



Dhirubhai Ambani  
Institute of Information and Communication Technology

## **“HIGHWAY CONSTRUCTION”**

**TEAM 8.**

**SECTION 8.**

**GROUP 3.**

<b><u>Name</u></b>	<b><u>ID</u></b>
<b>Harsh Acharya</b>	<b>202001131</b>
<b>Sarthak Sonagara</b>	<b>202001132</b>

# Index

1. Section1: Final version of SRS.	03
2. Section2: Noun Analysis and ERD	34
3. Section3: Normalization and Schema Refinement.	46
4. Section4: Final DDL Scripts and Queries.	54
5. Section5: Front-End Development	116

# **Section1: Final version of SRS.**

## **1. INTRODUCTION**

- 1.1. Purpose**
- 1.2. Intended audience and reading suggestion**
- 1.3. Product Scope**
- 1.4. Description**

## **2. FACT FINDING PHASE**

- 2.1. Background Reading**
    - 2.1.1. Description of reading**
    - 2.1.2. References**
    - 2.1.3. Combined requirement**
  - 2.2. Interview**
  - 2.3. Questionnaire**
  - 2.4. Observations**
- 3. Fact finding chart**
  - 4. List requirements**
  - 5. User categories and privileges**
  - 6. Assumption**
  - 7. Business constraints**

# **1. INTRODUCTION**

## *1.1. Purpose*

The purpose of this document is to present a detailed description of the highway construction. We are going to make a database which will cover all the details related to the topic to manage and to get the data.

As the number of highways increase overtime, the volume of registration and manual process of recording and updating each record is becoming tremendously tedious. The database will maximize work efficiency and it's easy to understand and use. It will provide easy, accurate and timely retrieval of Highway construction information.

It will store and provide details about

- Highway construction
- Keep track about maintenance updation and renovation thing
- The length, construction cost, boundaries between states ,construction companies, civil engineers and much more.
- Keep track of problems that arise after construction and whether it's resolved or not.
- And many details as described later in the document.

Storing these details, It will help us to efficiently find records about any highway and even after many years, we can check it and also if anything update is there then we can easily update new details into it.

## *1.2. Intended Audience and Reading Suggestions*

### For Learning purposes:

This document is intended for all the database management system students, faculties or the professional. The document describes the detailed case study about the topic, any computer domain student can refer to this document to learn, to suggest or to study examples of the design part of any software, or Database to be more precise. It can also be helpful to data scientists or researchers in some sense.

### For using purposes:

This is an SRS(System Requirement Specification) document to design

the database of “Highway Construction, Development and Maintenance”. Any user of this database can refer to this document to understand the structure and approaches that were used for making this database. Users like Government officials, Construction companies, Project Managers and many more can use this, even specific departments of government and companies and the public can use some part of the database and this document to gather corresponding information. There is a large number of users related to this database, eventually introduced and they can also retrieve the insights of the database.

**For technical purposes:**

Anyone who is part of designing the database should definitely read this document. Apart from designers, maintainers, programmers, database manipulators and all the lower level implementation related members must refer to this to understand the working and the structure of the database.

**Reading suggestions:**

The document has all the information about this database, different readers can use information accordingly. Product scope right after this section will be useful for using purposes. For technical purposes the “Description” section is provided. Users may find it convenient to read the document as it is. Users might skip some sections if they find them irrelevant.

### *1.3. Product Scope*

We are going to create a database for considerably large data which is real-life examples. To create a database we will use PostgreSQL software mainly. We will use some supplement software to design, represent and to provide results.

The objective is to store and efficiently provide each and every information related to the “Highway construction”. Our goal is to create such a “Useful” database, that provides most of the functionalities related to the topic in a professional designing manner. There are plenty of benefits associated with the creation of a “Good” database. It will be helpful to maintain and to get future insights for

any highway development related matter. It will efficiently manage all the necessary data and will relate information from different domains. The complexity to manage data will be reduced thoroughly.

The database is really helpful for corporate companies which are Civil-Construction Companies, Material providers, Governments. The database will provide ease of data management at a higher level. The data stored in the database will be useful for future projects and to analyze the statistics related to the business.

#### *1.4. Description*

##### Workflow:

The Highways are used for inter-state movements of goods and passengers.

After the state government's request and central government's approval, a tender can be released for a project of highway construction to the contract companies. On the basis of companies overall performance like cost, quality, time etc. one of the tenders is selected and the highway construction project starts.

After construction of a highway the maintenance of the highway is also assigned to some other organization. Some complaints, some events, some problems, some repairs can happen by the time and we should be able to manage them all. After many years, a highway might be renovated also.

##### Planning:

The government plays a major role here. At the very first, when the government decides to make or renovate highways. The decision gets passed and the whole process initiates. Government first passes the tender for the highway, after gathering tenders, it selects one of them from the data received from many companies. There are many departments in the government. NHAI - National Highway Authority of India. Different employees of this

department can decide to add a new highway by considering certain criteria. Government can also give ranks to the companies for awards. They assign various projects. They also solve problems related to construction or maintenance.

Information of all the contract companies and suppliers should be there to contact them for a new highway construction or to maintenance of an existing highway. Companies can also have data of their past work.

Highway itself has many properties like length, cities in which it falls, number of lanes, type of vehicle, etc. After construction of highway, there are tolls so toll has many properties toll number, spending for manage highway then collection of particular number toll. Highway pass through different states so state has define by its name property. On highway different vehicles are passes. So for allow different vehicles, vehicle has characteristics like type, weight and height.

For these purposes the government has to keep track of pending work and completed work so that it can work on pending projects by looking at past experiences. Government is include int select tender for construct particular highway then it also gives rank to companies as per its work and after looking other aspects. Governments should keep track of problems rises on different highways so solve it is a task in which government play as a role.

After construction, There is one completed project which contains information about completed project number, its start date, end date, cost and description. Then one major thing is highway maintenance. For it, the first issue / problem arises on the highway, then the government first looks at the project number and contacts its construction company so there is one responsible relation. In a project, the responsible relationship is with the person. Person can be any organization. Person has properties like organization name, id, Contact details like email, phone, etc.

Budget is one of the primary things. The work must be done in a provided fund. The government can be asked to reveal the budget of highway construction and maintenance or renovation projects. Tender details and collection of money from those highways are necessary. Different companies give their tender which contains information about cost, description, quality of material, expected time, etc. Then the selected company for the particular highway will construct the highway and complete the project so the project and company are also related to each other. Project has many properties like its number, type , current status, start time , expected end time, expected cost.

In short, highway authorities first check if a highway is needed, then take a project in hand, construct it, maintain it, keep track of problems in it, keep track of cost involved in all these things. For doing this work it must communicate with other organizations to be successfully done.

## 2. FACT FINDING PHASE

### 2.1. *Background Readings*

#### 2.1.1. *Description of reading*

- Road and highway projects database

This is a research paper by a company named “CRISIL”. This paper is on “Road and highway projects database”. This paper defines some methods that are used to award the construction companies of road and highways for their construction on the basis of given criteria.

From this pdf we got to know some more Functions that we can provide from the database. We also got some knowledge about adding more attributes.

For example we got the idea to add the Toll entity to have the information about toll collection of the highways.

We added expected end date, actual completion date, number of lanes attributes by referring to this research pdf index.

- Official website of Ministry of Road Transport and Highways

This is an official government website for road and highway construction information. It has many sections and real data of projects taken for the highway construction.

From this website we got familiar with the real scenarios. We understood about highway types and how authority declares highways based on the connectivity required. We were introduced to some boards and departments which make different decisions. We observed some projects like “Bharatmala”, “Vijaywada Ranchi”, “Char Dham Pariyojna”, etc. We added a “ranking” entity by studying this website. Government awards construction companies on various parameters.

- Spreadsheet on official website

We examined some tables and data fields from the sheets provided in the link below. This spreadsheet contains some more detailed attributes for ranking of companies and measures of their rank.

- Corporate Index Number

We got familiar with CIN numbers after reading the article from the website which is given below. A Corporate Identification Number (CIN) is a unique identification number that is assigned by the Registrar of Companies (ROC) to the companies registered in India. The ROC gives the CIN to the companies while issuing their Registration Certificate.

#### *2.1.2. References*

- [https://www.crisil.com/content/dam/crisil/our-businesses/quantix/infras\\_tructure-project-database/road-and-highway-projects-database-table-of\\_-contents.pdf](https://www.crisil.com/content/dam/crisil/our-businesses/quantix/infras_tructure-project-database/road-and-highway-projects-database-table-of_-contents.pdf)
- <https://morth.nic.in/about-highways>
- <https://morth.nic.in/sites/default/files/NHEA%202020%20Winners%20List.pdf>
- [https://morth.nic.in/suggestive-format-collection-and-reporting-data-us\\_ing-network-survey-vehicle-nsv-guidance-purpose](https://morth.nic.in/suggestive-format-collection-and-reporting-data-us_ing-network-survey-vehicle-nsv-guidance-purpose)
- <https://cleartax.in/s/cin-corporate-identification-number#:~:text=A%20Corporate%20Identification%20Number%20>

#### *2.1.3. Combined Requirements*

From background reading we got the basic idea that how will we start making required entities. We also got some attributes that we can add into it. In short, background reading helped to make the main entity highway and its attributes to start. Like highway construction can pass through many states, it has length, width, and similar common attributes, it has expected and actual completion dates, and the cost type of things associated with it. We also added some unique entities like tends and ranking because that is required as per our background reading. So, collectively we gathered many requirements that our database must fulfill from the background reading.

#### *2.2. Interviews*

## **1) Interview with construction company manager**

### Interview Plan:

Interviewee: 1) Shantilal Mehta (Role Play)

Designation: Construction Company Manager

Interviewer: 1) Harsh Acharya

Designation: Business Development Executive

2) Sarthak Sonagara

Designation: Developer

Date: 27/09/2022

Time: 11:30

Duration: 45 minutes

Place: Google meet

### Purpose of Interview:

Preliminary meeting to identify entire process, problems, solutions and requirements regarding highway construction and also how this tender cost value and quality calculate , which characteristics are more focused by government and how you try to fulfill it. Number of tenders selected by government of your company and prediction of number of tolls and collection of tolls.

### Agenda:

- Introduction and current status
- Initial ideas and about blue print then follow-up action to reduce cost and provide more quality as per requirements then finalize cost of particular highway.
- About the needs and complaints of the citizens and how these issues are handled.

### Documents to be brought to the interview:

- Any documents relating to the current highways in India, their length in particular state, highway number and state, highway connecting to which cities and states list and population around particular highway.

### Questions:

- 1) What is the current process of manufacturing the highway?
- 2) Are there any difficulties with the current system? If any, What difficulties are you encountering with the current interface?
- 3) Please comment if any new functionalities we can add to our database system, that will be helpful for you.

### Results of the interview:

- Currently, the government releases the tender of a particular highway and the deadline for submitting the tender value with detailed description of cost , quality ,etc.
- As per Current situation, Companies do not easily find records of highways like previously this highway when it was built ,how much cost and which company built it and when was it renovated.so it would be great if more transparency in communication between construction company and government.
- It would be great if provide easily accessible data , find old records easily, provide platform to directly communication with government , feedback about tender and also provide second chance to submit improve version of tender.

## **2) Interview with NHAI Government servant**

### Interview Plan:

Interviewee: 1) Karan Patel (Role Play)

Designation: Government servant at ministry of road transport and highways, India.

Interviewer: 1) Harsh Acharya

Designation: Business Development Executive

2) Sarthak Sonagara

Designation: Developer

Date: 28/09/2022

Time: 11:30

Duration: 1 hour

Place: Google meet

### Purpose of Interview:

To get the details of the entire process from the government point of view. At the government side which type of steps will be taken for this entire process of highway construction and upto its management.

### Agenda:

- Introduction and current status
- Get the details about how they make decisions about new highway criteria for renovation.
- Get the details about which type of problem arises then as a solution new construction do or just do the other solution.

### Documents to be brought to the interview:

- Any documents relating to the current highways in India, their length in particular state, highway number and state, highway connecting to which cities and states list and population around particular highway.

### Questions:

- 1) How do you decide to build a new highway?
- 2) What parameters do you take into account from the citizens' point of view?
- 3) Approximately how much time and cost it takes for every new construction and how do you manage the budget?
- 4) Can you tell us some functionalities that you want to use in the new system?

### Results of the interview:

- Mostly traffic jam issues, transportation increase, need connection between two cities, states and many other aspects for building a new highway.
- Mainly the government keeps watch on each and every constructed highway and also on normal roads. If they need a new highway or need reparation of the highway then the government take care of it.
- The parameters of the highway varies by location and natural conditions.

## **3) Interview with Government website developer**

### Interview Plan:

Interviewee: 1) Yash Ramanuj (Role Play)

Designation: Developer of government website

Interviewer: 1) Harsh Acharya

Designation: Business Development Executive

2) Sarthak Sonagara

Designation: Developer

Date: 28/09/2022

Time: 6:30 PM

Duration: 45 minutes

Place: Zoom meeting

### Purpose of Interview:

To understand the creation and maintenance of the system and management of different lists which contain details of highways and their kilometers in particular state.

### Agenda:

- Introduction and current status
- How do you handle updating the database and maintaining the system
- About the ideas for improving the performance of the system
- How you make the website attractive.

### Documents to be brought to the interview:

- Rough plan of building
- Any documents relating to the current system, analysis of the performance, and feedback of the customers.

### Questions:

- 1) How complex is the current system implementation? Is the current system able to answer all the queries?
- 2) How much traffic do you see on the website regularly? Which are the frequently viewed information?
- 3) What kind of feedback do you receive for new features?
- 4) How flexible is the current system, Is it easy to add new features, how do we start for the new system?

### Results of the interview:

- After the interview, we got to know that an attractive website plays an important role in publicity.

- We will increase the performance of the system by efficient implementation and a well structured system.
- For updating databases, we have to take care that other databases are not messed up(integrity is maintained).
- We have to start from basics and have to gather all the possible features that will be provided for better implementation knowledge.

*Combined Requirements:*

- From the interviews, we got the government and the contract company's point of view. We realized that we have to add a responsible entity which will take care of the contact information and the data about the engineers involved in the particular project. We got to know what is the process of the highway construction, what are the parameters related to it, What are the problems that currently users are facing. From interviews we added company, pending, completed and responsible entities and the insight of their respective attributes. Which are listed after this section.

### 2.3. Questionnaires

1.

Which State or Union Territory do you belong to? \*

1. Andhra Pradesh

2. Arunachal Pradesh

3. Assam

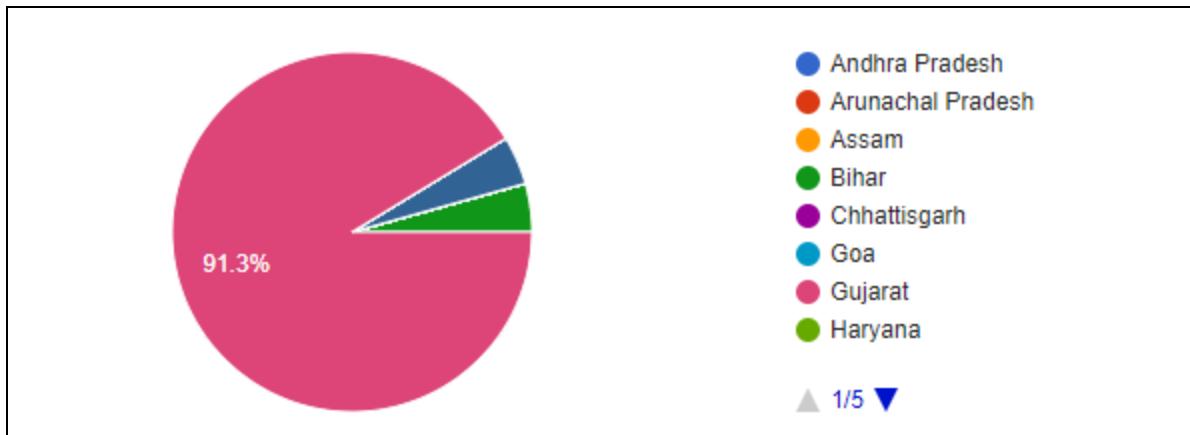
4. Bihar

5. Chhattisgarh

6. Goa

7. Gujarat

8. Haryana



### Intent of this question:

To get to know where the person lives.

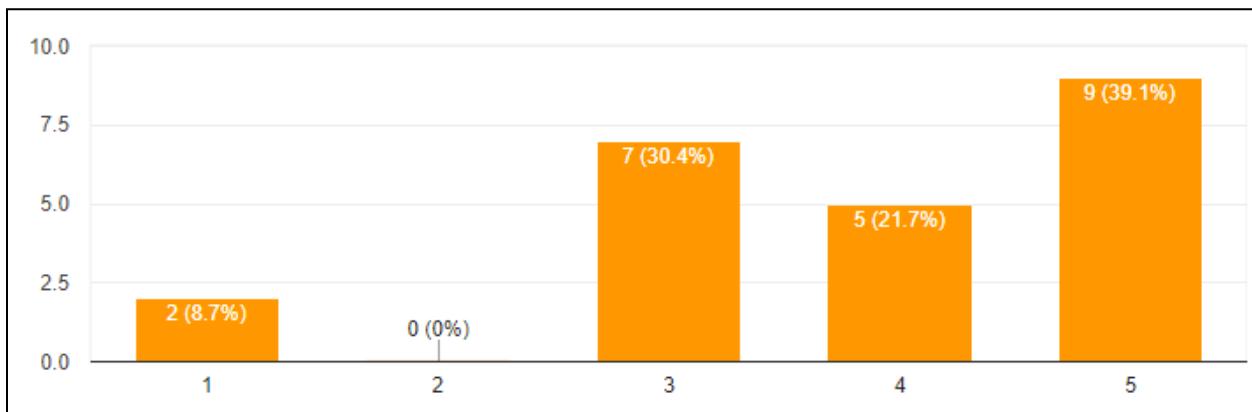
### Observation from the response:

The majority of persons will belong to Gujarat (from survey).

## 2.

How frequently have you travel by highways? \*

1	2	3	4	5
Less Often	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
				More Often



### Intent of this question:

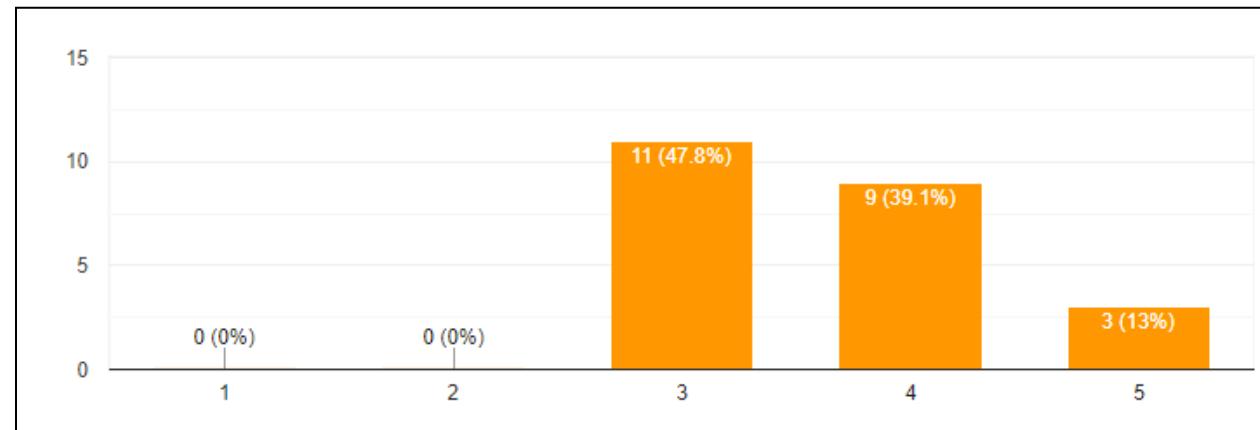
To get to know the frequency of the person travels by highways.

**Observation from the response:**

The majority of people use the highways more often.

**3.**

How was your traveling experience till now? \*



**Intent of this question:**

To get to know the person's traveling experience.

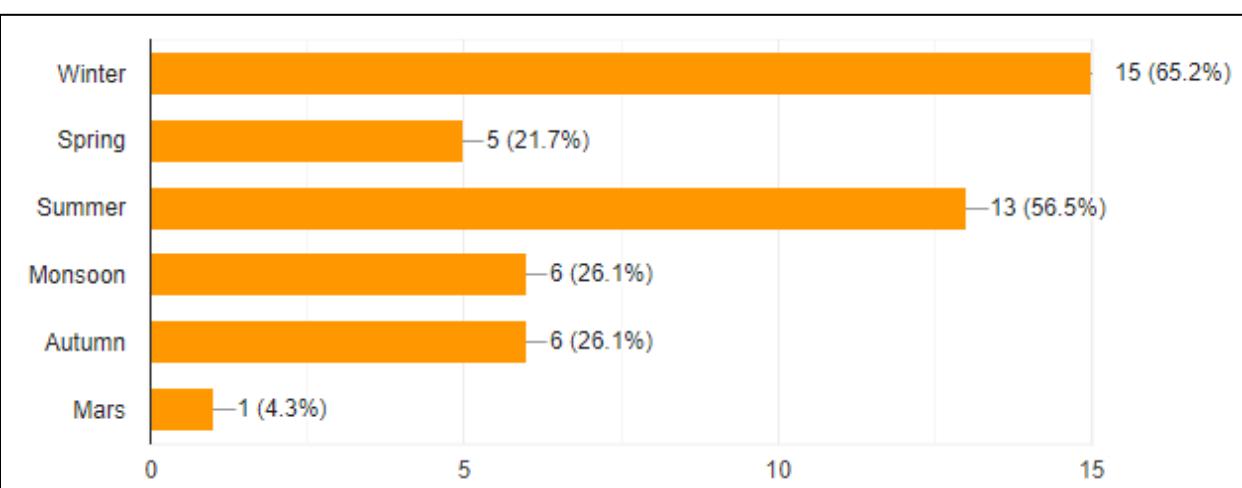
**Observation from the response:**

Mostly 3 ,4 and 5 are the experiences on the scale of 5.

**4.**

In which season do you travel most? \*

- Winter
- Spring
- Summer
- Monsoon
- Autumn



**Intent of this question:**

To get to know when people mostly travelling.

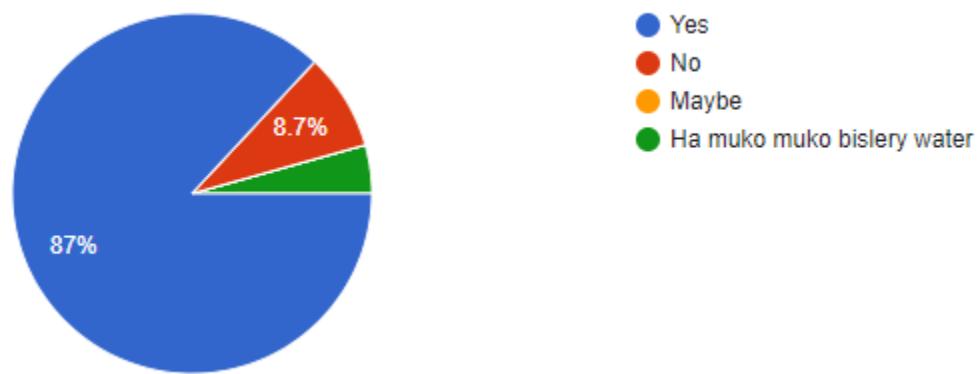
**Observation from the response:**

From this ,we can deduce that when festivals and weekends or summer vacation ,we can not do the construction or any repairing work if not emergency . Because it results in traffic jams and any other issues.

**5.**

In Summer, Do you wish to have Water Booths after proper distance? \*

- Yes
- No
- Maybe

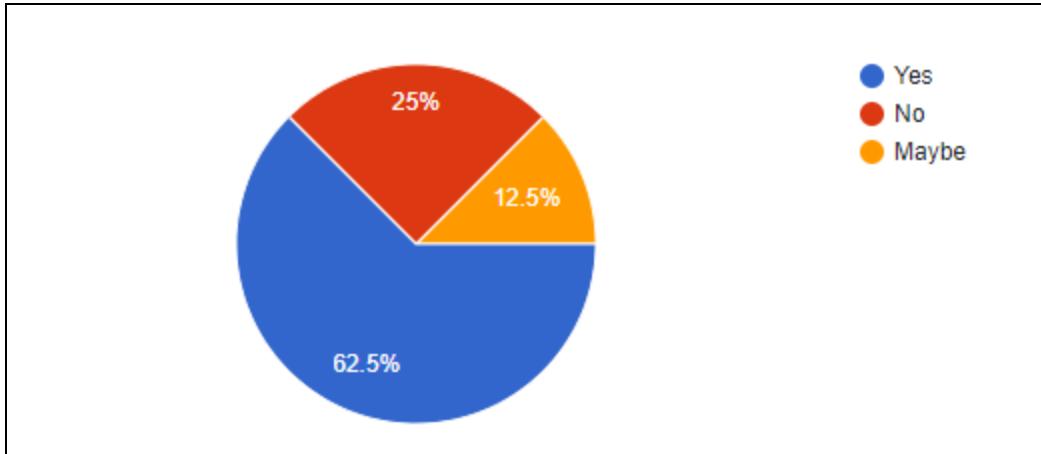


This question is just to know that travelers are satisfied or not with water booths during the summer.

## 6.

Are there any inconvenience caused due to water logging? \*

- Yes
- No
- Maybe



**Intent of this question:**

This question is to know public opinion about water logging during monsoon or any other reason.

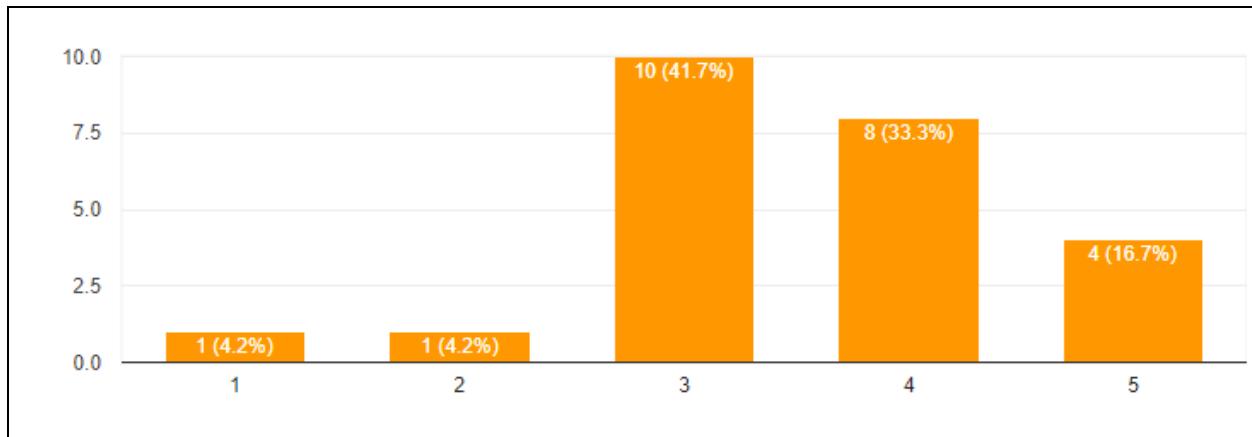
**Observation from the response:**

From the survey most people are facing the problem of water logging so construction companies and the government should take care of this problem and maintenance of constructed highways.

**7.**

How often you face traffic jam? \*

1	2	3	4	5	
Less Frequently	<input type="radio"/> More Frequently				



### Intent of this question:

This question is to know public opinion about traffic jam problems.

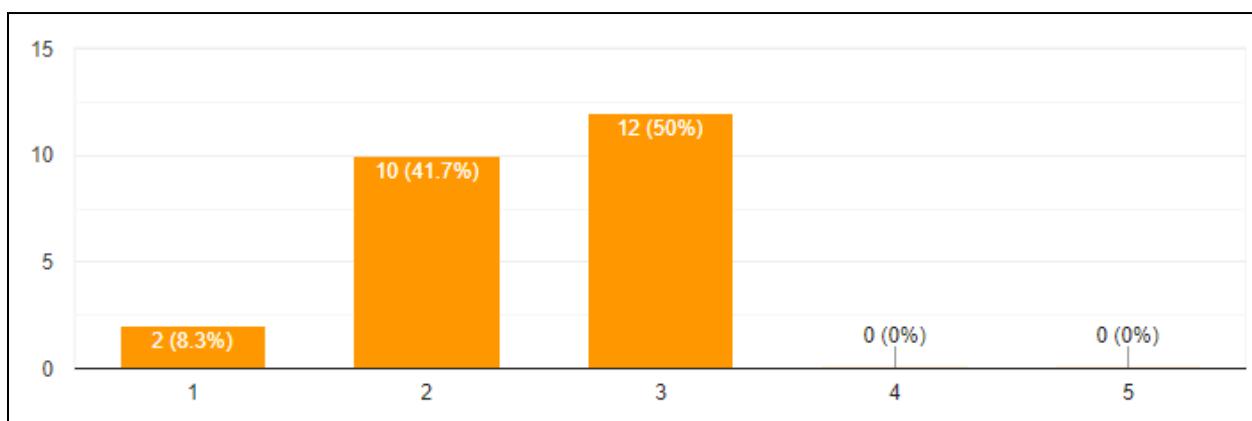
### **Observation from the response:**

From the survey most people have a normal response to traffic jams means they rarely face traffic jam issues on the highways.

8.

Rate condition of the highway in monsoon. \*

	1	2	3	4	5	
Worst	<input type="radio"/>	Best				



### **Intent of this question:**

This question is to know public opinion about highway conditions in monsoon.

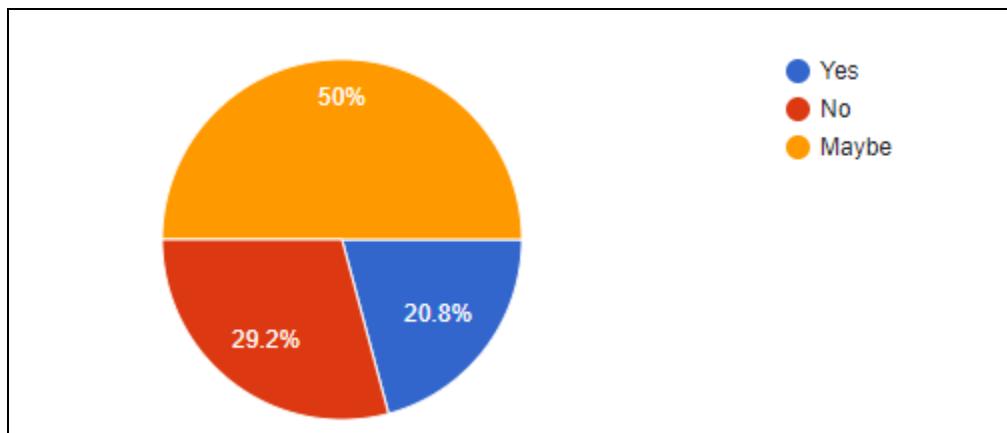
### **Observation from the response:**

From the survey most people have normal rating and some said it is the worst condition so during construction take care of this point that even in monsoon highway condition is best as much as possible.

**9.**

Are emergency services readily available? \*

- Yes
- No
- Maybe

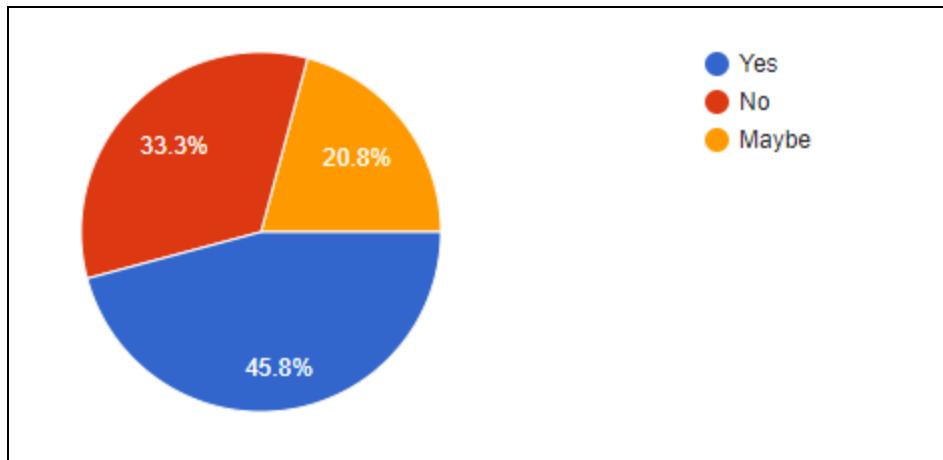


Mostly people don't know about emergency services but it should be.

**10.**

Is there the Highway have enough road facilities? (ex. hotels, petrol pumps,safe drinking water,clean toilets, parking space,etc.)

- Yes
- No
- Maybe



**Intent of this question:**

This question is to know public opinion about facilities on highway.

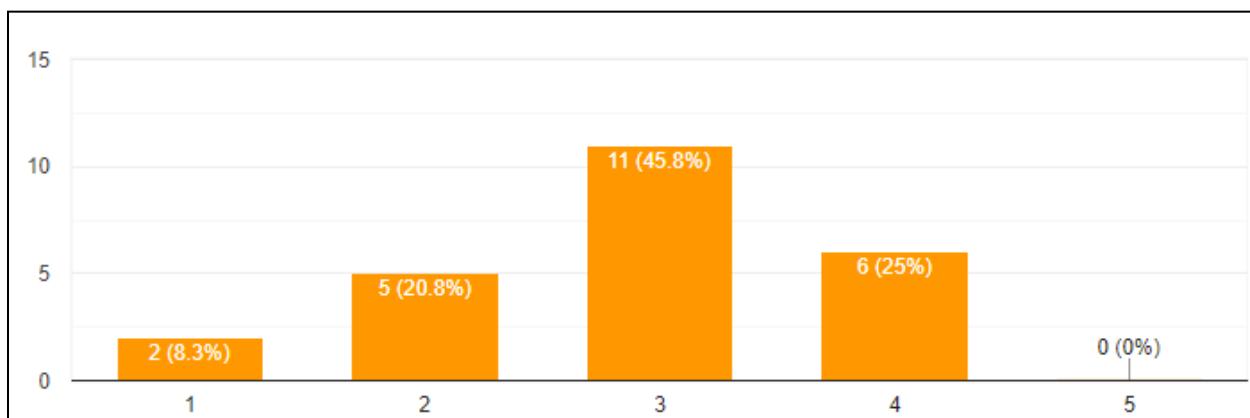
**Observation from the response:**

From survey most of people satisfies with facilities some of them not satisfies so it should be taken care.

**11.**

Rate the toll amount collected in terms of fairness.\*

1	2	3	4	5	
Unfair	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fair



**Intent of this question:**

This question is to know public opinion about toll amount.

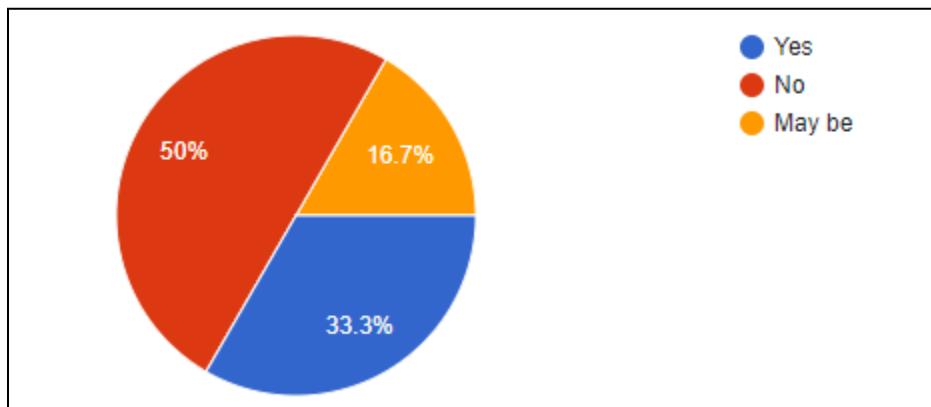
**Observation from the response:**

From survey most of people have normal opinion on fairness of toll amount but some people find it unfair so it should be as less as possible.but mostly now government do not take toll amount.

## 12.

Is it easy to drive at night? \*

- Yes
- No
- May be

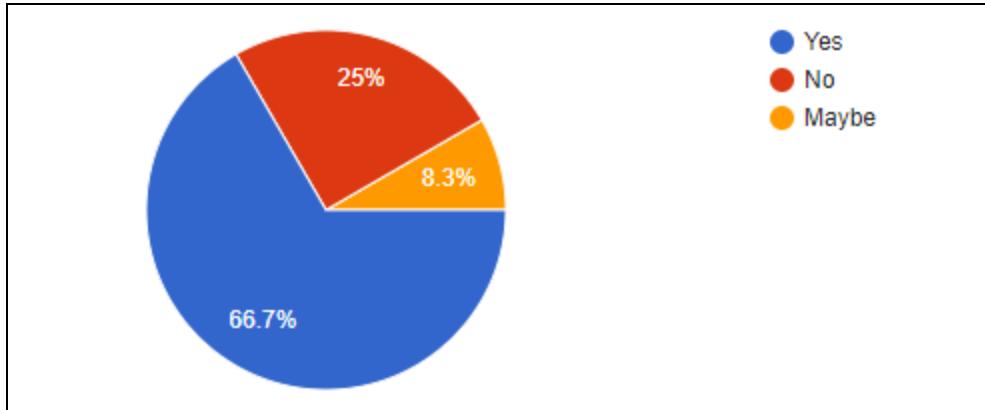


This question is to know easiness to drive on highway at night.

## 13.

Is there enough lights and road signs(Radium Lights,boards)on highway? \*

- Yes
- No
- Maybe



**Intent of this question:**

This question is to know public opinion about lights and road signs.

**Observation from the response:**

From survey most of people have satisfies with these facilities and some of them do not satisfy so it should be taken care .

**14.**

**Any other suggestions?**

Highway authority should repair the road and specially in monsoon they have to survey certain roads which are frequent in use . They have to put clean toilet on the highway . They must put the sign board of every further situation of road . Make overbridge so that traffic will reduce . Strict the traffic rules and find access some tip to reduce road accident

***Combined Requirements:***

- From the questionnaire we took the citizen's point of view in our requirement document. We asked many questions to the citizens and got feedback from them. From the feedback we found some constraints that we have to put on our attributes. Like construction can not be done on busy time. We also get the hierarchy of the database, like which user will be able to see or modify which things.

## *2.4. Observations*

### 1) Observations at site

System : Highway Construction

Observations by: Sarthak Sonagara (Database Developer)

Date: 12/9/2022

Time: 15:30

Duration: 1 hour

Place: Sola Bhagwat, Ahmedabad (Highway Construction Site)

#### Observations:

- The passengers moving from the construction site are facing high rush-traffic on their way.
- We can take surveys of users to get the time when there is less traffic.
- Difficult to construct highways by exact planning or map. It is requiring more time and resources than planned.
- We can be accurate on this error by taking multiple surveys or interviews and past analysis from the database.
- Natural calamities like rain, or rush of festivals are causing the stop of construction for some time.
- Contract companies must use the proper raw material for highway construction.
- For this we can add these details in the contract itself and make sure that the quality of the highway is maintained.
- Some government inspector must be there to inspect the work being done.

### 2) Observations at office

System : Construction Company Office

Observations by: Harsh Acharya (Database Developer)

Date: 13/9/2022

Time: 11:30

Duration: 30 minutes

Place: T&T construction company office, Ahmedabad

#### Observations:

- Companies are receiving tender with less description.
- There is high competition in the market for construction companies.

- Transparency in communication is an issue, stopping the good deals.
- Website for filling the tenders is not having a good interface.
- Past data is really helpful for companies to provide their approximate cost to the government and time it takes to construct or renovate the highway.
- Therefore having a proper nice database with all the required information helps a lot.

*Combined Requirements:*

- We got familiar with the real life scenario of highway construction and management after the observations. We added some entities like toll in our planning of the database also. By observations we observed the basic problems and the requirements that we will be trying to implement by creating the database.

### 3. FACT FINDING CHART

<b>Objective</b>	<b>Technique</b>	<b>Subject</b>	<b>Time Commitment</b>
To get the background knowledge of highway construction process	BackGround Reading	View of real data and similar work done	1 day
Preliminary meeting to identify problem, solution and process	Interview	Construction company manager	45 minutes
Details from the government point of view	Interview	Employee at NHAI	1 hour
To understand current system	Interview	Developer of government website	45 minutes
To understand the users' perspective	Questionnaire	Citizens of the city	1 day
To be familiar with real scenario	Observation	Highway construction site	1 hours
To feel company's environment	Observation	Construction company's office	30 minutes

## **4. LIST REQUIREMENTS**

- Database should be able to store mentioned data/perform mentioned functions :
- It should be able to store information of the tender that has been passed for specific highways and many companies. Tenders have different attributes of a highway, states and location of the highway, cost of the highway and the similar details. Companies should be able to read it and can submit their cost and description to the government.
- Database should also keep details of all the companies, their contact information, their past project, relation with the government for future projects and current highway maintenance.
- For passing the tender, the government should have a ranking of the companies on the basis of their past performance. The government also awards the companies time by time, for that reason there should be ranking among the companies.
- Database should store the attributes of the highway itself like number of lens, length, width, type of vehicle which can pass on it, building date, building cost, etc.
- The states in which a particular highway is falling is also needed for connectivity. For that it should keep information of states and union territories also.
- The highway may need a renovation. Highway may undergo maintenance also. Database should take care of the details of the contract of these events, cost associated with it, companies associated with it, starting date, ending date type of information.
- For future insights, details of the past projects' which are already completed are very important. Completed projects' details must be there in the database.
- For every project we need to store the contact details of the responsible person for that project. The responsible person could be a contract company manager or the government officer.
- Every type of vehicle cannot pass through the highway, therefore the type and the properties of the vehicle to check whether they are allowed to pass on it or not.
- The government charges the tolls at some points on the highway, so

the number of the tolls and the money collected from the toll should also be stored.

Our Planning :

1. COMPANY(Cid, Name, Contact{email + phone(multi-valued) + address})
2. HIGHWAS(No, Length, lanes, Description, type, age(derived from build date), Rating)
3. STATE(Name)
4. PENDING(P\_no, type, status, H\_no, Cid, start\_date, expected\_end\_date, description, cost, responsible)  
Build + renovate + maintain
5. COMPLETED(P\_no, end\_date)
6. VEHICLE(Type, Weight, Height)
7. RESPONSIBLE(Related\_people, field, organization, work)
8. TOLL(H\_no, no\_of\_tollbooth, spending, earning)
9. RANKING(Type, year, position, cid)
10. TENDER(Highway\_No, Cid, C\_Name, Quality, T\_value)

## 5. USER CATEGORIES AND PRIVILEGES

### Government

- **National Highway Authority of India Minister**

The minister has many accesses on the database. The minister can view pending status, completed status, can contact the company, can change the toll, can recommend a rank also. Therefore the minister has access to almost all the database.

- **National Highway Authority of India Employees**

The government employees must be able to contact the companies. They also have to modify or view highway attributes. They can also view pending and completed projects or be able to call the responsible person. But employees generally don't have access to the ranking of the companies or to the toll information.

- **Finance department employees**

The government's Finance department employees are only able to access some part of highway attributes, because the details of the highway are not necessary for them. Only highway number and cost related information like tender and the cost to build and to renovate is necessary for them. For abstraction and security purposes we must restrict these employees to get authority to view or modify unnecessary information.

- **Development officers**

They are also not able to view many of the data stored on the database. They just need information about the number of highways and the status of the highway for the development of society.

## Contract companies

- **Company CEO**

Companies do not have many privileges like the government. Company CEO is the lead person in the company, so the CEO can access the previously built highway data for future planning. They can also see their projects' status, cost, time related information. They can watch ranking and tender details also. But, Toll information, types of vehicles, and other companies' projects are out of their business. Only the Government can modify it.

- **Company employees**

Company's employee needs to know about the working projects of the company, and the responsible government person for the projects. But tender details are not accessible to them, other government related information is also not part of their scope.

## Citizens

- Citizens just have the information that which highways are there from and to which cities they are passing. All other information must be restricted to the citizens.

## **6. ASSUMPTION**

- To reduce the complexity we have assumed some things.
- A highway has equal properties at all the places. Like highways' width or type of vehicle allowed or speed limit or toll or number of lanes does not vary from different locations.
- We are assuming there must be some quality factor or rating for highways and contact companies for comparison between them. This is the measure of how work has been done, we will not entertain details of this measure. For example for highways, quality of cement, briks, paint, design and many more and for companies transparency, trust, standard, and many more.
- We do not take account of rare and complicated events like natural calamities, betray, and corruption. In short we assume fairness.

## **7. BUSINESS CONSTRAINTS**

- All contract companies must be genuine. They cannot leave a contract in between. Whatever contract is given to a company it should complete it.
- Market shortage should not happen. Raw material of highway construction like cement, concrete, sand should not go out of run, otherwise these factors highly affect construction cost and time.
- No. of suppliers and contract companies is defined. Based on capacity and cost we can award them. We can rank them on the basis of their performance and find a favorite supplier.
- We can set a limit on total cost and to the number of companies to avoid complexities. There is a certain budget associated with the department, every construction must be done in a given budget constraint.
- Number of resources like land, water are also limited. Number of highways needed is practically limited.

## Section 2: Noun Analysis and ERD

1. Noun Analysis
2. Rejected Nouns
3. Rejected Verbs
4. ER Diagram
5. Entity-Attribute

## 1. Extracted Nouns & Verbs

Sr.No.	Noun	Verb
1	Address	Allow
2	Age	Assign
3	Authority	Collect
4	Award	Communicate
5	Boundary	Complain
6	Budget	Complete
7	City	Construct
8	Collection	Contact
9	Company	Experience
10	Contract	Fill
11	Cost	For
12	Date	Have
13	Department	
14	Description	Maintain
15	E-mail	Make
16	End	Manage
17	Engineers	Movement
18	Expectation	Pass
19	Fund	Pending
20	Goods	Process
21	Government	Record
22	Height	Renovate

23	Highway	Repair
24	Length	Request
25	Location	Responsibility
26	Maintenance	Spend
27	Money	Take
28	Name	Update
29	No of lanes	
30	Number of highways	
31	Organization	
32	Passengers	
33	Performance	
34	Phone	
35	Position	
36	Problem	
37	Project	
38	Quality	
39	Query	
40	Rank	
41	Rating	
42	Registration	
43	Renovation	
44	Responsible	
45	Start	
46	State	
47	Status	

48	Supplier	
49	Tender	
50	Time	
51	Toll	
52	Type	
53	Type of vehicle	
54	User	
55	Vehicle	
56	Volume	
57	Weight	
58	Year	

## 2. Accepted Noun & Verbs list

Candidate Entity set	Candidate attribute set	Candidate relationship set
Highway	<u>Highway_no</u> , length, no_of_lanes, type, description, quality_rating, building_date, highway_status	Passing_through, Allows, Takes, Adds
Company	<u>Company_id</u> , name, type, contact{email, phone, address}, rating	Works_on(construct+maintena nce+renovate), Fills, Has

Rank	<u>Award_type</u> , <u>position</u> , <u>year</u>	Gives, Has
Vehicle	<u>Type</u> , <u>weight</u> , <u>height</u>	Allows
Completed	<u>Project_no</u> , <u>Project_type</u> , <u>starting_date</u> , <u>ending_date</u> , <u>cost</u> , <u>description</u>	Gets
Person	<u>Organization</u> , <u>Organization_id</u> , name, Contact{E-mail, phone}	Registers, Responsible for
States	<u>Name</u>	Passing through
Toll(Weak Entity)	<u>Toll_no</u> , <u>spending</u> , collection	Takes
Tender	<u>Highway_no</u> , <u>Quality_rating</u> , <u>Exp_cost</u> , <u>Exp_time</u>	Fills, for
Project	<u>Project_no</u> , <u>Project_type</u> , <u>Status</u> , <u>Highway_no</u> , <u>Starting_date</u> , <u>Expected_ending_date</u> , <u>Expected_cost</u>	Assigns, Gets, Works on, Responsible for
Government	<u>Employee_id</u> , <u>Employee_Department</u>	Adds, Selects, Gives, Assigns, Solves

### 3. Rejected Noun & Verbs list

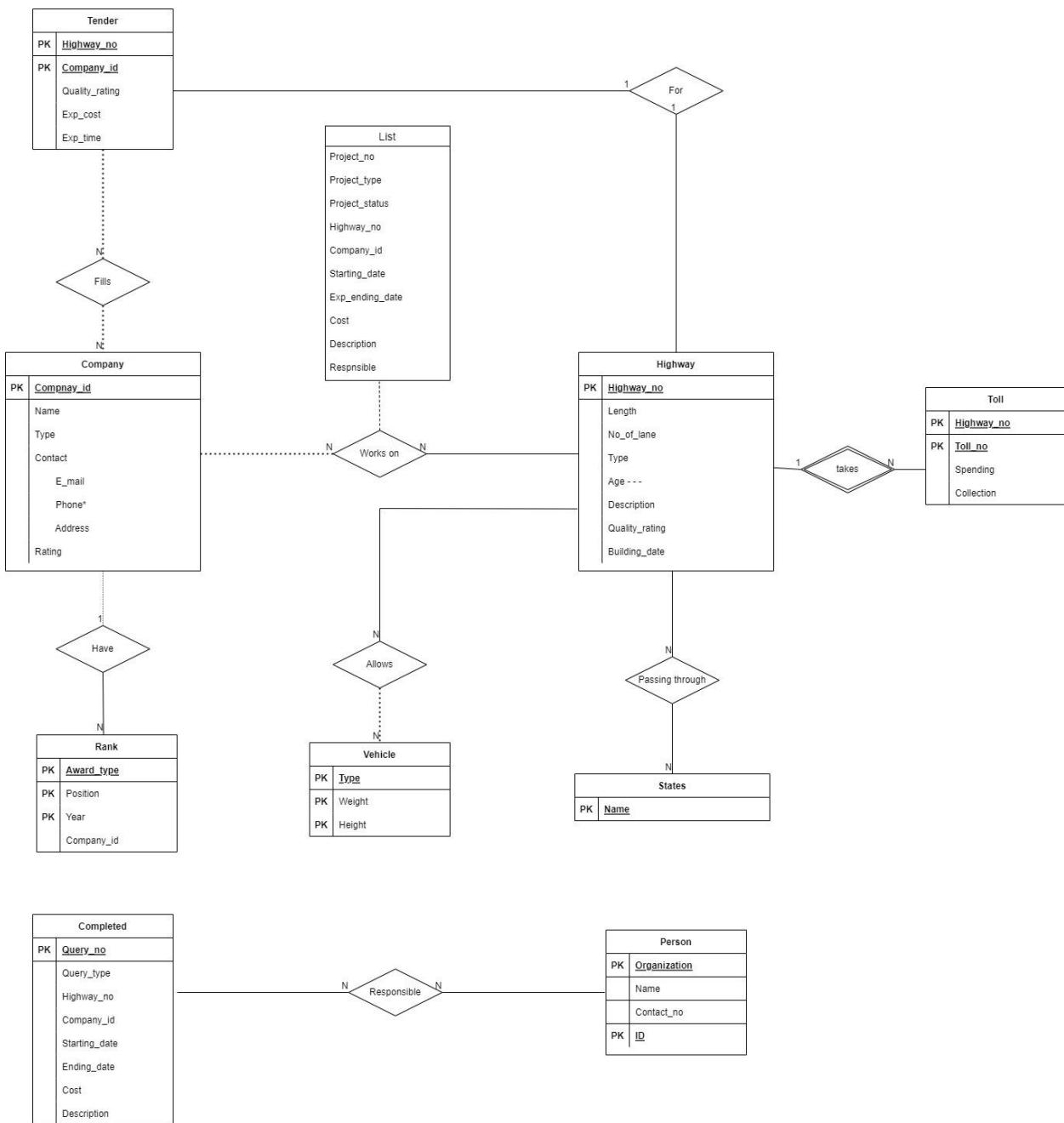
Sr.No.	Noun & Verb	Reason
1	Address	attributes
2	Age	attributes
3	Authority	general
4	Award	attributes
5	Boundary	irrelevant
6	Budget	duplicate
7	City	general
8	Collection	attributes
9	Contract	vague
10	Cost	attributes
11	Date	attributes
12	Description	attributes
13	E-mail	attributes
14	End	general
15	Engineers	duplicate
16	Expectation	general
17	Fund	duplicate
18	Goods	irrelevant
19	Status	attributes
20	Height	attributes
21	Length	attributes
22	Location	vague
23	Money	duplicate

24	Name	attributes
25	No of lanes	attributes
26	Number of highways	general
27	Organization	attributes
28	Passengers	irrelevant
29	Performance	general
30	Phone	attributes
31	Position	attributes
32	Department	attributes
33	Project	attributes
34	Quality	attributes
35	Query	vague
36	Rating	attributes
37	Registration	irrelevant
38	Start	general
39	Supplier	duplicate
40	Time	general
41	Type	attributes
42	Type of vehicle	attributes
43	User	general
44	Volume	vague
45	Weight	attributes
46	Year	attributes
47	Allow	associations
48	Assign	vague
49	Collect	attributes

50	Communicate	irrelevant
51	Complaint	general
52	Construct	associations
53	Contact	attributes
54	Experience	irrelevant
55	Fill	associations
56	For	associations
57	Have	associations
58	Maintain	associations
59	Make	general
60	Manage	general
61	Movement	vague
62	Pass	associations
63	Pending	associations
64	Process	general
65	Record	irrelevant
66	Renovate	associations
67	Repair	duplicate
68	Request	general
69	Responsibility	duplicate
70	Spend	attributes
71	Take	associations
72	Update	general

## ER-Diagram:

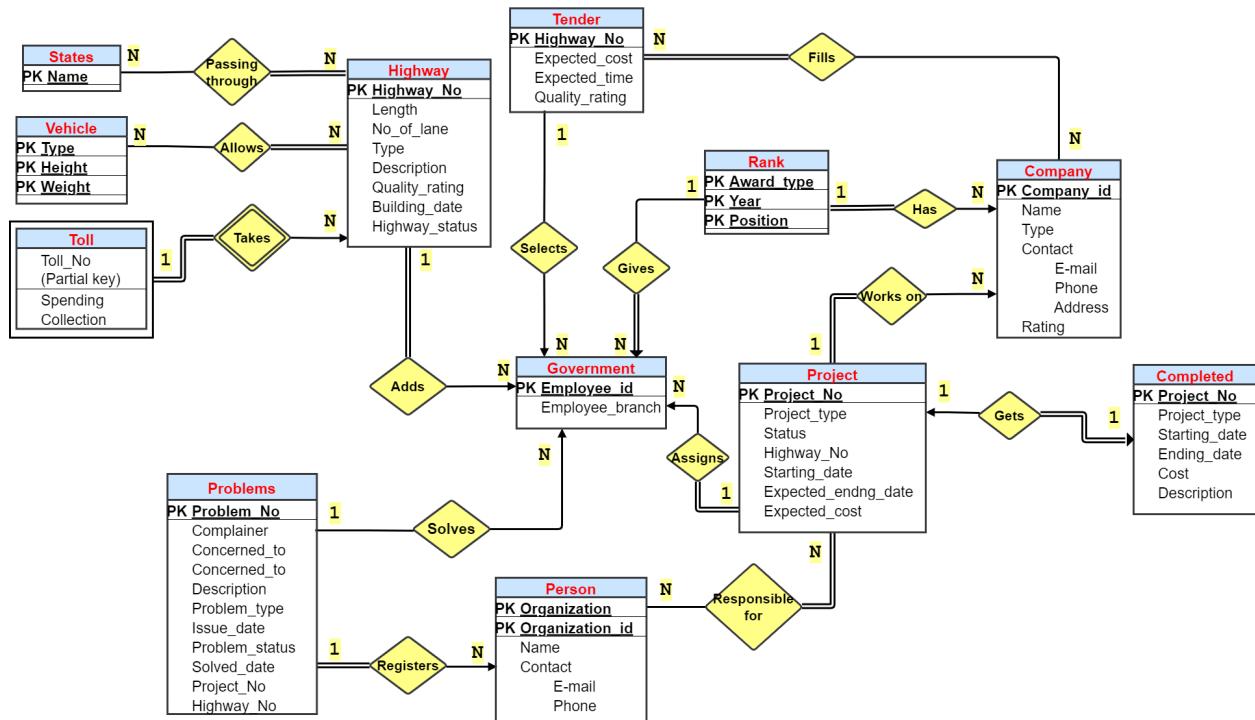
Version-1:



## Version 2:(Final)

DRAW.IO :

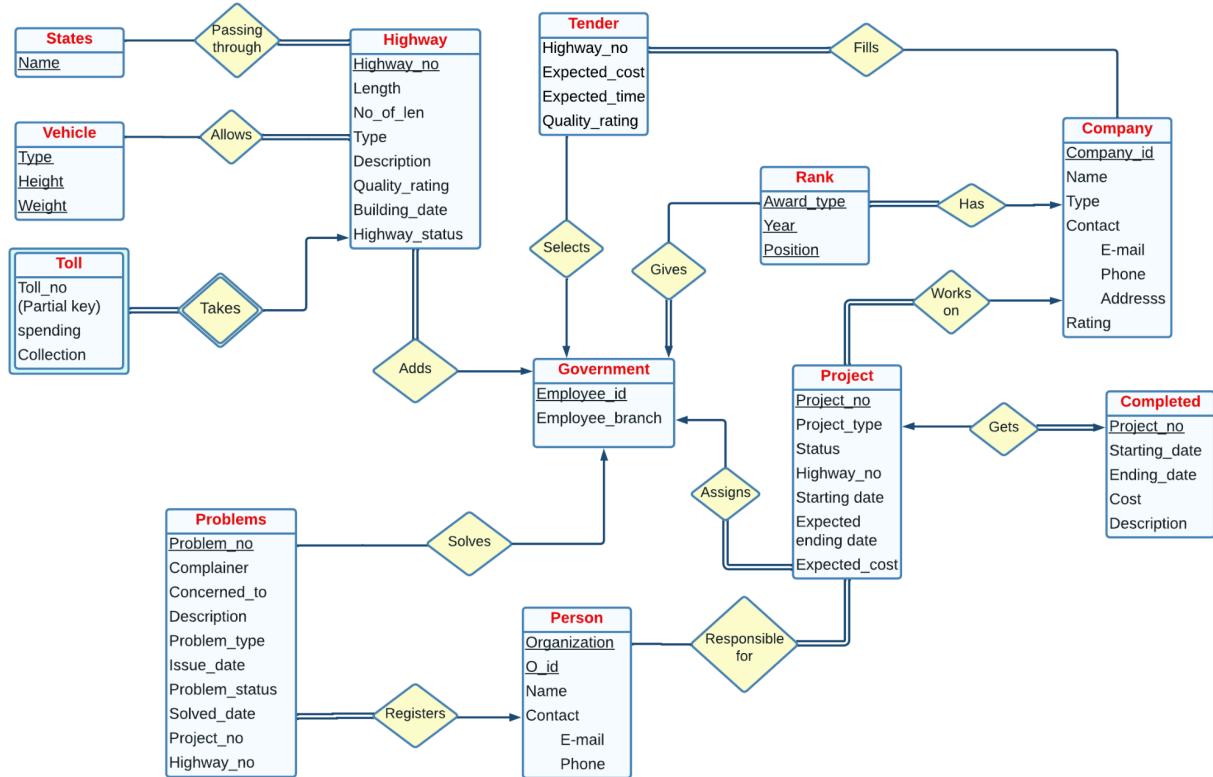
<https://drive.google.com/file/d/1bNy9r340mfdBHsJYxgMqVi0z1zMaa1o5/view?usp=sharing>



Here,

- Double line shows Total Participation.
- Numbers near the entities are the cardinalities of the entities for respective relations.
- Arrow also shows the same. Relation is One at the arrow side and Many at the non-arrow side.
- For example relation **Registers**(at the very bottom side of the diagram) between **Problems** and **Person**, is MANY to ONE. The cardinality of Problems is 1 and the cardinality of the Person is N indicating a person might have many problems but a problem is registered by a particular person. Also double line shows that the Problems has Total Participation, indicating that No problem can be there without a person' registration.

## LUCIDCHARTS :



Candidate Entity set	Candidate attribute set
Highway	Highway_no, length, no_of_lanes, type, description, quality_rating, building_date, highway_status
Company	Company_id, name, type, rating, email, address [phone was removed (multivalued)]
Rank	Award_type, position, year
Vehicle	Type, weight, height
Completed	Project_no, Project_type starting_date,

	ending_date, cost, description
Person	Organization, Organization_id, name, E-mail [phone was removed (multivalued)]
States	Name
Tender	Highway_no, Quality_rating, Exp_cost, Exp_time
Project	Project_no, Project_type, Status, Highway_no, Starting_date, Expected_ending_date, Expected_cost
Government	Employee_id, Employee_Department

Weak Entity :

Toll(Weak Entity)	Highway_no(from highway), Toll_no, spending, collection
-------------------	--

## Section 3: Normalization and Schema Refinement

1. Normalization
2. Schema refinement

Notations:

**BLUE** - decomposed from the composite attribute

**RED** - multivalued attribute

**ORANGE** - Primary key

**Purple** - Foreign key

**PINK** - Foreign key + Primary key

2)

Relation Tables	Table Attributes	Non trivial Functional Dependencies
Highway	<b>Highway_no</b> , length, no_of_lanes, type, description, quality_rating, building_date, highway_status, <b>Employee_id</b> (from Government)	Highway_no -> all Length, no_of_lanes -> type Highway_status -> quality_rating
Company	<b>Company_id</b> , name, type, rating, <b>email</b> , <b>address</b> [ <b>phone</b> was removed (multivalued)]	<b>Company_id</b> -> all
Rank	<b>Award_type</b> , <b>position</b> , <b>year</b> , <b>Employee_id</b> (from Government), <b>Company_id</b> (from Company)	<b>Award_type</b> , <b>position</b> , <b>year</b> -> all
Vehicle	<b>Type</b> , <b>weight</b> , <b>height</b>	Type, weight, height -> all
Completed	<b>Project_no</b> (from Project), Project_type, starting_date, ending_date, cost, description	<b>Project_no</b> -> all
Person	<b>Organization</b> , <b>Organization_id</b> , name,	<b>Organization</b> , <b>Organization_id</b> -> all

	<b>E-mail</b> [ <b>phone</b> was removed (multivalued)]	Organization -> E-mail
States	<b>Name</b>	Name -> all
Tender	<b>Highway_no</b> , <b>Quality_rating</b> , <b>Exp_cost</b> , <b>Exp_time</b> , <b>Employee_id</b> (from Government)	Highway_no -> all
Project	<b>Project_no</b> , <b>Project_type</b> , <b>Status</b> , <b>Highway_no</b> , <b>Starting_date</b> , <b>Expected_ending_date</b> , <b>Expected_cost</b> , <b>Employee_id</b> (from Government), <b>Company_id</b> (from Company)	<b>Project_no</b> -> all <b>Project_type</b> -> <b>Expected_ending_date</b> <b>Project_type</b> -> <b>Expected_cost</b>
Government	<b>Employee_id</b> , <b>Employee_Department</b>	<b>Employee_id</b> , <b>Employee_Department</b> -> all
Problems	<b>Problem_no</b> , <b>Complainier</b> , <b>Concerned_to</b> , <b>Description</b> , <b>Problem_type</b> , <b>Issue_date</b> , <b>Problem_status</b> , <b>Solved_date</b> , <b>Project_no</b> , <b>Highway_no</b> , <b>Employee_id</b> (from Government), <b>Organization</b> (from Person), <b>Organization_id</b> (from Person)	<b>Problem_no</b> -> all <b>Problem_type</b> -> <b>Concerned_to</b>

3)

<b>Relation Tables</b>	<b>Redundancies</b>	<b>Anomalies</b>
Highway	<p><b>Length, no_of_lanes -&gt; type</b></p> <p>For every repeated value of Length, no_of_lanes the type will be repeated also.</p>	<p>Without inserting the length we cannot determine the type.</p> <p>If we update the length, we have to update the type for every matching tuple.</p> <p>If we delete the no_of_length in the highway we cannot store the type of the highway.</p>
Company	No redundancies	No anomalies
Rank	No redundancies	No anomalies
Vehicle	No redundancies	No anomalies
Completed	No redundancies	No anomalies
Person	<p><b>Organization -&gt; Email</b></p> <p>For every repeated value of Organization, E-mail will be repeated also.</p>	<p>Without inserting organization we cannot determine the e-mail.</p> <p>If we update the organization, we have to update the e-mail for every matching tuple.</p> <p>If we delete the organization in the person we cannot store</p>

		the e-mail of the person.
States	No redundancies	No anomalies
Tender	No redundancies	No anomalies
Project	<p><b>Project_type -&gt;</b>  <b>Expected_ending_date,</b>  <b>Expected_cost</b></p> <p>For every repeated value of Project_type, Expected_ending_date and expected_cost will be repeated also.</p>	<p>Without inserting the Project_type we cannot determine expected_ending_date,Expected_cost of project.</p> <p>If we update the project_type, we have to update the expected_ending_date and expected_cost for every matching tuple.</p> <p>If we delete the project_type in the project we cannot store the expected ending date and cost of the project.</p>
Government	No redundancies	No anomalies
Problems	<p><b>Problem_type -&gt;</b>  <b>Concerned_to</b></p> <p>For every repeated value of Problem_type, Concerned_to will be repeated also.</p>	<p>Without inserting the Problem_type we cannot determine concerned_to.</p> <p>If we update the problem_type, we have to update the concerned_to for every matching tuple.</p> <p>If we delete the problem_type in the problems we cannot store the concerned_to of</p>

		problems.
--	--	-----------

#### 4) 1 NF :

- Since we have removed the “phone\_number” which had a multivalued attribute, There is no other multivalued attribute in the schema.
- Also, there is no composite attribute in the schema after decomposing email and address from contact attribute.
- Therefore we conclude that the schema is in 1NF.

#### 5) 2 NF:

- We have found out all the dependencies which are redundant in the schema.
- From which the following is the Partial Dependency:  
**Organization -> Email**
- Here, in the “person” table, “organization” is the part of the primary key (Organization, Organization\_id). It is determining a non-key attribute “e-mail”. Therefore it is a partial dependency and a “person” table is not in 2 NF.

For the solution of the redundancy, we need to decompose the Person table.

PERSON	Person	Organization, Organization_id, name
	Org_mail	Organization, E-mail

#### 6) 3 NF:

- From all the dependencies following are the Transitive dependencies:

`Project_type -> Expected_ending_date, Expected_cost`

`Problem_type -> Concerned_to`

`Length, no_of_lanes -> type`

- In the following dependencies, a non-key attribute is deciding another non-key attribute and a primary key always decides all the attributes. Therefore these are the transitive dependencies. These tables are not in 3 NF.
- So, we have to decompose the tables according to the following table.

7)

Project	Project	<code>Project_no,</code> <code>Project_type, Status,</code> <code>Highway_no,</code> <code>Starting_date,</code> <code>Employee_id(from Government),</code> <code>Company_id(from Company)</code>
	Expectation	<code>Project_type,</code> <code>Expected_ending_date, Expected_cost</code>
Problem	Problem	<code>Problem_no,</code> <code>Complainant,</code> <code>Description,</code> <code>Problem_type,</code> <code>Issue_date,</code> <code>Problem_status,</code> <code>Solved_date,</code> <code>Project_no,</code> <code>Highway_no,</code> <code>Employee_id(from Government),</code> <code>Organization(from Person),</code>

		<b>Organization_id</b> (from Person)
	Concerned	<b>Problem_type</b> , Concerned_to
Highway	Highway	<b>Highway_no</b> , length, no_of_lanes, description, quality_rating, building_date, highway_status, <b>Employee_id</b> (from Government)
	Highway_type	length, no_of_lanes, type

## Section4 : Final DDL Scripts and Queries.

1. DDL Script Snapshots
2. Data Snapshots
3. Queries and its output snapshots

## 1.DDL SCRIPT SNAPSHOTS:

```
-- Table: high.allows

-- DROP TABLE IF EXISTS high.allows;

CREATE TABLE IF NOT EXISTS high.allows
(
    v_type character varying COLLATE pg_catalog."default" NOT NULL,
    v_hight double precision NOT NULL,
    v_weight double precision NOT NULL,
    h_type character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT allows_pkey PRIMARY KEY (v_type, v_hight, v_weight, h_type),
    CONSTRAINT allows_v_type_v_hight_v_weight_fkey FOREIGN KEY (v_hight, v_type, v_weight)
        REFERENCES high.vehicle (height, type, weight) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.allows
OWNER to postgres;
```

```
-- Table: high.company

-- DROP TABLE IF EXISTS high.company;

CREATE TABLE IF NOT EXISTS high.company
(
    c_id bigint NOT NULL,
    name character varying COLLATE pg_catalog."default",
    rating double precision,
    email character varying COLLATE pg_catalog."default",
    address character varying COLLATE pg_catalog."default",
    type character varying COLLATE pg_catalog."default",
    CONSTRAINT company_pkey PRIMARY KEY (c_id)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.company
    OWNER to postgres;
```

```
-- Table: high.company_phone

-- DROP TABLE IF EXISTS high.company_phone;

CREATE TABLE IF NOT EXISTS high.company_phone
(
    c_id bigint NOT NULL,
    phone_no bigint NOT NULL,
    CONSTRAINT company_phone_pkey PRIMARY KEY (c_id, phone_no),
    CONSTRAINT company_phone_c_id_fkey FOREIGN KEY (c_id)
        REFERENCES high.company (c_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.company_phone
    OWNER to postgres;
```

```
-- Table: high.completed

-- DROP TABLE IF EXISTS high.completed;

CREATE TABLE IF NOT EXISTS high.completed
(
    p_no bigint NOT NULL,
    p_type character varying COLLATE pg_catalog."default",
    starting_date date,
    ending_date date,
    cost bigint,
    h_no bigint,
    c_id bigint,
    CONSTRAINT completed_pkey PRIMARY KEY (p_no)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.completed
    OWNER to postgres;
```

```
-- Table: high.fills

-- DROP TABLE IF EXISTS high.fills;

CREATE TABLE IF NOT EXISTS high.fills
(
    h_no bigint NOT NULL,
    c_id bigint NOT NULL,
    CONSTRAINT "Fills_pkey" PRIMARY KEY (h_no, c_id),
    CONSTRAINT "Fills_c_id_fkey" FOREIGN KEY (c_id)
        REFERENCES high.company (c_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.fills
    OWNER to postgres;
```

```
-- Table: high.government

-- DROP TABLE IF EXISTS high.government;

CREATE TABLE IF NOT EXISTS high.government
(
    emp_id bigint NOT NULL,
    emp_dep character varying COLLATE pg_catalog."default",
    CONSTRAINT government_pkey PRIMARY KEY (emp_id)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.government
OWNER to postgres;
```

```
-- Table: high.highway

-- DROP TABLE IF EXISTS high.highway;

CREATE TABLE IF NOT EXISTS high.highway
(
    h_no bigint NOT NULL,
    length bigint,
    no_of_lanes bigint,
    type character varying(50) COLLATE pg_catalog."default",
    quality_rating double precision,
    build_date date,
    emp_id bigint,
    CONSTRAINT highway_pkey PRIMARY KEY (h_no),
    CONSTRAINT highway_emp_id_fkey FOREIGN KEY (emp_id)
        REFERENCES high.government (emp_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.highway
OWNER to postgres;
```

```
-- Table: high.passing

-- DROP TABLE IF EXISTS high.passing;

CREATE TABLE IF NOT EXISTS high.passing
(
    state_name character varying COLLATE pg_catalog."default" NOT NULL,
    highway_no bigint NOT NULL,
    CONSTRAINT passing_pkey PRIMARY KEY (state_name, highway_no),
    CONSTRAINT "passing_Highway_no_fkey" FOREIGN KEY (highway_no)
        REFERENCES high.highway (h_no) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "passing_State_name_fkey" FOREIGN KEY (state_name)
        REFERENCES high.states (name) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.passing
    OWNER to postgres;
```

```
-- Table: high.person

-- DROP TABLE IF EXISTS high.person;

CREATE TABLE IF NOT EXISTS high.person
(
    o_name character varying COLLATE pg_catalog."default" NOT NULL,
    o_id bigint NOT NULL,
    name character varying COLLATE pg_catalog."default",
    email character varying COLLATE pg_catalog."default",
    CONSTRAINT "Person_pkey" PRIMARY KEY (o_id, o_name)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.person
    OWNER to postgres;
```

```
-- Table: high.person_phone

-- DROP TABLE IF EXISTS high.person_phone;

CREATE TABLE IF NOT EXISTS high.person_phone
(
    o_name character varying COLLATE pg_catalog."default" NOT NULL,
    o_id bigint NOT NULL,
    phone_no bigint NOT NULL,
    CONSTRAINT person_phone_pkey PRIMARY KEY (o_name, o_id, phone_no),
    CONSTRAINT person_phone_organization_o_id_fkey FOREIGN KEY (o_id, o_name)
        REFERENCES high.person (o_id, o_name) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.person_phone
    OWNER to postgres;
```

```
-- Table: high.problems

-- DROP TABLE IF EXISTS high.problems;

CREATE TABLE IF NOT EXISTS high.problems
(
    problem_no bigint NOT NULL,
    complainer character varying COLLATE pg_catalog."default",
    problem_type character varying COLLATE pg_catalog."default",
    issue_date date,
    problem_status character varying COLLATE pg_catalog."default",
    solved_date date,
    p_no bigint,
    h_no bigint,
    emp_id bigint,
    o_name character varying COLLATE pg_catalog."default",
    o_id bigint,
    CONSTRAINT problems_pkey PRIMARY KEY (problem_no),
    CONSTRAINT problems_emp_id_fkey FOREIGN KEY (emp_id)
        REFERENCES high.government (emp_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT problems_h_no_fkey FOREIGN KEY (h_no)
        REFERENCES high.highway (h_no) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT problems_o_name_o_id_fkey FOREIGN KEY (o_id, o_name)
        REFERENCES high.person (o_id, o_name) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
```

```
CONSTRAINT problems_p_no_fkey FOREIGN KEY (p_no)
    REFERENCES high.project (p_no) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE CASCADE
    NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.problems
OWNER to postgres;
```

```
-- Table: high.project

-- DROP TABLE IF EXISTS high.project;

CREATE TABLE IF NOT EXISTS high.project
(
    p_no bigint NOT NULL,
    p_type character varying COLLATE pg_catalog."default",
    h_no bigint,
    starting_date date,
    exp_ending_date date,
    exp_cost bigint,
    e_id bigint,
    c_id bigint,
    CONSTRAINT project_pkey PRIMARY KEY (p_no),
    CONSTRAINT project_c_id_fkey FOREIGN KEY (c_id)
        REFERENCES high.company (c_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT project_e_id_fkey FOREIGN KEY (e_id)
        REFERENCES high.government (emp_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.project
    OWNER to postgres;

-- Trigger: pro_comp

-- DROP TRIGGER IF EXISTS pro_comp ON high.project;

CREATE TRIGGER pro_comp
    AFTER DELETE
    ON high.project
    FOR EACH ROW
    EXECUTE PROCEDURE high.ins_del();
```

```
-- Table: high.rank

-- DROP TABLE IF EXISTS high.rank;

CREATE TABLE IF NOT EXISTS high.rank
(
    award_type character varying COLLATE pg_catalog."default" NOT NULL,
    "position" bigint NOT NULL,
    year bigint NOT NULL,
    emp_id bigint,
    c_id bigint,
    CONSTRAINT rank_pkey PRIMARY KEY (award_type, "position", year),
    CONSTRAINT rank_c_id_fkey FOREIGN KEY (c_id)
        REFERENCES high.company (c_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT rank_emp_id_fkey FOREIGN KEY (emp_id)
        REFERENCES high.government (emp_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.rank
    OWNER to postgres;
```

```
-- Table: high.responsible

-- DROP TABLE IF EXISTS high.responsible;

CREATE TABLE IF NOT EXISTS high.responsible
(
    p_no bigint NOT NULL,
    o_name character varying COLLATE pg_catalog."default" NOT NULL,
    o_id bigint NOT NULL,
    CONSTRAINT responsible_pkey PRIMARY KEY (p_no, o_name, o_id),
    CONSTRAINT responsible_o_name_o_id_fkey FOREIGN KEY (o_id, o_name)
        REFERENCES high.person (o_id, o_name) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT responsible_p_no_fkey FOREIGN KEY (p_no)
        REFERENCES high.project (p_no) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.responsible
    OWNER to postgres;
```

```
-- Table: high.states

-- DROP TABLE IF EXISTS high.states;

CREATE TABLE IF NOT EXISTS high.states
(
    name character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT states_pkey PRIMARY KEY (name)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.states
    OWNER to postgres;

-- Table: high.tender

-- DROP TABLE IF EXISTS high.tender;

CREATE TABLE IF NOT EXISTS high.tender
(
    h_no bigint NOT NULL,
    quality_rating double precision,
    exp_cost bigint,
    exp_time bigint,
    e_id bigint,
    CONSTRAINT tender_pkey PRIMARY KEY (h_no),
    CONSTRAINT tender_e_id_fkey FOREIGN KEY (e_id)
        REFERENCES high.government (emp_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.tender
    OWNER to postgres;
```

```
-- Table: high.toll

-- DROP TABLE IF EXISTS high.toll;

CREATE TABLE IF NOT EXISTS high.toll
(
    h_no bigint NOT NULL,
    toll_no bigint NOT NULL,
    spending bigint,
    collection bigint,
    CONSTRAINT toll_pkey PRIMARY KEY (h_no, toll_no),
    CONSTRAINT toll_h_no_fkey FOREIGN KEY (h_no)
        REFERENCES high.highway (h_no) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.toll
    OWNER to postgres;
```

```
-- Table: high.vehicle

-- DROP TABLE IF EXISTS high.vehicle;

CREATE TABLE IF NOT EXISTS high.vehicle
(
    type character varying COLLATE pg_catalog."default" NOT NULL,
    weight double precision NOT NULL,
    height double precision NOT NULL,
    CONSTRAINT vehicle_pkey PRIMARY KEY (type, weight, height)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS high.vehicle
OWNER to postgres;
```

```
-- FUNCTION: high.ins_del()

-- DROP FUNCTION IF EXISTS high.ins_del();

CREATE OR REPLACE FUNCTION high.ins_del()
RETURNS trigger
LANGUAGE 'plpgsql'
COST 100
VOLATILE NOT LEAKPROOF
AS $BODY$
declare num bigint; curdate date;
begin
    select count(*)+1 into num from high.completed;
    select current_date into curdate;
    insert into high.completed values(num,old.p_type,old.starting_date,curdate,NULL,old.h_no,old.v_no);
    return null;
end;
$BODY$;

ALTER FUNCTION high.ins_del()
OWNER TO postgres;
```

## 2.DATA SNAPSHOT :

1    `SELECT * FROM high.allows`

Data output    Messages    Notifications

	v_type [PK] character varying	v_hight [PK] double precision	v_weight [PK] double precision	h_type [PK] character varying
1	two_wheeler	3	100	national
2	two_wheeler	3	200	national
3	two_wheeler	5	500	national
4	two_wheeler	5	100	national
5	two_wheeler	5	200	national
6	four_wheeler	5	750	national
7	four_wheeler	5	1000	national
8	four_wheeler	5	1250	national
9	four_wheeler	5	1500	national
10	four_wheeler	5	2000	national
11	four_wheeler	7.5	750	national
12	four_wheeler	7.5	1000	national
13	four_wheeler	7.5	1250	national
14	four_wheeler	7.5	1500	national
15	four_wheeler	7.5	2000	national

Total rows: 65 of 65    Query complete 00:00:00.082

```
1 SELECT * FROM high.company
```

Data output Messages Notifications



	c_id [PK] bigint	name character varying	rating double precision	email character varying	address character varying	type character varying
1	1	Damon Jillis	2.12	djillis0@yale.edu	635 Sheridan Pass	maintenance
2	2	Tanitansy Trever	5.22	ttrever1@seattlet...	67 Lakewood Gar...	construction
3	3	Kain Bergin	2.93	kbergin2@godad...	4674 Burning Wo...	finance
4	4	Klara Ansteys	8	kansteys3@51.la	6 Graedel Center	maintenance
5	5	Bonnibelle Egent...	1.01	begentan4@virgi...	8 Barby Lane	construction
6	6	Pamelina Stanbri...	6.18	pstanbridge5@s...	34138 Bayside Ci...	supplier
7	7	Calhoun Bedle	9.55	cbedle6@sfgate....	6078 Luster Terr...	finance
8	8	Reena Aldin	2.88	raldin7@infoseek...	76 Dorton Crossi...	maintenance
9	9	Bradford Eisold	9.64	beisold8@miibei...	9 Old Shore Terra...	construction
10	10	Ambur Noakes	3.04	anoakes9@arste...	1 Nobel Plaza	maintenance
11	11	Thaine Sanchis	2.79	tsanchisa@jalbu...	9404 Rockefeller ...	supplier
12	12	Branden Absolom	7.87	babsolomb@milt...	449 Tennessee P...	maintenance
13	13	Halley Finlow	4.82	hfinlowc@china....	9 Badeau Crossing	maintenance
14	14	Jan Soffe	7.39	jsoffed@netscap...	47355 Ohio Way	supplier
15	15	Lillis Folbige	8.47	lfolbige@cnbc.c...	77968 Annamark...	construction

Total rows: 100 of 100

Query complete 00:00:00.057

```
1 SELECT * FROM high.company_phone
```

Data output Messages Notifications



	c_id [PK] bigint	phone_no [PK] bigint
1	1	9416897648
2	1	9827898920
3	2	9382211005
4	2	9866249508
5	3	9097839676
6	3	9989273032
7	4	9574297430
8	5	9828513669
9	6	9278496075
10	6	9446439826
11	7	9976525941
12	8	9124979484
13	8	9624830100
14	8	9889470591
15	9	9676259325

Total rows: 203 of 203

Query complete 00:00:00.062

```
1 SELECT * FROM high.completed
```

Data output Messages Notifications

	p_no [PK] bigint	p_type character varying	starting_date date	ending_date date	cost bigint	h_no bigint	c_id bigint
1	111	renovation	2015-05-02	2022-11-15	[null]	2	52
2	1	construction	2010-01-20	2017-02-28	7293316	1	65
3	2	construction	2010-01-28	2017-03-11	2947326	2	70
4	3	construction	2010-04-09	2017-03-19	6552449	3	43
5	4	construction	2010-04-09	2017-05-13	5193118	4	89
6	5	construction	2010-04-14	2017-05-31	5483027	5	64
7	6	construction	2010-05-07	2017-06-11	2346674	6	91
8	7	construction	2010-05-12	2017-06-17	5290913	7	39
9	8	construction	2010-06-21	2017-06-24	2623254	8	67
10	9	construction	2010-07-26	2017-06-29	3003637	9	20
11	10	construction	2010-08-06	2017-06-30	4859773	10	7
12	11	construction	2010-08-07	2017-07-02	3237774	11	2
13	12	construction	2010-08-20	2017-08-27	9647422	12	22
14	13	construction	2010-08-26	2017-09-12	2310530	13	81
15	14	construction	2010-09-12	2017-09-22	6765552	14	74

Total rows: 111 of 111 Query complete 00:00:00.061

```
1 SELECT * FROM high.fill
```

Data output Messages Notifications



	h_no [PK] bigint	c_id [PK] bigint
1	108	93
2	104	14
3	102	54
4	109	25
5	110	90
6	108	25
7	102	98
8	105	3
9	104	11
10	107	8
11	110	46
12	105	23
13	105	97
14	109	2
15	103	88

Total rows: 137 of 137

Query complete 00:00:00.064

```
1 SELECT * FROM high.government
```

Data output Messages Notifications



	emp_id [PK] bigint	emp_dep character varying
1	1	Development
2	2	Development
3	3	Development
4	4	Development
5	5	NHA
6	6	NHA
7	7	Ministry
8	8	Development
9	9	Development
10	10	Ministry
11	11	HR
12	12	Ministry
13	13	NHA
14	14	Development
15	15	Development

Total rows: 100 of 100

Query complete 00:00:00.103

```
1 SELECT * FROM high.highway
```

Data output Messages Notifications



	h_no [PK] bigint	length bigint	no_of_lanes bigint	type character varying (50)	quality_rating double precision	build_date date	emp_id bigint
1	1	415	2	rural	4.95	2015-10-26	26
2	2	587	3	urban	4.23	2014-10-24	16
3	3	459	2	state	7.52	2014-10-10	6
4	4	437	2	national	3.7	2015-02-26	98
5	5	451	2	national	8.55	2013-10-11	71
6	6	361	2	rural	3.2	2016-09-26	56
7	7	313	3	state	3.21	2014-06-23	99
8	8	506	3	rural	3.55	2012-06-12	94
9	9	448	4	rural	4.63	2013-01-09	91
10	10	400	4	state	4.18	2012-06-20	71
11	11	417	2	rural	4.25	2016-08-22	29
12	12	530	4	state	6.98	2014-12-11	89
13	13	497	2	state	5.36	2013-03-02	11
14	14	314	4	rural	1.13	2013-09-08	44
15	15	543	2	national	2.88	2015-08-08	64

Total rows: 100 of 100

Query complete 00:00:00.055

```
1 SELECT * FROM high.passing
```

Data output Messages Notifications

	state_name [PK] character varying	highway_no [PK] bigint
1	Uttarakhand	1
2	Haryana	1
3	Manipur	1
4	Bihar	1
5	Tamil Nadu	1
6	Kerala	1
7	Karnataka	1
8	Tamil Nadu	2
9	Goa	2
10	Meghalaya	2
11	Himachal Pradesh	2
12	Haryana	2
13	Bihar	2
14	Maharashtra	3
15	Himachal Pradesh	3

Total rows: 412 of 412

Query complete 00:00:00.148

```
1 SELECT * FROM high.person
```

Data output Messages Notifications



	o_name [PK] character varying	o_id [PK] bigint	name character varying	email character varying
1	Skinder	2021800	Elianora Melin	emelin0@cafepre...
2	Trunyx	2020364	Mal Bambury	mbambury1@us...
3	Photobug	2022209	Daniela Stonestr...	dstonestreet2@f...
4	Jaxnation	2020499	Faun Gurr	fgurr3@gravatar....
5	Rhynyx	2021564	Milli Stoller	mstoller4@poste...
6	Demizz	2022288	Jojo Vlasin	jvlasin5@pen.io
7	Skinte	2020967	Tyson Olland	tolland6@bloglin...
8	Flipstorm	2022482	Caroline Skates	cskates7@oracle...
9	Eamia	2022105	Inglis Prodrick	iprodrick8@chro...
10	Yata	2020687	Isabella Hyslop	ihyslop9@ed.gov
11	Eayo	2021183	Randal Cobbe	rcobbea@unesco...
12	Myworks	2020927	Maud Gallie	mgallieb@colum...
13	Devbug	2021228	Cheri Ouldcott	couldcottc@cens...
14	Feedfire	2021530	Moyna Symington	msymingtond@w...
15	Zoomdog	2021272	Jeno Spink	jspinke@omnitur...

Total rows: 100 of 100

Query complete 00:00:00.049

```
1 SELECT * FROM high.person_phone
```

Data output Messages Notifications

	o_name [PK] character varying	o_id [PK] bigint	phone_no [PK] bigint
1	Skinder	2021800	9408900256
2	Trunyx	2020364	9509707818
3	Photobug	2022209	9088675615
4	Jaxnation	2020499	9562781178
5	Rhynyx	2021564	9227771560
6	Demizz	2022288	9505144767
7	Skinte	2020967	9067363514
8	Flipstorm	2022482	9784289055
9	Eamia	2022105	9258774811
10	Yata	2020687	9536325270
11	Eayo	2021183	9786149890
12	Myworks	2020927	9457158072
13	Devbug	2021228	9594719579
14	Feedfire	2021530	9462190048
15	Zoomdog	2021272	9322800605

Total rows: 100 of 100

Query complete 00:00:00.053

```
1 SELECT * FROM high.problems
```

Data output Messages Notifications

	problem_no [PK] bigint	complainert character varying	problem_type character varying	issue_date date	problem_status character varying	solved_date date	p_no bigint	h_no bigint	emp_id bigint
1	2	Ashia Perot	other	2011-11-30	solved	2021-08-21	7	[null]	
2	6	Neala Bauman	safety	2012-01-05	solved	2018-04-06	[null]	43	
3	8	Domini Kemwal	pending work	2013-01-21	solved	2020-01-15	[null]	17	
4	10	Reeba Bowden	other	2013-11-01	solved	2019-12-20	2	[null]	
5	1	Mauricio Ladbrook	improper work	2015-01-30	unsolved	[null]	5	[null]	
6	3	Bryn Mennell	improper work	2015-06-02	unsolved	[null]	5	[null]	
7	4	Porty Arkwright	improper work	2015-09-15	unsolved	[null]	[null]	54	
8	7	Tonia Skerman	improper work	2012-02-19	unsolved	[null]	[null]	11	
9	9	Sanson Casiroli	safety	2013-12-20	unsolved	[null]	11	[null]	

Total rows: 9 of 9 Query complete 00:00:00.054

Ln 1, Col 28

```
1 SELECT * FROM high.project
```

Data output Messages Notifications

	p_no [PK] bigint	p_type character varying	h_no bigint	starting_date date	exp_ending_date date	exp_cost bigint	e_id bigint	c_id bigint
1	1	construction	96	2011-06-23	2023-10-26	8808832	60	4
2	2	construction	97	2013-08-21	2022-12-29	6219324	11	6
3	3	construction	98	2010-11-18	2025-06-15	8749870	19	21
4	4	construction	99	2016-02-11	2022-10-21	1818379	89	9
5	5	construction	100	2011-05-08	2023-01-04	9161092	54	57
6	6	renovation	34	2012-05-22	2024-09-17	4836138	32	93
7	7	renovation	4	2016-07-11	2022-01-23	7002926	77	64
8	8	renovation	10	2012-09-12	2025-08-17	7883235	65	8
9	10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74
10	11	maintenance	77	2016-09-15	2025-04-03	8609333	29	42
11	12	maintenance	62	2014-11-22	2026-06-05	4395020	19	71
12	9	renovation	50	2018-09-18	2026-08-19	2098765	13	40

Total rows: 12 of 12 Query complete 00:00:00.108

Ln

```
1 SELECT * FROM high.rank
```

Data output Messages Notifications



	award_type [PK] character varying	position [PK] bigint	year [PK] bigint	emp_id bigint	c_id bigint
1	nice work	1	2010	30	87
2	nice work	1	2011	56	78
3	nice work	1	2012	2	96
4	nice work	1	2013	79	64
5	nice work	1	2014	59	54
6	nice work	1	2015	90	97
7	nice work	1	2016	93	26
8	nice work	1	2017	13	67
9	nice work	1	2018	49	63
10	nice work	1	2019	46	84
11	nice work	1	2020	28	29
12	nice work	1	2021	4	44
13	nice work	1	2022	25	33
14	fast work	1	2010	41	5
15	fast work	1	2011	29	18

Total rows: 156 of 156

Query complete 00:00:00.063

```
1 SELECT * FROM high.responsible
```

Data output Messages Notifications



	p_no [PK] bigint	o_name [PK] character varying	o_id [PK] bigint
1	11	Skinder	2021800
2	7	Skinder	2021800
3	12	Photobug	2022209
4	3	Rhynyx	2021564
5	11	Rhynyx	2021564
6	4	Demizz	2022288
7	12	Flipstorm	2022482
8	5	Flipstorm	2022482
9	6	Eamia	2022105
10	8	Yata	2020687
11	2	Eayo	2021183
12	6	Eayo	2021183
13	1	Myworks	2020927
14	6	Myworks	2020927
15	7	Zoomdog	2021272

Total rows: 101 of 101

Query complete 00:00:00.125

```
1 SELECT * FROM high.states
```

Data output Messages Notifications



	name [PK] character varying
1	Andhra Pradesh
2	Arunachal Pradesh
3	Assam
4	Chhattisgarh
5	Goa
6	Gujarat
7	Haryana
8	Himachal Pradesh
9	Jharkhand
10	Karnataka
11	Kerala
12	Madhya Pradesh
13	Maharashtra
14	Manipur
15	Meghalaya

Total rows: 28 of 28    Query complete 00:00:00.077

```
1 SELECT * FROM high.tender
```

Data output    Messages    Notifications



	name [PK] character varying
1	Andhra Pradesh
2	Arunachal Pradesh
3	Assam
4	Chhattisgarh
5	Goa
6	Gujarat
7	Haryana
8	Himachal Pradesh
9	Jharkhand
10	Karnataka
11	Kerala
12	Madhya Pradesh
13	Maharashtra
14	Manipur
15	Meghalaya

Total rows: 28 of 28

Query complete 00:00:00.077

```
1 SELECT * FROM high.toll
```

Data output Messages Notifications

	h_no [PK] bigint	toll_no [PK] bigint	spending bigint	collection bigint
1	1	1	1093289	4329302
2	1	2	4544938	4582106
3	1	3	4753153	9906045
4	1	4	3493618	2774992
5	1	5	4506034	7270063
6	1	6	1816942	7150839
7	1	7	3415143	4928257
8	1	8	1256579	2765294
9	1	9	2946885	1299258
10	1	10	3140597	6497620
11	1	11	1102471	8879887
12	1	12	4743020	2135152
13	1	13	1587163	1789715
14	1	14	3100548	3535506
15	1	15	3943009	8328893

Total rows: 1000 of 1716

Query complete 00:00:00.061

```
1 SELECT * FROM high.vehicle
```

Data output Messages Notifications



	type [PK] character varying	weight [PK] double precision	height [PK] double precision
1	two_wheeler	100	3
2	two_wheeler	200	3
3	two_wheeler	500	5
4	two_wheeler	100	5
5	two_wheeler	200	5
6	four_wheeler	750	5
7	four_wheeler	1000	5
8	four_wheeler	1250	5
9	four_wheeler	1500	5
10	four_wheeler	2000	5
11	four_wheeler	750	7.5
12	four_wheeler	1000	7.5
13	four_wheeler	1250	7.5
14	four_wheeler	1500	7.5
15	four_wheeler	2000	7.5

Total rows: 24 of 24

Query complete 00:00:00.053

### 3.Questions :

#### 1. Display allowed vehicles on the highways of state Gujarat.

```
select v_type,v_hight,v_weight  
from high.allows t1  
join (select distinct type  
from high.highway  
where h_no in (select h_no  
from high.passing  
where state_name = 'Gujarat')) t2  
on (t1.h_type = t2.type)
```

Query    Query History

```
1 select v_type,v_hight,v_weight  
2 from high.allows t1  
3 join (select distinct type  
4 from high.highway  
5 where h_no in (select h_no  
6 from high.passing  
7 where state_name = 'Gujarat')) t2  
8 on (t1.h_type = t2.type)
```

Data output    Messages    Notifications

☰ + 📁 ⏮ 🗂️ 🗑️ 📥 🔍

	v_type character varying	v_hight double precision	v_weight double precision
1	two_wheeler	3	100
2	two_wheeler	3	200
3	two_wheeler	5	500
4	two_wheeler	5	100
5	two_wheeler	5	200
6	four_wheeler	5	750

Total rows: 65 of 65    Query complete 00:00:00.787

## 2. Display all the responsible person names for projects starting after the year 2015.

```
select distinct o_name
from high.responsible
where p_no in (
    select p_no
    from high.project
    where starting_date >= '2015-01-01' )
```

The screenshot shows a database query interface with the following sections:

- Query History:** Shows the executed SQL query.
- Data output:** Displays a table of vehicle data.
- Messages:** Shows the status "Query complete 00:00:00.787".

**Query History:**

```
1 select distinct o_name
2 from high.responsible
3 where p_no in (
4 select p_no
5 from high.project
6 where starting_date >= '2015-01-01' )
7
```

**Data output:**

	v_type character varying	v_hight double precision	v_weight double precision
1	two_wheeler	3	100
2	two_wheeler	3	200
3	two_wheeler	5	500
4	two_wheeler	5	100
5	two_wheeler	5	200
6	four_wheeler	5	750

Total rows: 65 of 65    Query complete 00:00:00.787

### 3. Display company name and email whose project has exceeded the deadline.

```
select name,email  
from high.company  
where c_id in (  
    select distinct c_id  
    from high.project  
    where exp_ending_date < (select current_date))
```

Query    Query History

```
1 select name,email  
2   from high.company  
3   where c_id in (  
4       select distinct c_id  
5       from high.project  
6       where exp_ending_date < (select current_date))  
7   |
```

Data output    Messages    Notifications

☰+ ↻ ⌂ ↴ 🗑️ 🔍 ↵

	name character varying	email character varying
1	Bradford Eisold	beisold8@miibei...
2	Shel Rodenhurst	srodenhurst1f@r...
3	Denise Mullins	dmullins1r@grav...

Total rows: 3 of 3    Query complete 00:00:21.029

#### 4. Display all the contact details of the company which is working on project number 10.

```
select name,email,address,phone_no  
from high.company_phone t1  
join  
(select name,email,address,c_id  
from high.company  
where c_id =  
(select c_id  
from high.project  
where p_no = 10)) t2  
on t1.c_id = t2.c_id
```

The screenshot shows a database query interface with the following details:

- Query Tab:** Contains the SQL code for the query.
- Data output Tab:** Contains the results of the query.
- Results:** A table with 2 rows and 5 columns. The columns are: name (character varying), email (character varying), address (character varying), and phone\_no (bigint). The data is as follows:

	name	email	address	phone_no
1	Keir Whittet	kwhittet21@spri...	36 Delaware Alley	9758817932
2	Keir Whittet	kwhittet21@spri...	36 Delaware Alley	9692189823

Total rows: 2 of 2 | Query complete 00:00:00.074

- 5. Display the total cost behind the renovation and the maintenance of all the highways which have length more than 500 km.**

```
select SUM(cost)
from high.completed
where h_no in
(select h_no
from high.highway
where length > 500)
```

The screenshot shows a database query editor interface. At the top, there are tabs for "Query" and "Query History". Below the tabs is the SQL query:

```
1 select SUM(cost)
2 from high.completed
3 where h_no in
4 (select h_no
5 from high.highway
6 where length > 500)
7
```

Below the query is a "Data output" section with tabs for "Messages" and "Notifications". The data output table has one row:

	sum
1	205678743

At the bottom of the interface, there is a status bar with the text "Total rows: 1 of 1" and "Query complete 00:00:00.064".

**6. Display all the project details running by the companies which have secured rank 1 in ‘fast work’ category.**

```
select      *
from high.project t1 join
(select c_id
from high.rank
where award_type='fast work' and position=1) t2
on (t1.c_id = t2.c_id)
```

The screenshot shows a database query interface with two main sections: 'Query' and 'Data output'.

**Query:**

```
1 select      *
2 from high.project t1 join
3 (select c_id
4 from high.rank
5 where award_type='fast work' and position=1) t2
6 on (t1.c_id = t2.c_id)
7 |
```

**Data output:**

	p_no bigint	p_type character varying	h_no bigint	starting_date date	exp_end_date date	exp_cost bigint	e_id bigint	c_id bigint	c_id bigint
1	10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74	74
2	5	construction	100	2011-05-08	2023-01-04	9161092	54	57	57

Total rows: 2 of 2    Query complete 00:00:00.072    Ln 7, Col 1

**7. Display the net worth of the tolls from the highways passing through the state Gujarat.**

```
select SUM(collection) - SUM(spending) AS net_worth  
from high.toll  
where h_no in  
(select h_no  
from high.passing  
where state_name = 'Gujarat')
```

The screenshot shows a database query interface with the following details:

- Query Tab:** Contains the SQL code for the query.
- Data output Tab:** Active tab, showing the results of the query.
- Results:** A table with one row:

	net_worth
1	4327470191
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste, Find, Delete, Import, Export, Refresh).
- Status Bar:** Shows "Total rows: 1 of 1" and "Query complete 00:00:00.100".

**8. Display name,email,award type of the companies which have achieved first three ranks in the year 2020.**

```
select name,position,award_type,email  
from high.company join high.rank on rank.c_id =  
company.c_id  
where year=2020
```

Query    Query History

---

```
1 select name,position,award_type,email  
2 from high.company join high.rank on rank.c_id = company.c_id  
3 where year=2020  
4
```

---

Data output    Messages    Notifications

---

The screenshot shows a database interface with a query editor and a results table. The query editor contains the SQL code provided above. The results table has four columns: name, position, award\_type, and email. The data consists of 12 rows, with the first 10 rows visible and the last two rows partially cut off. The columns are defined as follows:

	name character varying	position bigint	award_type character varying	email character varying
1	Dedie Owlner	1	eco freindly work	dowlnerj@ucsd.edu
2	Gregoor Steynor	1	nice work	gsteynors@guardian.co.uk
3	Wallis Terry	2	cooprative work	wterryw@umich.edu
4	Ellary Matuszak	3	cooprative work	ematuszak10@ox.ac.uk
5	Dot Sinkings	2	fast work	dsinkings18@linkedin.com
6	Dot Sinkings	1	fast work	dsinkings18@linkedin.com
7	Guthrie Server	1	cooprative work	gserver1h@mamy.cz
8	Eberto Poolton	3	eco freindly work	epoolton1n@jugem.jp
9	Buddy Aspole	3	fast work	baspole1v@berkeley.edu
10	Nicolina Daelman	3	nice work	ndaelman1w@amazon.de

Total rows: 12 of 12    Query complete 00:00:02.400

**9. Sort the company rating wise which has been filled with the tender for highway number 105.**

Select c\_id,name,rating

from high.company

where c\_id in

(select c\_id

from high.fills

where h\_no=105)

order by rating Desc

The screenshot shows a database query interface. The top section is a code editor with the following SQL query:

```
11  
12 Select c_id, name, rating  
13 from high.company  
14 where c_id in  
15 (select c_id  
16 from high.fills  
17 where h_no=105)  
18 order by rating Desc  
19  
20
```

The bottom section is a data grid displaying the results of the query. The columns are labeled: c\_id [PK] bigint, name character varying, and rating double precision. The data rows are:

	c_id [PK] bigint	name character varying	rating double precision
1	78	Martie Goor	9.38
2	83	Rutherford Lydiard	8.35
3	50	Rebekkah Alison	8.21
4	49	Gilbert Joslow	8.03
5	21	Cesaro Tremonte	7.25
6	41	Ceciley Brannon	5.69
7	94	Collin Sweetlove	5.22
8	28	Clo Kaiser	5.14

Total rows: 18 of 18    Query complete 00:00:00.150

**10. Provide expected ending date of all the renovation projects on going.**

```
select exp_ending_date  
from high.project  
where p_type='renovation'
```

Query    Query History

```
1 select exp_ending_date  
2 from high.project  
3 where p_type='renovation'  
4
```

Data output    Messages    Notifications

≡+

	exp_ending_date date	🔒
1	2024-09-17	
2	2022-01-23	
3	2025-08-17	
4	2022-08-09	

Total rows: 4 of 4    Query complete 00:00:03.619

## 11. Display distinct state names which having highways.

```
select distinct name  
from high.states
```

Query    Query History

```
1 select distinct name  
2 from high.states|
```

Data output    Messages    Notifications

≡+    ↗    ⏴    📁    🗑    🔍    ⏴

	name	✎
[PK]	character varying	
1	Haryana	
2	Telangana	
3	Madhya Pradesh	
4	Tripura	
5	Tamil Nadu	
6	Uttarakhand	

Total rows: 28 of 28    Query complete 00:00:09.102

**12. Display problem number ,problem status and problem type which are issue between year 2010 and 2015.**

```
select problem_no,problem_status,problem_type  
from high.problems  
where issue_date >= '2010-01-01' and  
issue_date<='2015-12-31'
```

Query    Query History

```
1 select problem_no,problem_status,problem_type  
2 from high.problems  
3 where issue_date >= '2010-01-01' and issue_date<='2015-12-31'
```

Data output    Messages    Notifications

☰ + ↻ ↽ 🗑️ 🔍 ⏪ ⏴ ⏵

	problem_no [PK] bigint	problem_status character varying	problem_type character varying
1	1	unsolved	improper work
2	2	solved	other
3	3	unsolved	improper work
4	4	unsolved	improper work
5	5	unsolved	pending work
6	6	solved	safety
7	7	unsolved	improper work
8	8	solved	pending work
9	9	unsolved	safety
10	10	solved	other

Total rows: 10 of 10    Query complete 00:01:11.651

✓ Successfully run. Tot

**13. Display all toll numbers and its collection of highway number 5.**

```
select toll_no, collection  
from high.toll  
where h_no=5
```

Query    Query History

```
1 select toll_no, collection  
2 from high.toll  
3 where h_no=5
```

Data output    Messages    Notifications

≡+

	toll_no bigint	collection bigint
1	1	2368235
2	2	2063476
3	3	6596968
4	4	2529036
5	5	9449767

Total rows: 5 of 5    Query complete 00:02:45.105

**14. Display contact details and the name of the persons responsible for all the on going projects.**

```
select distinct t3.name,t1.phone_no  
from high.person_phone t1 join high.responsible t2  
on (t1.o_name=t2.o_name and t1.o_id=t2.o_id)  
join high.person t3  
on (t1.o_name=t2.o_name and t1.o_id=t2.o_id)
```

The screenshot shows a database query interface with the following details:

- Query Tab:** Contains the SQL code for the query.
- Data output Tab:** Active tab, showing the results of the query.
- Results:** A table with two columns: "name" (character varying) and "phone\_no" (bigint). The data includes 7 rows of names and phone numbers.
- Total rows:** 1000 of 7000
- Query complete:** 00:00:00.104

	name character varying	phone_no bigint
1	Kayley Ascroft	9416782544
2	Serena Spykings	9105622636
3	Jojo Vlasin	9408900256
4	Rahel Greaser	9205503979
5	Malchy Gingell	9333751270
6	Elianora Melin	9505144767
7	Menard Narraway	9229588049
8	Indie Prendergast	0502746060

**15. Company with name “Willi Eick” fills the tender for the highway number 108.**

```
insert into high.fills
select 108, c_id
from high.company where name like 'Willi Eick';
```

The screenshot shows a database query interface with the following details:

- Query Tab:** Contains the SQL code:

```
1 insert into high.fills
2 select 108, c_id
3 from high.company where name like 'Willi Eick';
4
5
```
- Data output Tab:** Shows the result of the query:

```
INSERT 0 1
```
- Messages Tab:** Displays the message:

```
Query returned successfully in 90 msec.
```
- Status Bar:** Shows "Total rows: 100 of 100" and "Query complete 00:00:00.090".

## 16. Display average spending for a completed project.

```
select avg(cost)  
from high.completed
```

The screenshot shows a database query interface with the following details:

- Query Tab:** The tab is selected and shows the SQL query:

```
1 select avg(cost)  
2 from high.completed  
3
```

A message box displays "Loading..." with a circular progress icon.
- Data output Tab:** This tab is active and shows the result of the query:

	avg	numeric	lock
1	5216855.654		
- Messages Tab:** Shows "Total rows: 1 of 1".
- Notifications Tab:** Shows "Query complete 00:00:27.859".
- Toolbar:** Includes icons for new query, save, download, and refresh.

## 17. Increase the rating of the construction company which has constructed more than 1 highway by 1.

```
update high.company
set rating = rating + 1
where rating < 9 and type = 'construction' and c_id in
(select c_id
from high.completed
group by c_id
having count(*) > 1)
```

The screenshot shows a database query editor interface. At the top, there are tabs for "Query" and "Query History", with "Query" being the active tab. Below the tabs is a code editor containing the SQL query. The code is numbered from 1 to 9. The query itself is:

```
1 update high.company
2 set rating = rating + 1
3 where rating < 9 and type = 'construction' and c_id in
4 (select c_id
5 from high.completed
6 group by c_id
7 having count(*) > 1)
8 |
9
```

Below the code editor, there are three tabs: "Data output", "Messages", and "Notifications", with "Messages" being the active tab. The message area displays the result of the query execution:

UPDATE 5

Query returned successfully in 84 msec.

At the bottom of the interface, there is a status bar with the text "Total rows: 100 of 100" and "Query complete 00:00:00.084".

## 18. Display all company names whose rank was proposed by NHA government employees.

```
select name
from high.company
where c_id in(
    select c_id
    from high.rank
    where emp_id in(
        select emp_id
        from high.government
        where emp_dep = 'NHA'))
```

Query    Query History

---

```
1  select name
2  from high.company
3  where c_id in(
4  select c_id
5  from high.rank
6  where emp_id in(
7  select emp_id
8  from high.government
9  where emp_dep = 'NHA'))
10
```

---

Data output    Messages    Notifications

---

≡+

	name
1	Damon Jillis
2	Tanitansy Trever
3	Ambur Noakes
4	Thaine Sanchis
5	Halley Finlow
6	Maire Leachbogi

Total rows: 25 of 25    Query complete 00:00:50.715

**19. Display all the tender details for the highways having quality rating more than five.**

```
select h_no,exp_cost,exp_time,e_id  
from high.tender  
where h_no in (  
    select h_no  
    from high.highway  
    where quality_rating > 5.00  
)
```

The screenshot shows a database query interface with the following details:

- Query History:** The tab is selected, showing the executed SQL query.
- Query:** The SQL code is displayed in the main area.
- Data Output:** A progress bar indicates "Loading..." over the result table.
- Notifications:** No notifications are present.
- Toolbar:** Includes icons for new query, file operations, and refresh.
- Result Table:** A table with columns: h\_no [PK] bigint, exp\_cost bigint, exp\_time bigint, e\_id bigint. The data rows are:

	h_no [PK] bigint	exp_cost bigint	exp_time bigint	e_id bigint
1	3	4386519	30	6
2	5	6294098	53	71
3	12	5755555	70	89
4	13	2884540	19	11
5	16	4871597	52	6
6	18	3639692	54	21
7	19	6735599	44	93

Total rows: 48 of 48    Query complete 00:00:15.997

## **20. Make a view of ongoing construction project for construction company.**

create or replace view const as

select \*

from high.project

where p\_type = 'construction'

The screenshot shows a database query interface with the following details:

- Query Tab:** The tab is selected, showing the SQL code for creating a view:

```
1 create or replace view const as
2 select *
3 from high.project
4 where p_type = 'construction'
5
```
- Data output Tab:** This tab is visible below the Query tab.
- Messages Tab:** This tab is selected and shows the execution results:

```
CREATE VIEW
```

Query returned successfully in 3 secs 399 msec.
- Notifications Tab:** This tab is visible below the Messages tab.
- Bottom Status Bar:** Shows "Total rows: 156 of 156" and "Query complete 00:00:03.399".

Display details of created view.

```
select *  
from const
```

Query    Query History    Scratch Pad

```
1 create or replace view const as  
2 select *  
3 from high.project  
4 where p_type = 'construction'  
5  
6 select *  
7 from const
```

Data output    Messages    Notifications

	p_no bigint	p_type character varying	h_no bigint	starting_date date	exp_end_date date	exp_cost bigint	e_id bigint	c_id bigint
1	1	construction	96	2011-06-23	2023-10-26	8808832	60	4
2	2	construction	97	2013-08-21	2022-12-29	6219324	11	6
3	3	construction	98	2010-11-18	2025-06-15	8749870	19	21
4	4	construction	99	2016-02-11	2022-10-21	1818379	89	9
5	5	construction	100	2011-05-08	2023-01-04	9161092	54	57

Total rows: 5 of 5    Query complete 00:01:22.691

**Trigger : When a project gets done, it should be deleted from the project table and inserted in the completed table for future purposes. We have used trigger to insert the completed project in the completed table whenever a project gets deleted from the project table.**

```
create or replace function high.ins_del()
returns trigger
language 'plpgsql'
as $body$
declare num bigint; curdate date;
begin
    select count(*) into num from high.completed;
    select current_date into curdate;
    insert into high.completed
values(num,old.p_type,old.starting_date,curdate,NULL,old.h_
no,old.c_id);
    return null;
end;
$body$
```

```
create trigger pro_comp
after delete
on high.project
for each row
execute procedure high.ins_del();
```

```
delete from high.project
where p_no = 9
```

```
select *
from high.project
```

```
select *
from high.completed
```



The screenshot shows a PostgreSQL query editor interface with a "Query" tab selected. The code in the editor is as follows:

```
1 create or replace function high.ins_del()
2 returns trigger
3 language 'plpgsql'
4 as $body$
5 declare num bigint; curdate date;
6 begin
7     select count(*)+1 into num from high.completed;
8     select current_date into curdate;
9     insert into high.completed values(num,old.p_type,old.starting_date,curdate,NULL,old.h_no,old.p_name);
10    return null;
11 end;
12 $body$
13
14 create trigger pro_comp
15 after delete
16 on high.project
17 for each row
18 execute procedure high.ins_del();
```

Query    Query History

```
20 delete from high.project
21 where p_no = 9
22
23 select *
24 from high.project
25
26 select *
27 from high.completed
28
```

Data output    Messages    Notifications

	p_no [PK] bigint	p_type character varying	starting_date date	ending_date date	cost bigint	h_no bigint	c_id bigint
1	111	renovation	2015-05-02	2022-11-15	[null]	2	52
2	1	construction	2010-01-20	2017-02-28	7293316	1	65
3	2	construction	2010-01-28	2017-03-11	2947326	2	70
4	3	construction	2010-04-09	2017-03-19	6552449	3	43
5	4	construction	2010-04-09	2017-05-13	5193118	4	89
6	5	construction	2010-04-14	2017-05-31	5483027	5	64
7	6	construction	2010-05-07	2017-06-11	2346674	6	91
8	7	construction	2010-05-10	2017-06-17	5000012	7	20

Total rows: 111 of 111    Query complete 00:00:00.060

**Function : To check what duration of months a project has been taken, we have created a function called project\_duration(), which loops through the completed project and returns its duration in months.**

```
CREATE OR REPLACE FUNCTION high.project_duration()
RETURNS TABLE (
    project_no bigint,
    duration bigint
) AS
$body$
DECLARE
    var_r record;
BEGIN
    FOR var_r IN (SELECT * FROM high.completed)
    LOOP
        project_no := (var_r.p_no) ;
        duration := (var_r.ending_date - var_r.starting_date)/30;
        RETURN NEXT;
    END LOOP;
END;
$body$
LANGUAGE 'plpgsql';
```

```
SELECT high.project_duration()
```

```
20  SELECT high.project_duration()
```

Data output    Messages    Notifications



	project_duration record	lock
1	(111,91)	
2	(1,86)	
3	(2,86)	
4	(3,84)	
5	(4,86)	
6	(5,86)	
7	(6,86)	
8	(7,86)	

Total rows: 111 of 111

Query complete 00:00:00.072

## Section 5: Project Code with output Screenshots

1. Screenshots of the Website that connects the Database.
2. Code of front-end and back-end

# 1. Screenshots of the Website that connects the Database.

## 1. Home Page

Highway Projects

Project No	<input type="text"/>	Expected ending date	<input type="text"/>	Add
Project type	<input type="text"/>	Expected cost	<input type="text"/>	Update
Highway No	<input type="text"/>	Employee ID	<input type="text"/>	Delete
Starting date	<input type="text"/>	Company ID	<input type="text"/>	Custom Query

Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa
1	construction	96	2011-06-23	2023-10-26	8808832	60	4
2	construction	97	2013-08-21	2022-12-29	6219324	11	6
3	construction	98	2010-11-18	2025-06-15	8749870	19	21
4	construction	99	2016-02-11	2022-10-21	1818379	89	9
5	construction	100	2011-05-08	2023-01-04	9161092	54	57
6	renovation	34	2012-05-22	2024-09-17	4836138	32	93
7	renovation	4	2016-07-11	2022-01-23	7002926	77	64
8	renovation	10	2012-09-12	2025-08-17	7883235	65	8
9	renovation	2	2015-05-02	2022-08-09	8174064	50	52
10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74

## 2.“Projects” table

Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa
1	construction	96	2011-06-23	2023-10-26	8808832	60	4
2	construction	97	2013-08-21	2022-12-29	6219324	11	6
3	construction	98	2010-11-18	2025-06-15	8749870	19	21
4	construction	99	2016-02-11	2022-10-21	1818379	89	9
5	construction	100	2011-05-08	2023-01-04	9161092	54	57
6	renovation	34	2012-05-22	2024-09-17	4836138	32	93
7	renovation	4	2016-07-11	2022-01-23	7002926	77	64
8	renovation	10	2012-09-12	2025-08-17	7883235	65	8
9	renovation	2	2015-05-02	2022-08-09	8174064	50	52
10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74

### 3.Insert a Project

- Project number 35 is inserted ,

Highway Projects																																																																																																
Project No	35	Expected ending date	2025-06-22	Add																																																																																												
Project type	construction	Expected cost	8749888	Update																																																																																												
Highway No	50	Employee ID	20	Delete																																																																																												
Starting date	2010-11-20	Company ID	40	Custom Query																																																																																												
<table border="1"> <thead> <tr> <th>Project</th> <th>Project type</th> <th>Highway No</th> <th>Starting date</th> <th>Expected ending</th> <th>Expected Cost</th> <th>Employee</th> <th>Compa</th> </tr> </thead> <tbody> <tr><td>4</td><td>construction</td><td>99</td><td>2016-02-11</td><td>2022-10-21</td><td>1818379</td><td>89</td><td>9</td></tr> <tr><td>5</td><td>construction</td><td>100</td><td>2011-05-08</td><td>2023-01-04</td><td>9161092</td><td>54</td><td>57</td></tr> <tr><td>6</td><td>renovation</td><td>34</td><td>2012-05-22</td><td>2024-09-17</td><td>4836138</td><td>32</td><td>93</td></tr> <tr><td>7</td><td>renovation</td><td>4</td><td>2016-07-11</td><td>2022-01-23</td><td>7002926</td><td>77</td><td>64</td></tr> <tr><td>8</td><td>renovation</td><td>10</td><td>2012-09-12</td><td>2025-08-17</td><td>7883235</td><td>65</td><td>8</td></tr> <tr><td>9</td><td>renovation</td><td>2</td><td>2015-05-02</td><td>2022-08-09</td><td>8174064</td><td>50</td><td>52</td></tr> <tr><td>10</td><td>maintenance</td><td>5</td><td>2015-12-10</td><td>2023-04-15</td><td>6496398</td><td>58</td><td>74</td></tr> <tr><td>11</td><td>maintenance</td><td>77</td><td>2016-09-15</td><td>2025-04-03</td><td>8609333</td><td>29</td><td>42</td></tr> <tr><td>12</td><td>maintenance</td><td>62</td><td>2014-11-22</td><td>2026-06-05</td><td>4395020</td><td>19</td><td>71</td></tr> <tr><td>35</td><td>construction</td><td>50</td><td>2010-11-20</td><td>2025-06-22</td><td>8749888</td><td>20</td><td>40</td></tr> </tbody> </table>									Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa	4	construction	99	2016-02-11	2022-10-21	1818379	89	9	5	construction	100	2011-05-08	2023-01-04	9161092	54	57	6	renovation	34	2012-05-22	2024-09-17	4836138	32	93	7	renovation	4	2016-07-11	2022-01-23	7002926	77	64	8	renovation	10	2012-09-12	2025-08-17	7883235	65	8	9	renovation	2	2015-05-02	2022-08-09	8174064	50	52	10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74	11	maintenance	77	2016-09-15	2025-04-03	8609333	29	42	12	maintenance	62	2014-11-22	2026-06-05	4395020	19	71	35	construction	50	2010-11-20	2025-06-22	8749888	20	40
Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa																																																																																									
4	construction	99	2016-02-11	2022-10-21	1818379	89	9																																																																																									
5	construction	100	2011-05-08	2023-01-04	9161092	54	57																																																																																									
6	renovation	34	2012-05-22	2024-09-17	4836138	32	93																																																																																									
7	renovation	4	2016-07-11	2022-01-23	7002926	77	64																																																																																									
8	renovation	10	2012-09-12	2025-08-17	7883235	65	8																																																																																									
9	renovation	2	2015-05-02	2022-08-09	8174064	50	52																																																																																									
10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74																																																																																									
11	maintenance	77	2016-09-15	2025-04-03	8609333	29	42																																																																																									
12	maintenance	62	2014-11-22	2026-06-05	4395020	19	71																																																																																									
35	construction	50	2010-11-20	2025-06-22	8749888	20	40																																																																																									

### 4. Edit Project

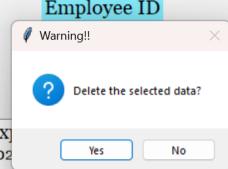
- Update project type construction to renovation of project number 35.

Highway Projects																																																																																																
Project No	35	Expected ending date	2025-06-22	Add																																																																																												
Project type	renovation	Expected cost	8749888	Update																																																																																												
Highway No	50	Employee ID	20	Delete																																																																																												
Starting date	2010-11-20	Company ID	40	Custom Query																																																																																												
<table border="1"> <thead> <tr> <th>Project</th> <th>Project type</th> <th>Highway No</th> <th>Starting date</th> <th>Expected ending</th> <th>Expected Cost</th> <th>Employee</th> <th>Compa</th> </tr> </thead> <tbody> <tr><td>4</td><td>construction</td><td>99</td><td>2016-02-11</td><td>2022-10-21</td><td>1818379</td><td>89</td><td>9</td></tr> <tr><td>5</td><td>construction</td><td>100</td><td>2011-05-08</td><td>2023-01-04</td><td>9161092</td><td>54</td><td>57</td></tr> <tr><td>6</td><td>renovation</td><td>34</td><td>2012-05-22</td><td>2024-09-17</td><td>4836138</td><td>32</td><td>93</td></tr> <tr><td>7</td><td>renovation</td><td>4</td><td>2016-07-11</td><td>2022-01-23</td><td>7002926</td><td>77</td><td>64</td></tr> <tr><td>8</td><td>renovation</td><td>10</td><td>2012-09-12</td><td>2025-08-17</td><td>7883235</td><td>65</td><td>8</td></tr> <tr><td>9</td><td>renovation</td><td>2</td><td>2015-05-02</td><td>2022-08-09</td><td>8174064</td><td>50</td><td>52</td></tr> <tr><td>10</td><td>maintenance</td><td>5</td><td>2015-12-10</td><td>2023-04-15</td><td>6496398</td><td>58</td><td>74</td></tr> <tr><td>11</td><td>maintenance</td><td>77</td><td>2016-09-15</td><td>2025-04-03</td><td>8609333</td><td>29</td><td>42</td></tr> <tr><td>12</td><td>maintenance</td><td>62</td><td>2014-11-22</td><td>2026-06-05</td><td>4395020</td><td>19</td><td>71</td></tr> <tr><td>35</td><td>renovation</td><td>50</td><td>2010-11-20</td><td>2025-06-22</td><td>8749888</td><td>20</td><td>40</td></tr> </tbody> </table>									Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa	4	construction	99	2016-02-11	2022-10-21	1818379	89	9	5	construction	100	2011-05-08	2023-01-04	9161092	54	57	6	renovation	34	2012-05-22	2024-09-17	4836138	32	93	7	renovation	4	2016-07-11	2022-01-23	7002926	77	64	8	renovation	10	2012-09-12	2025-08-17	7883235	65	8	9	renovation	2	2015-05-02	2022-08-09	8174064	50	52	10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74	11	maintenance	77	2016-09-15	2025-04-03	8609333	29	42	12	maintenance	62	2014-11-22	2026-06-05	4395020	19	71	35	renovation	50	2010-11-20	2025-06-22	8749888	20	40
Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa																																																																																									
4	construction	99	2016-02-11	2022-10-21	1818379	89	9																																																																																									
5	construction	100	2011-05-08	2023-01-04	9161092	54	57																																																																																									
6	renovation	34	2012-05-22	2024-09-17	4836138	32	93																																																																																									
7	renovation	4	2016-07-11	2022-01-23	7002926	77	64																																																																																									
8	renovation	10	2012-09-12	2025-08-17	7883235	65	8																																																																																									
9	renovation	2	2015-05-02	2022-08-09	8174064	50	52																																																																																									
10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74																																																																																									
11	maintenance	77	2016-09-15	2025-04-03	8609333	29	42																																																																																									
12	maintenance	62	2014-11-22	2026-06-05	4395020	19	71																																																																																									
35	renovation	50	2010-11-20	2025-06-22	8749888	20	40																																																																																									

## 5.Delete Completed Project

- Delete the tuple which has project number 35,

**Highway Projects**

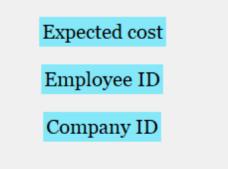
<b>Project No</b>	35	<b>Expected ending date</b>	2025-06-22	 Warning!! Delete the selected data? <input type="button" value="Yes"/> <input type="button" value="No"/>
<b>Project type</b>	renovation	<b>Expected cost</b>	8749888	
<b>Highway No</b>	50	<b>Employee ID</b>	20	
<b>Starting date</b>	2010-11-20		40	

Project	Project type	Highway No	Starting date	Ex	Cost	Employee	Compa
4	construction	99	2016-02-11	202	89	9	57
5	construction	100	2011-05-08	2023-01-04	9161092	54	93
6	renovation	34	2012-05-22	2024-09-17	4836138	32	64
7	renovation	4	2016-07-11	2022-01-23	7002926	77	8
8	renovation	10	2012-09-12	2025-08-17	7883235	65	52
9	renovation	2	2015-05-02	2022-08-09	8174064	50	74
10	maintenance	5	2015-12-10	2023-04-15	6496398	29	42
11	maintenance	77	2016-09-15	2025-04-03	8609333	19	71
12	maintenance	62	2014-11-22	2026-06-05	4395020	8749888	20
35	renovation	50	2010-11-20	2025-06-22		40	

Add
Update
Delete
Custom Query
Reset
Select

**Highway Projects**

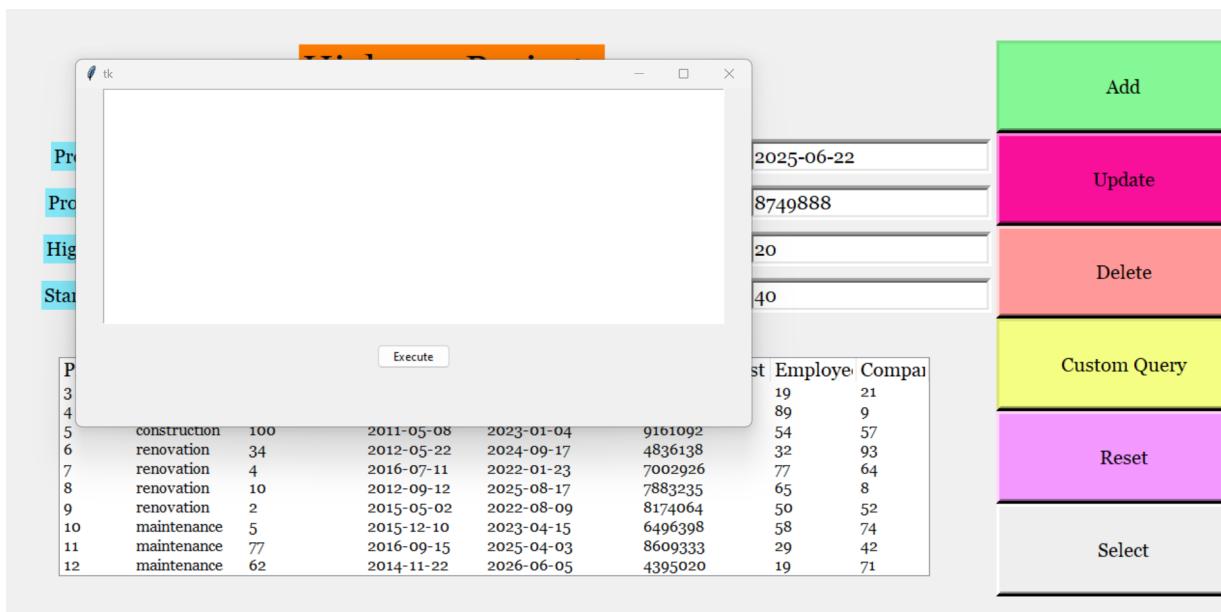
<b>Project No</b>	35	<b>Expected ending date</b>	2025-06-22	 Warning!! Delete the selected data? <input type="button" value="Yes"/> <input type="button" value="No"/>
<b>Project type</b>	renovation	<b>Expected cost</b>	8749888	
<b>Highway No</b>	50	<b>Employee ID</b>	20	
<b>Starting date</b>	2010-11-20	<b>Company ID</b>	40	

Project	Project type	Highway No	Starting date	Expected ending	Expected Cost	Employee	Compa
3	construction	98	2010-11-18	2025-06-15	8749870	19	21
4	construction	99	2016-02-11	2022-10-21	1818379	89	9
5	construction	100	2011-05-08	2023-01-04	9161092	54	57
6	renovation	34	2012-05-22	2024-09-17	4836138	32	93
7	renovation	4	2016-07-11	2022-01-23	7002926	77	64
8	renovation	10	2012-09-12	2025-08-17	7883235	65	8
9	renovation	2	2015-05-02	2022-08-09	8174064	50	52
10	maintenance	5	2015-12-10	2023-04-15	6496398	58	74
11	maintenance	77	2016-09-15	2025-04-03	8609333	29	42
12	maintenance	62	2014-11-22	2026-06-05	4395020	19	71

Add
Update
Delete
Custom Query
Reset
Select

## 6. Custom Query window



**Query 1 : Display all completed projects which have project number less than equal to 10.**

The screenshot shows a query editor window with a code input area and a results pane. The code input area contains:

```
select *
from high.completed
where p_no <= 10
```

The results pane displays the output of the query:

```
-----  
1 construction 2010-01-20 2017-02-28 7293316 1  
2 construction 2010-01-28 2017-03-11 2947326 2  
3 construction 2010-04-09 2017-03-19 6552449 3  
4 construction 2010-04-09 2017-05-13 5193118 4  
5 construction 2010-04-14 2017-05-31 5483027 5  
6 construction 2010-05-07 2017-06-11 2346674 6  
7 construction 2010-05-12 2017-06-17 5290913 7  
8 construction 2010-06-21 2017-06-24 2623254 8  
9 construction 2010-07-26 2017-06-29 3003637 9  
10 construction 2010-08-06 2017-06-30 4859773 10  
-----
```

At the bottom of the results pane is an 'Execute' button.

## Query 2 : Display company id and name which have rating greater than or equal to nine.

```
select c_id, name  
from high.company  
where rating >= 9
```

```
-- -----  
7 Calhoun Bedle  
9 Bradford Eisold  
25 Jerrine Pennone  
27 Thekla Duetsche  
36 Hayward Capon  
38 Marketa Morbey  
75 Lowe Biesty  
78 Martie Goor  
98 Isidor Eddolls  
99 Von Ginnally  
-- -----
```

## 2. Code of front-end and back-end

```
import psycopg2
import psycopg2.extras
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import tkinter as tk
from tabulate import tabulate
table_name = "high.project"

#connection for pgadmin
def connection():
    conn = psycopg2.connect(
        host = 'localhost',
        dbname = 'hd',
        user = 'postgres',
        password = 'admin',
        port = 5432)
    return conn

def refreshTable():
    for data in my_tree.get_children():
        my_tree.delete(data)

    for array in read():
        my_tree.insert(parent="", index='end', iid=array, text="", values=(array), tag="orow")

    my_tree.tag_configure('orow', background="#EEEEEE", font=('Georgia', 12))
    my_tree.grid(row=7, column=0, columnspan=8, rowspan=10, padx=50, pady=40)

root = Tk()
root.title("Highway Projects")
root.geometry("200x200")
my_tree = ttk.Treeview(root)
```

```
#placeholders for entry
ph1 = tk.StringVar()
ph2 = tk.StringVar()
ph3 = tk.StringVar()
ph4 = tk.StringVar()
ph5 = tk.StringVar()
ph6 = tk.StringVar()
ph7 = tk.StringVar()
ph8 = tk.StringVar()

#placeholder set value function
def setph(word,num):
    if num ==1:
        ph1.set(word)
    if num ==2:
        ph2.set(word)
    if num ==3:
        ph3.set(word)
    if num ==4:
        ph4.set(word)
    if num ==5:
        ph5.set(word)
    if num ==6:
        ph6.set(word)
    if num ==7:
        ph7.set(word)
    if num ==8:
        ph8.set(word)

def read():
    conn = connection()
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM "+table_name)
    results = cursor.fetchall()
    conn.commit()
    conn.close()
```

```

return results

def add():
    p_no = str(p_noEntry.get())
    p_type = str(p_typeEntry.get())
    h_no = str(h_noEntry.get())
    starting_date = str(starting_dateEntry.get())
    exp_ending_date = str(exp_ending_dateEntry.get())
    exp_cost = str(exp_costEntry.get())
    e_id = str(e_idEntry.get())
    c_id = str(c_idEntry.get())

    if (p_no == "" or p_no == " ") or (p_type == "" or p_type == " ") or (h_no == "" or h_no == " ") or
    (starting_date == "" or starting_date == " ") or (exp_ending_date == "" or exp_ending_date == " ")
    or (exp_cost == "" or exp_cost == " ") or (e_id == "" or e_id == " ") or (c_id == "" or c_id == " "):
        messagebox.showinfo("Error", "Please fill up the blank entry")
        return
    else:
        try:
            conn = connection()
            cursor = conn.cursor()
            cursor.execute("INSERT INTO "+table_name+" VALUES
            ("+p_no+","+p_type+","+h_no+","+starting_date+","+exp_ending_date+","+exp_cost+","+e_i
            d+","+c_id+") ")
            conn.commit()
            conn.close()
        except:
            messagebox.showinfo("Error", "Project No already exist")
            return

    refreshTable()

def reset():
    decision = messagebox.askquestion("Warning!!", "Delete all data?")
    if decision != "yes":
        return
    else:

```

```

try:
    conn = connection()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM "+table_name)
    conn.commit()
    conn.close()

except:
    messagebox.showinfo("Error", "Sorry an error occurred")
    return

refreshTable()

def delete():
    decision = messagebox.askquestion("Warning!!", "Delete the selected data?")
    if decision != "yes":
        return
    else:
        selected_item = my_tree.selection()[0]
        deleteData = str(my_tree.item(selected_item)['values'][0])
        try:
            conn = connection()
            cursor = conn.cursor()
            cursor.execute("DELETE FROM "+table_name+" WHERE p_no='"+str(deleteData)+"'")
            conn.commit()
            conn.close()

        except:
            messagebox.showinfo("Error", "Sorry an error occurred")
            return

refreshTable()

def select():
    try:
        selected_item = my_tree.selection()[0]
        p_no = str(my_tree.item(selected_item)['values'][0])
        p_type = str(my_tree.item(selected_item)['values'][1])
        h_no = str(my_tree.item(selected_item)['values'][2])
        starting_date = str(my_tree.item(selected_item)['values'][3])
    
```

```

exp_ending_date = str(my_tree.item(selected_item)['values'][4])
exp_cost = str(my_tree.item(selected_item)['values'][5])
e_id = str(my_tree.item(selected_item)['values'][6])
c_id = str(my_tree.item(selected_item)['values'][7])

setph(p_no,1)
setph(p_type,2)
setph(h_no,3)
setph(starting_date,4)
setph(exp_ending_date,5)
setph(exp_cost,6)
setph(e_id,7)
setph(c_id,8)

except:
    messagebox.showinfo("Error", "Please select a data row")

def get_input(txt, lbl):
    inp = txt.get(1.0, "end-1c")
    try:
        conn = connection()
        cursor = conn.cursor()
        cursor.execute(inp)
        op = cursor.fetchall()
        conn.commit()
        conn.close()
    except:
        messagebox.showinfo("Error", "Sorry an error occurred")
    lbl.config(text="" + tabulate(op))

def search():
    win=Tk()
    # Set the geometry
    win.geometry("700x350")
    # Add a text widget
    text=Text(win, width=80, height=15)
    text.insert(END, "")
    text.pack()

```

```

# Create a Label widget
label=Label(win, text="", font=('Calibri 10'))
label.pack()

# Create a button to get the text input
b=ttk.Button(win, text="Execute", command= lambda : get_input(text,label))
b.pack()

def update():
    selectedp_no = ""

    try:
        selected_item = my_tree.selection()[0]
        selectedp_no = str(my_tree.item(selected_item)['values'][0])
    except:
        messagebox.showinfo("Error", "Please select a data row")

    p_no = str(p_noEntry.get())
    p_type = str(p_typeEntry.get())
    h_no = str(h_noEntry.get())
    starting_date = str(starting_dateEntry.get())
    exp_ending_date = str(exp_ending_dateEntry.get())
    exp_cost = str(exp_costEntry.get())
    e_id = str(e_idEntry.get())
    c_id = str(c_idEntry.get())

    if (p_no == "" or p_no == " ") or (p_type == "" or p_type == " ") or (h_no == "" or h_no == " ") or
    (starting_date == "" or starting_date == " ") or (exp_ending_date == "" or exp_ending_date == " ") or
    (exp_cost == "" or exp_cost == " ") or (e_id == "" or e_id == " ") or (c_id == "" or c_id == " "):
        messagebox.showinfo("Error", "Please fill up the blank entry")
        return

    else:
        try:
            conn = connection()
            cursor = conn.cursor()
            cursor.execute("UPDATE "+table_name+" SET p_no='"++
            p_no+"', p_type='"++
            p_type+"', h_no='"++
            h_no+"', starting_date='"++

```

```

starting_date+"", exp_ending_date)+"+
exp_ending_date+"", exp_cost)+"+
exp_cost+"", e_id)+"+
e_id+"", c_id)+"+
c_id+" WHERE p_no)+"+
selectedp_no+" ")
conn.commit()
conn.close()

except:
    messagebox.showinfo("Error", "Project No already exist")
    return

refreshTable()

label = Label(root, text="Highway Projects", font=('Georgia', 30),bg="#FD7B00")
label.grid(row=0, column=0, columnspan=7, rowspan=2, padx=50, pady=40)

p_noLabel = Label(root, text="Project No", font=('Georgia', 15),bg="#84E8F8")
p_typeLabel = Label(root, text="Project type", font=('Georgia', 15),bg="#84E8F8")
h_noLabel = Label(root, text="Highway No", font=('Georgia', 15),bg="#84E8F8")
starting_dateLabel = Label(root, text="Starting date", font=('Georgia', 15),bg="#84E8F8")
exp_ending_dateLabel = Label(root, text="Expected ending date", font=('Georgia',
15),bg="#84E8F8")
exp_costLabel = Label(root, text="Expected cost", font=('Georgia', 15),bg="#84E8F8")
e_idLabel = Label(root, text="Employee ID", font=('Georgia', 15),bg="#84E8F8")
c_idLabel = Label(root, text="Company ID", font=('Georgia', 15),bg="#84E8F8")

p_noLabel.grid(row=3, column=0, columnspan=1, padx=50, pady=5)
p_typeLabel.grid(row=4, column=0, columnspan=1, padx=50, pady=5)
h_noLabel.grid(row=5, column=0, columnspan=1, padx=50, pady=5)
starting_dateLabel.grid(row=6, column=0, columnspan=1, padx=50, pady=5)
exp_ending_dateLabel.grid(row=3, column=4, columnspan=1, padx=50, pady=5)
exp_costLabel.grid(row=4, column=4, columnspan=1, padx=50, pady=5)
e_idLabel.grid(row=5, column=4, columnspan=1, padx=50, pady=5)
c_idLabel.grid(row=6, column=4, columnspan=1, padx=50, pady=5)

p_noEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph1)

```

```

p_typeEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph2)
h_noEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph3)
starting_dateEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph4)
exp_ending_dateEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph5)
exp_costEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph6)
e_idEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph7)
c_idEntry = Entry(root, width=20, bd=5, font=('Georgia', 15), textvariable = ph8)

p_noEntry.grid(row=3, column=1, columnspan=3, padx=5, pady=0)
p_typeEntry.grid(row=4, column=1, columnspan=3, padx=5, pady=0)
h_noEntry.grid(row=5, column=1, columnspan=3, padx=5, pady=0)
starting_dateEntry.grid(row=6, column=1, columnspan=3, padx=5, pady=0)
exp_ending_dateEntry.grid(row=3, column=5, columnspan=3, padx=5, pady=0)
exp_costEntry.grid(row=4, column=5, columnspan=3, padx=5, pady=0)
e_idEntry.grid(row=5, column=5, columnspan=3, padx=5, pady=0)
c_idEntry.grid(row=6, column=5, columnspan=3, padx=5, pady=0)

addBtn = Button(
    root, text="Add", padx=65, pady=25, width=10,
    bd=5, font=('Georgia', 15), bg="#84F894", command=add)
updateBtn = Button(
    root, text="Update", padx=65, pady=25, width=10,
    bd=5, font=('Georgia', 15), bg="#F9109A", command=update)
deleteBtn = Button(
    root, text="Delete", padx=65, pady=25, width=10,
    bd=5, font=('Georgia', 15), bg="#FF9999", command=delete)
searchBtn = Button(
    root, text="Custom Query", padx=65, pady=25, width=10,
    bd=5, font=('Georgia', 15), bg="#F4FE82", command=search)
resetBtn = Button(
    root, text="Reset", padx=65, pady=25, width=10,
    bd=5, font=('Georgia', 15), bg="#F398FF", command=reset)
selectBtn = Button(
    root, text="Select", padx=65, pady=25, width=10,
    bd=5, font=('Georgia', 15), bg="#EEEEEE", command=select)

addBtn.grid(row=1, column=8, columnspan=1, rowspan=2)

```

```
updateBtn.grid(row=3, column=8, columnspan=1, rowspan=2)
deleteBtn.grid(row=5, column=8, columnspan=1, rowspan=2)
searchBtn.grid(row=7, column=8, columnspan=1, rowspan=2)
resetBtn.grid(row=9, column=8, columnspan=1, rowspan=2)
selectBtn.grid(row=11, column=8, columnspan=1, rowspan=2)

style = ttk.Style()
style.configure("Treeview.Heading", font=('Georgia', 15),)

my_tree['columns'] = ("Project No","Project type","Highway No","Starting date","Expected ending date","Expected cost","Employee ID","Company ID")

my_tree.column("#0", width=0, stretch=NO)
my_tree.column("Project No", anchor=W, width=75)
my_tree.column("Project type", anchor=W, width=150)
my_tree.column("Highway No", anchor=W, width=75)
my_tree.column("Starting date", anchor=W, width=150)
my_tree.column("Expected ending date", anchor=W, width=150)
my_tree.column("Expected cost", anchor=W, width=150)
my_tree.column("Employee ID", anchor=W, width=75)
my_tree.column("Company ID", anchor=W, width=75)

my_tree.heading("Project No", text="Project No", anchor=W)
my_tree.heading("Project type", text="Project type", anchor=W)
my_tree.heading("Highway No", text="Highway No", anchor=W)
my_tree.heading("Starting date", text="Starting date", anchor=W)
my_tree.heading("Expected ending date", text="Expected ending date", anchor=W)
my_tree.heading("Expected cost", text="Expected Cost", anchor=W)
my_tree.heading("Employee ID", text="Employee ID", anchor=W)
my_tree.heading("Company ID", text="Company ID", anchor=W)

refreshTable()

root.mainloop()
```