easily accessible on any OS-based system.

- Data requirement: For different classes of users, Data requirements will be different.
    - For customers who want to book various facilities will require to submit one of the id proofs and need to complete their user profile.
    - For owner/manager: Licence of their hotel/Touring agencies will be required.

- Transparency of Data:- Using Data integrity, users know what data is being collected and who has the right to access it and how they are able to interact with it and so all that information will be inaccessible to unauthorized users and privacy of data is at its fullest.
- Integrity:- Database should be error-free by storing accurate and consistent data in a database

5 User Classes and Characteristics

Customers

- To view various Hotels, tour plans, and tickets for traveling that are currently available.
- Customers can watch their past actions/activities.
- View-only mode

Manager (Admin)

- To view details of his/her customers.
- Booking records like to check what the number of customers did increase compared to the previous year and many more different functions.
- To change/edit the tour agencies, hotels, and new facilities that they are going to provide.
- To edit data of tour plans and hotels that are unavailable (or sold).

Developer

- To edit/change functionalities of the database.
- To add/remove policies and norms.
- To edit/view details of Hotels, touring agencies.
- Can manipulate and maintain the database.

6 Operating Environment

Software Requirements:-

- Android (Version 5.0 or above) or IOS

- Operating System - Windows(Any Browser)
- Database System - PostgreSQL

Hardware Requirements:-

- Smartphone (Android or iPhone) or Computer with stable internet connectivity.

External Interface Requirements:

- **Hotels and Flights Database connectivity**:- Provides data of the availability of rooms and flights and other resources required for customers. And we also need to access certain people's locations for our feature to work efficiently.

7 Product Functions

Customer:

- Can create/edit their profiles made during registration.
- Can check/book available tour plans, tickets, and hotels.
- Can cancel the booking of the tour plane, tickets, and hotel bookings.
- Can check or track their history of booking.
- Can sort the available facilities using their personal preferences like ratings, prices of hotels and tickets and the trips which can come under their budget.

Manager:

- If a customer requests to edit a particular booking data then the manager has the right and privilege to edit the specified data.
- To edit information regarding hotels, airlines, and tour agencies.
- To predict their performance during the present month/quarter/year they will have access to the bookings.
- Can provide offers and schemes on various facilities to the user.

Developer:

- If a customer requests to edit a particular field (login details) then the manager has the right and privilege to edit the specified data.
- To edit information regarding hotels, airlines, and tour agencies.
- To add/delete a hotel, airline, or tour agency.
- To predict their performance during the present month/quarter/year they will have access to the bookings.

8 Privileges

Functions	Customer	Admin/Owner	*Developer*
View Customer Details	YES	YES	YES
Update Customer Details	NO	NO	YES
View Destination Details	YES	YES	YES
Update Destination Details	NO	NO	YES
Update Booking Details	YES	YES	YES

Update Hotel Details	NO	YES	YES
Compare different Places	YES	YES	YES
Sort the places based on preferences(like price, facilities, ratings, review, etc)	YES	YES	YES
To edit/change functionalities of the database	NO	NO	YES
To add/remove policies and norms	NO	NO	YES

9 Assumptions

- Customers/Owner/Developers have access to a computer.
- Customers/Owner/Developers have a basic knowledge of using a computer.
- Customer/Owner will require an internet connection.
- We have access to a tourism database.

10 Business Constraints

- There is a time constraint since we have to develop the Database System in a short span.
- We need to use as few tables as we can to reduce the complexity of the database which is time-consuming.
- As the number of customers, hotels, and tour agencies can be high, the Storage required to store all this information will be high which is expensive.
- A team of developers or a developer is required to maintain the database.

Section2 : Noun Analysis

Problem Description:

In today's busy and fast world every **person** needs a break from their hectic schedule and wants to go out from their home and spend some time with their **family** and friends and freshen themselves. The best way to fulfill their desires of roaming away from **home** is traveling and touring to their desired **destinations**. Tourism is the act and process of spending time away from home in pursuit of recreation and relaxation. The tourism industry has generated a lot of revenue in recent years. So now it has become one of the main income sources for many countries. Tourism also helps in understanding the various cultural, geographical, and historical cultures of each country. It opens a whole new window to the world for traveling. Tourism also provides jobs for people. People are needed to provide tourists all the food, drink, transportation, Photography, and lodging.

In the present times, due to the covid situation people are not yet willing to travel to different destinations. So, the tourism industry which was running at a tremendous speed before the **covid-19** situation is now lagging behind and the people who were solely dependent on tourism are now unemployed and finding a way to cope up with this tough situation.

Why do we need a database?

The tourism industry has generated a lot of revenue in recent years. So now it has become one of the main income sources for many countries.In the **existing system**, each task is carried out manually and processing is also a tedious job. In the system travelers were maintaining time table details manually in **pen and paper**, which was time taking and costly. The **traveler** is not able to achieve its need in time and also the results may not be accurate. Because of the manual maintenance there are a number of difficulties and drawbacks exist in the system.

There are many cons of existing system(file system,pen and paper) like

- Large volumes of tourists, hotels, flights, cabs **data** can not be stored in one place.
- Multiple **users** can not read and modify the data at the same time.
- Manual maintenance and time consuming which can result in error.

- When we talk about pen and paper systems, they are not durable. There are chances of losing the data.
- While access to **text files or spreadsheets** can be secured, once someone has access to a file, they have access to all data within that file.

Whenever using a hand-written **register or book** for entering a particular data, it can have many difficult consequences which cannot be solved practically or if solved then it could be much expensive or time consuming. Thinking from a chain of hotels perspective, if a **branch manager** of the hotel situated at Mumbai wants to access a particular data from database of the other branch situated at Chennai, then if the hotel uses hand-written registers then it would be impractical to send the registers from Mumbai to Chennai via any mode of transportation.

To make all these things related to **tourism** easy and hassle-free, the **user** needs a document or application which can help them to plan their trip with efficiency.To overcome all of these disadvantages we are going to design an efficient and secure database and with the help of a database management system we will be able to perform various operations on our database like sorting the **facilities** as per our preferences which will make the database more robust and **customers** can know all information regarding trips with a couple of mouse clicks.Implementing effective **marketing strategies** like providing **deals and discounts** on kinds of stuff or providing **sanitization and cab services** from the hotel management itself to promote the **hotel's services**.

Sometimes **people** visit a place and find that there are no hotel,flights available. Which makes their trip not so joyful. By using our database they can check which **hotels** are available. They can also compare the facilities available. All these things will help them to plan their trip stress free.

Also some hotels are opened newly or are not so famous and require attention of **customers** hence our database will include all such features which will be beneficial for **hotel owners** too.

During these tough times many **livelihoods** based on tourism are finding it difficult to survive. So to overcome these **problems** all the **hotels and touring agencies** will have access to our basic tourist data. Which will help them to attract those tourists with the help of some offers. It will help both the tourists and organizations.

The **tourism industry** mainly works on trust. Whenever a **person** books his or her trip to any location via the internet then he/she mainly worries whether the **tickets** they have purchased have been confirmed or the **trip** to which they are going by the help of the tourism agency whether it would be successful and enjoyable or would turn into ashes. Considering the COVID **pandemic** situation the customer is always worried whether the stay or the hotel where they would be staying is perfectly safe or whether the **food** that they are being served is hygienic. So with the help of our **database**, we would like to assure the customers by adding some attributes which ensure whether the hotels they are booking or the flights/trains/buses by which they are traveling would be safe or not by giving an option of **ratings** of that particular service. We would also think of the host company which is providing the services to the customer that has planned for the trip whether they have completed the payments or not. We would also provide the information of the different **places** like what are the main points of attraction, etc.

The people who are now willing to come to normalcy are unable to trust any of the tourism agencies. After the covid situation, the cases of robbing the people and using unfair means to make profits high have increased. Many of the agencies are using many different types of methods to rob people such as in online platforms they take the payment through online mode and run away with the money. The **uneducated portion of the society** are being lured by offering them high discounts and unbelievable offers. The platforms which provide such schemes only accept the **payment** through online mode so that they can run away with people's money. And the people also get lured by these schemes. So, our database would provide an easy interface which can also help the uneducated portion of the society by giving them easy and suitable **features** like the ratings of the tourism agencies. There are also many cases where the tourism agency does not provide the services that were initially promised. This problem can also be solved by the feature of rating different tourism agencies.

Beneficiaries of our Database

Users who want to plan a trip can use various facilities given by various hotel agencies. Users who want to book tickets for **trains, flights** in an ordered manner. Sometimes users have a certain **budget** to spend, our database will help them to sort available facilities accordingly. It will also help them check their status of booking. During this covid times customers prefer contact less payment options for obvious reasons. Users can check the number of quarantine days at a particular destination. Users can also look for discounts available at different places by different organizations. There are some facilities which are

not so famous but value for money which gets unrecognized by users in general. This won't be the case here.

Hotel agencies who want to improve the services provided by them to attract more and more **customers** can compare their strategies and deals with their rivals and can improve their deals.There are many uses which should be available for a manager i.e. an airline/train/bus company manager or hotel **manager** who needs to see all the data (which he is allowed but not others). He can perform some of the functionalities like seeing the details of the customer who is using his/her **services** to verify whether he/she is an honourable citizen of the country or not, then he/she can also see the annual/monthly/weekly/daily growth of his/her services and calculate their **profits**, then he/she can also have some features like editing the information of his/her own services like the increase/decrease of **prices**, changes in the restaurant menu, etc. The airline/train manager can have some features like asking the customer whether he/she would have some **dinner/lunch/breakfast** to eat during their journey and whether they would choose vegetarian/non-vegetarian and many other different features. The manager also needs to state to the **database holder** whether the services that they are providing are available for that particular day or not and if not then the customer should also be stated, so the manager has the right to display whether they are available or not.

Developers who want to make an application related to the tourism industry. **Developer** has a right to edit the database based on the request made by the customer or manager. The **developers** have the privilege to edit the login details based on customers request and to improve the functionality by adding new features of that hotel or airline company. If any hotel or airline company wants to measure their performance over some period of time then developers can include these features and so **companies** can work accordingly and can make the system more friendly to Manager class.

What horizons can the product reach?

The foremost purpose of the product is to provide services to the **customer** as well as the **hotel agency**. It will also help the **users** to book **cabs**, **bus tickets**, **railway** reservations, **flights**, **hotels** all in a single place.As far as the travel industry is concerned, if a user faces any difficulties in booking their travel destinations, modes of transport, **accommodation**, etc they will approach other Traveling Management Systems. To avoid losing business or

customers, building a user-friendly booking tool is most important. It will provide users with a wide range of different **facilities** to choose from according to their choices and **budget**.

Tourism Management System is an integrated software developed for **tour** operating companies. The main aim of his project is to help tourism companies to manage their customers, **vehicles**, and **agents**. It makes all operations of the tour company easy and accurate. The standalone platform makes tourism management easy by handling agency **requests** and providing services for the customers located in different parts of the various **cities**. Different **modules** have been incorporated in this project to handle different parts and **sectors** of the tour management field. Maintaining a large number of hotel, **car**, **bus vendors**, agents in travel management systems are most beneficial. If you don't have enough availability of low-cost accommodation, it may tend towards **loss** of great market exposure eventually resulting in loss of business.

If any **Air/Hotel/Car** service is not operational due to any reason, the **user** expects travel management to send a clear **message** that this Service is Out of Policy. If any kind of problem has occurred during the **trip** then the user wants constant support from the place where he has booked/planned his trip which can be provided by us.

What requirements must be checked before making a database?

For safety requirements, the **system** shall save the data if the server goes down or any other harmful things happen to the server for the sake of safety. For security purposes, the **application** will be password protected or we can have an OTP system,i.e. when the user enters the application, the One Time Password (OTP) will be sent to the respective user.The **user** should be strictly advised that he/she should not share the login details and the same should be applied with the manager and developer. The planning and the logical structure of the **database** should not be shared with anyone except the trustful. Similarly, the applications made using the database must be accessible using trusted platforms only and not by any third-party apps.The complexity of the system should be kept in such an order that whenever we need to manipulate any entity or table data it should be done easily. If we want to insert new data/attributes to a table then it should not create data redundancy or inconsistency in the database. So the **system** needs to be made such that it can be maintained whenever needed.

The application will be easily accessible on any OS-based system.For different classes of users, **Data** requirements will be different. **Customers** who want to book various facilities will be required to submit one of the **id proofs** and need to complete their user profile.For

owner/manager: Licence of their hotel/Touring agencies will be required.Using Data **integrity**, users know what data is being collected and who has the right to access it and how they are able to interact with it and so all that information will be inaccessible to unauthorized users and **privacy** of data is at its fullest.**Database** should be error-free by storing accurate and consistent data in a database.

Noun Analysis:

Nouns	Verbs
person	break
family	freshen
home	roaming away
destinations	travel
covid-19	lagging behind
existing system	carried out manually and processing
pen and paper	time taking
traveler	achieve
data	stored
users	can not read and modify
register or book	entering
branch manager	wants
text files or spreadsheets	access
tourism	easy and hassle free
user	plan
facilities	sorting
customers	knowing
marketing strategies	implementation
deals and discounts	providing
sanitization and cab services	providing

hotel's services	promote
hotel	available
people	visit
customers	attention
hotel owners	beneficial
livelihood	tough times
problems	overcome
hotels and touring agencies	access
tourism industry	works on trust
person	worries
tickets	confirmed
trip	going
pandemic	worried
food	hygienic
database	assurance
ratings	giving option
places	providing information
uneducated people	luring
payment	accepting
features	easy and suitable
hotel	book
trains	book
flights	book
user	trip
budget	spend
Hotel	services
customers	compare
manager	seeing data
services	verify

database holder	provide
developer	edit
developer	privilege
companies	work
Cab	book
profit	calculate
price	increase/decrease
dinner/lunch/breakfast	eating
customer	services
hotel agency	reservation
cabs	travel
bus tickets	management
railway	maintain
facilities	market
tour	benefit
budget	booking
vehicles	handling
modules	tend
sector	choose
cost	provide
accommodation	operating
loss	faces
message	help
problem	resulting
platform	aim
cities	expects
Licence	submit
owner	error free
manager	store

data	collect
database	interact
integrity	
privacy	
id proofs	
system	save
application	protected
user	should not share
database	should not be shared
system	made
login id	security
password	security
manager name	
date	arrival
tax	
payment	mode
booking id	unique
agency name	
agency id	
email id	
arrive time	travelling
depart time	travelling
landmark	
location	
hotel id	
rating	verify
feedback	improve
facility	
tour package type	

tour package id	

Rejected Nouns and Verbs:

Rejected Nouns	Rejected Verbs
Person : Irrelevant	Break : Irrelevant
Family : Irrelevant	Freshen : Irrelevant
Home : Irrelevant	roaming away : Vague
Destinations : Duplicate	Travel : General
Covid-19 : General	lagging behind : Irrelevant
existing system : Vague	carried out manually and processing : General
pen and paper : Vague	time taking : Irrelevant
Traveler : Duplicate	Achieve : Irrelevant
Data : Vague	can not read and modify : Irrelevant
Users : Duplicate	Entering : Vague
register or book : Vague	Wants : Irrelevant
text files or spreadsheets : Vague	easy and hassle free : General
Tourism : Vague	Knowing : Irrelevant
deals and discounts : Duplicate	Implementation : General
sanitization and cab services : Duplicate	works on trust : Vague
hotel's services : Duplicate	Worries : General
People : Irrelevant	Confirmed : Attribute
Customers : Duplicate	Going : General
hotel owners : Duplicate	Worried : General
Livelihood : Vague	Hygienic : Vague
Problems : General	Assurance : General
tourism industry : General	giving option : General

Person : Irrelevant	providing information : Irrelevant
Tickets : Duplicate	Luring : Irrelevant
Trip : Vague	Accepting : General
Pandemic : General	easy and suitable : General
Food : Duplicate	Book : Attribute
Database : General	Book : Duplicates
Ratings : Attributes	Book : Duplicates
Places : Duplicate	Trip : General
uneducated people : Irrelevant	Spend : Vague
Features : Duplicate	Services : Attributes
Hotel : Duplicate	Compare : General
User : Duplicate	seeing data : Irrelevant
Hotel : Duplicate	Verify : General
Customers : Duplicate	Provide : General
Budget : General	Edit : General
Manager : Duplicate	Privilege : Vague
Services : Duplicate	Work : General
database holder : Duplicate	Book : Duplicate
Developer : Duplicate	Calculate : General
Companies : Duplicate	increase/decrease : General
Profit : Vague	Eating : Vague
Price: Attributes	Services : Attributes
dinner/lunch/breakfast : Duplicate	Reservation : Vague
Customer : Duplicate	Travel : General
hotel agency : Duplicate	Management : General
Cabs : Duplicate	Maintain : Vague
bus tickets : Duplicate	Market : General
Railway : Duplicate	Benefit : Attributes
Facilities : Duplicate	Booking : Duplicate

Tour : General	Handling : Irrelevant
Budget : General	Tend : Vague
Vehicles : Vague	Choose : General
Modules : Vague	Provide : Duplicate
Sector : Vague	Operating : Irrelevant
Cost : Duplicate	Faces : Irrelevant/Vague
Accommodation : General	Help : Vague
Loss : General	Resulting : Vague
Message : General	Aim : Irrelevant
Problem : General	Expects : Irrelevant
Platform : Irrelevant	Submit : General
Licence : Attribute	error free : General
Owner : General	Store : General
Manager : Duplicate	Collect : Irrelevant
Data : Vague	Interact : Vague
Database : General	Save : General
Integrity : General	Protected : Irrelevant
Privacy : General	should not share : Vague
id proofs : Attributes	should not be shared : Vague
System : General	Made : Irrelevant
Application : General	Arrival : Irrelevant
User : Duplicate	Mode : Irrelevant
Database : General	Unique : Irrelevant
System : General	Travelling : General
login id : Attributes	Travelling : Duplicate
Password : Attributes	Verify : Duplicate
Manager name : Attributes	Improve : Irrelevant
Date : Attributes	

Tax : Attributes	
Payment : Attributes	
Booking id : Association	
agency name : Attributes	
Agency id : Association	
Email id : Attributes	
Arrive time : Attributes	
Depart time : Attributes	
Landmark : Attributes	
Location : Duplicate	
Hotel id : Association	
Rating : Duplicate	
Facility : Duplicate	
Tour package type : Attributes	
Tour package id : Attributes	

Accepted Nouns And Verbs :

Accepted Nouns	Accepted Verbs
Branch Manager	stored
facilities	access
customers	plan
marketing strategies	sorting
hotel	providing
hotels and touring agencies	available
tickets	works on trust

payment	confirmed
trains	security
flights	book
developer	compare
Cab	Address
cities	
feedback	

ER Model:

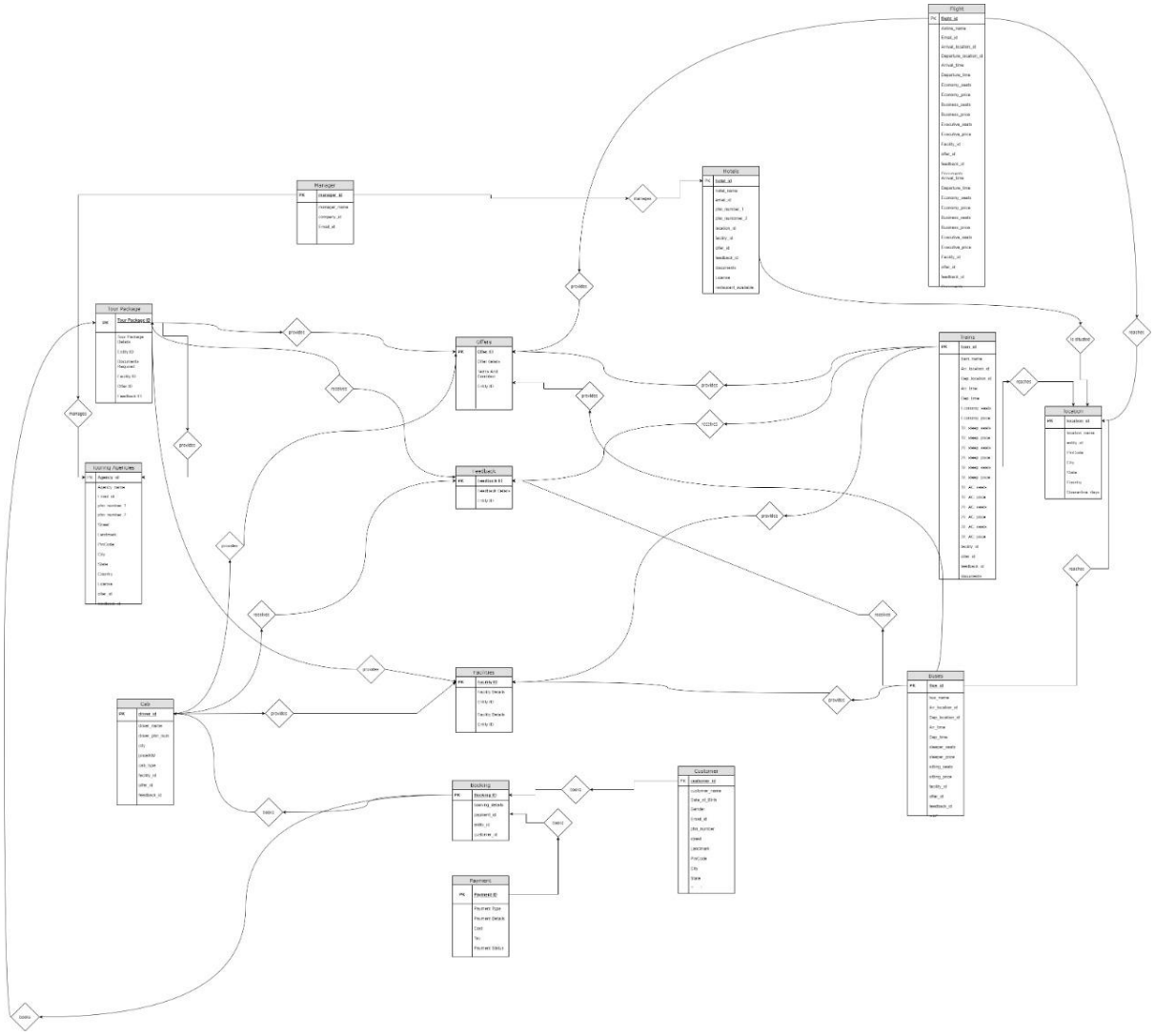
Candidate Entity Set	**Candidate Attribute**	**Candidate Relationship**
Customer	<u>Customer Id</u> Customer Name Date Of Birth Gender Email ID Phone Number Address Street Landmark Pin Code	Plan Compare Book
Manager	<u>Manager ID</u> Manager Name Email ID	Access providing
Touring Agencies	<u>Agency ID</u> Agency Name Email ID Phone Number 1 Phone Number 2 Address Street Landmark Pin Code License	Access providing

Hotels	Hotel ID Hotel Name Email ID Contact_Number Customer_Care_Number Location ID Facility ID Offer ID Feedback ID Documents License Restaurant Available	Confirmed Providing Book
Locations	Location ID Location Name Pin Code No. Of Quarantine Days	Address
Bookings	Booking ID Payment ID Booking Details Hotel ID Customer ID Transport ID Tour Package ID	Book Confirmed
Payment	Payment ID Payment Type Payment Details Cost Tax Payment Status	Security Confirmed
Facilities	Facility ID Facility Details	Providing Available
Feedback	Feedback ID Feedback Details Ratings	Store
Tour Package	Tour Package ID Tour Package Details Entity ID Documents Required	Providing Available

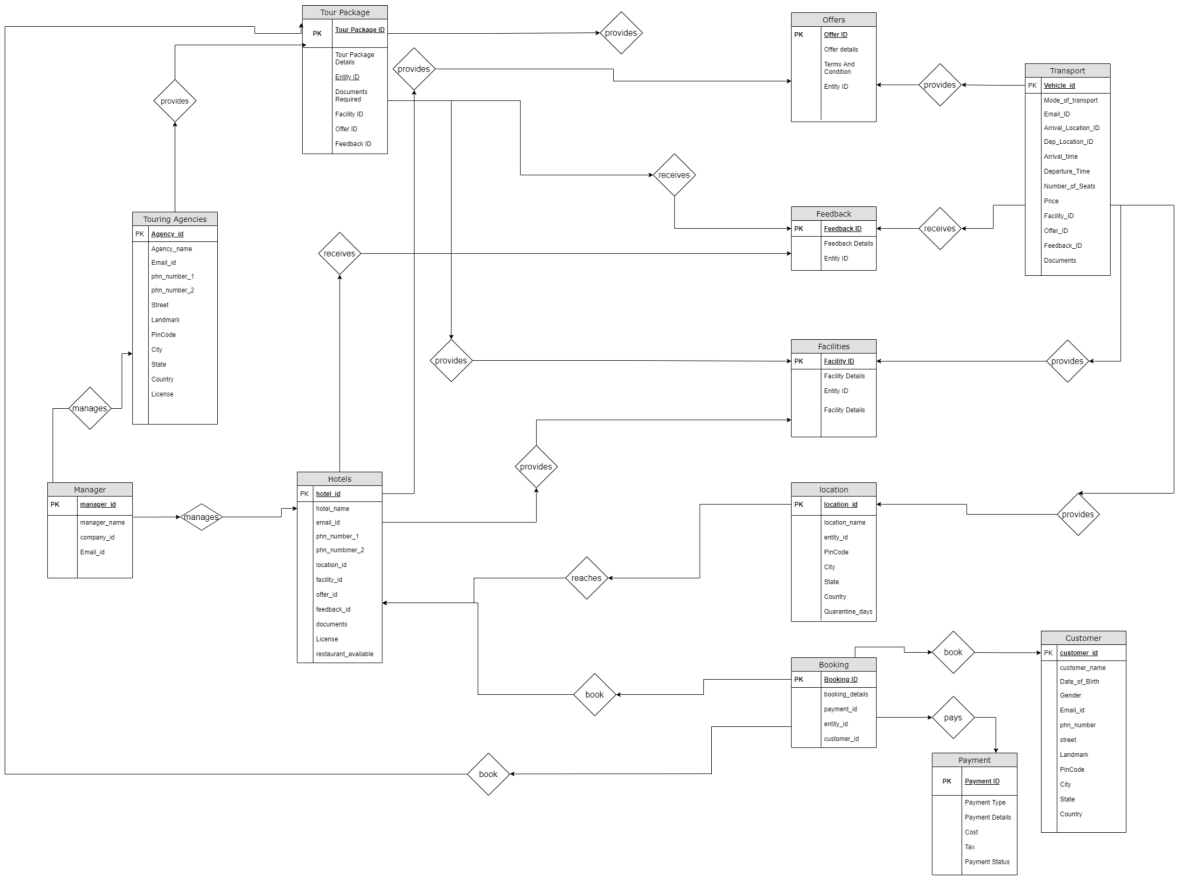
	Facility ID Offer ID Feedback ID	
Offers	Offer ID Offer Details Terms and Conditions	Providing Available
Transport	Vehicle_ID Mode_of_transportation Email_ID Arrival_location_ID Departure_location_id Arrival Time Departure Time Number_of_Seats Price Facility ID Offer ID Feedback ID Documents	Confirmed Providing Book

Section3 : ER-Diagrams all versions.

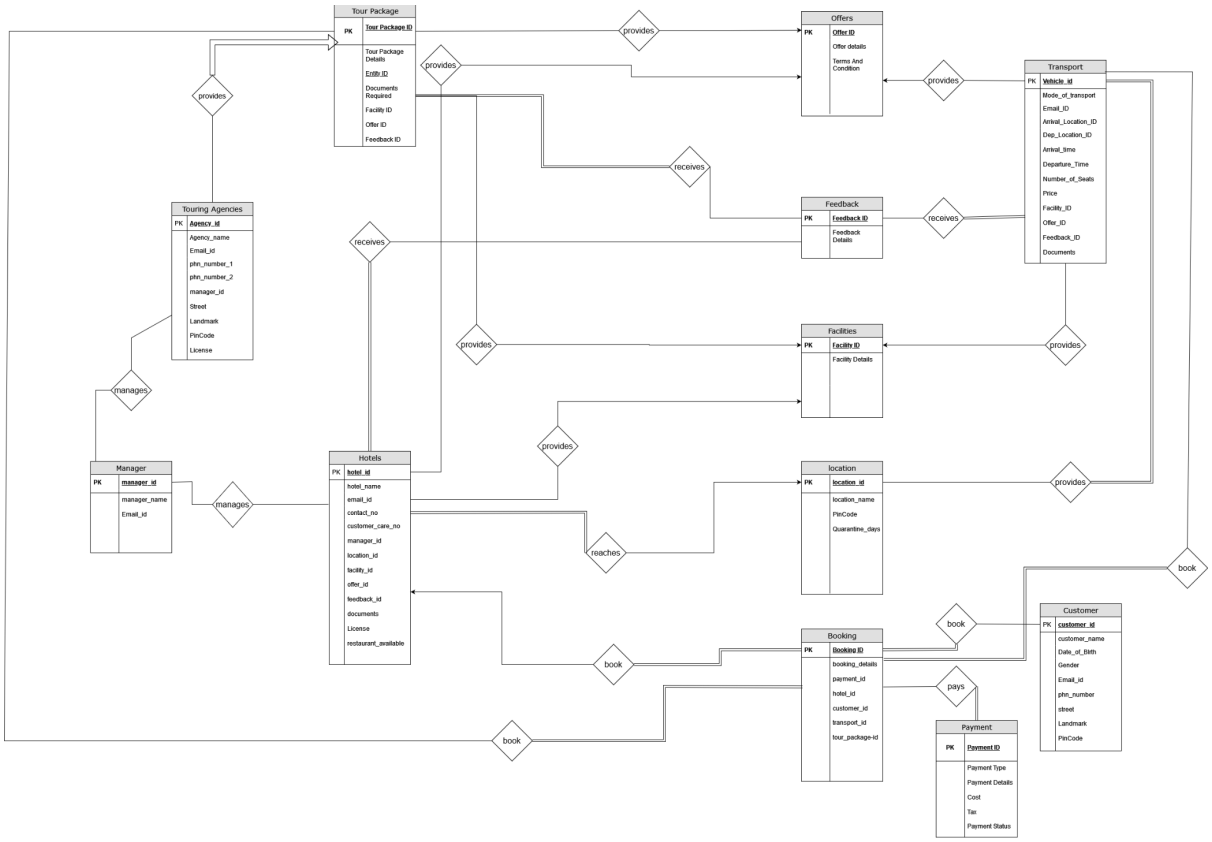
ER Diagram: Version 1



ER Diagram: Version 2



Final ER Diagram:



Link: https://app.diagrams.net/#G1WGSunJSdr04fRXFIS2NJ_LAEATAHpTLx

- Here in the above ER diagram, all the relationships represent the **Binary** relationship.

Section4: Conversion of Final ER-Diagram to Relational Model.

Relational Schema:

- Customer (**Customer_ID**,Customer_Name,Date_Of_Birth,Gender,Email ID,Phone Number,Street,Landmark,Pin Code)
- Manager (**Manager ID**,Manager Name,Email ID)
- TouringAgencies (**Agency ID**,Agency Name,Email ID,Phone Number 1,Phone Number 2,Manager ID,Street,Landmark,Pin Code,License)
- Hotel (**Hotel ID**,Hotel Name,Email ID,Contact_No,Customer_Care_No,Manager ID,Location ID,Facility ID,Offer ID,Feedback ID,Documents,License,Restaurant Available)
- Locations(**Location ID**,Location Name,Pin Code,No. Of Quarantine Days)
- Bookings(**Booking ID**,Payment ID,Booking Details,Hotel ID,Customer ID, Transport ID, Tour Package ID)
- Payment(**Payment ID**, Payment Type, Payment Details , Cost, Tax, Payment Status)
- Facilities(Facility ID,Facility Details)
- Feedback (**Feedback ID**,Feedback Details,Ratings)
- Tour Package(**Tour Package ID**,Tour Package Details,Entity ID,Documents Required,Facility ID,Offer ID,Feedback ID)
- Offers(**Offer ID**,Offer Details,Terms and Conditions)
- Transport(**Vehicle_ID**,Mode_of_transportation,Email_ID,Arrival_location_ID,Departure _location_id, Arrival Time, Departure Time, Number_of_Seats, Price, Facility ID, Offer ID, Feedback ID, Documents)

Section5 : Normalization and Schema Refinement

Functional Dependencies:

Entity Name	Dependencies
Customer	Customer_ID -> Customer_Name Customer_ID -> Date_Of_Birth Customer_ID -> Gender Customer_ID -> Email_ID Customer_ID -> Phone_Number Customer_ID -> (Street, Landmark, Pincode) Email_ID -> Customer_Name Email_ID -> Date_Of_Birth Email_ID -> Gender Email_ID -> Customer_ID Email_ID -> Phone_Number Email_ID -> (Street, Landmark, Pincode) Phone_Number-> Customer_Name Phone_Number-> Date_Of_Birth Phone_Number-> Gender Phone_Number-> Email_ID Phone_Number-> Customer_ID Phone_Number-> (Street, Landmark, Pincode)
Manager	Manager_ID -> Manager_name Manager_ID -> Email_ID Email_ID -> Manager_name Email_ID -> Manager_ID
Touring_Agencies	Agency_ID -> Agency_Name Agency_ID -> Email_ID Agency_ID -> Phone_number_1 Agency_ID -> Phone_number_2 Agency_ID -> Street Agency_ID -> Landmark

	Agency_ID -> Pin Code Agency_ID -> Country Agency_ID -> License Email_ID -> Agency_Name Email_ID -> Agency_ID Email_ID -> Phone_number_1 Email_ID -> Phone_number_2 Email_ID -> Street Email_ID -> Landmark Email_ID -> Pin Code Email_ID -> License License -> Agency_Name License -> Agency_ID License -> Phone_number_1 License -> Phone_number_2 License -> Street License -> Landmark License -> Pin Code License -> Email_ID Phone_number_1 -> Agency_Name Phone_number_1 -> Agency_ID Phone_number_1 -> License Phone_number_1 -> Phone_number_2 Phone_number_1 -> Street Phone_number_1 -> Landmark Phone_number_1 -> Pin Code Phone_number_1 -> Email_ID Phone_number_2 -> Agency_Name Phone_number_2 -> Agency_ID Phone_number_2 -> License Phone_number_2 -> Phone_number_1 Phone_number_2 -> Street Phone_number_2 -> Landmark Phone_number_2 -> Pin Code Phone_number_2 -> Email_ID
Hotel	Hotel_ID -> Hotel_Name Hotel_ID -> Email_ID Hotel_ID -> Contact_Number Hotel_ID -> Customer_Care_Number

	Hotel_ID -> Location_ID Hotel_ID -> Facility_ID Hotel_ID -> Offer_ID Hotel_ID -> Feedback_ID Hotel_ID -> Documents Hotel_ID -> License Hotel_ID -> Restaurant_Available Email_ID -> Hotel_Name Email_ID -> Hotel_ID Email_ID -> Contact_Number Email_ID -> Customer_Care_Number Email_ID -> Location_ID Email_ID -> Facility_ID Email_ID -> Offer_ID Email_ID -> Feedback_ID Email_ID -> Documents Email_ID -> License Email_ID -> Restaurant_Available Contact_Number -> Hotel_Name Contact_Number -> Email_ID Contact_Number -> Hotel_ID Contact_Number -> Customer_Contact_Number Contact_Number -> Location_ID Contact_Number -> Facility_ID Contact_Number -> Offer_ID Contact_Number -> Feedback_ID Contact_Number -> Documents Contact_Number -> License Contact_Number -> Restaurant_Available Customer_Care_Number -> Hotel_Name Customer_Care_Number -> Email_ID Customer_Care_Number -> Hotel_ID Customer_Care_Number -> Contact-Number Customer_Care_Number-> Location_ID Customer_Care_Number -> Facility_ID Customer_Care_Number -> Offer_ID Customer_Care_Number-> Feedback_ID Customer_Care_Number -> Documents Customer_Care_Number -> License

	Customer_Care_Number -> Restaurant_Available Location_ID -> Hotel_Name Location_ID -> Email_ID Location_ID -> Contact_Number Location_ID -> Customer_Care_Number Location_ID -> Hotel_ID Location_ID -> Facility_ID Location_ID -> Offer_ID Location_ID -> Feedback_ID Location_ID -> Documents Location_ID -> License Location_ID -> Restaurant_Available Feedback ID -> Hotel_Name Feedback ID-> Email_ID Feedback ID -> Contact_Number Feedback ID -> Customer_Care_Number Feedback ID -> Location_ID Feedback ID -> Facility_ID Feedback ID -> Offer_ID Feedback ID -> Hotel_ID Feedback ID -> Documents Feedback ID -> License Feedback ID -> Restaurant_Available Offer_ID -> Hotel_Name Offer_ID -> Email_ID Offer_ID -> Contact_Number Offer_ID -> Customer_Care_Number Offer_ID -> Location_ID Offer_ID -> Facility_ID Offer_ID -> Feedback_ID Offer_ID -> Hotel_ID Offer_ID -> Documents Offer_ID -> License Offer_ID -> Restaurant_Available Facility_ID-> Hotel_Name Facility_ID-> Email_ID Facility_ID-> Contact_Number Facility_ID-> Customer_Care_Number Facility_ID-> Location_ID

	Facility_ID-> Offer_ID Facility_ID-> Feedback_ID Facility_ID-> Hotel_ID Facility_ID-> Documents Facility_ID-> License Facility_ID-> Restaurant_Available License-> Hotel_Name License -> Email_ID License -> Contact_Number License -> Customer_Care_Number License -> Location_ID License -> Facility_ID License -> Offer_ID License -> Feedback_ID License -> Documents License -> Hotel_ID License -> Restaurant_Available
Location	Location_ID -> Location_name Location_ID -> Pin Code Location_ID -> No. Of Quarantine Days
Bookings	Booking_ID -> Payment ID Booking_ID -> Booking Details Booking_ID -> Customer ID Booking_ID -> Hotel_ID Booking_ID -> Tour_Package_ID Booking_ID -> Transport_ID Payment_ID -> Booking ID Payment_ID -> Booking Details Payment_ID -> Customer ID Payment_ID -> Hotel_ID Payment_ID -> Tour_Package_ID Payment_ID -> Transport_ID
Payment	Payment ID -> Payment Type Payment ID -> Payment Details Payment ID -> Cost Payment ID -> Tax Payment ID -> Payment Status
Facilities	Facility ID -> Facility Details

Feedback	Feedback ID -> Feedback Details Feedback ID -> Ratings
Tour Package	Tour_Package_ID -> Tour_Package_Details Tour_Package_ID -> Entity ID Tour_Package_ID -> Documents Required Tour_Package_ID -> Facility ID Tour_Package_ID -> Offer ID Tour_Package_ID -> Feedback ID Feedback_ID-> Tour_Package_Details Feedback_ID -> Tour_Package_ID Feedback_ID -> Documents Required Feedback_ID -> Facility_ID Feedback_ID -> Offer_ID Offer_ID-> Tour_Package_Details Offer_ID-> Tour_Package_ID Offer_ID-> Documents Required Offer_ID-> Facility_ID Offer_ID-> Feedback_ID Facility_ID-> Tour_Package_Details Facility_ID-> Tour_Package_ID Facility_ID-> Documents Required Facility_ID-> Feedback_ID Facility_ID-> Offer_ID
Offers	Offer_ID -> Offer Details Offer_ID -> Terms and Conditions
Transport	Vehicle_ID -> Mode_of_transportation Vehicle_ID -> Email_ID Vehicle_ID -> Arrival_location_ID Vehicle_ID -> Departure_location_id Vehicle_ID -> Arrival Time Vehicle_ID -> Departure Time Vehicle_ID -> Number_of_Seats Vehicle_ID -> Price Vehicle_ID -> Facility ID Vehicle_ID -> Offer ID Vehicle_ID -> Feedback ID Vehicle_ID -> Documents

	Email_ID -> Mode_of_transportation Email_ID-> Vehicle_ID Email_ID-> Arrival_location_ID Email_ID-> Departure_location_id Email_ID-> Arrival Time Email_ID-> Departure Time Email_ID-> Number_of_Seats Email_ID-> Price Email_ID-> Facility ID Email_ID-> Offer ID Email_ID-> Feedback ID Email_ID-> Documents Feedback_ID-> Mode_of_transportation Feedback_ID-> Vehicle_ID Feedback_ID-> Arrival_location_ID Feedback_ID-> Departure_location_id Feedback_ID-> Arrival Time Feedback_ID-> Departure Time Feedback_ID-> Number_of_Seats Feedback_ID-> Price Feedback_ID-> Facility_ID Feedback_ID-> Offer_ID Feedback_ID-> Email_ID Feedback_ID-> Documents Offer_ID-> Mode_of_transportation Offer_ID-> Vehicle_ID Offer_ID-> Arrival_location_ID Offer_ID-> Departure_location_id Offer_ID-> Arrival Time Offer_ID-> Departure Time Offer_ID-> Number_of_Seats Offer_ID-> Price Offer_ID-> Facility_ID Offer_ID-> Email_ID Offer_ID-> Feedback_ID Offer_ID-> Documents Facility_ID-> Mode_of_transportation Facility_ID-> Vehicle_ID Facility_ID-> Arrival_location_ID Facility_ID-> Departure_location_id Facility_ID-> Arrival Time

	Facility_ID-> Departure Time Facility_ID-> Number_of_Seats Facility_ID-> Price Facility_ID-> Offer_ID Facility_ID-> Email_ID Facility_ID-> Feedback_ID Facility_ID-> Documents

1. First Normal Form(1NF):

Every Table satisfies the First Normal Form as we do not have any **multivalued attributes** within the table.

2. Second Normal Form (2NF):

Second Normal Form (2NF) is based on the concept of functional dependency.It applies to a relation in which composite keys are present i.e. relations with a primary key composed of two or more attributes.In above mentioned schema each and every relation has **single attribute primary key** and hence every relation is in 2NF.Every Relation has no partial dependency i.e. no non - prime attributes are dependent on any proper subset of any candidate key of the table.

3. Redundancies:

In our database based on travel management, we do have some redundancies which are removed by removing the transitive dependencies present among different relations in the database. For example, the customer table contains attributes namely city, state and country. For different customers belonging to the same city, they have the same values of city, state and country so they add unwanted **redundancies in the database by using more space and memory**. So they can be removed by either removing them as attributes from the tables where they are included as attributes or making more tables having one of them as primary key which inturn satisfies third normal form.

4. 3rd Normal Form (3NF):

The **transitive dependencies** present in the tables need to be removed to be third normal form optimized. In the customer, touring agency and location table, the state attribute is dependent on the city, country is dependent on the state and so it implies that the country is dependent on the city. However a city, state and country can be found using pin code specified as an attribute. So here transitive dependency is present. We removed the city, state and country attributes from both the tables and so they became the third normal form optimized. Else all tables are already 3NF optimized.

Section6: SQL: Final DDL Scripts, Insert statements, 40 SQL Queries with Snapshots of output of each query.

Customer:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.customer
(
    customer_id character varying(50) NOT NULL,
    customer_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    date_of_birth date NOT NULL,
    gender "char" NOT NULL,
    email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    phone_number bigint NOT NULL,
    street character varying(100) COLLATE pg_catalog."default",
    landmark character varying(50) COLLATE pg_catalog."default",
    pincode bigint NOT NULL,
    CONSTRAINT customer_pkey PRIMARY KEY (customer_id)
)
```

Manager:

```
CREATE TABLE IF NOT EXISTS travel_agency_management."manager"
(
    manager_id bigint NOT NULL,
    manager_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "manager_pkey" PRIMARY KEY (manager_id)
)
```

Touring Agency:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.touring_agencies
(
    agency_id character varying(50) COLLATE pg_catalog."default" NOT NULL,
    agency_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
```

```
    email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    manager_id bigint NOT NULL,
    phone_number1 bigint NOT NULL,
    phone_number2 bigint,
    street character varying(100) COLLATE pg_catalog."default",
    landmark character varying(50) COLLATE pg_catalog."default",
    pincode bigint NOT NULL,
    license character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Touring_Agencies_pkey" PRIMARY KEY (agency_id),
    CONSTRAINT manager_fkey FOREIGN KEY (manager_id)
  REFERENCES travel_agency_management.manager (manager_id) MATCH SIMPLE
  ON UPDATE CASCADE
  ON DELETE CASCADE
)
```

Hotel:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.hotel
(
    hotel_id character varying(50) COLLATE pg_catalog."default" NOT NULL,
    hotel_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    contact_number bigint NOT NULL,
    customer_care_number bigint,
    manager_id bigint NOT NULL,
    location_id bigint NOT NULL,
    facility_id bigint NOT NULL,
    offer_id bigint NOT NULL,
    feedback_id bigint NOT NULL,
    documents character varying(150) COLLATE pg_catalog."default",
    license character varying(50) COLLATE pg_catalog."default" NOT NULL,
    restaurant_available boolean NOT NULL,
    CONSTRAINT hotel_pkey PRIMARY KEY (hotel_id),
    CONSTRAINT facility_fkey FOREIGN KEY (facility_id)
```

```
        REFERENCES travel_agency_management.facility (facility_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT feedback_fkey FOREIGN KEY (feedback_id)
        REFERENCES travel_agency_management.feedback (feedback_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT location_fkey FOREIGN KEY (location_id)
        REFERENCES travel_agency_management.location (location_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT manager_fkey FOREIGN KEY (manager_id)
        REFERENCES travel_agency_management.manager (manager_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID,
        CONSTRAINT offer_fkey FOREIGN KEY (offer_id)
        REFERENCES travel_agency_management.offers (offer_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```

Bookings:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.booking
(
    booking_id bigint NOT NULL,
    payment_id bigint NOT NULL,
    booking_details character varying(200) COLLATE pg_catalog."default" NOT NULL,
    customer_id  character varying(50)NOT NULL,
        hotel_id  character varying(50)NOT NULL,
        tour_package_id  character varying(50)NOT NULL,
        transport_id  character varying(50)NOT NULL,
    CONSTRAINT booking_pkey PRIMARY KEY (booking_id),
    CONSTRAINT customer_fkey FOREIGN KEY (customer_id)
      REFERENCES travel_agency_management.customer (customer_id) MATCH SIMPLE
      ON UPDATE CASCADE
      ON DELETE CASCADE
      NOT VALID,
```

```
    CONSTRAINT hotel_fkey FOREIGN KEY (hotel_id)
        REFERENCES travel_agency_management.hotel (hotel_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID,
    CONSTRAINT package_fkey FOREIGN KEY (tour_package_id)
        REFERENCES travel_agency_management.tour_package (tour_package_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID,
    CONSTRAINT payment_fkey FOREIGN KEY (payment_id)
        REFERENCES travel_agency_management.payment (payment_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID,
    CONSTRAINT transport_fkey FOREIGN KEY (transport_id)
        REFERENCES travel_agency_management.transport (vehicle_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
)
```

Location:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.location
(
        location_id bigint NOT NULL,
        location_name character varying COLLATE pg_catalog."default" NOT NULL,
        pincode bigint NOT NULL,
        number_of_quarantine_days bigint NOT NULL,
        CONSTRAINT location_pkey PRIMARY KEY (location_id)
)
```

Payment:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.payment
(
    payment_id bigint NOT NULL,
```

```
    payment_type character varying(50) COLLATE pg_catalog."default" NOT NULL,
    payment_details date NOT NULL,
    cost double precision NOT NULL,
    tax double precision NOT NULL,
    payment_status boolean NOT NULL,
    CONSTRAINT payment_pkey PRIMARY KEY (payment_id)
)
```

Facility:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.facility
(
    facility_id bigint NOT NULL,
    facility_details character varying(200) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT facility_pkey PRIMARY KEY (facility_id)
)
```

Feedback:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.feedback
(
    feedback_id bigint NOT NULL,
    feedback_details character varying(300) COLLATE pg_catalog."default" NOT NULL,
    rating integer NOT NULL,
    CONSTRAINT feedback_pkey PRIMARY KEY (feedback_id)
)
```

Tour Package:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.tour_package
(
        tour_package_id character varying(50) NOT NULL,
```

```
    tour_package_details character varying(1000) COLLATE pg_catalog."default" NOT
NULL,
    entity_id character varying(50) NOT NULL,
    documents_required character varying(200) COLLATE pg_catalog."default" NOT
NULL,
    facility_id bigint NOT NULL,
    offer_id bigint NOT NULL,
    feedback_id bigint NOT NULL,
    CONSTRAINT tour_package_pkey PRIMARY KEY (tour_package_id),
    CONSTRAINT facility_fkey FOREIGN KEY (facility_id)
    REFERENCES travel_agency_management.facility (facility_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT feedback_fkey FOREIGN KEY (feedback_id)
    REFERENCES travel_agency_management.feedback (feedback_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT offer_fkey FOREIGN KEY (offer_id)
    REFERENCES travel_agency_management.offers (offer_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    CONSTRAINT touring_fkey FOREIGN KEY (entity_id)
    REFERENCES travel_agency_management."touring_agencies" (agency_id) MATCH
SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
)
```

Offers:

```
CREATE TABLE IF NOT EXISTS travel_agency_management.offers
(
  offer_id bigint NOT NULL,
  offer_details character varying(300) COLLATE pg_catalog."default" NOT NULL,
  terms_and_conditions character varying(300) COLLATE pg_catalog."default" NOT NULL,
  CONSTRAINT offers_pkey PRIMARY KEY (offer_id)
)
```

Transport:

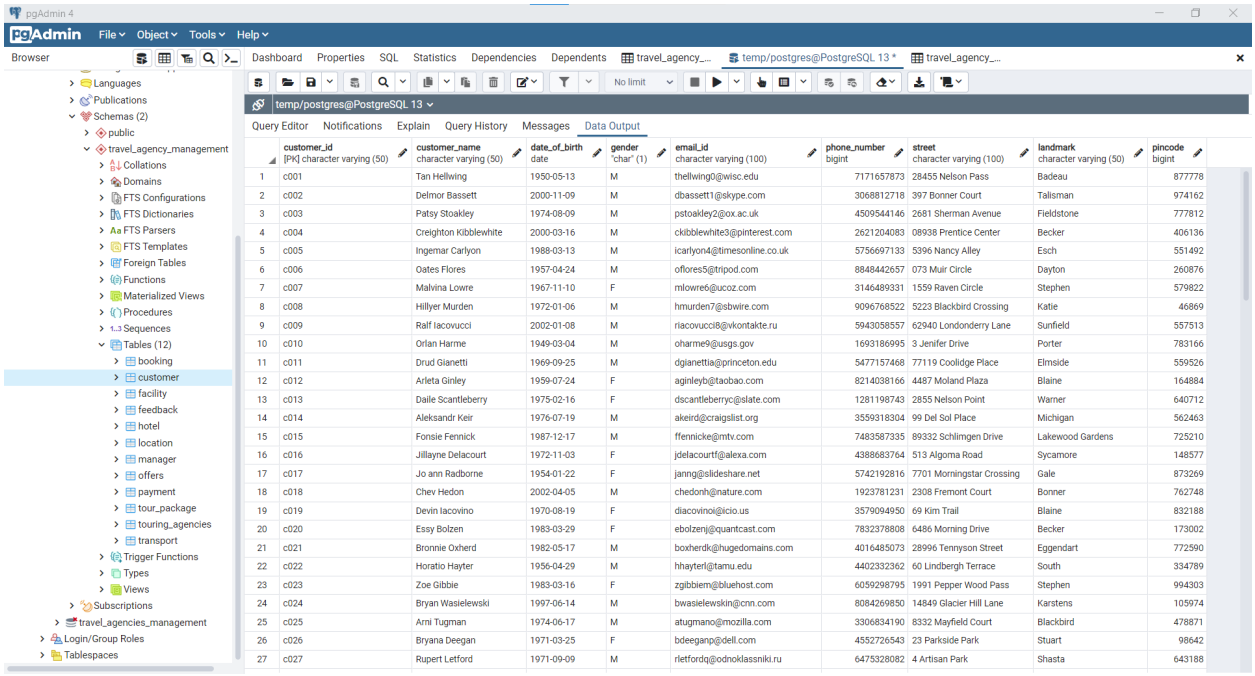
```
CREATE TABLE IF NOT EXISTS travel_agency_management.transport
(
        vehicle_id  character varying(50)NOT NULL,
        mode_of_transportation character varying(50) COLLATE pg_catalog."default" NOT
NULL,
        email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
        arrival_location_id bigint NOT NULL,
        departure_location_id bigint NOT NULL,
        arrival_time time without time zone NOT NULL,
        departure_time time without time zone NOT NULL,
        no_of_seat integer NOT NULL,
        price double precision NOT NULL,
        facility_id bigint NOT NULL,
        offer_id bigint NOT NULL,
        feedback_id bigint NOT NULL,
        document character varying(200) COLLATE pg_catalog."default" NOT NULL,
        CONSTRAINT transport_pkey PRIMARY KEY (vehicle_id),
        CONSTRAINT arr_location_fkey FOREIGN KEY (arrival_location_id)
        REFERENCES travel_agency_management.location (location_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT dep_location_fkey FOREIGN KEY (departure_location_id)
        REFERENCES travel_agency_management.location (location_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT facility_fkey FOREIGN KEY (facility_id)
        REFERENCES travel_agency_management.facility (facility_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
        CONSTRAINT feedback_fkey FOREIGN KEY (feedback_id)
        REFERENCES travel_agency_management.feedback (feedback_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
```

```
    CONSTRAINT offer_fkey FOREIGN KEY (offer_id)
    REFERENCES travel_agency_management.offers (offer_id) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
)
```

SQL Queries:

1. Show the details of all the customers.

```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM customer;
```

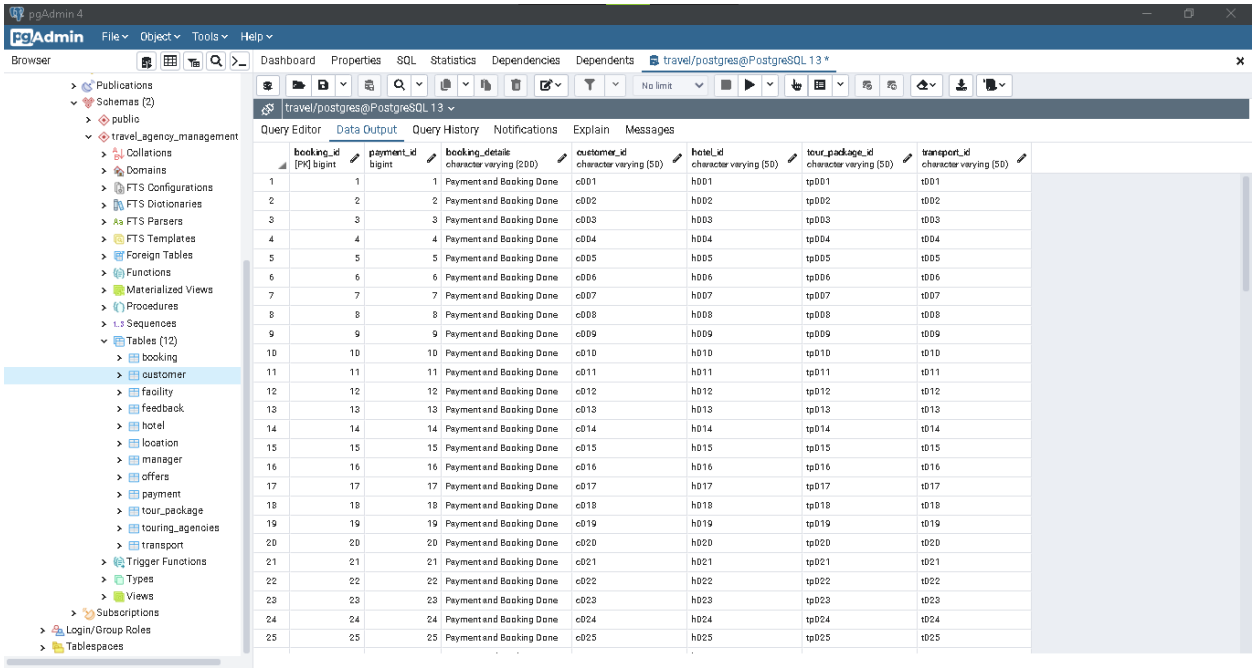


	customer_id [PK] character varying (50)	customer_name character varying (50)	date_of_birth date	gender "char" (1)	email_id character varying (100)	phone_number bigint	street character varying (100)	landmark character varying (50)	pincode bigint
1	c001	Tan Hellwing	1950-05-13	M	thellwing0@wisc.edu	7171657873	28455 Nelson Pass	Badeau	877778
2	c002	Delmor Bassett	2000-11-09	M	dbassett1@skype.com	3068812718	397 Bonner Court	Talisman	974162
3	c003	Patsy Stoakley	1974-08-09	M	pstoakley2@ox.ac.uk	4509544146	2681 Sherman Avenue	Fieldstone	777812
4	c004	Creighton Kibblewhite	2000-03-16	M	ckibblewhite3@pinterest.com	2621204083	08938 Prentice Center	Becker	406136
5	c005	Ingemar Carlyon	1988-03-13	M	icarlyon4@timesonline.co.uk	5756697133	5396 Nancy Alley	Esch	551492
6	c006	Oates Flores	1957-04-24	M	oflores5@tripod.com	8848442657	073 Muir Circle	Dayton	260876
7	c007	Malvina Lowre	1967-11-10	F	mlowre6@ucoz.com	3146489331	1559 Raven Circle	Stephen	579822
8	c008	Hillyer Murden	1972-01-06	M	hmurden7@sbwire.com	9096768522	5223 Blackbird Crossing	Katie	46869
9	c009	Ralf Iacovucci	2002-01-08	M	riacovucci8@vkontakte.ru	5943058557	62940 Londonderry Lane	Sunfield	557513
10	c010	Orlan Harme	1949-03-04	M	oharme9@usgs.gov	1693186995	3 Jenifer Drive	Porter	783166
11	c011	Drud Gianetti	1969-09-25	M	dgianettia@princeton.edu	5477157468	77119 Coolidge Place	Elmside	559526
12	c012	Arleta Ginley	1959-07-24	F	aginleyb@taobao.com	8214038166	4487 Moland Plaza	Blaine	164884
13	c013	Daile Scantleberry	1975-02-16	F	dscantleberryc@slate.com	1281198743	2855 Nelson Point	Warner	640712
14	c014	Aleksandr Keir	1976-07-19	M	akeird@craigslist.org	3559318304	99 Del Sol Place	Michigan	562463
15	c015	Fonsie Fennick	1987-12-17	M	ffennicke@mtv.com	7483587335	89332 Schlimgen Drive	Lakewood Gardens	725210
16	c016	Jillayne Delacourt	1972-11-03	F	jdelacourtf@alexa.com	4388683764	513 Algoma Road	Sycamore	148577
17	c017	Jo ann Radborne	1954-01-22	F	janng@slideshare.net	5742192816	7701 Morningstar Crossing	Gale	873269
18	c018	Chev Hedon	2002-04-05	M	chedonh@nature.com	1923781231	2308 Fremont Court	Bonner	762748
19	c019	Devin Iacovino	1970-08-19	F	diacovinoi@icio.us	3579094950	69 Kim Trail	Blaine	832188
20	c020	Essy Bolzen	1983-03-29	F	ebolzenj@quantcast.com	7832378808	6486 Morning Drive	Becker	173002
21	c021	Bronnie Oxherd	1982-05-17	M	boxherdk@hugedomains.com	4016485073	28996 Tennyson Street	Eggendart	772590
22	c022	Horatio Hayter	1956-04-29	M	hhayterl@tamu.edu	4402332362	60 Lindbergh Terrace	South	334789
23	c023	Zoe Gibbie	1983-03-16	F	zgibbiem@bluehost.com	6059298795	1991 Pepper Wood Pass	Stephen	994303
24	c024	Bryan Wasielewski	1997-06-14	M	bwasielewskin@cnn.com	8084269850	14849 Glacier Hill Lane	Karstens	105974
25	c025	Arni Tugman	1974-06-17	M	atugmano@mozilla.com	3306834190	8332 Mayfield Court	Blackbird	478871
26	c026	Bryana Deegan	1971-03-25	F	bdeeganp@dell.com	4552726543	23 Parkside Park	Stuart	98642
27	c027	Rupert Letford	1971-09-09	M	rletfordq@odnoklassniki.ru	6475328082	4 Artisan Park	Shasta	643188

```
No of Tuples = 100
```

2. Show the booking details of all the customers who have booked for any hotel, tour package, or transportation.

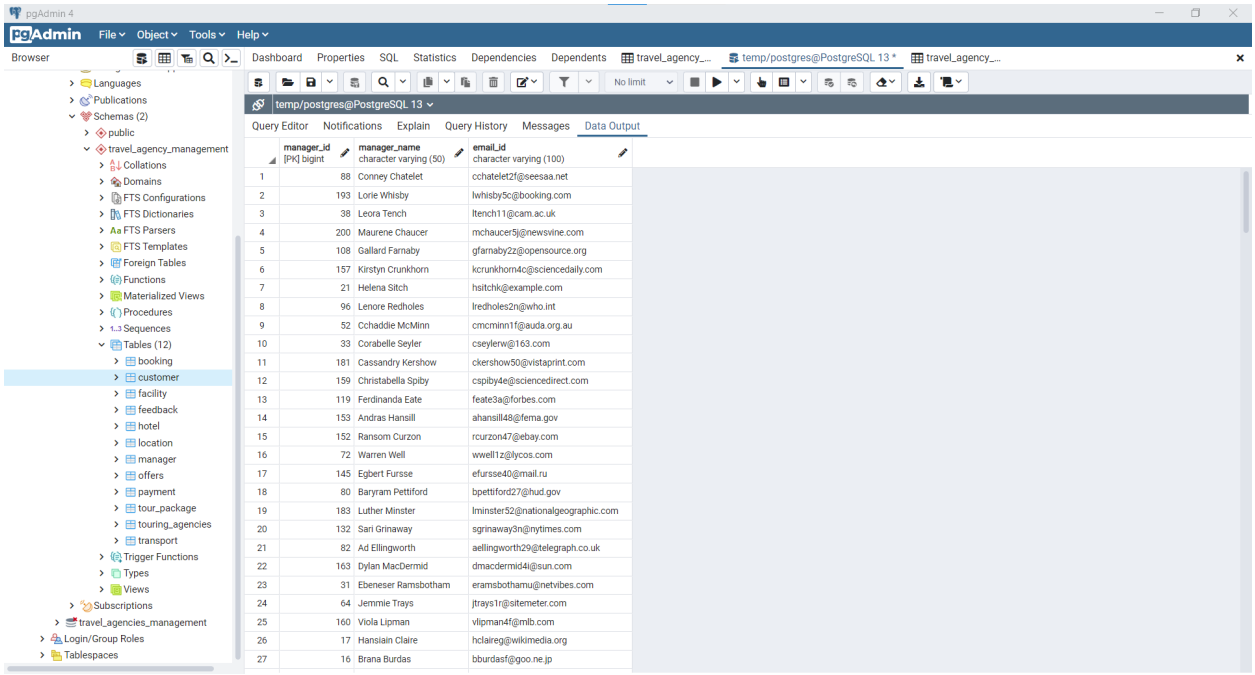
```
set search_path to travel_agency_management;
select * from booking;
```




```
No. of Tuples = 100
```

3. Show the details of all the distinct managers.

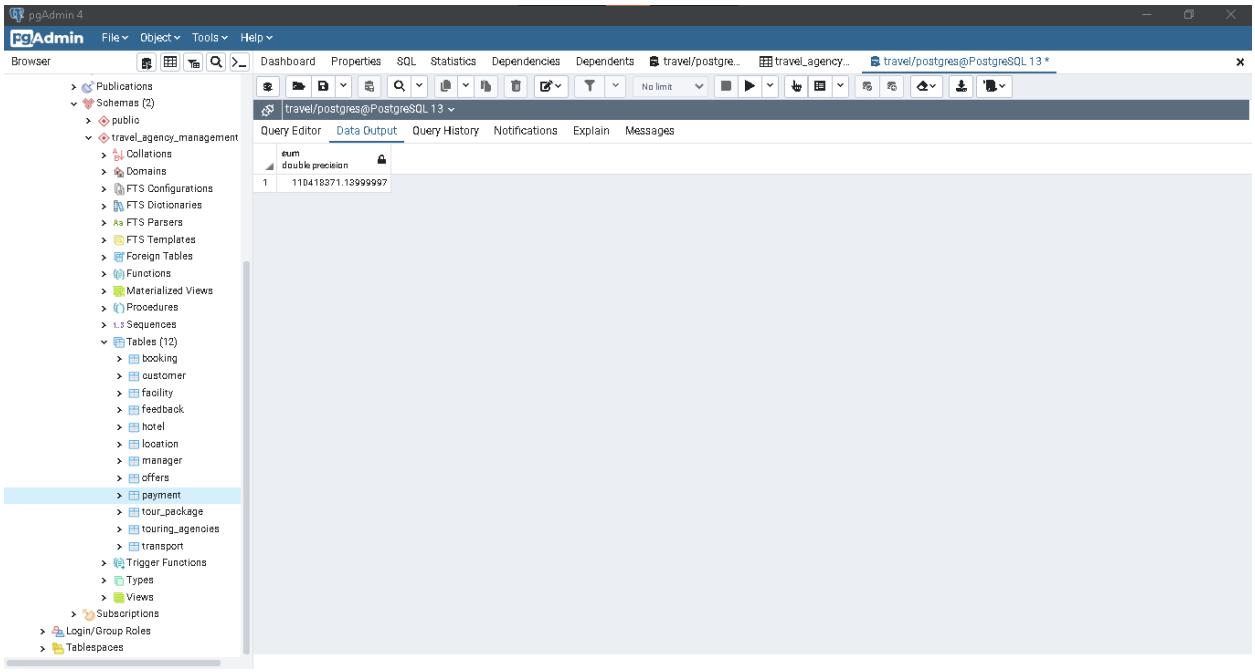
```
SET SEARCH_PATH TO travel_agency_management;
SELECT DISTINCT * FROM manager;
```




```
No of Tuples = 200.
```

4. Show the sum of all the payments made till now.

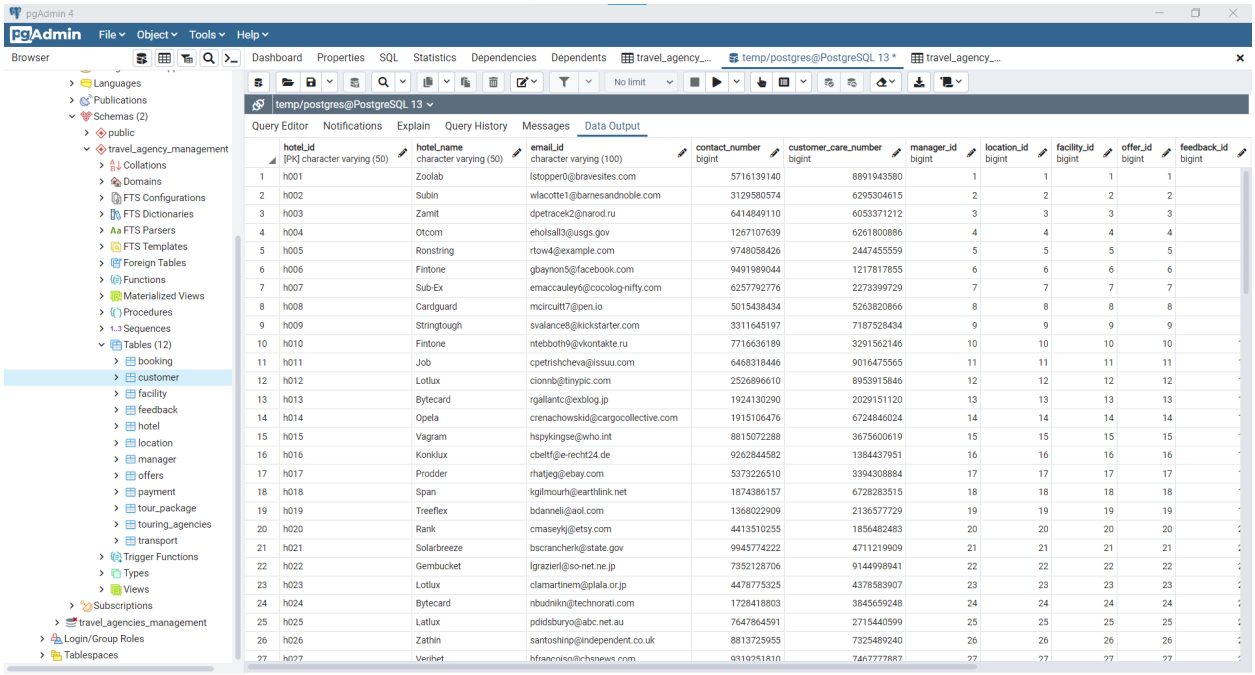
```
Set search_path to travel_agency_management;
select sum(cost) from payment;
```




```
No of Tuples = 1
```

5. Show the details of all the hotels registered in our database

```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM hotel;
```

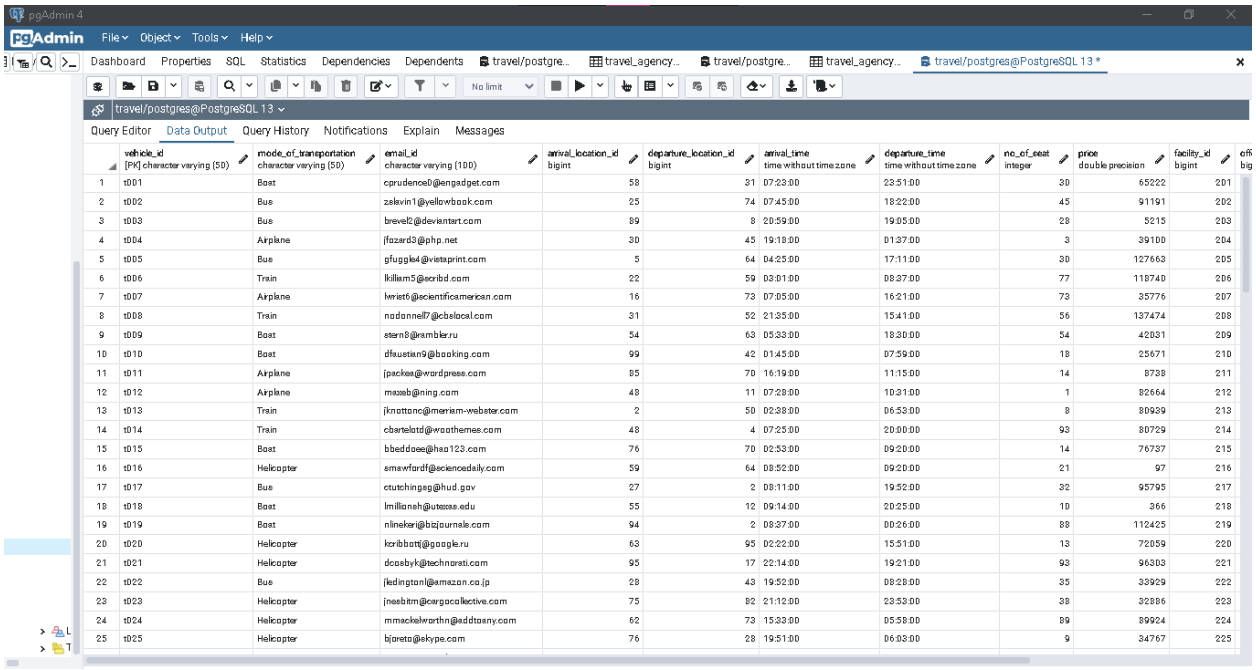


	hotel_id [PK] character varying (50)	hotel_name character varying (50)	email_id character varying (100)	contact_number bigint	customer_care_number bigint	manager_id bigint	location_id bigint	facility_id bigint	offer_id bigint	feedback_id bigint
1	h001	Zoolab	lstopper0@bravesites.com	5716139140	8891943580	1	1	1	1	
2	h002	Subin	wlacotte1@barnesandnoble.com	3129580574	6295304615	2	2	2	2	
3	h003	Zamit	dpetracek2@narod.ru	6414849110	6053371212	3	3	3	3	
4	h004	Otcom	eholsall3@usgs.gov	1267107639	6261800886	4	4	4	4	
5	h005	Ronstring	rtow4@example.com	9748058426	2447455559	5	5	5	5	
6	h006	Fintone	gbaynon5@facebook.com	9491989044	1217817855	6	6	6	6	
7	h007	Sub-Ex	emaccauley6@cocolog-nifty.com	6257792776	2273399729	7	7	7	7	
8	h008	Cardguard	mcircuitt7@pen.io	5015438434	5263820866	8	8	8	8	
9	h009	Stringtough	svalance8@kickstarter.com	3311645197	7187528434	9	9	9	9	
10	h010	Fintone	ntebboth9@vkontakte.ru	7716636189	3291562146	10	10	10	10	
11	h011	Job	cpetrishcheva@issuu.com	6468318446	9016475565	11	11	11	11	
12	h012	Lotlux	cionnb@tinypic.com	2526896610	8953915846	12	12	12	12	
13	h013	Bytecard	rgallantc@exblog.jp	1924130290	2029151120	13	13	13	13	
14	h014	Opela	crenachowskid@cargocollective.com	1915106476	6724846024	14	14	14	14	
15	h015	Vagram	hspykingse@who.int	8815072288	3675600619	15	15	15	15	
16	h016	Konklux	cbeltf@e-recht24.de	9262844582	1384437951	16	16	16	16	
17	h017	Prodder	rhatjeg@ebay.com	5373226510	3394308884	17	17	17	17	
18	h018	Span	kgilmourh@earthlink.net	1874386157	6728283515	18	18	18	18	
19	h019	Treeflex	bdanneli@aol.com	1368022909	2136577729	19	19	19	19	
20	h020	Rank	cmaseykj@etsy.com	4413510255	1856482483	20	20	20	20	
21	h021	Solarbreeze	bscrancherk@state.gov	9945774222	4711219909	21	21	21	21	
22	h022	Gembucket	lgrazierl@so-net.ne.jp	7352128706	9144998941	22	22	22	22	
23	h023	Lotlux	clamartinem@plala.or.jp	4478775325	4378583907	23	23	23	23	
24	h024	Bytecard	nbudnikn@technorati.com	1728418803	3845659248	24	24	24	24	
25	h025	Latlux	pdidsburyo@abc.net.au	7647864591	2715440599	25	25	25	25	
26	h026	Zathin	santoshinp@independent.co.uk	8813725955	7325489240	26	26	26	26	
27	h027	Veribet	bfrancoisq@cbsnews.com	9319251810	7467777887	27	27	27	27	

```
No of Tuples = 100.
```

6. Show the details of all the vehicles.

```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM transport;
```

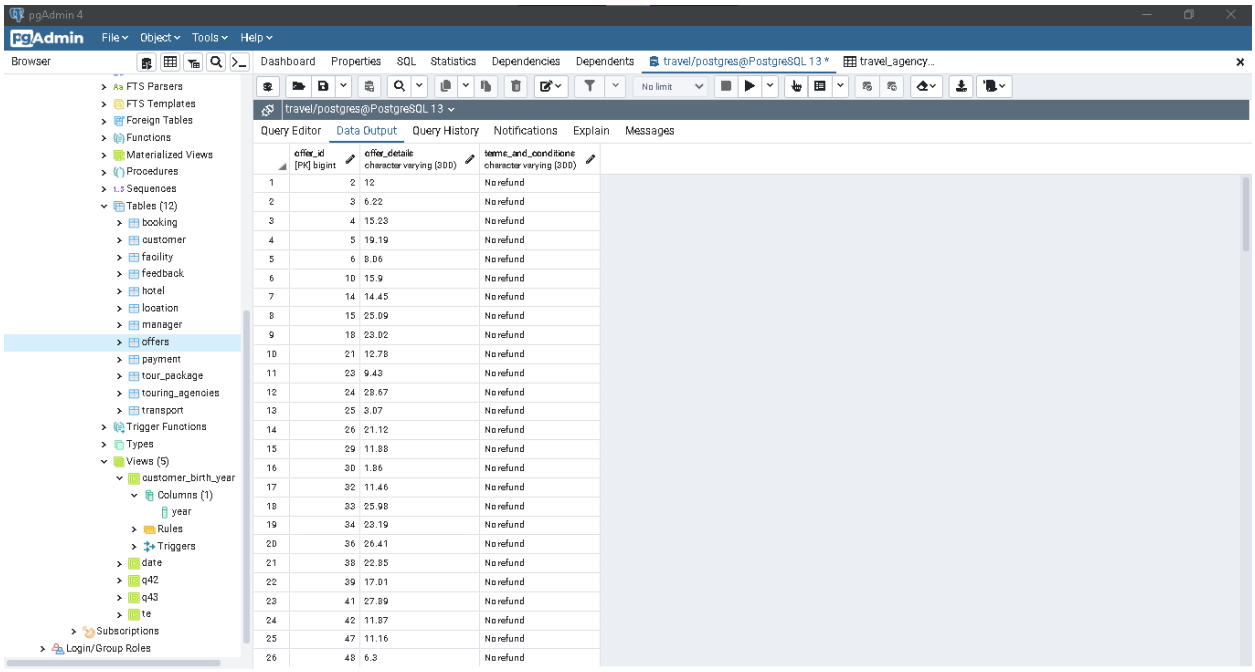


vehicle_id [PK] character varying (50)	mode_of_transportation character varying (50)	email_id character varying (100)	arrival_location_id bigint	departure_location_id bigint	arrival_time time without time zone	departure_time time without time zone	no_of_seat integer	price double precision	facility_id bigint
t001	Boat	cprudence0@engadget.com	58	31	07:23:00	23:51:00	30	65222	201
t002	Bus	zskvin1@yellowbook.com	25	74	07:45:00	18:22:00	45	91191	202
t003	Bus	brevel2@deviantart.com	89	8	20:59:00	19:05:00	28	5215	203
t004	Airplane	jfozard3@php.net	30	45	19:18:00	01:37:00	3	39100	204
t005	Bus	gfuggle4@vistaprint.com	5	64	04:25:00	17:11:00	30	127663	205
t006	Train	lkilliam5@scribd.com	22	59	03:01:00	08:37:00	77	118740	206
t007	Airplane	kwist6@scientificamerican.com	16	73	07:05:00	16:21:00	73	35776	207
t008	Train	nadonnell7@cbslocal.com	31	52	21:35:00	15:41:00	56	137474	208
t009	Boat	stern8@rambler.ru	54	63	05:33:00	18:30:00	54	42031	209
t010	Boat	dfaustian9@booking.com	99	42	01:45:00	07:59:00	18	25671	210
t011	Airplane	jpackes@wordpress.com	85	70	16:19:00	11:15:00	14	8738	211
t012	Airplane	mxseb@ning.com	48	11	07:28:00	10:31:00	1	82664	212
t013	Train	jknottsnc@merriam-webster.com	2	50	02:38:00	06:53:00	8	80939	213
t014	Train	cbartelotd@woothemes.com	48	4	07:25:00	20:00:00	93	80729	214
t015	Boat	bbeddoee@hao123.com	76	70	02:53:00	09:20:00	14	76737	215
t016	Helicopter	smawfordf@sciencedaily.com	59	64	08:52:00	09:20:00	21	97	216
t017	Bus	ctutchingsg@hud.gov	27	2	08:11:00	19:52:00	32	95795	217
t018	Boat	lmillionehh@utexas.edu	55	12	09:14:00	20:25:00	10	366	218
t019	Boat	nlinekeri@bizjournals.com	94	2	08:37:00	00:26:00	88	112425	219
t020	Helicopter	kcribbott@google.ru	63	95	02:22:00	15:51:00	13	72059	220
t021	Helicopter	dcosbyk@technorati.com	95	17	22:14:00	19:21:00	93	96303	221
t022	Bus	jledingtonl@amazon.co.jp	28	43	19:52:00	08:28:00	35	33929	222
t023	Helicopter	jnesbitm@cargocollective.com	75	82	21:12:00	23:53:00	38	32886	223
t024	Helicopter	mmackelworthn@addtoany.com	62	73	15:33:00	05:58:00	89	89924	224
t025	Helicopter	bjoretz@skype.com	76	28	19:51:00	06:03:00	9	34767	225

```
No of Tuples = 100
```

7. Show the details of all the offers where the term and conditions are ‘No Refund.’

```
set search_path to travel_agency_management;
select * from offers
where terms_and_conditions = 'No refund'
```

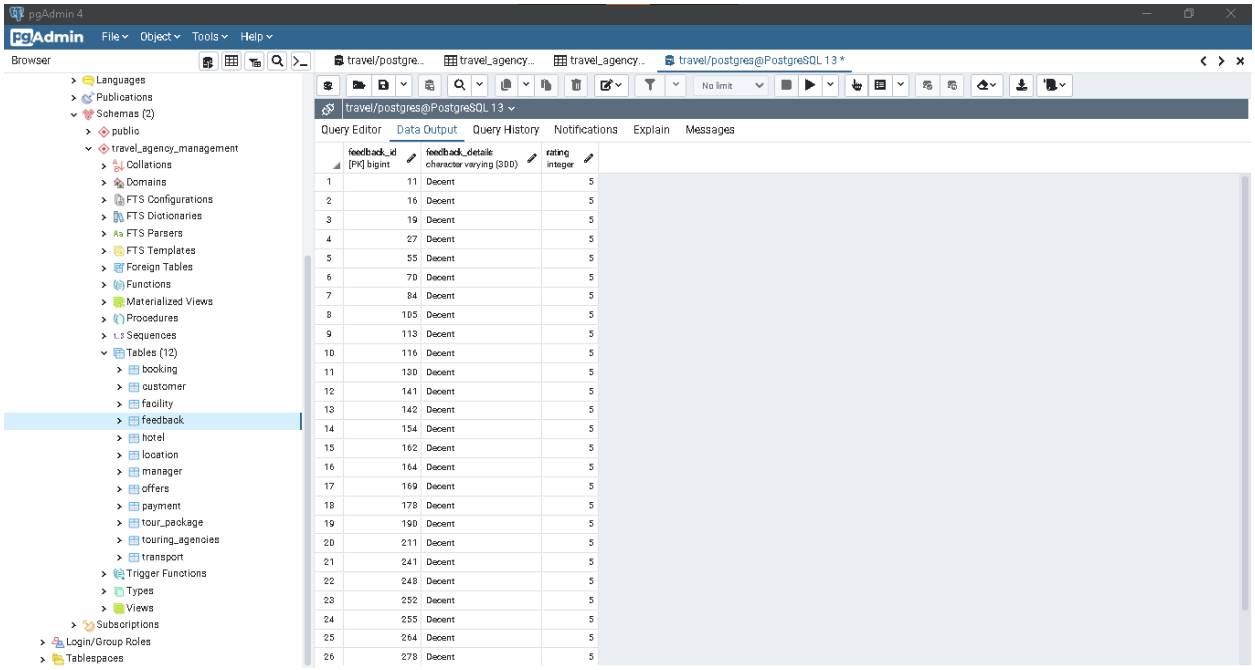


	offer_id [PK] bigint	offer_details character varying (300)	terms_and_conditions character varying (300)
1	2	12	No refund
2	3	6.22	No refund
3	4	15.23	No refund
4	5	19.19	No refund
5	6	8.06	No refund
6	10	15.9	No refund
7	14	14.45	No refund
8	15	25.09	No refund
9	18	23.02	No refund
10	21	12.78	No refund
11	23	9.43	No refund
12	24	28.67	No refund
13	25	3.07	No refund
14	26	21.12	No refund
15	29	11.88	No refund
16	30	1.86	No refund
17	32	11.46	No refund
18	33	25.98	No refund
19	34	23.19	No refund
20	36	26.41	No refund
21	38	22.85	No refund
22	39	17.01	No refund
23	41	27.89	No refund
24	42	11.87	No refund
25	47	11.16	No refund
26	48	6.3	No refund

```
No of Tuples - 159
```

8. Show the feedback details where the feedback given is ‘Decent.’

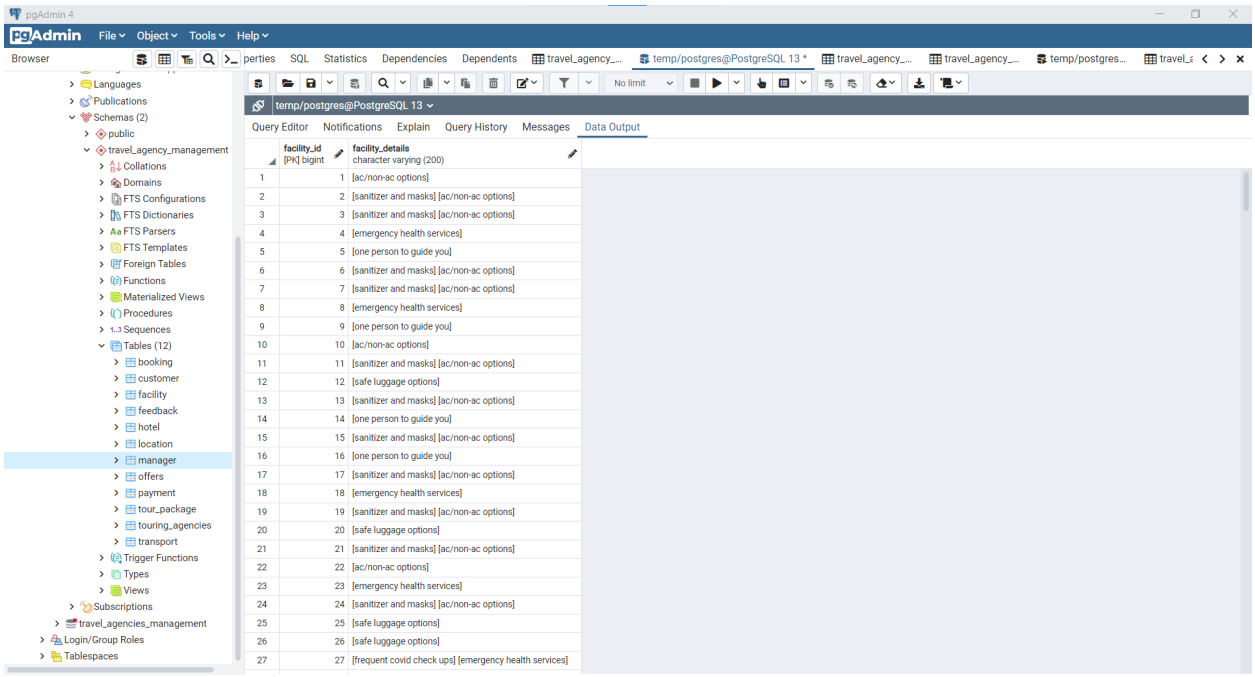
```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM feedback
WHERE feedback_details = 'Decent';
```




```
No. of Tuples - 28
```

9. Show all the facilities which are available for hotels, transport and tour packages.

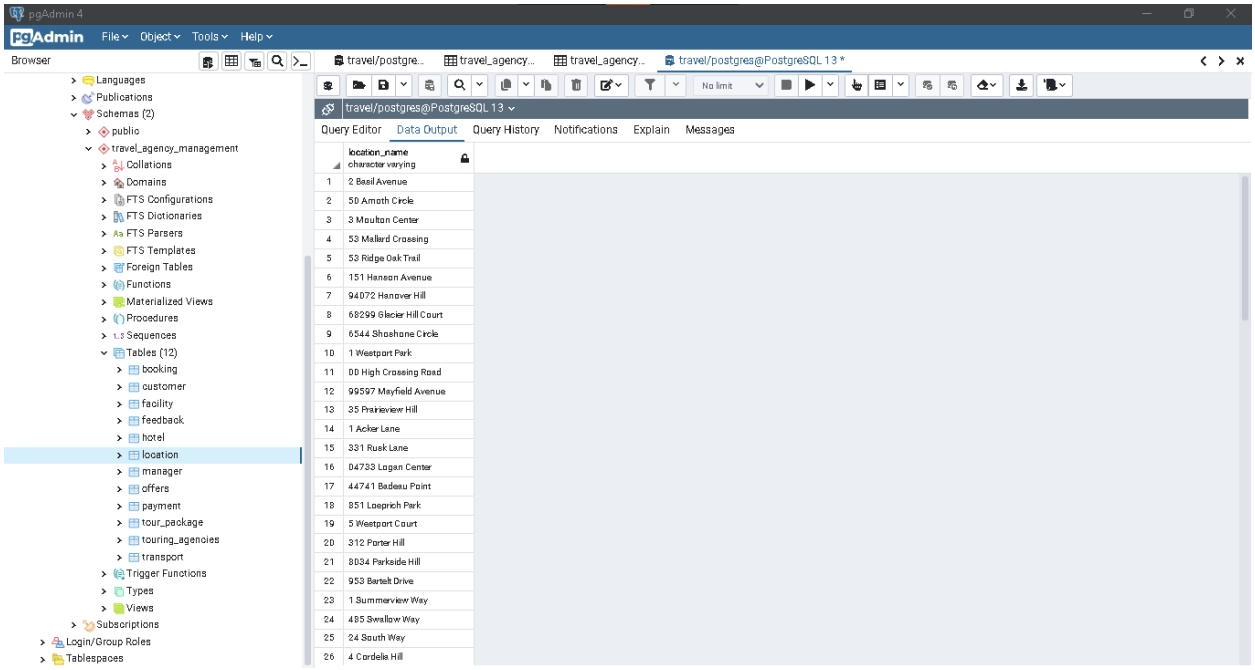
```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM facility;
```




```
No of Tuples - 300.
```

10. Show the details of different places where the number of quarantine days is less than 7 days.

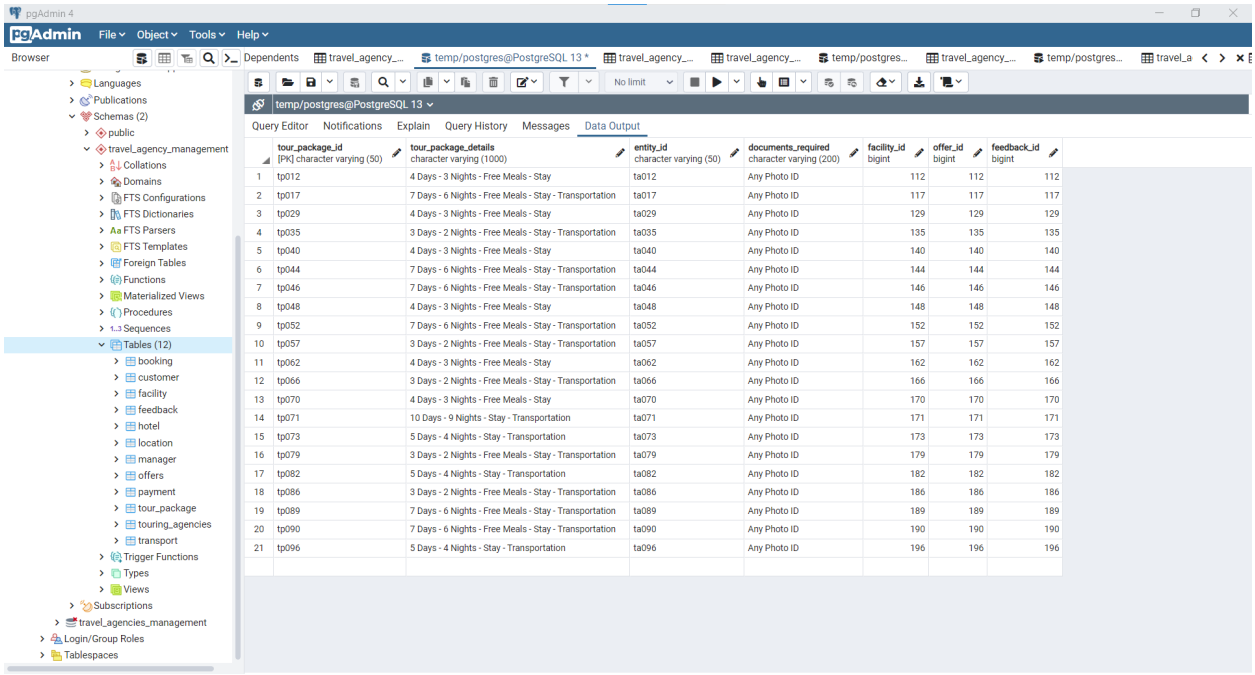
```
SET SEARCH_PATH TO travel_agency_management;
SELECT location_name FROM location
WHERE number_of_quarantine_days < 7;
```




```
No of tuples - 85
```

11. Show the details of different packages where any photo IDs are allowed.

```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM tour_package WHERE documents_required = 'Any Photo ID';
```

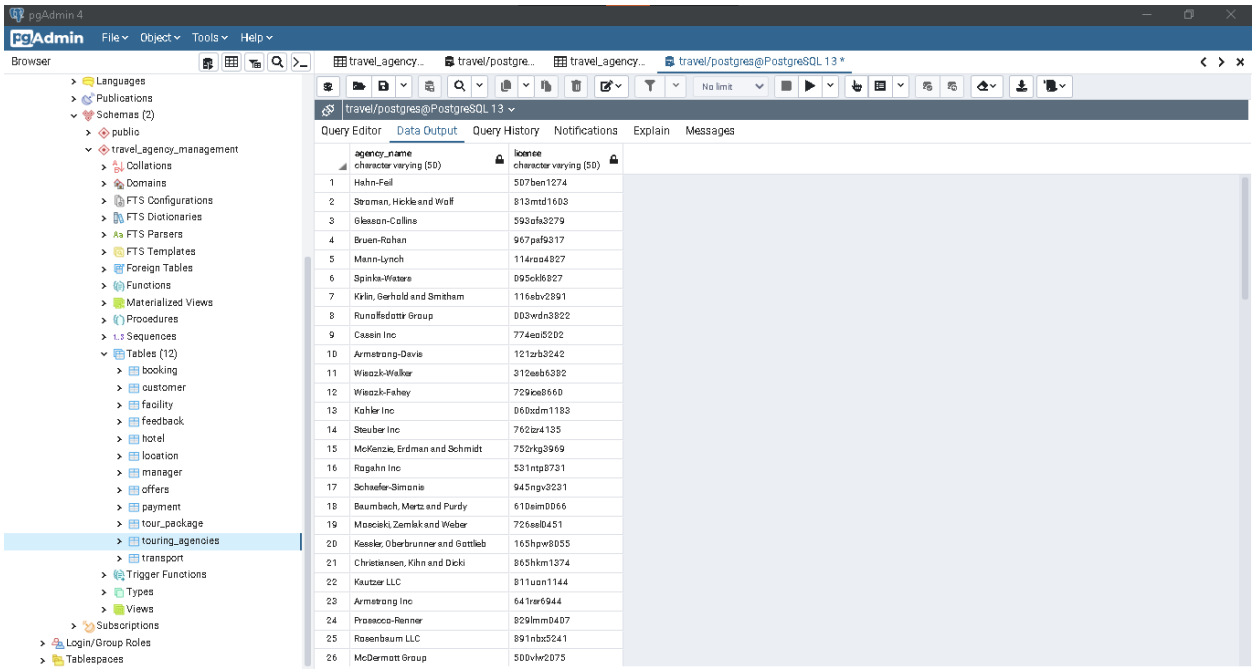


	tour_package_id [PK] character varying (50)	tour_package_details character varying (1000)	entity_id character varying (50)	documents_required character varying (200)	facility_id bigint	offer_id bigint	feedback_id bigint
1	tp012	4 Days - 3 Nights - Free Meals - Stay	ta012	Any Photo ID	112	112	112
2	tp017	7 Days - 6 Nights - Free Meals - Stay - Transportation	ta017	Any Photo ID	117	117	117
3	tp029	4 Days - 3 Nights - Free Meals - Stay	ta029	Any Photo ID	129	129	129
4	tp035	3 Days - 2 Nights - Free Meals - Stay - Transportation	ta035	Any Photo ID	135	135	135
5	tp040	4 Days - 3 Nights - Free Meals - Stay	ta040	Any Photo ID	140	140	140
6	tp044	7 Days - 6 Nights - Free Meals - Stay - Transportation	ta044	Any Photo ID	144	144	144
7	tp046	7 Days - 6 Nights - Free Meals - Stay - Transportation	ta046	Any Photo ID	146	146	146
8	tp048	4 Days - 3 Nights - Free Meals - Stay	ta048	Any Photo ID	148	148	148
9	tp052	7 Days - 6 Nights - Free Meals - Stay - Transportation	ta052	Any Photo ID	152	152	152
10	tp057	3 Days - 2 Nights - Free Meals - Stay - Transportation	ta057	Any Photo ID	157	157	157
11	tp062	4 Days - 3 Nights - Free Meals - Stay	ta062	Any Photo ID	162	162	162
12	tp066	3 Days - 2 Nights - Free Meals - Stay - Transportation	ta066	Any Photo ID	166	166	166
13	tp070	4 Days - 3 Nights - Free Meals - Stay	ta070	Any Photo ID	170	170	170
14	tp071	10 Days - 9 Nights - Stay - Transportation	ta071	Any Photo ID	171	171	171
15	tp073	5 Days - 4 Nights - Stay - Transportation	ta073	Any Photo ID	173	173	173
16	tp079	3 Days - 2 Nights - Free Meals - Stay - Transportation	ta079	Any Photo ID	179	179	179
17	tp082	5 Days - 4 Nights - Stay - Transportation	ta082	Any Photo ID	182	182	182
18	tp086	3 Days - 2 Nights - Free Meals - Stay - Transportation	ta086	Any Photo ID	186	186	186
19	tp089	7 Days - 6 Nights - Free Meals - Stay - Transportation	ta089	Any Photo ID	189	189	189
20	tp090	7 Days - 6 Nights - Free Meals - Stay - Transportation	ta090	Any Photo ID	190	190	190
21	tp096	5 Days - 4 Nights - Stay - Transportation	ta096	Any Photo ID	196	196	196

```
No of Tuples - 21.
```

12. Show the names of different touring agencies along with their license. (Only two attributes are needed in output).

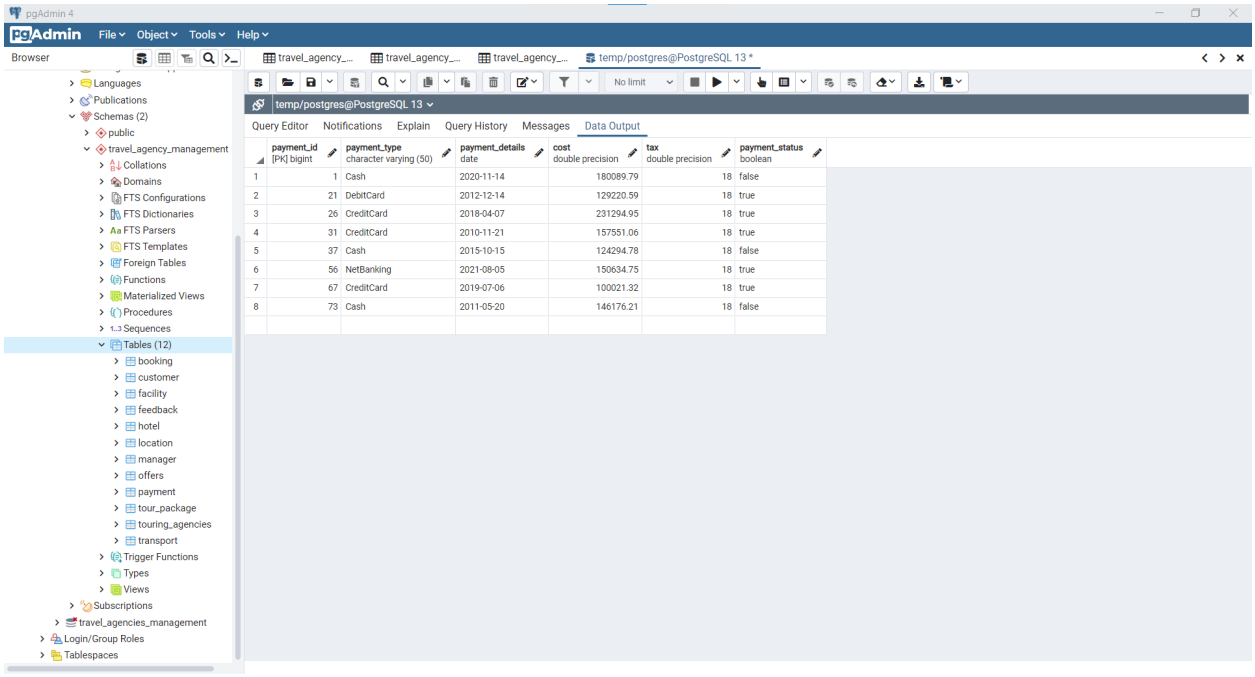
```
SET SEARCH_PATH TO travel_agency_management;
SELECT agency_name, license FROM touring_agencies;
```




```
No of Tuples = 100
```

13. Show the payment details where the amount is between 1,00,000 and 2,50,000.

```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM payment WHERE cost BETWEEN 100000 AND 250000;
```

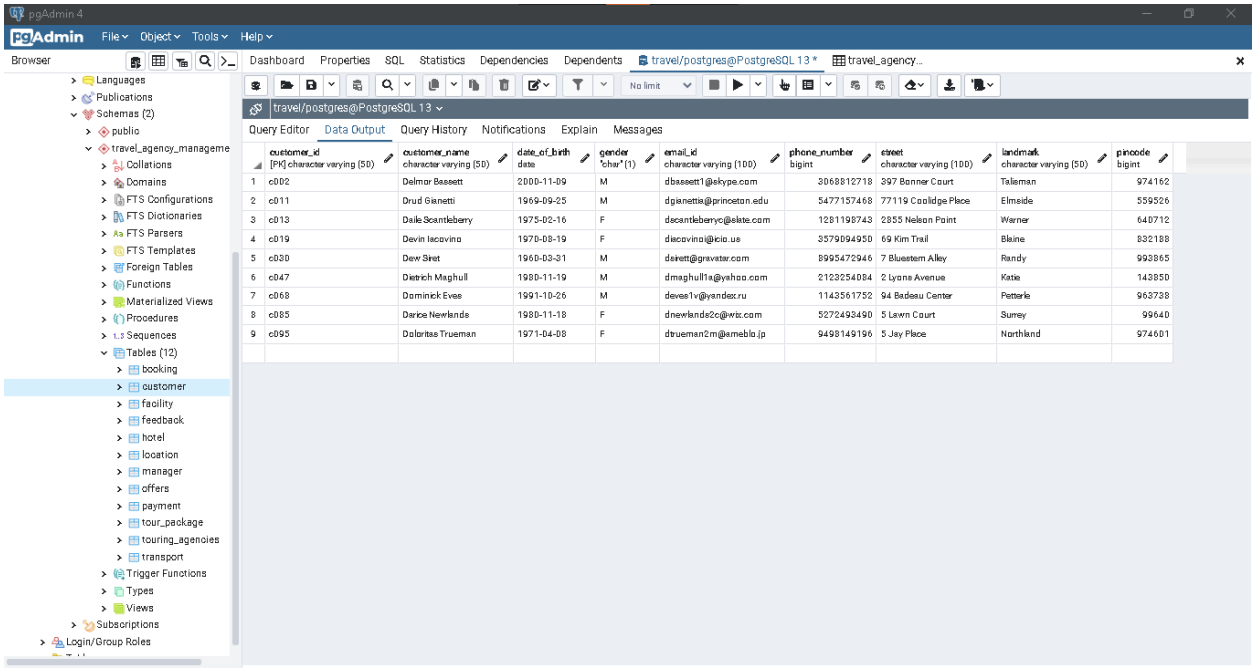


payment_id [PK] bigint	payment_type character varying (50)	payment_details date	cost double precision	tax double precision	payment_status boolean
1	Cash	2020-11-14	180089.79	18	false
21	DebitCard	2012-12-14	129220.59	18	true
26	CreditCard	2018-04-07	231294.95	18	true
31	CreditCard	2010-11-21	157551.06	18	true
37	Cash	2015-10-15	124294.78	18	false
56	NetBanking	2021-08-05	150634.75	18	true
67	CreditCard	2019-07-06	100021.32	18	true
73	Cash	2011-05-20	146176.21	18	false

```
No of Tuples = 8
```

14. Show the details of all the customers whose name starts with the letter ‘D,’

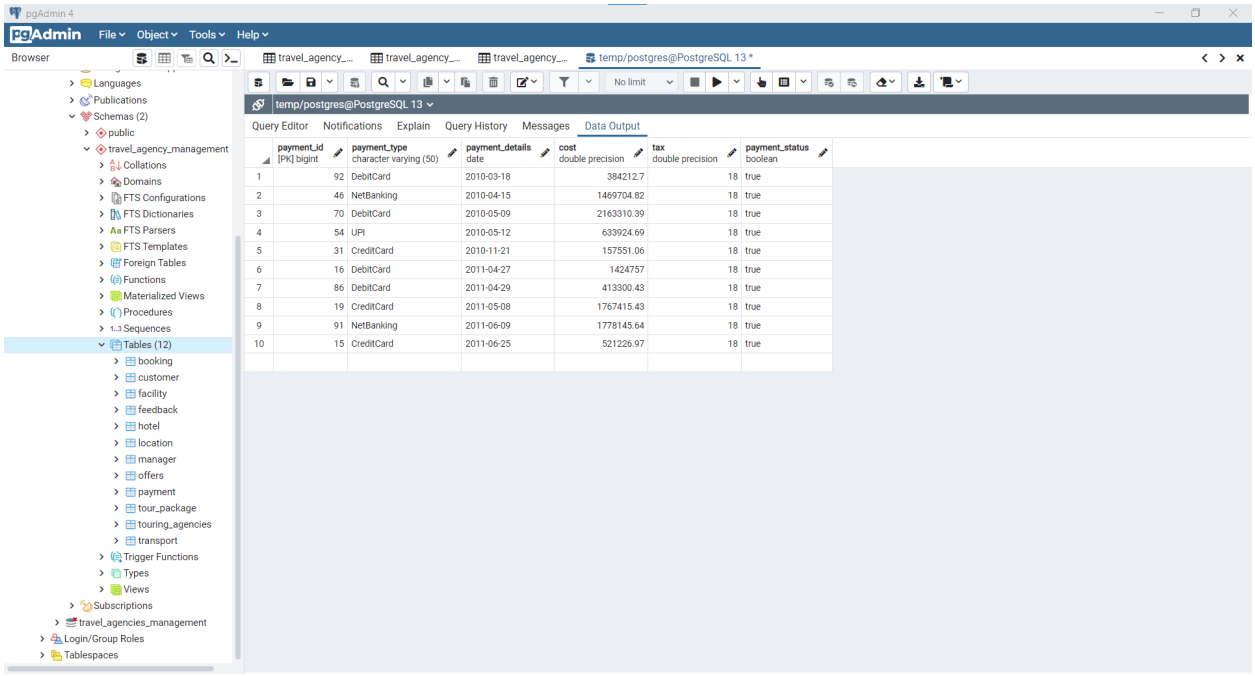
```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM customer
WHERE customer_name LIKE 'D%';
```




```
No of Tuples = 9
```

15. Show the details of the latest ten payments where the status is ‘True.’

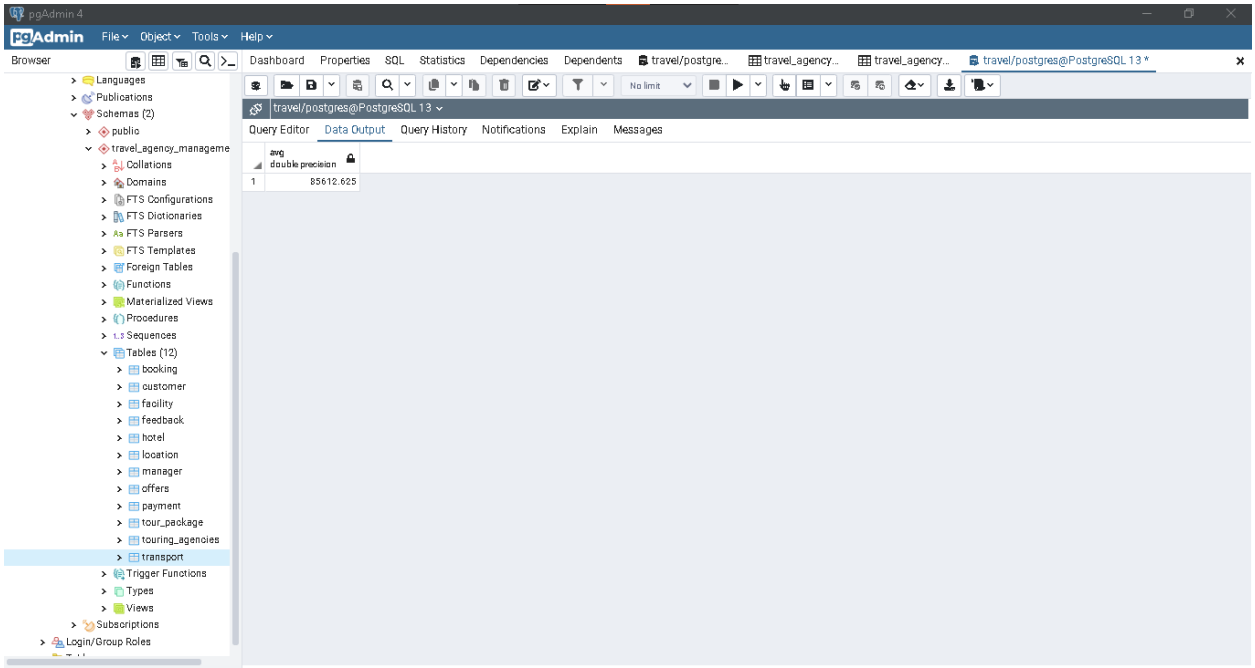
```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM payment WHERE payment_status = 'true'
ORDER BY payment_details
LIMIT 10;
```




```
No of Tuples = 10
```

16. Calculate the average price of transportation where buses are used.

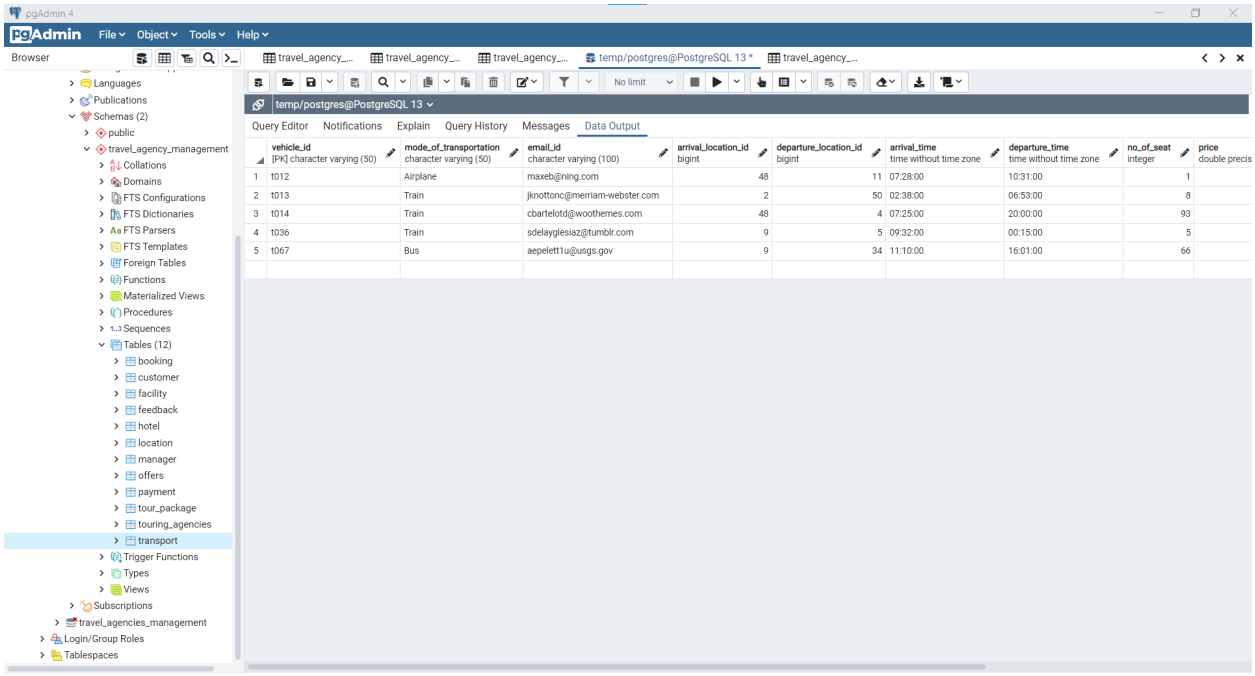
```
SET SEARCH_PATH TO travel_agency_management;
SELECT AVG(price) FROM transport
WHERE mode_of_transportation = 'Bus';
```




```
No of Tuples = 1
```

17. Count the number of locations where the arrival location ID is in (2,9,48) and departure location ID is in (4,5,11,34,50).

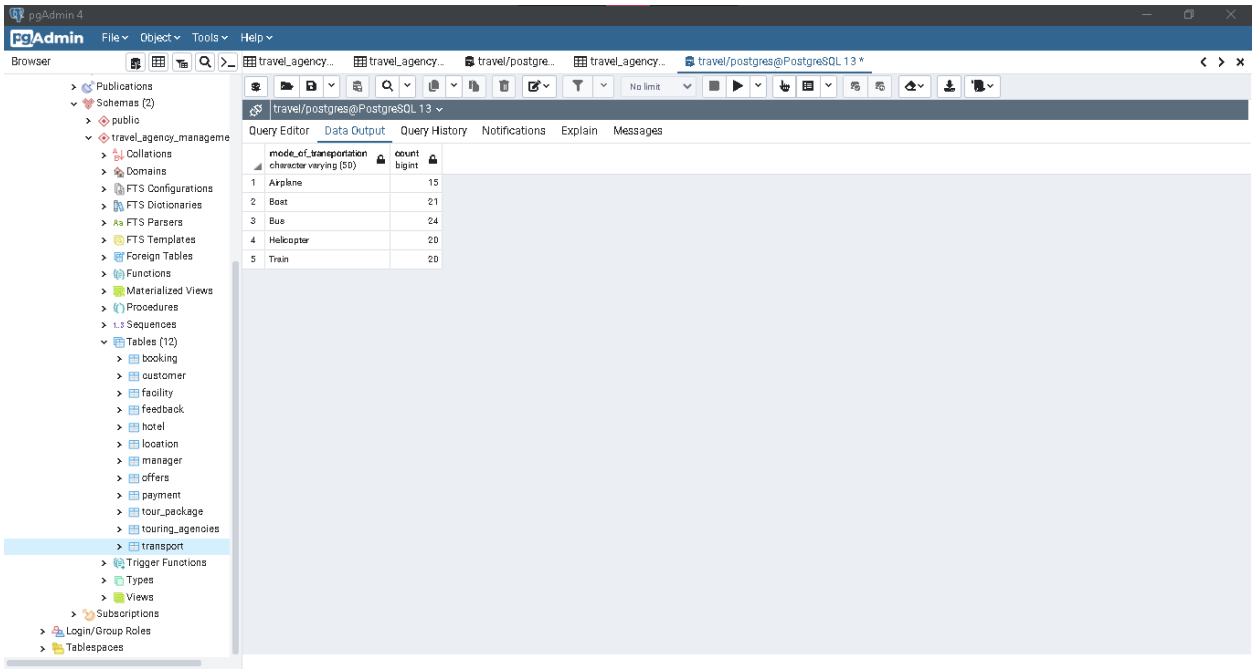
```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM transport WHERE arrival_location_id IN (2,9,48) AND
departure_location_id IN (4,5,11,34,50);
```




```
No of Tuples = 5.
```

18. Count the number of vehicles of different modes of transportation used for traveling.

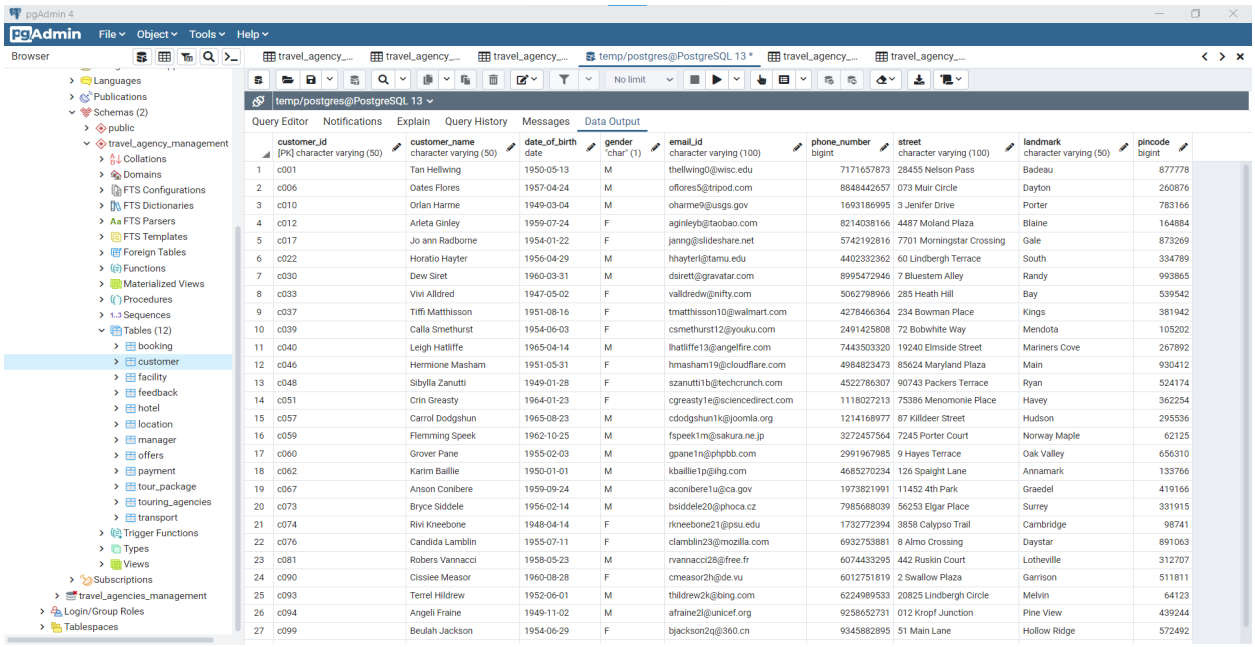
```
SET SEARCH_PATH TO travel_agency_management;
SELECT mode_of_transportation, count(vehicle_id) as count FROM transport
Group BY mode_of_transportation
ORDER BY mode_of_transportation
```




```
No of Tuples - 5
```

19. Show the details of all the customers whose age is greater than 55 years.

```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM customer WHERE date_of_birth < '1965-11-16';
```

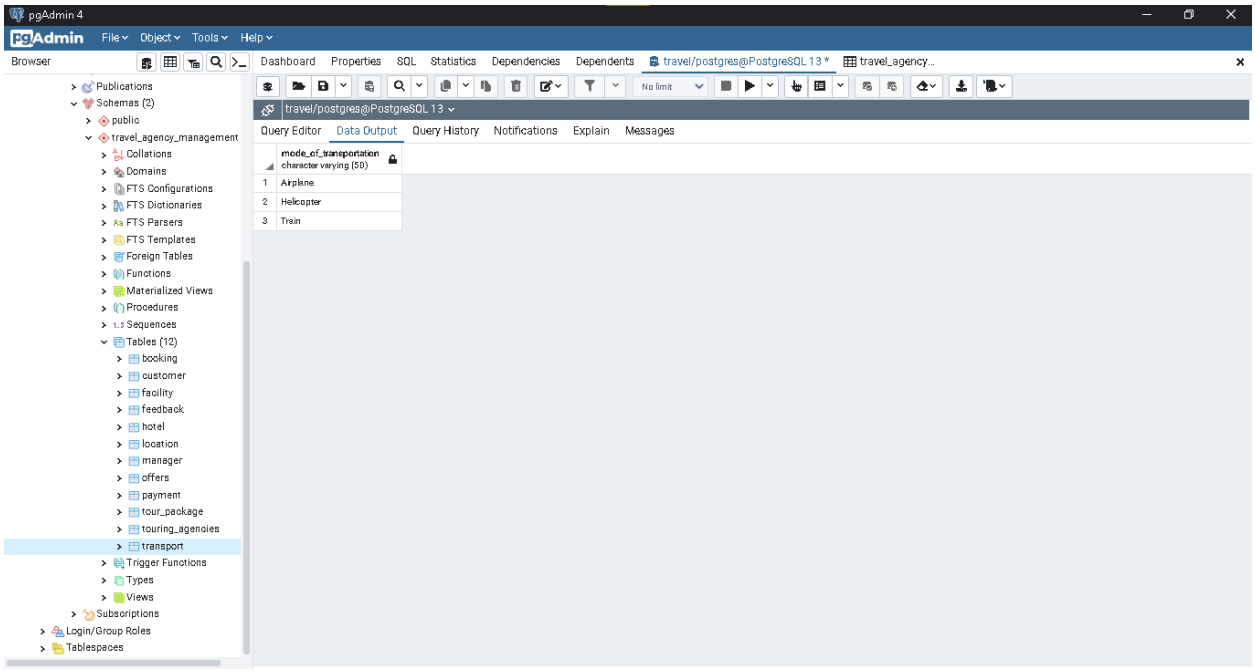


	customer_id [PK] character varying (50)	customer_name character varying (50)	date_of_birth date	gender "char" (1)	email_id character varying (100)	phone_number bigint	street character varying (100)	landmark character varying (50)	pincode bigint
1	c001	Tan Hellwing	1950-05-13	M	thellwing0@wisc.edu	7171657873	28455 Nelson Pass	Badeau	877778
2	c006	Oates Flores	1957-04-24	M	oflores5@tripod.com	8848442657	073 Muir Circle	Dayton	260876
3	c010	Orlan Harme	1949-03-04	M	oharme9@usgs.gov	1693186995	3 Jenifer Drive	Porter	783166
4	c012	Arleta Ginley	1959-07-24	F	aginleyb@taobao.com	8214038166	4487 Moland Plaza	Blaine	164884
5	c017	Jo ann Radborne	1954-01-22	F	janng@slideshare.net	5742192816	7701 Morningstar Crossing	Gale	873269
6	c022	Horatio Hayter	1956-04-29	M	hhayterl@tamu.edu	4402332362	60 Lindbergh Terrace	South	334789
7	c030	Dew Siret	1960-03-31	M	dsirett@gravatar.com	8995472946	7 Bluestem Alley	Randy	993865
8	c033	Vivi Alldred	1947-05-02	F	valldredw@nifty.com	5062798966	285 Heath Hill	Bay	539542
9	c037	Tiffi Matthisson	1951-08-16	F	tmatthisson10@walmart.com	4278466364	234 Bowman Place	Kings	381942
10	c039	Calla Smethurst	1954-06-03	F	csmethurst12@youku.com	2491425808	72 Bobwhite Way	Mendota	105202
11	c040	Leigh Hatliffe	1965-04-14	M	lhatliffe13@angelfire.com	7443503320	19240 Elmside Street	Mariners Cove	267892
12	c046	Hermione Masham	1951-05-31	F	hmasham19@cloudflare.com	4984823473	85624 Maryland Plaza	Main	930412
13	c048	Sibylla Zanutti	1949-01-28	F	szanutti1b@techcrunch.com	4522786307	90743 Packers Terrace	Ryan	524174
14	c051	Crin Greasty	1964-01-23	F	cgreasty1e@sciencedirect.com	1118027213	75386 Menomonie Place	Havey	362254
15	c057	Carrol Dodgshun	1965-08-23	M	cdodgshun1k@joomla.org	1214168977	87 Killdeer Street	Hudson	295536
16	c059	Flemming Speek	1962-10-25	M	fspeek1m@sakura.ne.jp	3272457564	7245 Porter Court	Norway Maple	62125
17	c060	Grover Pane	1955-02-03	M	gpane1n@phpbb.com	2991967985	9 Hayes Terrace	Oak Valley	656310
18	c062	Karim Baillie	1950-01-01	M	kbaillie1p@ihg.com	4685270234	126 Spaight Lane	Annamark	133766
19	c067	Anson Conibere	1959-09-24	M	aconibere1u@ca.gov	1973821991	11452 4th Park	Graedel	419166
20	c073	Bryce Siddele	1956-02-14	M	bsiddele20@phoca.cz	7985688039	56253 Elgar Place	Surrey	331915
21	c074	Rivi Kneebone	1948-04-14	F	rkneebone21@psu.edu	1732772394	3858 Calypso Trail	Cambridge	98741
22	c076	Candida Lamblin	1955-07-11	F	clamblin23@mozilla.com	6932753881	8 Almo Crossing	Daystar	891063
23	c081	Robers Vannacci	1958-05-23	M	rvannacci28@free.fr	6074433295	442 Ruskin Court	Lotheville	312707
24	c090	Cissiee Measor	1960-08-28	F	cmeasor2h@de.vu	6012751819	2 Swallow Plaza	Garrison	511811
25	c093	Terrel Hildrew	1952-06-01	M	thildrew2k@bing.com	6224989533	20825 Lindbergh Circle	Melvin	64123
26	c094	Angeli Fraine	1949-11-02	M	afraine2l@unicef.org	9258652731	012 Kropf Junction	Pine View	439244
27	c099	Beulah Jackson	1954-06-29	F	bjackson2q@360.cn	9345882895	51 Main Lane	Hollow Ridge	572492

```
No of Tuples = 27
```

20. Show the names of different modes of transportation whose average price is less than 80000.

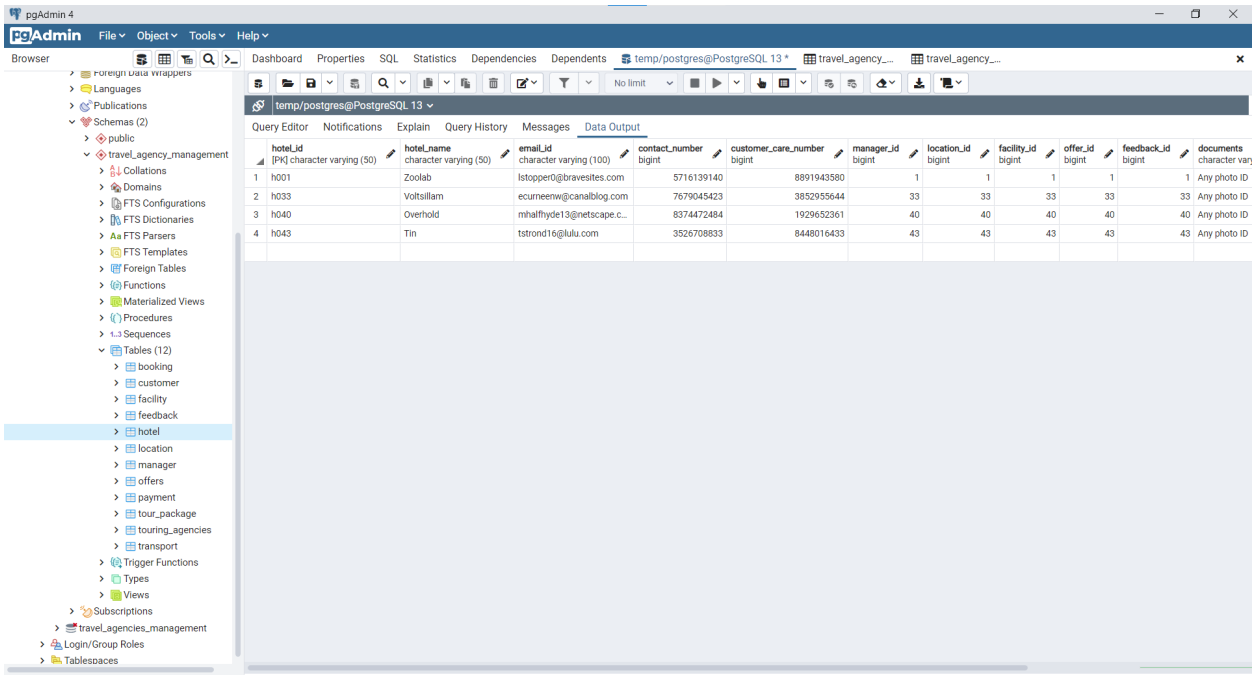
```
set search_path to travel_agency_management;
SELECT mode_of_transportation FROM transport
Group BY mode_of_transportation
having avg(price) <= 80000
ORDER BY mode_of_transportation
```




```
No of Tuples = 3
```

21. Show the details of the hotel where any photo ID can be used, and the license number should start with ‘9’.

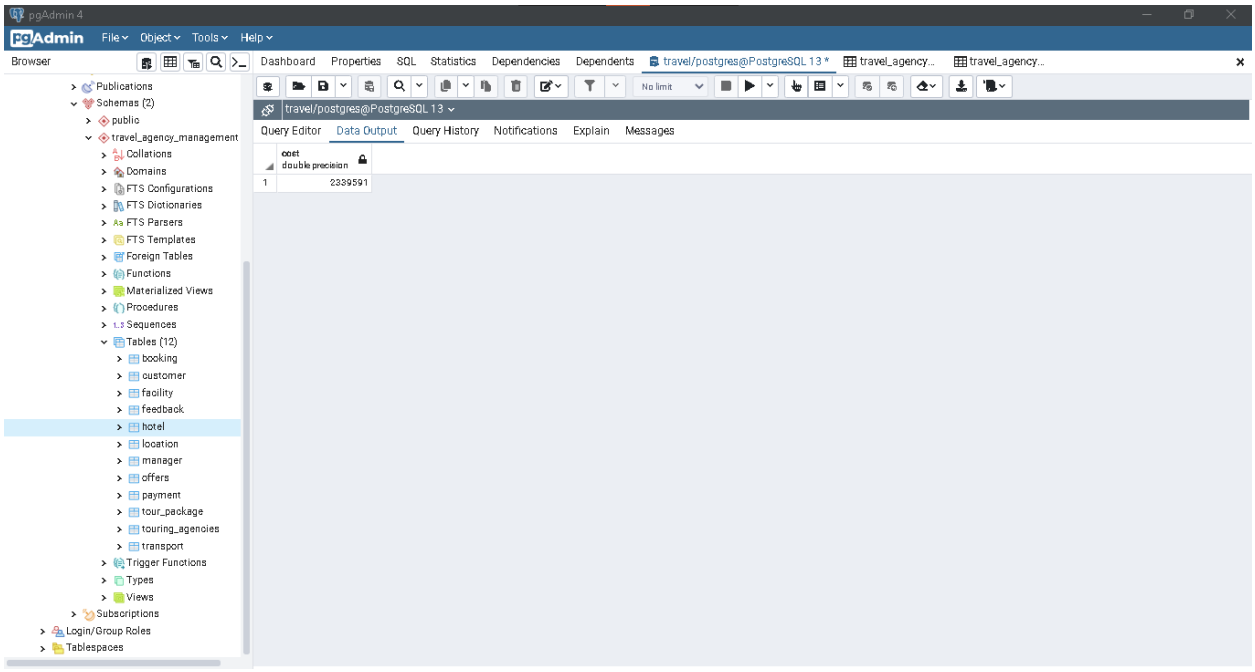
```
SET SEARCH_PATH TO travel_agency_management;
SELECT * FROM hotel WHERE documents = 'Any photo ID' AND license LIKE '9%';
```




```
No of Tuples - 4.
```

22. Show the third-highest payment made by a customer.

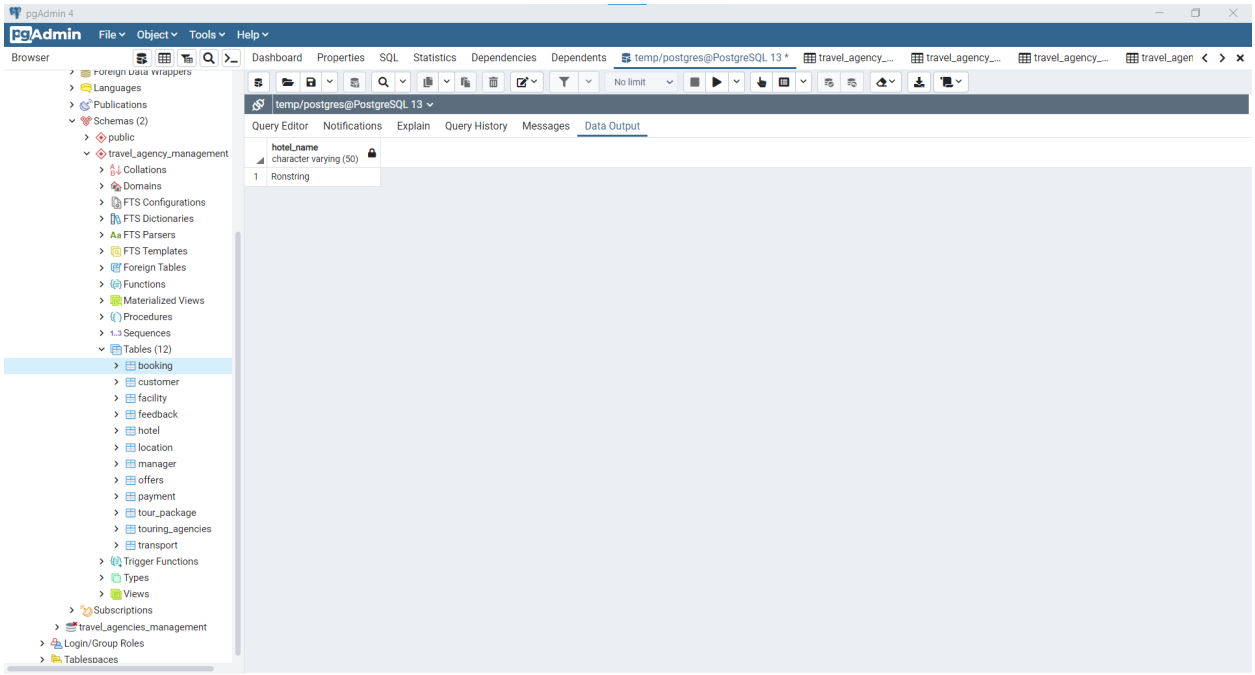
```
set search_path to travel_agency_management;
SELECT cost FROM
(SELECT distinct cost FROM payment
ORDER BY cost DESC
LIMIT 3) AS Comp
ORDER BY cost
LIMIT 1;
```




```
No of Tuples = 1
```

23. Show the name of the hotel where Ingemar Carlyon has stayed.

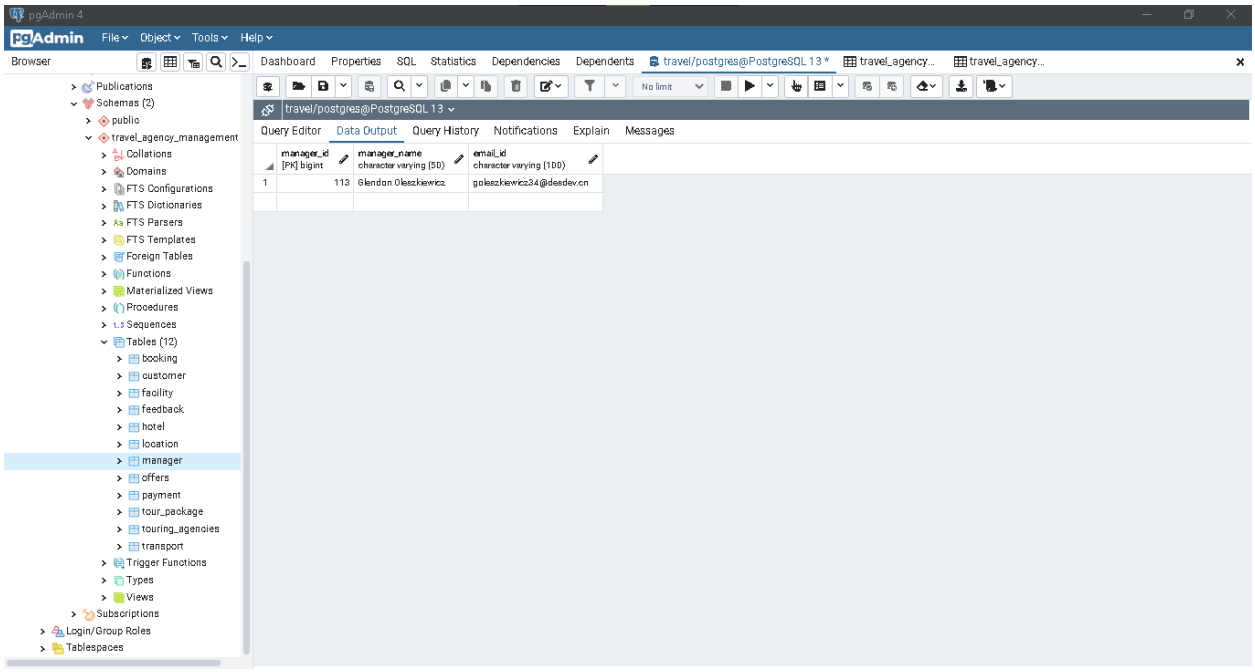
```
SET SEARCH_PATH TO travel_agency_management;
SELECT hotel_name
FROM booking
JOIN customer
ON customer.customer_id = booking.customer_id
JOIN hotel
ON hotel.hotel_id = booking.hotel_id
WHERE customer_name = 'Ingemar Carlyon';
```




```
No of Tuples = 1
```

24. Give the details of the manager of a touring agency named ‘Kohler Inc.’

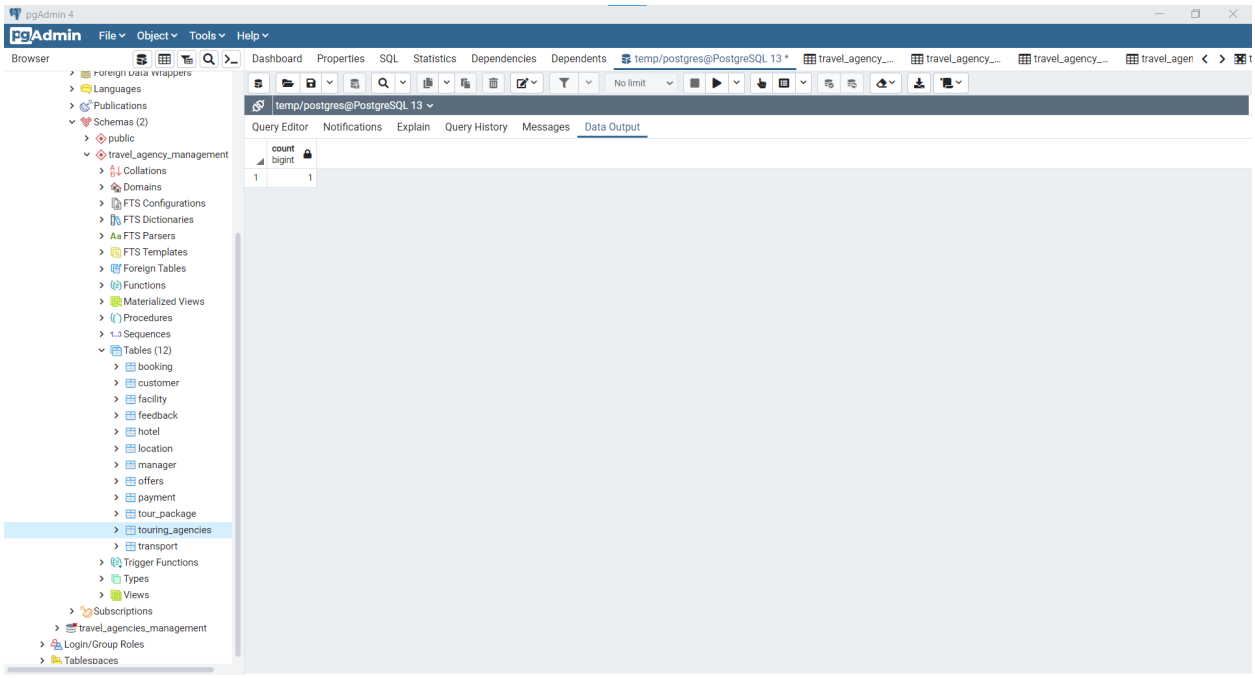
```
set search_path to travel_agency_management;
SELECT manager.manager_id, manager_name, manager.email_id
FROM manager
join touring_agencies
on manager.manager_id = touring_agencies.manager_id
where touring_agencies.agency_name = 'Kohler Inc'
```




```
No of Tuples = 1
```

25. For a particular vehicle ID (t054), show the number of bookings done.

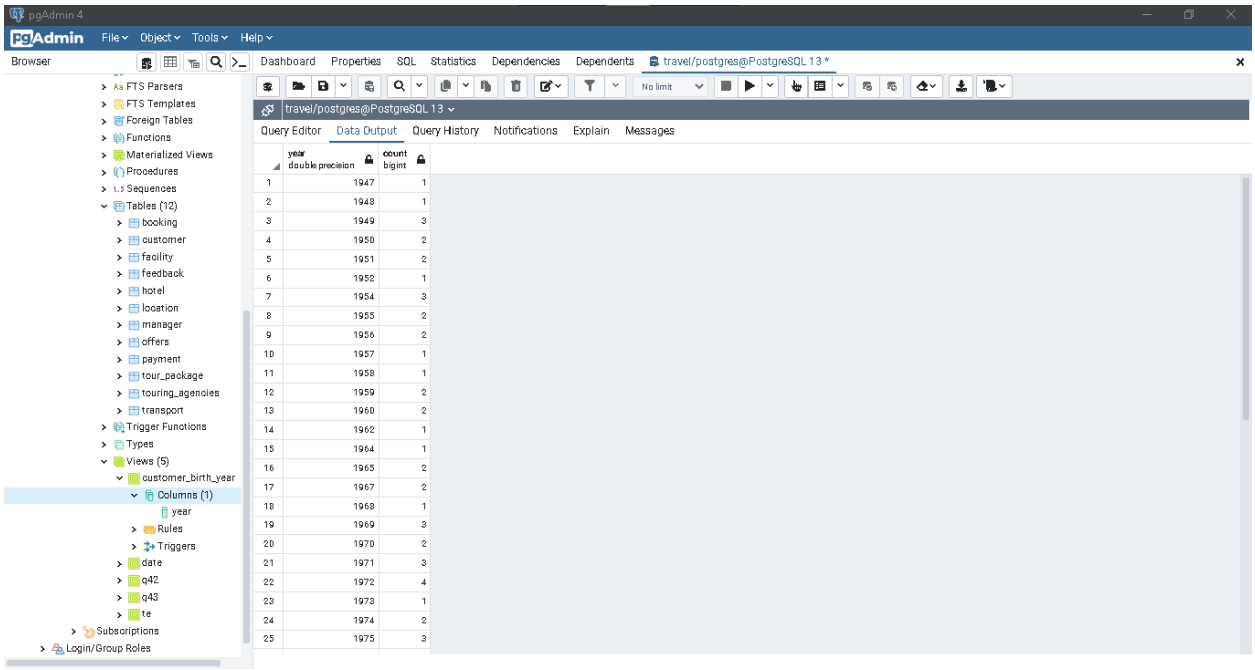
```
SET SEARCH_PATH TO travel_agency_management;
SELECT COUNT(*)FROM booking
JOIN transport ON booking.transport_id = transport.vehicle_id
WHERE transport_id = 't054';
```




```
No of Tuples - 1.
```

26. Count the number of customers born in every year right from 1947 to the present year.

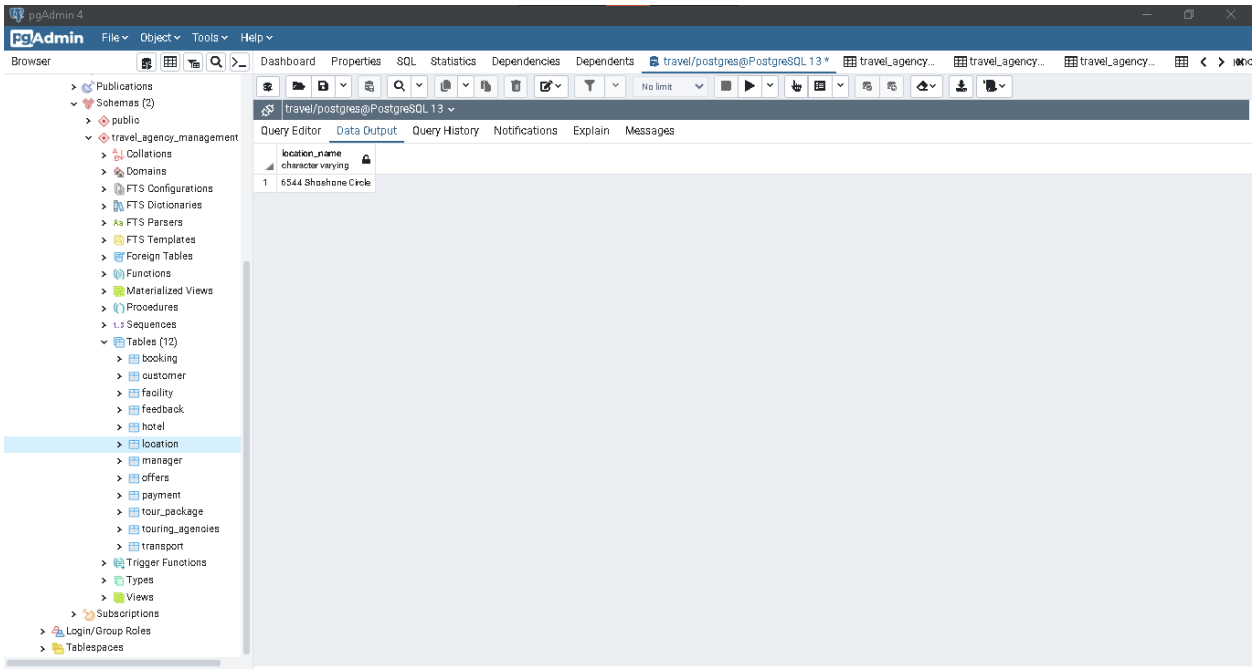
```
set search_path to travel_agency_management;
create view date as
select date_of_birth from customer;

create view customer_birth_year as
select extract (year from date_of_birth) as year from date;
select year ,count(*) from customer_birth_year
group by year
order by year asc;
```




```
No of Tuples = 49
```

27. Where is the Span hotel located?

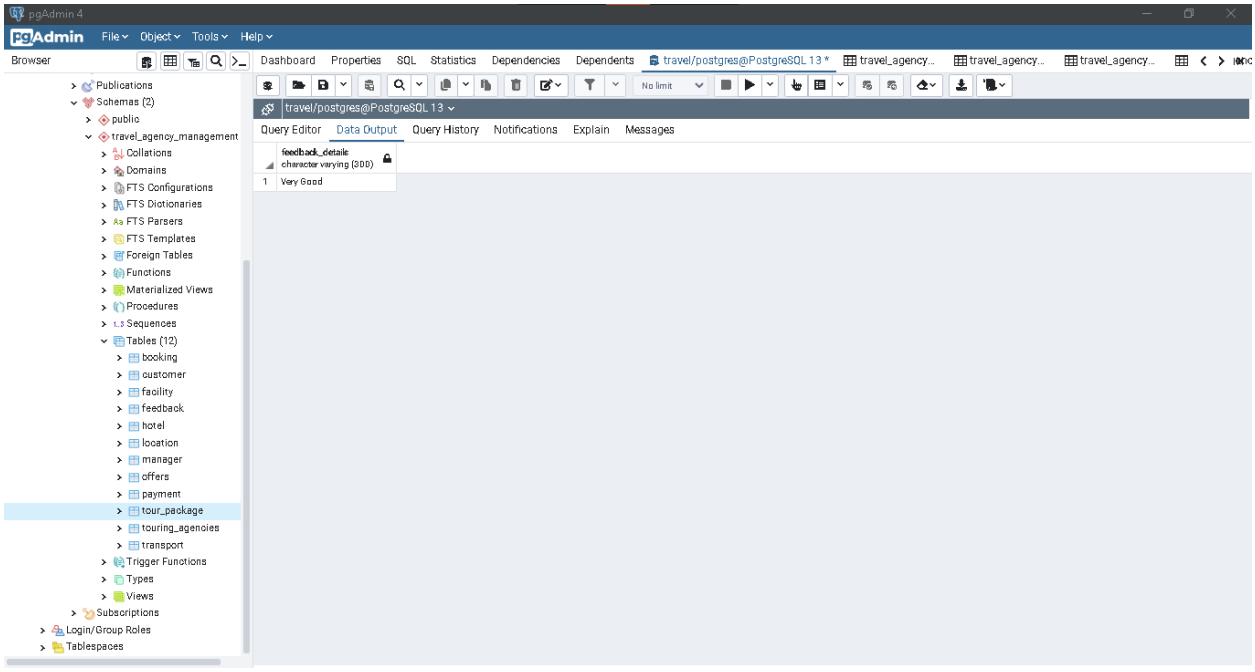
```
set search_path to travel_agency_management;
select location_name from location
join hotel on location.location_id = hotel.location_id
where hotel.hotel_name = 'Span'
```




```
No. of Tuple = 1
```

28. What feedback has been provided by the consumer for the tp056?

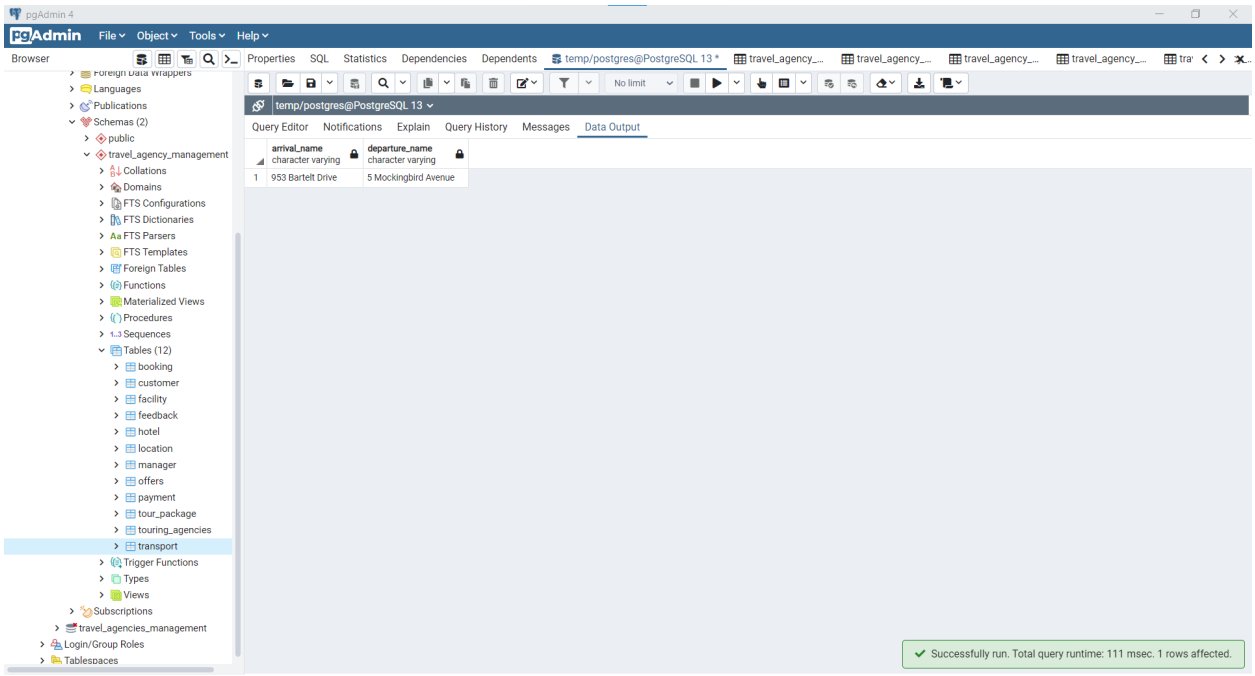
```
set search_path to travel_agency_management;
SELECT feedback_details
FROM booking
JOIN customer
ON customer.customer_id = booking.customer_id
JOIN tour_package
ON tour_package.tour_package_id = booking.tour_package_id
JOIN feedback
ON feedback.feedback_id = tour_package.feedback_id
WHERE tour_package.tour_package_id = 'tp056';
```




```
No. of Tuple = 1
```

29. What are the arrival location and departure location of transport with ID t012?

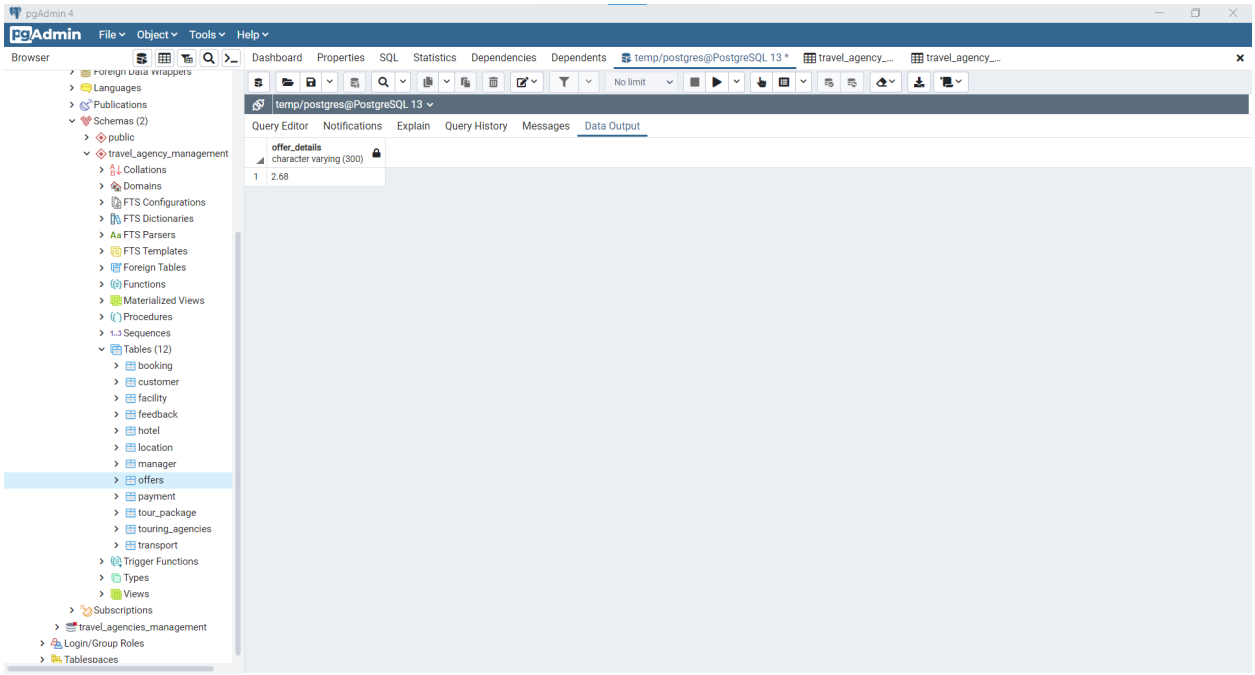
```
SET SEARCH_PATH TO travel_agency_management;
SELECT location_name AS arrival_name,(SELECT location_name AS
departure_name FROM location JOIN transport
ON transport.departure_location_id = location.location_id
WHERE vehicle_id = 't012')
FROM location
JOIN transport
ON transport.arrival_location_id = location.location_id
WHERE vehicle_id = 't012';
```




```
No of Tuples - 1.
```

30. What are the offers provided by hotel ‘Sub-Ex’.

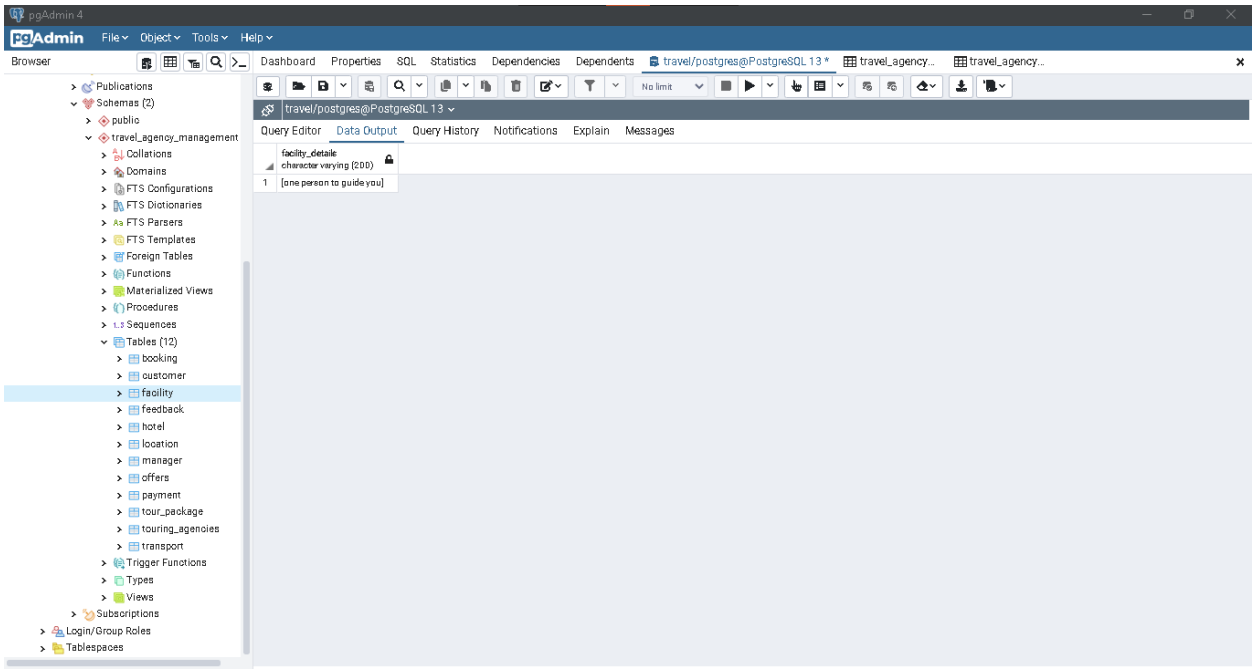
```
SET SEARCH_PATH TO travel_agency_management;
SELECT offer_details
FROM offers
JOIN hotel
ON offers.offer_id = hotel.offer_id
WHERE hotel_name = 'Sub-Ex';
```




```
No of Tuples - 1.
```

31. What are the facilities available for the tour package with ID tp025?

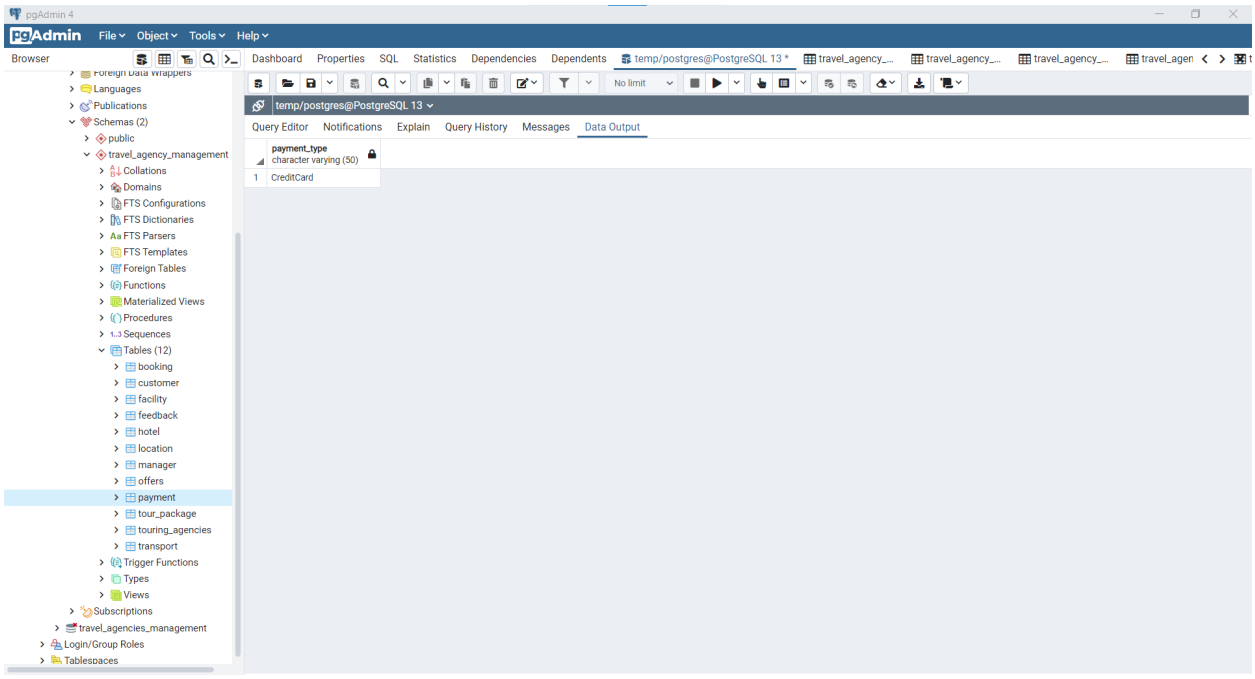
```
set search_path to travel_agency_management;
SELECT facility_details from tour_package
join facility
on tour_package.facility_id = facility.facility_id
where tour_package.tour_package_id = 'tp025'
```




```
No. of Tuple - 1
```

32. What is the payment made for booking with ID ‘22’?

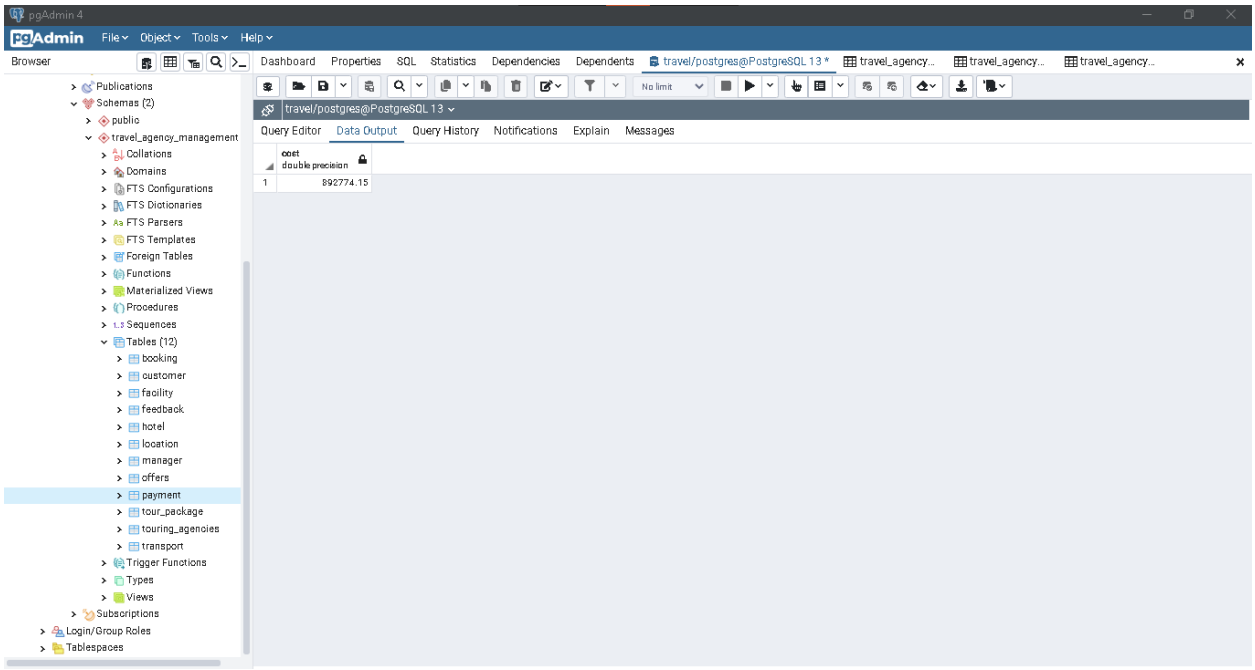
```
SET SEARCH_PATH TO travel_agency_management;
SELECT payment_type
FROM payment
JOIN booking
ON payment.payment_id = booking.payment_id
WHERE payment.payment_id = 22;
```




```
No of Tuples - 1.
```

33. What is the payment made by Zoe Gibbie?

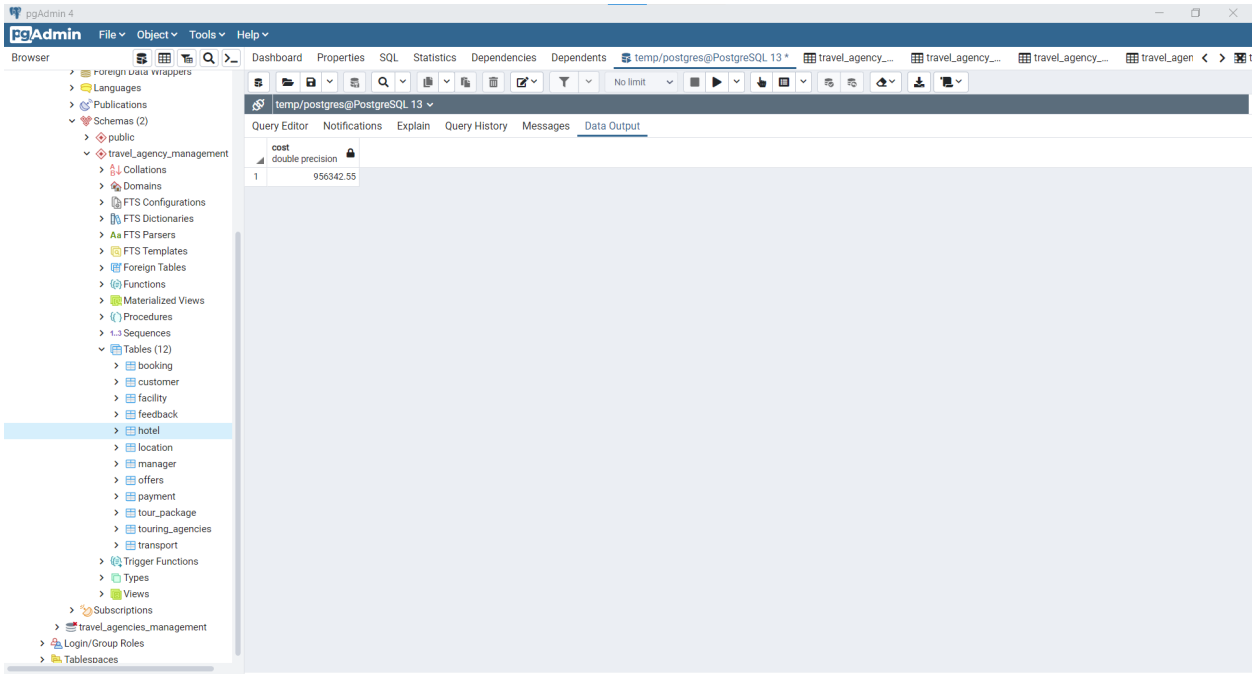
```
set search_path to travel_agency_management;
SELECT payment.cost
FROM booking
JOIN customer
ON customer.customer_id = booking.customer_id
JOIN payment
ON payment.payment_id = booking.payment_id
WHERE customer.customer_name = 'Zoe Gibbie';
```




```
No. of Tuple - 1
```

34. What is the payment received by hotel Span?

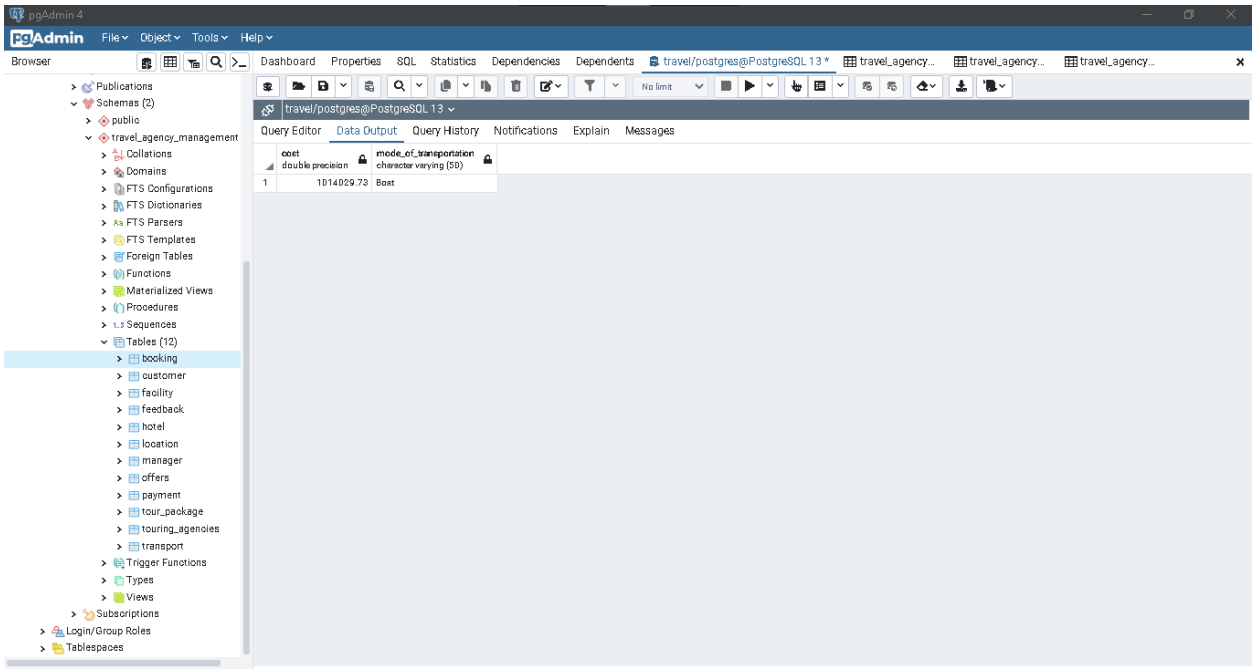
```
SET SEARCH_PATH TO travel_agency_management;
SELECT payment.cost
FROM booking
JOIN payment
ON payment.payment_id = booking.payment_id
JOIN hotel
ON hotel.hotel_id = booking.hotel_id
WHERE hotel_name = 'Span';
```




```
No of Tuples - 1.
```

35. What is the payment received by transport with ID t062 and also state the mode of transportation?

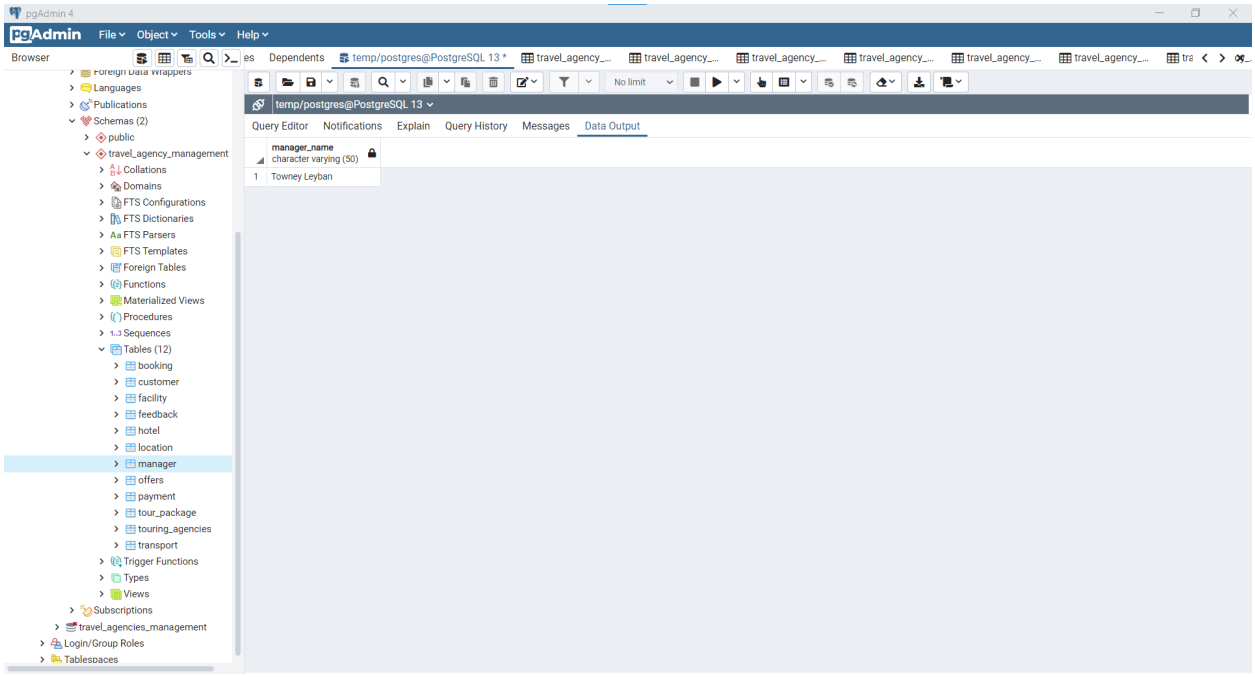
```
set search_path to travel_agency_management;
SELECT payment.cost, transport.mode_of_transportation
FROM booking
JOIN transport
ON transport.vehicle_id = booking.transport_id
JOIN payment
ON payment.payment_id = booking.payment_id
WHERE transport.vehicle_id = 't062';
```




```
No. of Tuple - 1
```

36. Who is the manager of the touring agency linked with the tour package of ID tp017?

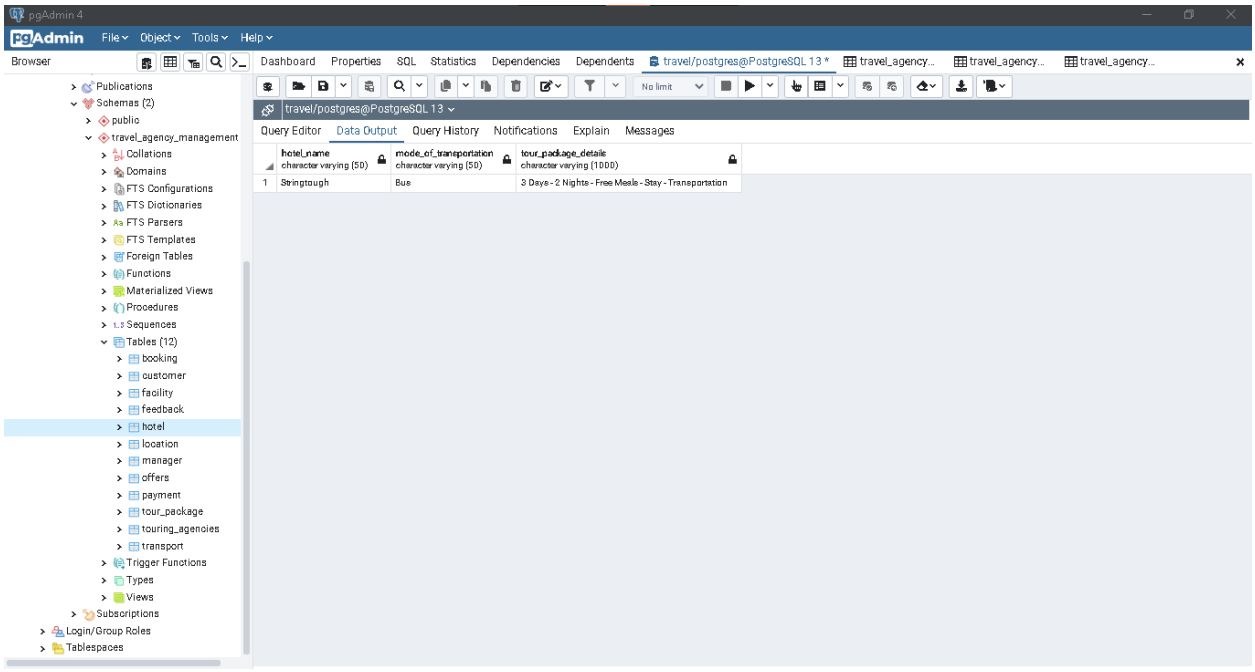
```
SET SEARCH_PATH TO travel_agency_management;
SELECT manager_name
FROM tour_package
JOIN touring_agencies
ON tour_package.entity_id = agency_id
JOIN manager
ON manager.manager_id = touring_agencies.manager_id
WHERE tour_package_id = 'tp017';
```




```
No of Tuples - 1.
```

37. Name of hotel, mode of transportation, and tour package for the booking ID 57.

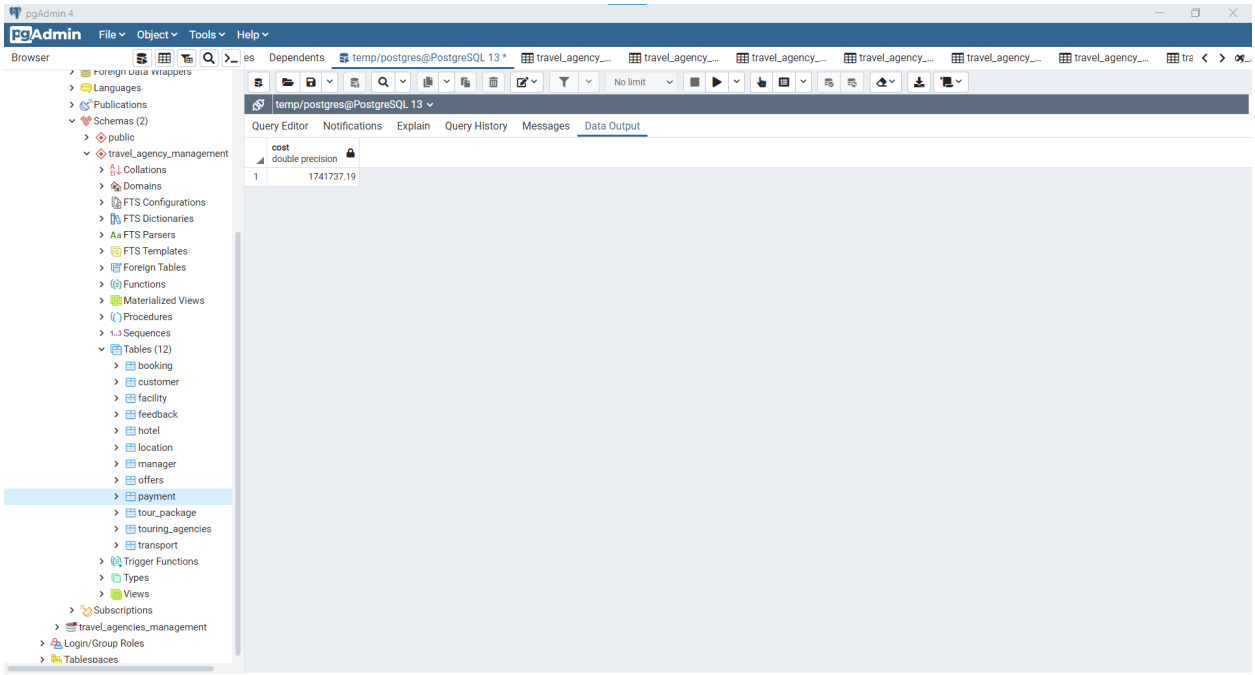
```
set search_path to travel_agency_management;
SELECT hotel_name, (SELECT mode_of_transportation FROM transport JOIN
booking
ON transport.vehicle_id = booking.transport_id
WHERE booking.booking_id = 57), (SELECT tour_package_details FROM
tour_package JOIN booking
ON tour_package.tour_package_id = booking.tour_package_id
WHERE booking.booking_id = 57)
FROM hotel
JOIN booking
ON hotel.hotel_id = booking.hotel_id
WHERE booking.booking_id = 57
```




```
No. of Tuple - 1.
```

38. What is the payment received by the touring agency ‘Bayer Inc’?

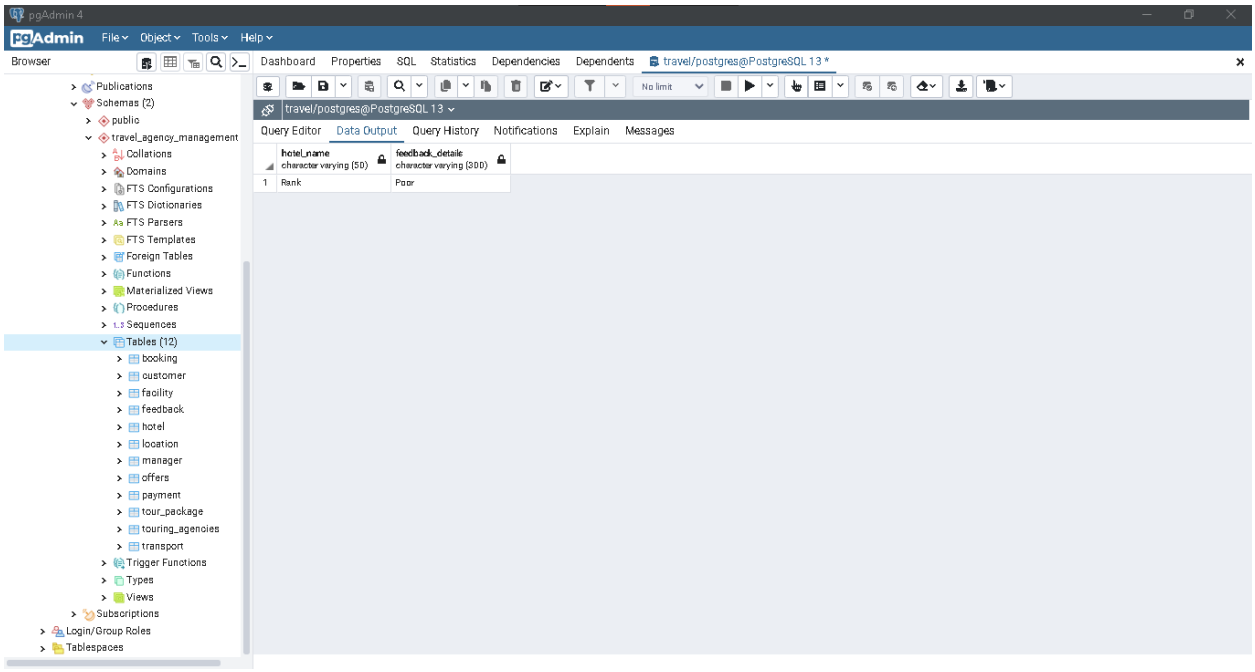
```
SET SEARCH_PATH TO travel_agency_management;
SELECT cost
FROM tour_package
JOIN touring_agencies
ON tour_package.entity_id = agency_id
JOIN booking
ON tour_package.tour_package_id = booking.tour_package_id
JOIN payment
ON booking.payment_id = payment.payment_id
WHERE agency_name = 'Bayer Inc';
```




```
No of Tuples - 1.
```

39. Show the facilities received by Essy Bolzen in the hotel where she stayed and also display the feedback provided by her.

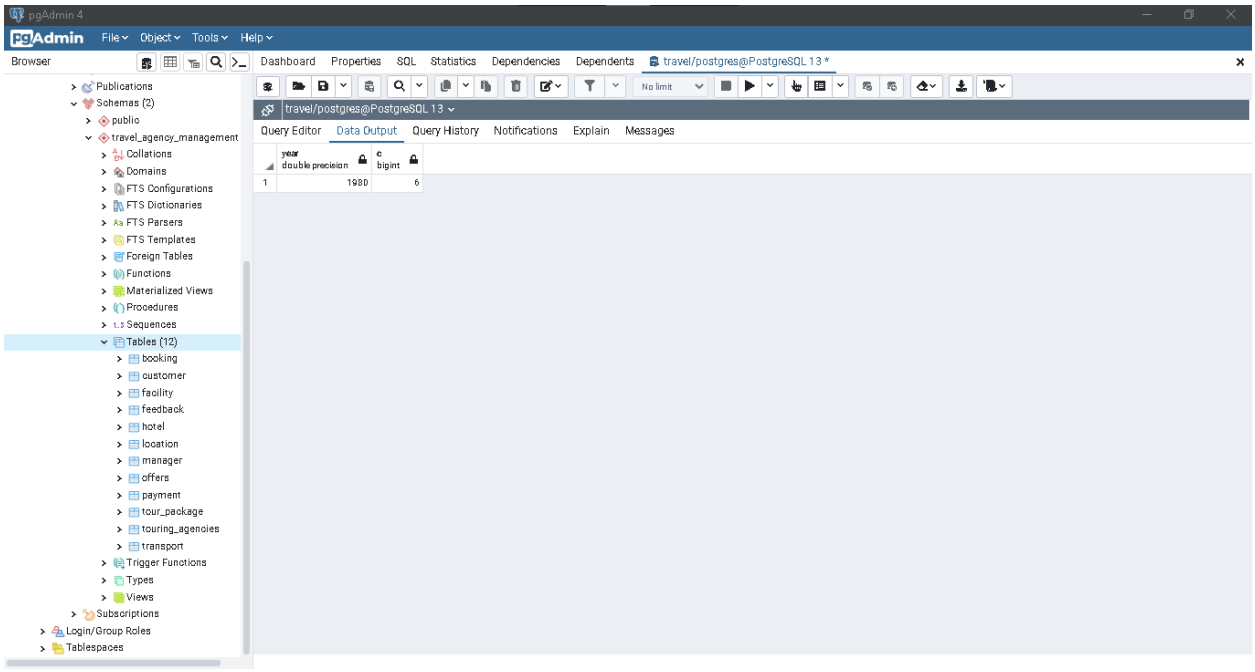
```
set search_path to travel_agency_management;
create view te as
select hotel_name,feedback_details from hotel as h
inner join feedback as f on h.feedback_id= f.feedback_id
inner join booking as b on b.hotel_id = h.hotel_id
inner join customer as c on b.customer_id = c.customer_id
where customer_name = 'Essy Bolzen';
select * from te;
```




```
No. of Tuple - 1
```

40. Show the year in which most of the customers are born.

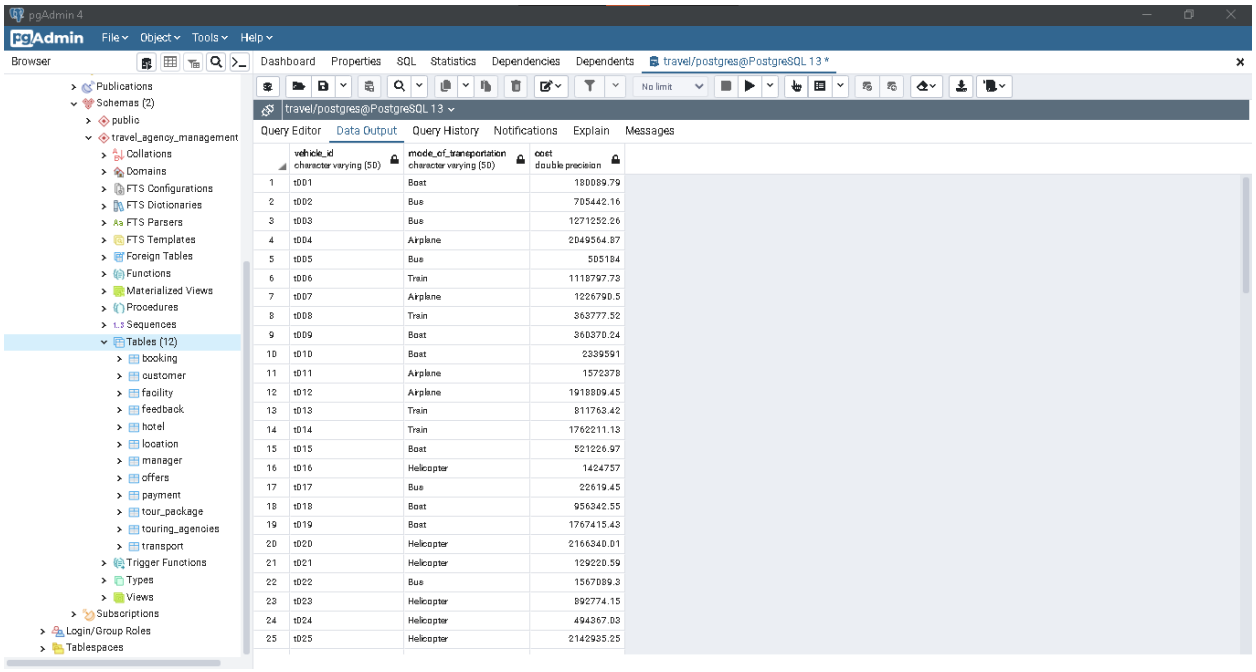
```
set search_path to travel_agency_management;
create view date as
select date_of_birth from customer;

create view customer_birth_year as
select extract (year from date_of_birth) as year from date;

select year ,count(*) as c from customer_birth_year
group by year
order by c desc
limit 1;
```




```
No. of Tuple - 1
```

41. Show the transport ID, mode of transportation, and payments received by them in a table.

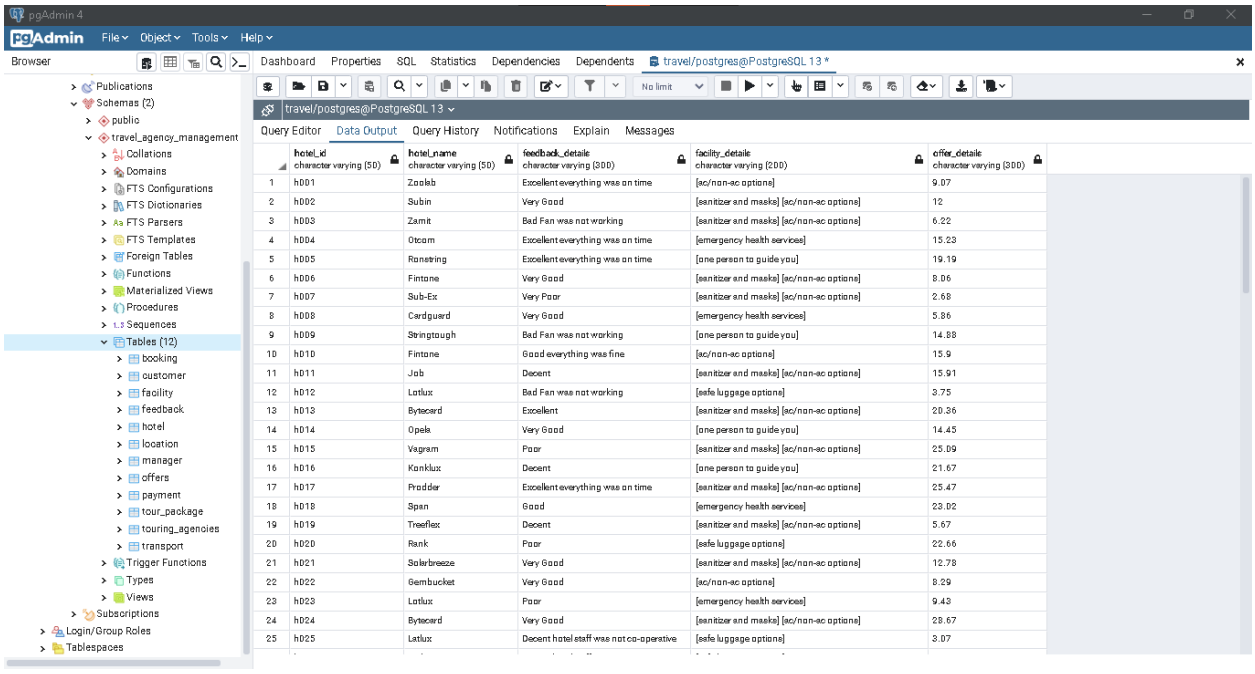
```
set search_path to travel_agency_management;
create view q42 as
select vehicle_id, mode_of_transportation, cost from transport as tr
inner join booking as b on tr.vehicle_id = b.transport_id
inner join payment as pa on b.payment_id = pa.payment_id;
select * from q42;
```




```
No. of Tuples - 100.
```

42. Show the hotel ID, hotel name, feedback details, facility details, and offer details in a table.

```
set search_path to travel_agency_management;
create view q43 as
select hotel_id, hotel_name, feedback_details,
facility_details,offer_details from hotel as h
inner join feedback as fe on fe.feedback_id = h.feedback_id
inner join facility as fa on fa.facility_id = h.facility_id
inner join offers as o on o.offer_id = h.offer_id
select * from q43;
```



	hotel_id character varying (50)	hotel_name character varying (50)	feedback_details character varying (300)	facility_details character varying (200)	offer_details character varying (300)
1	h001	Zoolab	Excellent everything was on time	[ac/non-ac options]	9.07
2	h002	Subin	Very Good	[sanitizer and masks] [ac/non-ac options]	12
3	h003	Zamit	Bad Fan was not working	[sanitizer and masks] [ac/non-ac options]	6.22
4	h004	Otcom	Excellent everything was on time	[emergency health services]	15.23
5	h005	Ronstring	Excellent everything was on time	[one person to guide you]	19.19
6	h006	Fintone	Very Good	[sanitizer and masks] [ac/non-ac options]	8.06
7	h007	Sub-Ex	Very Poor	[sanitizer and masks] [ac/non-ac options]	2.68
8	h008	Cardguard	Very Good	[emergency health services]	5.86
9	h009	Stringtough	Bad Fan was not working	[one person to guide you]	14.88
10	h010	Fintone	Good everything was fine	[ac/non-ac options]	15.9
11	h011	Job	Decent	[sanitizer and masks] [ac/non-ac options]	15.91
12	h012	Latlux	Bad Fan was not working	[safe luggage options]	3.75
13	h013	Bytecard	Excellent	[sanitizer and masks] [ac/non-ac options]	20.36
14	h014	Opela	Very Good	[one person to guide you]	14.45
15	h015	Vagram	Poor	[sanitizer and masks] [ac/non-ac options]	25.09
16	h016	Konklux	Decent	[one person to guide you]	21.67
17	h017	Prodder	Excellent everything was on time	[sanitizer and masks] [ac/non-ac options]	25.47
18	h018	Span	Good	[emergency health services]	23.02
19	h019	Treeflex	Decent	[sanitizer and masks] [ac/non-ac options]	5.67
20	h020	Rank	Poor	[safe luggage options]	22.66
21	h021	Solarbreeze	Very Good	[sanitizer and masks] [ac/non-ac options]	12.78
22	h022	Gembucket	Very Good	[ac/non-ac options]	8.29
23	h023	Latlux	Poor	[emergency health services]	9.43
24	h024	Bytecard	Very Good	[sanitizer and masks] [ac/non-ac options]	28.67
25	h025	Latlux	Decent hotel staff was not co-operative	[safe luggage options]	3.07

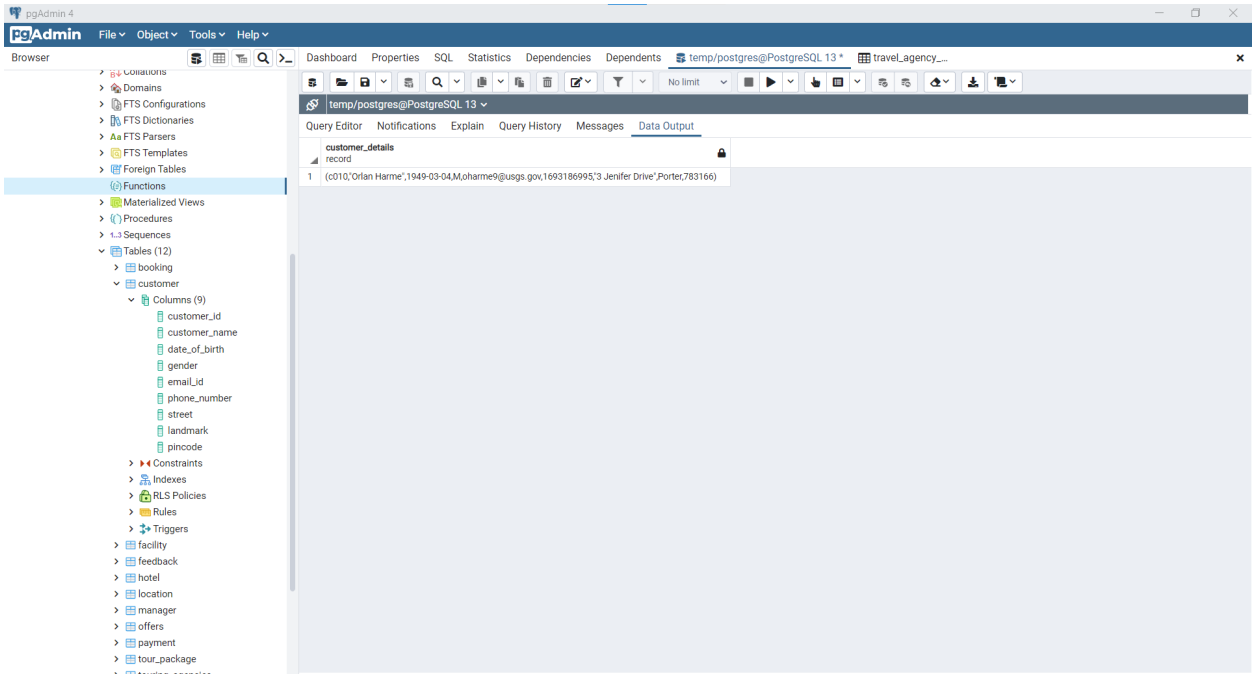
```
No. of Tuples - 100.
```

Functions:

1. A function that displays the details of a customer when called using a name.

```
SET SEARCH_PATH TO travel_agency_management;
CREATE OR REPLACE FUNCTION travel_agency_management.customer_details(IN
age_var character varying) RETURNS TABLE(a customer.customer_id%type,
b customer.customer_name%type,c customer.date_of_birth%type,d
customer.gender%type,e customer.email_id%type,f customer.phone_number%type,
g customer.street%type,h customer.landmark%type,i customer.pincode%type

LANGUAGE 'plpgsql'
VOLATILE
PARALLEL UNSAFE
COST 100 ROWS 1000
AS $BODY$
BEGIN
RETURN QUERY
SELECT * FROM travel_agency_management.customer where customer_name =
age_var;
END;
$BODY$;
```

```
SELECT "travel_agency_management"."customer_details"('Orlan Harme');
```

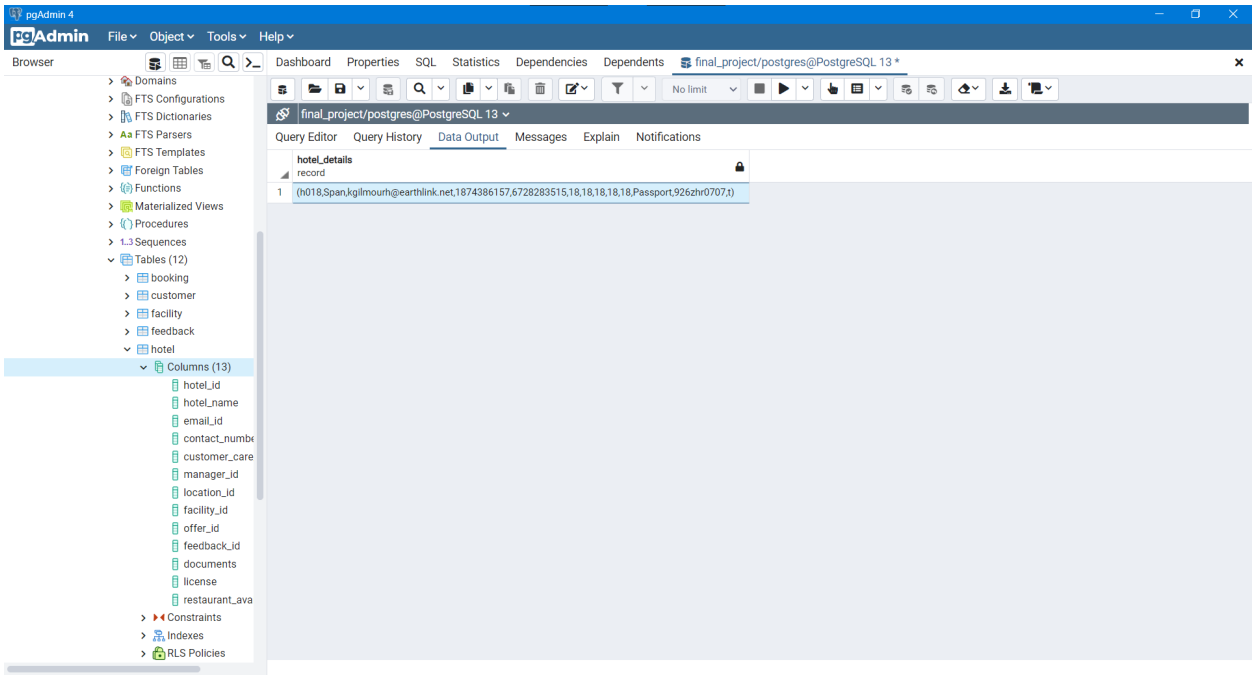


2. A function that displays the details of a hotel when called using a name.

```
SET SEARCH_PATH TO travel_agency_management;
CREATE OR REPLACE FUNCTION travel_agency_management.hotel_details(IN
hotel_var character varying) RETURNS TABLE(a hotel.hotel_id%type,
b hotel.hotel_name%type,c hotel.email_id%type,d hotel.contact_number%type,e
hotel.customer_care_number%type,f hotel.manager_id%type,
g hotel.location_id%type,h hotel.facility_id%type,i hotel.offer_id%type,j
hotel.feedback_id%type,l hotel.documents%type,n hotel.license%type,o
hotel.restaurant_available%type)

LANGUAGE 'plpgsql'
VOLATILE
PARALLEL UNSAFE
COST 100 ROWS 1000
AS $BODY$
BEGIN
RETURN QUERY
SELECT * FROM travel_agency_management.hotel where hotel_name = hotel_var;
END;
```

```
$BODY$;
```

```
SELECT "travel_agency_management"."hotel_details"('Span');
```



Triggers:

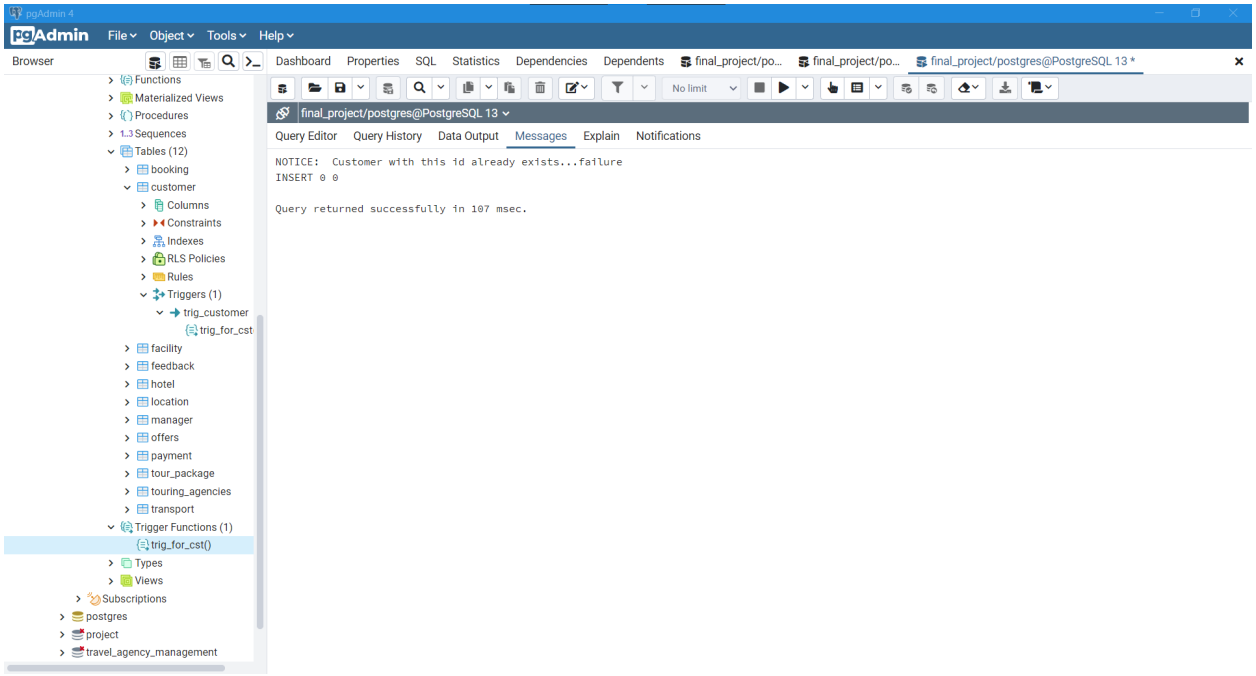
1. Create a trigger on Table customer to check if the Primary key ID already exists or not before inserting a new record. And send a custom reply instead of an error message.

Trigger Function Code:

```
Declare
e_id character varying(50);
Begin
select "customer_id" into e_id from "travel_agency_management"."customer"
where "customer_id" = new."customer_id";

if(e_id = new."customer_id") then
raise notice 'Customer with this id already exists...failure';
return NULL;
else
raise notice 'unique Customer ID... values inserted';
return new;
end if;
end
```

Trigger Code:

```
CREATE TRIGGER trig_customer
      BEFORE INSERT
      ON travel_agency_management.customer
      FOR EACH ROW
      EXECUTE FUNCTION travel_agency_management.trig_for_cst();
```



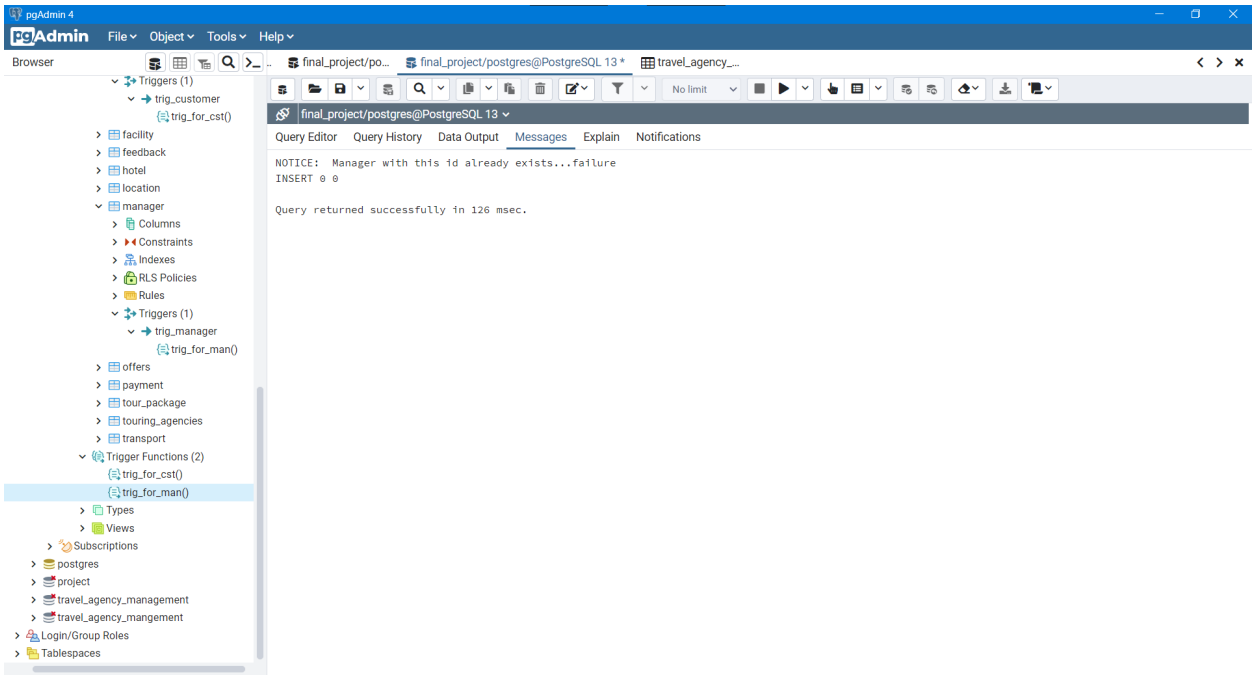
2. Create a trigger on Table manager to check if the Primary key ID already exists or not before inserting a new record. And send a custom reply instead of an error message.

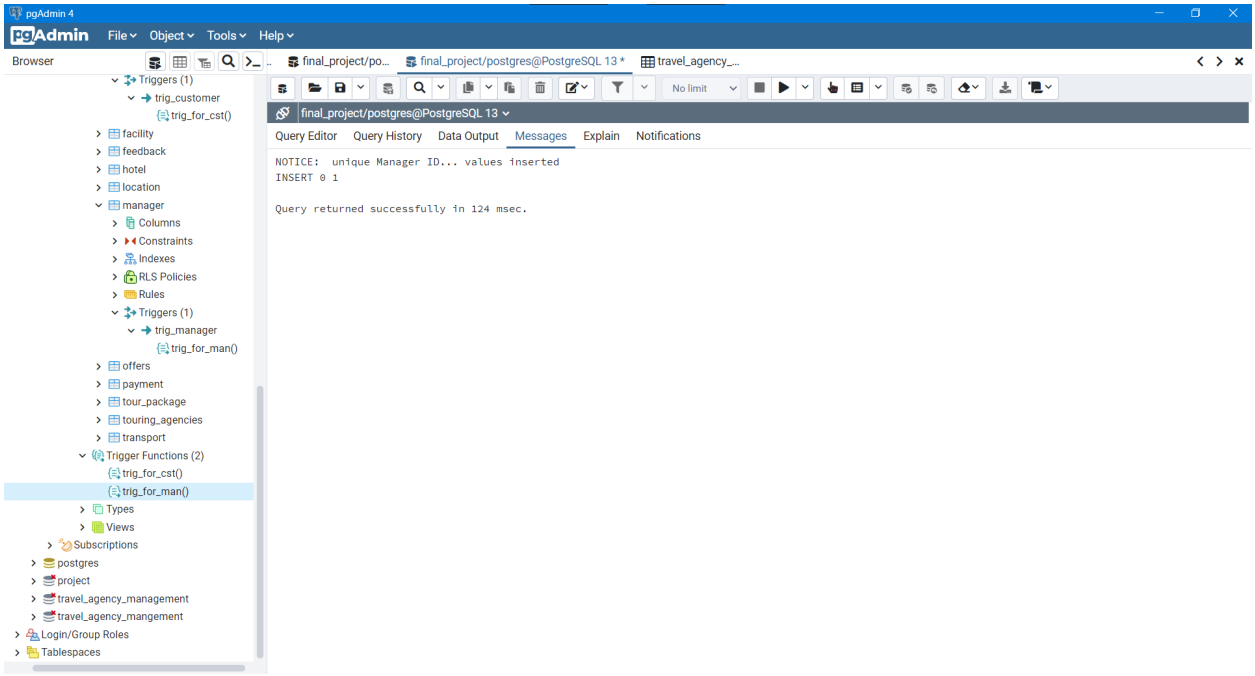
Trigger Function Code:

```
Declare
e_id bigint;
Begin
select "manager_id" into e_id from "travel_agency_management"."manager"
where "manager_id" = new."manager_id";

if(e_id = new."manager_id") then
raise notice 'Manager with this id already exists...failure';
return NULL;
else
raise notice 'unique Manager ID... values inserted';
return new;
end if;
end
```

Trigger Code:

```
CREATE TRIGGER trig_manager
      BEFORE INSERT
      ON travel_agency_management.manager
      FOR EACH ROW
      EXECUTE FUNCTION travel_agency_management.trig_for_man();
```



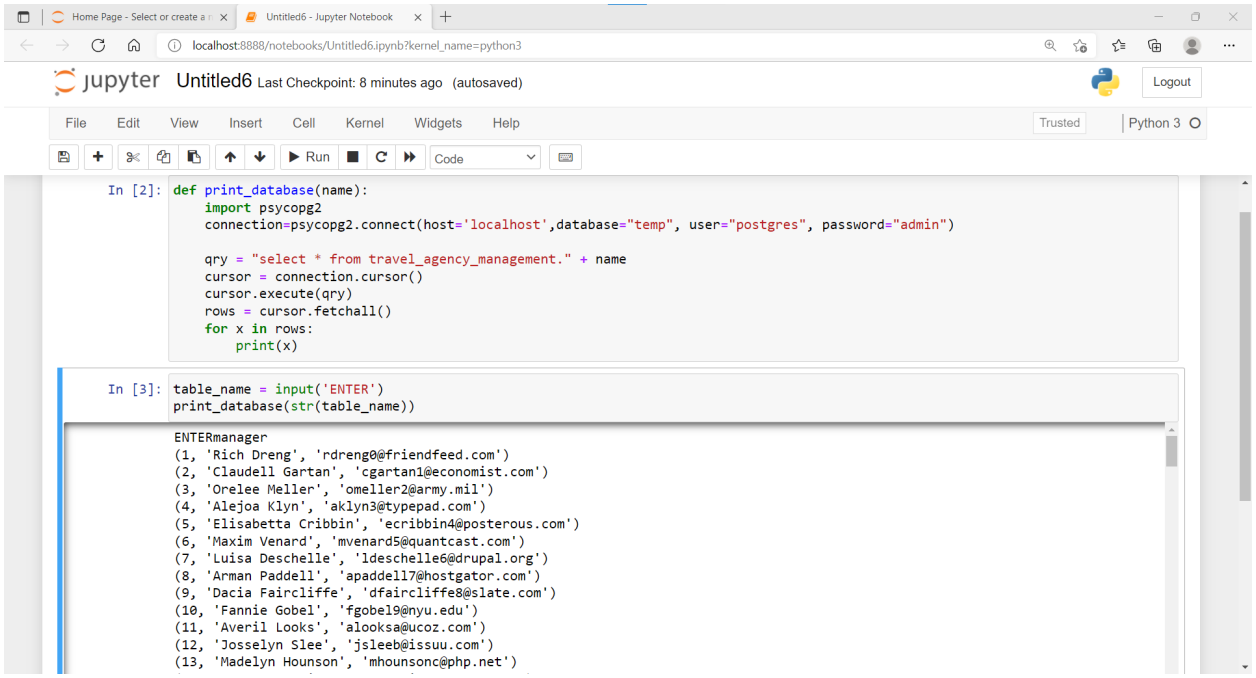

pgAdmin 4
File
Object
Tools
Help
Browser
final_project/po...
final_project/postgres@PostgreSQL 13 *
travel_agency_...
No limit
final_project/postgres@PostgreSQL 13
Query Editor
Query History
Data Output
Messages
Explain
Notifications
NOTICE: unique Manager ID... values inserted
INSERT 0 1
Query returned successfully in 124 msec.
Triggers (1)
trig_customer
trig_for_cst()
facility
feedback
hotel
location
manager
Columns
Constraints
Indexes
RLS Policies
Rules
Triggers (1)
trig_manager
trig_for_man()
offers
payment
tour_package
touring_agencies
transport
Trigger Functions (2)
trig_for_cst()
trig_for_man()
Types
Views
Subscriptions
postgres
project
travel_agency_management
travel_agency_mangement
Login/Group Roles
Tablespaces

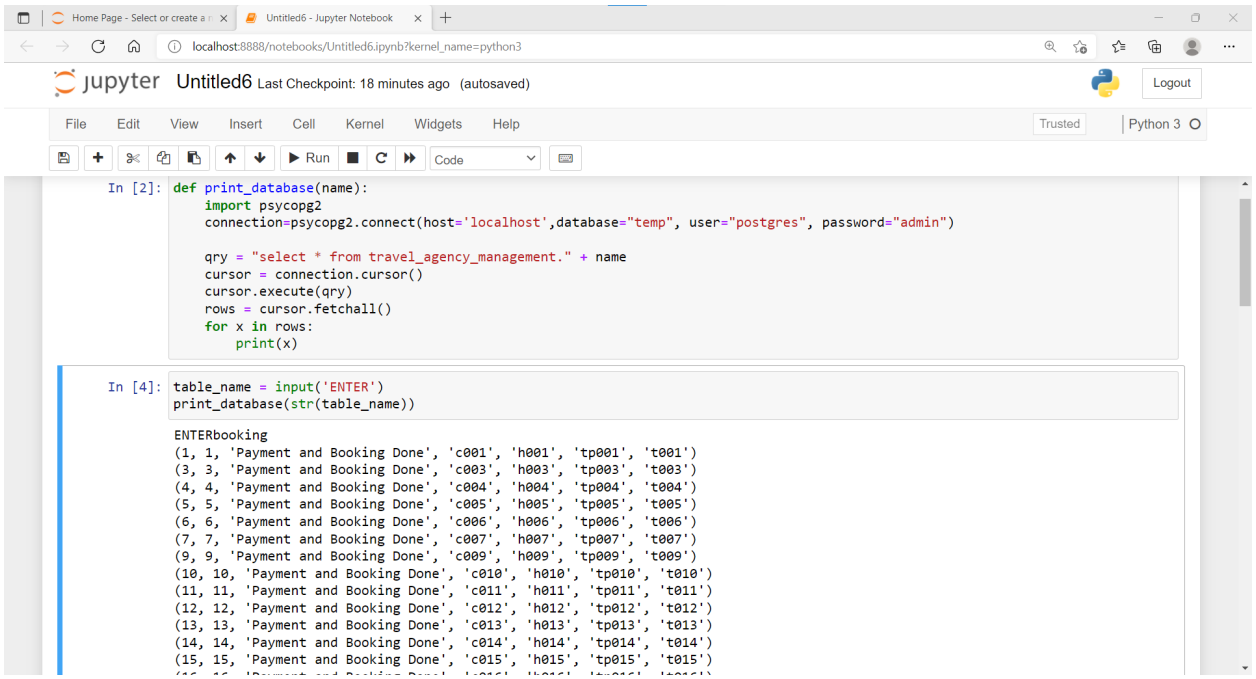
Section7: Project Code with output screenshots.

Retrieve Data Operation:

```
def print_database(name):
    import psycopg2
    connection=psycopg2.connect(host='localhost',database="temp",
user="postgres", password="admin")

    qry = "select * from travel_agency_management." + name
    cursor = connection.cursor()
    cursor.execute(qry)
    rows = cursor.fetchall()
    for x in rows:
        print(x)
```

```
table_name = input('ENTER')
print_database(str(table_name))
```

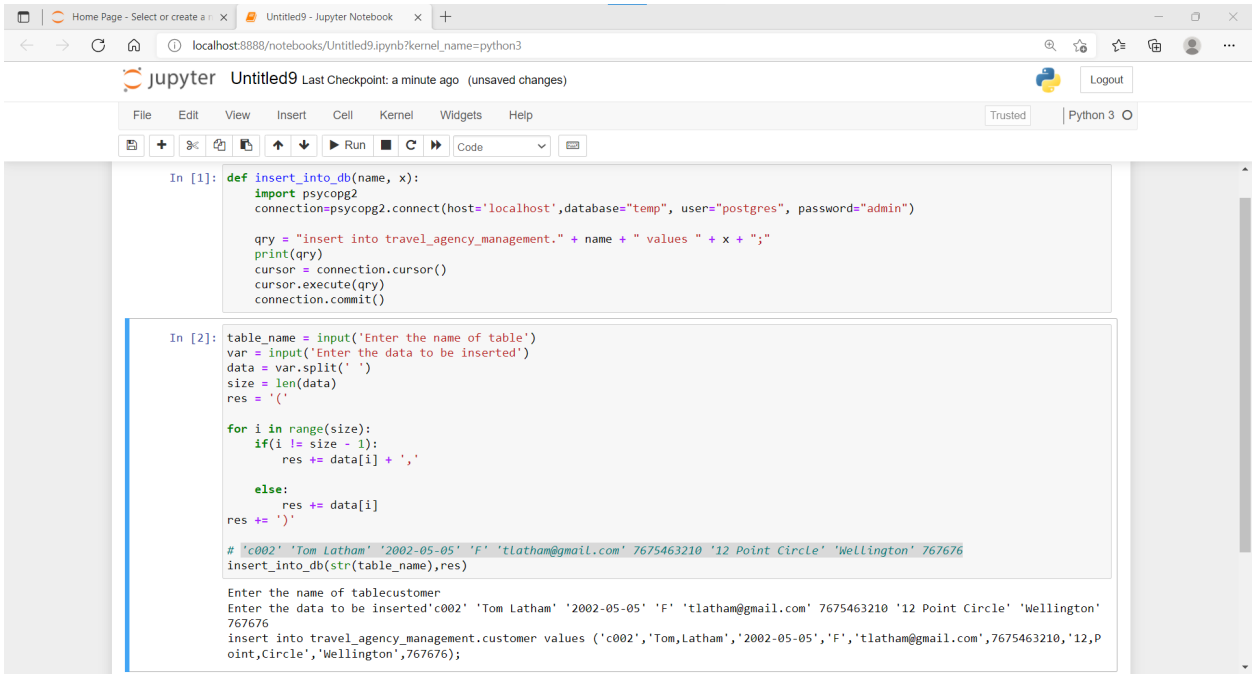

Home Page - Select or create a
Untitled6 - Jupyter Notebook
localhost:8888/notebooks/Untitled6.ipynb?kernel_name=python3
jupyter Untitled6 Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [2]:
def print_database(name):
import psycopg2
connection=psycopg2.connect(host='localhost',database="temp", user="postgres", password="admin")
qry = "select * from travel_agency_management." + name
cursor = connection.cursor()
cursor.execute(qry)
rows = cursor.fetchall()
for x in rows:
print(x)
In [3]:
table_name = input('ENTER')
print_database(str(table_name))
ENTERmanager
(1, 'Rich Dreng', 'rdreng0@friendfeed.com')
(2, 'Claudell Gartan', 'cgartan1@economist.com')
(3, 'Orelee Meller', 'omeller2@army.mil')
(4, 'Alejoa Klyn', 'aklyn3@typepad.com')
(5, 'Elisabetta Cribbin', 'ecribbin4@posterous.com')
(6, 'Maxim Venard', 'mvenard5@quantcast.com')
(7, 'Luisa Deschelle', 'ldeschelle6@drupal.org')
(8, 'Arman Paddell', 'apaddell7@hostgator.com')
(9, 'Dacia Faircliffe', 'dfaircliffe8@slate.com')
(10, 'Fannie Gobel', 'fgobel9@nyu.edu')
(11, 'Averil Looks', 'alooksa@ucoz.com')
(12, 'Josselyn Slee', 'jsleeb@issuu.com')
(13, 'Madelyn Hounson', 'mhounsonc@php.net')

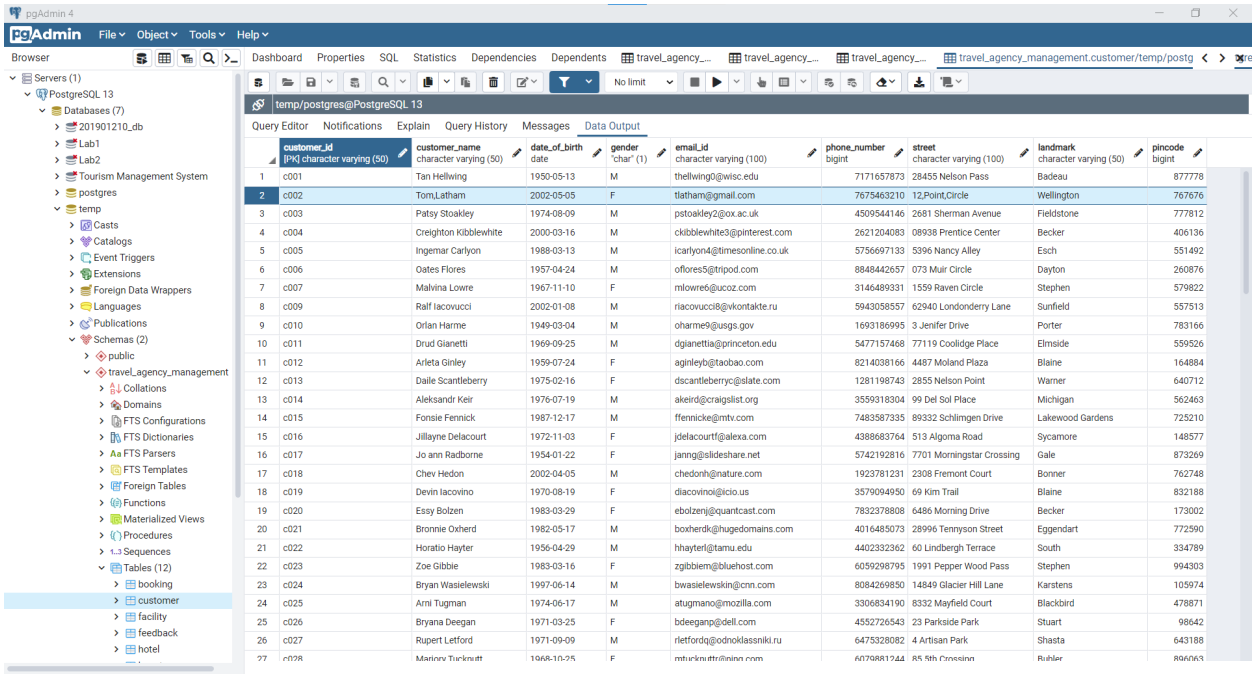

Home Page - Select or create a
Untitled6 - Jupyter Notebook
localhost:8888/notebooks/Untitled6.ipynb?kernel_name=python3
jupyter Untitled6 Last Checkpoint: 18 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [2]:
def print_database(name):
import psycopg2
connection=psycopg2.connect(host='localhost',database="temp", user="postgres", password="admin")
qry = "select * from travel_agency_management." + name
cursor = connection.cursor()
cursor.execute(qry)
rows = cursor.fetchall()
for x in rows:
print(x)
In [4]:
table_name = input('ENTER')
print_database(str(table_name))
ENTERbooking
(1, 1, 'Payment and Booking Done', 'c001', 'h001', 'tp001', 't001')
(3, 3, 'Payment and Booking Done', 'c003', 'h003', 'tp003', 't003')
(4, 4, 'Payment and Booking Done', 'c004', 'h004', 'tp004', 't004')
(5, 5, 'Payment and Booking Done', 'c005', 'h005', 'tp005', 't005')
(6, 6, 'Payment and Booking Done', 'c006', 'h006', 'tp006', 't006')
(7, 7, 'Payment and Booking Done', 'c007', 'h007', 'tp007', 't007')
(9, 9, 'Payment and Booking Done', 'c009', 'h009', 'tp009', 't009')
(10, 10, 'Payment and Booking Done', 'c010', 'h010', 'tp010', 't010')
(11, 11, 'Payment and Booking Done', 'c011', 'h011', 'tp011', 't011')
(12, 12, 'Payment and Booking Done', 'c012', 'h012', 'tp012', 't012')
(13, 13, 'Payment and Booking Done', 'c013', 'h013', 'tp013', 't013')
(14, 14, 'Payment and Booking Done', 'c014', 'h014', 'tp014', 't014')
(15, 15, 'Payment and Booking Done', 'c015', 'h015', 'tp015', 't015')

Inserting in the database:

```
def insert_into_db(name, x):
    import psycopg2
    connection=psycopg2.connect(host='localhost',database="temp",
user="postgres", password="admin")

    qry = "insert into travel_agency_management." + name + " values "
+ x + ";"
    print(qry)
    cursor = connection.cursor()
    cursor.execute(qry)
    connection.commit()
```

```
table_name = input('Enter the name of table')
var = input('Enter the data to be inserted')
data = var.split(' ')
size = len(data)
res = '('

for i in range(size):
    if(i != size - 1):
        res += data[i] + ','

    else:
        res += data[i]
res += ')'

# 'c002' 'Tom Latham' '2002-05-05' 'F' 'tlatham@gmail.com' 7675463210
'12 Point Circle' 'Wellington' 767676
insert_into_db(str(table_name),res)
```


Home Page - Select or create a
Untitled9 - Jupyter Notebook
localhost:8888/notebooks/Untitled9.ipynb?kernel_name=python3
jupyter Untitled9 Last Checkpoint: a minute ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [1]:
def insert_into_db(name, x):
import psycopg2
connection=psycopg2.connect(host='localhost',database="temp", user="postgres", password="admin")
qry = "insert into travel_agency_management." + name + " values " + x + ";"
print(qry)
cursor = connection.cursor()
cursor.execute(qry)
connection.commit()
In [2]:
table_name = input('Enter the name of table')
var = input('Enter the data to be inserted')
data = var.split(' ')
size = len(data)
res = '('
for i in range(size):
if(i != size - 1):
res += data[i] + ','
else:
res += data[i]
res += ')'
# 'c002' 'Tom Latham' '2002-05-05' 'F' 'tlatham@gmail.com' 7675463210 '12 Point Circle' 'Wellington' 767676
insert_into_db(str(table_name),res)
Enter the name of tablecustomer
Enter the data to be inserted'c002' 'Tom Latham' '2002-05-05' 'F' 'tlatham@gmail.com' 7675463210 '12 Point Circle' 'Wellington' 767676
insert into travel_agency_management.customer values ('c002','Tom,Latham','2002-05-05','F','tlatham@gmail.com',7675463210,'12,Point,Circle','Wellington',767676);


pgAdmin 4
pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents travel_agency_management.customer/temp/postg
No limit
temp/postgres@PostgreSQL 13
Query Editor Notifications Explain Query History Messages Data Output
Servers (1)
PostgreSQL 13
Databases (7)
201901210_db
Lab1
Lab2
Tourism Management System
postgres
temp
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (2)
public
travel_agency_management
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (12)
booking
customer
facility
feedback
hotel
customer_id [PK] character varying (50)
customer_name character varying (50)
date_of_birth date
gender "char" (1)
email_id character varying (100)
phone_number bigint
street character varying (100)
landmark character varying (50)
pincode bigint
1 c001 Tan Hellwing 1950-05-13 M thellwing0@wisc.edu 7171657873 28455 Nelson Pass Badeau 877778
2 c002 Tom,Latham 2002-05-05 F tlatham@gmail.com 7675463210 12,Point,Circle Wellington 767676
3 c003 Patsy Stoakley 1974-08-09 M pstoakley2@ox.ac.uk 4509544146 2681 Sherman Avenue Fieldstone 777812
4 c004 Creighton Kibblewhite 2000-03-16 M ckibblewhite3@pinterest.com 2621204083 08938 Prentice Center Becker 406136
5 c005 Ingemar Carlyon 1988-03-13 M icarlyon4@timesonline.co.uk 5756697133 5396 Nancy Alley Esch 551492
6 c006 Oates Flores 1957-04-24 M oflores5@tripod.com 8848442657 073 Muir Circle Dayton 260876
7 c007 Malvina Lowre 1967-11-10 F mlowre6@ucoz.com 3146489331 1559 Raven Circle Stephen 579822
8 c009 Ralf Iacovucci 2002-01-08 M riacovucci8@vkontakte.ru 5943058557 62940 Londonderry Lane Sunfield 557513
9 c010 Orlan Harme 1949-03-04 M oharme9@usgs.gov 1693186995 3 Jenifer Drive Porter 783166
10 c011 Drud Gianetti 1969-09-25 M dgianettia@princeton.edu 5477157468 77119 Coolidge Place Elmside 559526
11 c012 Arleta Ginley 1959-07-24 F aginleyb@taobao.com 8214038166 4487 Moland Plaza Blaine 164884
12 c013 Daile Scantleberry 1975-02-16 F dscantleberryc@slate.com 1281198743 2855 Nelson Point Warner 640712
13 c014 Aleksandr Keir 1976-07-19 M akeird@craigslist.org 3559318304 99 Del Sol Place Michigan 562463
14 c015 Fonsie Fennick 1987-12-17 M ffennicke@mtv.com 7483587335 89332 Schlimgen Drive Lakewood Gardens 725210
15 c016 Jillayne Delacourt 1972-11-03 F jdelacourtf@alexa.com 4388683764 513 Algoma Road Sycamore 148577
16 c017 Jo ann Radborne 1954-01-22 F janng@slideshare.net 5742192816 7701 Morningstar Crossing Gale 873269
17 c018 Chev Hedon 2002-04-05 M chedonh@nature.com 1923781231 2308 Fremont Court Bonner 762748
18 c019 Devin Iacovino 1970-08-19 F diacovinoi@icio.us 3579094950 69 Kim Trail Blaine 832188
19 c020 Essy Bolzen 1983-03-29 F ebolzenj@quantcast.com 7832378808 6486 Morning Drive Becker 173002
20 c021 Bronnie Oxherd 1982-05-17 M boxherdk@hugedomains.com 4016485073 28996 Tennyson Street Eggendart 772590
21 c022 Horatio Hayter 1956-04-29 M hhayterl@tamu.edu 4402332362 60 Lindbergh Terrace South 334789
22 c023 Zoe Gibbie 1983-03-16 F zgibbiem@bluehost.com 6059298795 1991 Pepper Wood Pass Stephen 994303
23 c024 Bryan Wasielewski 1997-06-14 M bwasielewskin@cnn.com 8084269850 14849 Glacier Hill Lane Karstens 105974
24 c025 Arni Tugman 1974-06-17 M atugmano@mozilla.com 3306834190 8332 Mayfield Court Blackbird 478871
25 c026 Bryana Deegan 1971-03-25 F bdeeganp@dell.com 4552726543 23 Parkside Park Stuart 98642
26 c027 Rupert Letford 1971-09-09 M rletfordq@odnoklassniki.ru 6475328082 4 Artisan Park Shasta 643188