DB_Project_Assignment

Name	Student Id
Ronit Jain	202001081
Nipun Shah	202001096
Faculty Name: Minal Bhise and Rachit Chhaya	
TA Mentor: Pinak Gajera	

Group No: 2

Section No: 6

Team Id: 6.1

Case Study

Beach Activity Management System:

All beach activity-related records should be created in this database. The beach-related activities like scuba diving, snorkeling, paragliding, flyboarding, windsurfing, and so on. Precautionary things like helmets, swimsuits, quality shoes, gloves, etc., for particular activities, must be appropriately managed for all customers. The other things related to taking pictures and videos are also included in this database.

Table of Contents:

1. Introduction

1.1. Purpose

Management Systems are systematic frameworks designed to manage an organization's procedures and data to improve data and processes. Management Systems can help to improve operations, manage risk, manage data, etc. Here we need to make a management system for some beaches where we need to manage beach activities.

There are multiple beach activities which include equipment, trainers, cost guards, customers, and other staff members working on the beach activities. There are different types of demands from customers and service providers which need to be managed. Databases of the equipment need to be addressed and checked that everything is working correctly as there are chances of accidents. A proper safety operation must be handled.

The primary purpose of making the database is to improve customers' experience and the beach activities and business of these activities. This would help maintain customers' safety and make their activities more adventurous and fun. It will hold all the records of customers, staff members, and the activities provided. It will help in the Improvement of services. Customers will know about the participants to decide which actions to participate in. Maintenance of equipment will help in the smooth and safe operation of activities. Service providers will learn about the lowest service participation to improve that service. It will also show the customer's activities held all around the place and their service providers and charges so they can plan accordingly.

1.2. Intended Audience and Reading Suggestions

The intended audience is the group of people for which a service or product is designed. So, as we know nowadays, the beaches are the main attraction of people, especially the young generation, because today's young age wants a different experience as an adventure. Youth and adults interested in watersports are the primary audiences for beach sports.

Many people are planning trips to visit beaches to relax, and the primary audience. In the broader scope, we see everyone is interested in any one of the beach activities.

Adventure sports like Scuba Diving, Snorkeling, Paragliding, Parasailing, Flyboarding, Windsurfing, Beach Volleyball, Jet Ski, Kite surfing, Beach Sprint rowing, Banana Ride, Speed Boat, Dolphin Exploration, FootVolley, Beach Camping, Swimming with Sharks and Sailing, etc. are primarily available for many beaches.

Reading Suggestions:

1. Beach Activities
2. Beach Notification Database User Guide
3. Beach Management

1.3. Product Scope

1.3.1. The purpose

Here we are providing the service of beach activities. It has a vast scope, and many people are attracted to these activities. These services can be improved for customers, and service providers will have more chances to earn good profits and provide safety.

Activities can be measured by how many people participate in that particular activity. And how can we improve that activity for further customers or clients? We can enhance the services for clients and customers by providing them with more safety and assurance that these activities are safer so that more people will participate. We can also maintain a reasonable price to be affordable for customers and profitable for the service providers. We can also introduce exciting prizes or offers to attract more customers.

Beach activities are prevalent and adventurous activities. They will attract more customers automatically if the prices are reasonable and the activities are safer. Principal customers for these activities will be youth who are more enthusiastic about them and ready to invest in them.

Furthermore, we can improve the quality of the services by adequately maintaining the types of equipment and bringing professional trainers and helpers who will help customers be more comfortable with the equipment and the technique and teach them how to do that activity. They will also help them prepare the errors, do’s, and don't they should follow during the training. This will help customers have a safer and more enjoyable experience, and they will also recommend this to other people who might be interested in participating in these activities.

1.3.2.Benefits and Goals

There are several benefits of this service which we are providing, which are as follows.

1. It will provide the administration with records of customers, trainers, and equipment and their conditions.
2. It will give administration reminders to maintain their equipment which is not in good condition.

3. It will help the administration improve the activity with minimum participation.
4. It will help customers select the activities suitable for them and see in which activity there is maximum participation.
5. This will also help customers to know where they can go and which service provider they need to go to for their demands.

1.4. Description

This software would provide the management of beach activities. A brief description of the functionalities of this software is as follows:

1. Create and Edit Activities:
It will allow administrators to create new activities and edit the old activities. It also provides information to the administration as to whether inactivity needs improvement or not. It will help administrators to implement the recent changes in activities that are done. It will also help them create new activities introduced in their system to bring more participation in those activities.

2. Management of Events on the beach:
This will manage any other events booked to be held on the beach. This software will help manage the event's date and time so that no two events collide on the same day. This will also contain the details of the booked customer and make sure that all the arrangements are made according to the customer’s demand.

3. Keep track of equipment and services:
This software keeps track of all equipment used in activities and the services of workers and equipment. This software will inform the administrator if any assistance is unavailable and is essential for training or customers. It will also maintain all the salary records of trainers and workers. It will also help the administrator to maintain the quality of the equipment.

4. Maintenance of the Equipment:

If any equipment is missing or damaged, it will inform the administration, and if the need for any equipment is unavailable, it also tells. If the number of customers increases or decreases, it also gives information about extra or less equipment for that activity.

5. Customers Friendly Software:

This software will be very friendly to customers because this software will track the records of all activities and this record will help the customers to know about that activity. This software helps in decision-making for customers. Customers will also be able to learn how to do this activity. They can also take the knowledge about that activity to do that activity. This software will also help them to select the most suitable action according to their choices. This will also let them know about the safety measures taken by the service providers, which will help them to select the safer activity.

6. Helps increase activity safety measures:

It enables the administrator to maintain the activity's safety by checking the equipment used for that particular activity. It will keep a record of the condition of equipment so that proper and time-to-time servicing is done on the equipment. It will also ensure appropriate safety measures are taken during the activity.

7. Keep track of profits and feedback on the activities:

This software will keep track of the expenses in providing the activity service to administrators so that they can charge the customer accordingly to make a profit in their business. It will also help them get feedback from the customers so that they can know which activity needs to be improved. They will also know which activity has maximum participation so that they can maintain that activity.

8. Regularly updating of Database:

There are instances where a planned activity needs to be postponed or canceled altogether due to rain or technical difficulty. In such cases, the administrator must inform all the customers about this and reschedule or cancel their scheduled activity.

The system allows us to edit a workout and access the customer's information for that particular activity.

9. Taking care of the images and videos of Customers:
Many customers want their pictures and videos while they enjoy the activity to keep their memory safe. This system will manage their photos and videos accordingly. It will also help administrators to manage the images and videos of their customers, which they can provide to customers.

2. Background Readings

2.1 Reading Summary:

1. Beach Management Activities:

This reading shows how the zones are created on the beach to manage the crowd. This division is based on the activities which are more popular and can have significant participation. So zones are classified into low, medium, and high-intensity recreation zones. Another way to categorize activities is the risk involved: Low, medium, and high conflicting activities.

2. Beach Safety:

In this article, we learn about the safety measures taken on the beach. We knew the beach safety flags and signs. We learn about the water animals that can come on the beach and their preventive measures. We also learned about the injuries that can happen on the beach.

3. Water Sports Safety Precautions:

In this article, we see how water sports or activities can be enjoyable and exciting, but if there are lapses in the precautions or safety measures, an injury might happen. This reading shows how to initially warm up your body so the chances of injury decrease. It also tells tips for water sporting activities. It describes the primary injuries that can occur, like sprains, strains in the neck or head area, concussions, lacerations, and Waterborne disease.

4. Beach Activities:

This document taught us how different commercial activities could be arranged on the beach. This document tells us how we can arrange promotional activities, filmings, marriage ceremonies, food services, surf schools, competitions to promote that activity, etc. All this activity brings the monitoring of the beach so that security is maintained so proper arrangements should be made accordingly. This also tells us how we have to make a system so we can arrange all the activities properly without any conflict of timings. It also has an idea about off-leash dog beaches, which would attract many people.

5. Water Sports Safety Guide:

This article shows the safety measures to be taken during water sports, rides on boats, and inland waterways. In this, we learned that there are several different water sports activities, and each activity will raise a new set of safety issues. For instance, safety requirements and concerns change depending on whether a person is water skiing on a lake or participating in offshore boat racing. The setting where the sport is being performed will determine what safety precautions must be taken.

6. Beach Activities and Games for Endless Fun:

In this article, we learned about the different types of beach-related activities which we can include in our database system. It also shows the most popular activities among the people. It also details the activities and how people can enjoy themselves on the beach.

7. Database System Concepts, by Abraham Silberschatz, Henry F. Korth, S. Sudarshan.

From this book, we understood the DBMS concepts like E-R Model, Database designing, etc.

2.2 Readings:

- Beach-Management-Activities:

Https://Www.Joondalup.Wa.Gov.Au/Wp-Content/Uploads/2018/07/2018-02-Beach-Management-Activities-Policy-1.Pdf

- Beach Safety:

Https://Www.Healthdirect.Gov.Au/Beach-Safety#:~:Text=To%20make%20sure%20you%20are,Or%20after%20a%20big%20meal.

- Water Sports Safety Precautions:

Https://Www.Patientfirst.Com/Blog/Water-Sports-Safety-Precautions

- Beach Activities

Https://Www.Slideshare.Net/Sallybayliss/Tracks-43592654v1summary-Documentcommercialactivityonbeachesfinal17march2014

- Water Sports Safety Guide

Https://Www.Realbuzz.Com/Articles-Interests/Sports-Activities/Article/Water-Sports-Safety-Guide/

- Beach Activities And Games For Endless Fun

Https://Playtivities.Com/30-Beach-Activities-And-Games-For-Endless-Fun/

- Database System Concepts, by Abraham Silberschatz, Henry F. Korth, S. Sudarshan.

2.3 List the combined Requirements gathered from Background Reading/s:

1. A well-functioning Database management system is required to maintain and update all the data related to activities, customers, and staff.
2. A user interface is also required so that customers can access all the information they need.
3. All the essential equipment and trainers associated with that activity data should be maintained.
4. Records of servicing of data should be maintained.
5. Administrators and trainers/staff will be assigned different roles to secure the customer's sensitive information.
6. The system should be designed in a way that it has fast performance.
7. All the precautionary measures should be taken during the activity to ensure customers' safety.
8. It should show administrators how to improve the commercial activities on the beach and maintain accurate data of all the activities.
9. Help administrators place all the safety boards and sign in proper places where they can reach the maximum audience.

3. Interviews

1. Interview 1:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Nishtha Jain(**Role Play**) **Designation**: Manager

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 24/9/2022 **Time:** 14:30

Duration: 30 minutes **Place:** Zoom

Purpose of Interview:

Preliminary meeting to identify how activities are organized and the problems in beach activity management.

Agenda:

1. Identify and discuss the beach activity management.
2. Identify and Discuss the problems that arise while managing activities.

Documents to be brought to the interview:

1. Identification Card
2. List of problems they have encountered

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Nishtha Jain(**Role Play**) **Designation**: Manager

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 24/9/2022 **Time:** 14:30

Duration: 30 minutes **Place:** Zoom

Purpose of Interview:

Preliminary meeting to identify how activities are organized and the problems in beach activity management.

Agenda:

1. Identify and discuss the beach activity management.
2. Identify and Discuss the problems that arise while managing activities.

Documents to be brought to the interview:

1. Identification Card
2. List of problems they have encountered

Summary:

1. From this interview, we learned how beach activities are organized and managed.
2. Several things need to be managed, like staff, trainers, equipment, etc., for the proper functioning of the activities.
3. The main task is to first manage the crowd on the beach. So that there are no problems related to crowd management.
4. Sometimes, some equipment is missing or damaged.
5. Waste management is also an essential issue on Indian beaches, as many people throw their garbage anywhere on the beach.

6. Sometimes, administrators or customers neglect the weather conditions in their excitement for that activity. This can lead to some accidents or mishappenings.
7. Medical facilities must be there; some administrators are careless about safety measures.
8. Strict rules should be implemented so that customers do not get into activities unsuitable for their age, height, or weight.

2. Interview 2:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Ronit Jain(**Role Play**) **Designation**: Administrator

Interviewer: 1) Nipun Shah **Designation**: Software Engineer

Date: 28/9/2022 **Time:** 16:00

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Preliminary meeting to identify the activities and equipment used in those activities.

Agenda:

1. List of Activities performed
2. Identify, and discuss the equipment used in the Activities

Documents to be brought to the interview:

1. Identification Card
2. List of activities
3. List of equipment for each activity

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Ronit Jain (**Role Play**) **Designation**: Administrator

Interviewer: 1) Nipun Shah **Designation**: Software Engineer

Date: 28/9/2022 **Time:** 16:00

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Preliminary meeting to identify the activities and equipment used in those activities.

Agenda:

1. List of Activities performed
2. Identify, and discuss the equipment used in the Activities

Documents to be brought to the interview:

1. Identification Card
2. List of activities
3. List of equipment for each activity

Summary:

1. In this interview, we learned about the activities organized on the beaches and the equipment required.
2. The information about famous water sports and water activities.
3. After the interview, we learned that the attractive visibility of activity plays an essential role in publicity.
4. We can increase the number of customers of the activity through efficient implementation and well-structured equipment
5. Some activities like scuba diving need oxygen cylinders, Buoyancy Control Devices (BCD) from Aqualung, fins, etc.
6. The most used equipment is the life jacket used in almost all activities.
7. We learned that the same equipment is used for some activities, and for some activities, different types of equipment are used to provide service.

8. We also learned how the different activities are managed and how all the equipment is managed.

3. Interview 3:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Nipun Shah (**Role Play**) **Designation**: Trainer

Interviewer: 1) Ronit Jain **Designation**: Designer

Date: 28/9/2022 **Time:** 20:00

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Preliminary meeting to understand the problems faced by the trainers during the activity.

Agenda:

1. Identify and discuss the problems faced by the trainers.

Documents to be brought to the interview:

1. Identification Card
2. Trainer Certification
3. List of problems they face

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Nipun Shah (**Role Play**) **Designation**: Trainer

Interviewer: 1) Ronit Jain **Designation**: Designer

Date: 28/9/2022 **Time:** 20:00

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Preliminary meeting to understand the problems faced by the trainers during the activity.

Agenda:

1. Identify and discuss the problems faced by the trainers.

Documents to be brought to the interview:

1. Identification Card
2. Trainer Certification
3. List of problems they face

Summary:

1. In this interview, we learned about the problems faced by the trainers.
2. They have issues regarding the equipment if they are not adequately maintained; they still have to adjust to ensure that their and the customer's lives don't get risked.
3. They have to adjust according to the customer's coordination level.
4. They must always be on alert during the activity and handle the customer properly.
5. They also face problems with the administration on the quality of the equipment that is inappropriate to use. Still, they are forced to use these because of pressure from the administrator.

4. Interview 4:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Rohit Rao (**Role Play**) **Designation**: Safety Operations Incharge

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 29/9/2022 **Time:** noon

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Preliminary meeting to understand the safety operations that are taken care off while managing the activities and during the activities.

Agenda:

1. Identify and discuss safety operations.
2. Problems that arise during safety operations.

Documents to be brought to the interview:

1. Identification Card
2. Certificate to act as Safety Operations Incharge
3. List of measures they take to manage safety operations
4. List of problems they face in managing the safety and security of people visiting beaches.

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Rohit Rao (**Role Play**) **Designation**: Safety Operations Incharge

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 29/9/2022 **Time:** noon

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Preliminary meeting to understand the safety operations that are taken care off while managing the activities and during the activities.

Agenda:

1. Identify and discuss safety operations.
2. Problems that arise during safety operations.

Documents to be brought to the interview:

1. Identification Card
2. Certificate to act as Safety Operations Incharge
3. List of measures they take to manage safety operations
4. List of problems they face in managing the safety and security of people visiting beaches.

Summary:

1. After the interview, we learned what safety operations are that are done or need to be done in some conditions.
2. We also got to know the problem which arises during safety operations.
3. Information about the equipment which is specially used in safety operations.
4. After the interview, we learned what lifeguards and lifesavers are and about the notation and flag placed on beaches. E.g., **no flag=no swim,** etc.
5. First aid kit is mandatory.

6. The essential people in safety operations are lifeguards, who must always be alert on the beach and take proper action if there is any mishap.

5. Interview 5:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Rohan Reddy (**Role Play**) **Designation**: Deputy Administrator

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 29/9/2022 **Time:** 13:00

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Meeting to understand the problems arise during the listing of activities and management of equipment.

Agenda:

1. Identify and discuss the problems in managing the activities and listing them in front of customers.
2. Identify and discuss the problems in managing the equipment related to each activity.

Documents to be brought to the interview:

1. Identification Card.
2. List of problems they face in managing the Activities.
3. List of problems they face in managing the Equipment.

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Rohan Reddy (**Role Play**) **Designation**: Deputy Administrator

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 29/9/2022 **Time:** 13:00

Duration: 30 minutes **Place:** Cafeteria

Purpose of Interview:

Meeting to understand the problems arise during the listing of activities and management of equipment.

Agenda:

1. Identify and discuss the problems in managing the activities and listing them in front of customers.
2. Identify and discuss the problems in managing the equipment related to each activity.

Documents to be brought to the interview:

1. Identification Card.
2. List of problems they face in managing the Activities.
3. List of problems they face in managing the Equipment.

Summary:

1. In this interview, we learned about the problems faced in managing the activities and equipment.
2. All the list of activities and trainers associated with them and the equipment associated with them are to be maintained.
3. Attendance of all the staff members and trainers is to be maintained, and their work is to be divided so that there is no problem and everything is worked out.
4. Equipment needs to be appropriately maintained, and servicing should be done from time to time so that they are in good condition.

5. Regular equipment maintenance will help increase the activity's safety, attracting more customers.
6. Cost of maintenance and equipment needs to be managed accordingly, so there is no loss and a good profit.

6. Interview 6:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Sanyam Jain (**Role Play**) **Designation**: Event Manager

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 30/9/2022 **Time:** 10:00

Duration: 30 minutes **Place:** Zoom

Purpose of Interview:

Preliminary meeting to understand about the event management and the production work that is done for an event.

Agenda:

1. Talk on event management
2. Production
3. Guest management

Documents to be brought to the interview:

1. Identification Card
2. List of events
3. List of equipment they require in production
4. Problems they face during an event

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Sanyam Jain (**Role Play**) **Designation**: Event Manager

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 30/9/2022 **Time:** 10:00

Duration: 30 minutes **Place:** Zoom

Purpose of Interview:

Preliminary meeting to understand about the event management and the production work that is done for an event.

Agenda:

1. Talk on event management
2. Production
3. Guest management

Documents to be brought to the interview:

1. Identification Card
2. List of events
3. List of equipment they require in production
4. Problems they face during an event

Summary:

1. In this interview, we learned how the events are managed on the beaches.
2. Events on beaches are pre-booked, and all the work is done according to the event.
3. Production work includes decoration, setting up of the stage (if required), chair setup, etc.
4. Event management has the very challenging task of managing every small requirement demanded by the customer.

5. There are many things that come on the spot and need to be arranged at that point of time only.
6. This work requires a more proper arrangement so that the minimum number of works comes up at the last point.

7. Interview 7:

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Dev Patel (**Role Play**) **Designation**: Customer

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 30/9/2022 **Time:** noon

Duration: 30 minutes **Place:** Room

Purpose of Interview:

Preliminary meeting to understand about the requirements and problems faced by the customers.

Agenda:

1. Requirements of customers
2. Customer's thoughts about the activities
3. Problems faced by the customer's

Documents to be brought to the interview:

1. Identification Card
2. List of requirements
3. List of problems

Beach Activity Management System: Mock Interview Summary

System: Beach Activity Management System

Project Reference: SF/SJ/2022/12

Interviewee: 1) Dev Patel (**Role Play**) **Designation**: Customer

Interviewer: 1) Ronit Jain **Designation**: Designer
2) Nipun Shah **Designation**: Software Designer

Date: 30/9/2022 **Time:** noon

Duration: 30 minutes **Place:** Room

Purpose of Interview:

Preliminary meeting to understand about the requirements and problems faced by the customers.

Agenda:

1. Requirements of customers
2. Customer's thoughts about the activities
3. Problems faced by the customer's

Documents to be brought to the interview:

1. Identification Card
2. List of requirements
3. List of problems

Summary:

1. This interview taught us about customers' perspectives and what they want from the service-providing companies.
2. We learned that customers want a platform to compare and select all their activities.
3. They also want that system to suggest an activity according to their need and experience.
4. They also feel that safety should be improved and risk should be minimal during an activity.

3.1 List the combined Requirements gathered from Interviews:

1. From the interviews we conducted, we learned about many aspects of the beach activities and their management.
2. We noted that safety is the most important thing which should be maintained properly and needs a lot of improvement.
3. All the equipment should be replaced with new ones and be maintained by properly servicing the equipment regularly.
4. We also see how many issues need to be addressed.
5. A system is needed to show and maintain all the data available to the administrators.
6. The system should show that no two events are booked on the same day or at the same time.
7. Systems also need to manage the data related to staff and trainers so that all the duties are fulfilled.
8. System should also manage the safety operation properly and maintain data for the first aid kits, ambulance, coast guards, lifeguards, etc.
9. It should also maintain the list of available equipment in the inventory and their condition. It should create a new list of equipment which are needed to buy.
10. It should also maintain the expenses and profit because, after all the expenses, it should be a customer-friendly profitable system.
11. It should also maintain a system where customers can see all the data related to their requirements and get suggestions for the best activity according to their requirements.

4. Questionnaire/s

1. Your Age *(Select an appropriate option)*

- ○ 11 - 18
- ○ 18 - 25
- ○ 25 - 35
- ○ 35+

2. How frequently do you visit Beach? *(Select an appropriate option)*

- ○ Weekly (1 or 2 times a Week)
- ○ Monthly (1 or 2 times a Month)
- ○ Yearly (1 or 2 times a year)
- ○ Rarely

3. How much are you interested in doing water sports on Beach? *(Circle a suitable option)*

1 / 2 / 3 / 4 / 5

4. Which are the activities you are most interested in? *(Choose appropriate options)*

- ☐ Scuba Diving
- ☐ Snorkeling
- ☐ Paragliding
- ☐ Parasailing
- ☐ Flyboarding
- ☐ Windsurfing
- ☐ Beach Volleyball
- ☐ Jet Ski
- ☐ Kite surfing
- ☐ Beach Sprint Rowing
- ☐ Banana Ride
- ☐ Speed Boat
- ☐ Dolphin Exploration

☐ Foot Volley

☐ Beach Camping

☐ Swimming with Sharks

Other: …….

5. Are you interested in getting professional training and doing an activity on your own?

○ Yes

○ No

○ Maybe

6. Do you want a Trainer?

○ Yes

○ No

○ Maybe

7. How satisfied are you with the safety measures followed till now during the activities?

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

8. How much are you worried about your safety during the activity?

1 / 2 / 3 / 4 / 5

9. How much are you satisfied with the equipment used during the water activity?

1 / 2 / 3 / 4 / 5

10. Do you want locker facilities to keep your belongings?

○ Yes

○ No

○ Maybe

11. Your views on the safety precautions taken and equipment used during the water activities.

……
……
……

12. Are you interested in purchasing your photos and videos captured during the activity?

- ○ Yes
- ○ No
- ○ Maybe

13. Are you interested in organizing any of your events on the beach?

- ○ Yes
- ○ No
- ○ Maybe

14. Any Suggestions for improvement?

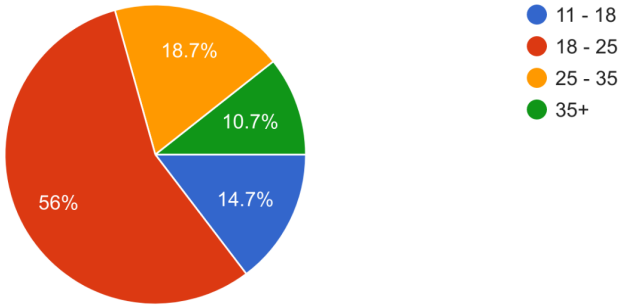
……
……

Google Form Link: https://forms.gle/UnsMBmwhaSTWVgPw9

4.1Summary:

Q1: Your Age

75 responses

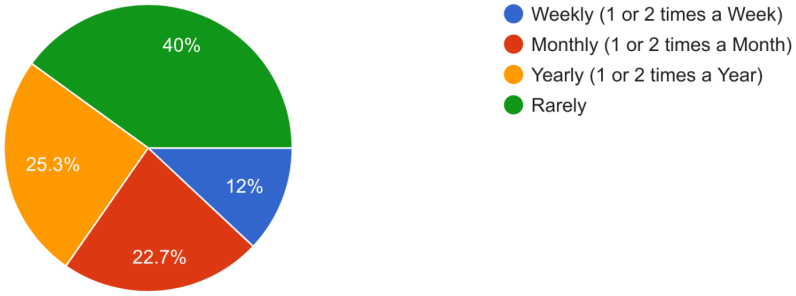


Intention: Through this question, we wanted to know the age range of people more likely to participate in beach activities.

Result: We can see that maximum participation is from ages 18 - 25. This means that youth are more interested in beach activities, and full participation will be from their side.

Q2: How frequently do you visit Beach?

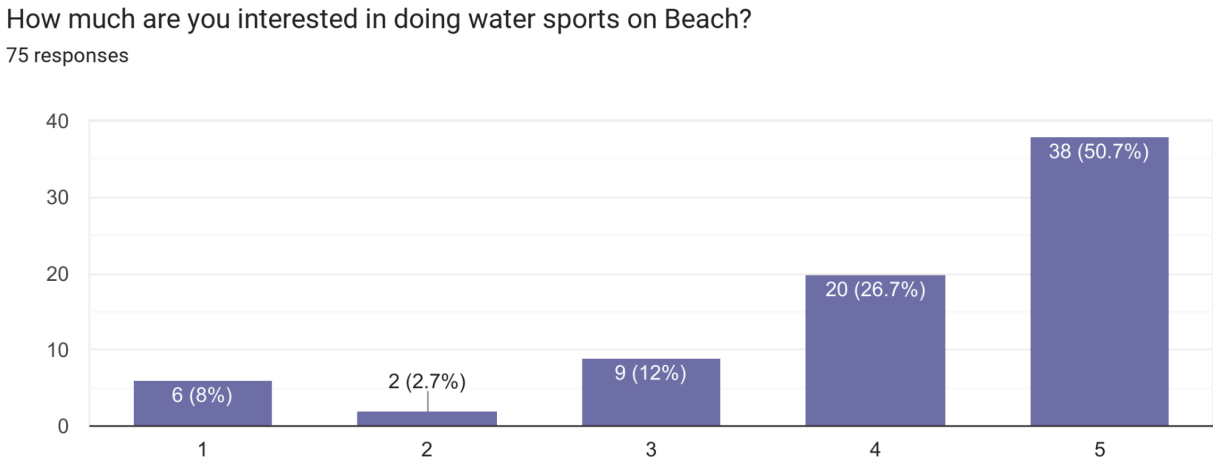
75 responses



Intention: From this question, we will see how frequently people visit beaches.

Result: From the responses, we can see that the maximum number of people rarely or in a year once or twice visit the beach.

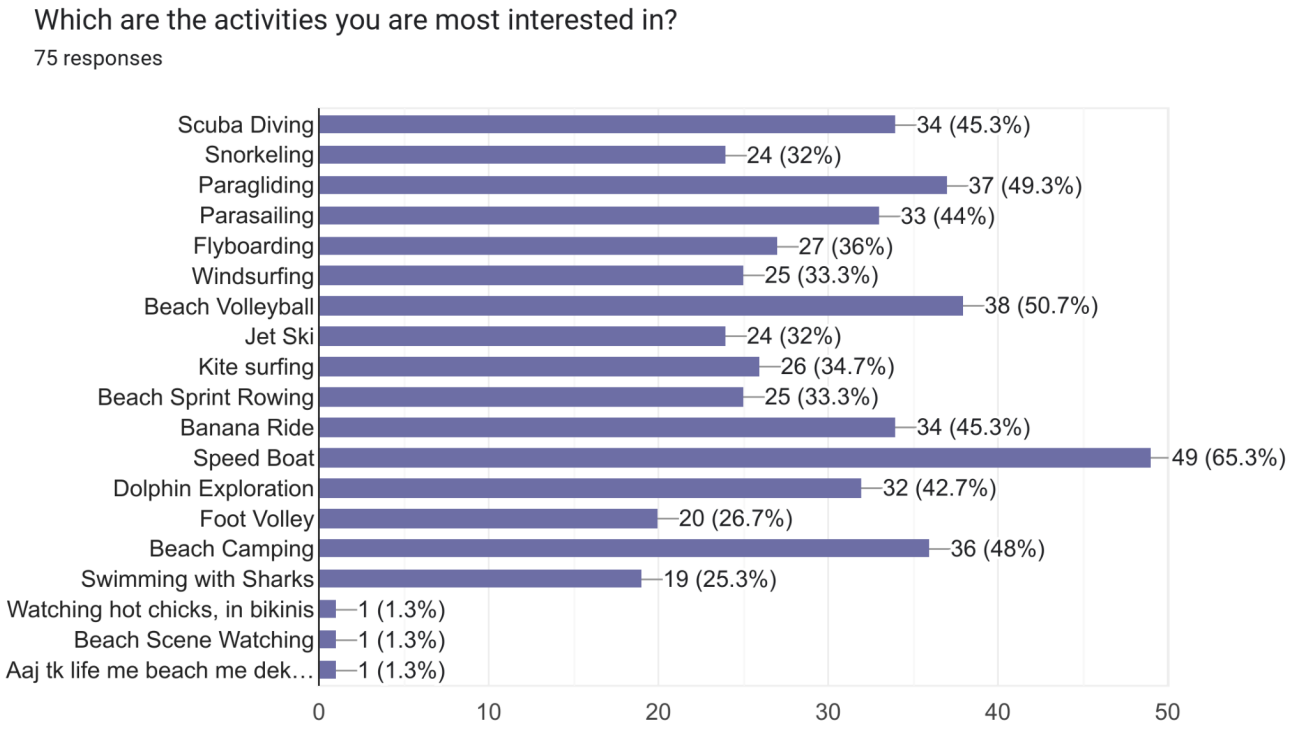
Q3:



Intention: We will see how many people are interested in watersports from this question.

Result: Most people are interested in water sports, and approximately 50 % are very interested in water sports.

Q4:

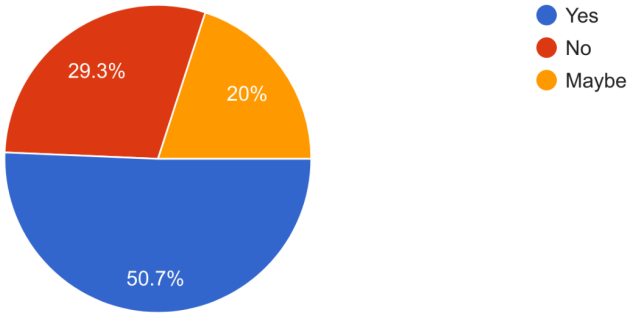


Intention: We see which beach activity maximum people are interested in from this question.

Result: Most people are interested in speed boats, beach camping, and scuba diving.

Q5: Are you interested in getting professional training and doing an activity on your own?

75 responses

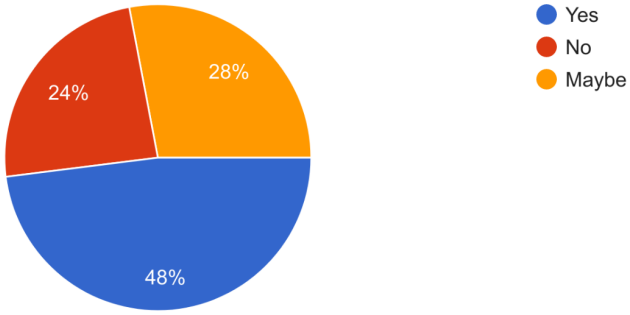


Intention: From this question, we will see how many people know about doing that activity so that we can arrange the trainer for them for that particular activity.

Result: Approximately 50% need a trainer, and approximately 36% of people can do activities independently.

Q6: Do you want a Trainer?

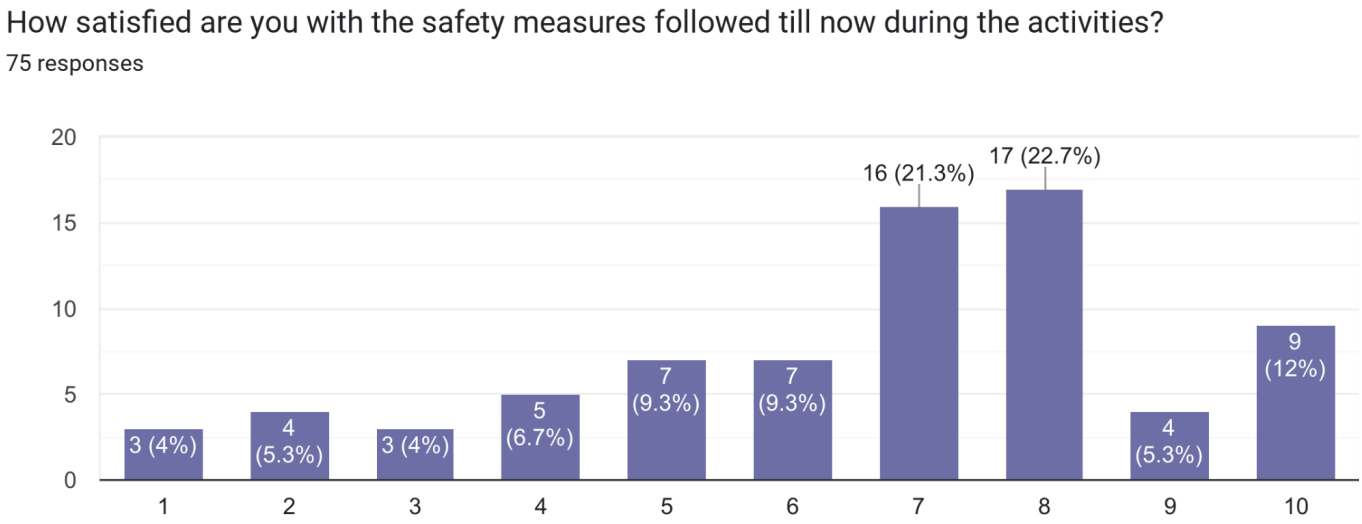
75 responses



Intention: How many people or customers do we need a trainer?

Result: As we collect the data from the above question, we can see that approximately 47% need a trainer and 30% don't.

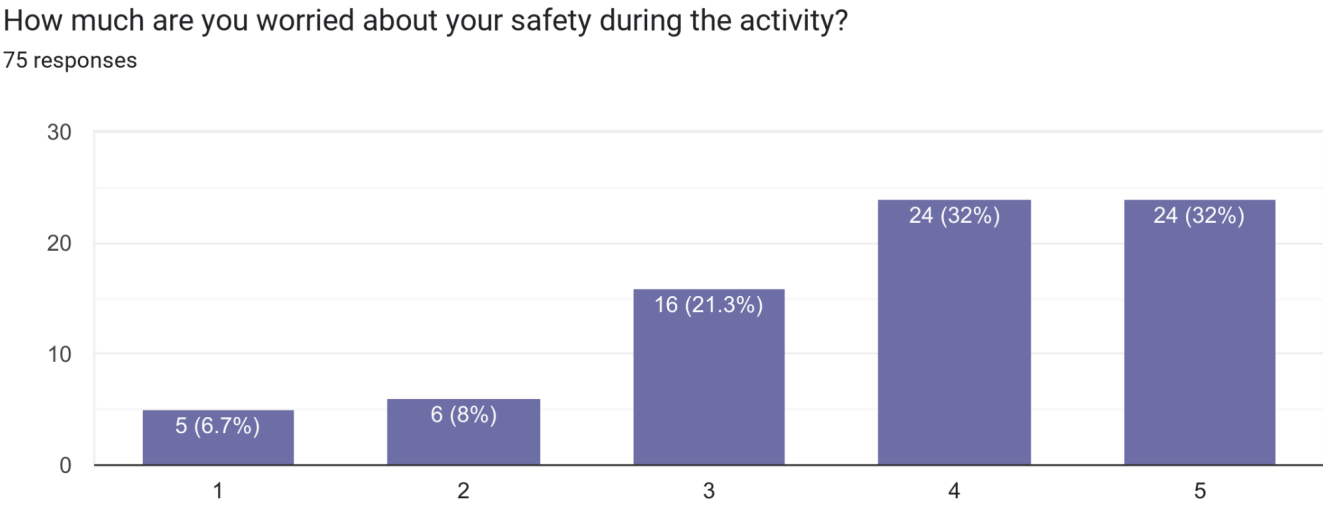
Q7:



Intention: From this question, we saw how many people are satisfied with the safety measures taken during the activities.

Result: We can see a mixed response in satisfaction with the safety measures. Maximum people have rated 7 out of 10, but some have also given less than 5.

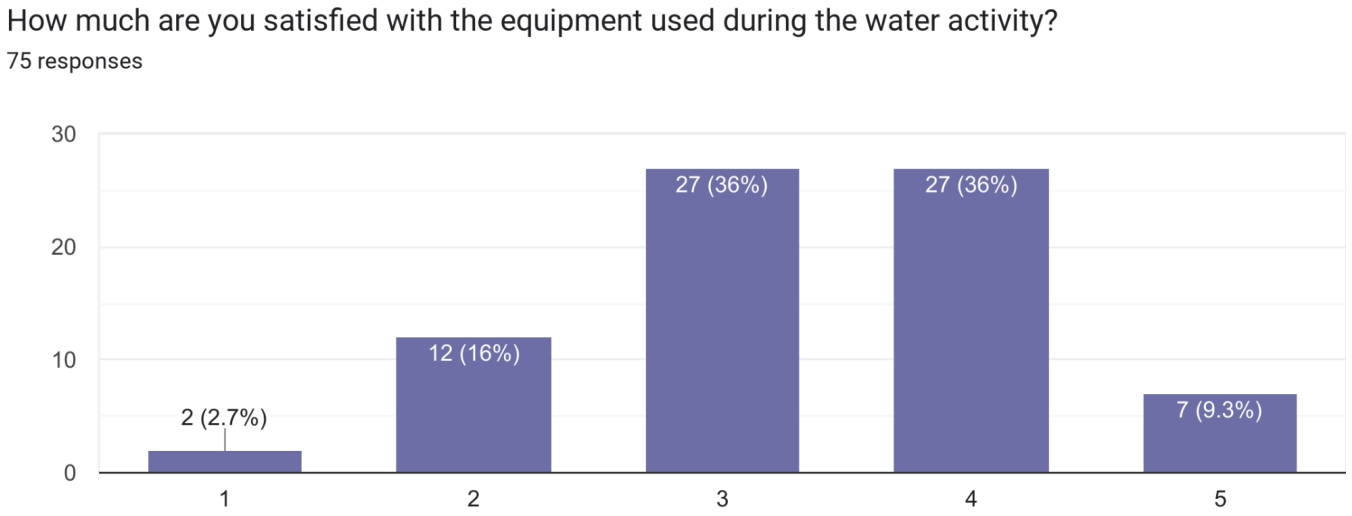
Q8:



Intention: From this question, we are trying to determine how much people worry about their safety during the activity.

Result: We can see from the results that around 61% of people have given ratings of 4 or 5 out of 5, which tells us that many people are worried about their safety during the rides.

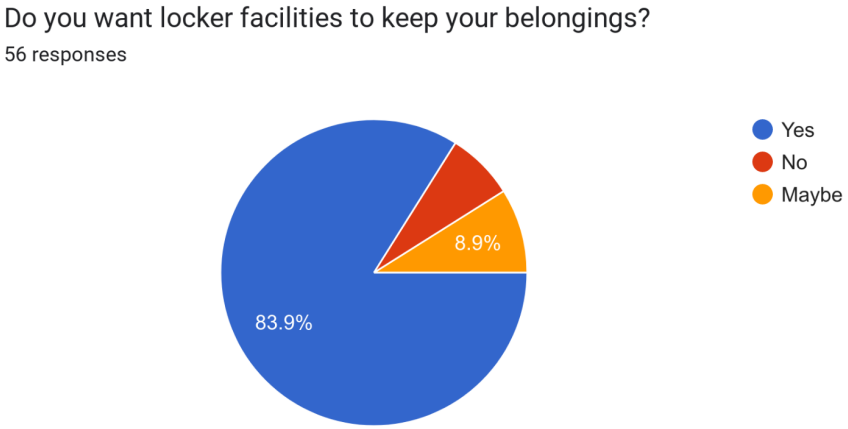
Q9:



Intention: From this question, we are trying to determine how much people are satisfied with the equipment used during the water activity.

Result: We can see from the results that around 70% of the people have voted 3 and 4 points out of 5. So people are not completely satisfied with the equipment used in the activities. This shows that there are still chances of improvement in the quality of the equipment.

Q10:



Intention: From this question, we see whether administrators need to arrange the locker for the people.

Result: From the results, we can see that 82.4% of people are interested in keeping a locker for their personal belongings.

Q11:

Your views on the safety precautions taken and equipment used during the water activities.

7 responses

Very bad , condoms must be of higher qualities
Great work
Regular inspection of equipments dont take placd
Ok
💩 💩 💩
They should maintain the safety measures and regularly check the equipments
Need to be more aware and careful for equipment

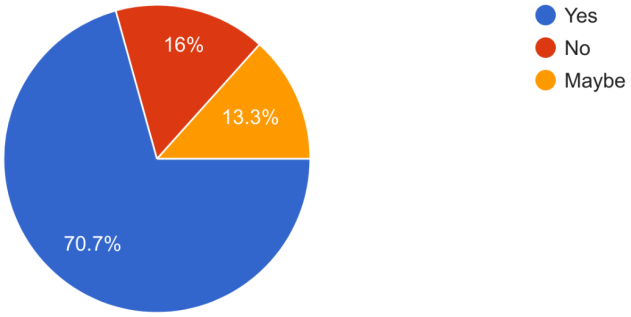
Intention: From this question, we intended to get people's views on the safety precautions and equipment used during water activities.

Result: We see people want administrators to be more careful and aware of the equipment used. People wish for a regular inspection of the equipment.

Q12:

Are you interested in purchasing your photos and videos captured during the activity?

75 responses

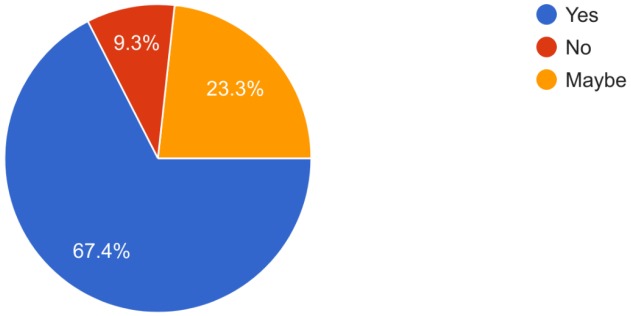


Intention: We see how many people are interested in getting their photos and videos from this question.

Result: From the above responses, we can see that approx 70% of people are interested in their pictures and videos.

Q13: Are you interested in organizing any of your events on beach?

43 responses



Intention: Some major events need more space so we can arrange the customers of other activities and the event management.

Result: We see that 65% of people are interested in organizing their events or special beach days.

Q14: Any Suggestions for improvement?

8 responses

- No
- Nude beaches would be a great idea , that way we can do a little cardio
- Women Safety& worried about Water safety
- Appropriate settings for relaxing should be made at the time of sunset & sunrise.

Intention: From this question, we wanted to know any other suggestions for improvement in the services.

Result: We can see concerns about women's safety, so that should be taken care of. We can also see people want facilities for relaxing during the sunset & sunrise.

4.1 List the combined Requirements gathered from Questionnaire:

1. From this questionnaire, we learned about customers' views on a particular set of questions.
2. We see that people aged 18 - 25 years are more interested in participating in the activities on the beach.
3. We can see that most people are interested in participating in water activities.
4. We see that people are worried about their safety during the activities, so that needs to be improved.
5. Many people are interested in having lockers for keeping their belongings safe on the beach. So there should be a management system where all the belongings are safe in the lockers.
6. Many people are also interested in arranging events on beaches, so there should be a system that helps manage those events.

5. Observation/s

Beach Activity Management System: Mock Interview Plan

System: Beach Activity Management System

Project Reference: SF/SJ/2022/Obv1

Observations by: Ronit Jain

Date: 30/9/2022 **Time:** 15:30

Duration: 45 minutes **Place:** Beach Activity Center

Observations:

1. No proper system was there to maintain the data of customers, hand made slips were distributed.
2. Information about the customers needed to be stored better.
3. Equipment was correctly stored so that no damage was there.
4. Trainers were well trained to perform all the activities available there. They were beneficial to customers.
5. Safety equipment was maintained well.
6. Many sorts of activities were available there.
7. Payment of all sorts was accepted for the convenience of customers.
8. All the security measures were implemented correctly.
9. Height and age restrictions in some activities were managed by asking customers their ages.
10. Customers were managed very smoothly.
11. Events on the beach were correctly organized, but improvements could have occurred.
12. Beach cleanliness was not appropriate. It should be appropriately managed. People should be careful about their garbage.

5.1 List the combined Requirements gathered from Observations:

1. We see that digitized systems should be used rather than handwritten slips.
2. Data should be managed from the equipment used during the activity so that all the equipment is maintained correctly and serviced regularly.
3. All the safety measures should be followed correctly, and there should be no lapses in the safety measures.
4. Safety drills should be conducted regularly so that staff members are adequately prepared for any situation.
5. Customer-friendly service should be provided to be happy with the service and revisit them.
6. Proper cleanliness should be maintained by managing proper dustbins on the beach, and there should be fines on those who throw garbage here and there.

6. Fact Finding Table

Objective	Technique	Subjects	Time Commitment
To get the background on beach activity management and the database management system.	Background Reading	Few similar projects related to beach activities and the activities.	2-day
Preliminary meeting to identify how the activities are organized and the problems faced by the organizers.	Interview	Manager	30 Minutes
Preliminary meeting to identify a list of activities and list of equipment required.	Interview	Administrator	30 Minutes
Meeting to understand the problems faced by the trainer.	Interview	Trainer	30 Minutes
Meeting to understand the safety operations on the beach and the issues in the system.	Interview	Safety Operations Incharge	30 Minutes
Meetings to understand problems while managing activities and equipment.	Interview	Deputy Administrator	30 Minutes
Meeting to know how various events are managed on the beaches.	Interview	Event Manager	30 Minutes
Meeting to know customers' views on the system and administration providing these services.	Interview	Customer	30 Minutes
To get to know people’s thoughts on the various parts of beach activity management	Questionnaire	Student	2 Day
To understand the beach activities management by observing.	Observation	Visit at beach	45 Minutes

7. List of Requirements

1. Data management of Customers: Frequency: 3

Information of customers needs to be appropriately stored and managed regularly. Customers must get their updates about the activity they have booked regularly so that customers can manage their time accordingly. This will help them efficiently work on their time management. Information about customers for promotional messages should also be saved for later use.

2. Management of Activities: Frequency: 2

Initially, activities should be selected, and their services should be decided.
All the people related to that activity should be assigned their jobs for managing the activities. Each activity should have a manager who will organize that activity and manage all the problems coming during the activity.

3. Management of Equipment: Frequency: 2

Equipment management is a must. Sufficient equipment should be available to manage all the activities and customers. The quality of each piece of equipment should be maintained by adequately storing it and providing it with proper servicing. All the equipment should be managed and stored correctly. While using the equipment, basic guidelines should be followed so that there is no severe damage to the equipment.

4. Management of Staff: Frequency: 2

All the staff members should be provided their role while joining. They should be given their daily to-do list, which they have to do daily. They should receive their salaries on the fifth day of the month. They should also be paid for the extra time they work. Their attendance and the time they are working can be measured by using the fingerprint attendance machine, which will note their joining time day and their leaving time of day to measure their working hours.

5. Safety Operations: Frequency: 2

This is the essential aspect of beach activity management. The safety of all the customers visiting the beach is essential. Safety Incharge Officer who will care for all the safety operations and drills on the beach should be appointed. He should also look

for no safety lapse during the activity. They should coordinate with the equipment manager that every piece of equipment used for the activity is in perfect condition. He should not allow the use of compromised equipment that might lead to risk. He should deploy all the lifeguards on the beach so they can cover the whole beach during their period.

6. Event management services: Frequency: 2

All events organized on the beach should be appropriately managed, and no double booking should be done for a specific area. Customers' demands must be appropriately managed, and their staff members should treat their guests very well.

7. Customer Friendly User face: Frequency: 1

There should be a platform where customers can view the services and pre book their activities. They can also know about the details of activities and all the essential information they require through that platform. This would also give them the data of participation in different activities to decide on their activity.

8. User Categories and Privileges

8.1 User Classes and Classification

1. Administrators:
They are the admin of the beach activities; they have access to all the data and rights to create/add/delete any information they want. They will have all the information about the activities, equipment, safety procedures, events, trainers, staff, customers, etc. They will also keep track of profits and losses.

2. Trainers:
They will have access to the activities allotted to them. They will also get the list of customers who are interested in getting trained and who will be participating in the activity.

3. Event Managers
They will have access to all organized events. They will also manage the bookings of events. They will also have access to the production material required. Staff members will be working on the event. Customer requirement list.

4. Equipment Manager
They will have access to all the list of equipment in the inventory. They will allocate different equipment to different activities. They will also maintain all the equipment by adequately storing and regularly checking them.

5. Booking Department
They will have all the access to book activities and the customer's database. They will manage that there is no issue in booking and no double booking is done for the same time.

6. Security Officer
Security must manage all the safety operations. He will have access to all the equipment quality and data of all the life jackets and lifeguards.

7. Customers
Customers will have access to view participation and details in all the activities to select the most suitable activity.

8.2 Privileges

- **List of user-class:**

1. Administrator
2. Trainers
3. Event Managers
4. Equipment Manager
5. Booking Department
6. Security Officer
7. Customers

- **List of privileges of user-class:**

1. Administrator

- All the manager's list
- Add/delete/update feature in every database
- Money transfer details
- Customers database
- Equipment database

- Event database
- Activities Database
- Safety Operations database
- Trainers and Staff database

2. Trainers

- Customers Database
- Activities assigned to them
- Equipments List

3. Event Managers

- Customers Database
- Event Booking Database
- Equipment List for Production
- Customer Requirement List
- Staff List
- Add/Update/Create Events list
- Add/Update/Create Event booking list

4. Equipment Manager

- Equipment Database
- Inventory Storage Database
- Activity Database
- Maintenance of Equipment
- Equipment Condition Database
- Add/Update/Create equipment

5. Booking Department

- Customers Database
- Booking Database
- Add/Update/Create booking
- Payment Database

6. Security Officer

- Equipment Condition Database
- Maintenance of Equipment
- Lifeguards List
- Safety Operations List
- Conduction of Safety Drills
- List of Staff members

7. Customers

- Activity Participation Database
- Add/Delete/Update their bookings
- View details about the activities and Event organization

9. Assumptions

1. We have assumed that the internet connectivity at the server is strong enough to handle all customer queries without crashing and no data loss occurs due to technical failures.
2. All the equipment is available for each activity. They all are assigned their equip_id and activities related to them.
3. All the equipment is also in good working condition.
4. All the staff members and trainers are given their specific id and password to access their systems. Their systems are encrypted and can not be hacked.
5. Customers can only participate in the allotted time only to change the time they have to request it before the buffer time. If the slot is available, it will only be allotted to them.
6. Customers doing pre-booking need to pay their payment while booking only.
7. Each and every activity would not be available at any beach. We assume that most of all activities are present at that beach.
8. We assume that each activity will have at least one trainer and at least one equipment.

10. Business Constraints

1. All the activities are in real-time, and the booking and time allocation process should occur instantaneously and be updated in real-time for the smooth running of the beach activities.
2. Amount of all the equipment is limited.
3. Capacity of activities is limited, i.e., only a limited number of customers can participate during a round.
4. Customers can take part in any number of activities they want.
5. Trainers are limited, so a trainer might have to handle multiple activities.
6. Lifeguards are limited, so they have to cover a larger beach area.
7. Some extra trainers, staff, and lifeguards will be called during weekends and holidays.

11. Section - 2 Noun Analysis

1. All Extracted Nouns and Verbs
2. Accepted nouns and Verbs list
3. Rejected Noun and Verbs List

1. All Extracted Nouns and Verbs

S.No.	Noun	Verbs
1	Management	manage
2	Management Systems	systematic
3	data	document
4	list	work
5	activities	list
6	manage	Risk
7	customers	demands

8	beaches	need
9	equipments	need
10	equipment	track
11	trainer	work
12	customer	experience
13	databases	improve
14	service providers	need
15	safety	proper
16	intended audience	keep
17	accidents	chances
18	Safety	operations
19	database	improve
20	intended audience	group
21	beaches	main
22	beaches	attraction
23	user	access
24	activities	business
25	Youth	young
26	customer	safety
27	adventurous	more
28	youth	interested
29	adult	interested
30	activities	fun
31	records	holds
32	services	improvement

33	participants	decide
34	participate	actions
35	equipment	maintains
36	operation	smooth
37	activities	safe
38	Service providers	learn
39	participation	improve
40	customer's activities	show
41	service	plan
42	activities	charges
43	product	designed
44	clients	more
45	customers	more
46	service	enhance
47	service	improve
48	People	attracted
49	sport	adventure
50	Beach activities	prevalent
51	Beach activities	adventurous
52	customers	attract
53	prices	reasonable
54	youth	enthusiastic
55	watersports	interested
56	Young generation	experience
57	beaches	adventure

58	trips	planning
59	beaches	visit
60	beaches	relax
61	Service	profits
62	customers	improve
63	Scuba Diving	
64	Snorkeling	
65	Paragliding	
66	Parasailing	
67	Flyboarding	
68	Windsurfing	
69	Beach Volleyball	
70	Jet Ski	
71	Kite surfing	
72	Beach Sprint rowing	
73	Banana Ride	
74	Speed Boat	
75	Dolphin Exploration	
75	FootVolley	
76	Beach Camping	
77	Swimming with Sharks	
78	Sailing	
79	activity	needs
80	administration	information
81	administrators	edit

82	administrators	create
83	customers	select
84	youth	invest
85	event	management
86	events	bookking
87	software	help
88	software	manage
89	event	collide
90	arrangements	demand
91	events	collide
92	customer	demand
93	Software	track
94	equipment	used
95	workers	service
96	software	inform
97	administrators	assistance
98	assistance	unavailable
99	customers	training
100	salary	maintain
101	customers	decision-making
102	equipment	missing
103	equipment	damaged
104	equipment	unavailable
105	customer	friendly
106	equipment	maintain

107	equipment	servicing
108	safety	measures
109	equipment	checking
110	expenses	track
111	activity	Maximum participation
112	customers	feedback
113	database	update
114	images	keep
115	videos	keep
116	activity	postpone
117	memory	safe
118	booking	reschedule
119	Booking	cancel
120	technical	difficulty
121	background	reading
122	zones	created
123	water animals	come
124	Trainer	trains
125	Trainer	activity

2. Accepted Nouns and Verbs List

S.No.	Candidate Entity Set	Candidate Attribute Set	Candidate Relationship Set
1	Customer	Customer_id, Customer_name, Date of birth: Age, Height, Weight, Gender, Activity_id, Booking_id	
2	Administrator	Admin_id, Admin_name	
3	Staff	Staff_id staff_name Department	
4	Trainers	Trainer_id, Trainer_name, Activity_id Activity_name	
5	Equipment	Equip_id Equip_name Equip_quantity Equip_condition Date_of_Purchase	
6	Security	Security_id, Security_name,	
7	Activities	Activity_id Activity_name Equip_id Trainer_id Activity_charge Activity_time	

8	Booking	Booking_id, Booking_date, Booking_time, Customer_id Payment_id: Payment_method Payment_Amount Activity_id	
9	Event	Event_id Event_name Event_time Event_date Event_requirement Admin_id Staff_id Booking_id	

3. Rejected Noun and Verbs List

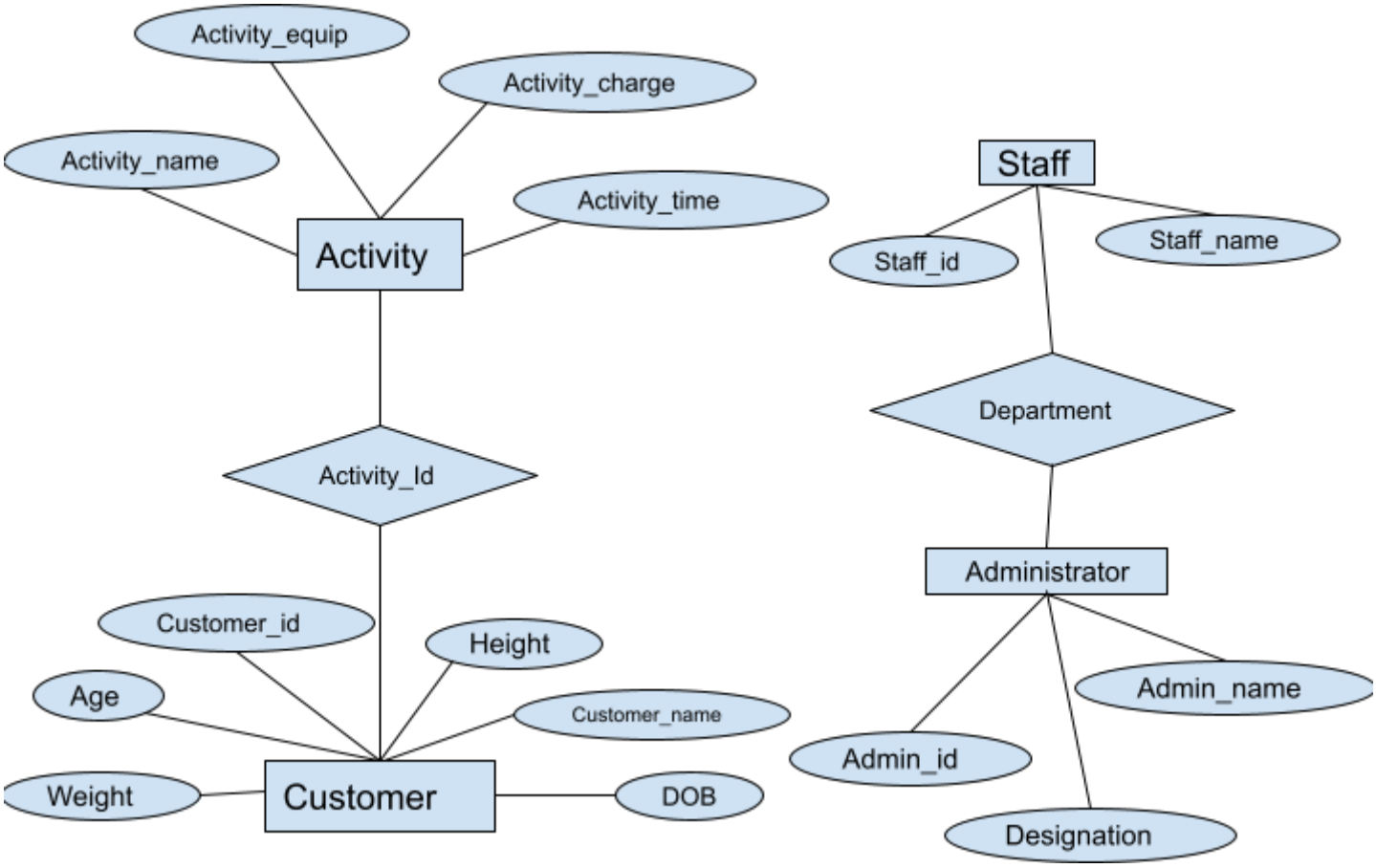
Sr. No.	Noun	Reject Reason
1	Management	Duplicate
2	Management Systems	General
3	data	duplicate
4	list	duplicate
6	manage	General
8	beaches	General
13	database	General
14	service providers	Duplicate
15	intended audience	Duplicate
16	accidents	General
17	Youth	duplicate
18	adventurous	General
19	adult	duplicate
20	records	General
21	operation	Irrelevant
22	product	Irrelevant
23	clients	Duplicate
24	service	Duplicate
25	People	General
26	sport	General
27	prices	Vague

28	watersports	Vague
29	Young generation	Vague
30	trips	duplicate
31	Scuba Diving	Vague
32	Snorkeling	Vague
33	Paragliding	Vague
34	Parasailing	Vague
35	Flyboarding	Vague
36	Windsurfing	Vague
37	Beach Volleyball	Vague
38	Jet Ski	Vague
39	Kite surfing	Vague
40	Beach Sprint rowing	Vague
41	Banana Ride	Vague
42	Speed Boat	Vague
43	Dolphin Exploration	Vague
44	FootVolley	Vague
45	Beach Camping	Vague
46	Swimming with Sharks	Vague
47	Sailing	irrelevant
48	software	General
49	arrangements	Duplicate
50	assistance	Associations
51	salary	Vague

52	expenses	Vague
53	images	Vague
54	videos	Vague
55	memory	Vague
56	technical	General
57	background	General
58	zones	irrelevant
59	water animals	irrelevant

12. ER Diagram:

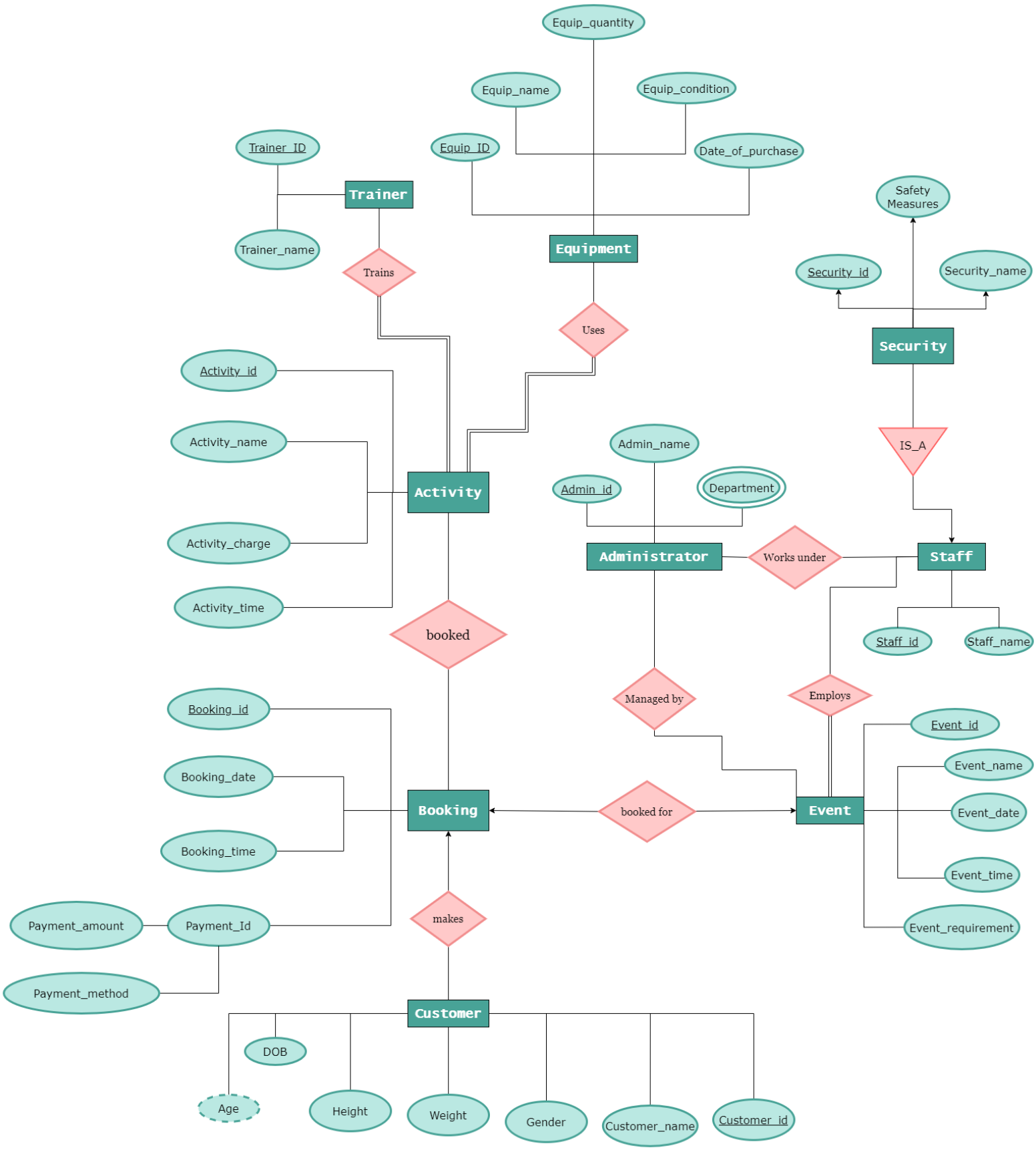
1. Version 0:



2. Version 1:


Equip_quantity
Equip_name
Equip_condition
Trainer_ID
Equip_ID
Date_of_purchase
Trainer
Trainer_name
Trains
Equipment
Uses
Security
Activity_id
Security_id
Security_name
Activity_name
Safety Measures
Admin_name
Activity
Admin_id
Department
Activity_charge
Administrator
Works under
Staff
Activity_time
booked
Staff_id
Staff_name
Booking_id
Managed by
Employs
Event_id
Booking_date
Event_name
Booking
booked for
Event
Event_date
Booking_time
Event_time
makes
Payment_amount
Payment_Id
Event_requirement
Payment_method
Customer
Customer_id
DOB
Age
Height
Weight
Customer_name

3. Version 2:


Equip_quantity
Equip_name
Equip_condition
Trainer_ID
Equip_ID
Date_of_purchase
Trainer
Trainer_name
Trains
Equipment
Uses
Safety Measures
Security_id
Security_name
Security
Activity_id
Activity_name
Activity_charge
Activity_time
Activity
Admin_name
Admin_id
Department
Administrator
Works under
IS_A
Staff
Staff_id
Staff_name
booked
Booking_id
Booking_date
Booking_time
Booking
Managed by
Employs
Event_id
Event_name
Event_date
Event_time
Event_requirement
booked for
Event
Payment_amount
Payment_Id
Payment_method
makes
Customer
DOB
Age
Height
Weight
Gender
Customer_name
Customer_id

4. Updated ER

S.No.	Entity	Attribute	Relationship
1	Customer	Customer_id, Customer_name, Date of birth: Age, Height, Weight, Gender, Activity_id, Booking_id	makes
2	Administrator	Admin_id, Admin_name	
3	Manager	Manager_id, Manager_name, Department	Manages, managed by, works under
4	Staff	Staff_id staff_name Department	Work under, Employs, is a
5	Trainers	Trainer_id, Trainer_name, Activity_id Activity_name	Trains, manages
6	Equipment	Equip_id Equip_name Equip_quantity Equip_condition Date_of_Purchase	Uses, manages
7	Security	Security_id, Security_name,	ISA
8	Activities	Activity_id Activity_name Equip_id Trainer_id Activity_charge	Booked, Manages, Uses, trains

		Activity_time	
9	Booking	Booking_id, Booking_date, Booking_time, Slot_availability, Customer_id Payment_id: Payment_method Payment_Amount Activity_id	Booked, mannages, booked for, makes
10	Event	Event_id Event_name Event_time Event_date Event_requirement Manager_id Staff_id Booking_id	Booked for, managed by, employs

13. Relationship Types:

1) Binary relationship:

a. Makes
b. Booked
c. Booked for
d. Manages
e. Trains
f. Uses
g. works under
h. Employs

2) Ternary relationship:

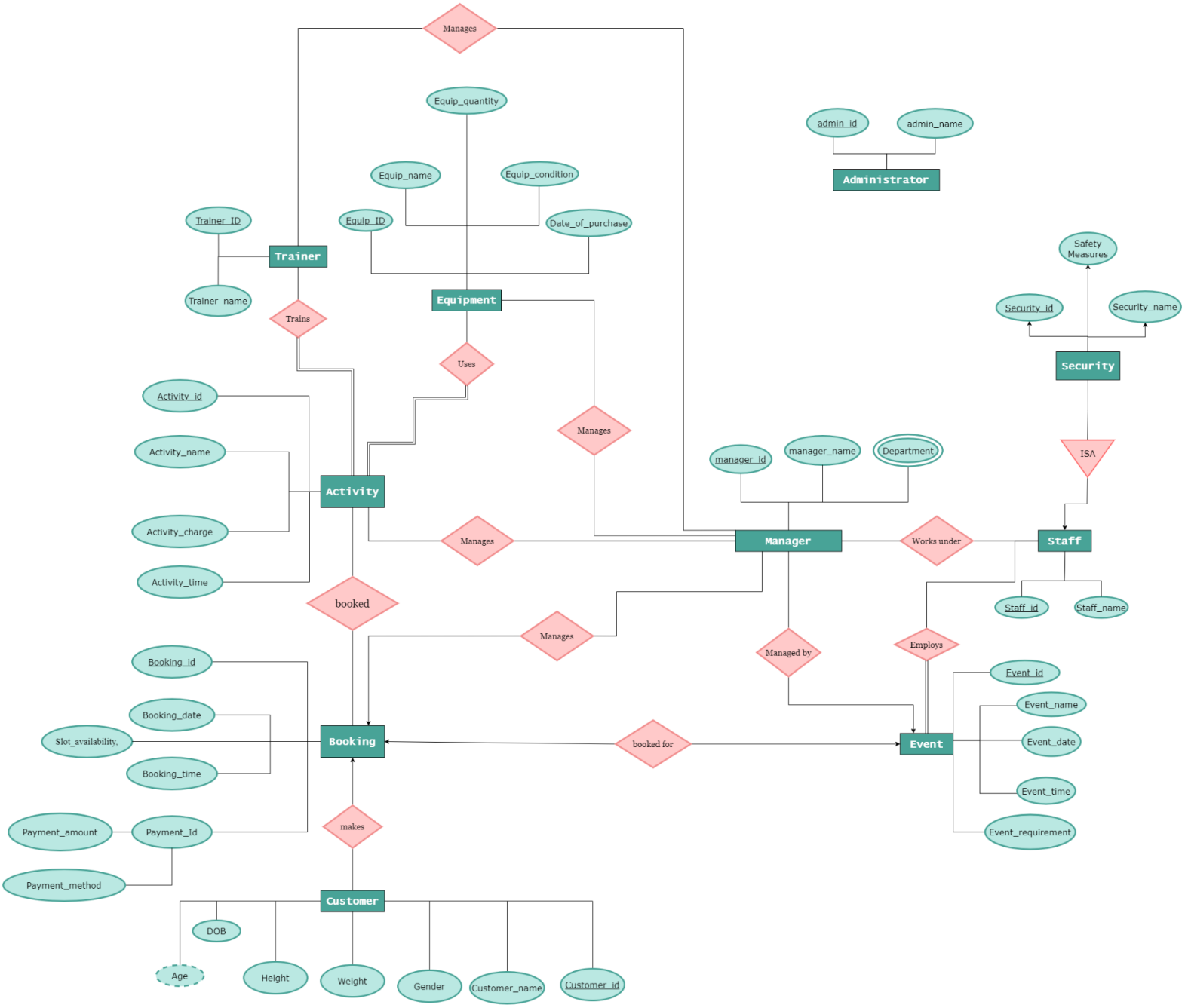
We don't have any ternary relationship.

3) Hierarchy relationship:

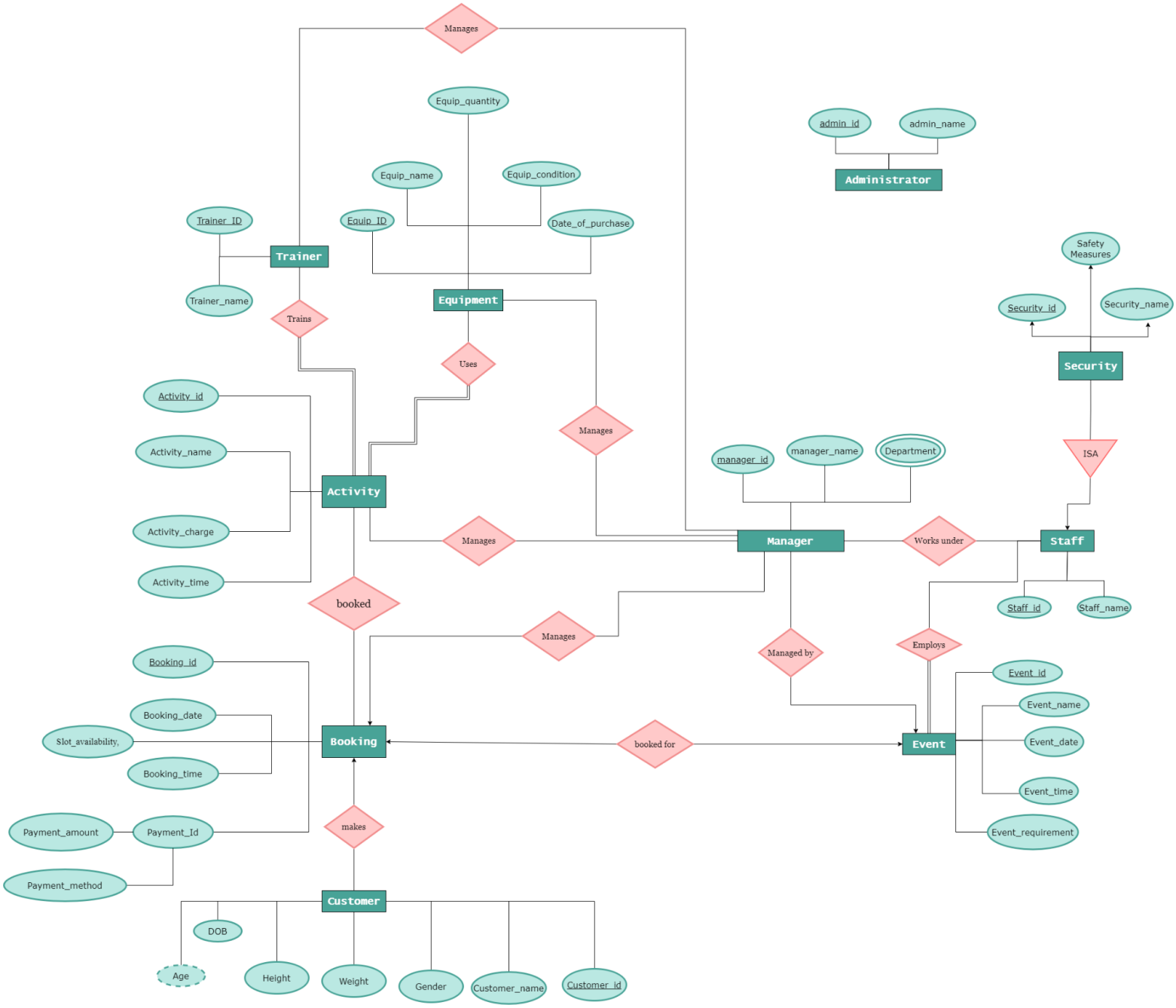
ISA is our hierarchy relationship: a security entity is a child of a staff entity.

- Here, we have created payment as an attribute of Booking. We can create it as an entity, but we created it as an attribute to booking because we can connect the payment received to the corresponding booking. This will help us in having more proper management of the amount.

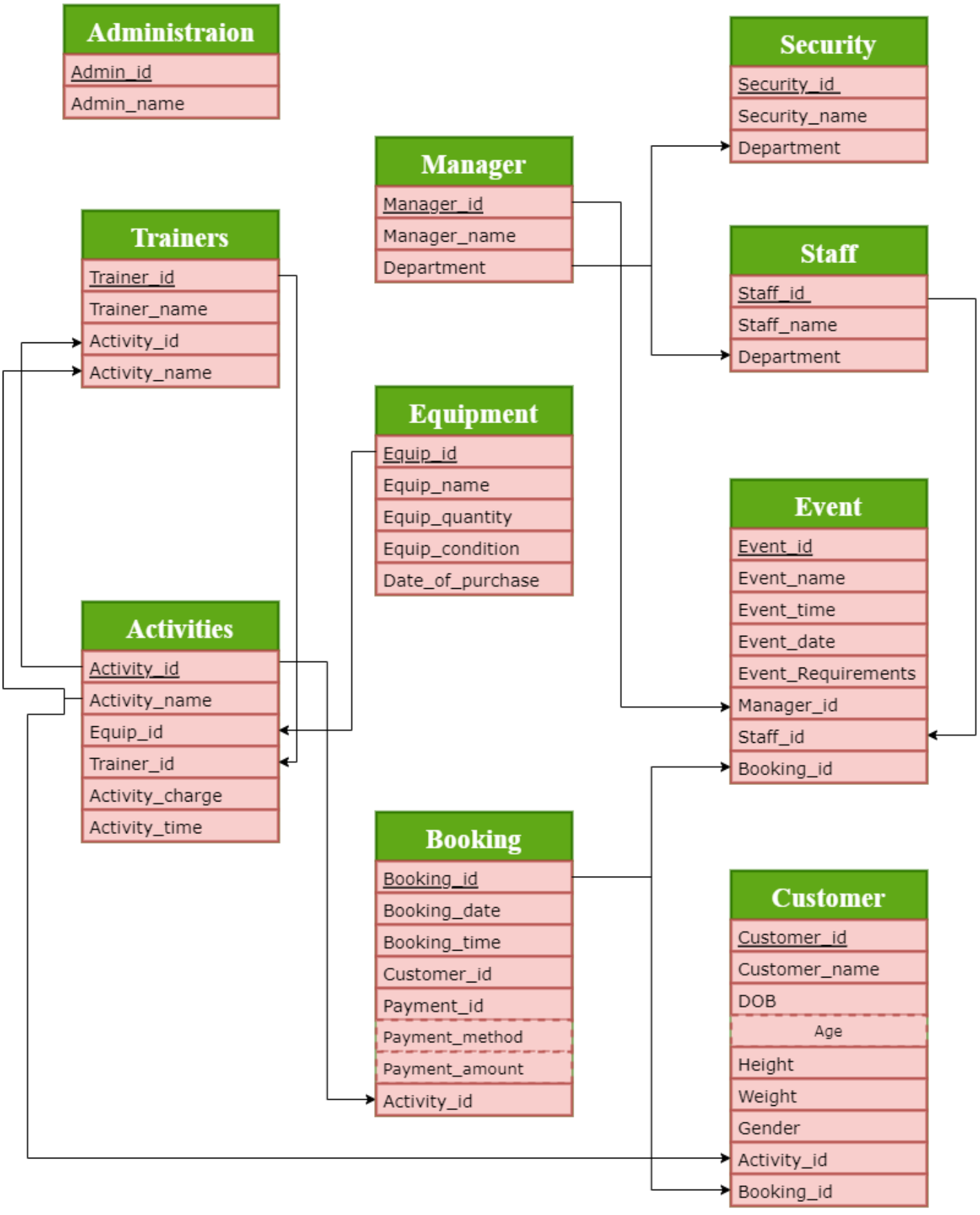
Version 3:



14. Final ERD:


Manages
Equip_quantity
admin_id
admin_name
Administrator
Equip_name
Equip_condition
Trainer_ID
Equip_ID
Date_of_purchase
Trainer
Safety Measures
Trainer_name
Trains
Equipment
Security_id
Security_name
Security
Uses
Activity_id
Manages
ISA
Activity_name
manager_id
manager_name
Department
Activity
Activity_charge
Manages
Manager
Works under
Staff
Activity_time
booked
Staff_id
Staff_name
Manages
Managed by
Employs
Booking_id
Event_id
Booking_date
Event_name
Slot_availability,
Booking
booked for
Event
Event_date
Booking_time
Event_time
makes
Payment_amount
Payment_Id
Event_requirement
Payment_method
Customer
DOB
Age
Height
Weight
Gender
Customer_name
Customer_id

15. Mapping E-R Model to Relational Model



16. Constraints:

In the above relational model we have 10 tables, each table has a primary key and some have foreign keys.

1. Administration

Here, admin_id is the primary key which will uniquely identify the administrator.

2. Manager

Here, manager_id is the primary key which will uniquely identify the manager.

3. Activity

Here, activity_id is the primary key which will uniquely identify the activities. Here Equip_id and Trainer_id are the foreign keys from equipment and trainers table, which will help us to identify the equipment related to that activity and the trainers related to that specific activity.

4. Equipment

Here, Equip_id is the primary key which will uniquely identify all the types of equipment present in the inventory.

5. Booking

Here, Booking_id is the primary key which will uniquely identify all the bookings. In this table customer_id and activity_id will be foreign keys from the table customers and activities which will help us to identify the customer who has done the booking and the activity for which it has booked to participate.

6. Customers

Here, customer_id is the primary key which will uniquely identify all the customers who have booked the activities or event. In This table we have activity_id and booking_id as foreign keys from the booking and activity table, which will help us to find the booking and activity of the customer.

7. Trainer

Here, trainer_id is the primary key which will uniquely identify all the trainers. In this table we will have activity_id and activity_name from the activities table as the foreign key to identify the activity to which the trainer is related and will train that activity.

8. Events

In this table, Event_id is the primary key which uniquely identifies all the events that are to be occurred. In this table we have booking_id from booking table, manager_id from manager table and staff_id from the staff table as foreign key so that we will about the manager who will be managing that event, booking_id will help us to see all the booking details and staff_id wll help us find the staff members who will be working for that particular event.

9. Staff

In this table, staff_id is the primary key which will uniquely identify the staff member.

10. Security

In this table, security_id is the primary key which will uniquely identify all the security staff. This table is also a child of the staff table.

17. Initial DDL Script:

```
--- Activity


-- Table: ba_ms.Activity


-- DROP TABLE ba_ms."Activity";


CREATE TABLE ba_ms."Activity"
(
    "Activty_name" "char" NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Acticity_charge" numeric NOT NULL,
    "Activity_time" numeric NOT NULL,
    "Trainer_Id" character varying COLLATE pg_catalog."default"
NOT NULL,
```

```
    "Equipment_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    CONSTRAINT "Activity_pkey" PRIMARY KEY ("Activity_id"),
    CONSTRAINT "Equipment_id" FOREIGN KEY ("Equipment_id")
        REFERENCES ba_ms."Equipment" ("Equipment_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Trainer_Id" FOREIGN KEY ("Trainer_Id")
        REFERENCES ba_ms."Trainers" ("Trainer_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Activity"
    OWNER to postgres;
-- Index: fki_Trainer_Id

-- DROP INDEX ba_ms."fki_Trainer_Id";

CREATE INDEX "fki_Trainer_Id"
    ON ba_ms."Activity" USING btree
    ("Trainer_Id" COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;
```

```
---Administration

-- Table: ba_ms.Administration


-- DROP TABLE ba_ms."Administration";


CREATE TABLE ba_ms."Administration"

(

    "Admin_id" character varying COLLATE pg_catalog."default" NOT
NULL,

    "Admin_name" "char" NOT NULL,

    CONSTRAINT "Administration_pkey" PRIMARY KEY ("Admin_id")

)


TABLESPACE pg_default;


ALTER TABLE ba_ms."Administration"

    OWNER to postgres;




---Booking


-- Table: ba_ms.Booking


-- DROP TABLE ba_ms."Booking";


CREATE TABLE ba_ms."Booking"
```

```
(
    "Booking_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Booking_date" date NOT NULL,
    "Booking_time" time without time zone NOT NULL,
    "Payment_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Payment_method" "char" NOT NULL,
    "Payment_amount" numeric NOT NULL,
    "Customer_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    CONSTRAINT "Booking_pkey" PRIMARY KEY ("Booking_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Customer_id" FOREIGN KEY ("Customer_id")
        REFERENCES ba_ms."Customer" ("Customer_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Booking"
    OWNER to postgres;
```

```
--Customer

-- Table: ba_ms.Customer


-- DROP TABLE ba_ms."Customer";


CREATE TABLE ba_ms."Customer"

(

    "Customer_id" character varying COLLATE pg_catalog."default"
NOT NULL,

    "Customer_name" "char" NOT NULL,

    "DOB" date[] NOT NULL,

    "Height" numeric[] NOT NULL,

    "Weight" numeric[] NOT NULL,

    "Gender" "char" NOT NULL,

    "Activity_id" character varying COLLATE pg_catalog."default"
NOT NULL,

    "Booking_id" character varying COLLATE pg_catalog."default"
NOT NULL,

    CONSTRAINT "Customer_pkey" PRIMARY KEY ("Customer_id"),

    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")

        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "Booking_id" FOREIGN KEY ("Booking_id")

        REFERENCES ba_ms."Booking" ("Booking_id") MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Customer"
    OWNER to postgres;




--Equipment

-- Table: ba_ms.Equipment

-- DROP TABLE ba_ms."Equipment";

CREATE TABLE ba_ms."Equipment"
(
    "Equipment_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Equipment_name" "char" NOT NULL,
    "Equipment_quantity" numeric NOT NULL,
    "Equipment_condition" "char" NOT NULL,
    "Date_of_purchase" date NOT NULL,
    CONSTRAINT "Equipment_pkey" PRIMARY KEY ("Equipment_id")
)
```

```
TABLESPACE pg_default;


ALTER TABLE ba_ms."Equipment"
    OWNER to postgres;




--Event
-- Table: ba_ms.Event


-- DROP TABLE ba_ms."Event";


CREATE TABLE ba_ms."Event"
(
    "Event_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Event_name" "char" NOT NULL,
    "Event_time" time without time zone NOT NULL,
    "Event_date" date NOT NULL,
    "Event_requirements" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Manager_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Staff_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Booking_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    CONSTRAINT "Event_pkey" PRIMARY KEY ("Event_id"),
```

```
    CONSTRAINT "Booking_id" FOREIGN KEY ("Booking_id")
        REFERENCES ba_ms."Booking" ("Booking_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Manager_id" FOREIGN KEY ("Manager_id")
        REFERENCES ba_ms."Manager" ("Manager_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Staff_id" FOREIGN KEY ("Staff_id")
        REFERENCES ba_ms."Staff" ("Staff_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Event"
    OWNER to postgres;




--Manageer

-- Table: ba_ms.Manager
```

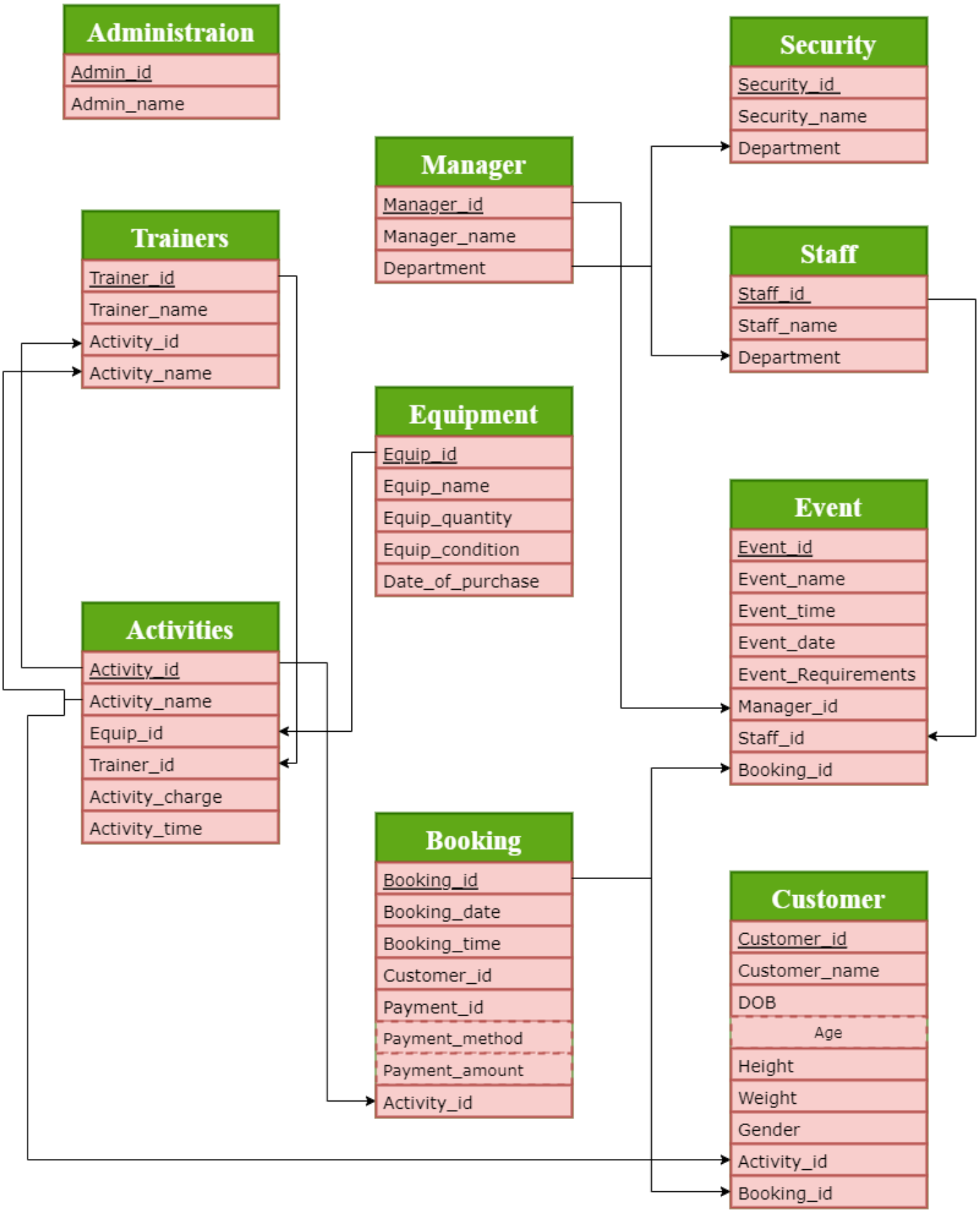
```
-- DROP TABLE ba_ms."Manager";


CREATE TABLE ba_ms."Manager"
(
    "Manager_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Manager_name" "char" NOT NULL,
    "Department" "char" NOT NULL,
    CONSTRAINT "Manager_id" PRIMARY KEY ("Manager_id")
)


TABLESPACE pg_default;


ALTER TABLE ba_ms."Manager"
    OWNER to postgres;







--Security
-- Table: ba_ms.Security


-- DROP TABLE ba_ms."Security";


CREATE TABLE ba_ms."Security"
(
    "Security_id" character varying COLLATE pg_catalog."default"
NOT NULL,
```

```
    "Security_name" "char" NOT NULL,
    "Department" "char" NOT NULL,
    CONSTRAINT "Security_pkey" PRIMARY KEY ("Security_id")
)

TABLESPACE pg_default;

ALTER TABLE ba_ms."Security"
    OWNER to postgres;




--Staff

-- Table: ba_ms.Staff

-- DROP TABLE ba_ms."Staff";

CREATE TABLE ba_ms."Staff"
(
    "Staff_id" character varying COLLATE pg_catalog."default" NOT
NULL,
    "Staff_name" "char" NOT NULL,
    "Department" "char" NOT NULL,
    CONSTRAINT "Staff_pkey" PRIMARY KEY ("Staff_id")
)
```

```
TABLESPACE pg_default;


ALTER TABLE ba_ms."Staff"
    OWNER to postgres;




--Trainers
-- Table: ba_ms.Trainers


-- DROP TABLE ba_ms."Trainers";


CREATE TABLE ba_ms."Trainers"
(
    "Trainer_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Trainer_name" "char" NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default"
NOT NULL,
    "Activity_name" "char" NOT NULL,
    CONSTRAINT "Trainers_pkey" PRIMARY KEY ("Trainer_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
```

```
TABLESPACE pg_default;


ALTER TABLE ba_ms."Trainers"
    OWNER to postgres;
```

18. Section - 3 Normalization & Schema Refinement:



1. Administration (<u>Admin_id</u>, Admin_name)

Primary Key: Admin_id

Dependencies: Admin_id -> Admin_name

Partial Key Dependency: None
Transitive Dependency: None

Redundancies: In Admin_name, it is possible that the same name is associated with a different Admin_id.

Anomalies:
Insert – None.
Update – None.
Delete – None.

Every attribute in this schema is single-valued (scalar), already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

2. Manager (Manager_id, Manager_name, department)

Primary Key: Manager_id

Dependencies:
Manager_id -> Manager_name,
Manager_id -> Department

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: In Manager_name, it is possible that the same name is associated with a different Manager_id. It might be possible that two manager_id are associated with the same department.

Anomalies:
Insert – None.
Update – None.
Delete – None.

Every attribute in this schema is single-valued (scalar), already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

3. Customer (Cutomer_id, Customer_name, DOB, age, Height, Weight, Gender, Activtiy_id, Booking_id)

Primary Key: Customer_id

Foreign Key: Activity_id, Booking_id

Dependencies:
Customer_id -> Customer_name,
Customer_id -> DOB,
Customer_id -> Age,
DOB -> Age,
Customer_id -> Height,
Customer_id -> Weight,
Customer_id -> Gender,

Partial Key Dependency: None

Transitive Dependency: Age is derived from DOB, and DOB is derived from Customer_id. We will drop the age attribute as it can be derived from the date of birth (DOB).

Redundancies: In Customer_name, it is possible that the same name is associated with a different Cuistomer_id.

Anomalies:
Insert – None.
Update – A customer's DOB cannot be updated once set.
Delete – Once assigned, one cannot delete the DOB or Activity_id of a particular customer.

In this, we drop booking_id, as in the booking table, there is customer_id through which we can get all the booking details.

Every attribute in this schema is single-valued (scalar), already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None

Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

a. It should be in the third standard form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

4. Booking (Booking_id, Booking_date, Booking_time, Customer_id, Payment_id, Payment_method, Payment_amount, Activity_id)

Primary Key: Booking_id

Foreign Key: Customer_id, Activity_id

Dependencies:
Booking_id -> Booking_date,
Booking_id -> Booking_time,
Booking_id -> Payment_id,
Booking_id -> Payment_method,
Booking_id -> Payment_amount

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – Once assigned, booking_id cannot be updated.
Delete – None.

Every attribute in this schema is single-valued (scalar), already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.
Hence, this relation is in BCNF as well.

5. Activity (Activity_id, Activity_name, Activity_charge, Activity_time, Equip_id, Trainer_id)

Primary Key: Activity_id

Foreign Key: Equip_id, Trainer_id

Dependencies:
Activity_id -> Activity_name,
Activity_id -> Activity_charge
Activity_id -> Activity_time

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – None
Delete – None.

Every attribute in this schema is not a single-valued (scalar), already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

a. It should be in the third standard form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

6. Equipment (Equip_id, Equip_name, Equip_quantity, Equip_condition, Date_of_purchase)

Primary Key: Equip_id

Dependencies:
Equip_id-> Equip_name,
Equip_id-> Equip_quantity,
Equip_id-> Equip_condition
Equip_id -> Date_of_purchase

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – None.
Delete – None.

Every attribute in this schema is single-valued (scalar), already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well

7. Trainers (Trainer_id, Trainer_name, Activity_id, activity_name)

Primary Key: Trainer_id

Foreign Key: Activity_id

Dependencies:
Trainer_id-> Trainer_name

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this, we drop activity_id and activity_name as this column is present in the Activity table, and trainer_id is given to them as a foreign key.

In this schema, every attribute is single-valued (scalar), making it already in 1NF. There is no partial dependency here, so it is in 2NF.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

8. Event (Event_id, Event_name, Event_time, Event_date, Event_requirements, manager_id, Staff_id, Booking_id)

Primary Key: Event_id

Foreign Key: Manager_id, Booking_id, Staff_id

Dependencies:
Event_id -> Event_name,
Event_id -> Event_time,
Event_id -> Event_date,
Event_id -> Event_requirements

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this schema, every attribute is single-valued (scalar), making it already in 1NF. There is no partial dependency here, so it is in 2NF as well.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.
Hence, this relation is in BCNF as well

9. Security (Security_id, Security_name, Department)

Primary Key: Security_id

Dependencies:
Security_id -> Security_name,
Security_id -> Department

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this schema, every attribute is single-valued (scalar) making it already in 1NF. There is no partial dependency here, so it is in 2NF as well.

2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,
1. It should be in the third normal form (3NF).
2. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well

10. Staff (Staff_id, Staff_name, department)

Primary Key: Staff_id

Dependencies:
Staff_id -> Staff_name,
Staff_id -> Department

Partial Key Dependency: None

Transitive Dependency: None

Redundancies: None

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this schema, every attribute is single-valued (scalar), making it already in 1NF. There is no partial dependency here, so it is in 2NF.

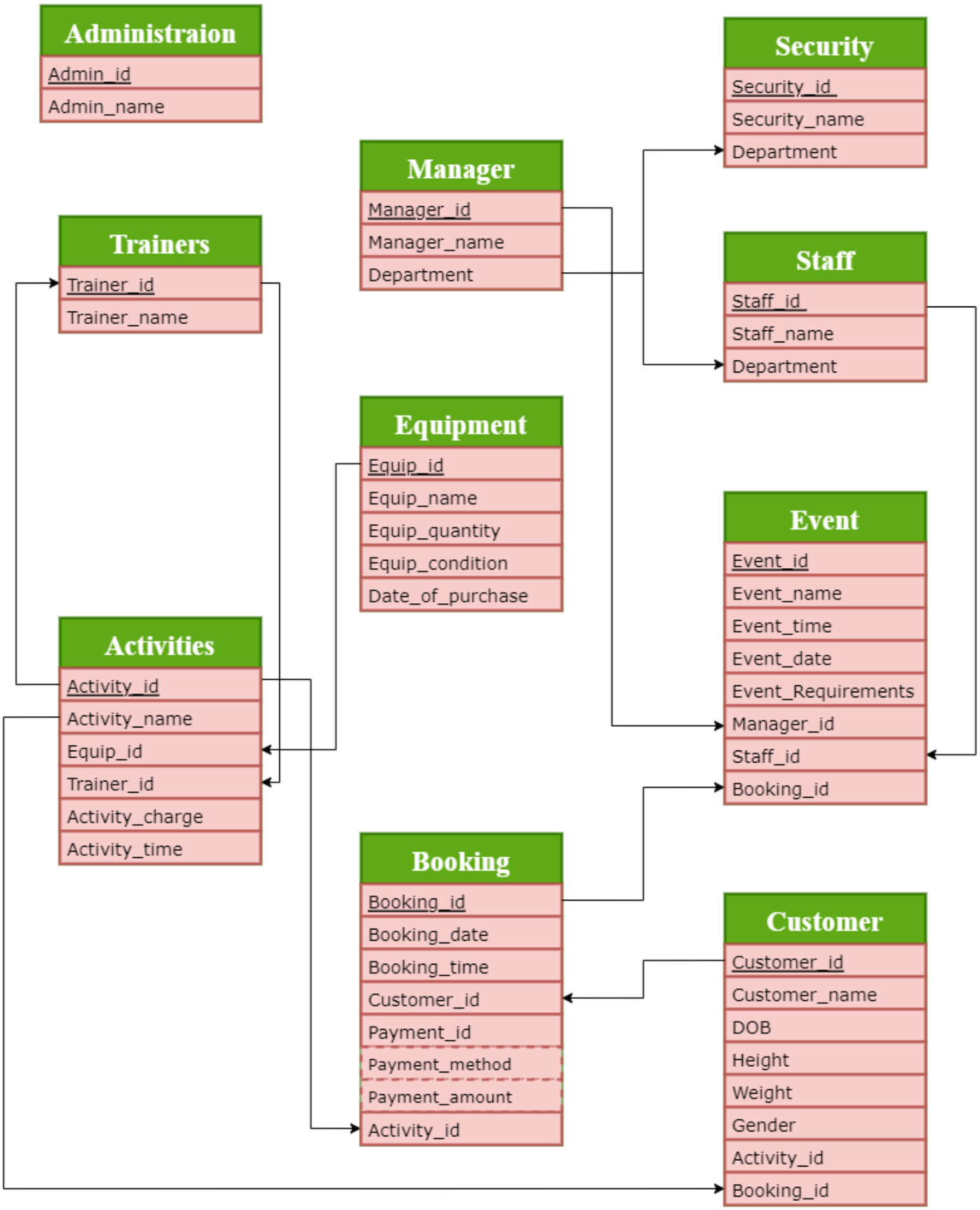
2NF Redundancies: None
Since there is no transitive dependency, it is also in 3NF. Thus, our final schema remains the same.

For a relation to be in **BCNF**,

1. It should be in the third normal form (3NF).
2. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

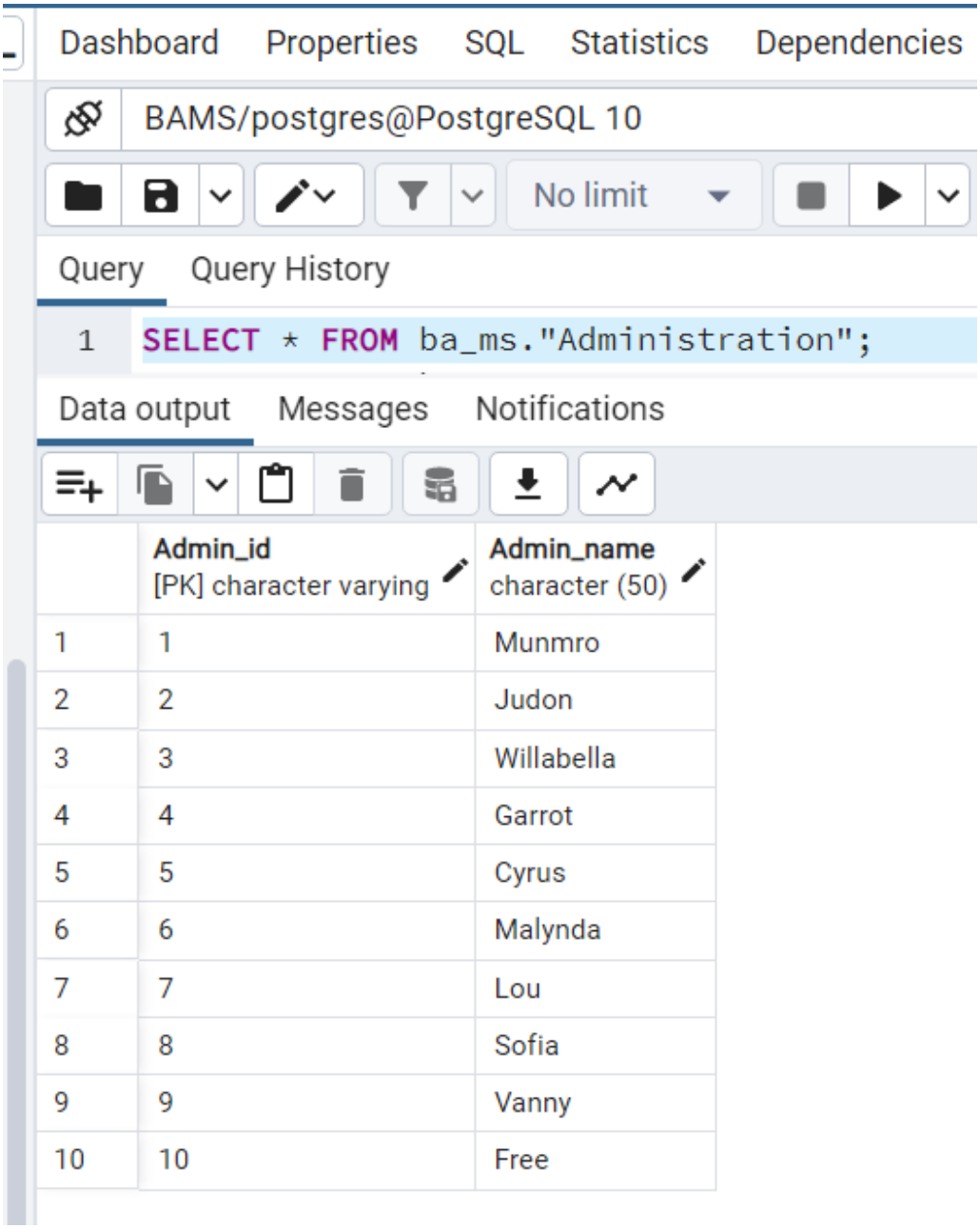
19. Updated Relational Model:



20. Final DDL and Select Statements

1. Administration (Admin_id, Admin_name)

```
---Administration

CREATE TABLE IF NOT EXISTS ba_ms."Administration"
(
    "Admin_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Admin_name" character(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Administration_pkey" PRIMARY KEY ("Admin_id")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Administration"
    OWNER to postgres;

COPY ba_ms."Administration"
FROM 'C:\Users\Public\sn_db\Administration.csv'
DELIMITER ',' CSV HEADER;
```



	Admin_id [PK] character varying	Admin_name character (50)
1	1	Munmro
2	2	Judon
3	3	Willabella
4	4	Garrot
5	5	Cyrus
6	6	Malynda
7	7	Lou
8	8	Sofia
9	9	Vanny
10	10	Free

2. Manager (Manager_id, Manager_name, department)

```
--- Manager ---

CREATE TABLE IF NOT EXISTS ba_ms."Manager"
(
    "Manager_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Manager_name" character(50) COLLATE pg_catalog."default" NOT NULL,
    "Department" character(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Manager_id" PRIMARY KEY ("Manager_id")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Manager"
    OWNER to postgres;


COPY ba_ms."Manager"
FROM 'C:\Users\Public\sn_db\Manager.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard Properties SQL Statistics Dependencies

BAMS/postgres@PostgreSQL 10

No limit

Query Query History

```
1  SELECT * FROM ba_ms."Administration";
2  SELECT * FROM ba_ms."Manager";
```

Data output Messages Notifications

	Manager_id [PK] character varying	Manager_name character (50)	Department character (50)
1	1	Jo	Activity
2	2	Clay	Equipment
3	3	Reggis	Staff
4	4	Elfrieda	Security
5	5	Babita	Trainer
6	6	Paloma	Event
7	7	Aurlie	Booking
8	8	Georgina	Activity
9	9	Sam	Equipment
10	10	Dre	Staff
11	11	Dirk	Security
12	12	Gerrilee	Trainer
13	13	Elmira	Event
14	14	Annmaria	Booking
15	15	Carolynn	Activity

Total rows: 15 of 15 Query complete 00:00:00.179

3. Staff (Staff_id, Staff_name, Department)

```
--- Staff ---

CREATE TABLE IF NOT EXISTS ba_ms."Staff"
(
    "Staff_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Staff_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Department" character(100) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Staff_pkey" PRIMARY KEY ("Staff_id")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Staff"
    OWNER to postgres;

COPY ba_ms."Staff"
FROM 'C:\Users\Public\sn_db\Staff.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard Properties SQL Statistics Dependencies Dependents

BAMS/postgres@PostgreSQL 10

No limit

Query Query History

```
2  SELECT * FROM ba_ms."Manager";
3  SELECT * FROM ba_ms."Staff";
```

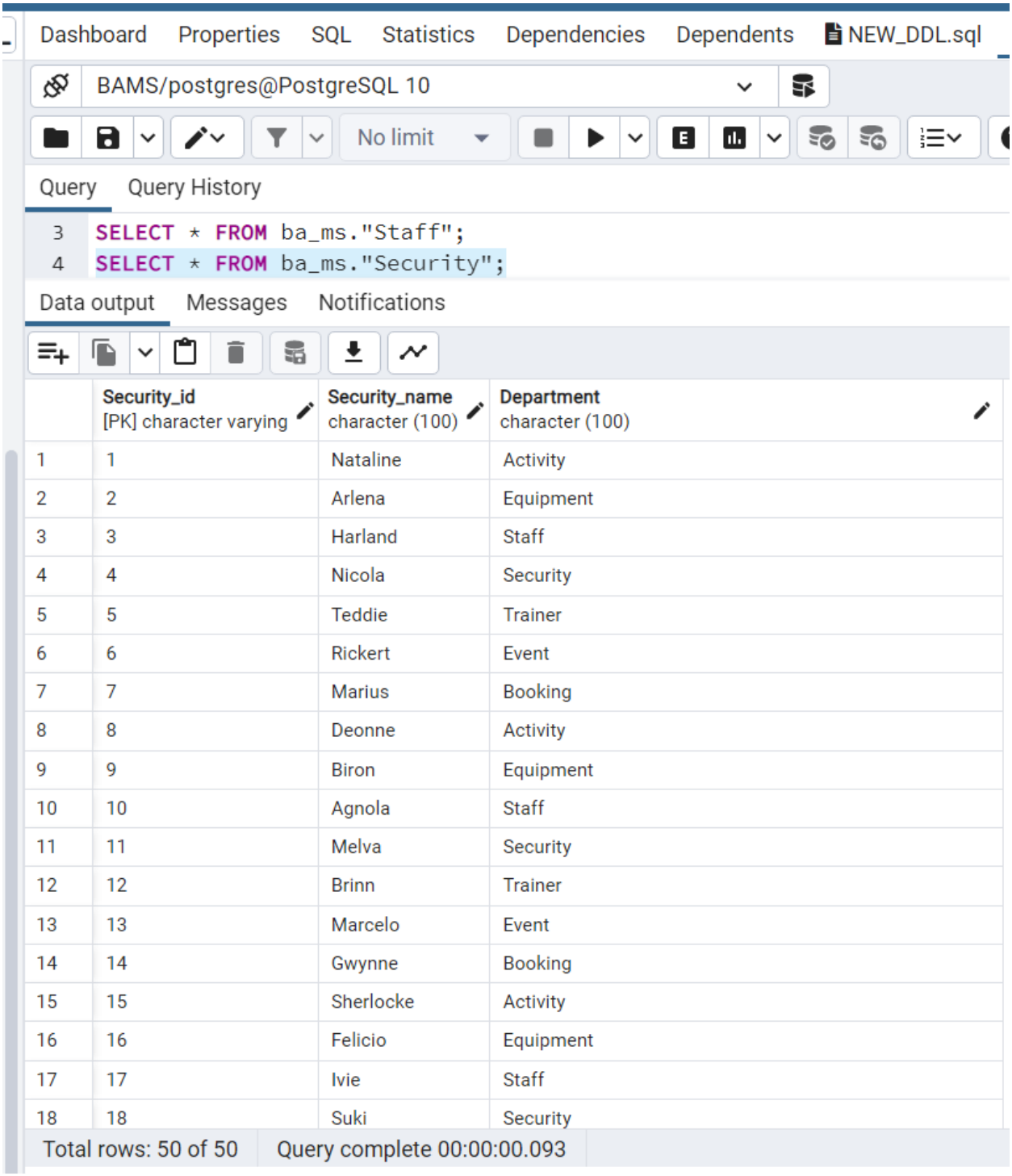
Data output Messages Notifications

	Staff_id [PK] character varying	Staff_name character (100)	Department character (100)
1	1	Karlik	Activity
2	2	Melosa	Equipment
3	3	Ame	Staff
4	4	Dawn	Security
5	5	Francesco	Trainer
6	6	Reyna	Event
7	7	Barbe	Booking
8	8	Ivy	Activity
9	9	Tamiko	Equipment
10	10	Wally	Staff
11	11	Diann	Security
12	12	Michael	Trainer
13	13	Stanleigh	Event
14	14	Florance	Booking
15	15	Claudia	Activity
16	16	Mimi	Equipment
17	17	Christoper	Staff
18	18	Eden	Security

Total rows: 50 of 50 Query complete 00:00:00.490

4. Security (Security_id, Security_name, Department)

```
--- Security ---

CREATE TABLE IF NOT EXISTS ba_ms."Security"
(
    "Security_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Security_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Department" character(100) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Security_pkey" PRIMARY KEY ("Security_id")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Security"
    OWNER to postgres;


COPY ba_ms."Security"
FROM 'C:\Users\Public\sn_db\Security.csv'
DELIMITER ',' CSV HEADER;
ACCESS
```


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
NEW_DDL.sql
BAMS/postgres@PostgreSQL 10
No limit
Query
Query History
3 SELECT * FROM ba_ms."Staff";
4 SELECT * FROM ba_ms."Security";
Data output
Messages
Notifications
Security_id
[PK] character varying
Security_name
character (100)
Department
character (100)
1 1 Nataline Activity
2 2 Arlena Equipment
3 3 Harland Staff
4 4 Nicola Security
5 5 Teddie Trainer
6 6 Rickert Event
7 7 Marius Booking
8 8 Deonne Activity
9 9 Biron Equipment
10 10 Agnola Staff
11 11 Melva Security
12 12 Brinn Trainer
13 13 Marcelo Event
14 14 Gwynne Booking
15 15 Sherlocke Activity
16 16 Felicio Equipment
17 17 Ivie Staff
18 18 Suki Security
Total rows: 50 of 50
Query complete 00:00:00.093

5. Customer (Customer_id, Customer_name, DOB, age, Height, Weight, Gender, Activtiy_id, Booking_id)

```
--- Customer ---

CREATE TABLE IF NOT EXISTS ba_ms."Customer"
(
    "Customer_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Customer_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Gender" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "DOB" date NOT NULL,
    "Height" numeric NOT NULL,
    "Weight" numeric NOT NULL,
    CONSTRAINT "Customer_pkey" PRIMARY KEY ("Customer_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Customer"
    OWNER to postgres;

COPY ba_ms."Customer"
FROM 'C:\Users\Public\sn_db\Customer.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | NEW_DDL.sql | Inserion.sql

BAMS/postgres@PostgreSQL 10

No limit

Query | Query History

```
9  SELECT * FROM ba_ms."Customer";
```

Data output | Messages | Notifications

	Customer_id [PK] character varying	Customer_name character (100)	Gender character (100)	Activity_id character varying	DOB date	Height numeric	Weight numeric
1	1	Pepillo	M	1	2003-07-09	163.07	84.84
2	2	Elise	F	2	2003-07-10	127.31	55.76
3	3	Raquela	F	3	2003-07-11	133.77	81.73
4	4	Martino	M	4	2003-07-12	154.11	101.24
5	5	Emmanuel	M	5	1990-12-11	157.57	75.65
6	6	Bordie	M	6	1990-12-12	104.37	112.1
7	7	Baudoin	M	7	2003-07-09	130.98	100.2
8	8	Hillel	M	8	1988-10-08	138.96	48.6
9	9	Glenda	F	9	1988-10-09	146.67	69.99
10	10	Lilly	F	10	1988-10-10	105.23	50.49
11	11	Cal	M	11	1988-10-11	113.79	91.99
12	12	Octavius	M	12	2000-04-08	119.33	113.92
13	13	Phillipe	M	13	2000-04-09	131.5	55.15
14	14	Maxwell	M	14	2000-04-10	165.64	53.78
15	15	Gare	M	15	2000-04-11	187.46	124.97
16	16	Jobye	F	16	2001-09-04	169.28	48.76
17	17	Abbe	M	17	2001-09-05	195.89	46.06
18	18	Vachel	M	18	2001-09-06	106.53	127.43

Total rows: 350 of 350 | Query complete 00:00:00.215

6. Booking (Booking_id, Booking_date, Booking_time, Customer_id, Payment_id, Payment_method, Payment_amount, Activity_id)

```
--- Booking ---

CREATE TABLE IF NOT EXISTS ba_ms."Booking"
(
    "Booking_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Booking_date" date NOT NULL,
    "Booking_time" time without time zone NOT NULL,
    "Payment_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Payment_method" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Payment_amount" numeric NOT NULL,
    "Activity_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Customer_id" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Booking_pkey" PRIMARY KEY ("Booking_id"),
    CONSTRAINT "Activity_id" FOREIGN KEY ("Activity_id")
        REFERENCES ba_ms."Activity" ("Activity_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Customer_id" FOREIGN KEY ("Customer_id")
        REFERENCES ba_ms."Customer" ("Customer_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Booking"
    OWNER to postgres;

COPY ba_ms."Booking"
FROM 'C:\Users\Public\sn_db\Booking.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | NEW_DDL.sql | Inserion.sql

BAMS/postgres@PostgreSQL 10

No limit

Query | Query History

```
5  SELECT * FROM ba_ms."Booking";
```

Data output | Messages | Notifications

	Booking_id [PK] character varying	Booking_date date	Booking_time time without time zone	Payment_id character varying	Payment_method character (100)	Payment_amount numeric	Activity_id character varying	Customer_id character varying
1	1	2023-05-01	00:37:00	2.01E+14	Cash	6073	1	1
2	2	2023-05-02	08:02:00	5.10E+15	Card	3169	2	2
3	3	2023-05-03	19:36:00	4.41E+15	Upi	1736	3	3
4	4	2023-05-04	02:01:00	5.60E+15	Cash	3204	4	4
5	5	2023-05-05	10:31:00	3.58E+15	Card	3564	5	5
6	6	2023-05-06	16:28:00	5.60E+15	Upi	8126	6	6
7	7	2023-05-07	18:43:00	3.55E+15	Cash	2597	7	7
8	8	2023-05-08	08:40:00	3.58E+15	Card	9213	8	8
9	9	2023-05-09	23:08:00	3.57E+15	Upi	3914	9	9
10	10	2023-05-10	17:53:00	3.53E+15	Cash	2443	10	10
11	11	2023-05-11	19:06:00	3.06E+13	Card	3360	11	11
12	12	2023-05-12	10:44:00	5.61E+16	Upi	6068	12	12
13	13	2023-05-13	05:54:00	3.55E+15	Cash	5812	13	13
14	14	2023-05-14	19:15:00	5.01E+15	Card	7198	14	14
15	15	2023-05-15	08:32:00	3.57E+15	Upi	7780	15	15
16	16	2023-05-16	10:02:00	3.55E+15	Cash	9715	1	16
17	17	2023-05-17	17:33:00	5.60E+15	Card	2676	2	17
18	18	2023-05-18	19:40:00	3.66E+13	Upi	4324	3	18

Total rows: 350 of 350 | Query complete 00:00:00.096

7. Event (Event_id, Event_name, Event_time, Event_date, Event_requirements, manager_id, Staff_id, Booking_id)

```
--- Event ---

CREATE TABLE IF NOT EXISTS ba_ms."Event"
(
    "Event_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Event_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Event_time" time without time zone NOT NULL,
    "Event_date" date NOT NULL,
    "Event_requirements" character varying COLLATE pg_catalog."default" NOT NULL,
    "Manager_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Staff_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Booking_id" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Event_pkey" PRIMARY KEY ("Event_id"),
    CONSTRAINT "Booking_id" FOREIGN KEY ("Booking_id")
        REFERENCES ba_ms."Booking" ("Booking_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Manager_id" FOREIGN KEY ("Manager_id")
        REFERENCES ba_ms."Manager" ("Manager_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Staff_id" FOREIGN KEY ("Staff_id")
        REFERENCES ba_ms."Staff" ("Staff_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Event"
    OWNER to postgres;

COPY ba_ms."Event"
FROM 'C:\Users\Public\sn_db\Event.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard Properties SQL Statistics Dependencies Dependents NEW_DDL.sql Inserion.sql

BAMS/postgres@PostgreSQL 10

No limit

Query Query History

```
10  SELECT * FROM ba_ms."Event";
```

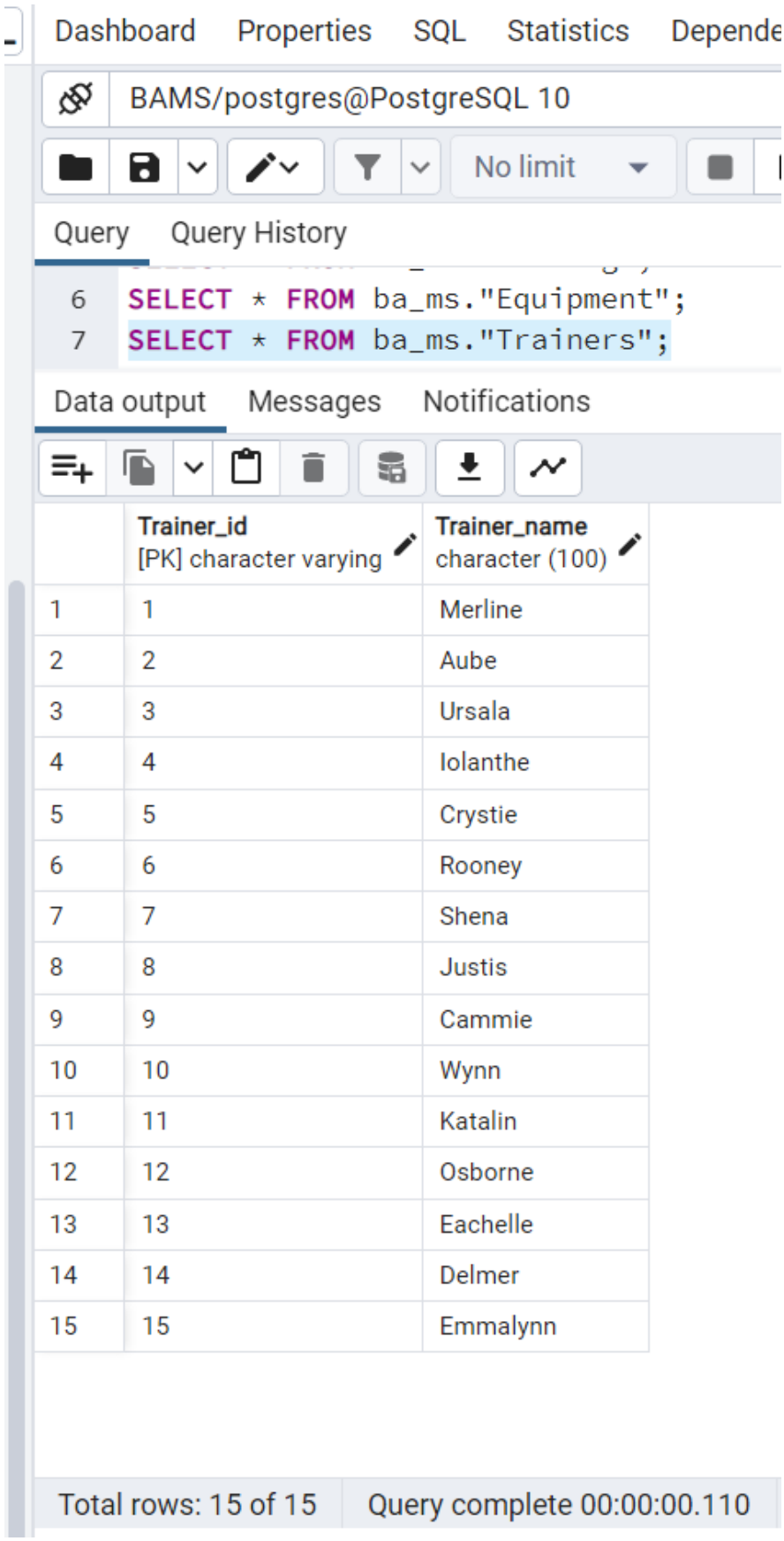
Data output Messages Notifications

	Event_id [PK] character varying	Event_name character (100)	Event_time time without time zone	Event_date date	Event_requirements character varying	Manager_id character varying	Staff_id character varying	Booking_id character varying
1	1	Birthday	00:29:00	2023-06-02	baloons	1	1	1
2	2	Wedding	02:40:00	2024-03-11	lights	2	2	2
3	3	Office Party	12:52:00	2023-11-09	cardboards	3	3	3
4	4	Marriage Party	20:05:00	2024-10-01	cartoons	4	4	4
5	5	Anniversary	18:41:00	2023-11-11	chairs	5	5	5
6	6	Fest	16:14:00	2023-11-12	tables	6	6	6
7	7	College Events	13:57:00	2023-11-13	stage	7	7	7
8	8	Companies Eve…	06:23:00	2023-11-14	photographer	8	8	8
9	9	Birthday	15:38:00	2023-11-15	cook	9	9	9
10	10	Wedding	19:16:00	2023-01-05	candles	10	10	10
11	11	Office Party	05:48:00	2023-01-06	baloons	11	11	11
12	12	Marriage Party	19:22:00	2023-01-06	lights	12	12	12
13	13	Anniversary	20:43:00	2023-01-07	cardboards	13	13	13
14	14	Fest	07:33:00	2024-04-01	cartoons	14	14	14
15	15	College Events	21:23:00	2024-04-02	chairs	15	15	15
16	16	Companies Eve…	05:19:00	2024-04-03	tables	1	16	16
17	17	Birthday	07:16:00	2023-08-07	stage	2	17	17
18	18	Wedding	16:57:00	2023-08-08	photographer	3	18	18

Total rows: 100 of 100 Query complete 00:00:00.263

8. Trainers (Trainer_id, Trainer_name, Activity_id, activity_name)

```
--- Trainers —

CREATE TABLE IF NOT EXISTS ba_ms."Trainers"
(
    "Trainer_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Trainer_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Trainers_pkey" PRIMARY KEY ("Trainer_id")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Trainers"
    OWNER to postgres;

COPY ba_ms."Trainers"
FROM 'C:\Users\Public\sn_db\Trainers.csv'
DELIMITER ',' CSV HEADER;
```


Dashboard
Properties
SQL
Statistics
Depende
BAMS/postgres@PostgreSQL 10
No limit
Query
Query History
6 SELECT * FROM ba_ms."Equipment";
7 SELECT * FROM ba_ms."Trainers";
Data output
Messages
Notifications
Trainer_id
[PK] character varying
Trainer_name
character (100)
1 1 Merline
2 2 Aube
3 3 Ursala
4 4 Iolanthe
5 5 Crystie
6 6 Rooney
7 7 Shena
8 8 Justis
9 9 Cammie
10 10 Wynn
11 11 Katalin
12 12 Osborne
13 13 Eachelle
14 14 Delmer
15 15 Emmalynn
Total rows: 15 of 15
Query complete 00:00:00.110

9. Activity (Activity_id, Activity_name, Activity_charge, Activity_time, Equip_id, Trainer_id)

```
--- Activity ---

CREATE TABLE IF NOT EXISTS ba_ms."Activity"
(
    "Activity_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Activity_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Activity_charge(in $)" integer NOT NULL,
    "Activity_time(in min)" integer NOT NULL,
    "Equip_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Trainer_id" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Activity_pkey" PRIMARY KEY ("Activity_id"),
    CONSTRAINT "Equip_id" FOREIGN KEY ("Equip_id")
        REFERENCES ba_ms."Equipment" ("Equipment_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Trainer_id" FOREIGN KEY ("Trainer_id")
        REFERENCES ba_ms."Trainers" ("Trainer_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Activity"
    OWNER to postgres;

COPY ba_ms."Activity"
FROM 'C:\Users\Public\sn_db\Activity.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard Properties SQL Statistics Dependencies Dependents NEW_DDL.sql Inserion.sql

BAMS/postgres@PostgreSQL 10

No limit

Query Query History

```
8  SELECT * FROM ba_ms."Activity";
```

Data output Messages Notifications

	Activity_id [PK] character varying	Activity_name character (100)	Activity_charge(in $) integer	Activity_time(in min) integer	Equip_id character varying	Trainer_id character varying
1	1	Scuba Diving	17	30	1	1
2	2	Snorkeling	12	35	2	2
3	3	Paragliding	10	20	3	3
4	4	Parasailing	15	25	4	4
5	5	Flyboarding	18	40	5	5
6	6	Windsurfing	16	45	6	6
7	7	Beach Volleyball	14	50	7	7
8	8	Jet Ski	9	60	8	8
9	9	Kite surfing	15	120	9	9
10	10	Beach Sprint ro...	6	90	10	10
11	11	Banana Ride	16	30	11	11
12	12	Speed Boat	15	45	12	12
13	13	Dolphin Explora...	17	60	13	13
14	14	FootVolley	12	25	14	14
15	15	Beach Camping	18	40	15	15
16	16	Beach Yoga	12	60	16	1
17	17	Tug of War	20	15	17	2
18	18	Swimming with...	30	45	18	3

Total rows: 20 of 20 Query complete 00:00:00.084

10. Equipment (Equip_id, Equip_name, Equip_quantity, Equip_condition, Date_of_purchase)

```
--- Equipment ---

CREATE TABLE IF NOT EXISTS ba_ms."Equipment"
(
    "Equipment_id" character varying COLLATE pg_catalog."default" NOT NULL,
    "Equipment_name" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Equipment_quantity" numeric NOT NULL,
    "Equipment_condition" character(100) COLLATE pg_catalog."default" NOT NULL,
    "Date_of_purchase" date NOT NULL,
    CONSTRAINT "Equipment_pkey" PRIMARY KEY ("Equipment_id")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS ba_ms."Equipment"
    OWNER to postgres;


COPY ba_ms."Equipment"
FROM 'C:\Users\Public\sn_db\Equipment.csv'
DELIMITER ',' CSV HEADER;
```

Dashboard Properties SQL Statistics Dependencies Dependents NEW_DDL.sql Inserion

BAMS/postgres@PostgreSQL 10

No limit

Query Query History

```
6  SELECT * FROM ba_ms."Equipment";
```

Data output Messages Notifications

	Equipment_id [PK] character varying	Equipment_name character (100)	Equipment_quantity numeric	Equipment_condition character (100)	Date_of_purchase date
1	1	Life Jacket	66	new	2018-01-09
2	2	Helmet	54	old	2020-02-09
3	3	Boat	26	need to change	2017-03-09
4	4	Knives	60	new	2020-04-09
5	5	Hinges	16	old	2020-05-09
6	6	Ropes	31	need to change	2019-06-09
7	7	Parachutes	20	new	2020-07-09
8	8	Knee Gaurds	13	old	2020-08-09
9	9	Albow Gaurds	70	need to change	2020-09-09
10	10	Wistles	76	new	2020-10-09
11	11	Speed Boat	52	old	2020-11-09
12	12	Rowing Boats	33	need to change	2020-12-09
13	13	First Aid Kits	23	new	2018-01-19
14	14	Harness	39	old	2020-02-02
15	15	Lifebuoys	10	need to change	2017-03-05
16	16	Distress Signal	45	new	2020-04-09
17	17	Dry suit	42	old	2020-05-09
18	18	Snorkel Mask	30	need to change	2019-06-09

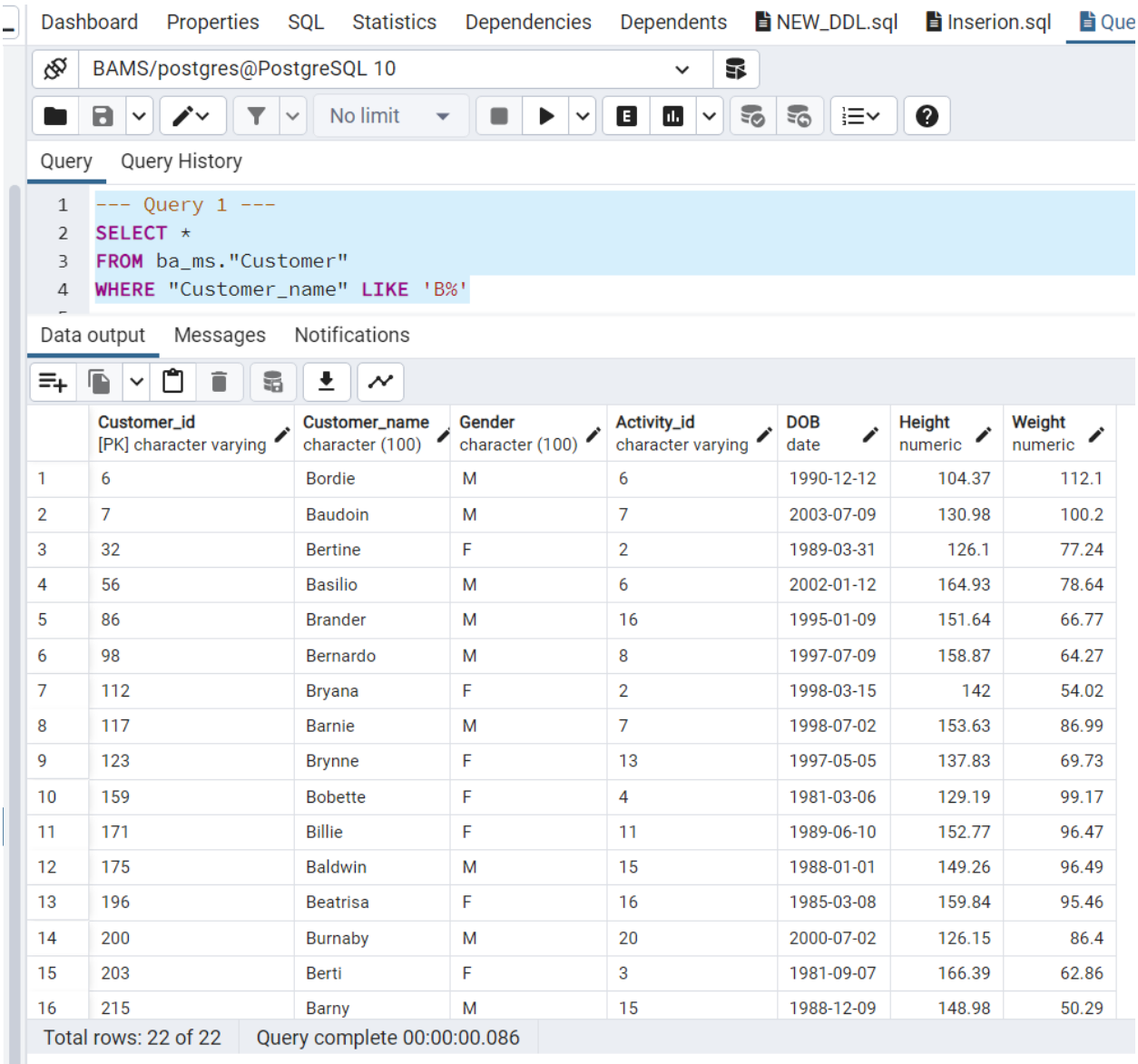
Total rows: 30 of 30 Query complete 00:00:00.153

21. SQL Queries

Q1: To find all customer details whose name starts with ‘B’.

Ans:

```
--- Query 1 ---
SELECT *
FROM ba_ms."Customer"
WHERE "Customer_name" LIKE 'B%'
```

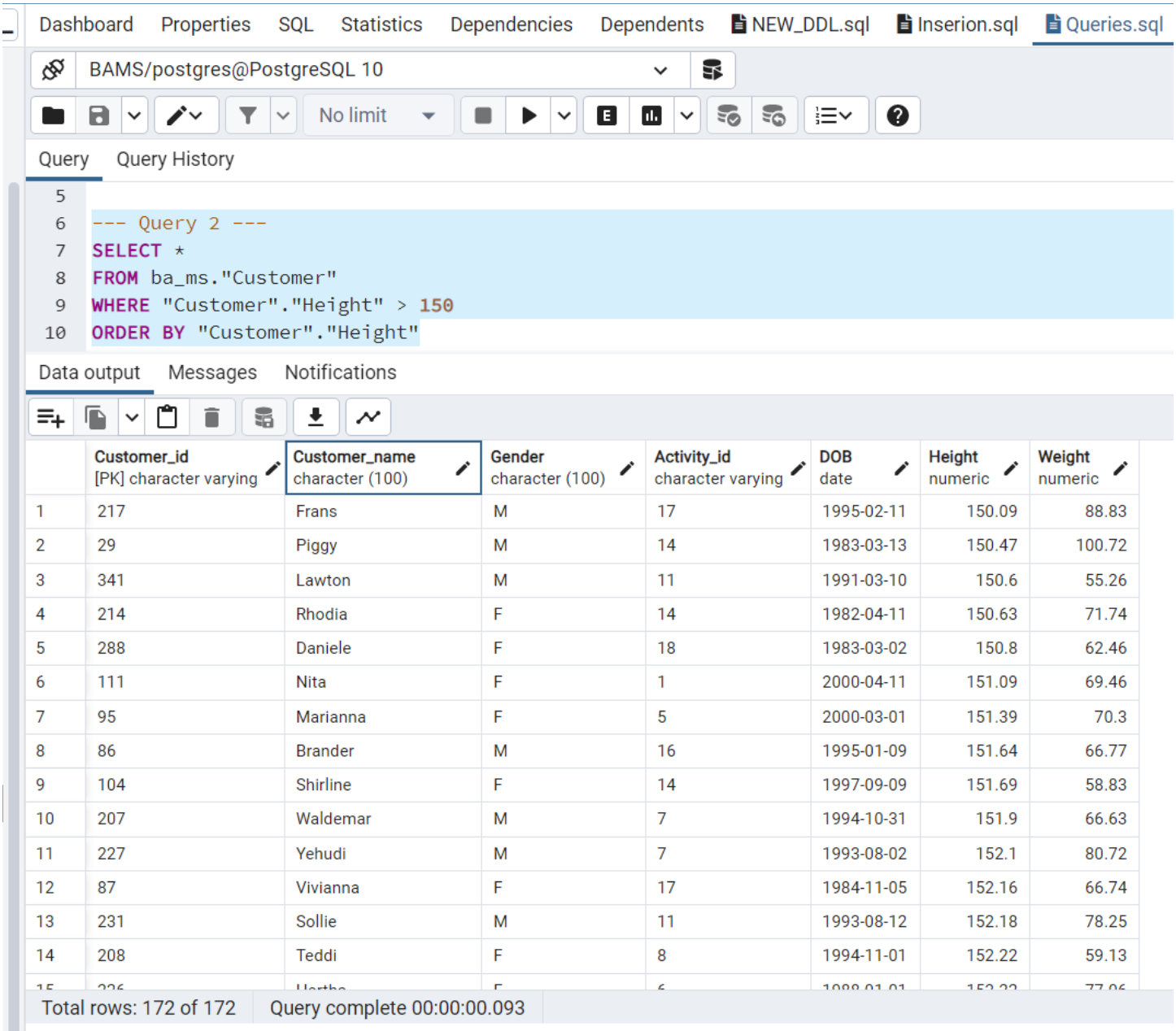


	Customer_id [PK] character varying	Customer_name character (100)	Gender character (100)	Activity_id character varying	DOB date	Height numeric	Weight numeric
1	6	Bordie	M	6	1990-12-12	104.37	112.1
2	7	Baudoin	M	7	2003-07-09	130.98	100.2
3	32	Bertine	F	2	1989-03-31	126.1	77.24
4	56	Basilio	M	6	2002-01-12	164.93	78.64
5	86	Brander	M	16	1995-01-09	151.64	66.77
6	98	Bernardo	M	8	1997-07-09	158.87	64.27
7	112	Bryana	F	2	1998-03-15	142	54.02
8	117	Barnie	M	7	1998-07-02	153.63	86.99
9	123	Brynne	F	13	1997-05-05	137.83	69.73
10	159	Bobette	F	4	1981-03-06	129.19	99.17
11	171	Billie	F	11	1989-06-10	152.77	96.47
12	175	Baldwin	M	15	1988-01-01	149.26	96.49
13	196	Beatrisa	F	16	1985-03-08	159.84	95.46
14	200	Burnaby	M	20	2000-07-02	126.15	86.4
15	203	Berti	F	3	1981-09-07	166.39	62.86
16	215	Barny	M	15	1988-12-09	148.98	50.29

Q2: To find all customers whose height exceeds 150 cm.

Ans:

```
--- Query 2 ---
SELECT *
FROM ba_ms."Customer"
WHERE "Customer"."Height" > 150
ORDER BY "Customer"."Height"
```

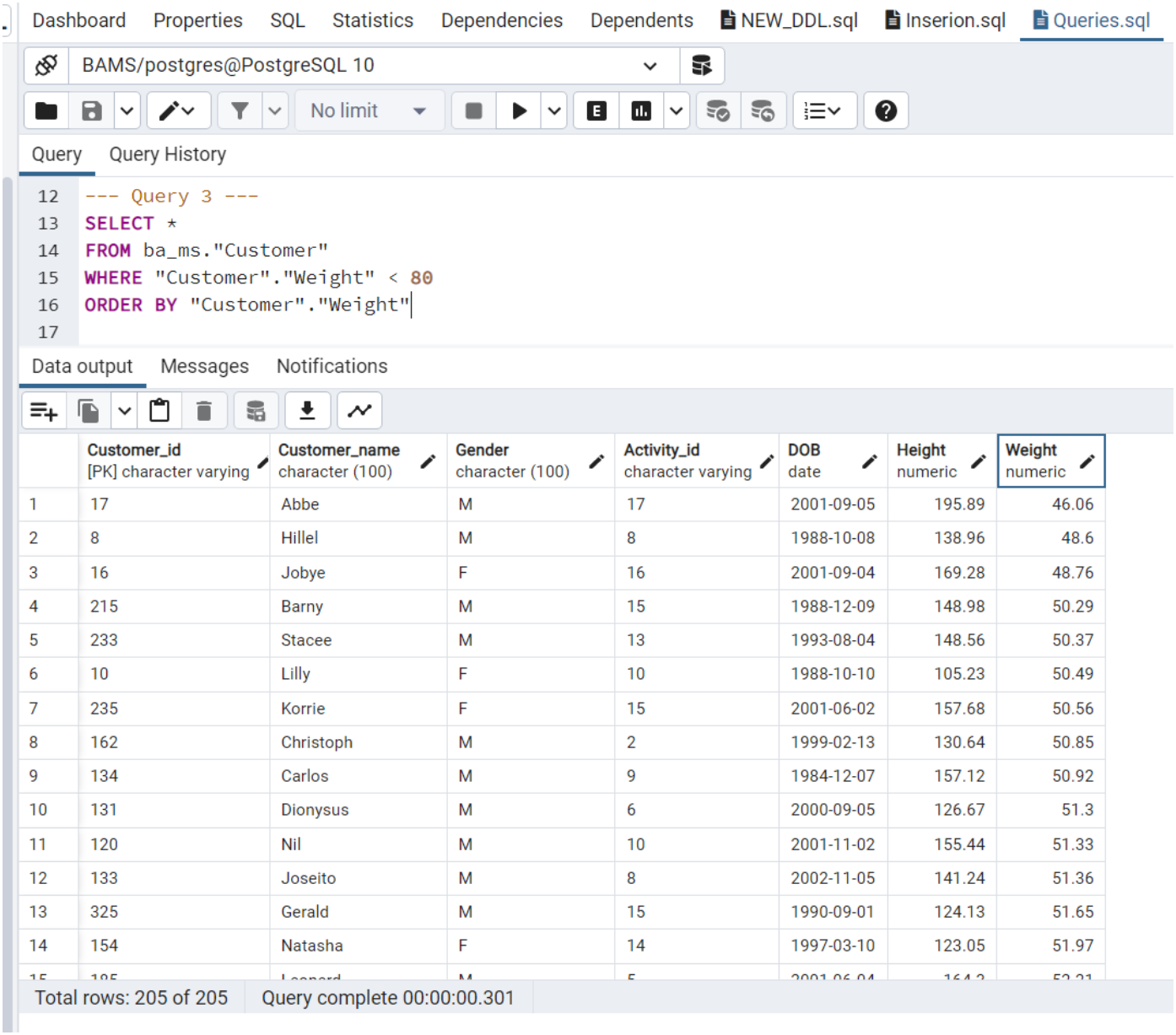


	Customer_id [PK] character varying	Customer_name character (100)	Gender character (100)	Activity_id character varying	DOB date	Height numeric	Weight numeric
1	217	Frans	M	17	1995-02-11	150.09	88.83
2	29	Piggy	M	14	1983-03-13	150.47	100.72
3	341	Lawton	M	11	1991-03-10	150.6	55.26
4	214	Rhodia	F	14	1982-04-11	150.63	71.74
5	288	Daniele	F	18	1983-03-02	150.8	62.46
6	111	Nita	F	1	2000-04-11	151.09	69.46
7	95	Marianna	F	5	2000-03-01	151.39	70.3
8	86	Brander	M	16	1995-01-09	151.64	66.77
9	104	Shirline	F	14	1997-09-09	151.69	58.83
10	207	Waldemar	M	7	1994-10-31	151.9	66.63
11	227	Yehudi	M	7	1993-08-02	152.1	80.72
12	87	Vivianna	F	17	1984-11-05	152.16	66.74
13	231	Sollie	M	11	1993-08-12	152.18	78.25
14	208	Teddi	F	8	1994-11-01	152.22	59.13

Q3: To find all customers whose weight is less than 80 Kg.

Ans:

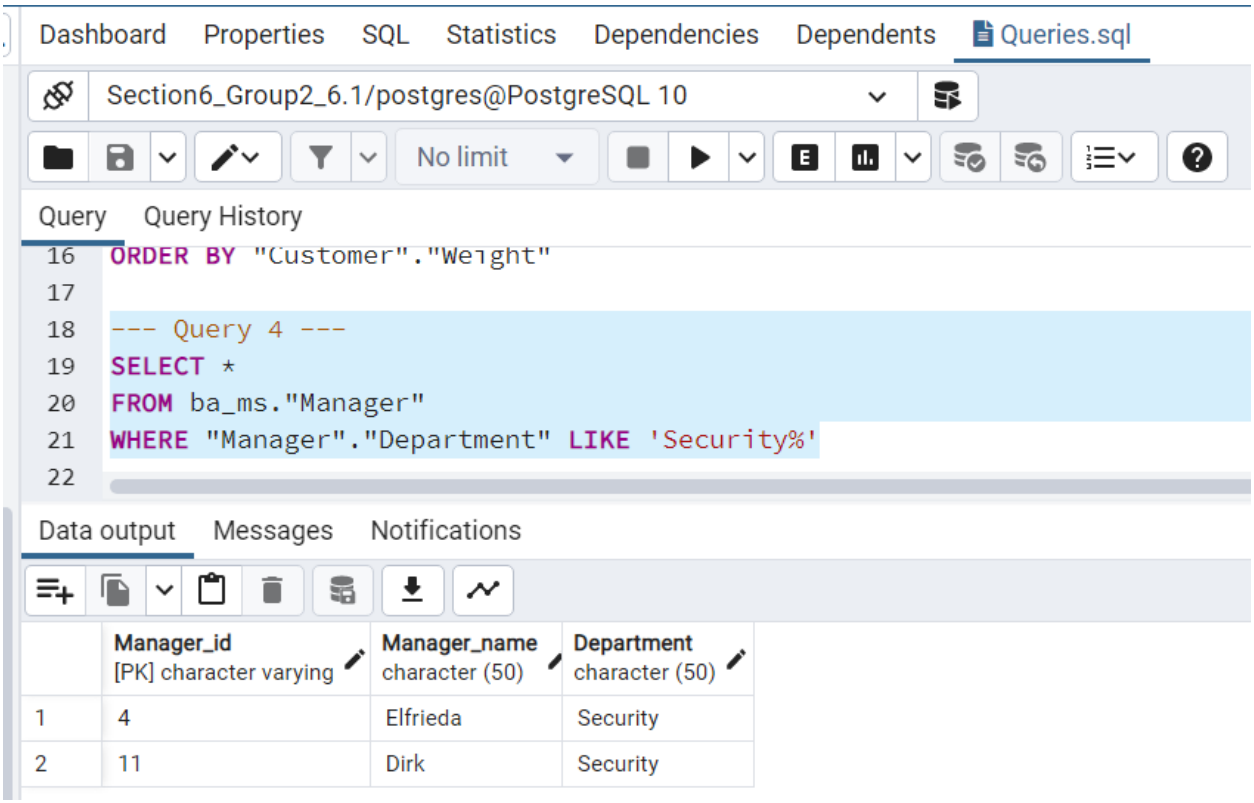
```
--- Query 3 ---
SELECT *
FROM ba_ms."Customer"
WHERE "Customer"."Weight" < 80
ORDER BY "Customer"."Weight"
```



Q4: To find all managers who work in the security department.

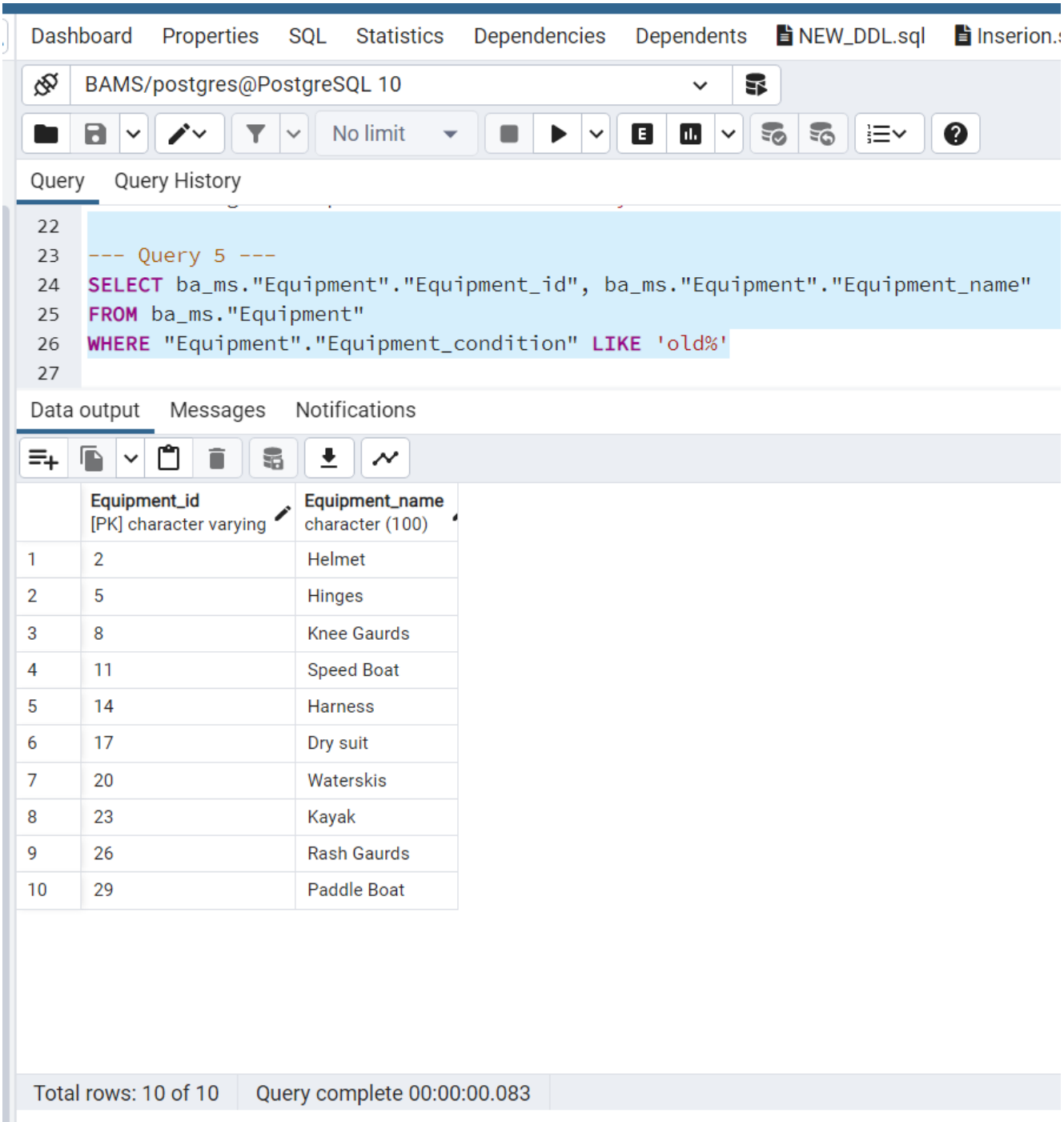
Ans:

```
--- Query 4 ---
SELECT *
FROM ba_ms."Manager"
WHERE "Manager"."Department" LIKE 'Security%'
```



Q5: Display Equipment_id and Equipment_name whose condition is old.
Ans:

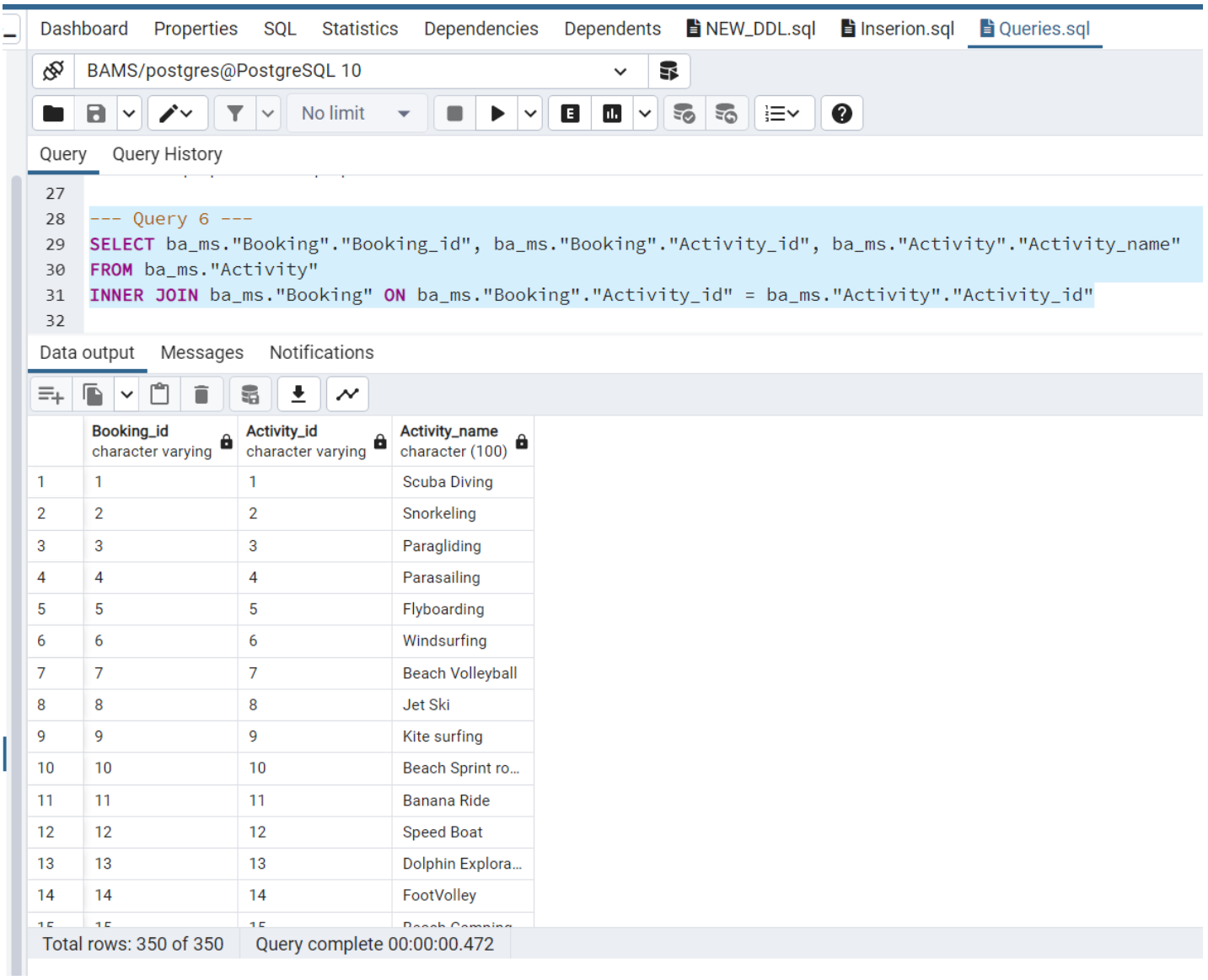
```
--- Query 5 ---
SELECT ba_ms."Equipment"."Equipment_id", ba_ms."Equipment"."Equipment_name"
FROM ba_ms."Equipment"
WHERE "Equipment"."Equipment_condition" LIKE 'old%'
```



Q6: Display all the activity names corresponding to all the bookings made.

Ans:

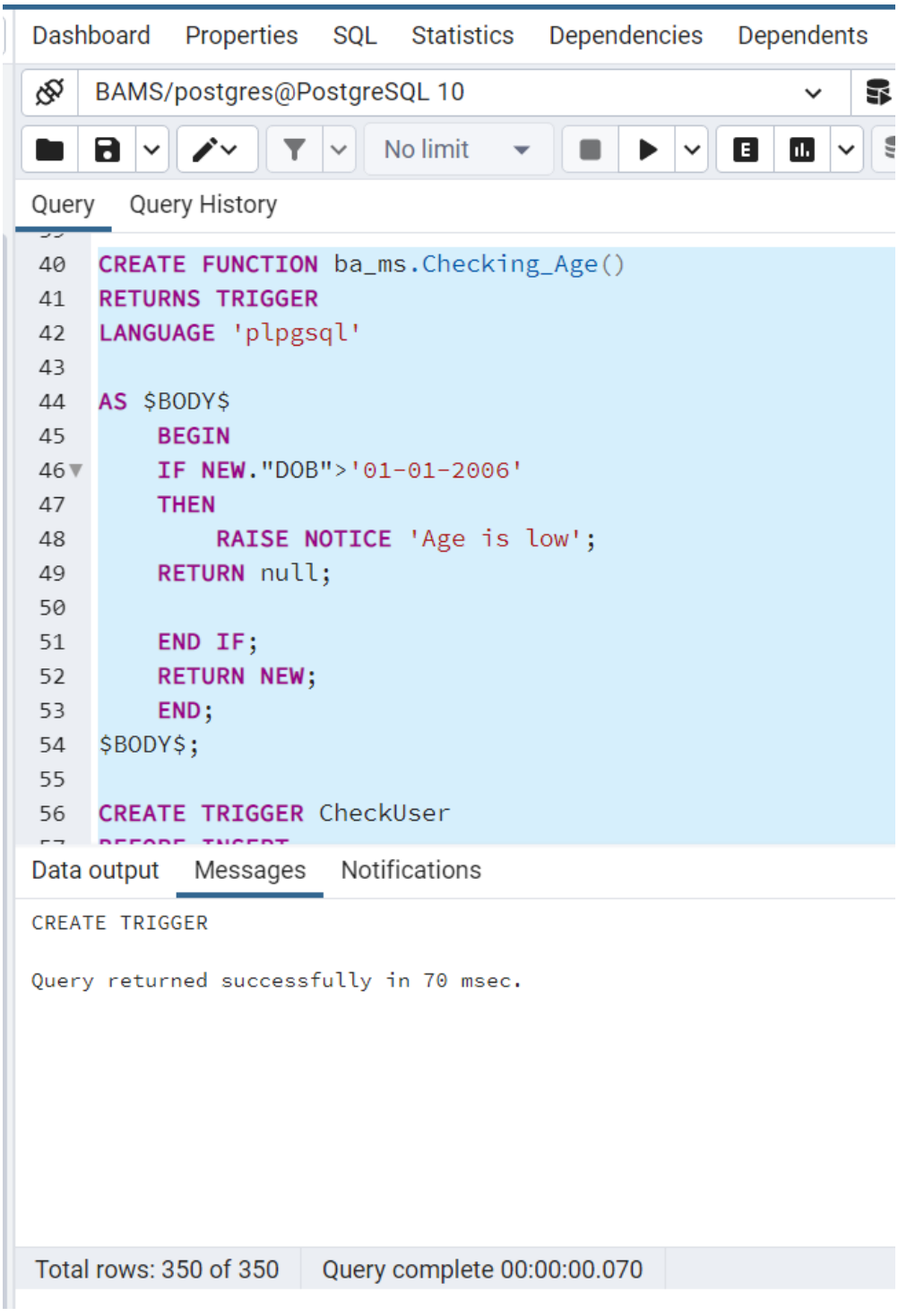
```
--- Query 6 ---
SELECT ba_ms."Booking"."Booking_id", ba_ms."Booking"."Activity_id",
ba_ms."Activity"."Activity_name"
FROM ba_ms."Activity"
INNER JOIN ba_ms."Booking" ON ba_ms."Booking"."Activity_id" =
ba_ms."Activity"."Activity_id"
```



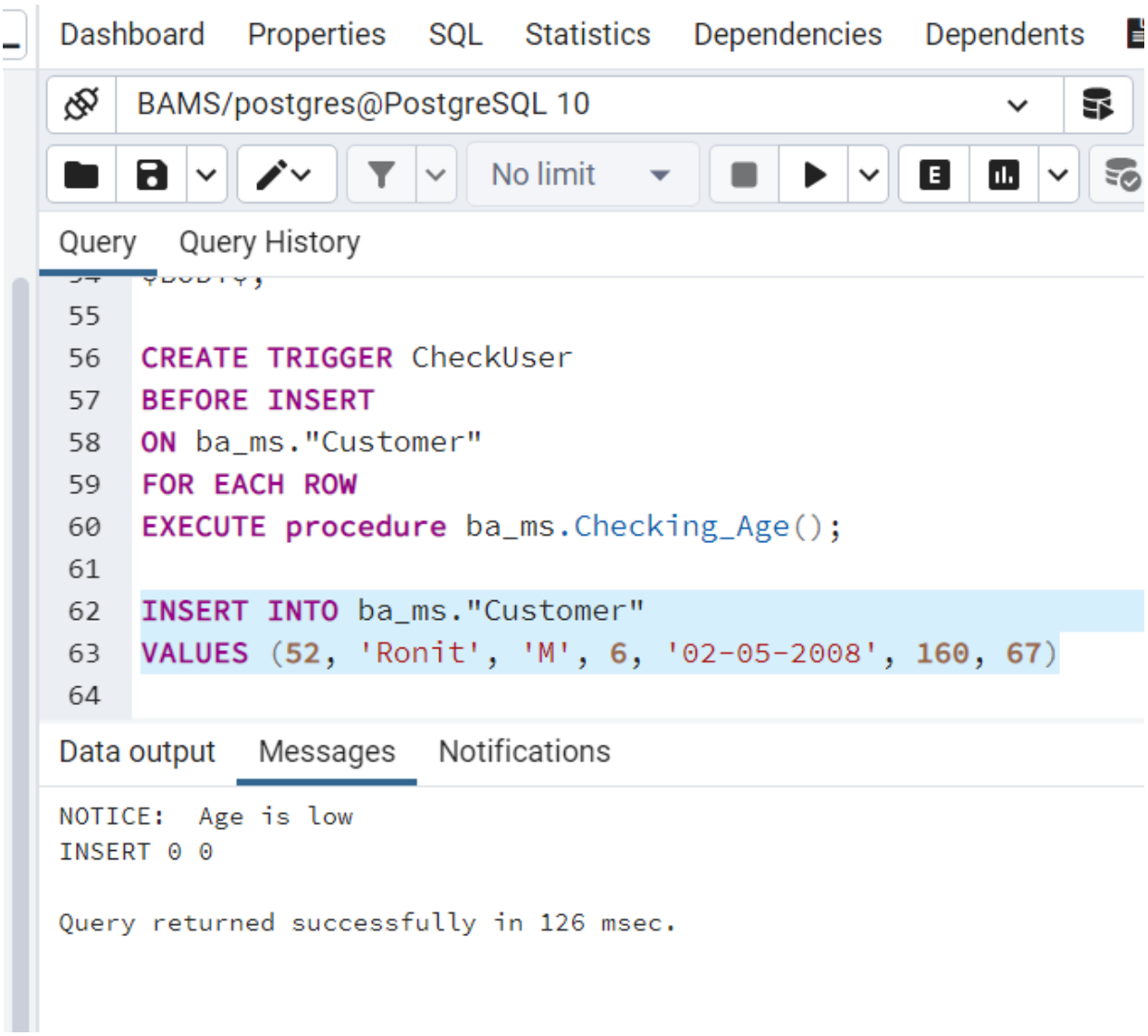
Q7: <u>Trigger:</u> This trigger helps us when any customer is below the required age then they can't insert their data in customer details.

Ans:

```
--- Query 7 (Trigger) ---

CREATE FUNCTION ba_ms.Checking_Age()
RETURNS TRIGGER
LANGUAGE 'plpgsql'

AS $BODY$
    BEGIN
    IF NEW."DOB">'01-01-2006'
    THEN
            RAISE NOTICE 'Age is low';
    RETURN null;

    END IF;
    RETURN NEW;
    END;
$BODY$;

CREATE TRIGGER CheckUser
BEFORE INSERT
ON ba_ms."Customer"
FOR EACH ROW
EXECUTE procedure ba_ms.Checking_Age();
```


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
BAMS/postgres@PostgreSQL 10
No limit
Query
Query History
40 CREATE FUNCTION ba_ms.Checking_Age()
41 RETURNS TRIGGER
42 LANGUAGE 'plpgsql'
43
44 AS $BODY$
45 BEGIN
46 IF NEW."DOB">'01-01-2006'
47 THEN
48 RAISE NOTICE 'Age is low';
49 RETURN null;
50
51 END IF;
52 RETURN NEW;
53 END;
54 $BODY$;
55
56 CREATE TRIGGER CheckUser
Data output
Messages
Notifications
CREATE TRIGGER
Query returned successfully in 70 msec.
Total rows: 350 of 350
Query complete 00:00:00.070

INSERT INTO ba_ms."Customer"
VALUES (52, 'Ronit', 'M', 6, '02-05-2008', 160, 67)



Q8: Function: This function returns the activity id of the activity who has maximum number of bookings.

Ans:

```
--- Query 8 (Function) ----

CREATE OR REPLACE FUNCTION ba_ms."Find_Max"()
RETURNS TABLE(ActivityID character varying)
LANGUAGE'plpgsql'
AS $body$
    DECLARE
    list_item record;
    max_cnt integer;

    BEGIN
            SELECT MAX(T.count) INTO max_cnt
            FROM (SELECT COUNT(*)
                    FROM ba_ms."Booking"
                    GROUP BY "Activity_id") AS T;

            CREATE TEMP TABLE Temp_Table(ActivityID character varying) ON COMMIT
DROP;

            FOR list_item IN (SELECT "Activity_id"
                                        FROM (SELECT "Activity_id", COUNT(*) AS cnt
                                                FROM ba_ms."Booking"
                                                GROUP BY "Activity_id") AS T
                                        WHERE T.cnt=max_cnt)

            LOOP
                INSERT INTO Temp_Table (ActivityId)
                VALUES (list_item."Activity_id");
            END LOOP;

            RETURN QUERY TABLE Temp_Table;
    END;
$body$;
```

Dashboard Properties SQL Statistics Dependencies Dependents NEW_DDL.sql Inserion.sql Qu

BAMS/postgres@PostgreSQL 10

No limit

Query Query History

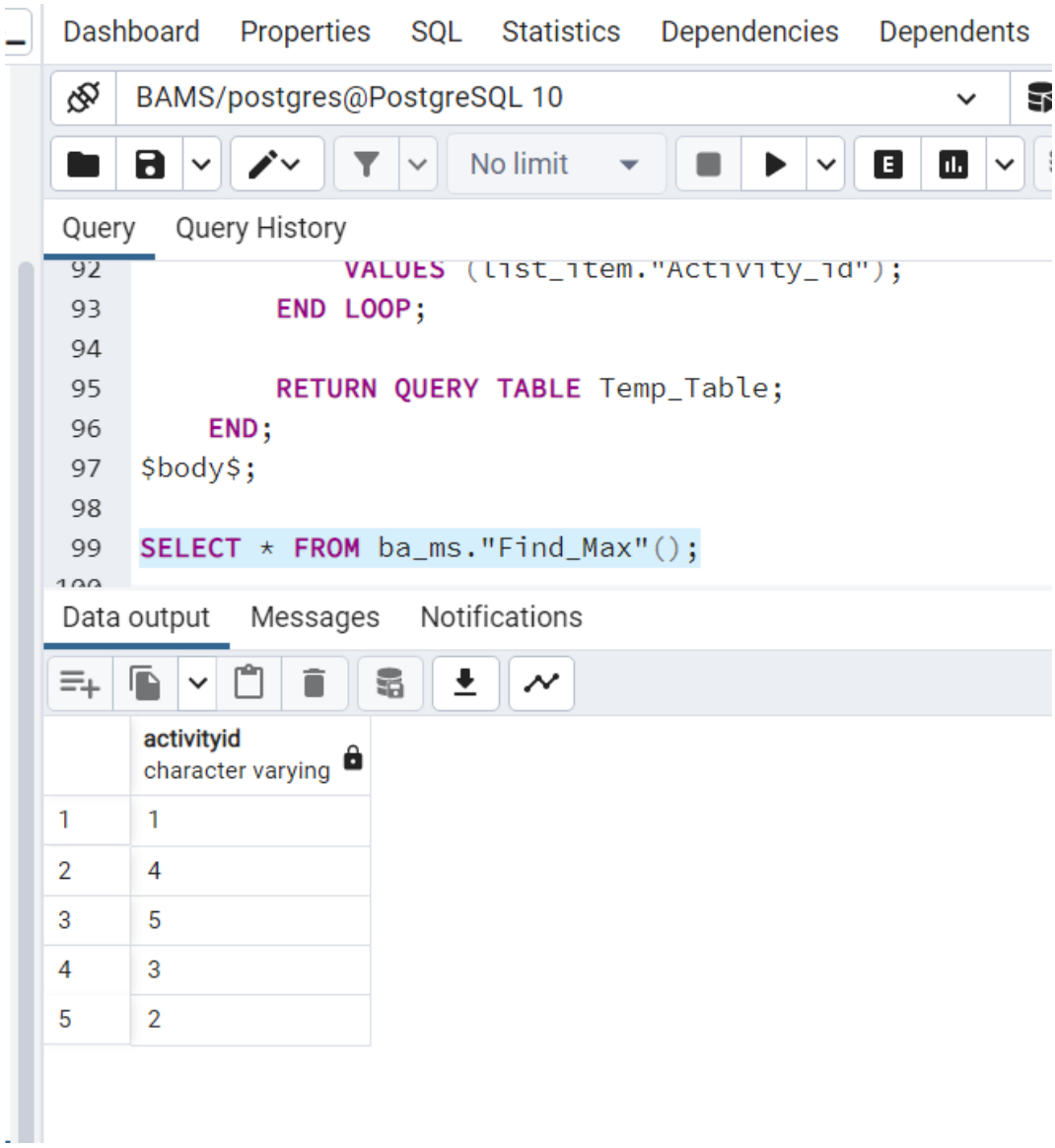
```
67
68  CREATE OR REPLACE FUNCTION ba_ms."Find_Max"()
69  RETURNS TABLE(ActivityID character varying)
70  LANGUAGE'plpgsql'
71  AS $body$
72      DECLARE
73      list_item record;
74      max_cnt integer;
75
76      BEGIN
77          SELECT MAX(T.count) INTO max_cnt
78          FROM (SELECT COUNT(*)
79                FROM ba_ms."Booking"
80                GROUP BY "Activity_id") AS T;
81
82          CREATE TEMP TABLE Temp_Table(ActivityID character varying) ON COMMIT DROP;
83
84          FOR list_item IN (SELECT "Activity_id"
85                            FROM (SELECT "Activity_id", COUNT(*) AS cnt
86                                  FROM ba_ms."Booking"
87                                  GROUP BY "Activity_id") AS T
88                            WHERE T.cnt=max_cnt)
```

Data output Messages Notifications

```
CREATE FUNCTION

Query returned successfully in 199 msec.
```

Total rows: 350 of 350 Query complete 00:00:00.199

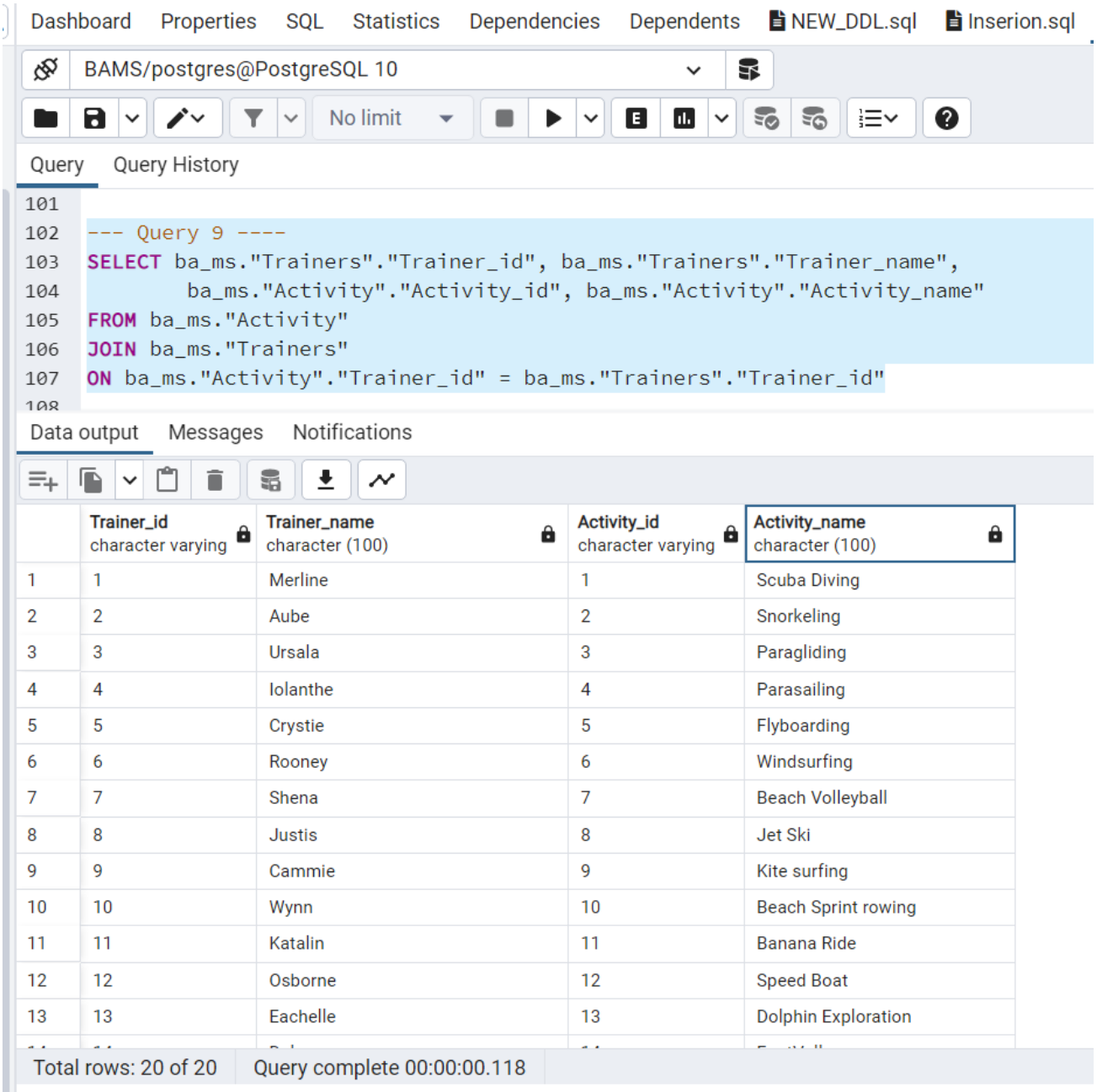
select * from ba_ms."Find_Max"();



Q9: Print names of trainers associated with corresponding activity.

Ans:

```
--- Query 9 ----
SELECT ba_ms."Trainers"."Trainer_id", ba_ms."Trainers"."Trainer_name",
        ba_ms."Activity"."Activity_id", ba_ms."Activity"."Activity_name"
FROM ba_ms."Activity"
JOIN ba_ms."Trainers"
ON ba_ms."Activity"."Trainer_id" = ba_ms."Trainers"."Trainer_id"
```

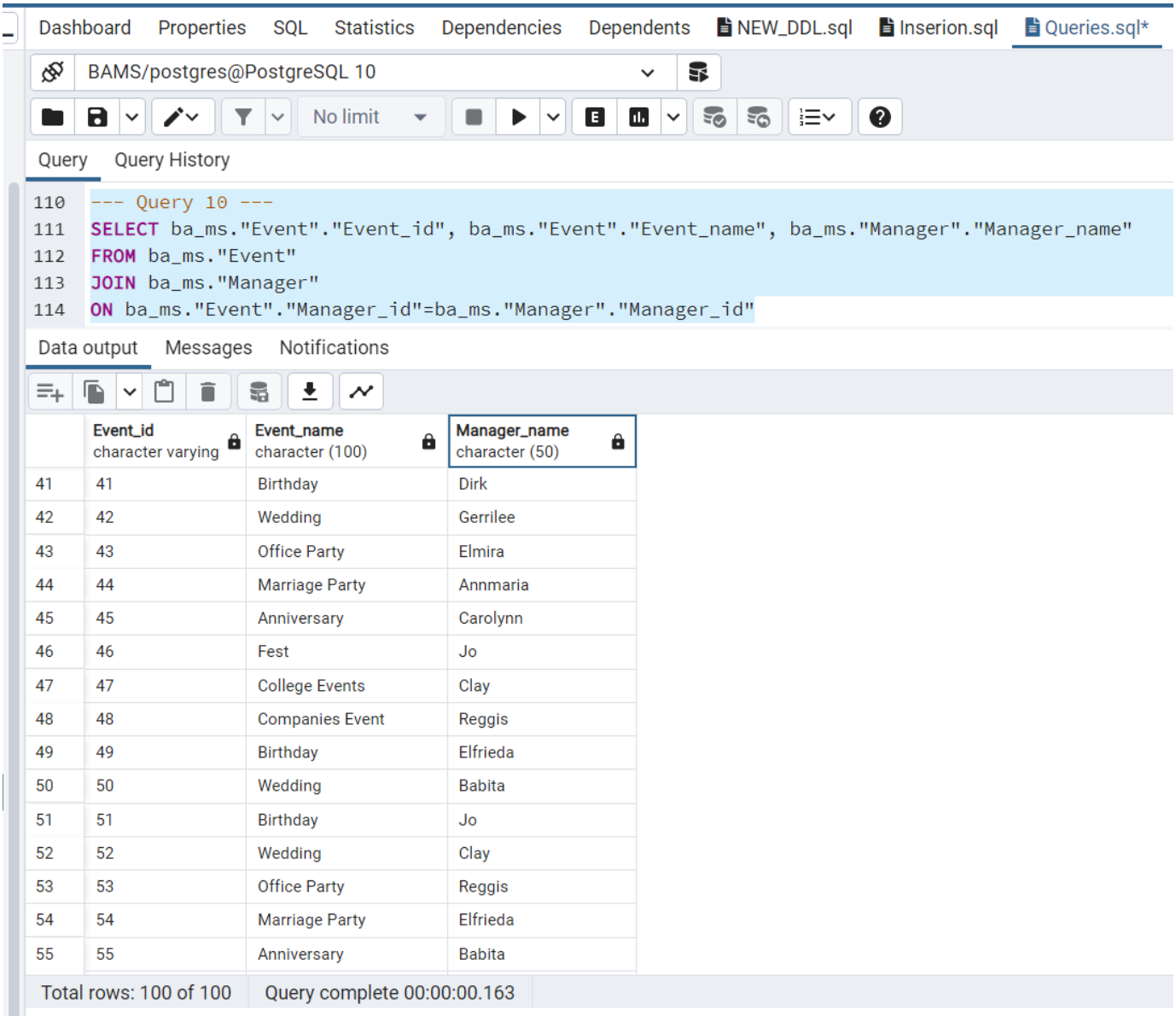


	Trainer_id character varying	Trainer_name character (100)	Activity_id character varying	Activity_name character (100)
1	1	Merline	1	Scuba Diving
2	2	Aube	2	Snorkeling
3	3	Ursala	3	Paragliding
4	4	Iolanthe	4	Parasailing
5	5	Crystie	5	Flyboarding
6	6	Rooney	6	Windsurfing
7	7	Shena	7	Beach Volleyball
8	8	Justis	8	Jet Ski
9	9	Cammie	9	Kite surfing
10	10	Wynn	10	Beach Sprint rowing
11	11	Katalin	11	Banana Ride
12	12	Osborne	12	Speed Boat
13	13	Eachelle	13	Dolphin Exploration

Q10: Print manager name associated with event id.

Ans:

```
--- Query 10 ---
SELECT ba_ms."Event"."Event_id", ba_ms."Event"."Event_name",
ba_ms."Manager"."Manager_name"
FROM ba_ms."Event"
JOIN ba_ms."Manager"
ON ba_ms."Event"."Manager_id"=ba_ms."Manager"."Manager_id"
```



Q11: Print names of staff members from equipment.

Ans:

```
--- Query 11 ---
Select *
FROM ba_ms."Staff"
WHERE ba_ms."Staff"."Department" Like 'Equipment%';
```

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Queries.sql*

Section6_Group2_6.1/postgres@PostgreSQL 10

No limit

Query | Query History

```
98
99  --- Query 11 ---
100 Select *
101 FROM ba_ms."Staff"
102 WHERE ba_ms."Staff"."Department" Like 'Equipment%';
103
104 --- Query 12 ---
105 SELECT ba_ms."Activity"."Activity_name", ba_ms."Activity"."Activity_time(in min)"
106 FROM ba_ms."Activity"
107 WHERE ba_ms."Activity"."Activity_time(in min)" >= 60
```

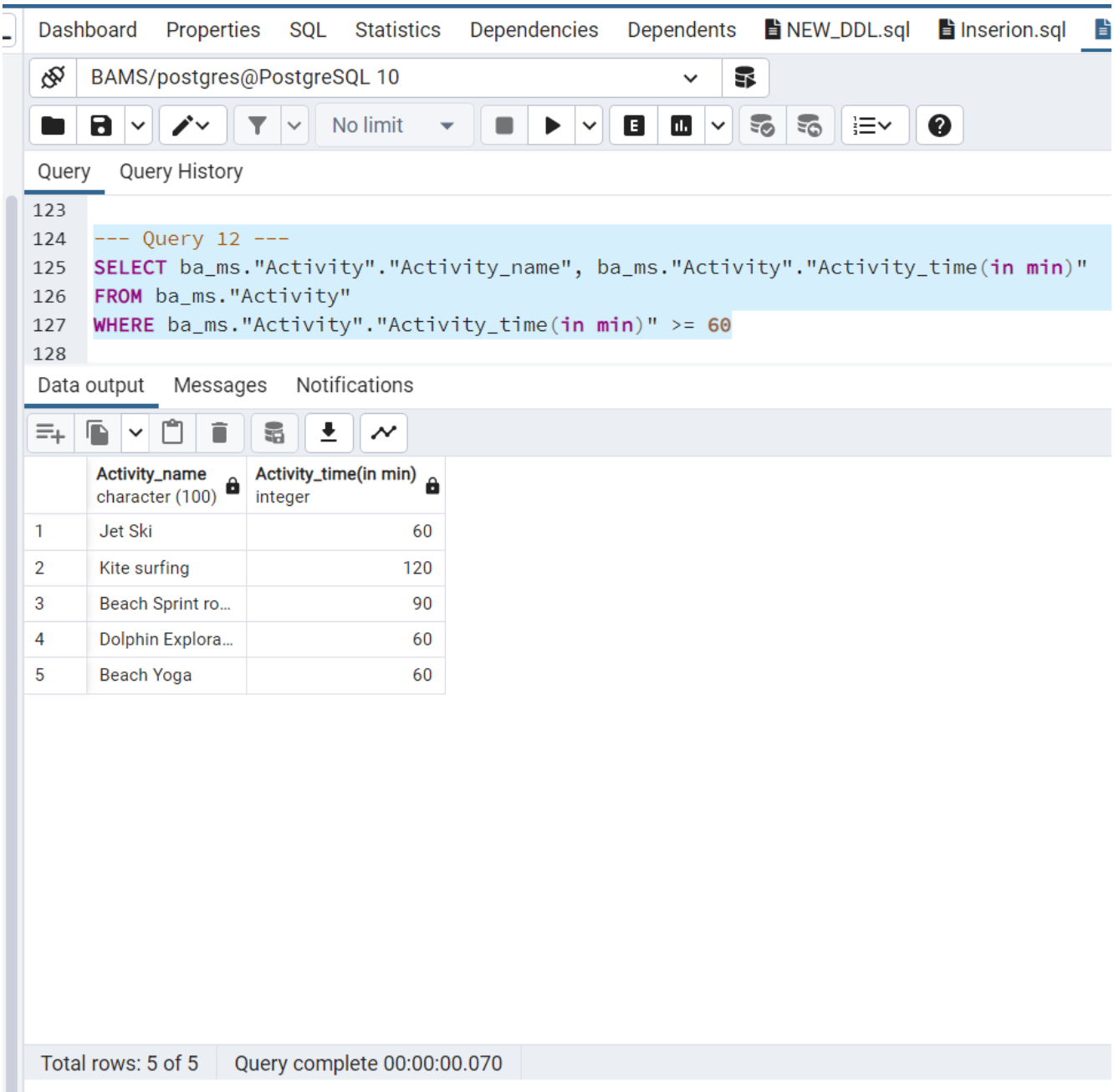
Data output | Messages | Notifications

	Staff_id [PK] character varying	Staff_name character (100)	Department character (100)
1	2	Melosa	Equipment
2	9	Tamiko	Equipment
3	16	Mimi	Equipment
4	23	Danika	Equipment
5	30	Georg	Equipment
6	37	Emmie	Equipment
7	44	Ky	Equipment

Total rows: 7 of 7 | Query complete 00:00:00.088

Q12: Print activity name whose activity time is greater or equal to 60 min.
Ans:

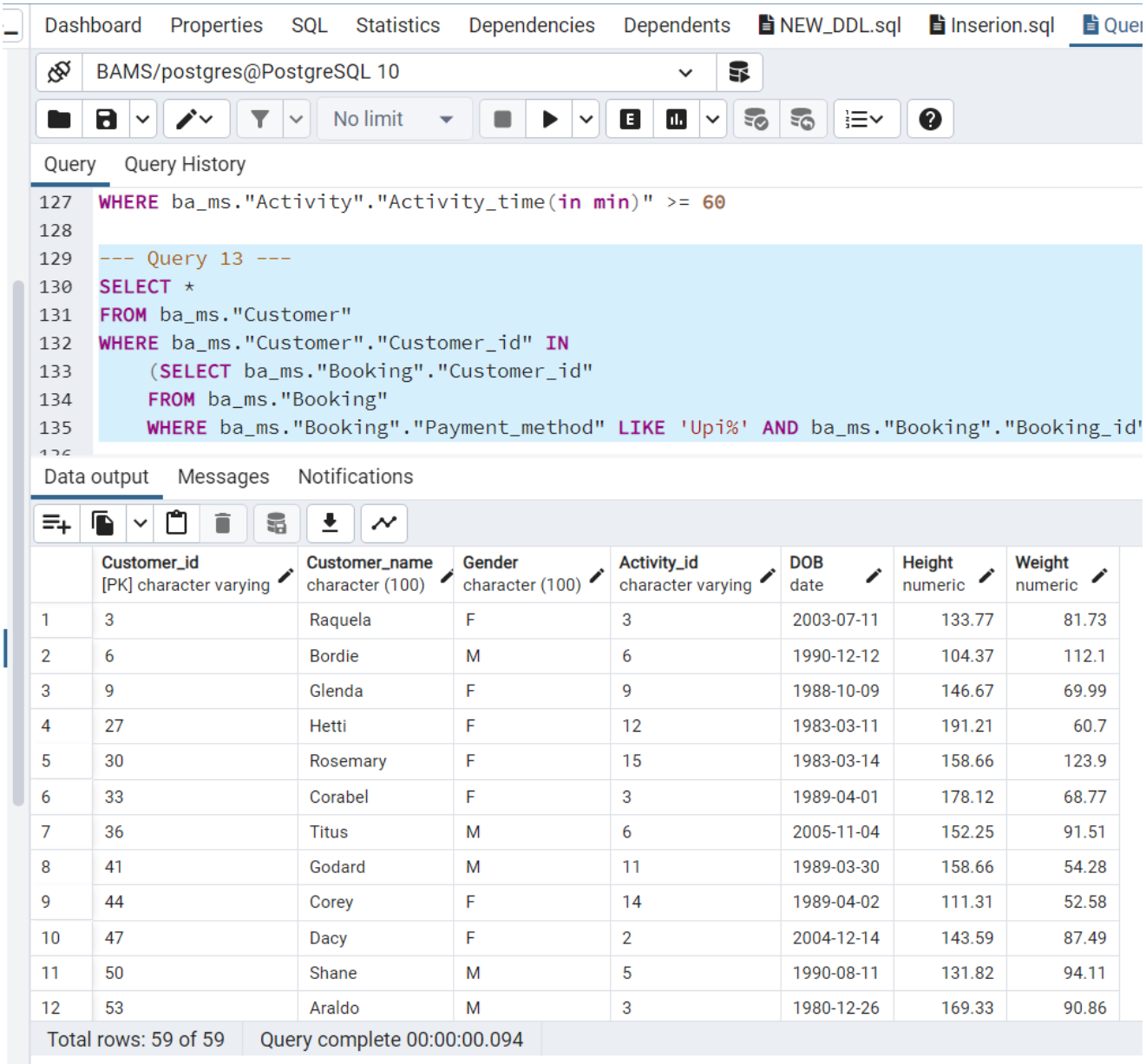
```
--- Query 12 ---
SELECT ba_ms."Activity"."Activity_name", ba_ms."Activity"."Activity_time(in min)"
FROM ba_ms."Activity"
WHERE ba_ms."Activity"."Activity_time(in min)" >= 60
```



Q13: Display the details of customers with booking IDs greater than 25 who have done payments by UPI.

Ans:

```
--- Query 13 ---
SELECT *
FROM ba_ms."Customer"
WHERE ba_ms."Customer"."Customer_id" IN
    (SELECT ba_ms."Booking"."Customer_id"
    FROM ba_ms."Booking"
    WHERE ba_ms."Booking"."Payment_method" LIKE 'Upi%' AND
ba_ms."Booking"."Booking_id" > '25')
```

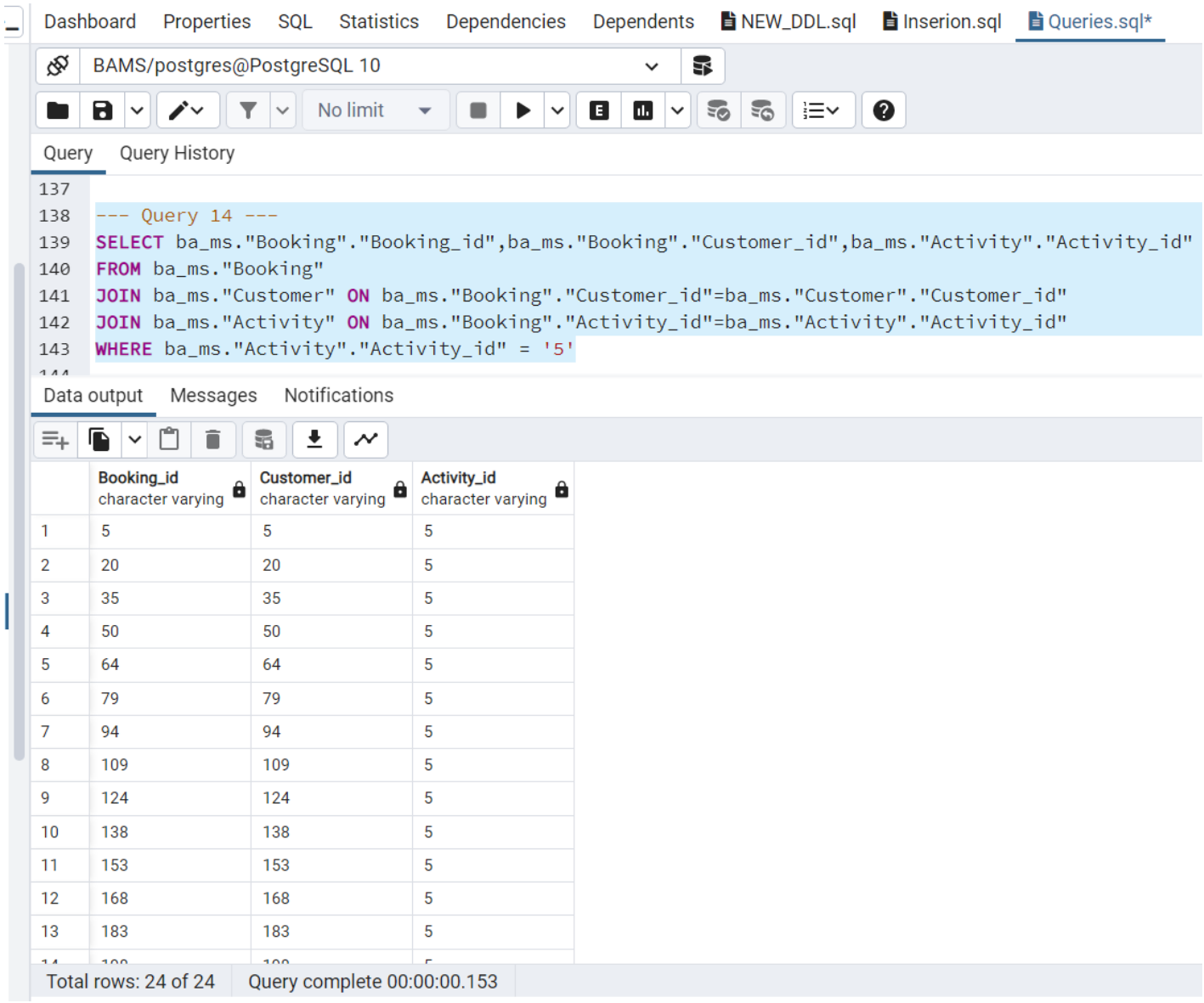


	Customer_id [PK] character varying	Customer_name character (100)	Gender character (100)	Activity_id character varying	DOB date	Height numeric	Weight numeric
1	3	Raquela	F	3	2003-07-11	133.77	81.73
2	6	Bordie	M	6	1990-12-12	104.37	112.1
3	9	Glenda	F	9	1988-10-09	146.67	69.99
4	27	Hetti	F	12	1983-03-11	191.21	60.7
5	30	Rosemary	F	15	1983-03-14	158.66	123.9
6	33	Corabel	F	3	1989-04-01	178.12	68.77
7	36	Titus	M	6	2005-11-04	152.25	91.51
8	41	Godard	M	11	1989-03-30	158.66	54.28
9	44	Corey	F	14	1989-04-02	111.31	52.58
10	47	Dacy	F	2	2004-12-14	143.59	87.49
11	50	Shane	M	5	1990-08-11	131.82	94.11
12	53	Araldo	M	3	1980-12-26	169.33	90.86

Q14: Display booking ID and customer ID for an activity ID = 5

Ans:

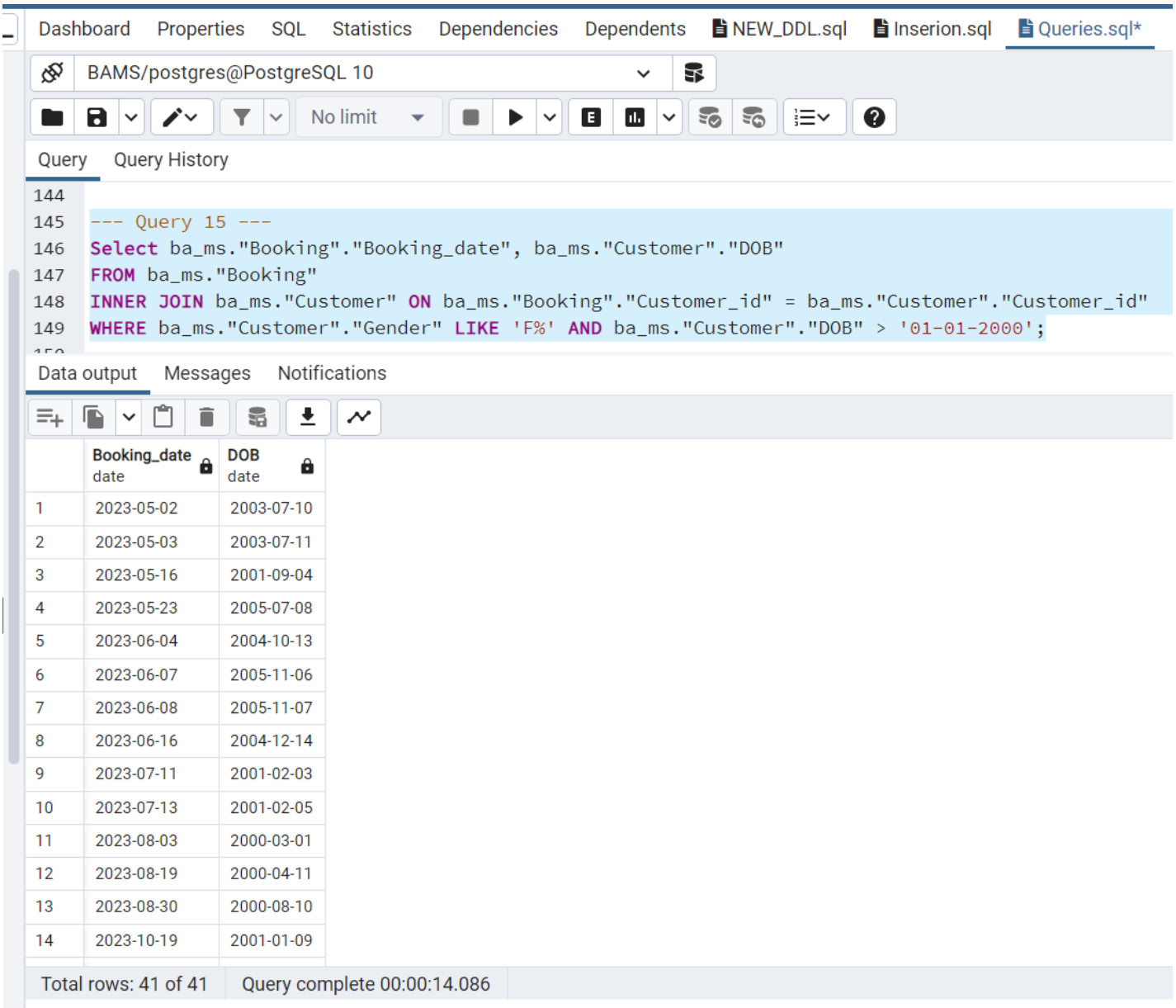
```
--- Query 14 ---
SELECT
ba_ms."Booking"."Booking_id",ba_ms."Booking"."Customer_id",ba_ms."Activity"."Activity_id"
FROM ba_ms."Booking"
JOIN ba_ms."Customer" ON
ba_ms."Booking"."Customer_id"=ba_ms."Customer"."Customer_id"
JOIN ba_ms."Activity" ON ba_ms."Booking"."Activity_id"=ba_ms."Activity"."Activity_id"
WHERE ba_ms."Activity"."Activity_id" = '5'
```



Q15: Find booking date of female customer whose DOB is greater than 01 - 01 - 2000 ?

Ans:

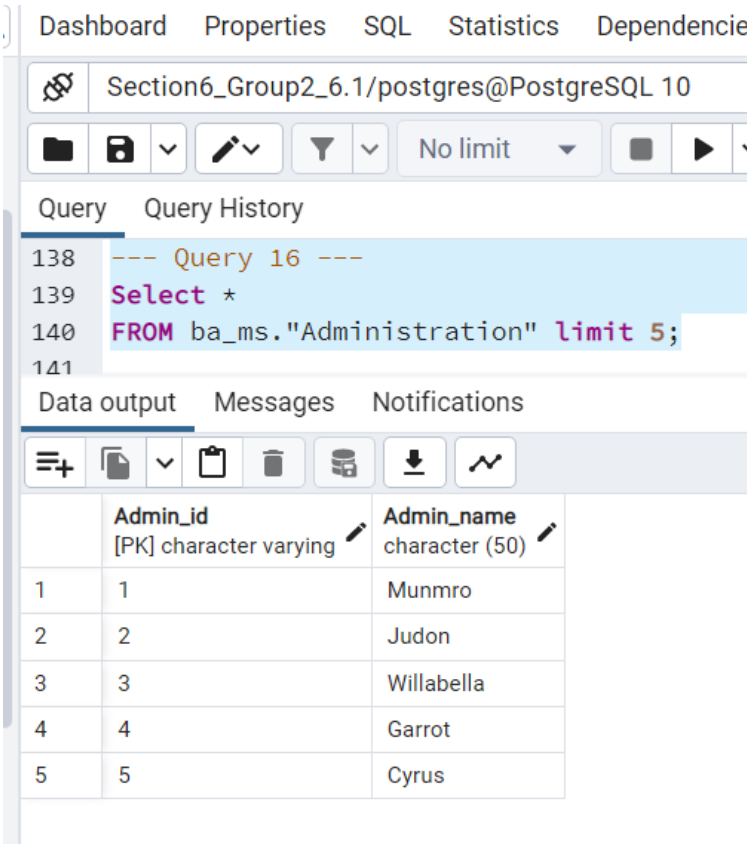
```
--- Query 15 ---
Select ba_ms."Booking"."Booking_date", ba_ms."Customer"."DOB"
FROM ba_ms."Booking"
INNER JOIN ba_ms."Customer" ON ba_ms."Booking"."Customer_id" =
ba_ms."Customer"."Customer_id"
WHERE ba_ms."Customer"."Gender" LIKE 'F%' AND ba_ms."Customer"."DOB" >
'01-01-2000';
```



Q16: Print the top 5 admin names.
Ans:

--- Query 16 ---

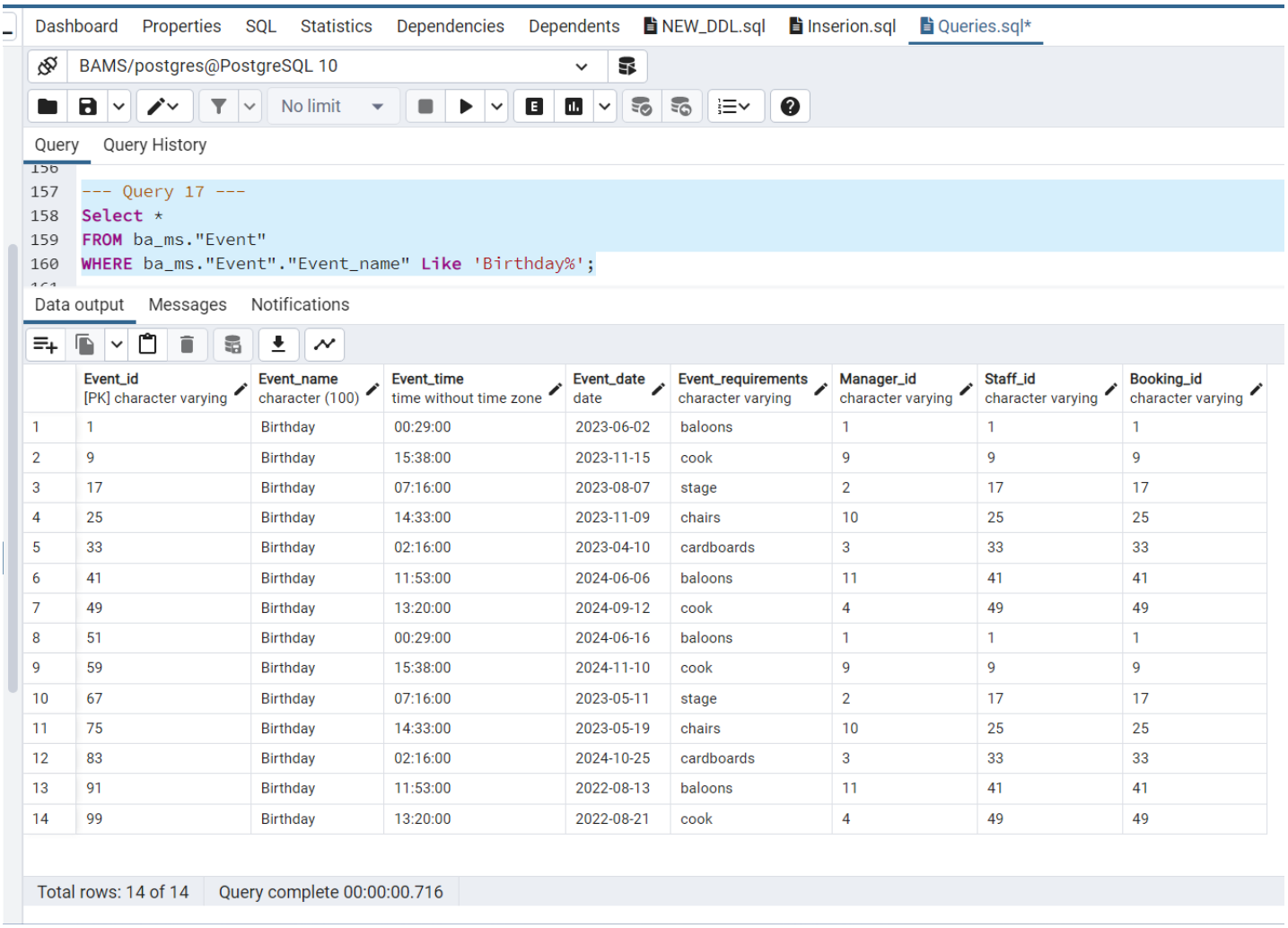
Select *
FROM ba_ms."Administration" limit 5;



Q17: Print details of all the birthday events.

Ans:

```
--- Query 17 ---
Select *
FROM ba_ms."Event"
WHERE ba_ms."Event"."Event_name" Like 'Birthday%';
```

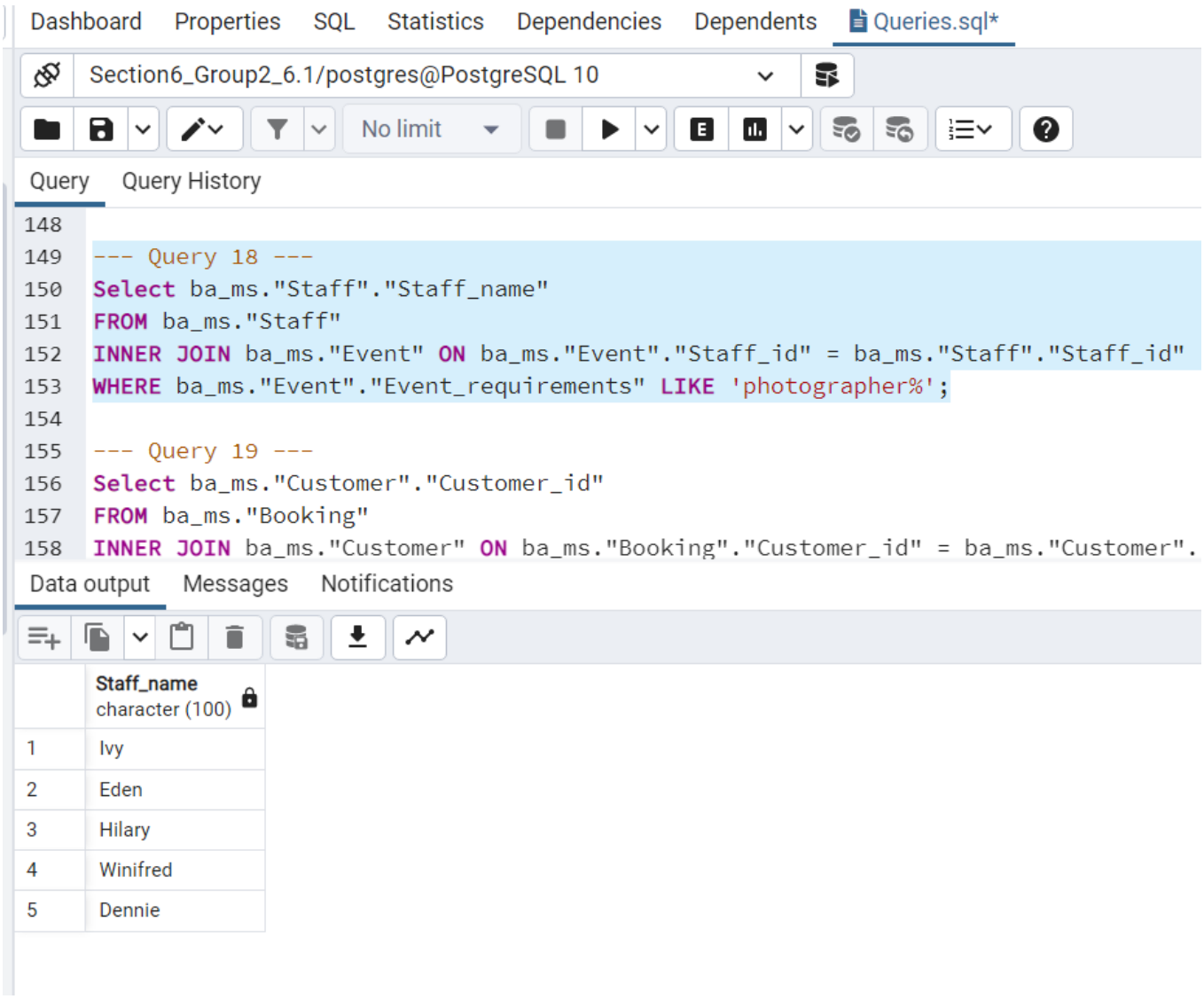


	Event_id [PK] character varying	Event_name character (100)	Event_time time without time zone	Event_date date	Event_requirements character varying	Manager_id character varying	Staff_id character varying	Booking_id character varying
1	1	Birthday	00:29:00	2023-06-02	baloons	1	1	1
2	9	Birthday	15:38:00	2023-11-15	cook	9	9	9
3	17	Birthday	07:16:00	2023-08-07	stage	2	17	17
4	25	Birthday	14:33:00	2023-11-09	chairs	10	25	25
5	33	Birthday	02:16:00	2023-04-10	cardboards	3	33	33
6	41	Birthday	11:53:00	2024-06-06	baloons	11	41	41
7	49	Birthday	13:20:00	2024-09-12	cook	4	49	49
8	51	Birthday	00:29:00	2024-06-16	baloons	1	1	1
9	59	Birthday	15:38:00	2024-11-10	cook	9	9	9
10	67	Birthday	07:16:00	2023-05-11	stage	2	17	17
11	75	Birthday	14:33:00	2023-05-19	chairs	10	25	25
12	83	Birthday	02:16:00	2024-10-25	cardboards	3	33	33
13	91	Birthday	11:53:00	2022-08-13	baloons	11	41	41
14	99	Birthday	13:20:00	2022-08-21	cook	4	49	49

Q18: Find staff_name whose event requirement is a photographer.

Ans:

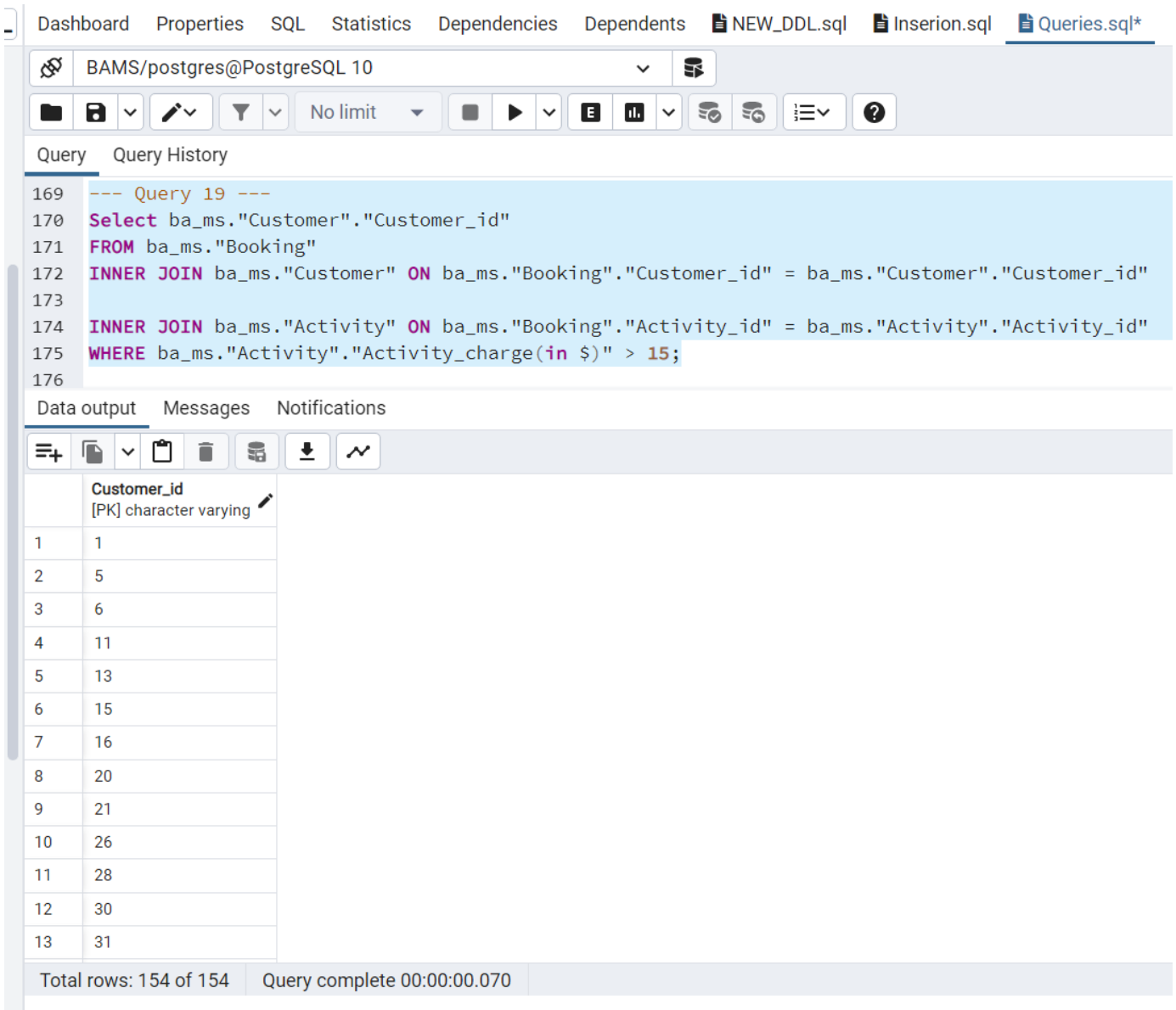
--- Query 18 ---
Select ba_ms."Staff"."Staff_name"
FROM ba_ms."Staff"
INNER JOIN ba_ms."Event" ON ba_ms."Event"."Staff_id" = ba_ms."Staff"."Staff_id"
WHERE ba_ms."Event"."Event_requirements" LIKE 'photographer%';



Q19: Find the customer_id of booking done more than $15.

Ans:

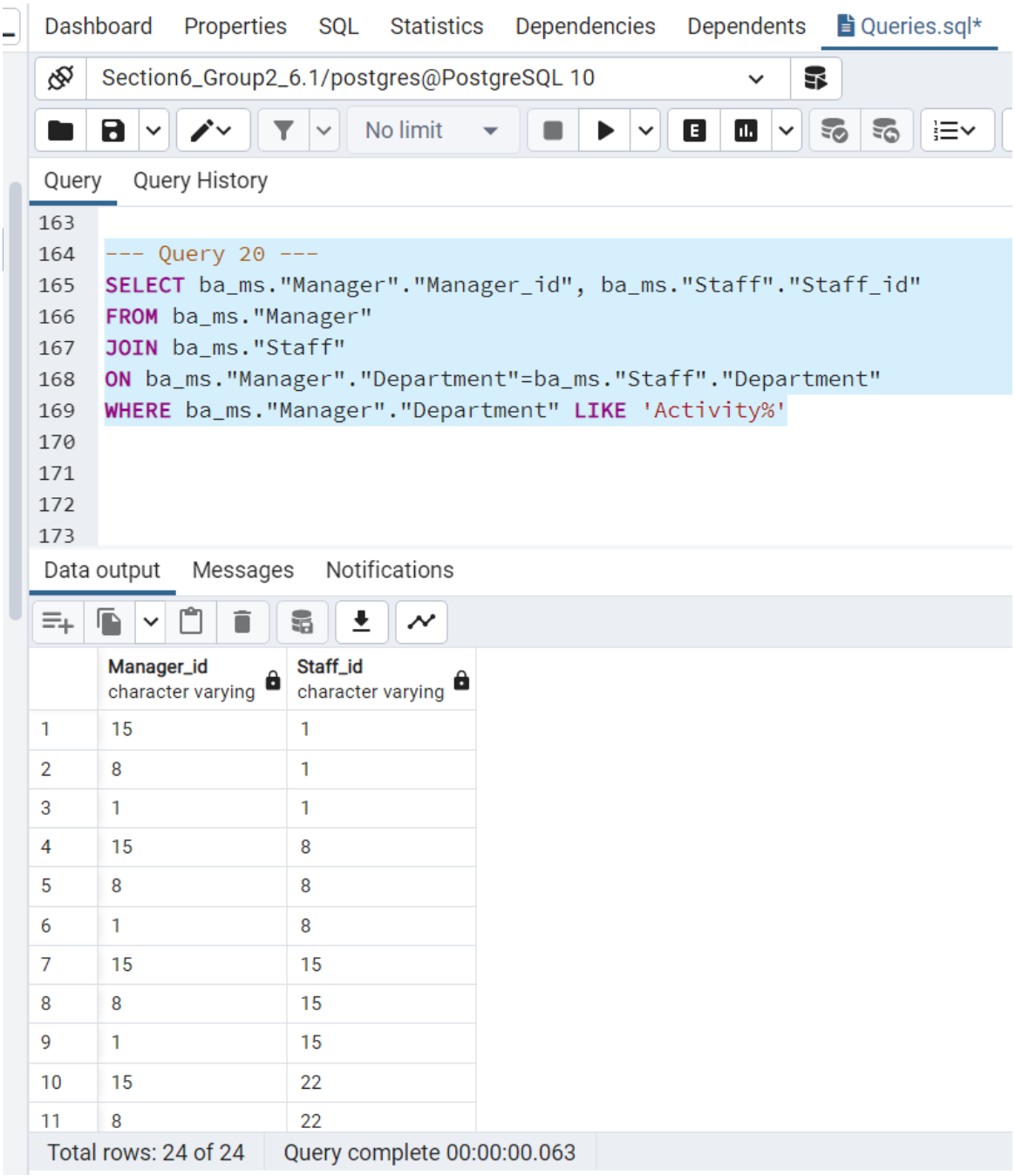
```
-- Query 19 --
Select ba_ms."Customer"."Customer_id"
FROM ba_ms."Booking"
INNER JOIN ba_ms."Customer" ON ba_ms."Booking"."Customer_id" =
ba_ms."Customer"."Customer_id"

INNER JOIN ba_ms."Activity" ON ba_ms."Booking"."Activity_id" =
ba_ms."Activity"."Activity_id"
WHERE ba_ms."Activity"."Activity_charge(in $)" > 15;
```



Q20: Print manager_id, staff_id where the department is activity.

Ans:

```
--- Query 20 ---
SELECT ba_ms."Manager"."Manager_id", ba_ms."Staff"."Staff_id"
FROM ba_ms."Manager"
JOIN ba_ms."Staff"
ON ba_ms."Manager"."Department"=ba_ms."Staff"."Department"
WHERE ba_ms."Manager"."Department" LIKE 'Activity%'
```



22. Front-End Development:

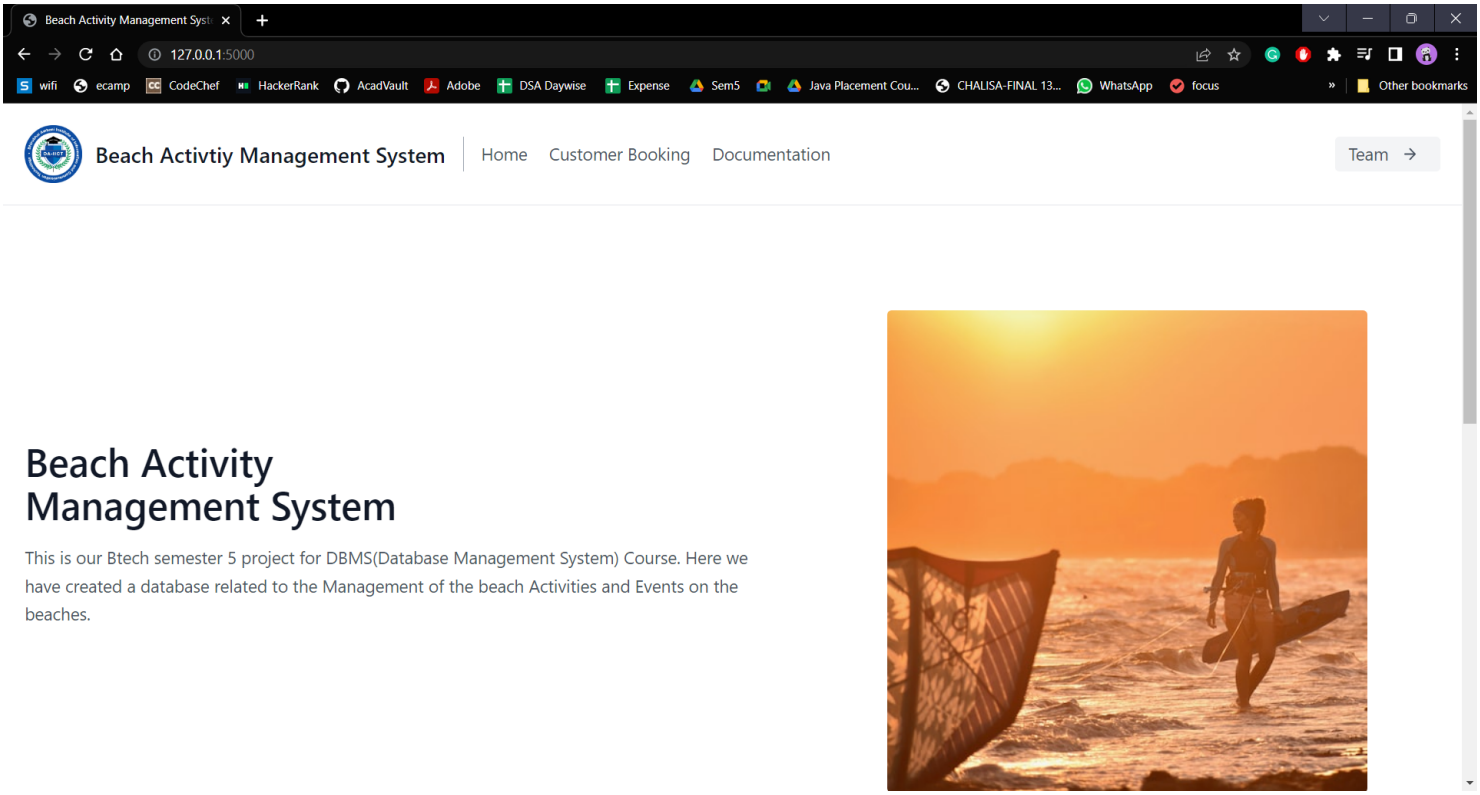
We are using python to connect our database. We have used Html code to create a webpage and CSS from a tailwind.

1. Connection Code:

```
from flask import Flask, render_template, request,session,redirect  ,jsonify
import psycopg2
from flask_session import Session
from json import dumps
from datetime import date
app = Flask(__name__)
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
app.static_folder='static'
Session(app)
conn = psycopg2.connect(host='localhost', database='bams',
                        user='postgres', password='ronit123')
cursor = conn.cursor()
user_id = 0
```

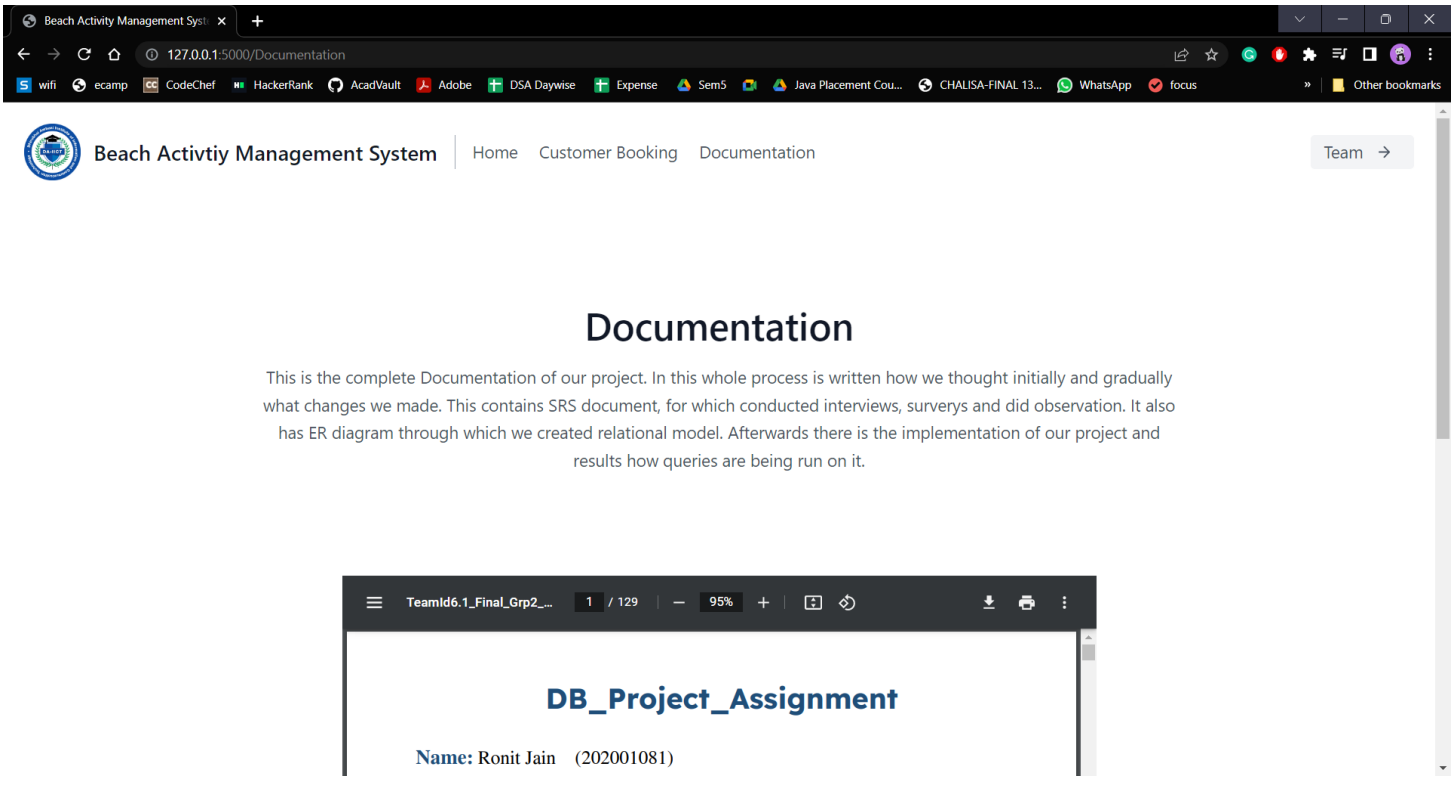
2. Functioning of Queries and Website:

Home Page:



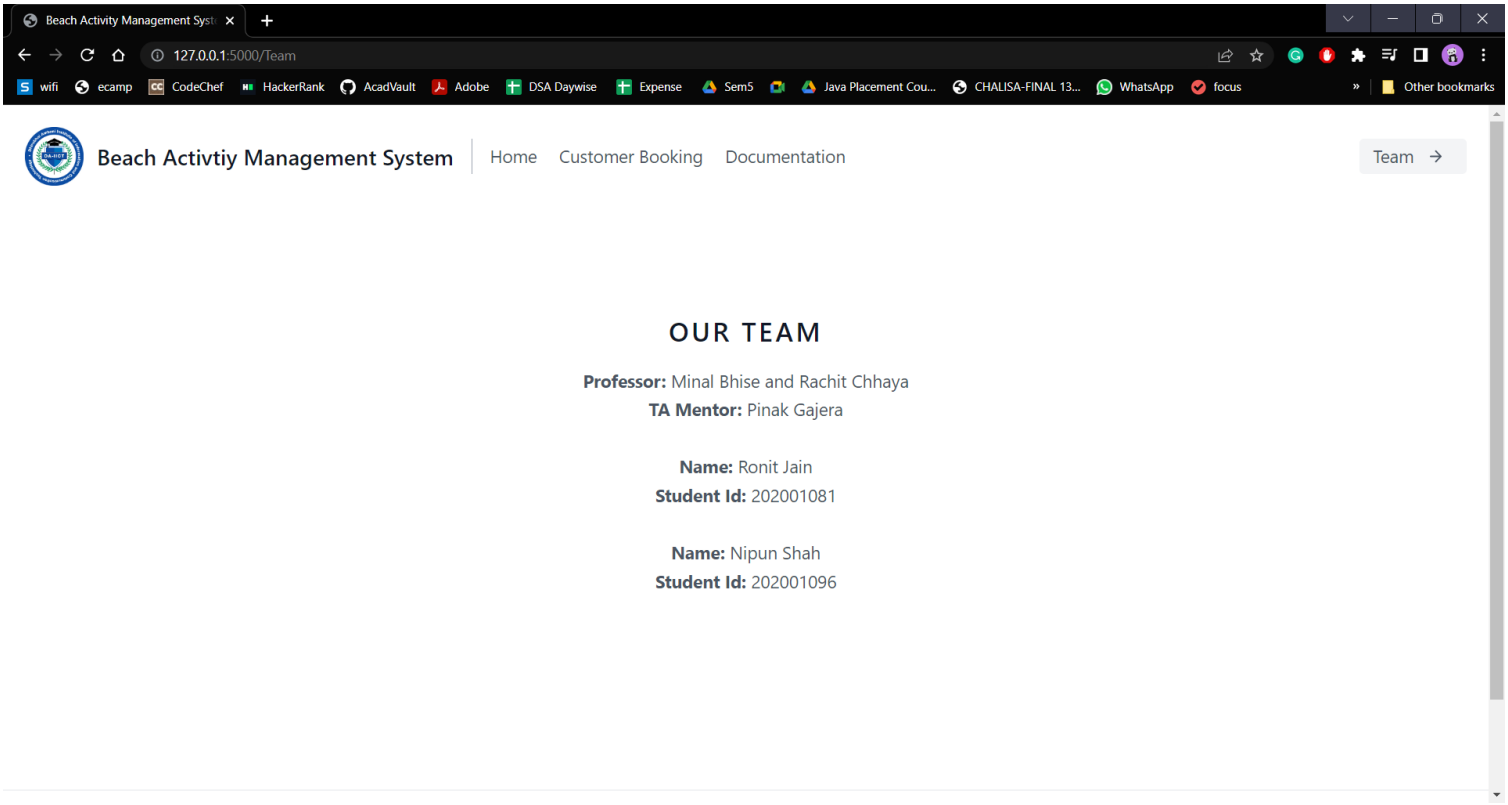
Documentation:

This page contains the pdf document of our project which shows all the details of the work done by us for creating database and beach management system.



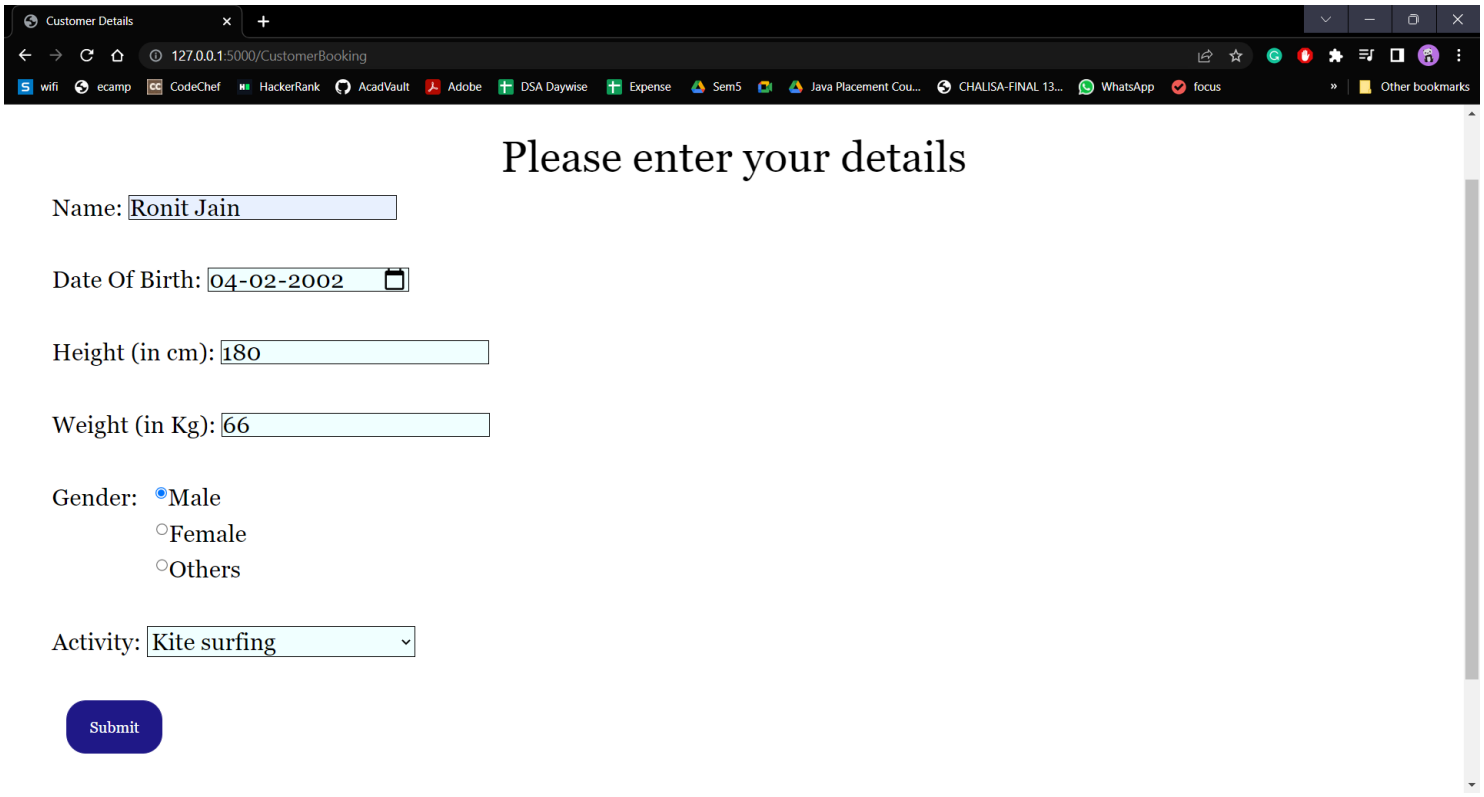
Team:

This page show details about our team, TA mentor and Professors.



Customer Booking:

This is our first page where customer enters his details about himself which when he submits are inserted into Customer table in our database and Customer_id is automatically generated.

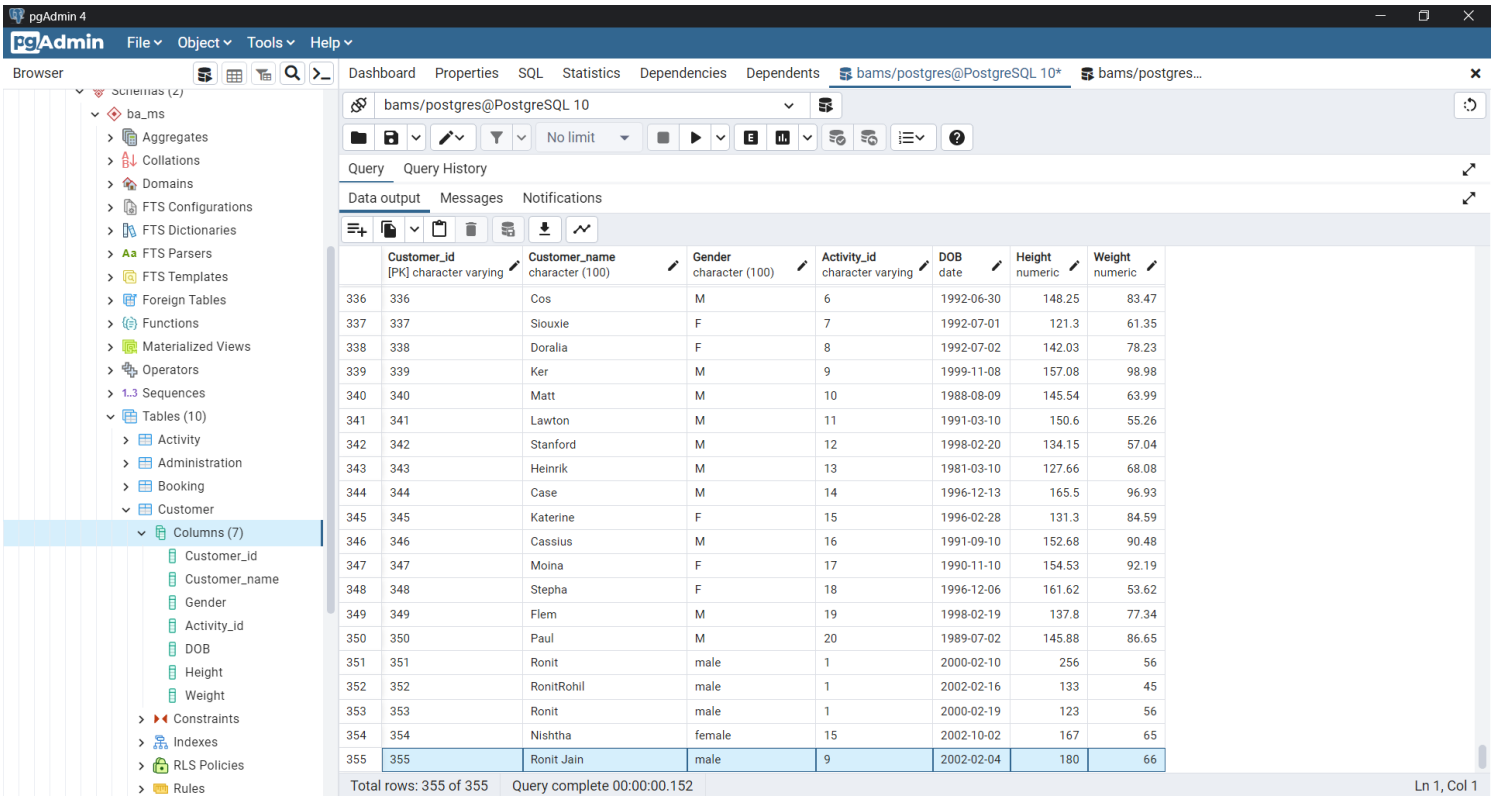


Here in this form we enter all the details and they are inserted in the customer table and Customer_id is automatically generated.

```
@app.route('/CustomerBooking', methods=['post','get'])
def CustomerBooking():
    print("Hello")
    if request.method == 'POST':
        Customer_name = request.form.get('name')
        DOB = request.form.get('dob')
        Height = request.form.get('height')
        Weight = request.form.get('weight')
        Gender = request.form.get('Gender')
        Activity_id = request.form.get('activity_id')
        cursor.execute('select count(*) from ba_ms.\"Customer\"')
        result = cursor.fetchone()
        count = result[0]+1
        print(result)
        insert_query = """ INSERT INTO ba_ms.\"Customer\" VALUES
(%s,%s,%s,%s,%s,%s,%s)"""
        try:
            record = (int(count), Customer_name, Gender, Activity_id, DOB,
int(Height), int(Weight))
```

```
            print(insert_query,record)
            cursor.execute(insert_query, record)
            conn.commit()
            count = cursor.rowcount
            print(count, "Record inserted successfully into Customer table")

        except:
            return "Invalid details"
    return render_template(Booking.html')
```

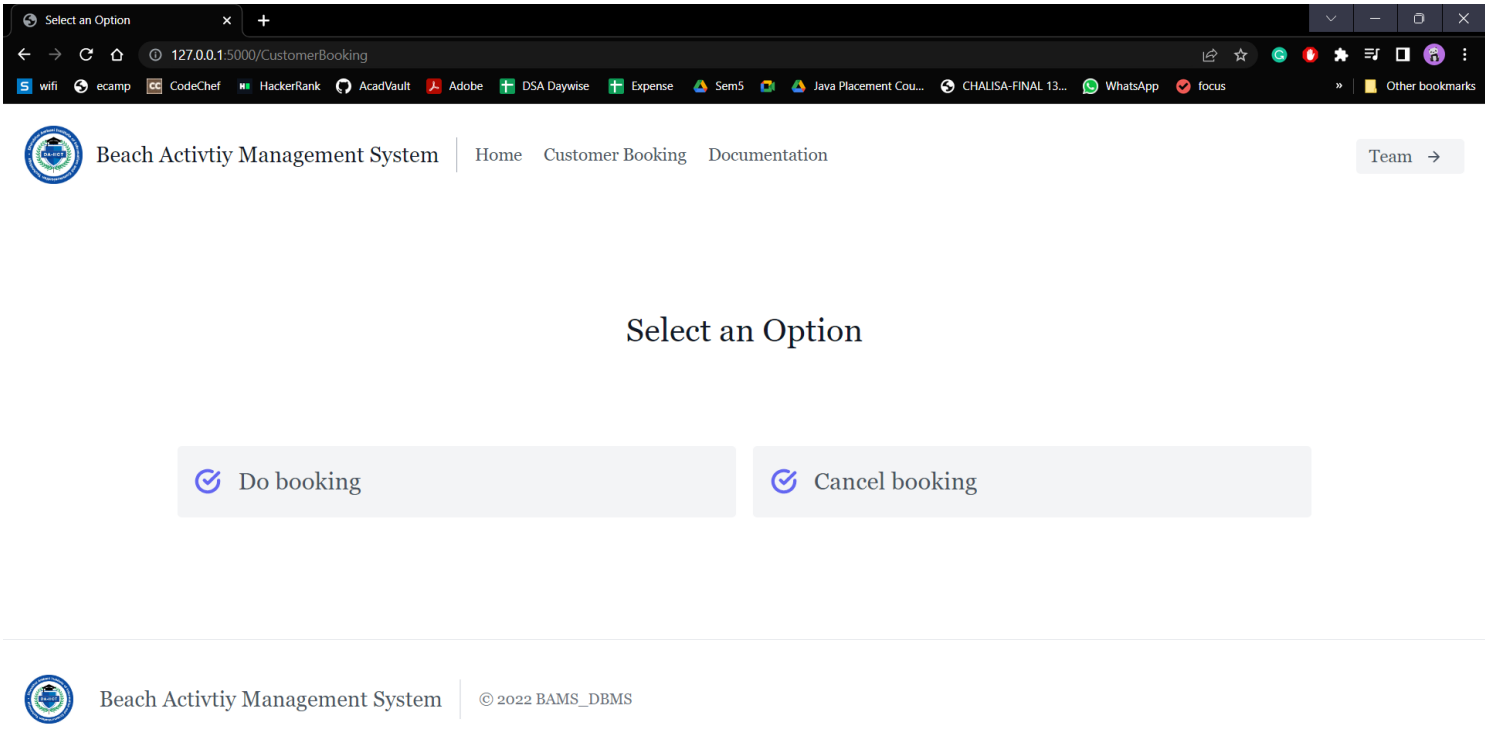
This code inserts data into Customer table entered on the website and generates automatically customer_id.



	Customer_id [PK] character varying	Customer_name character (100)	Gender character (100)	Activity_id character varying	DOB date	Height numeric	Weight numeric
336	336	Cos	M	6	1992-06-30	148.25	83.47
337	337	Siouxie	F	7	1992-07-01	121.3	61.35
338	338	Doralia	F	8	1992-07-02	142.03	78.23
339	339	Ker	M	9	1999-11-08	157.08	98.98
340	340	Matt	M	10	1988-08-09	145.54	63.99
341	341	Lawton	M	11	1991-03-10	150.6	55.26
342	342	Stanford	M	12	1998-02-20	134.15	57.04
343	343	Heinrik	M	13	1981-03-10	127.66	68.08
344	344	Case	M	14	1996-12-13	165.5	96.93
345	345	Katerine	F	15	1996-02-28	131.3	84.59
346	346	Cassius	M	16	1991-09-10	152.68	90.48
347	347	Moina	F	17	1990-11-10	154.53	92.19
348	348	Stepha	F	18	1996-12-06	161.62	53.62
349	349	Flem	M	19	1998-02-19	137.8	77.34
350	350	Paul	M	20	1989-07-02	145.88	86.65
351	351	Ronit	male	1	2000-02-10	256	56
352	352	RonitRohil	male	1	2002-02-16	133	45
353	353	Ronit	male	1	2000-02-19	123	56
354	354	Nishtha	female	15	2002-10-02	167	65
355	355	Ronit Jain	male	9	2002-02-04	180	66

Booking:

After submitting the details customer has two options from where he can either cancel the booking or make a new booking.



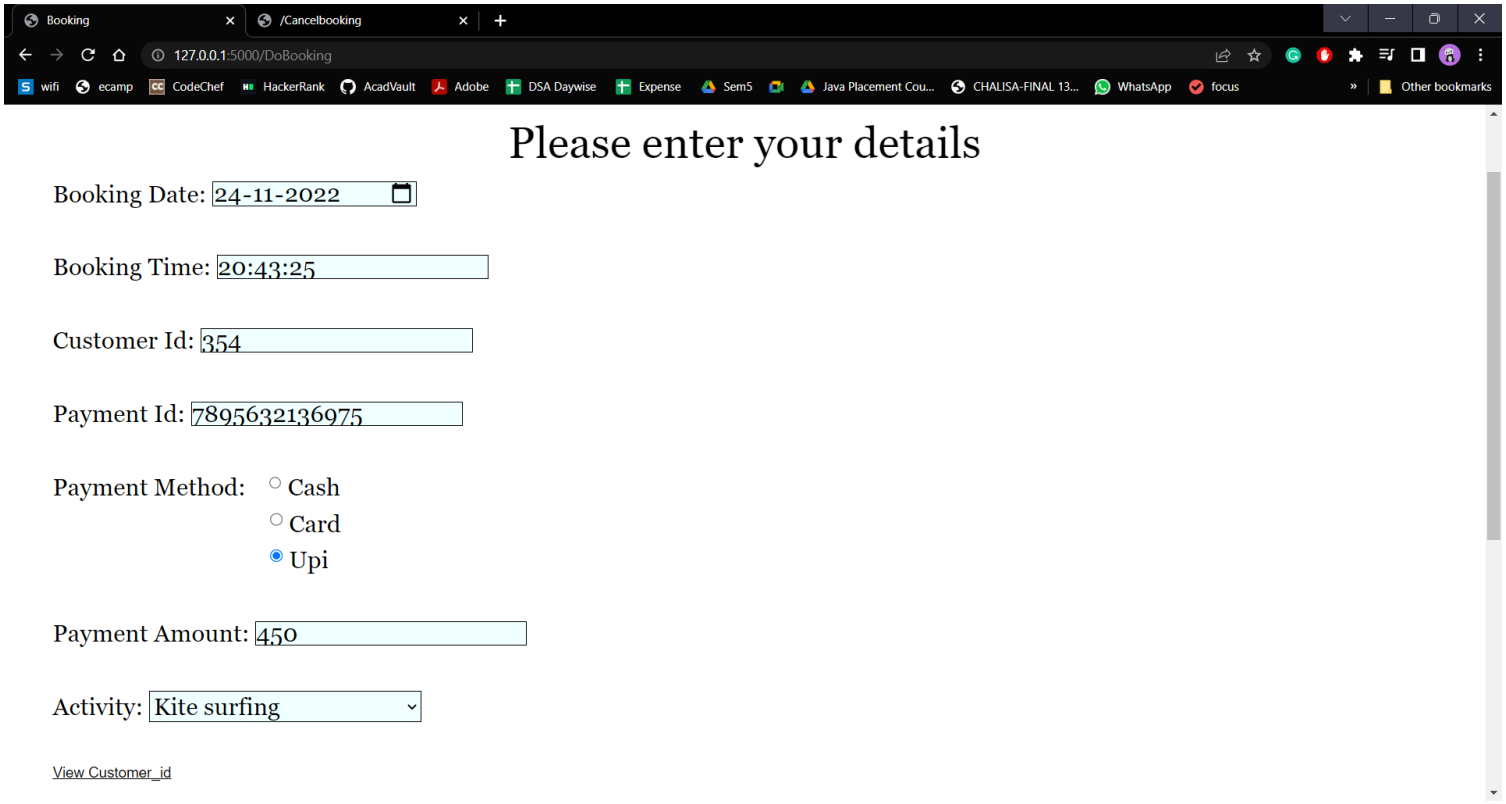
Do Booking:

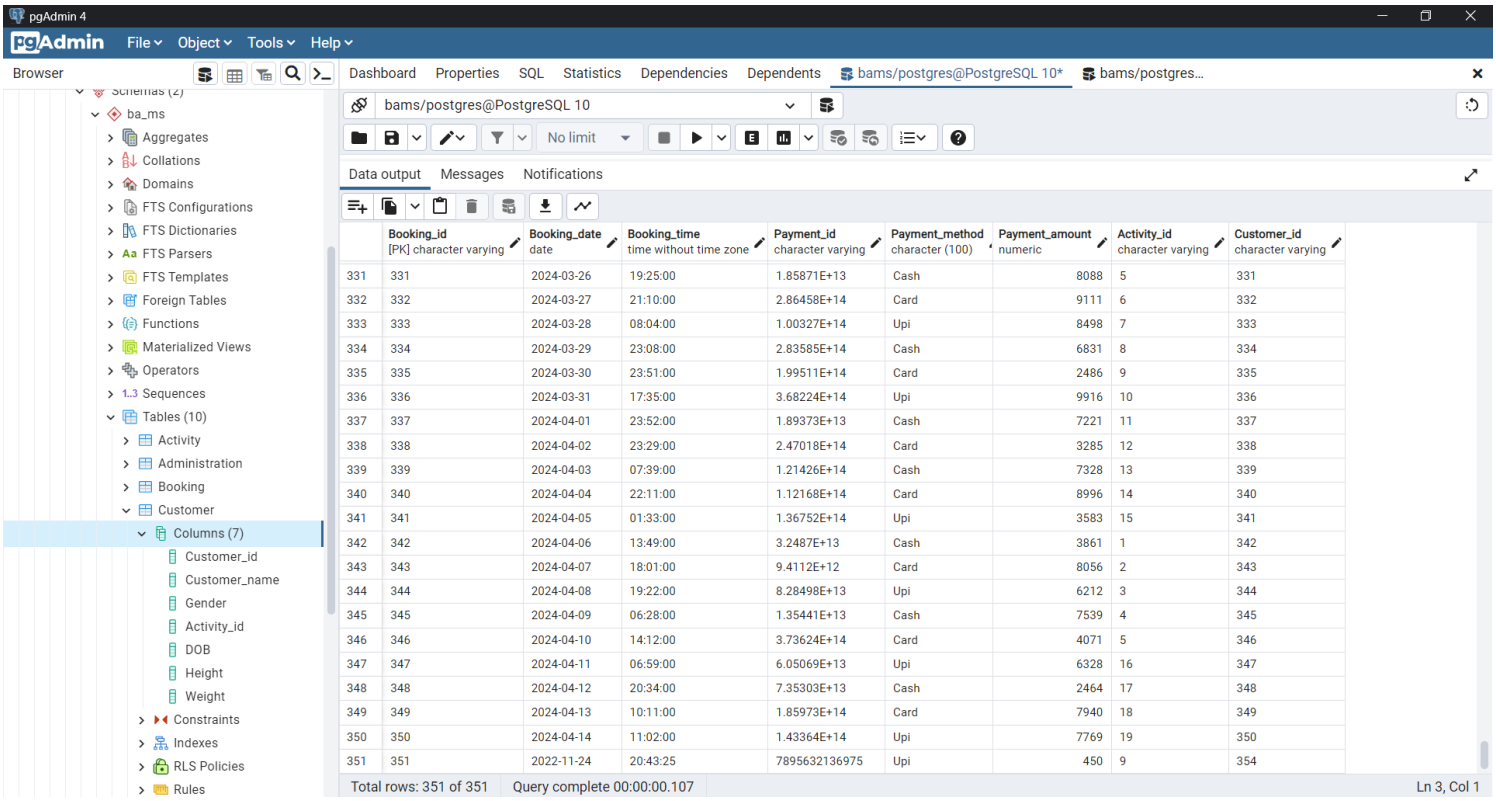
This page occurs when we select Do booking. This page helps customers to do booking. In this customer enter all the details required for booking. After submitting this data gets inserted into Booking table able booking_id is auto generated as in customer table.

```
@app.route('/DoBooking', methods=['post','get'])
def doBooking():
    print("Hello")
    if request.method == 'POST':
        Booking_date = request.form.get('Booking_date')
        Booking_time = request.form.get('Booking_time')
        Customer_id = request.form.get('Customer_id')
        Payment_id = request.form.get('Payment_id')
        Payment_method = request.form.get('Payment Method')
        Payment_amount = request.form.get('Payment_amount')
```

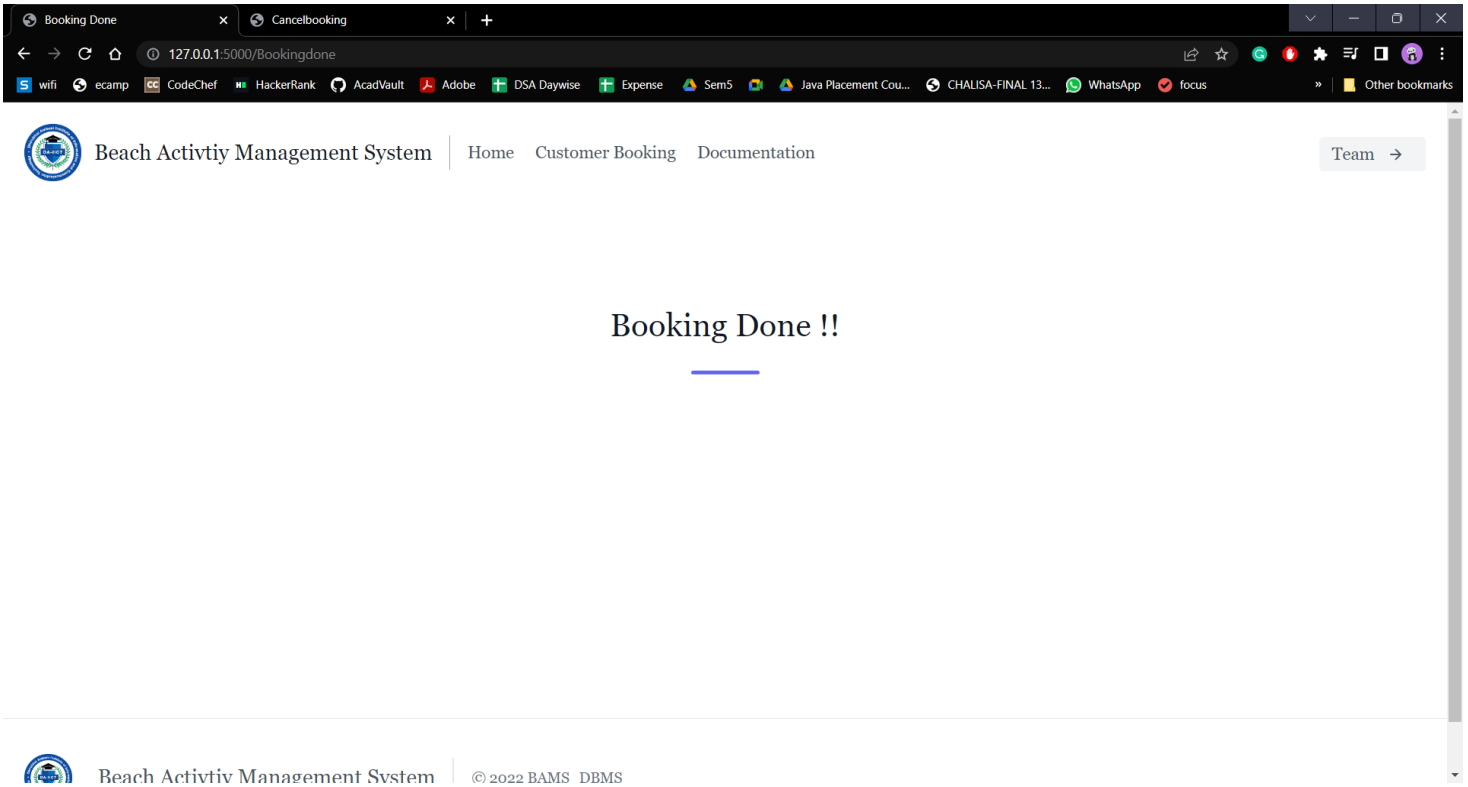
```
        Activity_id = request.form.get('activity_id')
        cursor.execute('select count(*) from ba_ms.\"Booking\"')
        result = cursor.fetchone()
        count = result[0]+1
        print(result)
        insert_query = """ INSERT INTO ba_ms.\"Booking\" VALUES
(%s,%s,%s,%s,%s,%s,%s,%s)"""
        try:
            record = (int(count), Booking_date, Booking_time, int(Payment_id),
Payment_method, int(Payment_amount), Activity_id, int(Customer_id))
            cursor.execute(insert_query, record)
            conn.commit()
            count = cursor.rowcount
            print(count, "Record inserted successfully into Booking table")
        except:
            return "Invalid details"
    return render_template('Booking_done.html')
```

This code inserts data into Booking table entered on the website and generates automatically Booking_id.



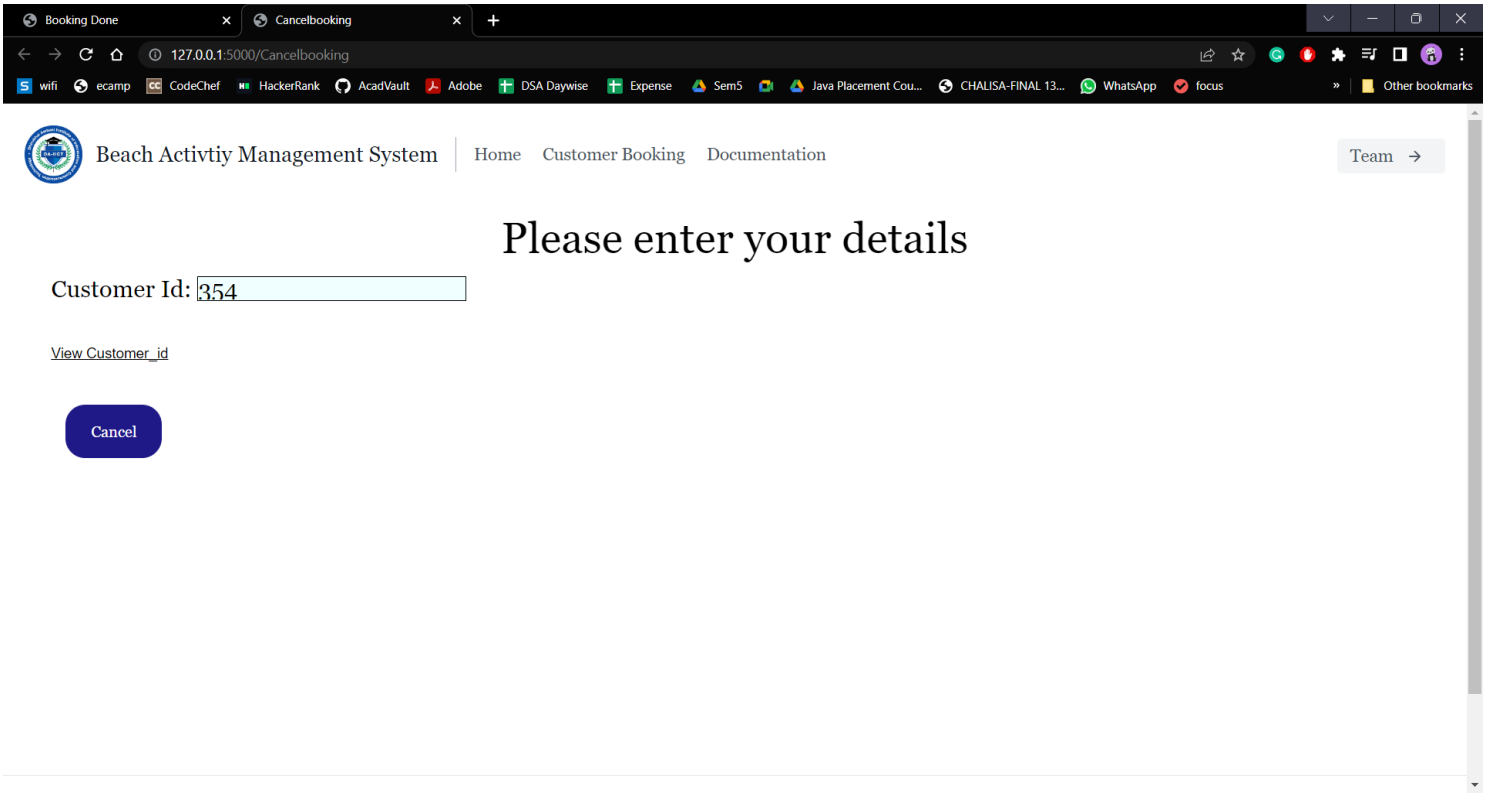


After submission this page comes which shows that booking is done.

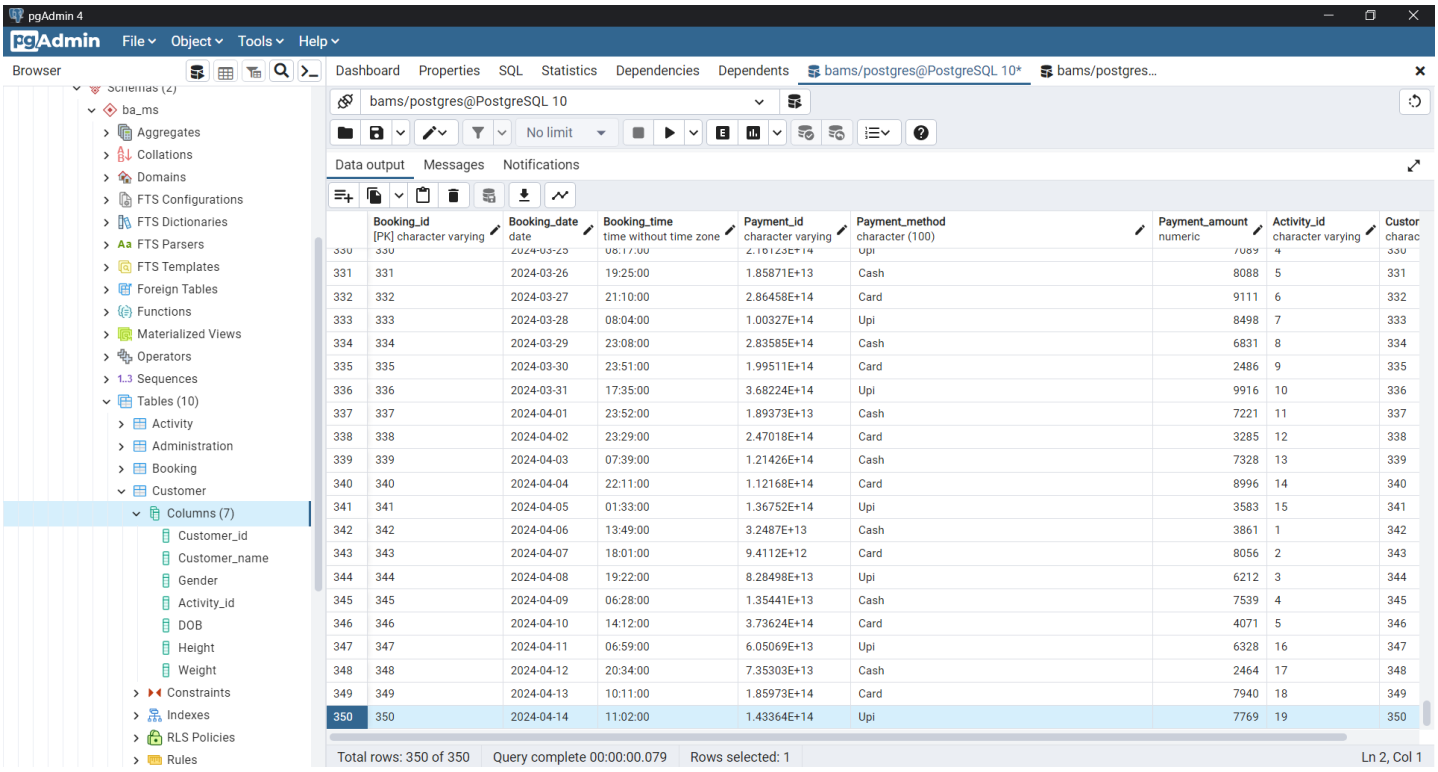


Cancel Booking:

This feature let us cancel our booking by entering customer_id in the form.




```python
@app.route('/Cancel', methods=['post','get'])
def CancelBooking():
    print("Hello")
    if request.method == 'POST':
        Customer_id = request.form.get('Customer_id')
        delete_query = """ DELETE FROM ba_ms.\"Booking\" where ba_ms.\"Booking\".
        "Customer_id\" = %s"""
        try:
            record = (Customer_id)
            cursor.execute(delete_query, record)
            conn.commit()
            count = cursor.rowcount
            print(count, "Record deleted successfully from Booking table")
        except:
            return "Invalid details"
    return redirect('/Canceldone')
```



This page shows that our booking is cancelled and deleted from the database and Booking Table.