Service request for civil works system database

DBMS - IT214

Team S10_T10

Instructor: Prof. Minal Bhise

Mentor TA: Mayank Jr, Dhairya

Team Members:

Vishvarajsinh Chauhan:201901015

Harsh Patel:201901016

Baveet Singh:201901256

Bhavesh Pandor:201901147

Dhirubhai Ambani

Institute of Information and Communication Technology

Table of Contents

Section 1
Final version of SRS

Understanding the Problem Domain of

Purpose:

Mainly, we are using database to manage inconsistency, difficult data accessing, security, reliable service, concurrent access and many such problems. Database manager is used to alter, add, update or delete the record. Here, we are making database based on Service request for civil works (like - plumbing, electrical problem, carpeting etc.) Mostly customers are not easily finding the services for their problems. Here our main purpose is to make database system is used to give customers to their needed services easily and in reliably way. This database help customers to get services for their problems in its requirements and also, they get services by good workers.

customer also can rate and give feedback for the services which they get and by the feedback, we can improve the database based on customers feedback and their requirements.

Intended Audience and Reading Suggestions:

The types of readers, the document is intended for, would be the ones with problems in plumbing, electric supply/low voltage, etc.

The SRS contains all the services, organized in the form of a database/table where we can have all the details of a service whether it is pending/finished, booking date, payment details, service details, worker and customer details, contact numbers, service id, also a feedback from the customers side, where customer can rate a service on the scale of 10, also an emergency column where if the customer is in hurry state, assign a priority service to the customer, in exchange of increased payment, also there is column “suggestion”, if a customer is unable to find a service he wants, the customer can put that in the suggestion column. If huge demand came to a new one, it can be added to the service section and begin servicing.

The reading sequence is depended to the type of service customer wants, so he can jump to service section that he wants, where it can choose a service and book a date for it or customer can go on with services section and then pick whichever he wants.

The rest of the SRS contains the basic functionalities of the system. Here, we are starting with the product scope and description for the company goal and brief introduction of the system. After that background readings, interview’s, questionnaires and observations helps users to familiars with the system. At the end we are providing information about the list of requirements, operating environment, product functions, privileges, assumptions and business constraints.

Product Scope:

Here, we have going to create service request for civil works system database. This database is very useful in application like mobile or OS application**.** In old days if we wish to call some electrician for electric work we have to go in market and search for the electric shop, in that there is lots of time wasted**.** Now, in today’s world everything is superfast everyone is hoping that all the work can be done through the home via mobile or laptop.

For that we have to create one application that provides every service-related household and payment systems. We can think as corporative market also if we build such a software company that provides services like plumbing, electric problem in the house, etc. in one application and the costs are also paid through that application hence the profit of the company goes high because we know that in India 34.4% of the population is using the internet and everybody faces some issues with many services in the house. Our goal is to serve who needs the services, workers have some good amount of money and also as a company to make profit.

Description:

We have created service request for civil works system database. In that database we create tables. In the first table we describe what kind of services we are providing to the customers. The second table contains recommendation, if customer cannot find the service, then he or she can recommend us to add the service in future. The third table contains the type of the services like customer wants to buy normal or fast service and it also displayed the cost for both the services. The fourth table contains customer details like phone number, address etc. The fifth table contains the payment methods like cash on delivery, online payment, etc. The sixth table contains the workers list for the services which is only for our preferences that which worker is on the duty or free. The seventh table is for service status the given service to particular worker is done or not. The eighth table contains the feedback of the customer. The ninth table contains the login information like customer_id, password, phone number.

Service request for civil works system database is the database that can be used in application perspective or website perspective also. In today's world the real-life example of service request for civil works system is 'Urban Company' in India. 'Urban Company' has starting with some barbers at first time. After that customer really wants the more services by urban company. Now onwards 'Urban Company' provides more than 15 services to the customers like repair & maintenance, home cleaning, homecare & design (Decorator), pest control, packers & movers, business services, event management, weddings & party management, health & wellness, salon, painter, Carpenter, Yoga instructor, massage, spa, makeup artists etc. Our database is fully concentrated on the customer requirements.

Our database is all about for providing services to the customers therefore the major part of the database is the service information. It contains the service_name and service_id. Now, the main concept is the customer requirements for that we have make one recommendation section for those customers who wants us to add some new services. Services are mainly two types one is normal service and fast service. In the fast service we mainly focus on the time. If customer selects the fast service than the priority of that customer is high than the normal service but for that customer needs to pay some extra amount. We keep data for the payments through via online payment or cash after service for the worker's payment. Now for the services we have to go to the customer's home for that we have to collect some customer information like customer name, address, phone number, etc. We have to assign the different customer_id to each customer so we can differentiate two customers with the same name.

Service status is another main concept of our database that we have to know that which service is complete or pending status that help us to schedule next services to the workers. We provide one more benefit to the customer that he/she can give feedback to us via scaling 0 to 10. From that it helps us for motivating to keep progressive and if some fault happens from our side than we can improve ourselves to provide better services. For our side the major concern is to schedule workers for the services. For that we have to maintain one table that contains information of the service_id which was same as the service information and simultaneously the worker's name, id, service status and queued service for that worker. It helps us to assigning new service to the new customer on which day or which time. For the payment to the workers, it all depends on the number of services in a day. It was clearly mentioned in the workers information that how much he/she able to earn in one day. This worker information table is private to the database owner and it cannot be show to the customer. This database helps us to make easy track of the services and make life easier.

Our database deals with the huge number of data because it tracks and stores the customer information, service information and feedback/ Rating points. It helps customer, service providers and administration as well. Customer can see the previous history of the services, payment details, and the discount of that time. Service provider can count the number of the services, it helps them to track record of their services like time, date, service name, payment done by the customer etc. Administration can make assumption on the basis of the previous services that some service is more popular than the other one, it can help to employee a greater number of workers to that particular service.

In the Application, the interface or the UI between user and application is a crucial part. The application must run properly with less bugs and efficient payment gateways system. If the user count of the App is very high, then through Endorsements and Ads, company's revenue is increased. Also, there is feature called "Super Service" in the application, in which if a particular service has less booking count, we can shift the service to this section. It contains all the service which are currently discounted. The discount attracts customers to the app, which increases the user visiting count as well as the chances of booking of any type service in that section. Even if there is a little loss there than usual, but it is better to have something than no service at all. Worker is then asked if that discounted price is enough for him to provide the service if he agrees then the service happens if not then another worker is asked. If that service is refused by all available workers, then the "service not available" option shows up.

Regular updating the application is another aspect, like festival sales attractive opening application page, cool picture and color pages, to attract youth people, etc. Anything we can increase revenue, we should know the actual and main requirements of our customers, some are,

Giving them trusted and high qualified professionals who are best in their services.

- Providing trusted, affordable, best price to quality services to customers.
- Update data model and app system by customer's rating and feedback for services.
- We should keep in mind with the delay of the services. Giving delayed service is a scratch in company's reputation as seen by customers.
- Make special ID for all the customer who taking service many times
- For maintaining diligently work we should give them small festival bonus and treat. also, the company profit and employee's salary increment in an organized way so that employee is satisfied with this salary also it doesn't affect companies profit.

Surveys are yet another important feature, we should regulate surveys of less questions per 1-2 months, it helps the company to know, what people want, and in understanding people's concerns. Most demanded service should have maximum workers to meet the demand also less demanded services should have less workers and can be moved to the "Super Service" section if necessary.

To interact more and more with people, there is special function called "Live chat". In this, there is a pop-up chat icon at the bottom of the screen in the application window. If a user has any issues regarding the service, then there are help providers there, which would help them in the live chat session, customers can ask any query like timings, prices, or anything which he want to ask within the business scope and also call them if a customer is more comfortable in talking over Voice calls.

In present time it is difficult to find the serviceman in local areas. We do not get service provider easily; we need to have a personal contact for this and it is the total waste of time because searching a services provider by contact is very time consuming. Many companies which are providing local services does not have a user-friendly app because they can't able to store data infrastructure and proper manner where in our database, we are stored data efficiently.

Our database must handle the large numbers of the data of the customers and services providers or workers. The main concern of the database is the real-life time accurate assigning the worker to the customer. If customer wants the fast services but at that time, we don't have any worker related that service then it was disaster therefore we have taken care of this type of situation. The two solutions are possible, one is we have to take one particular worker for the fast service and second is that in the queued time we have to assign more priority to the fast services but in this there is more delay in the

normal services that is quite bad impression on our system. The market is also the part of our system because the services is all about the supply and demand thing as example in the Covid -19 time the Demand is too low because of the safety reasons then how administration able to manage the customers and as well as the service providers. In the post covid time customers wants the services but with some terms like worker should have doubly vaccinated, he/she wearing the masks and gloves all the time many more. We have to track the details of the workers like experience, vaccination details, contact number, addresses, etc. Also, admin has to track on the service details like in pending mode, completed or delayed, it helps admin to manage the next assigned services.

Fact Finding Phase

Background Reading/s:

We know that in today's life it is easy to get plumbers, handyman, electrician because of mobile apps. So here we will see how data models going to do such type of work. This type of civil service apps is first found customers from different areas as much they can and after they are going to provide their services and get rating and feedback from their customers when that particular service is provided which customer needed. And from that rating and feedback from customers they improve their data model and system.

This apps store all the detail of services and it's also allowing customer to find the service provider by their needed service and its requirements (like – someone have some minor problem whereas other side someone has critical issue) and rate of services and many other aspects of services. The key fact of this type of service apps is to they are giving high qualified and trusted professionals which works within some criteria and all this thing is done by the service provider apps and it is make things easier for customers.

Here, we need to take care of about two main things, which is service providers and the customers. All the service providers are providing their services in several areas. A particular service providers manage its entire service process for his area. Based on the above things we need to make particular data model for each (ex. services, customers, customer and service provider details, rate of service, service category, payment details, rate per hour etc.) and relate it's with each other based on its relation. This all the data models are connected to each other and made the database which useful to both customers and service provider. Main purpose of generating the app and the database is to easily and reliably provide service to customers which we can do by this type of apps.

Here, we are mainly focusing on services provided to customer by company. In India most of the company which providing local services via app or any other online resources are not an Indian company. We only have one Indian company which is named urban company, it is providing most of the basic and local services which we are needed in daily life. The platform helps customers book professional home services such as beauty services and massage therapy for both women and men, cleaning, painting, plumbing, carpentry, appliance repair, and many more. This Indian company is present in Singapore, the UAE, and Australia as well. The main motive of the company is to provide the best services and provide the jobs for the professional workers at particular work. Since its launch, the company has evolved immensely, and currently, it offers services under two verticals – Beauty & Wellness, and Home Repairs & Maintenance. Service professionals are closed to 40,000, out of which 35000 are from India on its platform.

"These individuals [service providers] are the most atomic form of business units in our society... If India needs to grow, if consumers need to benefit, they have to be empowered" - As stated by Raghav Chandra, co-founder of Urban Company.

At the beginning of the starting the urban company in 2014, customers experienced issues in finding the services they truly required. Developer of the company is trying to solve this situation. They felt that there is much more gap in how people found services and in the manner in which they connected with the service providers. After that they had a idea of establishing an all-in-one platform. At initial point of the company, they build lead-generation model but over the time Urban Company adopted a full-stack model that onboards gig workers offering them financial assistance, skills training, access to branded tools and products, and a ready-to-serve market. This model is very good for the company because it provides the quality work to the company. This idea is new for the Indian market though the founders get millions of dollars of funds from the different ventures.

This company is hiring professionals which are providing their services individually in different areas. Urban company directly connect the customers with the professionals who are best in its service. Today, Urban company has more than 50000 professionals which are providing their services in best way possible. This company provide customers to fast access services through app, multiple payment methods, option to rate services and it is giving service providers to fix their charges for a service for particular time. It has a particular work flow which we are discussing now. Urban company has their own app from which user can go through all the services which are available and services provider which are ready to give services and customer can easily get the professional for their requirement for the service. Urban company charge the customers for services by two ways. First is to fixed the charge for service and other is service without any fixed charge. By the customer's request for services, professionals are giving the services and based on the service getting by customer, customers are giving feedback.

In current situation urban company gives their service in 30 cities of India like Agra, Ahmedabad, Bhopal, Bangalore, Indore, Jaipur, Kolkata, Mumbai, Lucknow, Vadodara, Delhi and so on the services are also available in international markets like Abu Dhabi Dubai, Sydney, Singapore.

Urban company changed his name to urban clap in January 2020 because in this huge change co-founder Abhiraj Bhal said, 'It is important to have a globally acceptable brand' and additional he said that 'urban Company is a simple name with universal appeals what remains unchanged in our commitment to offering reliable and affordable service at home. This is enabled by working closely with our service partners, helping them with up-skilling, financing, insurance, product, procurement, etc.'

Millions of people are familiar with the urban company because urban company always promoting their services by TV advertisement, online advertisement, Instagram network. Because of this company target a large number of audience and promote their service. They try to turn the viewers into their client with the medium of relative advertisement on television as well Instagram network. The online advertisement is very constructive as there are billions of users on Internet using social media. The urban company uses Google ads Facebook ads and YouTube ads for marketing their business also by posting attracting content on Instagram network for attract large number of people online. Working flow of customer and service provider in urban company: -

---

1) Customer

   Using the app, customer can go through all the list of resources and select the preferable service based on their requirements. This urban company's app allow customer to select any service they needed and they can choose it from range of choices for that particular service. Customers

can get all the information about service provider by that app. There are multiple payment methods for customer to pay for delivered service. They can rate the any particular service by their experience and give their valuable feedback to services or service providers.

2) Service provider

This app is allowing service providers to fix their rate for their services. When any customer requests any service provider for service, service provider will get a request on their profile and they can accept or reject it. When service provider accept any services request, alerts are received from the customers and service provider can find if they back out from work after accepting the request.

The main challenges for this type of companies are Matching Supply and Demand, Ensuring Quality of service and User Retention. Urban company is the marketplace for finding service professionals means they need to handle both the users and the service providers that means the effort for advertise is double. The trust is the main thing of the business but here if the service provider is doing not good than that affect the company not to the service provider because it is connected with the company's name. The user retention is very major challenge because if the customer wants the plumber and check the company's plumber status if it is available but 2-3 hours delay happens then for the next time customer think direct call the plumber and through not application.

A. References

- https://www.vertabelo.com/blog/data-model-design-a-mobile-app-marketplace-for-local-services/
- How UrbanClap (Now Urban Company) Works - Business Model Explained (oyelabs.com)
- https://www.analyticssteps.com/blogs/urbanclap-urban-company-success-story
- https://www.quora.com/What-are-the-challenges-of-startups-like-UrbanClap-and-how-do-you-overcome-them-specifically-for-vendor-management

B. List the combined Requirements gathered from Background Reading/s

- To provide services easily and reliably to customer and as fast as possible by customers mentioned requirements.

- Get all the updates of customer, services and service providers.
- Update data model and app system by customer's rating and feedback for services
- Give them trusted and high qualified professionals who are best in their services.
- Provide Different payment method to customers.
- Provide as much as services possible.
- Give category in services so customers can get their services by their requirements.
- Provide job to service provider by their skill.
- Efficiently manage customer and service provider.

Interview/s

1) Interview 1

Role Play Interview Plan

System: Service request for civil works system database

Interviewee: 1) Abhiraj Singh Bhal
Designation: CEO at Urban Company
Organization Details: Urban Company, New Delhi

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 09/10/2021 **Time:** 15:30

Duration: 75 minutes **Place:** Google Meet

Purpose of Interview:

To increase the number of services, improving quality of service, customer satisfaction, etc

Agenda:

Problems and concerns with services

Initial ideas

Follow-up actions

Documents to be brought to the interview:

Structured Database for managing data

---

2) Interview Summary 1

System: Service request for civil works system database

Interviewee: 1) Abhiraj Singh Bhal
Designation: CEO at Urban Company
Organization Details: Urban Company, New Delhi

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 09/10/2021 **Time:** 15:30

Duration: 75 minutes **Place:** Google Meet

Summary of Interview:

To increase the number of services, improving quality of service, customer satisfaction, etc

1. Quality is the key factor, if quality is improved, so the customer is satisfied, and that customer would choose us again for another service, also customer would suggest somebody for our service

2. The satisfied customer thus suggests others for the service, this is indirect marketing

3. First service discount, festive offers attract customer.

4. Giving the workers a Uniform is very important part, it increases brand value of the company

5. Providing sanitized and clean services ensures quality and COVID guidelines

6. Providing service rating in feedback and if is positive, the company is progressing.

7. Workers employed should be enough so that service can be done without much delay.

8. We must ensure affordable priced services, so that we can knockout our competitors.

9. If all these are followed, then the customer would definitely be satisfied

10. For Further Queries, reach out Anil and Anurag Joshi via email or contact no.

---

3) Interview 2

Role Play Interview Plan

System: Service request for civil works system database

Interviewee: Rohan dave , Ruchi dave

Designation: customer

Organization Details: customer of existing company

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 07/10/2021 **Time:** 14:30

Duration: 45 minutes **Place:** Cisco webex

Purpose of Interview:

Preliminary discuss with Rohan to know their opinion for building new company which give batter services with compare to the urban company

Agenda:

Problems with security and any other concerns

Current security procedures

Initial ideas

Follow-up actions

Documents to be brought to the interview:

Rough plan for building new service provider company.

Urban company latest database. (As document)

---

4) Interview 2 summary

System: Service request for civil works system database

Interviewee: Rohan dave , Ruchi dave

Designation: customer

Organization Details: customer of existing company

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 07/10/2021 **Time:** 14:30

Duration: 45 minutes **Place:** Cisco webex

summary of Interview:

Preliminary discuss with Rohan to know their opinion for building new company which give batter services with compare to the urban company

1. First of all, we need to know what customer requirement and how we can give satisfy services.

2. Before building the company, we must be aware with feedback off urban company and also make some brief database about what urban company was not to do for their customer and why did customer are not satisfied with their service.

3. Make some research and then improve the quality and the system which help to give fully satisfaction to the customer with compared to urban company.

4. For giving better service the company must be hire experienced and qualified employees which Can give better services and also, they all have good sense of humour.

5. We promoting our company in all sector with the help of some famous celebrities and also giving discount on first service and putting some special offer for attract the customer.

6. Make a large network of the employee which Located in all state district and villages because of this we must be ready to operate the request and give service at less time.

7. We make special ID for all the customer and give them special gift who take our services many times.

5) Interview 3

Role Play Interview Plan

System: Service request for civil works system database

Interviewee: Gauri Chauhan

Designation: Employee of urban company

Organization Details: Urban company office

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 06/10/2021 **Time:** 18:30

Duration: 45 minutes **Place:** Google Meet

Purpose of Interview:

Improving Worker's Day at Urban Company

Agenda:

Problems and concerns with workers/employees

Initial ideas

Actions to improve working environment.

Documents to be brought to the interview:

Structured idea or Database for managing environment and work load of the employee.

---

6) Interview 3 summary

System: Service request for civil works system database

Interviewee: Gauri Chauhan

Designation: Employee of urban company

Organization Details: Urban company office

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 06/10/2021 **Time:** 18:30

Duration: 45 minutes **Place:** Google Meet

Summary of Interview:

Improving Worker's Day at Urban Company

1. Workers and employees are the lifelines of the company, we should keep them always in a good mood.
2. We should not be too strict to them in matters of arriving and departure timing 5-10 minutes late is Okay.
3. Environment is key factor; the office should be cool and hygienic also there should be COVID guidelines followed.
4. We should manage company's profit and employee's salary increment in an organized way so that employee is satisfied with the salary, also it doesn't affect company's profit.
5. Small Festival Bonuses and treat for employee's doesn't affect company's profit but the employee's mood is delighted.
6. Overall Good conversations with the head is key, Head managers should not treat them as slaves and understand the worker's concerns.
7. These all would make an employee to enjoy and work at our company.
8. For Further queries, contact Interviewers.

---

7) Interview 4

Role Play Interview Plan

System: Service request for civil works system database

Interviewee: Alka parikh

Designation: Economist

Organization Details: DA-IICT faculty office

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 05/10/2021 **Time:** 17:30

Duration: 65 minutes **Place:** Zoom

Purpose of Interview:

Marketing ideas to increase Company's profit

Agenda:

Problems which block paths of profit

Efficient investment and increased profit

Deals which seems affordable but also increases company's market

Documents to be brought to the interview:

Structured idea and managed Database for managing investment and profit of the Company

---

8) Interview 4 summary

System: Service request for civil works system database

Interviewee: Alka parikh

Designation: Economist

Organization Details: DA-IICT faculty office

Interviewer: 1) Harsh Mehta **Designation**: student at DA-IICT

2) Jinal Chauhan **Designation**: student at DA-IICT

Date: 05/10/2021 **Time:** 17:30

Duration: 65 minutes **Place:** Zoom

Summary of Interview:

Marketing ideas to increase Company's profit

1. We have to design our type of services in such a way that if a customer comes for an affordable service but is also impressed for a little high-end service and buys that instead.

2. Advertisement is the Key factor, we should advertise the services and saying that the service starts at rupees = "the cheapest service price".

3. If a customer visits the application, always offer them discounts and coupons accordingly, where the services are less – releases coupons for it to attract users.

4. Advertisement should also be in areas where there is market potential, advertising where is not good market is waste.

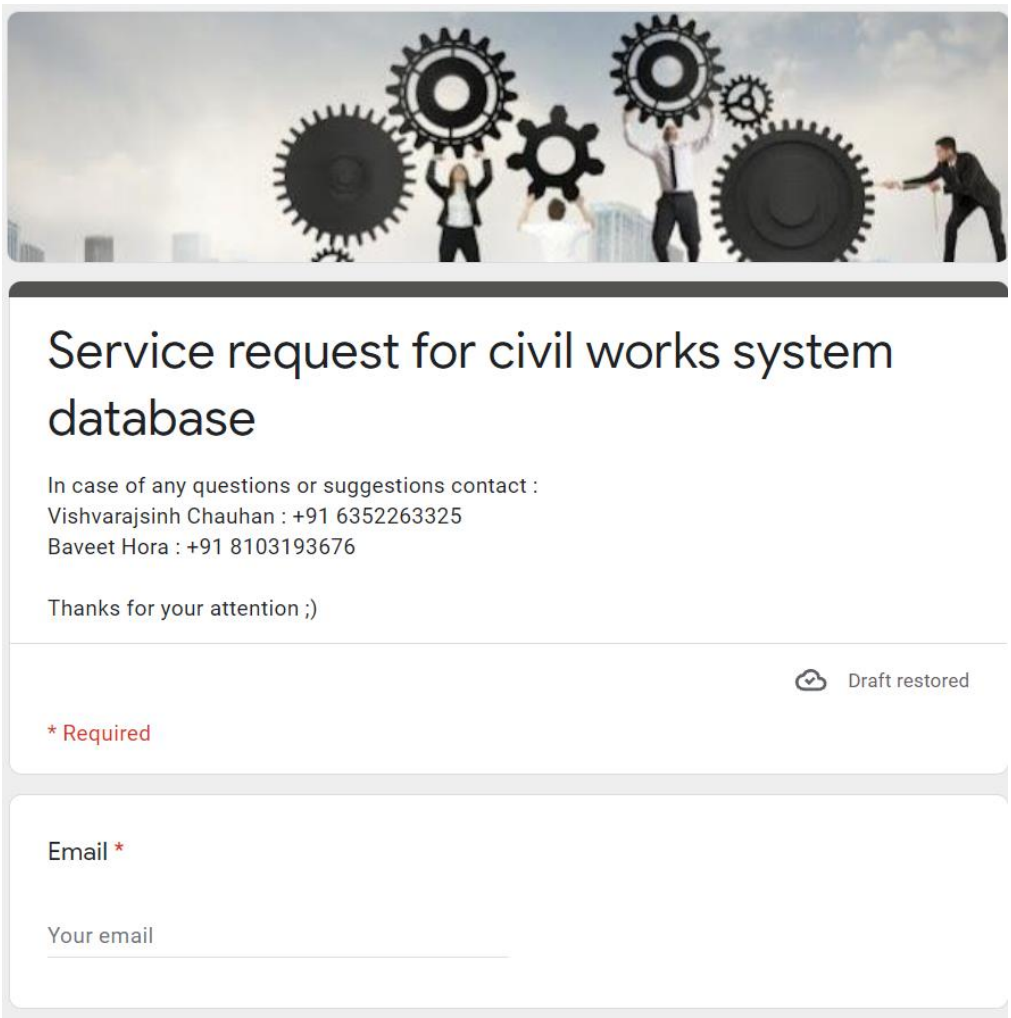
5. If the company is increasing, it would further get increased by sponsor and endorsements

6. The proper management of profit, investing, employee's salary is a challenge. We should see things where we can save investment, increase profit.
   E.g., service parts from a wholesale vendor.

Questionnaire/s –

We have collected 32 responses for our SRS project through google form. Following are the outcomes of our survey with some observations and google form's structure.

https://forms.gle/Vod38rzcECnvxZWCA



Name *

Your answer

Which age group do you belong to *

- ◯ 15 - 25
- ◯ 25 - 50
- ◯ 50+

Do you interested in online household services? *

- ◯ Yes
- ◯ No

If yes then which services are you looking for? *

- ☐ Plumbing
- ☐ Decorative Lighting
- ☐ Electrician
- ☐ Regular services for the AC , Refrigerator ,etc.
- ☐ Cleaner
- ☐ Salon
- ☐ None
- ☐ Other:

Urban Company is the only company in India which provides online household services. Do you want a new company with more services and compete with the Urban Company? *

- ◯ Yes
- ◯ No

If yes then which features are not good in Urban Company? *

- [ ] Not more services
- [ ] Quality of services
- [ ] Waiting time for the service
- [ ] Payment method
- [ ] Other: _____

Would you like to suggest any new feature in the platform that will serve you in a better way?

Your answer

Send me a copy of my responses.

Submit Clear form

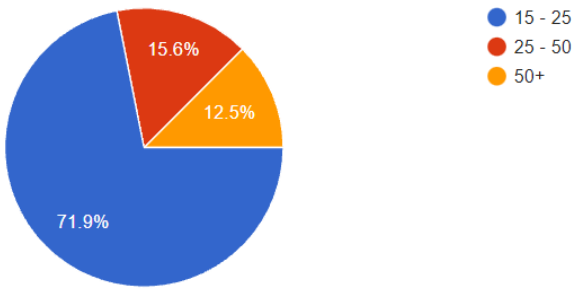
Never submit passwords through Google Forms.

reCAPTCHA
Privacy Terms

Responses of this survey: -

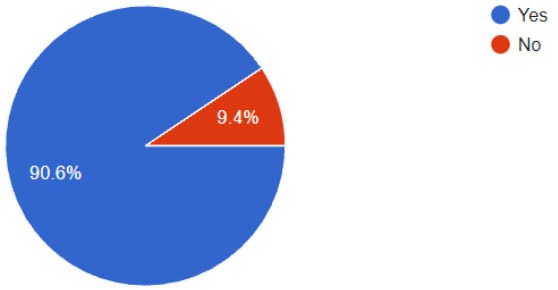
Which age group do you belong to

32 responses



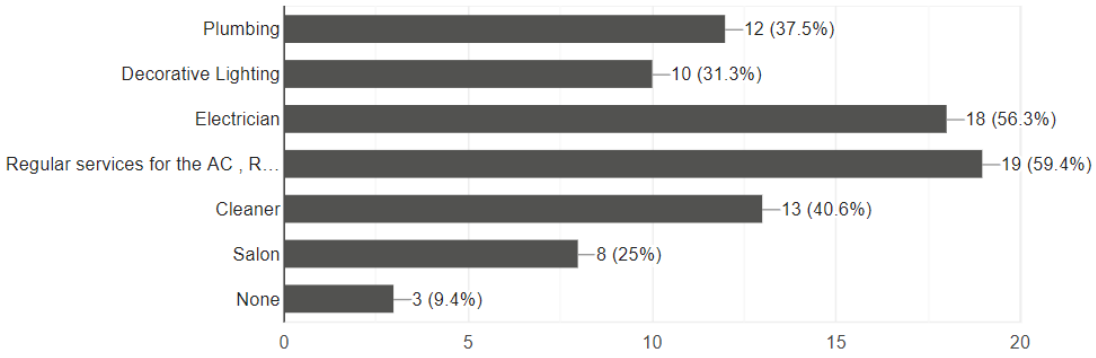
Do you interested in online household services?

32 responses



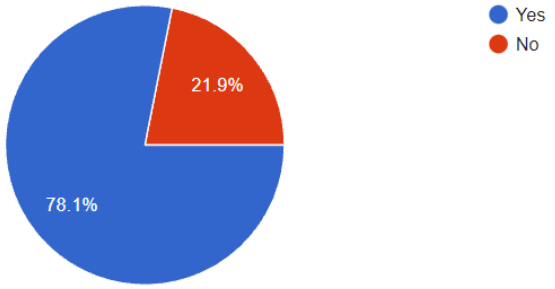
If yes then which services are you looking for?

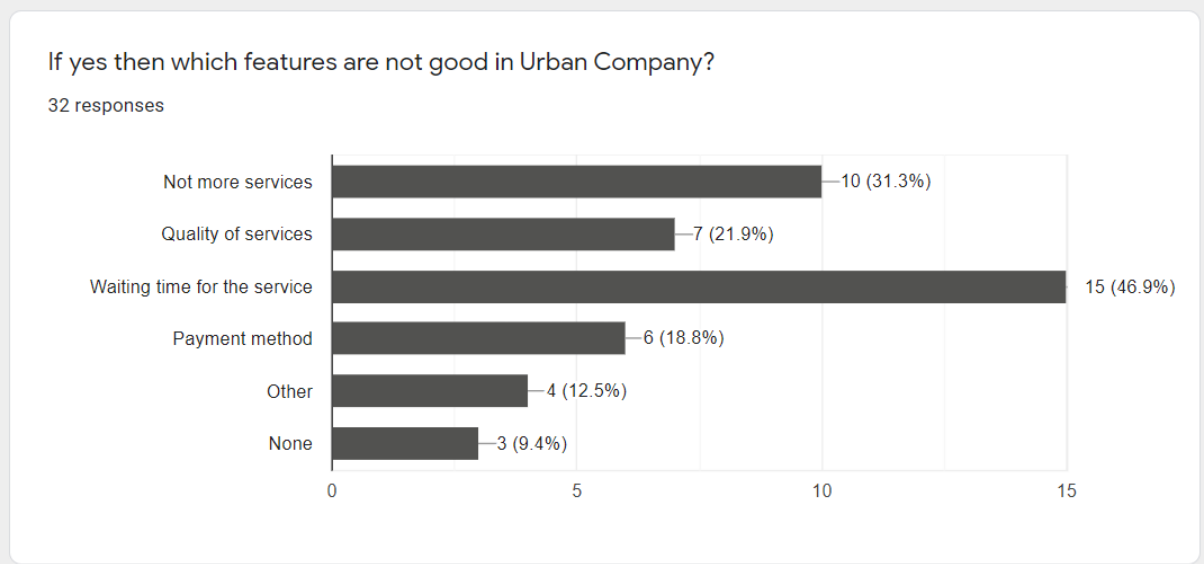
32 responses

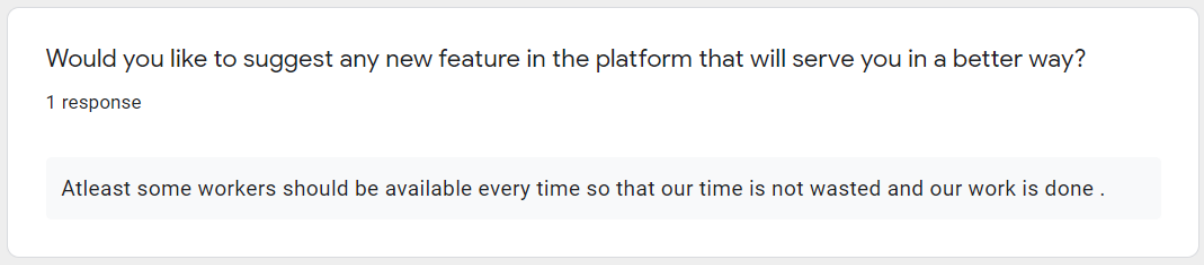


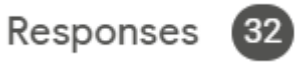
Urban Company is the only company in India which provides online household services. Do you want a new company with more services and compete with the Urban Company?

32 responses









Observations

Background Reading Observation

How mobile apps changed the way of working life of part time service workers?

They are actually were freelancers until this application. Now they have a stable area where they can get services and jobs more easily. Also, if a person moved to a new city, where can he get the service without knowing any worker out there. This app is a gem for that, it would provide you a quality and affordable service. Also, this apps store all the detail of services and it's also allowing customer to find the service provider by their needed service and its requirements.

The key fact of this apps is that we are giving high qualified and trusted professionals which works within some criteria and all this thing is done by the service provider apps and all these things makes "Getting a service" easier for customers.

All the efforts done by company in providing a nice service, managing workers and customer satisfaction as well as the profit of company is challenge. All the data models are connected to each other and made the database which useful to both customers and provider to manage all things properly. The second paragraph is actually based on references, it tells us the journey of the Urban Company-the only Indian service provider company. Today we all knew something about the company even if one

has not taken the service is due to their investments in Ads running on various platforms like Facebook, Instagram, YouTube, etc

Some prime features are:

As for **customers,**

They can go through all the list of resources and select the service based on their requirements. This app also provides customers with a wide range/big list of services and the best service to choose from. Customers can get all the information about service provider by that app. There are multiple payment methods, also customer can rate the any particular service which they received by their experience.

As for service **providers,**

When any customer requests a worker for service, service provider will get a request on their profile and they can accept or reject it. When service provider accepts any services request, alerts are received from the customers, also a notification is sent to the company, informing that this worker is busy in that service time. Also, a customer can find if they back out from work after accepting the request.

Finally, there should not be delay, as it's like daily need service, user can go to other worker/apps.

Interview Observation

All the interviews taken by the Urban Company's head peoples, employee, users and economist's Interviews are being discussed on,

- To increase the number of services, improving quality of service, customer satisfaction, etc
- Problems and concerns with services,
- Preliminary discuss with Rohan to know their opinion for building new company which give batter services with compare to the urban company
- Improving Worker's Day at Urban Company,
- Marketing ideas to increase Company's profit,
- and many points,

discussion about, to improve the company in each and every prospective were a lot,

Some are,

- Quality is the factor, if quality is improved, so the customer is satisfied, and that customer would choose us again for another service, also customer would suggest somebody for our service
- The satisfied customer thus suggests others for the service, this is indirect marketing
- First service discount, festive offers attract customers
- Giving the workers a Uniform is very important part, it increases brand value of the company
- Providing sanitized and clean services ensures quality and COVID guidelines

Marketing discussions like,

- Advertisement is factor, we should advertise the services and saying that the service starts at rupees = "the cheapest service price".

- If a customer visits the application, always offer them discounts and coupons accordingly, where the services are less – releases coupons for it to attract users.

- Advertisement should also be in areas where there is market potential, advertising where is not good market is waste.

Also, well-being of service providers is must as they are the lifelines of the company, In these areas, discussion were,

- Environment is the factor; the office should be
- cool and hygienic also there should be COVID guidelines followed.
- Small Festival Bonuses and treat for employee's doesn't affect company's profit but the employee's mood is delighted.
- Overall Good conversations with the head is key, Head managers should not treat them as slaves and understand the worker's concerns

Interviews were filled with great points, which when implemented, can boost the company in all areas – profit, quality, employees.

Questionnaire Observation

A total of 32 Responses received by our survey, we reached results

- Most Age group in survey = 15-25
- Do you interest in household services? = 90.6% interested
- If yes which service, are you looking for, most were- "regular services for AC".
- Urban Company is the only company that provides online household services in India? Do you want a company that provides more services to compete with urban company? – 78.1 % agreed to this.
- If yes then which features are not good in Urban Company? – Most Were Waiting time for the service – 46.9 %
- Would you like to suggest a new feature in the platform, that will serve you in a better a way? – Best was = "Some workers should be available all the time so that our time is not wasted in getting the service".

The Responses tells us people just not want to wait for the service, it's like going to app, book a service and just get it without any delay. Also, most of them want Regular services Of AC. So, it should get started asap.

Fact Finding Chart

objective	Technique	Subject(s)	Time commitment
To get background and history of the Urban company and information about to how company is hiring professionals and also there are some brief ideas of	Background Reading	Urban company report and Work flow and history of urban company	1 day

working flow for customers and service providers in Urban company's app			
To improve quality and quantity of services based on customer's feedback	Interview	CEO of the Urban Company	75 minutes
To establish business objectives. Scope to make new company and some basic information which is very useful to establish company based on customer's requirement and many other aspects	Interview	Business Development Executive and Developer – Urban company	45 minutes
To determine workers requirement and get idea of relationship between workers and managers of company	Interview	Most users	45 minutes
To find ways which we can make publicity of our company and way to manage profit, salary of worker, investment of company etc.	Interview	Economist	65 minutes
To get customers response and feedback about company and services which are needed in daily life	Questionaries	Service request for civil works	2 minutes
Summary of all the above points	Observation	All of the above	4 hours

List Requirements

- Give category in services so customers can get their services by their requirements.
- Give them trusted and high qualified professionals who are best in their services.
- Provide Different payment method to customers like online payments options (Phone Pe, Paytm, google pay) and offline payment option (Cash on service).
- Giving satisfied service to the customer.
- Update data model and app system by customer's rating and feedback for services. We have to carry the data of the customer's feedback.
- If customer trusted our company and taking many services from our company, then we have to maintain the trust by giving them special voucher or giving them some gifts or some offers.
- Be aware of timing. Give services as per time which recommend by the customer.
- For giving good service and improving rating of feedback of service it is important to higher educated and well experience employees.
- Make special ID for all the customer who taking service many times.

- Taking diligently work from employee and make a good progress.

- For maintaining diligently work we should give them small festival bonus and treat. also, the company profit and employee's salary increment in an organized way so that employee is satisfied with this salary also it doesn't affect companies profit.

- We should give advertisement in TV television, Internet, social media and also making some interested content and put it on Instagram, Facebook, and YouTube for attract large number of people.

- from the survey the company need to Maintaining their time for the service. Many people excuse that they are not satisfied with the company because of the waiting time for the service.

User Classes and Characteristics

1) List User Names with a basic description of user role in the system/database

Administrator, customer and service provider are the three main users of this system.

Administrator: -

Administrator can access the entire database. Administrator can be allowed to add, delete, update the customer, Administrator and service provider information or any other information from database where as customers and service provider does not have these privileges which Administrator has. Admin has privilege to add any service or delete it or it can be changing the rate and information of services where customer and services provider does not have this privilege.

Customers: -

Customers can access the home page and view the details of services, category of services, services providers and information regarding services and service providers to get those services. Customer also able to change his or her profile and give the rate and feedback to services and services providers and also customer can suggest any services through feedback which needed for it. Customer only have privilege to change his or her profile, customer has not any privilege to add, delete or update any information regarding administrator, services and service providers.

Service provider: -

Service provider can able to add, delete or update his or her profile. Service provider does not have any privilege to add, delete or update information of administrator and customers. Service provider can able to change information of his/her services (like – rate, charge per hour etc.) but it can't able to change any other service providers information or any other services which is not provided by his/her. Service provider can access the information of customers who request his/her for service to provide service but it has not privilege to change any customers information.

Electrician: -

Electricians are responsible for inspecting, testing, repairing, installing, and modifying electrical components and systems. Electricians general work at homes, businesses, and construction sites, and generally work as contractors.

Plumber: -

plumber's general purpose is to supervision, maintain the flow and drainage of water, air, and other gases by assembling, installing, and repairing pipes, fittings, and plumbing fixtures districtwide. Also Maintain and provide for the safe condition and operation of all plumbing systems in district facilities.

Yoga Instructor: -

The main duty for yoga instructors is to create a curriculum and lead groups through various levels and types of yoga practice. This can include doing administrative work for the class, getting equipment together and getting to know what participating clients want and Demonstrate practice and technique according to client's requirement.

House Keeping: -

In housekeeping, A Housekeeper is responsible for taking care of a building's general cleanliness to provide tidy and sanitary amenities to guests and residents. Their duties include cleaning floors, making beds and dusting surfaces throughout a home or other building.

Pest Control: -

Pest control workers are responsible for inspecting buildings and the surrounding property for signs of insects, rodents and other pests. They determine which treatment is best and use the proper baits or traps to remove them. Pest control workers also create barriers to prevent insects and rodents from re-entering.

Operating Environment

1) Hardware, Software or Connectivity Requirements

This type of web application solves the household problems and provide a platform to make interaction between customers and service providers to deliver the desired services.

Hardware requirement: -

At customer side the minimum hardware requirement is that we need RAM at least 512 MB, hard disk need at least 10 GB storage and processor need at least 1 GHz speed.

At server side the minimum hardware requirement is that we need RAM at least 1 GB, hard disk need at least 20 GB storage and processor need at least 2 GHz speed.

Software or connectivity requirement: -

At customer side we need good web browser (like – google chrome, Firefox or any other browser) and as part of connectivity we need Wi-Fi or mobile internet. We also need operating system like – windows or any other equivalent operating system.

At server side we need APACHE web server and as part of language we need HTML or JavaScript or any PHP version so server can understand the requests. We also need pgAdmin 4 as Database server. We need web browser, connectivity and operating system same as we require at customer side.

2) External Interface Requirements

There are 4 external interface requirements and all requirement need to be fulfilled at both server and customer side.

User Interface: -

Initially Web based Graphical User Interface (GUI) will be provided. Here, the Web portal must be completely menu driven and user friendly. In the GUI, there are various input forms and output screens are available with help of file as per requirement. So, the different people (ex. Customers, workers, administration etc.) can see the result as per requirement.

Hardware Interface: -

We know that this application completely designed on website, so we need a hardware which have faster internet connectivity and can able to host the web server. In hardware interface, it is required that our machine on which this app is running can able to run virtual software on web server and there must be a facility of future upgrade, memory storage etc.

Software Interface: -

We must need APACHE server to host the website. The Software must have many functionalities or language support (like – HTML, CSS, JavaScript etc.) to make interaction between user and server. There must be any operating system (like- window, Linux etc.) and web browser (like- google chrome, Firefox etc.) which can run this type of application and their also needed stable internet connectivity.

Connectivity Interface: -

Mostly all the interaction between customer and server is happen on website with HTTP protocol, so there must be support of HTTP protocol. And also, sometimes system needs to generate automated message or mails so there is need of email server as part of connectivity interface. We also need wi-fi or stable internet connectivity.

Product Functions

- **Login:**

If the customer is already registered then he/she have to fill the user_id / email_id and password for logging into the dashboard of the system. If the customer is new to platform, then he/she needs to register with contact details and new password. If the customer forgets the password, then we also provide the reset or update the password through security steps.

- **Customer information checker:**

We take the information about the customer for knowing the addresses and contact details of the customer. If the customer not fill the required information field, then it shows notice that “some information may not fill by you kindly fill that information”. If the customer fill faulty value in phone number like 10- digit is must for India then it also shows notice that number is wrong and same for email id.

- **Scheduling the services:**

Our system is real time application therefore many users are online at one point of time. This may lead to clash the same services to one worker therefore system has to check that the earlier service may set first in the queue of the worker's services information and second number may set at second number in the queue.

- **Cancel the services:**

If the customer book the normal service, then he/she can cancel the service by calling the cancel services function before one hour of the requested time. If the customer book the fast service, then then he/she can cancel the service by calling the cancel services function before 30 minutes of the requested time.

- **Finish the services:**

Many workers are assigned many services at one time. System has to track the service status of all the services that are demanded by the customers. If the service in the pending mode, then we can think that, that service is not entertained by any worker but if it is completed then it shows completed in the service status and it prompted users to feedback and rating page. Whether customer fill or reject the feedback rating page, system pass that service from the worker's service information and assign next service to that particular worker.

- **Worker's payment:**

We have many workers due to demand of the service. We have to pay the monthly salaries to the workers. For that we have to calculate his/her salary according to number of services. If worker one has N number of services and N/2 number of services has the full rating in the feedback then salary = N (one service value) + N/2 (Bonus on the service). This function is only for the administration and service provider, customer has no privilege to access this function.

- **Calculate the Profit and Expenditure:**

Every time company thinks about the profit and expenditure. This function calculates the profit of the company by deducting the service provider's cut and the cost of parts needed for the service. Admin can also manage the Finance and tax management according to profit and expenditure.

- **In-app chat function:**

In the application there is a pop-up chat icon at the bottom of the screen if a user has any issue regarding the service, then there our help provider will help them in the live chat session, customers can ask any query which they want to ask and also call them if necessary.

- **Super Service:**

If a particular service has less booking count, we can shift the service to a section called super service. It contains all the service which are currently discounted, the discount attracts customers here which increases the chances of booking of that service. Even if there is loss in there than usual, but it is better to have something than no service. Worker is then asked if it would provide service is that discounted price, if agrees then the service is done.

- **Service and payment history:**

We have one service and payment history function which can store all the services history which was done by the service provider in its section. Customers have an option called customer history which has the data of service history and payment history, customers can cross-check their payments and services accordingly.

Privileges:

- Price rate of the service:
  - read /update/ delete: Administrator
  - read: all the customer/service provider

- Contact details of company:
  - Read /update /delete: Administrator
  - Read: all the customer as well service provider

- Available service provider and also supporting staff:
  - read /update/ delete: service providing company like urban company (UC) which can alter the database according to the new announcement.
  - read /update: service provider/company staff
  - read: all to customer

- Service management:
  - Read /update/delete: Employees of the service providing company/ authorized staff of the company
  - Read: all the Customer/service provider

- Quick update on government policy:
  - read /update: service providing company and also the other government supporting company.
  - Read: all the customer / service provider

- Servicing status:
  - read /update/ delete: the head of service providing company and their mean staff.
  - Read: all the customer / service provider

- Special section for those who recommend company for add new service.
  - Read /update/ delete: service providing company
  - Read: all the customer as well service provider

- Feedback of the customer:
  - read /update/ delete: service providing company/main staff of the company
  - Read: all the customer as well service provider

- Important details about this service:
  - read /update / delete: service providing company/ main staff of the company
  - read/Update: Service provider as well company main staff
  - Read: call the customer as well service provider

- Complain box:
  - read /update/ delete: service providing company /main staff of the company /service provider/other government supporting company
  - read: call the customer as well service provider

- Number of customer details reason wise/number of service reason wise/available service provider reason wise/availability of tool for the service.
  - read all the customer details
  - update: only authorized by the service provider company, like example urban company (UC).

Assumptions:

- Customer can request their service only through the application or by the official website of the company, so it is mandatory to give them very good and fast server which never be go down.

- Most customers love receiving free information and learning about things that impact their regular life or the lives of those around them. So, think about what our customers really want to know, and which type of information would make their lives easier we have to put it on our site or application.

- Create blogs, eBook's, emails, and videos of the services which creates the bond between users and the company as well as invokes mutual trust.

- Internet connectivity is very important for the service request.

- Make brief information about the supply and demand curve like if demand of the services is increase then we get a good profit otherwise we might be faced loss.

Business Constraints

- To provide a reliable application on website from where customer and service provider can interact to each other for services and make app easy through artificial intelligence and machine learning.
- Go through the rating and feedback of the customers based on that improve the database and service information
- Price controlling for services must be important to satisfy the customer's requirement.
- Marketing strategy for the company.
- Mange the salary of employees, investment, expenditure on market effort.
- Provide a trusted and highly qualified service provider to customer.
- Make a strategy to expand the business in all over world and made the strategy which is different than other company and also made it attractive.
- Video advertisement on platform like YouTube, Instagram, Facebook and make interesting video which can catch the audience's interest.
- Solve the problems of customers and service providers to make a good and trustable relationship with customer and service provider.
- Find a funds and investors for company so we can face any critical issue if occur.
- Make a different payment methods option for customers.

Section 2
Final Noun Analysis

Extracted Nouns & Verbs from Problem Description

Nouns	Verbs
Database	connect /Manage/ Information/alter
Service Information	Information/reach
Service name	alter
Service ID	alter /Provide
Civil works	provide
Electrician	for
Painter	provide
Decorator	for
Plumber	provide
Home keeping services	Organized
Makeup artists	Assign
Lawn care	clean /beautiful
AC service & repair	need
Massage	Relax
Spa	Relax
repair & maintenance,	Provide/need
Event/ Party management	need
Health & Wellness	Wants
Yoga instructor	Manages
Pest control	need
Cost of services	price
System	manage
Customers	information
Customer ID	Provide
Customer name	Need
Customer Phone number	Need
Customer address	Need /Focus
House number	
Street	
PIN	
State	
recommendation	Facility
Delivery	collect
Application for Payment	wants
Service provides/Workers	Information/Qualified
Worker ID	Provide
Worker name	Need
Worker Phone number, email	Need
Worker Status	Supply
Duty	Deliver
Feedback/ Rating	Give
Carpenter	
Application/ Websites	Provide
Salon	
Service Request	Need
Service Date/Time	Implement
requirements	wants
Constraints	

Worker information	Reach
Track	
Owner	Author
Accurate	
Assigning	Connect
Service requests	
Priority	Check
Delay	
Market	Profit
Disaster	
Impression	Monitor
Demand	
Safety	Wants
Vaccination	
Gloves/ Masks	Need
Details	Information
admin	Authority
Stores	
History	Record
Service Payment	
Time/ Date	Implement
Employee/workers	Provide/service
internet	Connect
Add	
Attracts	Quality
Benefits	Profit
Support	Activity
Chat	Support
Areas	Information
Contact/ Phone numbers	Manage
Company	Facilitates
Data	Information
Efficiently	Quality
Situation	Test
Interface	Relationship
Bug	
Revenue	
features	facilities
Request	
Refused	
Available	Status
Attractive	
Application	Manage/Alter
Requirements	Need
Customer	Want
Affordable	
Data model	Update/Delete/Insert
Application system	Manage/Alter
Worker Salary	Provide
Organization	Manage
Worker	Provide
Business	Profit
Companies	Manage

Accepted Noun & Verbs list

Candidate Entity set	**Candidate attribute set**	**Candidate relationship set**
services	Service id	provide
Service category	Service_name	
Service provider/workers	Customer_id	Wants
Service information	Cost of service	
Worker salary	normal servive	Takes
History	Fast service	
Customer	Super service	Suggestion
Customer information	Worker_id	
Recommendation	Worker_name	Worker_service_info
Change password	Worker_Phone number	
Administrator	Worker_rating	
Customer review	Worker_status	
Service payment	Service_requests	
Application for payment	Service_date	Service/worker/customer history
	Service_time	
	salary	Salary_info
	bonus	Manage
	Total salary	
	Trnsaction-id	For
	Customer_name	
	Customer_password	
	Logout	
	Forgotten password	
	Customer_address	
	Street	
	State	
	PIN code	
	Country	
	Customer_email ID	Payment details
	Customer_Phone number	
	New password	Update
	Profit	
	Paytm	
	Phonepe	Facilities
	Google pay	
	Cash on services/direct cash	
	Feedback	Review
	Rating out of 10	

Rejected Noun & Verbs list

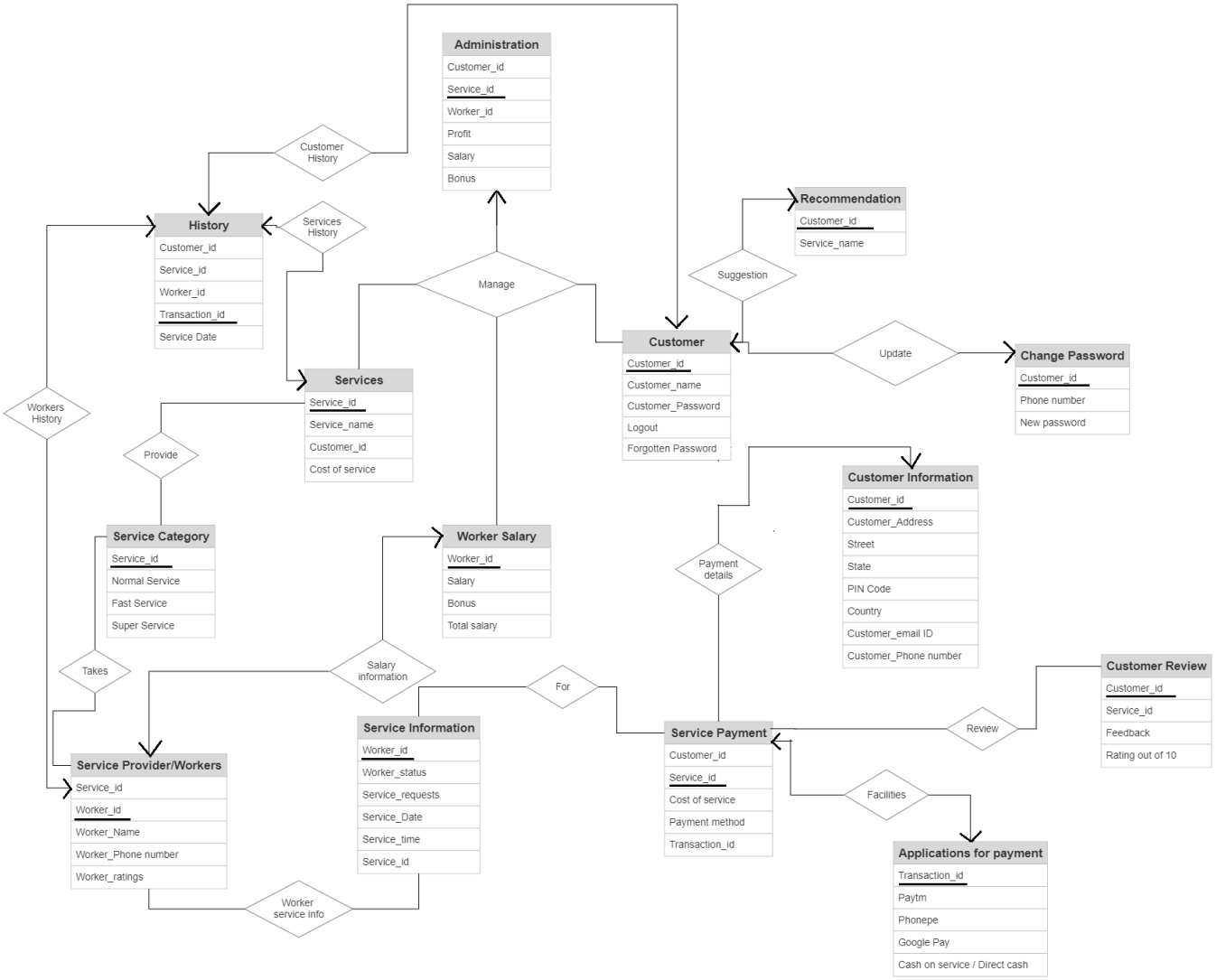
Noun/verbs	**Reject Reason**
Database	General
Service id	attributes
Service name	attributes
Civil works	Vague
Electrician	attributes
Painter	attributes
Decorator	attributes
Plumber	attributes
Home keeping services	attributes
Makeup artists	attributes
Lawn care	attributes
AC service & repair	attributes
Massage	attributes
Spa	attributes
repair & maintenance,	attributes
Event/ Party management	attributes
Health & Wellness	attributes
Yoga instructor	attributes
Pest control	attributes
Cost of services	attributes
Customer ID	attributes
Customer name	attributes
Customer Phone number	attributes
Customer address	attributes
House number	attributes
Street	attributes
PIN	attributes
State	attributes
System	attributes
Delivery	attributes
Worker ID	attributes
Worker name	attributes
Worker Phone number, email	attributes
Worker Status	attributes
Duty	vague
Feedback/ Rating	attributes
Carpenter	attributes
Application/ Websites	general
Salon	attributes
Service Request	attributes
Service Date/Time	attributes
requirements	general
Constraints	general
Track	General
Owner	Irrelevant
Accurate	Irrelevant
Assigning	Irrelevant

Care	Irrelevant
Service_requests	Attribute
Priority	Irrelevant
Delay	Irrelevant
Market	Irrelevant
Disaster	Irrelevant
Impression	Irrelevant
Demand	Irrelevant
Safety	Irrelevant
Vaccination	Irrelevant
Gloves/ Masks	Irrelevant
Details	General
admin	General
Stores	Irrelevant
Time/ Date	Attribute
Employee/workers	Attribute
internet	General
Add	General
Attracts	Vague
Benefits	Vague
Support	General
Chat	General
Areas	Attribute
Contact/ Phone numbers	Attribute
Company	Irrelevant
Data	Vague
Efficiently	Irrelevant
Situation	Irrelevant
Interface	Vague
Bug	Vague
Revenue	General
Request	General
Refused	General
Available	Duplicate
Attractive	Duplicate
Application	Duplicate
Requirements	Duplicate
Customer	Duplicate
Affordable	Duplicate
Data model	Duplicate
Application system	Duplicate
Organization	Duplicate
Worker	Duplicate
Business	Vague
Companies	Duplicate
Information/reach	General
alter	General
alter	Duplicate
Organized	Vague
Assign	Vague

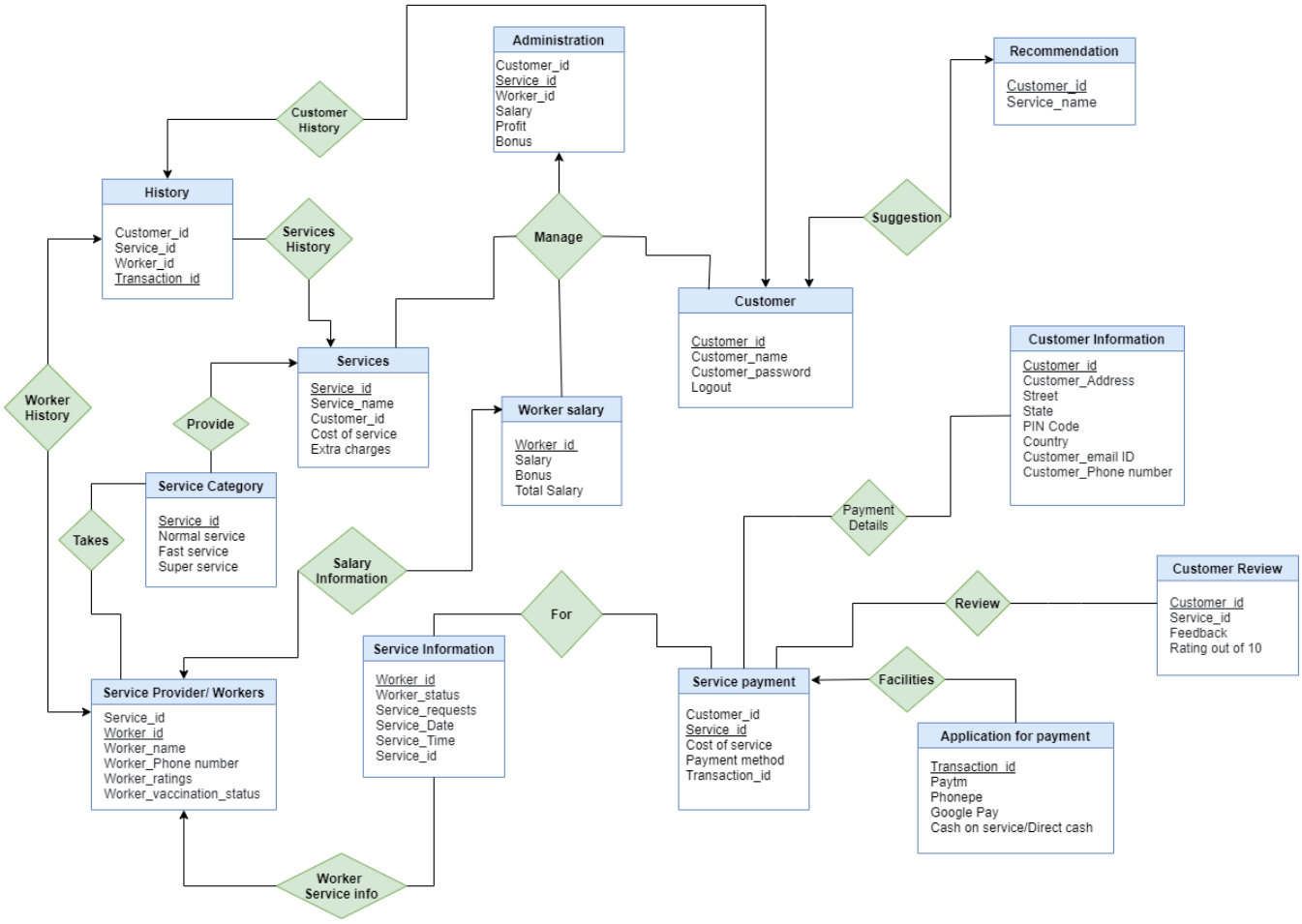
clean /beautiful	Irrelevant
need	Vague
Relax	Irrelevant
Relax	Duplicate
Provide/need	Duplicate
need	Duplicate
need	Duplicate
price	Vague
information	Duplicate
Need	Duplicate
Need	Duplicate
Need /Focus	Duplicate
collect	General
Information/Qualified	Duplicate
Need	Duplicate
Need	Duplicate
Supply	Vague
Deliver	Vague
Give	Vague
Need	Duplicate
Implement	General
Reach	Irrelevant
Author	General
Connect	General
Check	Vague
Profit	Vague
Monitor	vague
Information	Duplicate
Authority	Duplicate
Record	Irrelevant
Implement	Duplicate
Quality	Duplicate
Profit	Duplicate
Activity	Duplicate
Support	Duplicate
Information	Duplicate
Relationship	General
Status	General

Section 3
ER-Diagrams
(all versions)

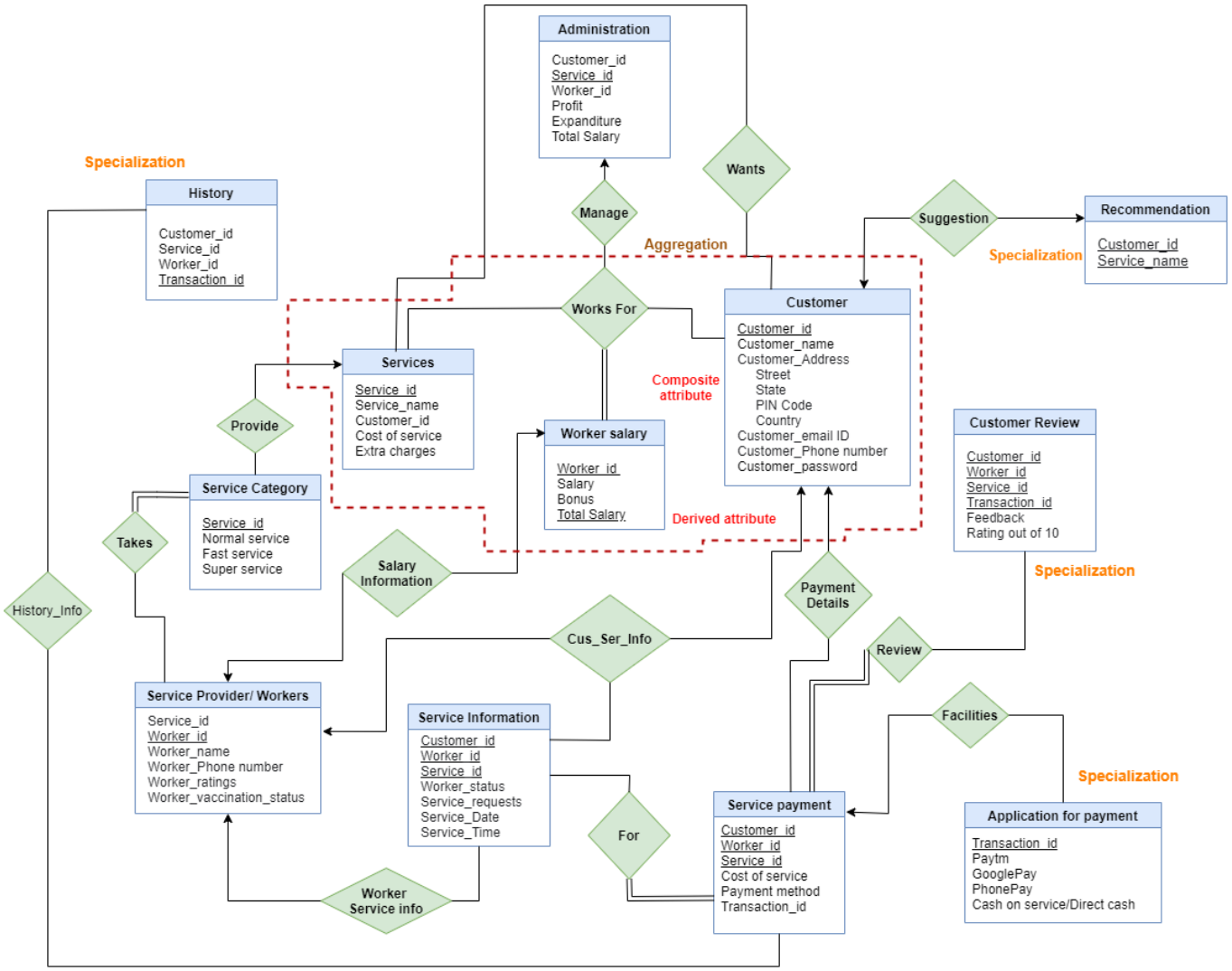
E - R Diagram V1: -



E - R Diagram V2: -



Final E-R Diagram:

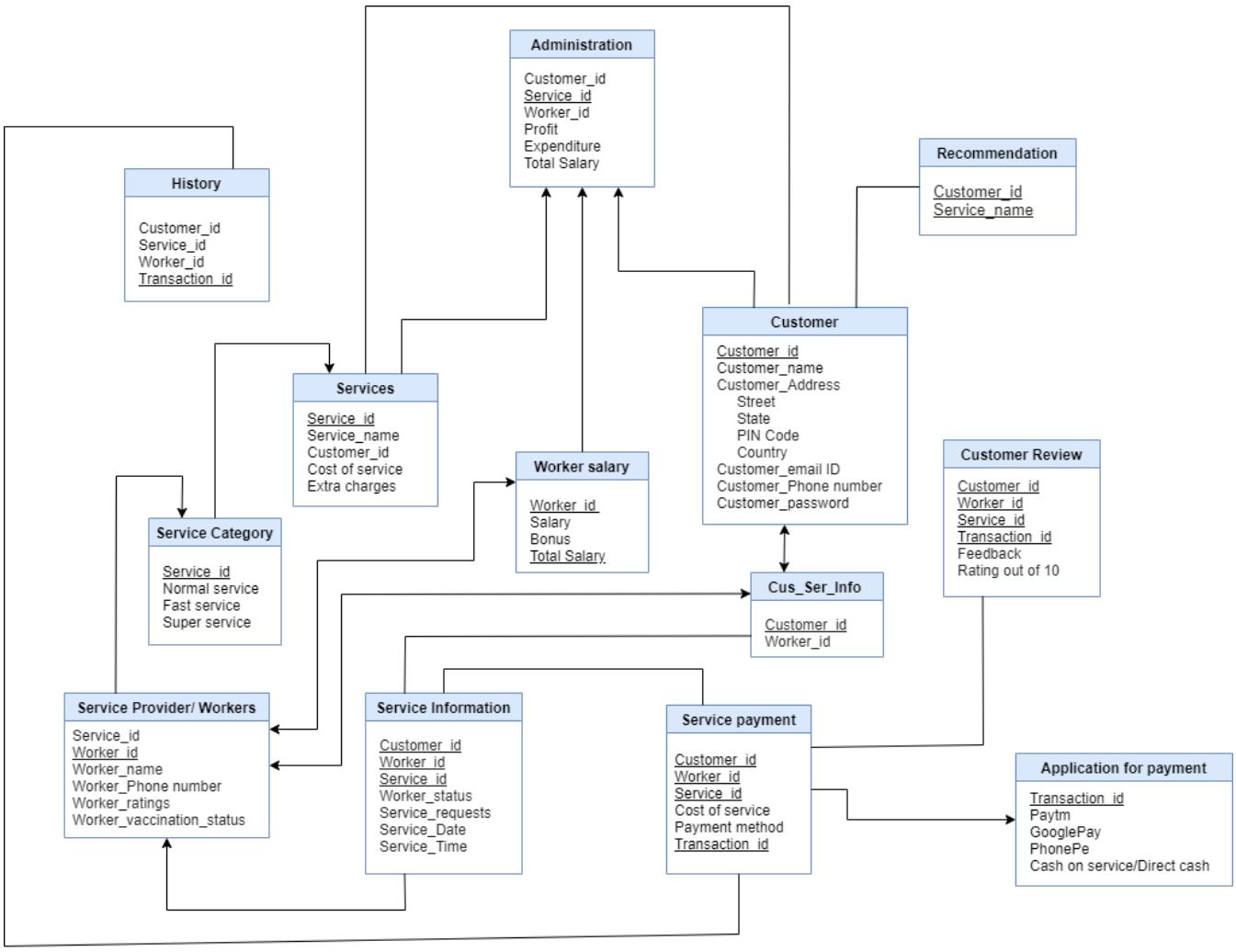


Section 4
Conversion of Final ER-Diagram to Relational Model

Mapping E-R Model to Relational Model

- **Customer** (**Customer_id**, Customer_name, Customer_Address , Street, Customer_State , Country, PIN_Code, Customer_email_ID , Customer_Phone_number ,Customer_password)
- **Services** (**Service_id** , service_name ,Cost_of_service ,Extra_charges)
- **Service_Category** (**Service_id** ,Normal_service , Fast_service , Super_service)
- **Service_Provider_Workers** (Service_id , **Worker_id** , Worker_name , Worker_Phone_number ,Worker_ratings, Worker_vaccination_status)
- **Administration** (Customer_id ,**Service_id** ,Worker_id ,Profit ,Expanditure ,Total_Salary)
- **Recommendation** (**Customer_id** ,**Service_name**)
- **Worker_salary** (**Worker_id** ,Salary ,Bonus ,**Total_Salary**)
- **Cus_Ser_Info** (**Customer_id** ,Worker_id)
- **Service_Information** (**Worker_id** ,Worker_status ,Service_requests, Service_Date, Service_Time, **Service_id** ,**Customer_id**)
- **Service_payment** (**Customer_id** ,**Worker_id** ,**Service_id** , Cost_of_service , Payment_method ,**Transaction_id**)
- **Application_for_payment** (**Transaction_id** ,Paytm ,GooglePay ,PhonePe, Cash_on_service_Direct_cash)
- **Customer_Review** (**Transaction_id** ,Customer_id ,Service_id , Worker_id , Feedback, Rating_out_of_10)
- **History** (Customer_id ,Service_id ,Worker_id , Service_Date, Service_Time, **Transaction_id**)

Relational model



Section 5
Normalization and Schema Refinement

Normalization and Schema Refinement

Customer (Customer_id, Customer_name, Customer_Address , Street, Customer_State , Country, PIN_Code, Customer_email_ID , Customer_Phone_number ,Customer_password)

- Primary key : Customer_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.
    2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
    1) Our relation is already in 3NF.
    2) For every functional dependency X → Y, X is the super key of the table.

Services (Service_id , service_name ,Cost_of_service ,Extra_charges **)**

- Primary key : Service_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    3) Our relation is already in 1NF.
    4) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    3) Our relation is already in 2NF.
    4) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
    3) Our relation is already in 3NF.
    4) For every functional dependency X → Y, X is the super key of the table.

Service_Category (Service_id ,Normal_service , Fast_service , Super_service **)**

- Primary key : Service_id
- Foregine Key : Service_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.
    2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :

1) Our relation is already in 3NF.
2) For every functional dependency X → Y, X is the super key of the table.

Service_Provider_Workers (Service_id , **Worker_id** , Worker_name , Worker_Phone_number ,Worker_ratings, Worker_vaccination_status **)**

- Primary key : Worker_id
- Foregine Key : Service_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.
    2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
    1) Our relation is already in 3NF.
    2) For every functional dependency X → Y, X is the super key of the table.

Administration (Customer_id ,**Service_id** ,Worker_id ,Profit ,Expanditure ,Total_Salary **)**

- Primary key : Service_id
- Foregine Key : Worker_id, Service_id, Customer_id, Total_Salary
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.
    2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
    1) Our relation is already in 3NF.
    2) For every functional dependency X → Y, X is the super key of the table.

Recommendation (Customer_id ,Service_name)

- Primary key : Customer_id, Service_name
- Foregine Key : Customer_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.

2) There is no transitive dependency exists in our relation.

- BCNF : Our relation is already in BCNF because :
    1) Our relation is already in 3NF.
    2) For every functional dependency X → Y, X is the super key of the table.

Worker_salary (Worker_id ,Salary ,Bonus **,Total_Salary)**

- Primary key : Worker_id, Total_Salary
- Foregine Key : Worker_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.
    2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
    1) Our relation is already in 3NF.
    2) For every functional dependency X → Y, X is the super key of the table.

Cus_Ser_Info (Customer_id ,Worker_id **)**

- Primary key : Customer_id
- Foregine Key : Customer_id, Worker_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
    1) Our relation is already in 1NF.
    2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
    1) Our relation is already in 2NF.
    2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
    1) Our relation is already in 3NF.
    2) For every functional dependency X → Y, X is the super key of the table.

Service_Information (Worker_id ,Worker_status ,Service_requests, Service_Date, Service_Time, **Service_id ,Customer_id)**

- Primary key : Worker_id, Service_id, Customer_id
- Foregine Key : Customer_id, Worker_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.

- Here, we have partial dependency between Worker_id and Worker_status, Service_requests because Worker_status,Service_requests are decided by Worker_id which is proper subset of

candidate key(PK) so for get rid of it we need to break down Service_Information table into two parts like

- 2NF
  - **Worker_Information2(<u>Worker_id</u>**, Worker_status, W_Service_requests)
  - **Service_Information(<u>Customer_id, Service_id, Worker_id,</u>** Service_Date, Serice_Time)
- 3NF : Our relation is already in 3NF because :
  1) Our relation is already in 2NF.
  2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
  1) Our relation is already in 3NF.
  2) For every functional dependency X → Y, X is the super key of the table.

Service_payment (<u>Customer_id ,Worker_id ,Service_id</u> , Cost_of_service , Payment_method **<u>,Transaction_id</u>)**

- Primary key : Worker_id, Service_id, Customer_id, Transaction_id
- Foregine Key : Worker_id, Service_id, Customer_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- Here, we have partial dependency between Service_id and Cost_of_service because Cost_of_service is decided by Service_id which is proper subset of candidate key(PK) so for get rid of it we need to break down Service_payment table into two parts like

- 2NF:
  - **Service_payment(<u>Customer_id, Worker_id, Service_id</u>**, Payment method, **<u>Transaction_id</u>**)
  - **Service_payment2(<u>Service_id</u>**, Cost_of_service)

- 3NF : Our relation is already in 3NF because :
  1) Our relation is already in 2NF.
  2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
  1) Our relation is already in 3NF.
  2) For every functional dependency X → Y, X is the super key of the table.

Application_for_payment (<u>Transaction_id</u> ,Paytm ,GooglePay ,PhonePe, Cash_on_service_Direct_cash **)**

- Primary key : Transaction_id
- Foregine Key : Transaction_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :

1) Our relation is already in 1NF.
2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.

- 3NF : Our relation is already in 3NF because :
  3) Our relation is already in 2NF.
  4) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
  3) Our relation is already in 3NF.
  4) For every functional dependency X → Y, X is the super key of the table.

Customer_Review (<u>Transaction_id</u> ,Customer_id ,Service_id , Worker_id , Feedback, Rating_out_of_10**)**

- Primary key : Transaction_id
- Foregine Key : Worker_id, Service_id, Customer_id, Transaction_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
  1) Our relation is already in 1NF.
  2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
  1) Our relation is already in 2NF.
  2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
  1) Our relation is already in 3NF.
  2) For every functional dependency X → Y, X is the super key of the table.

History (Customer_id ,Service_id ,Worker_id , Service_Date, Service_Time, **<u>Transaction_id</u>)**

- Primary key : Transaction_id
- Foregine Key : Worker_id, Service_id, Customer_id, Transaction_id
- 1NF : Our relation is already in 1NF and there are no multiple attributes.
- 2NF : Our relation is already in 2NF because :
  1) Our relation is already in 1NF.
  2) Non-prime attribute of the table is not dependent on proper subset of primary key , so there is no partial dependency.
- 3NF : Our relation is already in 3NF because :
  1) Our relation is already in 2NF.
  2) There is no transitive dependency exists in our relation.
- BCNF : Our relation is already in BCNF because :
  1) Our relation is already in 3NF.
  2) For every functional dependency X → Y, X is the super key of the table.

Final Relations with schema:

- **Customer** (**Customer_id**, Customer_first_name, Customer_last_name, Street, Customer_State , Country, PIN_Code, Customer_email_ID , Customer_Phone_number ,Customer_password)

- **Services** (**Service_id** , service_name ,Cost_of_service ,Extra_charges)

- **Service_Category** (**Service_id** ,Normal_service , Fast_service , Super_service)

- **Service_Provider_Workers** (**Service_id** , **Worker_id** , Worker_name , Worker_Phone_number ,Worker_ratings, Worker_vaccination_status)

- **Worker_salary** (**Worker_id** ,Salary ,Bonus ,**Total_Salary**)
- **Administration** (Customer_id ,**Service_id** ,Worker_id ,Profit ,Expanditure ,Total_Salary)

- **Recommendation** (**Customer_id** ,**Service_name**)

- **Cus_Ser_Info** (**Customer_id** ,Worker_id)

- **Service_Information** (**Worker_id** , Service_Date, Service_Time, **Service_id** ,**Customer_id**)

- **Worker_Information2**(**Worker_id**,Worker_status,W_Service_requests)

- **Service_payment** (**Customer_id** ,**Worker_id** ,**Service_id** , Payment_method ,**Transaction_id**)

- **Service_payment2** (**Service_id** , Cost_of_service)

- **Application_for_payment** (**Transaction_id** ,Paytm ,GooglePay ,PhonePe, Cash_on_service_Direct_cash)

- **Customer_Review** (**Transaction_id** ,Customer_id ,Service_id , Worker_id , Feedback, Rating_out_of_10)

- **History** (Customer_id ,Service_id ,Worker_id , Service_Date, Service_Time, **Transaction_id**)

Section 6
SQL

DDL Scripts

```
set search_path to cwsf_db

CREATE TABLE Customer
(
Customer_id int not NULL unique,
Customer_first_name varchar(30),
Customer_last_name varchar(30),
Street varchar(30),
PIN_Code int,
Customer_State varchar(30),
Customer_email_ID varchar(320),
Customer_Phone_number bigint not NULL Unique,
Customer_password varchar(16),
Customer_City varchar(16),
PRIMARY KEY (Customer_id)
)
```

```
CREATE TABLE Services
(
Service_id int not NULL unique,
service_name character varying(30),
Cost_of_service int not NULL,
Extra_charges int,
PRIMARY KEY (Service_id)
)
```

```
CREATE TABLE Service_Category
(
```

```
Service_id int not NULL,
Normal_service character varying(20),
Fast_service character varying(20),
Super_service character varying(20),
PRIMARY KEY(Service_id),
Foreign KEY (Service_id) REFERENCES Services(Service_id) ON
DELETE CASCADE ON UPDATE CASCADE
)
```

```
CREATE TABLE Service_Provider_Workers
(
Service_id int not NULL,
Worker_id int not NULL unique,
Worker_name character varying(15),
Worker_Phone_number int not NULL Unique,
Worker_ratings int,
Worker_vaccination_status character varying(15),
PRIMARY KEY(Worker_id,Service_id),
Foreign KEY (Service_id) REFERENCES Service_Category(Service_id)
ON UPDATE CASCADE)
```

```
CREATE TABLE Worker_salary
(
Worker_id int not NULL unique,
Salary int not NULL,
Bonus int default 0,
Total_Salary int not NULL,
PRIMARY KEY(Worker_id,Total_Salary),
Foreign KEY (Worker_id) REFERENCES
Service_Provider_Workers(Worker_id) ON DELETE CASCADE ON
```

```
UPDATE CASCADE
)
```

```
CREATE TABLE Administration
(
Customer_id int not NULL Unique,
Worker_id int not NULL Unique,
Profit int not NULL,
Expanditure int not NULL,
Total_Salary int not NULL,
PRIMARY KEY (Customer_id),
Foreign KEY (Customer_id) REFERENCES Customer(Customer_id)
ON DELETE CASCADE ON UPDATE CASCADE,
Foreign KEY (Worker_id,Total_Salary) REFERENCES
Worker_salary(Worker_id,Total_Salary) ON DELETE CASCADE ON
UPDATE CASCADE)
```

```
CREATE TABLE Recommendation
(
Customer_id int not NULL ,
Service_name character varying(30) not NULL,
PRIMARY KEY (Customer_id,Service_name),
Foreign KEY (Customer_id) REFERENCES Customer(Customer_id)
ON UPDATE CASCADE
)
```

```
CREATE TABLE Cus_Ser_Info
(
Customer_id int not NULL,
Worker_id int not NULL,
```

```
Primary key(Customer_id),
Foreign KEY (Customer_id) REFERENCES Customer(Customer_id)
ON UPDATE CASCADE,
Foreign KEY (Worker_id) REFERENCES
Service_Provider_Workers(Worker_id) ON UPDATE CASCADE
)
```

```
CREATE TABLE Service_Information
(
Worker_id int not NULL,
Service_Date date,
Service_Time time,
Service_id int not NULL,
Customer_id int not NULL,
PRIMARY KEY(Worker_id,Service_id,Customer_id),
Foreign KEY (Worker_id,Service_id) REFERENCES
Service_Provider_Workers(Worker_id,Service_id) ON
UPDATE CASCADE,
Foreign KEY (Customer_id) REFERENCES Cus_Ser_Info(Customer_id)
 ON UPDATE CASCADE
)
```

```
CREATE TABLE Worker_Information2
(
Worker_id int not null,
Worker_status character varying(10),
W_Service_requests int,
PRIMARY KEY(Worker_id),
Foreign KEY (Worker_id) REFERENCES
Service_Provider_Workers(Worker_id) ON DELETE CASCADE ON
```

```
UPDATE CASCADE
)
```

```
CREATE TABLE Service_payment
(
Customer_id int not NULL,
Worker_id int not NULL,
Service_id int not NULL,
Payment_method character varying(20),
Transaction_id int not NULL unique,
PRIMARY KEY(Transaction_id,Customer_id,Worker_id,Service_id),
Foreign KEY (Worker_id,Service_id,Customer_id) REFERENCES
Service_Information(Worker_id,Service_id,Customer_id) ON UPDATE
CASCADE)
```

```
CREATE TABLE Service_payment2
(
Service_id int not null,
Cost_of_service int not null,
PRIMARY KEY(Service_id),
Foreign KEY (Service_id) REFERENCES Service_Category(Service_id)
ON UPDATE CASCADE
)
```

```
CREATE TABLE Application_for_payment
(
Transaction_id int not NULL unique,
Paytm character varying(20),
GooglePay character varying(20),
```

```
PhonePe character varying(20),
Cash_on_service_Direct_cash character varying(20),
PRIMARY KEY(Transaction_id),
Foreign KEY (Transaction_id) REFERENCES
Service_Payment(Transaction_id) ON UPDATE CASCADE
)
```

```
CREATE TABLE Customer_Review
(
Customer_id int not NULL,
Service_id int not NULL,
Worker_id int not NULL,
Transaction_id int not NULL unique,
Feedback character varying(100),
Rating_out_of_10 int,
Primary key(Transaction_id),
Foreign KEY (Transaction_id,Customer_id,Worker_id,Service_id)
REFERENCES
Service_payment(Transaction_id,Customer_id,Worker_id,Service_id)
ON UPDATE CASCADE
)
```

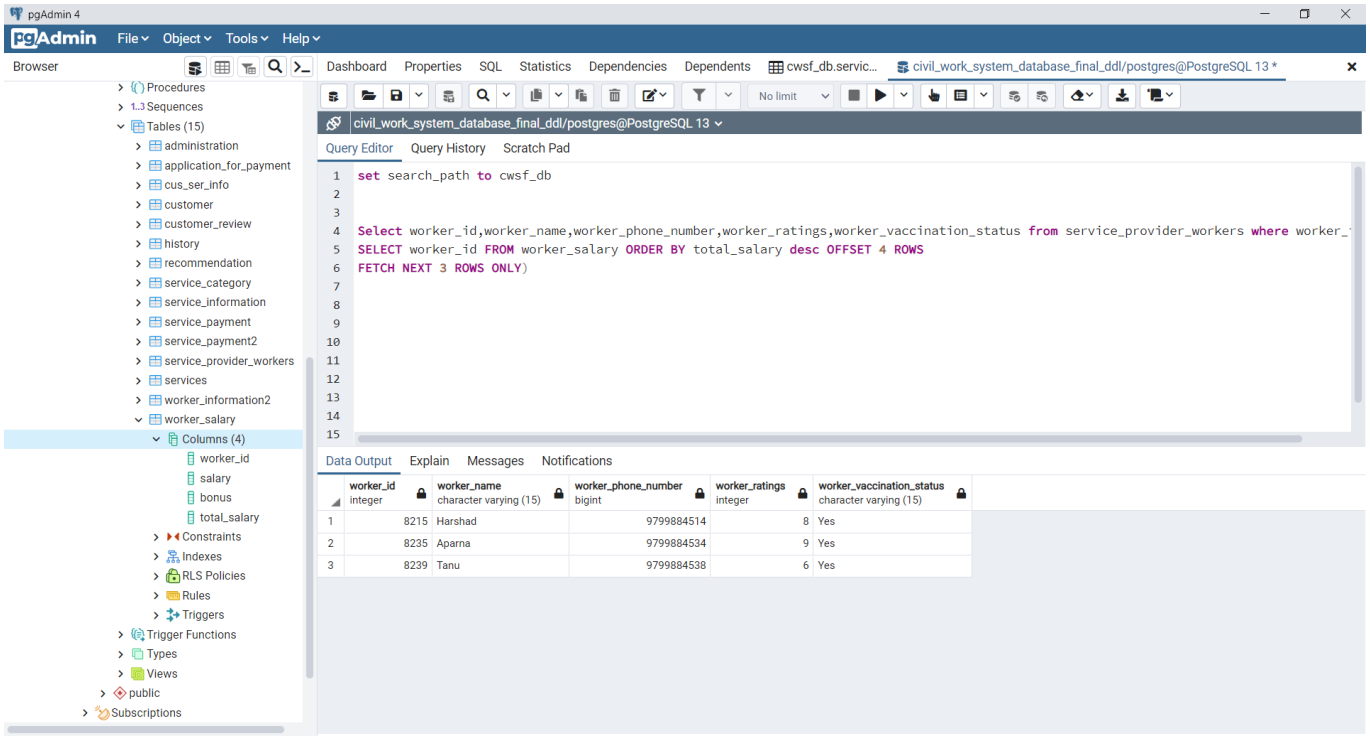
```
CREATE TABLE History
(
Customer_id int not NULL,
Service_id int not NULL,
Worker_id int not NULL,
Service_Date character varying(20),
Service_Time character varying(20),
Transaction_id int not NULL unique,
```

```
Primary key(Transaction_id),
Foreign KEY (Transaction_id,Customer_id,Worker_id,Service_id)
REFERENCES
Service_payment(Transaction_id,Customer_id,Worker_id,Service_id)
ON UPDATE CASCADE)
```

SQL Queries

1. **Find all worker details of 5th, 6th & 7th highest worker salary.**

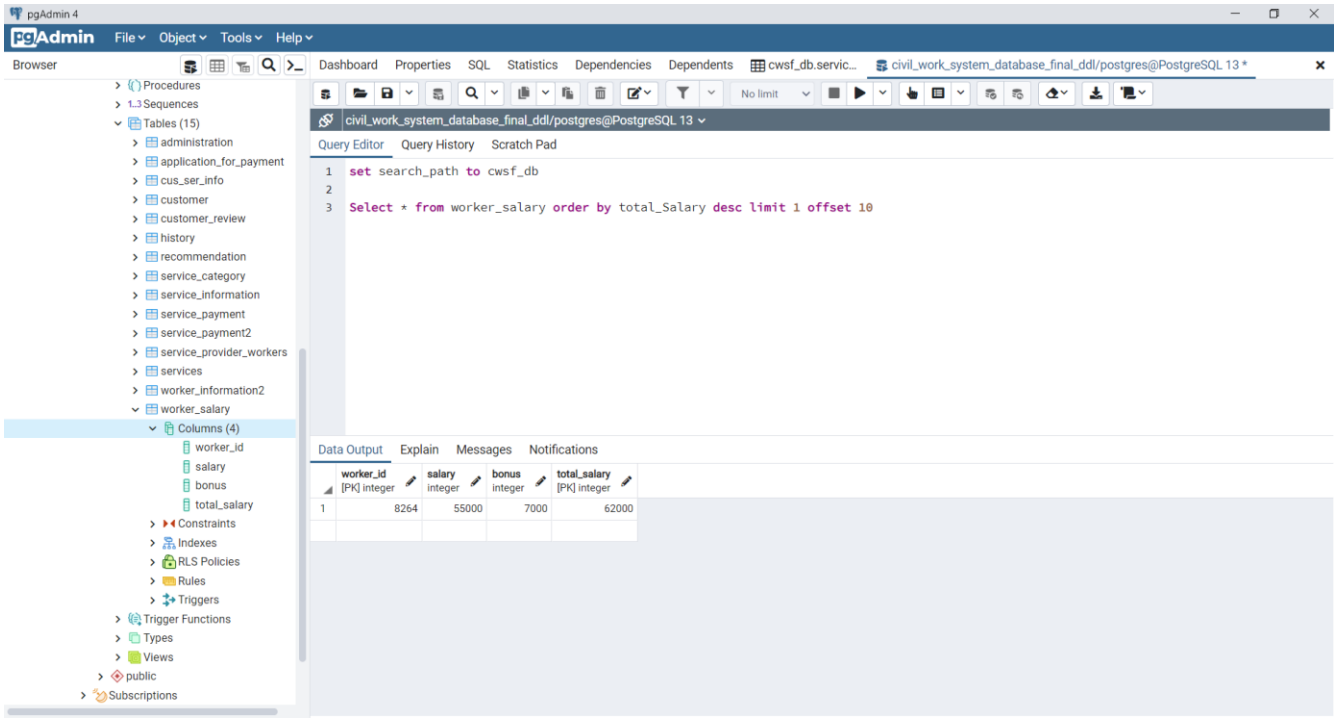
   - Select worker_id,worker_name,worker_phone_number,worker_ratings,worker_vaccination_status from service_provider_workers where worker_id in (SELECT worker_id FROM worker_salary ORDER BY total_salary desc OFFSET 4 ROWS FETCH NEXT 3 ROWS ONLY)



   - Number of tuples – 3

2. **Find Nth highest salary from worker salary table. As example we take 11th highest in the worker salary.**

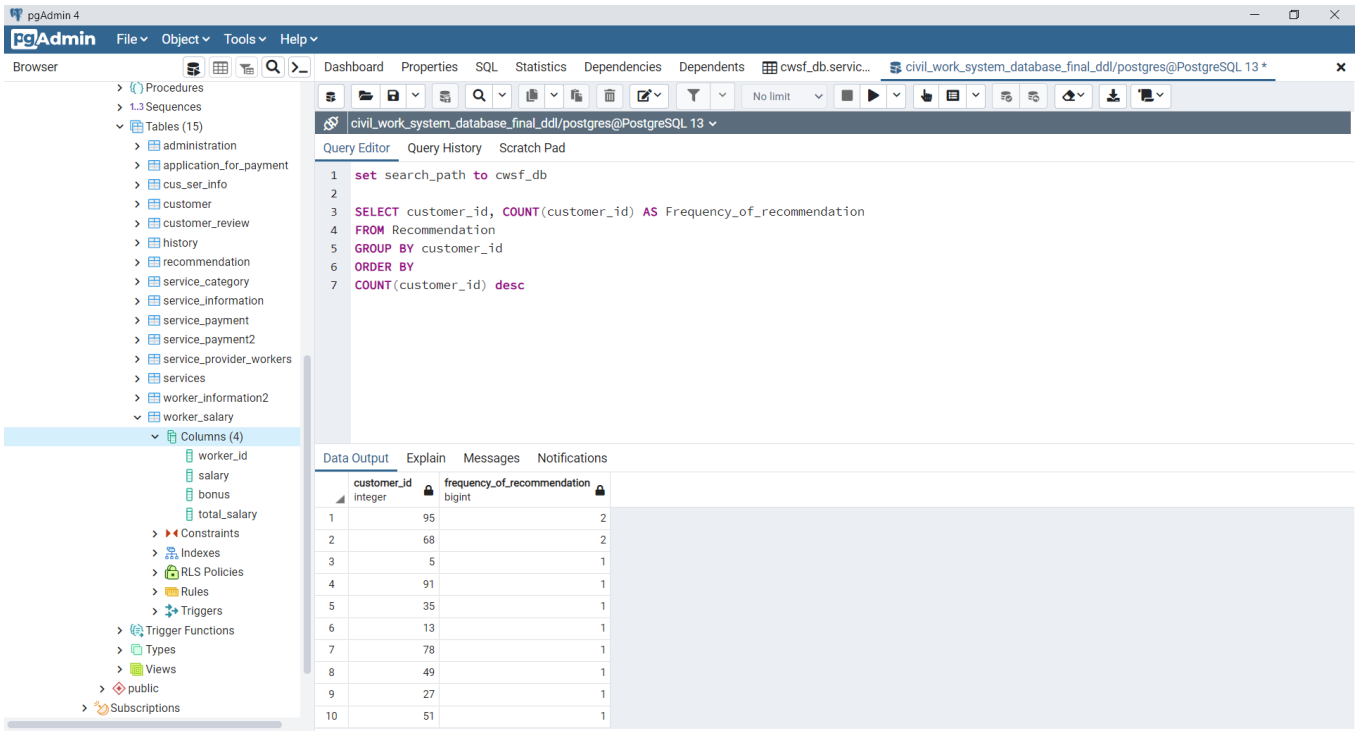
   - Select * from worker_salary order by total_Salary desc limit 1 offset 10



- Number of tuples – 1

3. Find frequency of customers for recommendation and print in decreasing order.

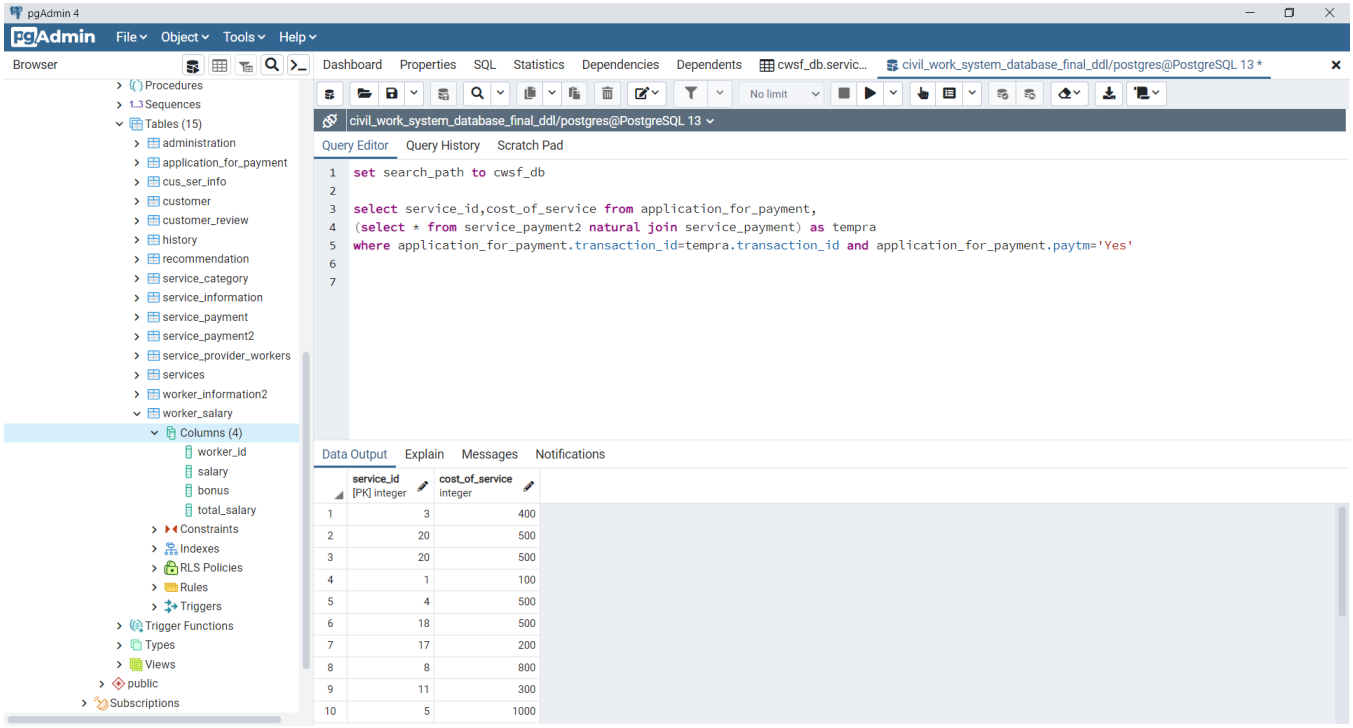
- SELECT customer_id, COUNT(customer_id) AS Frequency_of_recommendation FROM Recommendation GROUP BY customer_id ORDER BY COUNT(customer_id) desc



- Number of tuples – 10

4. **Find the service_id and cost of service which was paid through the payment method "Paytm".**

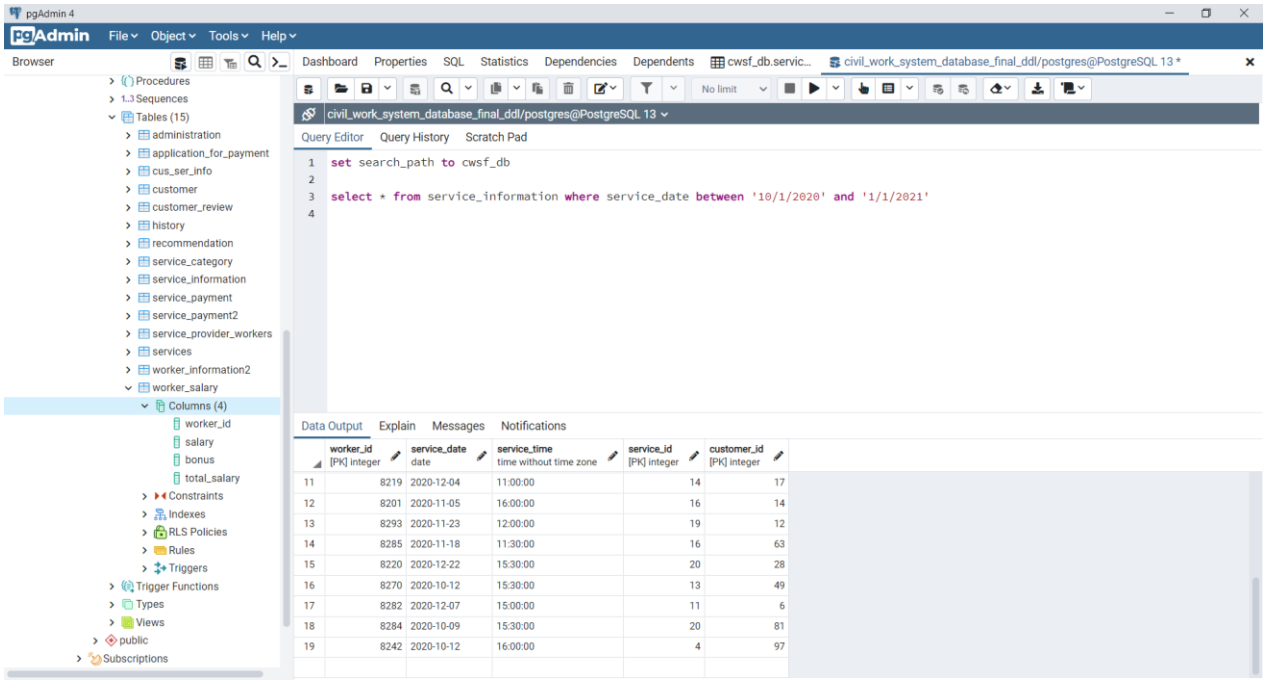
- select service_id,cost_of_service from application_for_payment,(select * from service_payment2 natural join service_payment) as tempra where application_for_payment.transaction_id=tempra.transaction_id and application_for_payment.paytm='Yes'



- Number of tuples – 10

5. **Find the number of services happened between any two dates.**

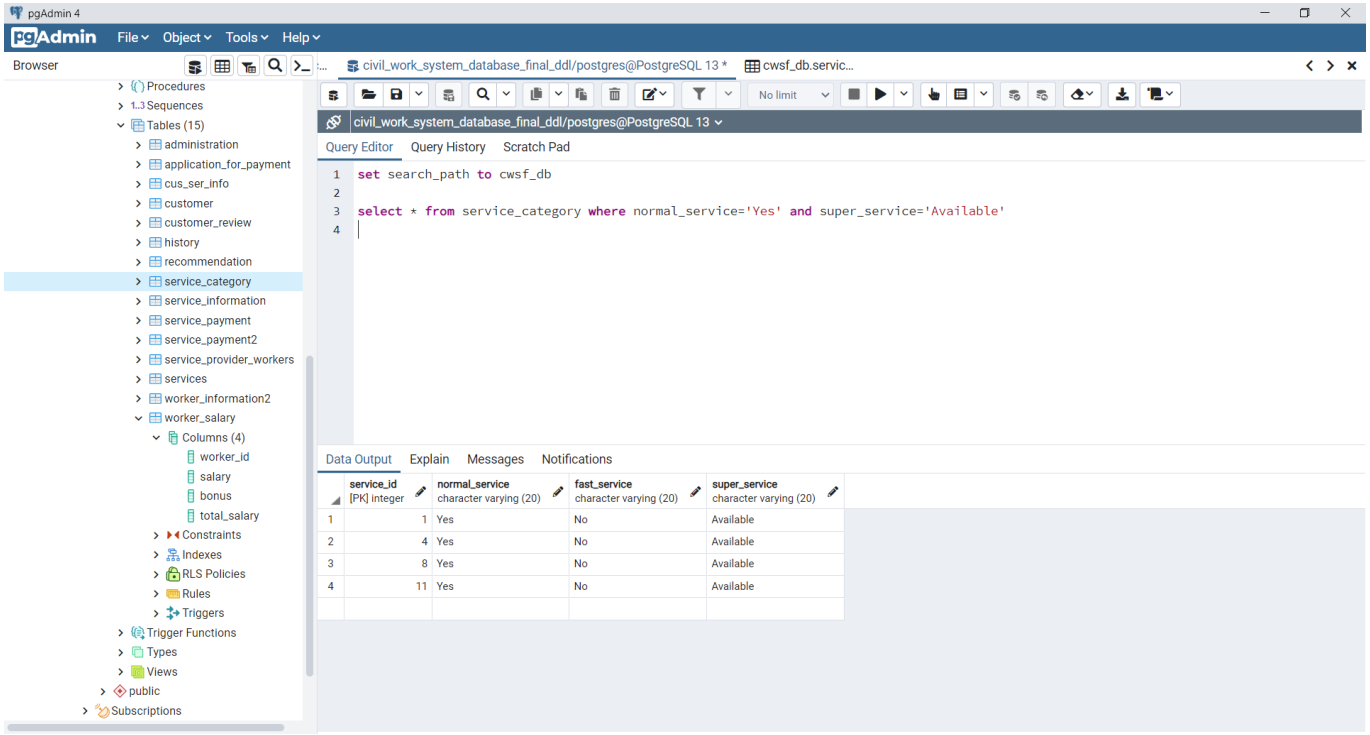
- select * from service_information where service_date between '10/1/2020' and '1/1/2021'



- Number of tuples – 19

6. Find the service type which is “normal” and “super” at the same time.

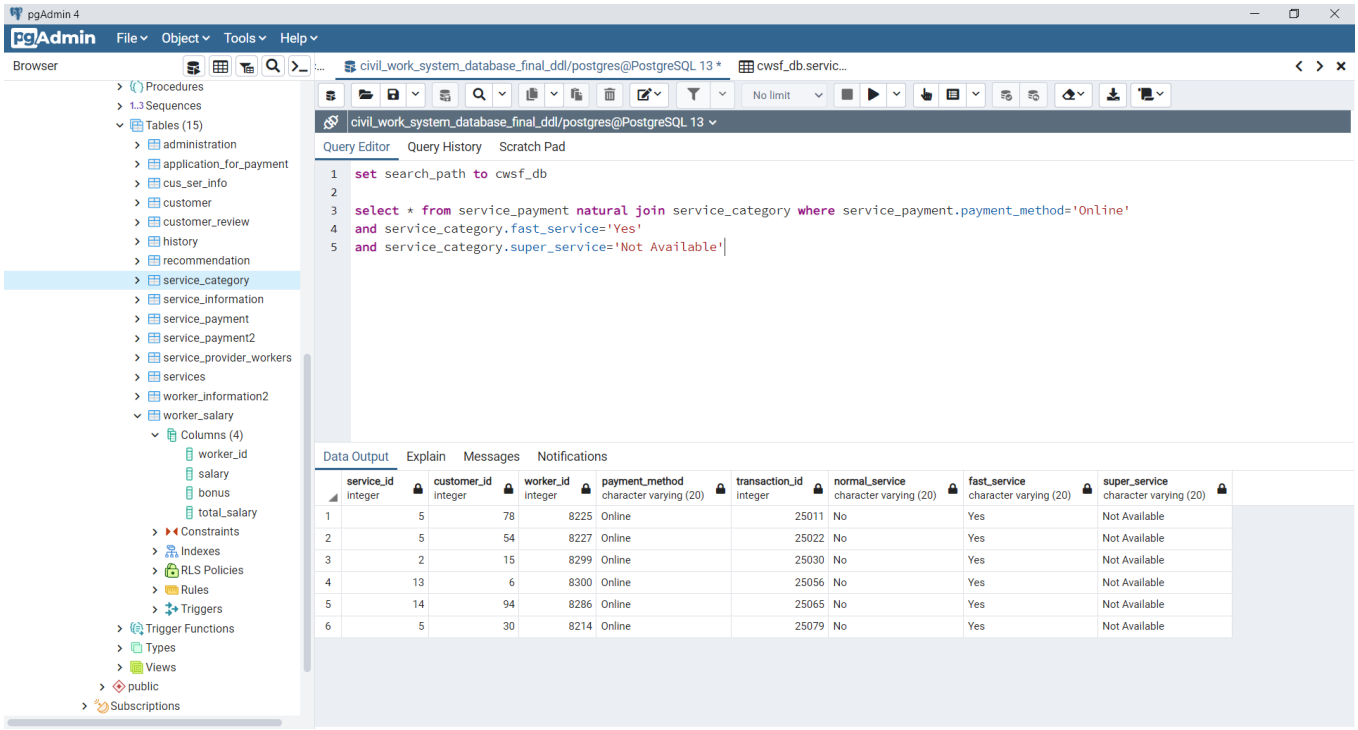
- select * from service_category where normal_service='Yes' and super_service='Available'



- Number of tuples – 4

7. **Display all the details of the service which is “Fast” type as well as not a super service at the same time.**

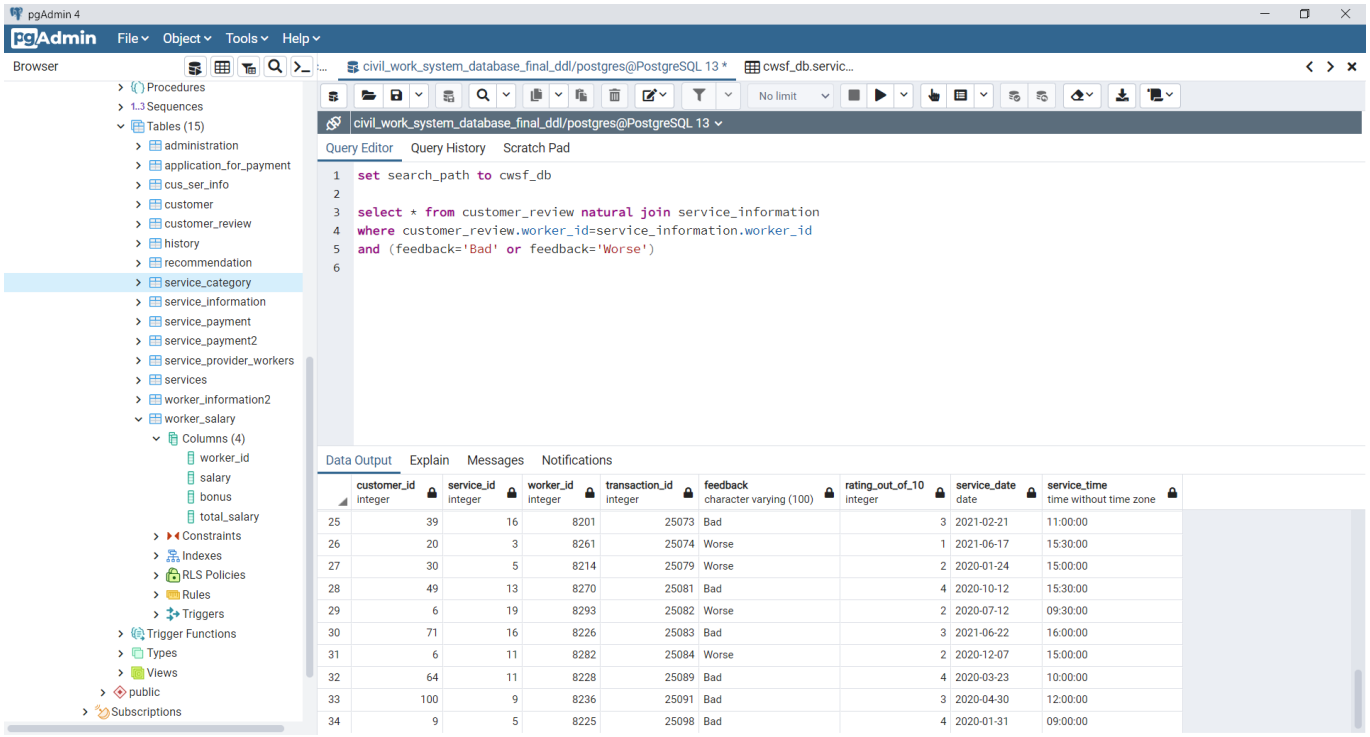
- select * from service_payment natural join service_category where service_payment.payment_method='Online' and service_category.fast_service='Yes' and service_category.super_service='Not Available'



- Number of tuples – 6

8. **Display the services which had been given bad reviews by customers. Also, display every other information of that particular service.**

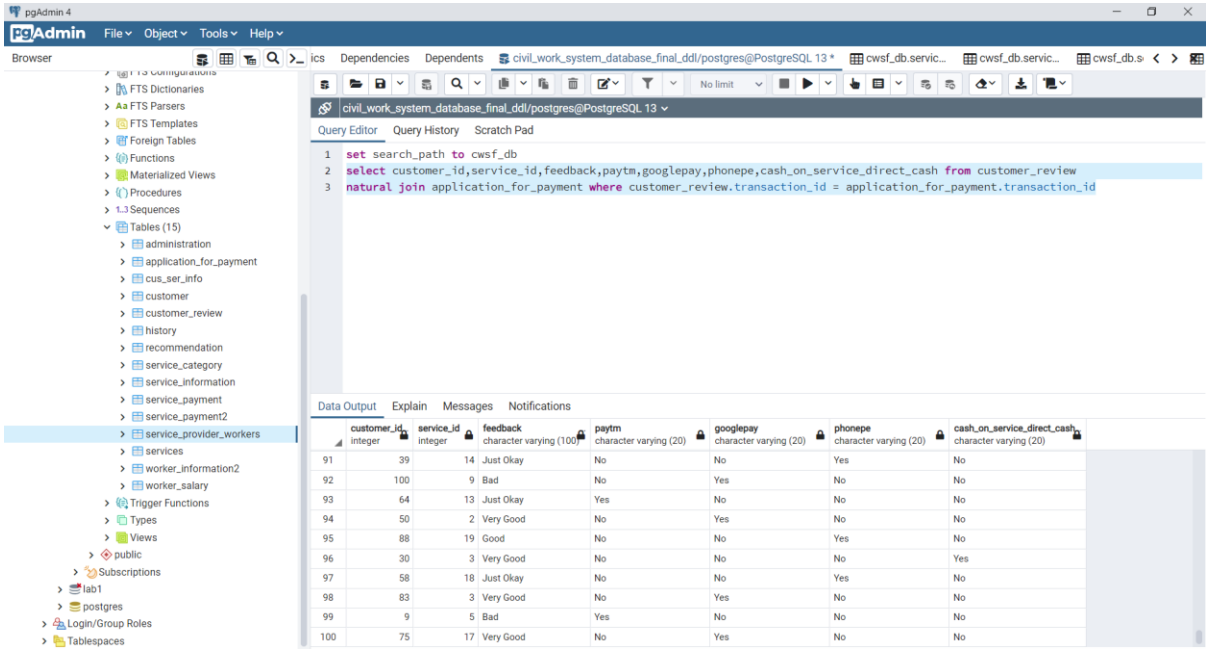
- select * from customer_review natural join service_information where customer_review.worker_id=service_information.worker_id and (feedback='Bad' or feedback='Worse')



- Number of tuples - 34

9. Find the customers with their service's payment methods.Also show the feedback of the customer

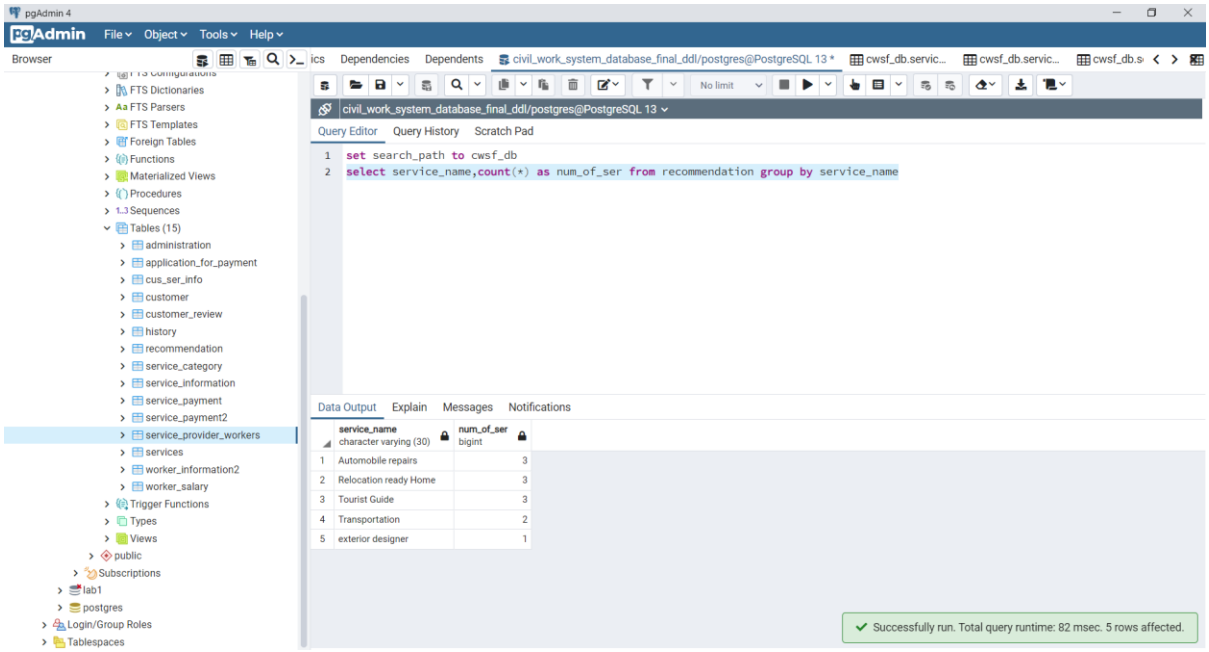
- Select customer_id,service_id,feedback,paytm,googlepay,phonepe, cash_on_service_direct_cash from customer_review natural join application_for_payment where customer_review.transaction_id = application_for_payment.transaction_id



- Number of tuples – 100

10. Find total number of duplicate recommened services by their names.

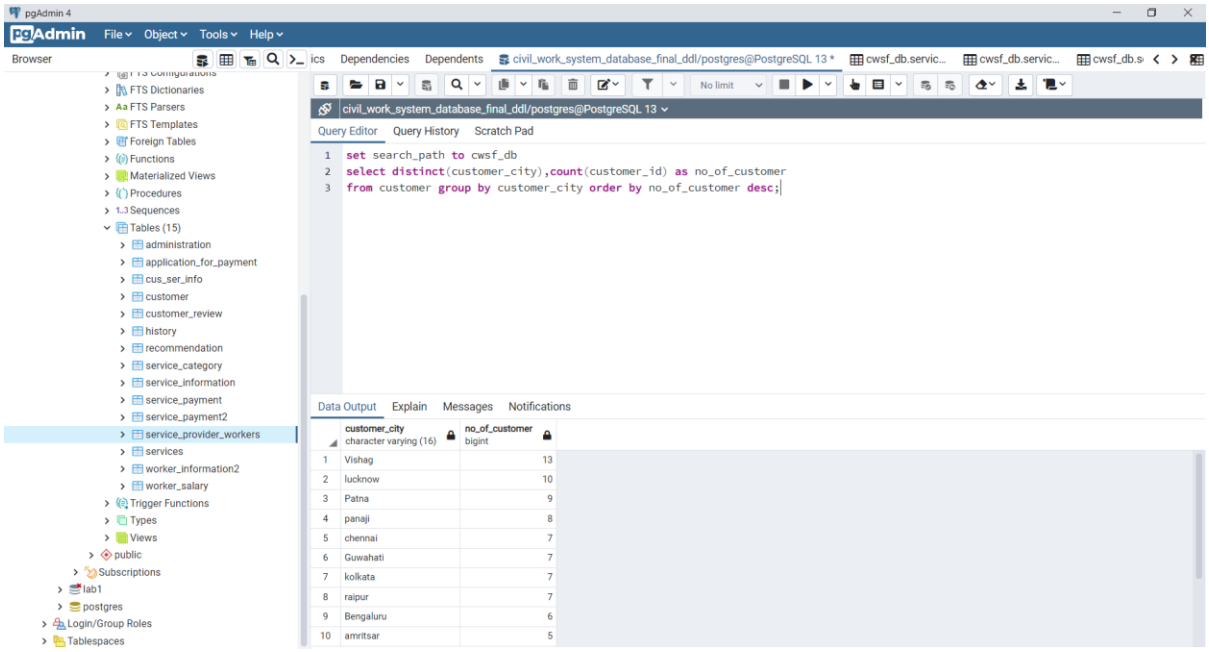
- select service_name,count(*) as num_of_ser from recommendation group by service_name



- Number of tuples – 5

11. Display number of customers by for each distinct city.

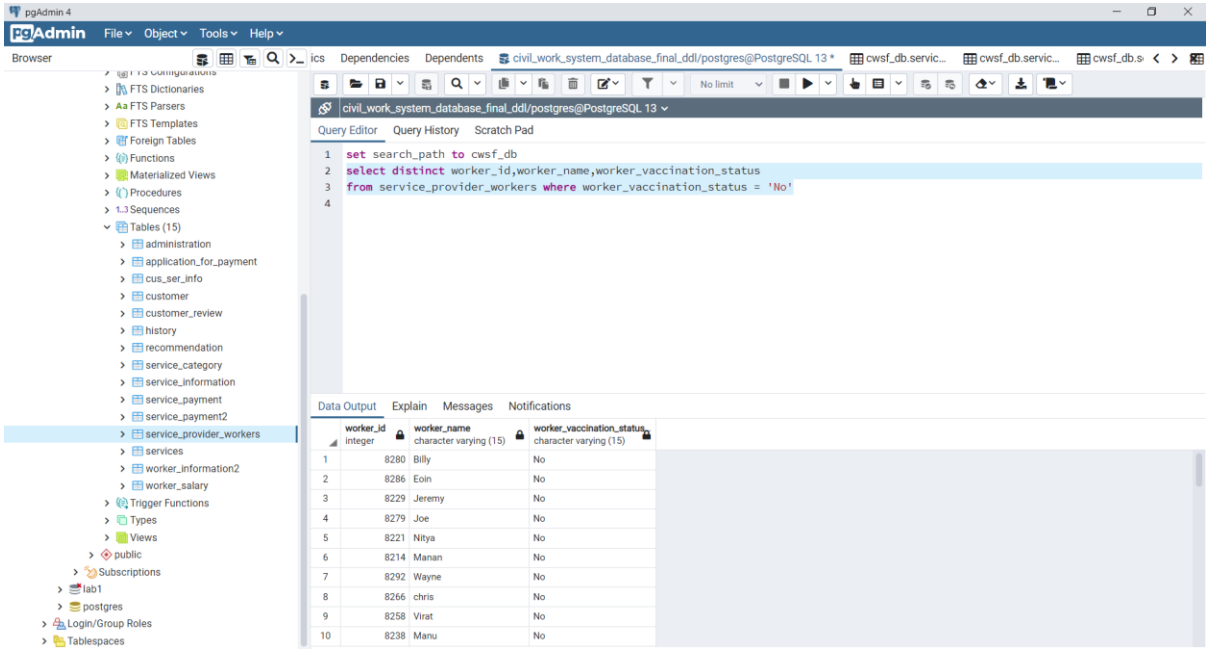
- select distinct(customer_city),count(customer_id) as no_of_customer from customer group by customer_city order by no_of_customer desc;



- Number of tuples - 10

12. Find worker_id,worker_name and worker_vaccination_status for those worker who is not vaccinated.

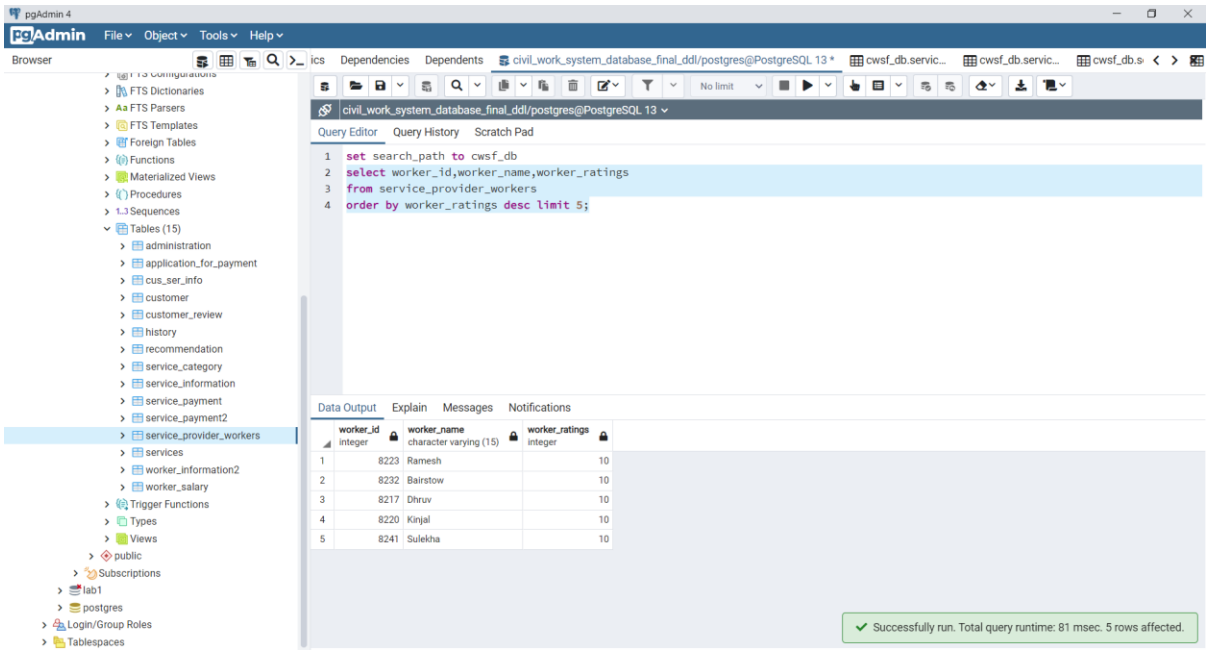
- select distinct worker_id,worker_name,worker_vaccination_status from service_provider_workers where worker_vaccination_status = 'No'



- Number of tuples - 10

13. Display top 5 worker_id ,worker_name and worker_ratings by their ratings.

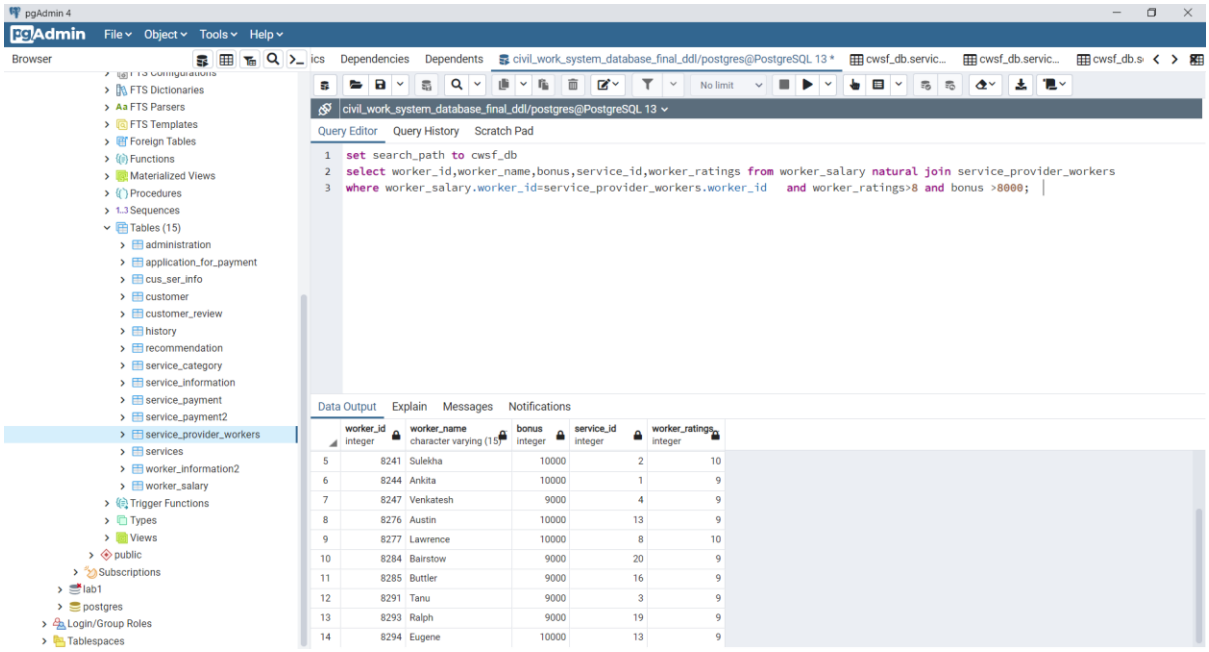
- select worker_id,worker_name,worker_ratings from service_provider_workers order by worker_ratings desc limit 5;



- Number of tuples - 5

14. Find the worker's information who gets more ratings as well as bonus greater than 8000.

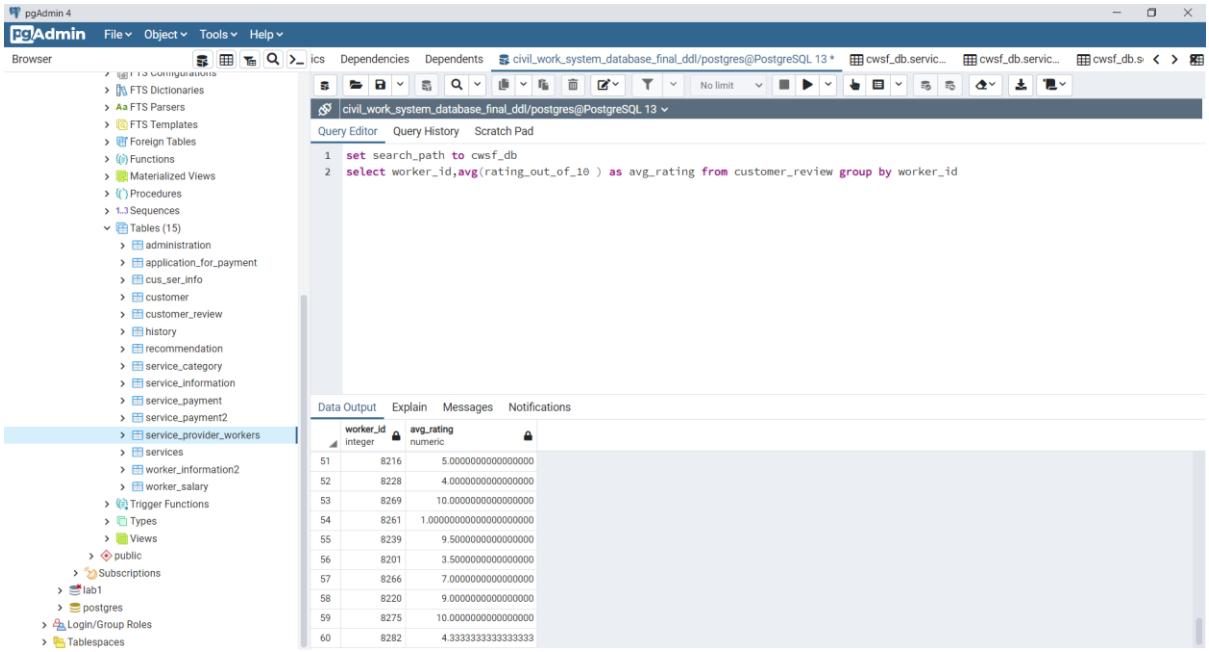
- select worker_id,worker_name,bonus,service_id,worker_ratings from worker_salary natural join service_provider_workers where worker_salary.worker_id=service_provider_workers.worker_id and worker_ratings>8 and bonus >8000;



- Number of tuples - 14

15. Find average rating given by customer for each worker.

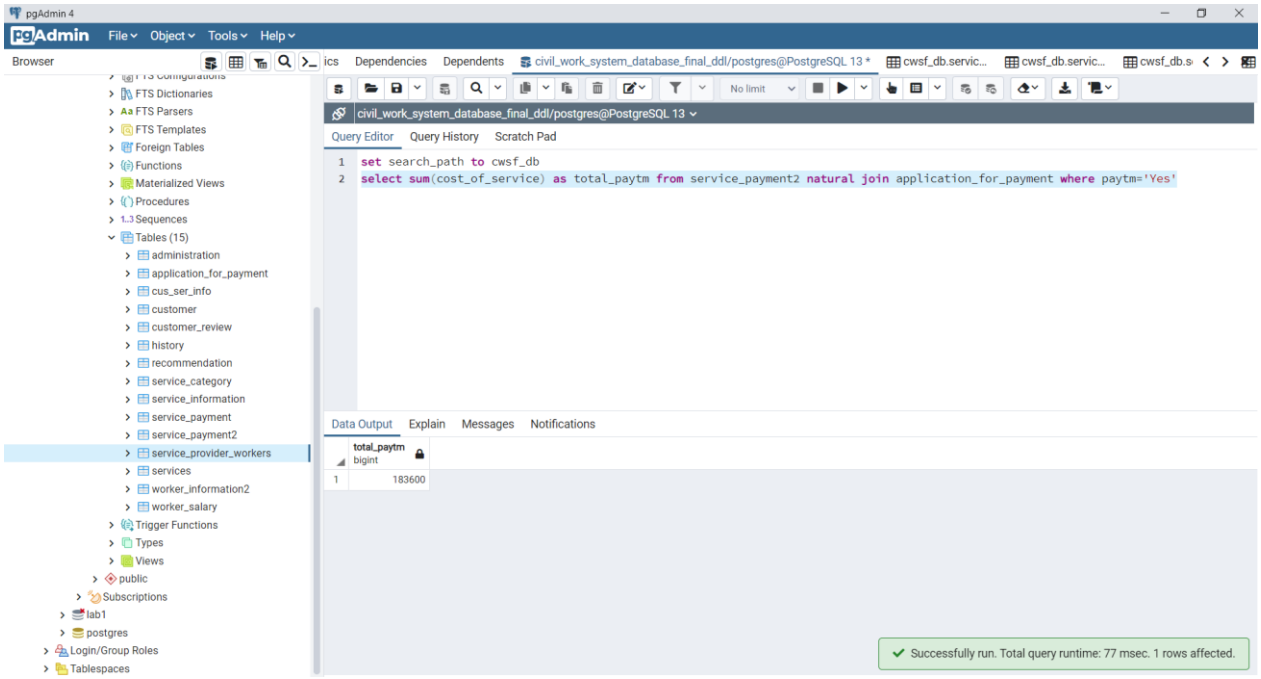
- select worker_id,avg(rating_out_of_10) as avg_rating from customer_review group by worker_id



- Number of tuples – 60

16. Find total payment we got through payment method=’Paytm’.

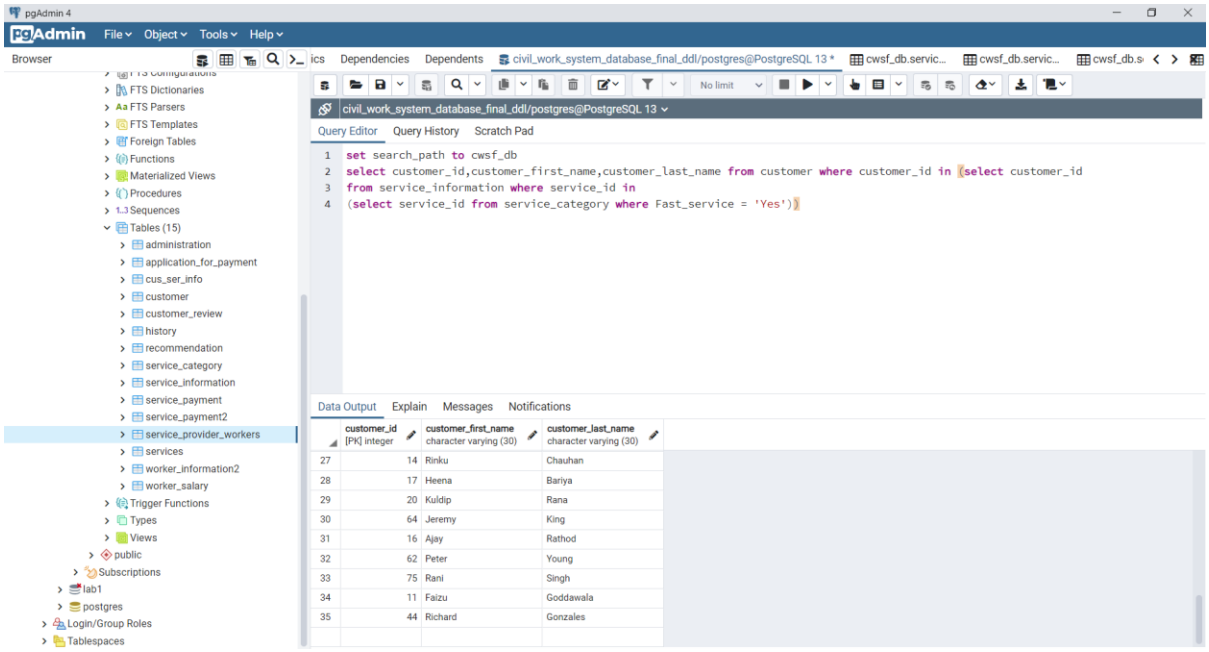
- select sum(cost_of_service) as total_paytm from service_payment2 natural join application_for_payment where paytm='Yes'



- Number of tuples – 1

17. Find the customers who take Fast services in past and show the customer_id and customer full name

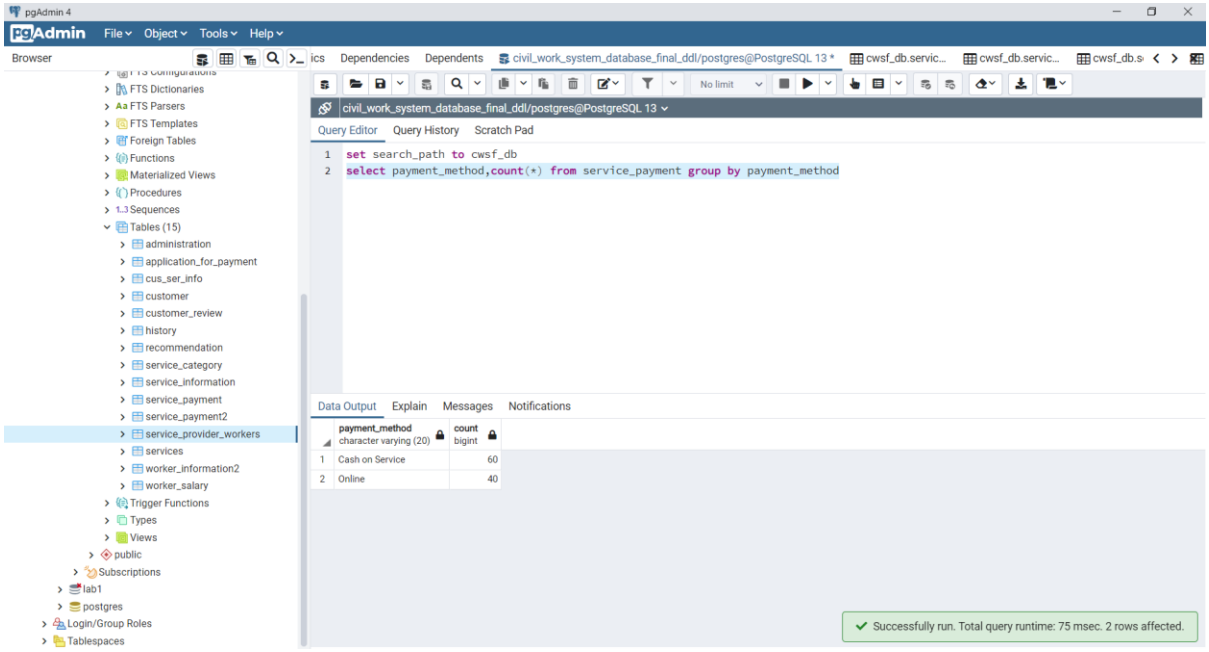
select customer_id,customer_first_name,customer_last_name from customer where customer_id in (select customer_id from service_information where service_id in (select service_id from service_category where Fast_service = 'Yes'))



- Number of tuples - 35

18. Find the number of customers for each payment method.

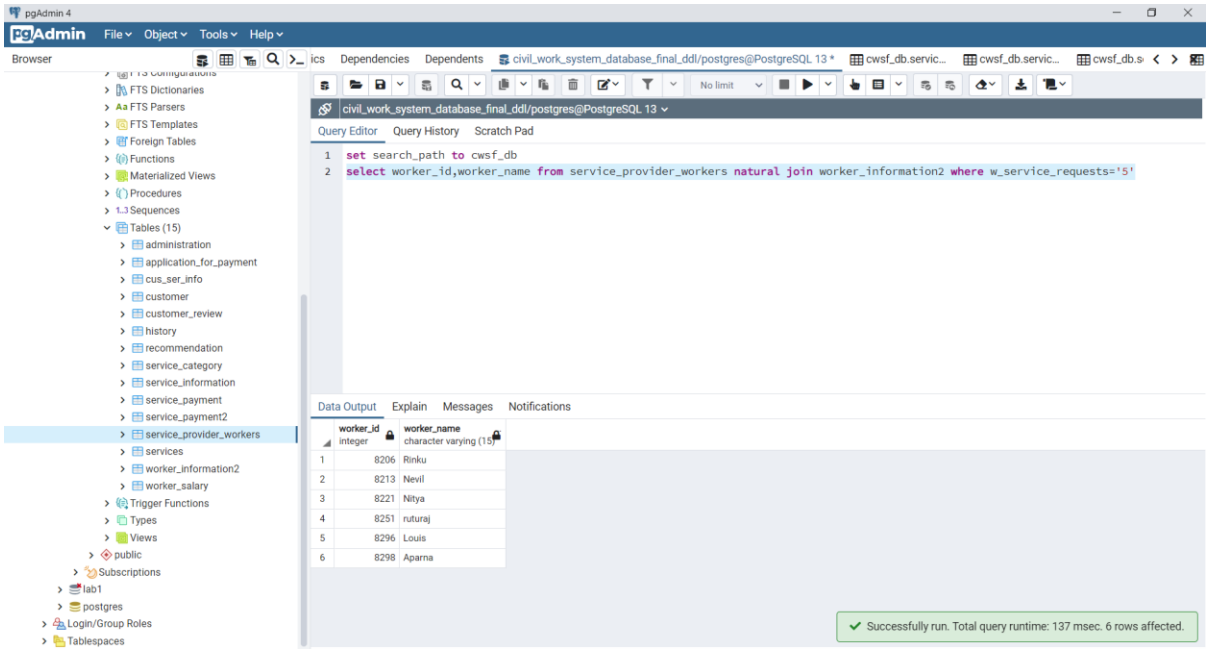
- select payment_method,count(*) from service_payment group by payment_method



- Number of tuples – 2

19. Find worker's name and id who is getting maximum service requests.

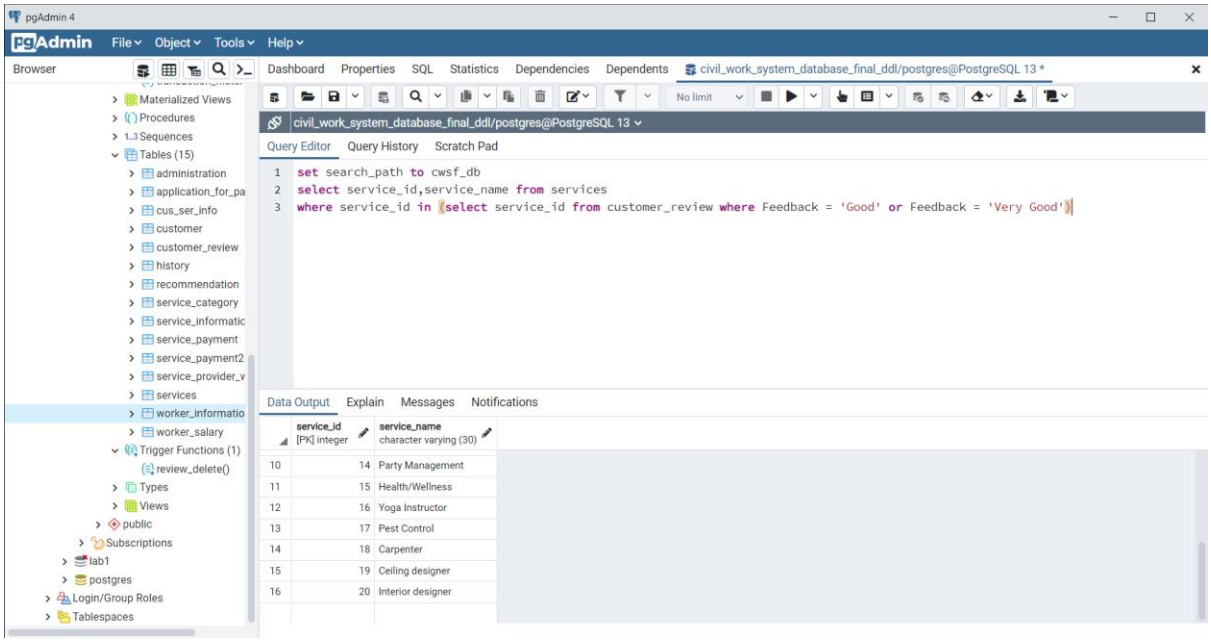
- select worker_id,worker_name from service_provider_workers natural join worker_information2 where w_service_requests='5'



- Number of tuples – 6

20. Find the services which has feedback ‘Good’ or ‘Very Good’.

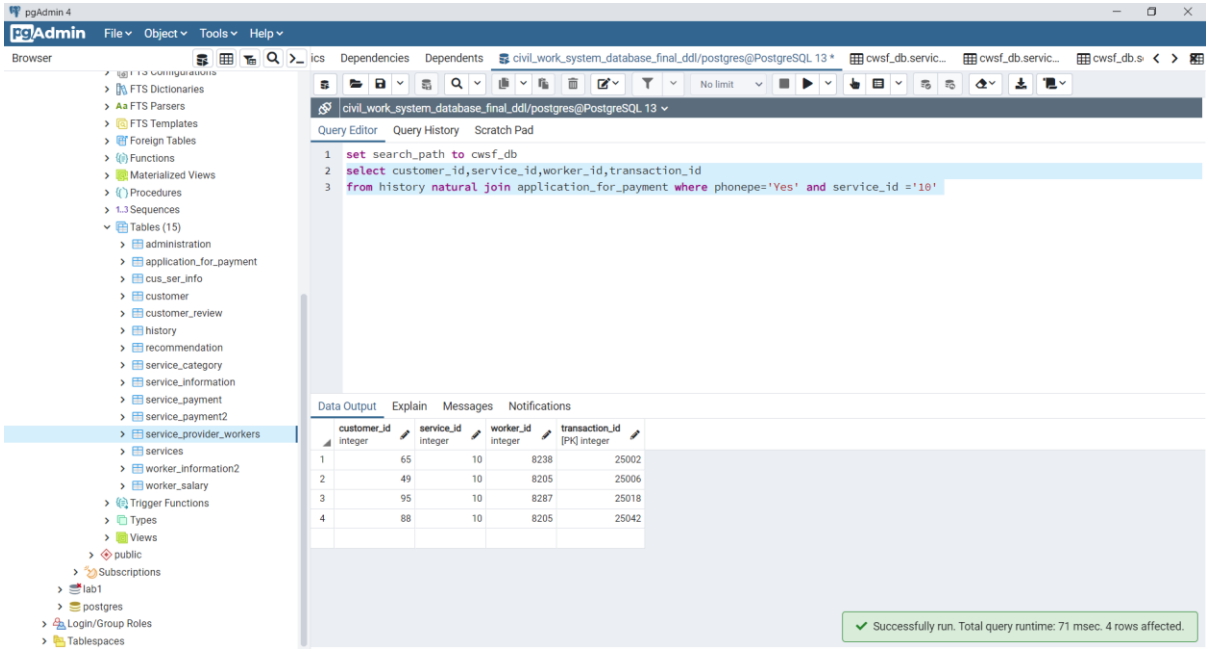
- select service_id,service_name from services where service_id in (select service_id from customer_review where Feedback = 'Good' or Feedback = 'Very Good')



- Number of tuples – 16

21. Display customer_id, service_id, worker_id and transaction_id details for service_id=’10’ and payment method is ‘phonepe’.

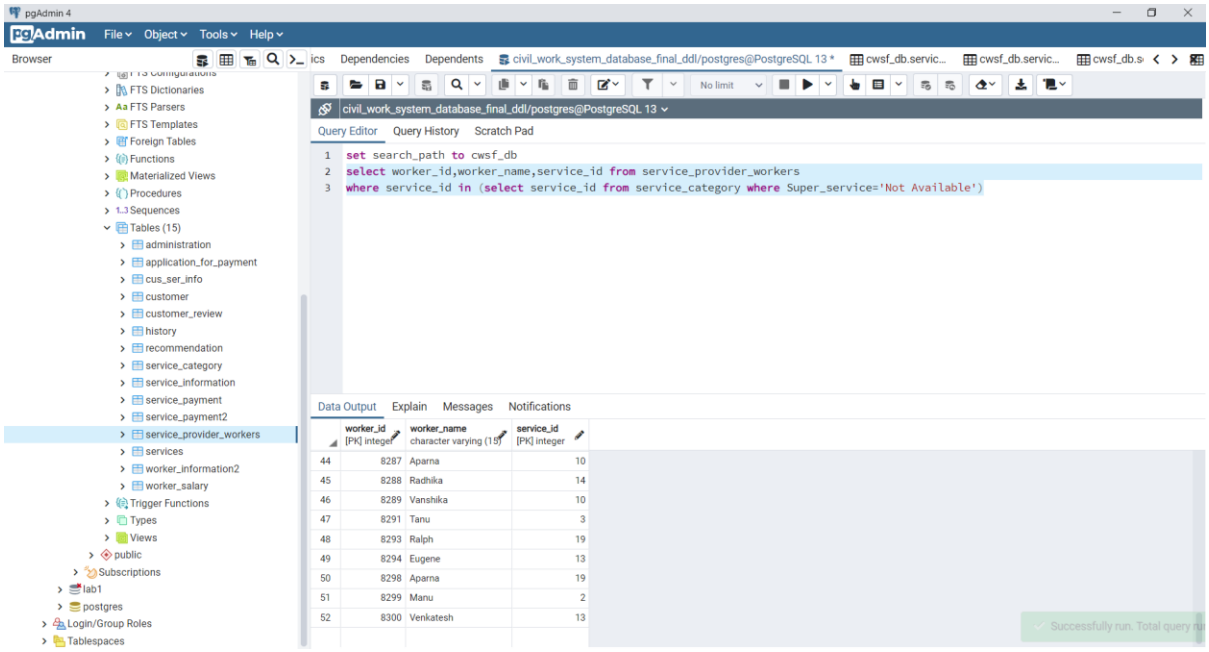
- select customer_id,service_id,worker_id,transaction_id from history natural join application_for_payment where phonepe='Yes' and service_id ='10'



- Number of tuples – 4

22. Find the workers who doesn't participate in super services with the service_id ,worker_id and worker_name.

- select worker_id,worker_name,service_id from service_provider_workers where service_id in (select service_id from service_category where Super_service='Not Available')

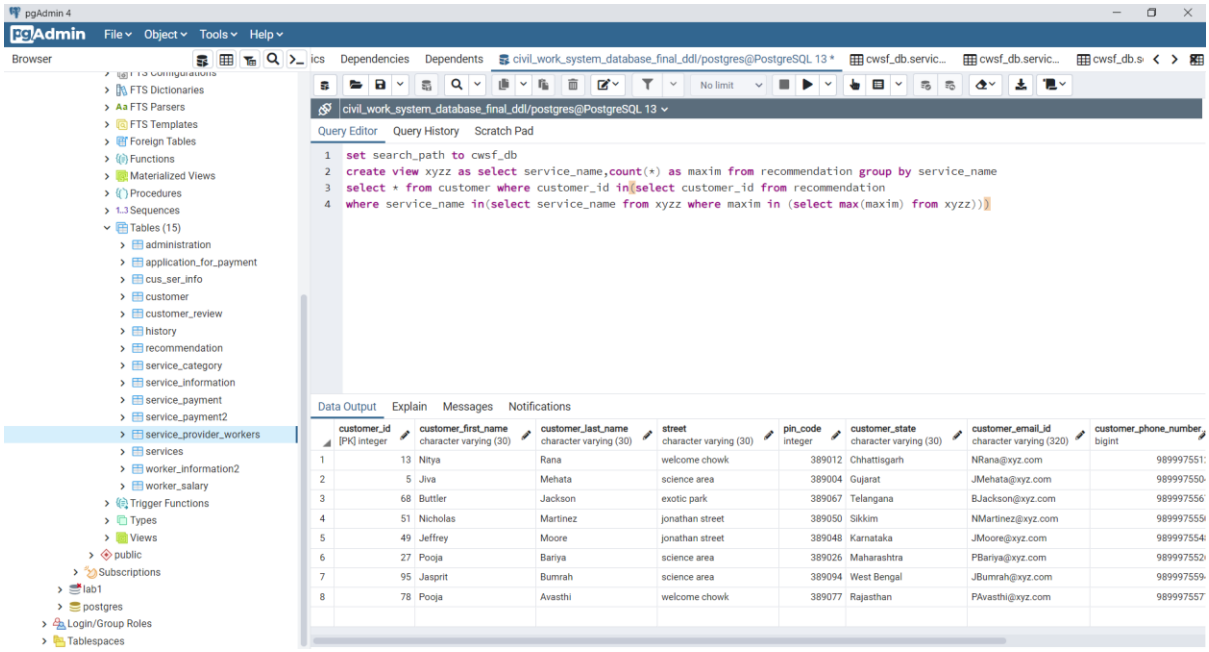


- Number of tuples - 52

23. Find all customer information for service_name who is more recommended by customer.

- create view xyzz as select service_name,count(*) as maxim from recommendation group by service_name

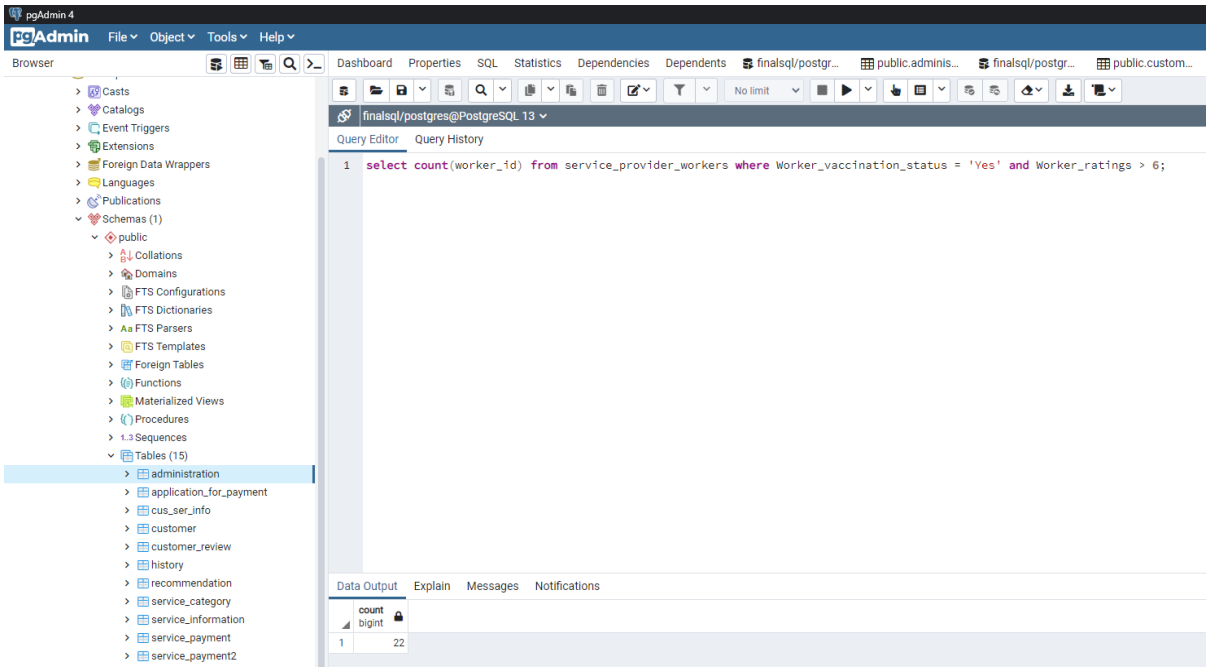
- select * from customer where customer_id in(select customer_id from recommendation where service_name in(select service_name from xyzz where maxim in (select max(maxim) from xyzz)))



- Number of tuples – 8

24. What is the total number of efficient workers (the workers which have completed vaccination and also are rated > 6) ?

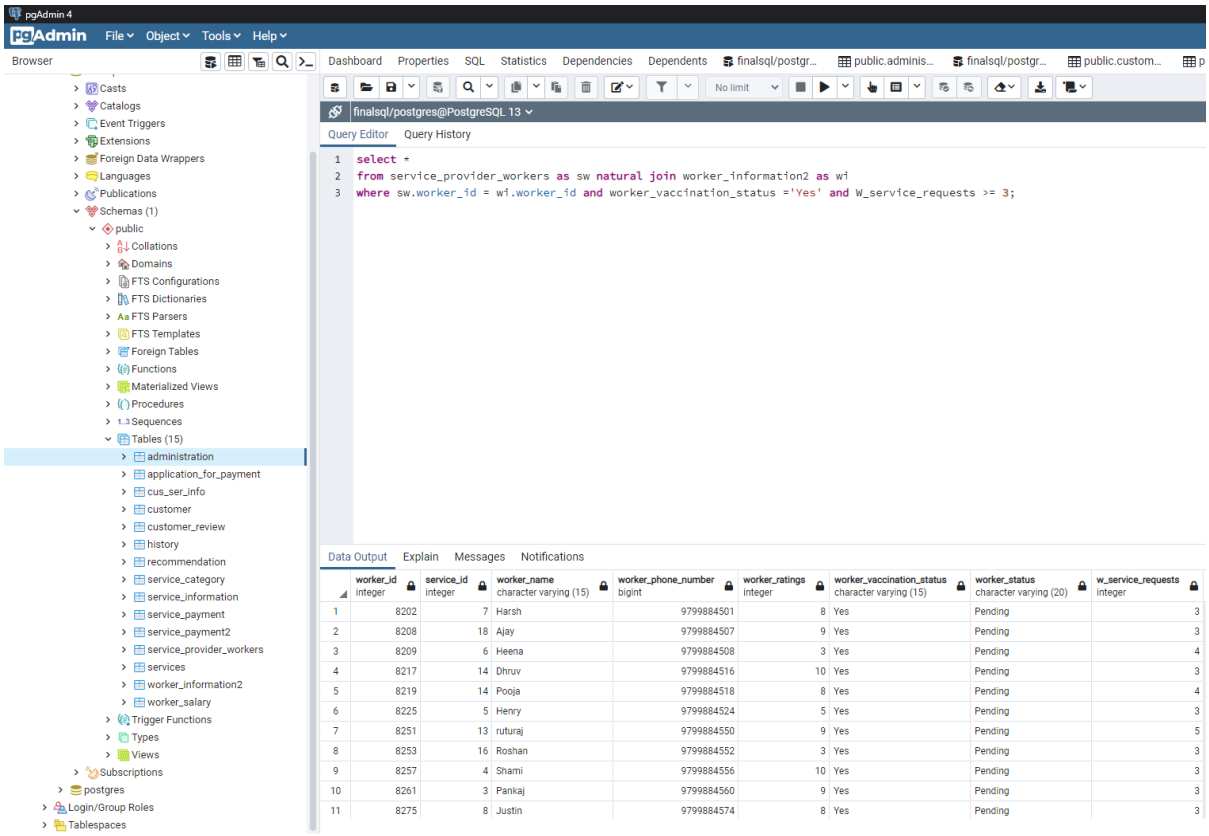
- select count(worker_id) from service_provider_workers where Worker_vaccination_status = 'Yes' and Worker_ratings > 6;



- Number of tuples – 1

25. Most demanding and efficient worker(which have request pending >=3 and are vaccinated)?

- select *
  from service_provider_workers as sw natural join worker_information2 as wi
  where sw.worker_id = wi.worker_id and worker_vaccination_status ='Yes' and W_service_requests >= 3;

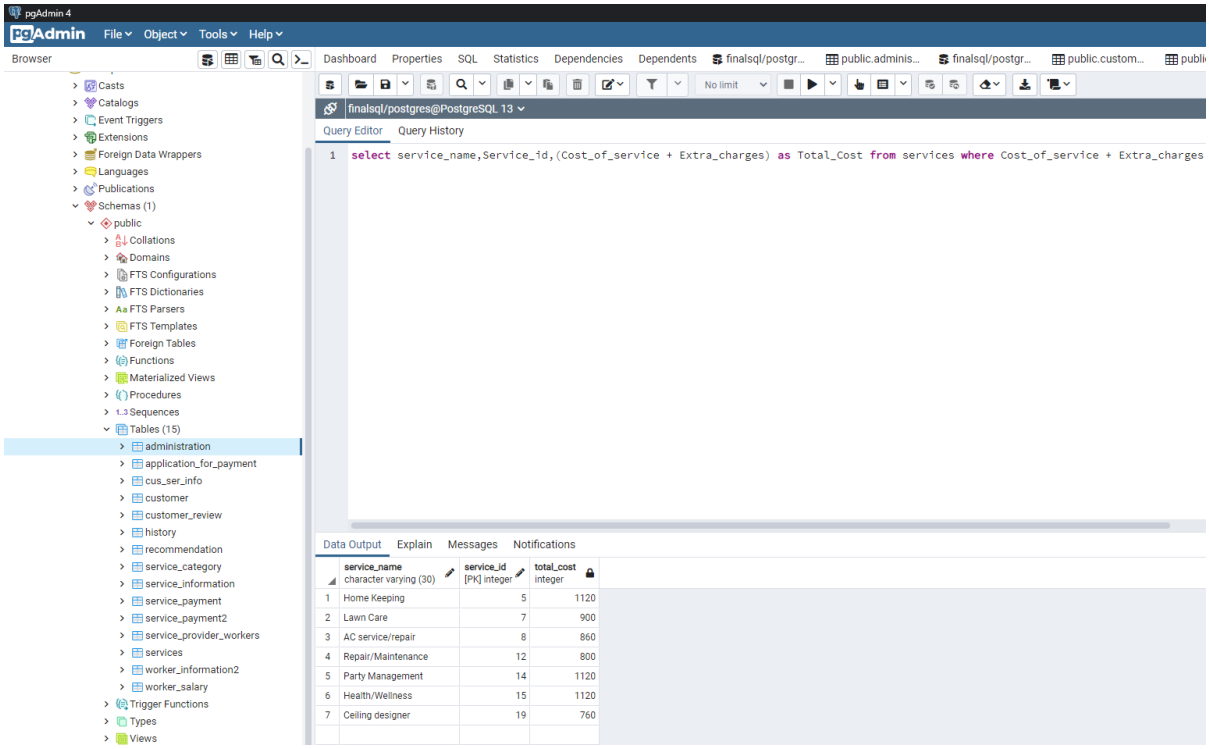


	worker_id integer	service_id integer	worker_name character varying (15)	worker_phone_number bigint	worker_ratings integer	worker_vaccination_status character varying (15)	worker_status character varying (20)	w_service_requests integer
1	8202	7	Harsh	9799884501	8	Yes	Pending	3
2	8208	18	Ajay	9799884507	9	Yes	Pending	3
3	8209	6	Heena	9799884508	3	Yes	Pending	4
4	8217	14	Dhruv	9799884516	10	Yes	Pending	3
5	8219	14	Pooja	9799884518	8	Yes	Pending	4
6	8225	5	Henry	9799884524	5	Yes	Pending	3
7	8251	13	ruturaj	9799884550	9	Yes	Pending	5
8	8253	16	Roshan	9799884552	3	Yes	Pending	3
9	8257	4	Shami	9799884556	10	Yes	Pending	3
10	8261	3	Pankaj	9799884560	9	Yes	Pending	3
11	8275	8	Justin	9799884574	8	Yes	Pending	3

- Number of tuples – 15

26. The service which costs the most out of all service and whose cost are greater than normal costs(700) ?

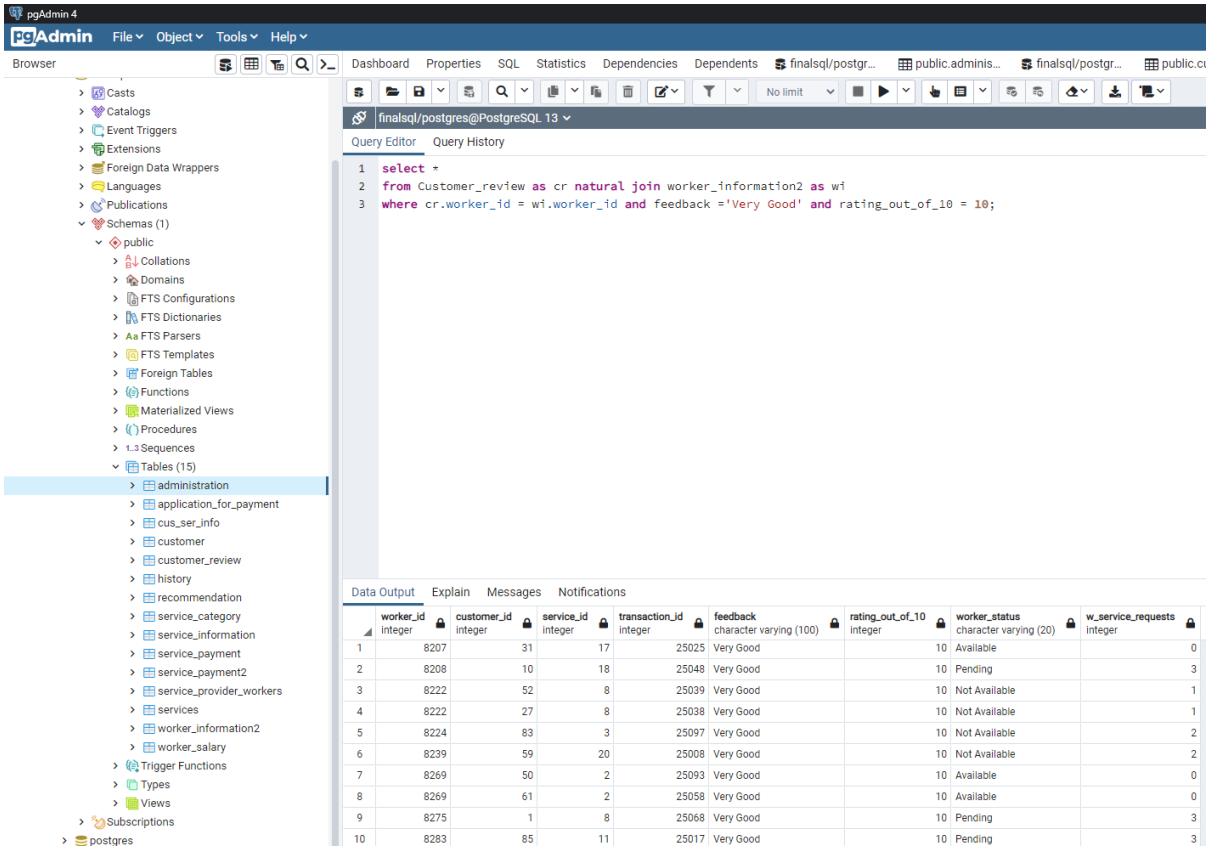
- select service_name,Service_id,(Cost_of_service + Extra_charges) as Total_Cost
  from services
  where Cost_of_service + Extra_charges >700;



- Number of tuples - 7

27. The Total number of perfect worker/services – (rating = 10, feedback-very good)?

- select *
  from Customer_review as cr natural join worker_information2 as wi
  where cr.worker_id = wi.worker_id and feedback ='Very Good' and rating_out_of_10 = 10;

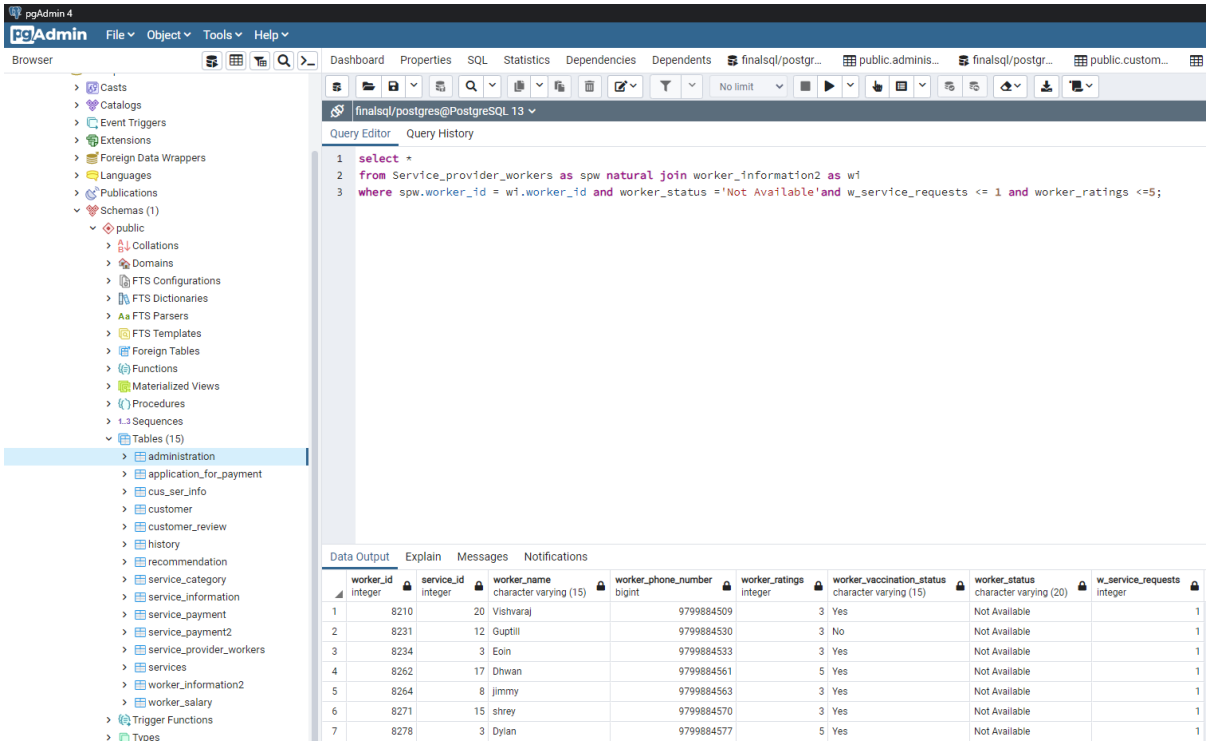


	worker_id integer	customer_id integer	service_id integer	transaction_id integer	feedback character varying (100)	rating_out_of_10 integer	worker_status character varying (20)	w_service_requests integer
1	8207	31	17	25025	Very Good	10	Available	0
2	8208	10	18	25048	Very Good	10	Pending	3
3	8222	52	8	25039	Very Good	10	Not Available	1
4	8222	27	8	25038	Very Good	10	Not Available	1
5	8224	83	3	25097	Very Good	10	Not Available	2
6	8239	59	20	25008	Very Good	10	Not Available	2
7	8269	50	2	25093	Very Good	10	Available	0
8	8269	61	2	25058	Very Good	10	Available	0
9	8275	1	8	25068	Very Good	10	Pending	3
10	8283	85	11	25017	Very Good	10	Pending	3

- Number of tuples - 11

28. The worst Performing worker (not vaccinated, less ratings and pending services <=1)?

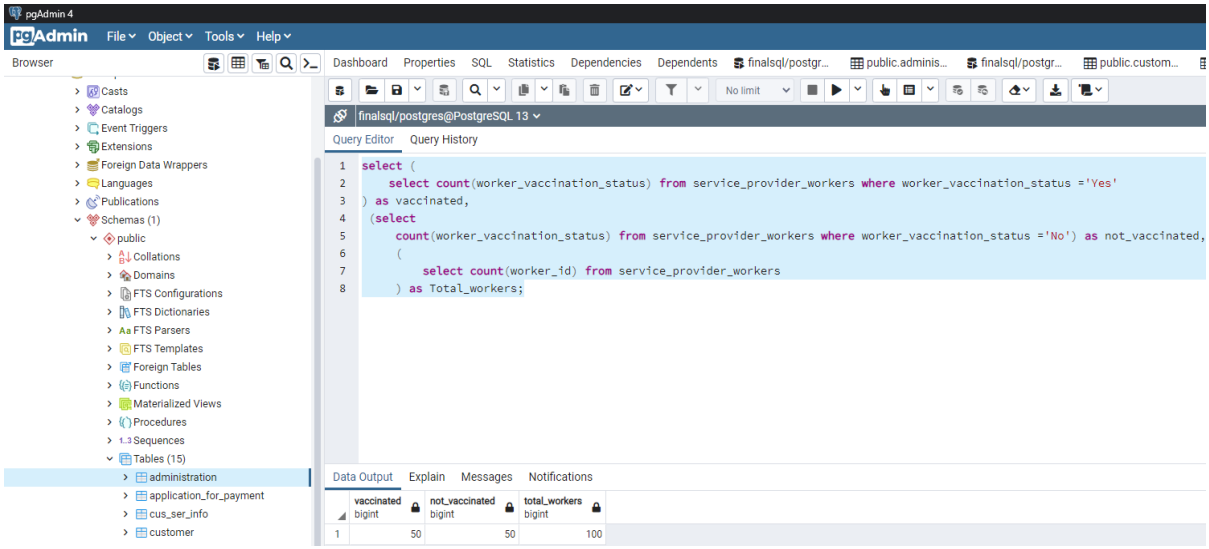
- select *
  from Service_provider_workers as spw natural join worker_information2 as wi
  where spw.worker_id = wi.worker_id and worker_status ='Not Available'and
  w_service_requests <= 1 and worker_ratings <=5;



- Number of tuples – 7

29. Total number of Vaccinated and not vaccinated worker?

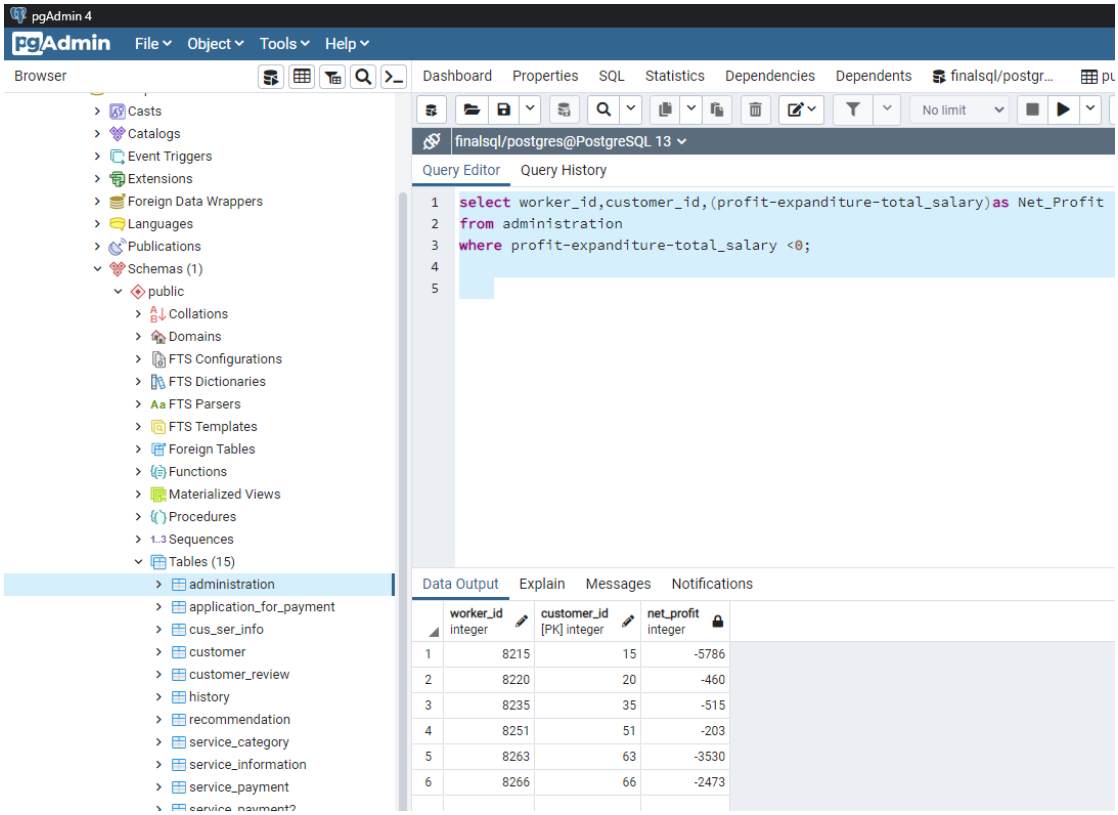
- ```
  select (select count(worker_vaccination_status) from service_provider_workers where worker_vaccination_status ='Yes'
  ) as vaccinated,
   (select
  		count(worker_vaccination_status) from service_provider_workers where worker_vaccination_status ='No') as not_vaccinated,
  		(
  			select count(worker_id) from service_provider_workers
  		) as Total_workers;
  ```
  ```



- Number of tuples – 1

30. Services which lead to a complete loss for the company (profit – expenditure-salary)?

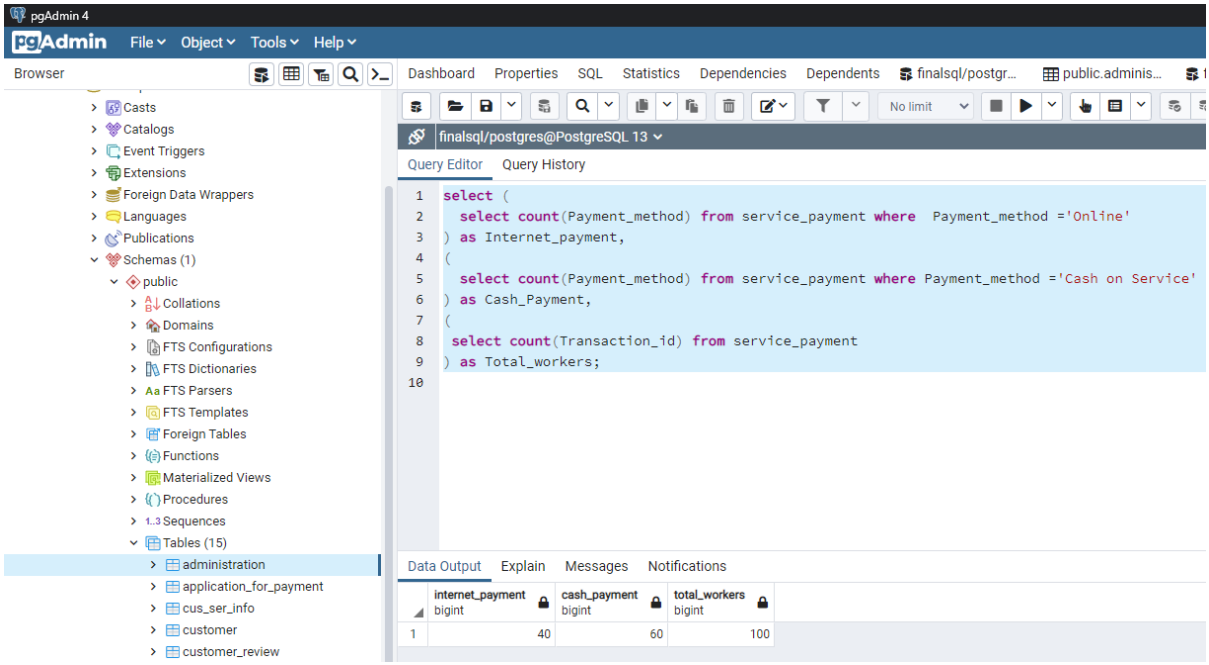
- select worker_id,customer_id,(profit-expanditure-total_salary) as Net_Profit from administration where profit-expanditure-total_salary <0;



- Number of tuples – 6

31. Total number of online vs the total number of offline services in the past 100 services ?

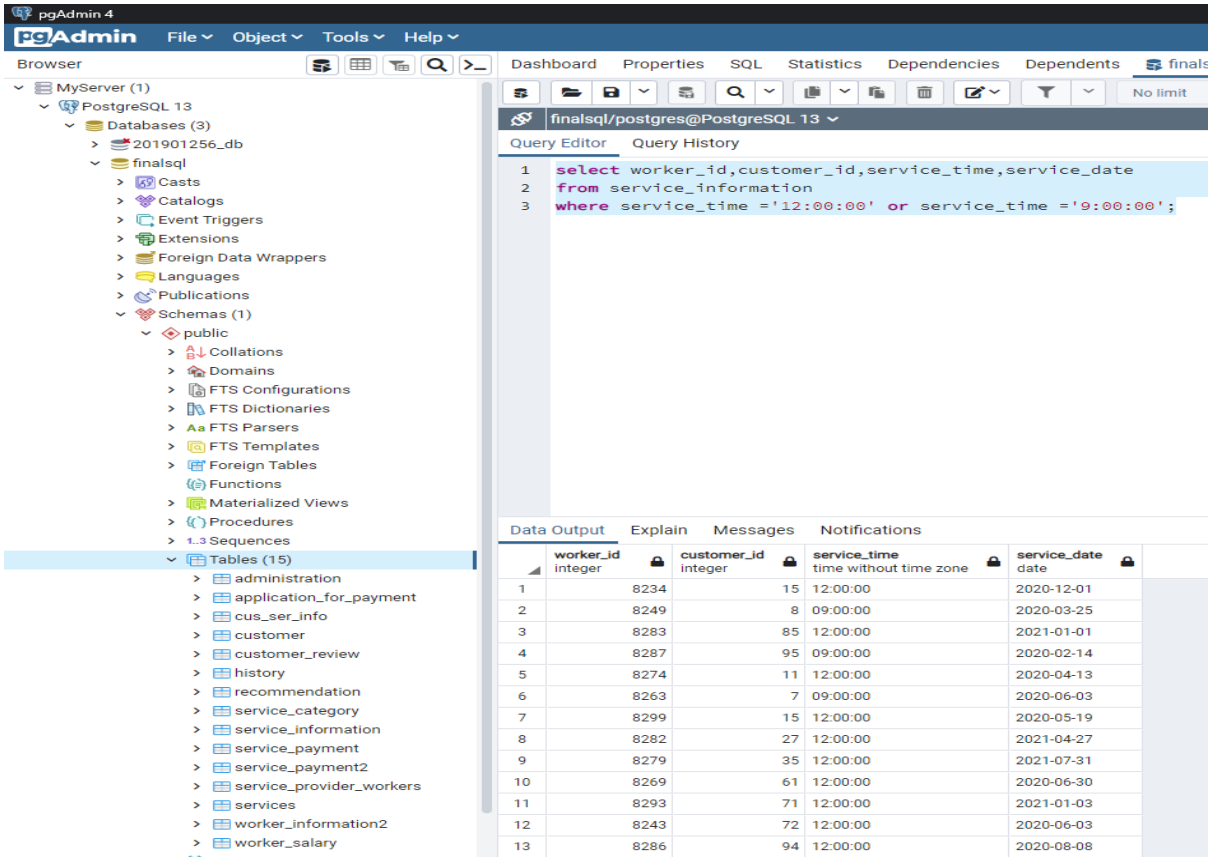
- select (
  select count(Payment_method) from service_payment where Payment_method ='Online'
) as Internet_payment,
(
  select count(Payment_method) from service_payment where Payment_method ='Cash on Service'
) as Cash_Payment,
(
 select count(Transaction_id) from service_payment
) as Total_workers;



- Number of tuples – 1

32. The wrong time services, service at breakfast or lunch times?

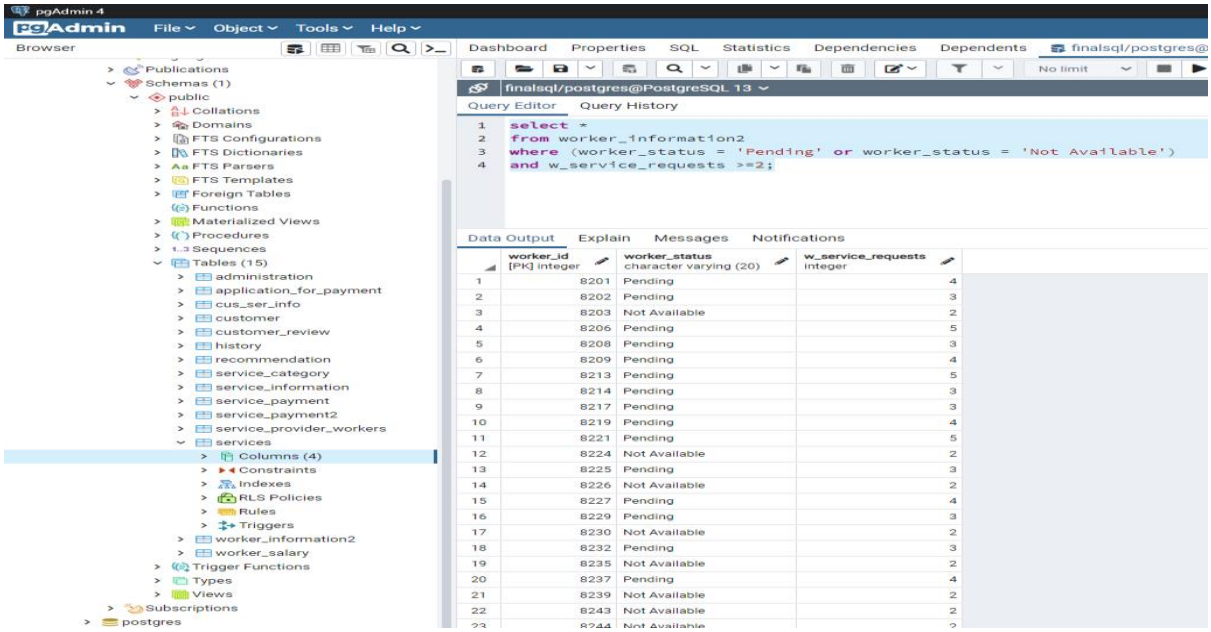
- select worker_id,customer_id,service_time,service_date
from service_information
where service_time ='12:00:00' or service_time ='9:00:00'



- Number of tuples – 19

33. Irresponsible workers – workers which are not available yet they have services to be done ?

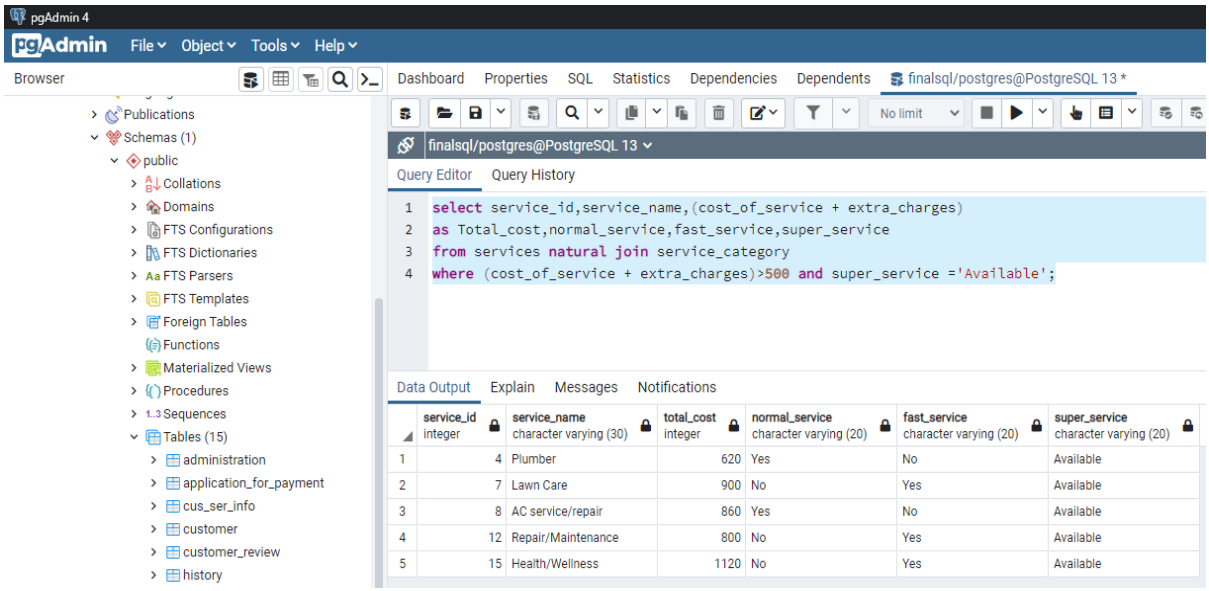
- ```
  select *
  from worker_information2
  where (worker_status = 'Pending' or worker_status = 'Not Available')
  and w_service_requests >=2;
  ```
  ```



- Number of tuples – 53

34. Services which are categorized in 'Super'(services in discount) but yet are expensive(>500)?

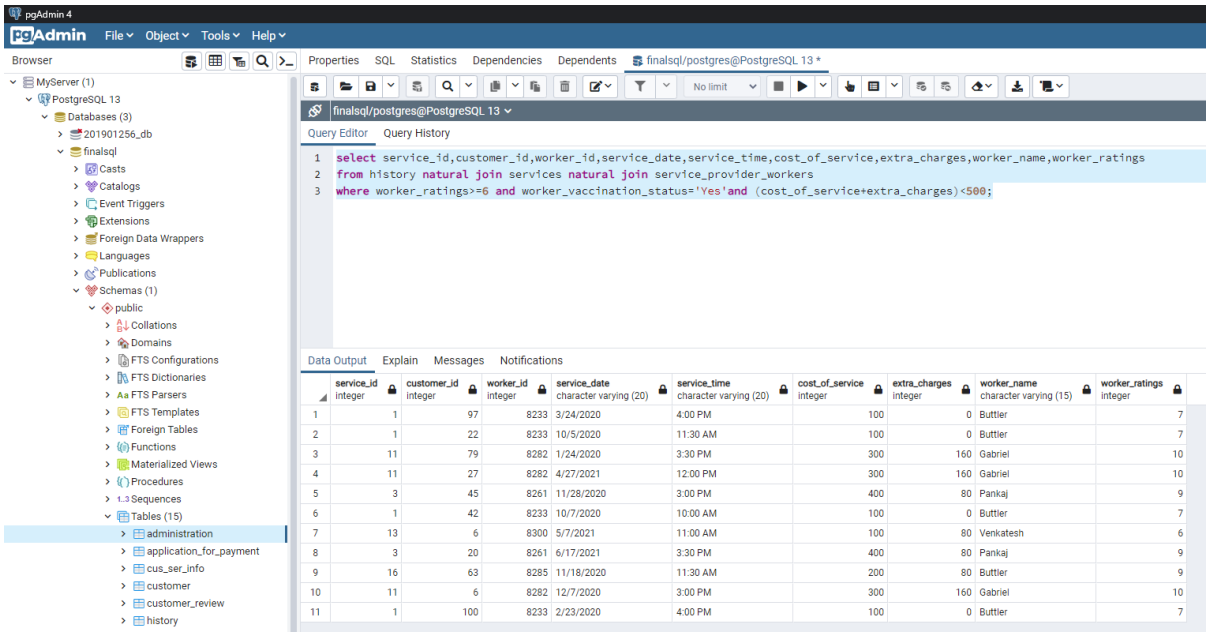
- select service_id,service_name,(cost_of_service + extra_charges) as Total_cost,normal_service,fast_service,super_service from services natural join service_category where (cost_of_service + extra_charges)>500 and super_service ='Available';



- Number of tuples - 5

35. All the details of the service which are successfully and safely done (less expensive <500, worker vaccinated, good rating, feedback)?

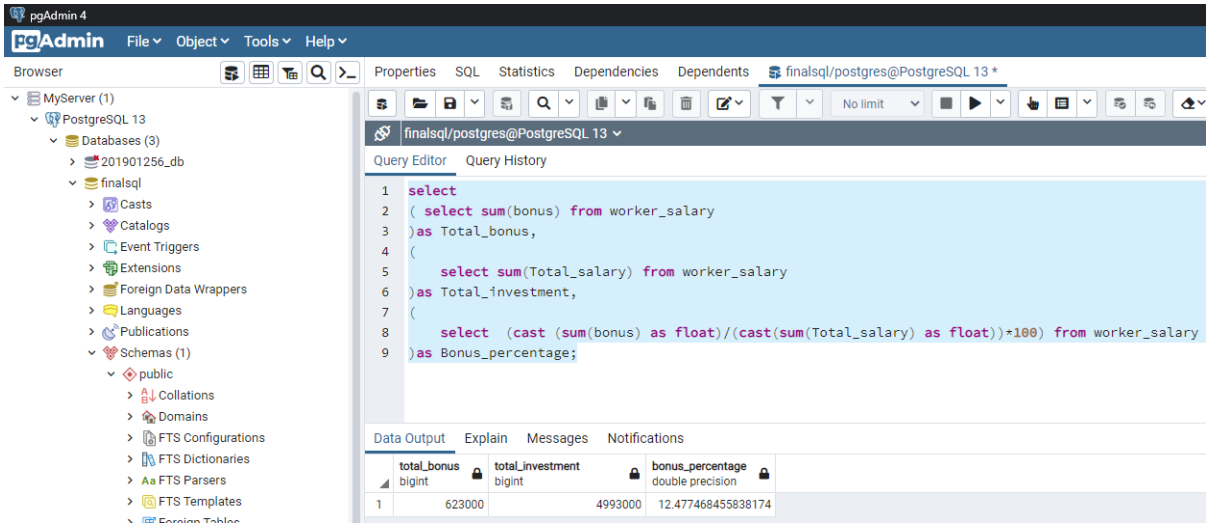
- select service_id, customer_id, worker_id, service_date, service_time, cost_of_service, extra_charges, worker_name, worker_ratings from history natural join services natural join service_provider_workers where worker_ratings>=6 and worker_vaccination_status='Yes'and (cost_of_service+extra_charges)<500;



- Number of tuples – 11

36. The extra investment of the company (bonus percentage in total_salary)?

- select
(select sum(bonus) from worker_salary
)as Total_bonus,
(
select sum(Total_salary) from worker_salary
)as Total_investment,
(
select (cast (sum(bonus) as float)/(cast(sum(Total_salary) as float))*100) from worker_salary
)as Bonus_percentage;



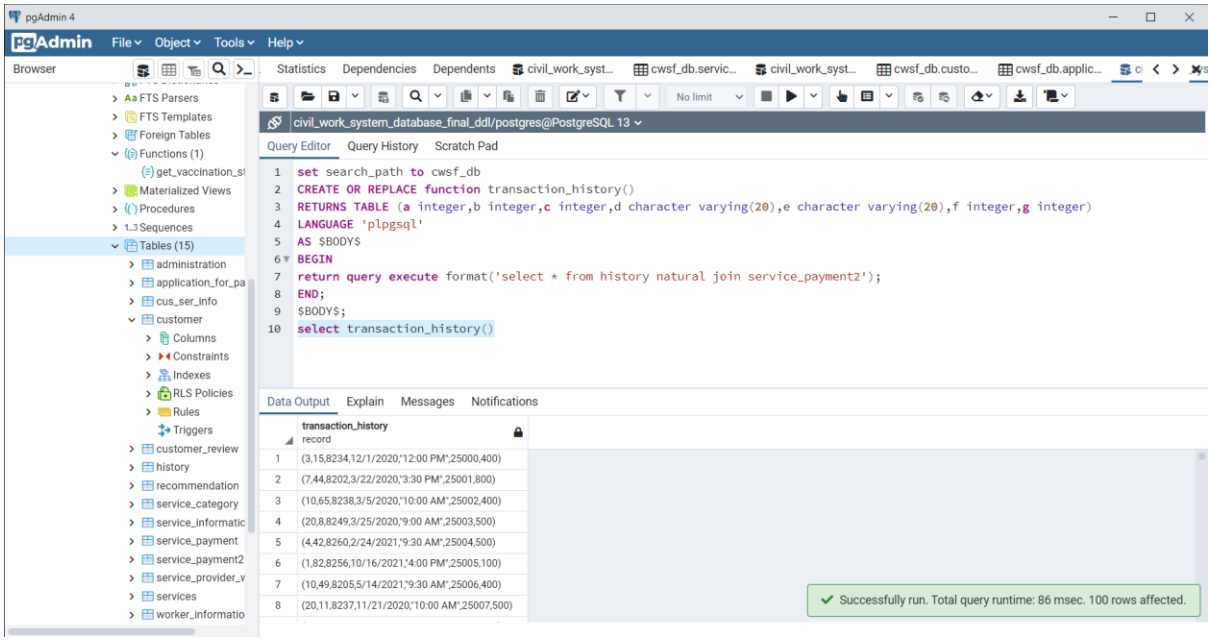
- Number of tuples – 1

37. Create function which will display history of services and its payment details.

- set search_path to cwsf_db

```
CREATE OR REPLACE function transaction_history()
RETURNS TABLE (a integer,b integer,c integer,d character varying(20),e character
varying(20),f integer,g integer)
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
return query execute format('select * from history natural join service_payment2');
END;
$BODY$;

select transaction_history()
```



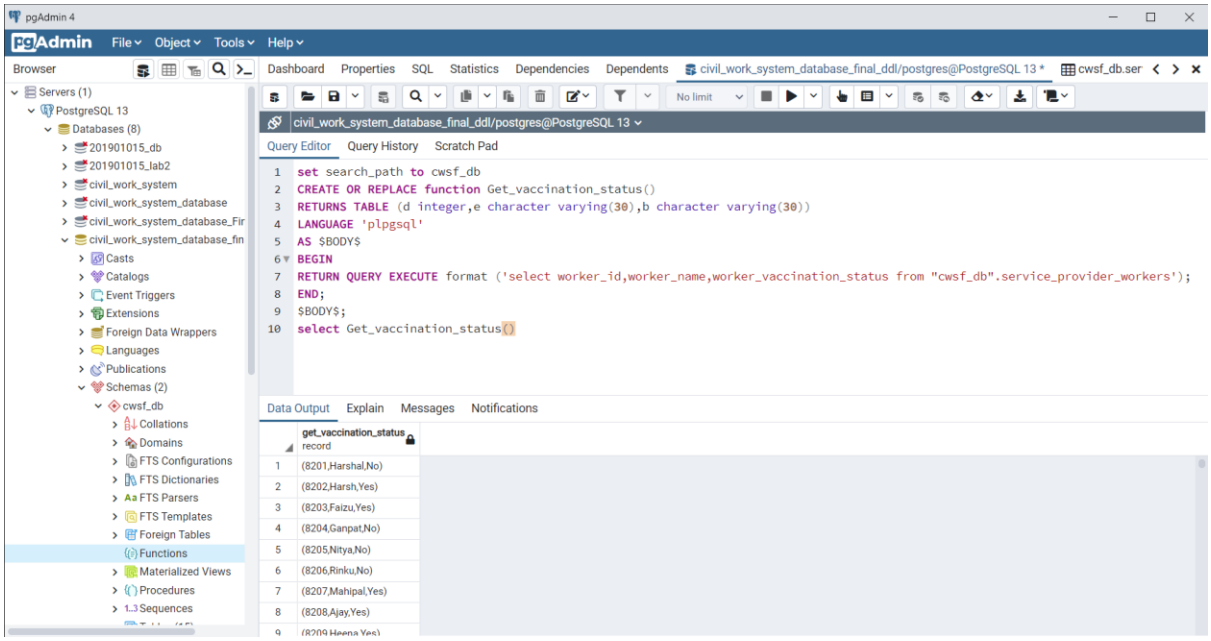
- Number of tuples – 8

38. Create function which will display worker's vaccination status with worker_id.

- set search_path to cwsf_db

```
CREATE OR REPLACE function Get_vaccination_status()
RETURNS TABLE (d integer,e character varying(30),b character varying(30))
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
RETURN QUERY EXECUTE format ('select
worker_id,worker_name,worker_vaccination_status from
"cwsf_db".service_provider_workers');
END;
$BODY$;

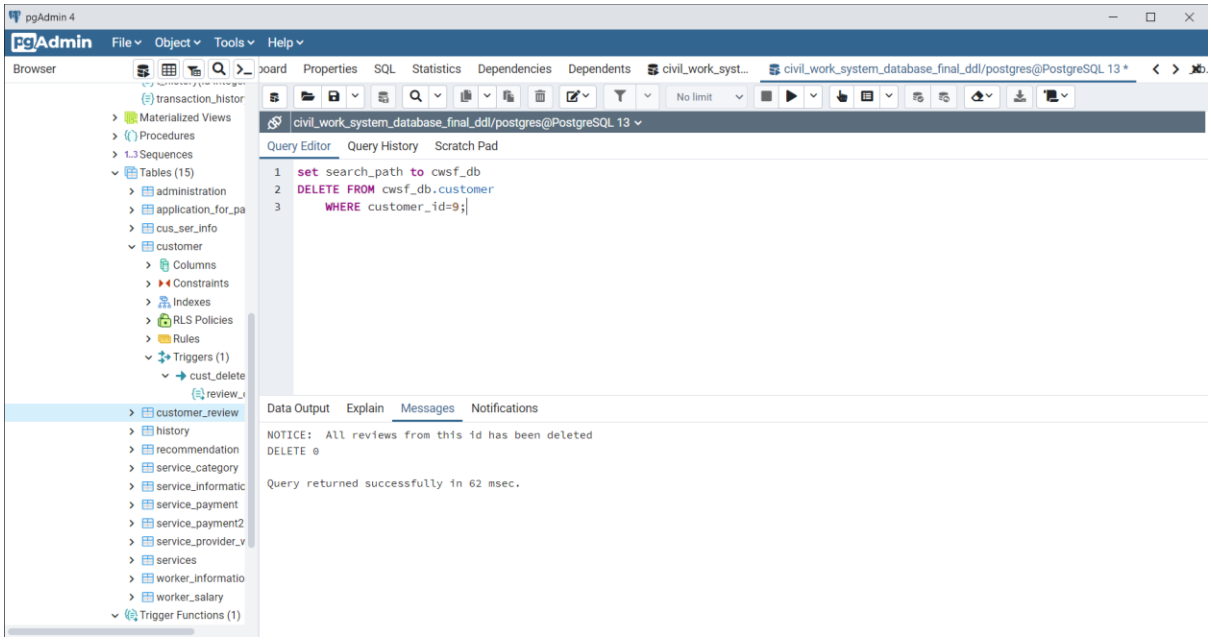
select Get_vaccination_status()
```



- Number of tuples – 100

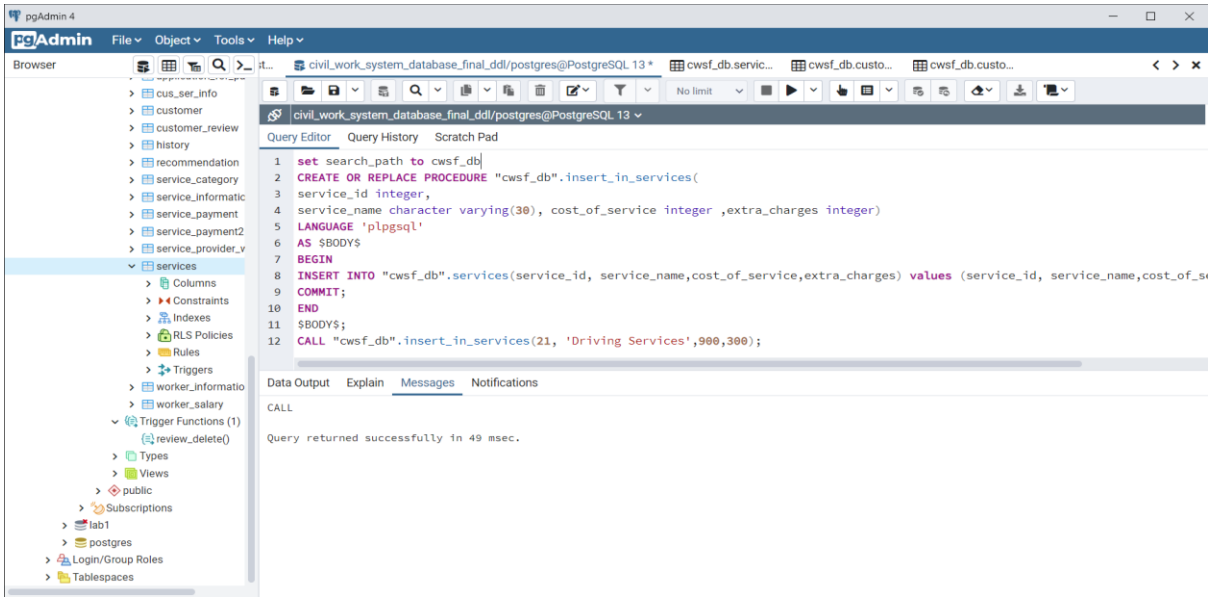
39. Create a trigger function that delete the reviews of the customer as soon as anyone delete the customer from customer table.

- ```
  CREATE OR REPLACE FUNCTION cwsf_db.review_delete()
      RETURNS trigger
      LANGUAGE 'plpgsql'
      VOLATILE
      COST 100
  AS $BODY$
  declare cid bigint;
  begin
  select "customer_id" into cid from cwsf_db.customer where
  "customer_id" = old."customer_id";
  if(cid = old."customer_id") then
  delete from cwsf_db.customer_review
  where customer_id = cid;
  raise notice 'All reviews from this id has been deleted';
  return null;
  else
  raise notice 'Reviews from this ID does not exist';
  return null;
  end if; end
  ```
  ```



40. Create a store procedure for inserting services in service table

- set search_path to cwsf_db
  CREATE OR REPLACE PROCEDURE "cwsf_db".insert_in_services(
  service_id integer,
  service_name character varying(30), cost_of_service integer ,extra_charges integer)
  LANGUAGE 'plpgsql'
  AS $BODY$
  BEGIN
  INSERT INTO "cwsf_db".services(service_id,
  service_name,cost_of_service,extra_charges) values (service_id,
  service_name,cost_of_service,extra_charges);
  COMMIT;
  END
  $BODY$;
  CALL "cwsf_db".insert_in_services(21, 'Driving Services',900,300);


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
civil_work_system_database_final_ddl/postgres@PostgreSQL 13 *
cwsf_db.servic...
cwsf_db.custo...
cwsf_db.custo...
No limit
civil_work_system_database_final_ddl/postgres@PostgreSQL 13
Query Editor
Query History
Scratch Pad
cus_ser_info
customer
customer_review
history
recommendation
service_category
service_informatic
service_payment
service_payment2
service_provider_v
services
Columns
Constraints
Indexes
RLS Policies
Rules
Triggers
worker_informatio
worker_salary
Trigger Functions (1)
review_delete()
Types
Views
public
Subscriptions
lab1
postgres
Login/Group Roles
Tablespaces
1 set search_path to cwsf_db
2 CREATE OR REPLACE PROCEDURE "cwsf_db".insert_in_services(
3 service_id integer,
4 service_name character varying(30), cost_of_service integer ,extra_charges integer)
5 LANGUAGE 'plpgsql'
6 AS $BODY$
7 BEGIN
8 INSERT INTO "cwsf_db".services(service_id, service_name,cost_of_service,extra_charges) values (service_id, service_name,cost_of_s
9 COMMIT;
10 END
11 $BODY$;
12 CALL "cwsf_db".insert_in_services(21, 'Driving Services',900,300);
Data Output
Explain
Messages
Notifications
CALL
Query returned successfully in 49 msec.

Section7
Project Code with output screenshots

Code and Snapshots:

We are implementing our database in html and python. Here, we are providing important codes only because there are many templates.

Code for this: -

Html and CSS file:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Services | Home</title>
    <style>
    * {
        box-sizing: border-box;
    }

    body {
        margin: 0px;
        font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
    }

    header h1 {
        text-align: center;
        font-size: 40px;
        color: white;
    }

    header {
        padding: 40px;
        background-color: rgb(17, 124, 143);
    }
```

```
nav {
    background-color: #333;
    overflow: hidden;
}

nav a {
    text-decoration: none;
    padding: 20px;
    text-align: center;
    float: left;
    color: white;
}

main {
    background-color: white;
    text-align: center;
}

h2 {
  text-align: center;
    padding: 20px;
}
h3 {
  word-spacing: 30px;
  text-align: center;
    padding: 20px;
}
h4 {
  text-align: center;
    padding: 20px;
}
```

```
section:nth-child(even) {
    background-color: rgb(241, 241, 241);
}
.column {
  float: left;
  width: 33.33%;
  padding: 5px;
}

/* Clear floats after image containers */
.row::after {
  content: "";
  clear: both;
  display: table;
}
section {
    min-height: 100vh;
}

video {
    padding: 10px;
}

iframe {
    height: 75vh;
}

img {
    padding: 10px;
}
```

```
section h5 {
    margin: 50px;
}

footer {
    background-color: #333;
    overflow: hidden;
}

footer a {
    text-decoration: none;
    padding: 20px;
    text-align: center;
    float: left;
    color: white;
}
.finalbutton{
  font-size: 125%;
  font-weight: bolder;
  text-align: center;
  padding: 20px;
  width:25%;
  background-color: rgb(17, 124, 143);
  border: 2px solid black;
  border-radius: 10px;
  color: white;
  margin: 10px;

}

.divbutton{
  text-align: center;
```

```
    margin-top: 10px;
   }
   .divbutton.finalbutton :hover{
    cursor: pointer;
   }
   </style>
  </head>
  <body>
   <header>
    <h1>The Indian Service Company</h1>
   </header>
   <nav>
    <a href="#">Home</a>
    <a href="#">Services</a>
    <a href="#">Workers</a>
    <a href="#">Payment</a>
    <a href="#">History</a>
    <a href="#">Contact Us</a>
   </nav>
   <body>
    <section>
     <h2>Services available</h2>
     <h4 >Click directly on the image to access the service &#128512;!!</h4>
     <a href="images\AC_ht.html" target="_blank"><img width="30%" height="15%"
src="images\AC.jpg" alt="The AC service promotion Photo"</a>

     <a href="images\paint_ht.html" target="_blank"><img width="30%" height="15%"
src="images\paint.jpg" alt="Painter image"></a>

     <a href="images\worker_ht.html" target="_blank"><img width="30%" height="15%"
src="images\worker.jpg" alt="Worker image"></a>
     <a href="images\AC_ht.html" target="_blank"><img width="30%" height="15%"
src="images\spa.png" alt="Worker image"></a>
```

<img width="30%" height="15%" src="images\pest-control-services.png" alt="Worker image">

<img width="30%" height="15%" src="images\plumber.jpg" alt="Worker image">

<img width="30%" height="15%" src="images\yoga instructor.jpg" alt="Worker image">

<img width="30%" height="15%" src="images\massagewomen.jpg" alt="Worker image">

</section>

<hr>

<button class="finalbutton" type="button" onclick= "window.location.href='http://127.0.0.1:5000/table'"name="button" target= "_blank" >Move to Tables</button>

<section>

<h2>The Owners</h2>

<h3>Baveet_Singh

Vishvarajsinh_Chauhan

Harsh_Patel</h3>

<h4>Mainly, we are using database to manage inconsistency, difficult data accessing, security, reliable service, concurrent access and many such problems. Database manager is used to alter, add, update or delete the record.</h4>

<h5>Here, we are making database based on Service request for civil works (like - plumbing, electrical problem, carpeting etc.) Mostly customers are not easily finding the services for their problems. Here our main purpose is to make database system is used to give customers to their needed services easily and in reliably way. This database help customers to get services for their problems in its requirements and also, they get services by good workers.

customer also can rate and give feedback for the services which they get and by the feedback, we can improve the database based on customers feedback and their requirements.</h5>

</section>

</body>

<footer>


```
        <a href="#">FAQs</a>
        <a href="#">Contact US</a>
        <a href="#">Headquater India</a>
        <a href="#">Headquater Australia</a>
        &copy;
      </ul>
    </footer>


  </body>
</html>
```

For connection of SQL and Webpage:

% Webpage%

```
from flask import Flask, render_template, request, redirect, url_for


from flask.json import dump
from numpy.lib import type_check
import webbrowser
import sql_interface as si
import numpy as np



app = Flask(__name__, template_folder='templates', static_folder='static')


tables = [  "administration", "application_for_payment", "cus_ser_info", "customer",
"customer_review", "history", "recommendation", "service_category", "service_information",
"service_payment",
          "service_payment2", "service_provider_workers", "services", "worker_information2",
"worker_salary" ]


#home page
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
```

```
        if request.form.get('showTable'):
            return redirect(url_for('selectTabletoShow'))
        elif request.form.get('editTable'):
            return redirect(url_for('selectTabletoEdit'))
        elif request.form.get('customQuery'):
            return redirect(url_for('customQuery'))
    return redirect(url_for('selectTabletoShow'))

#Custom query
@app.route('/custom', methods=['GET', 'POST'])
def customQuery():
    query = request.args.get('query')
    if query == None:
        query = ''

    if request.method == 'POST':
        if request.form.get('showTable'):
            return redirect(url_for('selectTabletoShow'))
        elif request.form.get('editTable'):
            return redirect(url_for('selectTabletoEdit'))

        elif request.form.get('Submit'):
            query = request.form.get('query')
            print(query)
            return redirect(url_for('customTable', query=query))

    return render_template('customQuery.html', query=query)

#custom Table show
@app.route('/custom/query', methods=['GET', 'POST'])
def customTable():
    query = request.args.get('query')
```

```
    if request.method == 'POST':
        if request.form.get('showTable'):
            return redirect(url_for('selectTabletoShow'))
        elif request.form.get('editTable'):
            return redirect(url_for('selectTabletoEdit'))

        elif request.form.get('Submit'):
            query = request.form.get('query')
            return redirect(url_for('customTable', query=query))
    data = si.get_data(query)
    if data == None:
        webbrowser.open_new("https://youtu.be/dQw4w9WgXcQ")
        return redirect(url_for('customQuery', query=query))
    return render_template('customTable.html', data=data, query=query)

#Table selceiotn page to show
@app.route('/table', methods=['GET','POST'])
def selectTabletoShow():
    if request.method == 'POST':
        if request.form.get('editTable'):
            return redirect(url_for('selectTabletoEdit'))
        elif request.form.get('customQuery'):
            return redirect(url_for('customQuery'))

        if request.form.get('tblnam')!= 'nothing':
            tableName = request.form.get('tblnam')
            print(tableName)
            return redirect(url_for('ShowTable', tableName=tableName))
    return render_template('selectTable.html', tables=tables)

#Table Select for editing
@app.route('/edit', methods=['GET','POST'])
```

```
def selectTabletoEdit():
    if request.method == 'POST':
        if request.form.get('showTable'):
            return redirect(url_for('selectTabletoShow'))
        elif request.form.get('customQuery'):
            return redirect(url_for('customQuery'))

        if request.form.get('tblnam')!= 'nothing':
            tableName = request.form.get('tblnam')
            print(tableName)
            return redirect(url_for('EditTable', tableName=tableName))
    return render_template('editTable.html', tables=tables)

# display tables
@app.route('/table/<tableName>', methods=['GET', 'POST'])
def ShowTable(tableName):
    if tableName not in tables:
        return redirect(url_for('selectTabletoShow'))
    if request.method == 'POST':
        if request.form.get('editTable'):
            return redirect(url_for('selectTabletoEdit'))
        elif request.form.get('customQuery'):
            return redirect(url_for('customQuery'))

        tableName = request.form.get('tblnam')
        print(tableName)
        return redirect(url_for('ShowTable', tableName=tableName))
    headings = np.array(si.get_data("select column_name from information_schema.columns where table_name='{}'ORDER BY ORDINAL_POSITION".format(tableName))).flatten()
    rows = np.array(si.get_data("select * from {}".format(tableName)))
    return render_template('table.html', headings=headings, rows = rows, tableName=tableName, tables=tables)
```

```
# Insert in table
@app.route('/edit/<tableName>', methods=['GET', 'POST'])
def EditTable(tableName):
    headings = np.array(si.get_data("select column_name from information_schema.columns where table_name='{}'ORDER BY ORDINAL_POSITION".format(tableName))).flatten()
    if tableName not in tables:
        return redirect(url_for('selectTabletoEdit'))
    if request.method == 'POST':
        if request.form.get('showTable'):
            return redirect(url_for('selectTabletoShow'))
        elif request.form.get('customQuery'):
            return redirect(url_for('customQuery'))


        elif request.form.get('Submit'):
            query = 'insert into {}('.format(tableName) + ', '.join(['%s' for _ in range(len(headings))]) + ') values(' + ', '.join(['\'%s\'' for _ in range(len(headings))]) + ')'
            temp = headings
            for i in headings:
                temp = np.append(temp, request.form.get(i))
            temp = tuple(temp)
            query = query% temp
            print(query)
            si.insert_data(query)
        elif request.form.get('tblnam') != 'nothing':
            tableName = request.form.get('tblnam')
            print(tableName)
            return redirect(url_for('EditTable', tableName=tableName))
    return render_template('edit.html', headings=headings, tableName=tableName, tables=tables)



if __name__ == '__main__':
    app.run(debug=True)
```

```
% Interface%
import psycopg2
from config import config

def connect():
    """ Connect to the PostgreSQL database server """
    conn = None
    try:
        # read connection parameters
        params = config()

        # connect to the PostgreSQL server
        print('Connecting to the PostgreSQL database...')
        conn = psycopg2.connect(**params)

        # create a cursor
        cur = conn.cursor()

            # execute a statement
        print('PostgreSQL database version:')
        cur.execute('SELECT version()')

        # display the PostgreSQL database server version
        db_version = cur.fetchone()
        print(db_version)

            # close the communication with the PostgreSQL
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
```

```
            conn.close()
            print('Database connection closed.')

def get_data(query):
    conn = None
    try:
        params = config()
        conn = psycopg2.connect(**params)
        cur = conn.cursor()
        cur.execute('set search_path to public')
        cur.execute(query)
        print("The number of parts: ", cur.rowcount)
        row = cur.fetchone()
        rows = []
        while row is not None:
            rows.append(row)
            print(row)
            row = cur.fetchone()
        return rows
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
            conn.close()

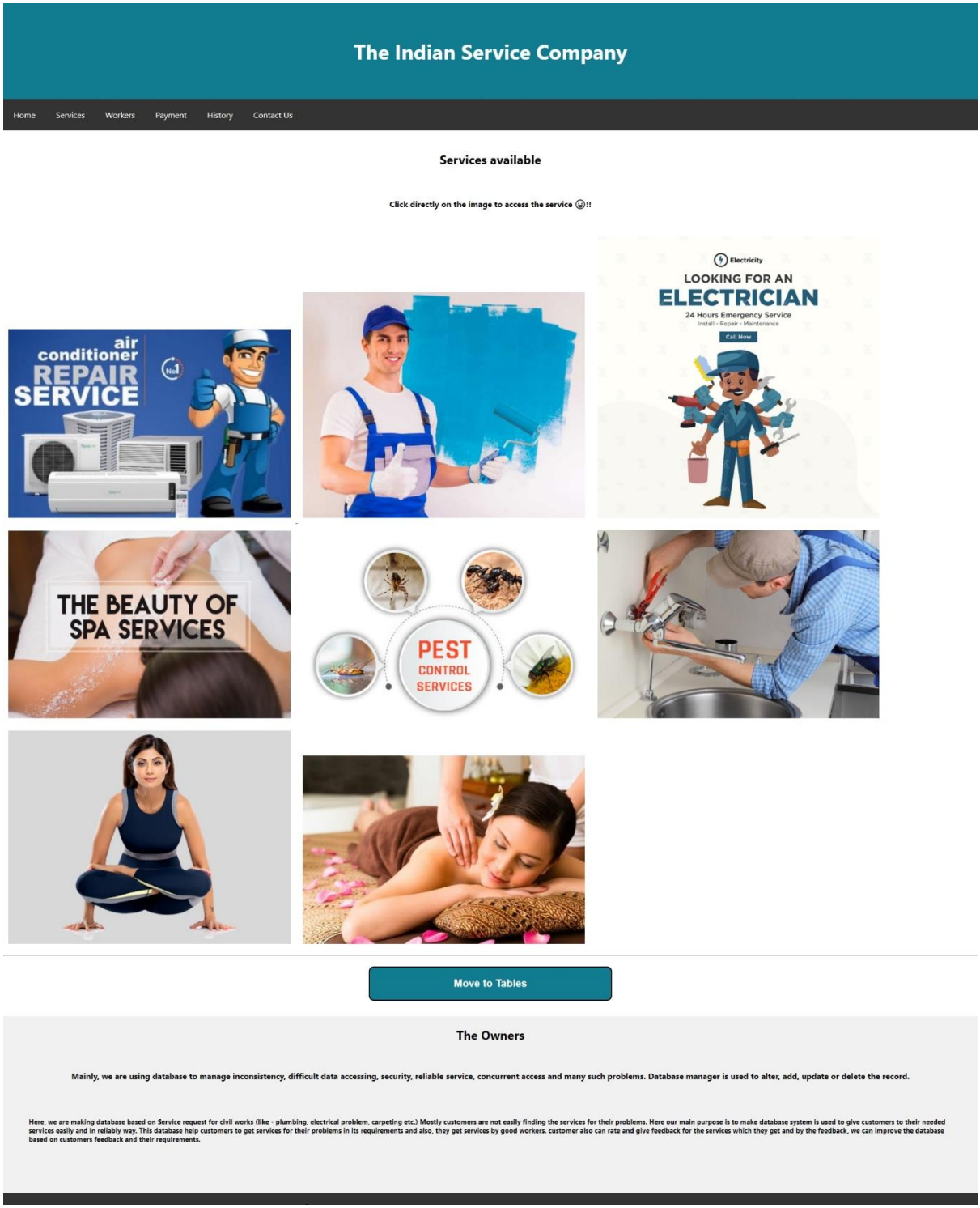
def insert_data(query):
    conn = None
    try:
        params = config()
        conn = psycopg2.connect(**params)
        cur = conn.cursor()
```

```
        cur.execute('set search_path to public')
        cur.execute(query)
        print("The number of parts: ", cur.rowcount)
        print(cur.statusmessage)
        conn.commit()
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        conn.rollback()
        print(error)
    finally:
        if conn is not None:
            conn.close()

if __name__ == '__main__':
    # data = get_data('employee 5')
    insert_data('insert into departments(department) values(\'hello\');')
```

Snapshots of the webpage and some functionalities of that webpages:

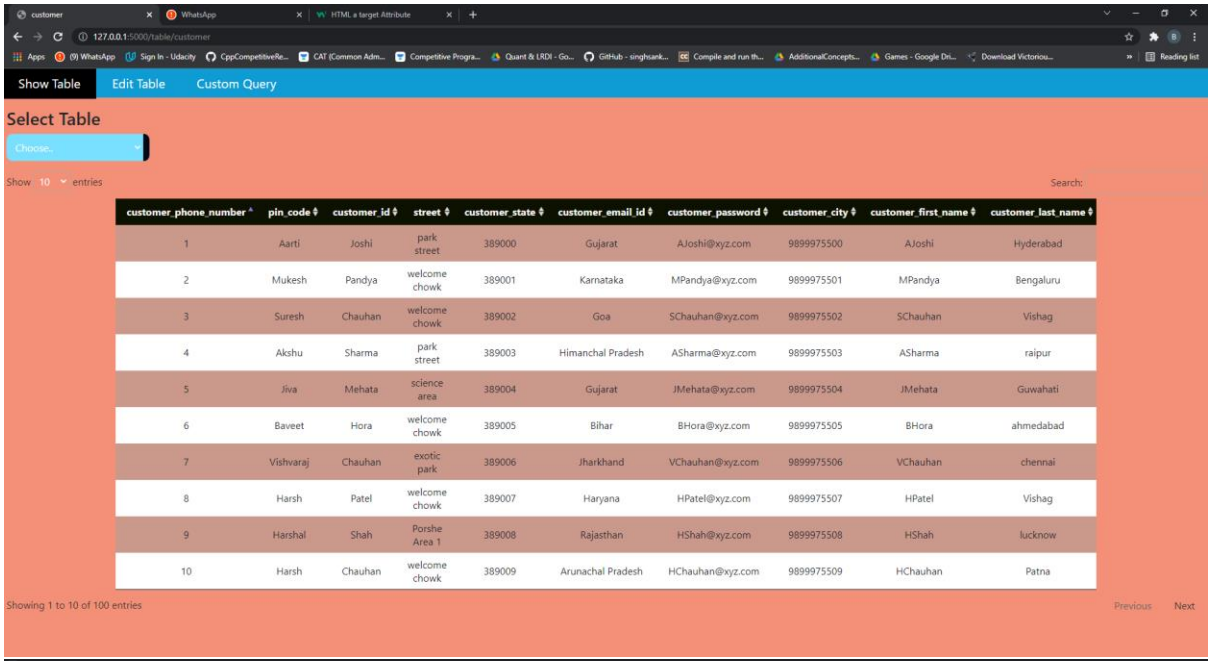
Home page:

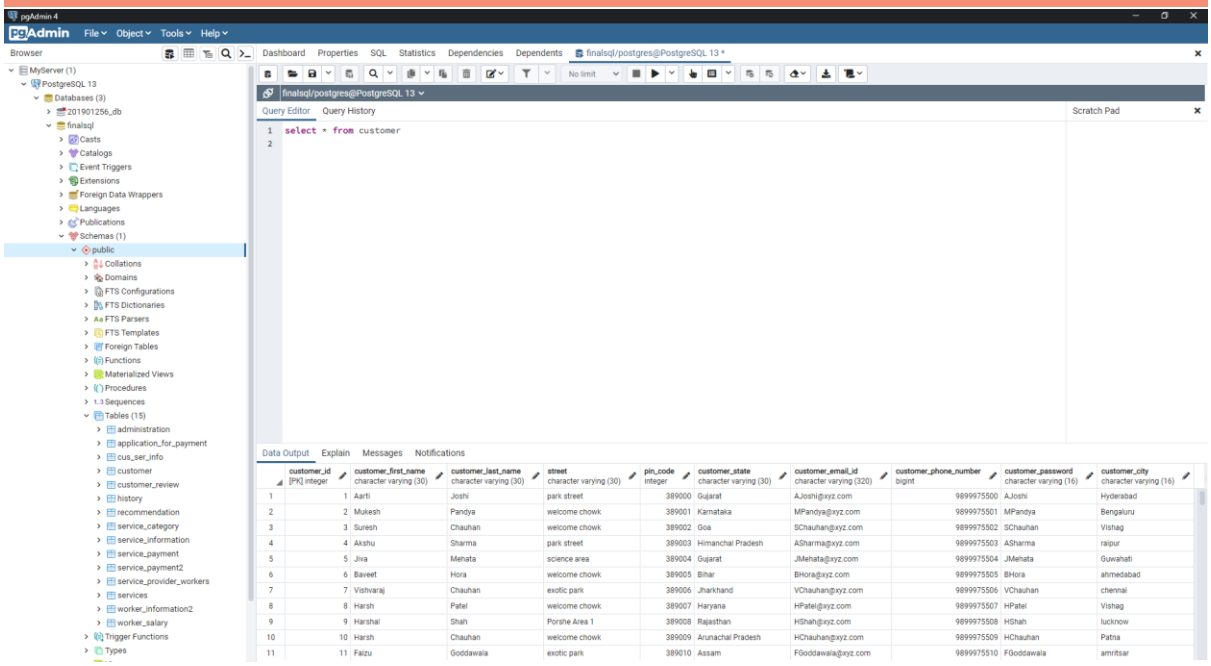


Here, we make one website named the Indian Service Company. It provides the basic operations like Home, Services, Workers, Payment, History, Contact us. For simplicity we make one button for database named “Move to Tables”. It provides three main features Show the table information, Edit the table option that provides the update, insert, delete and many more and last Custom query that we wish to compile in postures SQL.

Functionalities snapshots:

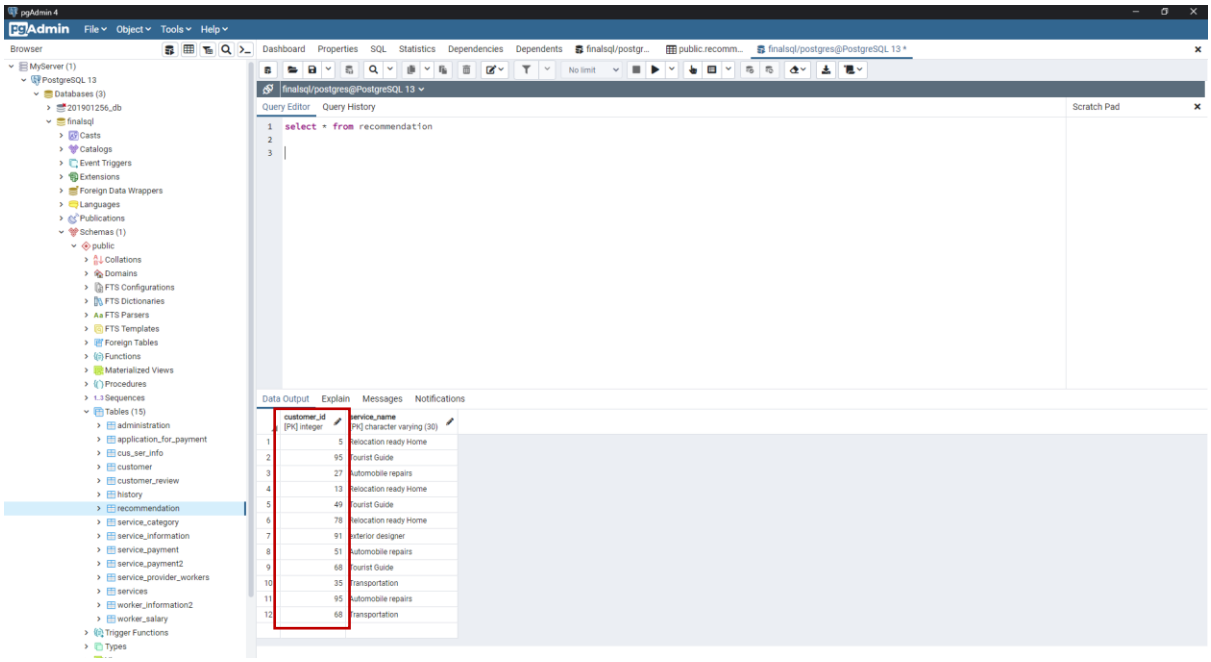
In the below image it shows the customer's table information. Here we can see the select table options that provides the list of the tables. If we want to see table information then we have to click on that table name. Next snapshot prove that the information is same for both the snapshots.





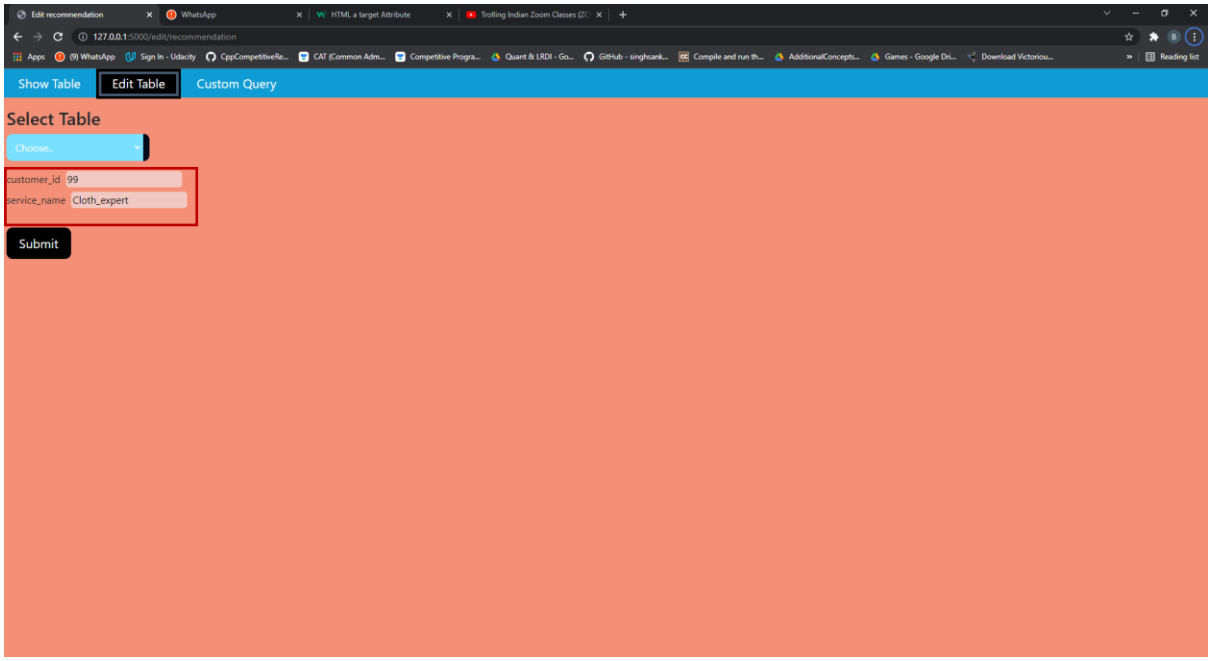
Now for insert/update/delete or many more functionalities there are edit table option that is clearly shown in below images. Here we are giving the example of insert the new recommendation in recommendation table.

Initial Recommendation table (Before Insertion operation):

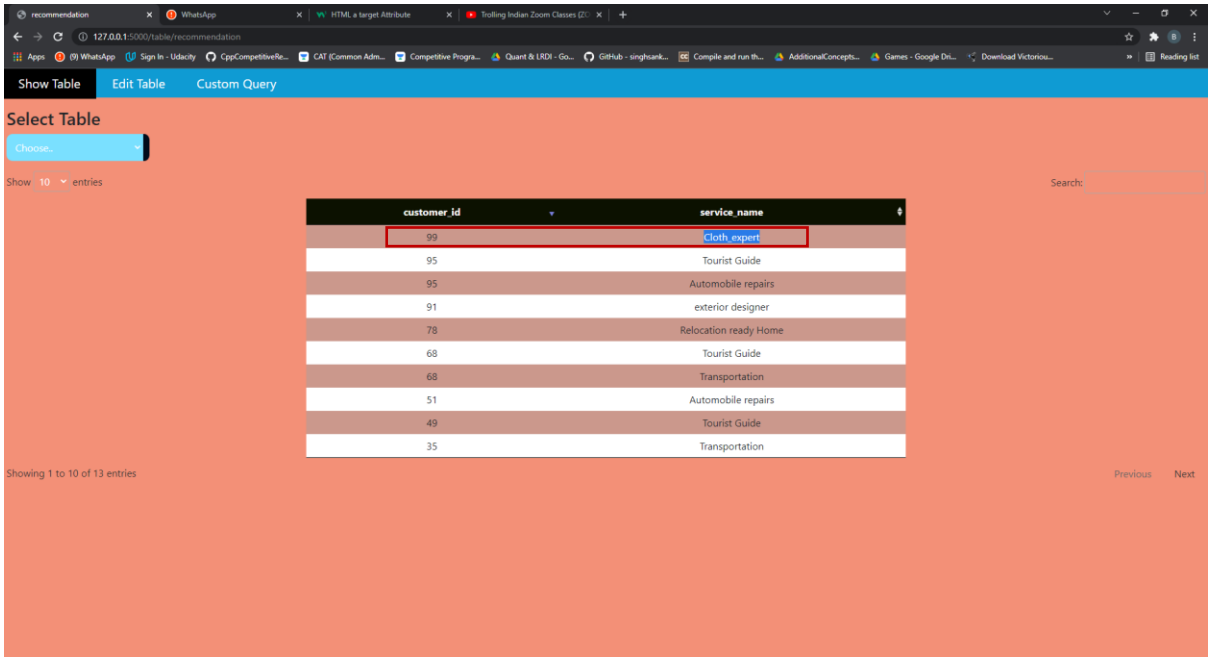


For Insertion:

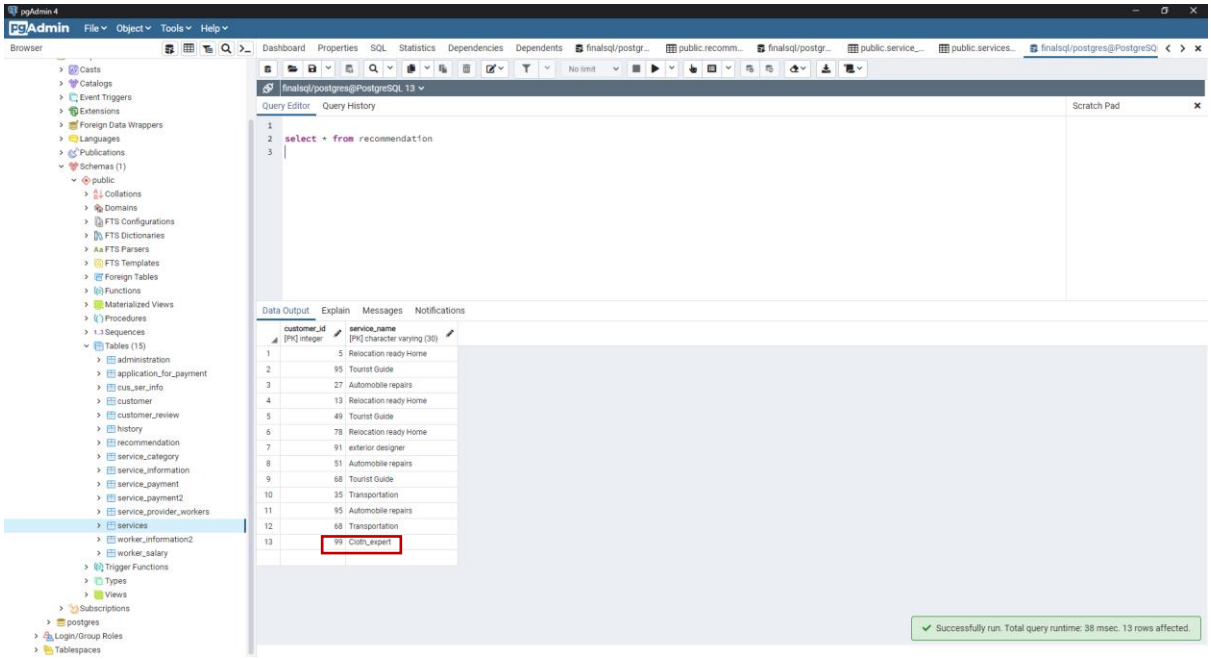
We have to write the information based on the table and submit that information



Updated Recommendation table (After Insertion operation) in Postgres and webpage too:



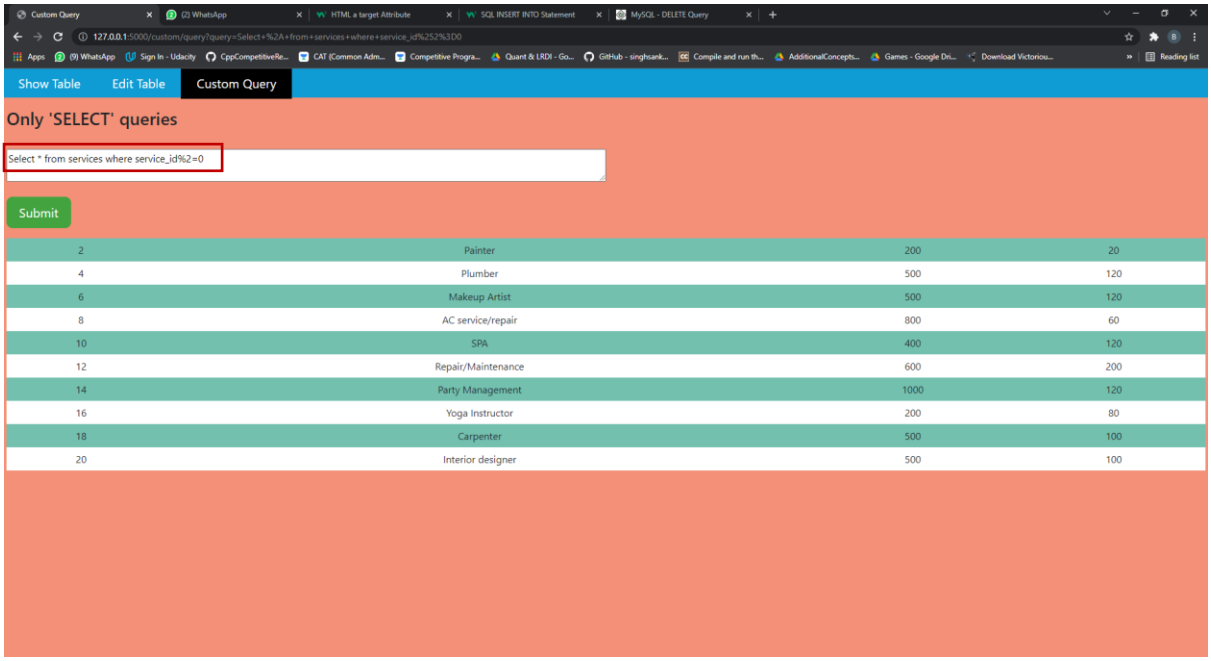
Webpage Recommendation table



Postgres Recommendation table

For Custom Query:

If we have to run some query operation then we don't need to go to the Postgres SQL we can directly run from the custom query option of the webpage. The example of this functionalities is given below by the snapshot.


Custom Query
(2) WhatsApp
HTML a target Attribute
SQL INSERT INTO Statement
MySQL - DELETE Query
127.0.0.1:5000/custom/query?query=Select+%2A+from+services+where+service_id%252%3D0
Show Table
Edit Table
Custom Query
Only 'SELECT' queries
Select * from services where service_id%2=0
Submit
2 Painter 200 20
4 Plumber 500 120
6 Makeup Artist 500 120
8 AC service/repair 800 60
10 SPA 400 120
12 Repair/Maintenance 600 200
14 Party Management 1000 120
16 Yoga Instructor 200 80
18 Carpenter 500 100
20 Interior designer 500 100