Database Management System

Final Project Report

Section 2 - Group 10

The Team
Aditya Shah - 201801008
Rishi Shivani - 201801073
Pavan Kotak - 201801120
Bhrugu Dave - 201801401

Table of Contents

Revision History

Name	Date	Reason for change	Version
Car rental service	**26/09/2020**		**1.0**
Car rental service	**30/09/2020**	**Finalizing the document after relevant steps.**	**2.0**
Car rental service	**03/10/2020**	**Revision of the document and updating the background reading**	**3.0**
Car rental service	**28/11/2020**	**Updating the document with current schemas**	**4.0**

Product Description

1. Purpose

The purpose of this product is to provide a solution for the problems faced by local car rental service providers like better connectivity among the multiple outlets, proper management of staff and cars within the outlet, availability of instantaneous records for the same, and an overall secure and adequate fund management system.

2. Intended Audience and Reading Suggestions

Stakeholders like the owner of the car rental service should go through the sections of product descriptions, different privileges are available for different users of this database, the fact-finding section, and the business constraint of this database. While staff members, who will be the administrators of this database, should go through the sections following the user classes and characteristics section. The tester of this database should go through the whole document starting from the fact-finding section. The developers should look in details mentioned in sections after the Fact Finding Chart which will give insights about the design and implementation of the required database.

3. Product Scope

This product mainly provides convenience to different car rental service providers who still maintain their records on paper or using excel sheets. This software offers convenient, secure, and reliable services for maintaining, updating, and deleting records from the database. Also, this product provides a relatively better interface for handling the records for the managerial staff. Thus, the scope of this database extends to services looking for a better database management system.

4. Description

In this modern world, everyone needs a form of transportation that is fast, convenient, and easy to access. This form of transportation isn't static for everyone, it will change person to person and even day to day. But for a while, now a good way to combine multiple forms of transportations i.e. having a personal car, hiring a taxi, etc has been to rent a car. Now, as more people are inclined towards this sort of ideology and since it gets a boost from start-ups advertising about the same thing, we end up in a place where the local rental

industry gets a boost. But there is a hindrance in the form of the convenience factor as most of the local rental places haven't yet migrated to an online platform so it's hard for the management to keep track of their cars and for the customers to browse their selection of cars before visiting the place. So, for digitizing the simple implementation of their workflow into a product that can be accessed by various types of users without a restriction of geography is required.

Now, since we are designing this product for the current client which is Ghost Car Rental Services, we must first properly understand the problems that they are facing and what their current workflow is so that we can provide a solution for them that is the most optimal. Therefore, firstly we need a detailed description of the current workflow and then speculate what solutions for the interface are possible.

The primary thing that any car rental place would need is cars, as without it there the place isn't really a car rental place. So, currently, GCRS maintains the records of cars in an excel spreadsheet, which is updatable by only the main branch as it is stored locally there. Now, this spreadsheet contains information along the lines of the license plate and chassis number of the cars. And any new cars that are purchased can only be managed by the main branch. From this there is a clear problem of the need to update the records at each and every branch as these updates, in some scenarios, can overwhelm the people doing these tasks at the main branch as the company expects the all the branches to have an updated record so that the customer has the most recent list of cars. This feature of concurrency is an important aspect, in the company's eye, is an important aspect of fast and efficient customer service.

Speaking of customer service, let's discuss the people who are the most important, nay the only ones who provide it, the clerical staff and the drivers. Currently, the management of drivers at GCRS can most aptly be termed as haphazard. The records of employed drivers and staff are maintained on books managed by the main branch using a unique employee ID. So, even if a branch wanted to hire a temp worker it has to coordinate with the main branch so that the main branch can have updated information on the workers, even updating personal information requires one to work with the main branch and of course, maintaining salary records is a nightmare. All the information on how much to pay each member of the staff is only available to view at the end of the month and all of this is sent by the main branch. The most obvious solution to this problem is to provide an easy to grasp and user-friendly UI, database synchronization between different franchises, and the main branch. This will lay

off some manual work done by the clerks of different franchises and also the main branch.

As it is obvious the central theme of all the discussions until this point relates to the staff of each and every branch. The records of staff are to be maintained branch-wise at each and every individual branch and as well as should be made available to the main branch. Thus for this purpose, branch managers are held responsible. It’s the job of the branch manager to maintain records of the fleet of cars under that branch, staff employed and answering any main issues caused at the branch. Also, they need to maintain concurrency with the main branch.

It is quite obvious that all the individual branches depend too much on the main branch and the owners also have to depend on the main branch staff for any sort of data. Thus, in the case of any sort of malfunction at the main branch, the company is very much vulnerable to important data loss. Also, let's not forget the point that all these records are maintained on either books or locally stored spreadsheets. Thus, the theft of data is also a very serious possibility. Thus, an integrated database would be the aptest solution. It would provide a concurrent database that could have limited access as well as it would give full access to the owners to the statistical data at any given time.

Up till now, we have just discussed the staff and the fleet of cars at a branch. Let us also see how the data of customers is being managed. At the moment, any customer can acquire the service of GCRS by either calling the branch or on spot booking. Before the car is handed over, the branch staff, more precisely the clerk, takes all the necessary information about the customer and proof ID before handing over the car. These records are maintained by a unique id which is given to the customer also, so that the next time the customer comes, he need not fill up the details form again. This system eliminates the redundancy at individual branch levels but causes problems across different branches. Suppose a customer registered at a particular branch and obtained a unique id for identification. But this id won’t work if he chooses to obtain services from a different branch. As such this data is branch limited, also, synchronization among the branches is not an option under the current system implemented for handling data. Thus, databases would maintain and provide synchronized data across branches about customers and also maintain uniqueness using customer ID given to each customer at time of registration.

So far we have discussed how different important data about individuals is being managed. But, we haven't discussed how the trip records as well as maintenance records are being handled. Suppose, a customer decides to acquire services of GCRS via a phone call. First of all, he is asked whether he has a unique customer id or not. If not he is asked to give all the required information over the phone call or is asked to come over to the branch for the same. Then he is asked, whether he requires a driver or not. If yes, then the call attendee checks whether a driver is available or not and assigns one for the same if available. Also, the customer is asked about what sort of car he wants, and it's the job of the staff to determine whether a car fulfilling the requirements is available or not. Thus, the staff has to do tedious work of going through for each and every booking request. Once this is fulfilled, the clerk has to jot down each and every trip in a record book describing which car is rented, which driver is allotted, and all sorts of required details. This record is maintained in books, thus, for calculating profit at any given time one has to calculate a significant amount of data. Once all these details are filled, the car is handed over and the pickup destination and time period with drop off destination is noted down after the trip is over. If the rented car is dropped on a different outlet, then it also causes some serious trouble for the clerks to synchronize the data for each trip and how to distribute the money between the outlets. Thus, intra and inter-city trip data updation is also a major issue.

One of the problems is that when a customer wants to rent a car, a clerk needs to show the available options, if a customer is booking via phone then the clerk describes the available cars according to the requirement of the customer, if a customer rents via on spot booking, then the clerk shows him the best option available. A customer would like to browse all the available options by himself and choose the best for him rather than the clerk showing him the car or describing the vehicle over the phone. So, we intend to develop a user-friendly interface for customers such that a customer can browse over the available options and filter the options according to his choice and rates of individual cars are also displayed which is maintained using a rate table.

We can maintain the records for all the franchises including the main branch by implementing an efficient database management system that will contain the staff data, customer data, fleet management, and records of the trips. This will provide an efficient solution to the concurrency of the data and all the databases can be synchronized to a cloud among the various outlets. This would result in a better and efficient way of inserting, updating, and also deleting the data into the

database. This will also help to ease the maintenance of the database among the various outlets.

The integration of the database will provide users of different classes with different privileges. This means that owners will have options for providing limited access to the database. Customers will have access to only the database of available cars on a particular franchise. Staff will also have limited roles to the database according to their job position such as clerks, branch managers, accountants, and drivers.

One feature that the company so far has failed to incorporate is the rating system for the vehicles and drivers given by the customers. We intend to implement such a system so that a customer can give a rating in the stars and if a customer wishes can also leave written comments along with the rating. This will help other customers to choose the vehicle and a driver considering the previous ratings of the respective car and driver. Implementing this system will also help the customer filter the available results according to the previous ratings and reviews. Also, the company can improve its facilities and services according to the reviews and ratings are given by the customers after the completion of the trip. This feature shall be integrated with the UI and displayed when asked.

Fact-Finding Phase

A. Background reading

To get a proper understanding of what customers of GCRS need and what the company owners require in order to run their business effectively we referred to different books and blogs.

Firstly, to get the best understanding of the company's most important aspect requirement we went through a blog that stated what a customer expects from a car rental service provider. For further insight, we circulated forms among different classes of customers the company had as well as on the internet.

Secondly, to gain insight into what exactly the company exactly needs for better management and effective handling of data, a couple of blogs and books were referred to list out, what various options the company has going forward and which options can the company afford.

To build a proper design to meet company’s requirements, we studied in depth about existing software namely Rent-a-car Manager, Rent Centric and Rent Syst available for managing car rental services. The first two software provides a solution which implements a centralized server maintaining database. While the third one implements cloud based database managing software.

The first Rent-a-car Manager provides several facilities such as fleet management, activity planners, cost management, client management, statistical reports, and web integration. We aim to implement fleet management,cost management, client management and web integration for easy to use and customer satisfaction. We aim to make the User Interface adaptive to different environments with its responsive design.[1]

The second Rent Centric provides a better fleet management system which extends to the vehicle tracking, maintenance and history record collecting,

[1] http://rentacar-manager.com/en/

financial management and fleet reporting. We intend to add GPS ID to each car and a customer as well as the staff can track the vehicle. Fleet management feature provides a complete overview of the state of the fleet - current location, physical condition, engagement, costs and income of each vehicle individually. This is overall a better software for fleet management compared to Rent-a-car Manager software.[2]

The last one Rent Syst is a software to automate business processes with full control of car rental and convenient order processing. Suitable for both large fleets and owners of one or more cars. With its help, order management is greatly simplified and also increases the revenue and profitability of each car. Our software allows you to take a different look at managing the auto business. With its help you can track individual cars and keep track of the available cars and help the customers keep track of their past rentals as well for a better experience.[3]

Sources:

1. https://fleetroot.com/blog/car-rental-software-the-best-guide-2020/
2. Dean Leffingwell and Don Widring, “Managing Software Requirements: A Use Case
3. https://www.rentcentric.com/products/car-rental-software
4. https://rentsyst.com/
5. http://rentacar-manager.com/en/
6. https://www.researchgate.net/publication/325253983_Online_Car_Rental_System_using_Web-Based_and_SMS_Technology

[2] https://rentsyst.com/

[3] https://www.rentcentric.com/products/car-rental-software

B. Interviews

Mainly three interviews were held for developing a proper design of system management, one with the owners, one with the branch manager, and one with the employees of various branches. The interview plan and summary of all interviews are attached here.

Griefingdor: (Role Play) Interview Plan For CEO

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Mr.Reeves (Role Play) **Designation**: CEO

Contact Details: gcrs_owner@gmail.com
Organization Details: Ghost car rental service
Interviewer:

1) Rishi Shiwani **Designation**: Business Development Executive
2) Bhrugu Dave **Designation**: Developer

Date: 29/09/2020 **Time**: 16:30
Duration: 1 hour **Place**: WebEx Meeting

Purpose of Interview:
Preliminary meeting to identify problems, bottlenecks, and requirements related to the current system of operation at Ghost car rental service.

Agenda:
Gaining ideas about the budget of the company.
Other business and implementation constraints.
Working of different outlets and data synchronization.
Features expected in the database.

Documents to be brought to the interview:
Rough plan of the UI and database
Any documents relating to its usage

Griefingdor: (Role Play) Interview Summary For CEO

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Mr.Reeves (Role Play) **Designation**: CEO

Contact Details: gcrs_owner@gmail.com
Organization Details: Ghost car rental service
Interviewer:

1) Rishi Shiwani **Designation**: Business Development Executive
2) Bhrugu Dave **Designation**: Developer

Date: 29/09/2020 **Time**: 16:30
Duration: 1 hour **Place**: WebEx Meeting

Results of Interview:

- The company is willing to allot a fair budget for effective implementation.
- Data synchronization among different outlets is absolutely necessary.
- There are no technical or business constraints expected to occur.
- Simple UI must be implemented for customers.
- Staff also shouldn't find it difficult to use.
- Different privileges allocations power should be provided.
- Data security is a major concern.

Griefingdor: (Role Play) Interview Plan For Branch Managers

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Hemant Chanda(Role Play) **Designation**: Branch Manager
2) Manish Kumar **Designation**: Branch Manager

Organization Details: Ghost car rental service
Interviewer:

1) Pavan Kotak **Designation**: Business Development Executive
2) Aditya Shah **Designation**: Developer

Date: 29/09/2020 **Time**: 18:30
Duration: 2 hours **Place**: WebEx Meeting

Purpose of Interview:
To get insight into the inner working of an individual branch and how they collaborate with other branches for data synchronization.

Agenda:
Different positions offered in every branch.
Different user classes, expected in the database.
Data synchronization.
Staff's technical skills.

Documents to be brought to the interview:
Rough plan of database and different functionalities of the database.
Any documents relating to its usage.

Griefingdor: (Role Play) Interview Summary For Branch Managers

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Hemant Chanda(Role Play) **Designation**: Branch Manager
2) Manish Kumar **Designation**: Branch Manager

Organization Details: Ghost car rental service
Interviewer:

1) Pavan Kotak **Designation**: Business Development Executive
2) Aditya Shah **Designation**: Developer

Date: 29/09/2020 **Time**: 18:30
Duration: 2 hours **Place**: WebEx Meeting

Results of Interview:

- At the moment, all reservations are made through calls and on-spot bookings.
- In every branch, 3 types of position have access to the records namely, clerk, accountant, and manager, and the rest of the staff is for technical and physical support
- The clerk is responsible for making proper entries into the record and the accountant handles all the billings and expenditure of the branch.
- The manager overlooks all the records and billings and in case of any data entry mistake, the manager is responsible for data updation.
- Most of the staff, who will access the database have a fair bit of software skills and are willing to adapt to the new norm.
- Thus, the staff should have relatively easy access to the database.

Griefingdor: (Role Play) Interview Plan For Accountants

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Nidhi Kesaria(Role Play) **Designation**: Accountant
2) Priyanka Sameja **Designation**: Accountant

Organization Details: Ghost car rental service
Interviewer:

1) Pavan Kotak **Designation**: Business Development Executive
2) Bhrugu Dave **Designation**: Developer

Date: 23/09/2020 **Time**: 18:30
Duration: 1.5 hours **Place**: WebEx Meeting

Purpose of Interview:
To get insight into the accounting aspects of the company and to know the requirements from the accounts side.

Agenda:
The different functions that the accountants use and what they require to perform their job.
The technical expertise of the accounting department.
The frequency of accessing the data.

Documents to be brought to the interview:
Rough plan of database and different functionalities of the database.
Any documents relating to its usage.

Griefingdor: (Role Play) Interview Summary For Accountants

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Nidhi Kesaria(Role Play) **Designation**: Accountant
2) Priyanka Sameja **Designation**: Accountant

Organization Details: Ghost car rental service
Interviewer:

1) Pavan Kotak **Designation**: Business Development Executive
2) Bhrugu Dave **Designation**: Developer

Date: 23/09/2020 **Time**: 18:30
Duration: 1.5 hours **Place**: WebEx Meeting
Results of Interview:

- At the moment all the relevant data is sent to them at the end of the month.
- The staff has a basic idea of using this kind of software.
- Accountants are asked for monthly reports and occasionally asked financial reports.
- Accountants must have access to the salaries of all staff members of their own branch.
- If possible, instead of writing functions, they would like a UI for the same.

Griefingdor: (Role Play) Interview Plan For Clerks

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Prashant Verma(Role Play) **Designation**: Clerk
2) Harsha Goyal **Designation**: Clerk
3) Sanjay Dutta **Designation**: Clerk

Organization Details: Ghost car rental service
Interviewer:

1) Pavan Kotak **Designation**: Business Development Executive
2) Bhrugu Dave **Designation**: Developer

Date: 23/09/2020 **Time**: 18:30
Duration: 2 hours **Place**: WebEx Meeting

Purpose of Interview:
To get insights into how are the records handled by the clerks and what problems are faced in synchronization across branches.

Agenda:
Different functionality expect in the new system
Software skills and technical abilities of the staff
Training sessions
Loopholes of the current system and expected solutions.

Documents to be brought to the interview:
Rough plan of database and different functionalities of the database.
Any documents relating to its usage.

Griefingdor: (Role Play) Interview Summary For Clerks

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Interviewee:

1) Prashant Verma(Role Play) **Designation**: Clerk
2) Harsha Goyal **Designation**: Clerk
3) Sanjay Dutta **Designation**: Clerk

Organization Details: Ghost car rental service
Interviewer:

1) Pavan Kotak **Designation**: Business Development Executive
2) Bhrugu Dave **Designation**: Developer

Date: 23/09/2020 **Time**: 18:30
Duration: 2 hours **Place**: WebEx Meeting
Results of Interview:

- The different functions that clerks need to make entries into the database.
- They also need access to the list of cars and drivers available at any particular moment.
- Aside, from this, the current system works mostly on excel sheets and sometimes becomes inefficient for entering records.
- Thus, a compact data entry system user interface is expected.
- The staff has sufficient knowledge about the software used for implementation. Thus, no extra training will be required.
- The most known problem was entering data at the same moment in the same branch resulted in data inconsistency.
- Thus, proper handling of race conditions is required.

These are the plans and summaries of each and every interview conducted. The main concerns of the company before implementing a new system is the data synchronization and better user interface at each and every level of usage of the database.

C. Questionnaires

Griefingdor – Car Rental Survey for Ghost Car Rental Services
Please circle your answers to the following questions:

1. How often do you rent a car to meet your needs?
Never / once a year / two or three times a year / every month / no regular pattern

2. Factors Car Rentals Customers use to select a Car rental agency?
Price / Reach /Availability of Cars/ Quality of Services / Fast Check out of the car

3. What is the most important factor when you book your own personal car hire?
Price/ Number of seats/ Discounts/ Driver/ Comfort

4. How often do you need to use your car for work during the day?
No regular pattern/Every week/ 2-3 times a month/ once a month

5. Will you like a User Interface which would help you to book efficiently?
Yes / No

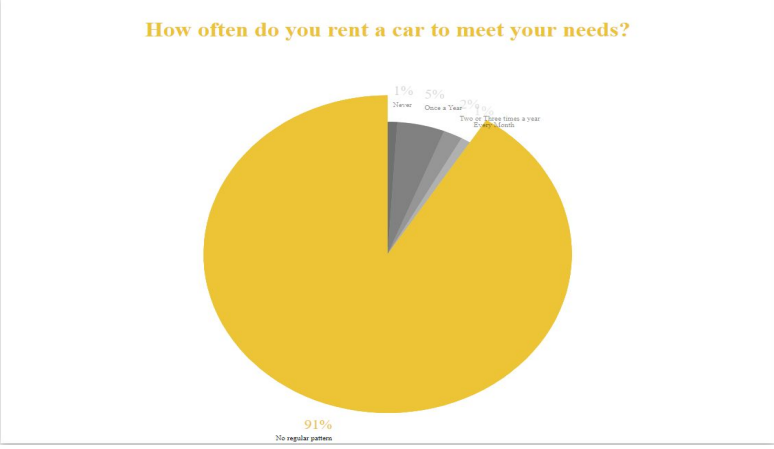
6. If you face problems, would you like to report it?
Yes / No

7. If Yes, How would you like to report it?
Message/ Call/ Website

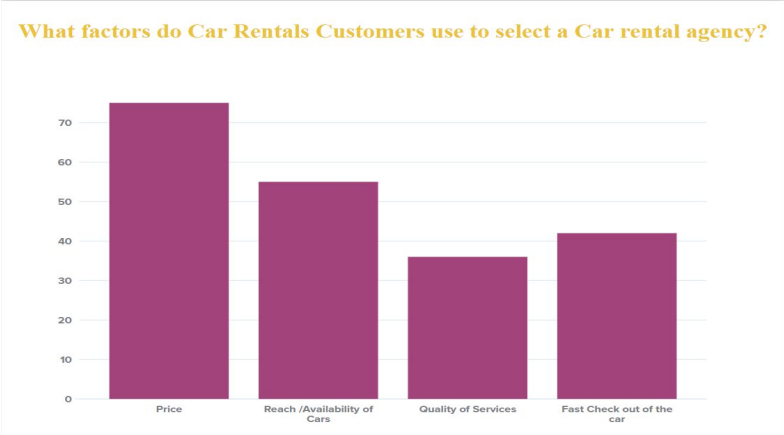
Your Name
Thank you for completing this questionnaire

D. Summary of Responses

1. How often do you rent a car to meet your needs?
   R: 91 percent of the audience responded with no regular pattern since most of them feel that there can be a need for a rented car at any time. Customers want to book a car any time they want so they want a 24/7 availability option for their immediate and urgent needs.

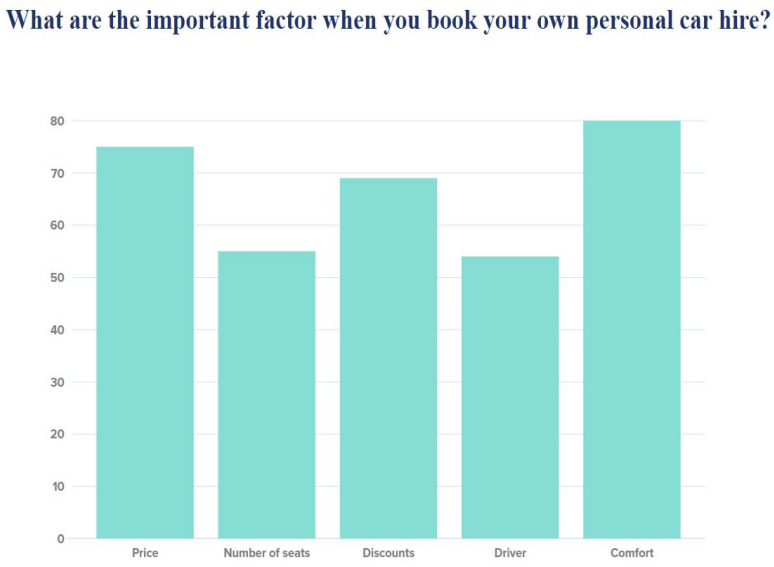


2. What factors do Car Rentals Customers use to select a Car rental agency?
   R: 75 percent of the respondents opted for the price since they wanted a fair price for their rented car and opted for discounted cars while 60 percent of respondents wanted fast check out of the car to minimize their time usage in the whole process. 55 percent of respondents preferred the availability of cars as a major factor for customer satisfaction.

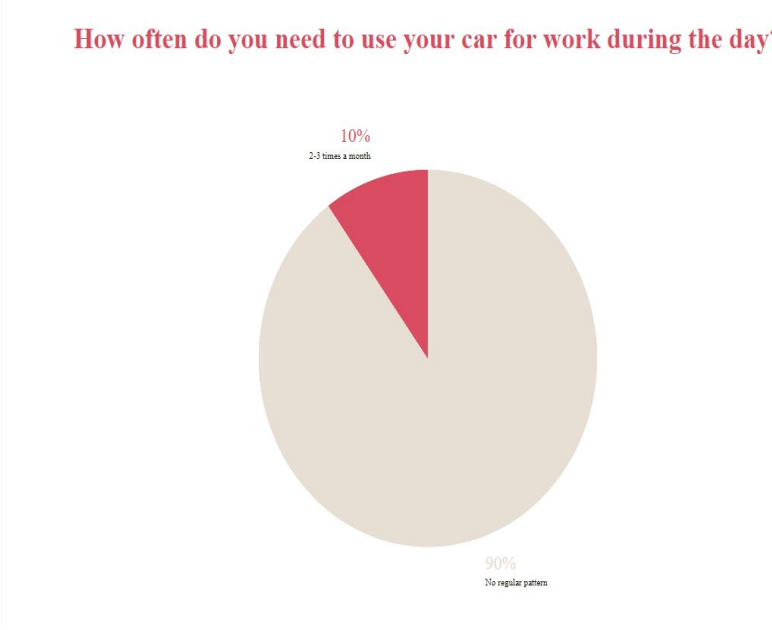


3. What is the most important factor when you book your own personal car hire?

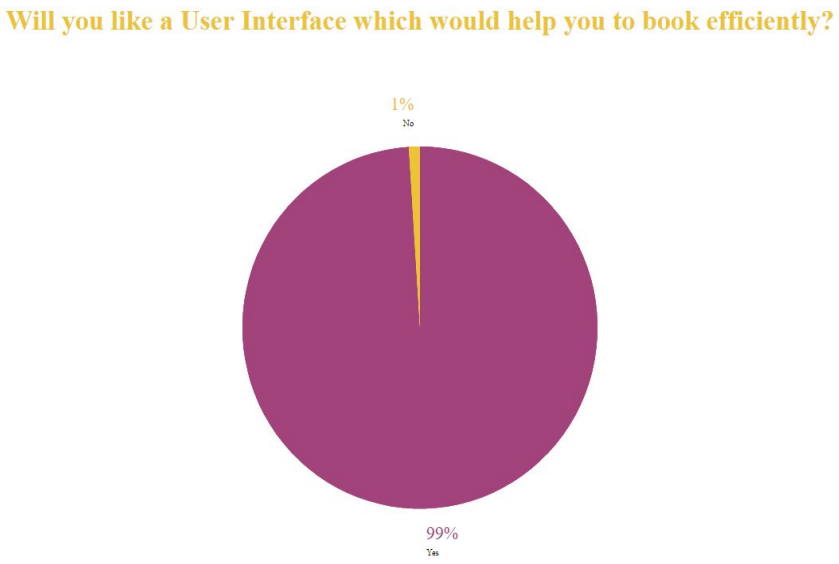
R: 80 percent of the respondents preferred comfort as they feel a car is worth renting when it is comfortable. 75 percent of respondents feel the price is the other dominating factor in this business.



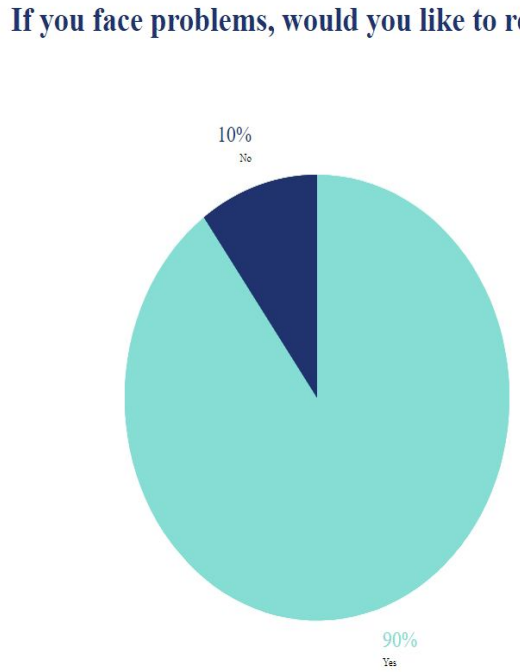
4. How often do you need to use your car for work during the day?
   R: 90 percent of the respondents feel there is no regular pattern in how often they use their car for work. 10 percent use it 2-3 times a month for work.



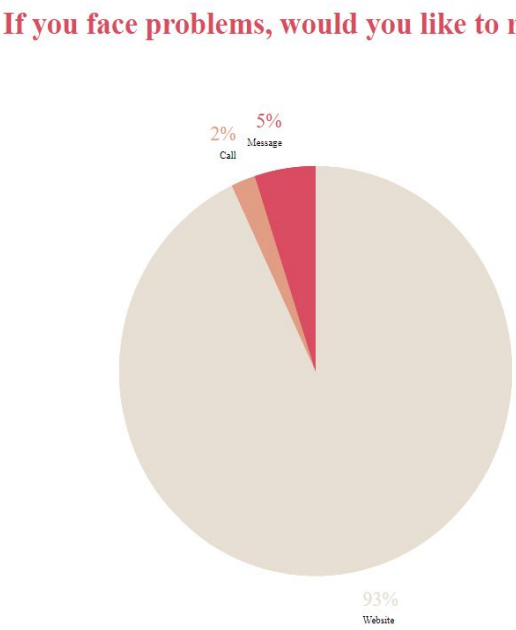
5. Will you like a User Interface which would help you to book efficiently?
   R: 99 percent of respondents opted for yes as its time-efficient to book from a user interface.



6. If you face problems, would you like to report it?
   R: 90 percent of respondents wanted to report the problems faced. They wanted to report the issues and wanted them to be resolved.



7. If Yes, How would you like to report it?
   R: 93 percent of the respondents wanted to report the problem on a website as it is the most convenient way.



Thus, we can clearly see that the customers have a clear opinion about having a compact UI for acquiring services is the utmost requirement. A platform for providing feedback should also exist.

Griefingdor: Observations

System: Ghost car rental service
Project Reference: SF/SJ/2020/9
Observers:

1) Pavan Kotak — **Designation**: Business Development Executive

Date: 23/09/2020 — **Time**: 18:30
Duration: 45 mins — **Place**: WebEx Meeting

Observations:

1. Customers prefer price over other facilities as they feel the price is more important for customer satisfaction. Price is the most dominating factor in the car rental business.
2. The second most important factor is the comfort as Customers crave for a comfortable rental car. Comfort is the biggest demand after a nominal price.
3. Customers seek the most comfortable vehicle according to their budget.
4. Customers prefer vehicles that have been maintained and cleaned regularly.
5. For an efficient booking service, customers prefer to have a user interface to book a car of their choice.
6. Customers like to go over several choices before making a decision.
7. Customers are eager to provide feedback on their rented vehicles/driver so it's a good marketing strategy to satisfy their demands and ask for their feedback to improve our services.
8. Lot of problems are caused and require much man work for synchronizing data across various outlets.

Thus, we observed that data integration is absolutely essential because current software is unable to provide an efficient solution.

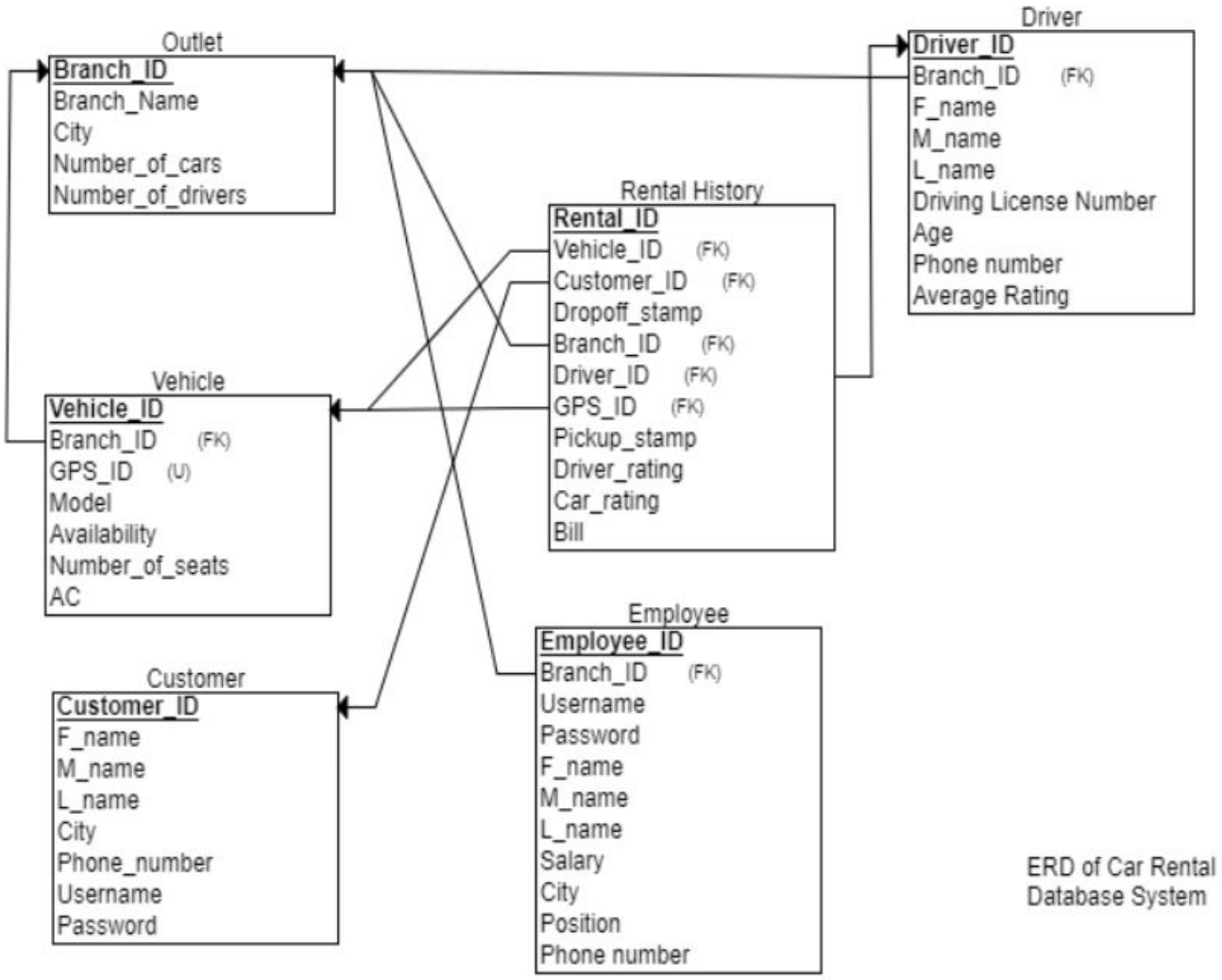
Fact-Finding Chart

Objective	Technique	Subject	Time Commitment
To get the background on the company	Background Reading	Company reports, Blogs, Wikipedia	1 day
To establish the company objective	Interview	CEO	1 hour
To get an insight into each branch and management	Interview	2 Branch Manager	2 hour 1 hour each
To find out the core business operations and what records and resources are kept	Interview	3 Clerk	2 hours
To find out the current accounting system works and what to establish the accounting for the new model	Interview	2 Accountant	1:30 hours 45 mins each
To follow up development of business understanding and features	Observations	-----	45 mins

Expected Deliverables

- The most important deliverable expected from the new system is data synchronization among all the branches of the company.
- The second important thing is limited access to the database. The owner can give access to the manager as required.
- Handling of data entries made at the same time i.e. race condition should be taken care of.
- To provide a better feedback system as well as display the same feedbacks efficiently on request.
- The owner although gave no constraints on the budget but expects a relatively affordable new management system for their business.
- Also, abstraction of implementation is also absolutely necessary.

Temporary Database Design



ERD of Car Rental Database System

- This is a rough entity relation design of the database required. Here, there are mainly 6 tables and the underlined attribute denotes the primary key of the table, and symbol (FK) determines the foreign key of the table while (U) determines that the given attribute must be unique.

User Classes and Characteristics

The main people who have been identified as regular system users as follow

1. CEO
2. Branch Manager
3. Accountant
4. Clerk
5. Customers

The users can be classified according to their frequency of use, a subset of product functions used technical expertise, security or privilege levels, educational level, or experience they have to the Car Rental System (CRS).

- CEO is supposed to have the following characteristics
    - CEO will have access to all the features and all the functions

- Branch managers are supposed to have the following characteristics
    - On the whole, the Branch manager and administrator spend comparatively lesser time than others on the system. Usually, they would access the system when they are required to make reports or something else.
    - They should have high technical expertise. They have to be fluent in the system usage and should be well aware of all the system functionalities.
    - Branch managers have access to all security levels as they have to access and maintain the system occasionally.
    - The branch manager and administrator should have experience in handling a system of this domain. He must be able to identify errors caused due to improper data entries.

- Clerk and Accountant are supposed to have the following characteristics
    - They spend the maximum amount of time on the system in the form of entering information, or updating the information.
    - They are not required to have technical expertise. The most they would need is to be computer literate to understand the working of the system and use it efficiently.
    - They have very low-security access. They cannot access quite a few functionalities of the system except the ones they really have to use.
    - They should have good communication skills and even if they have no experience in system usage, it’s not a big problem.
    - They should have experience in this area with a minimum diploma in IT.

- Customers are supposed to have the following characteristics
    - They have a minimal amount of access to the database.
    - They can just request their own profile.
    - They can also see how many cars and drivers are available on a particular franchise.
    - They can request a car and/or driver from the UI provided.
    - They are also provided with a GPS ID to track down the car at the time of booking services.
    - They can also see details of all the available drivers in order to facilitate overall better rental services.

Operating Environment

- Reliability:
    - The system shall be available for 90% of 24 hrs.
    - The system shall not be subject to failure for 20 hrs.
    - The system shall be repaired in at least one hour.
    - The system shall accurately deal with decimal numbers.

- Performance:
    - The system response for operations shall be at most 5 seconds.
    - The system shall handle five users per department.

- Supportability:
    - The system shall accommodate changes and enhancements that the development team is going to do in at most 5 days.
    - Also, easy management of authorization access is required.
    - A good, compact and simple user interface is required at each and every level of data access to the database.

- Design Constraints:
    - The system shall be designed according to IEEE Standards.
    - The system shall replace the existing system.
    - The system shall be programmed using JAVA.

- Connectivity Requirements:
    - For data synchronization, we have two options either to integrate our database into the cloud database system or we need to connect all database servers of each and every individual branch through physical links.
    - A centralized database system is required at both level inter and intra level for synchronizing data.

- Hardware Requirements:

    - The software should be run on any sort of desktop or laptop environment, regardless of the operating system.
    - The software also has the potential of running on tablets, but with a more simplified version.
    - Essential input/output devices are keyboards, mouse, and printers; nothing else is required but can be recommended if desired.

- Software Requirements:
    - To store all the records efficiently in our servers, we need at least 100 GB of server memory at each and every outlet.
    - Also, time to time backup of data is mandatory for extra safety.
    - RAM up to 16 GB for fast processing.
    - Server and domain host by Amazon

Product Function and User Privileges

1. Get Customer Information.
    - This function will fetch the detailed customer profile from the servers
    - Accessed by: Customers

2. Choose a package with vehicle model
    - This function will filter the data list according to the input parameters and display it to the user.
    - Accessed by: Customers

3. View Rental history
    - This function will display all the vehicle bookings made within the given time frame.
    - Accessed by: All user classes except customers

4. View car or driver average rating
    - This function returns the current average rating of a car or driver.
    - Accessed by: All user classes.

5. Create a final invoice
    - This function will generate the final invoice.
    - Accessed by: Accountant, Clerk

6. Collect payment
    - This function will calculate and collect the payable amount from the user
    - Accessed by: Clerk, Accountant, Customer

7. Get inventory report
    - This function will be accessible by the employee to check the number of vehicles and drivers available on the franchise.
    - Accessed by: All user classes except customer

8. Add vehicles
    - This function will allow employees to add the new vehicles and drivers into the system
    - Accessed by: Branch manager, Clerk

9. Get vehicle information
    - This function will display the information of the rented car to the user
    - Accessed by: All user classes.

10. Generate backup
    - This function will create the backup of the current database.
    - Accessed by: Branch manager

11. Get Tracking ID
    - This function will fetch GPS_ID associated with the booked car which allows one to track down the car.
    - Accessed by: All user classes except accountant.

12. Displaying Driver Details
    - This function will fetch details of the driver using the entered driver ID.
    - Accessed by: All user classes except accountant.

13. Displaying reviews
    - This function will display all car/driver reviews of previous trips.
    - Accessed by: All user classes except accountant.

Assumptions

- Each customer can book only one vehicle and/or driver.
- The rental price of cars is fixed and the same at every branch.
- Each customer who has rented the car has returned the car.
- The rate of the vehicle/driver will be per hour of usage.
- Each customer has returned the vehicle with no damage.
- Any damages on the trip are not accounted for by the company.
- Any car is tracked by just entering the GPS id.
- The salary of the driver doesn't depend on the number of fares completed.

Business Constraints

- The company hasn't specified any money related or space related constraint. Although, the only constraint mentioned was a Proper User Interface is absolutely necessary.
- For centralizing the database, the physical connection was checked out and cloud integration was recommended with minimal cost to the company.

Nouns and Verbs

Noun	Verb
Transportation	Digitize
Everyone	Change
Client	Visit
License Plate	Contains
Cars	Purchase
Chassis Number	Require
Records	Rate
Car List	Updated
Customer Service	Picking
Platform	Understand
Driver	Overwhelm
Clerical Staff	Provide
Worker	Hire
Salary Record	Acquire
Branch Manager	Provide
Accountant	Limited
Individual Branch	Eliminates
Proof ID	Handing
Employee ID	Coordinate
Unique ID	Pay
Details Form	Fill

Trip Record	Grasp
Maintenance Record	Synchronized
Customer	uses
Call Attendee	Assigns
City	Combine
Privileges	Integration
Rating	According
Review	Relate
Profit Record	Rent
Pickup Destination	Migrate
Dropoff Destination	Speculate
Time Period	Store
Money	Except
Modern World	Manage
Ideology	Delete
Start-up	Update
Local rental industry	Filter
Geography	Depends
Customer ID	Access
Excel Spreadsheet	Maintain
Task	Forget
Member	Discuss
Month	Fill
Problem	Implement

User Friendly UI	Handle
Franchise	Obtain
Owners	Check
Name	Determine
Data	Fulfil
Phone Call	Alot
Outlet	Calculate
Rate	Show
Rate Table	Books
Login Details	Works
	Parked
	Log in

Accepted Nouns and Verbs List

Candidate Entity Set	Candidate Attribute Set	Candidate Relationship Set
Customer Record	License plate	Maintaining
Car Record	Chassis number	Rate
Employee Record	Job position	Books
Maintenance Record	Proof ID	Parked
Rental Record	Employee ID	Works
Branch Record	Customer ID	Log in
Rate Table	Rating	
Login Details	Dropoff point	
	Pickup point	
	Time period	
	Salary	
	Rate	
	Phone number	
	City	
	Review	
	Branch ID	
	Car Type	
	Password	
	Access Type	

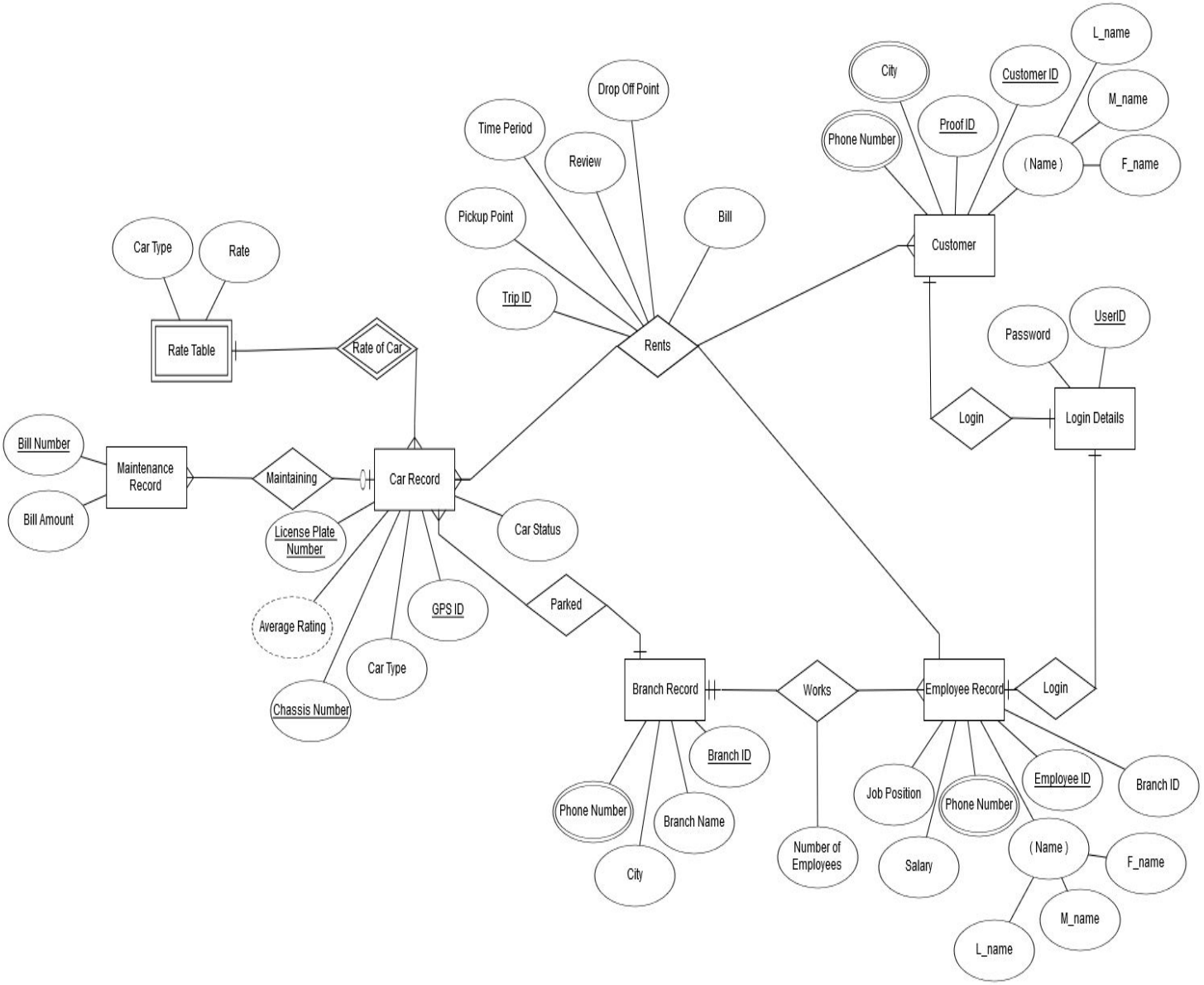
Rejected Nouns

Noun	Rejected Reason
Everyone	Vague
Transportation	Irrelevant
Customer Service	Association
Platform	General
Modern World	Vague
Call Attendee	Irrelevant
Details Form	Irrelevant
Ideology	Vague
Startup	Vague
Local Rental Industry	Vague
Geography	Vague
Excel Spreadsheet	Irrelevant
Task	Duplicate
Member	Duplicate
Month	Irrelevant
Solution	General
Problem	General
User Friendly UI	Irrelevant
Franchise	Duplicate
Owners	Duplicate
Name	Duplicate

Data	General
Client	Duplicate
Phone Call	Irrelevant
Outlet	Duplicate
Worker	Duplicate
Individual Branch	Duplicate
Unique ID	Duplicate
Money	Duplicate
Branch Manager	Attribute
Car List	Duplicate
Customer Service	Attribute
Platform	Generic
Driver	Attribute
Clerical Staff	Attribute
Worker	Duplicate
Branch Manager	Attribute
Accountant	Attribute
Individual Branch	Duplicate
Details Form	Irrelevant
Trip	Attribute
Customer	Attribute
City	Attribute
Privileges	Attribute
Rating	Attribute

Review	Attribute
Pickup Destination	Attribute
Dropoff Destination	Attribute
Time Period	Attribute
Money	Attribute
Rate	Attribute

Final ER Diagram



Functional Dependencies

1. Branch Record
    - (Branch ID) -> (BranchName, City) (Primary Key Dependencies)
2. Branch Record_PhoneNumber
    - (Branch ID,Phone Number)->(Branch ID,Phone Number)
3. Customer_city
    - (City, Customer ID)->(City, Customer ID)
4. Employee Record
    - Job position->Salary
    - Employee ID->(Branch ID, User ID, F_name, M_name, L_name, Job Position, Salary)
    - User ID->(Employee ID, Branch ID, F_name, M_name, L_name, Job Position, Salary)
5. Employee Record_PhoneNumber
    - (Phone Number, Employee ID)->(Phone Number, Employee ID)
6. Login Details
    - (User Id)->(Password)
7. Customer
    - (Customer ID)->(User ID, Proof ID, F_name, M_name, L_name)
    - (User ID)->(Customer ID, Proof ID, F_name, M_name, L_name)
8. Customer_PhoneNumber
    - (Phone Number, Customer ID)->(Phone Number, Customer ID)
9. Car Record
    - (Chassis Number)->(Chassis Number, Car Status, Car Type, Branch ID, License Plate Number, GPS ID, Average Rating)
    - (GPS ID)->(Chassis Number, Car Status, Car Type, Branch ID, License Plate Number, GPS ID, Average Rating)
    - (License Plate Number)->(Chassis Number, Car Status, Car Type, Branch ID, License Plate Number, GPS ID, Average Rating)
10. Rents
    - Trip Id ->(Employee ID, Customer ID, Chassis Number, Time Period, Drop Off Point, Pick Up Point, Bill, Review)
11. Maintenance Record
    - (Bill Number)->(Bill Amount, Chassis Number)
12. Rate Table
    - (Car Type)->(Rate)

1NF(First Normal Form)

1. All tables are in 1NF form. Because all multivalued attributes are already decomposed into other tables like employeerecord_phonenumber, etc.

2NF(Second Normal Form)

1. There is no partial dependency.

3NF(Third Normal Form)

In the employee table, a job position which is a non prime attribute determines another non prime attribute salary. Thus a transitive dependency is formed. Thus, another table

is formed consisting of job position and salary.

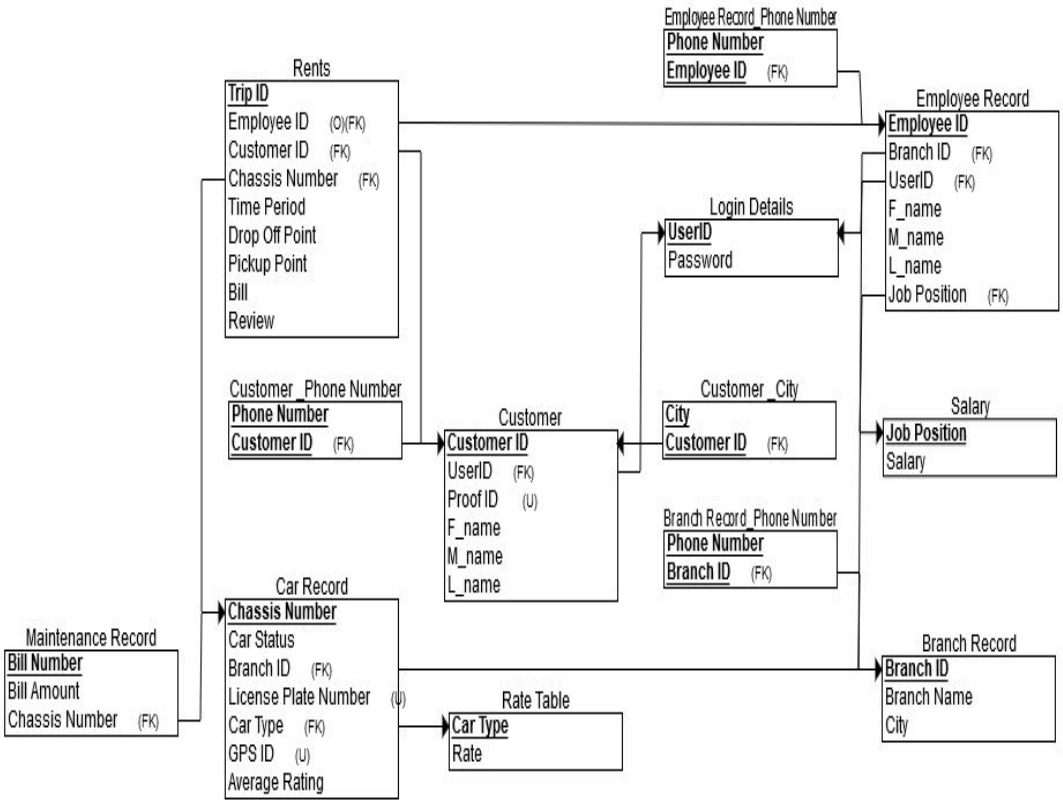
Final Schema

- Rate Table(Rate, Car Type)
- Maintenance Record(Bill Number, Bill Amount, Chassis Number)
- Rents(Trip ID, Employee ID, Customer ID, Chassis Number, Time Period, Drop Off Point, Pick Up Point, Bill, Review)
- Customer_Phone Number(Phone Number, Customer ID)
- Car Record(Chassis Number, Car Status, Car Type, Branch ID, License Plate Number, GPS ID, Average Rating)
- Customer(Customer ID, Proof ID, F_name, M_name, L_name)
- CustomerID_UserID(Customer ID, User ID)
- Employee Record_Phone Number(Phone Number, Employee ID)
- Login Details(User ID, Password)
- Employee Record(Employee ID, Branch ID, User ID, F_name, M_name, L_name, Job Position)
- EmployeeID_Salary(Employee ID, User ID)
- Customer_City(City, Customer ID)
- Branch Record_Phone Number(Phone Number, Branch ID)
- Branch Record(Branch ID, Branch Name, City)

CSV Files

For the data population, we have to import data using csv files. Here is the link for the same.
https://drive.google.com/file/d/1Vux8Msw3g2yjRtfSE0GQ1GtDs6iFF2LS/view?usp=sharing

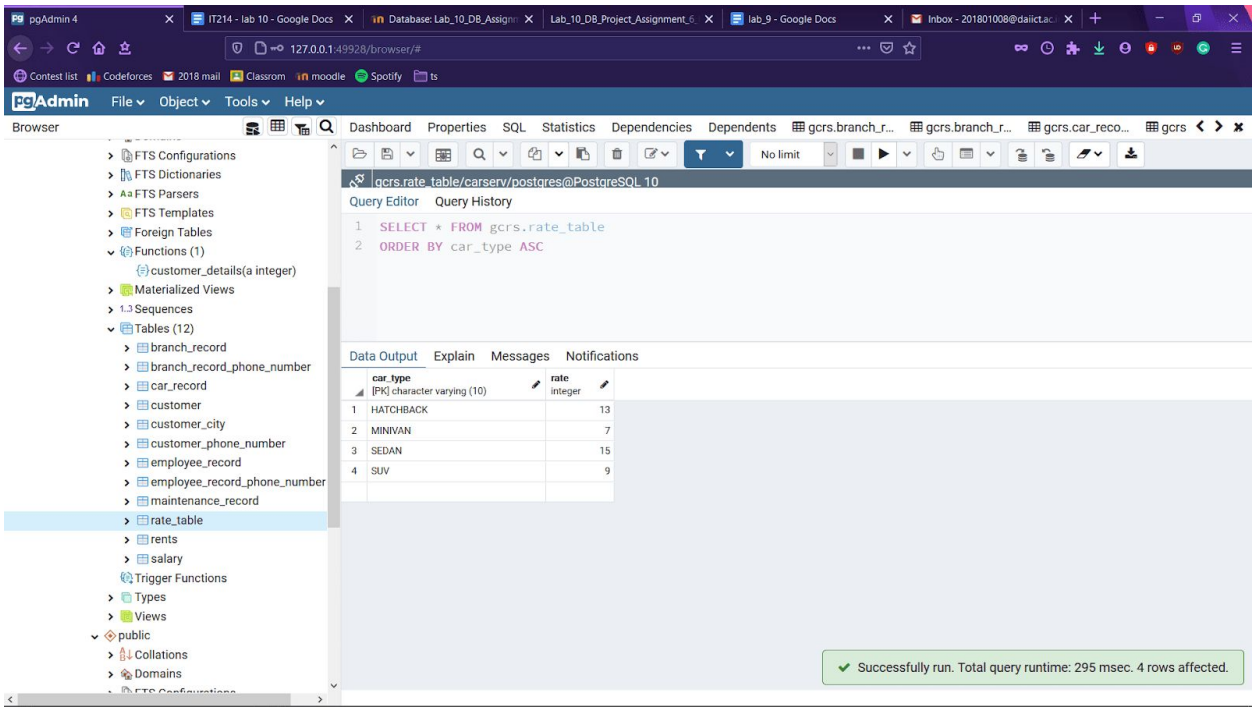
Final Relational Schema Diagram



DDL Scripts

Rate Table

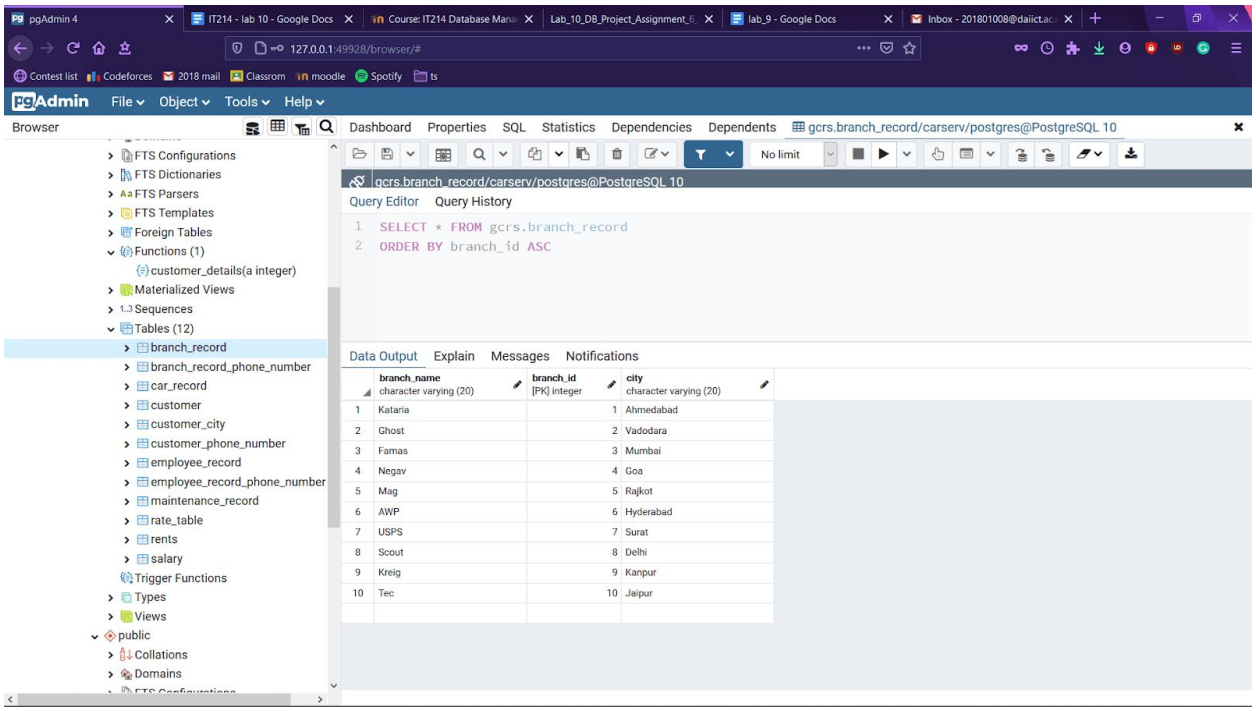
```
CREATE TABLE Rate_Table
(
  Car_Type VARCHAR(10) NOT NULL,
  Rate INT NOT NULL,
  PRIMARY KEY (Car_Type)
);
```



Number of records = 4

Branch Record

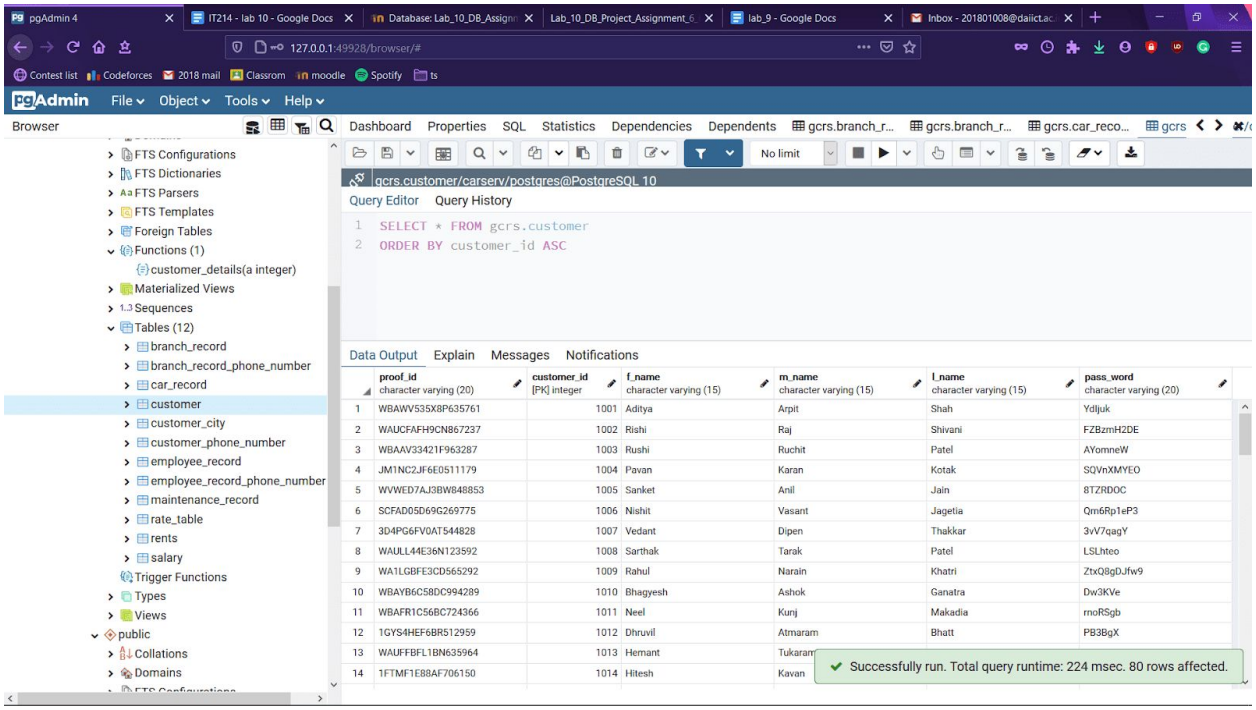
```
CREATE TABLE Branch_Record
(
  Branch_Name VARCHAR(20) NOT NULL,
  Branch_ID INT NOT NULL,
  City VARCHAR(20) NOT NULL,
  PRIMARY KEY (Branch_ID)
);
```



Number of records = 10

Customer

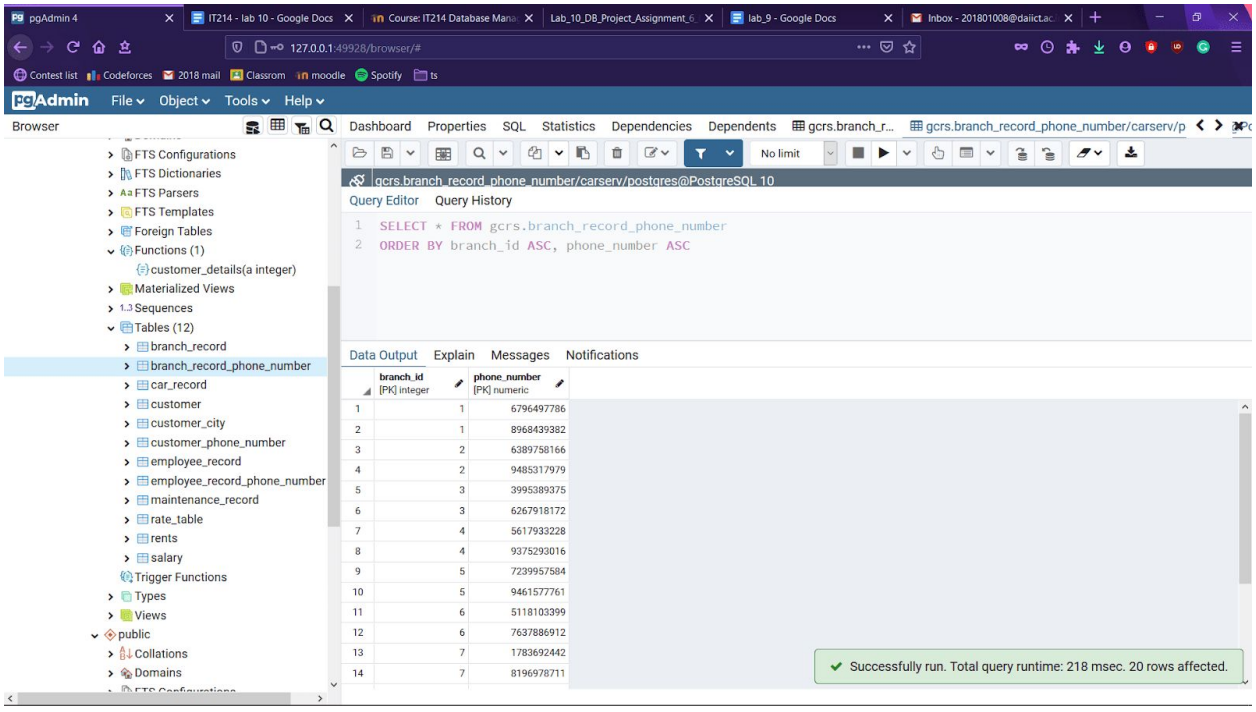
```
CREATE TABLE Customer
(
  Proof_ID  VARCHAR(20) NOT NULL,
  Customer_ID INT NOT NULL,
  F_name VARCHAR(15) NOT NULL,
  M_name VARCHAR(15) NOT NULL,
  L_name VARCHAR(15) NOT NULL,
  Pass_word VARCHAR(20) NOT NULL,
  PRIMARY KEY (Customer_ID),
  UNIQUE (Proof_ID)
);
```



Number of records = 80

Branch Record Phone Number

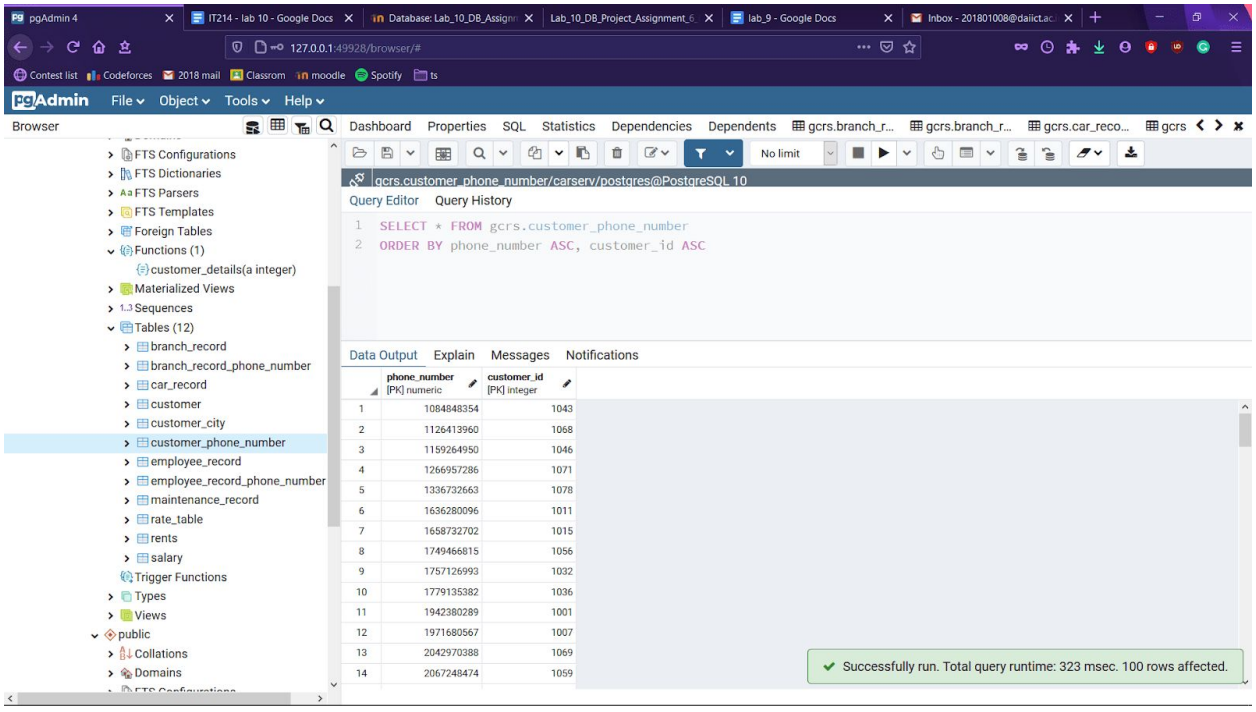
```
CREATE TABLE Branch_Record_Phone_Number
(
  Branch_ID INT NOT NULL,
  Phone_Number NUMERIC NOT NULL,
  PRIMARY KEY (Branch_ID, Phone_Number),
  FOREIGN KEY (Branch_ID) REFERENCES Branch_Record(Branch_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```



Number of records = 20

Customer Phone Number

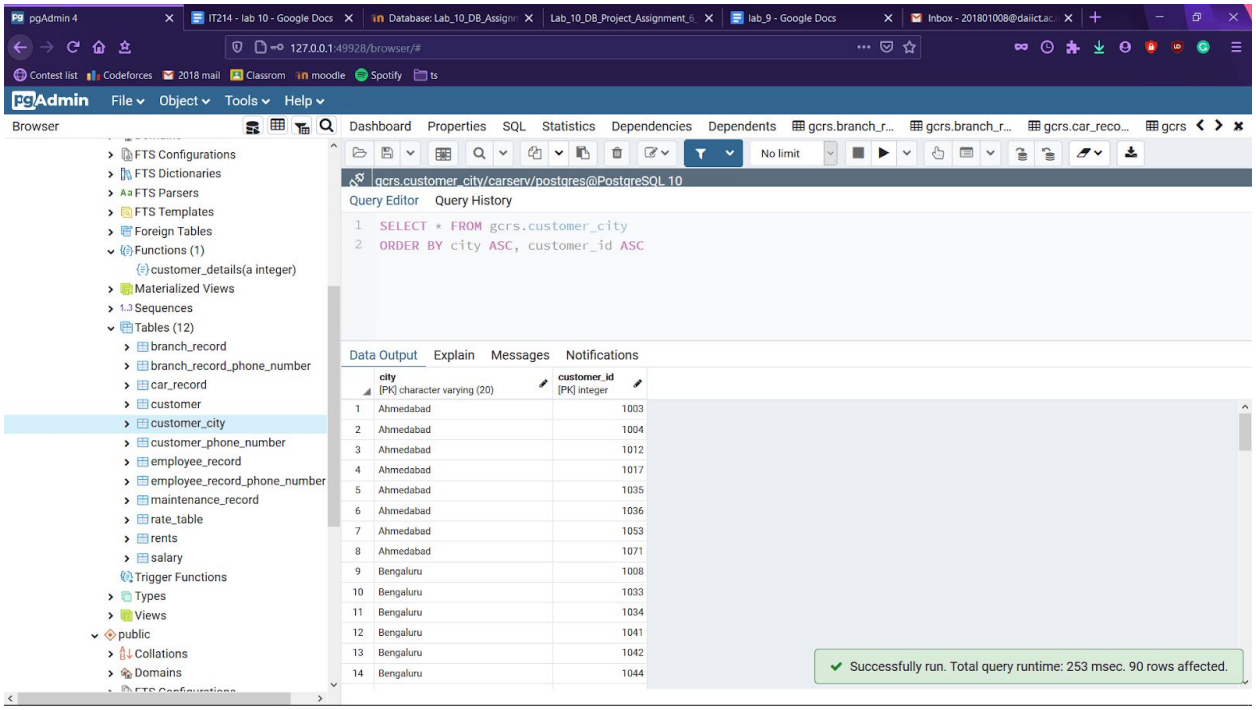
```
CREATE TABLE Customer_Phone_Number
(
  Phone_Number NUMERIC NOT NULL,
  Customer_ID INT NOT NULL,
  PRIMARY KEY (Phone_Number, Customer_ID),
  FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```



Number of records = 100

Customer City

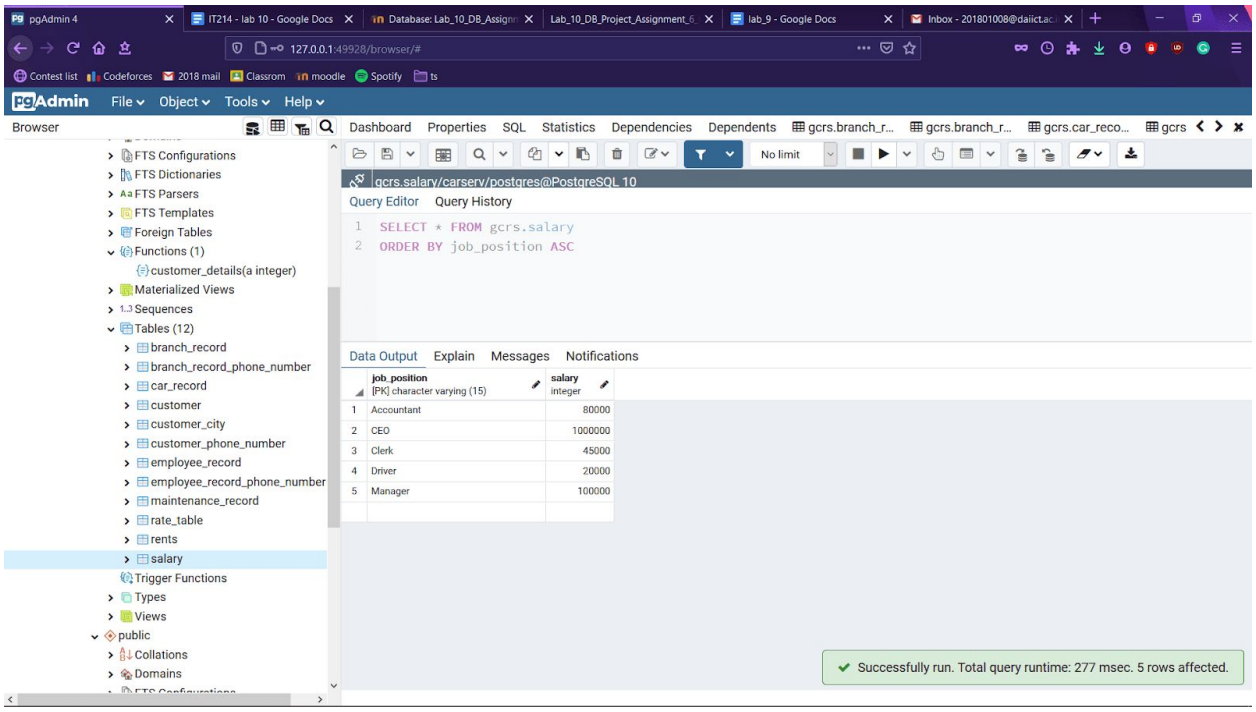
```
CREATE TABLE Customer_City
(
  City VARCHAR(20) NOT NULL,
  Customer_ID INT NOT NULL,
  PRIMARY KEY (City, Customer_ID),
  FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```



Number of records = 90

Salary

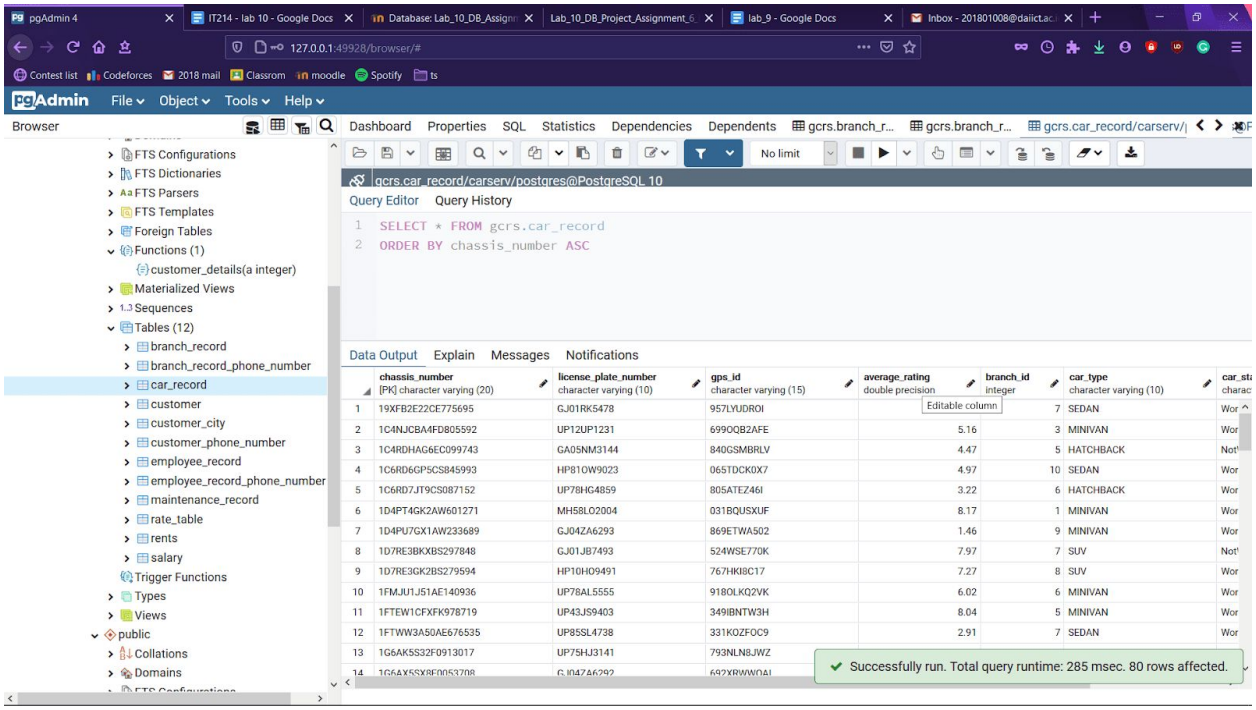
```
CREATE TABLE Salary
(
  Job_Position VARCHAR(15) NOT NULL,
  Salary INT NOT NULL,
  PRIMARY KEY (Job_Position)
);
```



Number of records = 5

Car Record

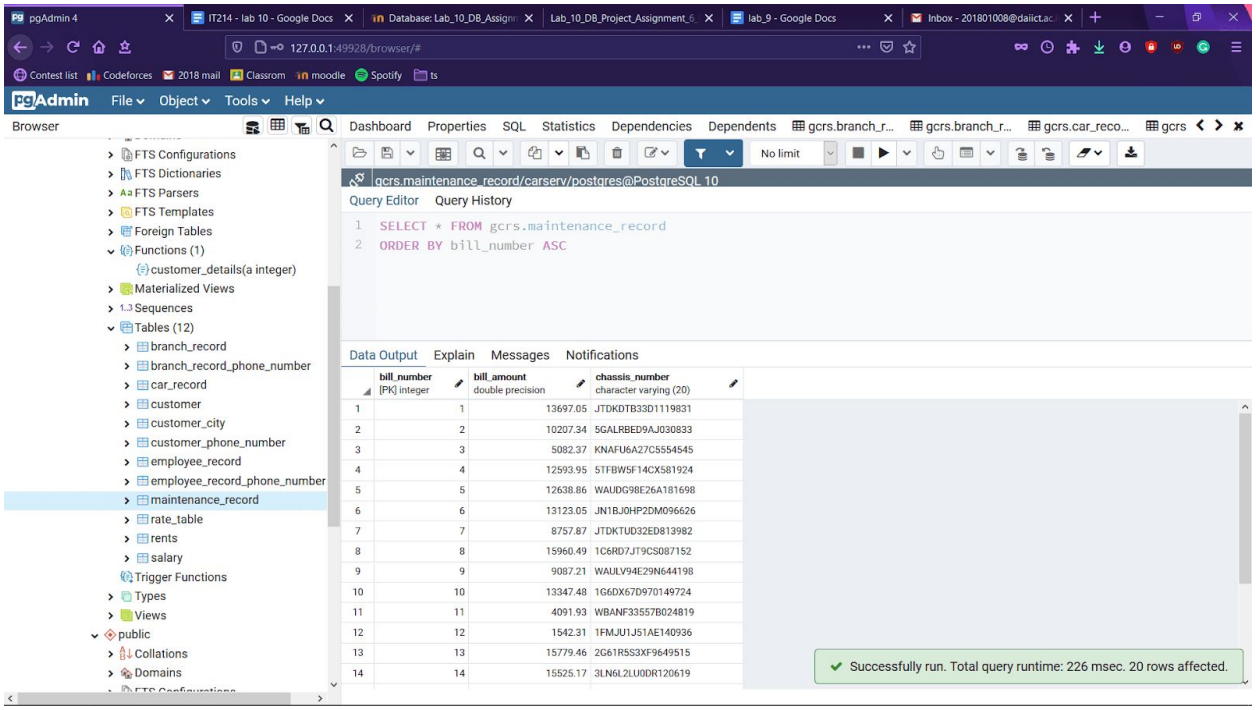
```
CREATE TABLE Car_Record
(
  Chassis_Number VARCHAR(20) NOT NULL,
  License_Plate_Number VARCHAR(10) NOT NULL,
  GPS_ID VARCHAR(15) NOT NULL,
  Average_Rating FLOAT NOT NULL,
  Branch_ID INT NOT NULL,
  Car_Type VARCHAR(10) NOT NULL,
  Car_Status VARCHAR(15) NOT NULL,
  PRIMARY KEY (Chassis_Number),
  FOREIGN KEY (Branch_ID) REFERENCES Branch_Record(Branch_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (Car_Type) REFERENCES Rate_Table(Car_Type)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  UNIQUE (License_Plate_Number),
  UNIQUE (GPS_ID)
);
```



Number of records = 80

Maintenance Record

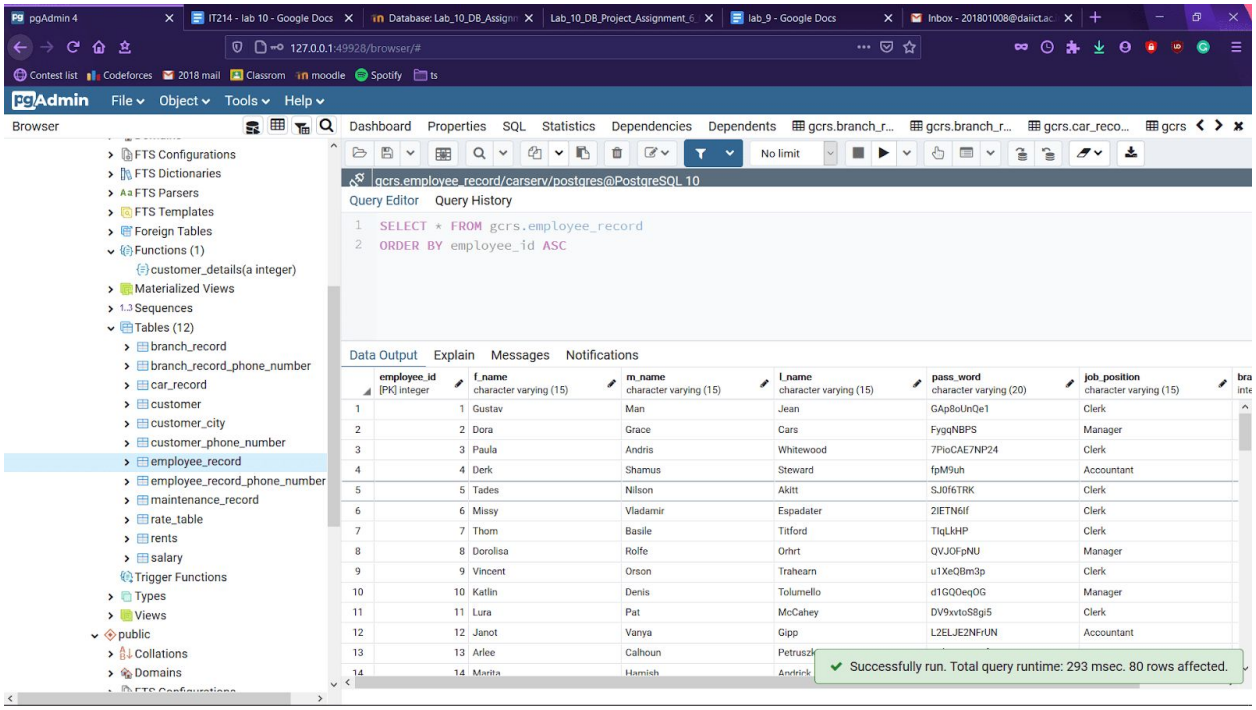
```
CREATE TABLE Maintenance_Record
(
  Bill_Number INT NOT NULL,
  Bill_Amount FLOAT NOT NULL,
  Chassis_Number VARCHAR(20) NOT NULL,
  PRIMARY KEY (Bill_Number),
  FOREIGN KEY (Chassis_Number) REFERENCES Car_Record(Chassis_Number)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```



Number of records = 20

Employee Record

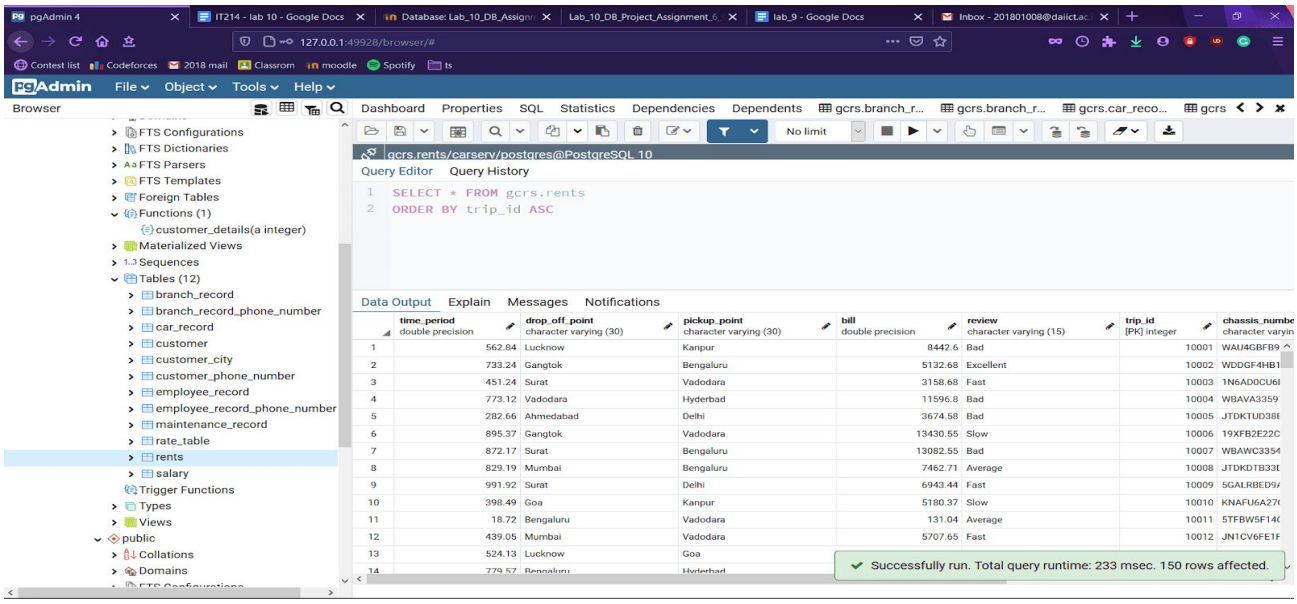
```
CREATE TABLE Employee_Record
(
  Employee_ID INT NOT NULL,
  F_name VARCHAR(15) NOT NULL,
  M_name VARCHAR(15) NOT NULL,
  L_name VARCHAR(15) NOT NULL,
  Pass_word VARCHAR(20) NOT NULL,
  Job_Position VARCHAR(15) NOT NULL,
  Branch_ID INT NOT NULL,
  PRIMARY KEY (Employee_ID),
  FOREIGN KEY (Branch_ID) REFERENCES Branch_Record(Branch_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (Job_Position) REFERENCES Salary(Job_Position)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```



Number of records = 80

Rents

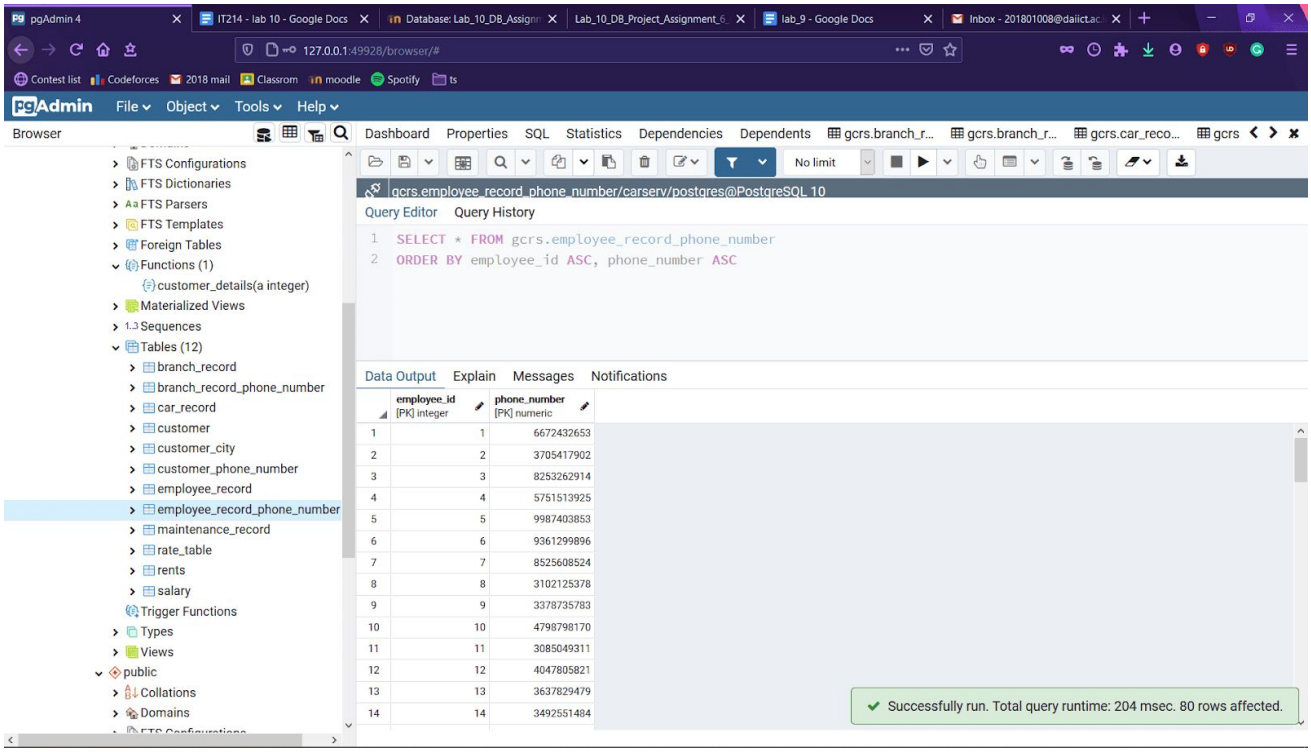
```
CREATE TABLE Rents
(
  Time_Period FLOAT NOT NULL,
  Drop_Off_Point VARCHAR(30) NOT NULL,
  Pickup_Point VARCHAR(30) NOT NULL,
  Bill FLOAT NOT NULL,
  Review VARCHAR(15) NOT NULL,
  Trip_ID INT NOT NULL,
  Chassis_Number VARCHAR(20) NOT NULL,
  Customer_ID INT NOT NULL,
  Employee_ID INT,
  PRIMARY KEY (Trip_ID),
  FOREIGN KEY (Chassis_Number) REFERENCES Car_Record(Chassis_Number)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (Employee_ID) REFERENCES Employee_Record(Employee_ID)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```



Number of records = 150

Employee Record Phone Number

```
CREATE TABLE Employee_Record_Phone_Number
(
  Employee_ID INT NOT NULL,
  Phone_Number NUMERIC NOT NULL,
  PRIMARY KEY (Phone_Number, Employee_ID),
  FOREIGN KEY (Employee_ID) REFERENCES Employee_Record(Employee_ID)
 ON DELETE CASCADE
  ON UPDATE CASCADE
 );
```

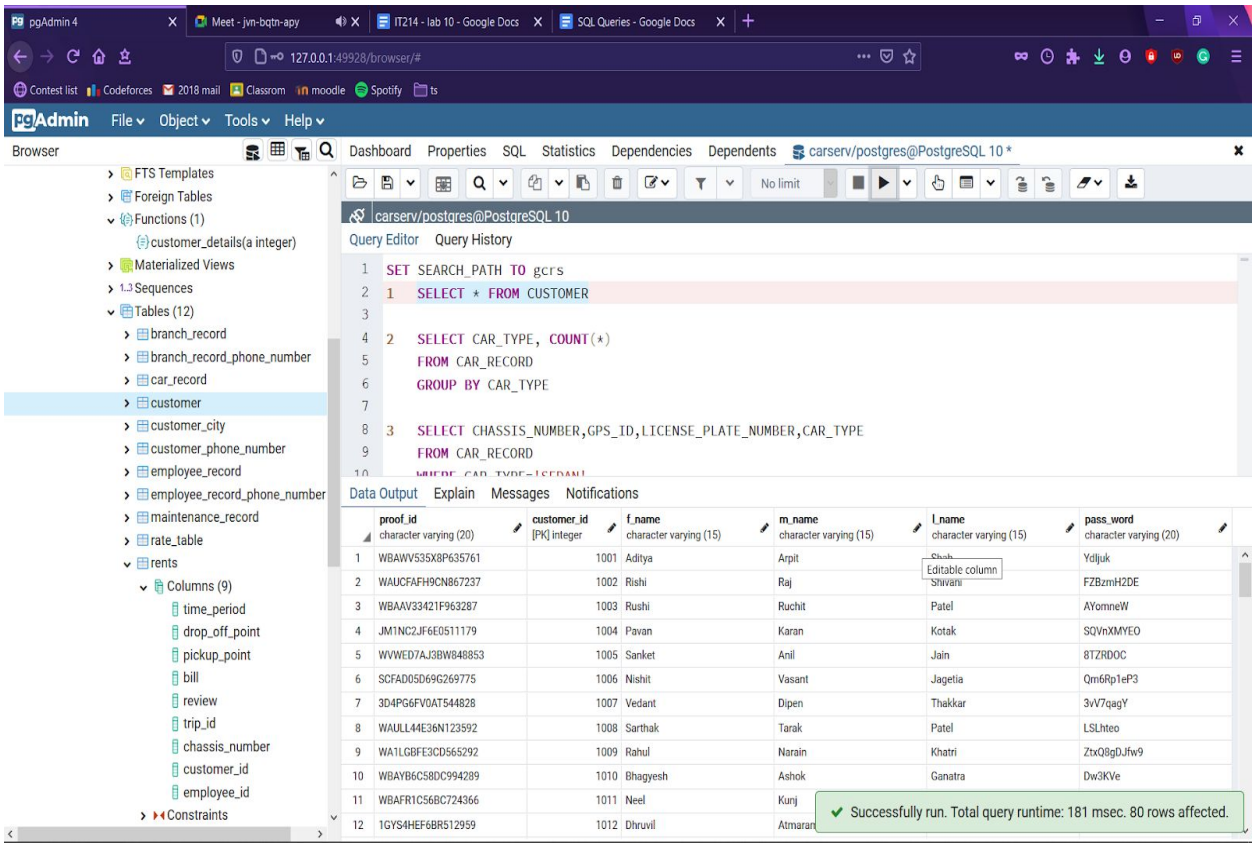


Number of records = 80

SQL Queries and Results

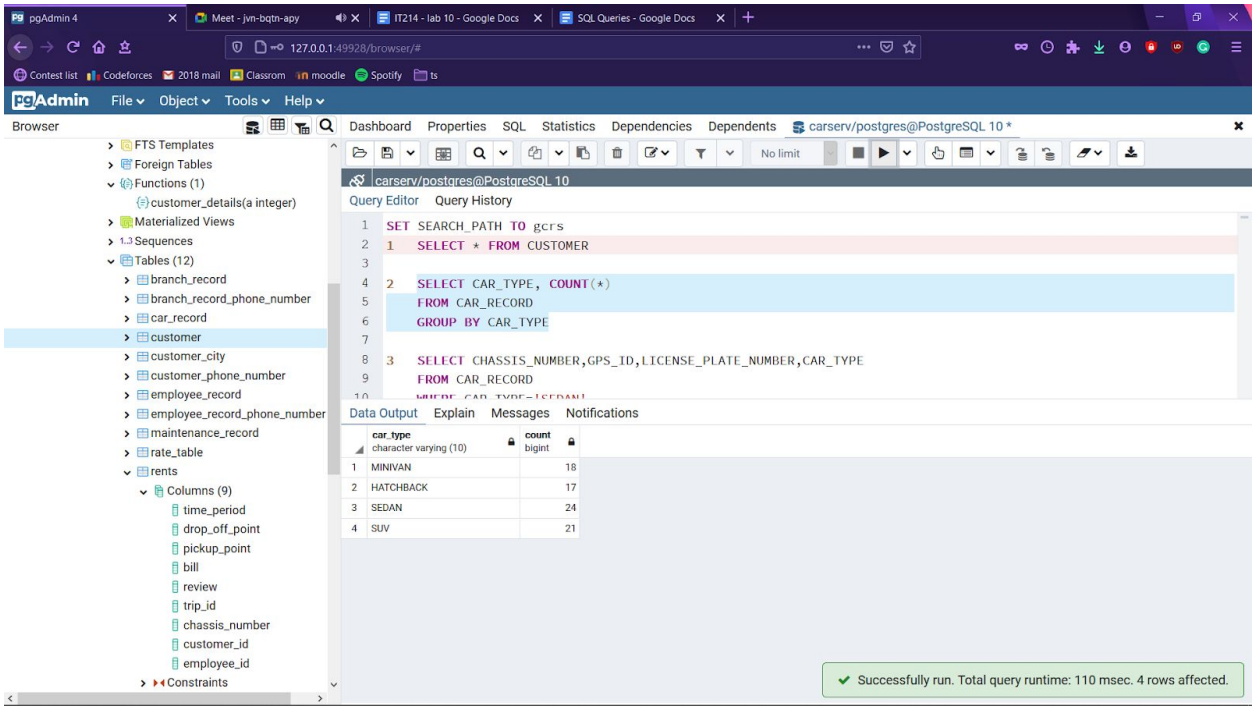
1. Select all records from customer

```
SELECT * FROM CUSTOMER
```



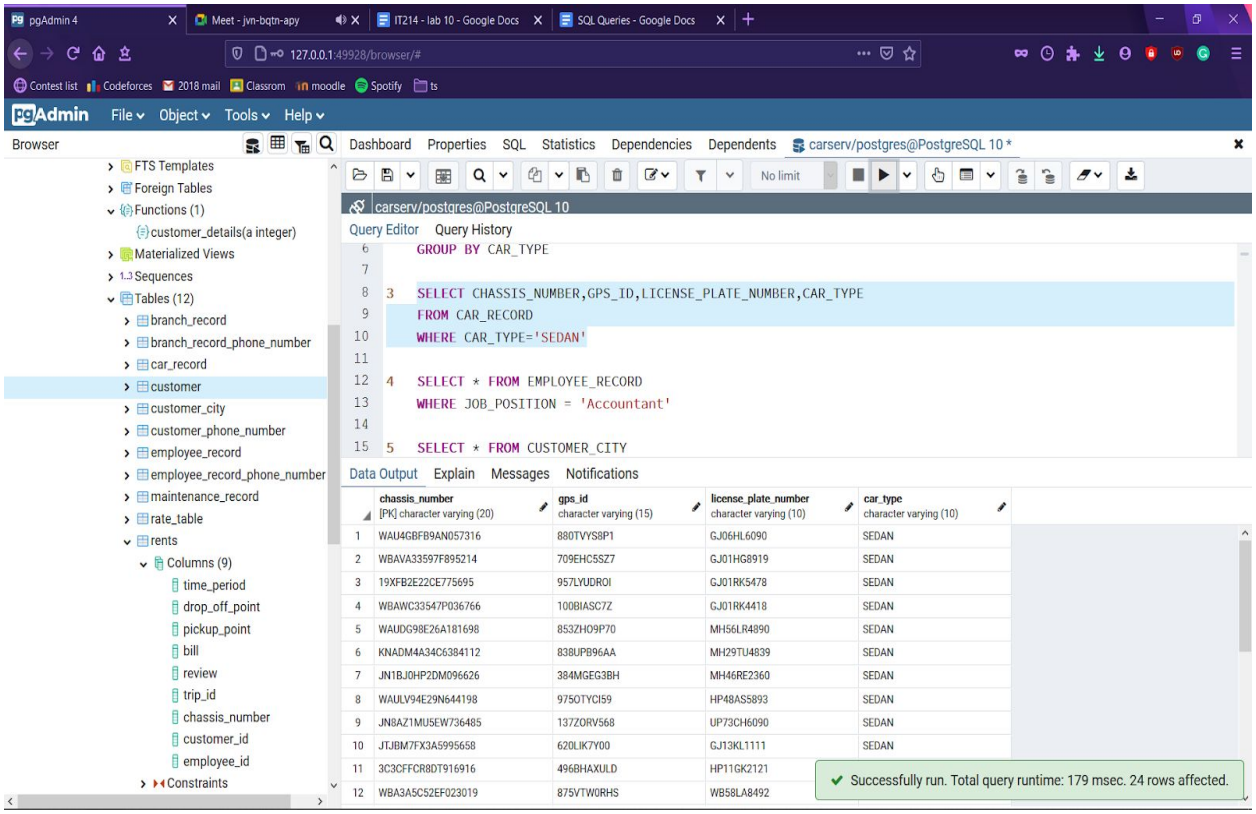
2 . Get count of number of cars category wise

```
SELECT CAR_TYPE, COUNT(*)
FROM CAR_RECORD
GROUP BY CAR_TYPE
```



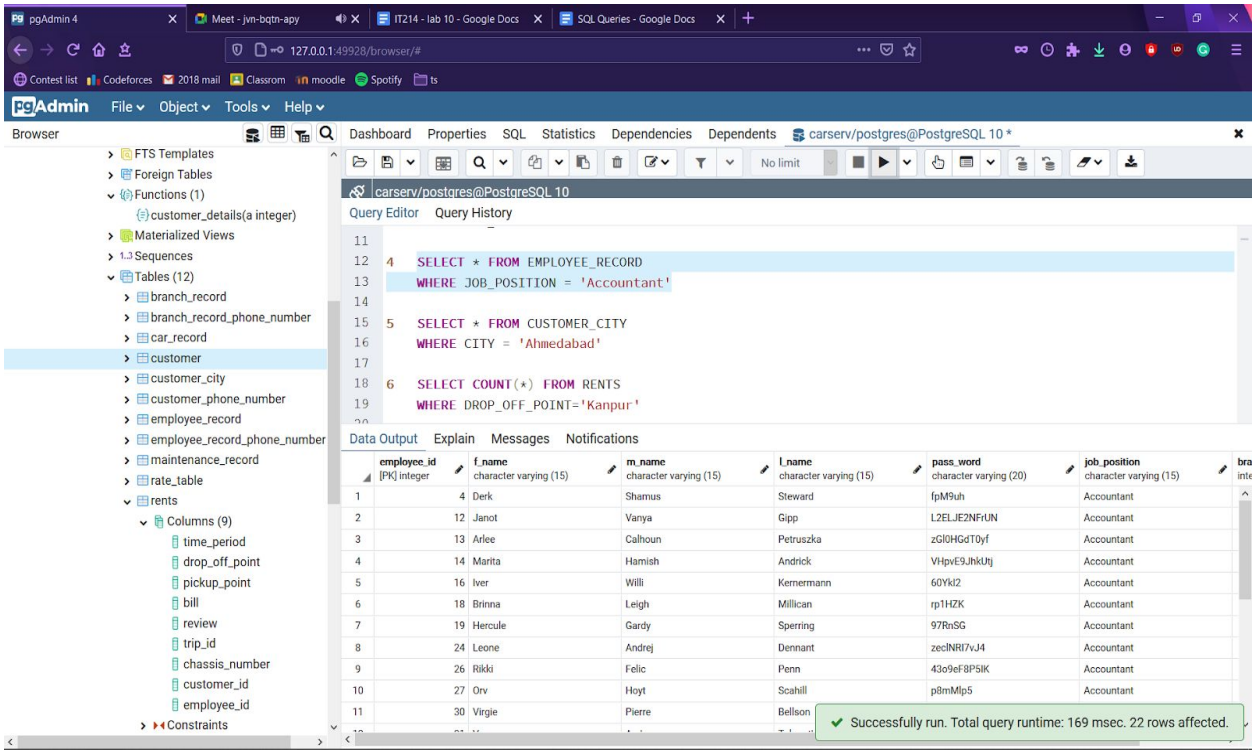
3 Get chassis number, license plate number and gps id where car category is sedan.

```
SELECT CHASSIS_NUMBER,GPS_ID,LICENSE_PLATE_NUMBER,CAR_TYPE
FROM CAR_RECORD
WHERE CAR_TYPE='SEDAN'
```



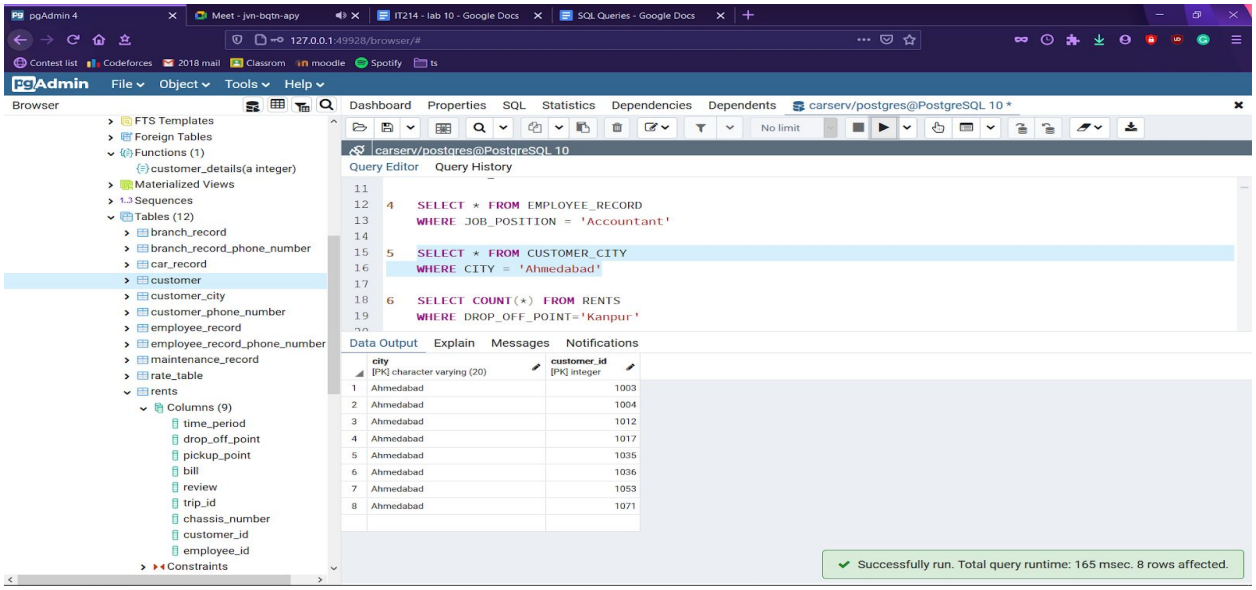
4 Select employees whose job position is accountant.

```
SELECT * FROM EMPLOYEE_RECORD
WHERE JOB_POSITION = 'Accountant'
```



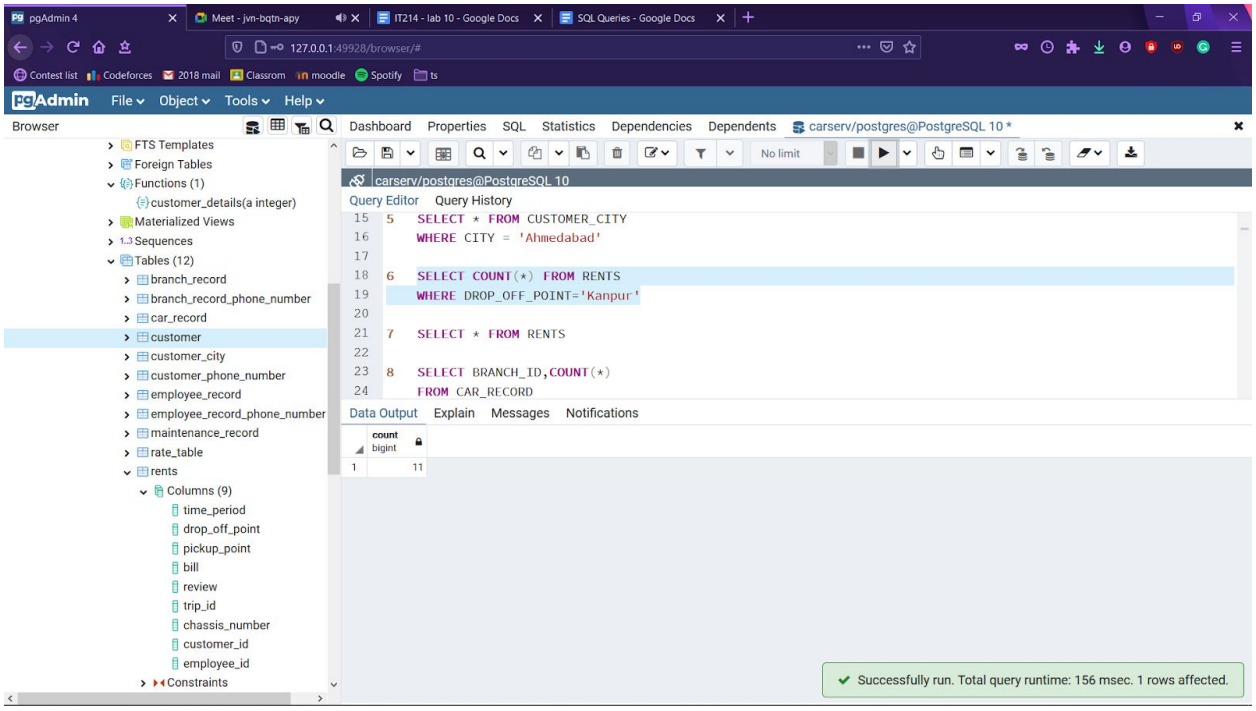
5 Get a customer id who lives in Ahmedabad.

```
SELECT * FROM CUSTOMER_CITY
WHERE CITY = 'Ahmedabad'
```



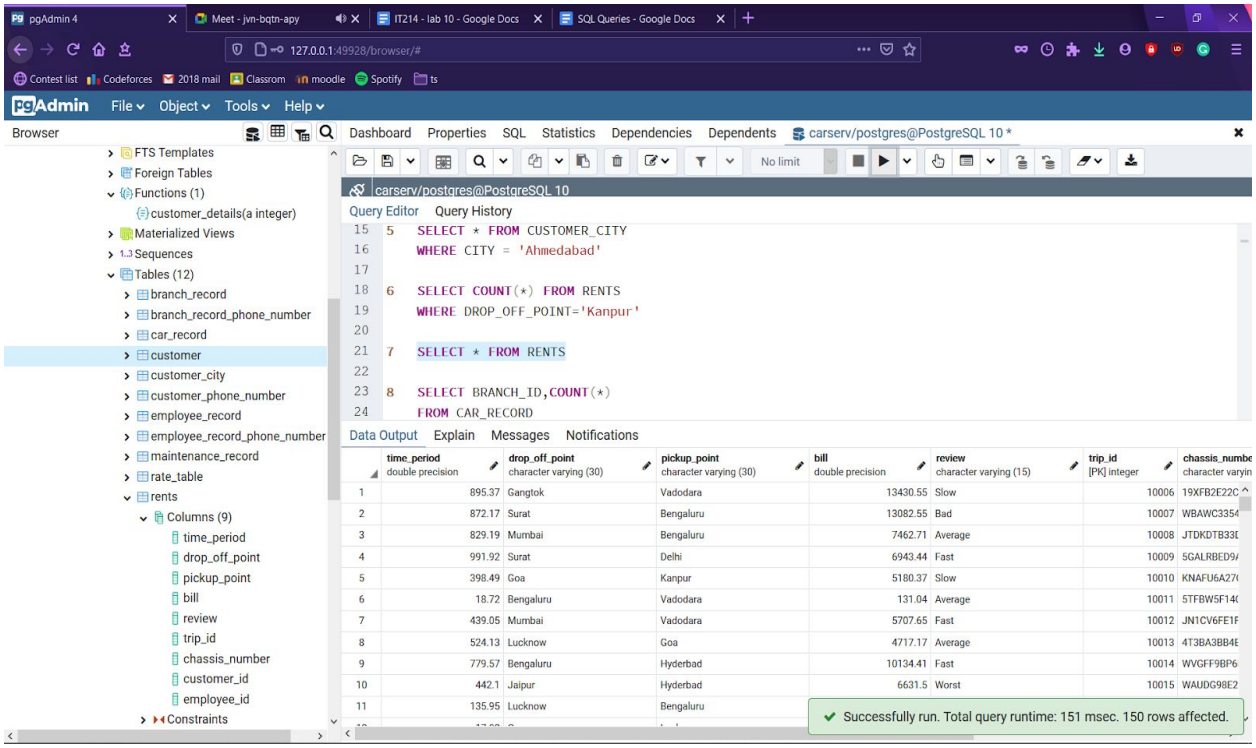
6 Count the number of rent histories whose drop-off point is Kanpur.

```
SELECT COUNT(*) FROM RENTS
WHERE DROP_OFF_POINT='Kanpur'
```



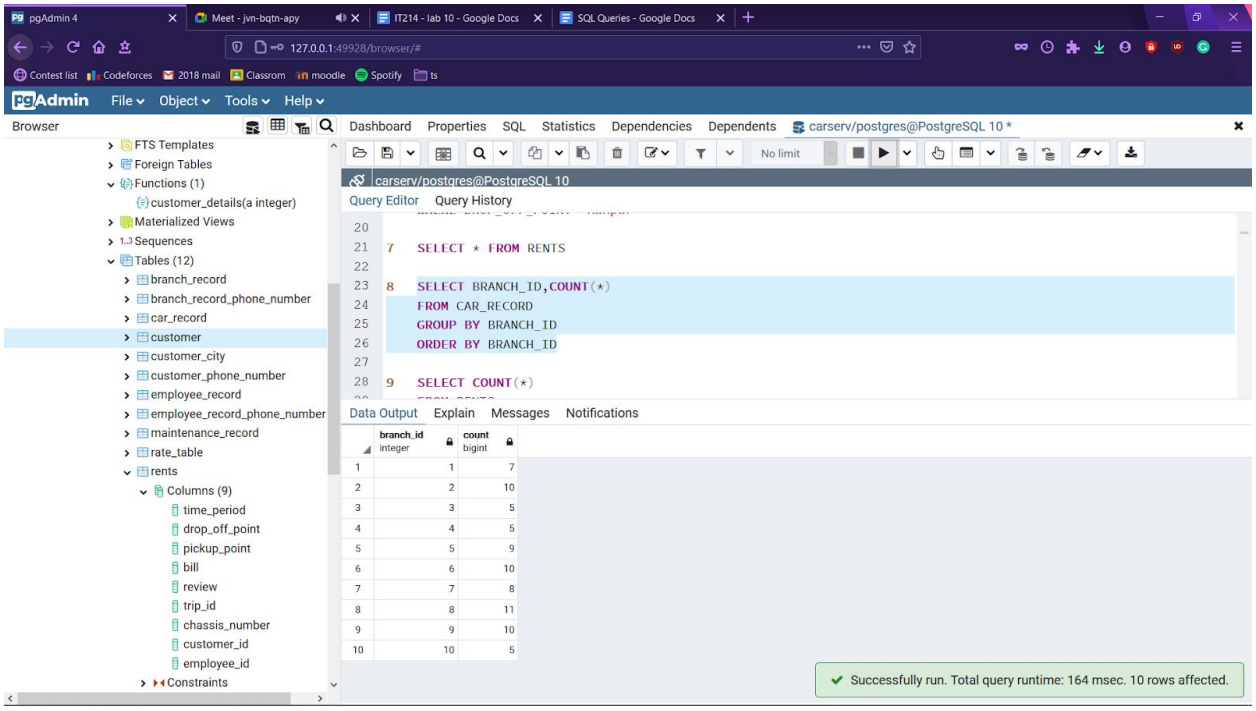
7 Display rental history.

SELECT * FROM RENTS



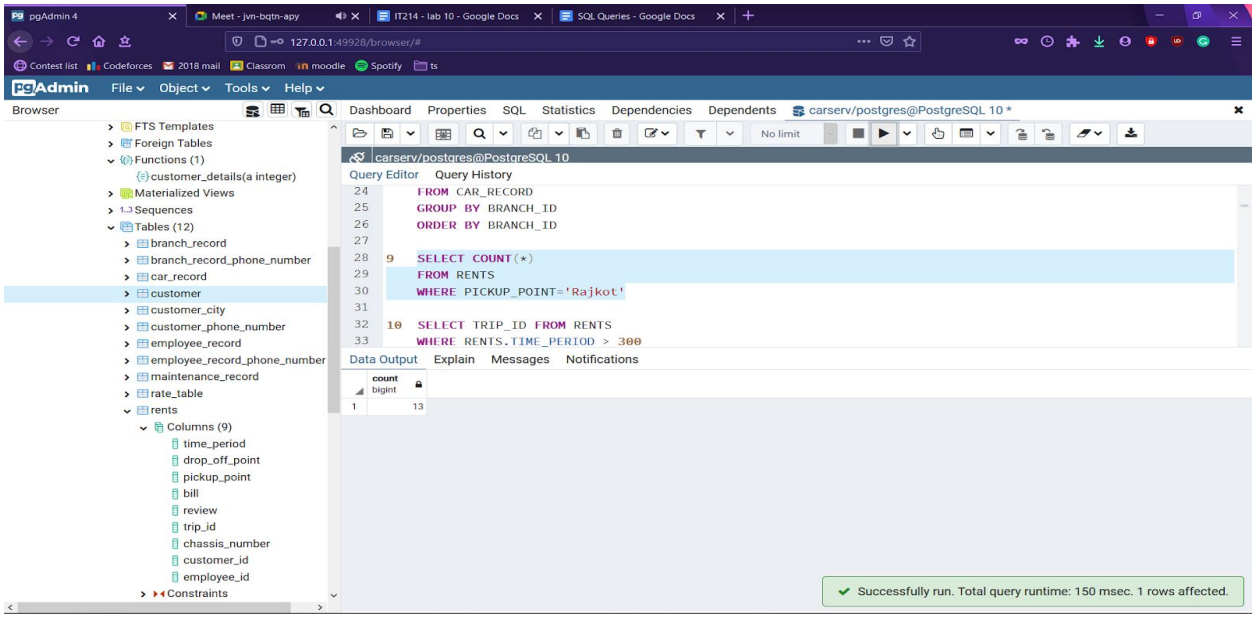
8 Find the number of cars in a particular branch.

```
SELECT BRANCH_ID,COUNT(*)
FROM CAR_RECORD
GROUP BY BRANCH_ID
ORDER BY BRANCH_ID
```



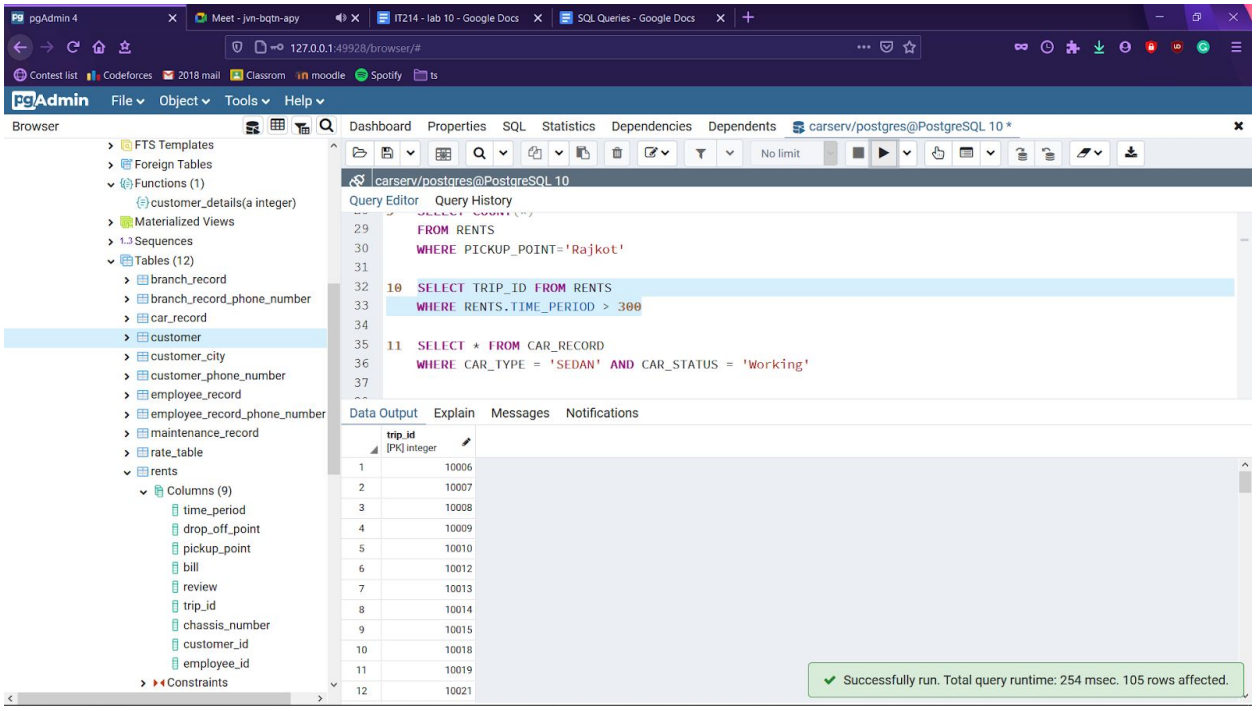
9 Count the number of rent histories whose pick-up point is Rajkot.

```
SELECT COUNT(*)
FROM RENTS
WHERE PICKUP_POINT='Rajkot'
```



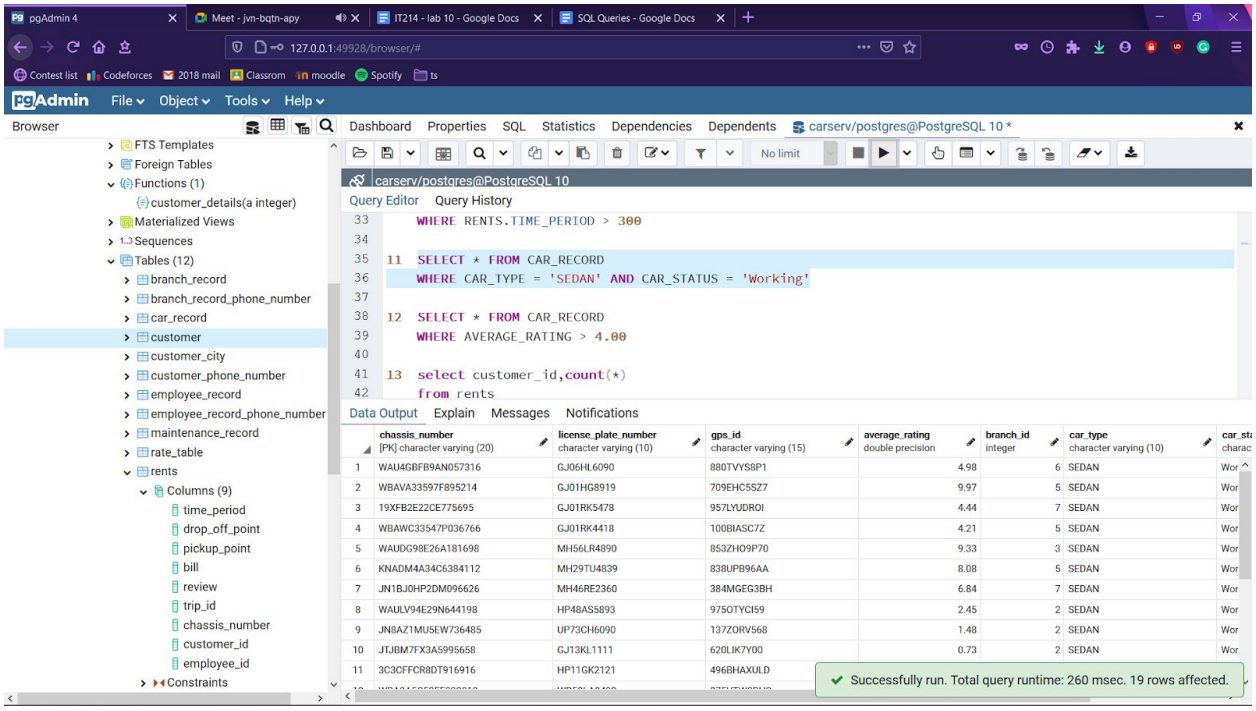
10 Display the rent id who used the car for more than 300 min

```
SELECT TRIP_ID FROM RENTS
WHERE RENTS.TIME_PERIOD > 300
```



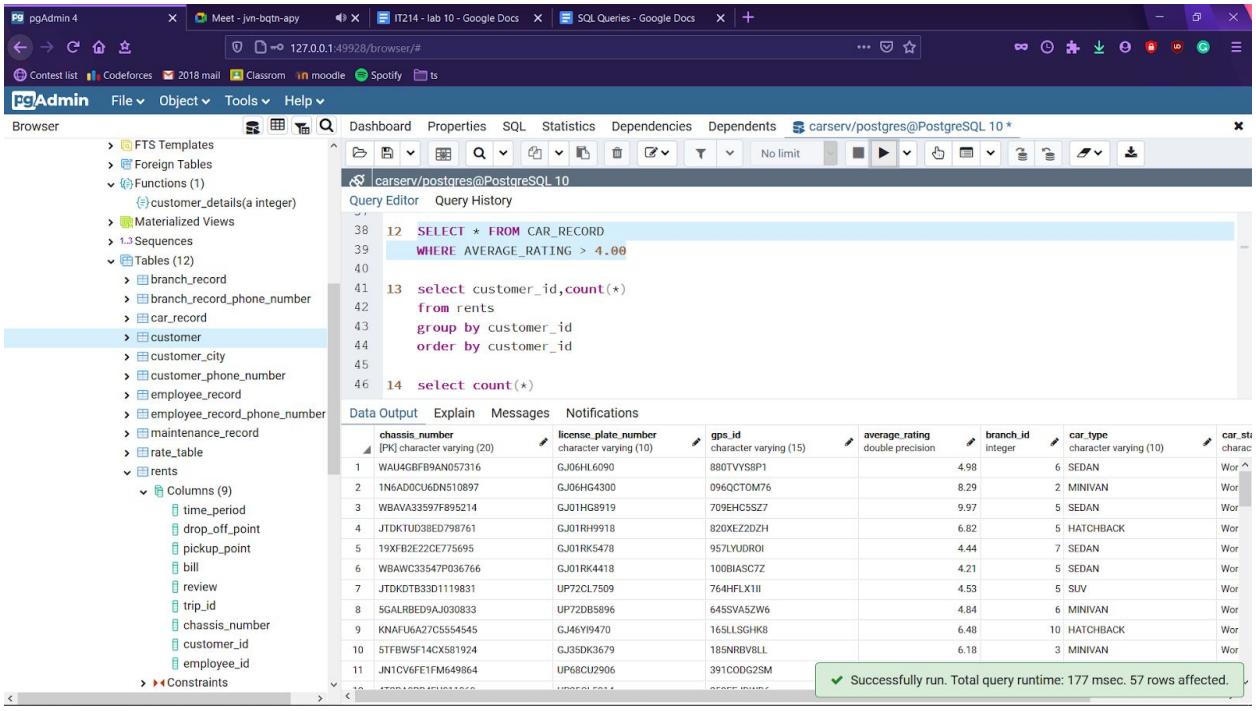
11 Select cars which are working and car type is sedan.

```
SELECT * FROM CAR_RECORD
WHERE CAR_TYPE = 'SEDAN' AND CAR_STATUS = 'Working'
```



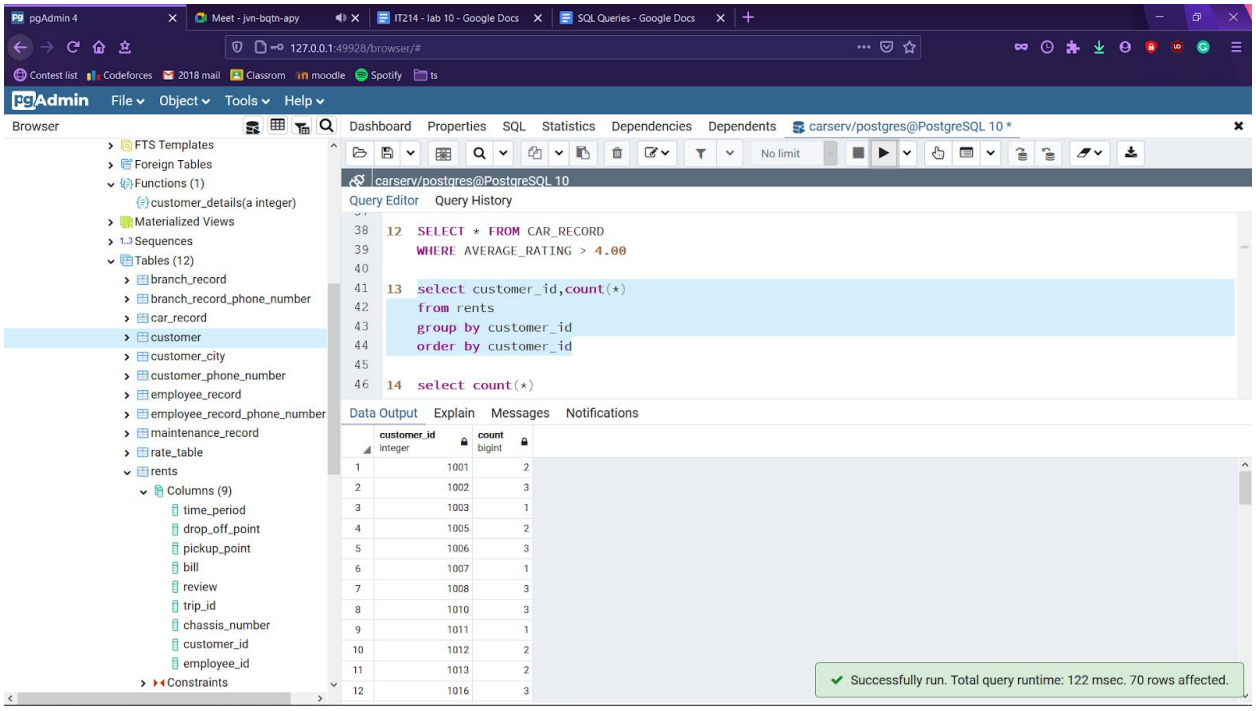
12 Select all cars with average rating higher than 4.00

```
SELECT * FROM CAR_RECORD
WHERE AVERAGE_RATING > 4.00
```



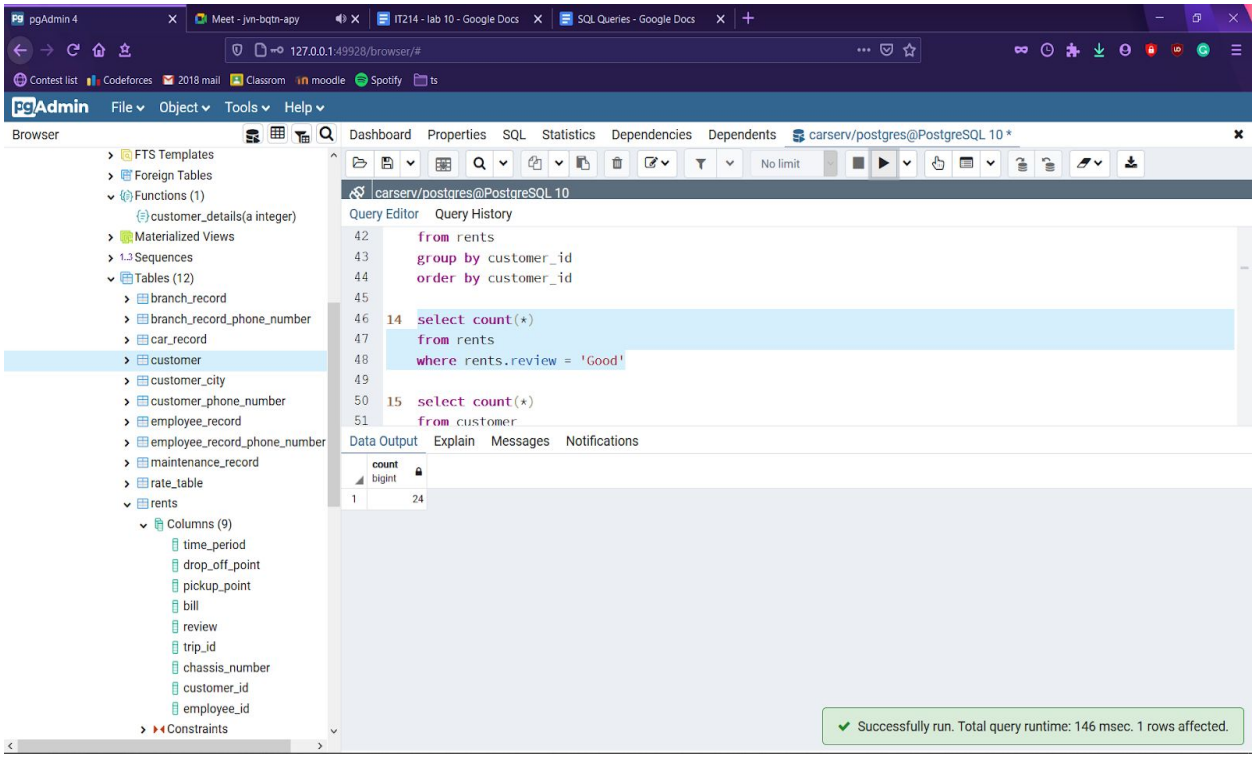
13 Find the number of bookings done by each customer.

```
select customer_id,count(*)
from rents
group by customer_id
order by customer_id
```



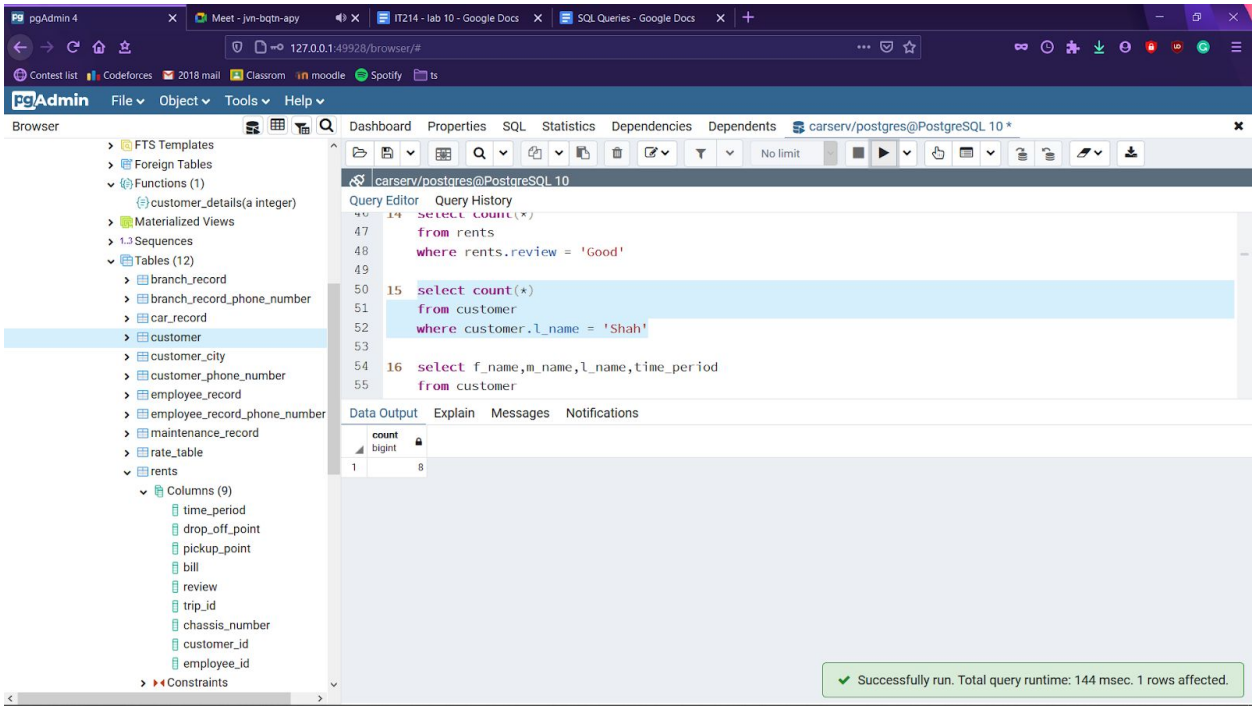
14 Find the number of customers who gave reviews as good.

```
select count(*)
from rents
where rents.review = 'Good'
```



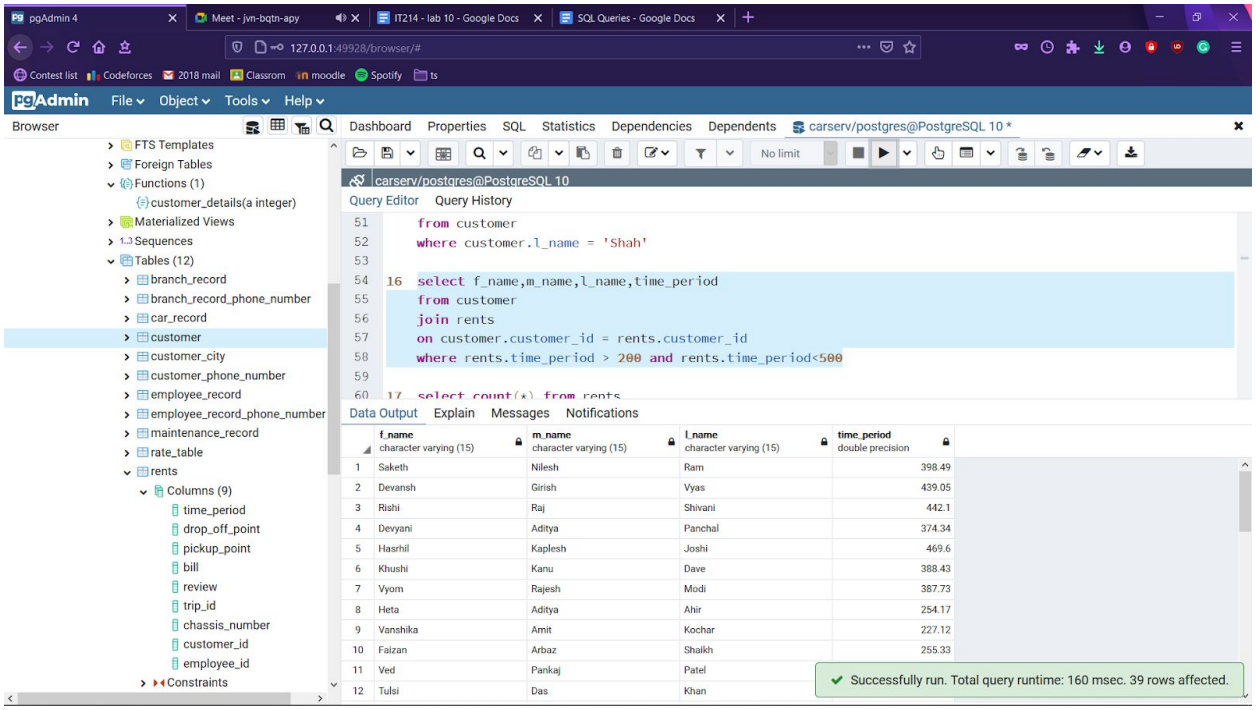
15 List the number of customers with the surname ” Shah “.

```
select count(*)
from customer
where customer.l_name = 'Shah'
```



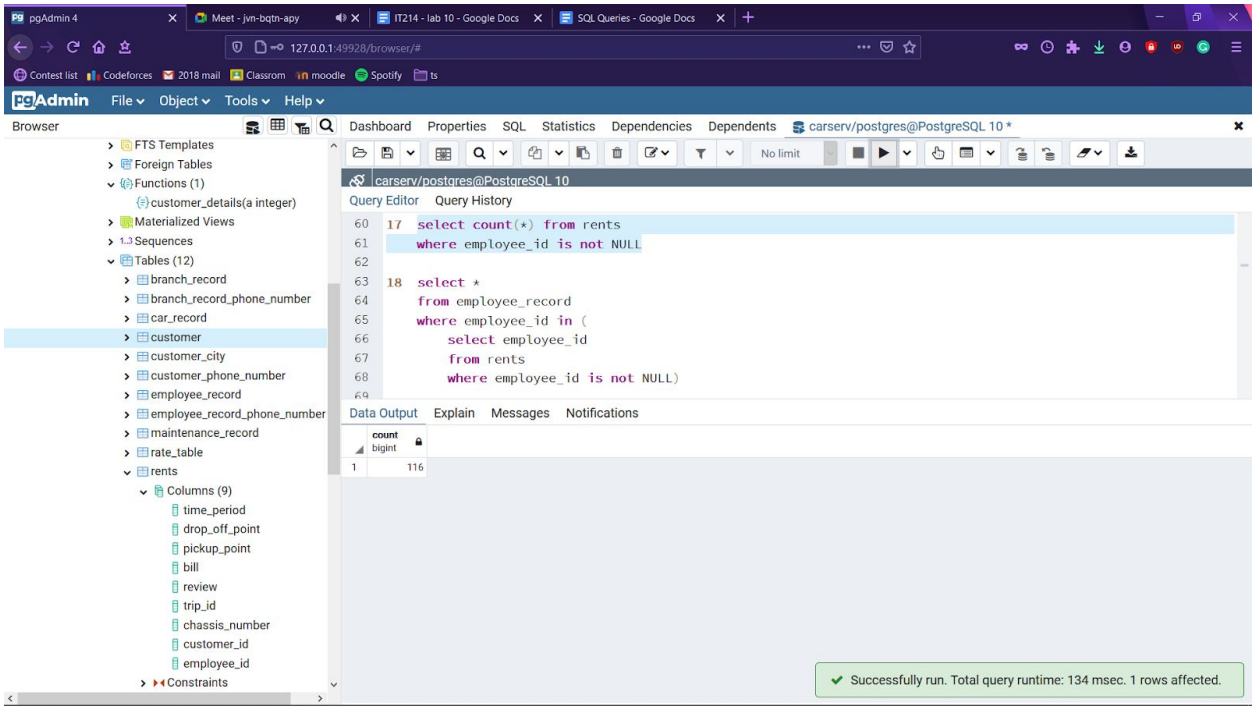
16 Find the customer details of who rented the car from 200 to 500 minutes.

```
select f_name,m_name,l_name,time_period
from customer
join rents
on customer.customer_id = rents.customer_id
where rents.time_period > 200 and rents.time_period<500
```



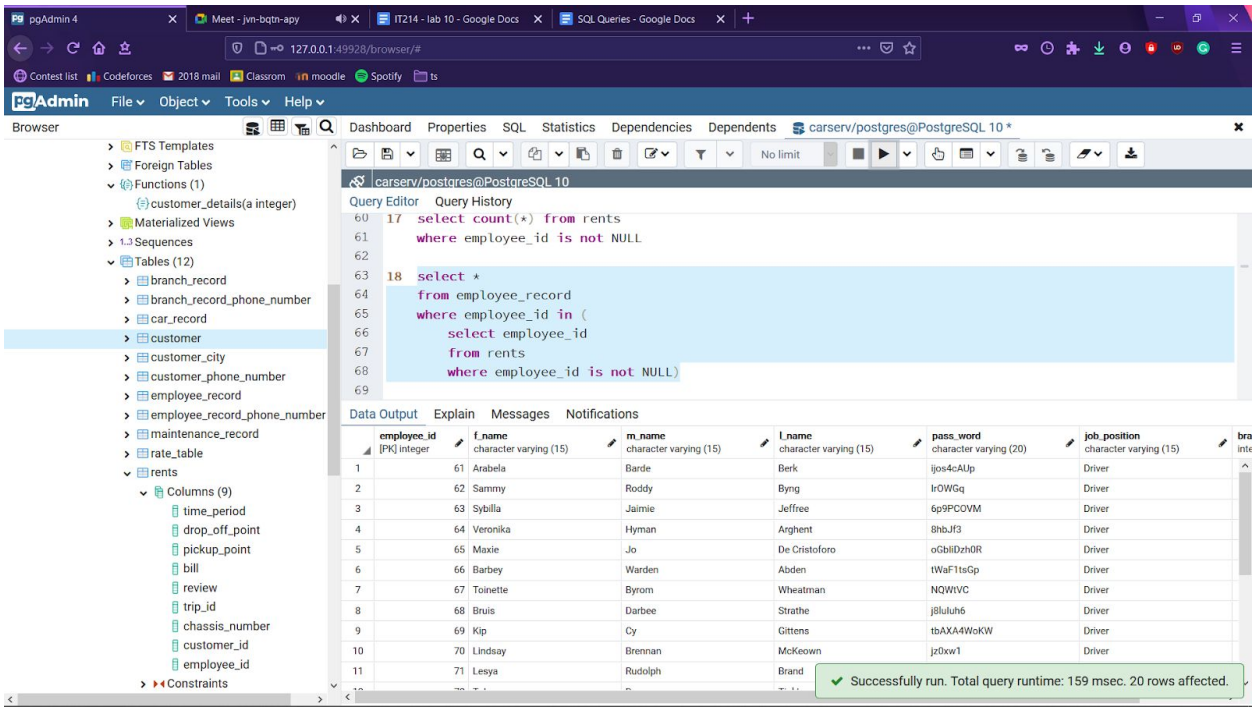
17 Find the number of customers who opted for drivers.

```
select count(*) from rents
where employee_id is not NULL
```



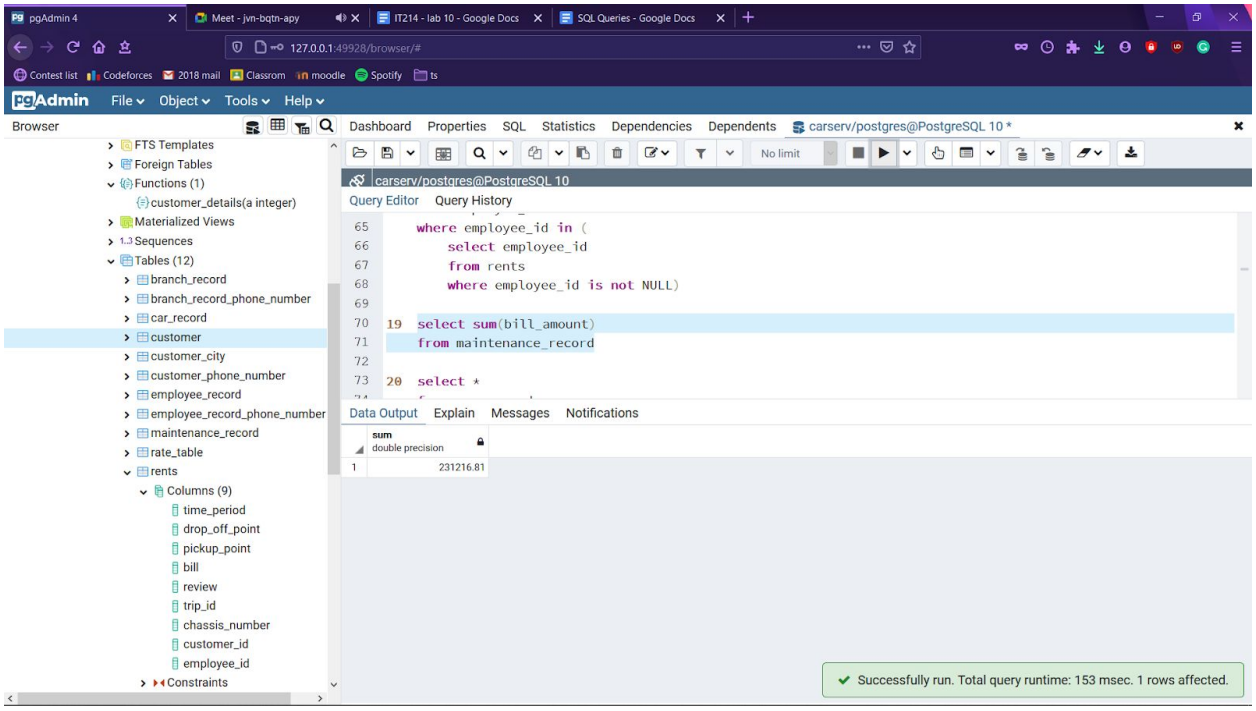
18 Find the employee details who drove the car for the customers.

```
select *
from employee_record
where employee_id in (
        select employee_id
        from rents
        where employee_id is not NULL)
```



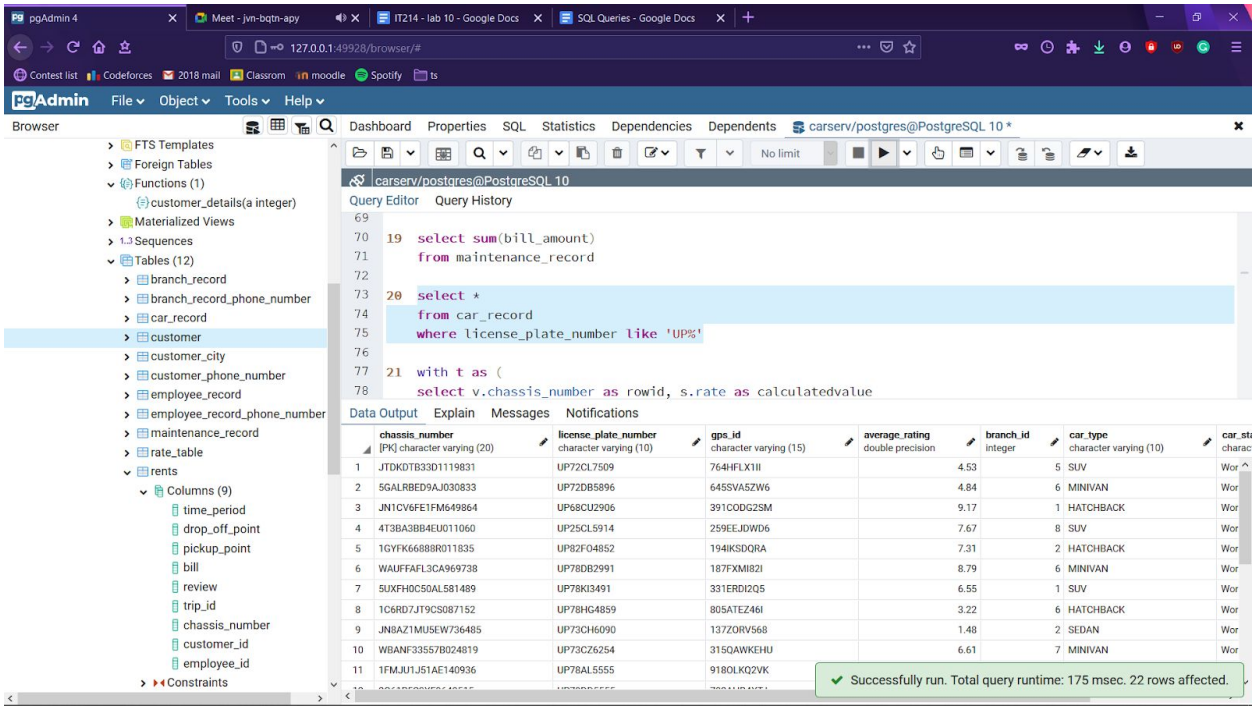
19 Find the total cost of all maintenance bills.

```
select sum(bill_amount)
from maintenance_record
```



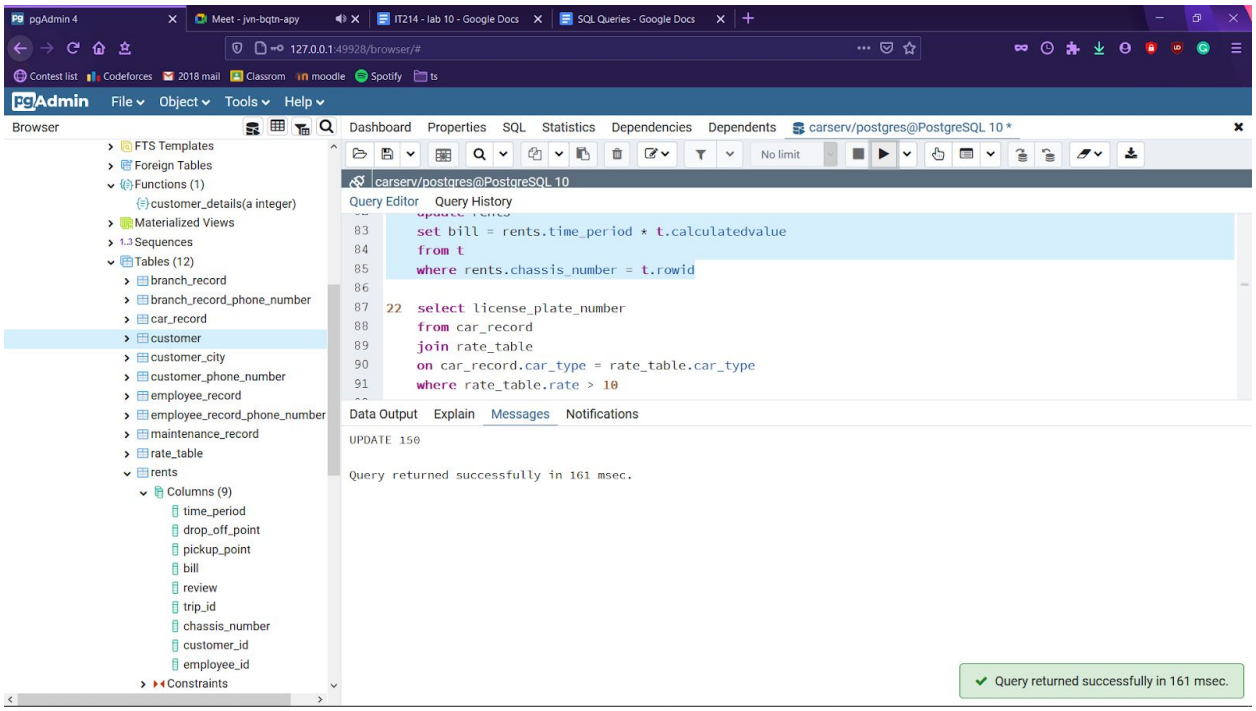
20 Find car details of cars registered in Uttar Pradesh.

```
select *
from car_record
where license_plate_number like 'UP%'
```



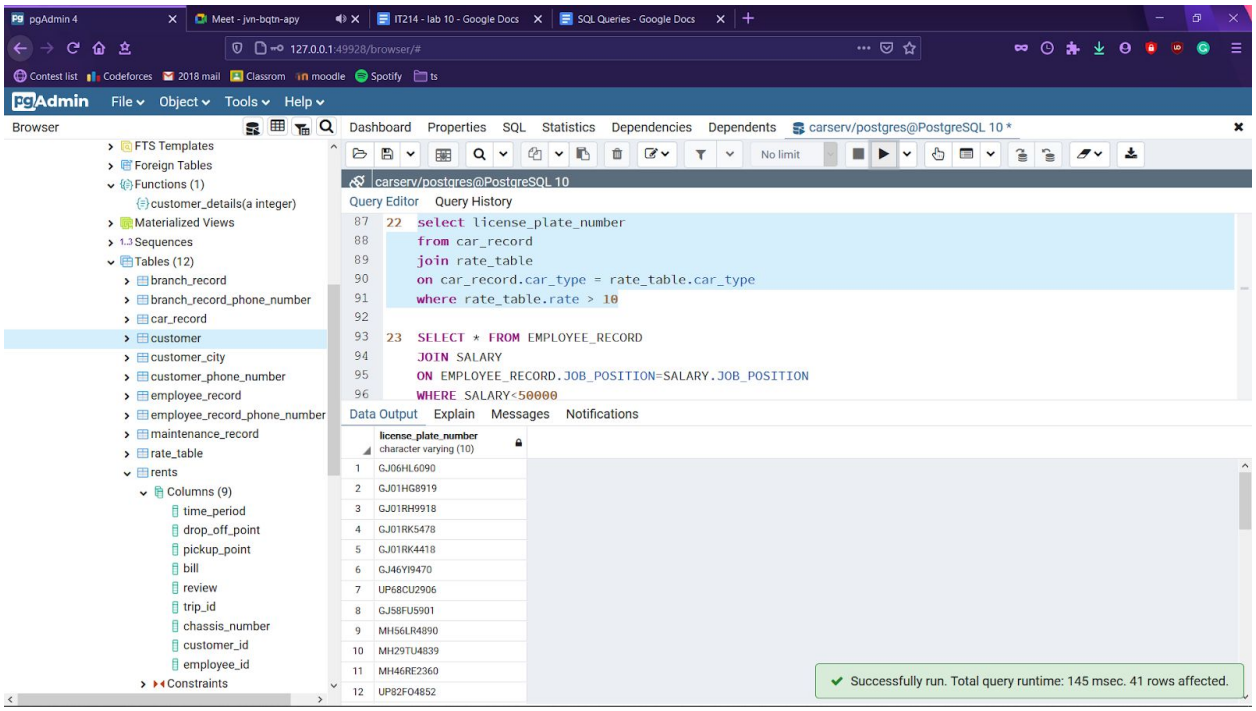
21 Update the bill for all rental records.

```
with t as (
        select v.chassis_number as rowid, s.rate as calculatedvalue
        from car_record as v
        join rate_table s on v.car_type = s.car_type
)
update rents
set bill = rents.time_period * t.calculatedvalue
from t
where rents.chassis_number = t.rowid
```



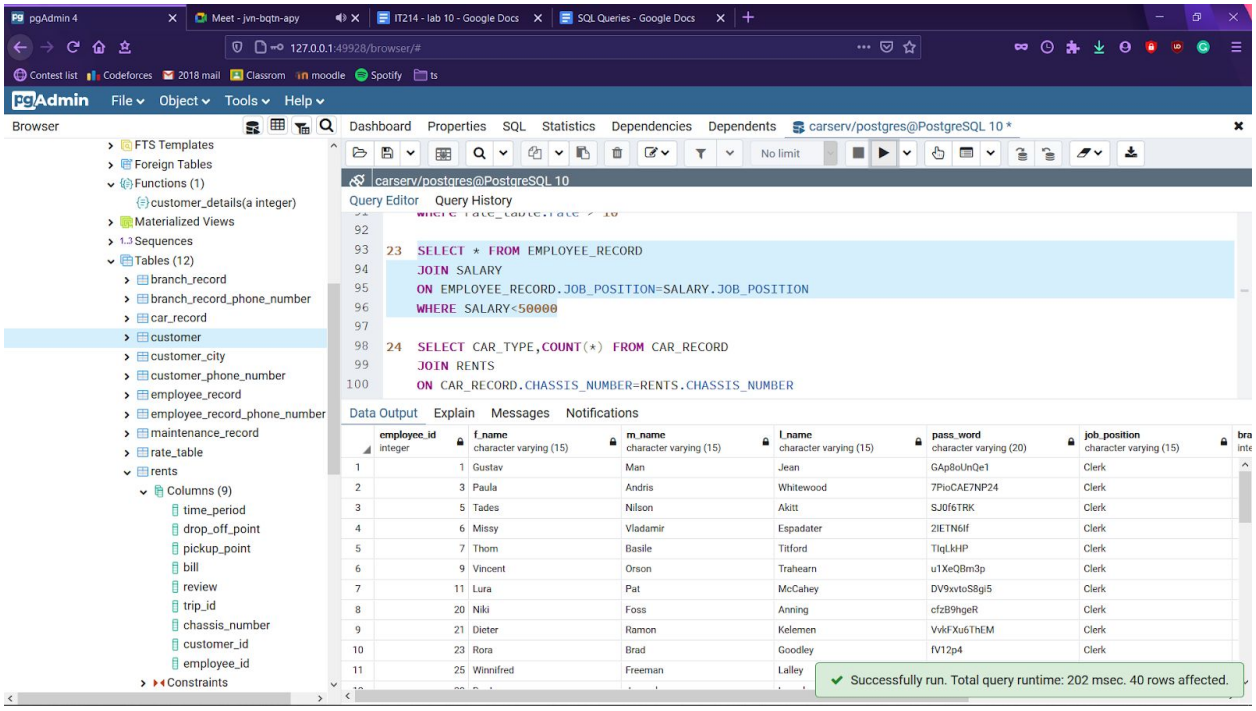
22 Display car's license plate number whose rent is greater than 10.

```
select license_plate_number
from car_record
join rate_table
on car_record.car_type = rate_table.car_type
where rate_table.rate > 10
```



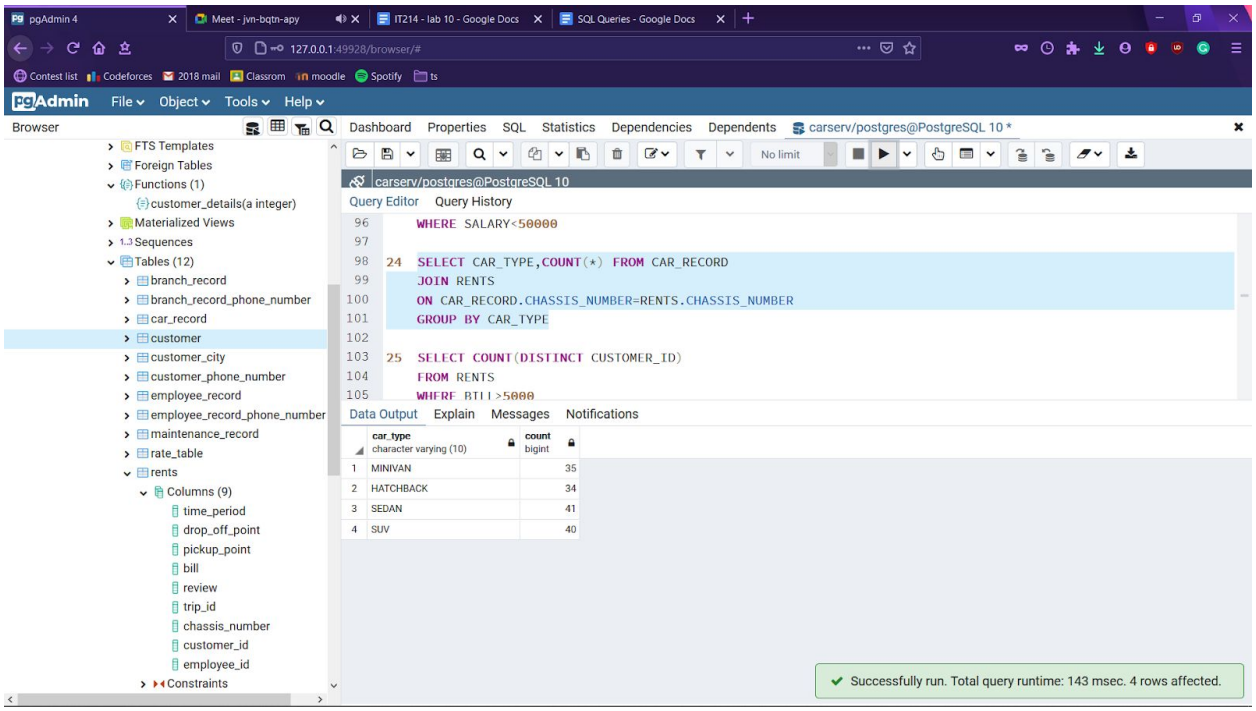
23 Find the employee details whose salary is less than 50000.

```
SELECT * FROM EMPLOYEE_RECORD
JOIN SALARY
ON EMPLOYEE_RECORD.JOB_POSITION=SALARY.JOB_POSITION
WHERE SALARY<50000
```



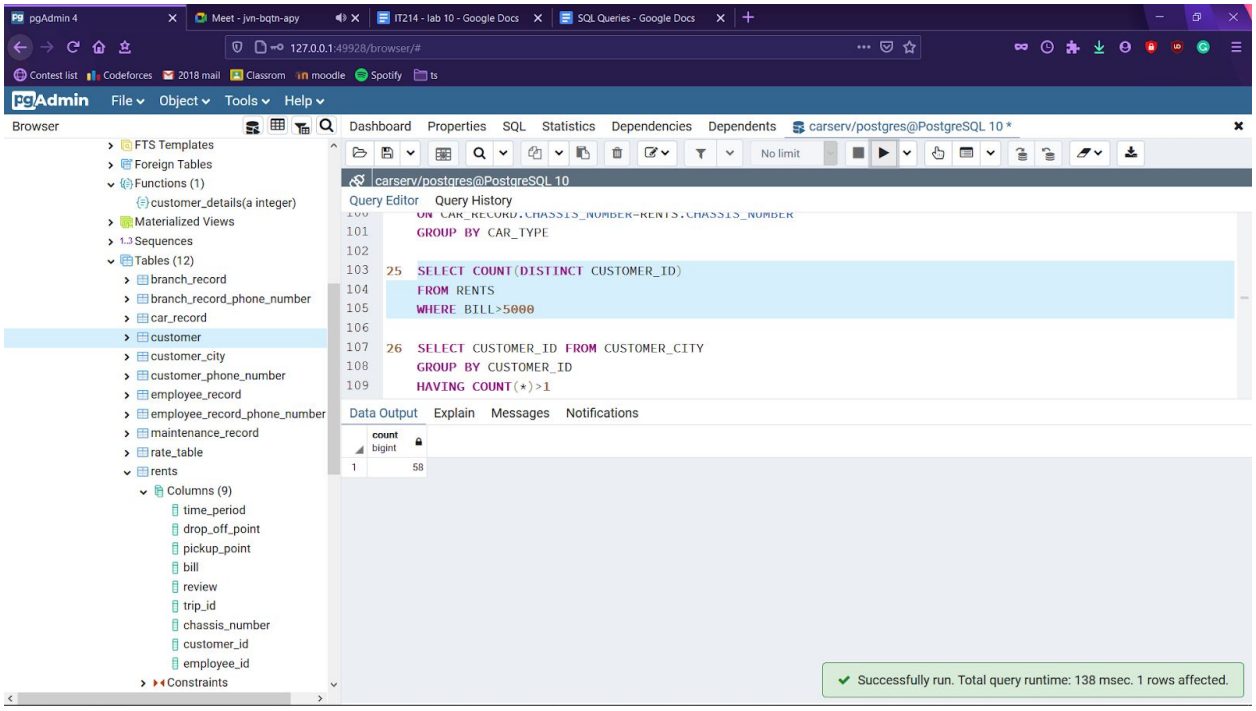
24 Select number of bookings done as per car type.

```
SELECT CAR_TYPE,COUNT(*) FROM CAR_RECORD
JOIN RENTS
ON CAR_RECORD.CHASSIS_NUMBER=RENTS.CHASSIS_NUMBER
GROUP BY CAR_TYPE
```



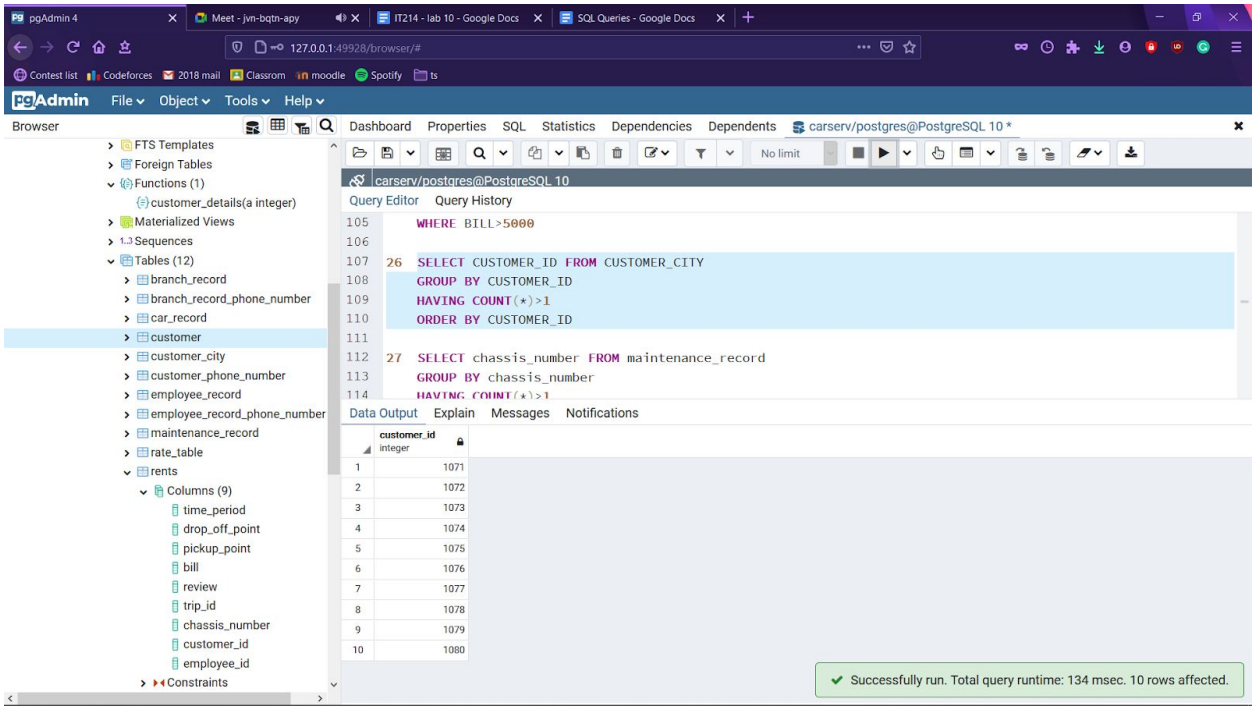
25 Select number of bookings done as per car type.

```
SELECT COUNT(DISTINCT CUSTOMER_ID)
FROM RENTS
WHERE BILL>5000
```



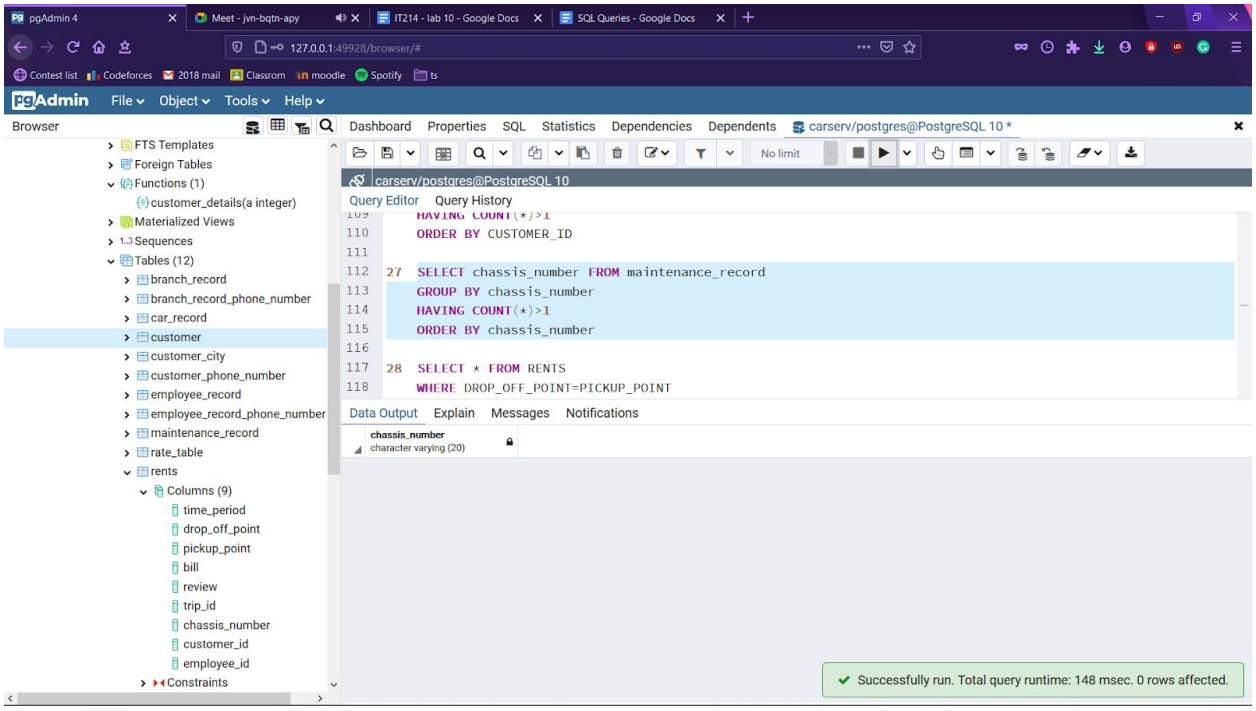
26 Count the number of customers who provided multiple cities as their residence.

```
SELECT CUSTOMER_ID FROM CUSTOMER_CITY
GROUP BY CUSTOMER_ID
HAVING COUNT(*)>1
ORDER BY CUSTOMER_ID
```



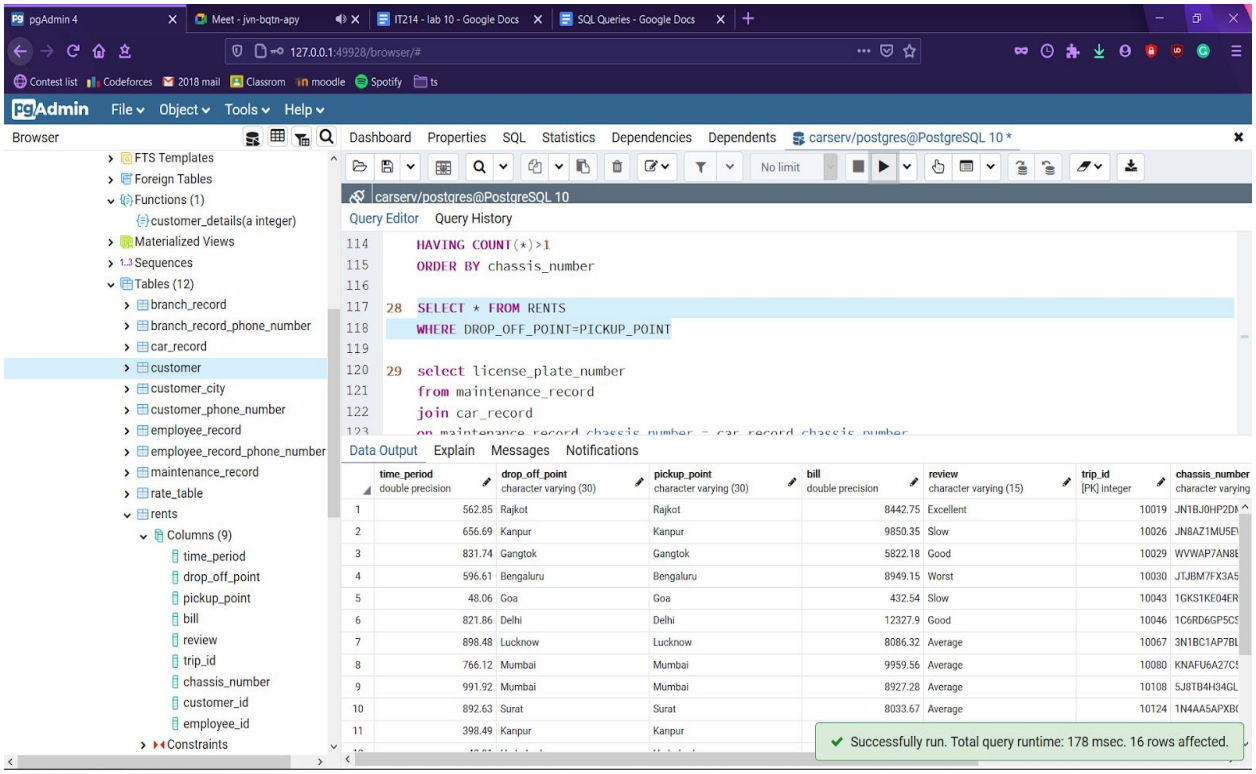
27 Display the cars which have multiple maintenance records.

```
SELECT chassis_number FROM maintenance_record
GROUP BY chassis_number
HAVING COUNT(*)>1
ORDER BY chassis_number
```



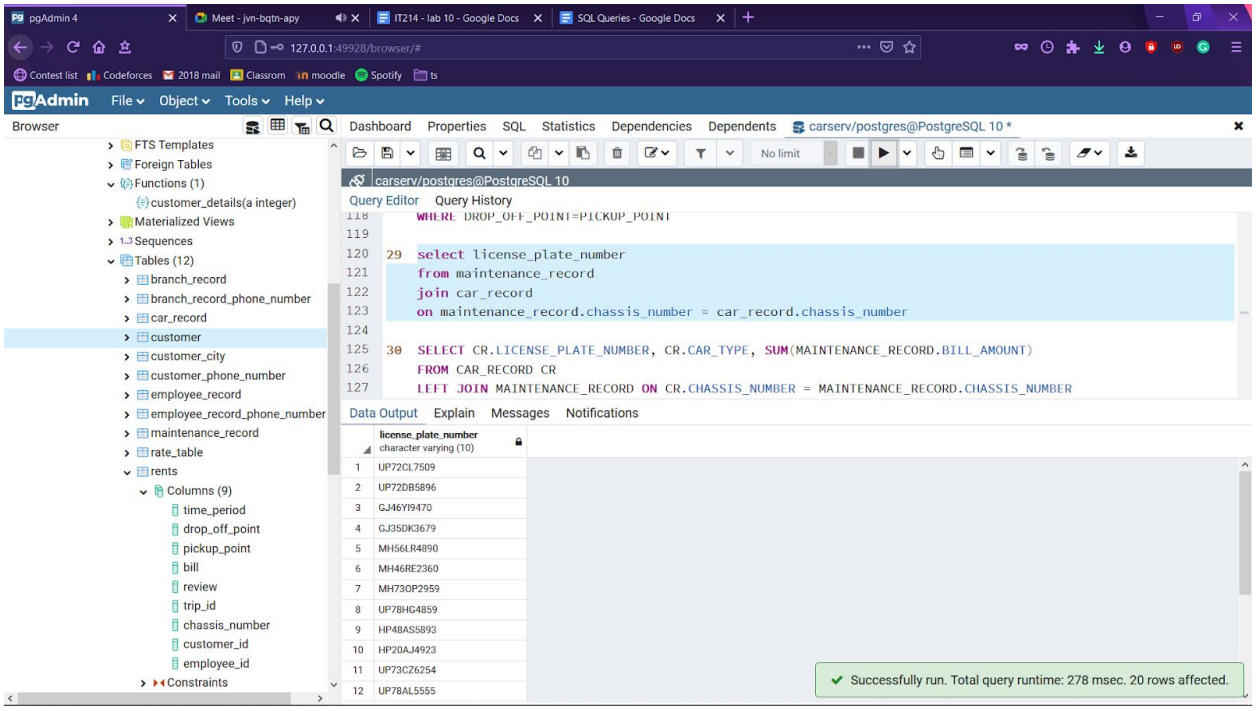
28 Display all the rental records where drop-off points and pick-up points are in the same city.

```
SELECT * FROM RENTS
WHERE DROP_OFF_POINT=PICKUP_POINT
```



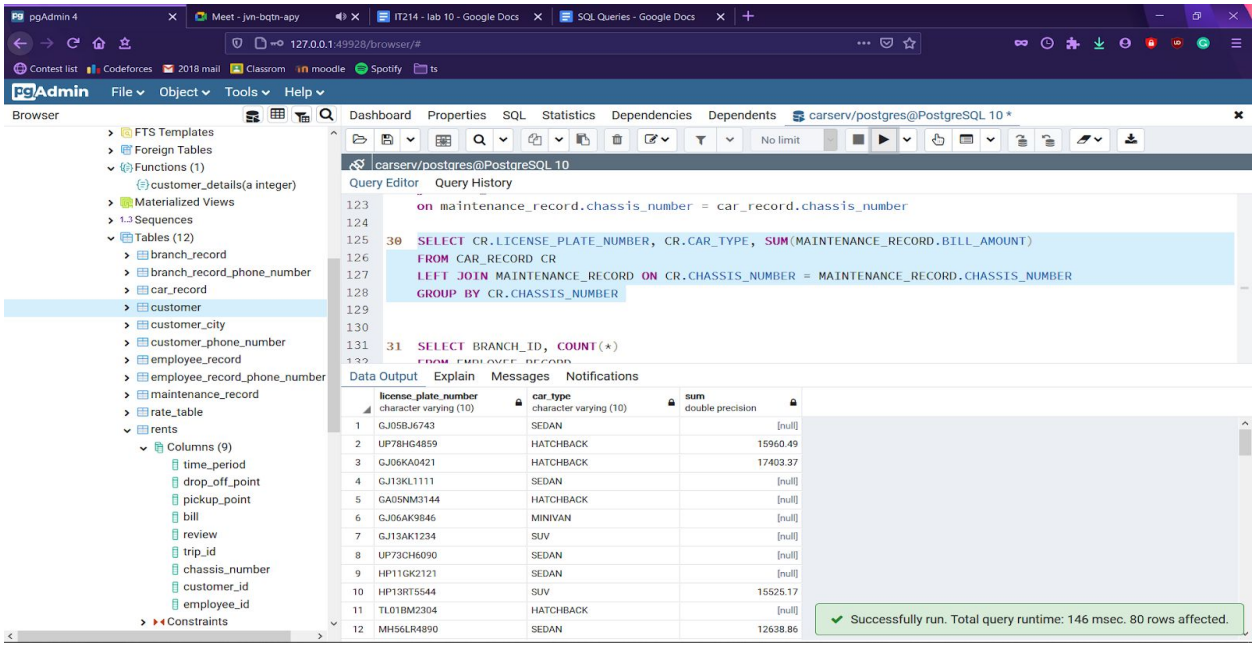
29 Find all cars who have a maintenance bill.

```
select license_plate_number
from maintenance_record
join car_record
on maintenance_record.chassis_number = car_record.chassis_number
```



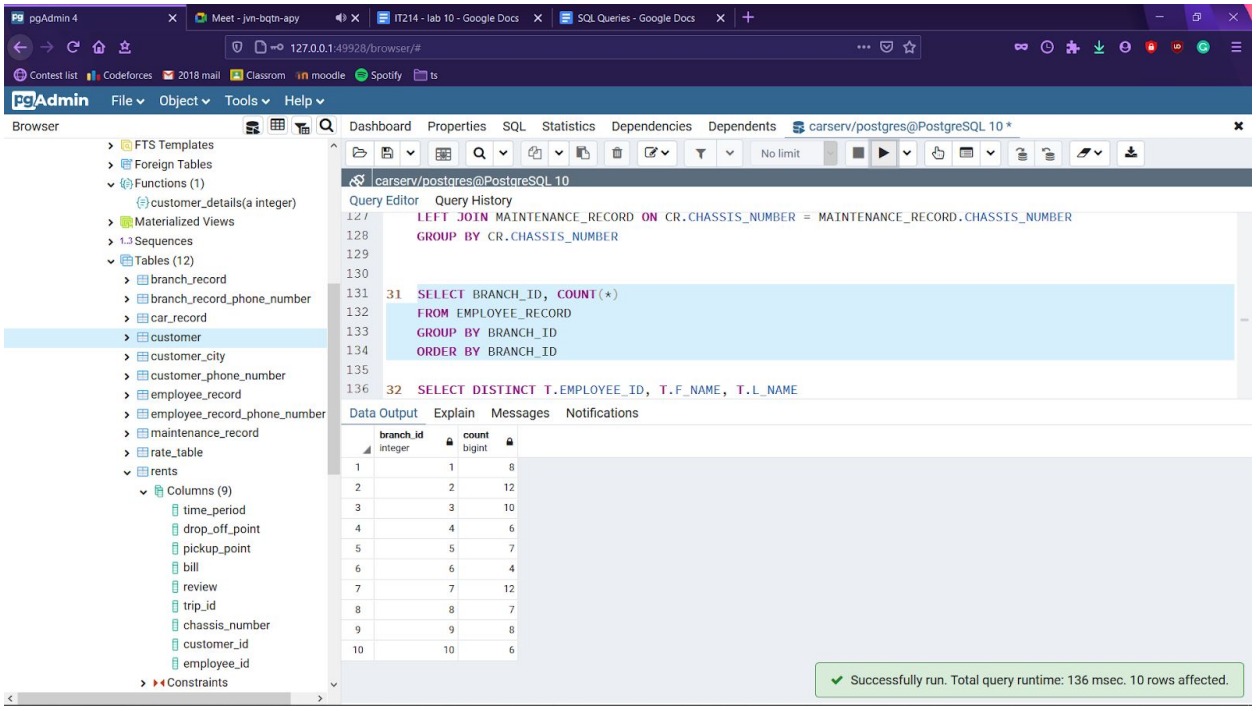
30 Find maintenance cost of each car.

```
SELECT CR.LICENSE_PLATE_NUMBER, CR.CAR_TYPE,
SUM(MAINTENANCE_RECORD.BILL_AMOUNT)
        FROM CAR_RECORD CR
        LEFT JOIN MAINTENANCE_RECORD ON CR.CHASSIS_NUMBER =
MAINTENANCE_RECORD.CHASSIS_NUMBER
        GROUP BY CR.CHASSIS_NUMBER
```



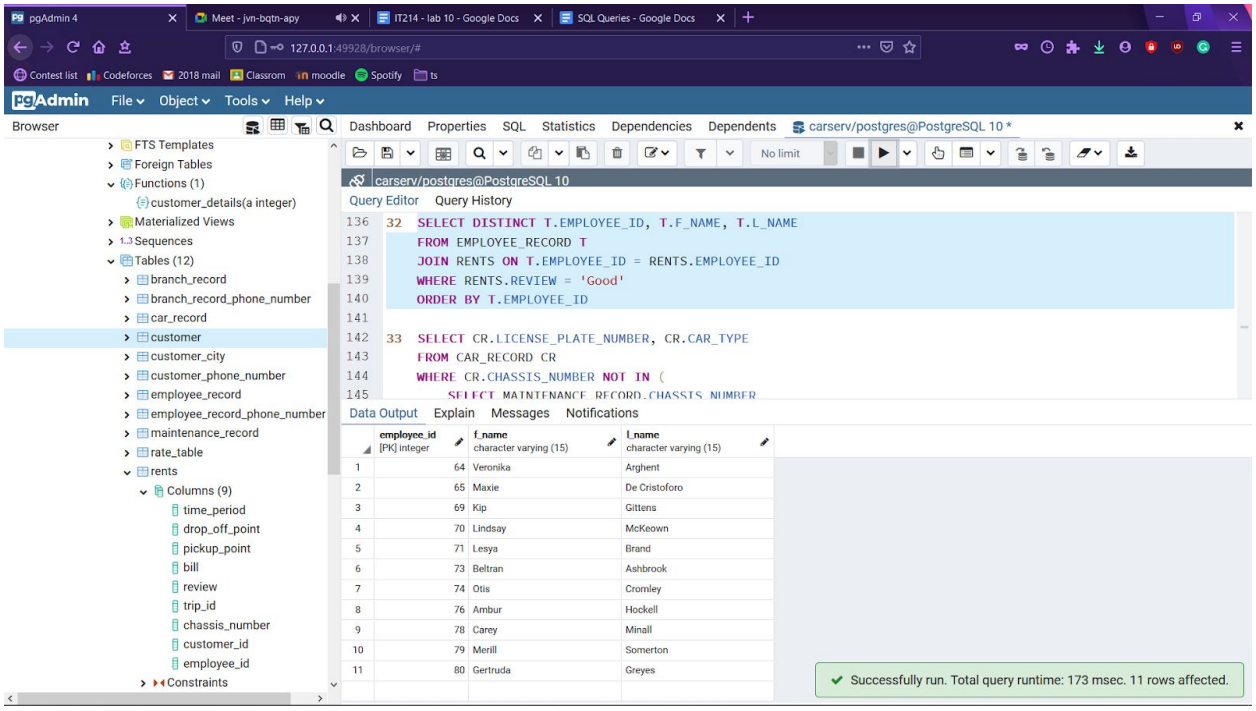
31 Find the number of employees branch wise.

```
SELECT BRANCH_ID, COUNT(*)
FROM EMPLOYEE_RECORD
GROUP BY BRANCH_ID
ORDER BY BRANCH_ID
```



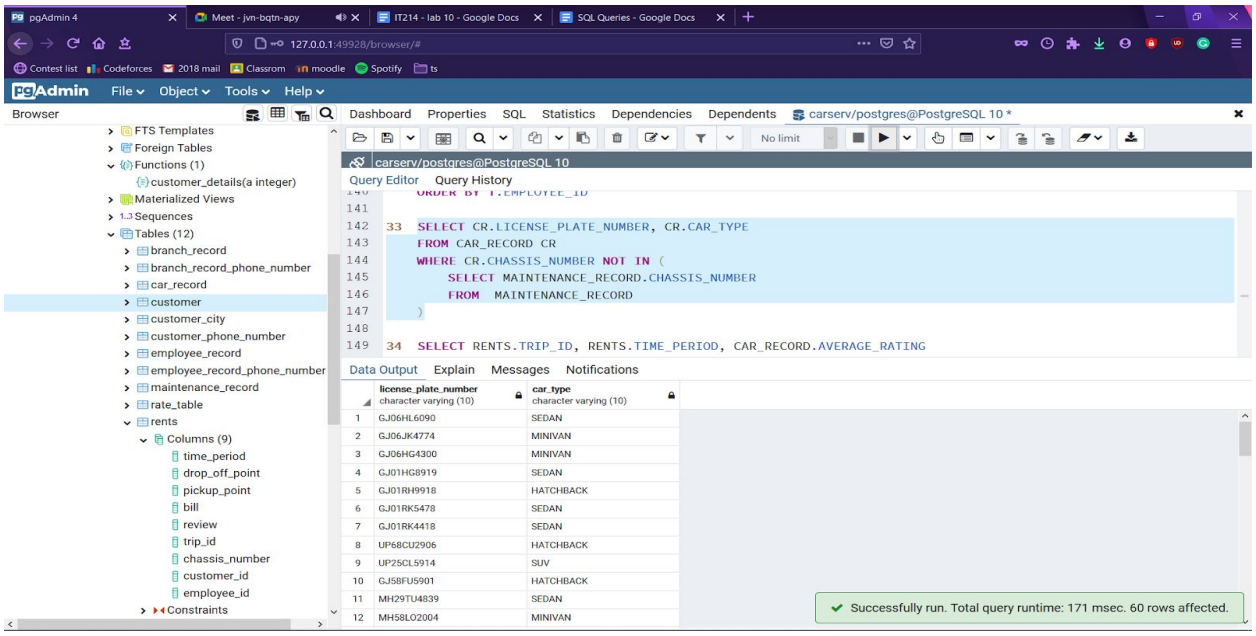
32 Find driver details who were reviewed as good.

```
SELECT DISTINCT T.EMPLOYEE_ID, T.F_NAME, T.L_NAME
FROM EMPLOYEE_RECORD T
JOIN RENTS ON T.EMPLOYEE_ID = RENTS.EMPLOYEE_ID
WHERE RENTS.REVIEW = 'Good'
ORDER BY T.EMPLOYEE_ID
```



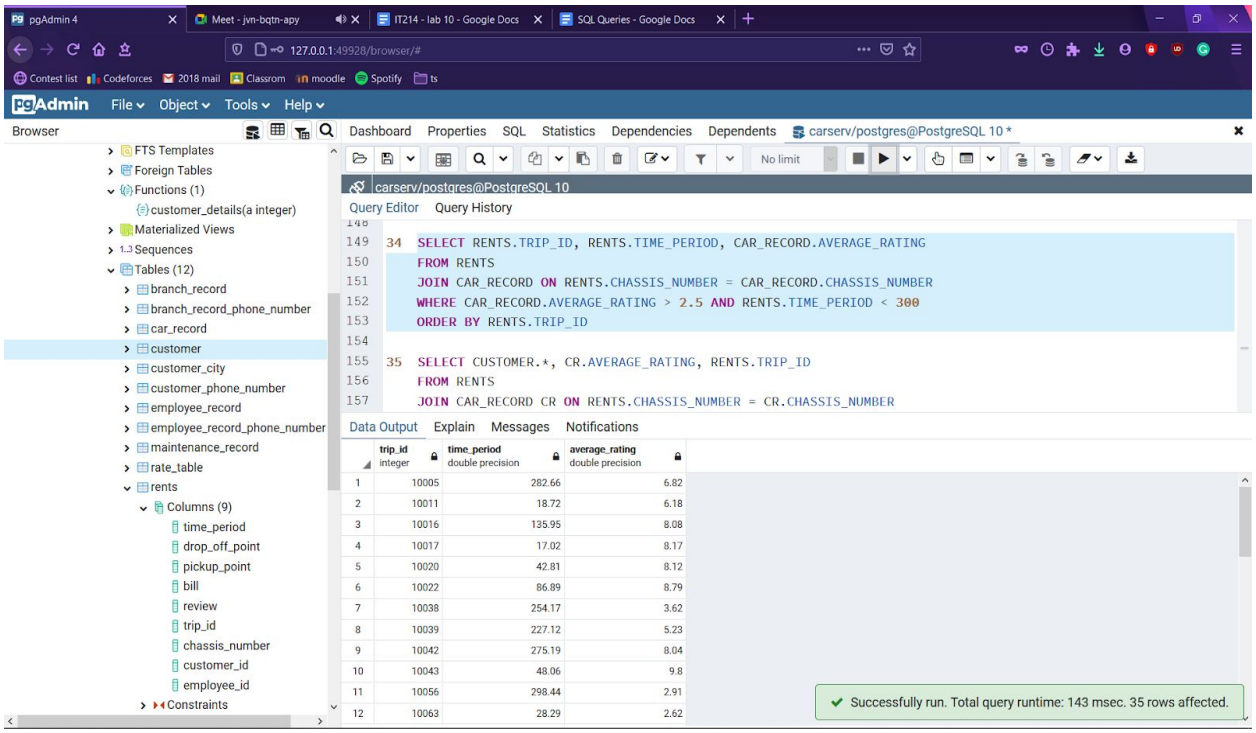
33 Display all cars who don't have a maintenance bill.

```
SELECT CR.LICENSE_PLATE_NUMBER, CR.CAR_TYPE
FROM CAR_RECORD CR
WHERE CR.CHASSIS_NUMBER NOT IN (
 SELECT MAINTENANCE_RECORD.CHASSIS_NUMBER
FROM  MAINTENANCE_RECORD
)
```



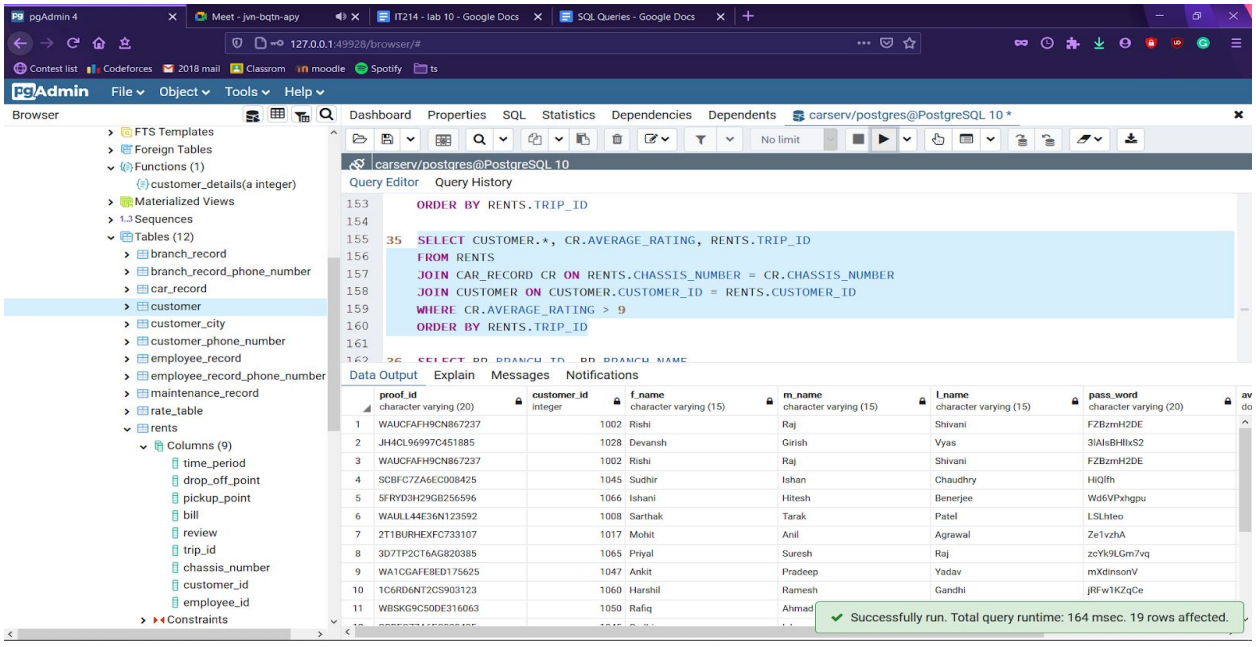
34 Display rent id who rented the car for less than 300 minutes and rating is higher than 2.5.

```
SELECT RENTS.TRIP_ID, RENTS.TIME_PERIOD,
CAR_RECORD.AVERAGE_RATING
FROM RENTS
JOIN CAR_RECORD ON RENTS.CHASSIS_NUMBER =
CAR_RECORD.CHASSIS_NUMBER
WHERE CAR_RECORD.AVERAGE_RATING > 2.5 AND RENTS.TIME_PERIOD < 300
ORDER BY RENTS.TRIP_ID
```



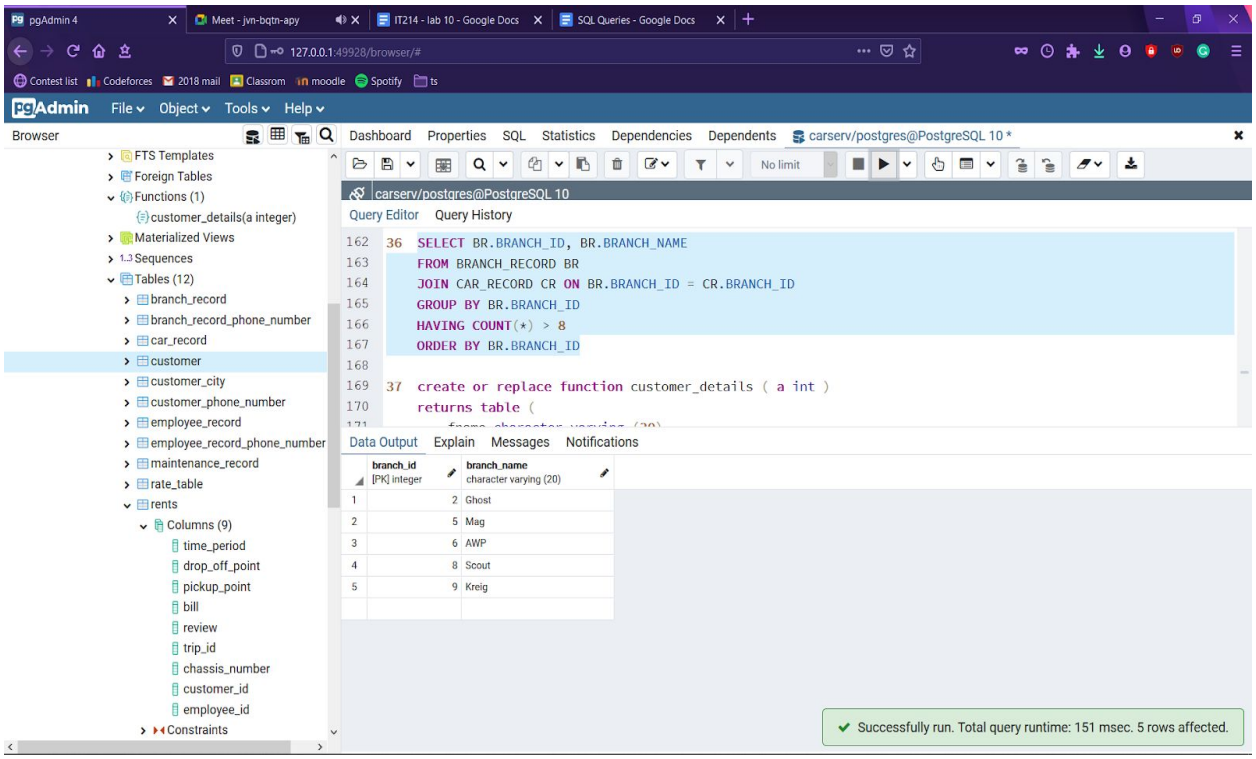
35 Display all the customer details who traveled in a car having average ratings higher than 9.

```
SELECT CUSTOMER.*, CR.AVERAGE_RATING, RENTS.TRIP_ID
FROM RENTS
JOIN CAR_RECORD CR ON RENTS.CHASSIS_NUMBER = CR.CHASSIS_NUMBER
JOIN CUSTOMER ON CUSTOMER.CUSTOMER_ID = RENTS.CUSTOMER_ID
WHERE CR.AVERAGE_RATING > 9
ORDER BY RENTS.TRIP_ID
```



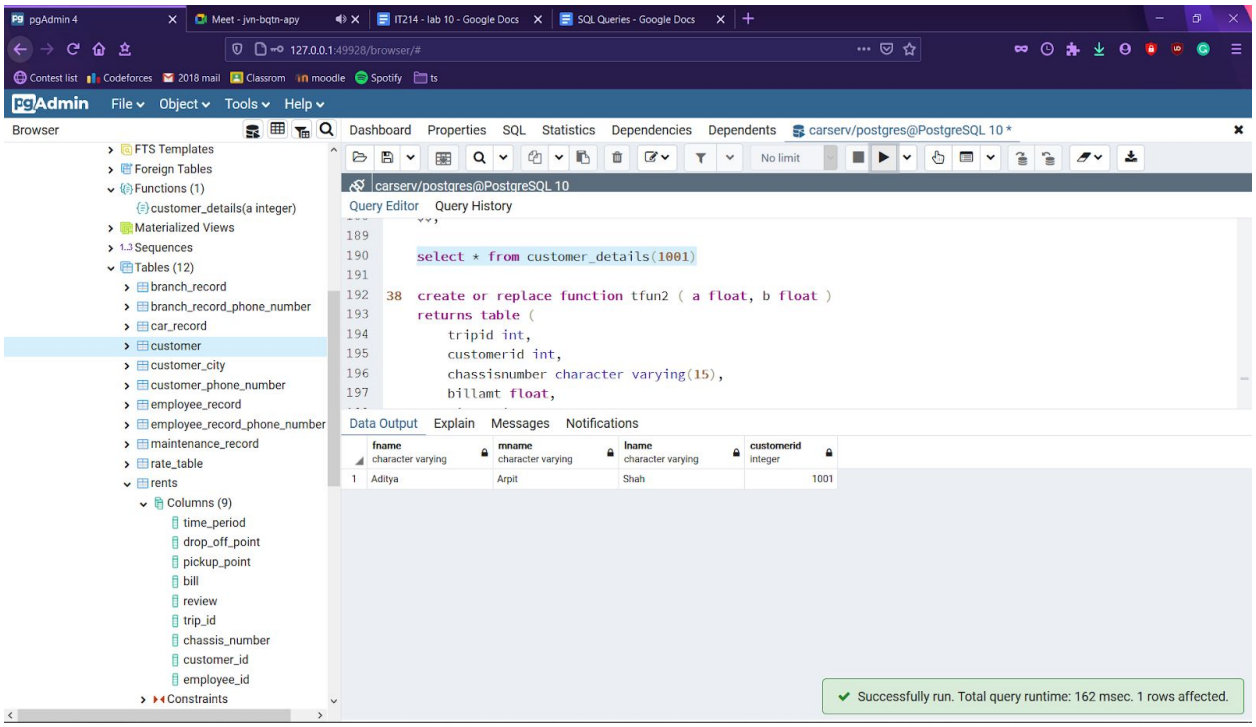
36 Display branch name which has more than 5 vehicles.

```
SELECT BR.BRANCH_ID, BR.BRANCH_NAME
FROM BRANCH_RECORD BR
JOIN CAR_RECORD CR ON BR.BRANCH_ID = CR.BRANCH_ID
GROUP BY BR.BRANCH_ID
HAVING COUNT(*) > 8
ORDER BY BR.BRANCH_ID
```



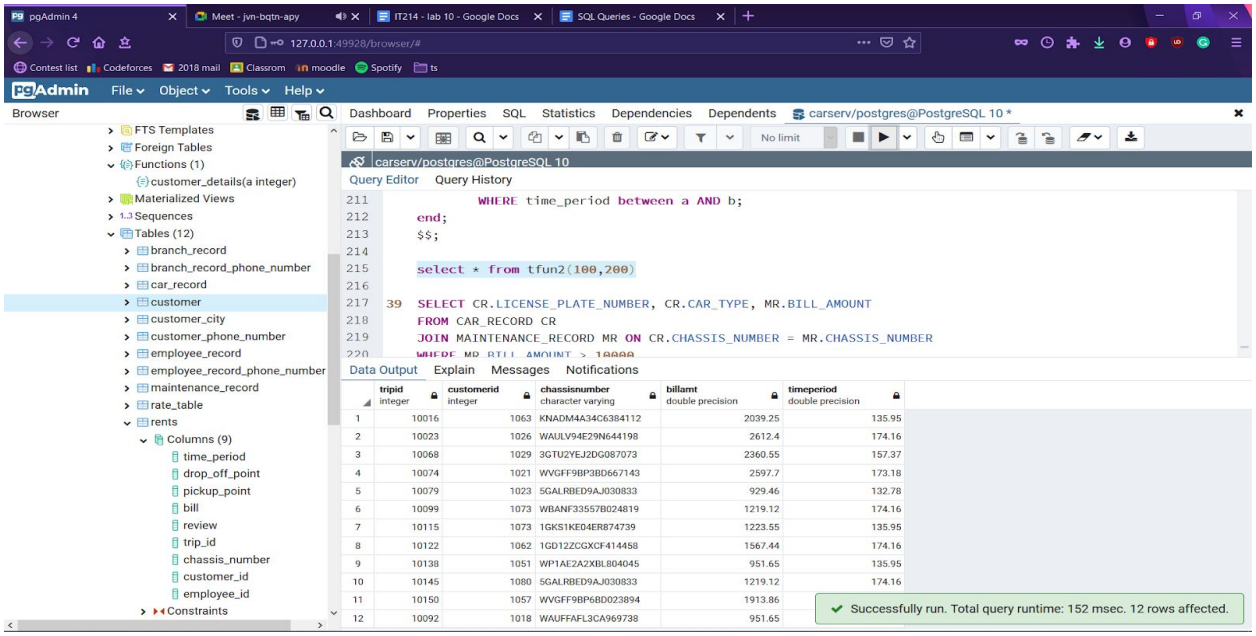
37 Create function which takes customer_id as input and returns customer details.

```
create or replace function customer_details ( a int )
   returns table (
         fname character varying (30),
         mname character varying (30),
         lname character varying (30),
         customerid int
   )
   language plpgsql
   as $$
   begin
         return query
               select
               f_name,
               m_name,
               l_name,
               customer_id
               from customer
               WHERE customer_id = a;
   end;
   $$;

   select * from customer_details(1001)
```



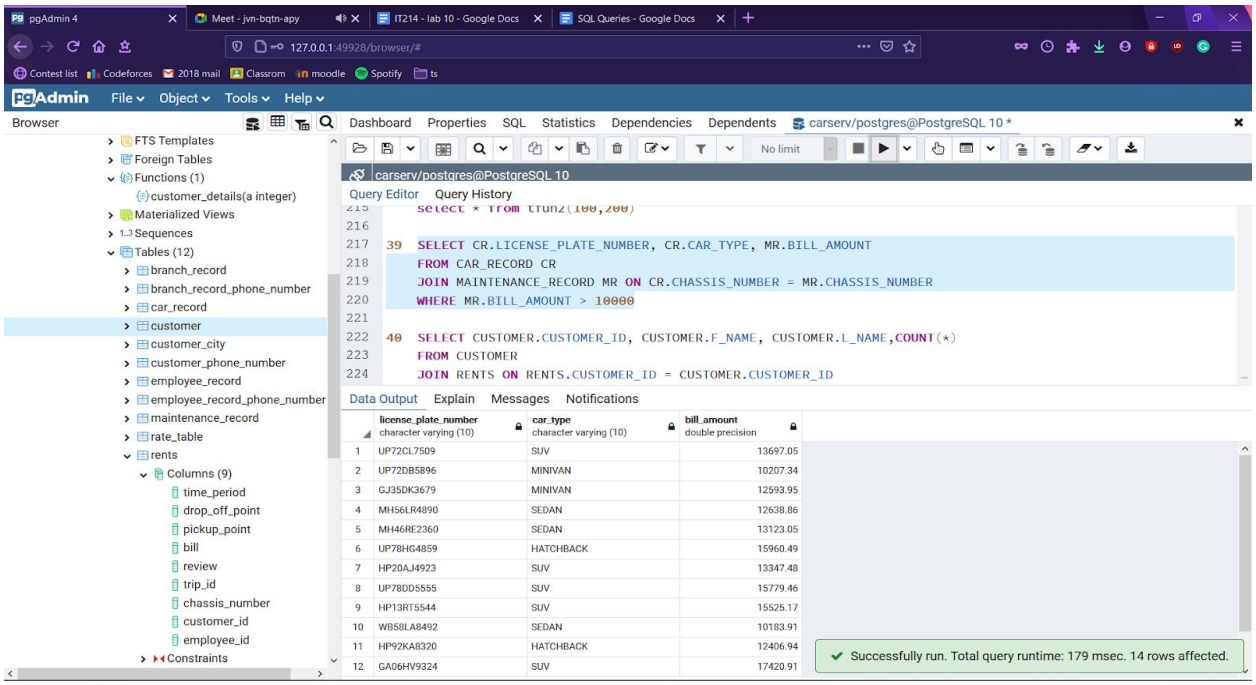
38 Create function which takes two time periods as input and displays all rental records where the car was rented for more than the first time period and less than the second time period.

```
create or replace function tfun2 ( a float, b float )
  returns table (
        tripid int,
        customerid int,
        chassisnumber character varying(15),
        billamt float,
        timeperiod float
  )
  language plpgsql
  as $$
  begin
        return query
              select
              trip_id,
              customer_id,
              chassis_number,
              bill,
              time_period
              from rents
              WHERE time_period between a AND b;
  end;
  $$;

  select * from tfun2(100,200)
```


pgAdmin 4
Meet - jvn-bqtn-apy
IT214 - lab 10 - Google Docs
SQL Queries - Google Docs
127.0.0.1:49928/browser/#
Contest list
Codeforces
2018 mail
Classrom
moodle
Spotify
ts
pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
carserv/postgres@PostgreSQL 10 *
No limit
FTS Templates
Foreign Tables
Functions (1)
customer_details(a integer)
Materialized Views
Sequences
Tables (12)
branch_record
branch_record_phone_number
car_record
customer
customer_city
customer_phone_number
employee_record
employee_record_phone_number
maintenance_record
rate_table
rents
Columns (9)
time_period
drop_off_point
pickup_point
bill
review
trip_id
chassis_number
customer_id
employee_id
Constraints
carserv/postgres@PostgreSQL 10
Query Editor
Query History
211 WHERE time_period between a AND b;
212 end;
213 $$;
214
215 select * from tfun2(100,200)
216
217 SELECT CR.LICENSE_PLATE_NUMBER, CR.CAR_TYPE, MR.BILL_AMOUNT
218 FROM CAR_RECORD CR
219 JOIN MAINTENANCE_RECORD MR ON CR.CHASSIS_NUMBER = MR.CHASSIS_NUMBER
220 WHERE MR.BILL_AMOUNT > 10000
Data Output
Explain
Messages
Notifications
tripid integer
customerid integer
chassisnumber character varying
billamt double precision
timeperiod double precision
1 10016 1063 KNADM4A34C6384112 2039.25 135.95
2 10023 1026 WAULV94E29N644198 2612.4 174.16
3 10068 1029 3GTU2YEJ2DG087073 2360.55 157.37
4 10074 1021 WVGFF9BP3BD667143 2597.7 173.18
5 10079 1023 5GALRBED9AJ030833 929.46 132.78
6 10099 1073 WBANF33557B024819 1219.12 174.16
7 10115 1073 1GKS1KE04ER874739 1223.55 135.95
8 10122 1062 1GD12ZCGXCF414458 1567.44 174.16
9 10138 1051 WP1AE2A2XBL804045 951.65 135.95
10 10145 1080 5GALRBED9AJ030833 1219.12 174.16
11 10150 1057 WVGFF9BP6BD023894 1913.86
12 10092 1018 WAUFFAFL3CA969738 951.65
Successfully run. Total query runtime: 152 msec. 12 rows affected.

39. Display car records whose maintenance bill is greater than 10000.

```
SELECT CR.LICENSE_PLATE_NUMBER, CR.CAR_TYPE, MR.BILL_AMOUNT
FROM CAR_RECORD CR
JOIN MAINTENANCE_RECORD MR ON CR.CHASSIS_NUMBER =
MR.CHASSIS_NUMBER
WHERE MR.BILL_AMOUNT > 10000
```



40 Display the customer id and number of cars rented by that customer.

```
SELECT CUSTOMER.CUSTOMER_ID, CUSTOMER.F_NAME,
CUSTOMER.L_NAME,COUNT(*)
FROM CUSTOMER
JOIN RENTS ON RENTS.CUSTOMER_ID = CUSTOMER.CUSTOMER_ID
GROUP BY CUSTOMER.CUSTOMER_ID
ORDER BY CUSTOMER.CUSTOMER_ID
```