Database Management System LAB 10

LabourList and wages

KANPARIYA CHINTAN MANISHBHAI
202001463

SMIT P. BHAVSAR
202001464

Team ID: 18

Case Study/Problem

INDEX

Section 1 : Final SRS version

1. Purpose

- A city includes more than one district and every district has a few labor supervisors.
- The reason of a labor manager is to supervise the work of each laborer that works below him.
- There are two types of wages Depending on the form of works, wages may be,weekly salary or an hourly salary.
- The hourly salary relies upon the entire range of hours spent in a day with the aid of using unique labor, whilst there may be no such limit for a weekly salary as labor could be paid on a weekly basis.
- We want to hold a file of all of the works achieved with the aid of using labor on a every day each and every district wise foundation to shop and distribute Wages.
- Main purpose is to simplify this LabourList and wages management to maintain daily wages, labor and supervisor details. From daily anonymous attendance to regular labor attendance. Inspect the safety and standards to work on the field for workers to avoid unwanted damages. Supervisor can Track work orders for various sites and the various locations at a single view.

2. Intended Audience and Reading Suggestions

- The Supervisor of the district can access the database for monitoring the progress of the works and can modify weekly or hourly wages of all labor.
- The head of the supervisor can modify supervisor salary.
- If a labor want to get high salary then labor can request to supervisor.
- The Supervisor can keep track of the requirements of precautionary things for particular work, working status of the activity and revenue generated from particular activity.
- Labor can see their wages, their working day etc.
- Labor can apply for leave and s/he will not get wages for that day.
- Supervisor also apply for leave.
- In case of accident / threat, man count & individual details can be traced and actions can be taken towards them.

- An adult worker shall work over 9 hours per day or 48 hours per week and overtime shall be double the regular wages. A female worker can work from 6 am to 7 pm.
- Equal compensation for equal work must be paid regardless of gender.
- The living wage is higher than the minimum wage and is designed so that a full-time worker should be able to support themselves and a small family at that wage.
- Recording of wage and labor-related data (registration of personnel information,holiday and sick leave balances).

Minimum Wages:

• Min wage needs to be paid at least the national/local/industrial minimum wage as per the law
• Min wage are reviewed and revised annually
• Min wage should be paid only for basic working hours

Payment of Wages:

• Wages need to be paid on time as per legal law
• No under-payment of basic wages, bonuses/benefits and overtime
• Social security payments need to be paid to concerned authorities on time as per law
• Workers have direct access to their wages
• Workers have a bank account where they receive their wage (preferably)
• Worker know how their wages are calculated
• Workers know how to read the pay slip
• Pay slip is provided in local language
• Pay slip includes all relevant wage information (holidays, days and hours of work, all components of wage, deductions, rest days, incentives or bonuses)
• Deduction should be legal/ compliance with legal law
• Any deduction, other than legal, should be done with consent of worker
• Wages should not be withheld

Payment of Working Hours:

• Workers are provided one day off in seven days
• Workers are provided breaks during working periods
• Facility has a formal request or approval process for urgent

Recruitment

• No child labour

• No forced labour
• Induction training for workers and supervisor

3. Product Scope

- It can create opportunities for unemployed labour.
- It can keep all records of wages which paid to labor and also keep record of all labor by different labor work.
- It enables paperless record keeping & calculations avoiding human errors,labor id, license no, expiry document with alert notifications,complete shift, overtime, comp- off management as per configurable company business rules.
- It contains information about right number of people, right kind of people at the right place, right time, doing the right things for which they are suited for the achievement of goals of the organization.

4. Description

The system contains the following tables:

1. **Regions**
2. **Company_deatils**
3. **Labors**
4. **Work_history**
5. **Department**
6. **Locations**
7. **Leave**
8. **Wages**
9. **Supervisor**
10. **Manager**
11. **Owner**

The user can get the following information about the project from the given system:

- **Regions:** region_id,region_name
  ➔ In this table region_id is primary key

This shows the basic details about the region of the company.

- **Company_details:**company_id,company_type,street_add,highest_no_worker,region_id
  - ➔ In this table company_id is primary key.

  Here all the labors and supervisors can view the company details if they want to work in a particular company.

- **Labors:**labor_id,first_name,last_name,gender,phone_no,dept_id,company_id,supervisor_id,age,wage,hire_date,region_id,address,labor_type
  - ➔ In this table labor_id is primary key; dept_id and company_id are foreign keys.

  Here all labors and supervisors of the company/factory can view basic details/information about labor.

- **Work_history:** labor_id,start_date,end_date,dept_id,company_id
  - ➔ In this table labor_id is primary key; company_id and dept_id are foreign keys.

  Company/factory can view the work history of all labors like start date and end date.

- **Department:** dept_id,name,wages_range,company_id
  - ➔ In this table dept_id is primary key and company_id is foreign key.

  Here supervisor or owner can modify the wages range according to requests of wages by labors.

- **Locations:** street_add,pin_code,city,state,country,region_id
  - ➔ in this table pin_code is primary key and region_id is foreign key

  Here labors can select and view their work locations.

- **Leave:** leave_id,labor_id,date,reason
  - ➔ in this table leave_idis primary key and labor_id is foreign key

  Here labors can apply or request for leave due to any emergency or illness only.

- **Wages:**wages_id,labor_id,dept_id,payment_id,date,receipt,total_amount,wages_type
  - ➔ in this table wages_id is primary key and labor_id is foreign key

  Here supervisors can see wages details of their labors which have already been paid and Labors can view receipt of their wages.

- **supervisor:**supervisor_id,f_name,l_name,gender,phone_no,dept_id,company_id,salary,hire_date,region_id,manager_id,address
  - ➔ in this table supervisor_id is primary key; dept_id and company_id are foreign keys.

  Here all supervisors of the company/factory can view basic details/information about labor.

- **Manager:**manager_id,f_name,l_name,gender,phone_no,dept_id,company_id,salary,hire_date,region_id,address
  - ➔ in this table manager_id is primary key; dept_id and company_id are foreign keys.

  Here all Manager of the company/factory can view basic details/information about labor and supervisor.

- **Owner:**owner_id,f_name,l_name,gender,phone_no,company_id,address
  - ➔ in this table owner_id is primary key;company_id is foreign keys.

  Owner of the company/factory can view basic details/information about manager, labor and supervisor.

+

Refrence:

https://hmgroup.com/wp-content/uploads/2020/10/Wage-Management-System-Guideline.pdf

https://www.sedex.com/wp-content/uploads/2020/05/Sedex-1.1-Labour-Management-Control-EN.pdf

2. Document the Requirements Collection/ Fact Finding Phase

ii. Background Reading/s

1. Description of each reading and references:

- Database System Concepts(Seventh edition), by Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- From this book, we understood the DBMS concepts like E-R Model, Database designing etc.
- https://www.sedex.com/wp-content/uploads/2020/05/Sedex-1.1-Labour-Management-Control-EN.pdf

2. List the combined Requirements gathered from Background Reading:

● A Database management system is required to maintain and update all the information about labors, wages and the company details.
● All the basic facilities that are expected to be available like keeping a record of labors and their contact details, etc.
● Company owner and the Supervisor will be assigned different roles that will ensure the security of labors.
● Labors will be assigned different types of view permissions to show details about their wages,wages type and company details in which they are working

(iii) Interview

Interview Plan
System : LaborList and Wages
Project Reference :
https://hmgroup.com/wp-content/uploads/2020/10/Wage-Management-System-Guideline.pdf
Interviewee : Mit Sharma(Role Play)
Designation : Supervisor at cement factory

Interviewer:
1)Chintan Kanpariya
Designation : Business Development Executive

2)Smit Bhavsar
Designation : Developer

Date: 28/9/2022 Time : 14:30

Duration : 45 minutes Place: Google Meet

Purpose of Interview :
Preliminary meeting to identify problems and requirements regarding safety of the Labors.

Agenda :

- Introduction and current status
- Initial ideas and follow-up action regarding the laborlist and wages management at the company.
- Need of labor according to their gender.
- About the needs and complaints of the labors and how you handle those issues.

Documents to be brought to the Interview:

- Any documents relating to the current activities and current needs of the labors.
- List of Precautionary things required for each working activities.

Interview Summary:

System : LaborList and Wages
Project Reference :
https://hmgroup.com/wp-content/uploads/2020/10/Wage-Management-System-Guideline.pdf

Interviewee : Mit Sharma(Role Play)
Designation : Supervisor at cement factory

Interviewer:
1)Chintan Kanpariya

Designation : Business Development Executive

2)Smit Bhavsar
Designation : Developer

Date: 26/09/2022 **Time:** 3:25

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

Purpose of Interview:
Preliminary meeting to identify problems and requirements regarding working activities of labors and their wages.

Results of the Interview:

- Regular feedback from supervisor about labors and their works will be taken and improvements will be done to the working activities accordingly.
- For the security of labors, the machines for work will be checked from time to time.
- Women have different needs to men that should be considered when developing processes around labour management – for example women may be more likely to experience harassment or discrimination at work due because of their gender.
- Strict rules will be implemented so that labors will not be allowed to do some working activities that are not suitable for their age.
- Training and communication should consider language needs of migrant workers – if migrant workers are not fluent in your workplace language, they will need health and safety instructions translated and explained to them in their own language, in order to be safe.

Interview Plan

System: LaborList and Wages

Project Reference:
https://hmgroup.com/wp-content/uploads/2020/10/Wage-Management-System-Guideline.pdf
Interviewee:

1)Ved Patel(Role Play)
Designation: Labor at Turner Construction

Interviewer:
1)Chintan kanpariya
Designation : Business Development Executive

2)Smit Bhavsar
Designation : Developer

Date: 28/09/2022 **Time**: 5:00

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

Purpose of Interview:
● The overall opinion of the working activities of labor and their wages.

Agenda:
● Introduction and current status
● Current labor working activities and wages are satisfactory or not for labors.
● Working Activities of labors or wages that are dissatisfactory or need improvement.
● work that were more suitable to labors.

Interview Summary:
System: LaborList and Wages
Interviewee:
1)Ved Patel(Role Play)
Designation: Labor at Turner Construction

Interviewer:
1)Chintan Kanpariya
Designation: Business Development Executive

2)Smit Bhavsar
Designation: Developer

Date: 26/09/2022 **Time**: 5:45

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

Purpose of Interview:

● The overall opinion of the labors,their working activities and wages.

Results of the Interview:

- Employment should be freely chosen.
- No discrimination should be practised.
- The ability of labors to do the work.
- Working conditions should be safe and hygienic.
- Working hours should not excessive.
- Living wages should be paid.
- Life insurance should be provided.
- No harsh of inhumane treatment should be allowed.

Combined Requirements gathered from the Interviews:

● Feedback should be received from the labors groups about different aspects of the working activities and suggestions.

● For the safety of labors, the machines for work will be checked from time to time.

● Company/factory should provide living rent and life insurance to their labors.

iv. Questionnaires

Google form link: https://forms.gle/x2hfkhxXw7AMDU7dA

LaborList & Wages

This survey will help us to build database of LaborList and their Wages and also it will be helpful for improvements in our database design.

202001463@daiict.ac.in (not shared) Switch account

* Required

According to you, How important is to create this software ? *

	1	2	3	4	5	
Not important	○	○	○	○	○	Extremely important

Which functionalities you would prefer to have in this software ? *

- [ ] Basic Information of company/factory
- [ ] amount of wages to be given to labors by company/factory
- [ ] labor request to work in company/factory and allocation of work which suited for labor
- [ ] Requests for improve wages by labors
- [ ] Request for leaves by supervisors or labors
- [ ] Any deduction in wages against indisciplinary action by labors
- [ ] working status of the activity and revenue generated from particular activity
- [ ] Other: _____

According to you, How frequently would company/factory use this software ? *

	1	2	3	4	5	
Hardly Sometimes	○	○	○	○	○	Very often

What do you think that Which company/factory information would labor require most often? *

- ◯ Private
- ◯ Government

What do you think that which type of wage would be preferred by labor? *

- ◯ Hourly Wage
- ◯ Weekly Wage

What do you think that Which type of payment method would be preferred by labor? *

- ◯ Online Payment
- ◯ Cash

According to you , should company/factory provide life insurance to labor? *

- ◯ Yes
- ◯ No

According to you , should company/factory give work below 18 age person? *

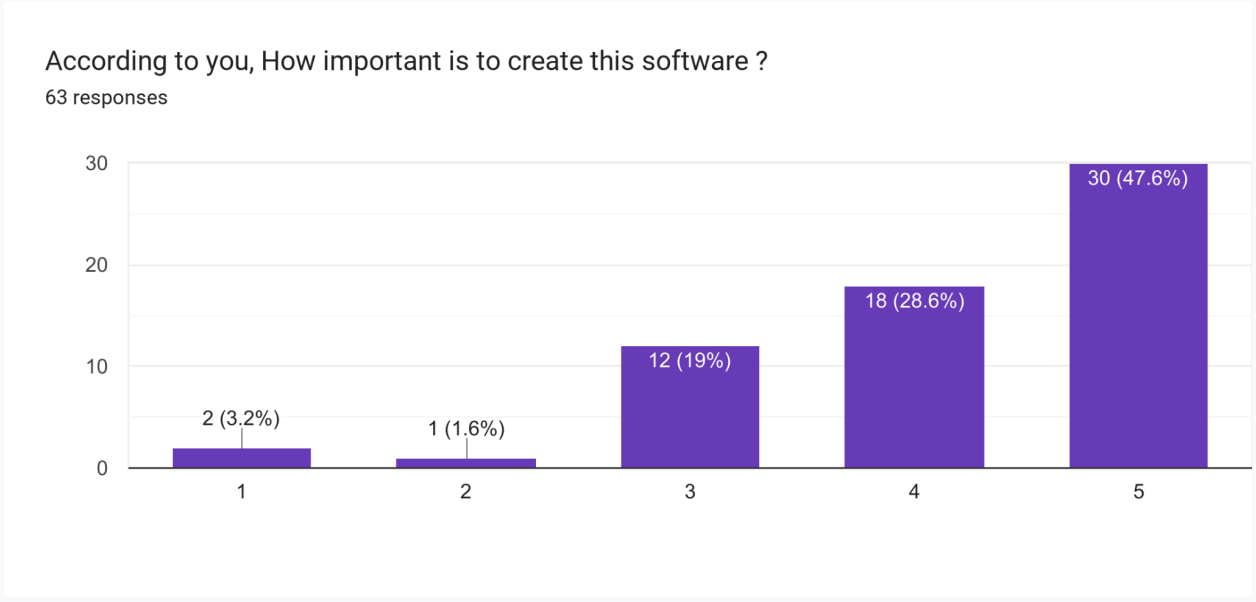
- ◯ Yes
- ◯ No

Any other Suggestions for the software ?

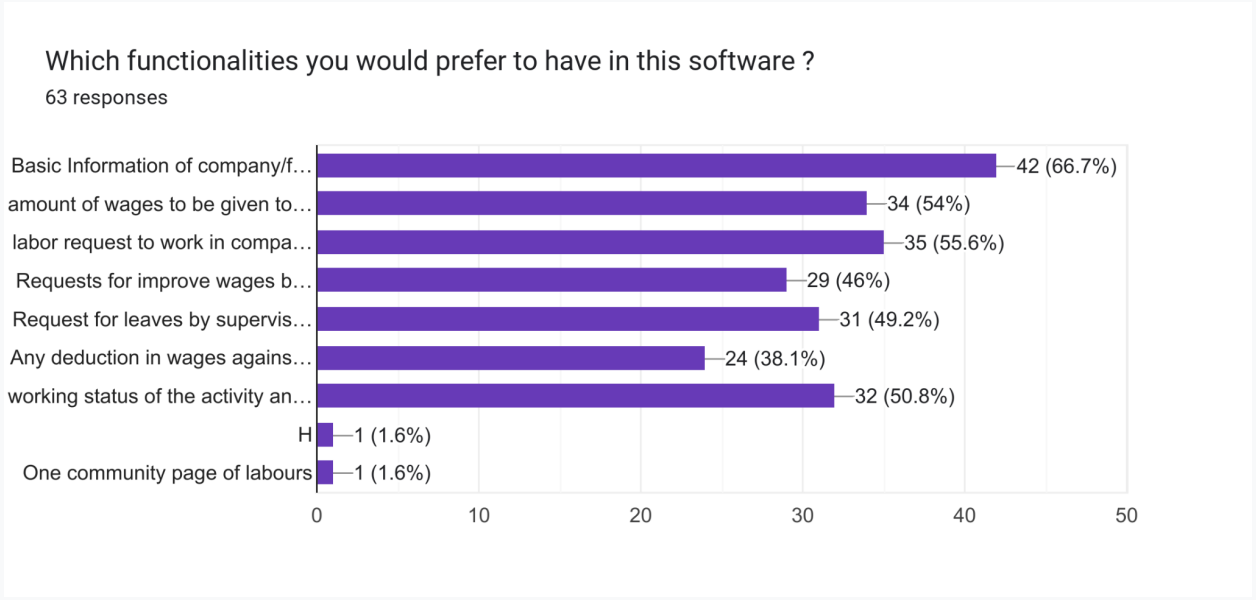
Your answer

Responses we got from the above questionnaire:

1



2

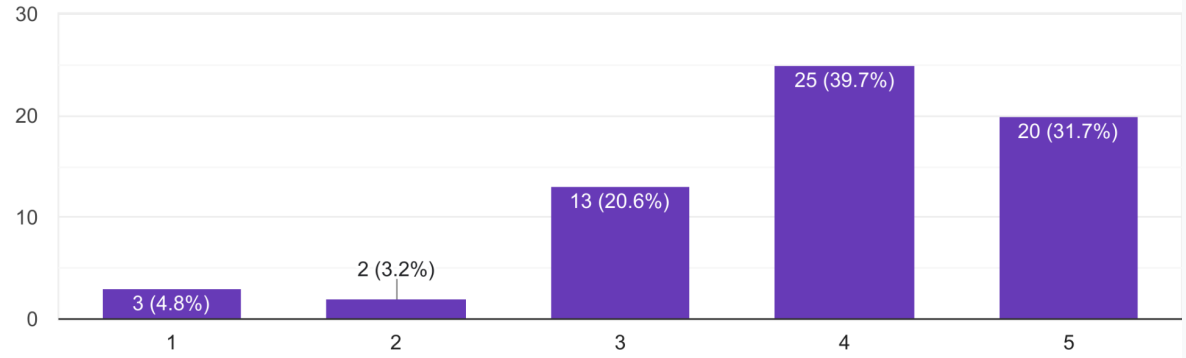


\

3

According to you, How frequently would company/factory use this software ?

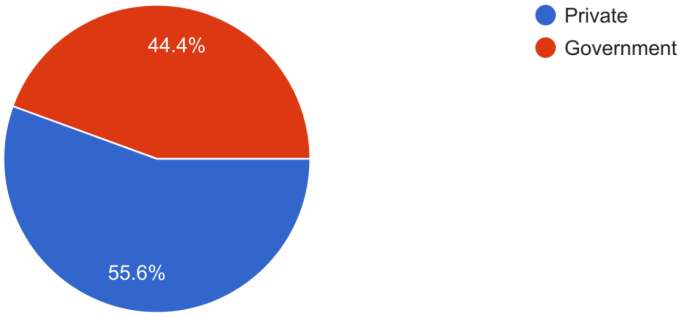
63 responses



4

What do you think that Which company/factory information would labor require most often?

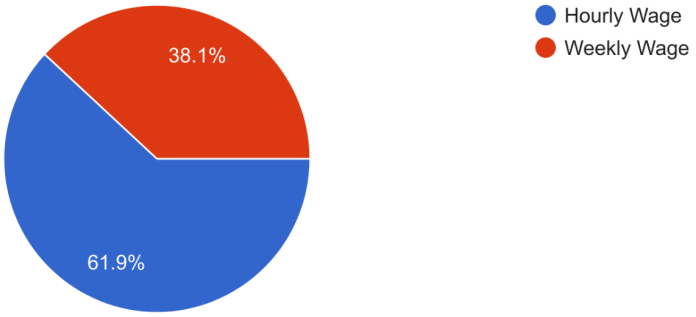
63 responses



5

What do you think that which type of wage would be preferred by labor?

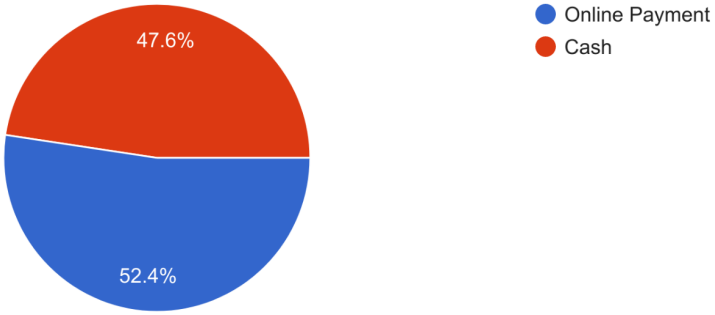
63 responses



6

What do you think that Which type of payment method would be preferred by labor?

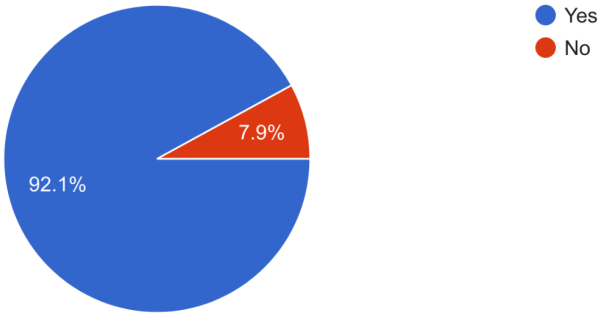
63 responses



7

According to you , should company/factory provide life insurance to labor?

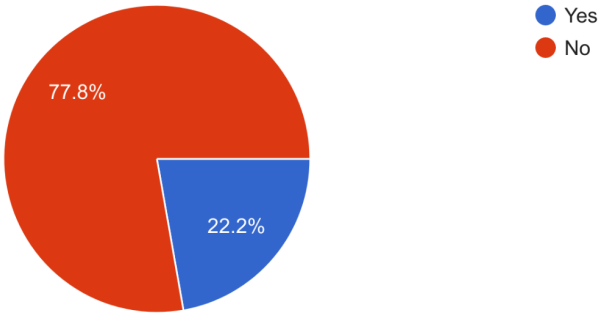
63 responses



8

According to you , should company/factory give work below 18 age person?

63 responses



v. Observations

- According to the people this laborlist and wages software is very important to create.
- According to the people labors want to know **Basic Information of company/factory**, **amount of wages to be given to labors by company/factory**, **labor request to work in company/factory and allocation of work which suited for labor**, **Requests for improve wages by labors**, **Request for leaves by supervisors or labors**, **Any deduction in wages against indisciplinary action by labors**, **working status of the activity and revenue generated from particular activity**
- According to people company/factory will be interested to use this software to store laborslist and their wages.
- Labors will be more interested to know the details about private companies/factories rather than the government.
- Most labors would prefer hourly wages rather than weekly wages.
- According to people, laborers would like both payment methods (cash/online) equally.
- Company/factory should provide life insurance to labors.
- Comapny/factory should not give work to below 18 age person.

Combined requirements:

- company/factory Keep record of wages according to the department.
- Below 18 shouldn't be allowed to work in factory.
- Company/factory should take care about the security of labors at working place.
- Company/factory should consider the request of labors to improve their wages if relatable.

3. Fact Finding Chart :

Objective	Technique	subject(s)	Time
To get idea about the software of labor management	Background Reading	Company Requirements and Reports	0.5 day
To identify problems regarding working activities of labors	Interview	Supervisor at cement factory	45 min
To find Requirements regarding working activities of labors	interview	Supervisor at cement factory	45 min
to identify problems wages of labor	interview	Supervisor	45 min
to identify problems and requirements regarding safety of the Labors	interview	Manager	45 min
To find out the overall opinion of the working activities of labor and their wages.	interview	Labor at Turner Construction	45 min
To find out need of labor during working activities	interview	Labor at Turner Construction	45 min
To get idea of additional functionality of software	Questionnaire	People	1 day

4. Requirements

- Software should contain
    - ➔ Basic Information of company/factory,
    - ➔ amount of wages to be given to labors by company/factory
    - ➔ labor request to work in company/factory and allocation of work which suited for labor, Requests for improve wages by labors
    - ➔ Request for leaves by supervisors or labors
    - ➔ Any deduction in wages against bad disciplinary actions by labors
    - ➔ working status of the activity and revenue generated from particular activity
- company/factory will be interested to use this software to store laborslist and their wages.
- Labors will be more interested to know the details about private companies/factories rather than the government.
- Most labors would prefer hourly wages rather than weekly wages.
- labor would like both payment methods (cash/online) equally therefore software should contain secure payment methods.
- Labor can apply for life insurance by software.
- Companies/factories should not give work to people below 18.
- If labors feel any inconvenience in machines then they can complain by software.
- If labors want leave except holidays they can apply or request for leave also.

5. User Categories and Privileges

- **Company/factory Owner:**
  Company Admin can add details about company and also can modify details of company.
- **Company/factory Manager :**
  A manager is at a higher level in an organization than a supervisor. Managers can modify the salary of supervisor and labor as well and also approve leave requests of supervisor and labor.

- **Company/factory Supervisor :**
  A supervisor is at a higher level in an organization than a labors. Supervisors can modify the salary of labor as well and also approve leave requests of labor.
- **Company/factory Labor :**
  Labor is at a lower level in an organization.labor can see details about the company/factory and apply for that company/factory to work.labor can also request to increase wages and also request for leaves.

6. Assumptions :

- All labors and supervisor who want to work in company/factory have a smartphone to apply for work.
- All labors and supervisors can able to understand software language.
- All labors and supervisor have a bank account in any of the banks.
- Company/factory is following all the rule of labor act which has been given by the government.
- Labor and Supervisor have online net banking account if the company/factory follows an online payment system.

7. Business Constraints :

- Some labors and supervisors who want to work in a company/factory do not have a smartphone to apply for work.
- Some labors and supervisors can not able to understand software language.
- Some labors and supervisors do not have a bank account in any of the banks.
- Some companies/factories are not following all the rule of labor act which has been given by the government.
- Some Labors and Supervisors do not have online net banking accounts if the company/factory follows an online payment system.

Description:

- A Database management system is required to maintain and update all the information about labors, wages and the company details.
- All the basic facilities that are expected to be available like keeping a record of labors and their contact details, etc.
- Company/factory owner and the Supervisor will be assigned different roles.
- Company/factory also provide surety of safety of labors during working hours.
- Labors will be assigned different types of view permissions to show details about their wages,wages type and company details in which they are working
- Software should contain
    - ➔ Basic Information of company/factory,
    - ➔ amount of wages to be given to labors by company/factory
    - ➔ labor request to work in company/factory and allocation of work which suited for labor, Requests for improve wages by labors
    - ➔ Request for leaves by supervisors or labors=
    - ➔ Any deduction in wages against bad disciplinary actions by labors
    - ➔ working status of the activity and revenue generated from particular activity
- company/factory will be interested to use this software to store laborslist and their wages.
- Labors will be more interested to know the details about private companies/factories rather than the government.
- Most labors would prefer hourly wages rather than weekly wages.
- labor would like both payment methods (cash/online) equally therefore software should contain secure payment methods.
- Labor can apply for life insurance by software.
- Companies/factories should not give work to people below 18.
- If labors feel any inconvenience in machines then they can complain by software.
- If labors want leave except holidays they can apply or request for leave also.

The system contains the following tables:

1. **Regions**
2. **Company_deatils**
3. **Labors**
4. **Work_history**
5. **Department**
6. **Locations**
7. **Leave**
8. **Wages**
9. **Supervisor**
10. **Manager**
11. **Owner**

The user can get the following information about the project from the given system:

- **Regions:** region_id,region_name
  ➔ In this table region_id is primary key
  This shows the basic details about the region of the company.
- **Company_details:**company_id,company_type,street_add,highest_no_worker,region_id
  ➔ In this table company_id is primary key.
  Here all the labors and supervisors can view the company details if they want to work in a particular company.

- **Labors:**labor_id,first_name,last_name,gender,phone_no,dept_id,company_id,supervisor_id,age,wage,hire_date,region_id,address,labor_type
  ➔ In this table labor_id is primary key; dept_id and company_id are foreign keys.
  Here all labors and supervisors of the company/factory can view basic details/information about labor.

- **Work_history:** labor_id,start_date,end_date,dept_id,company_id
  ➔ In this table labor_id is primary key; company_id and dept_id are foreign keys.

Company/factory can view the work history of all labors like start date and end date.

- **Department:** dept_id,name,wages_range,company_id
  - ➔ In this table dept_id is primary key and company_id is foreign key.

  Here supervisor or owner can modify the wages range according to requests of wages by labors.

- **Locations:** street_add,pin_code,city,state,country,region_id
  - ➔ in this table pin_code is primary key and region_id is foreign key

  Here labors can select and view their work locations.

- **Leave:** leave_id,labor_id,date,reason
  - ➔ in this table leave_idis primary key and labor_id is foreign key

  Here labors can apply or request for leave due to any emergency or illness only.

- **Wages:**wages_id,labor_id,dept_id,payment_id,date,receipt,total_amount,wages_type
  - ➔ in this table wages_id is primary key and labor_id is foreign key

  Here supervisors can see wages details of their labors which have already been paid and Labors can view receipt of their wages.

- **supervisor:**supervisor_id,f_name,l_name,gender,phone_no,dept_id,company_id,salary,hire_date,region_id,manager_id,address
  - ➔ in this table supervisor_id is primary key; dept_id and company_id are foreign keys.

  Here all supervisors of the company/factory can view basic details/information about labor.

- **Manager:**manager_id,f_name,l_name,gender,phone_no,dept_id,company_id,salary,hire_date,region_id,address

➔ in this table manager_id is primary key; dept_id and company_id are foreign keys.

Here all Manager of the company/factory can view basic details/information about labor and supervisor.

- **Owner:**owner_id,f_name,l_name,gender,phone_no,company_id,address
  ➔ in this table owner_id is primary key;company_id is foreign keys.

  Owner of the company/factory can view basic details/information about manager, labor and supervisor.

Section 2 : Noun & Verb Analysis

1. Noun (Verb) Analysis:

Sr. no.	Noun	Verb
1	Management system	be
2	city	
3	district	
4	Labor information	record
5	labor	work
6	supervisor	Monitor
7	company	working
8	owner	
9	manager	
10	record	keep
11	factory	
12	contact details	keep
13	wage	give
14	leave	request
15	revenue	generate
16	software	store
17	cash	pay
18	Online banking	pay
19	Wages type	prefer

20	Company type	interest
21	Life insurance	provide
22	machine	complain
23	holiday	request
24	Region	
25	age	
26	gender	
27	Details	Modify
28	state	
29	Country	
30	Street	
31	Address	add/remove
32	Salary	Credit
33	Location	
34	date	
35	Hire date	record
36	precautionary thing	
37	accident	
38	Minimum wage	need
39	Pay slip	provide
40	organization	Goal

2. Accepted Noun or verb list

Sr No	Candidate entity set	Candidate attribute set	Candidate relationship set
1	labor	Labor id,first name,last name,gender,phone no,dept id,company id,supervisor id,age,wage,hire date,region id,address,labor type	work_in
2	supervisor	Supervisor id,first name,last name,gender,phone no,dept id,company id,salary,hire date,region id,manager id,address	monitor,manage
3	manager	Manager id,first name,last name,gender,phone no,dept id,company id,salary,hire date,region id,address	manage
4	owner	Owner id,first name,last name,gender,phone no,company id,address	manage
5	Company details	Company id,company type,street add,highest no. worker,region id	-
6	region	Region id,region name	

7	Work history	Labor id,start date,end date,dept id,company id	
8	location	Street add,pin code,city,state,country,loaction_ id	
9	Department	Dept id,name,wages range,company id	
10	leave	Leave id,labor id, date,reason	has
11	wages	wages id,labor id, dept id,payment id,date,receipt,total amount,wages type	gets

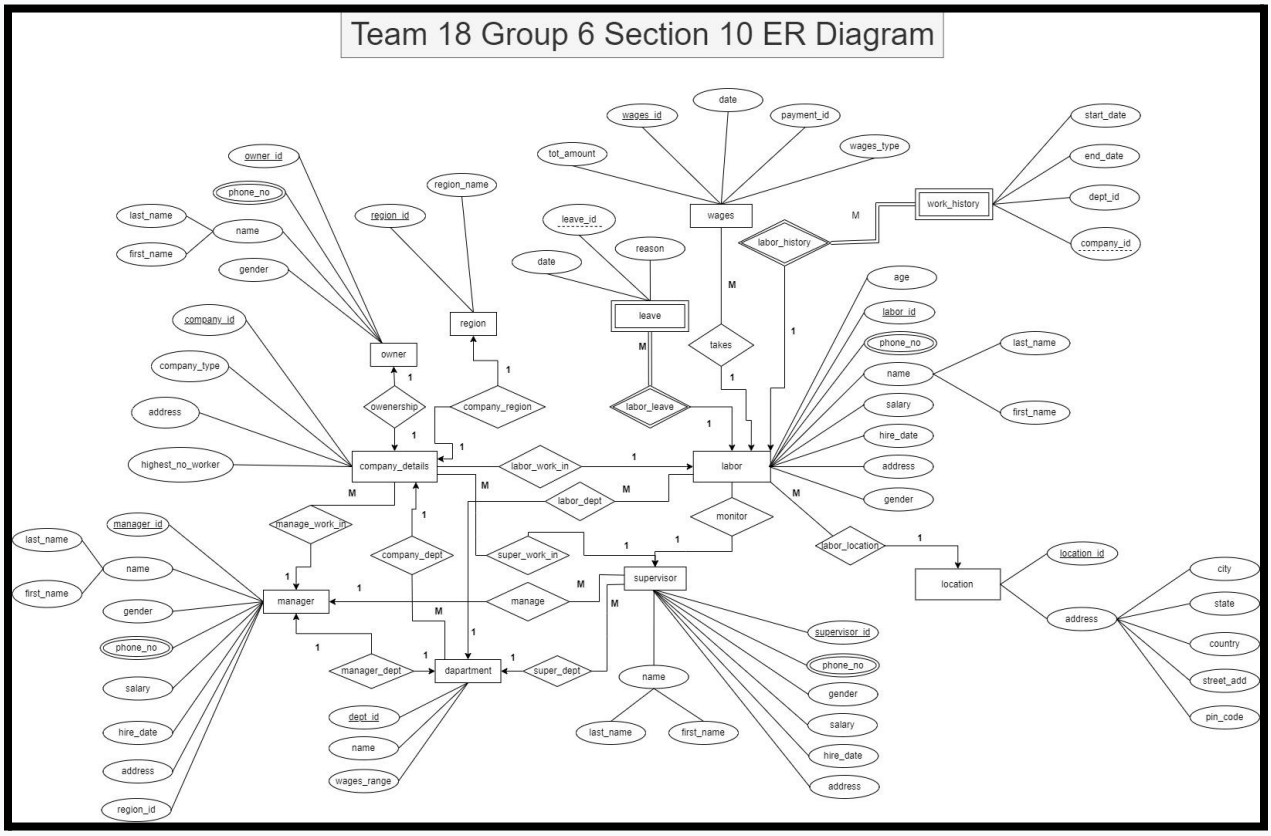
3. Rejected Nouns or verbs:

Sr no.	Noun	Reject reason
1	Management system	General
2	date	vague
3	precautionary thing	Irrelevant
4	Accident	Irrelevant
5	revenue	Irrelevant
6	Monitor	Irrelevant
7	Be	General
8	Keep	General

9	add/remove	General
10	software	General
11	holiday	vague
12	Minimum wage	vague
13	Organization	General
14	Goal	Irrelevant
15	Provide	Irrelevant
16	Need	Irrelevant
17	complain	Irrelevant
18	Life insurance	vague

Section 3 : ER Diagrams

ER Diagram: (Final Version)



Section 4 : Mapping E-R model to Relational Model

2. Mapping E-R model to Relational Model:

- **Regions(**region_id,region_name)
  - ➔ In this table region_id is primary key

- **Company_details(**company_id,company_type,street_add,highest_no_worker,region_id)
  - ➔ In this table company_id is primary key.

- **Labors(**labor_id,first_name,last_name,gender,dept_id,company_id,supervisor_id,age,wage,hire_date,location_id,address,labor_type)
  - ➔ In this table labor_id is primary key;
    dept_id,supervisor_id,location_id and company_id are foreign keys.

- **Work_history(**labor_id,dept_id,company_id,start_date,end_date)
  - ➔ In this table labor_id,company_id and dept_id are composite primary key;
    labor_id is foreign key.

- **Department(**dept_id,name,wages_range,company_id)
  - ➔ In this table dept_id is primary key;
    company_id is foreign key.

- **Locations(**location_id,street_add,city,state,country)
  - ➔ in this table pin_code is primary key

- **Leave(**leave_id,labor_id,date,reason)
  - ➔ in this table leave_id is primary key;
    labor_id is foreign key

- **Wages(**wages_id,payment_no,labor_id,date,total_amount,wages_type)
  - ➔ in this table wages_id and payment_no are composite primary key;

labor_id is foreign key

- **supervisor(**supervisor_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,manager_id,address)
  - ➔ in this table supervisor_id is primary key;
    dept_id,location_id,manager_id and company_id are foreign keys.

- **Manager(**manager_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,address)
  - ➔ in this table manager_id is primary key;
    dept_id, location_id and company_id are foreign keys.

- **Owner(**owner_id,first_name,last_name,gender,company_id,address)
  - ➔ in this table owner_id is primary key;
    company_id is foreign key.

- **Labor_phone(**labor_id, phone_no)
  - ➔ in this table labor_id and phone_no is composite primary key;
    labor_id is foreign key

- **Supervisor_phone(**supervisor_id, phone_no)
  - ➔ in this table supervisor_id and phone_no is composite primary key;
    supervisor_id is foreign key

- **Owner_phone(**owner_id, phone_no)
  - ➔ in this table owner_id and phone_no is composite primary key;
    owner_id is foreign key

- **Manager_phone(**manager_id, phone_no)
  - ➔ in this table manager_id and phone_no is composite primary key;
    Manager is foreign key

3. Create DDL Scripts

Region

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".region
(
    region_id integer NOT NULL,
    region_name character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT region_pkey PRIMARY KEY (region_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".region
    OWNER to postgres;
```

Company_details

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".company_details
(
    company_id integer NOT NULL,
    company_type character varying COLLATE pg_catalog."default" NOT NULL,
    street_add character varying COLLATE pg_catalog."default",
    highest_no_worker integer,
    region_id integer NOT NULL,
    CONSTRAINT company_details_pkey PRIMARY KEY (company_id),
    CONSTRAINT region_id FOREIGN KEY (region_id)
        REFERENCES "labor_wages_T18".region (region_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".company_details
    OWNER to postgres;
```

Department

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".department
(
    dept_id integer NOT NULL,
    name character varying COLLATE pg_catalog."default" NOT NULL,
    wages_range character varying COLLATE pg_catalog."default",
    company_id integer NOT NULL,
    CONSTRAINT department_pkey PRIMARY KEY (dept_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".department
    OWNER to postgres;
```

Location

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".location
(
    location_id integer NOT NULL,
    street_add character varying COLLATE pg_catalog."default" NOT NULL,
    city character varying COLLATE pg_catalog."default" NOT NULL,
    state character varying COLLATE pg_catalog."default" NOT NULL,
    contry character varying COLLATE pg_catalog."default",
    CONSTRAINT location_pkey PRIMARY KEY (location_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".location
```

```
    OWNER to postgres;
```

Owner

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".owner
(
    owner_id integer NOT NULL,
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    address character varying COLLATE pg_catalog."default",
    company_id integer NOT NULL,
    CONSTRAINT owner_pkey PRIMARY KEY (owner_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".owner
    OWNER to postgres;
```

Manager

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".manager
(
    manager_id integer NOT NULL,
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    dept_id integer NOT NULL,
    company_id integer NOT NULL,
    location_id integer NOT NULL,
    salary bigint NOT NULL,
```

```
    hire_date date,
    address character varying COLLATE pg_catalog."default",
    CONSTRAINT manager_pkey PRIMARY KEY (manager_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT dept_id FOREIGN KEY (dept_id)
        REFERENCES "labor_wages_T18".department (dept_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT location_id FOREIGN KEY (location_id)
        REFERENCES "labor_wages_T18".location (location_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".manager
    OWNER to postgres;
```

Supervisor

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".supervisor
(
    supervisor_id integer NOT NULL,
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    dept_id integer NOT NULL,
    company_id integer NOT NULL,
    location_id integer NOT NULL,
    salary bigint NOT NULL,
    hire_date date,
    manager_id integer NOT NULL,
    address character varying COLLATE pg_catalog."default",
```

```
    CONSTRAINT supervisor_pkey PRIMARY KEY (supervisor_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT dept_id FOREIGN KEY (dept_id)
        REFERENCES "labor_wages_T18".department (dept_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT location_id FOREIGN KEY (location_id)
        REFERENCES "labor_wages_T18".location (location_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT manager_id FOREIGN KEY (manager_id)
        REFERENCES "labor_wages_T18".manager (manager_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".supervisor
    OWNER to postgres;
```

Labor

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".labors
(
    labor_id integer NOT NULL,
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    dept_id integer NOT NULL,
    company_id integer NOT NULL,
    supervisor_id integer NOT NULL,
    age integer NOT NULL,
    wage bigint NOT NULL,
    hire_date date NOT NULL,
```

```
    location_id integer NOT NULL,
    address character varying COLLATE pg_catalog."default",
    labor_type character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT labors_pkey PRIMARY KEY (labor_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT dept_id FOREIGN KEY (dept_id)
        REFERENCES "labor_wages_T18".department (dept_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT location_id FOREIGN KEY (location_id)
        REFERENCES "labor_wages_T18".location (location_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT supervisor_id FOREIGN KEY (supervisor_id)
        REFERENCES "labor_wages_T18".supervisor (supervisor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".labors
    OWNER to postgres;
```

Leave

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".leave
(
    leave_id integer NOT NULL,
    labor_id integer NOT NULL,
    date date NOT NULL,
    reason character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT leave_pkey PRIMARY KEY (leave_id),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".leave
    OWNER to postgres;
```

Wages

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".wages
(
    wages_id integer NOT NULL,
    labor_id integer NOT NULL,
    payment_no integer NOT NULL,
    date date NOT NULL,
    total_amount bigint NOT NULL,
    wages_type character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT wages_pkey PRIMARY KEY (wages_id, payment_no),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".wages
    OWNER to postgres;
```

Work_history

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".work_history
(
    labor_id integer NOT NULL,
    dept_id integer NOT NULL,
```

```
    company_id integer NOT NULL,
    start_date date NOT NULL,
    end_date date NOT NULL,
    CONSTRAINT work_history_pkey PRIMARY KEY (labor_id, dept_id, company_id),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".work_history
    OWNER to postgres;
```

Labor_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".labor_phone
(
    labor_id integer NOT NULL,
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT labor_phone_pkey PRIMARY KEY (labor_id, phone_no),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".labor_phone
    OWNER to postgres;
```

Supervisor_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".supervisor_phone
(
    supervisor_id integer NOT NULL,
```

```
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT supervisor_phone_pkey PRIMARY KEY (supervisor_id, phone_no),
    CONSTRAINT supervisor_id FOREIGN KEY (supervisor_id)
        REFERENCES "labor_wages_T18".supervisor (supervisor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".supervisor_phone
    OWNER to postgres;
```

Owner_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".owner_phone
(
    owner_id integer NOT NULL,
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT owner_phone_pkey PRIMARY KEY (owner_id, phone_no),
    CONSTRAINT owner_id FOREIGN KEY (owner_id)
        REFERENCES "labor_wages_T18".owner (owner_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".owner_phone
    OWNER to postgres;
```

Manager_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".manager_id
(
    manager_id integer NOT NULL,
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT manager_id_pkey PRIMARY KEY (manager_id, phone_no),
```

```
    CONSTRAINT manager_id FOREIGN KEY (manager_id)
        REFERENCES "labor_wages_T18".manager (manager_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".manager_id
    OWNER to postgres;
```

Section 5 : Normalization & Schema Refinement

❖ Normalization and Schema refinement:

1. Functional Dependencies:

- **Regions(region_id** → region_name)
- **Company_details(company_id**→company_type,company_name,street_add,highest_no_worker,region_id)
- **Labors(labor_id**→first_name,last_name,dept_id,company_id,supervisor_id,wage,hire_date,location_id,labor_type,gender,age,address)
- **supervisor(supervisor_id**→first_name,last_name,dept_id,company_id,location_id,salary,hire_date,manager_id, gender,address)
- **Manager(manager_id**→first_name,last_name,dept_id,company_id,location_id,salary,hire_date, gender,address)
- **Owner(owner_id**→first_name,last_name,company_id,
  first_name,last_name → gender,address)
- **Department(dept_id** → name,wages_range,company_id)
- **Locations(location_id** → street_add,city,state,country)
- **Leave(leave_id** → labor_id,date,reason)
- **Wages(wages_id**,**payment_no**→labor_id,wages_type,date,total_amount)
- **Work_history(labor_id**,**dept_id,company_id**→start_date,end_date**)**
- **Labor_phone:** No dependencies
- **Supervisor_phone:** No dependencies

- **Owner_phone:** No dependencies

- **Manager_phone:** No dependencies

❖ Redundancies and Anomalies:

- **Regions(**region_id,region_name)

  region_id is the primary key Which can't be NULL.
  There is no redundancy in this relation.
  There are no transitive dependencies.
  There are no anomalies in this relation

- **Company_details(**company_id,company_name,company_type,street_add ,highest_no_worker,region_id)

  company_id is the primary key Which can't be NULL.
  Foreign Key region_id references to regions.
  There is no redundancy in this relation.
  There are no transitive dependencies.
  There are no anomalies in this relation

- **Labors(**labor_id,first_name,last_name,gender,dept_id,company_id,supervi sor_id,age,wage,hire_date,location_id,address,labor_type)

  labor_id is the primary key Which can't be NULL.
  Foreign Key dept_id references to Department.
  Foreign Key company_id references to Company_details.
  Foreign Key supervisor_id references to Supervisor.
  Foreign Key location_id references to Location.

  Redundancy: we can drop location_id , dept_id and company_id because we have references to supervisor which references to manager which already contain location_id,dept_id and company_id.

- **supervisor(**supervisor_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,manager_id,address)

  supervisor_id is the primary key Which can't be NULL.
  Foreign Key dept_id references to Department.
  Foreign Key company_id references to Company_details.
  Foreign Key manager_id references to Manager.
  Foreign Key location_id references to Location.

  Redundancy: we can drop location_id , dept_id and company_id because we have references to manager which already contain location_id,dept_id and company_id.

- **Manager(**manager_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,address)

  manager_id is the primary key Which can't be NULL.
  Foreign Key dept_id references to Department.
  Foreign Key company_id references to Company_details.
  Foreign Key location_id references to Location.

  Redundancy: There could be multiple location_id associated with one labor if he already worked at some place earlier and not work at other branch.

- **Owner(**owner_id,first_name,last_name,gender,company_id,address)

  There are no anomalies in this entity.
  owner_id is the primary key Which can't be NULL.
  Foreign Key company_id references to Company_details.
  Because of the above dependency, we also get a transitive dependency.
  So we can remove this dependency by adding a new table of

- **Work_history(**labor_id,dept_id,company_id,start_date,end_date)

  labor_id,company_id and dept_id are composite primary keys, Which can't be NULL.

Foreign key labor_id references Labors
There is no transitive dependency.

- **Department(**dept_id,name,wages_range,company_id)

  dept_id is the primary key Which can't be NULL.
  Foreign Key company_id references to Company_details.
  There is no redundancy in this relation.
  There are no transitive dependencies.
  There are no anomalies in this relation.

- **Locations(**location_id,street_add,city,state,country)

  Location_id is the primary key Which can't be NULL.
  There is no redundancy in this relation.
  There are no transitive dependencies.
  There are no anomalies in this relation.

- **Leave(**leave_id,labor_id,date,reason)

  leave_id is the primary key Which can't be NULL.
  Foreign Key labor_id references to labors.
  There is no redundancy in this relation.
  There are no transitive dependencies.

  ➔ Anomalies:
  Insert: We can not add a reason until it is not associated with leave id.

  Delete: If we delete the tuple of a particular reason, then it will also delete information about a specific leave of labor.

  Update: We have to update maintenance priority for the same reason every time.

- **Wages(**wages_id,payment_no,labor_id,date,total_amount,wages_type)

Wages_id and paymment_no are the primary keys which can't be NULL.
Foreign Key labor_id references to labors.
There is no redundancy in this relation.
There are no transitive dependencies.

- **Labor_phone(**labor_id, phone_no)
  in this table labor_id and phone_no is composite primary key;
  labor_id is foreign key which can't be NULL
  There is no redundancy in this relation.
  There are no transitive dependencies.

- **Supervisor_phone(**supervisor_id, phone_no)
  in this table supervisor_id and phone_no is composite primary key,supervisor_id is foreign key which can't be NULL
  There is no redundancy in this relation.
  There are no transitive dependencies.

- **Owner_phone(**owner_id, phone_no)
  in this table owner_id and phone_no is composite primary key;
  owner_id is a foreign key which can't be NULL
  There is no redundancy in this relation.
  There are no transitive dependencies.

- **Manager_phone(**manager_id, phone_no)
  manager_id and phone_no is composite primary key; Manager is a foreign key which can't be NULL.
  There is no redundancy in this relation.
  There are no transitive dependencies.

❖ Normalization upto 3NF/BCNF

- **Regions(**region_id,region_name)
  Since every attribute value is atomic the relation is in 1NF.

Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Company_details(**company_id,company_name,company_type,street_add ,highest_no_worker,region_id)

  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  Every candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Labors(**labor_id,first_name,last_name,gender,dept_id,company_id,supervi sor_id,age,wage,hire_date,location_id,address,labor_type)

  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  Every candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Work_history(**labor_id,dept_id,company_id,start_date,end_date)
  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  Every candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Department(**dept_id,name,wages_range,company_id)
  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.

candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Locations(**location_id,street_add,city,state,country)
  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Leave(**leave_id,labor_id,date,reason)
  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Wages(**wages_id,payment_no,labor_id,date,total_amount,wages_type)
  Since every attribute value is atomic the relation is in 1NF.
  Since there are two values in PK and also there is only one candidate key the relation is in 2NF.
  candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **supervisor(**supervisor_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,manager_id,address)

  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  Every candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Manager(**manager_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,address)

Since every attribute value is atomic the relation is in 1NF.
Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
Every candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Owner(**owner_id,first_name,last_name,gender,company_id,address)
  Since every attribute value is atomic the relation is in 1NF.

  Since every attribute value is atomic the relation is in 1NF.
  Since there is only one value in PK and also there is only one candidate key the relation is in 2NF.
  Every candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Labor_phone(**labor_id, phone_no)
  Since every attribute value is atomic the relation is in 1NF.
  Since there are two values in PK and also there is only one candidate key the relation is in 2NF.
  candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Supervisor_phone(**supervisor_id, phone_no)
  Since every attribute value is atomic the relation is in 1NF.
  Since there are two values in PK and also there is only one candidate key the relation is in 2NF.
  candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Owner_phone(**owner_id, phone_no)
  Since every attribute value is atomic the relation is in 1NF.
  Since there are two values in PK and also there is only one candidate key the relation is in 2NF.

candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

- **Manager_phone(**manager_id, phone_no)
  Since every attribute value is atomic the relation is in 1NF.
  Since there are two values in PK and also there is only one candidate key the relation is in 2NF.
  candidate key is on LHS of the functional dependencies so the relation is in BCNF.this implies that the relation is in 3NF.

❖ List of Final relations with schema:

- **Regions(**region_id,region_name)

➔ In this table region_id is primary key

- **Company_details(**company_id,company_type,street_add,highest_no_worker,region_id,company_name)

➔ In this table company_id is primary key.

- **Labors(**labor_id,first_name,last_name,gender,supervisor_id,age,wage,hire_date,address,labor_type)

➔ In this table labor_id is primary key; dept_id,supervisor_id ,location_id and company_id are foreign keys.

- **Work_history(**labor_id,dept_id,company_id,start_date,end_date)

➔ In this table labor_id,company_id and dept_id are composite primary key; labor_id is foreign key.

- **Department(**dept_id,name,wages_range,company_id)

➔ In this table dept_id is primary key;
company_id is foreign key.

- **Locations(**location_id,street_add,city,state,country)

➔ in this table pin_code is primary key

- **Leave(**leave_id,labor_id,date,reason)

➔ in this table leave_id is primary key;
labor_id is foreign key

- **Wages(**wages_id,payment_no,labor_id,date,total_amount,wages_type)

➔ in this table wages_id and payment_no are composite primary key;
labor_id is foreign key

- **supervisor(**supervisor_id,first_name,last_name,gender,salary,hire_date,manager_id,address)

➔ in this table supervisor_id is primary key; dept_id,location_id,
manager_id and company_id are foreign keys.

- **Manager(**manager_id,first_name,last_name,gender,dept_id,company_id,location_id,salary,hire_date,address)

➔ in this table manager_id is primary key;
dept_id, location_id and company_id are foreign keys.

- **Owner(**owner_id,first_name,last_name,gender,company_id,address)

➔ in this table owner_id is primary key;
company_id is foreign key.

- **Labor_phone(**labor_id, phone_no)

➔ in this table labor_id and phone_no is composite primary key;
labor_id is foreign key

- **Supervisor_phone(**supervisor_id, phone_no)

➔ in this table supervisor_id and phone_no is composite primary key;
supervisor_id is foreign key

- **Owner_phone(**owner_id, phone_no)

➔ in this table owner_id and phone_no is composite primary key;
owner_id is foreign key

- **Manager_phone(**manager_id, phone_no)

➔ in this table manager_id and phone_no is composite primary key; Manager is foreign key

Section 6 : DDL Scripts

❖ DDL Scripts

Region

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".region
(
    region_id integer NOT NULL,
    region_name character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT region_pkey PRIMARY KEY (region_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".region
    OWNER to postgres;
```

Company_details

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".company_details
(
    company_id integer NOT NULL,
    company_type character varying COLLATE pg_catalog."default" NOT NULL,
    street_add character varying COLLATE pg_catalog."default",
    highest_no_worker integer,
    region_id integer NOT NULL,
    company_name character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT company_details_pkey PRIMARY KEY (company_id),
    CONSTRAINT region_id FOREIGN KEY (region_id)
        REFERENCES "labor_wages_T18".region (region_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
```

```
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".company_details
    OWNER to postgres;
```

Department

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".department
(
    dept_id integer NOT NULL,
    name character varying COLLATE pg_catalog."default" NOT NULL,
    wages_range character varying COLLATE pg_catalog."default",
    company_id integer NOT NULL,
    CONSTRAINT department_pkey PRIMARY KEY (dept_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".department
    OWNER to postgres;
```

Location

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".location
(
    location_id integer NOT NULL,
    street_add character varying COLLATE pg_catalog."default" NOT NULL,
    city character varying COLLATE pg_catalog."default" NOT NULL,
    state character varying COLLATE pg_catalog."default" NOT NULL,
    contry character varying COLLATE pg_catalog."default",
```

```
    CONSTRAINT location_pkey PRIMARY KEY (location_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".location
    OWNER to postgres;
```

Owner

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".owner
(
    owner_id integer NOT NULL,
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    address character varying COLLATE pg_catalog."default",
    company_id integer NOT NULL,
    CONSTRAINT owner_pkey PRIMARY KEY (owner_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".owner
    OWNER to postgres;
```

Manager

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".manager
(
    manager_id integer NOT NULL,
```

```
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    dept_id integer NOT NULL,
    company_id integer NOT NULL,
    location_id integer NOT NULL,
    salary bigint NOT NULL,
    hire_date date,
    address character varying COLLATE pg_catalog."default",
    CONSTRAINT manager_pkey PRIMARY KEY (manager_id),
    CONSTRAINT company_id FOREIGN KEY (company_id)
        REFERENCES "labor_wages_T18".company_details (company_id) MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT dept_id FOREIGN KEY (dept_id)
        REFERENCES "labor_wages_T18".department (dept_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT location_id FOREIGN KEY (location_id)
        REFERENCES "labor_wages_T18".location (location_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".manager
    OWNER to postgres;
```

Supervisor

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".supervisor
(
    supervisor_id integer NOT NULL,
```

```
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    salary bigint NOT NULL,
    hire_date date,
    manager_id integer NOT NULL,
    address character varying COLLATE pg_catalog."default",
    CONSTRAINT supervisor_pkey PRIMARY KEY (supervisor_id),
    CONSTRAINT manager_id FOREIGN KEY (manager_id)
        REFERENCES "labor_wages_T18".manager (manager_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".supervisor
    OWNER to postgres;
```

Labor

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".labors
(
    labor_id integer NOT NULL,
    first_name character varying COLLATE pg_catalog."default" NOT NULL,
    last_name character varying COLLATE pg_catalog."default" NOT NULL,
    gender character varying COLLATE pg_catalog."default" NOT NULL,
    supervisor_id integer NOT NULL,
    age integer NOT NULL,
    wage bigint NOT NULL,
    hire_date date NOT NULL,
    address character varying COLLATE pg_catalog."default",
    labor_type character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT labors_pkey PRIMARY KEY (labor_id),
    CONSTRAINT supervisor_id FOREIGN KEY (supervisor_id)
        REFERENCES "labor_wages_T18".supervisor (supervisor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

```
TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".labors
    OWNER to postgres;
```

Leave

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".leave
(
    leave_id integer NOT NULL,
    labor_id integer NOT NULL,
    date date NOT NULL,
    reason character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT leave_pkey PRIMARY KEY (leave_id),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".leave
    OWNER to postgres;
```

Wages

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".wages
(
    wages_id integer NOT NULL,
    labor_id integer NOT NULL,
    payment_no integer NOT NULL,
    date date NOT NULL,
    total_amount bigint NOT NULL,
    wages_type character varying COLLATE pg_catalog."default" NOT NULL,
```

```
    CONSTRAINT wages_pkey PRIMARY KEY (wages_id, payment_no),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".wages
    OWNER to postgres;
```

Work_history

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".work_history
(
    labor_id integer NOT NULL,
    dept_id integer NOT NULL,
    company_id integer NOT NULL,
    start_date date NOT NULL,
    end_date date NOT NULL,
    CONSTRAINT work_history_pkey PRIMARY KEY (labor_id, dept_id, company_id),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".work_history
    OWNER to postgres;
```

Labor_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".labor_phone
(
    labor_id integer NOT NULL,
```

```
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT labor_phone_pkey PRIMARY KEY (labor_id, phone_no),
    CONSTRAINT labor_id FOREIGN KEY (labor_id)
        REFERENCES "labor_wages_T18".labors (labor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".labor_phone
    OWNER to postgres;
```

Supervisor_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".supervisor_phone
(
    supervisor_id integer NOT NULL,
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT supervisor_phone_pkey PRIMARY KEY (supervisor_id, phone_no),
    CONSTRAINT supervisor_id FOREIGN KEY (supervisor_id)
        REFERENCES "labor_wages_T18".supervisor (supervisor_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".supervisor_phone
    OWNER to postgres;
```

Owner_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".owner_phone
(
    owner_id integer NOT NULL,
```

```
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT owner_phone_pkey PRIMARY KEY (owner_id, phone_no),
    CONSTRAINT owner_id FOREIGN KEY (owner_id)
        REFERENCES "labor_wages_T18".owner (owner_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".owner_phone
    OWNER to postgres;
```

Manager_phone

```
CREATE TABLE IF NOT EXISTS "labor_wages_T18".manager_phone
(
    manager_id integer NOT NULL,
    phone_no character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT manager_id_pkey PRIMARY KEY (manager_id, phone_no),
    CONSTRAINT manager_id FOREIGN KEY (manager_id)
        REFERENCES "labor_wages_T18".manager (manager_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "labor_wages_T18".manager_phone
    OWNER to postgres;
```

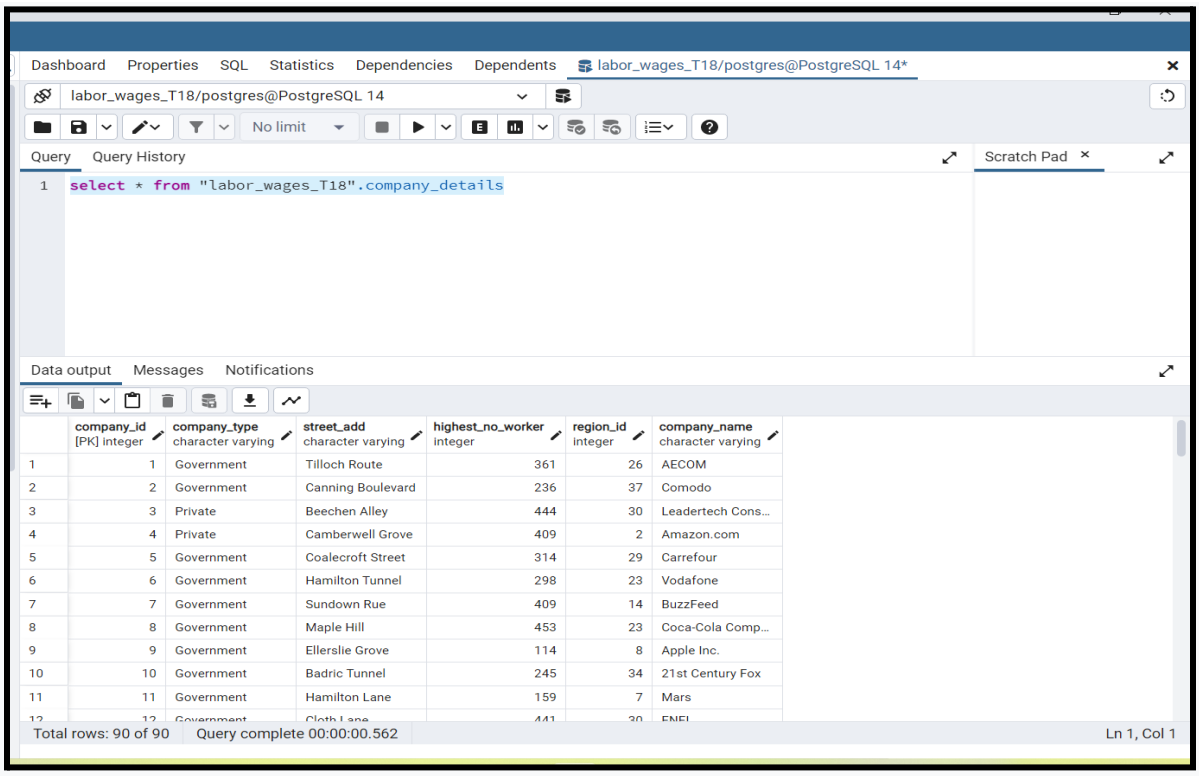
Section 7 : SQL Queries

❖ SQL Queries:

1) Show all company details.

➢

```
select *
from "labor_wages_T18".company_details
```



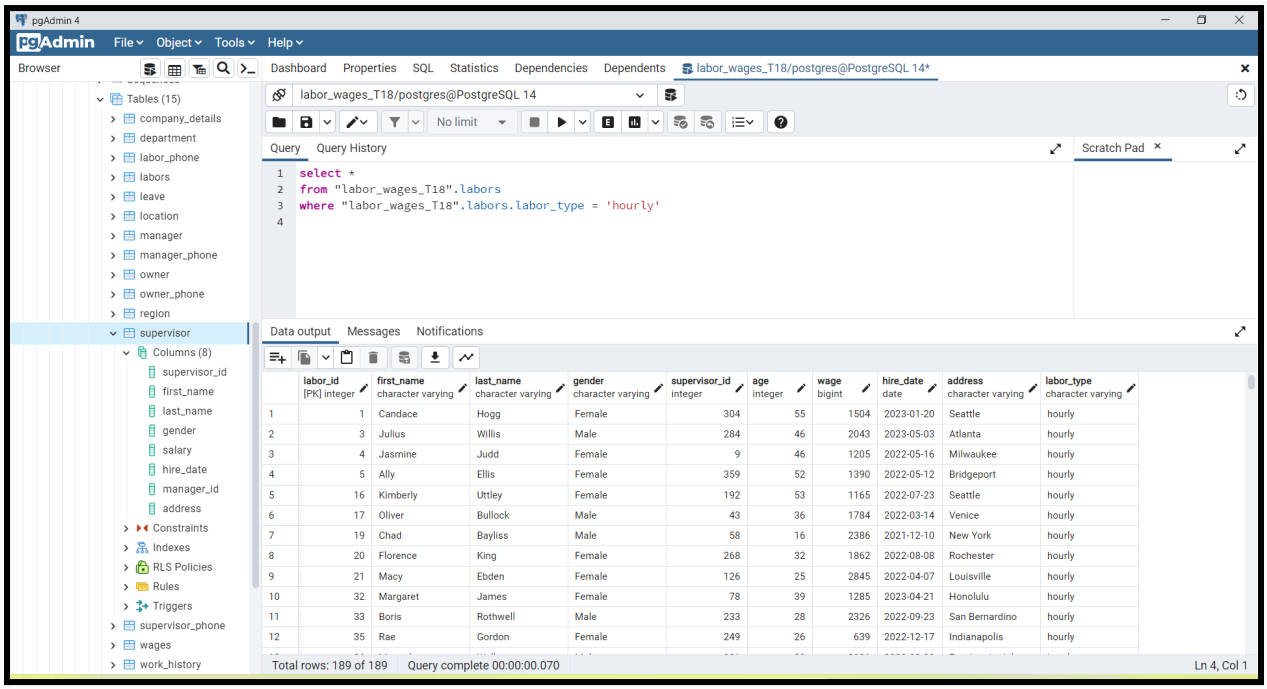
	company_id [PK] integer	company_type character varying	street_add character varying	highest_no_worker integer	region_id integer	company_name character varying
1	1	Government	Tilloch Route	361	26	AECOM
2	2	Government	Canning Boulevard	236	37	Comodo
3	3	Private	Beechen Alley	444	30	Leadertech Cons...
4	4	Private	Camberwell Grove	409	2	Amazon.com
5	5	Government	Coalecroft Street	314	29	Carrefour
6	6	Government	Hamilton Tunnel	298	23	Vodafone
7	7	Government	Sundown Rue	409	14	BuzzFeed
8	8	Government	Maple Hill	453	23	Coca-Cola Comp...
9	9	Government	Ellerslie Grove	114	8	Apple Inc.
10	10	Government	Badric Tunnel	245	34	21st Century Fox
11	11	Government	Hamilton Lane	159	7	Mars

- There is a total of 90 company data in this database.

2) Show Labor details with hourly wages type.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.labor_type = 'hourly'
```

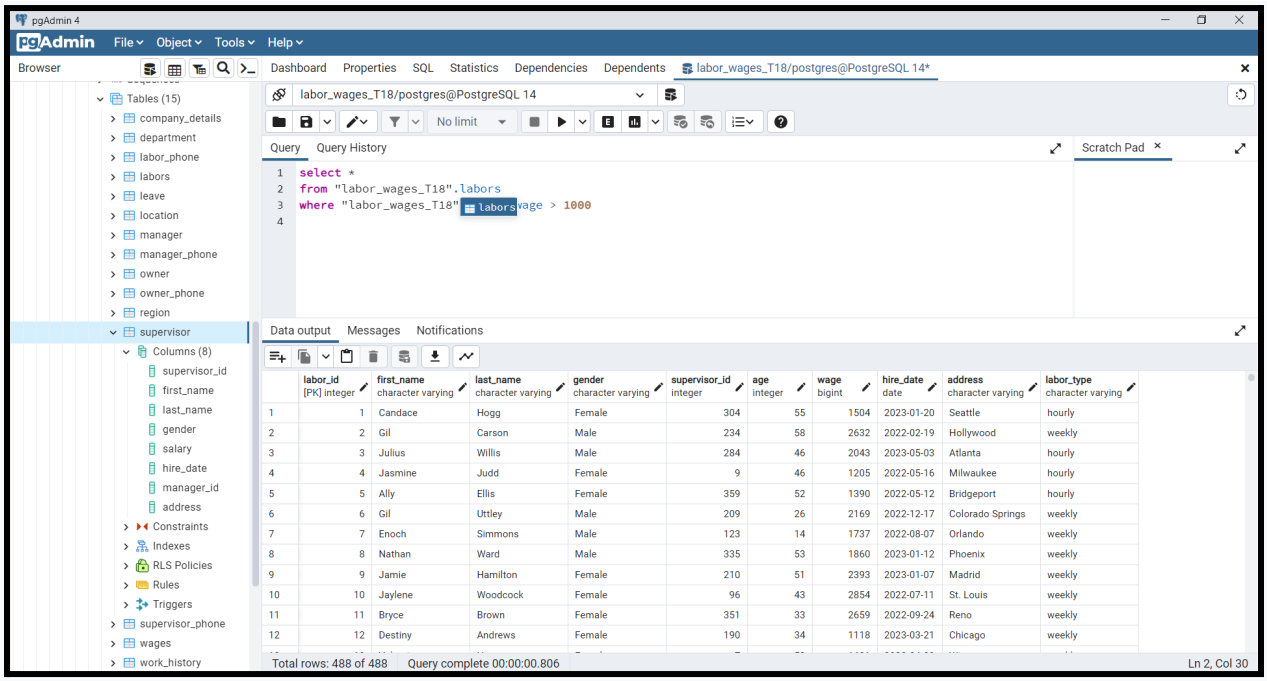


- Total Number of labors with hourly wages type is 189.

3) Show Labors with wages greater than 1000.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.wage > 1000
```

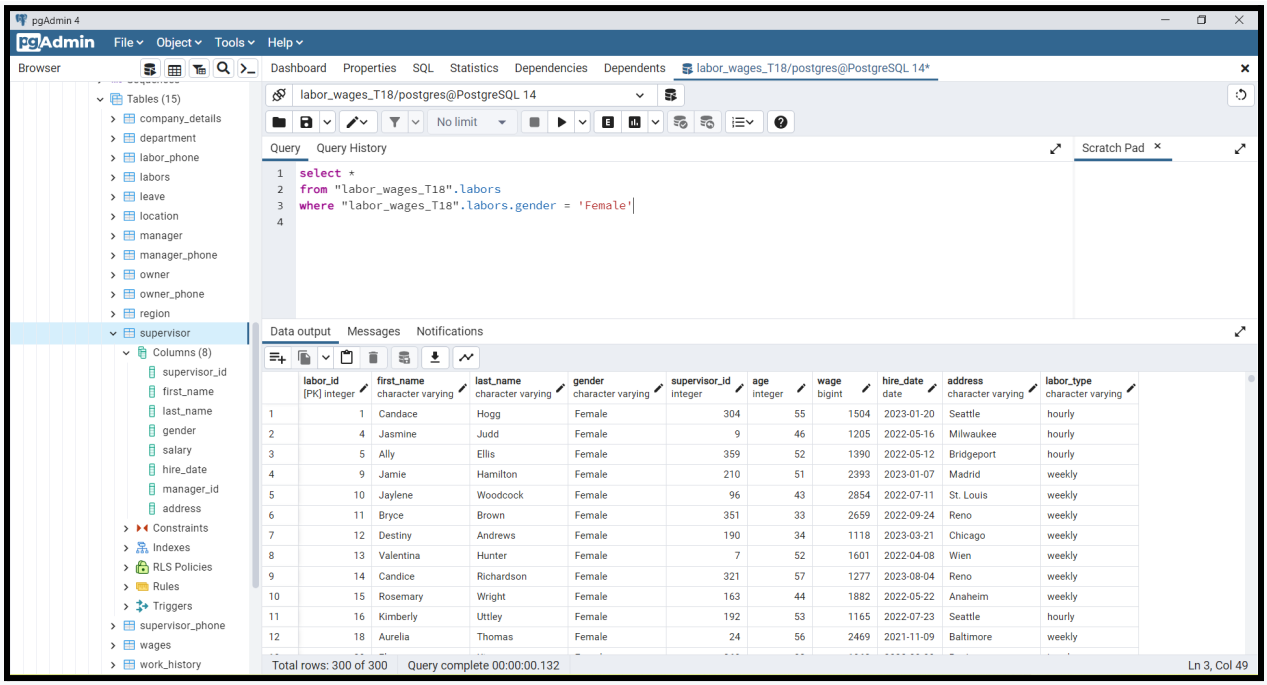


- Total No of labors with hourly wages type is 488.

4) Show the Total number of female labors.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.gender = 'Female'
```

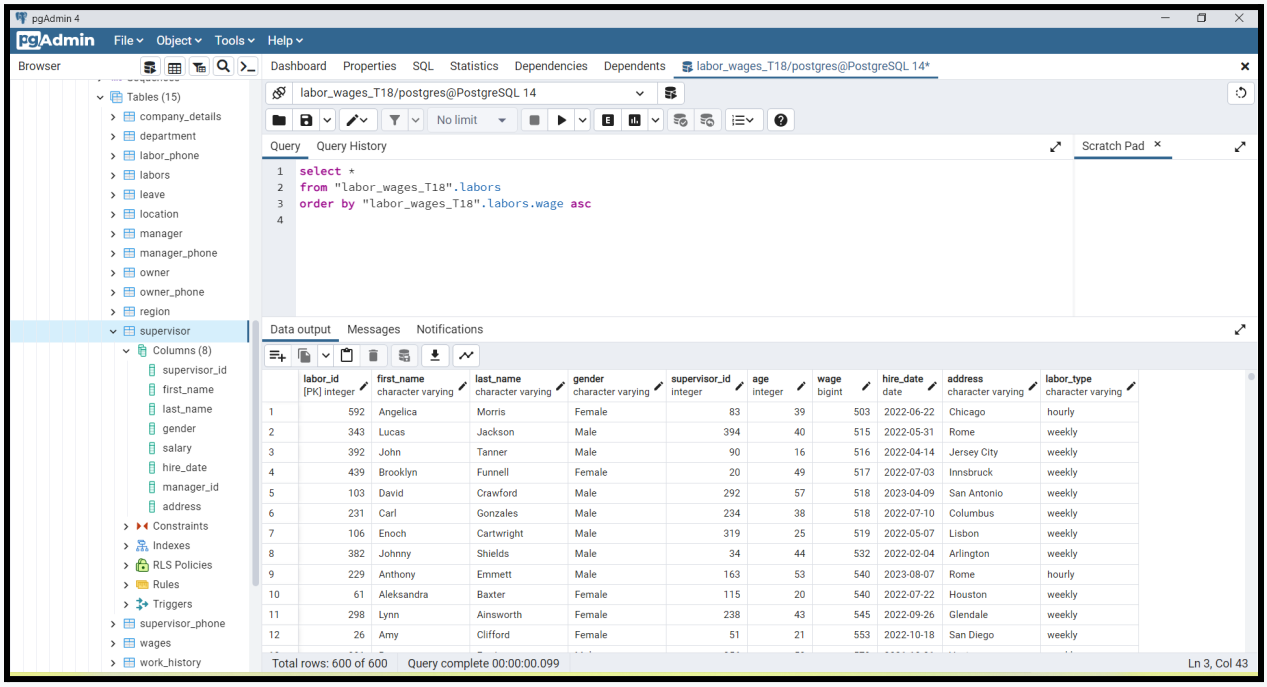


- There are a total 300 Female labors.

5) Show labors details in ascending order of their wages.

➢

```
select *
from "labor_wages_T18".labors
order by "labor_wages_T18".labors.wage asc
```

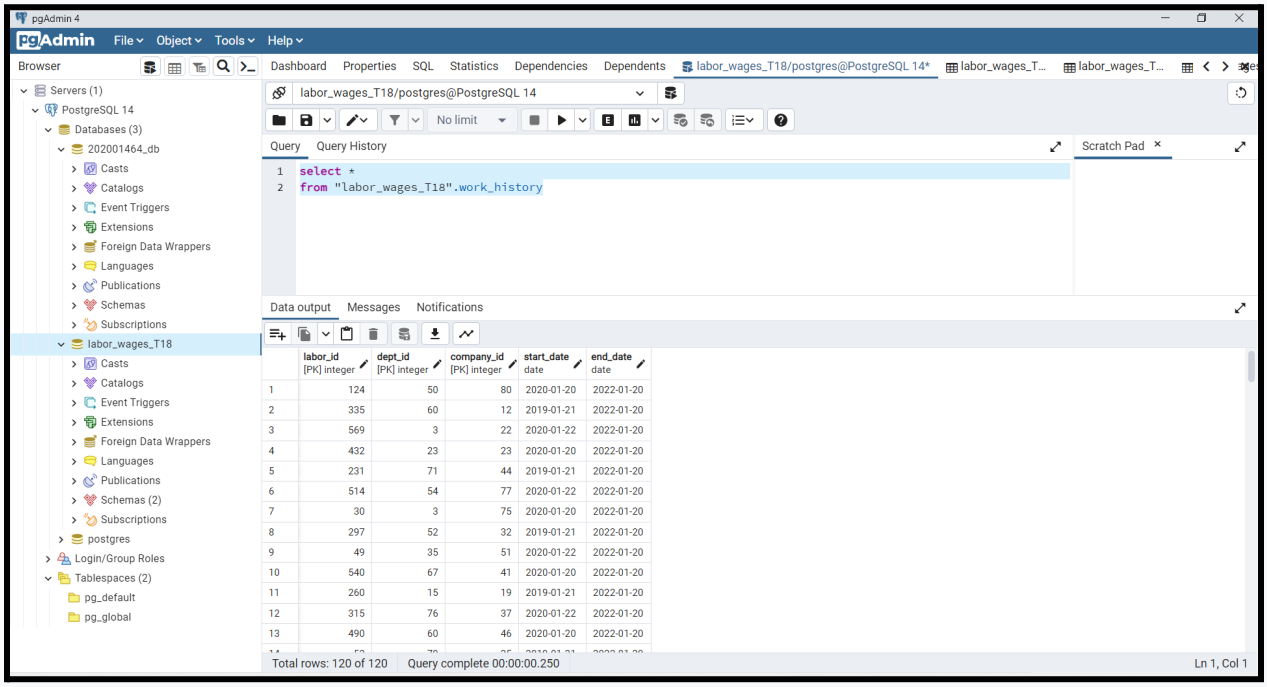


- There are a total 600 labors.

6) show the work history of the workers who have worked before.

➢

```
select *
from "labor_wages_T18".work_history
```

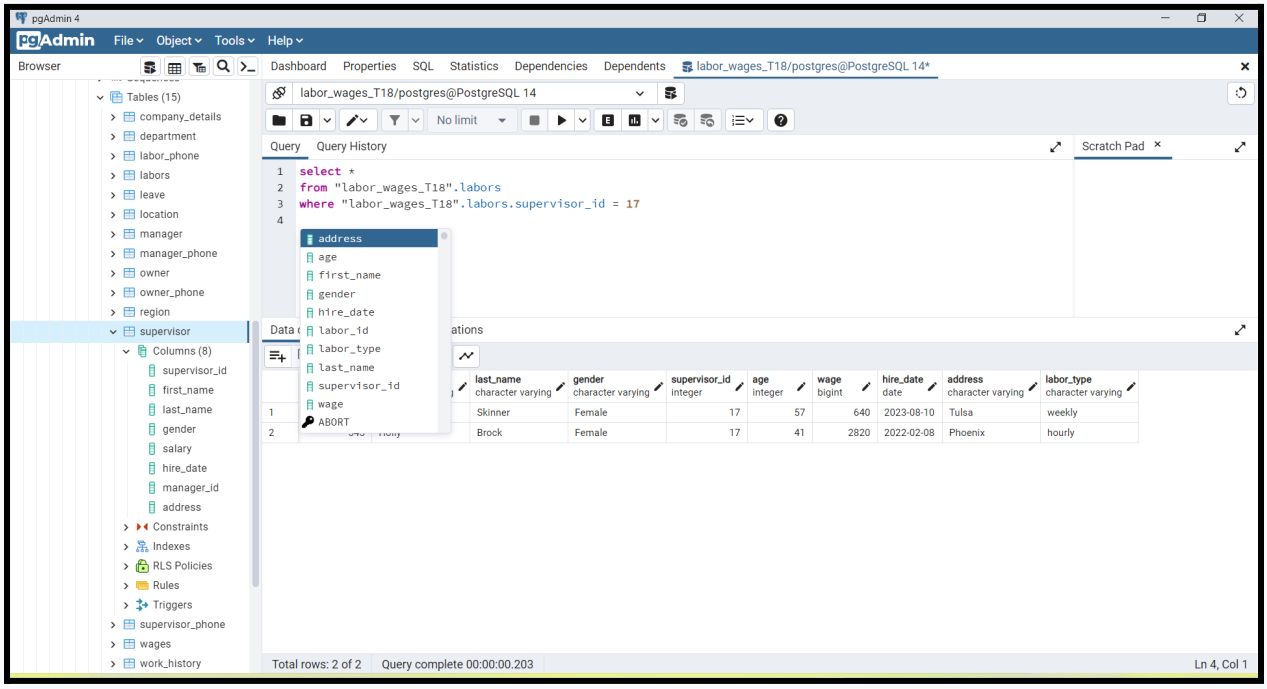


- There are total 120 labors who worked before.

7) show the details of labors who worked under a supervisor whose id is 17.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.supervisor_id = 17
```

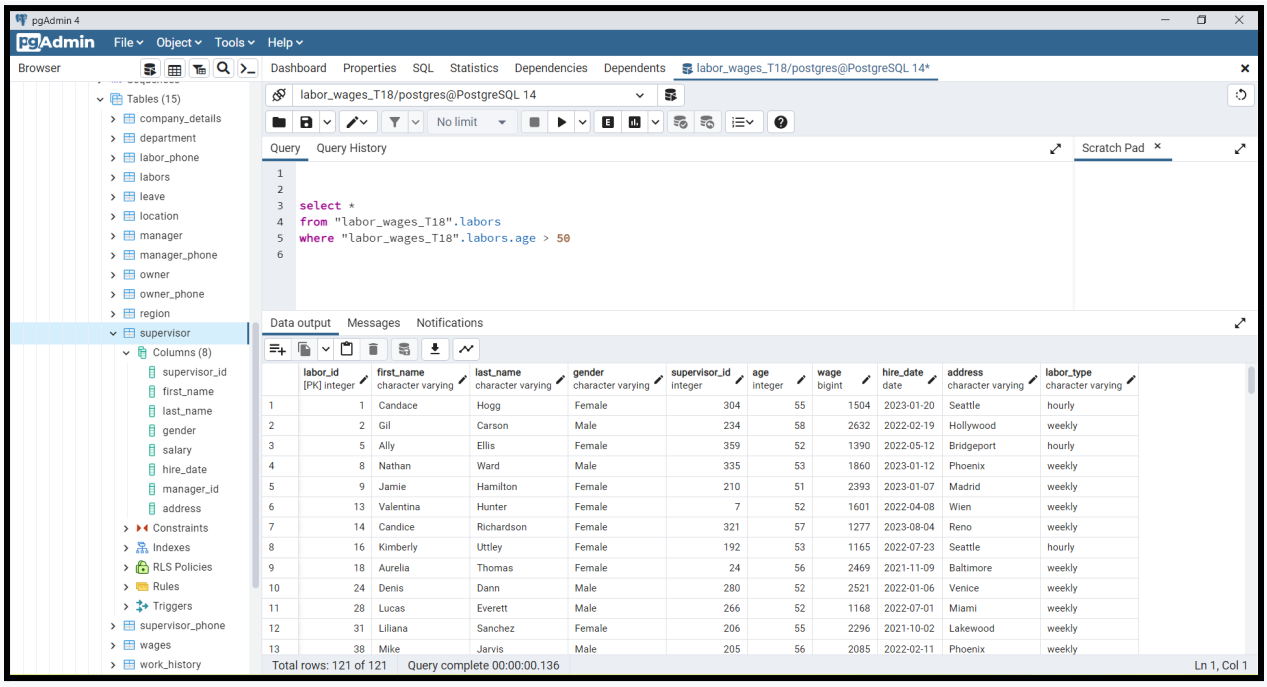


- There are only 2 labors who worked under supervisor id 17.

8) show the details of labors whose age is more then 50.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.age > 50
```



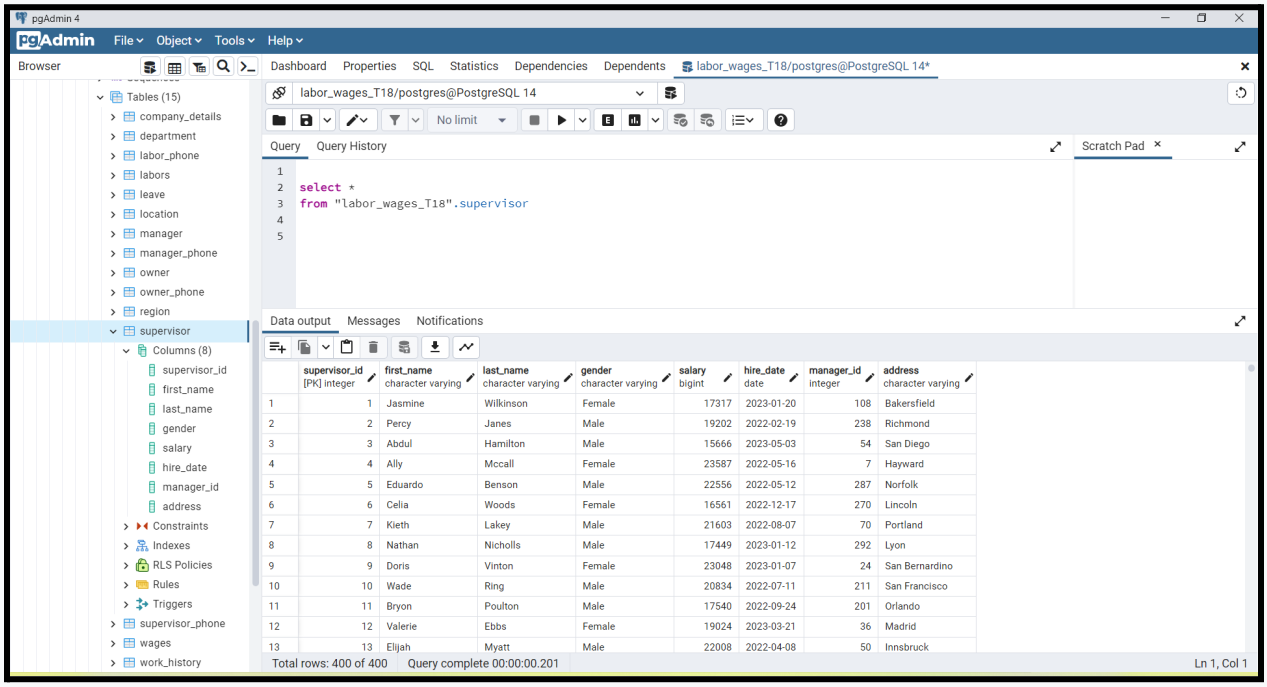
	labor_id [PK] integer	first_name character varying	last_name character varying	gender character varying	supervisor_id integer	age integer	wage bigint	hire_date date	address character varying	labor_type character varying
1	1	Candace	Hogg	Female	304	55	1504	2023-01-20	Seattle	hourly
2	2	Gil	Carson	Male	234	58	2632	2022-02-19	Hollywood	weekly
3	5	Ally	Ellis	Female	359	52	1390	2022-05-12	Bridgeport	hourly
4	8	Nathan	Ward	Male	335	53	1860	2023-01-12	Phoenix	weekly
5	9	Jamie	Hamilton	Female	210	51	2393	2023-01-07	Madrid	weekly
6	13	Valentina	Hunter	Female	7	52	1601	2022-04-08	Wien	weekly
7	14	Candice	Richardson	Female	321	57	1277	2023-08-04	Reno	weekly
8	16	Kimberly	Uttley	Female	192	53	1165	2022-07-23	Seattle	hourly
9	18	Aurelia	Thomas	Female	24	56	2469	2021-11-09	Baltimore	weekly
10	24	Denis	Dann	Male	280	52	2521	2022-01-06	Venice	weekly
11	28	Lucas	Everett	Male	266	52	1168	2022-07-01	Miami	weekly
12	31	Liliana	Sanchez	Female	206	55	2296	2021-10-02	Lakewood	weekly
13	38	Mike	Jarvis	Male	205	56	2085	2022-02-11	Phoenix	weekly

- There is a total of 121 labors whose age is more than 50.

9) show the details of all supervisors.

➢

```
select *
from "labor_wages_T18".supervisor
```



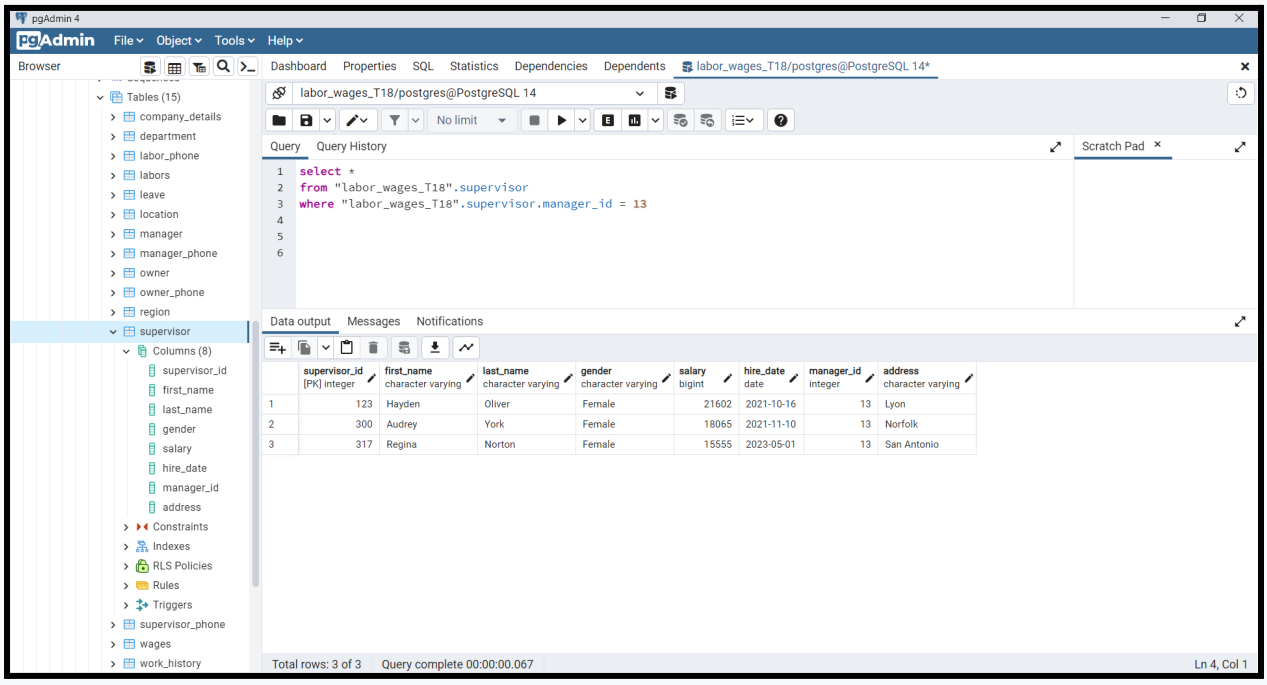
	supervisor_id [PK] integer	first_name character varying	last_name character varying	gender character varying	salary bigint	hire_date date	manager_id integer	address character varying
1	1	Jasmine	Wilkinson	Female	17317	2023-01-20	108	Bakersfield
2	2	Percy	Janes	Male	19202	2022-02-19	238	Richmond
3	3	Abdul	Hamilton	Male	15666	2023-05-03	54	San Diego
4	4	Ally	Mccall	Female	23587	2022-05-16	7	Hayward
5	5	Eduardo	Benson	Male	22556	2022-05-12	287	Norfolk
6	6	Celia	Woods	Female	16561	2022-12-17	270	Lincoln
7	7	Kieth	Lakey	Male	21603	2022-08-07	70	Portland
8	8	Nathan	Nicholls	Male	17449	2023-01-12	292	Lyon
9	9	Doris	Vinton	Female	23048	2023-01-07	24	San Bernardino
10	10	Wade	Ring	Male	20834	2022-07-11	211	San Francisco
11	11	Bryon	Poulton	Male	17540	2022-09-24	201	Orlando
12	12	Valerie	Ebbs	Female	19024	2023-03-21	36	Madrid
13	13	Elijah	Myatt	Male	22008	2022-04-08	50	Innsbruck

- There are a total of 400 supervisors.

10) show the details of a supervisor who has worked under a manager whose id is 13.

➢

```
select *
from "labor_wages_T18".supervisor
where "labor_wages_T18".supervisor.manager_id = 13
```

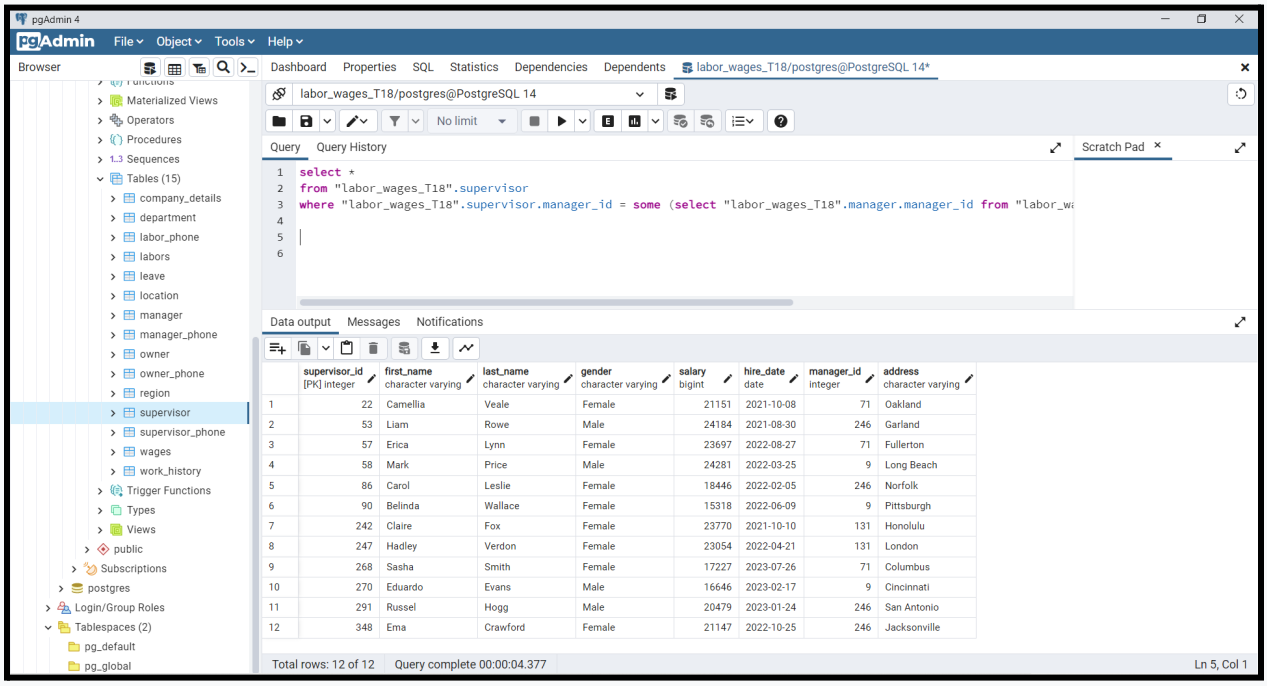


- There are a total of 3 supervisors who has worked under the manager with id 13.

11) Show the details of supervisors who worked at dept id 5.

➢

```
select *
from "labor_wages_T18".supervisor
where "labor_wages_T18".supervisor.manager_id = some (select
"labor_wages_T18".manager.manager_id from
"labor_wages_T18".manager where "labor_wages_T18".manager.dept_id
= 5)
```

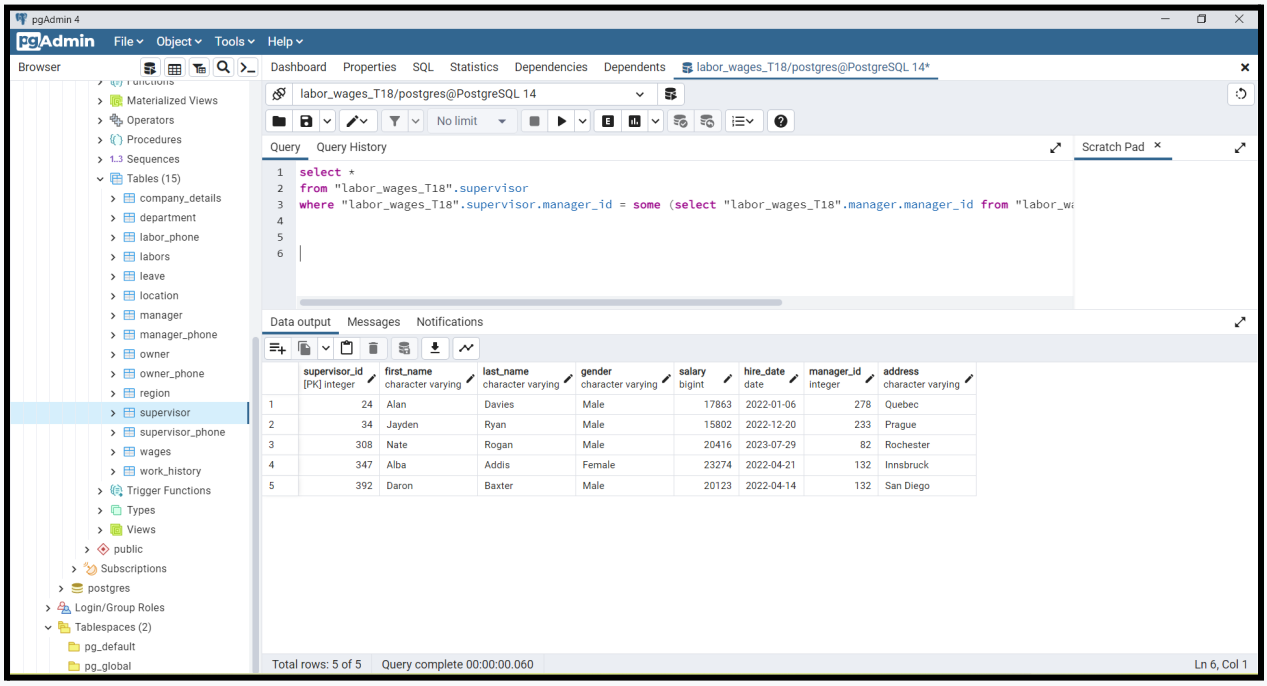


- There are a total of 12 supervisors who has worked at dept id 5.

12) Show the details of the supervisors who has worked at location id 5.

➢

```
select *
from "labor_wages_T18".supervisor
where "labor_wages_T18".supervisor.manager_id = some (select
"labor_wages_T18".manager.manager_id from
"labor_wages_T18".manager where
"labor_wages_T18".manager.location_id = 5)
```

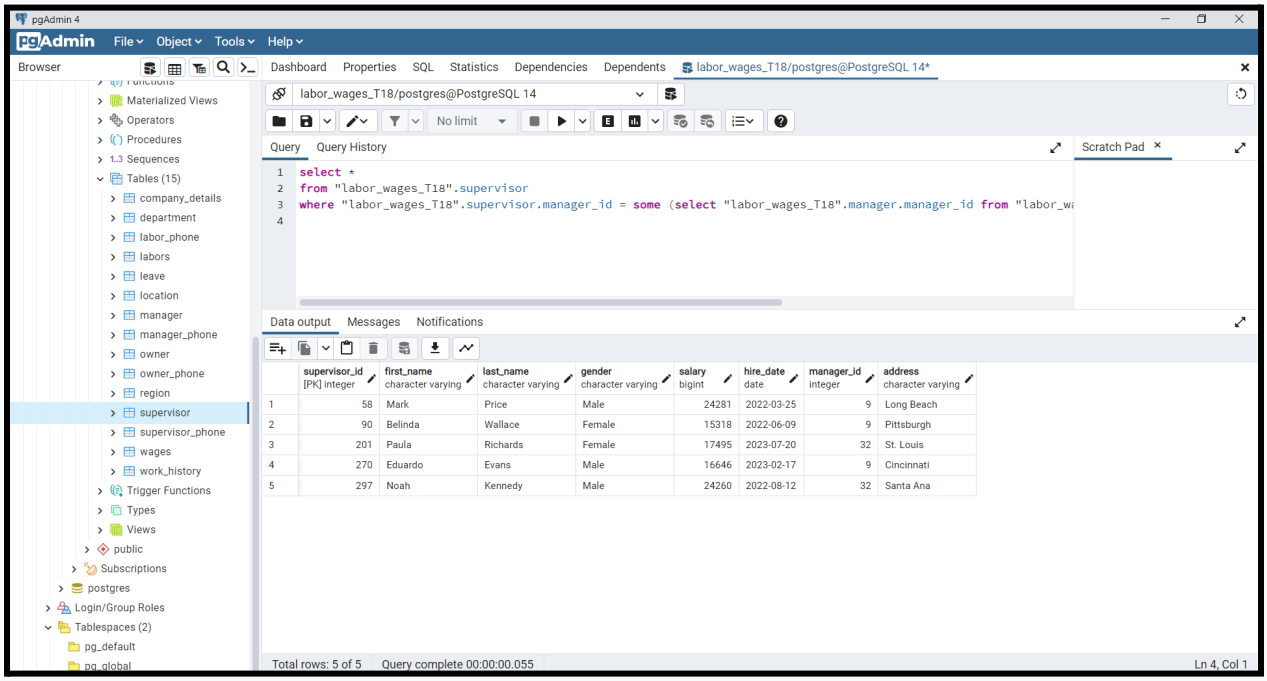


- There are a total of 5 supervisors who has worked at location id 5.

13) Show the details of the supervisor who has worked at company id 25.

➢

```
select *
from "labor_wages_T18".supervisor
where "labor_wages_T18".supervisor.manager_id = some (select
"labor_wages_T18".manager.manager_id from
"labor_wages_T18".manager where
"labor_wages_T18".manager.company_id = 25)
```

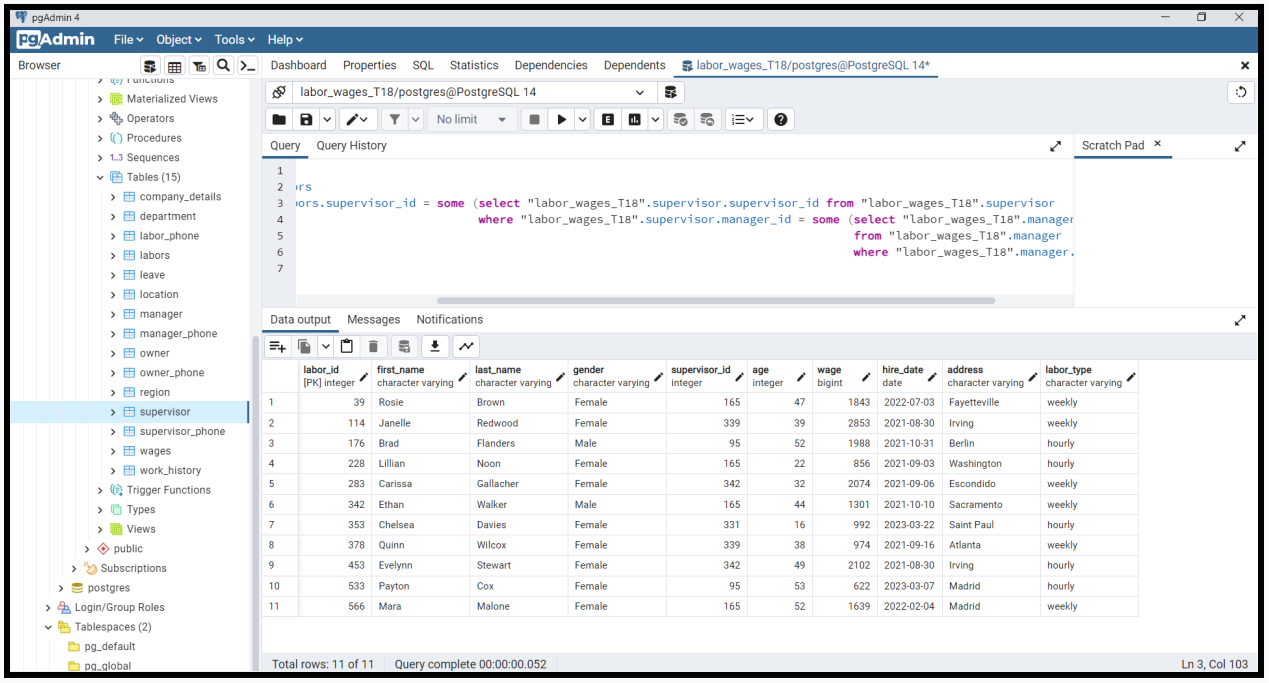


- There are total 5 supervisors who have worked at company id 25.

14) Show the details of labors who has worked at dept id 25.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.supervisor_id = some
(select   "labor_wages_T18".supervisor.supervisor_id
 from "labor_wages_T18".supervisor
 where "labor_wages_T18".supervisor.manager_id = some
 (select "labor_wages_T18".manager.manager_id
 from "labor_wages_T18".manager
 where "labor_wages_T18".manager.dept_id = 25))
```



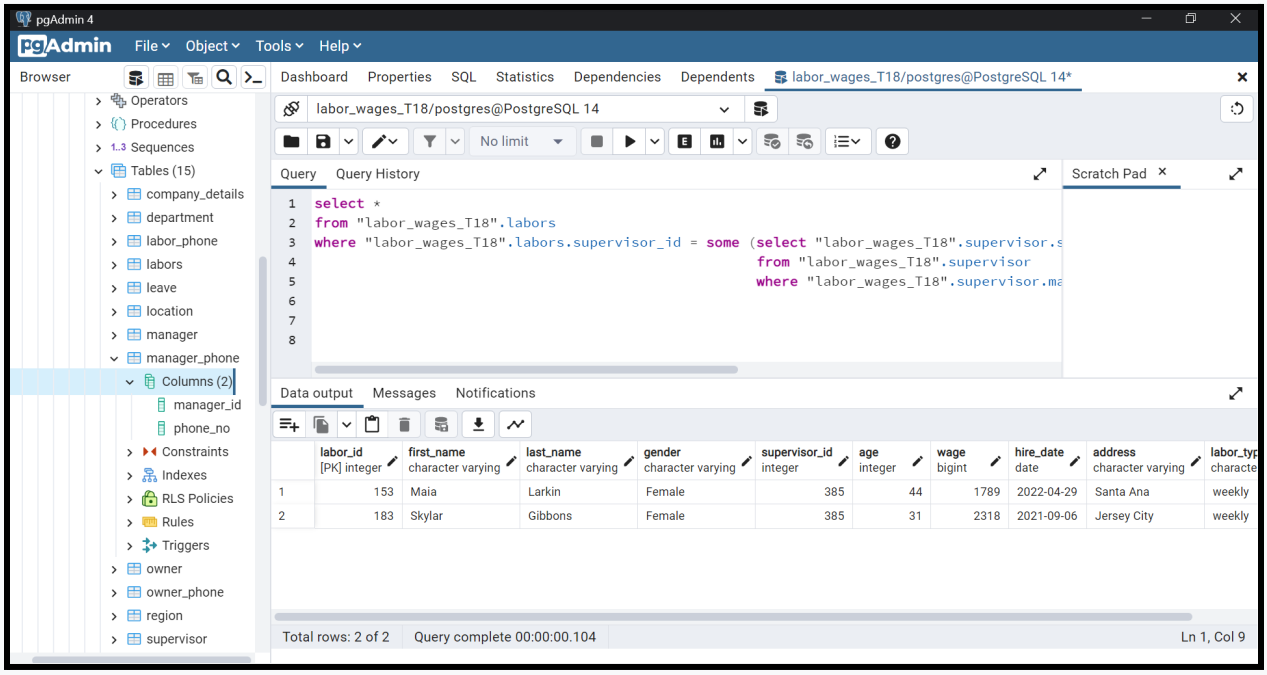
labor_id [PK] integer	first_name character varying	last_name character varying	gender character varying	supervisor_id integer	age integer	wage bigint	hire_date date	address character varying	labor_type character varying
1	39	Rosie	Brown	Female	165	47	1843	2022-07-03	Fayetteville	weekly
2	114	Janelle	Redwood	Female	339	39	2853	2021-08-30	Irving	weekly
3	176	Brad	Flanders	Male	95	52	1988	2021-10-31	Berlin	hourly
4	228	Lillian	Noon	Female	165	22	856	2021-09-03	Washington	hourly
5	283	Carissa	Gallacher	Female	342	32	2074	2021-09-06	Escondido	weekly
6	342	Ethan	Walker	Male	165	44	1301	2021-10-10	Sacramento	weekly
7	353	Chelsea	Davies	Female	331	16	992	2023-03-22	Saint Paul	hourly
8	378	Quinn	Wilcox	Female	339	38	974	2021-09-16	Atlanta	weekly
9	453	Evelynn	Stewart	Female	342	49	2102	2021-08-30	Irving	hourly
10	533	Payton	Cox	Female	95	53	622	2023-03-07	Madrid	hourly
11	566	Mara	Malone	Female	165	52	1639	2022-02-04	Madrid	weekly

- There are total 11 labors who have worked at dept id 25.

15) Show the details of labors who has worked at company id 18.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.supervisor_id = some (select
"labor_wages_T18".supervisor.supervisor_id
from "labor_wages_T18".supervisor
where "labor_wages_T18".supervisor.manager_id = some (select
"labor_wages_T18".manager.manager_id
from "labor_wages_T18".manager
where "labor_wages_T18".manager.company_id = 18))
```

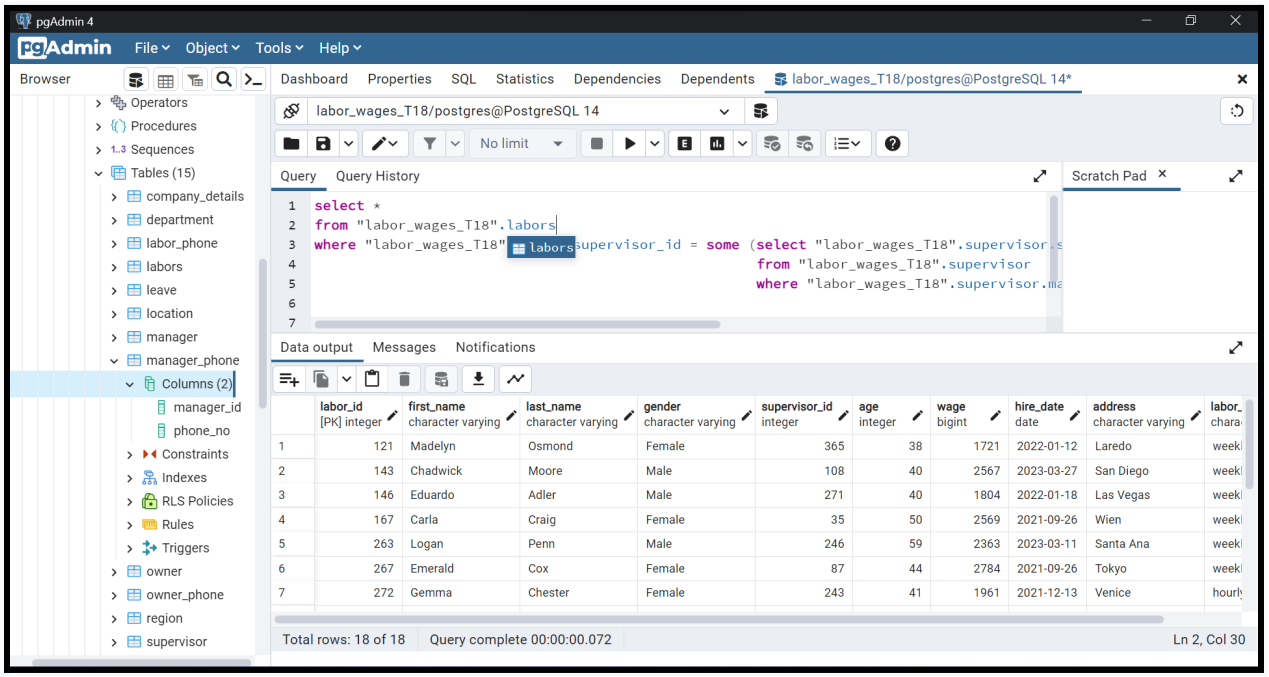


- There are total 2 labors who have worked at company id 18.

16) Show the details of labors who has worked at location id 7.

➢

```
select *
from "labor_wages_T18".labors
where "labor_wages_T18".labors.supervisor_id = some (select
"labor_wages_T18".supervisor.supervisor_id
from "labor_wages_T18".supervisor
where "labor_wages_T18".supervisor.manager_id = some (select
"labor_wages_T18".manager.manager_id
from "labor_wages_T18".manager
where "labor_wages_T18".manager.location_id = 7))
```



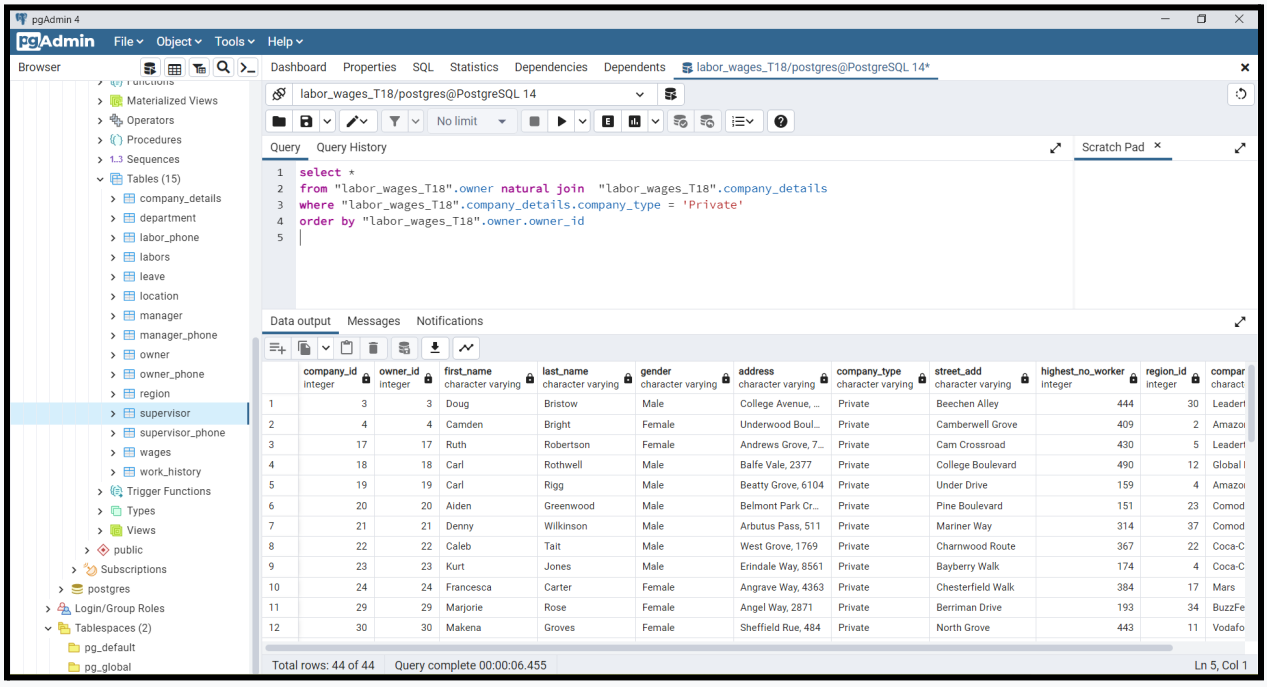
	labor_id [PK] integer	first_name character varying	last_name character varying	gender character varying	supervisor_id integer	age integer	wage bigint	hire_date date	address character varying
1	121	Madelyn	Osmond	Female	365	38	1721	2022-01-12	Laredo
2	143	Chadwick	Moore	Male	108	40	2567	2023-03-27	San Diego
3	146	Eduardo	Adler	Male	271	40	1804	2022-01-18	Las Vegas
4	167	Carla	Craig	Female	35	50	2569	2021-09-26	Wien
5	263	Logan	Penn	Male	246	59	2363	2023-03-11	Santa Ana
6	267	Emerald	Cox	Female	87	44	2784	2021-09-26	Tokyo
7	272	Gemma	Chester	Female	243	41	1961	2021-12-13	Venice

- There are total 18 supervisors who have worked at location id 7.

17) Find owner Details whose company is private and sort inorder of owner id.

➢

```
select *
from "labor_wages_T18".owner natural join
"labor_wages_T18".company_details
where "labor_wages_T18".company_details.company_type = 'Private'
order by "labor_wages_T18".owner.owner_id
```

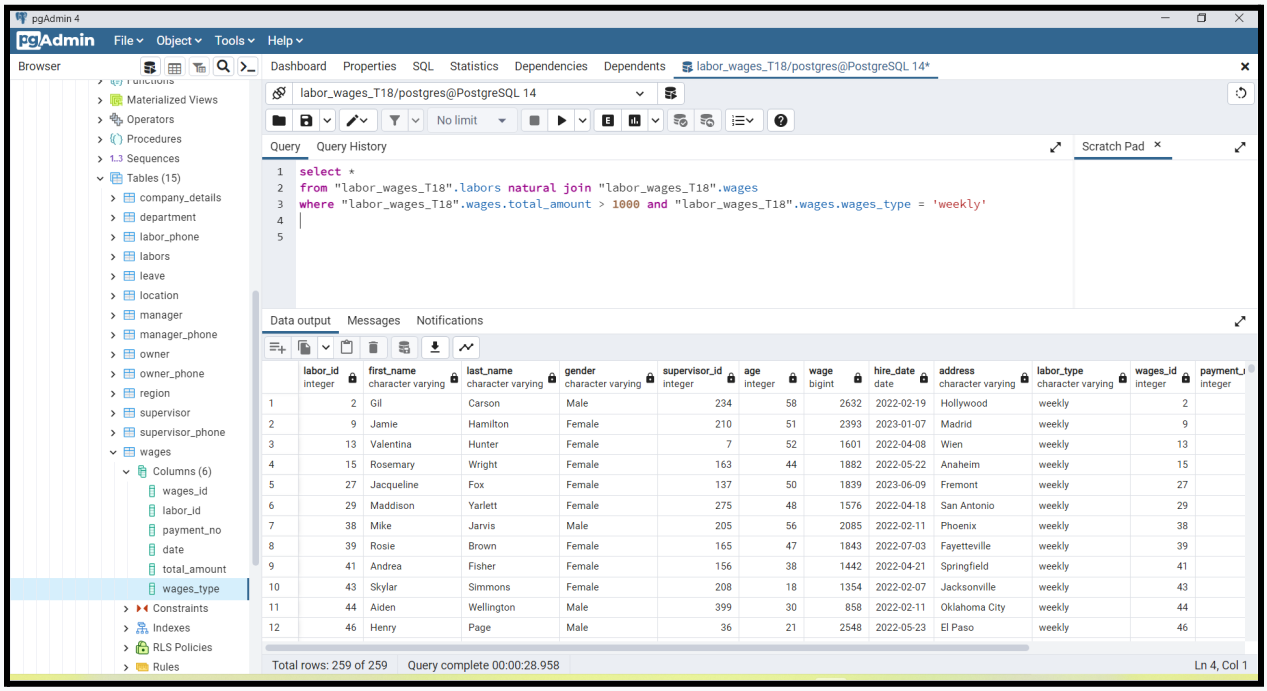


- There are only 44 owners with Private Company.

18) Display all the details of labors whose wages are greater than 1000 and wages type is weekly.

➢

```
select *
from "labor_wages_T18".labors natural join "labor_wages_T18".wages
where "labor_wages_T18".wages.total_amount > 1000 and
"labor_wages_T18".wages.wages_type = 'weekly'
```



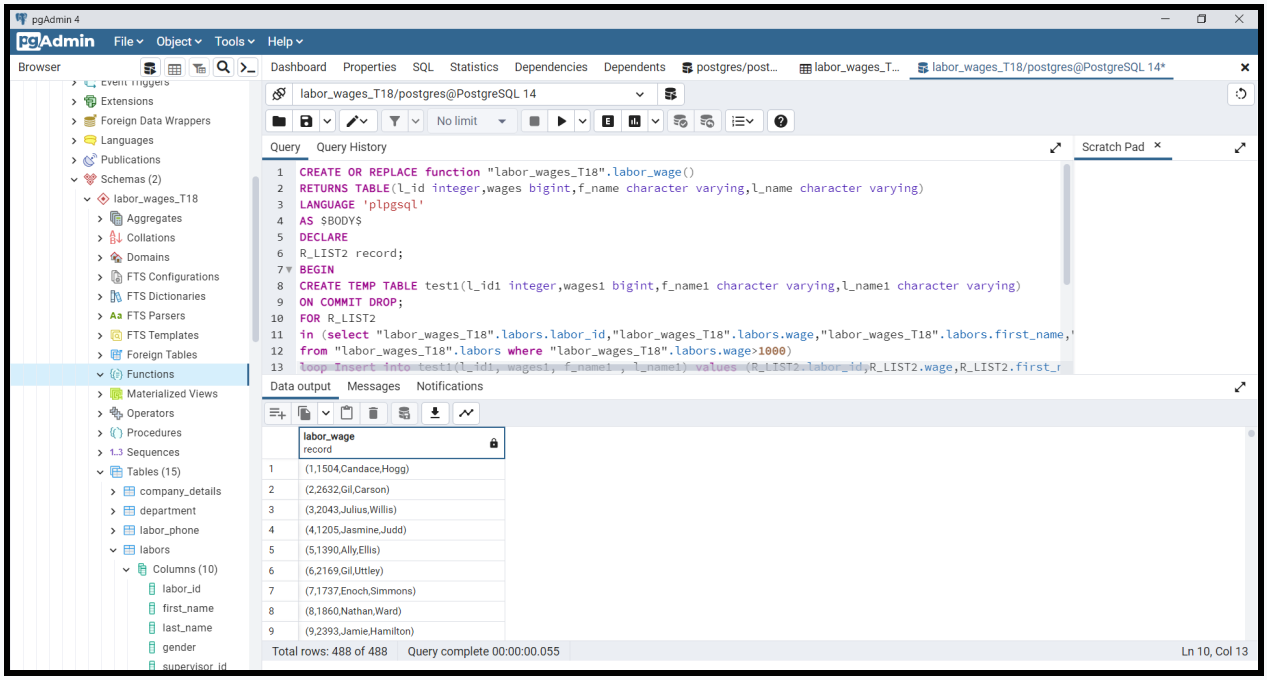
	labor_id integer	first_name character varying	last_name character varying	gender character varying	supervisor_id integer	age integer	wage bigint	hire_date date	address character varying	labor_type character varying	wages_id integer	payment_i integer
1	2	Gil	Carson	Male	234	58	2632	2022-02-19	Hollywood	weekly	2	
2	9	Jamie	Hamilton	Female	210	51	2393	2023-01-07	Madrid	weekly	9	
3	13	Valentina	Hunter	Female	7	52	1601	2022-04-08	Wien	weekly	13	
4	15	Rosemary	Wright	Female	163	44	1882	2022-05-22	Anaheim	weekly	15	
5	27	Jacqueline	Fox	Female	137	50	1839	2023-06-09	Fremont	weekly	27	
6	29	Maddison	Yarlett	Female	275	48	1576	2022-04-18	San Antonio	weekly	29	
7	38	Mike	Jarvis	Male	205	56	2085	2022-02-11	Phoenix	weekly	38	
8	39	Rosie	Brown	Female	165	47	1843	2022-07-03	Fayetteville	weekly	39	
9	41	Andrea	Fisher	Female	156	38	1442	2022-04-21	Springfield	weekly	41	
10	43	Skylar	Simmons	Female	208	18	1354	2022-02-07	Jacksonville	weekly	43	
11	44	Aiden	Wellington	Male	399	30	858	2022-02-11	Oklahoma City	weekly	44	
12	46	Henry	Page	Male	36	21	2548	2022-05-23	El Paso	weekly	46	

- There are 259 labors whose wages are greater than 1000 and wages type is weekly.

19) Find the labor id for wages greater than 1000. Return a temp table with wages id, wages and first_name,last_name in the result table.

➢

```
CREATE OR REPLACE function "labor_wages_T18".labor_wage()
RETURNS TABLE(l_id integer,wages bigint,f_name character
varying,l_name character varying)
LANGUAGE 'plpgsql'
AS $BODY$
DECLARE
R_LIST2 record;
BEGIN
CREATE TEMP TABLE test1(l_id1 integer,wages1 bigint,f_name1
character varying,l_name1 character varying)
ON COMMIT DROP;
FOR R_LIST2
in (select
"labor_wages_T18".labors.labor_id,"labor_wages_T18".labors.wage,"labor
_wages_T18".labors.first_name,"labor_wages_T18".labors.last_name
from "labor_wages_T18".labors where
"labor_wages_T18".labors.wage>1000)
loop Insert into test1(l_id1, wages1, f_name1 , l_name1) values
(R_LIST2.labor_id,R_LIST2.wage,R_LIST2.first_name,R_LIST2.last_nam
e);
end loop;
RETURN QUERY TABLE test1;
END;
$BODY$;

select "labor_wages_T18".labor_wage()
```

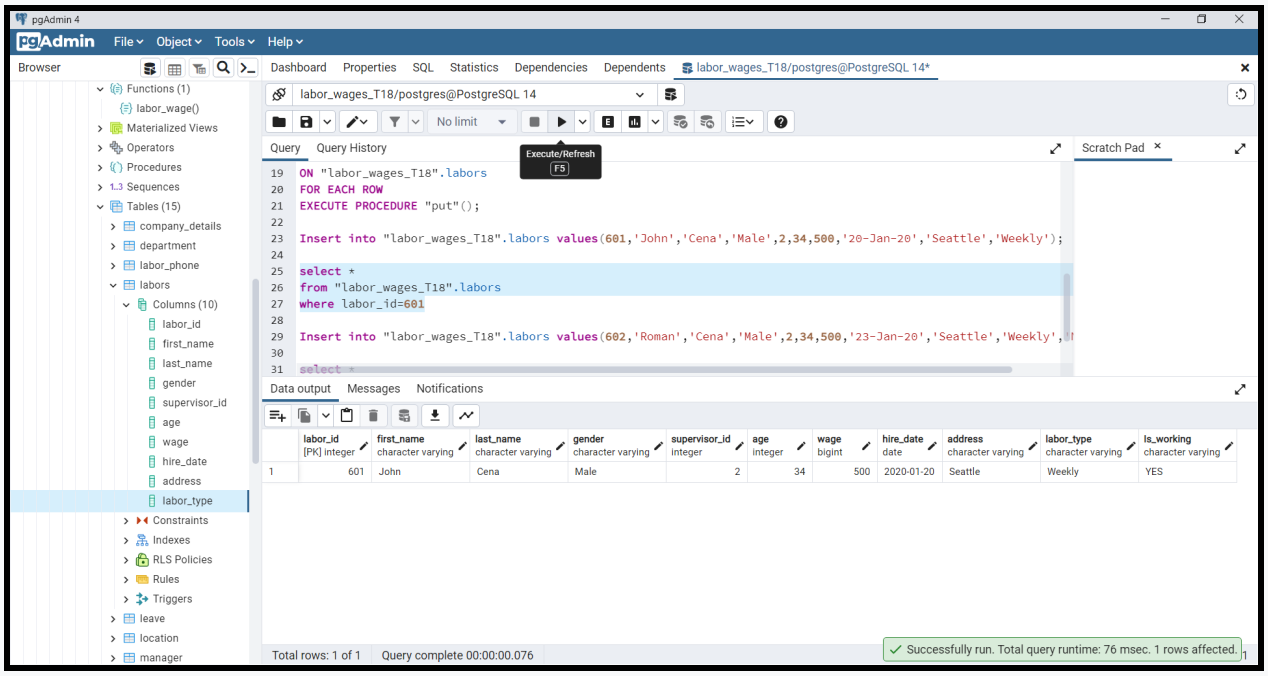


- There are a total of 488 whose wage is more than 1000.

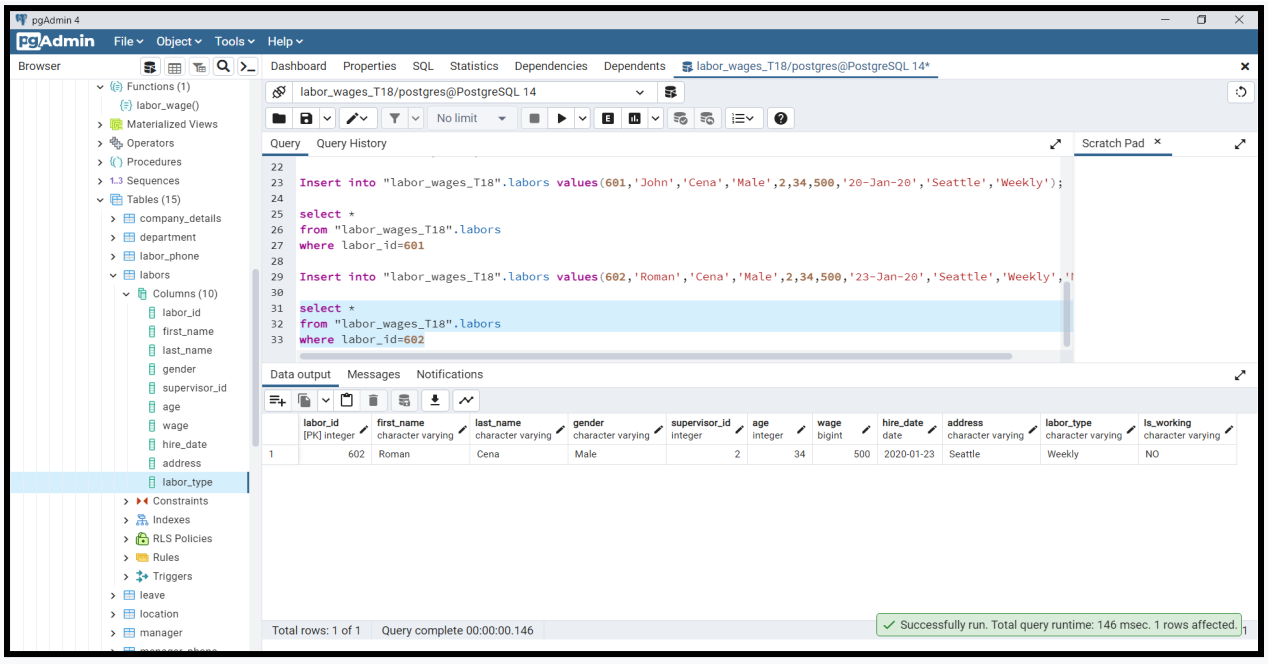
20) Create a column “Is Working” in the labors table. Create a trigger to put the default value Yes for every new entry if nothing is given by the user. Insert new records and check the functionality of Triggers.

➢

```
ALTER TABLE "labor_wages_T18".labors
ADD COLUMN "Is_working" VARCHAR;

CREATE OR REPLACE FUNCTION "put"()
RETURNS trigger
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
if New."Is_working" is NULL THEN
UPDATE "labor_wages_T18".labors SET "Is_working"='YES'
WHERE NEW."labor_id"="labor_id";
End if;
RETURN NEW;
END
$BODY$;

CREATE TRIGGER Trigger_insert_2
AFTER INSERT
ON "labor_wages_T18".labors
FOR EACH ROW
EXECUTE PROCEDURE "put"();

Insert into "labor_wages_T18".labors
values(601,'John','Cena','Male',2,34,500,'20-Jan-20','Seattle','Weekly');

select *
from "labor_wages_T18".labors
where labor_id=601
```

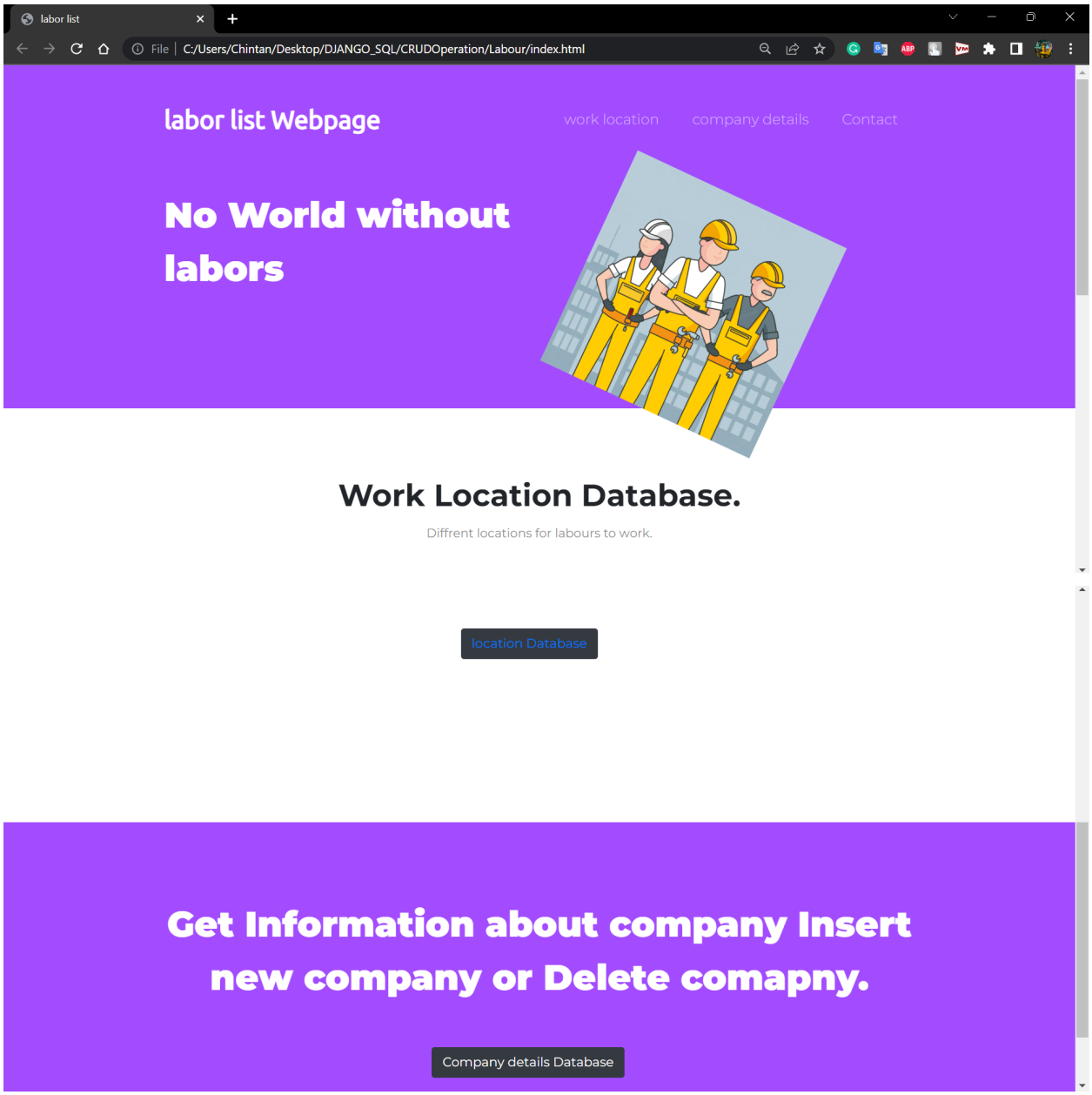



```
Insert into "labor_wages_T18".labors
values(602,'Roman','Cena','Male',2,34,500,'23-Jan-20','Seattle','Weekly','NO');

select *
from "labor_wages_T18".labors
where labor_id=602
```



Section 8 : Project Code with Output

Home Page



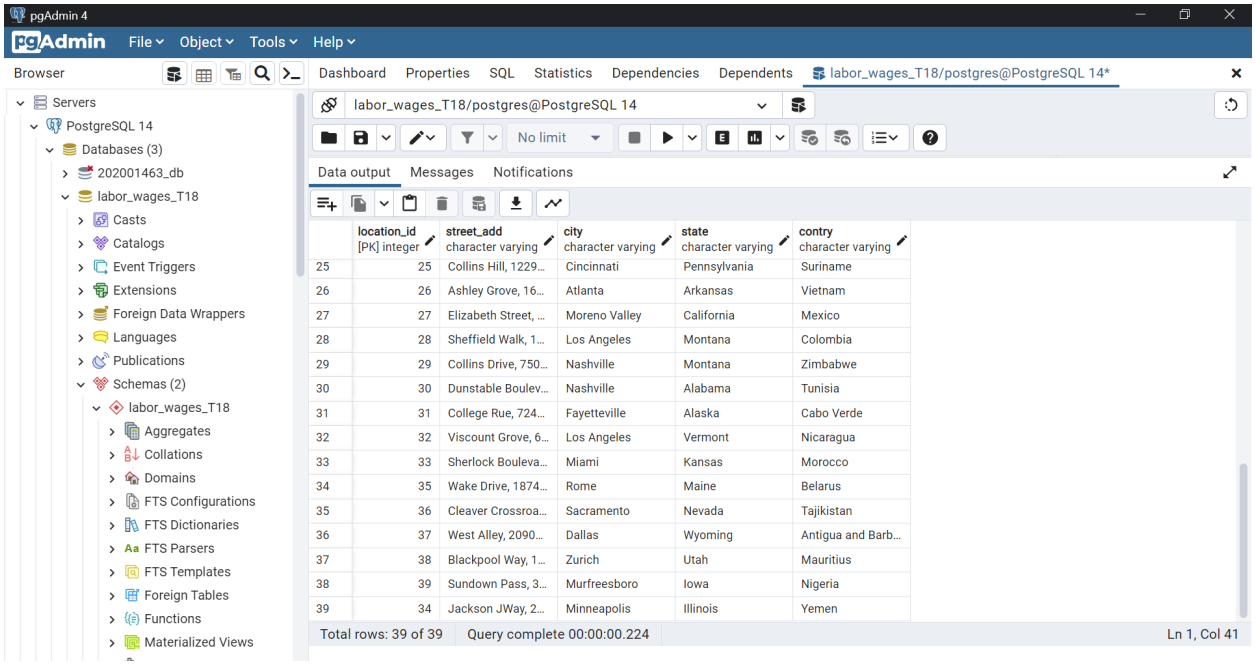
Now When We click on location database we can see location table.

Labor work location details

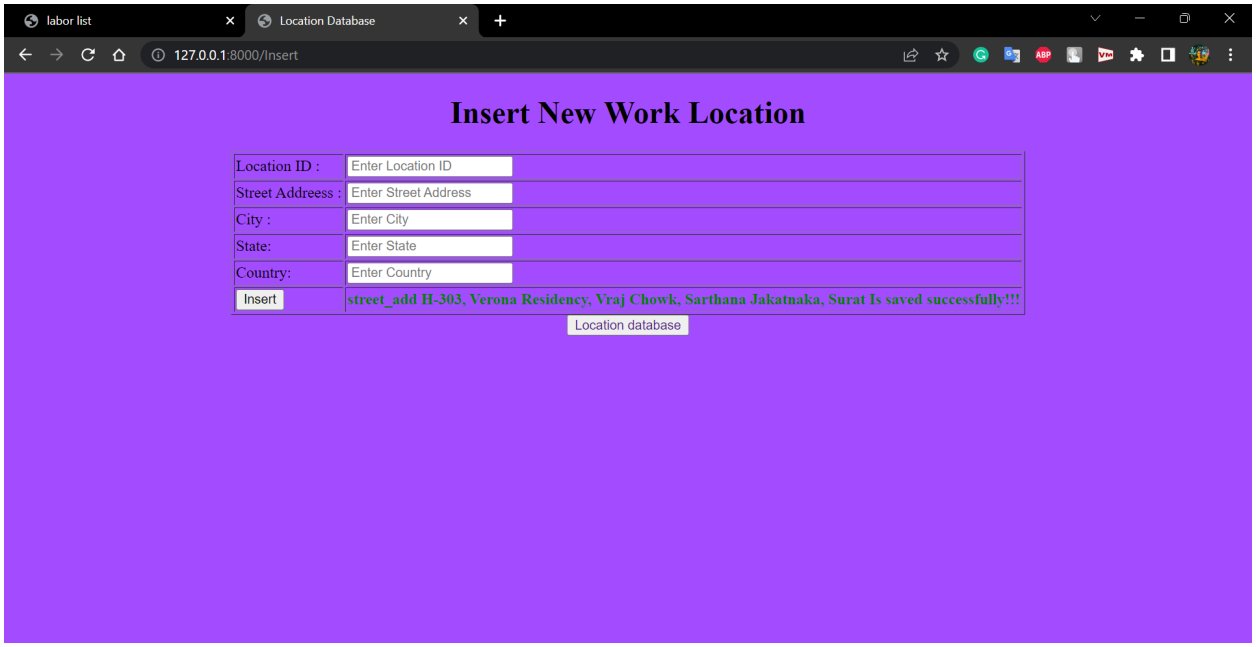
Add New Location

Location ID	Street Address	City	State	Country		
1	Bond Drive, 794660197	Baltimore	Missouri	Sudan, South	Edit	Delete
2	Maple Drive, 689299604	Scottsdale	Vermont	Poland	Edit	Delete
3	Eldon Avenue, 1154456684	Scottsdale	South Carolina	Sri Lanka	Edit	Delete
4	Thomas More Drive, 1872454581	Milano	Colorado	Kazakhstan	Edit	Delete
5	Yoakley Way, 1239519437	El Paso	Oregon	Mongolia	Edit	Delete
6	Collins Walk, 2001791006	Atlanta	Nevada	Slovakia	Edit	Delete
7	Lake Avenue, 894690276	Norfolk	North Carolina	Georgia	Edit	Delete
8	Blore Crossroad, 889083386	Worcester	Kentucky	Nicaragua	Edit	Delete
9	Chapel Walk, 1611096456	Lisbon	Alaska	Sweden	Edit	Delete
10	Bazely Avenue, 728128856	Colorado Springs	South Dakota	Spain	Edit	Delete
11	Angel Alley, 2024900926	Venice	Ohio	Jordan	Edit	Delete
12	Tiptree Route, 1305671246	Henderson	West Virginia	Australia	Edit	Delete
13	Norfolk Tunnel, 2136717172	Colorado Springs	Indiana	Ireland	Edit	Delete
14	Byland Hill, 1942951036	Worcester	Wisconsin	Mauritius	Edit	Delete
15	Carolina Way, 2001680832	Hollywood	Arizona	Slovenia	Edit	Delete
16	Bolingbroke Street, 1777222994	Huntsville	Alabama	Ukraine	Edit	Delete
17	Coal Wharf Alley, 1881900694	Santa Ana	Mississippi	Angola	Edit	Delete
18	East Tunnel, 383133521	New Orleans	Vermont	Israel	Edit	Delete
19	Blackpool Grove, 135812131	Rome	West Virginia	Algeria	Edit	Delete
20	Hammersmith Hill, 1281151949	Phoenix	Delaware	Mongolia	Edit	Delete
21	Sundown Walk, 295118919	Kansas City	Oregon	Myanmar	Edit	Delete
22	Camdenhurst Rue, 864127423	Henderson	Tennessee	Pakistan	Edit	Delete
23	Cliff Route, 1631758222	Saint Paul	Kansas	Iraq	Edit	Delete
24	Battersea Alley, 395121542	San Francisco	Oregon	Kenya	Edit	Delete
25	Collins Hill, 1229312369	Cincinnati	Pennsylvania	Suriname	Edit	Delete
26	Ashley Grove, 1662866123	Atlanta	Arkansas	Vietnam	Edit	Delete
27	Elizabeth Street, 384781444	Moreno Valley	California	Mexico	Edit	Delete
28	Sheffield Walk, 1735338303	Los Angeles	Montana	Colombia	Edit	Delete
29	Collins Drive, 750360612	Nashville	Montana	Zimbabwe	Edit	Delete
30	Dunstable Boulevard, 1586602146	Nashville	Alabama	Tunisia	Edit	Delete
31	College Rue, 724985039	Fayetteville	Alaska	Cabo Verde	Edit	Delete
32	Viscount Grove, 611769865	Los Angeles	Vermont	Nicaragua	Edit	Delete
33	Sherlock Boulevard, 1997912390	Miami	Kansas	Morocco	Edit	Delete
35	Wake Drive, 1874737725	Rome	Maine	Belarus	Edit	Delete
36	Cleaver Crossroad, 703404092	Sacramento	Nevada	Tajikistan	Edit	Delete
37	West Alley, 2090294340	Dallas	Wyoming	Antigua and Barbuda	Edit	Delete
38	Blackpool Way, 1245628548	Zurich	Utah	Mauritius	Edit	Delete
39	Sundown Pass, 322860470	Murfreesboro	Iowa	Nigeria	Edit	Delete
34	Jackson JWay, 2028720650	Minneapolis	Illinois	Yemen	Edit	Delete

Pg admin location table we can see the same data.



Now Add the New Work Location



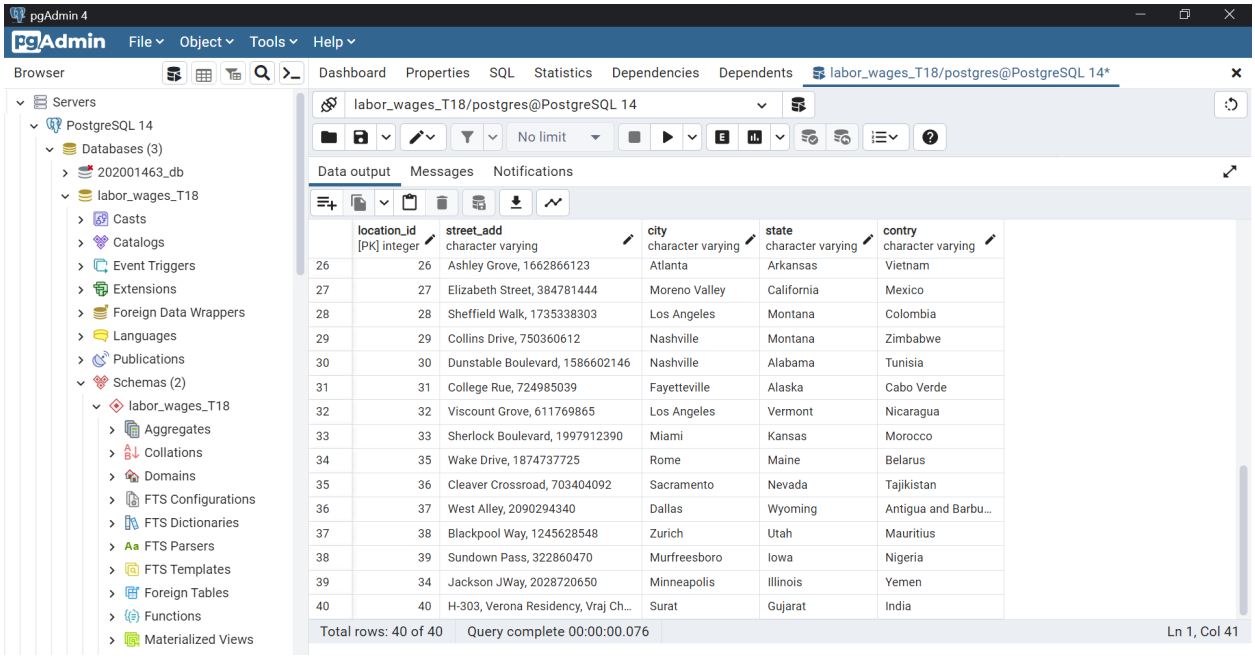
Now we can see added location

Labor work location details

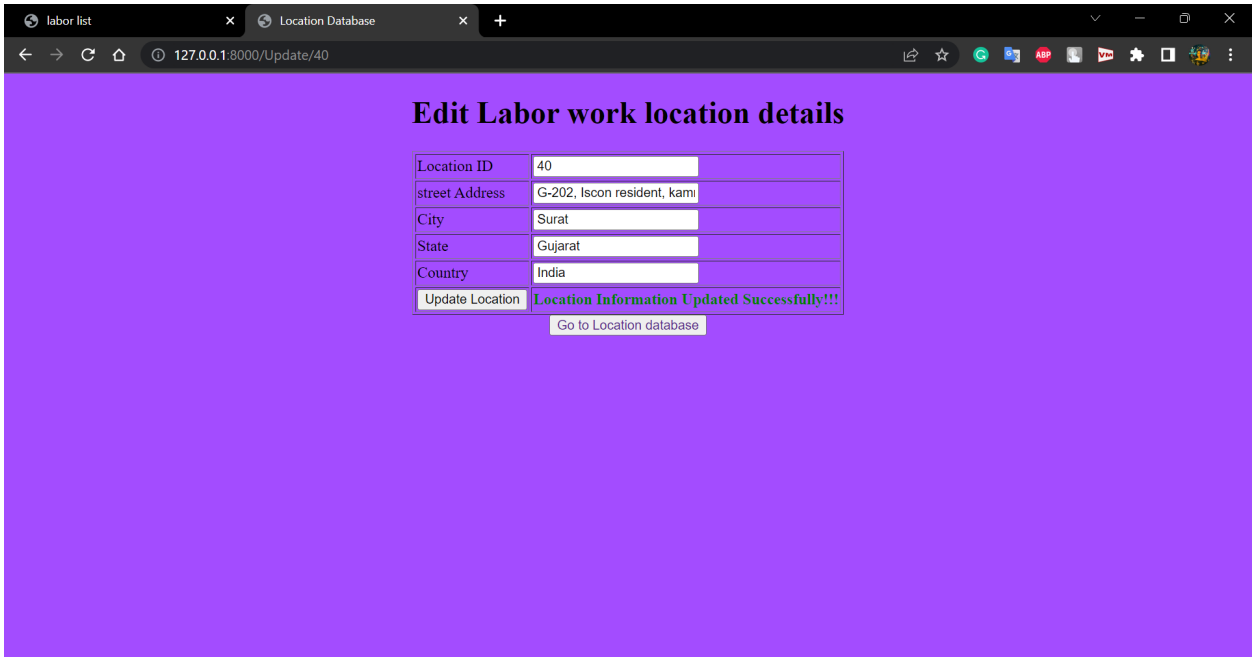
Add New Location

Location ID	Street Address	City	State	Country		
1	Bond Drive, 794660197	Baltimore	Missouri	Sudan, South	Edit	Delete
2	Maple Drive, 689299604	Scottsdale	Vermont	Poland	Edit	Delete
3	Eldon Avenue, 1154456684	Scottsdale	South Carolina	Sri Lanka	Edit	Delete
4	Thomas More Drive, 1872454581	Milano	Colorado	Kazakhstan	Edit	Delete
5	Yoakley Way, 1239519437	El Paso	Oregon	Mongolia	Edit	Delete
6	Collins Walk, 2001791006	Atlanta	Nevada	Slovakia	Edit	Delete
7	Lake Avenue, 894690276	Norfolk	North Carolina	Georgia	Edit	Delete
8	Blore Crossroad, 889083386	Worcester	Kentucky	Nicaragua	Edit	Delete
9	Chapel Walk, 1611096456	Lisbon	Alaska	Sweden	Edit	Delete
10	Bazely Avenue, 728128856	Colorado Springs	South Dakota	Spain	Edit	Delete
11	Angel Alley, 2024900926	Venice	Ohio	Jordan	Edit	Delete
12	Tiptree Route, 1305671246	Henderson	West Virginia	Australia	Edit	Delete
13	Norfolk Tunnel, 2136717172	Colorado Springs	Indiana	Ireland	Edit	Delete
14	Byland Hill, 1942951036	Worcester	Wisconsin	Mauritius	Edit	Delete
15	Carolina Way, 2001680832	Hollywood	Arizona	Slovenia	Edit	Delete
16	Bolingbroke Street, 1777222994	Huntsville	Alabama	Ukraine	Edit	Delete
17	Coal Wharf Alley, 1881900694	Santa Ana	Mississippi	Angola	Edit	Delete
18	East Tunnel, 383133521	New Orleans	Vermont	Israel	Edit	Delete
19	Blackpool Grove, 135812131	Rome	West Virginia	Algeria	Edit	Delete
20	Hammersmith Hill, 1281151949	Phoenix	Delaware	Mongolia	Edit	Delete
21	Sundown Walk, 295118919	Kansas City	Oregon	Myanmar	Edit	Delete
22	Camdenhurst Rue, 864127423	Henderson	Tennessee	Pakistan	Edit	Delete
23	Cliff Route, 1631758222	Saint Paul	Kansas	Iraq	Edit	Delete
24	Battersea Alley, 395121542	San Francisco	Oregon	Kenya	Edit	Delete
25	Collins Hill, 1229312369	Cincinnati	Pennsylvania	Suriname	Edit	Delete
26	Ashley Grove, 1662866123	Atlanta	Arkansas	Vietnam	Edit	Delete
27	Elizabeth Street, 384781444	Moreno Valley	California	Mexico	Edit	Delete
28	Sheffield Walk, 1735338303	Los Angeles	Montana	Colombia	Edit	Delete
29	Collins Drive, 750360612	Nashville	Montana	Zimbabwe	Edit	Delete
30	Dunstable Boulevard, 1586602146	Nashville	Alabama	Tunisia	Edit	Delete
31	College Rue, 724985039	Fayetteville	Alaska	Cabo Verde	Edit	Delete
32	Viscount Grove, 611769865	Los Angeles	Vermont	Nicaragua	Edit	Delete
33	Sherlock Boulevard, 1997912390	Miami	Kansas	Morocco	Edit	Delete
35	Wake Drive, 1874737725	Rome	Maine	Belarus	Edit	Delete
36	Cleaver Crossroad, 703404092	Sacramento	Nevada	Tajikistan	Edit	Delete
37	West Alley, 2090294340	Dallas	Wyoming	Antigua and Barbuda	Edit	Delete
38	Blackpool Way, 1245628548	Zurich	Utah	Mauritius	Edit	Delete
39	Sundown Pass, 322860470	Murfreesboro	Iowa	Nigeria	Edit	Delete
34	Jackson JWay, 2028720650	Minneapolis	Illinois	Yemen	Edit	Delete
40	H-303, Verona Residency, Vraj Chowk, Sarthana Jakatnaka, Surat	Surat	Gujarat	India	Edit	Delete

Pg admin location table



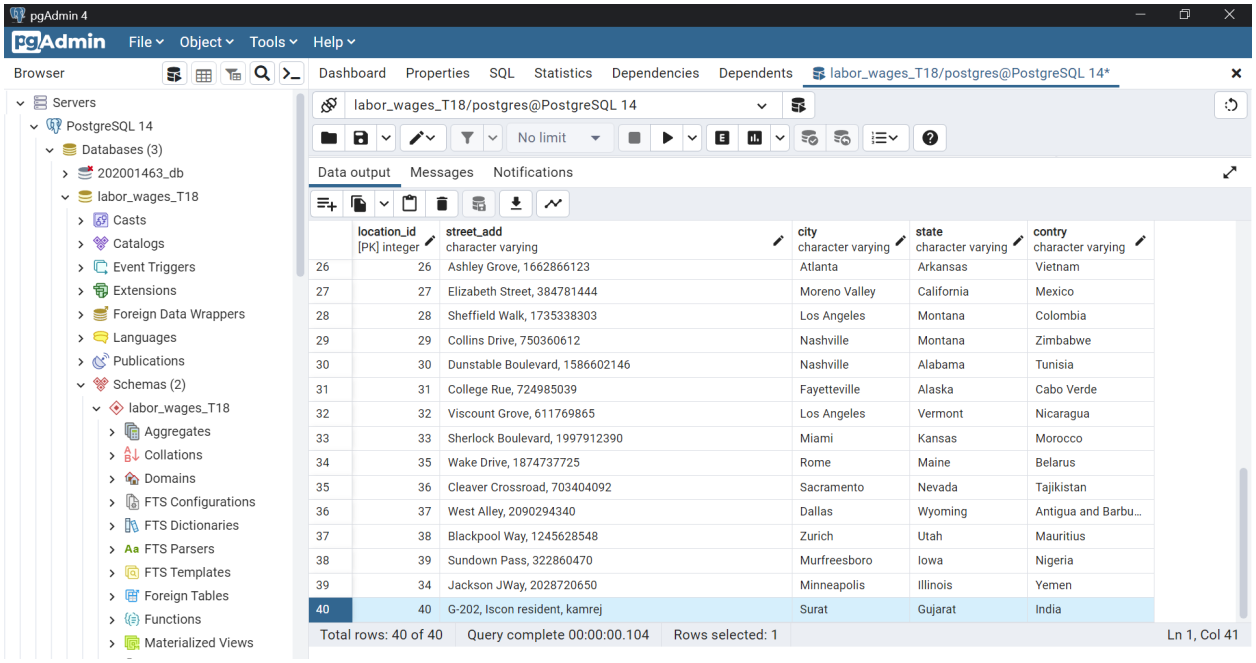
Edit location



We can see that for location id = 40, street address is edited.

20	Hammersmith Hill, 1281151949	Phoenix	Delaware	Mongolia	Edit	Delete
21	Sundown Walk, 295118919	Kansas City	Oregon	Myanmar	Edit	Delete
22	Camdenhurst Rue, 864127423	Henderson	Tennessee	Pakistan	Edit	Delete
23	Cliff Route, 1631758222	Saint Paul	Kansas	Iraq	Edit	Delete
24	Battersea Alley, 395121542	San Francisco	Oregon	Kenya	Edit	Delete
25	Collins Hill, 1229312369	Cincinnati	Pennsylvania	Suriname	Edit	Delete
26	Ashley Grove, 1662866123	Atlanta	Arkansas	Vietnam	Edit	Delete
27	Elizabeth Street, 384781444	Moreno Valley	California	Mexico	Edit	Delete
28	Sheffield Walk, 1735338303	Los Angeles	Montana	Colombia	Edit	Delete
29	Collins Drive, 750360612	Nashville	Montana	Zimbabwe	Edit	Delete
30	Dunstable Boulevard, 1586602146	Nashville	Alabama	Tunisia	Edit	Delete
31	College Rue, 724985039	Fayetteville	Alaska	Cabo Verde	Edit	Delete
32	Viscount Grove, 611769865	Los Angeles	Vermont	Nicaragua	Edit	Delete
33	Sherlock Boulevard, 1997912390	Miami	Kansas	Morocco	Edit	Delete
35	Wake Drive, 1874737725	Rome	Maine	Belarus	Edit	Delete
36	Cleaver Crossroad, 703404092	Sacramento	Nevada	Tajikistan	Edit	Delete
37	West Alley, 2090294340	Dallas	Wyoming	Antigua and Barbuda	Edit	Delete
38	Blackpool Way, 1245628548	Zurich	Utah	Mauritius	Edit	Delete
39	Sundown Pass, 322860470	Murfreesboro	Iowa	Nigeria	Edit	Delete
34	Jackson JWay, 2028720650	Minneapolis	Illinois	Yemen	Edit	Delete
40	G-202, Iscon resident, kamrej	Surat	Gujarat	India	Edit	Delete

Pg admin location table



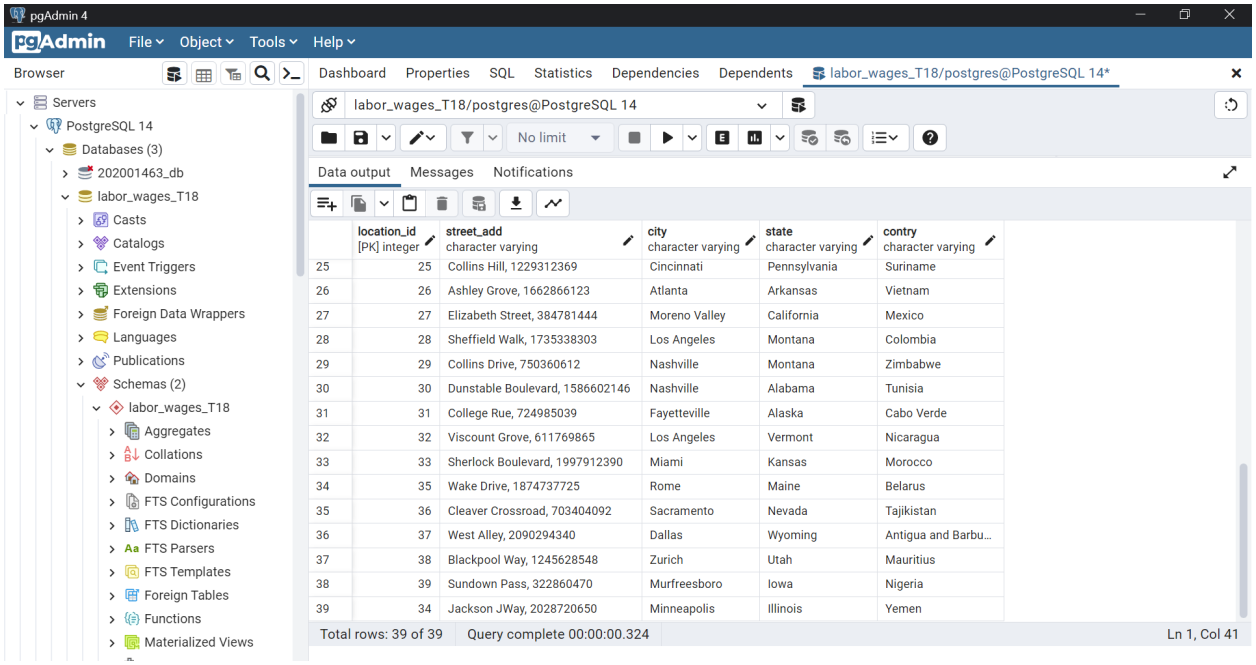
Now delete the data for which location id = 40

20	Hammersmith				Edit	Delete
21	Sundown Wa				Edit	Delete
22	Camdenhurst				Edit	Delete
23	Cliff Route, 1				Edit	Delete
24	Battersea Alley, 395121542	San Francisco	Oregon	Kenya	Edit	Delete
25	Collins Hill, 1229312369	Cincinnati	Pennsylvania	Suriname	Edit	Delete
26	Ashley Grove, 1662866123	Atlanta	Arkansas	Vietnam	Edit	Delete
27	Elizabeth Street, 384781444	Moreno Valley	California	Mexico	Edit	Delete
28	Sheffield Walk, 1735338303	Los Angeles	Montana	Colombia	Edit	Delete
29	Collins Drive, 750360612	Nashville	Montana	Zimbabwe	Edit	Delete
30	Dunstable Boulevard, 1586602146	Nashville	Alabama	Tunisia	Edit	Delete
31	College Rue, 724985039	Fayetteville	Alaska	Cabo Verde	Edit	Delete
32	Viscount Grove, 611769865	Los Angeles	Vermont	Nicaragua	Edit	Delete
33	Sherlock Boulevard, 1997912390	Miami	Kansas	Morocco	Edit	Delete
35	Wake Drive, 1874737725	Rome	Maine	Belarus	Edit	Delete
36	Cleaver Crossroad, 703404092	Sacramento	Nevada	Tajikistan	Edit	Delete
37	West Alley, 2090294340	Dallas	Wyoming	Antigua and Barbuda	Edit	Delete
38	Blackpool Way, 1245628548	Zurich	Utah	Mauritius	Edit	Delete
39	Sundown Pass, 322860470	Murfreesboro	Iowa	Nigeria	Edit	Delete
34	Jackson JWay, 2028720650	Minneapolis	Illinois	Yemen	Edit	Delete
40	G-202, Iscon resident, kamrej	Surat	Gujarat	India	Edit	Delete

For Location id = 40, information deleted from the table

20	Hammersmith Hill, 1281151949	Phoenix	Delaware	Mongolia	Edit	Delete
21	Sundown Walk, 295118919	Kansas City	Oregon	Myanmar	Edit	Delete
22	Camdenhurst Rue, 864127423	Henderson	Tennessee	Pakistan	Edit	Delete
23	Cliff Route, 1631758222	Saint Paul	Kansas	Iraq	Edit	Delete
24	Battersea Alley, 395121542	San Francisco	Oregon	Kenya	Edit	Delete
25	Collins Hill, 1229312369	Cincinnati	Pennsylvania	Suriname	Edit	Delete
26	Ashley Grove, 1662866123	Atlanta	Arkansas	Vietnam	Edit	Delete
27	Elizabeth Street, 384781444	Moreno Valley	California	Mexico	Edit	Delete
28	Sheffield Walk, 1735338303	Los Angeles	Montana	Colombia	Edit	Delete
29	Collins Drive, 750360612	Nashville	Montana	Zimbabwe	Edit	Delete
30	Dunstable Boulevard, 1586602146	Nashville	Alabama	Tunisia	Edit	Delete
31	College Rue, 724985039	Fayetteville	Alaska	Cabo Verde	Edit	Delete
32	Viscount Grove, 611769865	Los Angeles	Vermont	Nicaragua	Edit	Delete
33	Sherlock Boulevard, 1997912390	Miami	Kansas	Morocco	Edit	Delete
35	Wake Drive, 1874737725	Rome	Maine	Belarus	Edit	Delete
36	Cleaver Crossroad, 703404092	Sacramento	Nevada	Tajikistan	Edit	Delete
37	West Alley, 2090294340	Dallas	Wyoming	Antigua and Barbuda	Edit	Delete
38	Blackpool Way, 1245628548	Zurich	Utah	Mauritius	Edit	Delete
39	Sundown Pass, 322860470	Murfreesboro	Iowa	Nigeria	Edit	Delete
34	Jackson JWay, 2028720650	Minneapolis	Illinois	Yemen	Edit	Delete

Pg admin location table



Now When We click on company database we can see company table

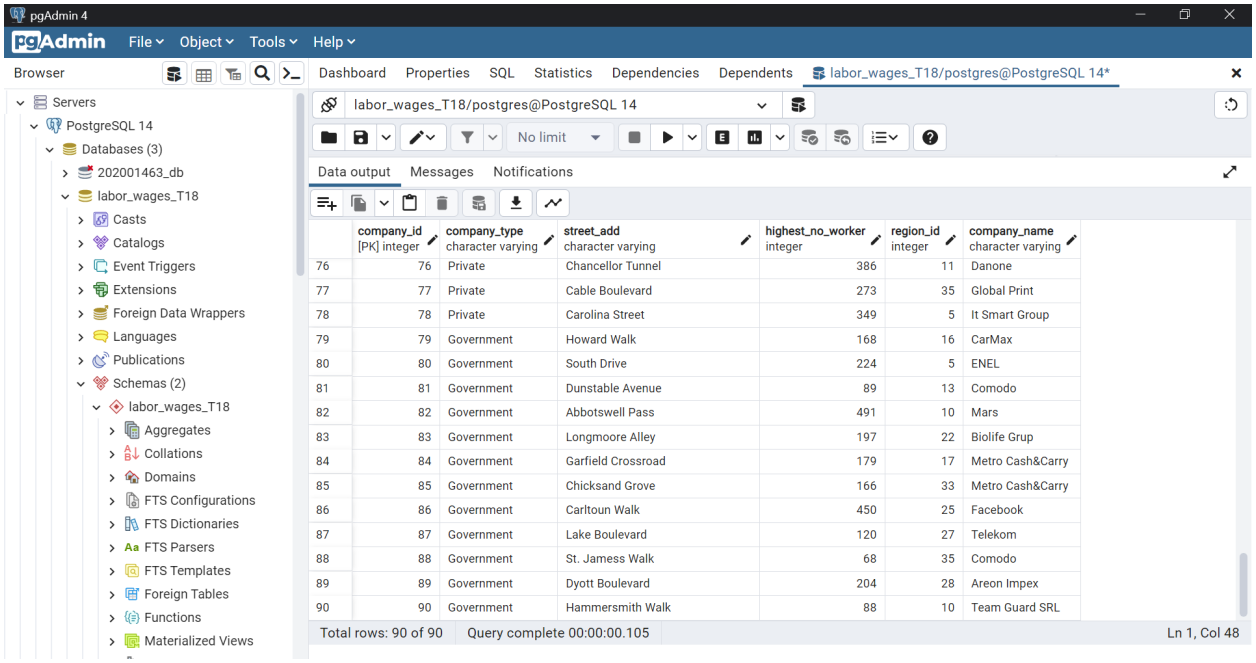
Company details

Add New Company

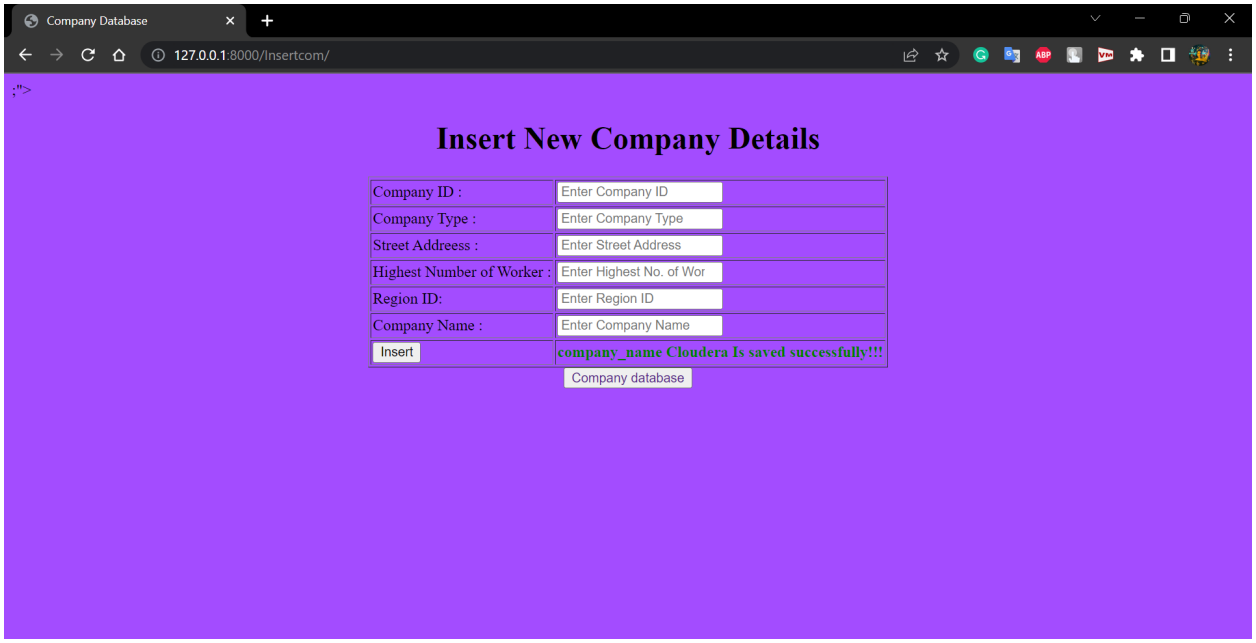
Company ID	Company Type	Street Address	Highest No. Worker	Region ID	Company Name		
1	Government	Tilloch Route	361	26	AECOM	Edit	Delete
2	Government	Canning Boulevard	236	37	Comodo	Edit	Delete
3	Private	Beechen Alley	444	30	Leadertech Consulting	Edit	Delete
4	Private	Camberwell Grove	409	2	Amazon.com	Edit	Delete
5	Government	Coalecroft Street	314	29	Carrefour	Edit	Delete
6	Government	Hamilton Tunnel	298	23	Vodafone	Edit	Delete
7	Government	Sundown Rue	409	14	BuzzFeed	Edit	Delete
8	Government	Maple Hill	453	23	Coca-Cola Company	Edit	Delete
9	Government	Ellerslie Grove	114	8	Apple Inc.	Edit	Delete
10	Government	Badric Tunnel	245	34	21st Century Fox	Edit	Delete
11	Government	Hamilton Lane	159	7	Mars	Edit	Delete
12	Government	Cloth Lane	441	30	ENEL	Edit	Delete
13	Government	Kimberley Crossroad	294	12	Coca-Cola Company	Edit	Delete
14	Government	Bective Crossroad	143	34	Telekom	Edit	Delete
15	Government	Sherlock Road	462	15	Erickson	Edit	Delete
16	Government	Chesterfield Grove	258	21	It Smart Group	Edit	Delete
17	Private	Cam Crossroad	430	5	Leadertech Consulting	Edit	Delete
18	Private	College Boulevard	490	12	Global Print	Edit	Delete
19	Private	Under Drive	159	4	Amazon.com	Edit	Delete
20	Private	Pine Boulevard	151	23	Comodo	Edit	Delete
21	Private	Mariner Way	314	37	Comodo	Edit	Delete
22	Private	Charnwood Route	367	22	Coca-Cola Company	Edit	Delete
23	Private	Bayberry Walk	174	4	Coca-Cola Company	Edit	Delete
24	Private	Chesterfield Walk	384	17	Mars	Edit	Delete
25	Government	Marina Way	289	3	ENEL	Edit	Delete
26	Government	Balfour Rue	405	6	21st Century Fox	Edit	Delete
27	Government	Clyde Lane	157	39	Amazon.com	Edit	Delete
28	Government	Camera Alley	487	4	Biolife Grup	Edit	Delete
29	Private	Berriman Drive	193	34	BuzzFeed	Edit	Delete
30	Private	North Grove	443	11	Vodafone	Edit	Delete
31	Private	Chester Boulevard	475	35	DynCorp	Edit	Delete
32	Private	Adelaide Route	308	14	Leadertech Consulting	Edit	Delete
33	Private	Chestnut Rue	442	34	Biolife Grup	Edit	Delete
34	Private	Gate Way	490	17	Areon Impex	Edit	Delete
35	Private	Bayberry Lane	410	21	It Smart Group	Edit	Delete
36	Private	Blore Tunnel	363	30	Leadertech Consulting	Edit	Delete
37	Private	Callcott Lane	169	26	Zepter	Edit	Delete
38	Private	Bayberry Walk	238	9	Mars	Edit	Delete
39	Private	East Vale	427	8	Amazon.com	Edit	Delete
40	Government	Gautrey Road	218	20	Metro Cash&Carry	Edit	Delete
41	Government	English Crossroad	138	31	Global Print	Edit	Delete

47	Government	Coborn Pass	262	5	DynCorp	Edit	Delete
48	Private	Endsleigh Vale	150	6	Mars	Edit	Delete
49	Private	Longman Walk	100	8	ENEL	Edit	Delete
50	Private	Tiptree Lane	370	7	DynCorp	Edit	Delete
51	Private	Timber Boulevard	89	25	Coca-Cola Company	Edit	Delete
52	Private	Meadow Alley	308	3	Facebook	Edit	Delete
53	Private	Longman Lane	244	29	Leadertech Consulting	Edit	Delete
54	Private	Bempton Crossroad	99	6	Comcast	Edit	Delete
55	Private	Jackson Road	51	14	Demaco	Edit	Delete
56	Private	Carpenter Lane	470	2	It Smart Group	Edit	Delete
57	Private	Beaumont Grove	101	31	Amazon.com	Edit	Delete
58	Private	Cobden Way	92	38	AECOM	Edit	Delete
59	Private	Buttonwood Road	212	28	Apple Inc.	Edit	Delete
60	Private	Bletchley Crossroad	463	24	Vodafone	Edit	Delete
61	Private	Monroe Crossroad	272	15	Amazon.com	Edit	Delete
62	Private	Chestnut Rise Rue	393	36	ENEL	Edit	Delete
63	Private	Falconberg Road	462	17	UPC	Edit	Delete
64	Government	Baynes Grove	286	19	Telekom	Edit	Delete
65	Government	Armory Grove	300	38	Zepter	Edit	Delete
66	Government	Victorian Pass	142	6	Apple Inc.	Edit	Delete
67	Government	Balham Tunnel	378	13	Mars	Edit	Delete
68	Government	Camden Walk	317	9	DynCorp	Edit	Delete
69	Government	Bolingbroke Lane	203	7	Team Guard SRL	Edit	Delete
70	Government	Archery Drive	152	37	AECOM	Edit	Delete
71	Government	Chestnut Rise Tunnel	469	17	Erickson	Edit	Delete
72	Private	Cliff Pass	359	34	Vodafone	Edit	Delete
73	Private	Maple Alley	373	27	Zepter	Edit	Delete
74	Private	Canal Way	328	10	BuzzFeed	Edit	Delete
75	Private	King Grove	288	18	Facebook	Edit	Delete
76	Private	Chancellor Tunnel	386	11	Danone	Edit	Delete
77	Private	Cable Boulevard	273	35	Global Print	Edit	Delete
78	Private	Carolina Street	349	5	It Smart Group	Edit	Delete
79	Government	Howard Walk	168	16	CarMax	Edit	Delete
80	Government	South Drive	224	5	ENEL	Edit	Delete
81	Government	Dunstable Avenue	89	13	Comodo	Edit	Delete
82	Government	Abbotswell Pass	491	10	Mars	Edit	Delete
83	Government	Longmoore Alley	197	22	Biolife Grup	Edit	Delete
84	Government	Garfield Crossroad	179	17	Metro Cash&Carry	Edit	Delete
85	Government	Chicksand Grove	166	33	Metro Cash&Carry	Edit	Delete
86	Government	Carltoun Walk	450	25	Facebook	Edit	Delete
87	Government	Lake Boulevard	120	27	Telekom	Edit	Delete
88	Government	St. Jamess Walk	68	35	Comodo	Edit	Delete
89	Government	Dyott Boulevard	204	28	Areon Impex	Edit	Delete
90	Government	Hammersmith Walk	88	10	Team Guard SRL	Edit	Delete

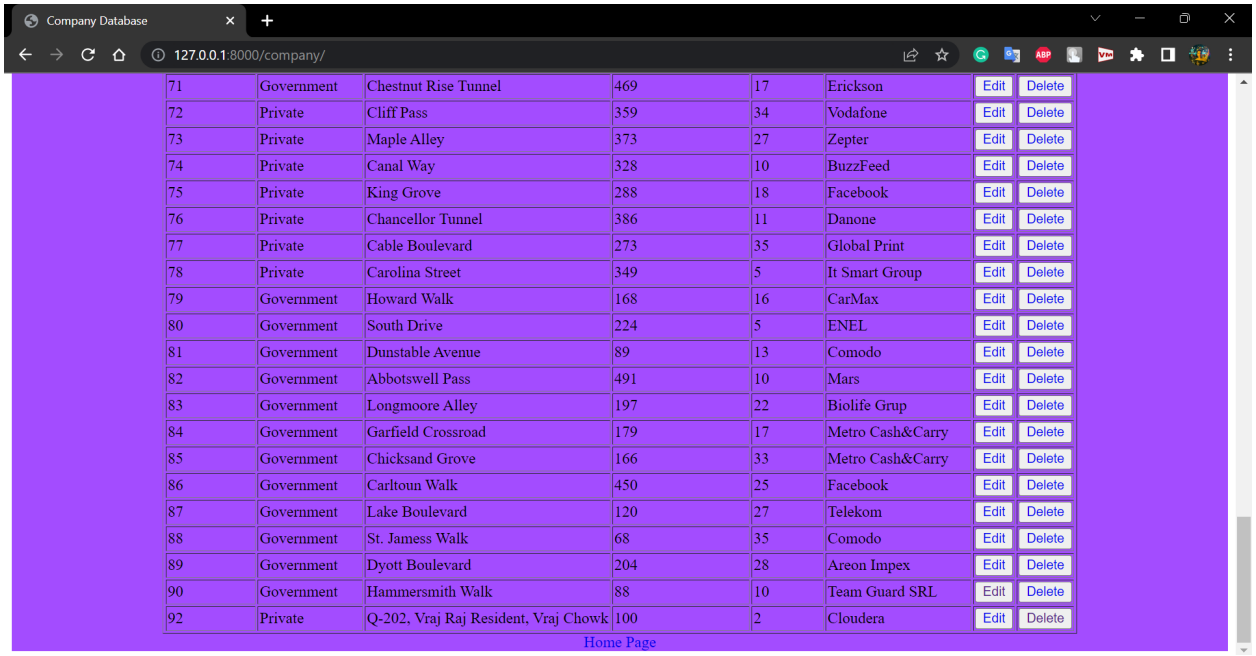
Pg admin company table



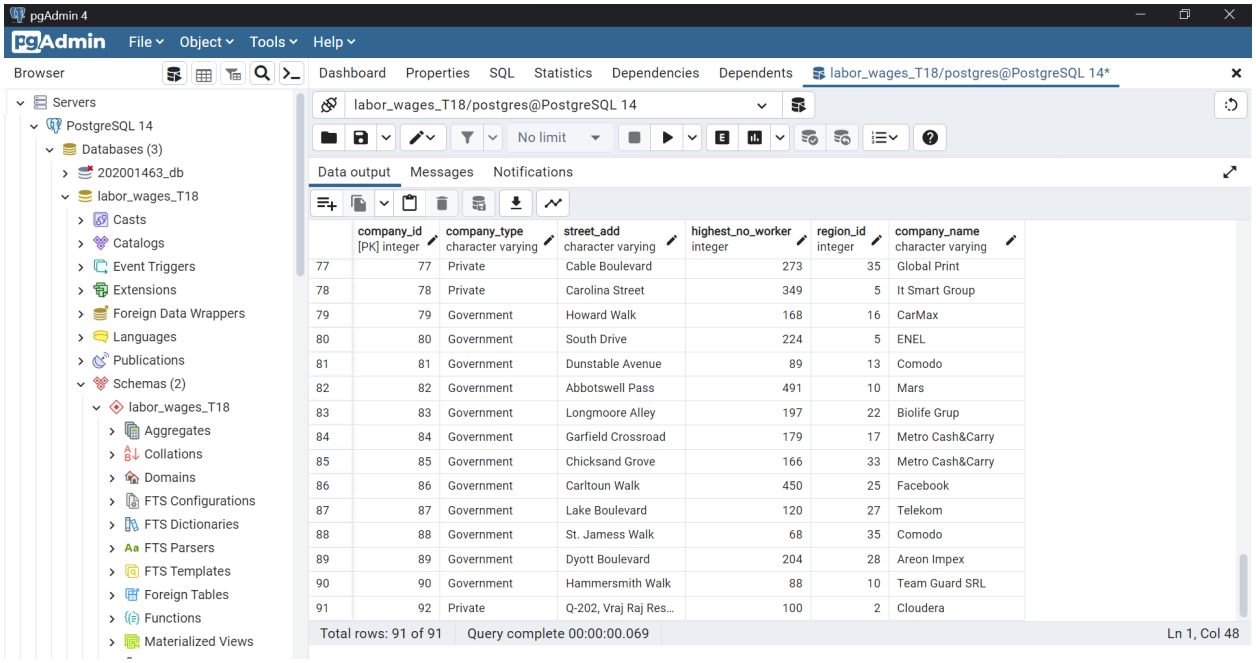
Now Add the New company detail



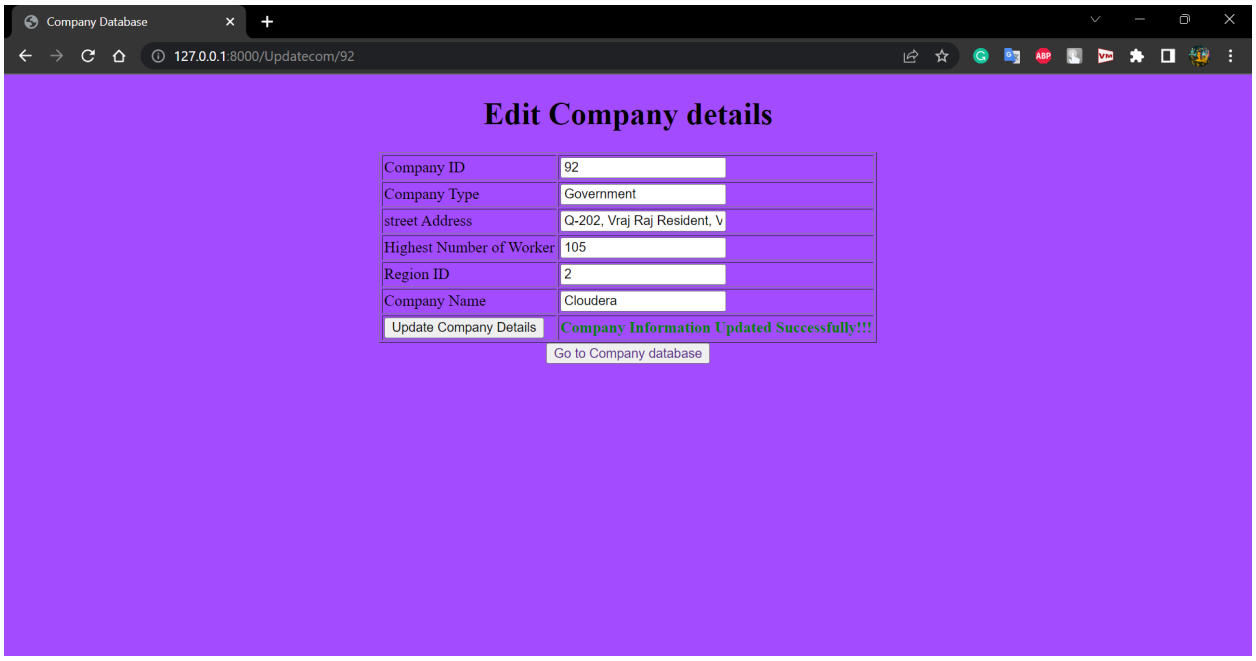
New table company after add new company details



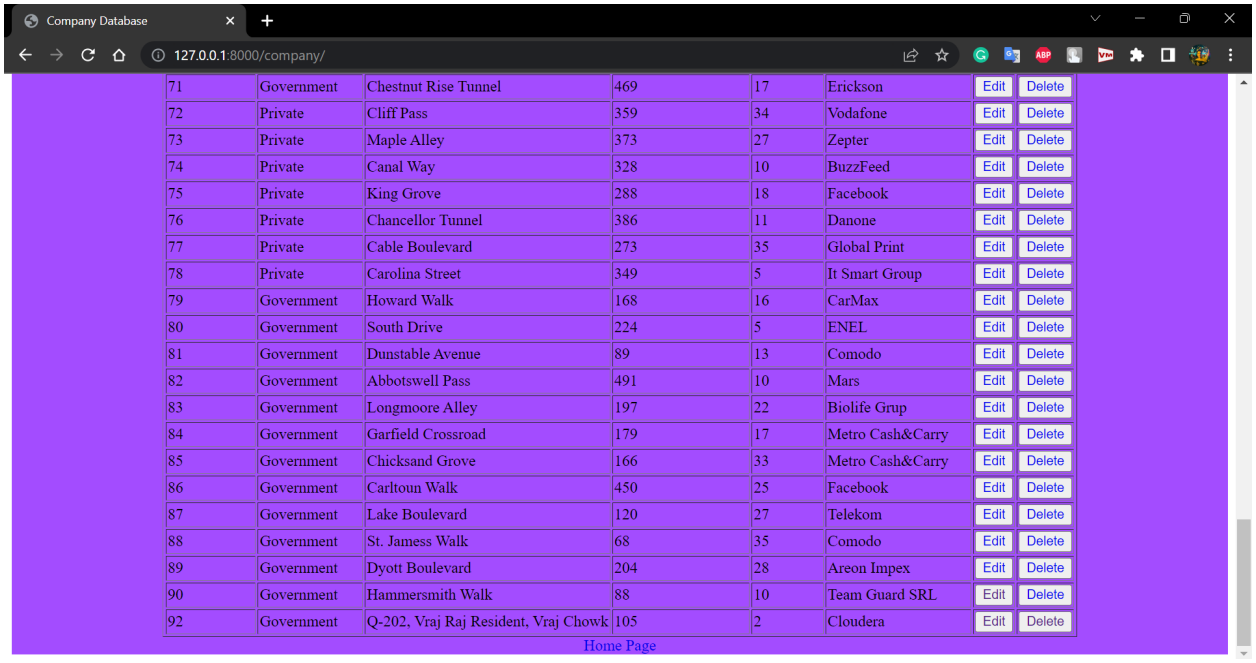
Pg admin company details table



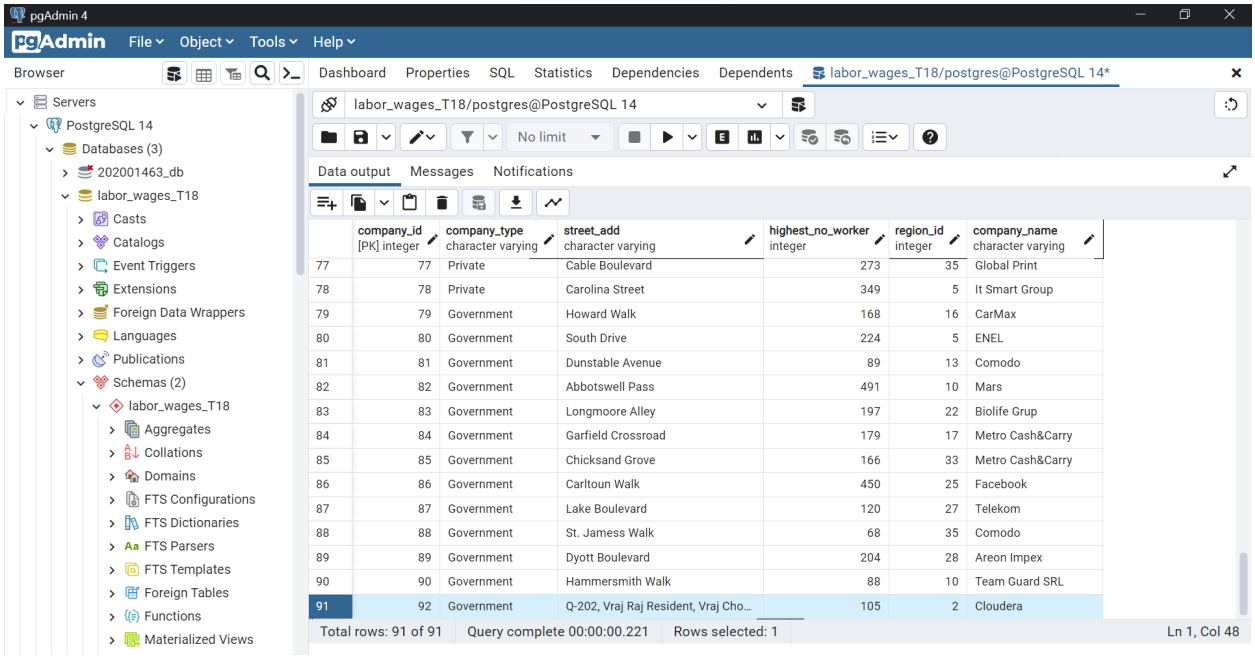
Now edit the company detail of company id 92.



Company table after edit.

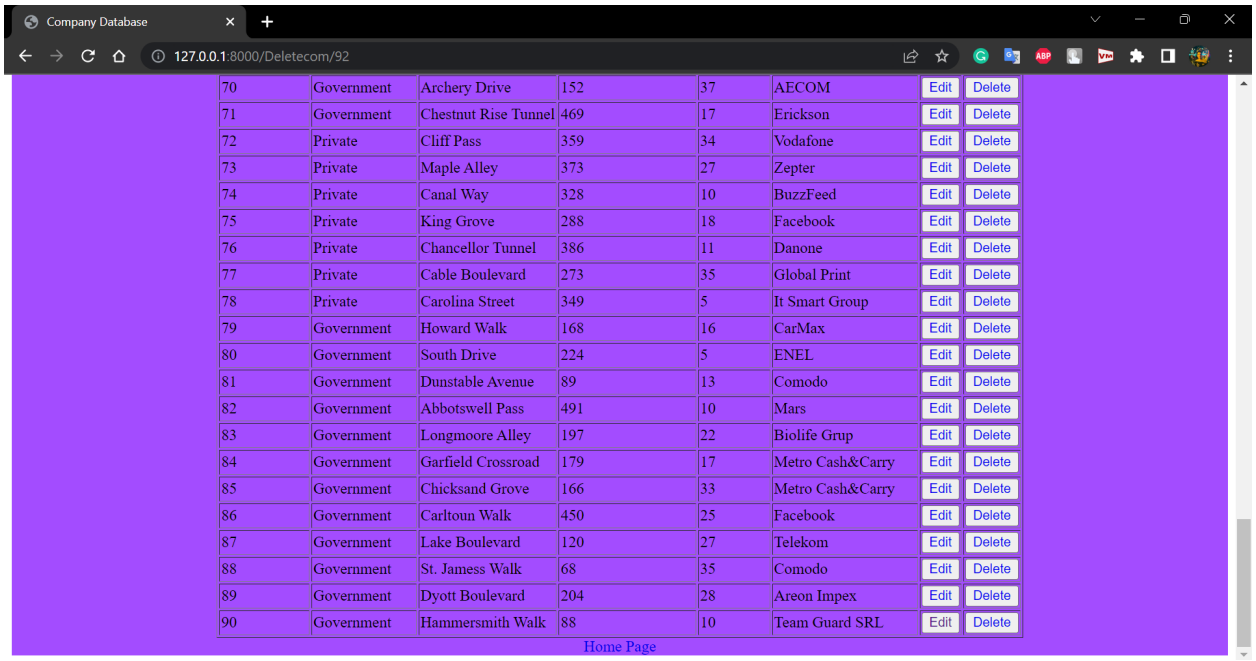


Pg admin table after edit.

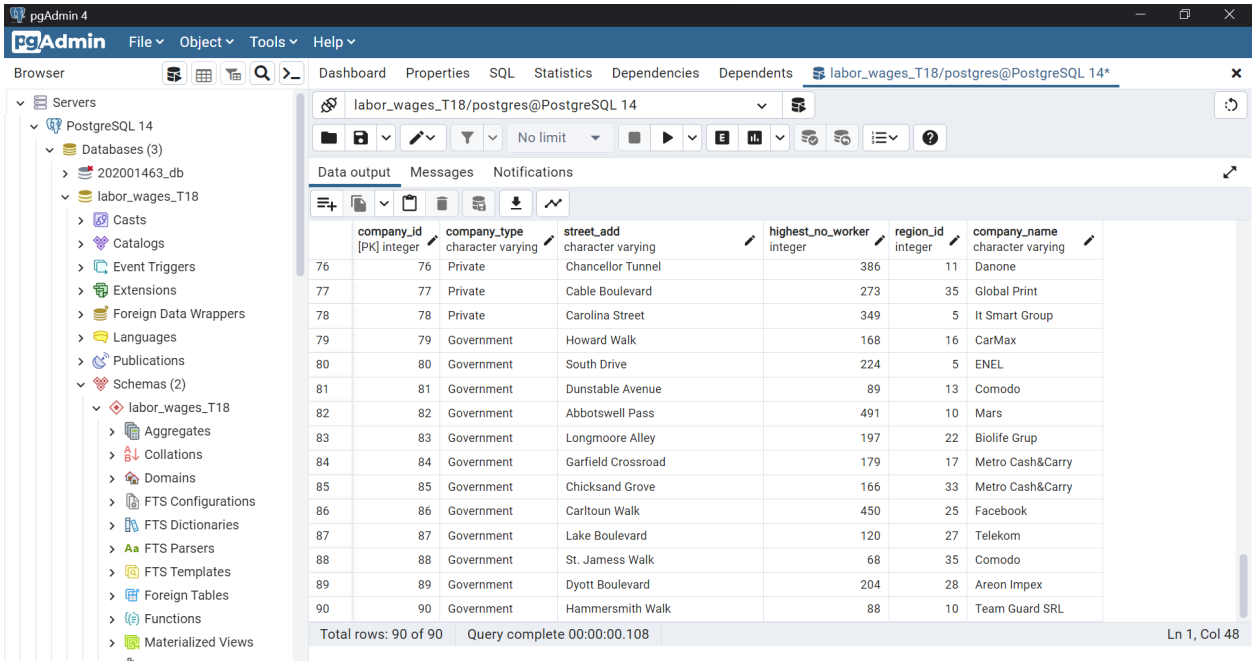


Now delete the data of company id 92.

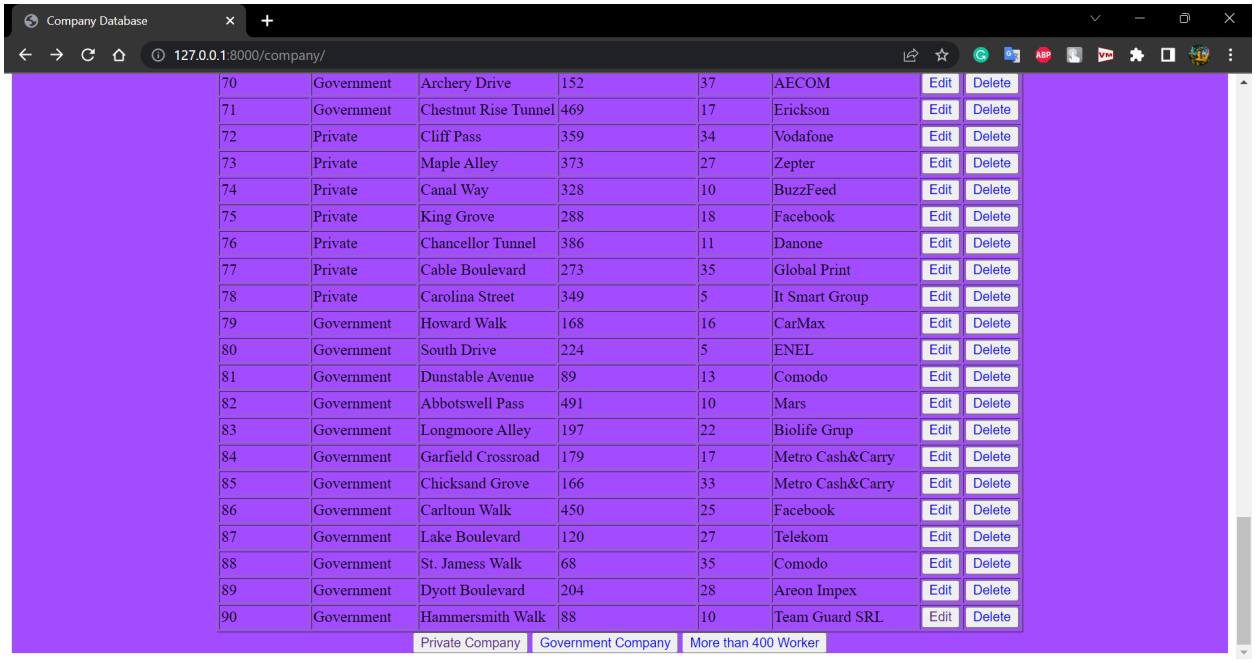
Company Table after delete.



Pg admin table after delete.



Query Buttons



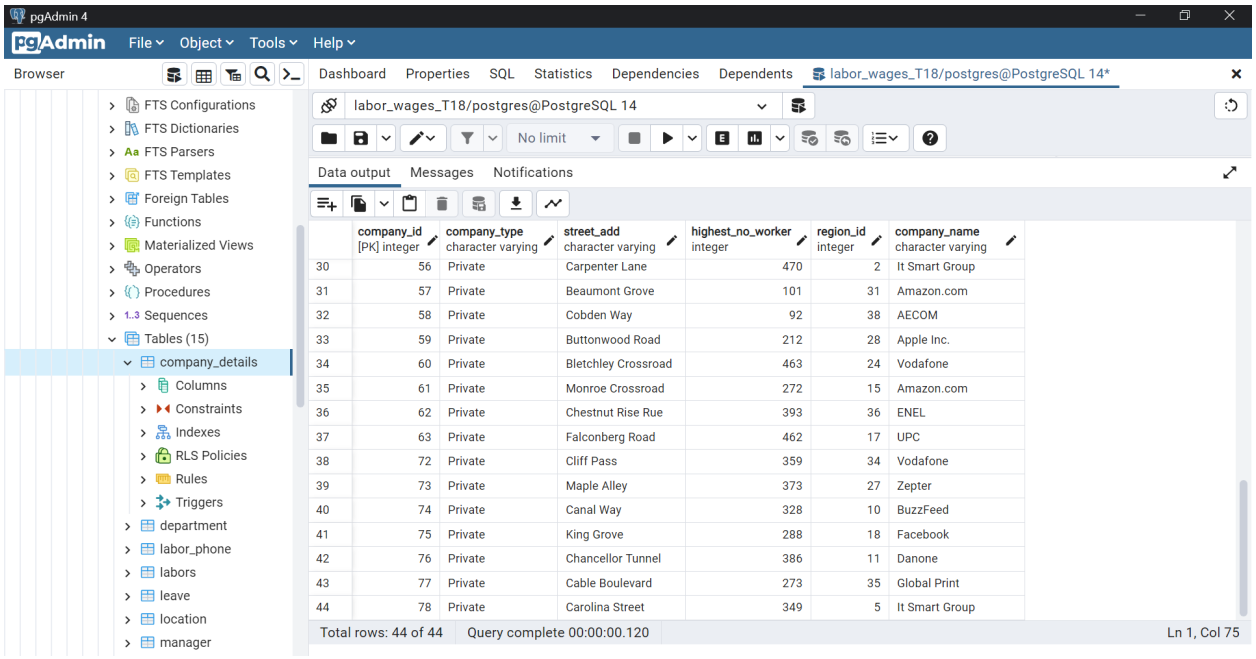
If we want to private company

Private Company details

Company ID	Company Type	Street Address	Highest No. Worker	Region ID	Company Name
3	Private	Beechen Alley	444	30	Leadertech Consulting
4	Private	Camberwell Grove	409	2	Amazon.com
17	Private	Cam Crossroad	430	5	Leadertech Consulting
18	Private	College Boulevard	490	12	Global Print
19	Private	Under Drive	159	4	Amazon.com
20	Private	Pine Boulevard	151	23	Comodo
21	Private	Mariner Way	314	37	Comodo
22	Private	Charnwood Route	367	22	Coca-Cola Company
23	Private	Bayberry Walk	174	4	Coca-Cola Company
24	Private	Chesterfield Walk	384	17	Mars
29	Private	Berriman Drive	193	34	BuzzFeed
30	Private	North Grove	443	11	Vodafone
31	Private	Chester Boulevard	475	35	DynCorp
32	Private	Adelaide Route	308	14	Leadertech Consulting
33	Private	Chestnut Rue	442	34	Biolife Grup
34	Private	Gate Way	490	17	Areon Impex
35	Private	Bayberry Lane	410	21	It Smart Group
36	Private	Blore Tunnel	363	30	Leadertech Consulting
37	Private	Callcott Lane	169	26	Zepter
38	Private	Bayberry Walk	238	9	Mars
39	Private	East Vale	427	8	Amazon.com
48	Private	Endsleigh Vale	150	6	Mars
49	Private	Longman Walk	100	8	ENEL
50	Private	Tiptree Lane	370	7	DynCorp
51	Private	Timber Boulevard	89	25	Coca-Cola Company
52	Private	Meadow Alley	308	3	Facebook
53	Private	Longman Lane	244	29	Leadertech Consulting
54	Private	Bempton Crossroad	99	6	Comcast
55	Private	Jackson Road	51	14	Demaco
56	Private	Carpenter Lane	470	2	It Smart Group
57	Private	Beaumont Grove	101	31	Amazon.com
58	Private	Cobden Way	92	38	AECOM
59	Private	Buttonwood Road	212	28	Apple Inc.
60	Private	Bletchley Crossroad	463	24	Vodafone
61	Private	Monroe Crossroad	272	15	Amazon.com
62	Private	Chestnut Rise Rue	393	36	ENEL
63	Private	Falconberg Road	462	17	UPC
72	Private	Cliff Pass	359	34	Vodafone
73	Private	Maple Alley	373	27	Zepter
74	Private	Canal Way	328	10	BuzzFeed
75	Private	King Grove	288	18	Facebook
76	Private	Chancellor Tunnel	386	11	Danone
77	Private	Cable Boulevard	273	35	Global Print
78	Private	Carolina Street	349	5	It Smart Group

Company database

Pg admin table for private company



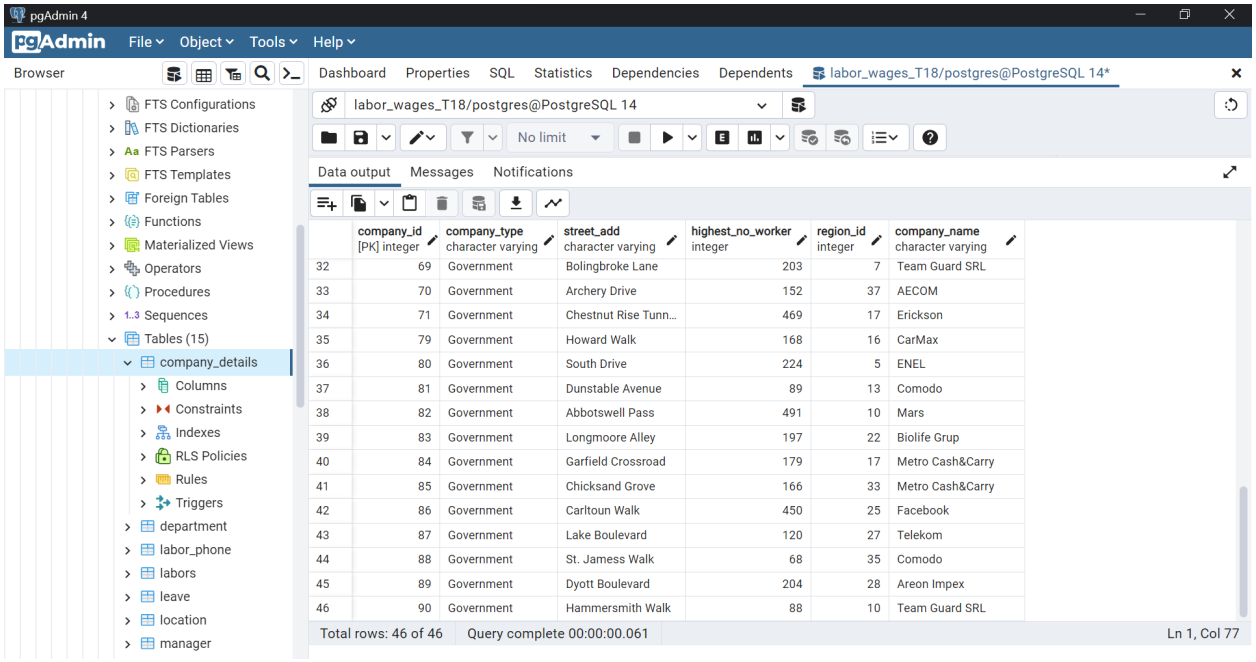
If we want to Government Company

Government Company details

Company ID	Company Type	Street Address	Highest No. Worker	Region ID	Company Name
1	Government	Tilloch Route	361	26	AECOM
2	Government	Canning Boulevard	236	37	Comodo
5	Government	Coalecroft Street	314	29	Carrefour
6	Government	Hamilton Tunnel	298	23	Vodafone
7	Government	Sundown Rue	409	14	BuzzFeed
8	Government	Maple Hill	453	23	Coca-Cola Company
9	Government	Ellerslie Grove	114	8	Apple Inc.
10	Government	Badric Tunnel	245	34	21st Century Fox
11	Government	Hamilton Lane	159	7	Mars
12	Government	Cloth Lane	441	30	ENEL
13	Government	Kimberley Crossroad	294	12	Coca-Cola Company
14	Government	Bective Crossroad	143	34	Telekom
15	Government	Sherlock Road	462	15	Erickson
16	Government	Chesterfield Grove	258	21	It Smart Group
25	Government	Marina Way	289	3	ENEL
26	Government	Balfour Rue	405	6	21st Century Fox
27	Government	Clyde Lane	157	39	Amazon.com
28	Government	Camera Alley	487	4	Biolife Grup
40	Government	Gautrey Road	218	20	Metro Cash&Carry
41	Government	English Crossroad	138	31	Global Print
42	Government	Western Route	117	11	BuzzFeed
43	Government	Boadicea Rue	460	29	Amazon.com
44	Government	Oxford Tunnel	112	24	It Smart Group
45	Government	Boleyn Crossroad	205	4	21st Century Fox
46	Government	Central Grove	272	14	Danone
47	Government	Coborn Pass	262	5	DynCorp
64	Government	Baynes Grove	286	19	Telekom
65	Government	Armory Grove	300	38	Zepter
66	Government	Victorian Pass	142	6	Apple Inc.
67	Government	Balham Tunnel	378	13	Mars
68	Government	Camden Walk	317	9	DynCorp
69	Government	Bolingbroke Lane	203	7	Team Guard SRL
70	Government	Archery Drive	152	37	AECOM
71	Government	Chestnut Rise Tunnel	469	17	Erickson
79	Government	Howard Walk	168	16	CarMax
80	Government	South Drive	224	5	ENEL
81	Government	Dunstable Avenue	89	13	Comodo
82	Government	Abbotswell Pass	491	10	Mars
83	Government	Longmoore Alley	197	22	Biolife Grup
84	Government	Garfield Crossroad	179	17	Metro Cash&Carry
85	Government	Chicksand Grove	166	33	Metro Cash&Carry
86	Government	Carltoun Walk	450	25	Facebook
87	Government	Lake Boulevard	120	27	Telekom
88	Government	St. Jamess Walk	68	35	Comodo
89	Government	Dyott Boulevard	204	28	Areon Impex
90	Government	Hammersmith Walk	88	10	Team Guard SRL

Company database

Pg admin for table Government company



	company_id [PK] integer	company_type character varying	street_add character varying	highest_no_worker integer	region_id integer	company_name character varying
32	69	Government	Bolingbroke Lane	203	7	Team Guard SRL
33	70	Government	Archery Drive	152	37	AECOM
34	71	Government	Chestnut Rise Tunn...	469	17	Erickson
35	79	Government	Howard Walk	168	16	CarMax
36	80	Government	South Drive	224	5	ENEL
37	81	Government	Dunstable Avenue	89	13	Comodo
38	82	Government	Abbotswell Pass	491	10	Mars
39	83	Government	Longmoore Alley	197	22	Biolife Grup
40	84	Government	Garfield Crossroad	179	17	Metro Cash&Carry
41	85	Government	Chicksand Grove	166	33	Metro Cash&Carry
42	86	Government	Carltoun Walk	450	25	Facebook
43	87	Government	Lake Boulevard	120	27	Telekom
44	88	Government	St. Jamess Walk	68	35	Comodo
45	89	Government	Dyott Boulevard	204	28	Areon Impex
46	90	Government	Hammersmith Walk	88	10	Team Guard SRL

If we want to company more than 400.

More than 400 Worker Company Details

Company ID	Company Type	Street Address	Highest No. Worker	Region ID	Company Name
3	Private	Beechen Alley	444	30	Leadertech Consulting
4	Private	Camberwell Grove	409	2	Amazon.com
7	Government	Sundown Rue	409	14	BuzzFeed
8	Government	Maple Hill	453	23	Coca-Cola Company
12	Government	Cloth Lane	441	30	ENEL
15	Government	Sherlock Road	462	15	Erickson
17	Private	Cam Crossroad	430	5	Leadertech Consulting
18	Private	College Boulevard	490	12	Global Print
26	Government	Balfour Rue	405	6	21st Century Fox
28	Government	Camera Alley	487	4	Biolife Grup
30	Private	North Grove	443	11	Vodafone
31	Private	Chester Boulevard	475	35	DynCorp
33	Private	Chestnut Rue	442	34	Biolife Grup
34	Private	Gate Way	490	17	Areon Impex
35	Private	Bayberry Lane	410	21	It Smart Group
39	Private	East Vale	427	8	Amazon.com
43	Government	Boadicea Rue	460	29	Amazon.com
56	Private	Carpenter Lane	470	2	It Smart Group
60	Private	Bletchley Crossroad	463	24	Vodafone
63	Private	Falconberg Road	462	17	UPC
71	Government	Chestnut Rise Tunnel	469	17	Erickson
82	Government	Abbotswell Pass	491	10	Mars
86	Government	Carltoun Walk	450	25	Facebook

Company database

Pg admin table for highest number of worker is more than 400.