Task Management System
S10_T5

DBMS - IT214

Instructor: Prof. Minal Bhise
Mentor TA: Dhairya Patel

Team Members :

201901080 : Boricha Tarun
201901087 : Makwana Vivek
201901091 : Rohit Mahima
201901140 : Patel Khushbu

Dhirubhai Ambani Institute of Information and Communication Technology

Index

Section 1
Final version of SRS

[A]

Introduction :

1. Purpose:-

- ➔ Provide users with an interface through which they may log into the system.
- ➔ Provide project managers with an interface through which they may initialize a new project.
- ➔ Provide project managers with an interface through which they may dynamically add or remove processes from a particular project.
- ➔ Provide project managers with a system alerting them of inconsistencies in process dependencies among the processes chosen for a project.
- ➔ Provide project managers as well as team members with an interface through which they may view all of the processes for a project in the form of a dependency graph.
- ➔ Provide project managers as well as team members with an interface through which they may update the inputs, outputs, and status of a process.
- ➔ Obtain information from and make changes to the database quickly and accurately.

2. Intended Audience and Reading Suggestions:-

This section of the software requirement specification (SRS) document is written for a more general audience, this part is intended for individuals directly involved in the development of projects. This includes hardware-software developers, project consultants, finance manager and team managers and employees. In our hardware manufacturing company, we need mainly engineers who are experts in hardware components and related software like assembly language and microcontroller level coding.

In the general reading suggestion it includes goals and objectives, project scope, general system details, and some major constraints associated with the intended platform which is generally used by the manager.

People or employees who are working in designing a product, they should read the design part, which includes rules and some good samples of design. In our company it is the design of hardware chip, IC, components.

Readers interested in how to organize and handle data should consult data design who are database engineers, which covers data structures and flow patterns utilized by the system.Restrictions, Limitations, and Constraints discusses the general constraints in the project.

Testing engineers interested in the hardware and software testing process should consult the testing issue, which offers a list of test cases, expected responses, information about the testing machine and other pertinent information.

Other side of the reading suggestion is for market people, who use the company's product. For them there is documentation which includes detailed specifications about the product, for example in our company suppose the product is headphone then It includes general info(color,model name, brand name), connector type, from factor, frequency, operating system compatibility etc.

3. Product Scope :

This product has a wide range of applications. So this project helps a firm to be managed very easily. By using this task management system we can keep track of the daily progress of tasks. Some of the functionalities of this task management system is described below :

login : An official can have access to this application through a login id and password.
Display task list : It will display all the tasks assigned to the employees.
Create a new project: For creating a new project
progress: If a task is going on then it will indicate how much work is done and it will keep track of the project's process without talking with the group leader or manager.
Task assigning : The higher authorities can assign a new task to the employees by using it.
Correction comments: If something is going in the wrong direction while executing the task then a correction comment will be displayed to indicate the invalidness of the execution.
Insert/Delete/Update : Manager or employee can insert/Delete/update any information /task .

4. Description

One of the hardest things about running any project is dealing with the messy middle. You might have a vision of what things should look like when you're done, and a good idea of where you are now. But getting there? It's safe to say even the best teams can get lost along the way.

There is an easy tool that can guide your team through the messy middle: Task management.

Instead of huge, vague, and unclear project goals, tasks are clear, descriptive, step-by-step instructions. By breaking down every part of your project into a detailed task, you get a better picture of how you're going to bring it to life. Otherwise, it's like trying to do a puzzle without any idea of what it looks like.

Task management is both a science and an art. But with the right methods and a proper task management system in place, you can ensure your projects run smoothly from start to finish.

What is task management?

Task management is a pretty simple idea. But simple doesn't always mean easy.

In its most basic form, task management is the process of managing a task through its life cycle—from planning, testing, tracking, to reporting on the outcome.

Task Management involves managing all aspects of a task, from its status and priority, to the time, human, and financial resources it needs.

When you're working with a team, proper task management is the foundation for productive days. Task management tools and techniques give you a detailed and up-to-date view of all the moving parts of a project. It helps everyone on your team stay in-sync, productive, and on schedule.

And as an added bonus, breaking projects and milestones down into specific tasks can be a huge motivator.

A powerful task management tool or software is your best friend when it comes to organizing, assigning, tracking, and reporting on tasks.

Not only will it keep all your info in one place that's visible to the entire team, but it also lets you see your task and project progress and where you might be overcommitting resources.

So, now that we understand why task management is so important to running projects, what's the best way to go about actually doing it?

The basics of task management are to capture, organize, and assign everything that needs to get done to complete your project. The more detail you can get upfront, the more people will know what needs to be done.

As with everything when it comes to project management, the only method that matters is the one that works for your team. So while this guide will give you everything you need to know to start properly managing tasks, make sure to experiment and see what your team likes or doesn't.

How to properly prioritize tasks

You can't just look at your giant list and assume you'll know what to do next. Instead, every task needs to be properly prioritized to know what needs to happen to keep the project running smoothly. But this isn't always easy.

Some tasks will naturally be higher priority than others. But there will also be some nebulous ones that you'll be stumped by. You'll want to start by defining the scale you're going to prioritize on. Something like:

- Critical
- High

- Medium
- Low

Tasks by their very definition are tiny. But the return they bring you and your team are massive.

Take the time to break down tasks into their smallest pieces, prioritize them ruthlessly, and find a task management software that gives you visibility, flexibility, and awareness (like Planio) and you'll be on your way to turning that big, scary, audacious project into an actionable, step-by-step plan.

Hardware Manufacturing Task Management System :

1). This task management system will allow us to manipulate the data very easily. Maybe in earlier times this was done on paper which was very hard to deal with. This system also includes a login system for companies and employees.

2). By this system companies which are working for each other and suppliers become more connected. And the employees who are working in various tasks connected with a virtual platform where they can share ideas and make work more perfect. Managers also have direct contact to all employees, So all employees get equal work according to their skill and expertise like software part, hardware level coding, hardware manufacturing. Managers can make sure that all employees are working properly and there is no such thing like someone is loaded with lots of work and someone is free.

3). It is very important to keep information about raw materials. How much is in stock?, From where we will get if stock is empty or in case a company is providing right now that stop providing then which company is next.

4). This system is very useful In order to update database information which comes from employees, customers, other companies(which supply raw materials, components) and for those who access the database and keep updating the Data.

5). Database has been updated with real time so that all the information like employee suggestions or other suggestions from head office, so that company can make appropriate and efficient decisions in working projects.

6). One important factor for task management is safety of workers, safety is important while working on various projects.
There must be all information in the database about the department of the company which is at high risk of accident, and precautions to be taken for working in those areas. If an accident happens then what to do to control the situation very fast. One person should monitor the activity of those areas via a cctv camera .

7). This task management system contains following :

i). Information Storage: This stores all details about employees(ID, department, expertise, phone no., etc), all manager details, required customer details, all product details and lots of things.

ii). Data security:
The major role of the software database comes when we talk of the security level of databases because it's very necessary to keep all the information private and safe from malicious activities. Because of one attack or mistake, the whole system can fail.

Data security is there so that no one can manipulate any information on other employees details or read some personal info. For this we have a login(username and password) system. An authorised person only can access particular data like manager's data can't be accessed by employees. All customers' accounts are also protected, it must be protected with high security because some of the accounts are linked with the customer's debit card, credit card etc.

iii). Customers' data: Containing all the information about customers (customer id, name, phone no. , purchased items, date of purchase).

iv). Update easily: Anyone with permission can update that particular data very easily. And when we need to update the structure of the database, It will also be easy to do by a database expert.

v). All device compatibility: In this database all the information is kept in the way such that other devices with less memory and low speed processor can easily access the data.

8). After production of a hardware, the person who is testing it has to write all the things he noticed and require changes. So it can be taken under consideration during the final testing. So there should be facility to track the testing part.

9). There is a whole section in the database for advertisement, where all our advertisement videos and posters and details about advertisers are kept. Manager can track information about all advertised mediums and how it is going. so that our product can reach the people and other job interested people apply to the company.

10). We classified customer feedback and employee feedback into two different groups so we maintain the simplicity of the database.

11). After the production is complete our task is to put the products into market and keep track of sales. Keep all details about product goes where if we are exporting in another country, profit we get from them, taxes etc.

12). Limitation of this database is that it does not support all speaking languages yet, so people who are using it need to know that particular language (ex. English, Hindi).

[B]

1. Background Reading

- **Description of each reading done**

The IT hardware sector basically comprises activities such as need assessment and definition, development, pre-production processes, production, sales and service support.

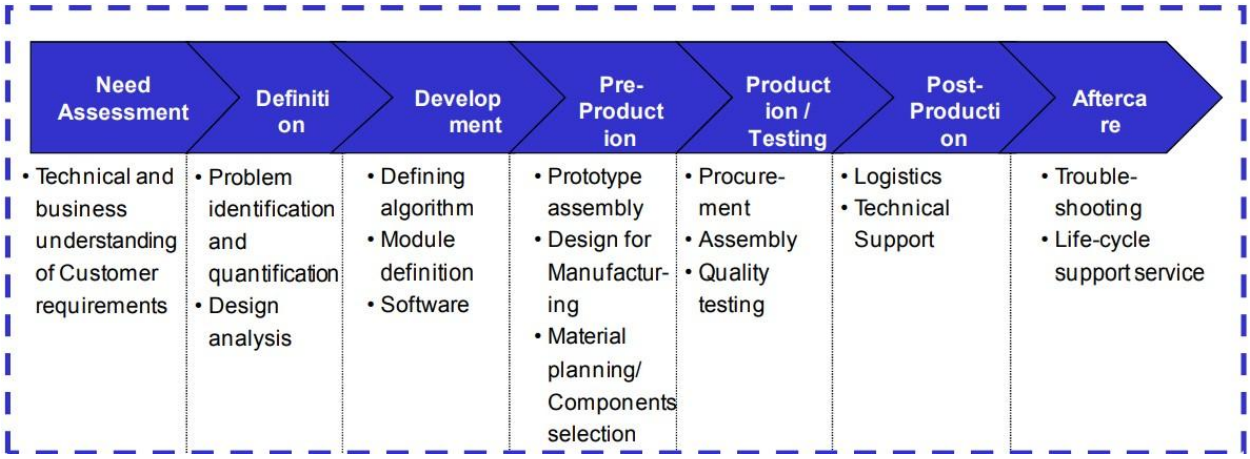


Figure : Core processes in the IT hardware sector

1. Need Assessment: For this, a cross functional team composed of people with background in engineering, manufacturing, materials management, and sales and marketing assess the technical and business aspects of the customer requirements. These are then taken up in further detail by the definition team.
2. Definition: Based on the need assessment, the engineering team works to define the customer application requirements and help finalise the exact hardware requirements/software configuration. The design concept is then chalked out considering the hardware configuration and cost estimates during the complete manufacturing process.
3. Development: With the aid of algorithm engineering, i.e. a combination of theoretical algorithm design with real-world data via a hardware device, verification of algorithm results and behaviour is ascertained. Definition module includes packaging of various units which have a well-defined interface with the other components, and developing the enabling software for hardware to function.
4. Pre-Production: The pre-production process begins with prototype assembly to identify the manufacturing process and finalise the product design. The design for manufacturing analysis is

performed to point out cost reduction and system improvements at the prototype level. Further, to achieve desired quality and price level, best available components are identified for the product design.

5. Production/Testing: Best-in-class components at competitive prices need to be procured from the existing/new vendors developed for the assembly process. Assembly services shall provide complete box-build and full system assembly for the computing hardware and finally conclude the production process with functional and system testing.
6. Post-Production: This includes arranging logistics and providing technical support for the product.
7. Aftercare: This includes post sales trouble-shooting and repair and maintenance. Managing and providing life-cycle support service for the product.

- **References**

1. Microsoft Word - MAIT Report on Manpower for eleectronics industry Novemberr v0 25 (meity.gov.in)

- **List the combined Requirements gathered from Background Reading/s**.

1) For manufacturing of Hardware devices, we need raw materials which are useful for production of the hardware. We can take the example of a computer system and mobile processor, for which we need ICs and circuits.
2) After the hardware manufacturing part is done ,we need to test it. So I can fix those issues. But even after testing some issues are still there. So we have to take the review of the customers. And work accordingly on those issues.
3) Hence we can give technical support, trouble-shooting and life cycle support service for the customer as an aftercare service.

2. Interview/s

Hardware Manufacturing Company/Interview Plan

System: Hardware Manufacturing System**(**HMS)

Interviewee: 1) Mahima Rohit **Designation**: CEO at HMS **Contact Details:**

Email : mahimarohit@gmail.com

Phone no: 9510345713

Organization Details: 676 , Near Home Town super market ,Bengaluru

Interviewer: 1) Rohan Shah **Designation**: Hardware & Networking Engineer

2) Jigar Patel **Designation**: Hardware Engineer

Date: 05/10/2021 **Time:** 12:00

Duration: 30 minutes **Place:** Google meet

Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding security of task management system

Agenda:

Problems with security
Problem with lack of management
Initial ideas
Follow-up actions

Documents to be brought to the interview:

documents relating to current security procedures

documents relating to the suggestion

Interview Summary

System: Hardware Manufacturing System

Interviewee: 1) Mahima Rohit **Designation**: CEO at HMS

Contact Details:

Email : mahimarohit@gmail.com

Phone No : 9510345713

Organization Details: 676 , Near Home Town super market,Bengaluru

Interviewer: 1) Rohan Shah **Designation**: Hardware and networking engineer **2)** Jigar Patel **Designation**: Hardware engineer

Date: 5/10/2021 **Time:** 12:00

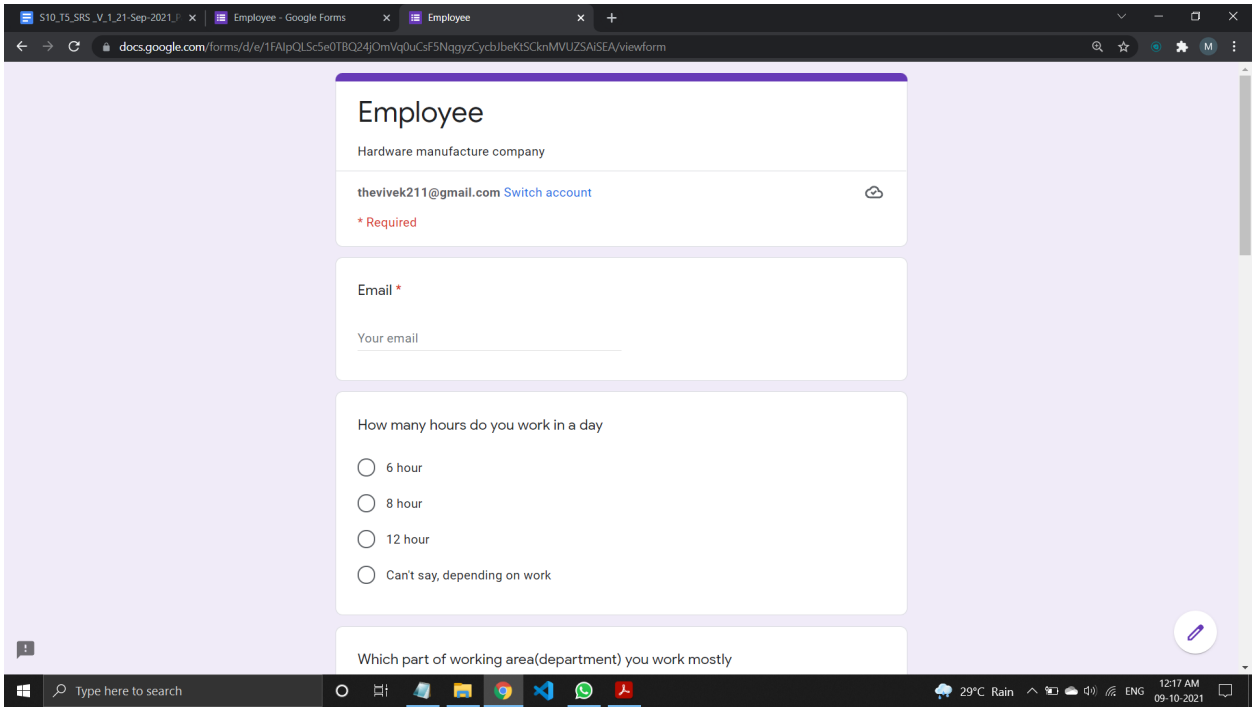
Duration: 30 minutes **Place:** Google meet

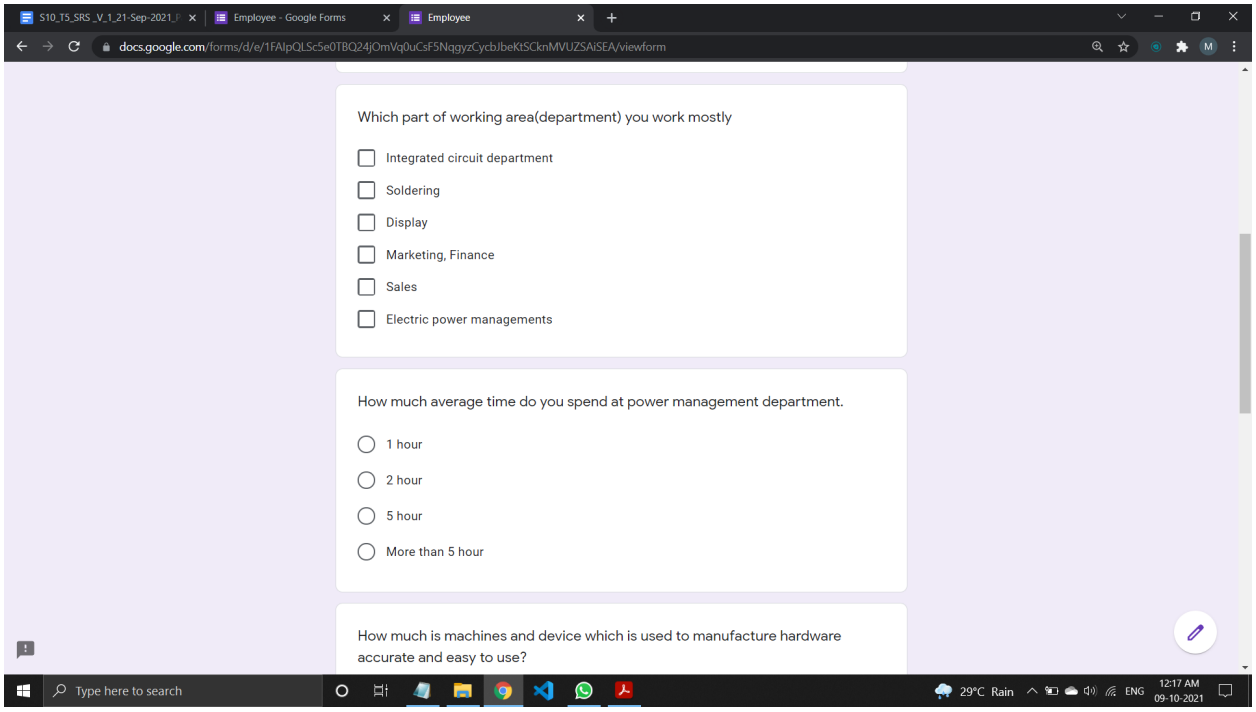
Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding security of task management systems.

- ★ Not very effective security in the system
- ★ Secure login
- ★ Some information is accessed by only the specified person and not by all
- ★ Priority of task

3) Questionnaire:





How much is machines and device which is used to manufacture hardware accurate and easy to use?

	1	2	3	4	5	
Very good	○	○	○	○	○	Very bad

Have you ever found safety issue, security issue in system?

- ○ Yes
- ○ No

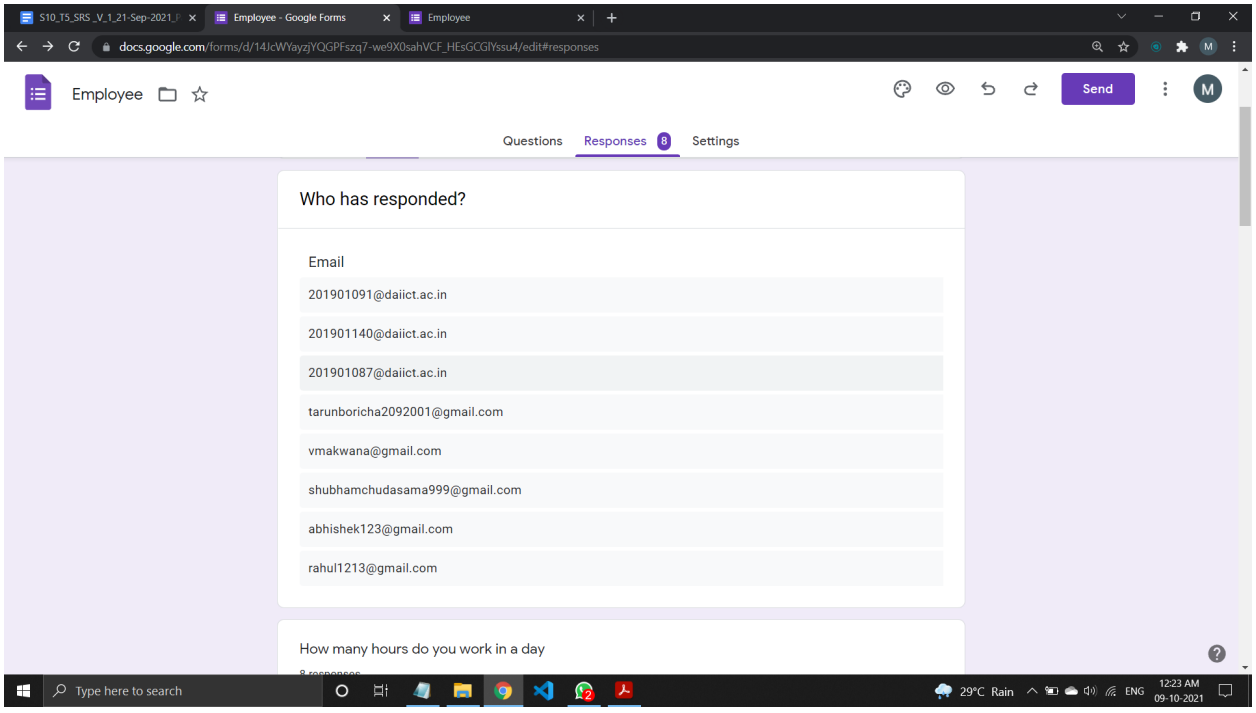
If you found any, write here without fail.

Your answer

If not found but want more security then also write here, give suggestion.

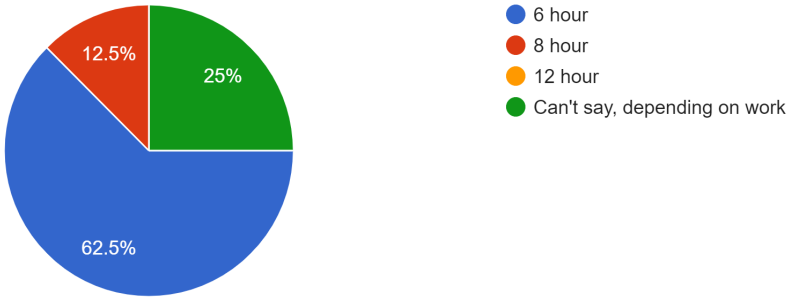
Your answer

Any other suggestion of comment



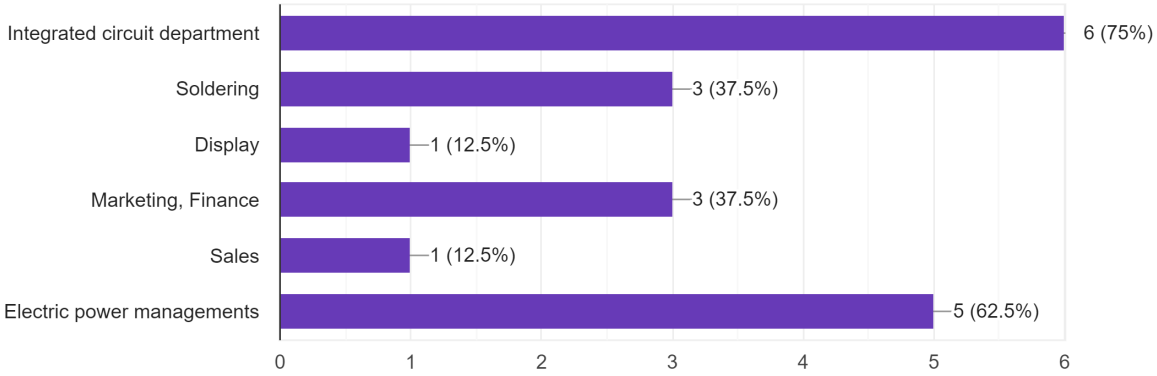
How many hours do you work in a day

8 responses



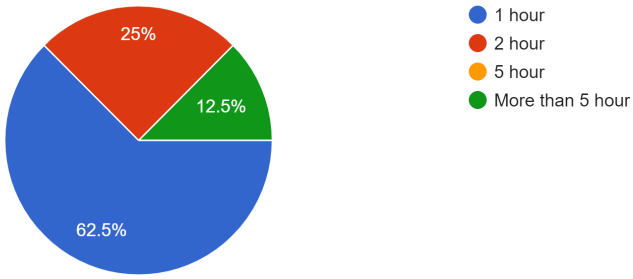
Which part of working area(department) you work mostly

8 responses



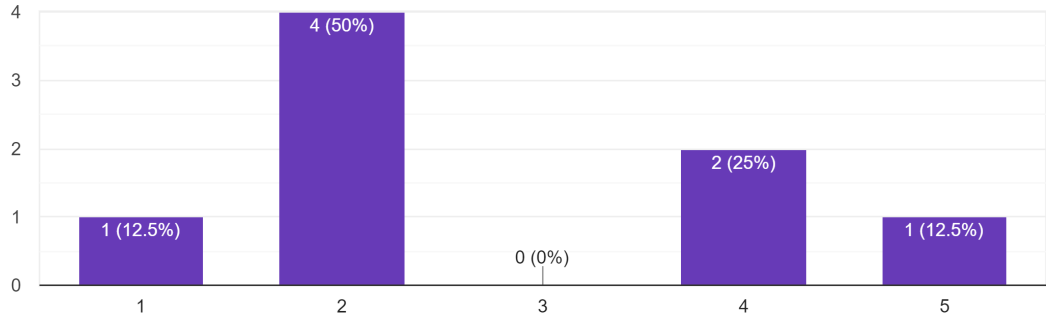
How much average time do you spend at power management department.

8 responses



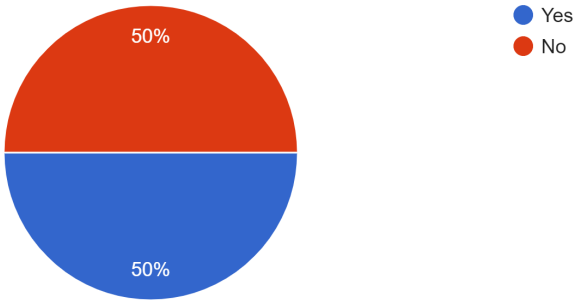
How much is machines and device which is used to manufacture hardware accurate and easy to use?

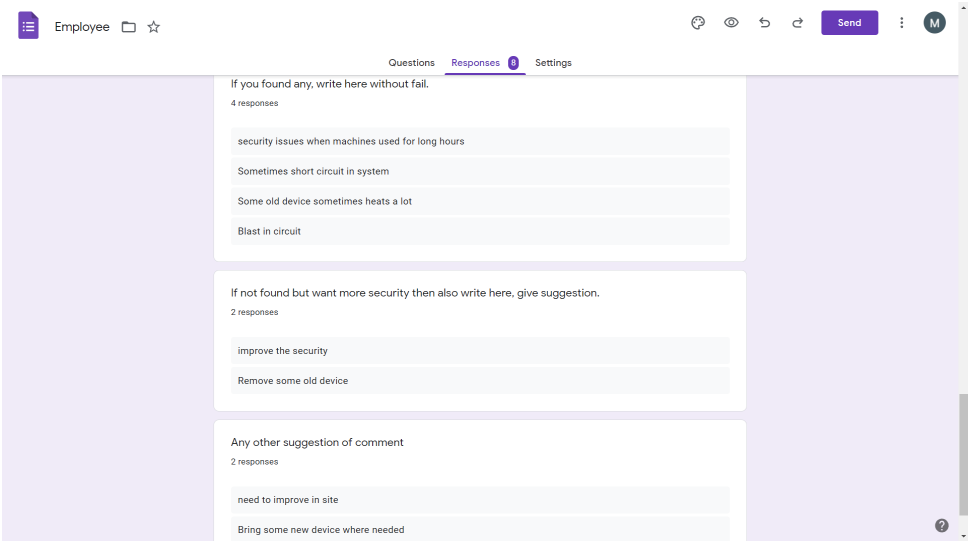
8 responses

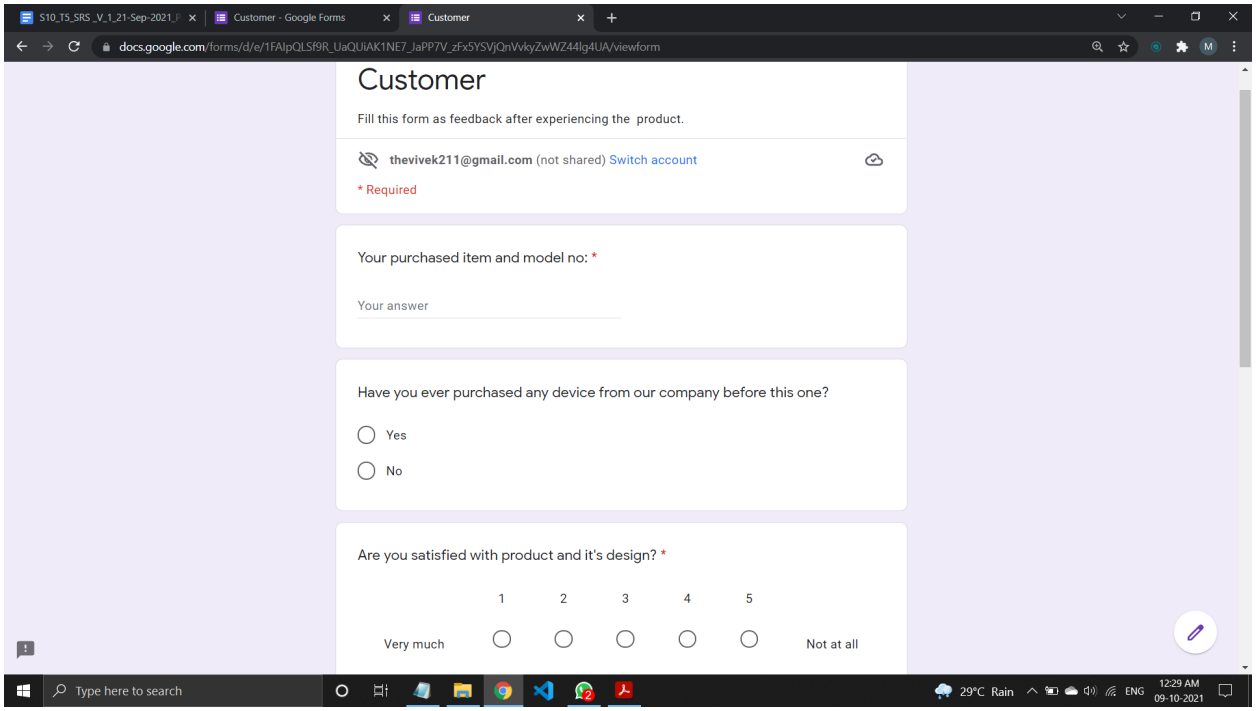


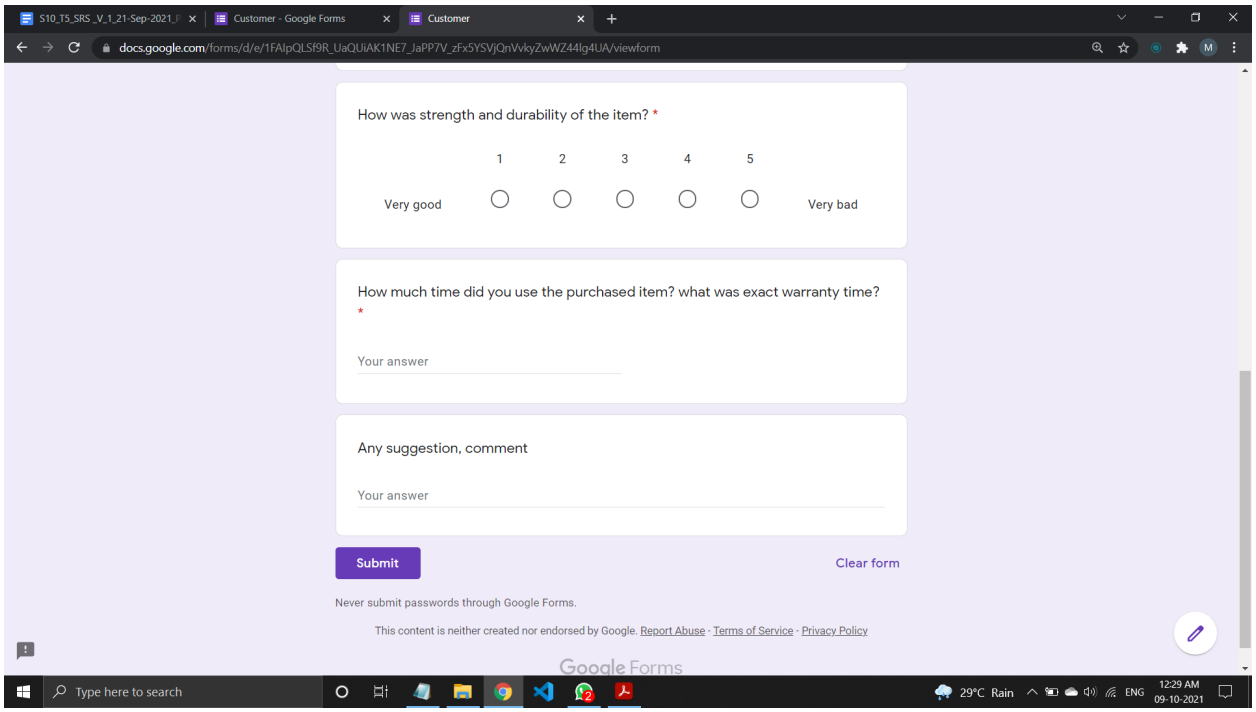
Have you ever found safety issue, security issue in system?

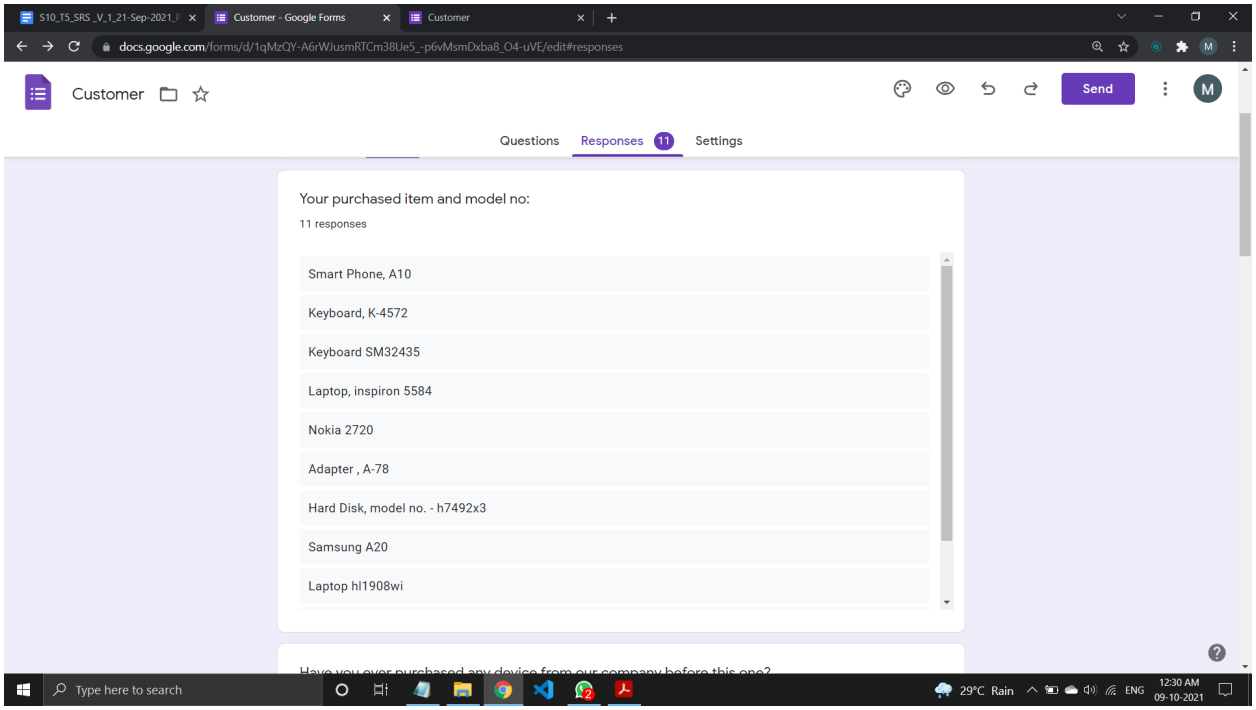
8 responses





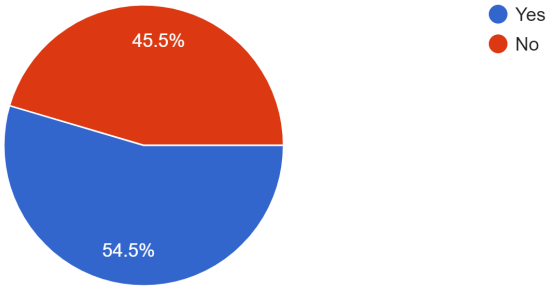

S10_T5_SRS _V_1_21-Sep-2021_F
Customer - Google Forms
Customer
docs.google.com/forms/d/e/1FAIpQLSf9R_UaQUiAK1NE7_JaPP7V_zFx5YSVjQnVvkyZwWZ44lg4UA/viewform
Customer
Fill this form as feedback after experiencing the product.
thevivek211@gmail.com (not shared) Switch account
* Required
Your purchased item and model no: *
Your answer
Have you ever purchased any device from our company before this one?
Yes
No
Are you satisfied with product and it's design? *
1 2 3 4 5
Very much
Not at all
Type here to search
29°C Rain
ENG
12:29 AM
09-10-2021


S10_T5_SRS _V_1_21-Sep-2021_F
Customer - Google Forms
Customer
docs.google.com/forms/d/e/1FAIpQLSf9R_UaQUiAK1NE7_JaPP7V_zFx5YSVjQnVvkyZwWZ44lg4UA/viewform
How was strength and durability of the item? *
1 2 3 4 5
Very good
Very bad
How much time did you use the purchased item? what was exact warranty time? *
Your answer
Any suggestion, comment
Your answer
Submit
Clear form
Never submit passwords through Google Forms.
This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy
Google Forms
Type here to search
29°C Rain
ENG
12:29 AM
09-10-2021



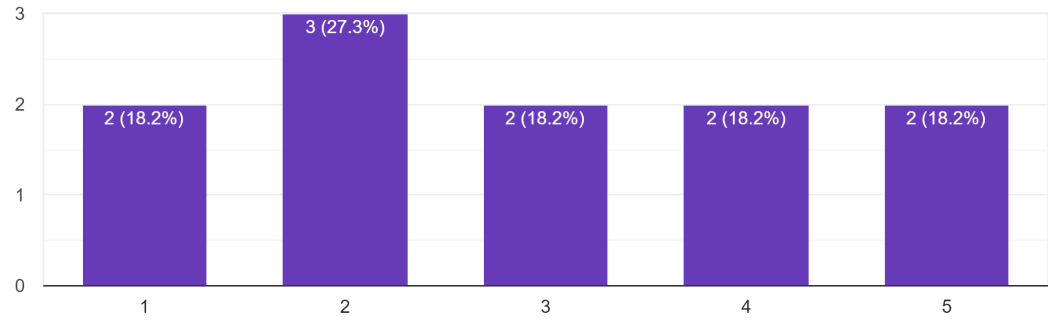
Have you ever purchased any device from our company before this one?

11 responses



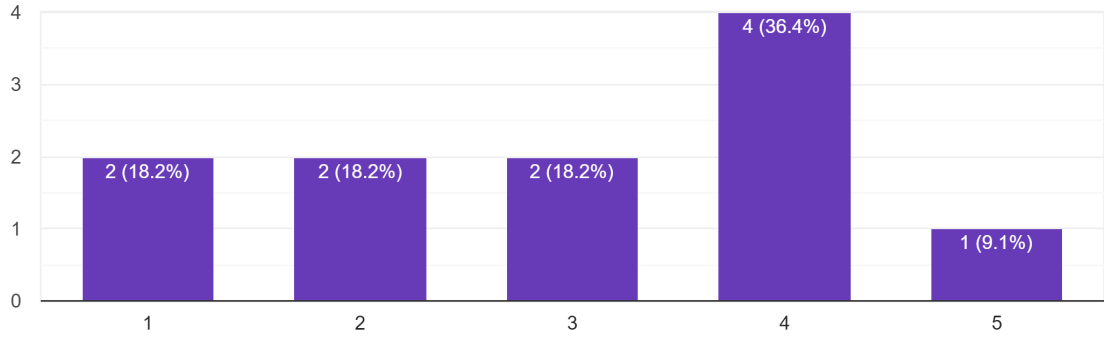
Are you satisfied with product and it's design?

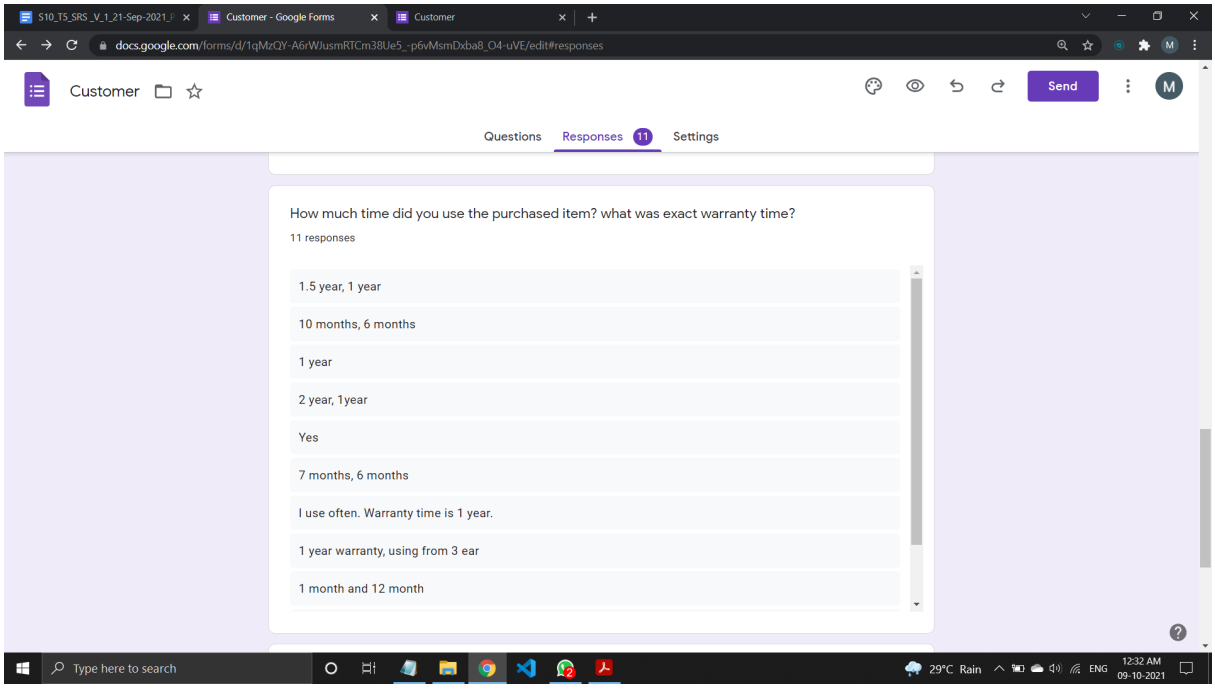
11 responses

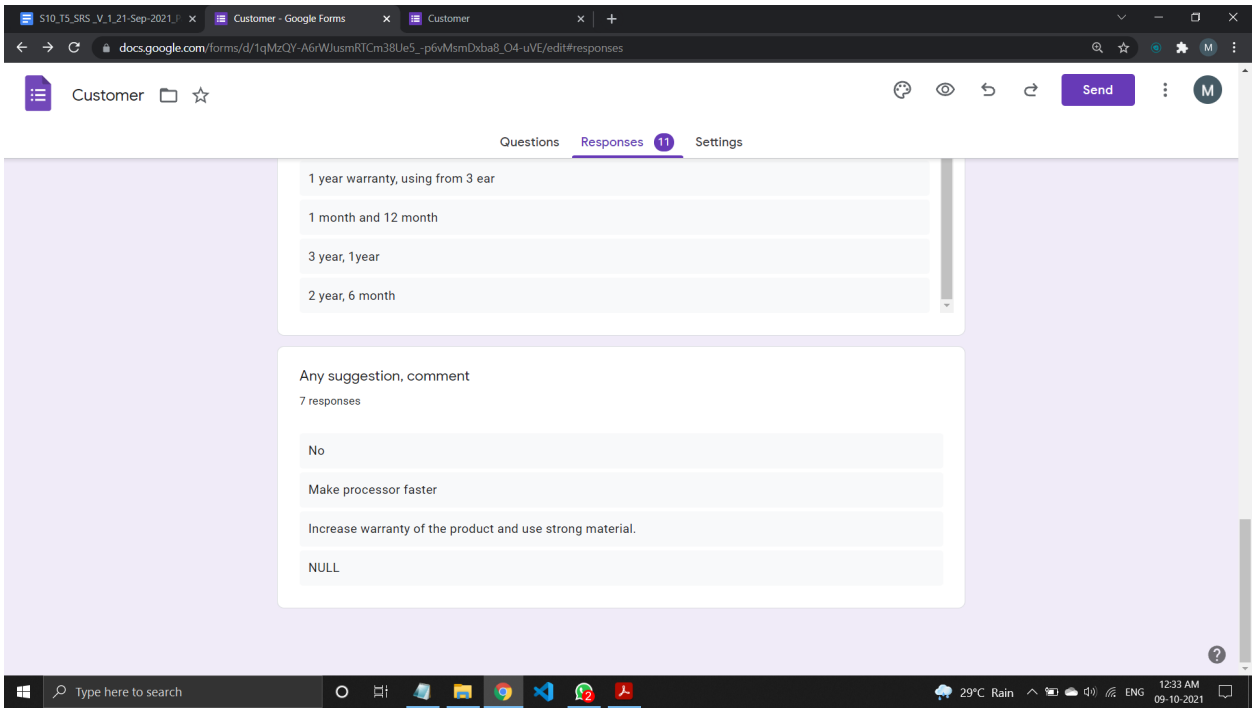


How was strength and durability of the item?

11 responses







Requirements:

There are some complaints from employees mostly to increase security of the system. Machine heat, short circuit, old device not working properly. This requires taking it into consideration very first as soon as possible.

From the customer side they want more time in warranty, aspect faster processor. This needs to improve.

4. Observations: -

Hardware Manufacturing Company: Observations

System : Hardware Manufacturing System

Project Reference : CJ/WI/2019/07

Observations by: Tarun (Company employee)

Date: 10/09/2019 **Time:** 11:30

Duration: 30 minutes **Place:** Head Office, Mumbai

Observations:

1. Need to improve in the response and needful action time of feedback for customers.
2. Need to improve product hardware efficiency.
3. Need to upgrade some devices in the working area: computer, printer and security camera.
4. Products like mobile processors need 5nm base architecture to improve their power efficiency and improved performance.

[C]. Fact Finding Chart

Objective	Technique	Subject(s)	Time commitment
To get background on Hardware related companies and their strategy of management	Background Reading	Hardware company sites like intel, snapdragon and mediatek and some articles	4 hour
To determine customer feedback	Questionnaire	Google form	2 days
To determine employee feedback	Questionnaire	Google Form	2 days
To solve problem of lack of security	Interview	Role play	1 hour
Summary	Observation	All above	1 hour

[D]. List Requirements

We can collect all the required information regarding the project assigned and we can work on it on paper.After the complete task is done on paper ,we can give it to the system we are working on.This is helpful when we have some chances of changes in the information.

- **Details about security :** The security of company related information like on the current project the company is working on is a very important aspect nowadays.So it is required to provide security to the information we have and the security about the project ideas.

- **Details of Product :**We can start our task after we get all the information about the project.It is good if we can collect all the data and information that we are going to use in the task.

- **Employees Details :**To assign the task with no ambiguity ,we can simply assign the task to the employees ,if we have the necessary information about the employee like subject of expertise and the work experience.It helps a lot when we are assigning the task.

- **Customer Details :**Whenever the issue is raised by the customer for the product ,we need to gather some details of the customer.So we can easily commute with them.It is our responsibility to provide the security to the customers details.

- **Update on the Product :**After we took an action on the product for the further development ,we need to track those changes.So in future, whenever the previous version is needed we can easily get access to that product.

[E]. User Classes and Characteristics

Manufacturing is the basic building block of any organization and the primary reason why an organization exists. Manufacturing is the process that involves crafting of a product as per the company demand. Manufacturing is not only the work of labor but requires a number of professionals at management level who work hard to ensure the smooth running of the entire manufacturing process.

The major levels in the manufacturing management hierarchy are described as designations from the highest manufacturing management level to the lowest manufacturing management level.

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

- **Chief Executive Officer :** The highest management head of the manufacturing company or department is always the CEO or simply saying Chief Executive Officer of the company. Entire work associated with manufacturing from planning to implementing and executing is done under the supervision of this professional who has years of vital experience of the field and is an integral part of the company.

- **Project Manager :**A project manager in the manufacturing management hierarchy is the associate professional that works in compilation with the manufacturing head on a specific project or for all the projects. Project managers can have the liability of taking care of the planning, implementation and closing of any manufacturing project in a company.

- **Senior Engineers :** Next in the manufacturing management hierarchy are the senior engineers who undertake the accountability of the management of the development, designing and manufacturing of the product being manufactured. They play a fundamental yet imperative role in the manufacturing management hierarchy.

- **Architect :**Architects are the skillful professionals who are responsible to finalize the final layout of the product to be manufactured. In some companies they are referred to as designers depending upon the work being tackled.

- **Head of Machinery Operations :**The machinery and tasks associated with the tools, equipment, machinery is under the supervision of these management professionals.
- **Head of the Sales** – The sales head supervises the promotion as well as sale of the manufactured product in order to work for the growth of the company.

- **Junior Engineers** – These professionals in the manufacturing management hierarchy play an important role since they perform the ground level work for the senior officials.

[F]. Operating environment :

1). Computer windows, mac
2). Database to manage all produced products.
3). Wifi router for connectivity.
4).- Soldering machine.
- Automatic IC Forming Machine
- High Precision Double Spindles CNC Making Machine with Knife Magazine for Mobile Phone Tempered Glass
- Making Machine Mobile Phone Battery Making Machine Pouch Cell Making
- 300t High Speed Plastic Mobile Phone Case Making Machine Injection Molding Machine Machinery Price
- Mobile Phone Touch Screen Making Machine, Automatic Mini Vacuum Lcd
- 180t Horizontal Cold Chamber Aluminum Die Casting Machine for Making Hardware
- High Precision Double Spindles CNC Making Machine with Tool Magazine for Hardware Fittings and Accessories

[G]. Product Functions :

Providing information about the below objects :

- ★ Details about employees of the company
- ★ Details about project managers of the company
- ★ Projects details
- ★ Task details
- ★ Assignment of task
- ★ Keep track of the task progress
- ★ Comment on the task
- ★ Customer details

[H]. Privileges

- ★ Details about Employees of the company
  Read - Some of the details can read all and some of the details can read only authorized person of the company
  Update : Only authorized person or that person can update

- ★ Details about project managers of the company
  Read/Update/Delete - Only authorized person can read and Update this details

- ★ Project Details

Read : All Employees of the company can read this details
Update : Only project manager can update this details

★ Task details
Read / Update : All can read and update this details

★ Assignment of task
Read : All can read
Update : Only project manager can update

★ Keep track of the task progress
Read : All can read
Update : Only the person who works on that task can update

★ Comment on task
Read : All can read
Update : Project manager or the person who works on that task can update

★ Customer Details
Read : All can read
Update : Only customer can update

[I]. Assumptions

1. Quick implementation in new projects.
2. Companies provide proper functionality to their employees for new projects.
3. Head office quickly gives feedback on working projects.
4. Customer's problems are solved quickly.
5. Employees can update databases very easily.

[J]. Business constraints:

Business constraints is a factor which affect negatively in the growth of the business.
1). Hardware products usually depend on many suppliers, materials, electronic components, manufacturing, etc. These things are not something you will make by yourself. Sometimes Raw materials are not available or available at high prices. So these dependencies could make hardware businesses inefficient.

2). Hardware requires a whole different set of skills in engineers. It is essential to have the right number of people with the appropriate skills to enable the business to achieve its business objectives, sometimes hard to hire skilled employees.
3). Hardware businesses suffer from high costs and low margins.
4) Companies need to make themselves powerful by time to time in order to stay in the market and in competition.
5) It is important to identify the nature of customers and their requirements through market research.
6) Common business constraints include budget and time restrictions, resource limitations, and resource skill limitations,size of the market,quality of direction and management,transportation costs, money management.

Section 2
Noun Analysis

Table 1 :

All Extracted Nouns &
Verbs from Problem Description

Nouns	**Verbs**
employee	Assigned
Project	Creating
Task	Working
Company	Located
Product	Manufacturing
Skills	Required
Security	Ensuring
Staff	Allocate
Name	Managed
Work	Assigned
Safety	Resolved
Salary	Allotted
Date	Sentencing
System	Edit
Database	Maintained
Gender	Recorded
Attributes	Associated
Information	Included
Number	Required
Date of birth	Recorded
Managers	Managing
Hardware	Required

Software	Required
ID	Provided
Age	Recorded
Department	Handling
Expertise	Needed
Phone number	Required
Data security	Authorised
Login	Required
User name	Information
Password	Protected
Customer	Using
Customer id	Required
Item	Production
Memory	Useful
System	Regarding
Processor	Working
Raw materials	Required
Components	Making
Feedback	Useful
Safety	Required
Issues	Resolved

Employee ID	Required
Project Ideas	Information
Working Hours	Assigned
Marketing Strategy	Information
Payment Details	Information
Project Details	Required
Project type	Allocated
Assigned Employee	Assigned
Project ID	Required
Documents	Information
production	completed
Taxes	Pay

Table 2 :

Accepted Noun & Verbs list

Candidate entity Set	**Candidate attribute set**	**Candidate relationship set**
User	user_id user_password user_employeeID user_AuthenticationMode	Required

Project	Project_id Project_title Project_startTime Project_endTime project_status	Assigned
Task	task_id task_name task_employeeID task_projectTitle start_time end_time task_status	Allotted
Customer	customer_ID customer_name selected_item cutomer_contact	Purchased by
Employee	Employee_ID Employee_name Employee_address Date_of_birth Employee_phoneNo Employee_email	Working In

Product	product_ID product_name Manufacturing_date Warranty Total_No_product	Produced
Payment	payment_id customer_id payment_amount payment_date payment_mode	To be paid
Customer feedback	product_id customer_id customer__name selected_item feedback_description	feedback
Company	Company_id Company_name Contact Address	Data
	team_id	

Team	team_name No_of_employee project_id	Assign
Manager	Department Office_address	Manage

Table 3:

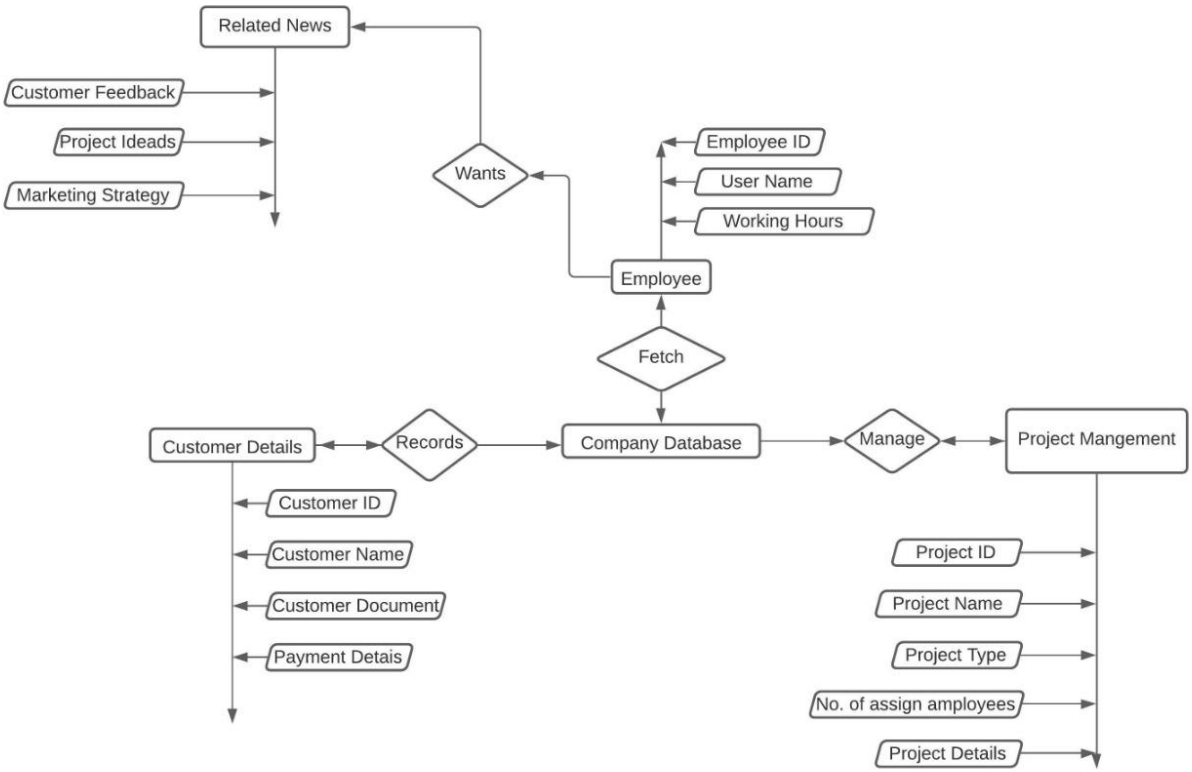
Rejected Noun & Verbs list from Problem Description

Noun	**Reject Reason**
Online	Irrelevant
Date	General
System	Irrelevant
Database	Irrelevant
Name	General
Information	Vague
Safety	Duplicate
Staff	General
Relation	General
Number	General
Attributes	General
System	Duplicate

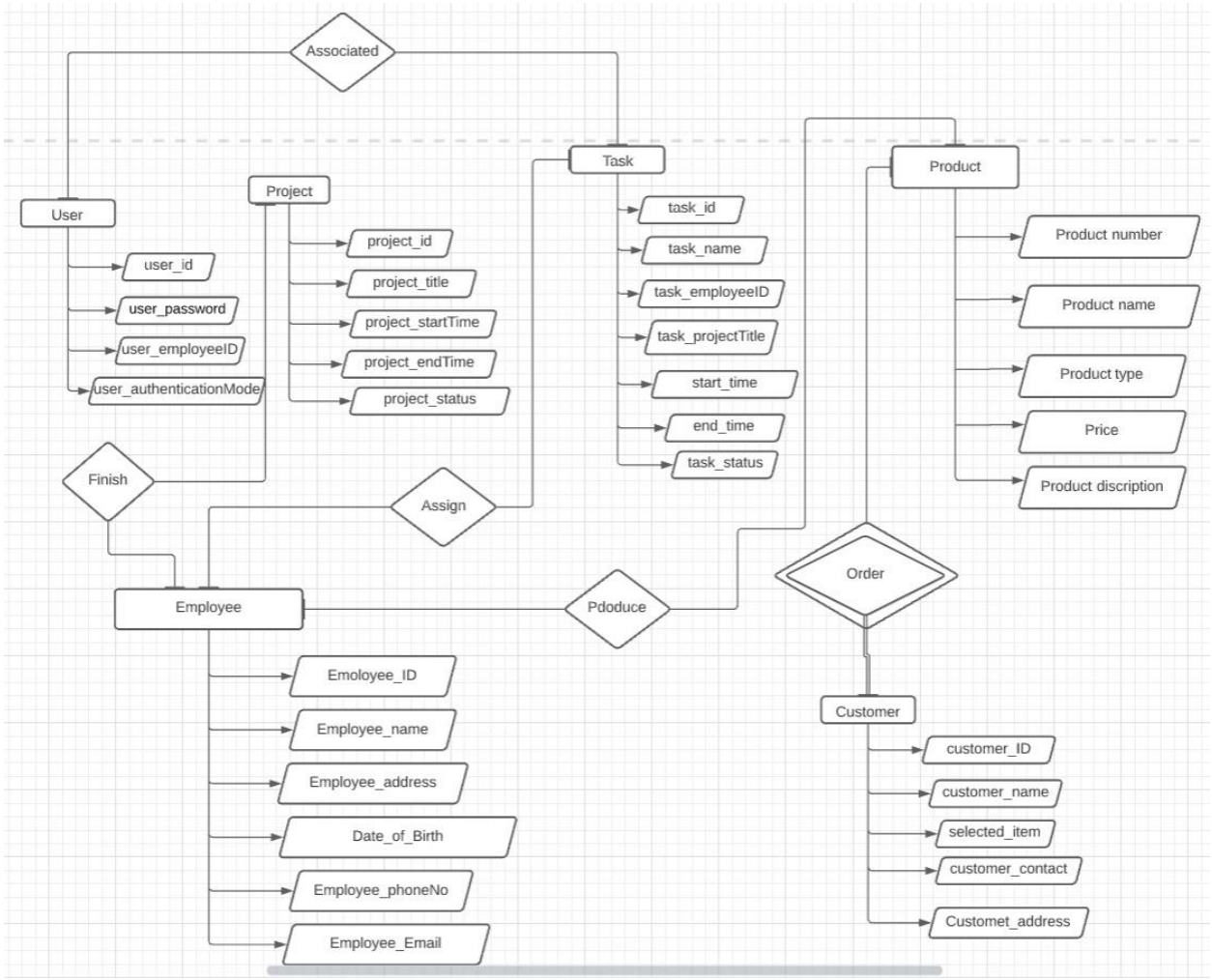
Section 3
ER-Diagrams all versions

II. Develop the ER Diagram (ERD)

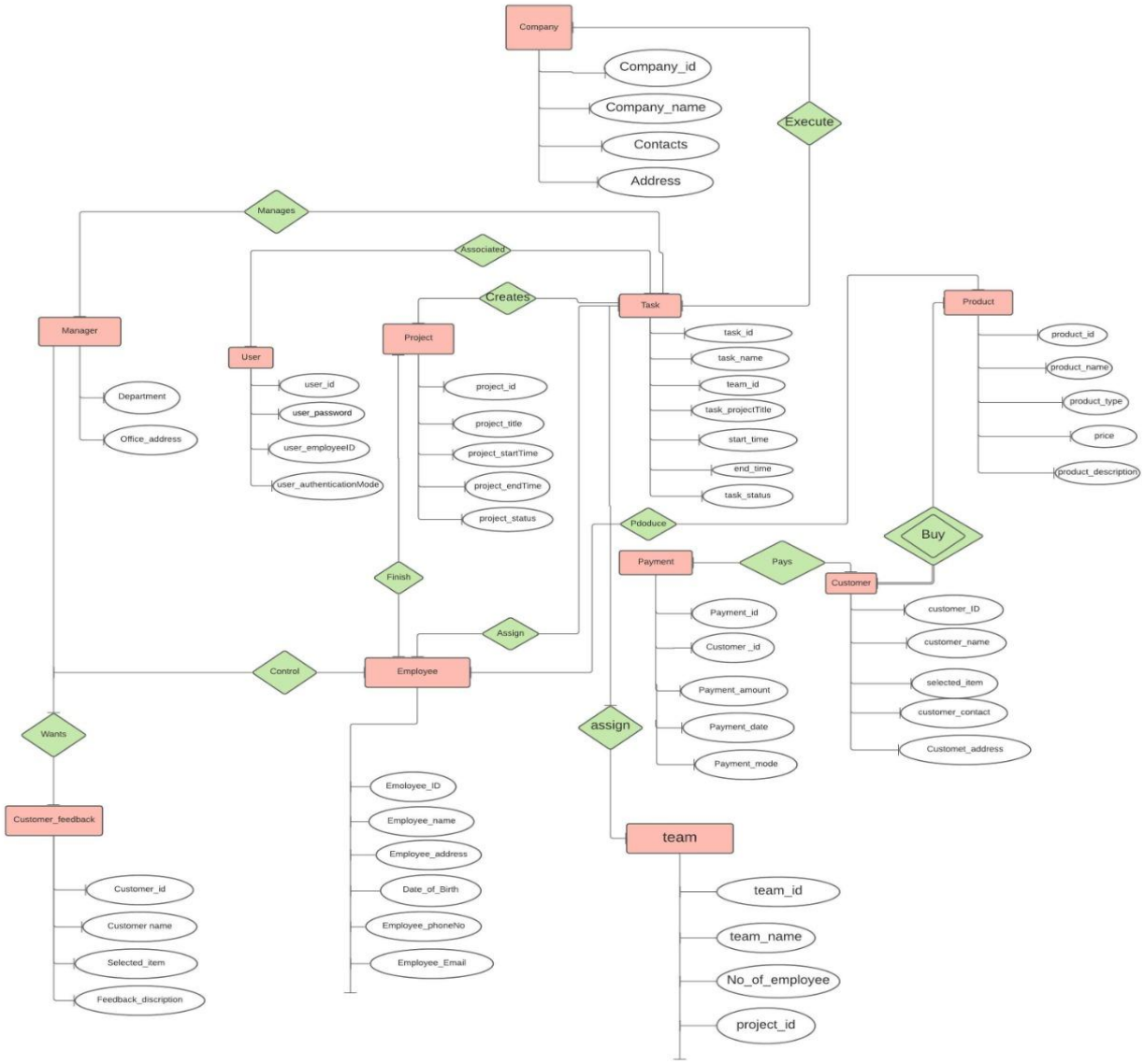
- **Version 1: -**



Version 2: -


Associated
User
user_id
user_password
user_employeeID
user_authenticationMode
Project
project_id
project_title
project_startTime
project_endTime
project_status
Task
task_id
task_name
task_employeeID
task_projectTitle
start_time
end_time
task_status
Product
Product number
Product name
Product type
Price
Product discription
Finish
Assign
Order
Employee
Pdoduce
Emoloyee_ID
Employee_name
Employee_address
Date_of_Birth
Employee_phoneNo
Employee_Email
Customer
customer_ID
customer_name
selected_item
customer_contact
Customet_address

Version 3: -


Company
Company_id
Company_name
Contacts
Address
Execute
Manages
Associated
Creates
Manager
Department
Office_address
User
user_id
user_password
user_employeeID
user_authenticationMode
Project
project_id
project_title
project_startTime
project_endTime
project_status
Task
task_id
task_name
team_id
task_projectTitle
start_time
end_time
task_status
Product
product_id
product_name
product_type
price
product_description
Pdoduce
Buy
Payment
Pays
Customer
customer_ID
customer_name
selected_item
customer_contact
Customet_address
Payment_id
Customer _id
Payment_amount
Payment_date
Payment_mode
Finish
Assign
Control
Employee
Wants
assign
Customer_feedback
Customer_id
Customer name
Selected_item
Feedback_discription
Emoloyee_ID
Employee_name
Employee_address
Date_of_Birth
Employee_phoneNo
Employee_Email
team
team_id
team_name
No_of_employee
project_id

Section 4
Conversion of Final ER-Diagram to Relational Model

Mapping E-R Model to Relational Model: -

• Company(company_id, company_name, contacts, address)

• User(user_id, user_password, user_employeeID, user_authenticationMode)

• Manager(id, department, office_address)

• Employee(employee_id, employee_name, employee_address, date_of_birth, employee_phoneNo, employee_email)

• Project(project_id, project_title, project_startTime, project_endTime, project_status)

• Team(team_id, team_name, no_of_employee, project_id)

• Task(task_id, task_name, task_employeeID, task_projectTitle, task_status, task_startTime, task_endTime)

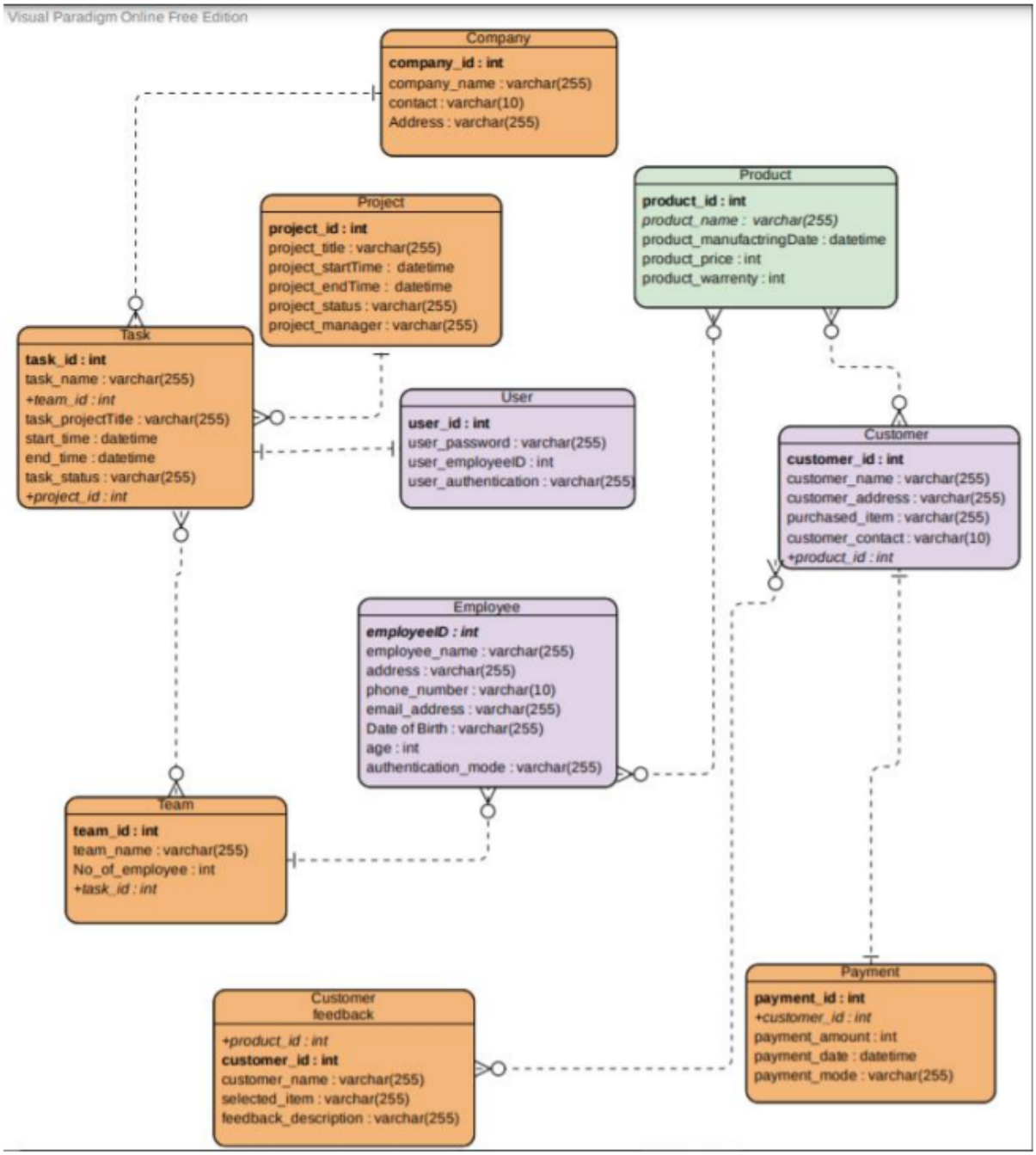
• Product(Product_numer, Product_name, Product_type, Product_price, Product_description)

• Costumer(customer_id, customer_name, selected_item, customer_contact, customer_address)

• Payment(payment_id, paymet_amount, Payment_date, customer_id, payment_mode)

• Customer Feedback(customer_id, customer_name, selected_item, feedback_description)

RELATIONAL SCHEMA: -



Section 5
Normalization and Schema Refinement

Normalization & Schema Refinement :

1. Company(<u>company_id</u>, company_name, contacts, address)

PK dependency : company_id -> company_name, contacts, address

FD : company_id -> company_name
company_id -> contacts
company_id -> address

Partial Key Dependency : None

Transitive Dependency : None

Redundancies : if a company has two different contacts then there will be redundancy in the table.

Anomalies: -

Insert: - we cannot insert company_name, contacts, address without company_id.

Deletion: - There won't be any anomaly in deletion.

Update: - There won't be any anomaly in update

1NF:- The table is not in 1NF because there can be multiple contact numbers. To remove this company_id, company_name, address will be repeated.

2NF: - there is no partial dependency so the table is already in 2NF.

3NF: - there is no transitive dependency so the table is already in 3NF.

2. User(<u>user_id</u>, user_name, user_password)

PK dependency : user_id -> user_password, user_name
FD : user_id -> user_password
user_id -> user_name

Partial Key Dependency : None

Transitive Dependency : None

Redundancies : None.

Anomalies: -

There won't be any anomaly in insert,update and deletion.

1NF:- There are no multi-valued attributes here, so it is already in 1NF

2NF: - there is no partial dependency so the table is already in 2NF.

3NF: - there is no transitive dependency so the table is already in 3NF.

3. Employee(employee_id, employee_name, employee_address, date_of_birth, employee_contact, employee_email)

PK dependency :
{employee_id} -> employee_name, employee_address, date_of_birth, employee_contact, employee_email}

FD : employee_id -> employee_name
employee_id -> employee_address
employee_id -> date_of_birth
employee_id -> employee_contact
employee_id -> employee_email

Partial Key Dependency : here partial dependency occurs because if an employee has two contacts then there will be the same employee_email with different contacts.

Transitive Dependency : None

Anomalies:

Insert: - there is no anomaly in insert.

Update: - if we update email id then it will occur more than once because the same employee has two different contacts.

Delete: - there is no anomaly in delete.

1NF: - if an employee has two different addresses like current and permanent then there will be 2 values in the same attribute. So we need change primary key (employee_id employee_contact)

Employee(employee_id,employee_contact employee_name, employee_cur_address, employee_per_address, date_of_birth, employee_email)

2NF: -
There is partial dependency in the table so we need to divide this table.

Employee(employee_id,employee_contact)
Employeeinfo(employee_id, employee_name, employee_cur_address, employee_per_address, date_of_birth, employee_email)

Here employee_id in employeeinfo table is foreign key.

3NF: - there is no transitive dependency in the table so it is already in 3NF.

So now it is on BCNF.

4. Project(project_id, project_title, project_startTime, project_workstatus, project_endTime)

PK dependency : project_id -> project_title, project_startTime, project_workstatus, project_endTime

FD : project_id -> project_title
project_id -> project_startTime
project_id -> project_workstatus
project_id -> project_endTime
Project_status -> project_endTime

Partial Key Dependency : None

Transitive Dependency : Project_status can find project end time which is transitive dependency.

Redundancies : Some projects are starting or ending at the same time so information will repeat.

Anomalies: -

Insert: - we can not insert project_endTime without inserting project_startTime.

Update: - There is no update anomaly.

Delete: - There is no delete anomaly.

1NF:- There are no multi-valued attributes here, so it is already in 1NF

2NF: - there is no partial dependency so the table is already in 2NF.
3NF: - there is transitive dependency in the table so we need to divide this table.

Project(project_id, project_title, project_startTime)
Projectinfo(Project_id, project_workstatus, project_endTime)

Now it is already in BCNF.

5. Team(team_id, team_name, no_of_employee, project_id)

PK dependency : team_id -> team_name, no_of_employee, project_id

Here project_id is foreign key

FD : team_id -> team_name
team_id -> no_of_employee
team_id -> project_id

Partial Key Dependency : None

Transitive Dependency : None

Redundancies : None

Anomalies: -

Insert: - there is no anomaly in insert.

Update: - there is no anomaly in the update.

Delete: - if we delete project_id then all project details will lost

1NF:- There are no multi-valued attributes here, so it is already in 1NF

2NF: - there is no partial dependency so the table is already in 2NF.

3NF: - there is no transitive dependency so the table is already in 3NF.

Hence table is in BCNF

6. Task(task_id, task_name, employee_id, task_status, task_startTime, task_endTime)

PK dependency : task_id -> task_name, employee_id, task_status, task_startTime, task_endTime

FD : task_id -> task_name
task_id -> task_employeeID
task_id -> task_projectTitle

task_id -> task_status
task_id -> task_startTime
task_id -> task_endTime
task_status -> task_endTime

Partial Key Dependency : There won't be any partial dependencies.

Transitive Dependency : There is transitive dependency that we can find on task_endTime by current task_status.

Redundancies : There can be multiple employees working on the same task. It can create redundancy.

Anomalies:-

Insert:- If the next task is supposed to be done on a particular day, this can't be inserted in tables without task_id.
Delete:- There is no delete anomali.
Update:- There is no update anomaly

1NF:- There can be multiple employees working on the same task. To Convert in 1NF task_id, task_name, task_status, task_startTime, task_endTime will be repeated.

2NF:- Table already in 2NF.

3NF:- There is transitive dependency from task_status to task_endTime, so we have to create another table containing task_id, task_status, task_endTime like this,
task_completion(task_id, task_status, task_endTime)
Task(task_id, task_name, employee_id, task_startTime)

7. Product(Product_id, Product_name, Product_type, Product_price, Product_description)

PK dependency : Product_id -> Product_name, Product_type, Product_price, Product_description

FD : Product_id -> Product_name
Product_id -> Product_type
Product_id -> Product_price
Product_id -> Product_description
Product_name -> product_description
Product_name -> product_price

Partial Key Dependency : There is no partial dependency.

Transitive Dependency : There is transitive
Dependency. Because from product_name we can get product_price and product description.

Redundancies : Product name, price and description can be repeated for same product.

Anomalies:-

Insert: We can't add product description and product type without id, even if we know product description before production.
Delete: There is no delete anomaly.
Update: To update a price or description we need to change all price and description.

1NF: Table is already in 1NF.
2NF: It is also in 2NF.
3NF: There is transitive dependency so we need to remove that to convert into 3NF.
Product_name -> product_description, product_price
Product_id -> Product_name, Product_type

8. Customer(<u>customer_id</u>, customer_name, selected_item, customer_contact, customer_address)

PK dependency : customer_id -> customer_name, selected_item, customer_contact, customer_address
(Customer_name, customer_contact) -> customer_id, selected_item, customer_address.

FD : customer_id -> customer_name
customer_id -> selected_item
customer_id -> customer_contact
customer_id -> customer_address
(Customer_name, customer_contact) ->
Customer_id
(Customer_name, customer_contact) ->
selected_item
(Customer_name, customer_contact) ->
customer_address

Partial Key Dependency : None

Transitive Dependency : There is no transitive dependency.

Redundancies : If customers purchase more than one item then other information will repeat.

Anomalies :-

Insert:- can't insert product item without customer's id or name, contact.
Delete:- There is no deletion anomaly.
Update:- If a customer's contact or address will change then it will update on all purchased items.

1NF:- Table is not in 1NF as there can more than one value in purchased item and contact number, to convert in 1NF we have to repeat customer_id, customer_name, customer_address
2NF:- There is no partial dependency, so the table is in 2NF.
3NF: There is no transitive dependency so it is in 3NF.

9. Payment(<u>payment_id</u>, payment_amount, Payment_date, <u>customer_id</u>, payment_mode)

PK dependency : payment_id -> payment_amount, Payment_date, customer_id, payment_mode

FD : payment_id -> payment_amount
payment_id -> payment_date
payment_id -> customer_id
payment_id -> payment_mode
(customer_id, payment_mode) -> payment_id
(customer_id, payment_mode) ->
payment_date
(customer_id, payment_mode) ->
Payment_amount
customer_id -> payment_id
customer_id -> payment_amount
customer_id -> Payment_date

Partial Key Dependency : There is partial dependency as customer_id can find payment_id, payment_amount and payment_date. To remove this we can make new tables.

Transitive Dependency : There is no transitive dependency.

Redundancies : There is no redundancy.

Anomalies:-

Inset:- No insertion anomalies.
Delete:- No delete anomalies.
Update:- No updation anomalies.

1NF:- Table is already in 1NF as there is no multivalued attributes.
2NF:- It is not in 2NF, there is partial dependency so we make new table as follows,
Payment_id -> customer_id
Payment_id -> payment_amount, payment_date, payment_mode.
3NF:- Table is in 3rd normal form already as there is no transitive dependency.

10. Customer Feedback(<u>customer_id</u>, <u>product_id</u>, selected_item, feedback_description)

PK dependency : customer_id -> customer_name, selected_item,feedback_description

FD : customer_id -> customer_name

customer_id -> selected_item
customer_id -> feedback_description
Item_id -> selected_item

Partial Key Dependency : Here selected item is non prime attribute and it is functionally dependent on customer_id.
Transitive Dependency : There is no dependency.
Redundancies : Lot's of customers find the same item so selected_item and product_id are the same for many customers.

Anomalies:-

Insert:- can't insert customer_name, selected_item, Item_id, feedback_description without customer_id.
Delete:- No delete anomalies.
Update:- No update anomalies.

1NF:- It is already in 1NF.
2NF:- For this we divide this table into two table,

Customer_id- > product_id
Customer_id -> selected_item, feedback_description
3NF:- already in 3NF. Yet it is in BCNF.

Section 6
SQL

Final schema

1. **company_id** -> { company_name, contacts, address }
   Key:- company_id

2. **user_id** -> {user_password, user_name }
   Key:-user_id

3. **Employee_id, employee_contact** -> { employee_id, employee_contact}
   Key:- employee_id, employee_contact

4. **employee_id** -> { employee_name, employee_cur_address,employee_per_address, date_of_birth, employee_email}
   Key:- employee_id

5. **Project_id ->** {project_title, project_startTime}
Project_id -> project_workstatus,
project_endTime

   Key :- project_id

6. **Team_id, project_id** -> { team_name, no_of_employee, project_id}
   Key :- team_id

7. **task_id** -> { task_name, employee_id, task_startTime }
   task_id -> {task_status, task_endTime,)

   Key :- task_id

8. **Product_id** ->{ Product_name, Product_type}
   Product_name -> product_description, product_price
   Key :- Product_id

9. **customer_id** -> { customer_name, selected_item, customer_contact, customer_address }
   Key :- customer_id

10. **Payment_id** -> payment_mode
    customer_id -> payment_id, payment_amount, payment_date.
    Key :- payment_id, customer_id

11. **Customer_id**- > product_id
    Customer_id -> selected_item, feedback_description

    Key :- customer_id, product_id

DDL Script: -

1. **Company: -**

```
CREATE TABLE IF NOT EXISTS prct."Company"
(
    company_id bigint NOT NULL,
    company_name character varying(255),
    contacts bigint,
    address character varying(255),
    PRIMARY KEY (company_id)
);
```

2. **User: -**

```
CREATE TABLE IF NOT EXISTS prct."User"
(
    user_id bigint NOT NULL,
    user_name character varying(255),
    user_password character varying(255),
    PRIMARY KEY (user_id)
);
```

3. **Employee: -**

```
CREATE TABLE IF NOT EXISTS prct."Employee"
(
    employee_id bigint NOT NULL,
    employee_contact bigint NOT NULL,
    PRIMARY KEY (employee_id, employee_contact),
    CONSTRAINT employee_id FOREIGN KEY (employee_id)
        REFERENCES prct."Employee info" (employee_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

4. **Employeeinfo: -**

```
CREATE TABLE IF NOT EXISTS "S10"."Employee info"
(
    employee_id bigint NOT NULL,
    employee_name character varying(255),
    employee_cur_address character varying(255),
    employee_per_address character varying(255),
    date_of_birth date,
    employee_email character varying(255),
    PRIMARY KEY (employee_id)
);
```

5. **Projectinfo: -**

```
CREATE TABLE IF NOT EXISTS prct."Projectinfo"
(
   project_id bigint NOT NULL,
   project_workstatus date,
   "project_endTime" date,
   PRIMARY KEY (project_id)
);
```

6. **Project: -**

```
CREATE TABLE IF NOT EXISTS prct."Project"
(
   project_id bigint NOT NULL,
   project_title character varying(255),
   "project_startTime" date,
   PRIMARY KEY (project_id),
   CONSTRAINT project_id FOREIGN KEY (project_id)
      REFERENCES prct."Projectinfo" (project_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID
);
```

7. **Team: -**

```
CREATE TABLE IF NOT EXISTS prct."Team"
(
   team_id bigint NOT NULL,
   team_name character varying(255),
   no_of_employee bigint,
   project_id bigint NOT NULL,
   PRIMARY KEY (team_id, project_id),
   CONSTRAINT project_id FOREIGN KEY (project_id)
      REFERENCES prct."Projectinfo" (project_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID
);
```

8. **Task: -**

```
CREATE TABLE IF NOT EXISTS prct."Task"
(
   task_id bigint NOT NULL,
   task_name character varying(255),
   employee_id bigint,
   " task_startTime" date,
   PRIMARY KEY (task_id)
);
```

9. **Task Completion: -**

```
CREATE TABLE IF NOT EXISTS prct."Task_completion"
(
   task_id bigint NOT NULL,
   task_status date,
   "task_endTime" date,
   PRIMARY KEY (task_id),
   CONSTRAINT task_id FOREIGN KEY (task_id)
      REFERENCES prct."Task" (task_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID
);
```

10. **Product: -**

```
CREATE TABLE IF NOT EXISTS prct."Product"
(
   product_id bigint NOT NULL,
   "Product_name" character varying(255),
   "Product_type" character varying(255),
   PRIMARY KEY (product_id)
);
```

11. **Productinfo: -**

```
CREATE TABLE IF NOT EXISTS prct."Product_info"
(
   product_id bigint NOT NULL,
   product_price bigint,
   product_dec character varying(255),
   PRIMARY KEY (product_id),
   CONSTRAINT product_id FOREIGN KEY (product_id)
      REFERENCES prct."Product" (product_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID
);
```

12. **Customer: -**

```
CREATE TABLE IF NOT EXISTS prct."Customer"
(
   customer_id bigint NOT NULL,
   customer_name character varying(255),
```

```
    selected_item character varying(255),
    customer_contact bigint,
    customer_address character varying(255),
    PRIMARY KEY (customer_id)
);
```

13. **Payment: -**

```
CREATE TABLE IF NOT EXISTS prct."Payment"
(
    "Payment_id" bigint NOT NULL,
    payment_amount bigint,
    payment_date date,
    payment_mode character varying(255),
    PRIMARY KEY ("Payment_id")
);
```

14. **Paymentinfo: -**

```
CREATE TABLE IF NOT EXISTS prct."Paymentinfo"
(
    payment_id bigint NOT NULL,
    customer_id bigint NOT NULL,
    PRIMARY KEY (payment_id, customer_id),
    CONSTRAINT payment_id FOREIGN KEY (payment_id)
        REFERENCES prct."Payment" ("Payment_id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT customer_id FOREIGN KEY (customer_id)
        REFERENCES prct."Customer" (customer_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

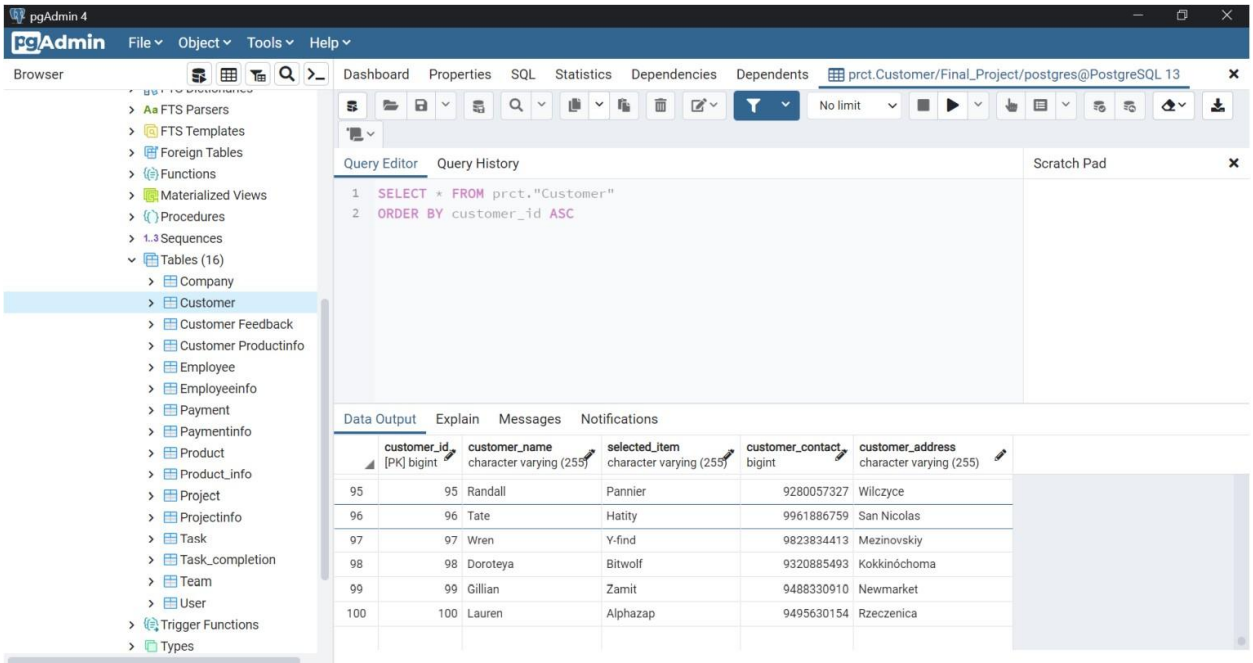
15. **Customer Feedback: -**

```
CREATE TABLE IF NOT EXISTS prct."Customer Feedback"
(
    customer_id bigint NOT NULL,
    selected_item character varying(255),
    feedback_description character varying(255),
    PRIMARY KEY (customer_id)
);
```

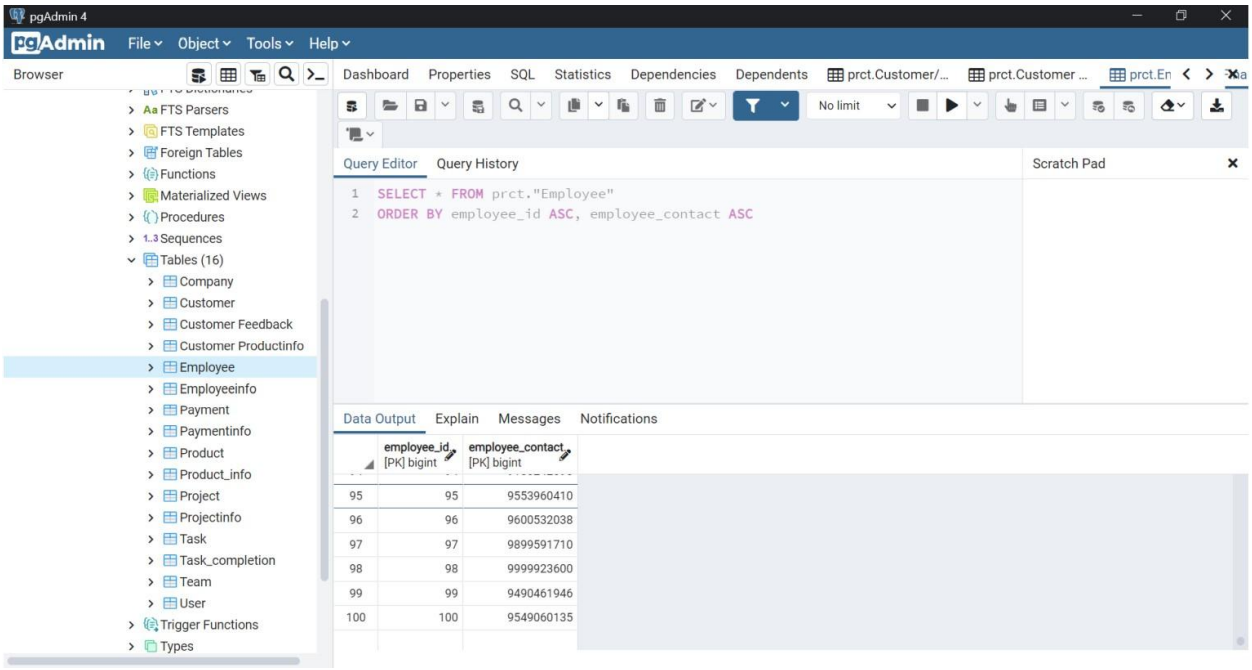
16. **Customer productinfo: -**

```
CREATE TABLE IF NOT EXISTS prct."Customer Productinfo"
(
    customer_id bigint NOT NULL,
    product_id bigint NOT NULL,
    PRIMARY KEY (customer_id, product_id),
    CONSTRAINT customer_id FOREIGN KEY (customer_id)
        REFERENCES prct."Customer" (customer_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT product_id FOREIGN KEY (product_id)
        REFERENCES prct."Product" (product_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

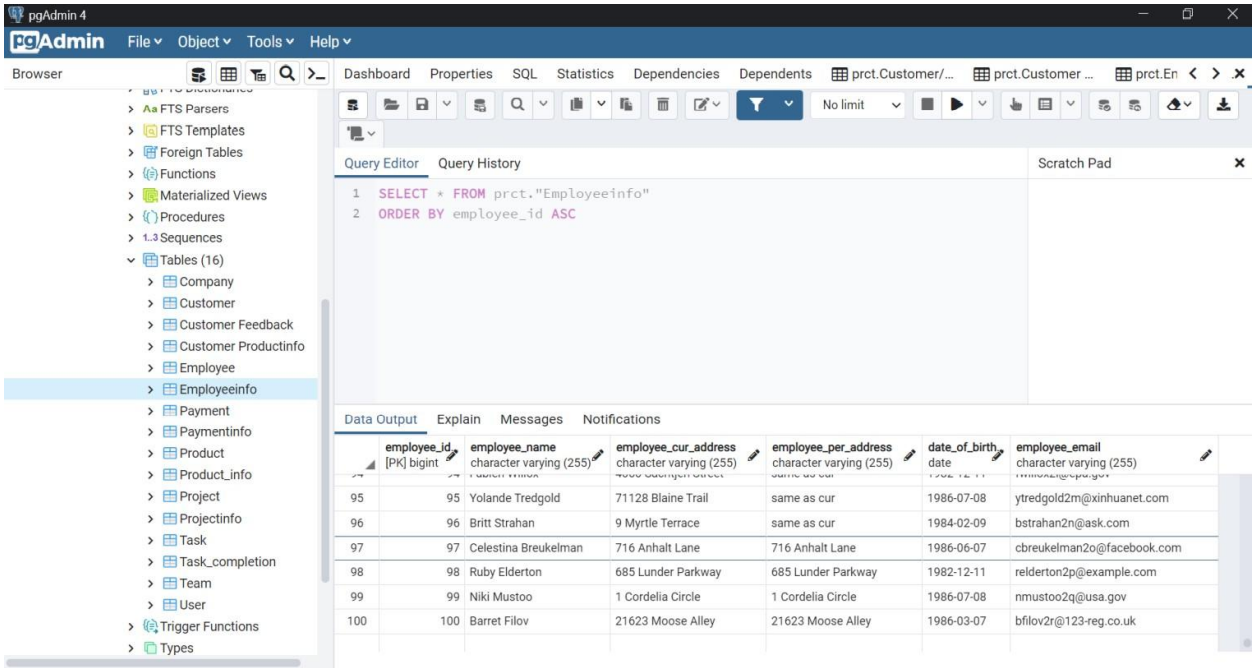
Customer: -



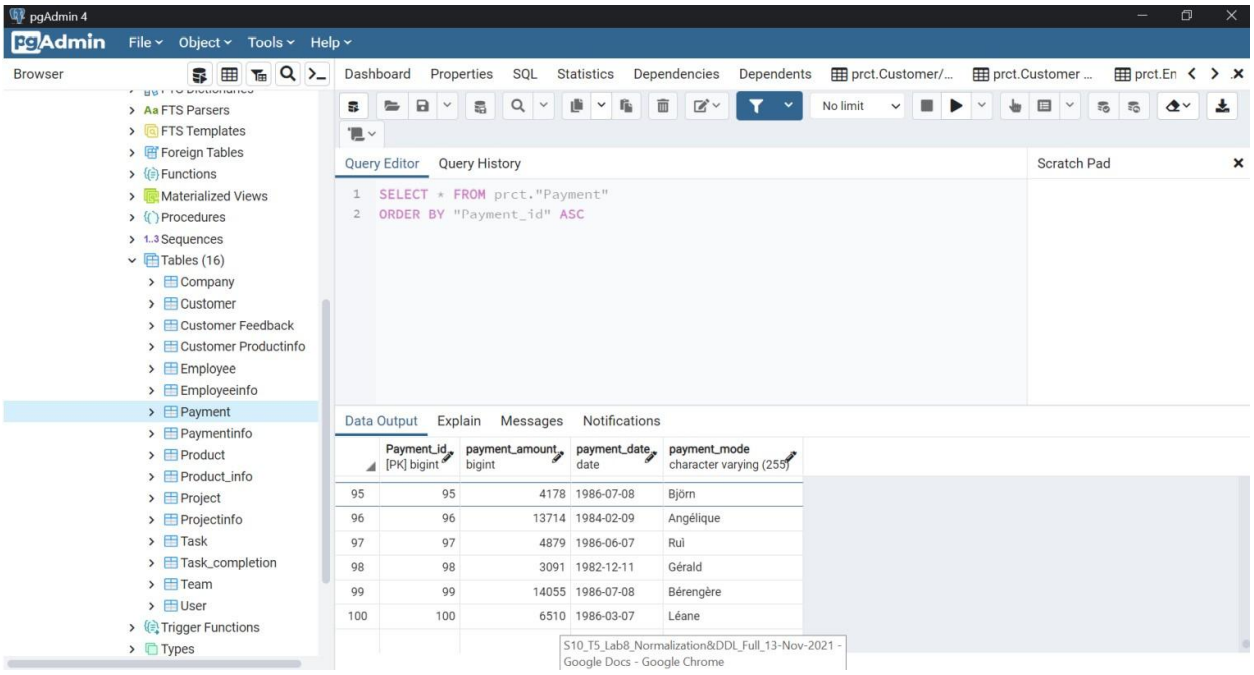
Employee: -



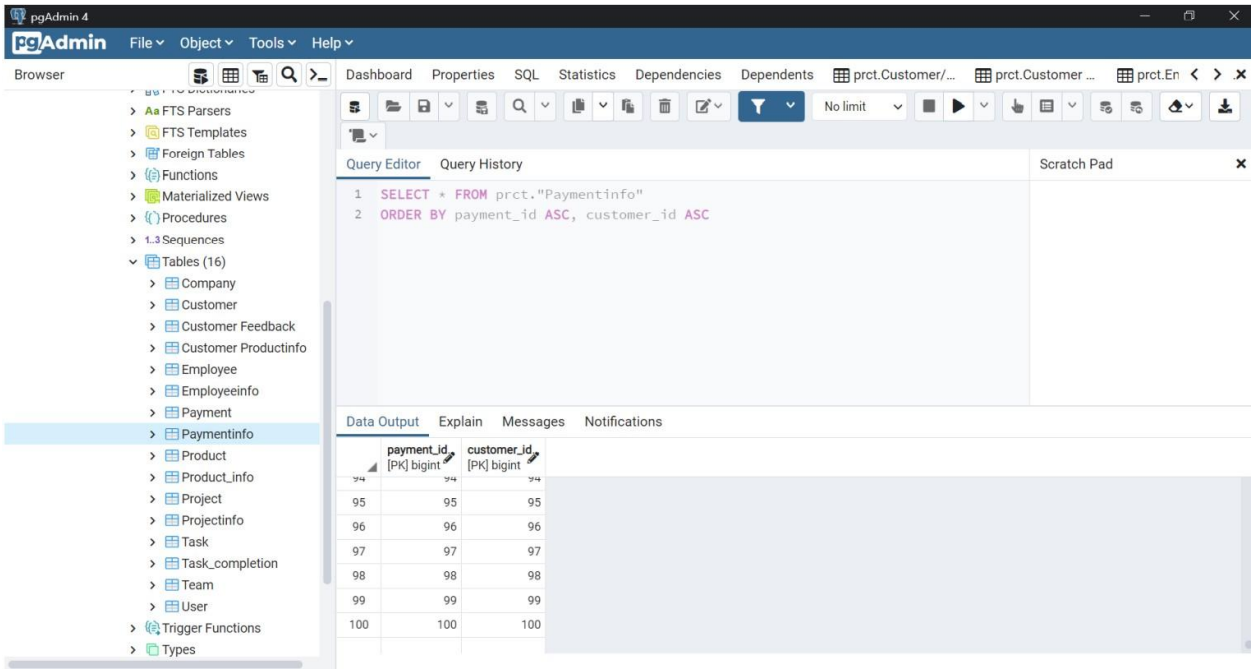
Employeeinfo: -



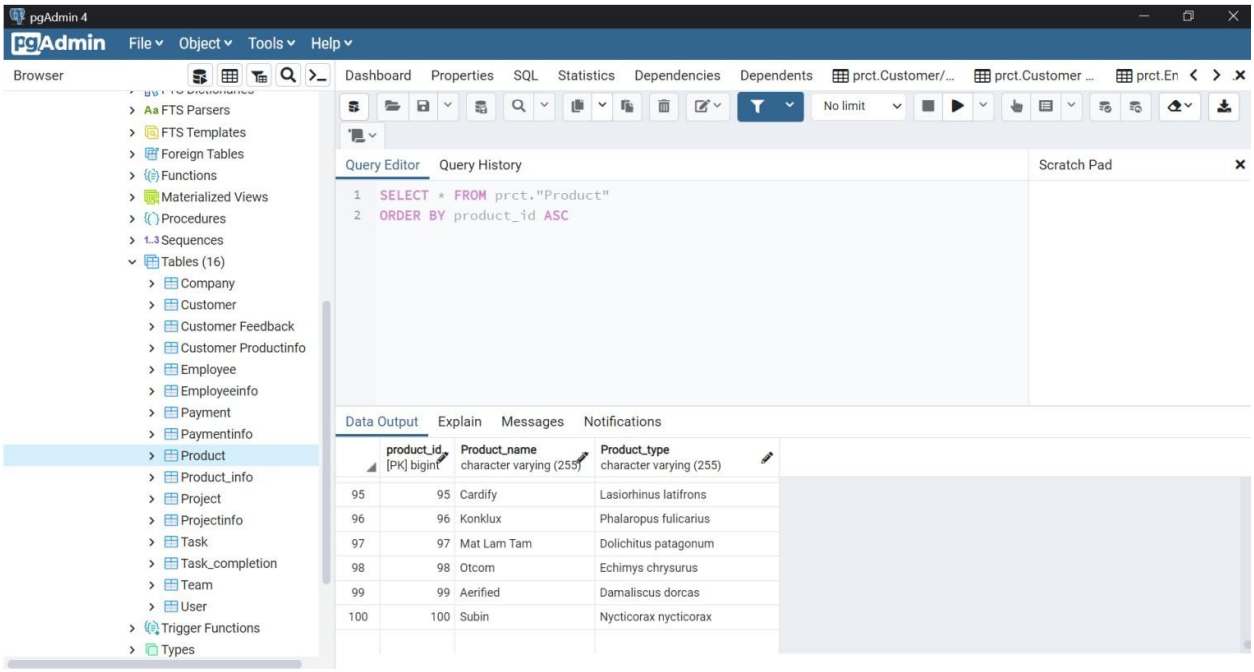
Payment: -



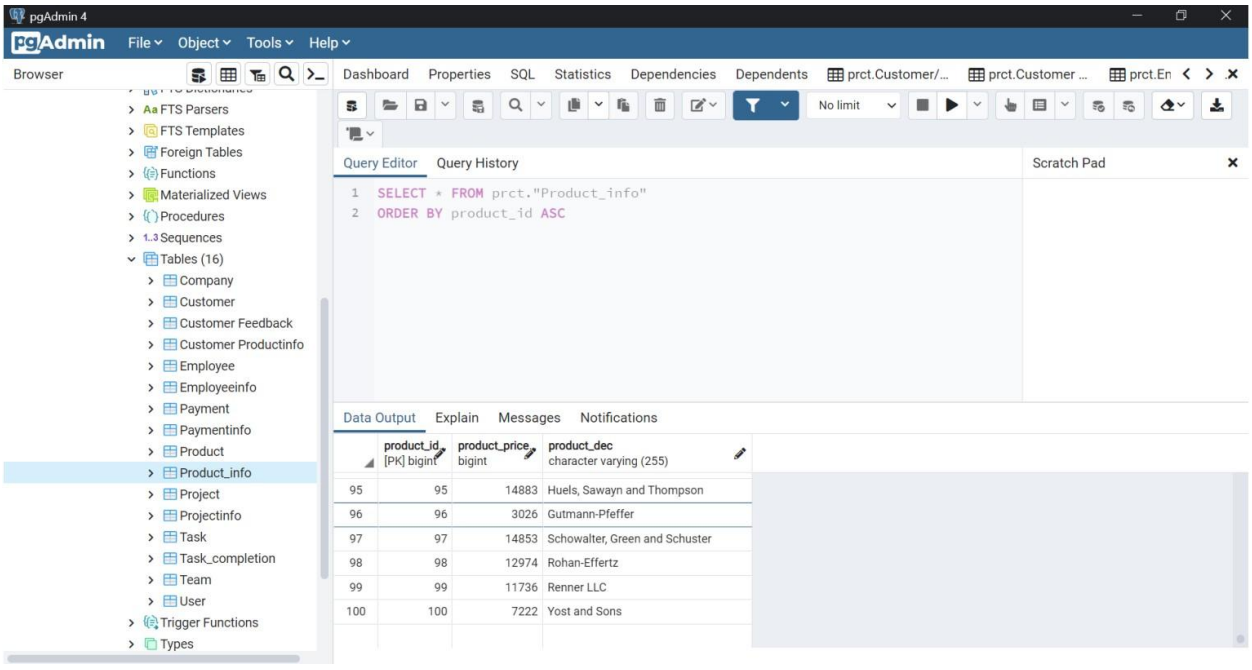
Paymentinfo: -



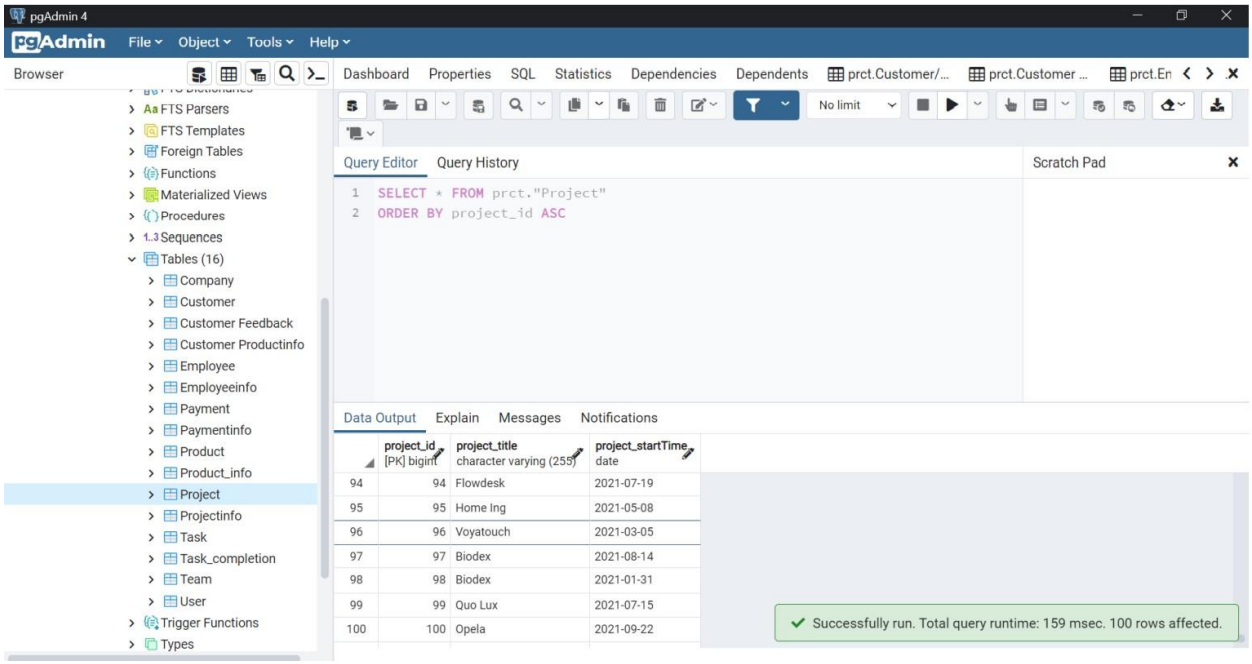
Product: -



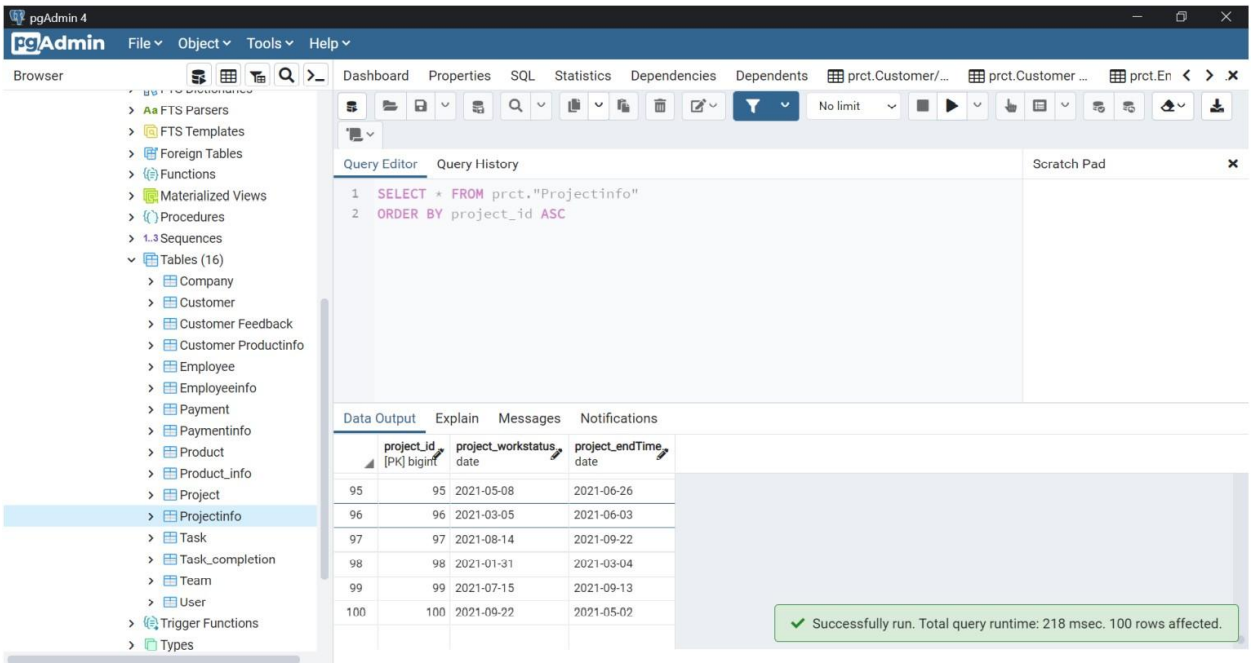
Product_info: -



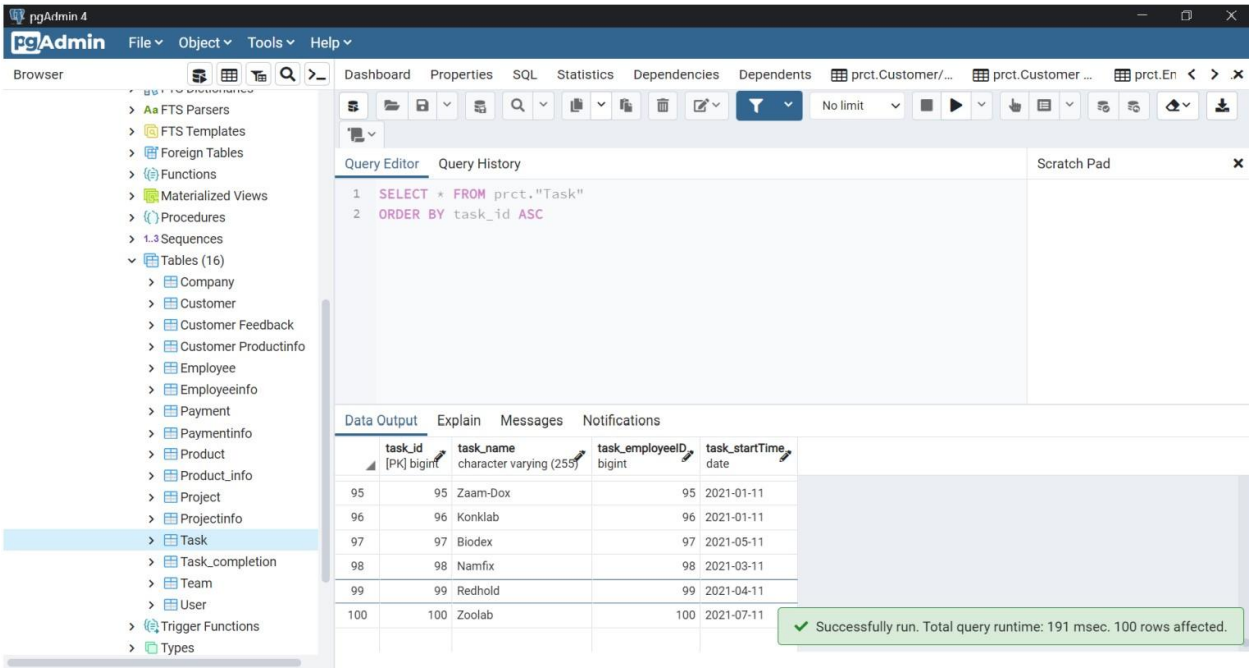
Project: -



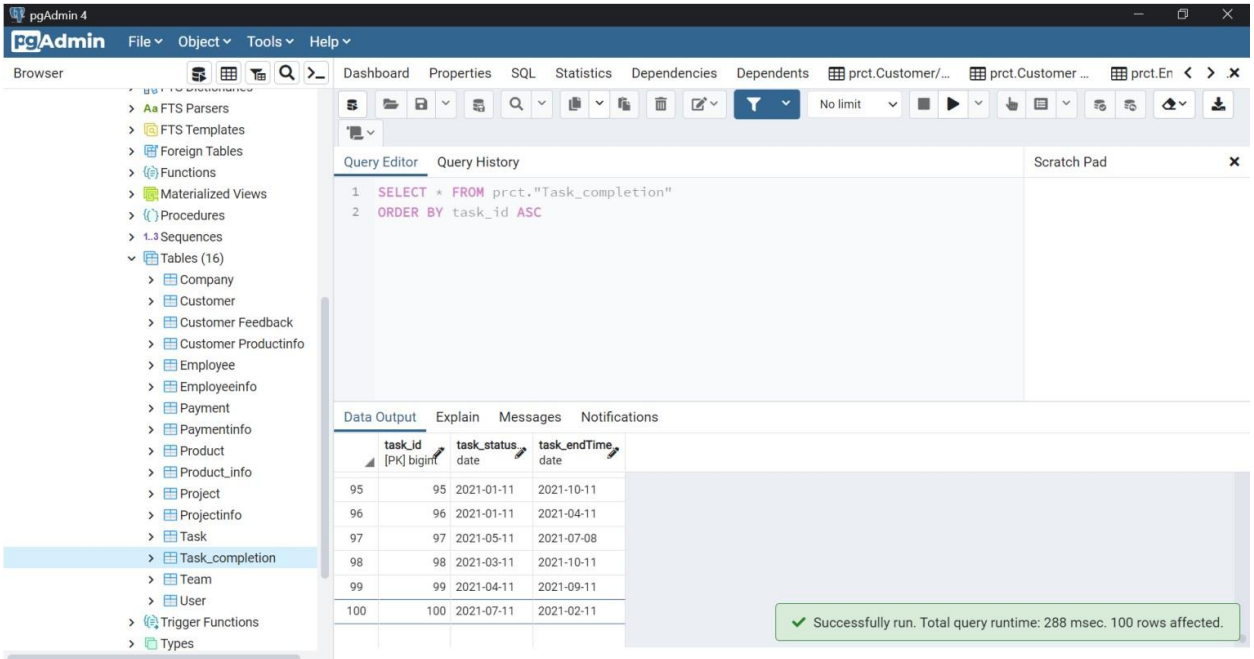
Projectinfo: -



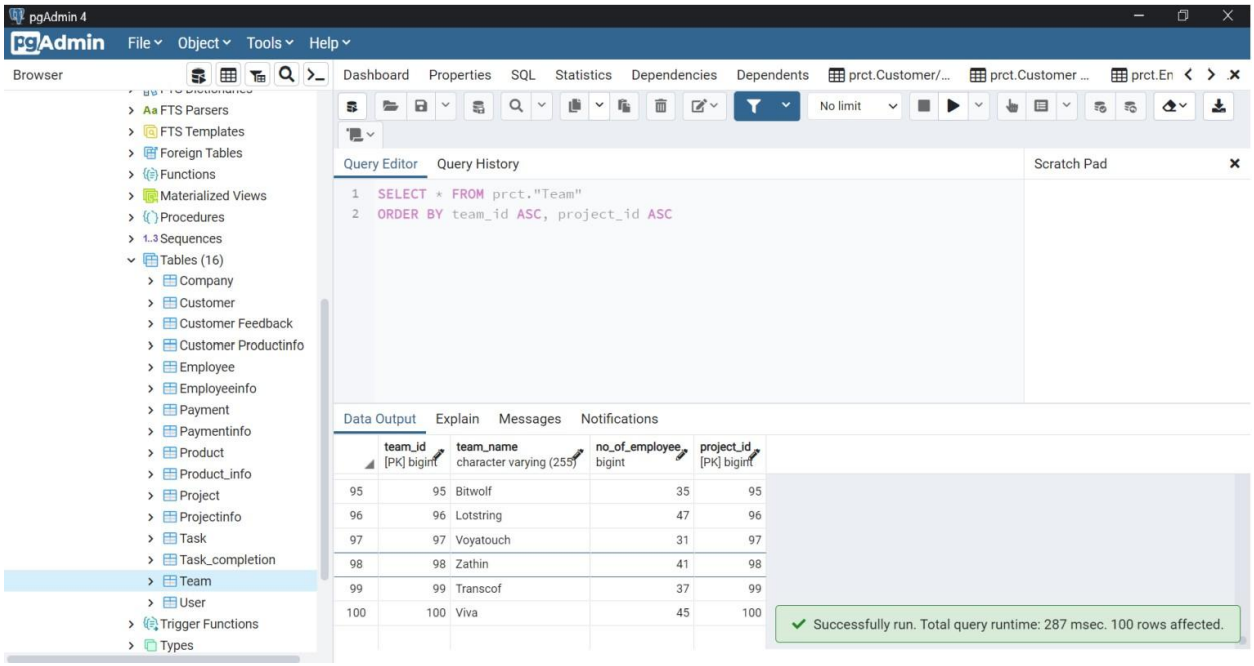
Task: -



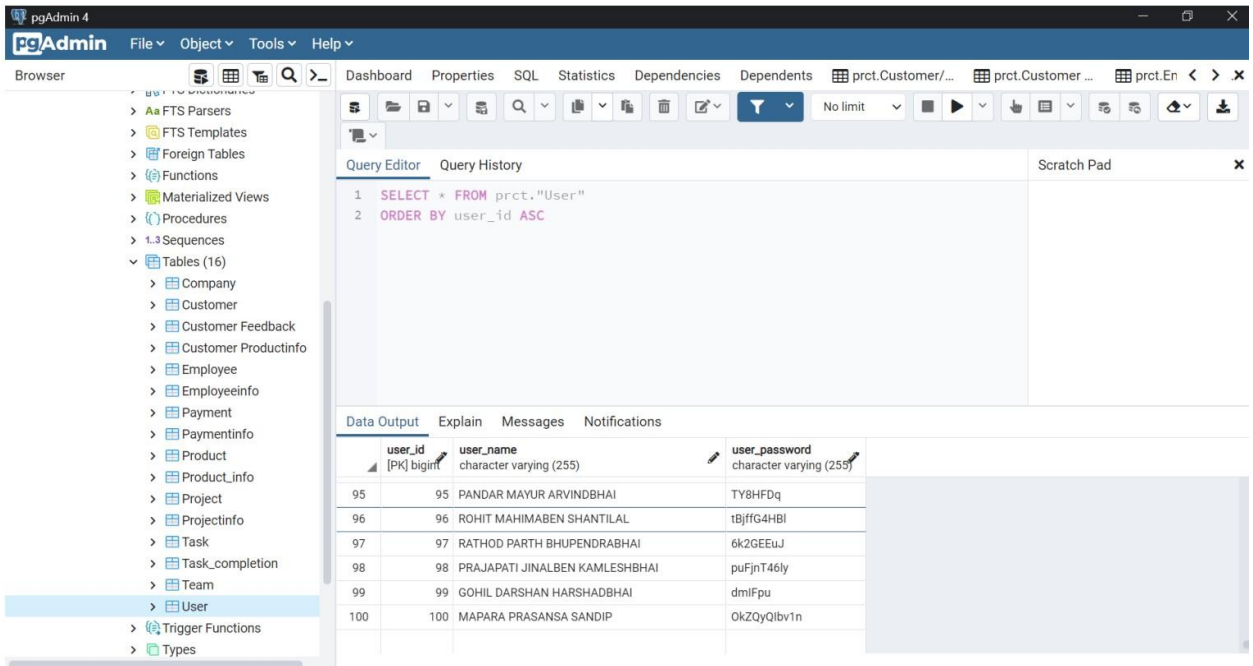
Task_completion: -



Team: -

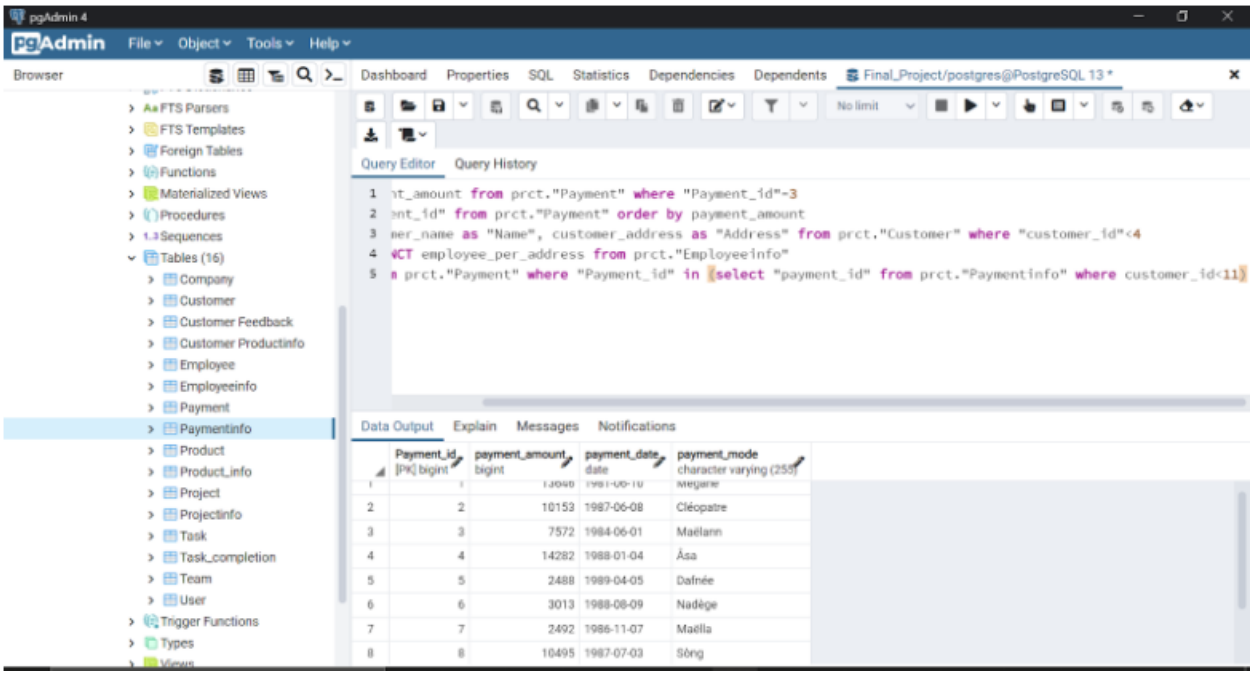

pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents prct.Customer/... prct.Customer ... prct.En
No limit
Query Editor Query History
Scratch Pad
1 SELECT * FROM prct."Team"
2 ORDER BY team_id ASC, project_id ASC
Data Output Explain Messages Notifications
team_id [PK] bigint
team_name character varying (255)
no_of_employee bigint
project_id [PK] bigint
95 95 Bitwolf 35 95
96 96 Lotstring 47 96
97 97 Voyatouch 31 97
98 98 Zathin 41 98
99 99 Transcof 37 99
100 100 Viva 45 100
Successfully run. Total query runtime: 287 msec. 100 rows affected.
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (16)
Company
Customer
Customer Feedback
Customer Productinfo
Employee
Employeeinfo
Payment
Paymentinfo
Product
Product_info
Project
Projectinfo
Task
Task_completion
Team
User
Trigger Functions
Types

User: -

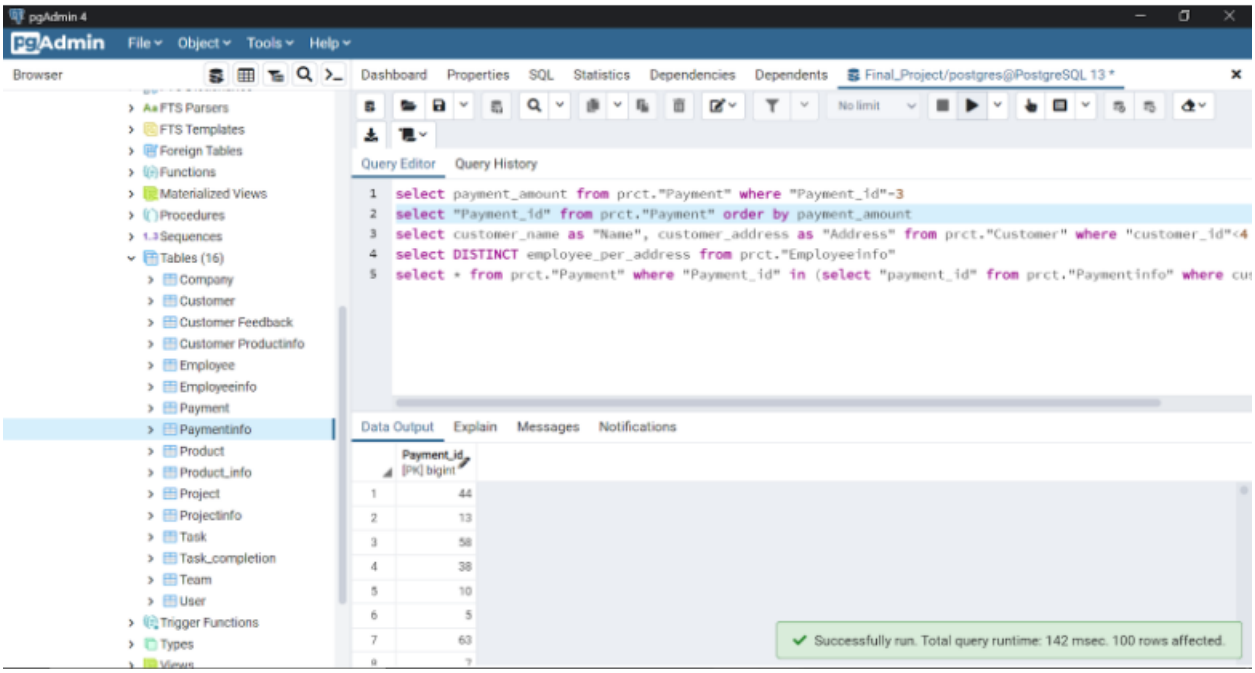

pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents prct.Customer/... prct.Customer ... prct.En
No limit
Query Editor Query History
Scratch Pad
1 SELECT * FROM prct."User"
2 ORDER BY user_id ASC
Data Output Explain Messages Notifications
user_id [PK] bigint
user_name character varying (255)
user_password character varying (255)
95 95 PANDAR MAYUR ARVINDBHAI TY8HFDq
96 96 ROHIT MAHIMABEN SHANTILAL tBjffG4HBl
97 97 RATHOD PARTH BHUPENDRABHAI 6k2GEEuJ
98 98 PRAJAPATI JINALBEN KAMLESHBHAI puFjnT46ly
99 99 GOHIL DARSHAN HARSHADBHAI dmlFpu
100 100 MAPARA PRASANSA SANDIP OkZQyQIbv1n
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (16)
Company
Customer
Customer Feedback
Customer Productinfo
Employee
Employeeinfo
Payment
Paymentinfo
Product
Product_info
Project
Projectinfo
Task
Task_completion
Team
User
Trigger Functions
Types

QUERY: -

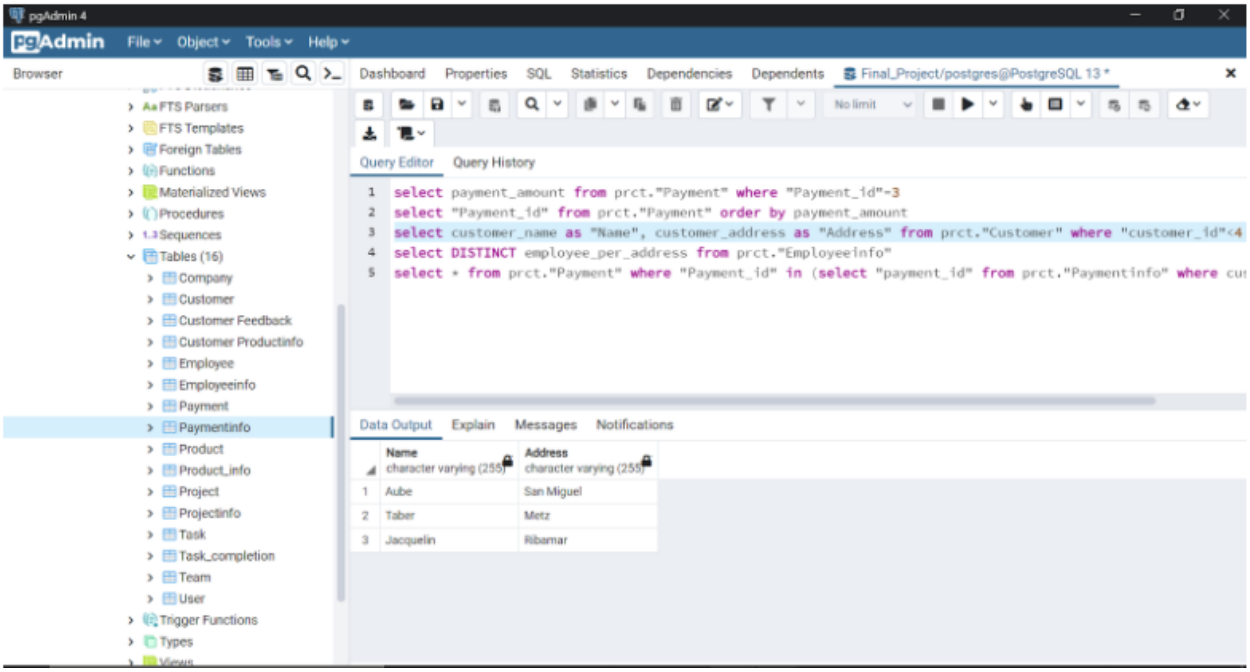
1) select payment_amount from prct."Payment" where "Payment_id"=3



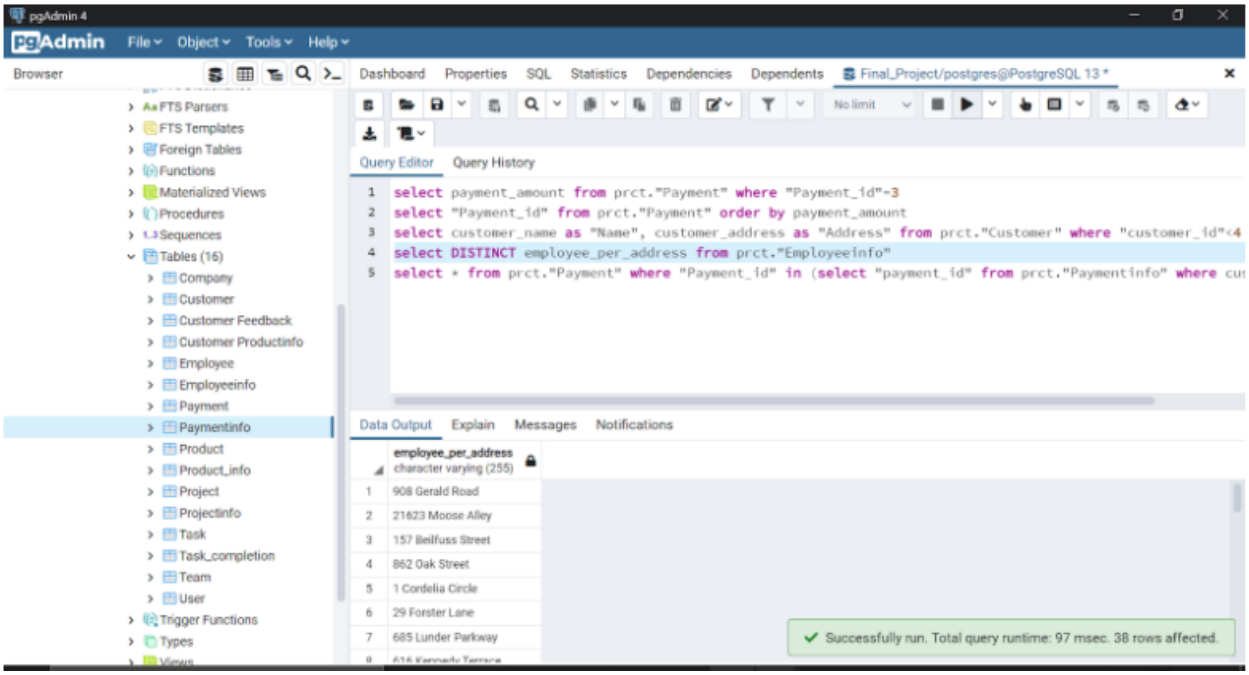
2) select "Payment_id" from prct."Payment" order by payment_amount



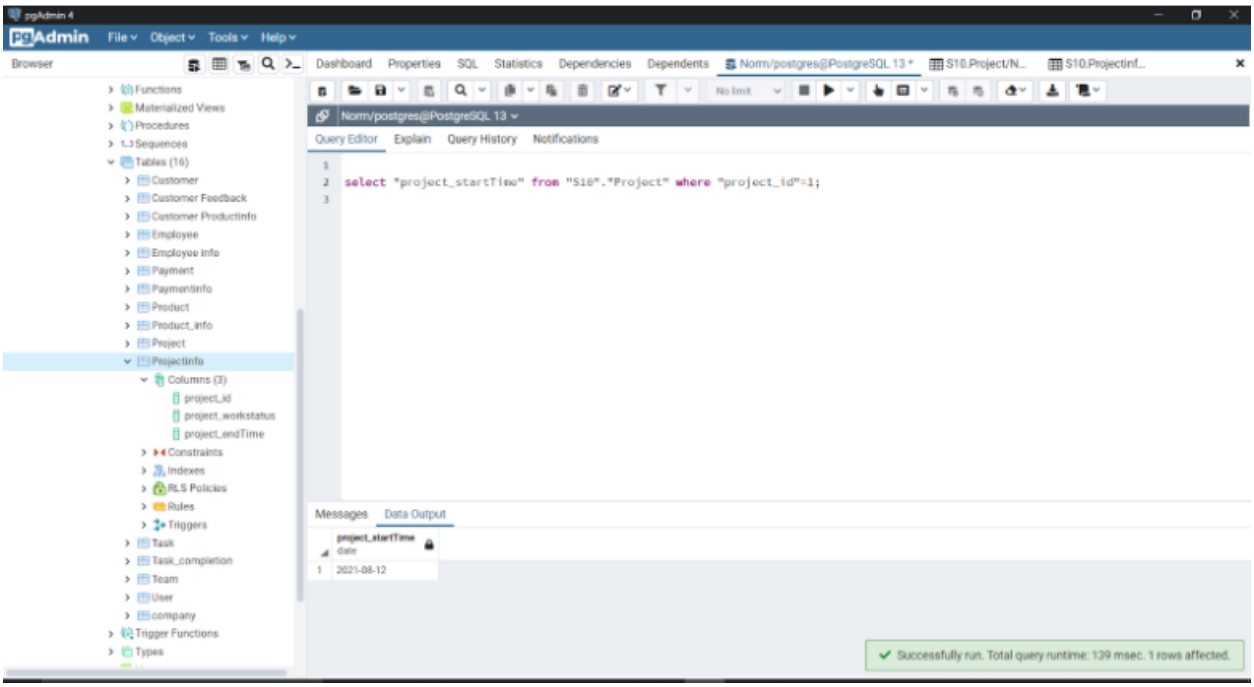
3) select customer_name as "Name", customer_address as "Address" from prct."Customer" where "customer_id"<4



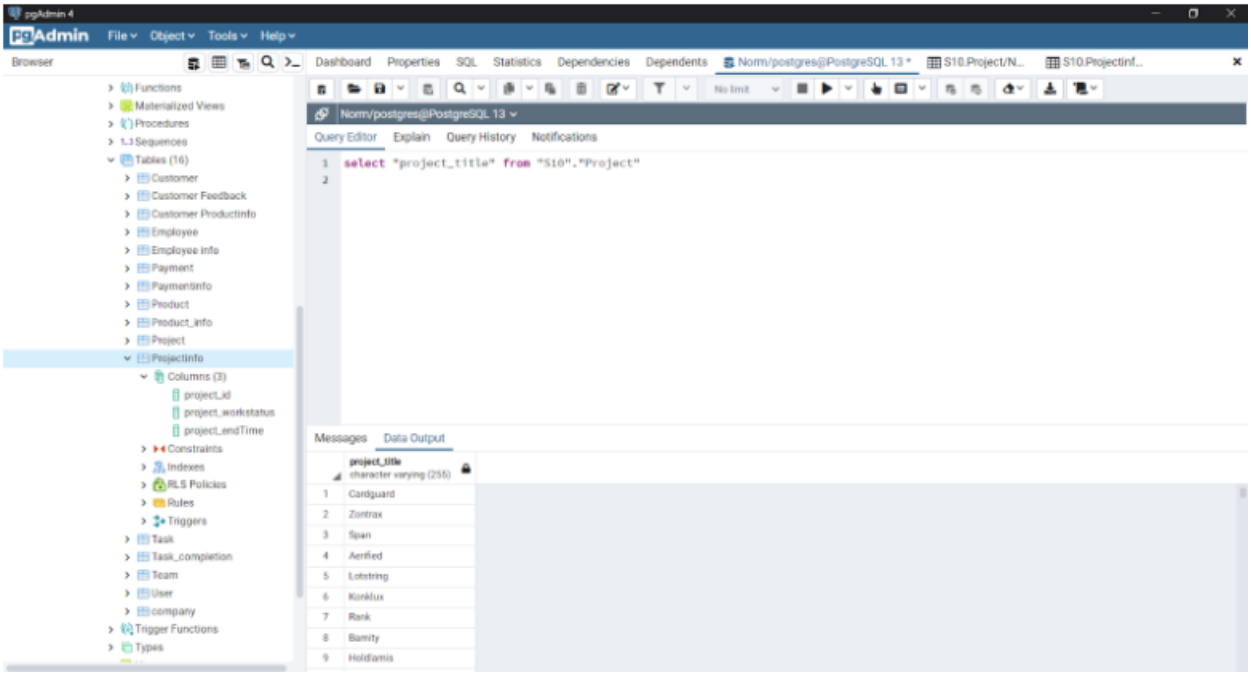
4) select DISTINCT employee_per_address from prct."Employeeinfo"



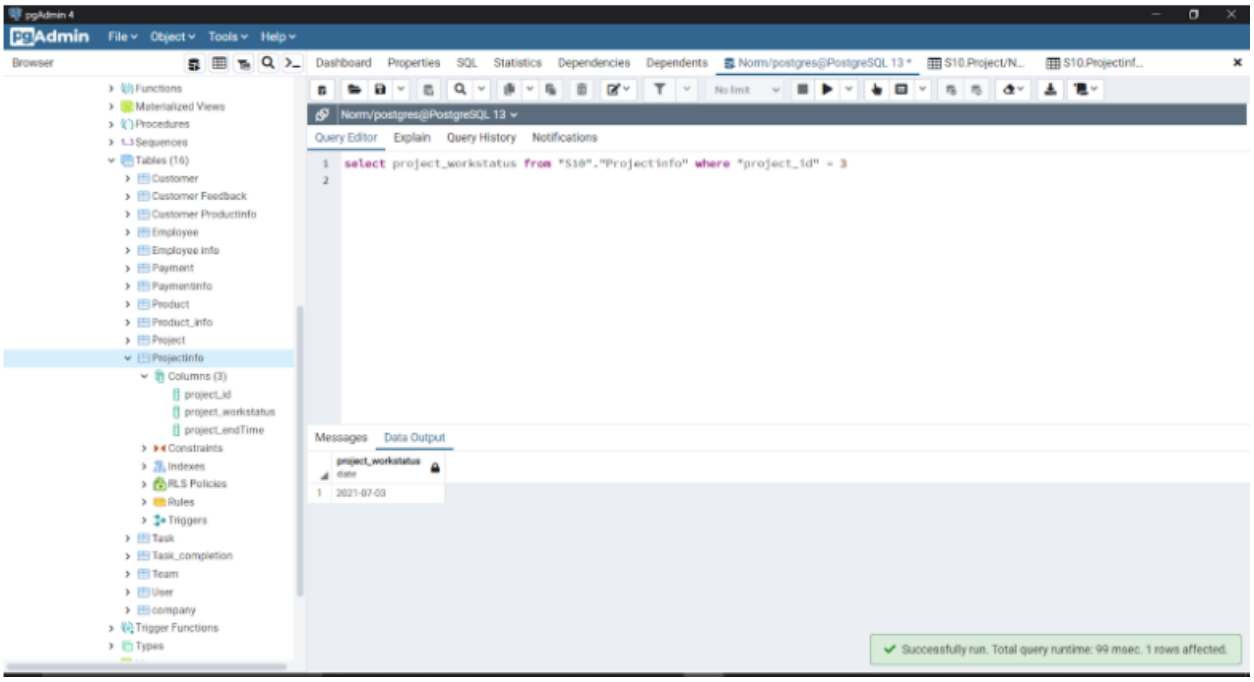
5) select "project_startTime" from "S10"."Project" where "project_id"=1;



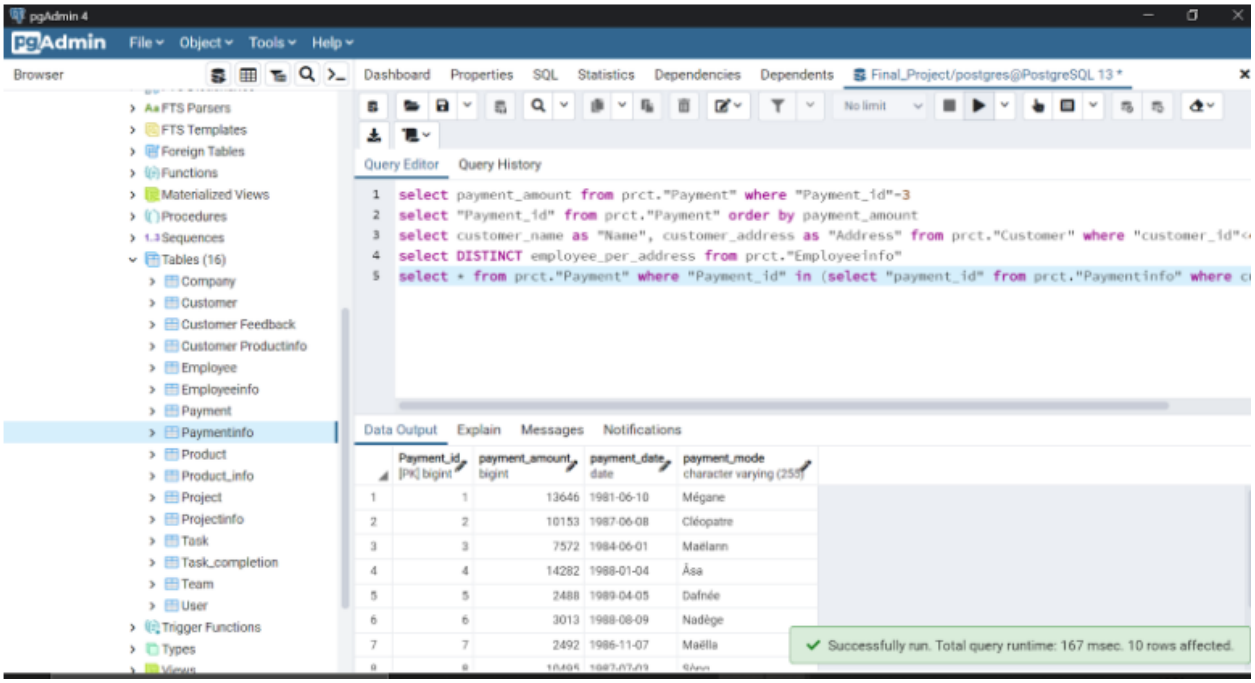
6) select "project_title" from "S10"."Project"



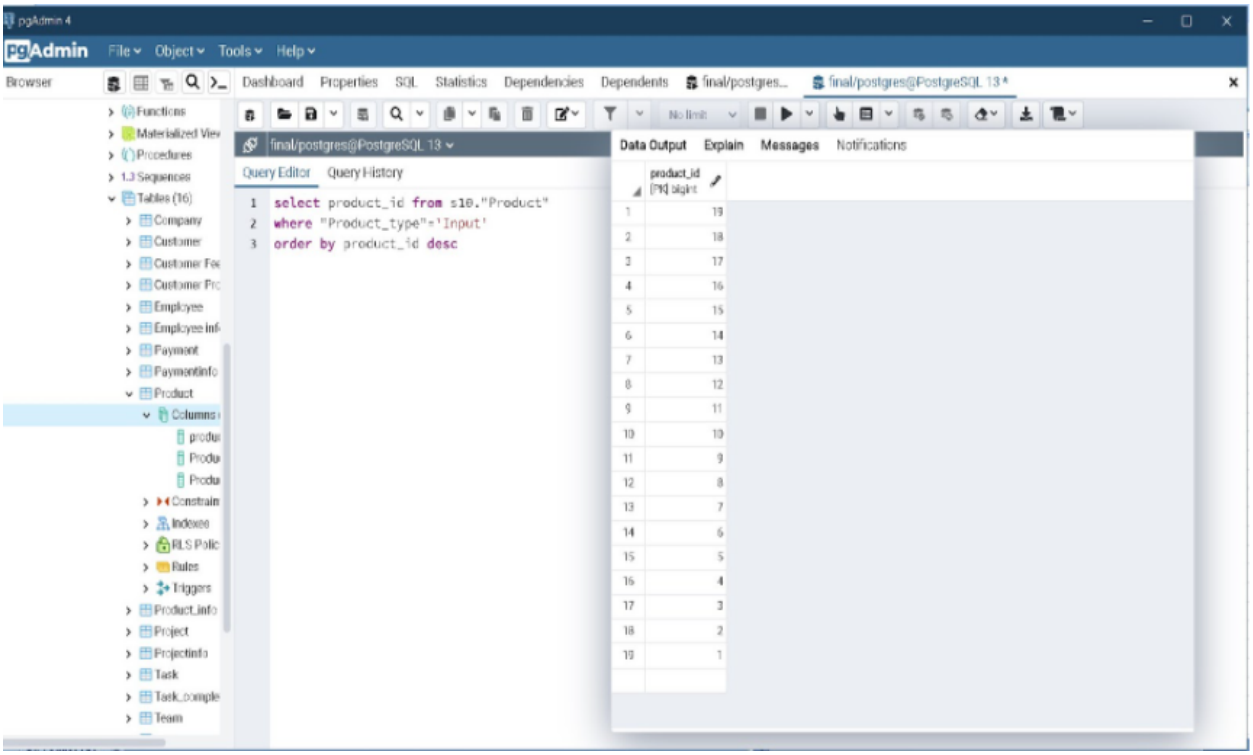
7) select project_workstatus from "S10"."Projectinfo" where "project_id" = 3



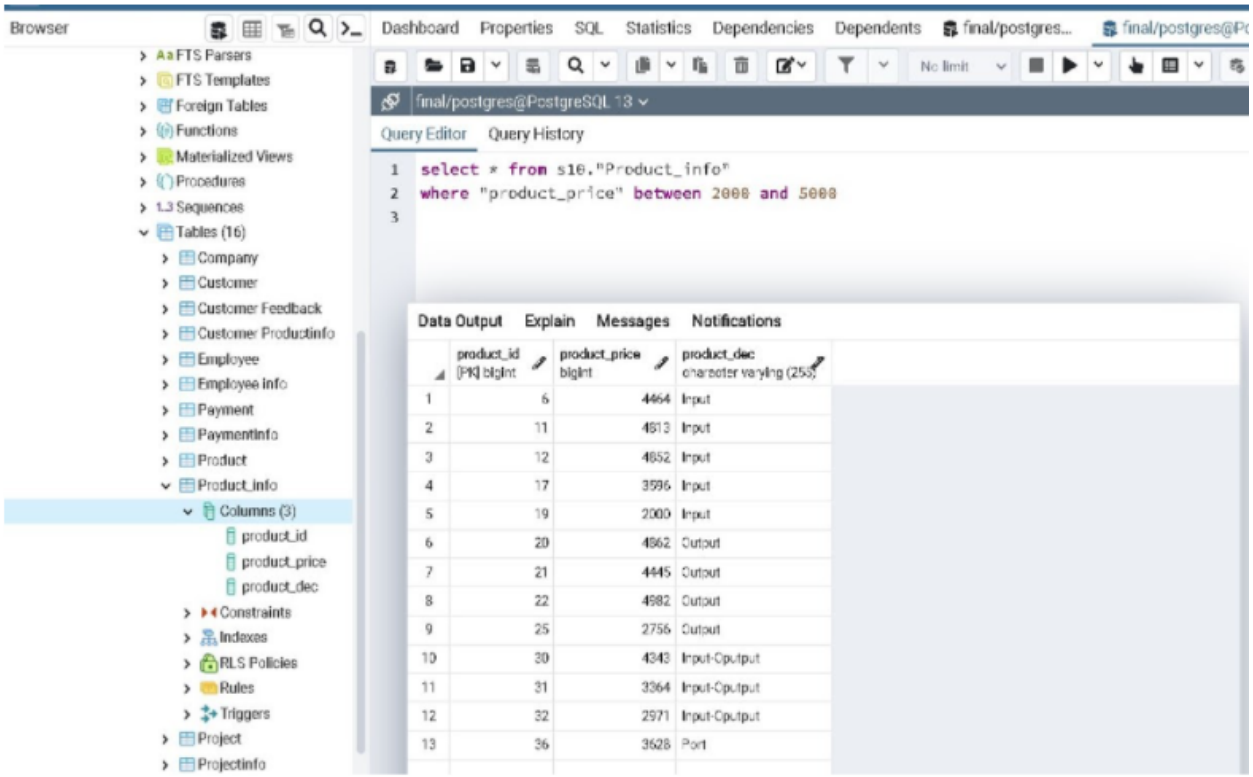
8) select * from prct."Payment" where "Payment_id" in (select "payment_id" from prct."Paymentinfo" where customer_id<11)



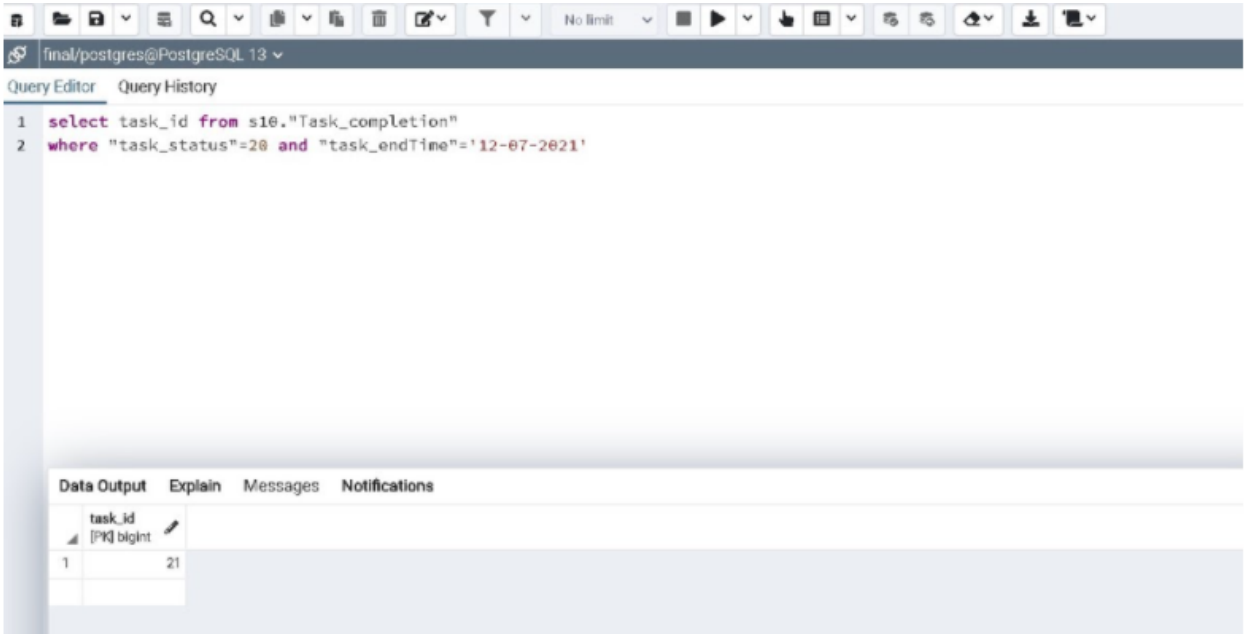
9) select product_id from “S10”. “product” where Product_type=‘Input’ order by product_id desc



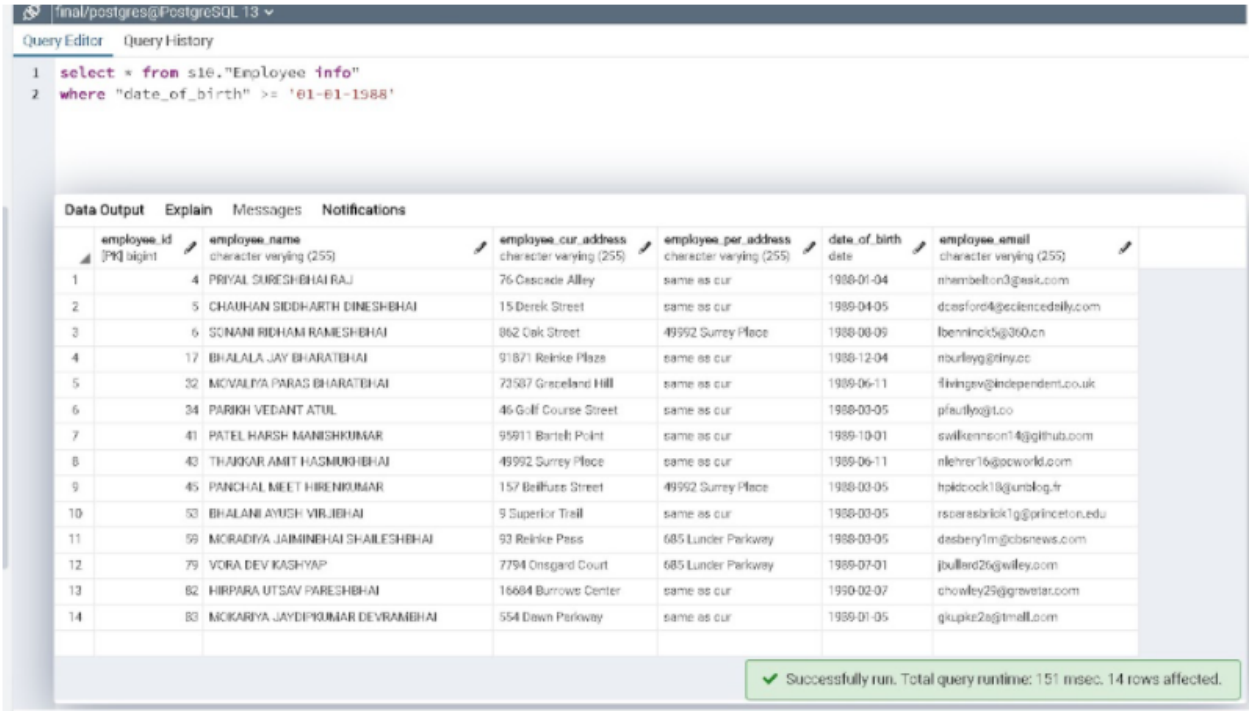
10) Select all details from Product_info table where product_price is between rupees 2000 to 5000.



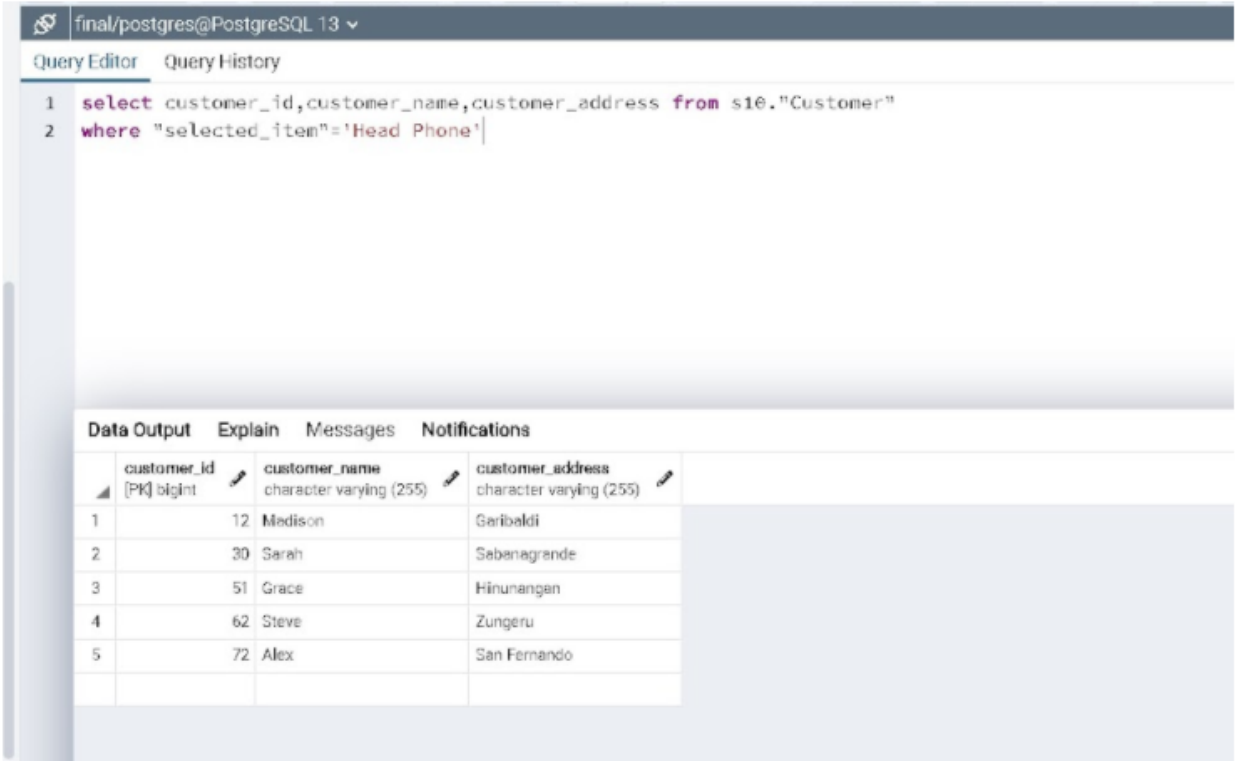
11) select task_id where task_status is 20 and task_endTime is 12-7-2021.



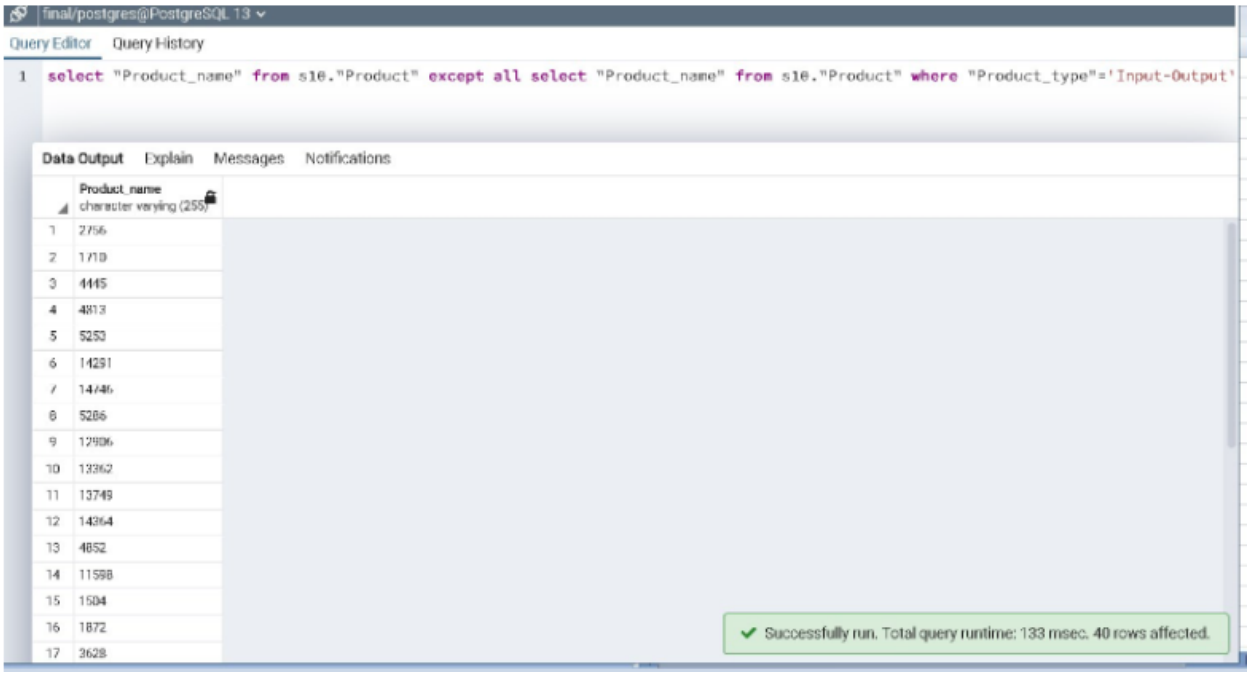
12) Select all details from Employee info table where date of birth of employee is greater than or equal to 1-1-1988



13) Select customer id,customer name and customer address from Customer table where selected item is ‘Head Phone’



14) Select Product_name except all Product where "Product_type"='Input-Output'



15) Select all details from the task table of the employees who are born before 23-10-1983.

final/postgres@PostgreSQL 13

Query Editor | Query History

```
1  SELECT * FROM s10."Task"
2  WHERE employee_id in (
3  SELECT employee_id
4  FROM s10."Employee info"
5  WHERE date_of_birth < '23-10-1983'
6  )
7
```

Data Output | Explain | Messages | Notifications

	task_id [PK] bigint	task_name character varying (255)	employee_id bigint	task_startTime date
1	3	taskA3	28	2021-08-18
2	4	taskB1	26	2021-06-25
3	18	taskF3	19	2021-06-04
4	19	taskG1	48	2021-06-21
5	21	taskG3	20	2021-07-08
6	26	taskI2	1	2021-02-03
7	27	taskI3	18	2021-02-11
8	34	taskL1	52	2021-09-02
9	38	taskM2	23	2021-11-14
10	40	taskN1	37	2021-01-10
11	43	taskO1	11	2021-10-09
12	44	taskO2	14	2021-10-19
13	49	taskQ1	56	2021-02-27
14	50	taskQ2	13	2021-03-04

16) Select team_name as “Team Name” and project_title as “Project Name” from Team and Project table.

final/postgres@PostgreSQL 13

Query Editor | Query History

```
select team_name as "Team Name", project_title as "Project Name"
from s10."Team" join s10."Project"
on s10."Team".team_id=s10."Project".project_id
```

Data Output | Explain | Messages | Notifications

	Team Name character varying (255)	Project Name character varying (255)
1	Karma Gigantic	ProjectA
2	The Titans	ProjectB
3	Internet Explorers	ProjectC
4	Tech Connect	ProjectD
5	Blue Screens	ProjectE
6	Temp	ProjectF
7	Java Our Souls	ProjectG
8	Solarbreeze	ProjectH
9	Zemit	ProjectI
10	SubEx	ProjectJ

17) Find out "Employee info"."employee_id","Employee info"."employee_name","Employee"."employee_contact".

final/postgres@PostgreSQL 13

Query Editor | Query History

```
Select "Employee info"."employee_id","Employee info"."employee_name","Employee"."employee_contact"
from s10."Employee info" full join s10."Employee"
on "Employee"."employee_id" = "Employee info"."employee_id";
```

Data Output | Explain | Messages | Notifications

	employee_id bigint	employee_name character varying (255)	employee_contact bigint
1	1	PATEL VISHVAKUMARI BHARATBHAI	9452445437
2	2	SURAPANENI SRI AKHIL	9797969040
3	3	DHAVALKUMAR HASMUKHBHAI SOLANKI	9730463356
4	4	PRIYAL SURESHBHAI RAJ	9650778056
5	5	CHAUHAN SIDDHARTH DINESHBHAI	9572496054
6	6	SONANI RIDHAM RAMESHBHAI	9446598752
7	7	DIYORA MANAN MANOJBHAI	9429538863
8	8	VAGHASIYA CHIRAG JIVRAJBHAI	9726620624
9	9	DHOLA AXIT KISHORBHAI	9416361746
10	10	PATEL AKSH BHARATLAL	9502208341
11	11	SAVALIYA KEVAL DINESHKUMAR	9979867473
12	12	SHAH HARMIBEN ATULBHAI	9185191099

Successfully run. Total query runtime: 275 msec. 100 rows affected.

18) Find out product id,product name,product type,product price.

final/postgres@PostgreSQL 13

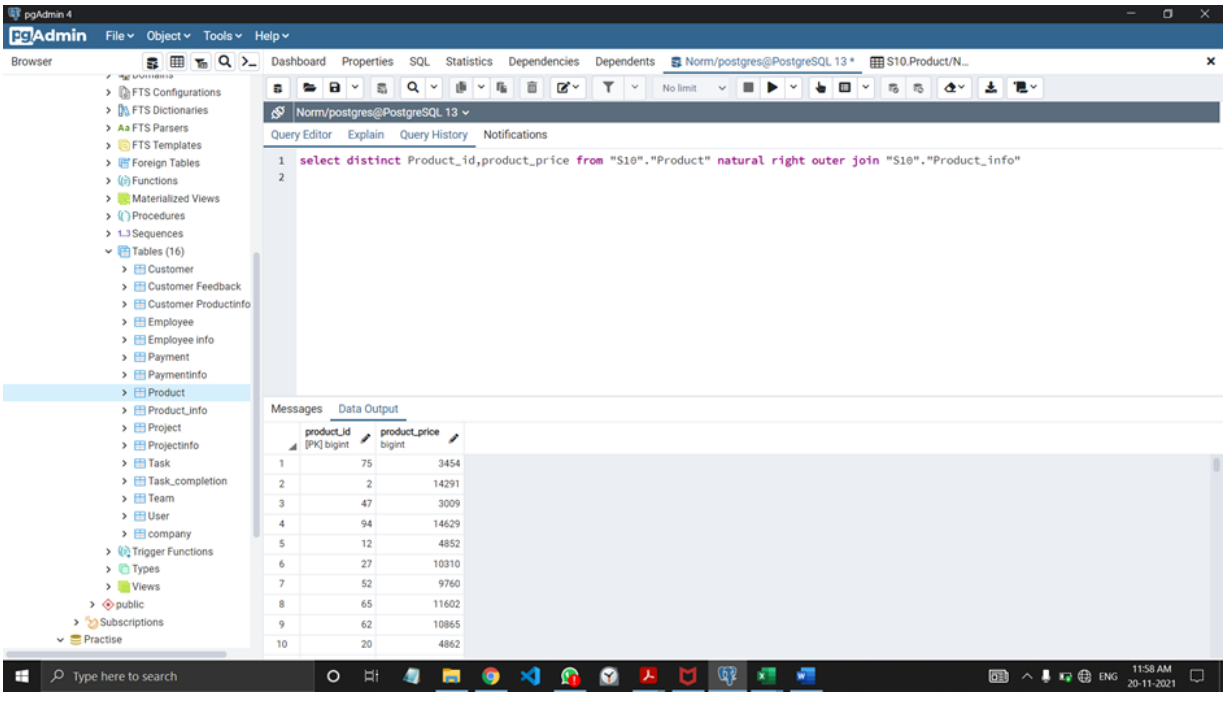
Query Editor | Query History

```
1  select "Product".product_id,"Product"."Product_name","Product"."Product_type","Product_info".product_price
2  from s10."Product_info" LEFT JOIN s10."Product"
3  ON s10."Product".product_id = s10."Product_info".product_id
4
5
```

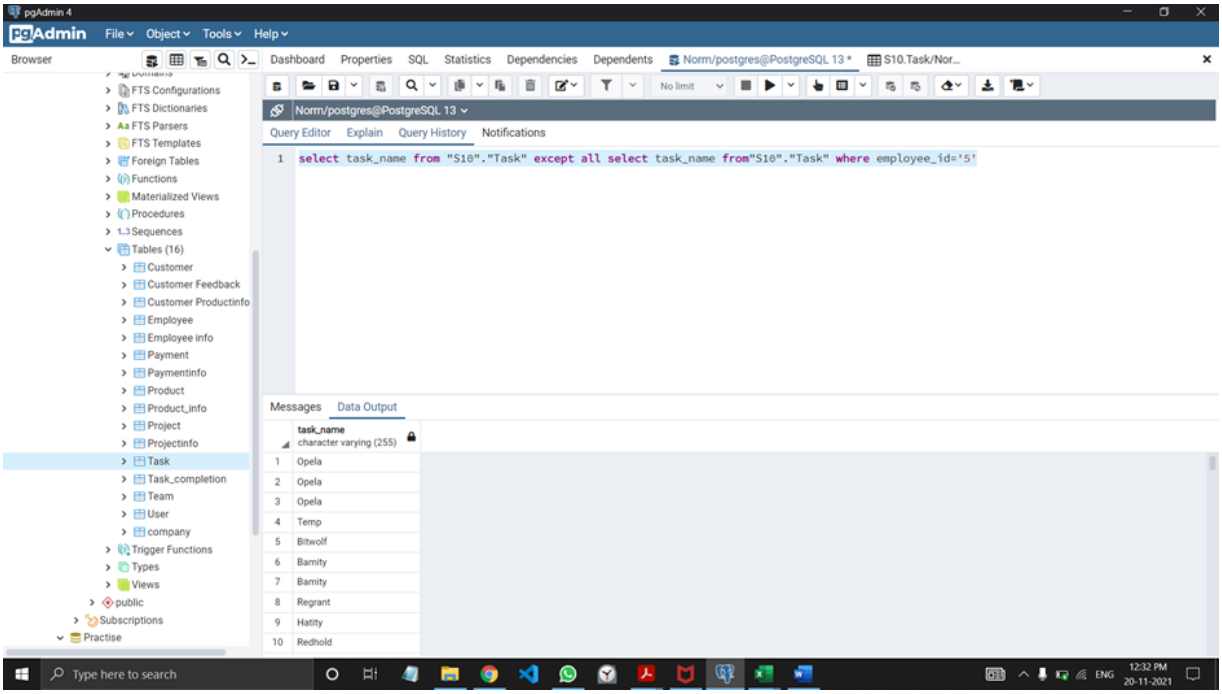
Data Output | Explain | Messages | Notifications

	product_id bigint	Product_name character varying (255)	Product_type character varying (255)	product_price bigint
1	1	14746	Input	14746
2	2	14291	Input	14291
3	3	8277	Input	8277
4	4	5253	Input	5253
5	5	14364	Input	14364
6	6	4464	Input	4464
7	7	8074	Input	8074
8	8	9882	Input	9882
9	9	11375	Input	11375
10	10	14753	Input	14753
11	11	4813	Input	4813
12	12	4852	Input	4852
13	13	7506	Input	7506
14	14	1872	Input	1872
15	15	13362	Input	13362
16	16	13749	Input	13749
17	17	3596	Input	3596

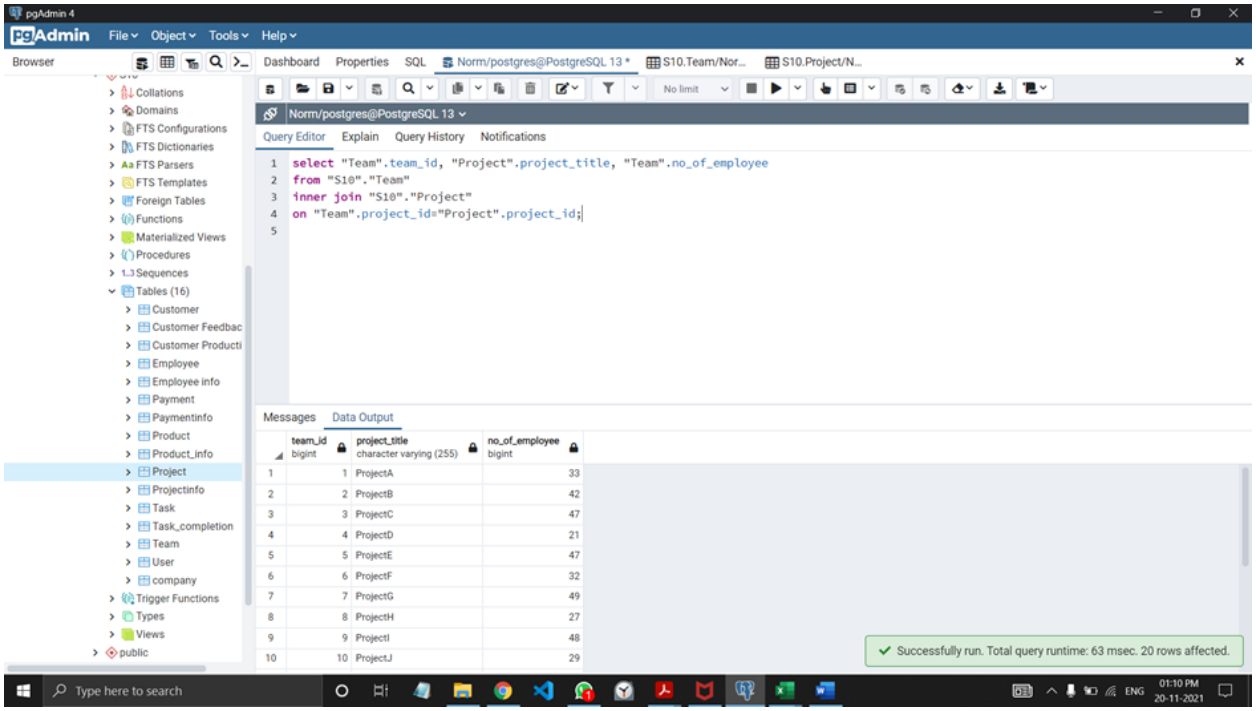
19) select distinct Product_id,product_price from "S10"."Product" natural right outer join "S10"."Product_info"



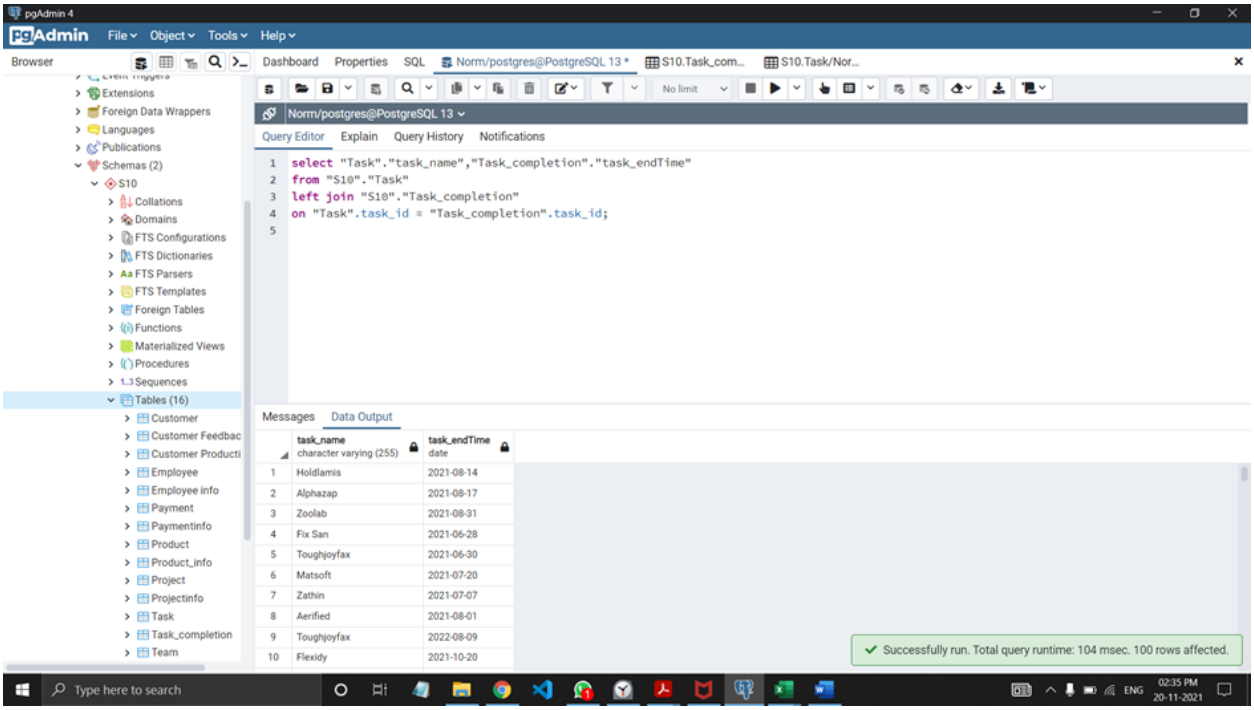
20) select task_name from "S10"."Task" except all select task_name from"S10"."Task" where employee_id='5'



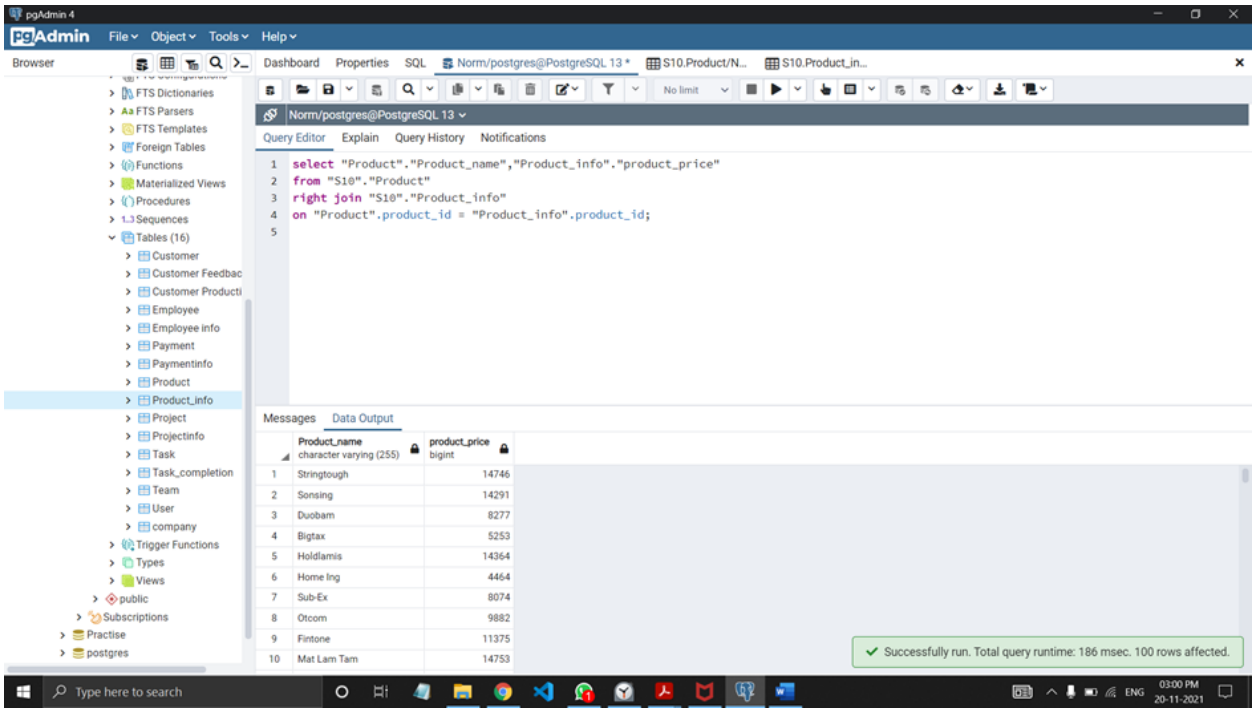
21) select "Team".team_id, "Project".project_title, "Team".no_of_employee from "S10"."Team" inner join "S10"."Project" on"Team".project_id="Project".project_id;



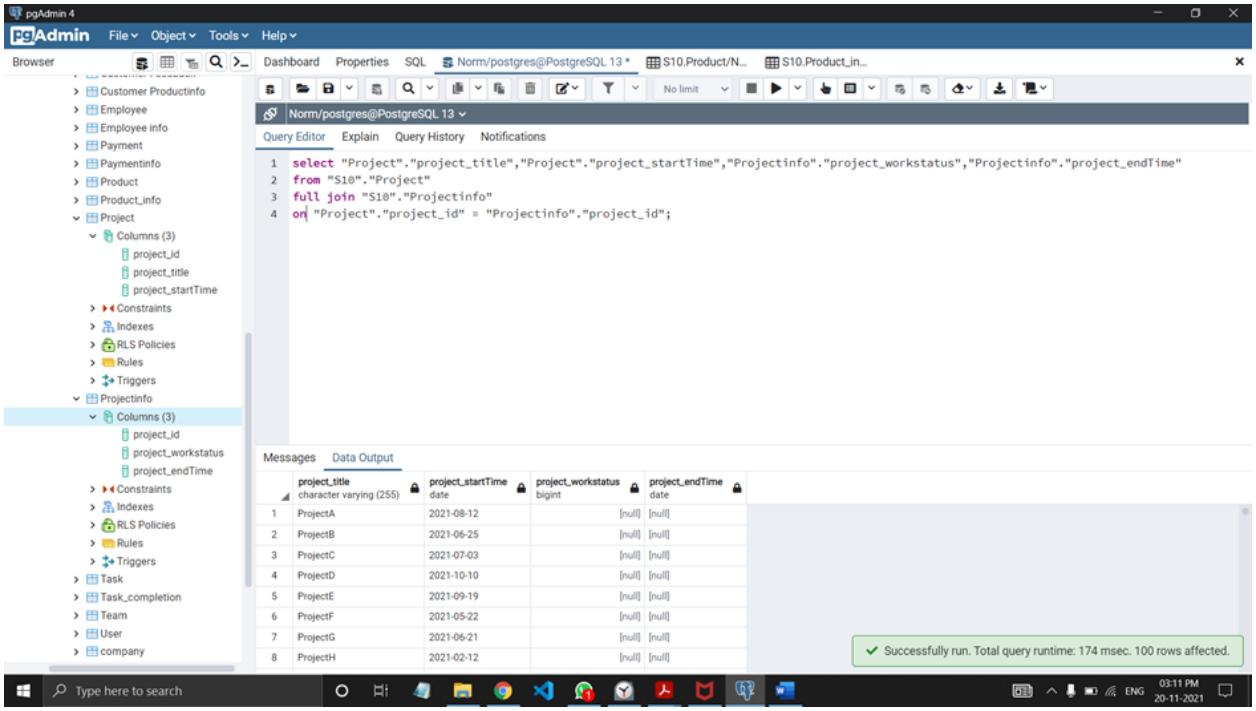
22) select "Task"."task_name","Task_completion"."task_endTime" from "S10"."Task" left join "S10"."Task_completion" on "Task".task_id = "Task_completion".task_id;



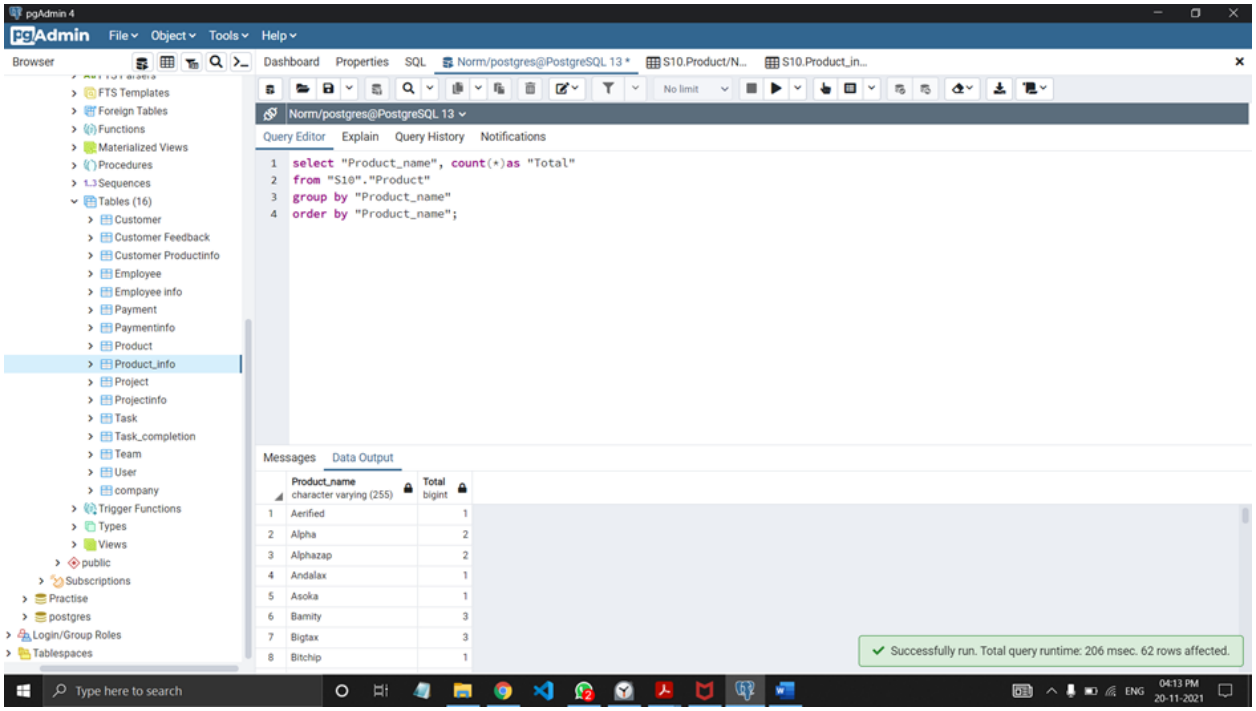
23) select "Product"."Product_name","Product_info"."product_price" from "S10"."Product" right join "S10"."Product_info" on "Product".product_id = "Product_info".product_id;



24) Select"Project"."project_title","Project"."project_startTime","Projectinfo"."project_workstatus", "Projectinfo"."project_endTime" from "S10"."Project" full join "S10"."Projectinfo" on "Project"."project_id" = "Projectinfo"."project_id";

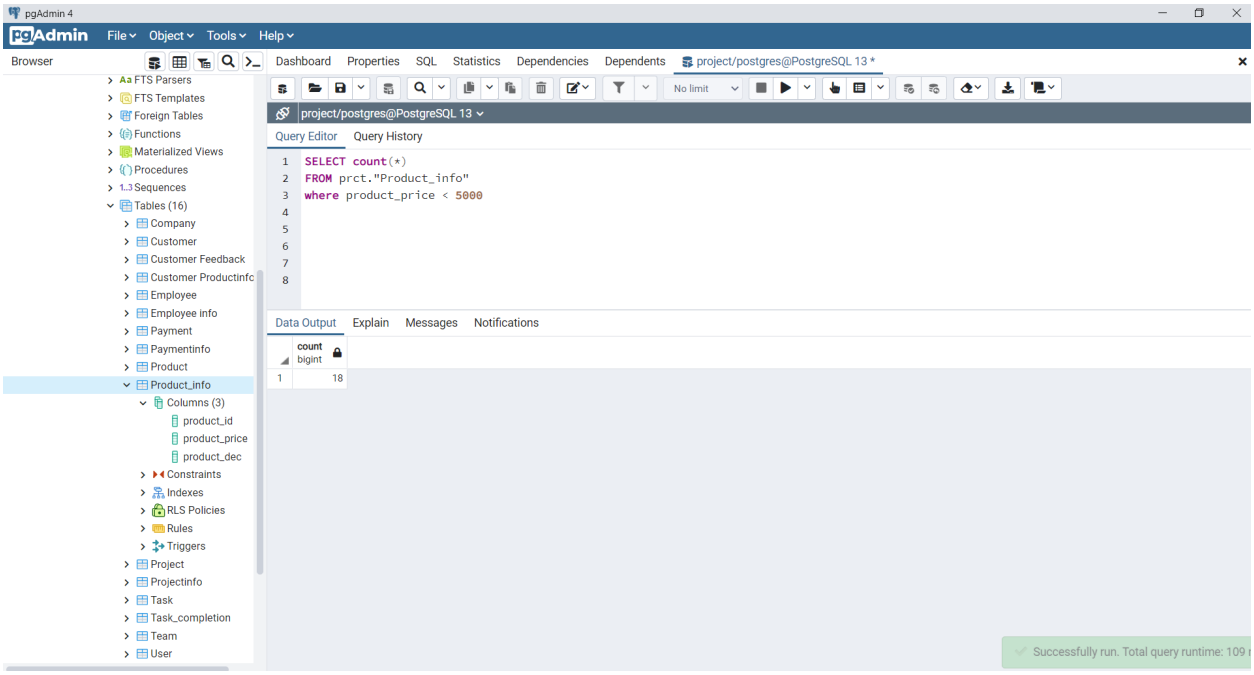


25) select "Product_name", count(*)as "Total" from "S10"."Product" group by "Product_name" order by "Product_name";



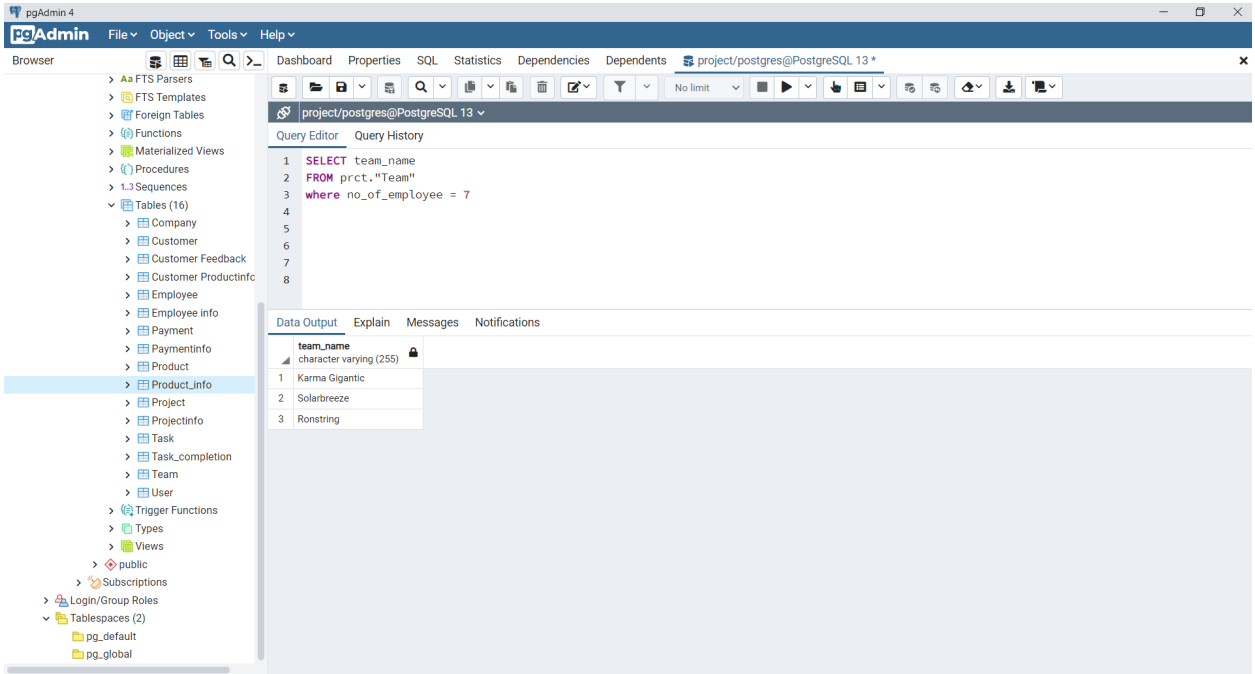
26) Find total number of product which have price less than 5000.

SELECT count(*) FROM prct.”Product_info” WHERE product_price < 5000



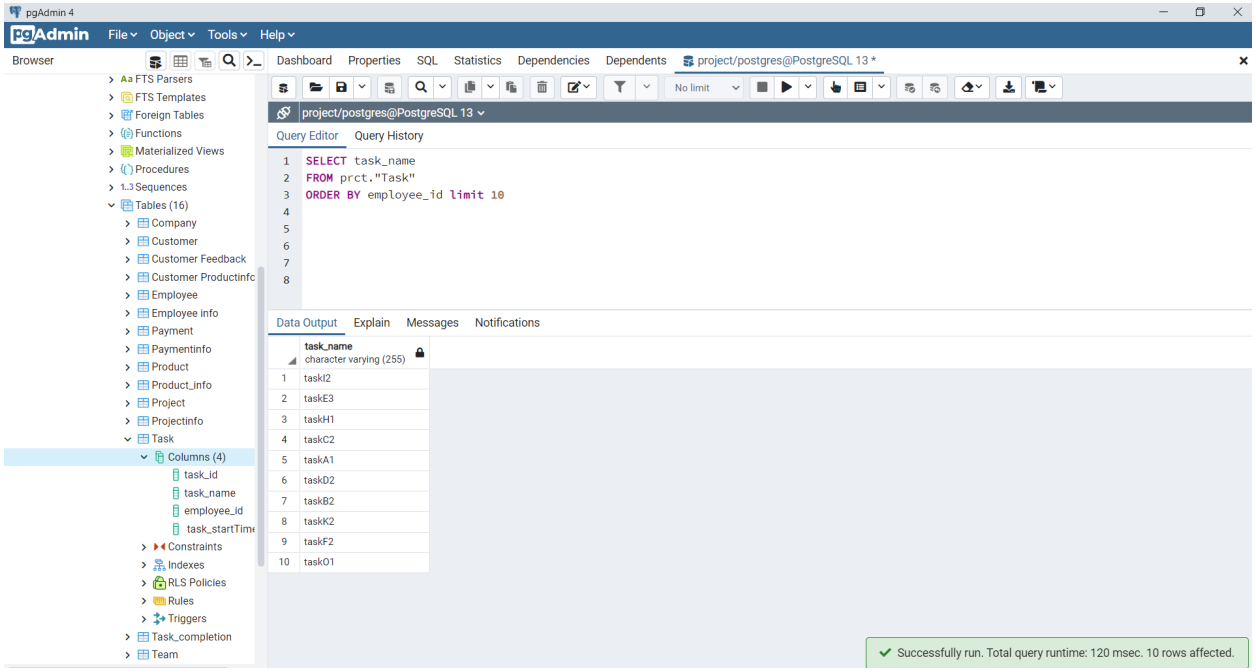
27) Find all the team from “Team” table which have 7 employee in Team.

SELECT team_name FROM prct.”Team” Where no_of_employee = 7



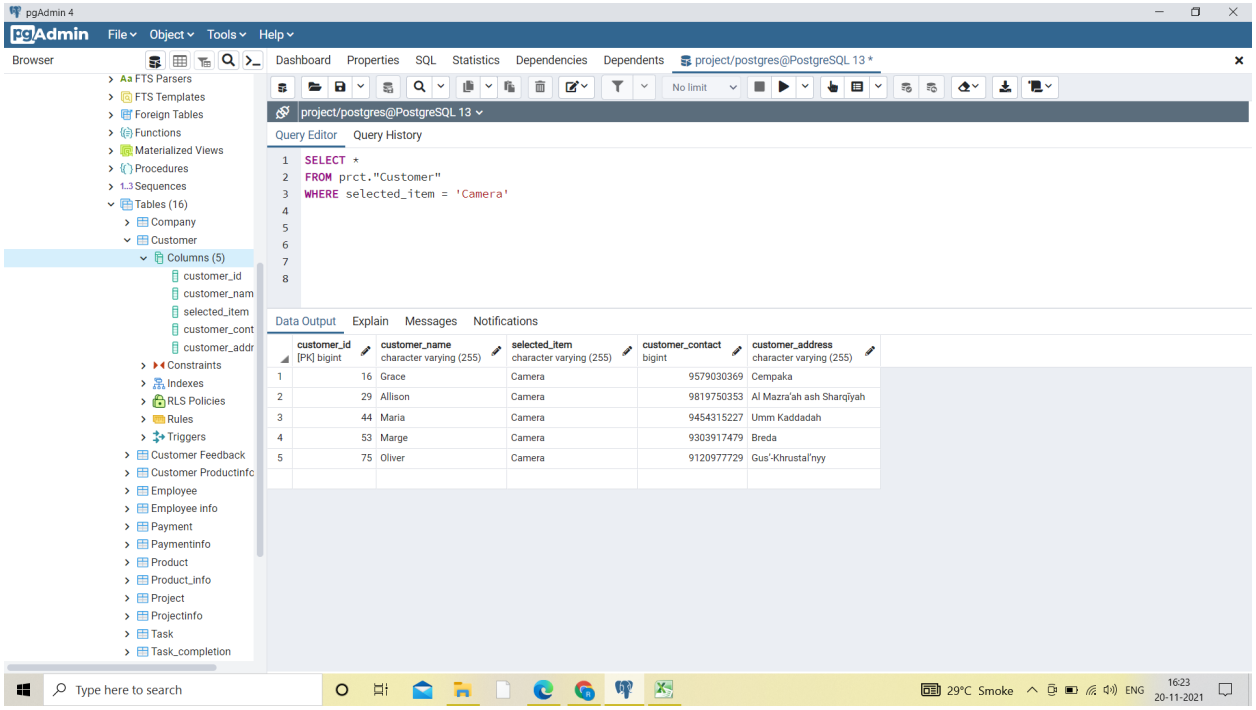
28) Find task from “Task” table which is assigned to employee id 1 to 10.

SELECT task_name FROM prct.”Task” ORDER BY employee_id limit 10



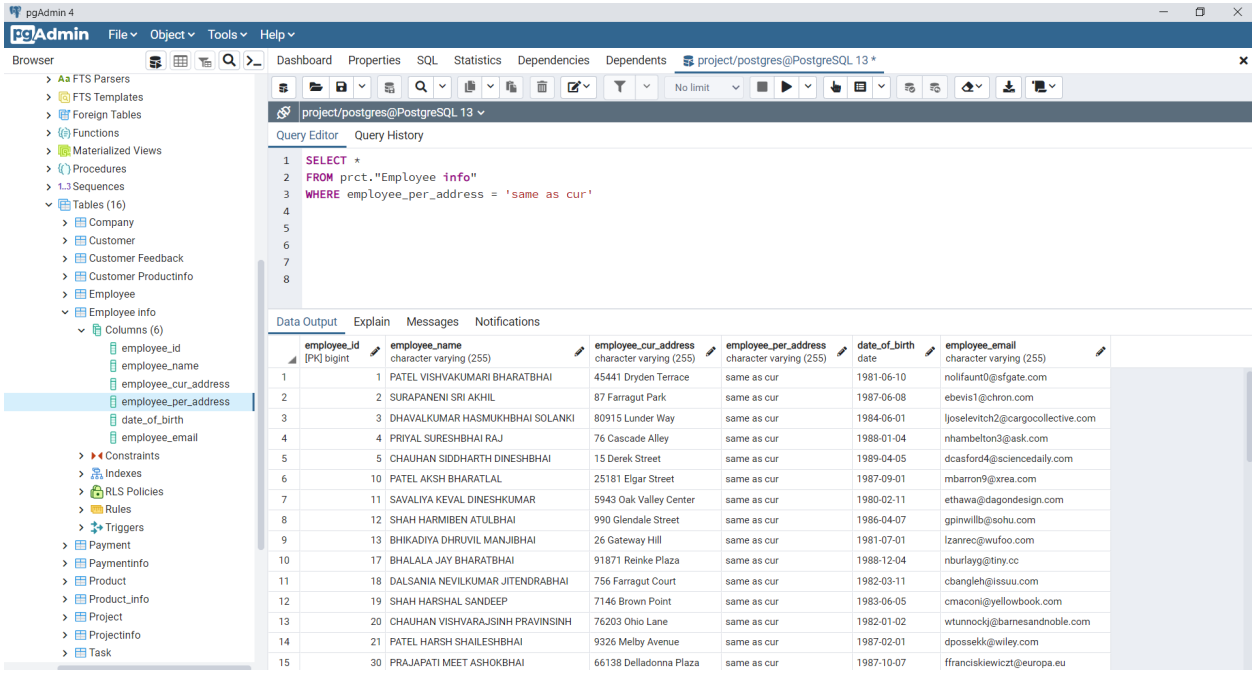
29) Find all the details of customer from “Customer” table who bought camera.

SELECT * FROM prct.”Customer” WHERE selected_item = ‘Camera’



30) Find all the employee whose cur_address is same as per_address.

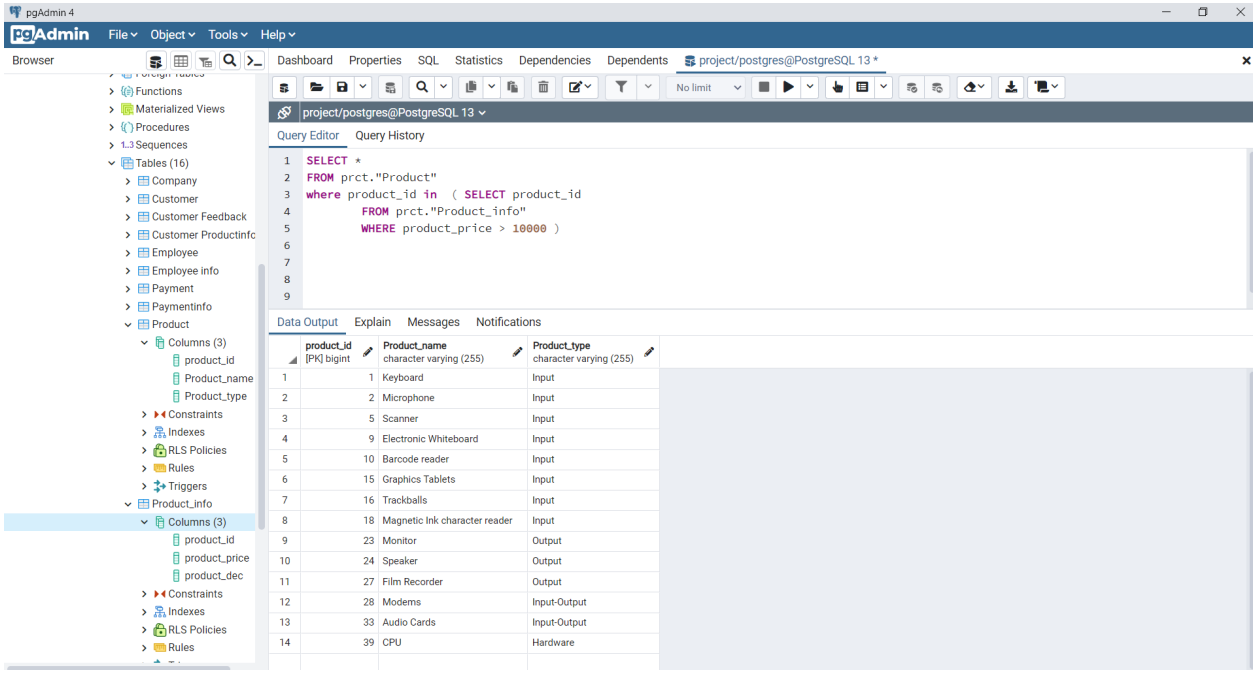
SELECT * FROM prct.”Employee info” WHERE employee_per_address = ‘same as cur’



	employee_id [PK] bigint	employee_name character varying (255)	employee_cur_address character varying (255)	employee_per_address character varying (255)	date_of_birth date	employee_email character varying (255)
1	1	PATEL VISHVAKUMARI BHARATBHAI	45441 Dryden Terrace	same as cur	1981-06-10	nolifaunt0@sfgate.com
2	2	SURAPANENI SRI AKHIL	87 Farragut Park	same as cur	1987-06-08	ebevis1@chron.com
3	3	DHAVALKUMAR HASMUKHBHAI SOLANKI	80915 Lunder Way	same as cur	1984-06-01	ljoselevitch2@cargocollective.com
4	4	PRIYAL SURESHBHAI RAJ	76 Cascade Alley	same as cur	1988-01-04	nhambelton3@ask.com
5	5	CHAUHAN SIDDHARTH DINESHBHAI	15 Derek Street	same as cur	1989-04-05	dcasford4@sciencedaily.com
6	10	PATEL AKSH BHARATLAL	25181 Elgar Street	same as cur	1987-09-01	mbarron9@xrea.com
7	11	SAVALIYA KEVAL DINESHKUMAR	5943 Oak Valley Center	same as cur	1980-02-11	ethawa@dagondesign.com
8	12	SHAH HARMIBEN ATULBHAI	990 Glendale Street	same as cur	1986-04-07	gpinwillb@sohu.com
9	13	BHIKADIYA DHRUVIL MANJIBHAI	26 Gateway Hill	same as cur	1981-07-01	lzanrec@wufoo.com
10	17	BHALALA JAY BHARATBHAI	91871 Reinke Plaza	same as cur	1988-12-04	nburlayg@tiny.cc
11	18	DALSANIA NEVILKUMAR JITENDRABHAI	756 Farragut Court	same as cur	1982-03-11	cbangleh@issuu.com
12	19	SHAH HARSHAL SANDEEP	7146 Brown Point	same as cur	1983-06-05	cmaconi@yellowbook.com
13	20	CHAUHAN VISHVARAJSINH PRAVINSINH	76203 Ohio Lane	same as cur	1982-01-02	wtunnockj@barnesandnoble.com
14	21	PATEL HARSH SHAILESHBHAI	9326 Melby Avenue	same as cur	1987-02-01	dpossekk@wiley.com
15	30	PRAJAPATI MEET ASHOKBHAI	66138 Delladonna Plaza	same as cur	1987-10-07	ffranciskiewiczt@europa.eu

31) Find all the products from product table which have price greater than 10000.

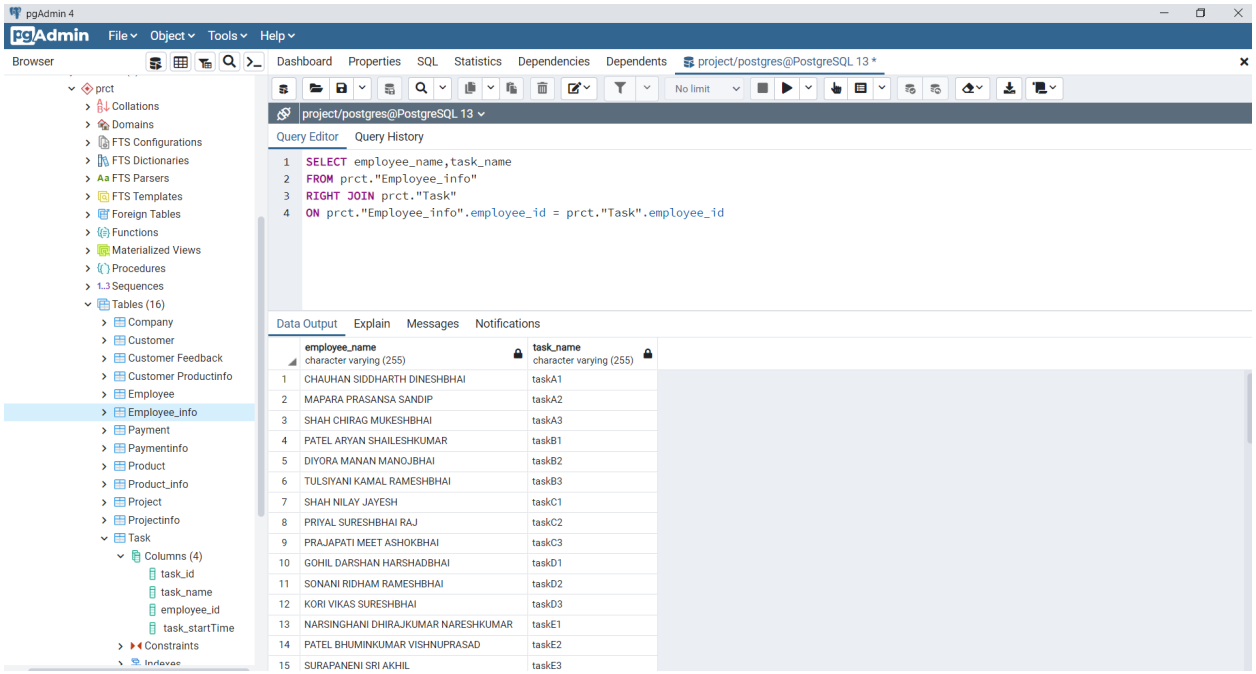
SELECT * FROM prct.”Product” WHERE product_id in (SELECT product_id FROM prct.”Product_info” WHERE product_price > 10000)



	product_id [PK] bigint	Product_name character varying (255)	Product_type character varying (255)
1	1	Keyboard	Input
2	2	Microphone	Input
3	5	Scanner	Input
4	9	Electronic Whiteboard	Input
5	10	Barcode reader	Input
6	15	Graphics Tablets	Input
7	16	Trackballs	Input
8	18	Magnetic Ink character reader	Input
9	23	Monitor	Output
10	24	Speaker	Output
11	27	Film Recorder	Output
12	28	Modems	Input-Output
13	33	Audio Cards	Input-Output
14	39	CPU	Hardware

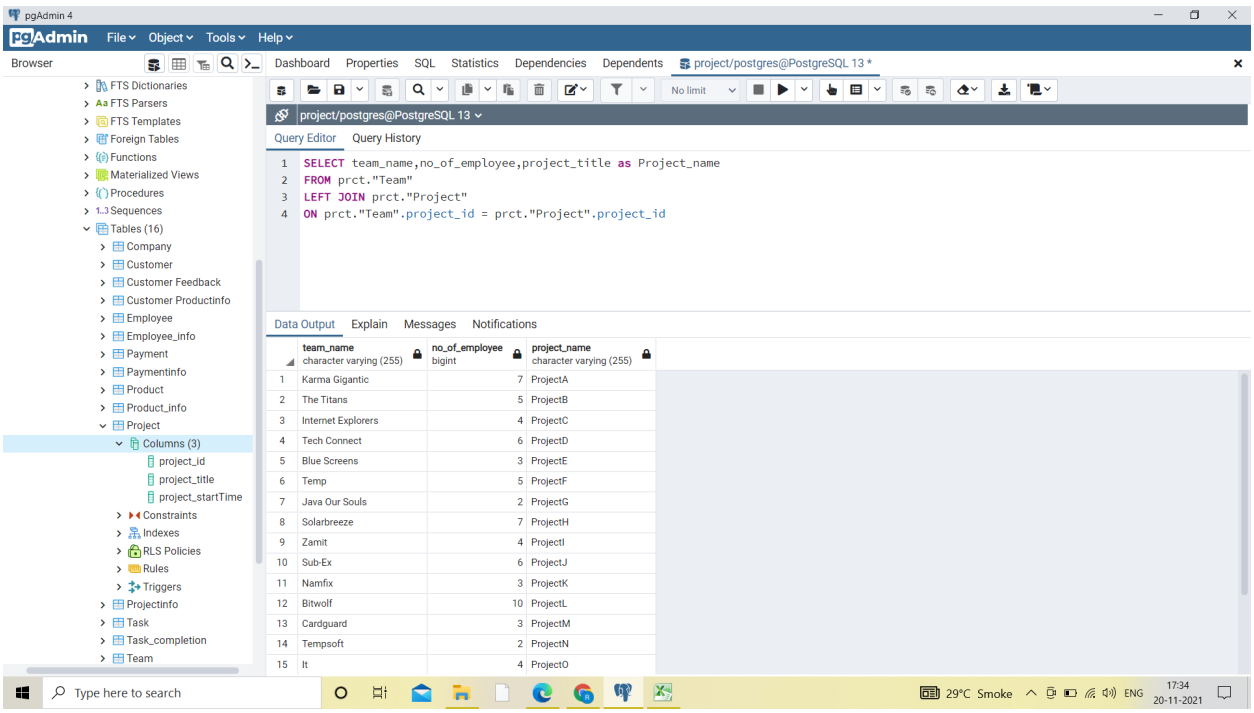
32) Perform Right Outer Join between “Employee_info” table and “Task” table. Find employee_name from “employee_info” table and task_name from “Task” table.

SELECT employee_name,task_name FROM prct.”Employee_info” RIGHT JOIN prct.”Task” ON prct.”Employee_info”.employee_id = prct.”Task”.employee_id



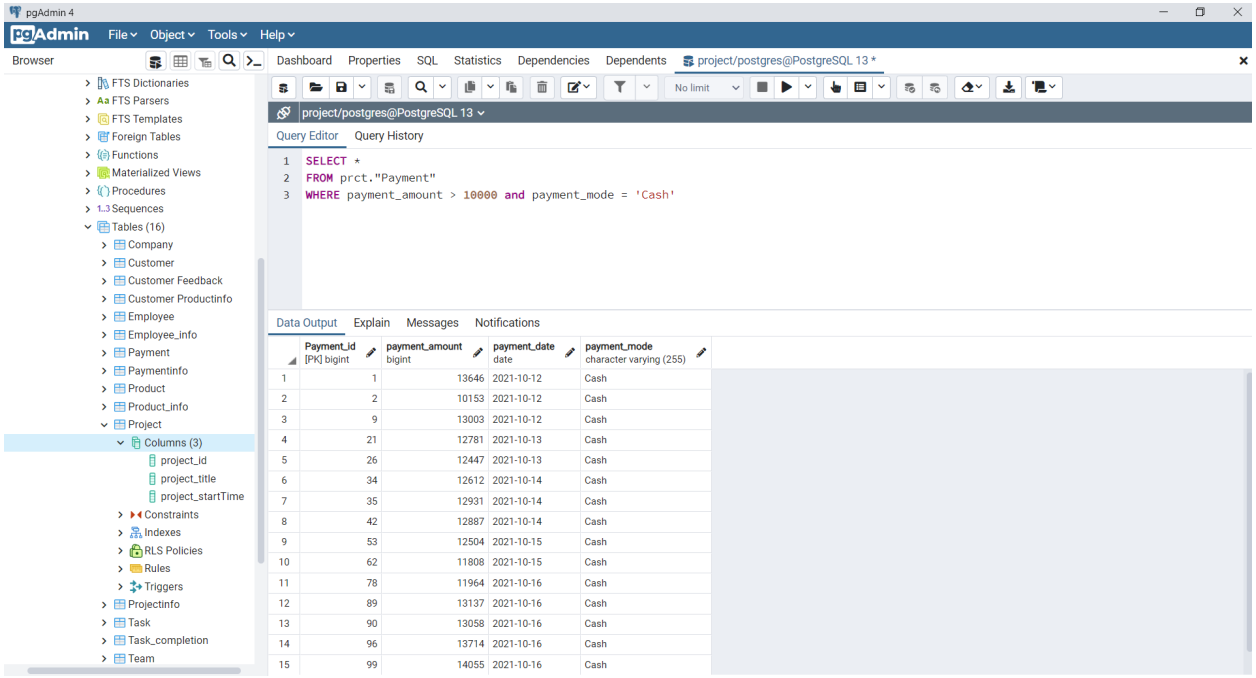
33) Perform Left Outer Join between “Team” table and “Project” table and find team_name, no_of_employee from “Team” table and project_name from “Project” table.

SELECT team_name,no_of_employee,project_title ad Project_name FROM prct.”Team” LEFT JOIN prct.”Project” ON prct.”Team”.project_id = prct.”Project”.project_id



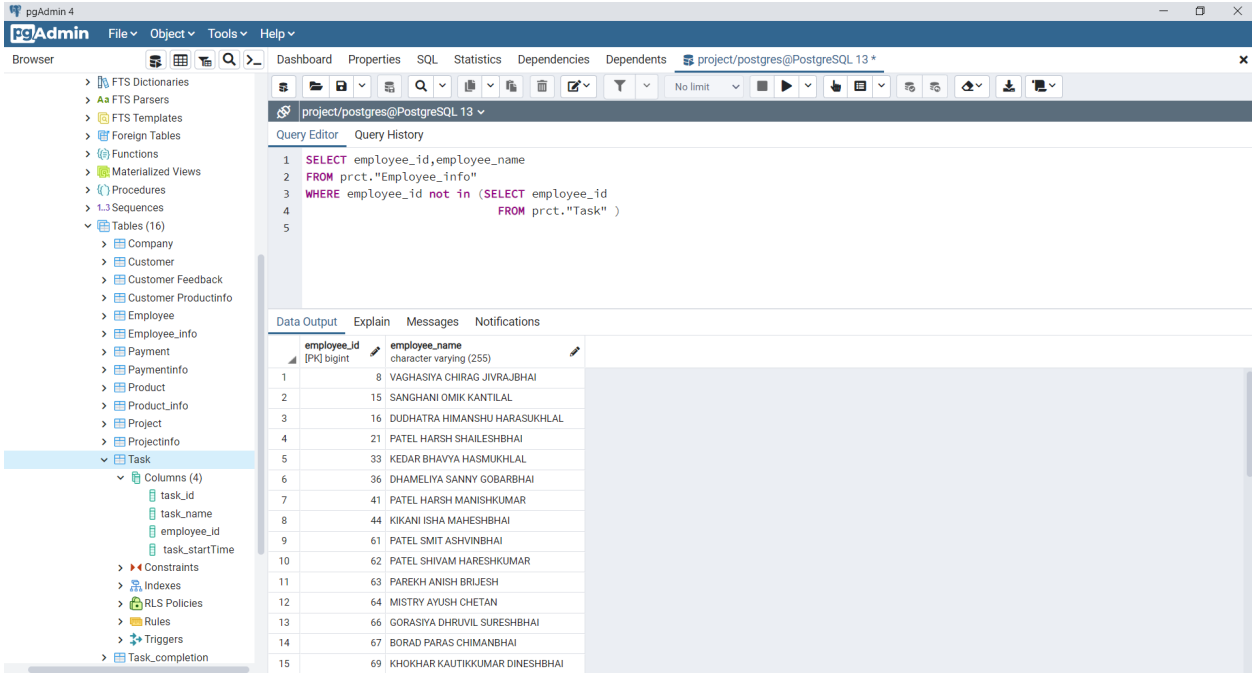
34) Find all the payment where payment_amount is greater than 10000 and payment_mode is Cash.

SELECT * FROM prct.”Payment” WHERE payment_amount > 10000 and paymet_mode = ‘Cash’

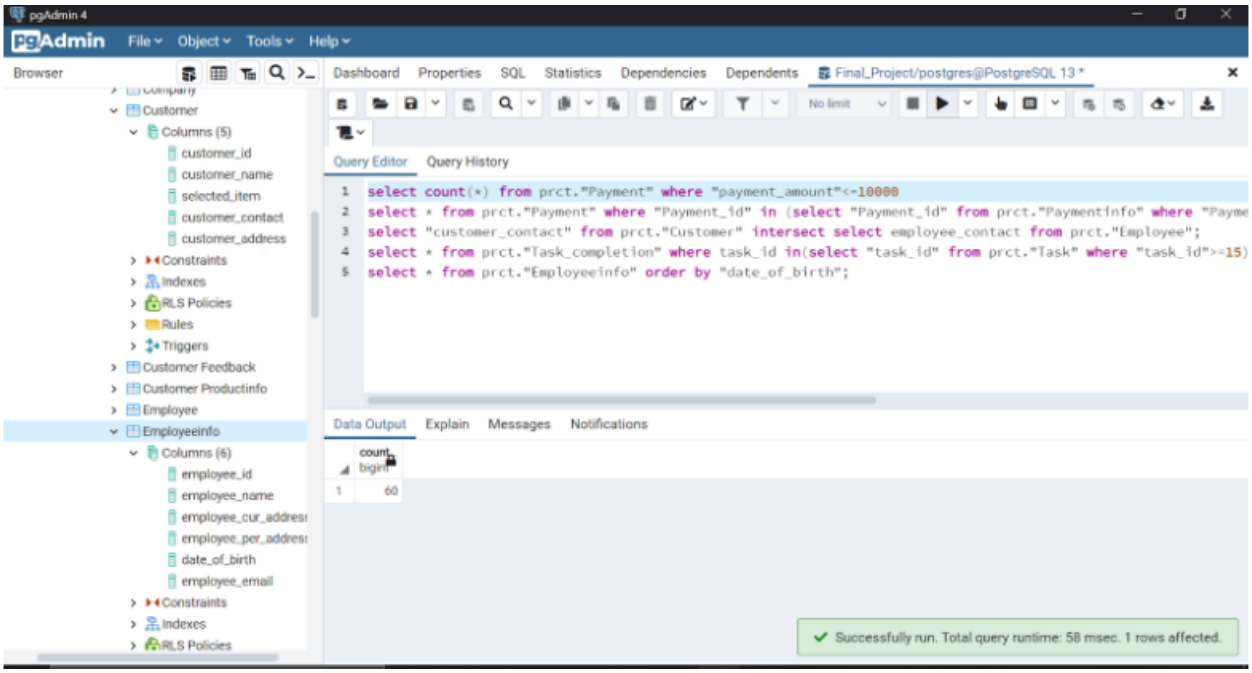


35) Find all employees who has not assigned any task.

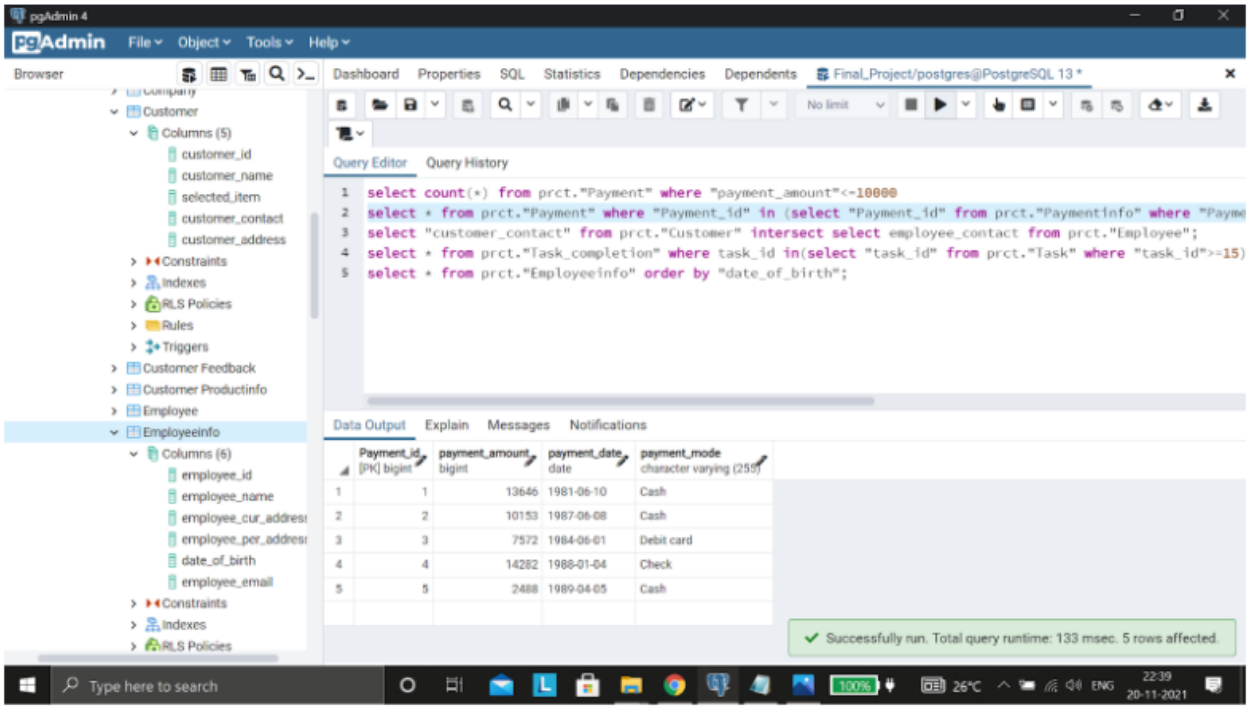
SELECT employee_id,employee_name FROM prct.”Employee_info” WHERE employee_id not in (SELECT employee_id FROM prct.”Task”)



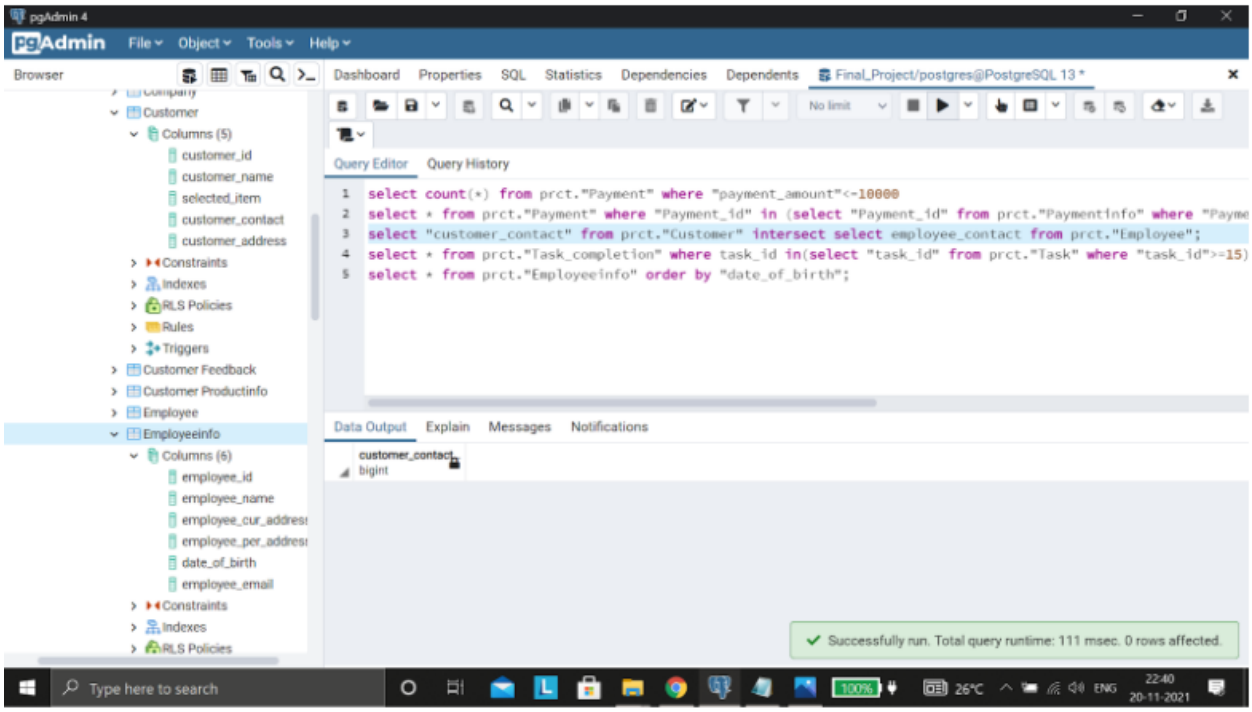
36) select count(*) from prct."Payment" where "payment_amount"<=10000



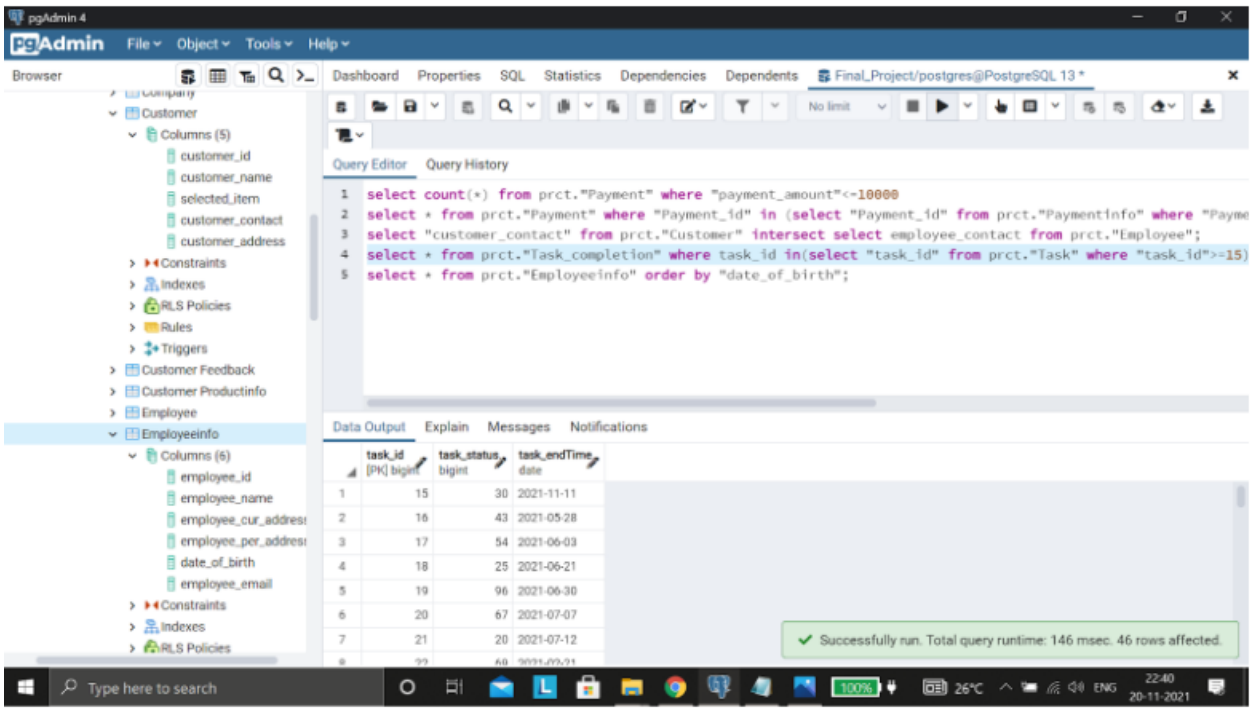
37) select * from prct."Payment" where "Payment_id" in (select "Payment_id" from prct."Paymentinfo" where "Payment_id" in (select "employee_id" from prct."Employeeinfo" where "employee_id" <=5));



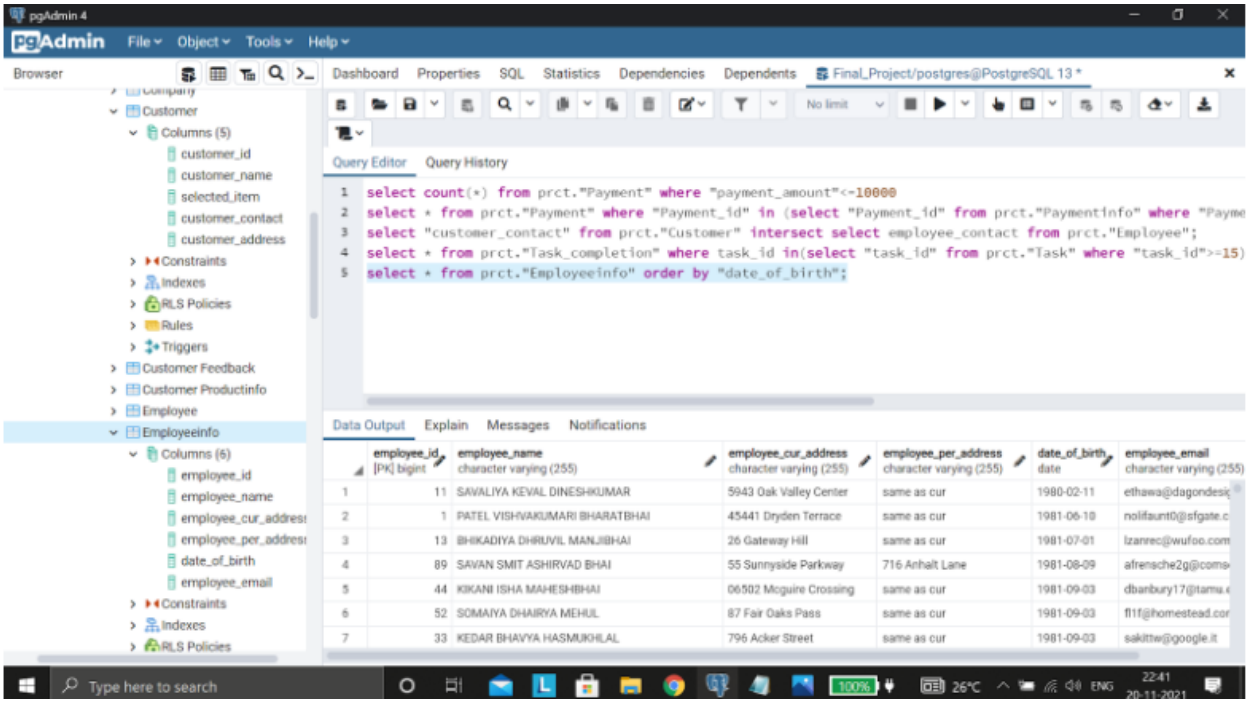
38) select "customer_contact" from prct."Customer" intersect select employee_contact from prct."Employee";



39) select * from prct."Task_completion" where task_id in(select "task_id" from prct."Task" where "task_id">=15)



40) select * from prct."Employeeinfo" order by "date_of_birth";



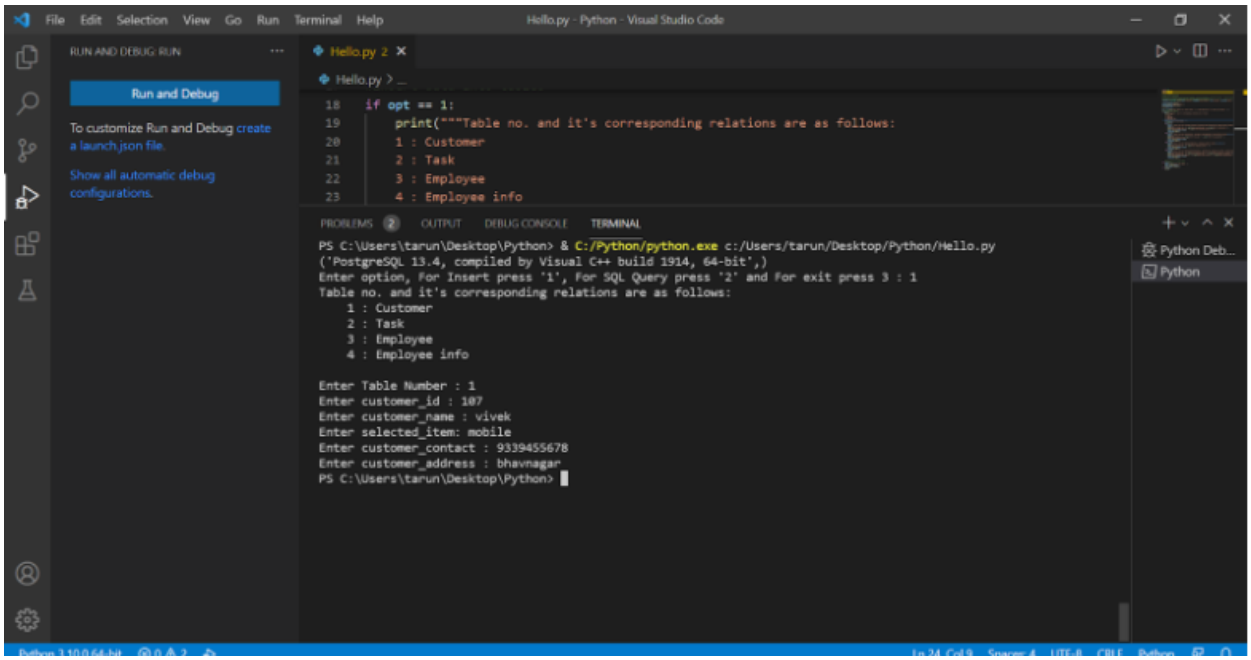
Section 7
Project code with output screenshots

Tool: - Visual Studio
Language: - Python
Code: -

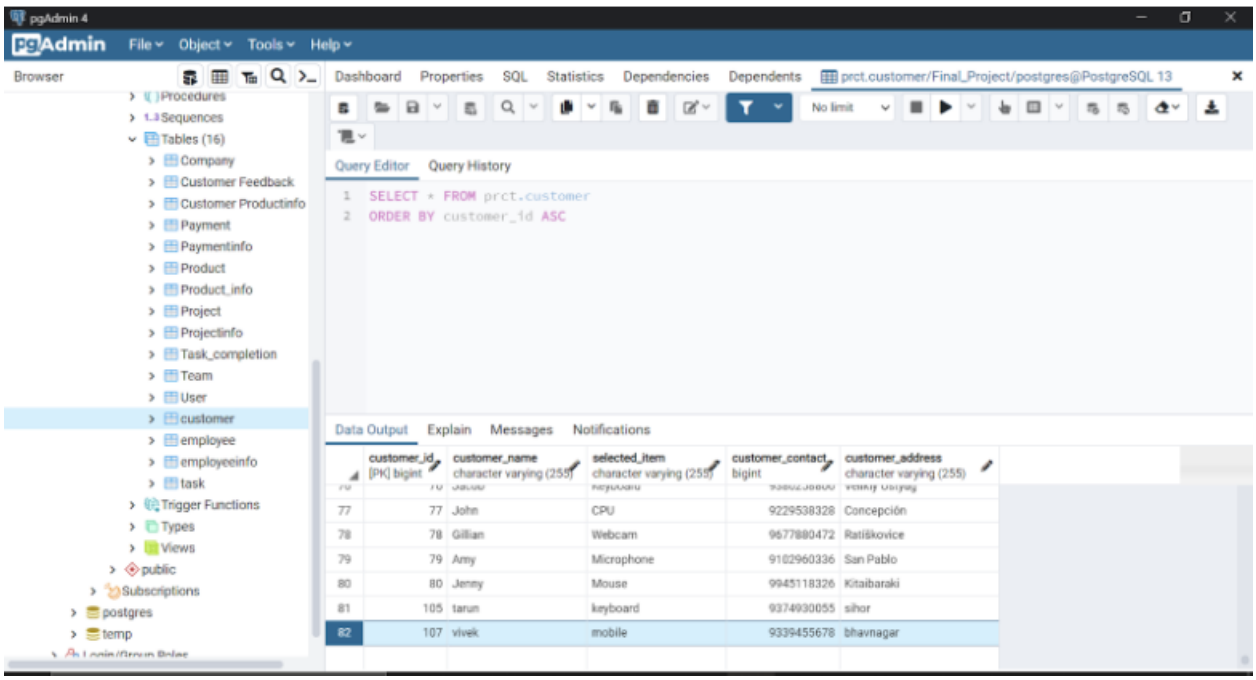
```
import time
import pymonetdb
import psycopg2
#lpip3 install paymonetdb

#connection = pymonetdb.connect(username="monetdb", password ="monetdb",
hostname="Localhost", database="demo")
#conn=psycopg2.connect("dbname=suppliers user=postgres
password="postgres")
connection = psycopg2.connect(host="localhost", database="Final_Project",
user="postgres", password="admin")

cursor=connection.cursor()
cursor.execute('SELECT version()')
db_version=cursor.fetchone()
print(db_version)

opt = int(input("Enter option, For Insert press '1', For SQL Query press
'2' and For exit press 3 : "))
#insert data into tables
if opt == 1:
    print("""Table no. and it's corresponding relations are as follows:
    1 : Customer
    2 : Task
    3 : Employee
    4 : Employee info
    """)
    val=input("Enter Table Number : ")
    v = int(val)
    if(v == 1):
        Insert = 'Insert into
prct.customer(customer_id,customer_name,selected_item,customer_contact,cus
tomer_address) values(%s,%s,%s,%s,%s)'
        value = (
```

```
        input("Enter customer_id : "), input("Enter customer_name : "),
input("Enter selected_item: "),
        input("Enter customer_contact : "), input("Enter customer_address
: "))
        cursor.execute(Insert, value)
        connection.commit()
    elif(v == 2):
        Insert = 'Insert into prct.task(task_id, task_name,
task_employeeid, task_startTime) values(%s,%s,%s,%s)'
        value = (
        input("Enter task_id : "), input("Enter task_name : "),
input("Enter task_employeeid: "),
        input("Enter task_startTime : "))
        cursor.execute(Insert, value)
        connection.commit()
    elif(v == 3):
        Insert = 'Insert into prct.employee(employee_id, employee_contact)
values(%s,%s,%s,%s,%s)'
        value = (
        input("Enter employee_id : "), input("Enter employee_contact : "))
        cursor.execute(Insert, value)
        connection.commit()
    elif(v == 4):
        Insert = 'Insert into prct.employeeinfo(employee_id,
employee_name, employee_cur_address, employee_per_address, date_of_birth,
employee_email) values(%s,%s,%s,%s,%s,%s)'
        value = (
        input("Enter employee_id : "), input("Enter employee_name : "),
input("Enter employee_cur_address: "),
        input("Enter employee_per_address : "), input("Enter date_of_birth
: "), input("Enter employee_email : "))
        cursor.execute(Insert, value)
        connection.commit()
    #Execute query of User's choice
elif opt == 2:
    Query = input("Enter your query : ")
    cursor.execute(Query)
    print(cursor.fetchall())
elif opt == 3:
    exit(0)
```

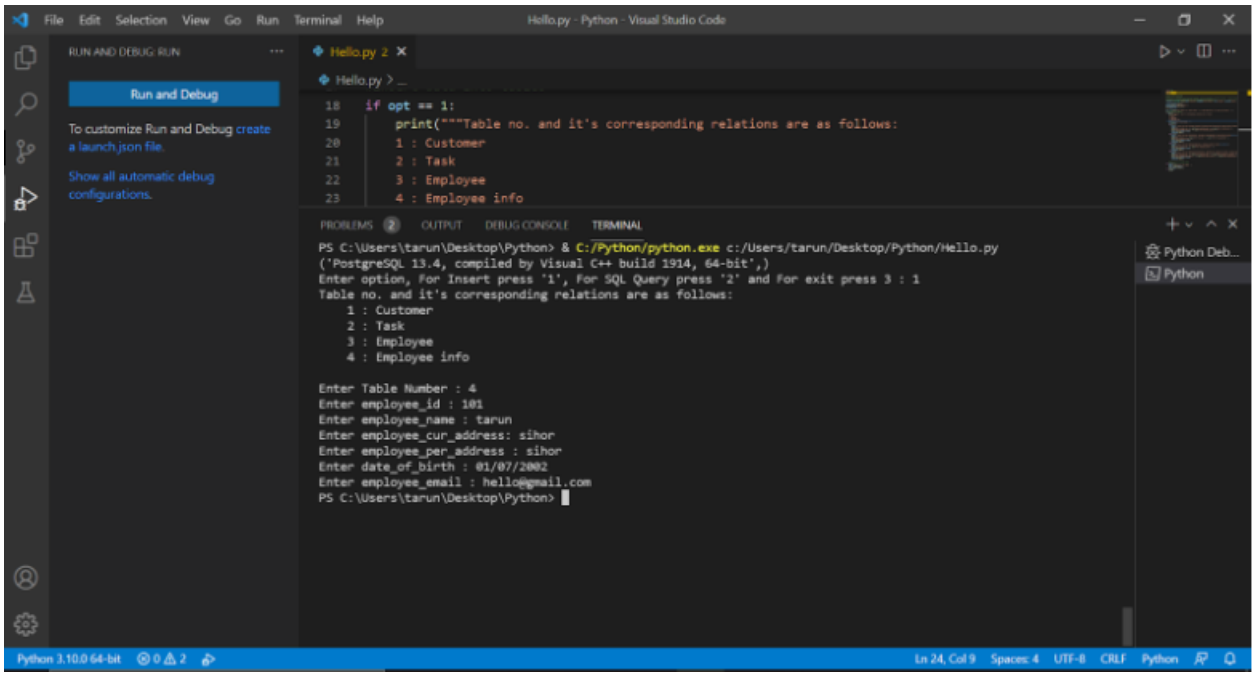
- **Insert into Customer: -**



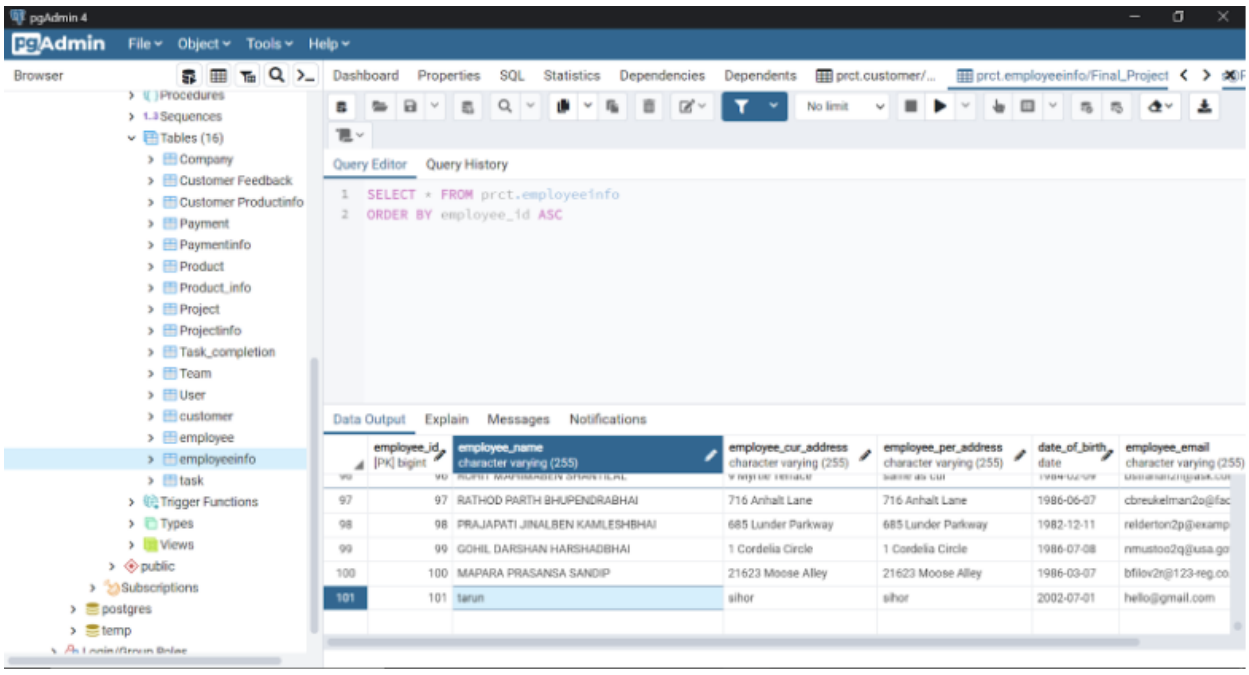
- **Output**: -



- **Insert into Employee info: -**

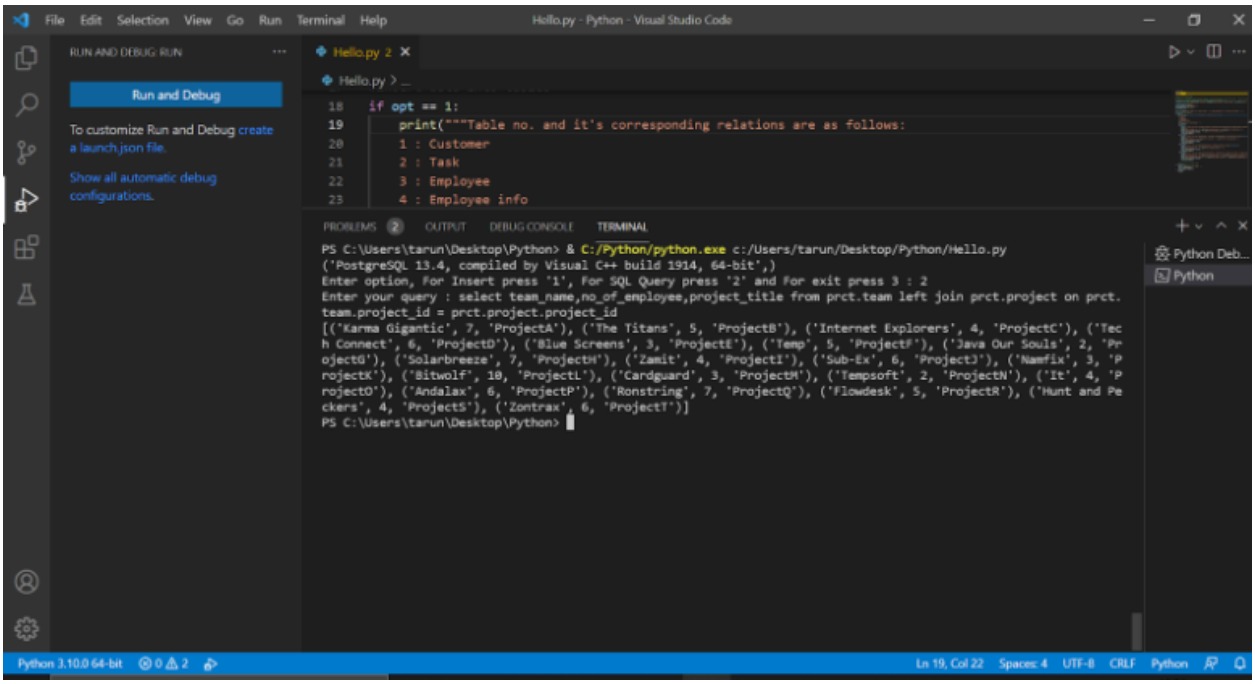


Output: -

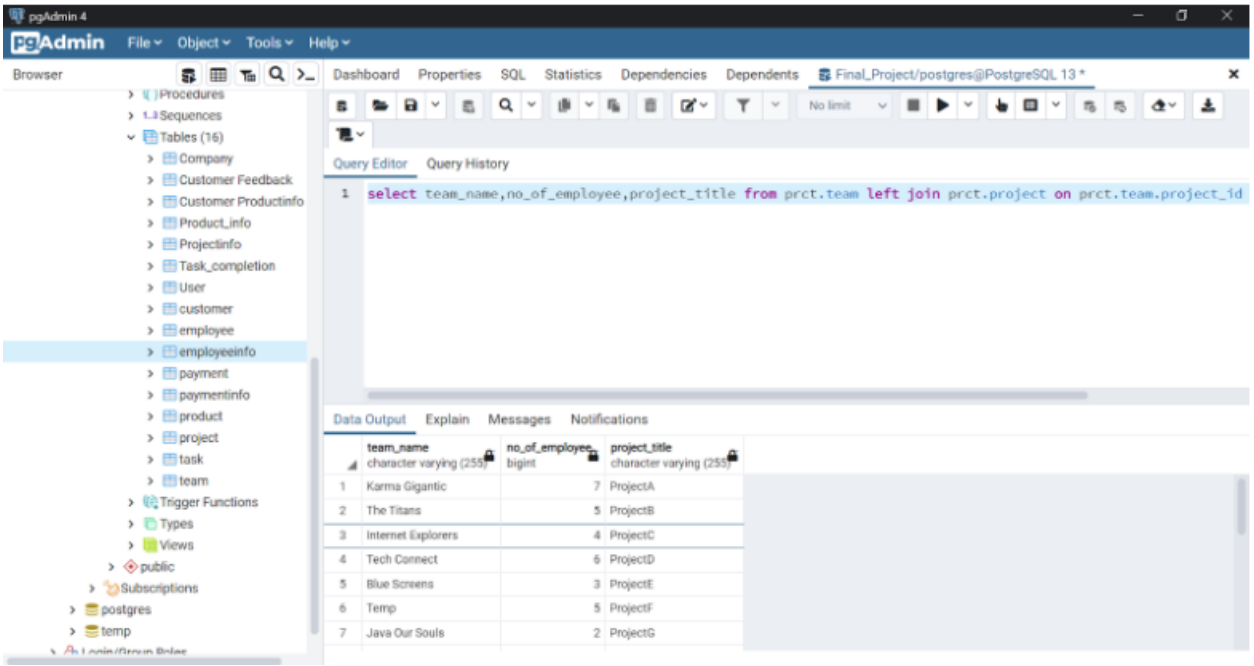


Execute Query: -

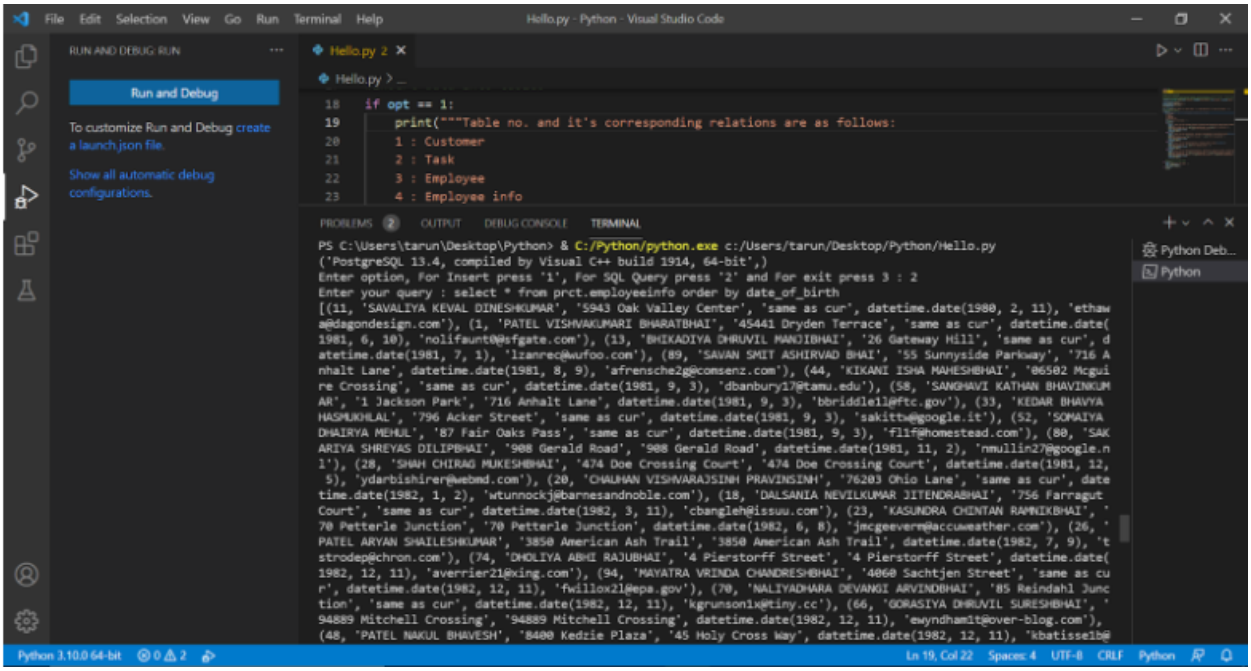
- select team_name,no_of_employee,project_title from prct.team left join prct.project on prct.team.project_id = prct.project.project_id



- Output: -



- select * from prct.employeeinfo order by date_of_birth



- Output: -