Wood Product Lifecycle Management System

Group -55
202412108-Jenil Soni
202412090-Ayush Shah

Institute
DAIICT

Date of submission
13/11/2024

Chapter 1-Software Requirements Specification (SRS)

1 Problem Description

1 Region and Tree Management

Description

This fundamental element entails monitoring and maintaining information about forest areas and the trees that inhabit them. It entails keeping thorough records of every forest region, monitoring the specifics of each tree, and documenting harvest data. Effective resource management and sustainable forestry techniques depend on this basis.

Key Issues

- Ensuring that information on forest regions, such as geographic coordinates, total area, soil conditions, and climate data, is accurate and reliable.
- Tree tracking is the process of keeping accurate and thorough records of the types, ages, growth rates, and overall health of trees as well as any possible illnesses or pests.
- Harvest Records: Precisely recording the quantity of trees felled, the dates of harvest, the techniques employed, and the loggers in charge of each operation.

- Efficiency of Queries: Creating and refining a strong querying system that can efficiently and precisely retrieve data on tree populations, harvest records, and logger activities throughout various time periods and regions.
- Environmental Impact Assessment: Combining the ability to evaluate and forecast how harvesting operations may affect nearby ecosystems.

2. Harvest Tracking

Description

This entity painstakingly records every aspect of each harvest event. It documents the exact quantity and kind of trees cut down, the logger who was in charge, and the location of the incident. Furthermore, it monitors the amount and calibre of raw wood produced following every harvest, serving as a vital component in the chain of distribution.

Key Issues

- Event Documentation: Ensuring thorough and accurate recording of every harvest event, including the tools utilized, the climate, and any noteworthy incidents.
- Wood tracking: Putting in place a reliable system to measure and monitor the quantity of raw wood produced with accuracy, and giving each batch a unique identity to ensure traceability.
- Logger accountability involves keeping thorough records of the loggers in charge of every harvest event, together with information about their credentials, performance indicators, and commitment to sustainable harvesting methods.
- During harvest events, real-time data collection systems should be developed in order to increase accuracy and facilitate quick decision-making.
- Yield Prediction: Using historical data and present forest conditions to anticipate harvest yields through predictive analytics.

3. Wood Processing

Description

This part handles the complex conversion of raw wood into different processed forms. To guarantee quality control and effective resource use along the processing chain, it keeps track of several processing phases and documents which facilities are in charge of each transformation step.

Key Issues

- Steps of Processing: Precisely monitoring and recording every phase of wood processing, encompassing particular treatments, tools employed, and quality assurance protocols.
- Facility Management: Keeping thorough documentation of all wood processing facilities' capacities, abilities, efficiency measures, and adherence to industry norms.
- Query Capabilities: Developing an advanced query system to swiftly ascertain which facilities handled which batches of wood, the specifics of each processing step, and the characteristics of the wood that were ultimately produced.
- Quality Control: Implementing rigorous quality control measures at each processing stage and tracking any defects or issues.
- Resource Optimization: Analyzing processing data to optimize resource allocation, minimize waste, and improve overall efficiency in the wood processing chain.

4. Manufacturing

Description

This organisation monitors the flow of processed wood to producers and the subsequent production of wood goods. It keeps thorough records of the manufacturers that receive processed wood, the amounts that are used, and the particular wood products that are produced. This element is essential for figuring out where products come from and making sure that resources for processed wood are used effectively.

Key Issues

- Wood Allocation: Keeping track of the kinds, grades, and quantities of processed wood that are sent to various producers in an accurate and timely manner.
- Product tracking is the process of keeping thorough records of wood products, which include information on pricing, quality grades, specs, and the maker of each product.
- Inventory management: Putting in place a sophisticated system to monitor the quantities of completed goods and processed wood at various manufacturers' locations.
- Production Efficiency: Analyzing manufacturing data to identify bottlenecks, improve production efficiency, and optimize resource utilization.
- Sustainability Metrics: Incorporating sustainability metrics into the manufacturing process, tracking factors such as energy use, waste generation, and carbon footprint.

5. Customer Purchases

Description

This component tracks all customer purchases of wood products. It records detailed information about each transaction, including customer identity, specific products purchased, quantities, prices, and dates of purchase. This data is crucial for sales analysis, customer relationship management, and tracking product lifecycles.

Key Problems

- Purchase Records: Preserving complete, safe, and accurate records of every purchase made by customers while adhering to data protection laws.
- Customer data management is the process of keeping accurate and comprehensive records of a customer's preferences, purchasing history, and contact information.
- Sales Analysis: Creating strong analytical tools to examine sales information, spot patterns, project demand, and learn about the preferences and actions of customers.
- Customer relationship management (CRM) systems and purchase data integration: This technique combines both to enhance customer support and marketing tactics.
- Product Lifecycle Tracking: Setting up mechanisms to monitor a sold product's lifecycle, either for warranty or recycling purposes later on.

6. Recycling

Description

This final component manages the recycling of wood products at the end of their lifecycle. It tracks recycling events in detail, including information about the products being recycled, the specific recycling methods employed, and the contractors responsible for carrying out the recycling process. This entity is crucial for promoting sustainability and circular economy principles in the wood industry.

Key Problems

- Documentation of Recycling: Ensuring thorough and precise documentation of all recycling activities, encompassing the source of recycled materials, the recycling techniques employed, and the final materials or products.
- Contractor management is the process of keeping thorough records of all contractors who handle recycling, including information about their qualifications, performance standards, and adherence to environmental laws.
- Query Efficiency: Creating a system that efficiently queries data to find recycled products fast, connect them to the original consumers and recycling providers, and provide environmental impact and recycling efficiency reports.
- Material Recovery Tracking: Putting in place mechanisms to monitor the kinds and amounts of materials that are recovered during recycling procedures, as well as their eventual usage or sale.
- Creating plans and procedures to include recycled materials back into the production process is known as "circular economy integration," and it aims to create a closed-loop system in the wood sector.

2. Requirement Collection

Background Reading

Forest Regions

Link- https://pwonlyias.com/forest-in-india/#types-of-forest-in-india

Type - Article

Author - PWOnlyIAS

Date - February 27, 2024

1. Tropical Evergreen Forests: Dense and green year-round, found in regions like the Western Ghats and the Andaman Islands.
2. Deciduous Forests: Monsoon forests, shedding leaves in dry seasons, common in the Himalayas and Central India.
3. Thorn Forests: Found in arid areas, with sparse vegetation like Babul and Neem.
4. Montane & Alpine Forests: Spread across the Himalayas, supporting species like Rhododendron.
5. Mangroves: Crucial for coastal protection, found in areas like the Sundarbans.

Tree Species

Link - https://www.naturalmangoes.com/10-best-timber-tree-varieties-in-india/
Type- Article
Author - Unknown
Date - November 12 , 2022

India is home to a diverse range of timber species, which are widely used in construction, furniture making, and various other industries. Key timber species in India include:

1. Teak (Tectona grandis): Known for its durability and resistance to termites, teak is one of the most prized hardwoods in India. It is commonly found in both natural and plantation forests.
2. Sal (Shorea robusta): Sal is another significant timber species used in construction due to its strength and resistance to decay. It is harvested primarily from natural forests.
3. Neem (Azadirachta indica): A fast-growing tree that is resistant to pests and termites, neem is frequently used in furniture making and construction. Its pest-resistant qualities make it particularly valuable.
4. Mahogany (Swietenia macrophylla): Valued for its rich reddish-brown color and ease of working, mahogany is used in high-end furniture and interior decoration. It is found in tropical forests, especially in South India.
5. Eucalyptus: This fast-growing tree is commonly planted for timber production in India and is used in paper, pulp, and plywood industries.
6. Rosewood (Dalbergia latifolia): Known for its fine grain and deep red color, rosewood is a hardwood popular for making durable furniture and decorative items.

The growth cycle of a tree

Link - https://ecotree.green/en/blog/the-life-cycle-of-a-tree
Type - Article
Author - Kia Eg
Date - Feb 8 , 2022

1. Seed Germination: A tree's life begins with a seed, which, when exposed to proper moisture and warmth, germinates. The seed develops roots (radicle) and a shoot, emerging as a seedling from the soil.
2. Seedling Stage: At this stage, the young tree starts producing leaves, relying on nutrients stored in the seed. It is vulnerable to external factors like pests, weather, and disease
3. Sapling Stage: After establishing itself, the tree becomes a sapling, typically about 3 feet tall. This stage is marked by rapid growth, but the tree is not yet mature enough to produce seeds or flowers.
4. Mature Tree: The tree reaches sexual maturity and begins to reproduce by producing flowers, fruit, or seeds. This phase can last decades or even centuries depending on the species.
5. Elderly Decline: In the final phase, the tree's growth slows, and it becomes more susceptible to diseases. Decaying trees play an important ecological role, providing habitat and nutrients to various organisms.

Techniques for Forest Regeneration

Link-https://synopsisias.com/blog/natural-and-artificial-regeneration-of-forests-forestry-optional-for-upsc-ifs-category?category_slug=silviculture-general

Type-Article

Author - Synopsis IAS

Date - October 12 , 2023

1. Natural Regeneration: Allowing forests to regenerate naturally through seed dispersal and growth without human intervention. This method is cost-effective and preserves genetic diversity.

2. Artificial Regeneration: Human-assisted methods include planting tree seedlings, direct seeding, or using vegetative propagation techniques like cloning or grafting.

3. Site Preparation: Clearing debris, controlling competing vegetation, and improving soil conditions can enhance the success of regeneration efforts.

4. Genetic Selection: Choosing tree species and individuals with desirable traits, such as resistance to pests or adaptability to changing climates, can improve the long-term viability of regenerated forests.

Wood processing

Link-https://www.deskera.com/blog/wood-manufacturing-process/?formCode=MG0AV3

Type of document-Article
Author - Rhema Hans

Wood Processing Stages:

- Tree Selection and Harvesting: Choosing the right trees and cutting them down.
- Sawing: Cutting logs into rough lumber.
- Drying: Reducing the moisture content to prevent warping and cracking.
- Planing: Smoothing the wood surface to a consistent thickness.
- Moulding: Shaping wood into specific profiles or mouldings.
- Sanding: Smoothing the wood surface to a fine finish.
- Grading: Inspecting and sorting wood based on quality and defects.
- Treating: Applying chemicals to protect wood from decay, insects, and fungi.
- Finishing: Applying coatings to enhance the appearance and durability of wood.

Tools used for processing

Link-https://disciplines.ng/wood-production-tools-and-equipment/?formCode=MG0AV3

Author-Unknown

Tree Selection and Harvesting: Feller bunchers and skidders are used to cut and transport trees.

Sawing: Headrig saws, band saws, and circular saws cut logs into rough lumber.

Drying: Kilns and dehumidifiers reduce moisture content to prevent warping and cracking.

Planing: Planers and jointers smooth the wood surface to a consistent thickness.

Moulding: Moulders and shapers shape wood into specific profiles or mouldings.

Sanding: Belt sanders, orbital sanders, and hand sanding blocks smooth the wood surface to a fine finish.

Grading: Grading machines and visual inspection tools inspect and sort wood based on quality and defects.

Treating: Pressure treatment equipment and dip tanks apply chemicals to protect wood from decay, insects, and fungi.

Finishing: Sprayers, brushes, and rollers apply coatings to enhance the appearance and durability of wood.

Joinery: Routers, chisels, and biscuit joiners create joints to connect wood pieces together.

Cutting: Table saws, miter saws, and jigsaws cut wood into specific sizes and shape

Wood Quality

Link-https://www.deskera.com/blog/best-practices-for-wood-manufacturing-quality-control/#key-elements-of-wood-manufacturing-quality-control

Type-Article
Author - Rahema Hans

1 Key Elements of Wood Manufacturing Quality Control

Raw Material Inspection: The first step is inspecting wood logs or boards for defects like cracks, knots, and insect damage to ensure they are suitable for manufacturing.
Moisture Content Control: Controlling moisture is crucial as it affects the wood's quality. Excess moisture can cause warping and decay, while too little can make the wood brittle.
Quality Control at Each Stage: Continuous quality checks during each manufacturing stage—sawing, planing, sanding, and finishing—help identify and correct defects early.
Equipment Maintenance: Regular maintenance and calibration of manufacturing equipment ensure consistent product quality.
Testing and Certification: Products are tested and certified to meet industry standards, building customer trust and ensuring product safety.
Continuous Improvement: Ongoing evaluation and enhancement of manufacturing processes, based on customer feedback and internal assessments, help improve product quality over time.

Types of Defects:

1. Knots: These are areas where branches grew, potentially weakening the wood and affecting its appearance.
2. Checks and Splits: Cracks caused by uneven drying or moisture changes, which can reduce the wood's strength.
3. Warping and Twisting: Resulting from improper drying or storage, leading to wood that is not flat or straight.
4. Grain Defects: Occur when the wood is cut improperly, misaligning growth rings and weakening the product.
5. Decay and Insect Damage: Caused by inadequate treatment or storage, reducing durability.

Wood Usage

Link-https://www.britannica.com/science/wood-plant-tissue/Hygroscopicity

Type-Article
Author - George Thomas Tsoumis
Date - July 30 , 2024

Primary Products

- Roundwood Products: Includes poles and posts used for utilities and construction, often treated with preservatives.

- Sawn Wood: Lumber, including railroad ties, is produced from logs through sawing. Sawmill operations involve debarking, sawing (using band, frame, or circular saws), resawing, edging, and trimming. Lumber is graded based on visual defects.

- Veneer: Thin wood sheets, produced mainly by rotary-cutting or slicing. Veneers are used in plywood and furniture. The yield is less than 50% of the original wood volume, but veneers are valuable.

Plywood and Laminated Wood

- Plywood: Made from layers of veneer glued together with alternating grain directions, resulting in a stable and versatile panel product used in construction, furniture, and more.

- Laminated Wood: Made by gluing parallel layers of lumber to create large, strong structural components like beams and arches.

- Particleboard: Produced from wood particles glued together, available in various densities and used for furniture, paneling, and structural

applications. Types include single-layer, three-layer, and oriented strand board (OSB).

- Fibreboard: Made from wood fibers and used in insulation and construction. Types include hardboard and medium-density fibreboard (MDF), used in furniture, paneling, and siding.

- Pulp and Paper: Wood is processed into pulp through mechanical, chemical, or semichemical methods. This pulp is then used to produce paper products.

Wood Recycle

Link-https://recyclinginside.com/wood-recycling/

Type-Article
Author - Unkonwn

Recycling waste wood is a multi-step process that transforms used wood into valuable materials, reducing waste and promoting sustainability. This process involves several stages, each crucial for ensuring that the wood is properly processed and repurposed.

When recycling waste wood, it must be processed in several stages – mainly including these steps:

- Receipt, weighing, and quality control
- Sorting in accordance with the waste wood grades
- Primary shredding
- Separating of recyclable material (e.g. metals)
- Secondary (fine) shredding/granulating

Recycled wood grades:

Grade A: “Clean” recycled wood – material produced from pallets and secondary manufacture, etc., and suitable for making animal bedding and mulches.

Grade B: Industrial feedstock grade – including grade A material plus construction and demolition waste, is suitable for making panelboard.

Grade C: Fuel grade is made from all of the above material and municipal collections and civic amenity sites and can be used for biomass fuel.
Grade D: Hazardous waste – This includes all grades of wood, including treated material such as fencing and trackwork, and requires disposal at special facilities.

Link-https://www.tomra.com/en/waste-metal-recycling/media-center/news/2022/wood-recycling-from-waste-to-resource-traditional-or-modern

Type-Article
Author - Valerio Sama
Date - May 12 , 2022

Waste wood generation remains high, with Germany producing around 8 million tonnes annually from construction, demolition, municipal waste, and wood processing. While 80% of this wood is used for energy, only 20% is recycled into products like particleboard. As the push for circularity and resource efficiency grows, improving wood recycling processes to maximize material recovery is crucial.

Wood Types and Processing Wood waste consists of processed and non-processed fractions. Non-processed wood is especially valuable for producing high-quality particleboards, while processed wood includes MDF, plywood, and coated materials. Effective recycling requires advanced sorting technologies to remove impurities and recover valuable materials.

Conventional Sorting Methods

Multiple-Stage Cleaning Systems: Large systems that use various techniques to remove contaminants but often require additional purification steps.

Air Separators: Use air blasts to separate lighter wood from heavier contaminants. Limited in precision and effectiveness.

Water Separators: Employ a sink-float method to separate light wood from heavier materials. Simple but less effective for advanced sorting needs.

Carbon Emission

Link-https://ecotree.green/en/how-much-co2-does-a-tree-absorb

Type-Article
Author - Unknown

Average CO2 Absorption: A tree absorbs between 10 to 40 kg of CO2 per year, with an average estimate of approximately 25 kg per year.

Calculation of CO2 Absorption:

Chemical Composition of Wood:

- Wood is composed of 50-80% cellulose, which is mainly carbon.
- The composition of wood is roughly: 50% carbon, 42% oxygen, 6% hydrogen, 1% nitrogen, and 1% mineral matter.

Photosynthesis Process:

- Trees absorb CO2 and water, using sunlight to convert them into glucose and oxygen. The glucose nourishes the tree, while oxygen is released into the atmosphere.

Water Content in Trees:

- Trees contain about 50% water, which can vary from 60% to 200% depending on species and environmental conditions.

Carbon Content Calculation:

- To calculate carbon content, the tree's dry mass (excluding water) is considered.

Annual CO2 Absorption Estimate:

- For a 35-year-old tree, which absorbs 871.63 kg of CO2 over its lifetime, the average annual absorption is approximately 25 kg of CO2 per year.

- Species and Age Factors:Different tree species have varying densities and carbon storage capacities. Young trees absorb CO2 more quickly due to faster growth, while older trees have greater density, which can result in higher total carbon storage over their lifetime.

Ethical wood harvest

Link-https://forestry.com/forestry-management/sustainable-timber-harvesting/?formCode=MG0AV3

Type-Article

Author-Gilbert Griffin

Ethical Wood Harvesting Methods:

Selective Logging: Only mature trees are harvested, leaving younger trees to grow and maintain forest structure

Reduced Impact Logging (RIL): Techniques that minimize damage to the forest floor and surrounding vegetation, such as using specialized machinery and planning careful extraction paths

Continuous Cover Forestry: Maintaining a permanent forest cover by selectively harvesting individual trees or small groups, promoting biodiversity and ecosystem health

Shelterwood Cutting: Harvesting in phases to create a shelter for seedlings, ensuring natural regeneration

Seed-Tree Cutting: Leaving a few seed-producing trees to regenerate the forest naturally

Coppicing: Cutting trees close to the ground to encourage new shoots to grow from the stump

Variable Retention Harvesting: Retaining certain trees or patches of forest to preserve habitat and biodiversity

Link-https://8billiontrees.com/carbon-offsets-credits/carbon-ecological-footprint-calculators/how-much-carbon-does-a-tree-capture/

Type calculator for tree
Author - Georgette Kilgore
Date - March 28, 2024

Calculating How Much Carbon Does a Tree Absorb:

The calculator above will help you determine how many trees you need to plant to offset your carbon footprint. Others can also allow you to compare the different types of trees and calculate how much carbon dioxide your favorite tree can capture in a year.

If you're planting trees to offset your carbon footprint, it helps to know which species absorb the most carbon. Some tree benefits calculators determine which tree species you can grow to contribute the most carbon capture.

Follow these steps:

Step 1: Determine whether you are using the calculator for a single tree or a group of trees to determine how much carbon a tree captures.

Step 2: Enter the trunk circumference (single tree) or the number of trees, and the approximate age of the tree(s).

Step 3: View calculation results immediately, including how much oxygen the trees produce, and how much carbon emissions they store.

Tree count

Link -https://saiwa.ai/blog/counting-trees/?formCode=MG0AV3

Author-Unknown

Field Sampling

- Fixed-Area Plots: A representative sample of the forest is selected, and tree counts are conducted within that sample area1
- This method ensures accurate and unbiased results2
- Random Sampling: Trees are counted within randomly selected plots to estimate the total number of trees in a larger area3
- Stratified Sampling: The forest is divided into different strata or sections, and samples are taken from each stratum to ensure all areas are represented4

2. Distance Sampling

- Line Transect Sampling: A line is established, and trees within a certain distance from the line are counted5
- This method is useful for estimating tree density over larger areas6
- Point Transect Sampling: Points are established, and trees within a certain radius from each point are counted7
- This method is also used to estimate tree density.

3 Remote Sensing

- Satellite Imagery: High-resolution images from satellites are used to identify and count trees over large areas8

- LiDAR (Light Detection and Ranging): Laser scanning technology is used to create detailed 3D maps of forested areas, allowing for accurate tree counts.

4. Manual Counting

- Direct Counting of tree using Id

Wood Recycling Equipment

Link - https://www.recyclingtoday.org/blogs/news/explore-wood-recycling-and-wood-recycling-equipment

Author - Unknown

Wood recycling equipment plays a crucial role in the recycling process by efficiently handling and processing wood waste. Here are some key types of wood recycling equipment:

- Wood Chippers/Shredders: These machines are used to reduce large pieces of wood waste into smaller chips or shreds, which can then be used for various applications such as mulch, biomass fuel, or composite wood products.

- Grinders: Wood grinders are designed to grind wood waste into finer particles or sawdust. This processed material can be used for particleboard production, animal bedding, or as a fuel source.

- Sorting and Screening Equipment: These machines are used to separate different types of wood waste based on size, density, or other criteria. This helps in preparing the waste for further processing or recycling.

- Magnetic Separators: Often used in conjunction with wood recycling equipment, these devices remove metal contaminants from the wood waste stream to prevent damage to processing equipment and ensure product quality.

- Briquetting Machines: These machines compress wood waste into briquettes or pellets, which can be used as fuel or feedstock in various industries, including biomass energy production.

- Incinerators and Boilers: In some cases, wood waste is burned directly as a fuel source in specialized incinerators or boilers, generating heat or electricity.

References

Forest Regions	https://pwonlyias.com/forest-in-india/#types-of-forest-in-india
Tree Species	https://www.naturalmangoes.com/10-best-timber-tree-varieties-in-india/
The Growth Cycle of Tree	https://ecotree.green/en/blog/the-life-cycle-of-a-tree
Techniques for Forest Regeneration	https://synopsisias.com/blog/natural-and-artificial-regeneration-of-forests-forestry-optional-for-upsc-ifs-category?category_slug=silviculture-general
Wood Processing	https://duffieldtimber.com/the-workbench/timber-trends/how-is-wood-manufactured-and-processed
Wood Quality	https://www.deskera.com/blog/best-practices-for-wood-manufacturing-quality-control/#key-elements-of-wood-manufacturing-quality-control
Wood Usage	https://www.britannica.com/science/wood-plant-tissue/Hygroscopicity
Wood Recycle	https://recyclinginside.com/wood-recycling/ https://www.tomra.com/en/waste-metal-recycling/media-center/news/2022/wo

	od-recycling-from-waste-to-resource-traditional-or-modern
Carbon Emission	https://ecotree.green/en/how-much-co2-does-a-tree-absorb https://8billiontrees.com/carbon-offsets-credits/carbon-ecological-footprint-calculators/how-much-carbon-does-a-tree-capture/
Tree count	https://saiwa.ai/blog/counting-trees/?formCode=MG0AV3
Wood Recycling Equipment	https://www.recyclingtoday.org/blogs/news/explore-wood-recycling-and-wood-recycling-equipment
Tool used for processing	https://disciplines.ng/wood-production-tools-and-equipment/?formCode=MG0AV3
Ethical wood harvest	https://forestry.com/forestry-management/sustainable-timber-harvesting/?formCode=MG0AV3

IT Solutions : Interview Plan

System : Wood Product Lifecycle Management

Project Reference : xyz

Participants : Ayush Shah
Jenil Soni

Date : 9/10/2024 Time : 2:00 P.M.

Duration : 2 hours Place : CEP-207

Puspose of Interview :

The purpose is to uncover critical deficiencies in the current system and define precise requirements for the new system to address these issues effectively. This ensures that the new system not only resolves existing problems but also aligns with strategic goals and supports future growth.

Agenda :

Problems of Wood lifecycle
Process of Wood product from Scratch
Impact of Deforestation
Understandings with Current System
Market Scope of original wood

Documentation :

The files which we mention at references.

IT Solutions : Interview Summary

System : Wood Product Lifecycle Management

Project Reference : xyz

Participants : Ayush Shah
Jenil Soni

Date : 9/10/2024 Time : 2:00 P.M.

Duration : 2 hours Place : CEP-207

Puspose of Interview :

The purpose is to uncover critical deficiencies in the current system and define precise requirements for the new system to address these issues effectively. This ensures that the new system not only resolves existing problems but also aligns with strategic goals and supports future growth.

1. They primarily use Oak , Timber and pine.
2. The stages include initial cutting, kiln drying, milling, and final finishing
3. They use a combination of manual logs and a legacy software system for tracking
4. In the current system , Data is recorded manually and entered into our software system, which leads to errors and inefficiencies.
5. Faces the problems of deforestation , illegal logging , Maintenance , Transportation , climate change.
6. Market Scope in Furniture goods , Construction Industry , Crafts and Artisan Work
7. Further discussions needed when more information is available and Follow-up questions will be prepared.

Questionnaire

1 Name : _____

2 How familiar are you with processed wood products (e.g., plywood, MDF, veneer)?

- Very familiar
- Somewhat familiar
- Not familiar

3 Do you know the difference between natural wood and processed wood?

- Yes, I'm aware of the differences.
- I've heard about it, but not sure of the details.
- No, I don't know the difference.

4 Where do you usually purchase wood or wood-based products?

- Furniture stores
- Hardware stores
- Online retailers
- Other: _____

5 How important is the environmental impact of wood products when making a purchase decision?

- Very important
- Somewhat important
- Not important

6 In your daily life, which types of wood-based products do you interact with or use the most?

- Furniture (e.g., tables, chairs, beds, etc.)
- Building materials (e.g., flooring, doors, window frames)
- Interior decor (e.g., wall panels, wooden art, picture frames)
- Stationery or packaging (e.g., paper, notebooks, cardboard)

- Kitchenware (e.g., cutting boards, wooden utensils)
- Other: _____

7 Are you aware of the concept of sustainable forestry (i.e., harvesting wood in a way that ensures forests regenerate)?

- Yes
- No

8 What qualities do you look for when purchasing processed wood products? (Check all that apply)

- Durability
- Price
- Appearance/Finish
- Environmental friendliness
- Availability

9 Have you ever encountered quality issues with processed wood products?

- Yes
- No
- Not applicable

Observation

Based on the questionnaire, here are some observations that could provide insights into customer awareness, preferences, and attitudes toward wood products and sustainability:

Observations

1. Awareness and Familiarity with Processed Wood Products:

- Respondents' familiarity with processed wood products (e.g., plywood, MDF, veneer) will reveal their knowledge level, which may influence purchasing decisions and expectations for quality.

- Those familiar with the differences between natural and processed wood may value certain characteristics (such as appearance or durability) differently than those less knowledgeable.

2.Purchasing Habits:

- Understanding where respondents typically buy wood products (furniture stores, hardware stores, online retailers, etc.) can indicate preferred shopping channels, which may influence product availability and marketing approaches.

- This also suggests the types of products customers are likely exposed to and may affect their brand loyalty or openness to sustainable options.

3. Importance of Environmental Impact:

- The responses on how environmental impact affects purchase decisions will help assess the importance customers place on sustainability.

- High importance on environmental impact suggests a growing preference for sustainably sourced products, which can guide companies to emphasize eco-friendly practices and certifications.

4. Common Wood-Based Products Used Daily:

- Identifying the most common types of wood products in daily life (furniture, building materials, decor, etc.) provides insight into high-demand product categories and their perceived functionality or importance in consumers' lives.

- This information can help focus sustainable practices on the most frequently used items.

5. Awareness of Sustainable Forestry:

- Gauging awareness of sustainable forestry practices indicates the extent to which customers are informed about sustainable sourcing. Those who are aware may be more supportive of sustainability initiatives.

- This awareness level can help tailor educational campaigns to promote sustainable forestry among those less familiar.

6. Qualities Valued in Processed Wood Products:

- The qualities respondents seek (durability, price, appearance, environmental friendliness, availability) reveal priorities when choosing products, helping manufacturers understand which features to emphasize in marketing.

- A high interest in environmental friendliness and durability, for example, might encourage a focus on robust, eco-certified products.

7. Experience with Quality Issues:

- Knowing whether customers have experienced quality issues with processed wood products provides insight into potential areas of improvement in product manufacturing and quality control.

- Common quality concerns, if identified, could guide companies to address specific pain points, improving customer satisfaction and brand trust.

.

3 Fact Finding Chart

Objective	Techniques	Subject	Time Commitment
To get the better knowledge of the wood product lifecycle and the growth cycle of a tree.	Background Readings	Online articles	1 hour
To get the knowledge of forest regions and tree species.	Background Readings	Online articles	1 hour
To gain understandings of wood processing , wood quality , wood usage , Wood recycle , Wood recycle Equipments	Background Readings	Online articles	5 hour
To get better Understandings about the Tools used for Processing the wood , Tree Count , Carbon Emission , Ethical Wood Harvest .	Background Readings	Online articles	3 hour
To get the core process we discuss with the wood facility manager and get the knowledge about the process of wood in	Interview	Wood facility manager	1 hour

manufacturing company.			
To get knowledge of current system.	Interview	Wood facility manager	1 hour
To know about carbon emission.	Background Readings	Online Articles	0.5 hour
Customer usage of wood and preference of goods of wood.	Survey	Questionnaire	-

4 Requirement list

1. Region and Tree Management

- Geographical and Environmental Data Recording:
    - Maintain accurate information on forest regions, including coordinates, area, soil conditions, and climate data.
- Tree Tracking:
    - Track individual trees with data on species, age, growth rates, health status, and potential pests or diseases.
- Harvest Record Management:
    - Document the quantity of trees harvested, dates, techniques used, and logger details for each event.
- Efficient Querying System:
    - Develop a querying system for quick retrieval of data on tree populations, harvest records, and logger activities across regions and time frames.
- Environmental Impact Assessment Tools:
    - Implement tools to analyze environmental impact, predict ecosystem changes, and assess the effects of harvesting on forest health.

2. Harvest Tracking

- Detailed Harvest Event Documentation:
    - Record all aspects of each harvest event, including tools, climate conditions, and any notable incidents.
- Raw Wood Tracking System:
    - Implement a system to accurately measure raw wood yield and assign unique identifiers to batches for traceability.
- Logger Accountability System:
    - Track loggers responsible for each harvest event, including credentials, performance, and sustainable harvesting compliance.
- Real-Time Data Collection:

    - Develop real-time data capture methods to enhance accuracy and enable swift decision-making during harvest events.
- Yield Prediction Analytics:
    - Integrate predictive analytics to forecast yields based on historical data and current forest conditions.

3. Wood Processing

- Processing Stage Monitoring:
    - Track each batch through all processing stages with details on specific treatments, tools used, and quality control measures.
- Facility Management:
    - Maintain data on facility capacity, capabilities, efficiency metrics, and compliance with industry standards.
- Advanced Query System:
    - Enable a querying system to identify processing details, facilities involved, and final wood product specifications.
- Quality Control Integration:
    - Implement quality control checks at each processing stage, with records of any defects or issues.
- Resource Optimization Analytics:
    - Analyze processing data to improve resource allocation, minimize waste, and enhance processing chain efficiency.

4. Manufacturing

- Processed Wood Allocation System:
    - Track and allocate processed wood batches, recording types, grades, and quantities sent to each manufacturer.
- Product Tracking and Lifecycle Management:
    - Maintain records for all manufactured products, including specifications, quality grades, and production details.
- Inventory Management System:
    - Implement a system to monitor available quantities of processed wood and finished products at each manufacturing location.
- Production Efficiency Analysis:

    - Analyze data to identify production bottlenecks, optimize workflows, and enhance resource utilization.
- Sustainability Metrics:
    - Track sustainability-related data, such as energy consumption, waste generated, and carbon footprint, within the manufacturing process.

5. Customer Purchases

- Secure Purchase Record System:
    - Store accurate and secure records of each customer purchase, including product details, quantities, and purchase dates.
- Customer Data Management:
    - Maintain comprehensive customer profiles with preferences, purchase history, and contact information.
- Sales Analysis Tools:
    - Develop analytical tools for sales pattern analysis, demand forecasting, and insights into customer preferences.
- Customer Relationship Management Integration:
    - Integrate purchase data with CRM systems to improve customer support and marketing strategies.
- Product Lifecycle Tracking:
    - Implement mechanisms to track product lifecycles, supporting warranties and future recycling efforts.

6. Recycling

- Recycling Event Documentation:
    - Document each recycling activity with details on recycled products, methods used, and materials recovered.
- Contractor Management:
    - Maintain records on recycling contractors, including qualifications, performance metrics, and compliance with environmental regulations.
- Efficient Query System:
    - Create a querying system to quickly retrieve data on recycled products, contractor details, and environmental impact reports.
- Material Recovery Tracking:

    - Implement tracking of recovered materials, including types, quantities, and future uses or sales.
- Circular Economy Integration:
    - Develop plans to reincorporate recycled materials into production, fostering a closed-loop system in the wood industry.

5 Users and their privileges

1. Forest Manager

- Role: Manages regions, trees, and environmental data.
- Privileges:
    - SELECT, INSERT, UPDATE on:
        - Region
        - Tree_Count
        - Region_Location
        - Harvest
        - Harvest_Report
        - Region_Harvest
    - SELECT on:
        - Logger
        - Logger_Contact
    - EXECUTE on functions related to region data queries and environmental impact assessments.

2. Logger

- Role: Logs harvest events and related data.
- Privileges:
    - SELECT, INSERT on:
        - Harvest
        - Harvest_Logger
        - Logger
        - Logger_Contact
    - SELECT on:
        - Region
        - Tree_Count
    - EXECUTE on functions related to yield predictions and harvest event tracking.

3. Processing Facility Manager

- Role: Manages wood processing activities and facility data.
- Privileges:
    - SELECT, INSERT, UPDATE on:
        - Batch
        - Wood_Processing
        - Processing_Stage
        - Inventory
        - Facility
        - Facility_Address
    - EXECUTE on functions for querying facility utilization, inventory levels, and batch processing status.

4. Manufacturer

- Role: Manages manufacturing of wood products.
- Privileges:
    - SELECT, INSERT, UPDATE on:
        - Manufacturer
        - Manufacturer_Batch
        - Product_Inventory
        - Wood_Product
        - Transaction
    - SELECT on:
        - Batch
        - Inventory
        - Facility
    - EXECUTE on functions related to wood allocation, production tracking, and inventory management.

5. Customer Relations Manager

- Role: Manages customer data, transactions, and sales analysis.
- Privileges:
    - SELECT, INSERT, UPDATE on:
        - Customer
        - Customer_Contact
        - Sales
        - Wood_Product
    - EXECUTE on functions related to purchase history analysis and customer data management.

6. Recycling Manager

- Role: Manages recycling events, contractors, and waste disposal data.
- Privileges:
    - SELECT, INSERT, UPDATE on:
        - Recycle
        - Waste_Disposal
        - Recycle_Method
        - Disposal_Method
    - SELECT on:
        - Wood_Product
        - Facility
    - EXECUTE on functions for material recovery tracking and environmental impact reporting.

7. Admin

- Role: Full access to manage all database tables and user privileges.
- Privileges:
    - ALL PRIVILEGES on all tables, views, and functions.
    - GRANT privileges to other users as needed.

Chapter 2 - Database Design

1. Noun Analysis

Table 1: All Extracted Nouns & Verbs from Problem Description

Nouns	Verbs
Region	Manage
Tree	Maintain
Population	Track
Data	Support
Forest	Record
Tree Count	Prevent
Environmental Report	Use
Resources	Evaluate
Harvest	Predict
Logger	Require
Yield	Log
Harvest Event	Link
Harvest Method	Enforce
Sustainability Score	Enable
Carbon Absorption	Analyze

Batch	Allocate
Inventory	Manufacture
Processing Stage	Purchase
Facility	Dispose
Capacity	Recycle
Quality Control	Document
Product	Manage
Manufacturer	Optimize
Transaction	Meet
Customer	Understand
Purchase	Adjust
Recycling Event	Support
Contractor	Measure
Method	Contribute
Waste	Track
Emissions	Assess
Lifecycle	
Compliance	
Traceability	
Sustainability	

Table 2: Accepted Noun and Verb Lists

Candidate Entity Set	**Candidate Attribute Set**	**Candidate Relationship Set**
Forest Region	ID, Name, Location, Type	Contains
		Manages
Tree	ID, Name, Type, Age	Grows in
		Belongs to
Processing Stage	Stage Name	Involves
Harvest	Quantity, Schedule, Actual Time	Occurs in
		Utilizes
Wood Product	Type, Quantity of Tree Used, Carbon Footprint	Produced from
		Transforms
Recycle	Product, Method, Date	Recycles
		Generates Report on

Table 3: Rejected Noun & Verbs List

Noun	Reject Reason
Geographic Coordinates	Considered an attribute of Forest Region (Location)
Soil Conditions	Attribute of Forest Region (Type could cover this)
Climate Data	Attribute of Forest Region or tracked as part of reports
Logger	Not a core entity; related to Harvest as a role or assignment
Logger Accountability	Considered as part of Logger's role tracking; not separate entity
Harvest Event	Redundant; represented by the "Harvest" entity
Wood Tracking	This represents a process, not an entity
Customer	Not part of wood lifecycle as defined by the ER
Purchase Records	Sales-related, out of scope for this system's main focus

Sustainability Metrics	Attribute rather than an entity; tracked with reports
Carbon Footprint	An attribute of Wood Product
Customer Relationship Management	Out of scope; pertains more to CRM functions
Inventory Management	Process or function, not an entity
Query System	A functionality, not an entity
Environmental Impact Assessment	This represents a report, not a standalone entity

Verb	Reject Reason
Monitors	Implied in tracking attributes, not as core relationship
Tracks	This is a functionality, not an explicit relationship
Documents	General verb; covered by "Records" or "Contains"
Allocates	Out of scope for core ER structure
Predicts	Related to analytics, not core data relationships
Assigns	A task or process, not part of core ER
Maintains	Implied in tracking attributes or ownership
Schedules	Part of Harvest attributes, not a separate relationship
Generates	General; more specific relationships used

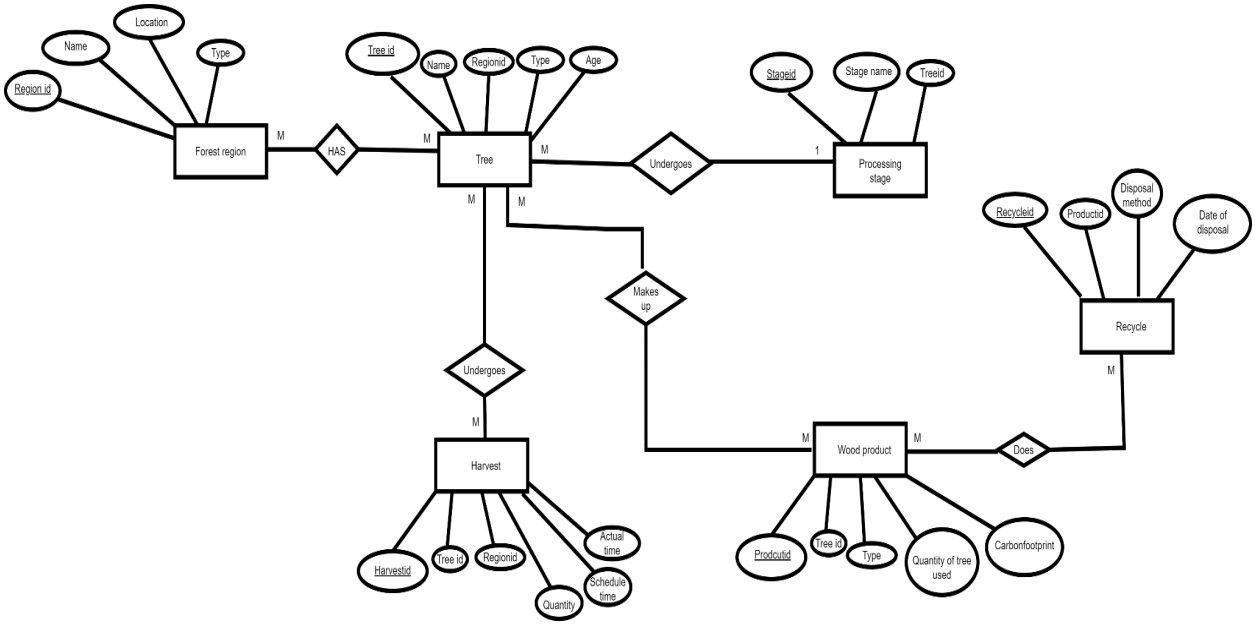
Converts	Too vague; "Transforms" is used instead
Produces	Replaced with more specific relationship terminology

Finalize Noun and Verb List

Final Noun & Verb List (Based on ER Requirements)

Candidate Entity Set	**Candidate Attribute Set**	**Candidate Relationship Set**
Forest Region	ID, Name, Location, Type	Contains
		Manages
Tree	ID, Name, Type, Age	Grows in
		Belongs to
Processing Stage	Stage Name	Involves
Harvest	Quantity, Schedule, Actual Time	Occurs in
		Utilizes
Wood Product	Type, Quantity of Tree Used, Carbon Footprint	Produced from
		Transforms
Recycle	Product, Method, Date	Recycles
		Generates Report on

2. Schema and ER Diagram Design



Schema

Forest region-region_id,name,location,type
Tree-tree_id,name,region_id,type,age
Processing_Stage-stage_id,stage_name,tree_id
Harvest-harvest_id,tree_id,region_id,quantity,schedule_time,actual_time
Wood product-product_id,tree_id,type,quantity,carbon_footprint
Recycle-recycle_id,prodcut_id,disposal_method,date of disposal

ER Diagram Improvement

i. Identify Entity types.
1. Add information like weak entity set/s, the identifier for weak entity set/s to ERD.

1 Inventory
Reason for Weak Entity: The Inventory entity seems to be dependent on the wood processing entity, as it represents the stock of processed wood stored.
Identifier: Batch_ID to uniquely identify inventory records within the facility.
Related Strong Entity: Wood processing
Identifying Relationship: A "stores in" relationship links Inventory to wood processing.

2 Product Inventory
Reason for Weak Entity: Product Inventory depends on the wood product entity, as it stores products that the manufacturer has produced and made available for sale.
Identifier: Product_ID and Manufacturer_ID, as it logs inventory for specific products manufactured by a particular manufacturer.
Related Strong Entity: wood product
Identifying Relationship: A "stores in" relationship links product Inventory to wood product.

3 Wood inspection
Reason: Wood inspection depends on Wood Product as it grades specific products based on their condition or quality. It does not have its own unique identifier and instead relies on Product_ID from Wood Product.
identifier for weak entity:Product_ID
Strong entity-wood product
Identifying relationship -undergoes

4 tree_count

Reason:Tree Count depends on the Region entity for its identity. The Tree Count is defined within a specific region and cannot exist without being tied to a Region. The Tree Count is associated with a Region via the RegionID (which is a foreign key), and its identity (or record) relies on the region to which it pertains.

Identifier for weak entity-Region_id

Strong entity-region

Identifying relationship-has

5 wood processing

Reason: Wood Processing depends on the Batch, Facility, and Processing Stage entities for its identity. The Wood Processing event occurs only when a specific batch of wood is processed at a particular facility and during a specific processing stage. It cannot exist independently and is defined within the context of these entities. Thus, Wood Processing is inherently tied to a specific batch, facility, and processing stage, and its record depends on these entities for identification.

Identifier for weak entity:
Batch_ID, Facility_ID, Processing_stage_ID

Strong entities:Batch,Facility,Processing Stage

Identifying relationship:
Batch "has" Wood Processing
Facility "has" Wood Processing
Processing Stage "defines" Wood Processing

Relationships in the Diagram

1. Region and Harvest

Type: Simple Association
Explanation: Each Region can have multiple Harvest records, and each Harvest can occur across multiple Regions, capturing their association without dependency.

2. Logger and Harvest

Type: Simple Association
Explanation: Logger performs tasks in Harvest so simple Association more accurately represents the assignment of loggers to harvests.

3. Harvest and Wood Processing

Type: Simple Association
Explanation: This is a sequential, process-driven link where harvested logs move to processing stages. There is no hierarchy or whole-part dependency.

4. Wood Processing and Inventory

Type: Simple Association
Explanation: Inventory stores processed wood,Inventory is simply where processed wood is tracked, making this a Simple Association.

5. Wood Processing and Facility
Type: simple association

6 Wood processing and processing stages

Type: Aggregation
Explanation: Wood Processing is a larger process composed of multiple Processing Stages (e.g., sawing, drying), each contributing to the final product. These stages exist only as part of the overall Wood Processing, making this an Aggregation.

7. Inventory and Transaction

Type: Simple Association
Explanation: Transactions record the movement or sale of items from Inventory, which is a basic association reflecting inventory records rather than an ownership or part dependency.

8. Transaction → Manufacturer

Type: Simple Association
Explanation: Each Transaction is simply an event between Inventory and Manufacturer, making it a Simple Association.

9. Manufacturer → Wood Product

Type: Hierarchy
Explanation: Manufacturer creates Wood Product, forming a One-to-Many hierarchical link where one manufacturer oversees or "owns" many products.

10 wood product and inventory

Type: Aggregation
Explanation: Inventory holds various Wood Products as part of stock management. Wood Products are stored within Inventory but are independent items that can exist outside of it, making this an Aggregation.

11 Inventory and sales

Type: Simple Association
Explanation: Inventory provides items for Sales Transactions, where each transaction reduces the quantity of products in Inventory. Since Sales Transaction depends on Inventory but does not represent a larger structure or ownership, this is best described as a Simple Association.

12 sales and user

Type: Association (Simple)
Explanation: Sales Transaction is linked to a User, representing the customer who initiates or completes the transaction. Each Sales Transaction is associated with a specific User but remains independent, as Sales Transactions are recorded separately from User details. This makes it a Simple Association, where the User is connected to the transaction as a participant.

13. Wood Product → User

Type: Simple Association
Explanation: This relationship reflects ownership or usage of products. Simple Association is appropriate.

14 Wood Product → Wood Inspection

Type: Simple Association
Explanation: Wood Inspection assesses the quality of Wood Product, but it does not "own" or oversee it as a parent. This relationship is best represented as a Simple Association.

15 Wood Inspection and Recycle

Type: Association (Simple)
Explanation: Wood Inspection assesses if a product can be repurposed. If suitable, it is directed to Recycle for processing and reuse. This relationship is a Simple Association, as Wood Inspection simply directs products to Recycle without creating additional dependency.

16 Wood Inspection and Disposal

Type: Simple Association
Explanation: When Wood Inspection deems a product unfit for recycling, it sends it to Disposal. This relationship is also a Simple Association, where Wood Inspection determines that a product should be discarded, but without further interdependence between the inspection and disposal processes.

17. Harvest Report → Harvest

Type: Simple Association
Explanation: Harvest Report provides data on Harvest, reflecting an informational association rather than a hierarchical or whole-part dependency.

18. Harvest Report → Region

Type: Simple Association
Explanation: Each Harvest Report may relate to a specific Region without implying dependency or hierarchy.

19. Recycle → Wood Processing

Type: Simple Association
Explanation: Recycle links to Wood Processing to indicate that recycled wood can be reprocessed, forming a cycle in the process but not a recursive or hierarchical dependedancy

Entity Set

Entities	Attributes
Region	RegionID , Region_name, R_Treecount_ID, Area,Location_ID,Report_ID
Region_location	Location_id,city,state
Tree_count	region_ID,tree_count,Date
region_harvest	region_ID,harvest_ID
Harvest	Harvest_ID , Batch_ID,Harvest_date,volume, Harvested_Tree_count, Replanted_Tree_count,
Harvest_Report	Report_ID,Harvest_ID,report_date,tree_density, sustanibility_score,carbon_absorbtion
Region_harvest_ report	Region_id ,harvest_report_id
Logger	Logger_ID , first_name,middle_name,last_name
Logger_contact	Logger_ID,contact_number
harvest_logger	harvest_ID,logger_ID

Batch	Batch_ID,Facility_ID,total_quantity,total_volume
Facility	Facility_ID ,Address_ID,Facility_name, Owner_name
Wood_processing	(Batch_ID,facility_id,processing_stage_id), status,quantity,processing_Start_date,processing_end_date
processing_stage	processing_stage_ID,processing_stage_name,
Inventory	(Batch_ID,Facility_ID),total_quantity,available_quantity
Transaction	Transaction_ID,Batch_ID,Facility_ID,Manufacturer_ID, quantity_purchased,purchase_date
Manufacturer_Batch	Manufacturer_ID,Batch_ID,quantity_owned
Manufacturer	Manufacturer_ID , Address_ID,sales_ID,Name,wood_purchased
Wood_Product	Product_ID ,Manufacturer_ID,User_ID, Product_Type, Wood_used_quantity,volume, Production_date,usage_Status
product_inventory	(Product_ID,Manufacturer_ID),total_quantity,available_quantity
User	User_ID, Customer_name

User_contact	User_ID,contact_number
sales	sales_ID, User_ID ,product_ID, Purchase_date, Quantity_purchased
wood_inspection	Product_ID,condition,decision
Waste_disposal	Disposal_ID, product_ID,disposal_method_ID,disposal_date
disposal_method	Disposal_method_ID,Disposal_method_name
Recycle_method	Recycle_method_ID , Recycle_method_name
Recycle	Recycle_ID,Product_ID,recycle_method_ID,recycle_date
Facility Address	facility_Address_ID , Street, City, State, Postal_code
manufacturer_phone	(manufacturer_ID,contact_number)
Manufacturer_Address	facility_Address_ID , Street, City, State, Postal_code

Entity vs. Attribute

Logger as an Entity

- In the Harvest table, Logger could be an attribute, storing details like the logger's name or ID.
- To improve data transparency and enable detailed tracking of logger activities across multiple harvests, Logger was made a separate entity.
- This allows storing additional logger-specific information (such as contact details and qualifications) and maintaining relationships with multiple harvest records.

Region location as an entity

- Location might have been considered as an attribute of the Region entity. This would mean that information about the geographical location (such as city, state, or coordinates) would be directly stored within the Region table.
-
- By making Location its own entity, you store location information only once, even if multiple regions share the same location. This allows for efficient management of location data, as any update to a location (such as a change in the name of a city or state) only needs to be made in one place.

Facility address as an entity

- **Address** might have been considered an attribute within the **Facility** entity. This would mean storing the address (such as street, city, state, postal code) directly within the **Facility** table.
- If multiple facilities share the same address (for example, in the case of a large complex with multiple buildings), storing the address in each **Facility** record would lead to unnecessary duplication. This increases the amount of data stored and can make it harder to manage updates or changes.

- By creating a **Facility_Address** entity, we store address information only once. Multiple facilities that share the same address can reference the same address record, avoiding duplication and ensuring data consistency

Manufacturer address as an entity

- Address might have been considered an attribute within the Manufacturer entity. This would mean storing the address (such as street, city, state, postal code) directly within the Manufacturer table

- if multiple manufacturers are located at the same address (e.g., multiple branches of a manufacturer operating from the same industrial complex), storing the address in each manufacturer's record would lead to unnecessary duplication. This increases the amount of data stored and makes it harder to manage changes to address information.

- By creating an Address entity, address information is stored once, and manufacturers reference the address through a foreign key. This ensures that if multiple manufacturers are located at the same address, there is no duplication in the data.

Processing stage as an entity

- Processing Stage might have been considered an attribute within the Wood Processing table. This would mean that the stage (e.g., "Cutting," "Sanding," or "Finishing") is directly stored as an attribute in the Wood Processing record for each batch or processing operation.

- Storing the Processing Stage as an attribute in the Wood Processing table would restrict the ability to track additional information about each stage, such as the duration, materials used, equipment, or worker responsible for each stage. This can make it harder to analyze or report on processing performance.

- Treating Processing Stage as an entity avoids redundancy. Instead of repeating the same data across multiple Wood Processing records, we store it once in a Processing Stage table and link it to each processing record via a foreign key. This ensures the database is normalized and adheres to best practices.

Wood Inspection as an Entity

- Although Wood inspection could be stored as an attribute within the Wood Product entity, creating it as a separate entity enables detailed storage of grading information for each product, such as grade level, quality score, and inspection notes.
- This structure supports multiple grades for a single product and promotes consistency in quality assessment.

User as an Entity

- User details could be added as attributes in the Wood Product table to indicate ownership or use.
- Defining User as a separate entity enables detailed tracking of customer or end-user information
- This structure supports relationship management and customer-specific analysis, which can help improve product offerings and customer satisfaction.

Transaction as an Entity

- Transaction details, like purchase dates and quantities, could be stored within manufacturer
- By making Transaction a separate entity, you can record detailed transaction data, including purchase dates, quantities, and buyer information.
- This allows a record of every transaction and provides a comprehensive history of product movement, which aids in inventory management and financial reporting.

Disposal as an Entity

- Disposal status could be an attribute within recycle to show if the product is disposed of.
- As a separate entity, Disposal enables the tracking of disposal methods, dates, and reasons, along with any environmental impacts.
- This helps maintain a detailed disposal history, which is beneficial for compliance with environmental regulations and assessing product lifecycle end stages.

Recycle method as an entity

- Recycle Method could have been considered an attribute within the Recycle table, where you might store the recycling method (e.g., "Mechanical Processing," "Chemical Processing," "Biodegradable" methods) directly as an attribute.

- If the same Recycle Method (e.g., "Mechanical Processing") is used across multiple recycling events, storing this as an attribute would lead to redundancy in the Recycle table, as the same method would be repeated for each record.

- By creating a Recycle Method entity, we normalize the database and avoid redundancy. Each Recycle Method is stored only once and linked to multiple recycling activities via a foreign key.

Disposal method as an entity

- Disposal Method could have been considered an attribute within the Waste Disposal table. You might have stored the type of disposal method (e.g., "Landfill," "Incineration," "Composting") directly as an attribute in the Waste Disposal table.

- If multiple waste disposal records use the same method (e.g., "Landfill"), it would lead to redundant data in the Waste Disposal table. Storing the same method for each record would unnecessarily repeat the information.

- By treating Disposal Method as an entity, we can avoid redundancy. Each disposal method is stored once in the Disposal Method table and linked to multiple waste disposal activities via a foreign key. This ensures that the data is normalized and reduces the risk of inconsistencies.

Entity vs. Relationships

Wood Processing

- Wood Processing could be represented as a set of relationships between Harvest and facility.
- Creating Wood Processing as a separate entity allows capturing multiple processing stages, batch numbers, processing dates, and other specific data points that describe each stage of wood handling.
- By defining Wood Processing as an entity, you can manage detailed records of each batch's processing stages, making it easier to track wood from harvest to final products, and allowing for consistent data retrieval.

Wood Inspection

- Wood Inspection could be represented as a simple relationship between Wood Product and Recycle/Disposal outcomes.
- Creating Wood Inspection as a separate entity allows capturing specific inspection data, product conditions, inspection dates, and assessment criteria that guide whether a product should be recycled or disposed of.
- By defining Wood Inspection as an entity, you can manage detailed records of each product's condition and inspection outcome, facilitating efficient tracking through the inspection process and ensuring consistent, data-driven decisions for recycling or disposal.

Binary vs ternary

Original Ternary Relationship

1 Harvest → Wood Processing → Logger

- This ternary relationship originally represented the scenario where a Harvest undergoes Wood Processing and is also associated with a specific Logger responsible for conducting the harvesting tasks.

Converted to Binary Relationships:

Logger → Harvest

- This binary relationship connects Logger to Harvest, representing that each logger is assigned to conduct specific harvest tasks. It allows tracking which logger performed a given harvest activity.

Harvest → Wood Processing

- This binary relationship shows that each Harvest leads to various Wood Processing stages, representing the transformation of raw materials from harvest into processed wood. This relationship focuses on the flow of materials and stages within the processing phase.

2 Wood Product → Wood Grading → Recycle

- Represents a scenario where a Wood Product is graded, and the grading decision can lead to Recycling.

Converted to Binary Relationships:

- Wood Product → Wood Grading: Links Wood Product to Wood Grading, where each product has one or more quality assessments.
- Wood Grading → Recycle: Connects Wood Grading to Recycle, showing that certain products are recycled based on their grade.

3 Wood Grading → Recycle → Disposal

- This ternary relationship would represent a flow where, based on Wood Grading, a product may go to Recycle, and if unsuitable for recycling, it may then be directed to Disposal. The grading decision dictates whether a product enters the recycling process or proceeds directly to disposal.

Converted to Binary Relationships

Wood Grading → Recycle

- A binary relationship indicating that certain grades lead a product to Recycle. This represents the initial decision based on the grading assessment.

Recycle → Disposal

- A binary relationship showing that if a product cannot be recycled, it moves to Disposal. This connection allows tracking products that have gone through recycling but are ultimately disposed of.

4 Harvest Report → Harvest → Region

- In this original ternary relationship, a Harvest Report provides information about a specific Harvest activity within a certain Region. The report links data about both the harvest event and the region in which it occurred, providing insights into location-specific harvesting details.

Converted to Binary Relationships

Harvest Report → Harvest

- Description: This binary relationship links each Harvest Report to a specific Harvest event, detailing information about that particular harvest activity (such as date, volume, and tree density).

Harvest Report → Region

- Description: This binary relationship connects Harvest Report to Region, indicating the location where the harvest took place. This association allows each report to be tied to its respective region without directly involving the harvest entity in this connection.

5 Wood Processing → Processing Stage → Facility

- In this original ternary relationship, Wood Processing is connected to both Processing Stage and Facility. This means that a single instance of wood processing encompasses the specific processing stage and the facility where that processing occurs.

Converted to Binary Relationships

Wood Processing → Processing Stage

- Description: This binary relationship links each Wood Processing instance to a specific Processing Stage. It details the stage (e.g., sawing, drying, or treating) that the wood is currently undergoing. This relationship allows for clear tracking of the processing steps involved in the lifecycle of the wood product.

Wood Processing → Facility

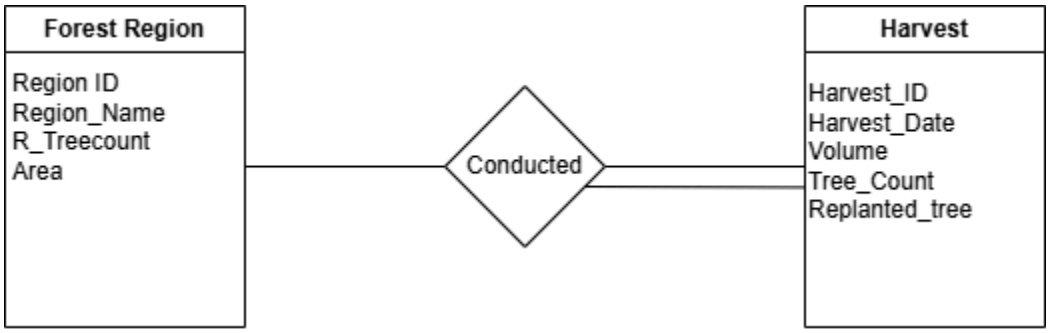
- Description: This binary relationship connects Wood Processing to Facility, indicating the specific facility where the processing occurs. This association allows for the identification of the location responsible for handling the wood processing without needing to reference the processing stage in this connection. It enhances clarity by separating the location data from the stage data, simplifying management and reporting.

Aggregation vs.Ternary Relationship.

No aggregation relationship

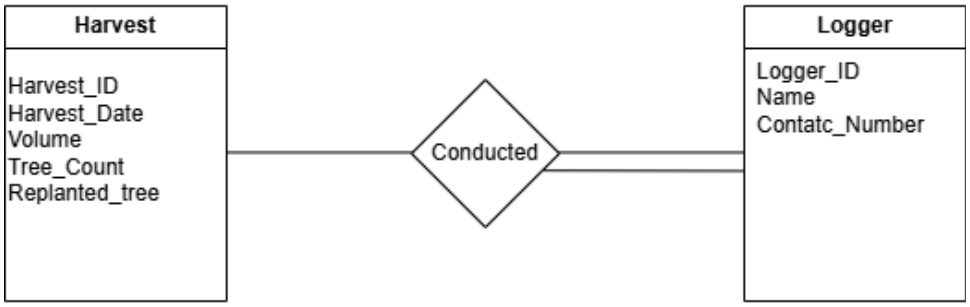
Many to Many Relation table

1) Forest_Region - Harvest



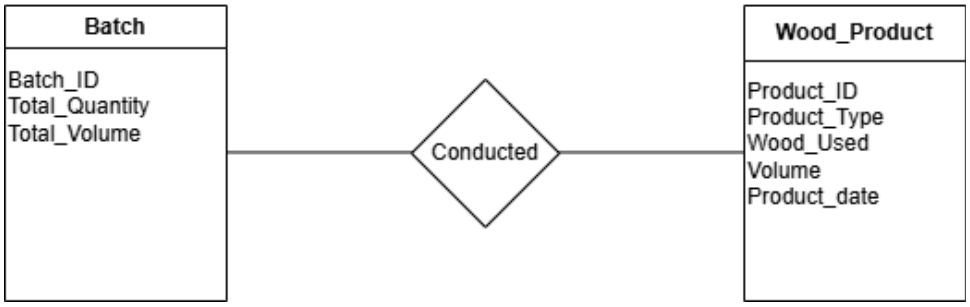
- Conducted(Region_ID,Harves_ID)

2) Harvest - Logger



- Conducted(Harvest_ID,Logger_ID)

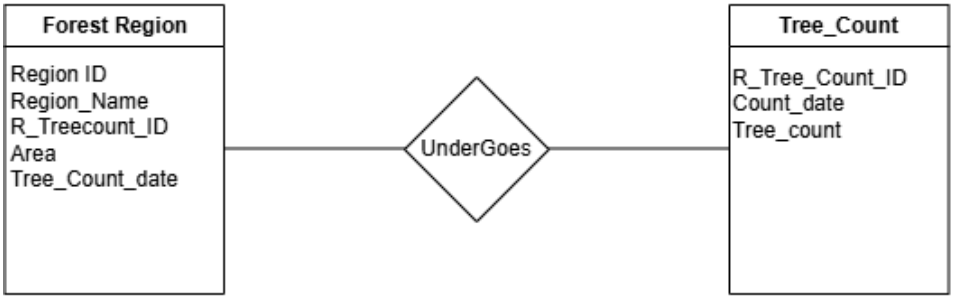
3) Batch - Wood Product



- Form(Batch_ID,Product_ID)

One to Many Relation table

1) Forest_Region - Tree_Count

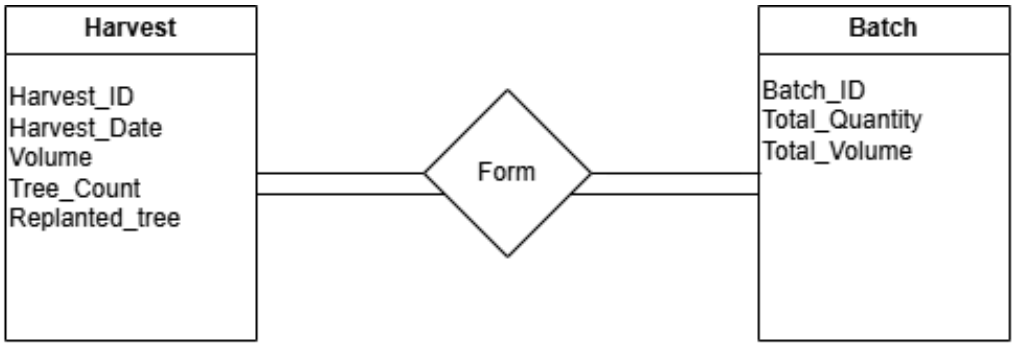


- Undergoes(Region_ID)

2) Forest_Region - Harvest_Report

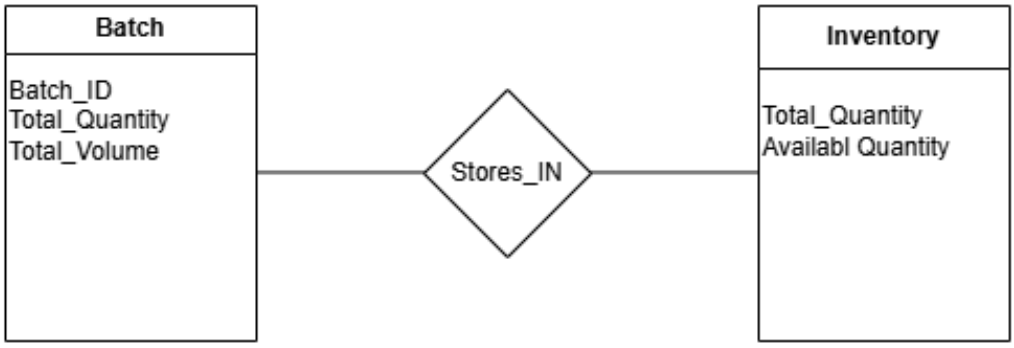
- Basis On(Region_ID,Report_ID)

3) Harvest - Batch



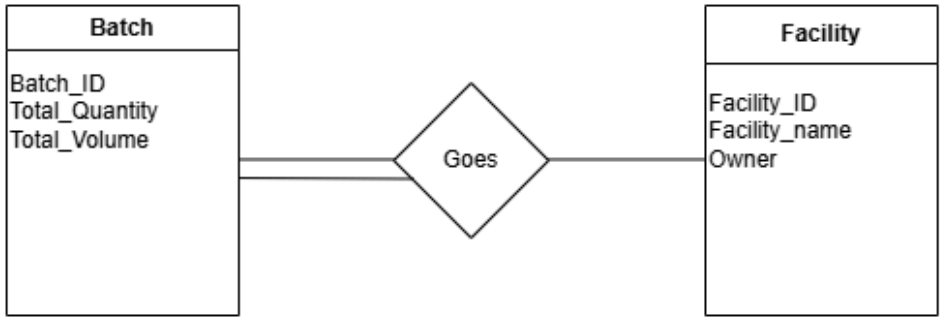
- Form(Harvest_ID,Batch_ID)

4) Batch - Inventory



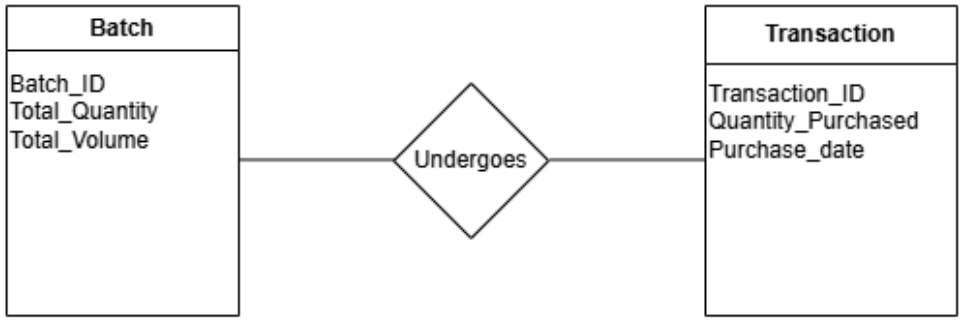
- Stores_IN(Batch_ID)

5) Batch - Facility



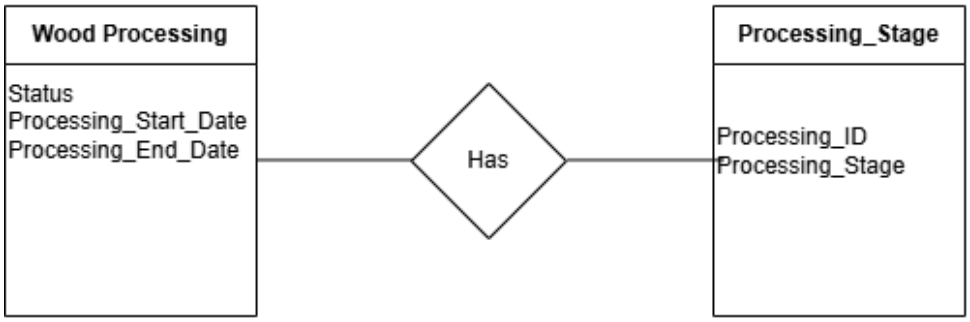
- Goes(Batch_ID,Facility_ID)

6) Batch - Transactions



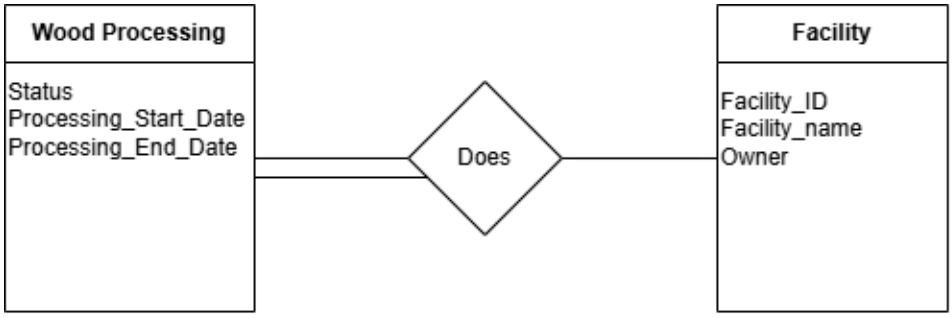
- Undergoes(Batch_ID,Transaction_ID)

7) Wood_Processing - Proceessing_Stage



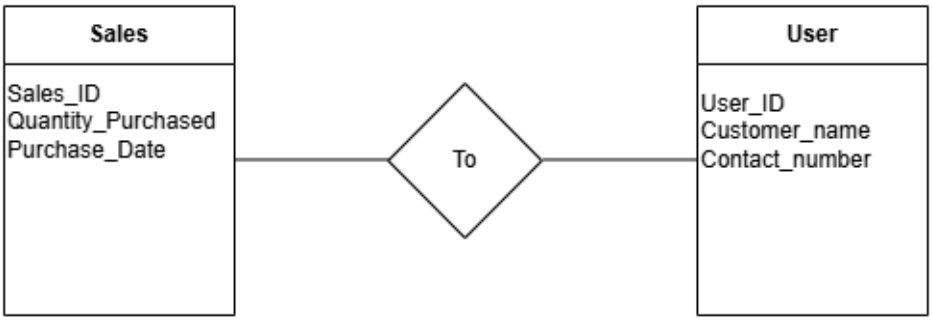
- Has(<u>Processing_ID</u>)

8) Wood_Processing - Facility



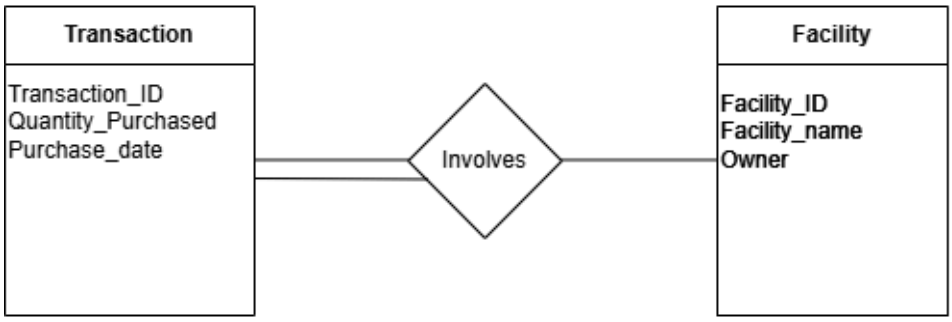
- Does(<u>Batch_ID,Facility_ID</u>)

9) Sales - User



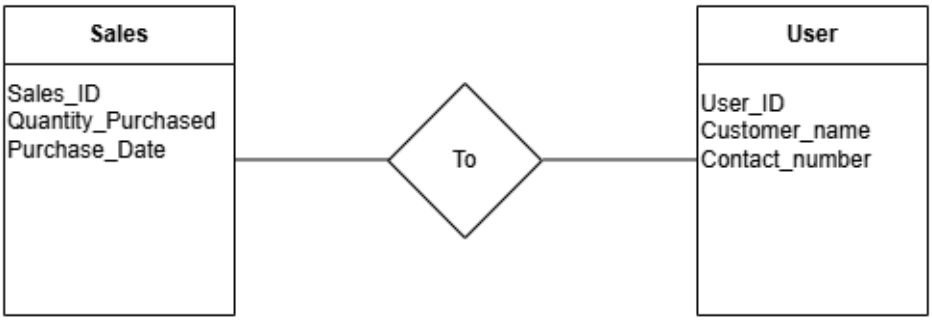
- Goes(Sales_ID,User_ID)

10) Sales - User



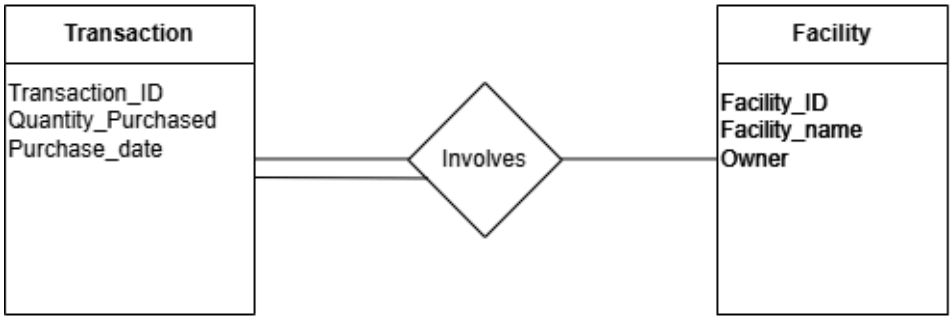
- Involves(Transaction_ID,Facility_ID)

11) Sales - User



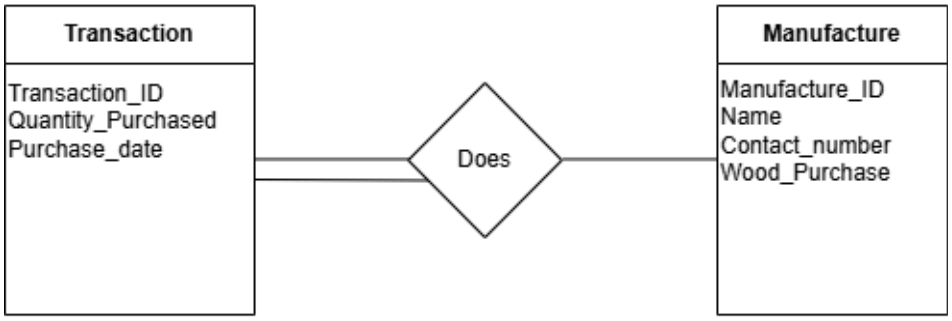
- Goes(Sales_ID,User_ID)

12) Transaction - Facility



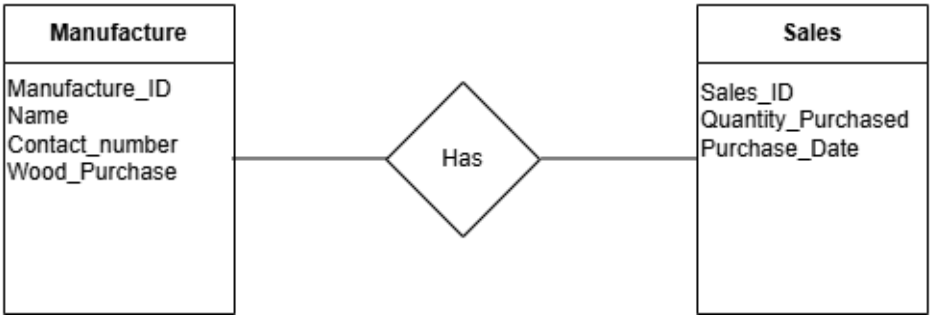
- Involves(Transaction_ID,Facility_ID)

13) Transaction - Manufacture



- Involves(Transaction_ID,Manufacture_ID)

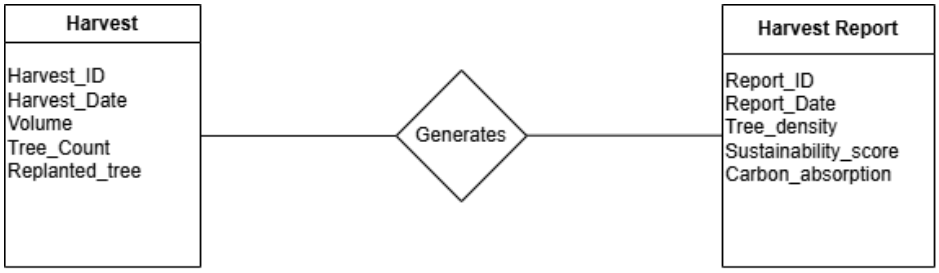
14) Manufacture - Sales



- Has(Manufacture_ID,Sales_ID)

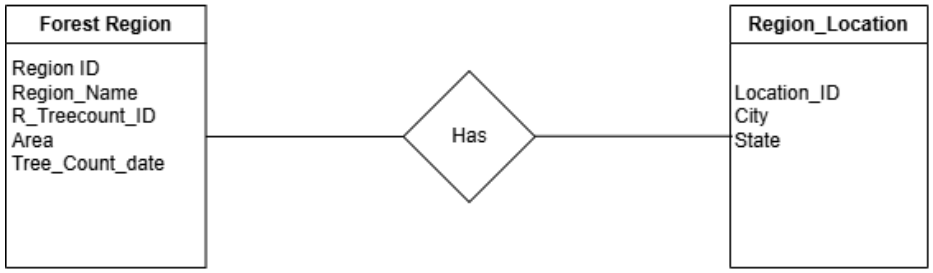
One to One
Relation table

1) Harvest - Harvest_report



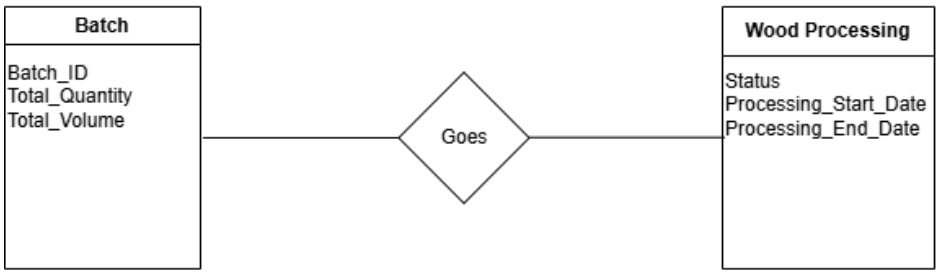
- Generates(Harvest_ID,Report_ID)

2) Forest_Region - Region_Location



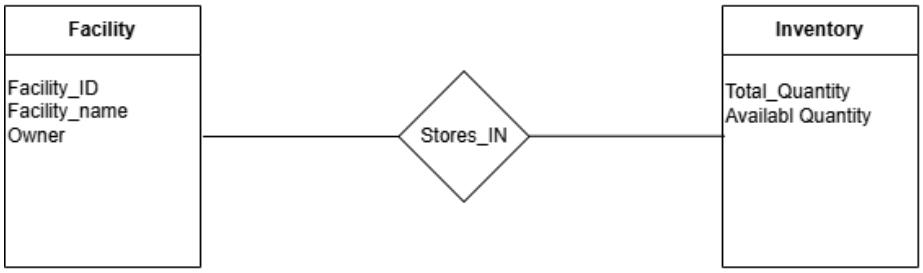
- Has(Region_ID,Location_ID)

3) Batch - Wood_Processing



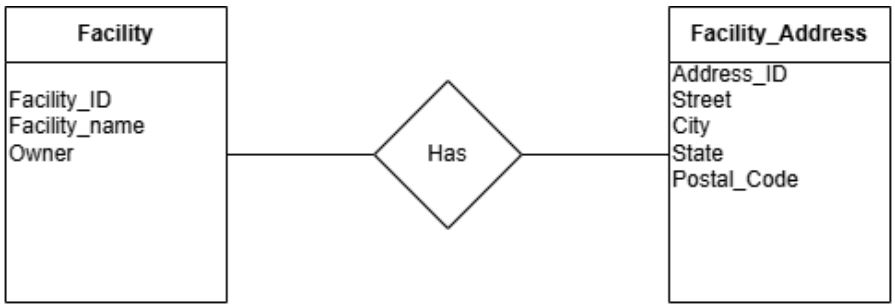
- Goes(Batch_ID)

4) Facility - Inventory



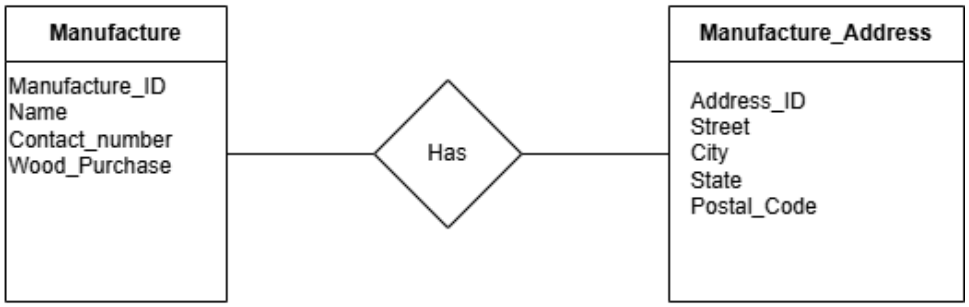
- Generates(Facility_ID)

5) Facility - Facility_Address



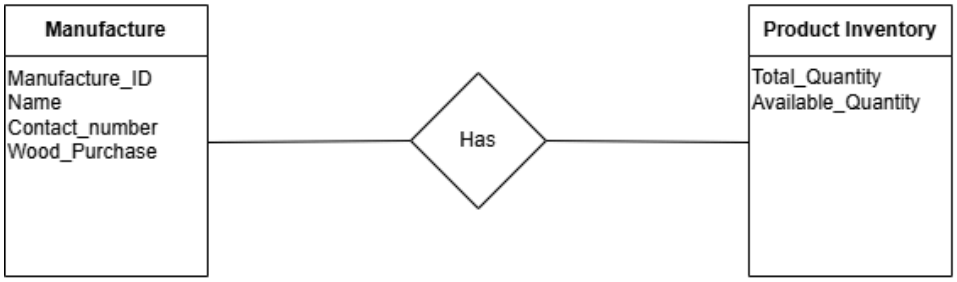
- Has(Facility_ID,Addres_ID)

6) Manufacture - Manufacture_Address



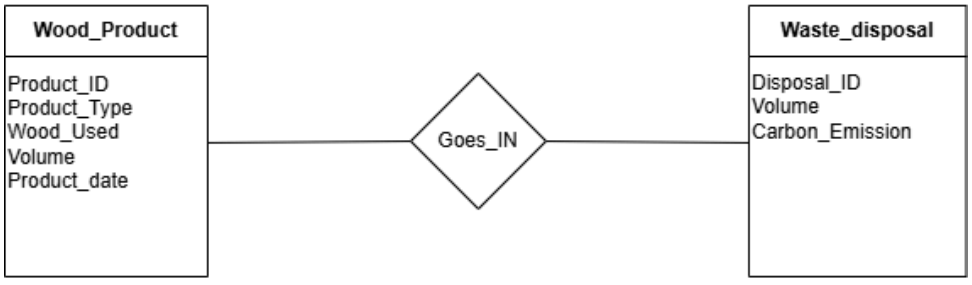
- Has(Manufacture_ID,Addres_ID)

7) Manufacture - Product_Inventory



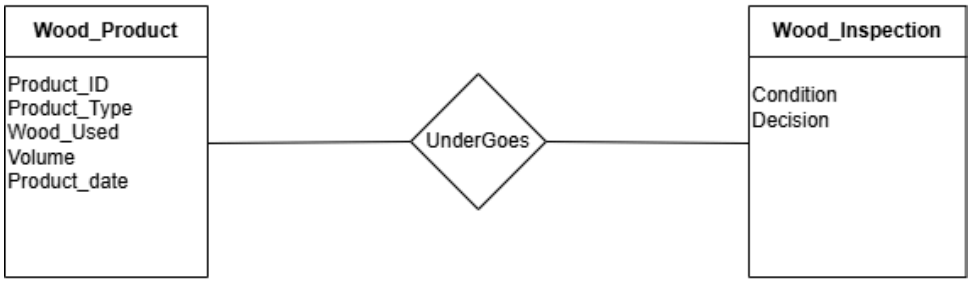
- Has(Manufacture_ID)

8) Wood_Product - Waste_Disposal



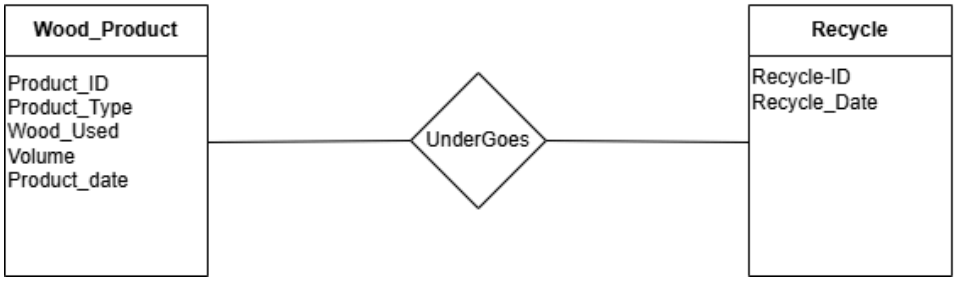
- Goes(Product_ID,Disposal_ID)

9) Wood_Product - Wood_Inspection



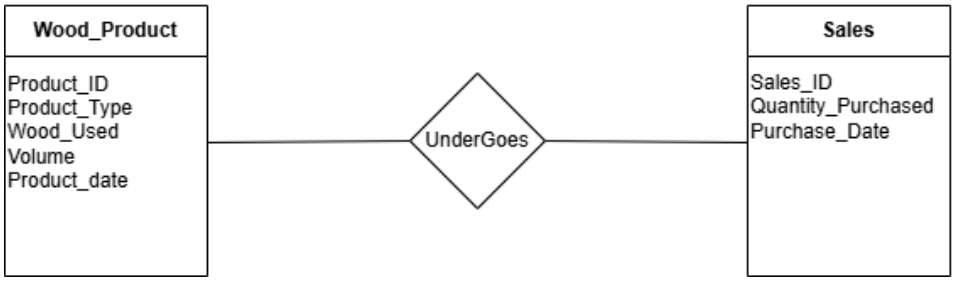
- Undergoes(Product_ID)

10) Wood_Product - Recycle



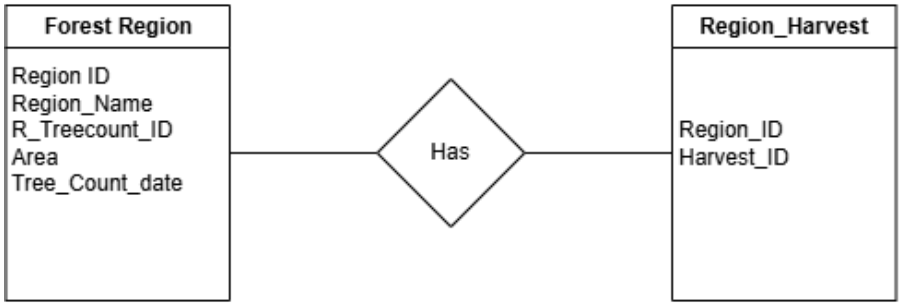
- Undergoes(Product_ID,Recycle_ID)

11) Wood_Product - Sales



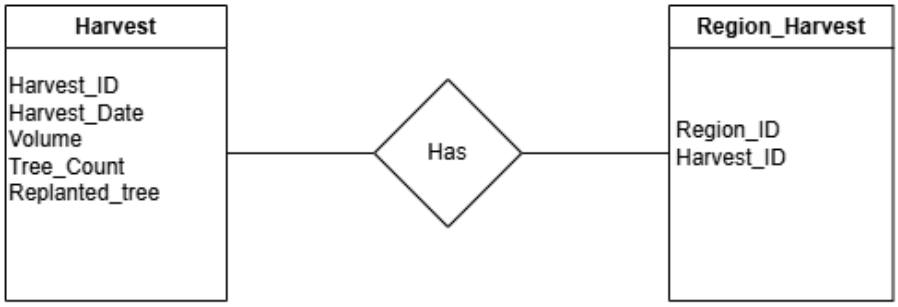
- Undergoes(Product_ID,Sales_ID)

12) Forest_Region - Region_Harvest



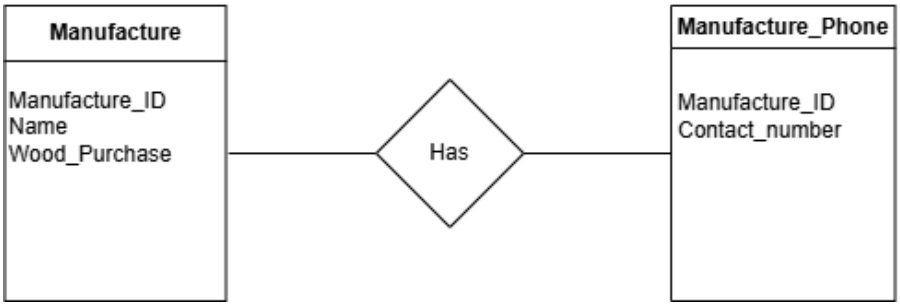
- Has(Region_ID)

13) Harvest - Region_Harvest



- Has(Harvest_ID)

14) Manufacture - Manufacture_Contact



- Has(Manufacture_ID)

Identify total participation & improve ERD Version 2.

1 region and region_location-both have
2 region and tree count-both have
3 region and
4 harvest and batch
5 Facility and Address: If there's a thick line between these entities, it implies that every facility must have an address, indicating that address participation is total in relation to a facility.

6 Batch and Facility: A thick line might indicate that every batch must be associated with a facility, suggesting total participation of the Batch entity in the relationship with Facility.

7 Logger and Logger Contact: If there's a thick line here, it would imply that each logger must have a contact in the system, making Logger totally participate in the Logger Contact relationship.

8 Wood Product and Sales: If there’s a thick line between these entities, it suggests that every wood product must be involved in a sales transaction, showing total participation of Wood Product in the Sales relationship.

9 Region and Tree Count: A thick line between these entities would imply that every region has a corresponding tree count, making Region fully participate in the Tree Count relationship.

10 Region and Tree Count

Total participation of Tree Count in Region, meaning every tree count record must be associated with a region.

Region and Harvest Report

Total participation of Harvest Report in Region, meaning every harvest report must be associated with a specific region.
Facility and Address

Total participation of Facility in Address, indicating every facility must have an address associated with it.
Batch and Facility

Total participation of Batch in Facility, suggesting that every batch must be stored in a facility.
Logger and Logger Contact

Total participation of Logger Contact in Logger, meaning each logger in the system is expected to have a contact entry.
Wood Product and Sales

Total participation of Sales in Wood Product, indicating every wood product must eventually be sold or assigned a sales record.
Harvest and Region

Total participation of Harvest in Region, which means each harvest event must be associated with a specific region, possibly for tracking sustainability practices.
Harvest and Batch

Total participation of Batch in Harvest, suggesting every batch is created as a result of a specific harvest or group of harvests.
Manufacturer and Manufacturer Address

Total participation of Manufacturer in Manufacturer Address, implying that every manufacturer must have an associated address in the system.
Wood Product and Wood Inspection

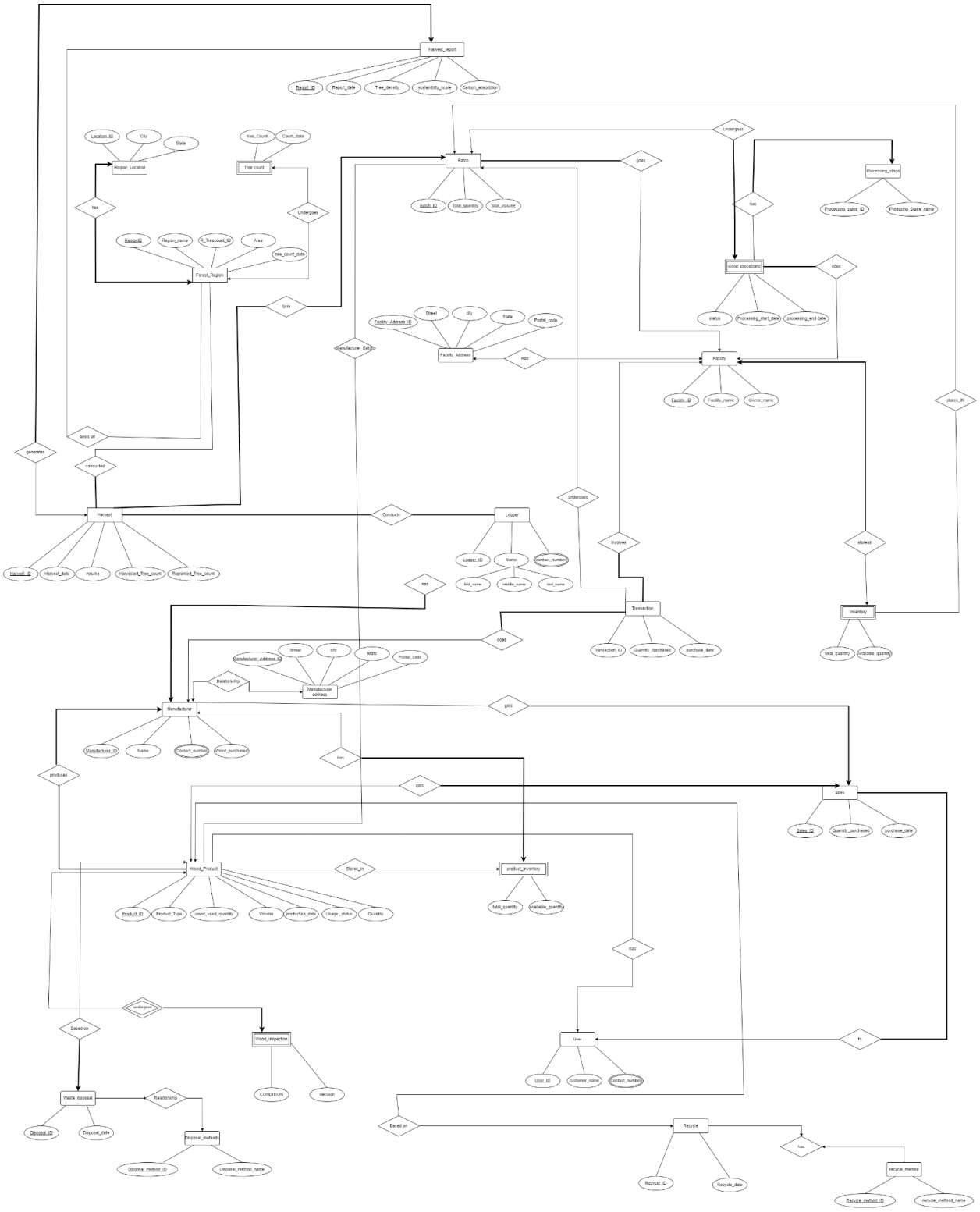
Total participation of Wood Product in Wood Inspection, meaning each wood product must go through an inspection before disposal or recycling.
Product Inventory and Wood Product

Total participation of Product Inventory in Wood Product, which implies that every wood product must have an inventory record.
Transaction and Manufacturer

Total participation of Transaction in Manufacturer, suggesting that every transaction is associated with a manufacturer who purchases wood or processed materials.
Recycle and Recycle Method

Total participation of Recycle in Recycle Method, indicating that every recycling event uses a specific recycling method.
Disposal and Disposal Method

Total participation of Disposal in Disposal Method, meaning every disposal event must be carried out with a specified method.


Harvest_report
Report_ID
Report_date
Tree_density
sustainility_score
Carbon_absorbtion
Location_ID
City
State
Region_Location
has
Tree_Count
Count_date
Tree count
Undergoes
RegionID
Region_name
Is_treecount_ID
Area
tree_count_date
Forest_Region
Spots
Ranch
Batch_ID
Total_quantity
total_volume
goes
Undergoes
Processing_stage
Processing_stage_ID
Processing_Stage_name
has
wood_processing
does
status
Processing_start_date
processing_end date
Facility_Address_ID
Street
city
State
Postal_code
Facility_Address
Has
Facility
Facility_ID
Facility_name
Owner_name
stores_IN
Manufacturer_Batch
basis on
generates
conducted
Harvest
Conducts
Logger
undergoes
Harvest_ID
Harvest_date
volume
Harvested_Tree_count
Replanted_Tree_count
Logger_ID
Name
Contact_number
first_name
middle_name
last_name
has
involves
Transaction
Transaction_ID
Quantity_purchased
purchase_date
storesh
Inventory
total_quantity
available_quantity
does
Manufacturer_Address_ID
fstreet
city
State
Postal_code
Manufacturer address
Relationship
gets
Manufacturer
Manufacturer_ID
Name
Contact_number
Wood_purchased
has
produces
gets
sales
Sales_ID
Quantity_purchased
purchase_date
Wood_Product
Stored_in
product_inventory
Product_ID
Product_Type
wood_used_quantity
Volumes
production_date
Usage_status
Quantity
total_quantity
available_quantity
has
undergone
Based on
Wood_inspection
CONDITION
decision
User
User_ID
customer_name
Contact_number(s)
fo
Waste_disposal
Relationship
Disposal_ID
Disposal_date
Disposal_methods
Disposal_method_ID
Disposal_method_name
Based on
Recycle
Recycle_ID
Recycle_date
has
recycle_method
Recycle_method_ID
recycle_method_name

5. Create DDL Scripts

1. Region Table

```
CREATE TABLE Region (
   RegionID INT PRIMARY KEY,
   Region_name VARCHAR(255) NOT NULL,
   R_Treecount_ID INT,
   Area DECIMAL(10, 2),
   Location_ID INT,
   Report_ID INT,
   FOREIGN KEY (Location_ID) REFERENCES Region_Location(Location_ID),
   FOREIGN KEY (Report_ID) REFERENCES Harvest_Report(Report_ID)
);
```

2. Region_Location Table

```
CREATE TABLE Region_Location (
   Location_ID INT PRIMARY KEY,
   City VARCHAR(255),
   State VARCHAR(255)
);
```

3. Tree_Count Table

```
CREATE TABLE Tree_Count (
   Region_ID INT PRIMARY KEY,
   Tree_count INT,
   Date DATE,
   FOREIGN KEY (Region_ID) REFERENCES Region(RegionID)
);
```

4. Region_Harvest Table

```
CREATE TABLE Region_Harvest (
    Region_ID INT,
    Harvest_ID INT,
    PRIMARY KEY (Region_ID, Harvest_ID),
    FOREIGN KEY (Region_ID) REFERENCES Region(RegionID),
    FOREIGN KEY (Harvest_ID) REFERENCES Harvest(Harvest_ID)
);
```

5. Harvest Table

```
CREATE TABLE Harvest (
    Harvest_ID INT PRIMARY KEY,
    Batch_ID INT,
    Harvest_date DATE,
    Volume DECIMAL(10, 2),
    Harvested_Tree_count INT,
    Replanted_Tree_count INT,
    FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID)
);
```

6. Harvest_Report Table

```
CREATE TABLE Harvest_Report (
    Report_ID INT PRIMARY KEY,
    Harvest_ID INT,
    Report_date DATE,
    Tree_density DECIMAL(10, 2),
    Sustainability_score DECIMAL(10, 2),
    Carbon_absorption DECIMAL(10, 2),
    FOREIGN KEY (Harvest_ID) REFERENCES Harvest(Harvest_ID)
);
```

7. Region_Harvest_Report Table

```
CREATE TABLE Region_Harvest_Report (
   Region_ID INT,
   Harvest_Report_ID INT,
   PRIMARY KEY (Region_ID, Harvest_Report_ID),
   FOREIGN KEY (Region_ID) REFERENCES Region(RegionID),
   FOREIGN KEY (Harvest_Report_ID) REFERENCES
Harvest_Report(Report_ID)
);
```

8. Logger Table

```
CREATE TABLE Logger (
   Logger_ID INT PRIMARY KEY,
   First_name VARCHAR(255),
   Middle_name VARCHAR(255),
   Last_name VARCHAR(255)
);
```

9. Logger_Contact Table

```
CREATE TABLE Logger_Contact (
   Logger_ID INT,
   Contact_number VARCHAR(15),
   PRIMARY KEY (Logger_ID, Contact_number),
   FOREIGN KEY (Logger_ID) REFERENCES Logger(Logger_ID)
);
```

10. Harvest_Logger Table

```
CREATE TABLE Harvest_Logger (
   Harvest_ID INT,
   Logger_ID INT,
   PRIMARY KEY (Harvest_ID, Logger_ID),
   FOREIGN KEY (Harvest_ID) REFERENCES Harvest(Harvest_ID),
   FOREIGN KEY (Logger_ID) REFERENCES Logger(Logger_ID)
);
```

11. Batch Table

```
CREATE TABLE Batch (
   Batch_ID INT PRIMARY KEY,
   Facility_ID INT,
   Total_quantity INT,
   Total_volume DECIMAL(10, 2),
   FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID)
);
```

12. Facility Table

```
CREATE TABLE Facility (
   Facility_ID INT PRIMARY KEY,
   Address_ID INT,
   Facility_name VARCHAR(255),
   Owner_name VARCHAR(255),
   FOREIGN KEY (Address_ID) REFERENCES
Facility_Address(Facility_Address_ID)
);
```

13. Wood_Processing Table

```
CREATE TABLE Wood_Processing (
   Batch_ID INT,
   Facility_ID INT,
   Processing_Stage_ID INT,
   Status VARCHAR(255),
   Quantity INT,
   Processing_Start_date DATE,
   Processing_end_date DATE,
   PRIMARY KEY (Batch_ID, Facility_ID),
   FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID),
   FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID),
   FOREIGN KEY (Processing_Stage_ID) REFERENCES
Processing_Stage(Processing_stage_ID)
);
```

14. Processing_Stage Table

```
CREATE TABLE Processing_Stage (
   Processing_stage_ID INT PRIMARY KEY,
   Processing_stage_name VARCHAR(255)
);
```

15. Inventory Table

```
CREATE TABLE Inventory (
   Batch_ID INT,
   Facility_ID INT,
   Total_quantity INT,
   Available_quantity INT,
   PRIMARY KEY (Batch_ID, Facility_ID),
   FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID),
   FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID));
```

16. Transaction Table

```
CREATE TABLE Transaction (
    Transaction_ID INT PRIMARY KEY,
    Batch_ID INT,
    Facility_ID INT,
    Manufacturer_ID INT,
    Quantity_purchased INT,
    Purchase_date DATE,
    FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID),
    FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID),
    FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID)
);
```

17. Manufacturer_Batch Table

```
CREATE TABLE Manufacturer_Batch (
    Manufacturer_ID INT,
    Batch_ID INT,
    Quantity_owned INT,
    PRIMARY KEY (Manufacturer_ID, Batch_ID),
    FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID),
    FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID)
);
```

18. Manufacturer Table

```
CREATE TABLE Manufacturer (
   Manufacturer_ID INT PRIMARY KEY,
   Address_ID INT,
   Sales_ID INT,
   Name VARCHAR(255),
   Wood_purchased INT,
   FOREIGN KEY (Address_ID) REFERENCES
Manufacturer_Address(Manufacturer_Address_ID),
   FOREIGN KEY (Sales_ID) REFERENCES Sales(Sales_ID)
);
```

19. Wood_Product Table

```
CREATE TABLE Wood_Product (
   Product_ID INT PRIMARY KEY,
   Manufacturer_ID INT,
   User_ID INT,
   Product_Type VARCHAR(255),
   Wood_used_quantity INT,
   Volume DECIMAL(10, 2),
   Production_date DATE,
   Usage_Status VARCHAR(255),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID),
   FOREIGN KEY (User_ID) REFERENCES User(User_ID)
);
```

20. Product_Inventory Table

```
CREATE TABLE Product_Inventory (
   Product_ID INT,
   Manufacturer_ID INT,
   Total_quantity INT,
   Available_quantity INT,
   PRIMARY KEY (Product_ID, Manufacturer_ID),
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID)
);
```

21. User Table

```
CREATE TABLE User (
   User_ID INT PRIMARY KEY,
   Customer_name VARCHAR(255)
);
```

22. User_Contact Table

```
CREATE TABLE User_Contact (
   User_ID INT,
   Contact_number VARCHAR(15),
   PRIMARY KEY (User_ID, Contact_number),
   FOREIGN KEY (User_ID) REFERENCES User(User_ID)
);
```

23. Sales Table

```
CREATE TABLE Sales (
   Sales_ID INT PRIMARY KEY,
   User_ID INT,
   Product_ID INT,
   Purchase_date DATE,
   Quantity_purchased INT,
   FOREIGN KEY (User_ID) REFERENCES User(User_ID),
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID)
);
```

24. Wood_Inspection Table

```
CREATE TABLE Wood_Inspection (
   Product_ID INT,
   Condition VARCHAR(255),
   Decision VARCHAR(255),
   PRIMARY KEY (Product_ID),
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID)
);
```

25. Waste_Disposal Table

```
CREATE TABLE Waste_Disposal (
   Disposal_ID INT PRIMARY KEY,
   Product_ID INT,
   Disposal_Method_ID INT,
   Disposal_Date DATE,
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID),
   FOREIGN KEY (Disposal_Method_ID) REFERENCES
Disposal_Method(Disposal_Method_ID)
);
```

26. Disposal_Method Table

```
CREATE TABLE Disposal_Method (
   Disposal_Method_ID INT PRIMARY KEY,
   Disposal_Method_Name VARCHAR(255)
);
```

27. Recycle_Method Table

```
CREATE TABLE Recycle_Method (
   Recycle_Method_ID INT PRIMARY KEY,
   Recycle_Method_Name VARCHAR(255)
);
```

28. Recycle Table

```
CREATE TABLE Recycle (
   Recycle_ID INT PRIMARY KEY,
   Product_ID INT,
   Recycle_Method_ID INT,
   Recycle_Date DATE,
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID),
   FOREIGN KEY (Recycle_Method_ID) REFERENCES
Recycle_Method(Recycle_Method_ID)
);
```

29. Facility_Address Table

```
CREATE TABLE Facility_Address (
   Facility_Address_ID INT PRIMARY KEY,
   Street VARCHAR(255),
   City VARCHAR(255),
   State VARCHAR(255),
   Postal_code VARCHAR(10));
```

30. Manufacturer_Phone Table

```
CREATE TABLE Manufacturer_Phone (
   Manufacturer_ID INT,
   Contact_number VARCHAR(15),
   PRIMARY KEY (Manufacturer_ID, Contact_number),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID)
);
```

31. Manufacturer_Address Table

```
CREATE TABLE Manufacturer_Address (
   Manufacturer_Address_ID INT PRIMARY KEY,
   Street VARCHAR(255),
   City VARCHAR(255),
   State VARCHAR(255),
   Postal_code VARCHAR(10)
);
```

Chapter 3: Normalization of Database

Identify and list all types of dependencies (PK, FK, Functional Dependencies) for each relation

1. Region

- PK: RegionID
- FK:Location_ID,Report_ID
- FD:RegionID → Region_name, R_Treecount_ID, Area, Location_ID, Report_ID

2 Region_location

- PK:Location_ID
- FD:Location_id → city, state

3 Tree count

- PK:R_treecount_ID
- FD:R_treecount_ID → ,region_ID,tree_count,Date

4 Harvest

- PK:Harvest_ID
- FK:Batch_ID
- FD:Harvest_ID → Batch_ID, Harvest_date, volume, Harvested_Tree_count, Replanted_Tree_count

5 Region_Harvest

- PK:(Region_ID,Harvest_ID)

7 Harvest_Report Table

- PK:Report_ID
- FK:Harvest_ID
- FD:Report_ID → Harvest_ID, report_date, tree_density, sustainability_score, carbon_absorption

8 Logger Table

- PK:Logger_ID
- FD:Logger_ID → first_name, middle_name, last_name

9 Logger_contact Table

- PK:(logger_ID, contact_number)

10 Harvest_logger Table

- PK:(harvest_ID, logger_ID)

11 Batch Table

- PK:Batch_ID
- FK:Facility_ID
- FD:Batch_ID → Facility_ID, total_quantity, total_volume

12 Facility Table

- PK:Facility_ID
- FK:Address_ID
- FD:Facility_ID → Address_ID, Facility_name, Owner_name

13 Wood_processing

- PK:Batch_ID, facility_id, processing_stage_id
- FK:Batch_ID, facility_id, processing_stage_id
- FD:Batch_ID, facility_id, processing_stage_id → status, processing_Start_date, processing_end_date

14 processing_stage Table

- PK:processing_stage_ID
- FD:processing_stage_ID → processing_stage_name

15 Inventory Table

- PK:(Batch_ID, Facility_ID)
- FK:(Batch_ID, Facility_ID)
- FD:Batch_ID, Facility_ID → total_quantity, available_quantity
- Batch_ID→ total_quantity, available_quantity

16 Transaction

- PK:Transaction_ID
- FK:Batch_ID, Facility_ID, Manufacturer_ID
- FD:Transaction_ID → Batch_ID, Facility_ID, Manufacturer_ID, quantity_purchased, purchase_date

17 Manufacturer

- PK: Manufacturer_ID
- FK: Address_ID, sales_ID
- FD:Manufacturer_ID → Address_ID, sales_ID, Name, wood_purchased

18 Wood_Product

- PK: Product_ID
- FK: Manufacturer_ID, User_ID
- FD:Product_ID → Manufacturer_ID, User_ID, Product_Type, Wood_used_quantity, volume, Production_date, usage_Status

19 Product_inventory

- PK: (Product_ID, Manufacturer_ID) (composite key)
- FD:(Product_ID, Manufacturer_ID) → total_quantity, available_quantity

20 User

- PK: User_ID
- FD:User_ID → Customer_name

21 User_contact

- PK: (User_ID, contact_number) (composite key)
- FD:(User_ID, contact_number)

22 Sales

- PK: sales_ID
- FK: User_ID, product_ID
- FD:sales_ID → User_ID, product_ID, Purchase_date, Quantity_purchased

23 Wood_inspection

- PK: Product_ID
- FD:Product_ID → condition, decision

24 Waste_disposal

- PK: Disposal_ID
- FK: product_ID
- FD:Disposal_ID → product_ID,disposal_method_id_disposal_method_date

25 Recycle_method

- PK: Recycle_method_ID
- FD:Recycle_method_ID → Recycle_method_name

26 Recycle

- PK: Recycle_ID
- FK: Product_ID, recycle_method_ID
- FD:Recycle_ID → Product_ID, recycle_method_ID, recycle_date

27 Address

- PK: Address_ID
- FD:Address_ID → Street, City, State, Postal_code

28 Manufacturer_phone

- PK: (manufacturer_ID, contact_number) (composite key)
- FD:(manufacturer_ID, contact_number)

29 tree count

- PK:R_tree_count_ID
- FD:R_tree_count_ID→tree_count,date

30 Manufacturer_Batch

- PK:(Manufacturer_ID,Batch_ID))PK
- FK:Manufacturer_ID,Batch_ID
- FD:(Manufacturer_ID(FK),Batch_ID(FK))PK→ quantity_owned

Normalize the database up to 1NF (scalar values)

1. Region

RegionID	Region_name	R_Treecount	Area	Location_ID	Report_ID
1	Northern	1000	5000	1	1
2	Southern	800	3000	2	2
3	Eastern	1200	4500	3	3

2. Region_location

Location_ID	city	state
1	Ahmedabad	Gujarat
2	Surat	Gujarat
3	Vadodara	Gujarat

3. Tree_count

R_tree_count_ID	region_ID	tree_count	Date
1	1	500	2024-01-01
2	2	300	2024-02-01
3	3	450	2024-03-01

4. Region_harvest

region_ID	harvest_ID
1	1
2	2
3	3

5. Harvest

Harvest_ID	Batch_ID	Harvest_date	volume	Harvested_Tree_count	Replanted_Tree_count
1	1	2024-01-15	2000	500	200
2	2	2024-02-20	1500	300	100
3	3	2024-03-25	2500	450	150

6. Harvest_Report

Report_ID	Harvest_ID	report_date	tree_density	sustainability_score	carbon_absorption
1	1	2024-01-20	200	85	400
2	2	2024-02-25	150	75	300
3	3	2024-03-30	180	80	350

7. Logger

Logger_ID	first_name	middle_name	last_name
1	Rahul	K.	Sharma
2	Sneha	R.	Patel
3	Amit	J.	Joshi

8. Logger_contact

logger_ID	contact_number
1	9876543210
2	9123456789
3	9988776655

9. Harvest_logger

harvest_ID	logger_ID
1	1
2	2
3	1

10. Batch_formation

Batch_ID	Facility_ID	total_quantity	total_volume
1	1	1000	5000
2	2	500	2500
3	1	750	3750

11. Facility

Facility_ID	Address_ID	Facility_name	Owner_name
1	1	Wood Factory	Ramesh
2	2	Timber Plant	Priya
3	3	Lumber Yard	Deepak

12. Wood_processing

Batch_ID	facility_id	processing_stage_id	status	quantity	processing_Start_date	processing_end_date
1	1	1	Active	1000	2024-01-10	2024-01-20
2	2	2	Completed	500	2024-02-05	2024-02-15
3	1	1	Active	750	2024-03-10	2024-03-20

13. Processing_stage

processing_stage_ID	processing_stage_name
1	Cutting
2	Shaping
3	Finishing

14. Inventory

Batch_ID	Facility_ID	total_quantity	available_quanti ty
1	1	1000	800
2	2	500	300
3	1	750	600

15. Transaction

Transaction_ID	Batch_ID	Facility_ID	Manufacturer_ID	quantity_purchased	purchase_date
1	1	1	1	200	2024-01-21
2	2	2	2	150	2024-02-26
3	1	1	3	300	2024-03-30

16. Manufacturer_Batch

Manufacturer_ID	Batch_ID	quantity_owned
1	1	200
2	2	150
3	1	300

17. Manufacturer

Manufacturer_ID	Address_ID	sales_ID	Name	wood_purchased
1	1	1	Timber Co.	1000
2	2	2	Woodworks Inc.	500
3	3	3	Lumber LLC	750

18. Wood_Product

Product_ID	Manufacturer_ID	User_ID	Product_Type	Wood_used_quantity	volume	Production_date	usage_Status
1	1	1	Chair	10	50	2024-02-01	In Use
2	2	2	Table	20	100	2024-02-10	In Use
3	3	1	Desk	15	75	2024-03-01	In Use

19. Product_inventory

Product_ID	Manufacturer_ID	total_quantity	available_quantity
1	1	100	80
2	2	200	150
3	3	150	120

20. User

User_ID	Customer_name
1	Aditi
2	Ravi
3	Snehal

21. User_contact

User_ID	contact_number
1	9876543210
2	8765432109
3	7654321098

22. Sales

sales_ID	User_ID	product_ID	Purchase_date	Quantity_purchased
1	1	1	2024-02-15	1
2	2	2	2024-03-20	2
3	3	3	2024-04-05	1

23. Wood_inspection

Product_ID	condition	decision
1	Good	Pass
2	Fair	Repair
3	Poor	Reject

24 Waste_disposal Table

Disposal_ID (PK)	Product_ID (FK)	Disposal_method_ID (FK)	Disposal_date
1	101	1	2024-06-15
2	102	2	2024-06-20
3	103	1	2024-07-10

25 Disposal_method Table

Disposal_method_ID (PK)	Disposal_method_name
1	Burning
2	Recycling
3	Composting

26. Recycle_method

Recycle_method_ID	Recycle_method_name
1	Shredding
2	Composting
3	Reuse

27. Recycle

Recycle_ID	Product_ID	recycle_method_ID	recycle_date
1	1	1	2024-05-01
2	2	2	2024-05-15
3	3	1	2024-06-01

28. Address

Address_ID	Street	City	State	Postal_code
1	123 Main St	Ahmedabad	Gujarat	380001
2	456 Oak St	Surat	Gujarat	395001
3	789 Pine St	Vadodara	Gujarat	390001

29. manufacturer_phone

manufacturer_ID	contact_number
1	9876543210
2	9123456789
3	9988776655

In a 1NF-compliant database, all tables must have:

1. Scalar (atomic) values in each cell: Each cell must contain a single value.
2. Unique rows: Each row should be uniquely identifiable, usually achieved by having a primary key.
3. No repeating groups: There should be no repeating columns or groups of columns related to the same type of data within a table.

Each table in this schema is normalized to 1NF:

- All tables have atomic values (single values in each cell).
- Primary keys ensure unique identification of each row, preventing duplicate entries.
- Foreign keys establish relationships across tables (e.g., linking Product_ID in different tables without repeating information).
- The given data is also normalized to Second Normal Form (2NF) because it meets the requirements of 1NF and resolves any partial dependencies. In 2NF, each table must satisfy two conditions: it should already be in 1NF, and all non-key attributes should depend entirely on the primary key, rather than just a part of it. In the current schema, tables such as Region, Batch_formation, Harvest_Report, and Inventory store information where every non-key attribute is fully dependent on the table's primary key. For instance, in the Region table, attributes like Region_name, Area, and R_Treecount depend entirely on RegionID. Similarly, in the Wood_Product table, attributes like Product_Type, Wood_used_quantity, and usage_Status are fully dependent on the composite primary key of Product_ID and Manufacturer_ID, ensuring the data adheres to 2NF.

- Additionally, tables designed to handle many-to-many relationships, such as harvest_logger and Transaction, eliminate partial dependencies by using composite keys where appropriate, ensuring that all non-key attributes depend fully on the entire composite key rather than a subset of it. This structure minimizes redundancy and organizes data so that each non-key attribute directly relates to the primary key, reducing the risk of insertion, update, and deletion anomalies. Overall, the schema adheres to 2NF, improving data integrity by ensuring a logical, dependency-driven data structure.

Normalize it further to 3NF/BCNF (Remove Transitive Dependencies)

- To demonstrate that each table is in Third Normal Form (3NF), we need to confirm that:
- Each table is already in 2NF (no partial dependencies).
- No transitive dependencies exist within each table.
- Each entity, like Region, Logger, Harvest, and Wood_Product, is structured to avoid redundancy by linking related attributes through primary and foreign keys, thus removing partial and transitive dependencies. This design improves data integrity and optimizes storage, as no attributes rely indirectly on primary keys. The schema effectively isolates independent data points while preserving essential relationships through properly defined keys and associations, making it efficient and consistent for use in managing wood product lifecycle data.

Write down final relations with the schema.

1. Region
    - RegionID (PK), Region_name, R_Treecount, Area, Location_ID (FK), Report_ID (FK)
2. Region_Location
    - Location_ID (PK), City, State
3. Tree_Count
    - R_tree_count_ID (PK), Region_ID (FK), Tree_count, Date
4. Region_Harvest
    - Region_ID (PK, FK), Harvest_ID (PK, FK)
5. Harvest
    - Harvest_ID (PK), Batch_ID, Harvest_date, Volume, Harvested_Tree_count, Replanted_Tree_count
6. Harvest_Report
    - Report_ID (PK), Harvest_ID (FK), Report_date, Tree_density, Sustainability_score, Carbon_absorption
7. Logger
    - Logger_ID (PK), First_name, Middle_name, Last_name
8. Logger_Contact
    - Logger_ID (PK, FK), Contact_number
9. Harvest_Logger
    - Harvest_ID (PK, FK), Logger_ID (PK, FK)
10. Batch
    - Batch_ID (PK), Facility_ID (FK), Total_quantity, Total_volume
11. Facility
    - Facility_ID (PK), Address_ID (FK), Facility_name, Owner_name
12. Wood_Processing
    - Batch_ID (PK, FK), Facility_ID (PK, FK), Processing_stage_ID (FK), Status, Quantity, Processing_Start_date, Processing_end_date
13. Processing_Stage

- Processing_stage_ID (PK), Processing_stage_name

14. Inventory
    - Batch_ID (PK, FK), Facility_ID (PK, FK), Total_quantity, Available_quantity
15. Transaction
    - Transaction_ID (PK), Batch_ID (FK), Facility_ID (FK), Manufacturer_ID (FK), Quantity_purchased, Purchase_date
16. Manufacturer_Batch
    - Manufacturer_ID (PK, FK), Batch_ID (PK, FK), Quantity_owned
17. Manufacturer
    - Manufacturer_ID (PK), Address_ID (FK), Sales_ID, Name, Wood_purchased
18. Wood_Product
    - Product_ID (PK), Manufacturer_ID (FK), User_ID (FK), Product_Type, Wood_used_quantity, Volume, Production_date, Usage_Status
19. Product_Inventory
    - Product_ID (PK, FK), Manufacturer_ID (PK, FK), Total_quantity, Available_quantity
20. User
    - User_ID (PK), Customer_name
21. User_Contact
    - User_ID (PK, FK), Contact_number
22. Sales
    - Sales_ID (PK), User_ID (FK), Product_ID (FK), Purchase_date, Quantity_purchased
23. Wood_Inspection
    - Product_ID (PK, FK), Condition, Decision
24. Waste_Disposal

    Disposal_ID (PK),Product_ID,(FK), ,Disposal_Method_ID (FK) ,Disposal_Date
25. Disposal_Method

    Disposal_Method_ID (PK) , Disposal_Method_Name
26. Recycle_Method
    - Recycle_method_ID (PK), Recycle_method_name

27. Recycle
    - Recycle_ID (PK), Product_ID (FK), Recycle_method_ID (FK), Recycle_date
28. Facility Address
    - facility_Address_ID (PK), Street, City, State, Postal_code
29. Manufacturer_Phone
    - Manufacturer_ID (PK, FK), Contact_number

30 Manufacturer_address
Manufacturer_Address(PK), Street, City, State, Postal_code

31 region_harvest_report
region_id,report_id

Chapter 4: Implementation of Database

Revised DDL Scripts

1. Region Table

```
CREATE TABLE Region (
    RegionID INT PRIMARY KEY,
    Region_name VARCHAR(255) NOT NULL,
    R_Treecount_ID INT,
    Area DECIMAL(10, 2),
    Location_ID INT,
    Report_ID INT,
    FOREIGN KEY (Location_ID) REFERENCES Region_Location(Location_ID),
    FOREIGN KEY (Report_ID) REFERENCES Harvest_Report(Report_ID)
);
```

2. Region_Location Table

```
CREATE TABLE Region_Location (
    Location_ID INT PRIMARY KEY,
    City VARCHAR(255),
    State VARCHAR(255)
);
```

3. Tree_Count Table

```
CREATE TABLE Tree_Count (
   Region_ID INT PRIMARY KEY,
   Tree_count INT,
   Date DATE,
   FOREIGN KEY (Region_ID) REFERENCES Region(RegionID)
);
```

4. Region_Harvest Table

```
CREATE TABLE Region_Harvest (
   Region_ID INT,
   Harvest_ID INT,
   PRIMARY KEY (Region_ID, Harvest_ID),
   FOREIGN KEY (Region_ID) REFERENCES Region(RegionID),
   FOREIGN KEY (Harvest_ID) REFERENCES Harvest(Harvest_ID)
);
```

5. Harvest Table

```
CREATE TABLE Harvest (
   Harvest_ID INT PRIMARY KEY,
   Batch_ID INT,
   Harvest_date DATE,
   Volume DECIMAL(10, 2),
   Harvested_Tree_count INT,
   Replanted_Tree_count INT,
   FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID)
);
```

6. Harvest_Report Table

```
CREATE TABLE Harvest_Report (
   Report_ID INT PRIMARY KEY,
   Harvest_ID INT,
   Report_date DATE,
   Tree_density DECIMAL(10, 2),
   Sustainability_score DECIMAL(10, 2),
   Carbon_absorption DECIMAL(10, 2),
   FOREIGN KEY (Harvest_ID) REFERENCES Harvest(Harvest_ID)
);
```

7. Region_Harvest_Report Table

```
CREATE TABLE Region_Harvest_Report (
   Region_ID INT,
   Harvest_Report_ID INT,
   PRIMARY KEY (Region_ID, Harvest_Report_ID),
   FOREIGN KEY (Region_ID) REFERENCES Region(RegionID),
   FOREIGN KEY (Harvest_Report_ID) REFERENCES
Harvest_Report(Report_ID)
);
```

8. Logger Table

```
CREATE TABLE Logger (
   Logger_ID INT PRIMARY KEY,
   First_name VARCHAR(255),
   Middle_name VARCHAR(255),
   Last_name VARCHAR(255)
);
```

9. Logger_Contact Table

```
CREATE TABLE Logger_Contact (
   Logger_ID INT,
   Contact_number VARCHAR(15),
   PRIMARY KEY (Logger_ID, Contact_number),
   FOREIGN KEY (Logger_ID) REFERENCES Logger(Logger_ID)
);
```

10. Harvest_Logger Table

```
CREATE TABLE Harvest_Logger (
   Harvest_ID INT,
   Logger_ID INT,
   PRIMARY KEY (Harvest_ID, Logger_ID),
   FOREIGN KEY (Harvest_ID) REFERENCES Harvest(Harvest_ID),
   FOREIGN KEY (Logger_ID) REFERENCES Logger(Logger_ID)
);
```

11. Batch Table

```
CREATE TABLE Batch (
   Batch_ID INT PRIMARY KEY,
   Facility_ID INT,
   Total_quantity INT,
   Total_volume DECIMAL(10, 2),
   FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID)
);
```

12. Facility Table

```
CREATE TABLE Facility (
   Facility_ID INT PRIMARY KEY,
   Address_ID INT,
   Facility_name VARCHAR(255),
   Owner_name VARCHAR(255),
   FOREIGN KEY (Address_ID) REFERENCES
Facility_Address(Facility_Address_ID)
);
```

13. Wood_Processing Table

```
CREATE TABLE Wood_Processing (
   Batch_ID INT,
   Facility_ID INT,
   Processing_Stage_ID INT,
   Status VARCHAR(255),
   Quantity INT,
   Processing_Start_date DATE,
   Processing_end_date DATE,
   PRIMARY KEY (Batch_ID, Facility_ID),
   FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID),
   FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID),
   FOREIGN KEY (Processing_Stage_ID) REFERENCES
Processing_Stage(Processing_stage_ID)
);
```

14. Processing_Stage Table

```
CREATE TABLE Processing_Stage (
   Processing_stage_ID INT PRIMARY KEY,
   Processing_stage_name VARCHAR(255));
```

15. Inventory Table

```
CREATE TABLE Inventory (
    Batch_ID INT,
    Facility_ID INT,
    Total_quantity INT,
    Available_quantity INT,
    PRIMARY KEY (Batch_ID, Facility_ID),
    FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID),
    FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID)
);
```

16. Transaction Table

```
CREATE TABLE Transaction (
    Transaction_ID INT PRIMARY KEY,
    Batch_ID INT,
    Facility_ID INT,
    Manufacturer_ID INT,
    Quantity_purchased INT,
    Purchase_date DATE,
    FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID),
    FOREIGN KEY (Facility_ID) REFERENCES Facility(Facility_ID),
    FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID)
);
```

17. Manufacturer_Batch Table

```
CREATE TABLE Manufacturer_Batch (
   Manufacturer_ID INT,
   Batch_ID INT,
   Quantity_owned INT,
   PRIMARY KEY (Manufacturer_ID, Batch_ID),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID),
   FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID)
);
```

18. Manufacturer Table

```
CREATE TABLE Manufacturer (
   Manufacturer_ID INT PRIMARY KEY,
   Address_ID INT,
   Sales_ID INT,
   Name VARCHAR(255),
   Wood_purchased INT,
   FOREIGN KEY (Address_ID) REFERENCES
Manufacturer_Address(Manufacturer_Address_ID),
   FOREIGN KEY (Sales_ID) REFERENCES Sales(Sales_ID)
);
```

19. Wood_Product Table

```
CREATE TABLE Wood_Product (
   Product_ID INT PRIMARY KEY,
   Manufacturer_ID INT,
   User_ID INT,
   Product_Type VARCHAR(255),
   Wood_used_quantity INT,
   Volume DECIMAL(10, 2),
   Production_date DATE,
   Usage_Status VARCHAR(255),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID),
   FOREIGN KEY (User_ID) REFERENCES User(User_ID)
);
```

20. Product_Inventory Table

```
CREATE TABLE Product_Inventory (
   Product_ID INT,
   Manufacturer_ID INT,
   Total_quantity INT,
   Available_quantity INT,
   PRIMARY KEY (Product_ID, Manufacturer_ID),
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID)
);
```

21. User Table

```
CREATE TABLE User (
   User_ID INT PRIMARY KEY,
   Customer_name VARCHAR(255)
);
```

22. User_Contact Table

```
CREATE TABLE User_Contact (
   User_ID INT,
   Contact_number VARCHAR(15),
   PRIMARY KEY (User_ID, Contact_number),
   FOREIGN KEY (User_ID) REFERENCES User(User_ID)
);
```

23. Sales Table

```
CREATE TABLE Sales (
   Sales_ID INT PRIMARY KEY,
   User_ID INT,
   Product_ID INT,
   Purchase_date DATE,
   Quantity_purchased INT,
   FOREIGN KEY (User_ID) REFERENCES User(User_ID),
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID)
);
```

24. Wood_Inspection Table

```
CREATE TABLE Wood_Inspection (
   Product_ID INT,
   Condition VARCHAR(255),
   Decision VARCHAR(255),
   PRIMARY KEY (Product_ID),
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID)
);
```

25. Waste_Disposal Table

```
CREATE TABLE Waste_Disposal (
   Disposal_ID INT PRIMARY KEY,
   Product_ID INT,
   Disposal_Method_ID INT,
   Disposal_Date DATE,
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID),
   FOREIGN KEY (Disposal_Method_ID) REFERENCES
Disposal_Method(Disposal_Method_ID)
);
```

26. Disposal_Method Table

```
CREATE TABLE Disposal_Method (
   Disposal_Method_ID INT PRIMARY KEY,
   Disposal_Method_Name VARCHAR(255)
);
```

27. Recycle_Method Table

```
CREATE TABLE Recycle_Method (
   Recycle_Method_ID INT PRIMARY KEY,
   Recycle_Method_Name VARCHAR(255)
);
```

28. Recycle Table

```
CREATE TABLE Recycle (
   Recycle_ID INT PRIMARY KEY,
   Product_ID INT,
   Recycle_Method_ID INT,
   Recycle_Date DATE,
   FOREIGN KEY (Product_ID) REFERENCES Wood_Product(Product_ID),
   FOREIGN KEY (Recycle_Method_ID) REFERENCES
Recycle_Method(Recycle_Method_ID)
);
```

29. Facility_Address Table

```
CREATE TABLE Facility_Address (
   Facility_Address_ID INT PRIMARY KEY,
   Street VARCHAR(255),
   City VARCHAR(255),
   State VARCHAR(255),
   Postal_code VARCHAR(10)
);
```

30. Manufacturer_Phone Table

```
CREATE TABLE Manufacturer_Phone (
   Manufacturer_ID INT,
   Contact_number VARCHAR(15),
   PRIMARY KEY (Manufacturer_ID, Contact_number),
   FOREIGN KEY (Manufacturer_ID) REFERENCES
Manufacturer(Manufacturer_ID)
);
```

31. Manufacturer_Address Table

```
CREATE TABLE Manufacturer_Address (
   Manufacturer_Address_ID INT PRIMARY KEY,
   Street VARCHAR(255),
   City VARCHAR(255),
   State VARCHAR(255),
   Postal_code VARCHAR(10)
);
```

Database Population

INSERT Statements

1.Insert in Loaction

```
INSERT INTO Region_Location (Location_ID, City, State) VALUES
(1, 'Mumbai', 'Maharashtra'),
(2, 'Delhi', 'Delhi'),
(3, 'Bengaluru', 'Karnataka'),
(4, 'Hyderabad', 'Telangana'),
(5, 'Ahmedabad', 'Gujarat'),
(6, 'Chennai', 'Tamil Nadu'),
(7, 'Kolkata', 'West Bengal'),
(8, 'Surat', 'Gujarat'),
(9, 'Pune', 'Maharashtra'),
(10, 'Jaipur', 'Rajasthan'),
(11, 'Lucknow', 'Uttar Pradesh'),
(12, 'Kanpur', 'Uttar Pradesh'),
(13, 'Nagpur', 'Maharashtra'),
(14, 'Indore', 'Madhya Pradesh'),
(15, 'Thane', 'Maharashtra'),
(16, 'Bhopal', 'Madhya Pradesh'),
(17, 'Visakhapatnam', 'Andhra Pradesh'),
(18, 'Pimpri-Chinchwad', 'Maharashtra'),
(19, 'Patna', 'Bihar'),
(20, 'Vadodara', 'Gujarat'),
(21, 'Ghaziabad', 'Uttar Pradesh'),
(22, 'Ludhiana', 'Punjab'),
(23, 'Agra', 'Uttar Pradesh'),
(24, 'Nashik', 'Maharashtra'),
(25, 'Faridabad', 'Haryana'),
(26, 'Meerut', 'Uttar Pradesh'),
(27, 'Rajkot', 'Gujarat'),
(28, 'Kalyan-Dombivli', 'Maharashtra'),
```

(29, 'Vasai-Virar', 'Maharashtra'),
(30, 'Varanasi', 'Uttar Pradesh'),
(31, 'Srinagar', 'Jammu & Kashmir'),
(32, 'Aurangabad', 'Maharashtra'),
(33, 'Dhanbad', 'Jharkhand'),
(34, 'Amritsar', 'Punjab'),
(35, 'Navi Mumbai', 'Maharashtra'),
(36, 'Allahabad', 'Uttar Pradesh'),
(37, 'Ranchi', 'Jharkhand'),
(38, 'Howrah', 'West Bengal'),
(39, 'Coimbatore', 'Tamil Nadu'),
(40, 'Jabalpur', 'Madhya Pradesh'),
(41, 'Gwalior', 'Madhya Pradesh'),
(42, 'Vijayawada', 'Andhra Pradesh'),
(43, 'Jodhpur', 'Rajasthan'),
(44, 'Madurai', 'Tamil Nadu'),
(45, 'Raipur', 'Chhattisgarh'),
(46, 'Kota', 'Rajasthan'),
(47, 'Guwahati', 'Assam'),
(48, 'Chandigarh', 'Chandigarh'),
(49, 'Solapur', 'Maharashtra'),
(50, 'Hubli-Dharwad', 'Karnataka'),
(51, 'Bareilly', 'Uttar Pradesh'),
(52, 'Moradabad', 'Uttar Pradesh'),
(53, 'Mysore', 'Karnataka'),
(54, 'Gurgaon', 'Haryana'),
(55, 'Aligarh', 'Uttar Pradesh'),
(56, 'Jalandhar', 'Punjab'),
(57, 'Tiruchirappalli', 'Tamil Nadu'),
(58, 'Bhubaneswar', 'Odisha'),
(59, 'Salem', 'Tamil Nadu'),
(60, 'Mira-Bhayandar', 'Maharashtra'),
(61, 'Thiruvananthapuram', 'Kerala'),
(62, 'Bhiwandi', 'Maharashtra'),
(63, 'Saharanpur', 'Uttar Pradesh'),

(64, 'Guntur', 'Andhra Pradesh'),
(65, 'Amravati', 'Maharashtra'),
(66, 'Bikaner', 'Rajasthan'),
(67, 'Noida', 'Uttar Pradesh'),
(68, 'Jamshedpur', 'Jharkhand'),
(69, 'Bhilai', 'Chhattisgarh'),
(70, 'Cuttack', 'Odisha'),
(71, 'Firozabad', 'Uttar Pradesh'),
(72, 'Kochi', 'Kerala'),
(73, 'Bhavnagar', 'Gujarat'),
(74, 'Dehradun', 'Uttarakhand'),
(75, 'Durgapur', 'West Bengal'),
(76, 'Asansol', 'West Bengal'),
(77, 'Nanded', 'Maharashtra'),
(78, 'Kolhapur', 'Maharashtra'),
(79, 'Ajmer', 'Rajasthan'),
(80, 'Gulbarga', 'Karnataka');

2.Insert in Address

INSERT INTO Address (Address_ID, Street, City, State, Postal_code) VALUES
(1, 'MG Road', 'Mumbai', 'Maharashtra', '400001'),
(2, 'Church Street', 'Bengaluru', 'Karnataka', '560001'),
(3, 'Connaught Place', 'Delhi', 'Delhi', '110001'),
(4, 'Park Street', 'Kolkata', 'West Bengal', '700016'),
(5, 'Anna Salai', 'Chennai', 'Tamil Nadu', '600002'),
(6, 'SG Highway', 'Ahmedabad', 'Gujarat', '380054'),
(7, 'Banjara Hills', 'Hyderabad', 'Telangana', '500034'),
(8, 'FC Road', 'Pune', 'Maharashtra', '411004'),
(9, 'C G Road', 'Ahmedabad', 'Gujarat', '380009'),
(10, 'JLN Marg', 'Jaipur', 'Rajasthan', '302017'),
(11, 'Hazratganj', 'Lucknow', 'Uttar Pradesh', '226001'),
(12, 'Civil Lines', 'Kanpur', 'Uttar Pradesh', '208001'),

(13, 'Sitabuldi', 'Nagpur', 'Maharashtra', '440012'),
(14, 'MP Nagar', 'Bhopal', 'Madhya Pradesh', '462011'),
(15, 'New Market', 'Bhopal', 'Madhya Pradesh', '462003'),
(16, 'Lawsons Bay', 'Visakhapatnam', 'Andhra Pradesh', '530017'),
(17, 'Kalyani Nagar', 'Pune', 'Maharashtra', '411006'),
(18, 'Boring Road', 'Patna', 'Bihar', '800001'),
(19, 'Alkapuri', 'Vadodara', 'Gujarat', '390007'),
(20, 'Kavi Nagar', 'Ghaziabad', 'Uttar Pradesh', '201002'),
(21, 'Mall Road', 'Amritsar', 'Punjab', '143001'),
(22, 'Sapna Sangeeta Road', 'Indore', 'Madhya Pradesh', '452001'),
(23, 'VIP Road', 'Raipur', 'Chhattisgarh', '492001'),
(24, 'Vasant Vihar', 'Thane', 'Maharashtra', '400601'),
(25, 'Model Town', 'Ludhiana', 'Punjab', '141002'),
(26, 'Hinjawadi', 'Pune', 'Maharashtra', '411057'),
(27, 'Gomti Nagar', 'Lucknow', 'Uttar Pradesh', '226010'),
(28, 'DLF Phase 1', 'Gurgaon', 'Haryana', '122002'),
(29, 'Sector 62', 'Noida', 'Uttar Pradesh', '201309'),
(30, 'Gandhi Nagar', 'Jammu', 'Jammu & Kashmir', '180004'),
(31, 'Bypass Road', 'Varanasi', 'Uttar Pradesh', '221002'),
(32, 'Akurdi', 'Pimpri-Chinchwad', 'Maharashtra', '411035'),
(33, 'Kaloor', 'Kochi', 'Kerala', '682017'),
(34, 'Ashok Nagar', 'Udaipur', 'Rajasthan', '313001'),
(35, 'Chandni Chowk', 'Delhi', 'Delhi', '110006'),
(36, 'Sadar Bazar', 'Meerut', 'Uttar Pradesh', '250001'),
(37, 'MG Road', 'Indore', 'Madhya Pradesh', '452007'),
(38, 'Bariatu', 'Ranchi', 'Jharkhand', '834009'),
(39, 'Belur', 'Howrah', 'West Bengal', '711202'),
(40, 'R S Puram', 'Coimbatore', 'Tamil Nadu', '641002'),
(41, 'Rajendra Nagar', 'Patna', 'Bihar', '800016'),
(42, 'Nungambakkam', 'Chennai', 'Tamil Nadu', '600034'),
(43, 'Vidhyadhar Nagar', 'Jaipur', 'Rajasthan', '302023'),
(44, 'Ring Road', 'Surat', 'Gujarat', '395002'),
(45, 'Mahanagar', 'Lucknow', 'Uttar Pradesh', '226006'),
(46, 'Sakchi', 'Jamshedpur', 'Jharkhand', '831001'),
(47, 'Kankarbagh', 'Patna', 'Bihar', '800020'),

(48, 'Ambattur', 'Chennai', 'Tamil Nadu', '600053'),
(49, 'Madhapur', 'Hyderabad', 'Telangana', '500081'),
(50, 'Race Course Road', 'Bengaluru', 'Karnataka', '560001'),
(51, 'Malad West', 'Mumbai', 'Maharashtra', '400064'),
(52, 'Dharampeth', 'Nagpur', 'Maharashtra', '440010'),
(53, 'Aliganj', 'Lucknow', 'Uttar Pradesh', '226024'),
(54, 'Worli', 'Mumbai', 'Maharashtra', '400018'),
(55, 'Golf Green', 'Kolkata', 'West Bengal', '700095'),
(56, 'Navi Peth', 'Pune', 'Maharashtra', '411030'),
(57, 'Pratap Nagar', 'Jodhpur', 'Rajasthan', '342004'),
(58, 'Sadar', 'Nagpur', 'Maharashtra', '440001'),
(59, 'Bhawanipur', 'Kolkata', 'West Bengal', '700025'),
(60, 'Velachery', 'Chennai', 'Tamil Nadu', '600042'),
(61, 'Marine Lines', 'Mumbai', 'Maharashtra', '400020'),
(62, 'Civil Lines', 'Agra', 'Uttar Pradesh', '282002'),
(63, 'Sector 9', 'Faridabad', 'Haryana', '121006'),
(64, 'Cantonment', 'Jabalpur', 'Madhya Pradesh', '482001'),
(65, 'Manewada', 'Nagpur', 'Maharashtra', '440024'),
(66, 'Bhagwan Nagar', 'Nagpur', 'Maharashtra', '440027'),
(67, 'Vyttila', 'Kochi', 'Kerala', '682019'),
(68, 'Saket', 'Delhi', 'Delhi', '110017'),
(69, 'Sector 15', 'Chandigarh', 'Chandigarh', '160015'),
(70, 'Rohini', 'Delhi', 'Delhi', '110085'),
(71, 'Sector 22', 'Gurgaon', 'Haryana', '122001'),
(72, 'Dharavi', 'Mumbai', 'Maharashtra', '400017'),
(73, 'Hawa Mahal Road', 'Jaipur', 'Rajasthan', '302002'),
(74, 'Uppal', 'Hyderabad', 'Telangana', '500039'),
(75, 'Lajpat Nagar', 'Delhi', 'Delhi', '110024'),
(76, 'Sector 5', 'Panchkula', 'Haryana', '134109'),
(77, 'Channi Himmat', 'Jammu', 'Jammu & Kashmir', '180015'),
(78, 'Bodakdev', 'Ahmedabad', 'Gujarat', '380054'),
(79, 'Vashi', 'Navi Mumbai', 'Maharashtra', '400703'),
(80, 'Bandra West', 'Mumbai', 'Maharashtra', '400050');

3.Insert in Logger

```
INSERT INTO Logger (Logger_ID, First_name, Middle_name, Last_name)
VALUES
(1, 'Raj', 'Kumar', 'Singh'),
(2, 'Anil', 'Pratap', 'Verma'),
(3, 'Sunil', 'Chandra', 'Sharma'),
(4, 'Vikram', 'Deep', 'Thakur'),
(5, 'Arun', 'Rajesh', 'Mishra'),
(6, 'Rakesh', 'Dev', 'Chauhan'),
(7, 'Amit', 'Singh', 'Yadav'),
(8, 'Ravi', 'Kumar', 'Patel'),
(9, 'Kiran', 'Lal', 'Choudhary'),
(10, 'Vijay', 'Kumar', 'Pandey'),
(11, 'Sandeep', 'Rao', 'Mehta'),
(12, 'Rajesh', 'Prasad', 'Jain'),
(13, 'Suresh', 'Chandra', 'Rathore'),
(14, 'Rohit', 'Dev', 'Bhatt'),
(15, 'Mahesh', 'Nath', 'Kapoor'),
(16, 'Dinesh', 'Narayan', 'Shukla'),
(17, 'Manoj', 'Vishnu', 'Joshi'),
(18, 'Prakash', 'Ratan', 'Desai'),
(19, 'Shiv', 'Nath', 'Mali'),
(20, 'Yogesh', 'Kiran', 'Bansal'),
(21, 'Nilesh', 'Kumar', 'Reddy'),
(22, 'Vinod', 'Pratap', 'Goyal'),
(23, 'Harish', 'Kumar', 'Dubey'),
(24, 'Naveen', 'Chandra', 'Pillai'),
(25, 'Mohan', 'Raj', 'Dhawan'),
(26, 'Sanjay', 'Anil', 'Kumar'),
(27, 'Ajay', 'Narayan', 'Choudhury'),
(28, 'Pankaj', 'Raj', 'Saxena'),
(29, 'Ashok', 'Kumar', 'Thakur'),
(30, 'Vivek', 'Lal', 'Rana'),
(31, 'Hemant', 'Rajesh', 'Tyagi'),
(32, 'Deepak', 'Narendra', 'Chandra'),
```

(33, 'Ganesh', 'Shiv', 'Rao'),
(34, 'Anup', 'Vikram', 'Bhagat'),
(35, 'Tarun', 'Kumar', 'Mehra'),
(36, 'Arvind', 'Kailash', 'Bora'),
(37, 'Balraj', 'Mohan', 'Trivedi'),
(38, 'Prem', 'Nath', 'Pathak'),
(39, 'Narendra', 'Dev', 'Sahu'),
(40, 'Sudhir', 'Prakash', 'Chopra'),
(41, 'Lokesh', 'Shankar', 'Nair'),
(42, 'Naresh', 'Raj', 'Yogi'),
(43, 'Jitendra', 'Singh', 'Rathod'),
(44, 'Harpreet', 'Singh', 'Maan'),
(45, 'Parag', 'Kumar', 'Ahluwalia'),
(46, 'Umesh', 'Prakash', 'Seth'),
(47, 'Suraj', 'Kumar', 'Mahajan'),
(48, 'Bhupendra', 'Singh', 'Gupta'),
(49, 'Shailesh', 'Kumar', 'Tripathi'),
(50, 'Tej', 'Singh', 'Deshmukh'),
(51, 'Kamlesh', 'Narayan', 'Rana'),
(52, 'Anirudh', 'Singh', 'Parihar'),
(53, 'Rajiv', 'Kumar', 'Pandit'),
(54, 'Sanjiv', 'Nath', 'Dixit'),
(55, 'Devendra', 'Kumar', 'Joshi'),
(56, 'Vishal', 'Kumar', 'Rao'),
(57, 'Kailash', 'Nath', 'Bharti'),
(58, 'Mahendra', 'Singh', 'Patil'),
(59, 'Yashpal', 'Kumar', 'Choudhary'),
(60, 'Gaurav', 'Raj', 'Rawat'),
(61, 'Santosh', 'Kumar', 'Negi'),
(62, 'Brijesh', 'Dev', 'Bhardwaj'),
(63, 'Raghav', 'Shankar', 'Nath'),
(64, 'Pranav', 'Vikram', 'Shah'),
(65, 'Sharad', 'Kumar', 'Solanki'),
(66, 'Deepesh', 'Ratan', 'Shinde'),
(67, 'Sachin', 'Kumar', 'Tiwari'),

(68, 'Kartik', 'Nath', 'Singh'),
(69, 'Mayank', 'Raj', 'Mishra'),
(70, 'Roshan', 'Nath', 'Verma'),
(71, 'Bharat', 'Narayan', 'Bisht'),
(72, 'Ashish', 'Vikram', 'Malhotra'),
(73, 'Niraj', 'Pratap', 'Arora'),
(74, 'Chirag', 'Singh', 'Jain'),
(75, 'Sumit', 'Narendra', 'Kaur'),
(76, 'Kamal', 'Raj', 'Purohit'),
(77, 'Navin', 'Prakash', 'Dhingra'),
(78, 'Rahul', 'Kumar', 'Mehta'),
(79, 'Parth', 'Singh', 'Chaudhary'),
(80, 'Ishaan', 'Nath', 'Bajpai');

4. Insert in Disposal_Method

INSERT INTO Disposal_Method (Disposal_Method_ID,
Disposal_Method_Name) VALUES
(1, 'Landfill'),
(2, 'Incineration'),
(3, 'Composting'),
(4, 'Biodegradation'),
(5, 'Chemical Treatment');

5.Insert in Recycle Method

INSERT INTO Recycle_Method (Recycle_Method_ID, Recycle_Method_Name)
VALUES
(1, 'Mechanical Recycling'),
(2, 'Chemical Recycling'),
(3, 'Thermal Recycling'),
(4, 'Biological Recycling'),
(5, 'Energy Recovery');

6.Insert in Facility

```
INSERT INTO Facility (Facility_ID, Address_ID, Facility_name, Owner_name)
VALUES
(1, 1, 'GreenWood Processing Plant', 'Rajesh Kumar'),
(2, 2, 'Eco Timber Manufacturing', 'Amit Verma'),
(3, 3, 'Woodland Resources', 'Ravi Patel'),
(4, 4, 'Forest Works Inc.', 'Suresh Sharma'),
(5, 5, 'PrimeWood Solutions', 'Anil Gupta'),
(6, 6, 'NatureBuild Facility', 'Manoj Thakur'),
(7, 7, 'EcoPly Industries', 'Deepak Joshi'),
(8, 8, 'Sunrise Wood Plant', 'Naveen Mehta'),
(9, 9, 'GreenLine Woodworks', 'Kiran Rao'),
(10, 10, 'Forest Products Ltd.', 'Vinod Jain'),
(11, 11, 'TimberCrafts', 'Harish Patel'),
(12, 12, 'Earthwise Solutions', 'Ashok Kapoor'),
(13, 13, 'NatureWood Inc.', 'Arvind Desai'),
(14, 14, 'Timber Dynamics', 'Vijay Malhotra'),
(15, 15, 'EnviroWood', 'Santosh Rao'),
(16, 16, 'Wood Source Corp.', 'Rakesh Yadav'),
(17, 17, 'GreenLife Manufacturing', 'Mohan Pillai'),
(18, 18, 'TimberTech Ltd.', 'Sanjay Tyagi'),
(19, 19, 'Green Wood Solutions', 'Anand Tiwari'),
(20, 20, 'Earth Timber', 'Hemant Sharma'),
(21, 21, 'EcoBoard Industries', 'Parag Dubey'),
(22, 22, 'Sustainable Timber Ltd.', 'Bhavesh Jain'),
(23, 23, 'BioPly Ltd.', 'Ravi Singh'),
(24, 24, 'WoodSmart Facility', 'Ajay Kumar'),
(25, 25, 'Forest Friendly Plant', 'Tej Chauhan'),
(26, 26, 'Timber Alliance', 'Rajiv Kumar'),
(27, 27, 'BioWood Works', 'Narendra Gupta'),
(28, 28, 'Natural Resources Facility', 'Aman Verma'),
(29, 29, 'Eco Wood Systems', 'Sunil Rathi'),
(30, 30, 'GreenCircle Products', 'Balraj Mehta'),
(31, 31, 'Eco Timber Works', 'Deepak Choudhury'),
(32, 32, 'Sustainable Solutions', 'Shyam Deshmukh'),
```

(33, 33, 'Forest Innovations', 'Hemant Kapoor'),
(34, 34, 'WoodEco Plant', 'Raj Prasad'),
(35, 35, 'Prime Ply Solutions', 'Sachin Rao'),
(36, 36, 'TimberPro Ltd.', 'Vivek Chopra'),
(37, 37, 'NaturalPlank', 'Lokesh Jain'),
(38, 38, 'BioResource Corp.', 'Shiv Agrawal'),
(39, 39, 'Green Timbers', 'Navin Bansal'),
(40, 40, 'EcoWise Plant', 'Vikas Tripathi'),
(41, 41, 'WoodPlast Solutions', 'Rohit Das'),
(42, 42, 'Eco Forest Products', 'Yogesh Singh'),
(43, 43, 'ForestMax Solutions', 'Gaurav Thakur'),
(44, 44, 'TimberSource Industries', 'Anand Kaur'),
(45, 45, 'EcoCore Ltd.', 'Prakash Yogi'),
(46, 46, 'Green Innovations Facility', 'Harpreet Bajaj'),
(47, 47, 'BioWorks Inc.', 'Bharat Naidu'),
(48, 48, 'PureTimber Plant', 'Umesh Garg'),
(49, 49, 'Greenfield Products', 'Rajesh Tiwari'),
(50, 50, 'EcoTree Manufacturing', 'Suresh Desai'),
(51, 51, 'NatureWorks Facility', 'Ganesh Verma'),
(52, 52, 'SustainableCraft', 'Rohit Rao'),
(53, 53, 'Bio Timber Co.', 'Tejas Reddy'),
(54, 54, 'TimberWorks Facility', 'Vikram Choudhary'),
(55, 55, 'ForestWise Plant', 'Dev Patel'),
(56, 56, 'EcoGen Timber', 'Kailash Arora'),
(57, 57, 'NatureWood Solutions', 'Mahesh Malhotra'),
(58, 58, 'TimberCare Industries', 'Ravi Bhandari'),
(59, 59, 'EnviroTimber', 'Nitin Jain'),
(60, 60, 'Eco Dynamics', 'Pradeep Singh'),
(61, 61, 'GreenProcess Plant', 'Manoj Joshi'),
(62, 62, 'EcoPlanet Industries', 'Yash Sharma'),
(63, 63, 'ForestPly', 'Navin Agarwal'),
(64, 64, 'Prime Resources', 'Ajay Pandit'),
(65, 65, 'GreenCore Plant', 'Arun Kumar'),
(66, 66, 'Eco Resource Works', 'Sanjay Khanna'),
(67, 67, 'ForestPro Facility', 'Rakesh Mehta'),

(68, 68, 'WoodHub', 'Rohit Shah'),
(69, 69, 'Eco Innovations', 'Rahul Vyas'),
(70, 70, 'BioSphere Facility', 'Raghav Tripathi'),
(71, 71, 'EarthLine Timber', 'Kamlesh Yadav'),
(72, 72, 'GreenTrack Plant', 'Naveen Kumar'),
(73, 73, 'Eco Timber Processing', 'Vikrant Patel'),
(74, 74, 'GreenWays Manufacturing', 'Shailesh Rawat'),
(75, 75, 'TimberEdge', 'Anuj Verma'),
(76, 76, 'WoodRidge Plant', 'Prem Chandra'),
(77, 77, 'Forest Line Industries', 'Narendra Rai'),
(78, 78, 'Sustainable Timberworks', 'Brijesh Kaul'),
(79, 79, 'GreenMile Facility', 'Tarun Goel'),
(80, 80, 'WoodStock Inc.', 'Dharmendra Gupta');

7.Insert in Customer

INSERT INTO Customer (Customer_ID, Customer_name) VALUES
(1, 'Ravi Sharma'),
(2, 'Anita Verma'),
(3, 'Kiran Patel'),
(4, 'Sunil Joshi'),
(5, 'Meera Kapoor'),
(6, 'Vikram Gupta'),
(7, 'Priya Desai'),
(8, 'Rajesh Kumar'),
(9, 'Neha Rao'),
(10, 'Amit Singh'),

(11, 'Suresh Yadav'),
(12, 'Manisha Mehta'),
(13, 'Dinesh Agarwal'),
(14, 'Lata Prasad'),
(15, 'Vijay Bhatt'),
(16, 'Naveen Jain'),
(17, 'Pooja Malhotra'),
(18, 'Ramesh Chauhan'),
(19, 'Shalini Nair'),
(20, 'Vikas Chandra'),
(21, 'Aarti Pillai'),
(22, 'Rohit Mishra'),
(23, 'Nisha Tiwari'),
(24, 'Anil Kapoor'),
(25, 'Suman Das'),
(26, 'Ashok Sinha'),
(27, 'Seema Rai'),
(28, 'Mohan Dubey'),
(29, 'Geeta Reddy'),
(30, 'Harish Bhatia'),
(31, 'Lalit Joshi'),
(32, 'Pooja Deshmukh'),
(33, 'Aman Saxena'),
(34, 'Ritu Nanda'),
(35, 'Sanjay Verma'),
(36, 'Neelam Chaudhary'),
(37, 'Aakash Chopra'),
(38, 'Shivani Rathore'),
(39, 'Raj Malhotra'),
(40, 'Rekha Goyal'),
(41, 'Ravi Tripathi'),
(42, 'Sonia Mehra'),
(43, 'Devendra Solanki'),
(44, 'Karuna Rawat'),
(45, 'Nirmal Shetty'),

(46, 'Shreya Singh'),
(47, 'Ajay Kumar'),
(48, 'Bhavna Mehta'),
(49, 'Suraj Patel'),
(50, 'Vidya Pandey'),
(51, 'Rohan Kaur'),
(52, 'Priyanka Rathi'),
(53, 'Narendra Chauhan'),
(54, 'Meghna Dixit'),
(55, 'Alok Sharma'),
(56, 'Sarita Ghosh'),
(57, 'Rakesh Yadav'),
(58, 'Smita Jain'),
(59, 'Umesh Rane'),
(60, 'Manoj Shukla'),
(61, 'Indira Singh'),
(62, 'Ashwin Taneja'),
(63, 'Rashmi Nair'),
(64, 'Nikhil Kapoor'),
(65, 'Sneha Bansal'),
(66, 'Vishal Choudhary'),
(67, 'Kavita Sahu'),
(68, 'Arjun Dev'),
(69, 'Ishita Agarwal'),
(70, 'Tarun Chopra'),
(71, 'Namrata Verma'),
(72, 'Anand Narayan'),
(73, 'Leena Joshi'),
(74, 'Kailash Kothari'),
(75, 'Siddharth Shetty'),
(76, 'Rani Trivedi'),
(77, 'Mahesh Desai'),
(78, 'Sunita Vaidya'),
(79, 'Rahul Yadav'),
(80, 'Anjana Rana');

8.Insert in Logger

```
INSERT INTO Logger_Contact (Logger_ID, Contact_number) VALUES
(1, '9876543210'),
(2, '8765432109'),
(3, '7654321098'),
(4, '6543210987'),
(5, '5432109876'),
(6, '4321098765'),
(7, '3210987654'),
(8, '2109876543'),
(9, '1098765432'),
(10, '9988776655'),
(11, '8877665544'),
(12, '7766554433'),
(13, '6655443322'),
(14, '5544332211'),
(15, '4433221100'),
(16, '3322110099'),
(17, '2211009988'),
(18, '1100998877'),
(19, '9098765432'),
(20, '8098765432'),
(21, '7098765432'),
(22, '6098765432'),
(23, '5098765432'),
(24, '4087654321'),
(25, '3087654321'),
(26, '2087654321'),
(27, '1087654321'),
(28, '9987665544'),
(29, '8876554433'),
(30, '7765443322'),
(31, '6654332211'),
```

(32, '5543221100'),
(33, '4432110099'),
(34, '3321009988'),
(35, '2210098877'),
(36, '1100987766'),
(37, '9087654321'),
(38, '8076543210'),
(39, '7065432109'),
(40, '6054321098'),
(41, '5043210987'),
(42, '4032109876'),
(43, '3021098765'),
(44, '2010987654'),
(45, '1009876543'),
(46, '9998887776'),
(47, '8887776665'),
(48, '7776665554'),
(49, '6665554443'),
(50, '5554443332'),
(51, '4443332221'),
(52, '3332221110'),
(53, '2221110009'),
(54, '1110009998'),
(55, '9008877665'),
(56, '8007766554'),
(57, '7006655443'),
(58, '6005544332'),
(59, '5004433221'),
(60, '4003322110'),
(61, '3002211009'),
(62, '2001100998'),
(63, '1000098877'),
(64, '9988776654'),
(65, '8877665543'),
(66, '7766554432'),

```
(67, '6655443321'),
(68, '5544332210'),
(69, '4433221109'),
(70, '3322110098'),
(71, '2211009987'),
(72, '1100998876'),
(73, '9099887765'),
(74, '8098776654'),
(75, '7097665543'),
(76, '6096554432'),
(77, '5095443321'),
(78, '4084332210'),
(79, '3083221109'),
(80, '2082110098');
```

9.Insert in Processing_Stage

```
INSERT INTO Processing_Stage (Processing_stage_ID, Processing_stage_name)
VALUES
(1, 'Sawing'),
(2, 'Drying'),
(3, 'Planing'),
(4, 'Moulding'),
(5, 'Sanding'),
(6, 'Grading'),
(7, 'Treating'),
(8, 'Finishing');
```

10.Insert in Manufacturer_Address

INSERT INTO Manufacture_Address (Street, City, State, Postal_code) VALUES
('123 Green Lane', 'Mumbai', 'Maharashtra', '400001'),
('456 Oak Street', 'Delhi', 'Delhi', '110001'),
('789 Pine Avenue', 'Bengaluru', 'Karnataka', '560001'),
('101 Maple Drive', 'Chennai', 'Tamil Nadu', '600001'),
('202 Birch Road', 'Kolkata', 'West Bengal', '700001'),
('303 Cedar Boulevard', 'Hyderabad', 'Telangana', '500001'),
('404 Elm Street', 'Ahmedabad', 'Gujarat', '380001'),
('505 Walnut Crescent', 'Pune', 'Maharashtra', '411001'),
('606 Cherry Lane', 'Jaipur', 'Rajasthan', '302001'),
('707 Chestnut Court', 'Lucknow', 'Uttar Pradesh', '226001'),
('808 Willow Way', 'Surat', 'Gujarat', '395001'),
('909 Redwood Boulevard', 'Nagpur', 'Maharashtra', '440001'),
('1010 Fir Place', 'Indore', 'Madhya Pradesh', '452001'),
('1111 Ash Drive', 'Patna', 'Bihar', '800001'),
('1212 Hickory Street', 'Chandigarh', 'Chandigarh', '160001'),
('1313 Poplar Road', 'Bhopal', 'Madhya Pradesh', '462001'),
('1414 Pine Hill', 'Coimbatore', 'Tamil Nadu', '641001'),
('1515 Cedar Lane', 'Ludhiana', 'Punjab', '141001'),
('1616 Elmwood Drive', 'Gurugram', 'Haryana', '122001'),
('1717 Maple Street', 'Vijayawada', 'Andhra Pradesh', '520001'),
('1818 Oakridge Lane', 'Agra', 'Uttar Pradesh', '282001'),
('1919 Birch Court', 'Noida', 'Uttar Pradesh', '201301'),
('2020 Redwood Road', 'Vadodara', 'Gujarat', '390001'),
('2121 Willow Street', 'Meerut', 'Uttar Pradesh', '250001'),
('2222 Chestnut Hill', 'Kanpur', 'Uttar Pradesh', '208001'),
('2323 Ashwood Avenue', 'Madurai', 'Tamil Nadu', '625001'),
('2424 Fir Road', 'Bhubaneswar', 'Odisha', '751001'),
('2525 Pine Lane', 'Tiruchirappalli', 'Tamil Nadu', '620001'),
('2626 Cedar Crescent', 'Rajkot', 'Gujarat', '360001'),
('2727 Oakwood Drive', 'Thane', 'Maharashtra', '400601'),
('2828 Maple Road', 'Faridabad', 'Haryana', '121001'),
('2929 Redwood Street', 'Patiala', 'Punjab', '147001'),

('3030 Chestnut Road', 'Jammu', 'Jammu & Kashmir', '180001'),
('3131 Ashwood Crescent', 'Kochi', 'Kerala', '682001'),
('3232 Willow Hill', 'Vellore', 'Tamil Nadu', '632001'),
('3333 Poplar Avenue', 'Ranchi', 'Jharkhand', '834001'),
('3434 Pinewood Lane', 'Agartala', 'Tripura', '799001'),
('3535 Cedar Court', 'Durgapur', 'West Bengal', '713201'),
('3636 Oak Drive', 'Mysuru', 'Karnataka', '570001'),
('3737 Maple Crescent', 'Kochi', 'Kerala', '682005'),
('3838 Redwood Avenue', 'Kurnool', 'Andhra Pradesh', '518001'),
('3939 Chestnut Boulevard', 'Jalandhar', 'Punjab', '144001'),
('4040 Willow Court', 'Gaya', 'Bihar', '823001'),
('4141 Ashwood Hill', 'Kollam', 'Kerala', '691001'),
('4242 Fir Crescent', 'Ambala', 'Haryana', '134001'),
('4343 Pine Avenue', 'Dibrugarh', 'Assam', '786001'),
('4444 Cedar Street', 'Udaipur', 'Rajasthan', '313001'),
('4545 Oak Court', 'Jammu', 'Jammu & Kashmir', '180002'),
('4646 Maple Boulevard', 'Silchar', 'Assam', '788001'),
('4747 Redwood Crescent', 'Mangalore', 'Karnataka', '575001'),
('4848 Chestnut Avenue', 'Aurangabad', 'Maharashtra', '431001'),
('4949 Willow Boulevard', 'Siliguri', 'West Bengal', '734001'),
('5050 Ash Street', 'Bhopal', 'Madhya Pradesh', '462010'),
('5151 Fir Boulevard', 'Bareilly', 'Uttar Pradesh', '243001'),
('5252 Pinewood Avenue', 'Aligarh', 'Uttar Pradesh', '202001'),
('5353 Cedar Hill', 'Jamshedpur', 'Jharkhand', '831001'),
('5454 Oakwood Boulevard', 'Cuttack', 'Odisha', '753001'),
('5555 Maple Court', 'Guwahati', 'Assam', '781001'),
('5656 Redwood Crescent', 'Hazaribagh', 'Jharkhand', '825301'),
('5757 Chestnut Hill', 'Bikaner', 'Rajasthan', '334001'),
('5858 Willow Avenue', 'Sonipat', 'Haryana', '131001'),
('5959 Ash Lane', 'Raipur', 'Chhattisgarh', '492001'),
('6060 Fir Avenue', 'Jabalpur', 'Madhya Pradesh', '482001'),
('6161 Pine Boulevard', 'Puducherry', 'Puducherry', '605001'),
('6262 Cedar Boulevard', 'Kozhikode', 'Kerala', '673001'),
('6363 Oakwood Crescent', 'Gwalior', 'Madhya Pradesh', '474001'),
('6464 Maple Hill', 'Bhubaneswar', 'Odisha', '751004'),

('6565 Redwood Street', 'Shivpuri', 'Madhya Pradesh', '473551'),
('6666 Chestnut Lane', 'Karnal', 'Haryana', '132001'),
('6767 Willow Crescent', 'Navi Mumbai', 'Maharashtra', '400705'),
('6868 Ashwood Avenue', 'Raichur', 'Karnataka', '584101'),
('6969 Fir Hill', 'Baran', 'Rajasthan', '325205'),
('7070 Pine Lane', 'Dewas', 'Madhya Pradesh', '455001'),
('7171 Cedar Road', 'Tumkur', 'Karnataka', '572101'),
('7272 Oak Boulevard', 'Kochi', 'Kerala', '682003'),
('7373 Maple Drive', 'Jammu', 'Jammu & Kashmir', '180008'),
('7474 Redwood Avenue', 'Jabalpur', 'Madhya Pradesh', '482002'),
('7575 Chestnut Crescent', 'Bokaro', 'Jharkhand', '827001'),
('7676 Willow Boulevard', 'Gurugram', 'Haryana', '122018'),
('7777 Ashwood Lane', 'Patna', 'Bihar', '800005'),
('7878 Fir Boulevard', 'Nellore', 'Andhra Pradesh', '524001'),
('7979 Pinewood Road', 'Varanasi', 'Uttar Pradesh', '221001'),
('8080 Cedar Crescent', 'Jalna', 'Maharashtra', '431203');

11.Insert in Region

INSERT INTO Region (RegionID, Region_name, R_Treecount_ID, Area, Location_ID) VALUES
(1, 'Sundarbans', 1, 5000, 1),
(2, 'Western Ghats', 2, 2500, 2),
(3, 'Kailash Sacred Landscape', 3, 3000, 3),
(4, 'Kanha National Park', 4, 1200, 4),
(5, 'Ranthambore', 5, 1000, 5),
(6, 'Periyar Tiger Reserve', 6, 1100, 6),
(7, 'Corbett Tiger Reserve', 7, 1300, 7),
(8, 'Sariska Tiger Reserve', 8, 700, 8),
(9, 'Kaziranga National Park', 9, 1150, 9),

(10, 'Manas National Park', 10, 2800, 10),
(11, 'Bandhavgarh National Park', 11, 1050, 11),
(12, 'Gir National Park', 12, 1300, 12),
(13, 'Jim Corbett National Park', 13, 1300, 13),
(14, 'Nanda Devi and Valley of Flowers', 14, 3000, 14),
(15, 'Tropical Wet Evergreen Forests', 15, 1500, 15),
(16, 'Great Himalayan National Park', 16, 4200, 16),
(17, 'Nilgiri Biosphere Reserve', 17, 1300, 17),
(18, 'Madhya Pradesh Forest Region', 18, 15000, 18),
(19, 'North East India Forest Region', 19, 22000, 19),
(20, 'Rajasthan Arid Zone', 20, 3000, 20),
(21, 'Andaman & Nicobar Islands', 21, 8000, 21),
(22, 'Bhitarkanika Mangrove Reserve', 22, 500, 22),
(23, 'Madhya Pradesh', 23, 10000, 23),
(24, 'Kumarakom Bird Sanctuary', 24, 240, 24),
(25, 'Sundarbans Tiger Reserve', 25, 6000, 25),
(26, 'Loktak Lake Wetland', 26, 300, 26),
(27, 'Keoladeo National Park', 27, 290, 27),
(28, 'Chilika Lake', 28, 1000, 28),
(29, 'Assam Forests', 29, 12000, 29),
(30, 'North Western Ghats', 30, 2300, 30),
(31, 'Nanda Devi', 31, 800, 31),
(32, 'Sikkim Himalayan Biodiversity', 32, 1400, 32),
(33, 'Nilgiri Wildlife Sanctuary', 33, 2500, 33),
(34, 'Dudhwa National Park', 34, 1300, 34),
(35, 'Sikhote-Alin Biosphere Reserve', 35, 2500, 35),
(36, 'Sundarbans Reserve Forest', 36, 7000, 36),
(37, 'Tungabhadra Forests', 37, 1100, 37),
(38, 'Kumaon Himalayas', 38, 5000, 38),
(39, 'West Bengal Forests', 39, 11000, 39),
(40, 'Tamil Nadu Forests', 40, 15000, 40),
(41, 'Kerala Forests', 41, 12000, 41),
(42, 'Himachal Pradesh', 42, 6000, 42),
(43, 'Uttarakhand', 43, 10000, 43),
(44, 'Uttar Pradesh Forests', 44, 8000, 44),

(45, 'Gujarat Coastal Forests', 45, 4500, 45),
(46, 'Bihar Forests', 46, 6500, 46),
(47, 'Jharkhand Forests', 47, 8500, 47),
(48, 'Punjab Forests', 48, 4700, 48),
(49, 'Haryana Forests', 49, 3800, 49),
(50, 'Chhattisgarh Forests', 50, 6200, 50),
(51, 'Madhya Pradesh Teak Forests', 51, 9500, 51),
(52, 'Bengal Forests', 52, 7000, 52),
(53, 'Karnataka Forests', 53, 14000, 53),
(54, 'Maharashtra Forests', 54, 10500, 54),
(55, 'Madhya Pradesh Rainforest', 55, 4500, 55),
(56, 'Odisha Coastal Forests', 56, 5500, 56),
(57, 'Ganga River Basin Forests', 57, 9000, 57),
(58, 'Madhya Pradesh Deciduous Forests', 58, 12000, 58),
(59, 'Sikkim Himalayan Biozone', 59, 1000, 59),
(60, 'Arunachal Pradesh Rainforest', 60, 3000, 60),
(61, 'Assam Rainforests', 61, 4500, 61),
(62, 'Kerala Rainforests', 62, 8000, 62),
(63, 'Mizoram Forests', 63, 4200, 63),
(64, 'Nagaland Forests', 64, 3700, 64),
(65, 'Tripura Forests', 65, 2800, 65),
(66, 'Manipur Forests', 66, 3400, 66),
(67, 'Meghalaya Forests', 67, 2900, 67),
(68, 'Chhattisgarh Sal Forests', 68, 5000, 68),
(69, 'Bihar Deciduous Forests', 69, 4400, 69),
(70, 'Nagaland Evergreen Forests', 70, 1500, 70),
(71, 'Karnataka Dry Deciduous', 71, 10000, 71),
(72, 'Gujarat Scrub Forests', 72, 5300, 72),
(73, 'Andhra Pradesh Forests', 73, 7000, 73),
(74, 'Himalayan Temperate Forests', 74, 6500, 74),
(75, 'Rajasthan Forests', 75, 3000, 75),
(76, 'Assam Floodplain Forests', 76, 4000, 76),
(77, 'Jharkhand Mixed Forests', 77, 7000, 77),
(78, 'West Bengal Forests', 78, 9000, 78),
(79, 'Tropical Rainforests', 79, 12000, 79),

(80, 'Madhya Pradesh Dry Forests', 80, 7500, 80);

12.Insert in Region

INSERT INTO Region (RegionID, Region_name, R_Treecount_ID, Area, Location_ID) VALUES
(1, 'Sundarbans', 1, 5000, 1),
(2, 'Western Ghats', 2, 2500, 2),
(3, 'Kailash Sacred Landscape', 3, 3000, 3),
(4, 'Kanha National Park', 4, 1200, 4),
(5, 'Ranthambore', 5, 1000, 5),
(6, 'Periyar Tiger Reserve', 6, 1100, 6),
(7, 'Corbett Tiger Reserve', 7, 1300, 7),
(8, 'Sariska Tiger Reserve', 8, 700, 8),
(9, 'Kaziranga National Park', 9, 1150, 9),
(10, 'Manas National Park', 10, 2800, 10),
(11, 'Bandhavgarh National Park', 11, 1050, 11),
(12, 'Gir National Park', 12, 1300, 12),
(13, 'Jim Corbett National Park', 13, 1300, 13),
(14, 'Nanda Devi and Valley of Flowers', 14, 3000, 14),
(15, 'Tropical Wet Evergreen Forests', 15, 1500, 15),
(16, 'Great Himalayan National Park', 16, 4200, 16),
(17, 'Nilgiri Biosphere Reserve', 17, 1300, 17),
(18, 'Madhya Pradesh Forest Region', 18, 15000, 18),
(19, 'North East India Forest Region', 19, 22000, 19),
(20, 'Rajasthan Arid Zone', 20, 3000, 20),
(21, 'Andaman & Nicobar Islands', 21, 8000, 21),
(22, 'Bhitarkanika Mangrove Reserve', 22, 500, 22),
(23, 'Madhya Pradesh', 23, 10000, 23),
(24, 'Kumarakom Bird Sanctuary', 24, 240, 24),
(25, 'Sundarbans Tiger Reserve', 25, 6000, 25),
(26, 'Loktak Lake Wetland', 26, 300, 26),
(27, 'Keoladeo National Park', 27, 290, 27),
(28, 'Chilika Lake', 28, 1000, 28),
(29, 'Assam Forests', 29, 12000, 29),

(30, 'North Western Ghats', 30, 2300, 30),
(31, 'Nanda Devi', 31, 800, 31),
(32, 'Sikkim Himalayan Biodiversity', 32, 1400, 32),
(33, 'Nilgiri Wildlife Sanctuary', 33, 2500, 33),
(34, 'Dudhwa National Park', 34, 1300, 34),
(35, 'Sikhote-Alin Biosphere Reserve', 35, 2500, 35),
(36, 'Sundarbans Reserve Forest', 36, 7000, 36),
(37, 'Tungabhadra Forests', 37, 1100, 37),
(38, 'Kumaon Himalayas', 38, 5000, 38),
(39, 'West Bengal Forests', 39, 11000, 39),
(40, 'Tamil Nadu Forests', 40, 15000, 40),
(41, 'Kerala Forests', 41, 12000, 41),
(42, 'Himachal Pradesh', 42, 6000, 42),
(43, 'Uttarakhand', 43, 10000, 43),
(44, 'Uttar Pradesh Forests', 44, 8000, 44),
(45, 'Gujarat Coastal Forests', 45, 4500, 45),
(46, 'Bihar Forests', 46, 6500, 46),
(47, 'Jharkhand Forests', 47, 8500, 47),
(48, 'Punjab Forests', 48, 4700, 48),
(49, 'Haryana Forests', 49, 3800, 49),
(50, 'Chhattisgarh Forests', 50, 6200, 50),
(51, 'Madhya Pradesh Teak Forests', 51, 9500, 51),
(52, 'Bengal Forests', 52, 7000, 52),
(53, 'Karnataka Forests', 53, 14000, 53),
(54, 'Maharashtra Forests', 54, 10500, 54),
(55, 'Madhya Pradesh Rainforest', 55, 4500, 55),
(56, 'Odisha Coastal Forests', 56, 5500, 56),
(57, 'Ganga River Basin Forests', 57, 9000, 57),
(58, 'Madhya Pradesh Deciduous Forests', 58, 12000, 58),
(59, 'Sikkim Himalayan Biozone', 59, 1000, 59),
(60, 'Arunachal Pradesh Rainforest', 60, 3000, 60),
(61, 'Assam Rainforests', 61, 4500, 61),
(62, 'Kerala Rainforests', 62, 8000, 62),
(63, 'Mizoram Forests', 63, 4200, 63),
(64, 'Nagaland Forests', 64, 3700, 64),

(65, 'Tripura Forests', 65, 2800, 65),
(66, 'Manipur Forests', 66, 3400, 66),
(67, 'Meghalaya Forests', 67, 2900, 67),
(68, 'Chhattisgarh Sal Forests', 68, 5000, 68),
(69, 'Bihar Deciduous Forests', 69, 4400, 69),
(70, 'Nagaland Evergreen Forests', 70, 1500, 70),
(71, 'Karnataka Dry Deciduous', 71, 10000, 71),
(72, 'Gujarat Scrub Forests', 72, 5300, 72),
(73, 'Andhra Pradesh Forests', 73, 7000, 73),
(74, 'Himalayan Temperate Forests', 74, 6500, 74),
(75, 'Rajasthan Forests', 75, 3000, 75),
(76, 'Assam Floodplain Forests', 76, 4000, 76),
(77, 'Jharkhand Mixed Forests', 77, 7000, 77),
(78, 'West Bengal Forests', 78, 9000, 78),
(79, 'Tropical Rainforests', 79, 12000, 79),
(80, 'Madhya Pradesh Dry Forests', 80, 7500, 80);

13.Insert in Tree_Count

INSERT INTO Tree_Count (R_Treecount_ID, Tree_Count, Count_Date) VALUES
(1, 100000, '2024-01-01'),
(2, 85000, '2024-01-02'),
(3, 120000, '2024-01-03'),
(4, 90000, '2024-01-04'),
(5, 95000, '2024-01-05'),
(6, 110000, '2024-01-06'),
(7, 130000, '2024-01-07'),
(8, 105000, '2024-01-08'),
(9, 115000, '2024-01-09'),
(10, 100000, '2024-01-10'),
(11, 95000, '2024-01-11'),
(12, 105000, '2024-01-12'),
(13, 100000, '2024-01-13'),
(14, 120000, '2024-01-14'),
(15, 130000, '2024-01-15'),

(16, 135000, '2024-01-16'),
(17, 100000, '2024-01-17'),
(18, 110000, '2024-01-18'),
(19, 115000, '2024-01-19'),
(20, 120000, '2024-01-20'),
(21, 140000, '2024-01-21'),
(22, 100000, '2024-01-22'),
(23, 135000, '2024-01-23'),
(24, 100000, '2024-01-24'),
(25, 110000, '2024-01-25'),
(26, 105000, '2024-01-26'),
(27, 100000, '2024-01-27'),
(28, 115000, '2024-01-28'),
(29, 120000, '2024-01-29'),
(30, 110000, '2024-01-30'),
(31, 90000, '2024-01-31'),
(32, 130000, '2024-02-01'),
(33, 95000, '2024-02-02'),
(34, 120000, '2024-02-03'),
(35, 105000, '2024-02-04'),
(36, 110000, '2024-02-05'),
(37, 130000, '2024-02-06'),
(38, 125000, '2024-02-07'),
(39, 115000, '2024-02-08'),
(40, 140000, '2024-02-09'),
(41, 125000, '2024-02-10'),
(42, 135000, '2024-02-11'),
(43, 100000, '2024-02-12'),
(44, 130000, '2024-02-13'),
(45, 120000, '2024-02-14'),
(46, 110000, '2024-02-15'),
(47, 100000, '2024-02-16'),
(48, 105000, '2024-02-17'),
(49, 115000, '2024-02-18'),
(50, 120000, '2024-02-19'),

```
(51, 110000, '2024-02-20'),
(52, 100000, '2024-02-21'),
(53, 130000, '2024-02-22'),
(54, 110000, '2024-02-23'),
(55, 105000, '2024-02-24'),
(56, 120000, '2024-02-25'),
(57, 100000, '2024-02-26'),
(58, 130000, '2024-02-27'),
(59, 120000, '2024-02-28'),
(60, 115000, '2024-02-29'),
(61, 110000, '2024-03-01'),
(62, 105000, '2024-03-02'),
(63, 120000, '2024-03-03'),
(64, 115000, '2024-03-04'),
(65, 110000, '2024-03-05'),
(66, 100000, '2024-03-06'),
(67, 105000, '2024-03-07'),
(68, 120000, '2024-03-08'),
(69, 115000, '2024-03-09'),
(70, 130000, '2024-03-10'),
(71, 125000, '2024-03-11'),
(72, 110000, '2024-03-12'),
(73, 120000, '2024-03-13'),
(74, 130000, '2024-03-14'),
(75, 115000, '2024-03-15'),
(76, 100000, '2024-03-16'),
(77, 125000, '2024-03-17'),
(78, 130000, '2024-03-18'),
(79, 110000, '2024-03-19'),
(80, 120000, '2024-03-20');
```

14.Insert in Harvest

```
-- Harvest Insertions (80 queries, each batch will have 2 harvests)
```

```
INSERT INTO Harvest (Harvest_ID, Batch_ID, Harvest_date, Volume,
Harvested_Tree_count, Replanted_Tree_count) VALUES
(1, 1, '2024-11-01', 5000, 200, 150),
(2, 1, '2024-11-02', 5200, 180, 140),
(3, 2, '2024-11-03', 4800, 210, 160),
(4, 2, '2024-11-04', 5100, 220, 170),
(5, 3, '2024-11-05', 5300, 230, 180),
(6, 3, '2024-11-06', 4900, 250, 200),
(7, 4, '2024-11-07', 5200, 240, 190),
(8, 4, '2024-11-08', 4800, 210, 170),
(9, 5, '2024-11-09', 5000, 200, 150),
(10, 5, '2024-11-10', 5100, 220, 180),
(11, 6, '2024-11-11', 5200, 230, 190),
(12, 6, '2024-11-12', 5300, 240, 200),
(13, 7, '2024-11-13', 5000, 210, 160),
(14, 7, '2024-11-14', 5100, 220, 180),
(15, 8, '2024-11-15', 4900, 250, 200),
(16, 8, '2024-11-16', 4800, 230, 180),
(17, 9, '2024-11-17', 5300, 240, 200),
(18, 9, '2024-11-18', 5200, 210, 170),
(19, 10, '2024-11-19', 5100, 220, 180),
(20, 10, '2024-11-20', 4900, 250, 200),
(21, 11, '2024-11-21', 4800, 230, 170),
(22, 11, '2024-11-22', 5000, 200, 150),
(23, 12, '2024-11-23', 5200, 240, 190),
(24, 12, '2024-11-24', 5100, 220, 180),
(25, 13, '2024-11-25', 5300, 230, 190),
(26, 13, '2024-11-26', 4800, 210, 160),
(27, 14, '2024-11-27', 5000, 240, 200),
(28, 14, '2024-11-28', 5100, 220, 180),
(29, 15, '2024-11-29', 5200, 230, 190),
(30, 15, '2024-11-30', 5300, 240, 200),
(31, 16, '2024-12-01', 5000, 200, 150),
(32, 16, '2024-12-02', 4900, 250, 200),
(33, 17, '2024-12-03', 4800, 220, 180),
```

(34, 17, '2024-12-04', 5100, 230, 190),
(35, 18, '2024-12-05', 5200, 210, 170),
(36, 18, '2024-12-06', 5300, 240, 200),
(37, 19, '2024-12-07', 4800, 230, 180),
(38, 19, '2024-12-08', 4900, 220, 180),
(39, 20, '2024-12-09', 5000, 210, 160),
(40, 20, '2024-12-10', 5100, 240, 200),
(41, 21, '2024-12-11', 5200, 230, 190),
(42, 21, '2024-12-12', 5300, 240, 200),
(43, 22, '2024-12-13', 4800, 200, 150),
(44, 22, '2024-12-14', 4900, 210, 170),
(45, 23, '2024-12-15', 5100, 220, 180),
(46, 23, '2024-12-16', 5200, 230, 190),
(47, 24, '2024-12-17', 5300, 240, 200),
(48, 24, '2024-12-18', 4800, 200, 150),
(49, 25, '2024-12-19', 4900, 250, 200),
(50, 25, '2024-12-20', 5000, 230, 190),
(51, 26, '2024-12-21', 5100, 220, 180),
(52, 26, '2024-12-22', 5200, 240, 200),
(53, 27, '2024-12-23', 5300, 230, 190),
(54, 27, '2024-12-24', 4800, 210, 160),
(55, 28, '2024-12-25', 5000, 220, 180),
(56, 28, '2024-12-26', 5100, 230, 190),
(57, 29, '2024-12-27', 5200, 240, 200),
(58, 29, '2024-12-28', 5300, 250, 200),
(59, 30, '2024-12-29', 4800, 230, 180),
(60, 30, '2024-12-30', 4900, 200, 150),
(61, 31, '2024-12-31', 5100, 220, 180),
(62, 31, '2025-01-01', 5200, 230, 190),
(63, 32, '2025-01-02', 5300, 240, 200),
(64, 32, '2025-01-03', 4800, 210, 160),
(65, 33, '2025-01-04', 5000, 220, 180),
(66, 33, '2025-01-05', 5100, 230, 190),
(67, 34, '2025-01-06', 5200, 240, 200),
(68, 34, '2025-01-07', 5300, 250, 200),

```
(69, 35, '2025-01-08', 4800, 230, 180),
(70, 35, '2025-01-09', 4900, 200, 150),
(71, 36, '2025-01-10', 5100, 220, 180),
(72, 36, '2025-01-11', 5200, 230, 190),
(73, 37, '2025-01-12', 5300, 240, 200),
(74, 37, '2025-01-13', 4800, 210, 160),
(75, 38, '2025-01-14', 4900, 220, 180),
(76, 38, '2025-01-15', 5100, 230, 190),
(77, 39, '2025-01-16', 5200, 240, 200),
(78, 39, '2025-01-17', 5300, 250, 200),
(79, 40, '2025-01-18', 4800, 230, 180),
(80, 40, '2025-01-19', 4900, 200, 150);
```

15.Insert in Batch

```
-- Batch Insertions (40 queries)

INSERT INTO Batch (Batch_ID, Facility_ID, Total_quantity, Total_volume)
VALUES
(1, 1, 380, 10200),  -- Harvest 1 + Harvest 2
(2, 2, 390, 9600),   -- Harvest 3 + Harvest 4
(3, 3, 400, 10400),  -- Harvest 5 + Harvest 6
(4, 4, 410, 10200),  -- Harvest 7 + Harvest 8
(5, 5, 430, 10600),  -- Harvest 9 + Harvest 10
(6, 6, 470, 10600),  -- Harvest 11 + Harvest 12
(7, 7, 450, 10400),  -- Harvest 13 + Harvest 14
(8, 8, 470, 9800),   -- Harvest 15 + Harvest 16
(9, 9, 470, 10600),  -- Harvest 17 + Harvest 18
(10, 10, 450, 10200),-- Harvest 19 + Harvest 20
(11, 11, 460, 10400),-- Harvest 21 + Harvest 22
(12, 12, 470, 10400),-- Harvest 23 + Harvest 24
(13, 13, 460, 10600),-- Harvest 25 + Harvest 26
(14, 14, 460, 10100),-- Harvest 27 + Harvest 28
(15, 15, 470, 10600),-- Harvest 29 + Harvest 30
(16, 16, 450, 10400),-- Harvest 31 + Harvest 32
```

```
(17, 17, 450, 9600), -- Harvest 33 + Harvest 34
(18, 18, 460, 10200),-- Harvest 35 + Harvest 36
(19, 19, 470, 10600),-- Harvest 37 + Harvest 38
(20, 20, 450, 10000),-- Harvest 39 + Harvest 40
(21, 21, 470, 10200),-- Harvest 41 + Harvest 42
(22, 22, 450, 10400),-- Harvest 43 + Harvest 44
(23, 23, 460, 10600),-- Harvest 45 + Harvest 46
(24, 24, 470, 10200),-- Harvest 47 + Harvest 48
(25, 25, 460, 10800),-- Harvest 49 + Harvest 50
(26, 26, 480, 10600),-- Harvest 51 + Harvest 52
(27, 27, 470, 10800),-- Harvest 53 + Harvest 54
(28, 28, 470, 10400),-- Harvest 55 + Harvest 56
(29, 29, 460, 10600),-- Harvest 57 + Harvest 58
(30, 30, 470, 9600), -- Harvest 59 + Harvest 60
(31, 31, 480, 10200),-- Harvest 61 + Harvest 62
(32, 32, 490, 10600),-- Harvest 63 + Harvest 64
(33, 33, 470, 10400),-- Harvest 65 + Harvest 66
(34, 34, 480, 10600),-- Harvest 67 + Harvest 68
(35, 35, 490, 10800),-- Harvest 69 + Harvest 70
(36, 36, 460, 10200),-- Harvest 71 + Harvest 72
(37, 37, 480, 10600),-- Harvest 73 + Harvest 74
(38, 38, 490, 10200),-- Harvest 75 + Harvest 76
(39, 39, 480, 10800),-- Harvest 77 + Harvest 78
(40, 40, 460, 10200);-- Harvest 79 + Harvest 80
```

16. Insert in harvest_Logger

```
INSERT INTO Harvest_Logger (Harvest_ID, Logger_ID) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
```

(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(15, 15),
(16, 16),
(17, 17),
(18, 18),
(19, 19),
(20, 20),
(21, 21),
(22, 22),
(23, 23),
(24, 24),
(25, 25),
(26, 26),
(27, 27),
(28, 28),
(29, 29),
(30, 30),
(31, 31),
(32, 32),
(33, 33),
(34, 34),
(35, 35),
(36, 36),
(37, 37),
(38, 38),
(39, 39),
(40, 40),
(41, 41),
(42, 42),

(43, 43),
(44, 44),
(45, 45),
(46, 46),
(47, 47),
(48, 48),
(49, 49),
(50, 50),
(51, 51),
(52, 52),
(53, 53),
(54, 54),
(55, 55),
(56, 56),
(57, 57),
(58, 58),
(59, 59),
(60, 60),
(61, 61),
(62, 62),
(63, 63),
(64, 64),
(65, 65),
(66, 66),
(67, 67),
(68, 68),
(69, 69),
(70, 70),
(71, 71),
(72, 72),
(73, 73),
(74, 74),
(75, 75),
(76, 76),
(77, 77),

(78, 78),
(79, 79),
(80, 80);

17.Insert in Region_Harvest

-- Region_Harvest Insertions (80 queries)

INSERT INTO Region_Harvest (Region_ID, Harvest_ID) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(15, 15),
(16, 16),
(17, 17),
(18, 18),
(19, 19),
(20, 20),
(21, 21),
(22, 22),
(23, 23),
(24, 24),
(25, 25),
(26, 26),

(27, 27),
(28, 28),
(29, 29),
(30, 30),
(31, 31),
(32, 32),
(33, 33),
(34, 34),
(35, 35),
(36, 36),
(37, 37),
(38, 38),
(39, 39),
(40, 40),
(41, 41),
(42, 42),
(43, 43),
(44, 44),
(45, 45),
(46, 46),
(47, 47),
(48, 48),
(49, 49),
(50, 50),
(51, 51),
(52, 52),
(53, 53),
(54, 54),
(55, 55),
(56, 56),
(57, 57),
(58, 58),
(59, 59),
(60, 60),
(61, 61),

```
(62, 62),
(63, 63),
(64, 64),
(65, 65),
(66, 66),
(67, 67),
(68, 68),
(69, 69),
(70, 70),
(71, 71),
(72, 72),
(73, 73),
(74, 74),
(75, 75),
(76, 76),
(77, 77),
(78, 78),
(79, 79),
(80, 80);
```

18.Insert in Customer_Contact

```
-- 80 Insertions for Customer_Contact table
INSERT INTO Customer_Contact (Customer_ID, Contact_number)
VALUES
(1, '9876543210'),
(2, '9887654321'),
(3, '9876541230'),
(4, '9801234567'),
(5, '9776543120'),
(6, '9908765432'),
(7, '9754321098'),
(8, '9765432109'),
(9, '9632587410'),
(10, '9623456789'),
```

(11, '9508765432'),
(12, '9807564321'),
(13, '9745612380'),
(14, '9790321456'),
(15, '9658743120'),
(16, '9665327108'),
(17, '9556374821'),
(18, '9536721940'),
(19, '9543126789'),
(20, '9612345678'),
(21, '9445678123'),
(22, '9423486571'),
(23, '9567812345'),
(24, '9435678912'),
(25, '9412365789'),
(26, '9234598710'),
(27, '9223475610'),
(28, '9256781243'),
(29, '9301827465'),
(30, '9392716450'),
(31, '9381045632'),
(32, '9371598740'),
(33, '9345671230'),
(34, '9458362170'),
(35, '9412598743'),
(36, '9556789012'),
(37, '9263475180'),
(38, '9193246570'),
(39, '9203456789'),
(40, '9376541230'),
(41, '9456789301'),
(42, '9418765432'),
(43, '9554321098'),
(44, '9498765430'),
(45, '9732465871'),

(46, '9632458710'),
(47, '9593426789'),
(48, '9621547980'),
(49, '9398765321'),
(50, '9456321098'),
(51, '9283746501'),
(52, '9375462380'),
(53, '9335819024'),
(54, '9364728593'),
(55, '9263478920'),
(56, '9256123480'),
(57, '9201765432'),
(58, '9346821059'),
(59, '9325478632'),
(60, '9378512460'),
(61, '9402361847'),
(62, '9573146802'),
(63, '9507642931'),
(64, '9482634570'),
(65, '9231765489'),
(66, '9364876125'),
(67, '9395187423'),
(68, '9273915470'),
(69, '9221345687'),
(70, '9293846502'),
(71, '9453820769'),
(72, '9251493820'),
(73, '9447236198'),
(74, '9365784921'),
(75, '9398674123'),
(76, '9302465710'),
(77, '9267842091'),
(78, '9403856217'),
(79, '9216748395'),
(80, '9225637810');

19.Insert in Inventory

```
-- 80 Insertions for Inventory table
INSERT INTO Inventory (Batch_ID, Facility_ID, Total_quantity,
Available_quantity)
VALUES
(1, 1, 5000, 4500),
(2, 1, 5200, 4900),
(3, 1, 4800, 4600),
(4, 2, 5100, 4800),
(5, 2, 5300, 5100),
(6, 2, 4900, 4700),
(7, 3, 5200, 4900),
(8, 3, 4800, 4600),
(9, 4, 5000, 4700),
(10, 4, 5100, 4900),
(11, 5, 5200, 5000),
(12, 5, 5300, 5100),
(13, 6, 5000, 4700),
(14, 6, 5100, 4800),
(15, 7, 4900, 4700),
(16, 7, 4800, 4500),
(17, 8, 5300, 5100),
(18, 8, 5200, 4900),
(19, 9, 5100, 4900),
(20, 9, 4900, 4600),
(21, 10, 4800, 4500),
(22, 10, 5000, 4700),
(23, 11, 5200, 4900),
(24, 11, 5300, 5100),
(25, 12, 5100, 4800),
(26, 12, 5000, 4700),
(27, 13, 4900, 4600),
(28, 13, 4800, 4500),
```

(29, 14, 5300, 5100),
(30, 14, 5200, 4900),
(31, 15, 5000, 4700),
(32, 15, 5100, 4800),
(33, 16, 5200, 4900),
(34, 16, 5300, 5100),
(35, 17, 4800, 4500),
(36, 17, 4900, 4700),
(37, 18, 5000, 4700),
(38, 18, 5100, 4800),
(39, 19, 5300, 5100),
(40, 19, 5200, 4900),
(41, 20, 5100, 4800),
(42, 20, 5000, 4700),
(43, 21, 4900, 4600),
(44, 21, 4800, 4500),
(45, 22, 5300, 5100),
(46, 22, 5200, 4900),
(47, 23, 5000, 4700),
(48, 23, 5100, 4800),
(49, 24, 5200, 4900),
(50, 24, 5300, 5100),
(51, 25, 4800, 4500),
(52, 25, 4900, 4700),
(53, 26, 5000, 4700),
(54, 26, 5100, 4800),
(55, 27, 5200, 4900),
(56, 27, 5300, 5100),
(57, 28, 5100, 4800),
(58, 28, 5000, 4700),
(59, 29, 4900, 4600),
(60, 29, 4800, 4500),
(61, 30, 5300, 5100),
(62, 30, 5200, 4900),
(63, 31, 5000, 4700),

(64, 31, 5100, 4800),
(65, 32, 5200, 4900),
(66, 32, 5300, 5100),
(67, 33, 4800, 4500),
(68, 33, 4900, 4700),
(69, 34, 5000, 4700),
(70, 34, 5100, 4800),
(71, 35, 5300, 5100),
(72, 35, 5200, 4900),
(73, 36, 5100, 4800),
(74, 36, 5000, 4700),
(75, 37, 4900, 4600),
(76, 37, 4800, 4500),
(77, 38, 5300, 5100),
(78, 38, 5200, 4900),
(79, 39, 5000, 4700),
(80, 39, 5100, 4800);

20.Insert in Manufacture

INSERT INTO Manufacturer (Manufacturer_ID, Address_ID, Sales_ID, Name, Wood_purchased) VALUES
(1, 1, 101, 'Tata Timber Ltd.', 50000),
(2, 2, 102, 'Birla Woodworks', 45000),
(3, 3, 103, 'Ashok Plywood Industries', 60000),
(4, 4, 104, 'Reliance Timber Corp.', 52000),
(5, 5, 105, 'Godrej Wood Processing', 47000),
(6, 6, 106, 'Mahindra Timber Products', 49000),
(7, 7, 107, 'L&T Wood Manufacturing', 51000),
(8, 8, 108, 'Nayak Timber Solutions', 53000),
(9, 9, 109, 'JSW Timber Ltd.', 55000),
(10, 10, 110, 'Hindalco Wood Pvt. Ltd.', 48000),
(11, 11, 111, 'Kochi Timber Industries', 56000),
(12, 12, 112, 'Sailwood Pvt. Ltd.', 50000),
(13, 13, 113, 'Essar Timber Co.', 46000),

(14, 14, 114, 'Tata Wood Processing', 52000),
(15, 15, 115, 'Mahindra Plywood Ltd.', 49000),
(16, 16, 116, 'ICICI Timber Solutions', 53000),
(17, 17, 117, 'Ambuja Timber Ltd.', 51000),
(18, 18, 118, 'Aditya Birla Timber Pvt. Ltd.', 45000),
(19, 19, 119, 'JK Plywood Works', 47000),
(20, 20, 120, 'Vardhman Timber Products', 55000),
(21, 21, 121, 'Bharat Timber Ltd.', 60000),
(22, 22, 122, 'Rathi Plywood Co.', 53000),
(23, 23, 123, 'Punj Lloyd Timber Ltd.', 54000),
(24, 24, 124, 'Gujarat Timber Enterprises', 56000),
(25, 25, 125, 'Sundaram Timber Solutions', 58000),
(26, 26, 126, 'Zenith Timber Manufacturing', 50000),
(27, 27, 127, 'Prism Timber Works', 52000),
(28, 28, 128, 'Unitech Wood Processing', 49000),
(29, 29, 129, 'Shree Timber Industries', 51000),
(30, 30, 130, 'Gokuldas Plywood Ltd.', 45000),
(31, 31, 131, 'Bansal Wood Works', 46000),
(32, 32, 132, 'Tata Wood Industries', 53000),
(33, 33, 133, 'IndoWood Manufacturing', 54000),
(34, 34, 134, 'Pinnacle Timber Pvt. Ltd.', 47000),
(35, 35, 135, 'Lalit Timber Co.', 50000),
(36, 36, 136, 'Kailash Wood Processing', 52000),
(37, 37, 137, 'Shakti Timber Ltd.', 56000),
(38, 38, 138, 'Dalmia Timber Pvt. Ltd.', 55000),
(39, 39, 139, 'Krishna Wood Solutions', 49000),
(40, 40, 140, 'Suraj Timber Products', 48000),
(41, 41, 141, 'Vikram Timber Works', 51000),
(42, 42, 142, 'Oberoi Timber Ltd.', 46000),
(43, 43, 143, 'Kumar Wood Products', 53000),
(44, 44, 144, 'SRS Timber Industries', 52000),
(45, 45, 145, 'Pioneer Wood Works', 54000),
(46, 46, 146, 'Vishal Timber Products', 55000),
(47, 47, 147, 'Siddhi Plywood Pvt. Ltd.', 60000),
(48, 48, 148, 'Tata Wood Manufacturing', 57000),

(49, 49, 149, 'Siddharth Timber Co.', 49000),
(50, 50, 150, 'Maruti Timber Pvt. Ltd.', 51000),
(51, 51, 151, 'Arihant Timber Ltd.', 54000),
(52, 52, 152, 'Panchal Timber Solutions', 56000),
(53, 53, 153, 'Sanjeev Plywood Ltd.', 52000),
(54, 54, 154, 'Rajasthan Timber Works', 49000),
(55, 55, 155, 'Laxmi Timber Manufacturing', 53000),
(56, 56, 156, 'Mohan Timber Pvt. Ltd.', 51000),
(57, 57, 157, 'Shree Shakti Wood', 54000),
(58, 58, 158, 'Gurukripa Timber Works', 55000),
(59, 59, 159, 'Vidyut Timber Ltd.', 48000),
(60, 60, 160, 'Rupesh Timber Pvt. Ltd.', 49000),
(61, 61, 161, 'Neelam Timber Solutions', 47000),
(62, 62, 162, 'Manoj Timber Manufacturing', 50000),
(63, 63, 163, 'Bansal Plywood Pvt. Ltd.', 53000),
(64, 64, 164, 'Jain Timber Products', 55000),
(65, 65, 165, 'Rama Timber Works', 52000),
(66, 66, 166, 'Shiv Timber Pvt. Ltd.', 56000),
(67, 67, 167, 'Pinnacle Timber Ltd.', 48000),
(68, 68, 168, 'Sandeep Timber Co.', 54000),
(69, 69, 169, 'Pallavi Timber Ltd.', 49000),
(70, 70, 170, 'Akash Timber Pvt. Ltd.', 51000),
(71, 71, 171, 'Om Plywood Industries', 50000),
(72, 72, 172, 'Sagar Timber Ltd.', 52000),
(73, 73, 173, 'Abhishek Timber Solutions', 54000),
(74, 74, 174, 'Jai Plywood Manufacturing', 56000),
(75, 75, 175, 'Mahavir Timber Ltd.', 55000),
(76, 76, 176, 'Vishnu Timber Pvt. Ltd.', 47000),
(77, 77, 177, 'Shivraj Wood Processing', 51000),
(78, 78, 178, 'Shree Timber Ltd.', 49000),
(79, 79, 179, 'Mahalaxmi Timber Works', 53000),
(80, 80, 180, 'Harsh Timber Pvt. Ltd.', 56000);

21.Insert in Sales

```
INSERT INTO Sales (Sales_ID, Customer_ID, Product_ID, Purchase_date,
Quantity_purchased) VALUES
(1, 1, 1, '2024-10-01', 10),
(2, 2, 2, '2024-10-02', 15),
(3, 3, 3, '2024-10-03', 12),
(4, 4, 4, '2024-10-04', 18),
(5, 5, 5, '2024-10-05', 20),
(6, 6, 6, '2024-10-06', 25),
(7, 7, 7, '2024-10-07', 8),
(8, 8, 8, '2024-10-08', 30),
(9, 9, 9, '2024-10-09', 22),
(10, 10, 10, '2024-10-10', 27),
(11, 11, 11, '2024-10-11', 14),
(12, 12, 12, '2024-10-12', 9),
(13, 13, 13, '2024-10-13', 17),
(14, 14, 14, '2024-10-14', 23),
(15, 15, 15, '2024-10-15', 16),
(16, 16, 16, '2024-10-16', 21),
(17, 17, 17, '2024-10-17', 11),
(18, 18, 18, '2024-10-18', 19),
(19, 19, 19, '2024-10-19', 25),
(20, 20, 20, '2024-10-20', 13),
(21, 21, 21, '2024-10-21', 24),
(22, 22, 22, '2024-10-22', 10),
(23, 23, 23, '2024-10-23', 16),
(24, 24, 24, '2024-10-24', 18),
(25, 25, 25, '2024-10-25', 28),
(26, 26, 26, '2024-10-26', 30),
(27, 27, 27, '2024-10-27', 9),
(28, 28, 28, '2024-10-28', 14),
(29, 29, 29, '2024-10-29', 12),
(30, 30, 30, '2024-10-30', 19),
(31, 31, 31, '2024-11-01', 20),
(32, 32, 32, '2024-11-02', 22),
(33, 33, 33, '2024-11-03', 17),
```

(34, 34, 34, '2024-11-04', 13),
(35, 35, 35, '2024-11-05', 24),
(36, 36, 36, '2024-11-06', 30),
(37, 37, 37, '2024-11-07', 21),
(38, 38, 38, '2024-11-08', 15),
(39, 39, 39, '2024-11-09', 19),
(40, 40, 40, '2024-11-10', 10),
(41, 41, 41, '2024-11-11', 23),
(42, 42, 42, '2024-11-12', 17),
(43, 43, 43, '2024-11-13', 14),
(44, 44, 44, '2024-11-14', 26),
(45, 45, 45, '2024-11-15', 21),
(46, 46, 46, '2024-11-16', 12),
(47, 47, 47, '2024-11-17', 19),
(48, 48, 48, '2024-11-18', 16),
(49, 49, 49, '2024-11-19', 14),
(50, 50, 50, '2024-11-20', 11),
(51, 51, 51, '2024-11-21', 24),
(52, 52, 52, '2024-11-22', 13),
(53, 53, 53, '2024-11-23', 18),
(54, 54, 54, '2024-11-24', 28),
(55, 55, 55, '2024-11-25', 20),
(56, 56, 56, '2024-11-26', 15),
(57, 57, 57, '2024-11-27', 22),
(58, 58, 58, '2024-11-28', 18),
(59, 59, 59, '2024-11-29', 25),
(60, 60, 60, '2024-11-30', 30),
(61, 61, 61, '2024-12-01', 17),
(62, 62, 62, '2024-12-02', 19),
(63, 63, 63, '2024-12-03', 21),
(64, 64, 64, '2024-12-04', 24),
(65, 65, 65, '2024-12-05', 12),
(66, 66, 66, '2024-12-06', 15),
(67, 67, 67, '2024-12-07', 28),
(68, 68, 68, '2024-12-08', 23),

(69, 69, 69, '2024-12-09', 18),
(70, 70, 70, '2024-12-10', 30),
(71, 71, 71, '2024-12-11', 25),
(72, 72, 72, '2024-12-12', 22),
(73, 73, 73, '2024-12-13', 20),
(74, 74, 74, '2024-12-14', 21),
(75, 75, 75, '2024-12-15', 19),
(76, 76, 76, '2024-12-16', 24),
(77, 77, 77, '2024-12-17', 16),
(78, 78, 78, '2024-12-18', 18),
(79, 79, 79, '2024-12-19', 11),
(80, 80, 80, '2024-12-20', 27);

22.Insert in Wood_Product

INSERT INTO Wood_Product (Product_ID, Manufacturer_ID, Customer_ID, Product_Type, Wood_used_quantity, Volume, Production_date) VALUES
(1, 1, 1, 'Furniture', 50, 15, '2024-10-01'),
(2, 2, 2, 'Wooden Panel', 30, 10, '2024-10-02'),
(3, 3, 3, 'Decking', 60, 20, '2024-10-03'),
(4, 4, 4, 'Wooden Flooring', 40, 12, '2024-10-04'),
(5, 5, 5, 'Wooden Furniture', 80, 25, '2024-10-05'),
(6, 6, 6, 'Wooden Door', 45, 13, '2024-10-06'),
(7, 7, 7, 'Wooden Window', 35, 11, '2024-10-07'),
(8, 8, 8, 'Wooden Toys', 25, 8, '2024-10-08'),
(9, 9, 9, 'Wooden Planks', 55, 17, '2024-10-09'),
(10, 10, 10, 'Wooden Panels', 50, 15, '2024-10-10'),
(11, 11, 11, 'Wooden Chair', 35, 10, '2024-10-11'),
(12, 12, 12, 'Wooden Table', 70, 22, '2024-10-12'),
(13, 13, 13, 'Wooden Cabinet', 60, 18, '2024-10-13'),
(14, 14, 14, 'Wooden Shelf', 40, 12, '2024-10-14'),
(15, 15, 15, 'Wooden Railing', 50, 14, '2024-10-15'),
(16, 16, 16, 'Wooden Partition', 45, 13, '2024-10-16'),
(17, 17, 17, 'Wooden Paneling', 60, 20, '2024-10-17'),
(18, 18, 18, 'Wooden Cabinets', 65, 21, '2024-10-18'),

(19, 19, 19, 'Wooden Furniture Set', 85, 28, '2024-10-19'),
(20, 20, 20, 'Wooden Rack', 50, 15, '2024-10-20'),
(21, 21, 21, 'Wooden Flooring', 60, 18, '2024-10-21'),
(22, 22, 22, 'Wooden Doors', 45, 13, '2024-10-22'),
(23, 23, 23, 'Wooden Blocks', 30, 10, '2024-10-23'),
(24, 24, 24, 'Wooden Planks', 55, 17, '2024-10-24'),
(25, 25, 25, 'Wooden Crate', 50, 15, '2024-10-25'),
(26, 26, 26, 'Wooden Box', 45, 14, '2024-10-26'),
(27, 27, 27, 'Wooden Toy', 20, 6, '2024-10-27'),
(28, 28, 28, 'Wooden Cabinet', 60, 18, '2024-10-28'),
(29, 29, 29, 'Wooden Bench', 40, 12, '2024-10-29'),
(30, 30, 30, 'Wooden Table', 70, 22, '2024-10-30'),
(31, 31, 31, 'Wooden Planks', 50, 16, '2024-11-01'),
(32, 32, 32, 'Wooden Railing', 45, 14, '2024-11-02'),
(33, 33, 33, 'Wooden Paneling', 55, 17, '2024-11-03'),
(34, 34, 34, 'Wooden Door', 30, 10, '2024-11-04'),
(35, 35, 35, 'Wooden Shelf', 40, 12, '2024-11-05'),
(36, 36, 36, 'Wooden Partition', 60, 18, '2024-11-06'),
(37, 37, 37, 'Wooden Toys', 25, 8, '2024-11-07'),
(38, 38, 38, 'Wooden Cabinet', 55, 16, '2024-11-08'),
(39, 39, 39, 'Wooden Chair', 35, 10, '2024-11-09'),
(40, 40, 40, 'Wooden Table', 70, 22, '2024-11-10'),
(41, 41, 41, 'Wooden Crate', 50, 15, '2024-11-11'),
(42, 42, 42, 'Wooden Box', 45, 14, '2024-11-12'),
(43, 43, 43, 'Wooden Flooring', 60, 18, '2024-11-13'),
(44, 44, 44, 'Wooden Paneling', 50, 15, '2024-11-14'),
(45, 45, 45, 'Wooden Railing', 60, 18, '2024-11-15'),
(46, 46, 46, 'Wooden Bench', 40, 12, '2024-11-16'),
(47, 47, 47, 'Wooden Chair', 30, 10, '2024-11-17'),
(48, 48, 48, 'Wooden Panel', 35, 11, '2024-11-18'),
(49, 49, 49, 'Wooden Shelf', 55, 17, '2024-11-19'),
(50, 50, 50, 'Wooden Table', 60, 18, '2024-11-20'),
(51, 51, 51, 'Wooden Cabinet', 65, 21, '2024-11-21'),
(52, 52, 52, 'Wooden Toy', 25, 8, '2024-11-22'),
(53, 53, 53, 'Wooden Chair', 45, 14, '2024-11-23'),

(54, 54, 54, 'Wooden Bench', 40, 12, '2024-11-24'),
(55, 55, 55, 'Wooden Partition', 50, 15, '2024-11-25'),
(56, 56, 56, 'Wooden Rack', 35, 10, '2024-11-26'),
(57, 57, 57, 'Wooden Panel', 60, 18, '2024-11-27'),
(58, 58, 58, 'Wooden Door', 50, 15, '2024-11-28'),
(59, 59, 59, 'Wooden Shelf', 55, 16, '2024-11-29'),
(60, 60, 60, 'Wooden Cabinet', 70, 22, '2024-12-01'),
(61, 61, 61, 'Wooden Chair', 35, 11, '2024-12-02'),
(62, 62, 62, 'Wooden Table', 70, 20, '2024-12-03'),
(63, 63, 63, 'Wooden Railing', 45, 14, '2024-12-04'),
(64, 64, 64, 'Wooden Box', 50, 15, '2024-12-05'),
(65, 65, 65, 'Wooden Crate', 45, 13, '2024-12-06'),
(66, 66, 66, 'Wooden Paneling', 55, 17, '2024-12-07'),
(67, 67, 67, 'Wooden Door', 30, 10, '2024-12-08'),
(68, 68, 68, 'Wooden Cabinet', 60, 18, '2024-12-09'),
(69, 69, 69, 'Wooden Bench', 35, 12, '2024-12-10'),
(70, 70, 70, 'Wooden Shelf', 50, 15, '2024-12-11'),
(71, 71, 71, 'Wooden Toy', 20, 6, '2024-12-12'),
(72, 72, 72, 'Wooden Chair', 40, 12, '2024-12-13'),
(73, 73, 73, 'Wooden Panel', 55, 17, '2024-12-14'),
(74, 74, 74, 'Wooden Crate', 50, 15, '2024-12-15'),
(75, 75, 75, 'Wooden Box', 45, 14, '2024-12-16'),
(76, 76, 76, 'Wooden Flooring', 60, 18, '2024-12-17'),
(77, 77, 77, 'Wooden Shelf', 50, 15, '2024-12-18'),
(78, 78, 78, 'Wooden Partition', 45, 14, '2024-12-19'),
(79, 79, 79, 'Wooden Paneling', 60, 18, '2024-12-20'),
(80, 80, 80, 'Wooden Bench', 40, 12, '2024-12-21');

23.Insert in Manufacture_Batch

-- Insert data into Manufacturer_Batch table
INSERT INTO Manufacturer_Batch (Manufacturer_ID, Batch_ID,
Quantity_owned) VALUES
(1, 1, 1000),
(2, 2, 2000),

(3, 3, 1500),
(4, 4, 1200),
(5, 5, 1800),
(6, 6, 1100),
(7, 7, 1300),
(8, 8, 1700),
(9, 9, 1600),
(10, 10, 1400),
(11, 11, 1500),
(12, 12, 1900),
(13, 13, 1000),
(14, 14, 1800),
(15, 15, 1600),
(16, 16, 1400),
(17, 17, 1300),
(18, 18, 1200),
(19, 19, 1100),
(20, 20, 1800),
(21, 21, 1600),
(22, 22, 1500),
(23, 23, 1700),
(24, 24, 1400),
(25, 25, 1300),
(26, 26, 1200),
(27, 27, 1100),
(28, 28, 1800),
(29, 29, 1500),
(30, 30, 1600),
(31, 31, 1400),
(32, 32, 1700),
(33, 33, 1200),
(34, 34, 1300),
(35, 35, 1800),
(36, 36, 1600),
(37, 37, 1500),

```
(38, 38, 1100),
(39, 39, 1400),
(40, 40, 1200);
```

24.Insert in Manufacture_Phone

```
-- Insert data into Manufacturer_Phone table
INSERT INTO Manufacturer_Phone (Manufacturer_ID, Contact_number)
VALUES
(1, '9876543210'),
(2, '9876543220'),
(3, '9876543230'),
(4, '9876543240'),
(5, '9876543250'),
(6, '9876543260'),
(7, '9876543270'),
(8, '9876543280'),
(9, '9876543290'),
(10, '9876543300'),
(11, '9876543310'),
(12, '9876543320'),
(13, '9876543330'),
(14, '9876543340'),
(15, '9876543350'),
(16, '9876543360'),
(17, '9876543370'),
(18, '9876543380'),
(19, '9876543390'),
(20, '9876543400'),
(21, '9876543410'),
(22, '9876543420'),
(23, '9876543430'),
(24, '9876543440'),
(25, '9876543450'),
(26, '9876543460'),
```

(27, '9876543470'),
(28, '9876543480'),
(29, '9876543490'),
(30, '9876543500'),
(31, '9876543510'),
(32, '9876543520'),
(33, '9876543530'),
(34, '9876543540'),
(35, '9876543550'),
(36, '9876543560'),
(37, '9876543570'),
(38, '9876543580'),
(39, '9876543590'),
(40, '9876543600'),
(41, '9876543610'),
(42, '9876543620'),
(43, '9876543630'),
(44, '9876543640'),
(45, '9876543650'),
(46, '9876543660'),
(47, '9876543670'),
(48, '9876543680'),
(49, '9876543690'),
(50, '9876543700'),
(51, '9876543710'),
(52, '9876543720'),
(53, '9876543730'),
(54, '9876543740'),
(55, '9876543750'),
(56, '9876543760'),
(57, '9876543770'),
(58, '9876543780'),
(59, '9876543790'),
(60, '9876543800'),
(61, '9876543810'),

(62, '9876543820'),
(63, '9876543830'),
(64, '9876543840'),
(65, '9876543850'),
(66, '9876543860'),
(67, '9876543870'),
(68, '9876543880'),
(69, '9876543890'),
(70, '9876543900'),
(71, '9876543910'),
(72, '9876543920'),
(73, '9876543930'),
(74, '9876543940'),
(75, '9876543950'),
(76, '9876543960'),
(77, '9876543970'),
(78, '9876543980'),
(79, '9876543990'),
(80, '9876544000');

25.Insert in Product_Inventory

-- Insert data into Product_Inventory table
INSERT INTO Product_Inventory (Product_ID, Manufacturer_ID, Total_quantity, Available_quantity) VALUES
(1, 1, 1000, 800),
(2, 2, 1200, 1000),
(3, 3, 1100, 900),
(4, 4, 1050, 850),
(5, 5, 1150, 950),
(6, 6, 1300, 1100),
(7, 7, 1250, 1050),
(8, 8, 1400, 1200),
(9, 9, 1350, 1150),
(10, 10, 1250, 1050),

(11, 11, 1100, 900),
(12, 12, 1050, 850),
(13, 13, 1200, 1000),
(14, 14, 1300, 1100),
(15, 15, 1250, 1050),
(16, 16, 1350, 1150),
(17, 17, 1400, 1200),
(18, 18, 1450, 1250),
(19, 19, 1300, 1100),
(20, 20, 1200, 1000),
(21, 21, 1350, 1150),
(22, 22, 1250, 1050),
(23, 23, 1400, 1200),
(24, 24, 1300, 1100),
(25, 25, 1200, 1000),
(26, 26, 1250, 1050),
(27, 27, 1100, 900),
(28, 28, 1050, 850),
(29, 29, 1200, 1000),
(30, 30, 1350, 1150),
(31, 31, 1300, 1100),
(32, 32, 1400, 1200),
(33, 33, 1250, 1050),
(34, 34, 1450, 1250),
(35, 35, 1300, 1100),
(36, 36, 1200, 1000),
(37, 37, 1100, 900),
(38, 38, 1050, 850),
(39, 39, 1250, 1050),
(40, 40, 1350, 1150),
(41, 41, 1300, 1100),
(42, 42, 1200, 1000),
(43, 43, 1100, 900),
(44, 44, 1250, 1050),
(45, 45, 1300, 1100),

(46, 46, 1350, 1150),
(47, 47, 1450, 1250),
(48, 48, 1400, 1200),
(49, 49, 1300, 1100),
(50, 50, 1200, 1000),
(51, 51, 1250, 1050),
(52, 52, 1100, 900),
(53, 53, 1050, 850),
(54, 54, 1350, 1150),
(55, 55, 1400, 1200),
(56, 56, 1250, 1050),
(57, 57, 1300, 1100),
(58, 58, 1200, 1000),
(59, 59, 1250, 1050),
(60, 60, 1350, 1150),
(61, 61, 1300, 1100),
(62, 62, 1250, 1050),
(63, 63, 1400, 1200),
(64, 64, 1200, 1000),
(65, 65, 1100, 900),
(66, 66, 1050, 850),
(67, 67, 1300, 1100),
(68, 68, 1350, 1150),
(69, 69, 1450, 1250),
(70, 70, 1400, 1200),
(71, 71, 1200, 1000),
(72, 72, 1250, 1050),
(73, 73, 1300, 1100),
(74, 74, 1350, 1150),
(75, 75, 1450, 1250),
(76, 76, 1400, 1200),
(77, 77, 1250, 1050),
(78, 78, 1300, 1100),
(79, 79, 1200, 1000),
(80, 80, 1100, 900);

26.Insert in Wood_Processing

```
-- Insert data into Wood_Processing table
INSERT INTO Wood_Processing (Batch_ID, Facility_ID, Processing_stage_ID,
Status, Quantity, Processing_Start_date, Processing_end_date) VALUES
(1, 1, 1, 'In Progress', 50, '2024-01-01', '2024-01-03'),
(2, 2, 2, 'Completed', 70, '2024-01-02', '2024-01-04'),
(3, 3, 3, 'In Progress', 30, '2024-01-03', '2024-01-05'),
(4, 4, 1, 'Completed', 40, '2024-01-04', '2024-01-06'),
(5, 5, 2, 'In Progress', 60, '2024-01-05', '2024-01-07'),
(6, 6, 3, 'In Progress', 20, '2024-01-06', '2024-01-08'),
(7, 7, 1, 'Completed', 100, '2024-01-07', '2024-01-09'),
(8, 8, 2, 'In Progress', 90, '2024-01-08', '2024-01-10'),
(9, 9, 3, 'Completed', 50, '2024-01-09', '2024-01-11'),
(10, 10, 1, 'In Progress', 80, '2024-01-10', '2024-01-12'),
(11, 11, 2, 'Completed', 60, '2024-01-11', '2024-01-13'),
(12, 12, 3, 'In Progress', 40, '2024-01-12', '2024-01-14'),
(13, 13, 1, 'Completed', 70, '2024-01-13', '2024-01-15'),
(14, 14, 2, 'In Progress', 90, '2024-01-14', '2024-01-16'),
(15, 15, 3, 'Completed', 30, '2024-01-15', '2024-01-17'),
(16, 16, 1, 'In Progress', 50, '2024-01-16', '2024-01-18'),
(17, 17, 2, 'Completed', 60, '2024-01-17', '2024-01-19'),
(18, 18, 3, 'In Progress', 80, '2024-01-18', '2024-01-20'),
(19, 19, 1, 'Completed', 55, '2024-01-19', '2024-01-21'),
(20, 20, 2, 'In Progress', 75, '2024-01-20', '2024-01-22'),
(21, 21, 3, 'Completed', 40, '2024-01-21', '2024-01-23'),
(22, 22, 1, 'In Progress', 95, '2024-01-22', '2024-01-24'),
(23, 23, 2, 'Completed', 80, '2024-01-23', '2024-01-25'),
(24, 24, 3, 'In Progress', 60, '2024-01-24', '2024-01-26'),
(25, 25, 1, 'Completed', 70, '2024-01-25', '2024-01-27'),
(26, 26, 2, 'In Progress', 100, '2024-01-26', '2024-01-28'),
(27, 27, 3, 'Completed', 90, '2024-01-27', '2024-01-29'),
(28, 28, 1, 'In Progress', 80, '2024-01-28', '2024-01-30'),
(29, 29, 2, 'Completed', 60, '2024-01-29', '2024-01-31'),
```

```
(30, 30, 3, 'In Progress', 40, '2024-01-30', '2024-02-01'),
(31, 31, 1, 'Completed', 50, '2024-02-01', '2024-02-03'),
(32, 32, 2, 'In Progress', 60, '2024-02-02', '2024-02-04'),
(33, 33, 3, 'Completed', 70, '2024-02-03', '2024-02-05'),
(34, 34, 1, 'In Progress', 90, '2024-02-04', '2024-02-06'),
(35, 35, 2, 'Completed', 50, '2024-02-05', '2024-02-07'),
(36, 36, 3, 'In Progress', 60, '2024-02-06', '2024-02-08'),
(37, 37, 1, 'Completed', 100, '2024-02-07', '2024-02-09'),
(38, 38, 2, 'In Progress', 80, '2024-02-08', '2024-02-10'),
(39, 39, 3, 'Completed', 70, '2024-02-09', '2024-02-11'),
(40, 40, 1, 'In Progress', 60, '2024-02-10', '2024-02-12'),
(1, 1, 2, 'Completed', 75, '2024-02-11', '2024-02-13'),
(2, 2, 3, 'In Progress', 85, '2024-02-12', '2024-02-14'),
(3, 3, 1, 'Completed', 55, '2024-02-13', '2024-02-15'),
(4, 4, 2, 'In Progress', 60, '2024-02-14', '2024-02-16'),
(5, 5, 3, 'Completed', 100, '2024-02-15', '2024-02-17'),
(6, 6, 1, 'In Progress', 45, '2024-02-16', '2024-02-18'),
(7, 7, 2, 'Completed', 80, '2024-02-17', '2024-02-19'),
(8, 8, 3, 'In Progress', 90, '2024-02-18', '2024-02-20'),
(9, 9, 1, 'Completed', 70, '2024-02-19', '2024-02-21'),
(10, 10, 2, 'In Progress', 60, '2024-02-20', '2024-02-22'),
(11, 11, 3, 'Completed', 80, '2024-02-21', '2024-02-23'),
(12, 12, 1, 'In Progress', 90, '2024-02-22', '2024-02-24'),
(13, 13, 2, 'Completed', 50, '2024-02-23', '2024-02-25'),
(14, 14, 3, 'In Progress', 70, '2024-02-24', '2024-02-26'),
(15, 15, 1, 'Completed', 95, '2024-02-25', '2024-02-27'),
(16, 16, 2, 'In Progress', 85, '2024-02-26', '2024-02-28'),
(17, 17, 3, 'Completed', 65, '2024-02-27', '2024-02-29'),
(18, 18, 1, 'In Progress', 55, '2024-02-28', '2024-03-01'),
(19, 19, 2, 'Completed', 75, '2024-03-01', '2024-03-03'),
(20, 20, 3, 'In Progress', 85, '2024-03-02', '2024-03-04'),
(21, 21, 1, 'Completed', 100, '2024-03-03', '2024-03-05'),
(22, 22, 2, 'In Progress', 65, '2024-03-04', '2024-03-06'),
(23, 23, 3, 'Completed', 90, '2024-03-05', '2024-03-07'),
(24, 24, 1, 'In Progress', 80, '2024-03-06', '2024-03-08');
```

27.Insert in Transaction

```
-- Insert data into Transaction table
INSERT INTO Transaction (Transaction_ID, Batch_ID, Facility_ID,
Manufacturer_ID, Quantity_purchased, Purchase_date) VALUES
(1, 1, 1, 1, 500, '2024-01-01'),
(2, 2, 2, 2, 600, '2024-01-02'),
(3, 3, 3, 3, 700, '2024-01-03'),
(4, 4, 4, 4, 800, '2024-01-04'),
(5, 5, 5, 5, 550, '2024-01-05'),
(6, 6, 6, 6, 450, '2024-01-06'),
(7, 7, 7, 7, 500, '2024-01-07'),
(8, 8, 8, 8, 650, '2024-01-08'),
(9, 9, 9, 9, 750, '2024-01-09'),
(10, 10, 10, 10, 900, '2024-01-10'),
(11, 11, 11, 11, 950, '2024-01-11'),
(12, 12, 12, 12, 800, '2024-01-12'),
(13, 13, 13, 13, 700, '2024-01-13'),
(14, 14, 14, 14, 600, '2024-01-14'),
(15, 15, 15, 15, 850, '2024-01-15'),
(16, 16, 16, 16, 550, '2024-01-16'),
(17, 17, 17, 17, 900, '2024-01-17'),
(18, 18, 18, 18, 1000, '2024-01-18'),
(19, 19, 19, 19, 1100, '2024-01-19'),
(20, 20, 20, 20, 1200, '2024-01-20'),
(21, 21, 21, 21, 950, '2024-01-21'),
(22, 22, 22, 22, 1050, '2024-01-22'),
(23, 23, 23, 23, 850, '2024-01-23'),
(24, 24, 24, 24, 900, '2024-01-24'),
(25, 25, 25, 25, 800, '2024-01-25'),
(26, 26, 26, 26, 700, '2024-01-26'),
(27, 27, 27, 27, 650, '2024-01-27'),
(28, 28, 28, 28, 600, '2024-01-28'),
```

(29, 29, 29, 29, 550, '2024-01-29'),
(30, 30, 30, 30, 800, '2024-01-30'),
(31, 31, 31, 31, 750, '2024-01-31'),
(32, 32, 32, 32, 950, '2024-02-01'),
(33, 33, 33, 33, 1050, '2024-02-02'),
(34, 34, 34, 34, 1150, '2024-02-03'),
(35, 35, 35, 35, 1200, '2024-02-04'),
(36, 36, 36, 36, 1000, '2024-02-05'),
(37, 37, 37, 37, 1100, '2024-02-06'),
(38, 38, 38, 38, 1200, '2024-02-07'),
(39, 39, 39, 39, 1300, '2024-02-08'),
(40, 40, 40, 40, 1400, '2024-02-09'),
(41, 1, 1, 1, 600, '2024-02-10'),
(42, 2, 2, 2, 700, '2024-02-11'),
(43, 3, 3, 3, 750, '2024-02-12'),
(44, 4, 4, 4, 850, '2024-02-13'),
(45, 5, 5, 5, 950, '2024-02-14'),
(46, 6, 6, 6, 1050, '2024-02-15'),
(47, 7, 7, 7, 1150, '2024-02-16'),
(48, 8, 8, 8, 1250, '2024-02-17'),
(49, 9, 9, 9, 1350, '2024-02-18'),
(50, 10, 10, 10, 1450, '2024-02-19'),
(51, 11, 11, 11, 950, '2024-02-20'),
(52, 12, 12, 12, 850, '2024-02-21'),
(53, 13, 13, 13, 1050, '2024-02-22'),
(54, 14, 14, 14, 1150, '2024-02-23'),
(55, 15, 15, 15, 1250, '2024-02-24'),
(56, 16, 16, 16, 1300, '2024-02-25'),
(57, 17, 17, 17, 1200, '2024-02-26'),
(58, 18, 18, 18, 1300, '2024-02-27'),
(59, 19, 19, 19, 1400, '2024-02-28'),
(60, 20, 20, 20, 1500, '2024-03-01'),
(61, 21, 21, 21, 950, '2024-03-02'),
(62, 22, 22, 22, 1050, '2024-03-03'),
(63, 23, 23, 23, 1150, '2024-03-04'),

(64, 24, 24, 24, 1250, '2024-03-05'),
(65, 25, 25, 25, 1350, '2024-03-06'),
(66, 26, 26, 26, 1450, '2024-03-07'),
(67, 27, 27, 27, 1550, '2024-03-08'),
(68, 28, 28, 28, 1650, '2024-03-09'),
(69, 29, 29, 29, 1750, '2024-03-10'),
(70, 30, 30, 30, 1850, '2024-03-11'),
(71, 31, 31, 31, 1950, '2024-03-12'),
(72, 32, 32, 32, 2050, '2024-03-13'),
(73, 33, 33, 33, 2150, '2024-03-14'),
(74, 34, 34, 34, 2250, '2024-03-15'),
(75, 35, 35, 35, 2350, '2024-03-16'),
(76, 36, 36, 36, 2450, '2024-03-17'),
(77, 37, 37, 37, 2550, '2024-03-18'),
(78, 38, 38, 38, 2650, '2024-03-19'),
(79, 39, 39, 39, 2750, '2024-03-20'),
(80, 40, 40, 40, 2850, '2024-03-21');

28.Insert in Recycle

-- Insert data into Recycle table
INSERT INTO Recycle (Recycle_ID, Product_ID, Recycle_Method_ID, Recycle_date) VALUES
(1, 1, 1, '2024-01-01'),
(2, 2, 2, '2024-01-02'),
(3, 3, 3, '2024-01-03'),
(4, 4, 4, '2024-01-04'),
(5, 5, 1, '2024-01-05'),
(6, 6, 2, '2024-01-06'),
(7, 7, 3, '2024-01-07'),
(8, 8, 4, '2024-01-08'),
(9, 9, 1, '2024-01-09'),
(10, 10, 2, '2024-01-10'),
(11, 11, 3, '2024-01-11'),
(12, 12, 4, '2024-01-12'),

(13, 13, 1, '2024-01-13'),
(14, 14, 2, '2024-01-14'),
(15, 15, 3, '2024-01-15'),
(16, 16, 4, '2024-01-16'),
(17, 17, 1, '2024-01-17'),
(18, 18, 2, '2024-01-18'),
(19, 19, 3, '2024-01-19'),
(20, 20, 4, '2024-01-20'),
(21, 21, 1, '2024-01-21'),
(22, 22, 2, '2024-01-22'),
(23, 23, 3, '2024-01-23'),
(24, 24, 4, '2024-01-24'),
(25, 25, 1, '2024-01-25'),
(26, 26, 2, '2024-01-26'),
(27, 27, 3, '2024-01-27'),
(28, 28, 4, '2024-01-28'),
(29, 29, 1, '2024-01-29'),
(30, 30, 2, '2024-01-30'),
(31, 31, 3, '2024-01-31'),
(32, 32, 4, '2024-02-01'),
(33, 33, 1, '2024-02-02'),
(34, 34, 2, '2024-02-03'),
(35, 35, 3, '2024-02-04'),
(36, 36, 4, '2024-02-05'),
(37, 37, 1, '2024-02-06'),
(38, 38, 2, '2024-02-07'),
(39, 39, 3, '2024-02-08'),
(40, 40, 4, '2024-02-09'),
(41, 1, 1, '2024-02-10'),
(42, 2, 2, '2024-02-11'),
(43, 3, 3, '2024-02-12'),
(44, 4, 4, '2024-02-13'),
(45, 5, 1, '2024-02-14'),
(46, 6, 2, '2024-02-15'),
(47, 7, 3, '2024-02-16'),

```
(48, 8, 4, '2024-02-17'),
(49, 9, 1, '2024-02-18'),
(50, 10, 2, '2024-02-19'),
(51, 11, 3, '2024-02-20'),
(52, 12, 4, '2024-02-21'),
(53, 13, 1, '2024-02-22'),
(54, 14, 2, '2024-02-23'),
(55, 15, 3, '2024-02-24'),
(56, 16, 4, '2024-02-25'),
(57, 17, 1, '2024-02-26'),
(58, 18, 2, '2024-02-27'),
(59, 19, 3, '2024-02-28'),
(60, 20, 4, '2024-02-29'),
(61, 21, 1, '2024-03-01'),
(62, 22, 2, '2024-03-02'),
(63, 23, 3, '2024-03-03'),
(64, 24, 4, '2024-03-04'),
(65, 25, 1, '2024-03-05'),
(66, 26, 2, '2024-03-06'),
(67, 27, 3, '2024-03-07'),
(68, 28, 4, '2024-03-08'),
(69, 29, 1, '2024-03-09'),
(70, 30, 2, '2024-03-10'),
(71, 31, 3, '2024-03-11'),
(72, 32, 4, '2024-03-12'),
(73, 33, 1, '2024-03-13'),
(74, 34, 2, '2024-03-14'),
(75, 35, 3, '2024-03-15'),
(76, 36, 4, '2024-03-16'),
(77, 37, 1, '2024-03-17'),
(78, 38, 2, '2024-03-18'),
(79, 39, 3, '2024-03-19'),
(80, 40, 4, '2024-03-20');
```

29.Insert in Wood_Inspection

```
-- Insert data into Wood_Inspection table
INSERT INTO Wood_Inspection (Product_ID, Condition, Decision) VALUES
(1, 'Good', 'Approved'),
(2, 'Good', 'Approved'),
(3, 'Damaged', 'Rejected'),
(4, 'Good', 'Approved'),
(5, 'Good', 'Approved'),
(6, 'Fair', 'Approved'),
(7, 'Damaged', 'Rejected'),
(8, 'Good', 'Approved'),
(9, 'Good', 'Approved'),
(10, 'Fair', 'Approved'),
(11, 'Good', 'Approved'),
(12, 'Damaged', 'Rejected'),
(13, 'Good', 'Approved'),
(14, 'Good', 'Approved'),
(15, 'Fair', 'Approved'),
(16, 'Damaged', 'Rejected'),
(17, 'Good', 'Approved'),
(18, 'Fair', 'Approved'),
(19, 'Good', 'Approved'),
(20, 'Damaged', 'Rejected'),
(21, 'Good', 'Approved'),
(22, 'Fair', 'Approved'),
(23, 'Good', 'Approved'),
(24, 'Damaged', 'Rejected'),
(25, 'Good', 'Approved'),
(26, 'Fair', 'Approved'),
(27, 'Damaged', 'Rejected'),
(28, 'Good', 'Approved'),
(29, 'Good', 'Approved'),
(30, 'Fair', 'Approved'),
(31, 'Damaged', 'Rejected'),
(32, 'Good', 'Approved'),
```

(33, 'Good', 'Approved'),
(34, 'Fair', 'Approved'),
(35, 'Damaged', 'Rejected'),
(36, 'Good', 'Approved'),
(37, 'Good', 'Approved'),
(38, 'Fair', 'Approved'),
(39, 'Damaged', 'Rejected'),
(40, 'Good', 'Approved'),
(41, 'Good', 'Approved'),
(42, 'Fair', 'Approved'),
(43, 'Damaged', 'Rejected'),
(44, 'Good', 'Approved'),
(45, 'Fair', 'Approved'),
(46, 'Good', 'Approved'),
(47, 'Damaged', 'Rejected'),
(48, 'Good', 'Approved'),
(49, 'Fair', 'Approved'),
(50, 'Good', 'Approved'),
(51, 'Damaged', 'Rejected'),
(52, 'Good', 'Approved'),
(53, 'Fair', 'Approved'),
(54, 'Good', 'Approved'),
(55, 'Damaged', 'Rejected'),
(56, 'Good', 'Approved'),
(57, 'Good', 'Approved'),
(58, 'Fair', 'Approved'),
(59, 'Damaged', 'Rejected'),
(60, 'Good', 'Approved'),
(61, 'Good', 'Approved'),
(62, 'Fair', 'Approved'),
(63, 'Damaged', 'Rejected'),
(64, 'Good', 'Approved'),
(65, 'Good', 'Approved'),
(66, 'Fair', 'Approved'),
(67, 'Damaged', 'Rejected'),

(68, 'Good', 'Approved'),
(69, 'Good', 'Approved'),
(70, 'Fair', 'Approved'),
(71, 'Damaged', 'Rejected'),
(72, 'Good', 'Approved'),
(73, 'Good', 'Approved'),
(74, 'Fair', 'Approved'),
(75, 'Damaged', 'Rejected'),
(76, 'Good', 'Approved'),
(77, 'Good', 'Approved'),
(78, 'Fair', 'Approved'),
(79, 'Damaged', 'Rejected'),
(80, 'Good', 'Approved');

30.Insert in Waste_Disposal

-- Insert data into Waste_Disposal table with unique disposal_id
INSERT INTO Waste_Disposal (Disposal_ID, Product_ID, Disposal_Method_ID, Disposal_Date) VALUES
(1, 1, 1, '2024-11-01'),
(2, 2, 2, '2024-11-02'),
(3, 3, 1, '2024-11-03'),
(4, 4, 2, '2024-11-04'),
(5, 5, 1, '2024-11-05'),
(6, 6, 3, '2024-11-06'),
(7, 7, 2, '2024-11-07'),
(8, 8, 1, '2024-11-08'),
(9, 9, 3, '2024-11-09'),
(10, 10, 2, '2024-11-10'),
(11, 11, 1, '2024-11-11'),
(12, 12, 2, '2024-11-12'),
(13, 13, 3, '2024-11-13'),
(14, 14, 1, '2024-11-14'),
(15, 15, 2, '2024-11-15'),
(16, 16, 3, '2024-11-16'),

(17, 17, 1, '2024-11-17'),
(18, 18, 2, '2024-11-18'),
(19, 19, 1, '2024-11-19'),
(20, 20, 3, '2024-11-20'),
(21, 21, 2, '2024-11-21'),
(22, 22, 1, '2024-11-22'),
(23, 23, 3, '2024-11-23'),
(24, 24, 2, '2024-11-24'),
(25, 25, 1, '2024-11-25'),
(26, 26, 3, '2024-11-26'),
(27, 27, 2, '2024-11-27'),
(28, 28, 1, '2024-11-28'),
(29, 29, 3, '2024-11-29'),
(30, 30, 2, '2024-11-30'),
(31, 31, 1, '2024-12-01'),
(32, 32, 2, '2024-12-02'),
(33, 33, 3, '2024-12-03'),
(34, 34, 1, '2024-12-04'),
(35, 35, 2, '2024-12-05'),
(36, 36, 3, '2024-12-06'),
(37, 37, 1, '2024-12-07'),
(38, 38, 2, '2024-12-08'),
(39, 39, 1, '2024-12-09'),
(40, 40, 3, '2024-12-10'),
(41, 41, 2, '2024-12-11'),
(42, 42, 1, '2024-12-12'),
(43, 43, 3, '2024-12-13'),
(44, 44, 2, '2024-12-14'),
(45, 45, 1, '2024-12-15'),
(46, 46, 3, '2024-12-16'),
(47, 47, 2, '2024-12-17'),
(48, 48, 1, '2024-12-18'),
(49, 49, 3, '2024-12-19'),
(50, 50, 2, '2024-12-20'),
(51, 51, 1, '2024-12-21'),

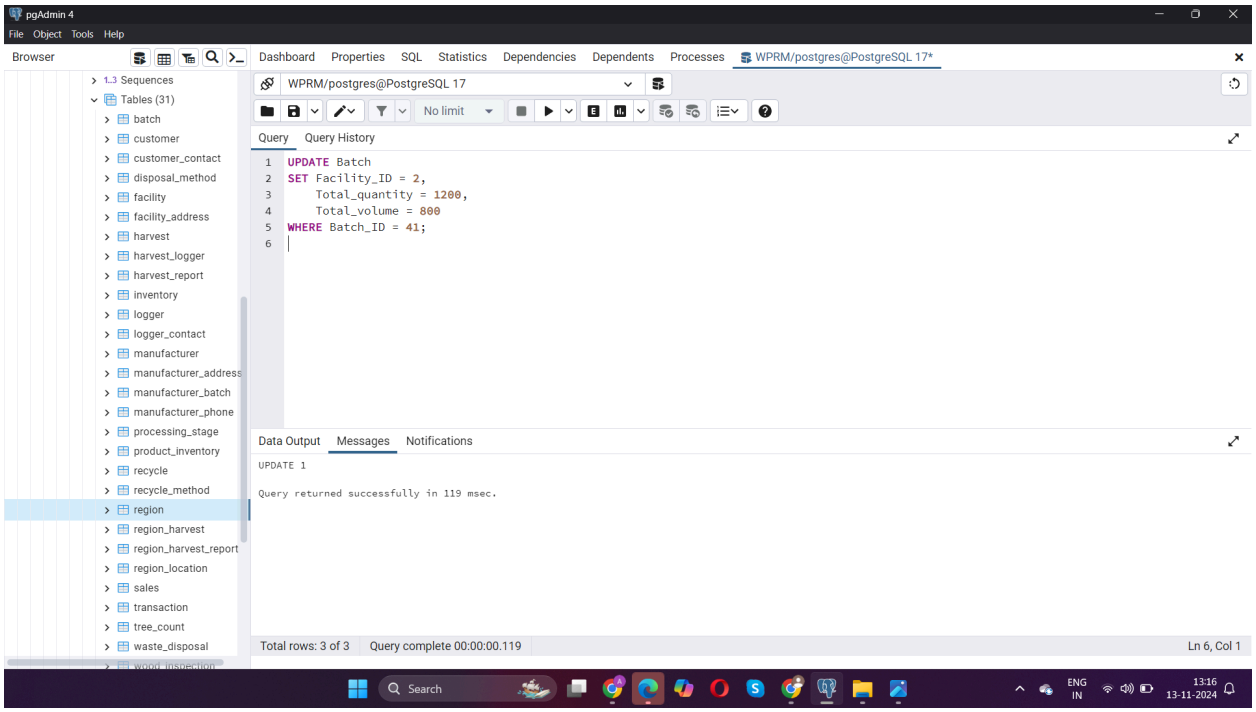
(52, 52, 3, '2024-12-22'),
(53, 53, 2, '2024-12-23'),
(54, 54, 1, '2024-12-24'),
(55, 55, 3, '2024-12-25'),
(56, 56, 2, '2024-12-26'),
(57, 57, 1, '2024-12-27'),
(58, 58, 3, '2024-12-28'),
(59, 59, 2, '2024-12-29'),
(60, 60, 1, '2024-12-30'),
(61, 61, 3, '2025-01-01'),
(62, 62, 2, '2025-01-02'),
(63, 63, 1, '2025-01-03'),
(64, 64, 3, '2025-01-04'),
(65, 65, 2, '2025-01-05'),
(66, 66, 1, '2025-01-06'),
(67, 67, 3, '2025-01-07'),
(68, 68, 2, '2025-01-08'),
(69, 69, 1, '2025-01-09'),
(70, 70, 3, '2025-01-10'),
(71, 71, 2, '2025-01-11'),
(72, 72, 1, '2025-01-12'),
(73, 73, 3, '2025-01-13'),
(74, 74, 2, '2025-01-14'),
(75, 75, 1, '2025-01-15'),
(76, 76, 3, '2025-01-16'),
(77, 77, 2, '2025-01-17'),
(78, 78, 1, '2025-01-18'),
(79, 79, 3, '2025-01-19'),
(80, 80, 2, '2025-01-20');

Query List

Basic queries

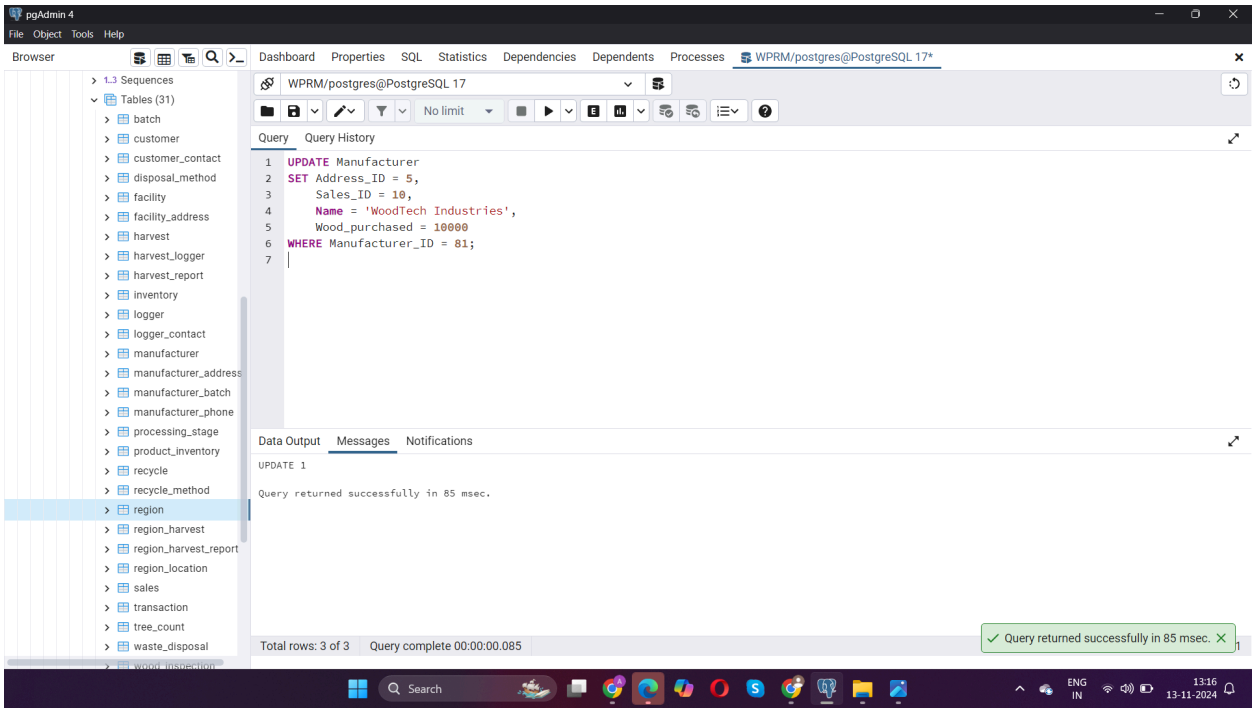
1. Update an existing batch in the Batch table:

```
UPDATE Batch SET Facility_ID = 2,
Total_quantity = 1200,
Total_volume = 800
WHERE Batch_ID = 41;
```



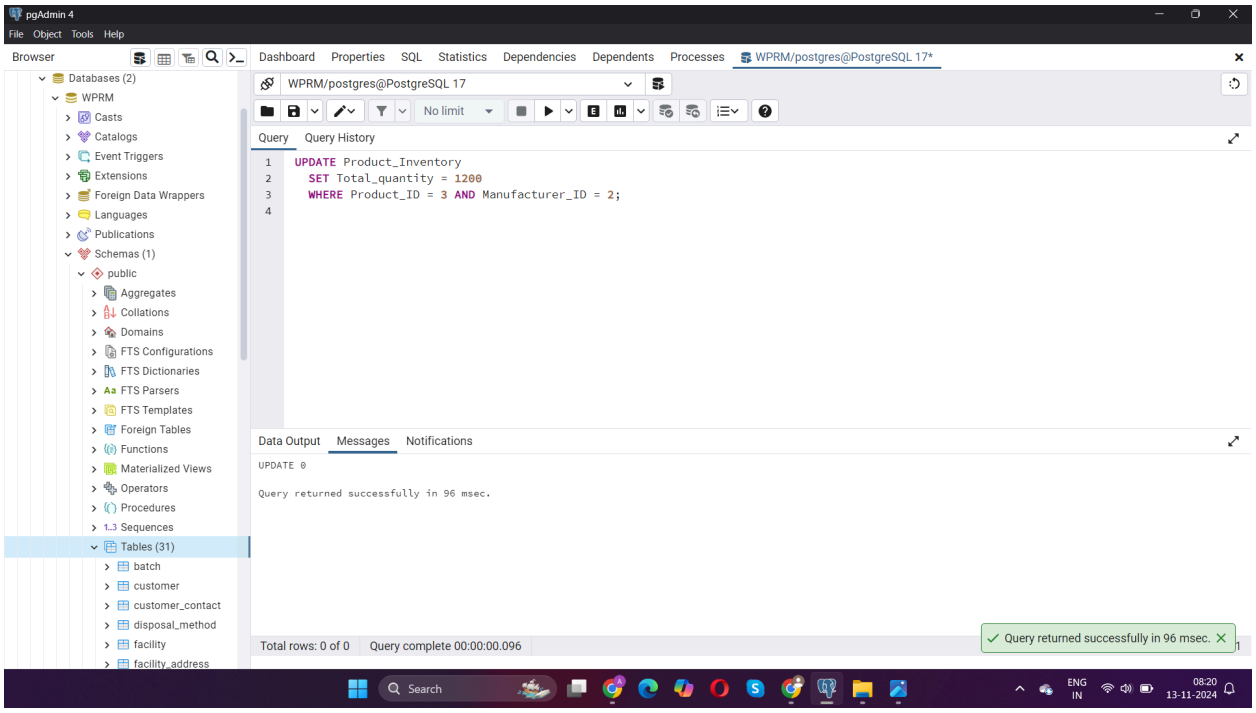
2. Update an existing manufacturer in the Manufacturer table:

```
UPDATE Manufacturer
SET Address_ID = 5,
    Sales_ID = 10,
    Name = 'WoodTech Industries',
    Wood_purchased = 10000
WHERE Manufacturer_ID = 81;
```



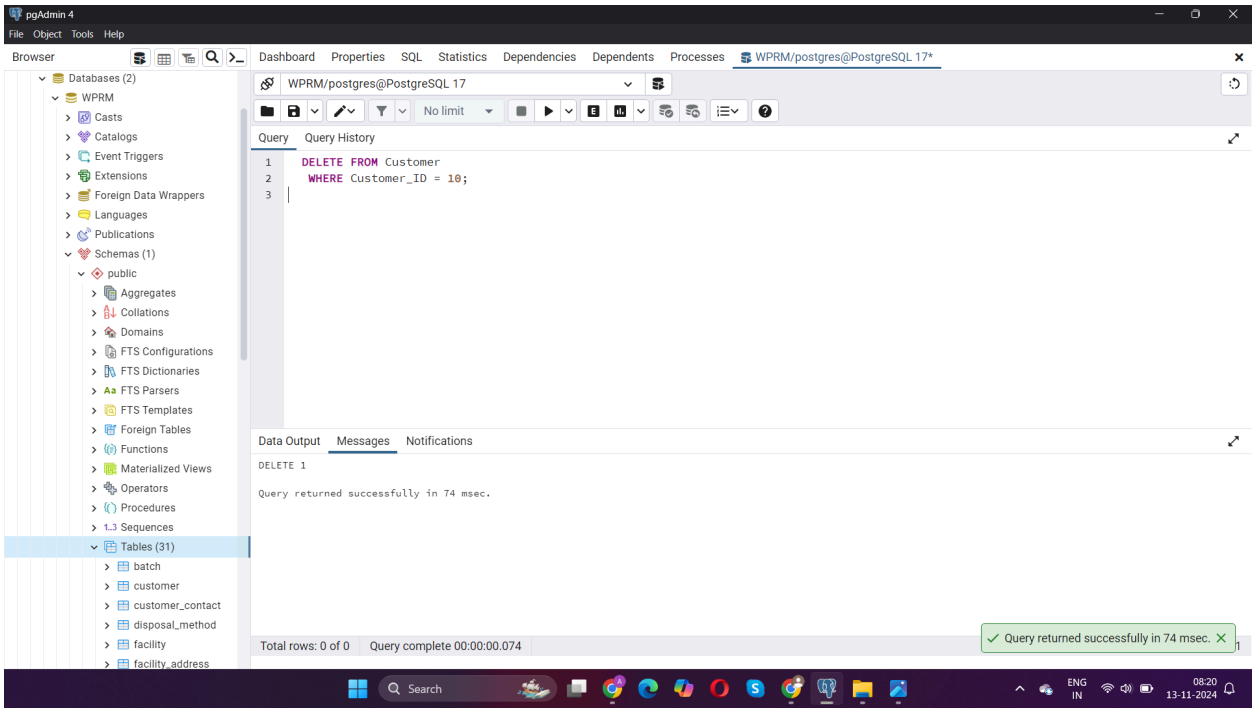
3. Update the total quantity in the `Product_Inventory` table:

```
UPDATE Product_Inventory
SET Total_quantity = 1200
WHERE Product_ID = 3 AND Manufacturer_ID = 2;
```



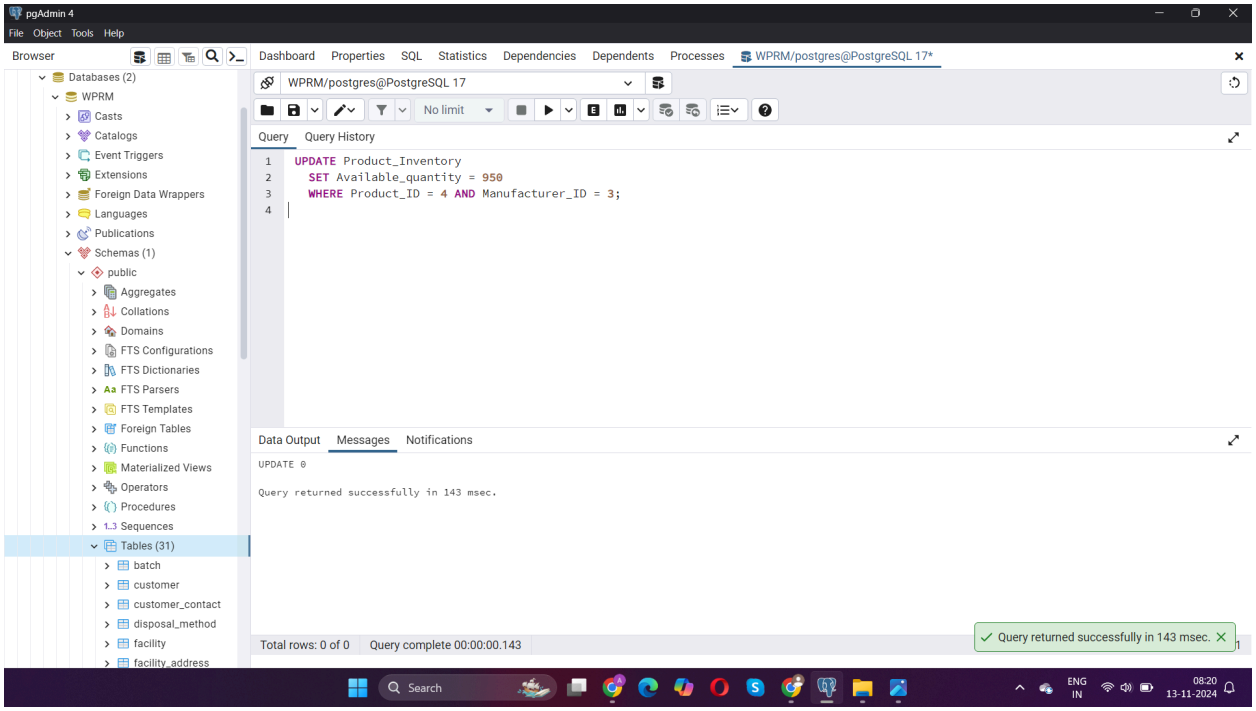
4. Delete a record from the `Customer` table:

```
DELETE FROM Customer
WHERE Customer_ID = 10;
```



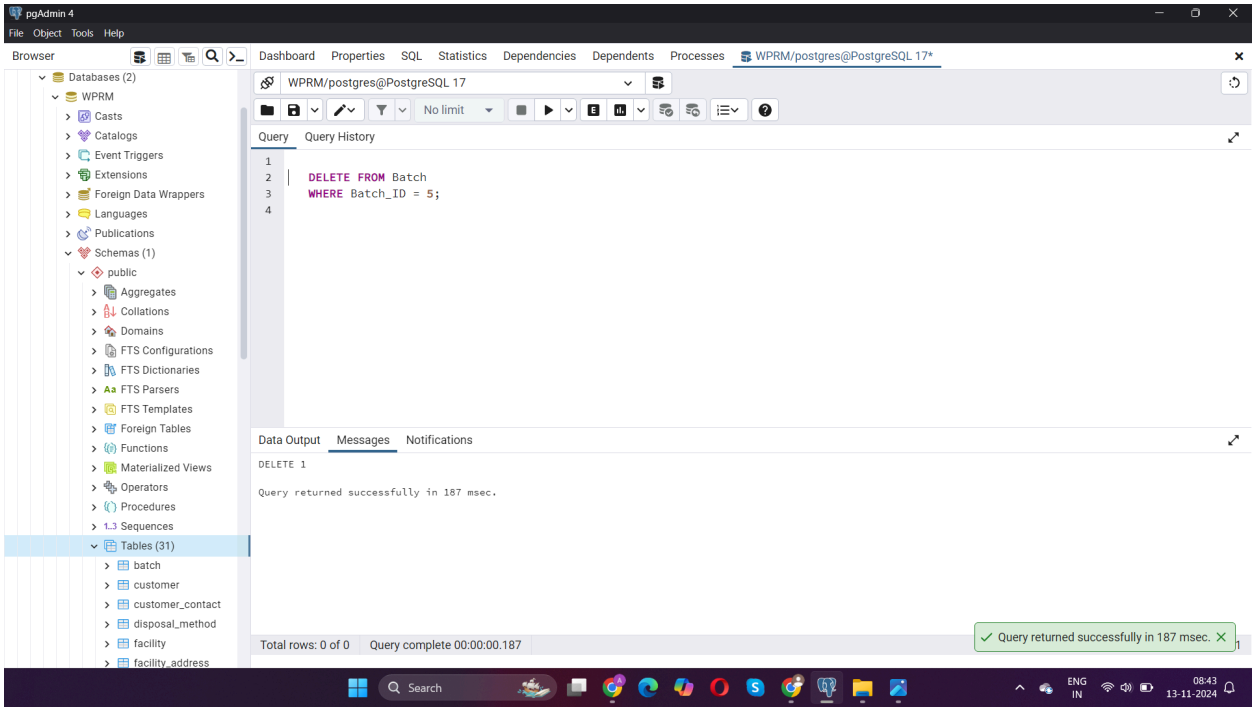
5. Update the available quantity in the `Product_Inventory` table:

```
UPDATE Product_Inventory
SET Available_quantity = 950
WHERE Product_ID = 4 AND Manufacturer_ID = 3;
```



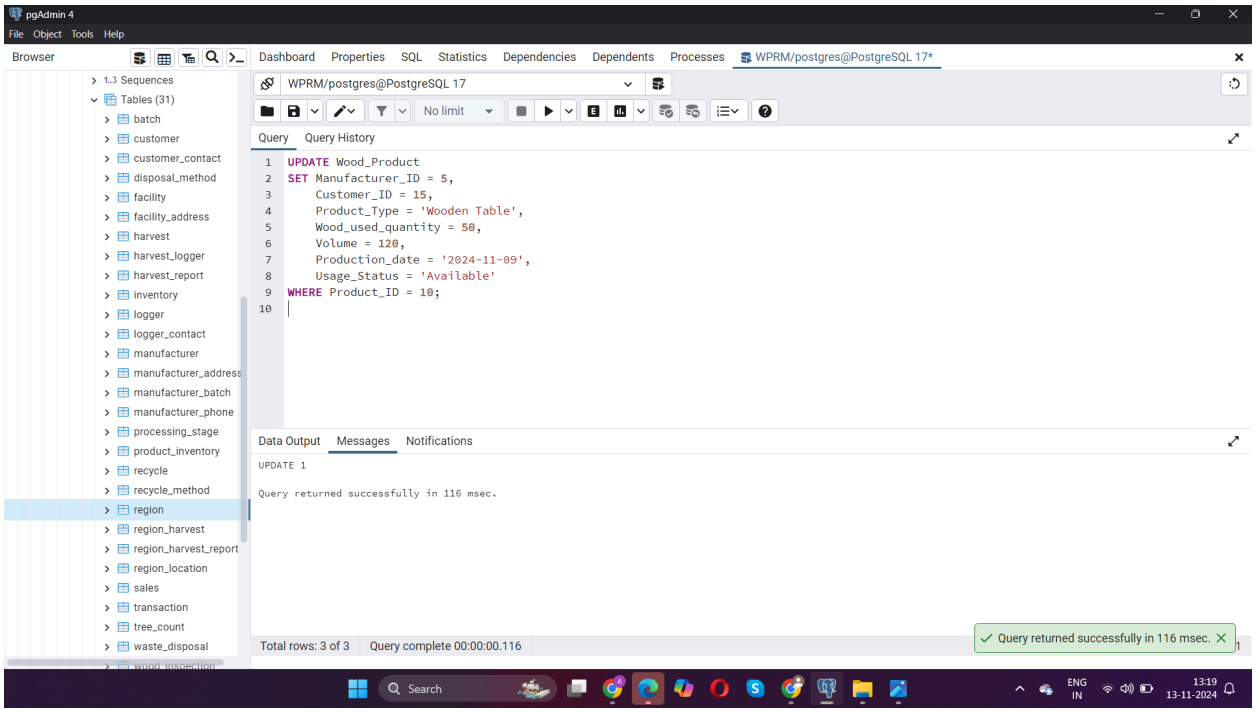
6. Delete a batch from the `Batch` table:

```
DELETE FROM Batch
WHERE Batch_ID = 5;
```



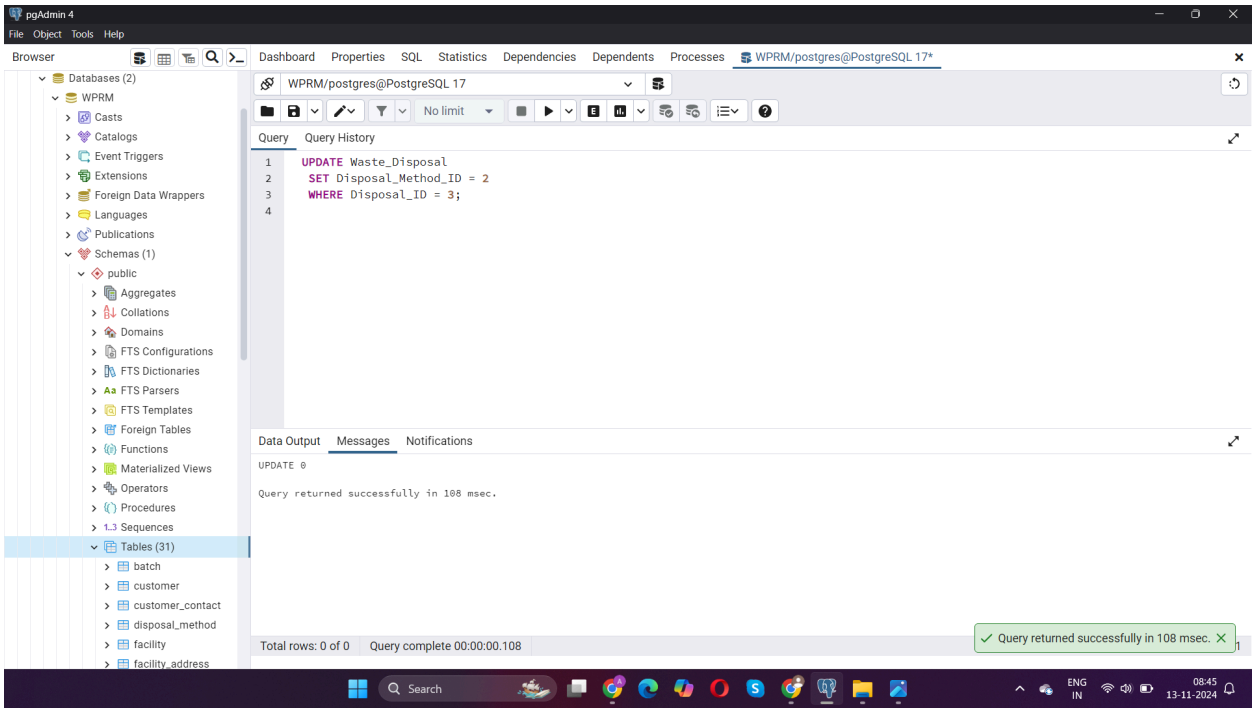
7. Update a new product into the `Wood_Product` table:

```
UPDATE Wood_Product
SET Manufacturer_ID = 5,
    Customer_ID = 15,
    Product_Type = 'Wooden Table',
    Wood_used_quantity = 50,
    Volume = 120,
    Production_date = '2024-11-09',
    Usage_Status = 'Available'
WHERE Product_ID = 10;
```



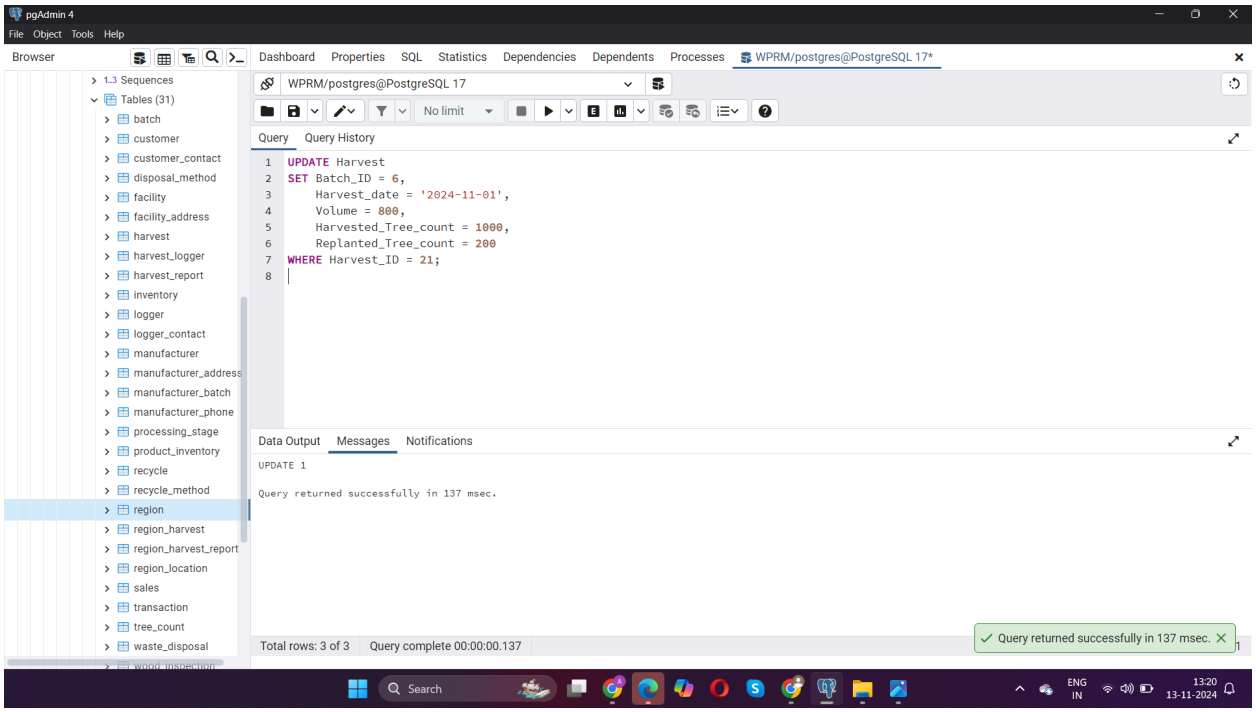
8. Change the disposal method in the `Waste_Disposal` table:

```
UPDATE Waste_Disposal
SET Disposal_Method_ID = 2
WHERE Disposal_ID = 3;
```



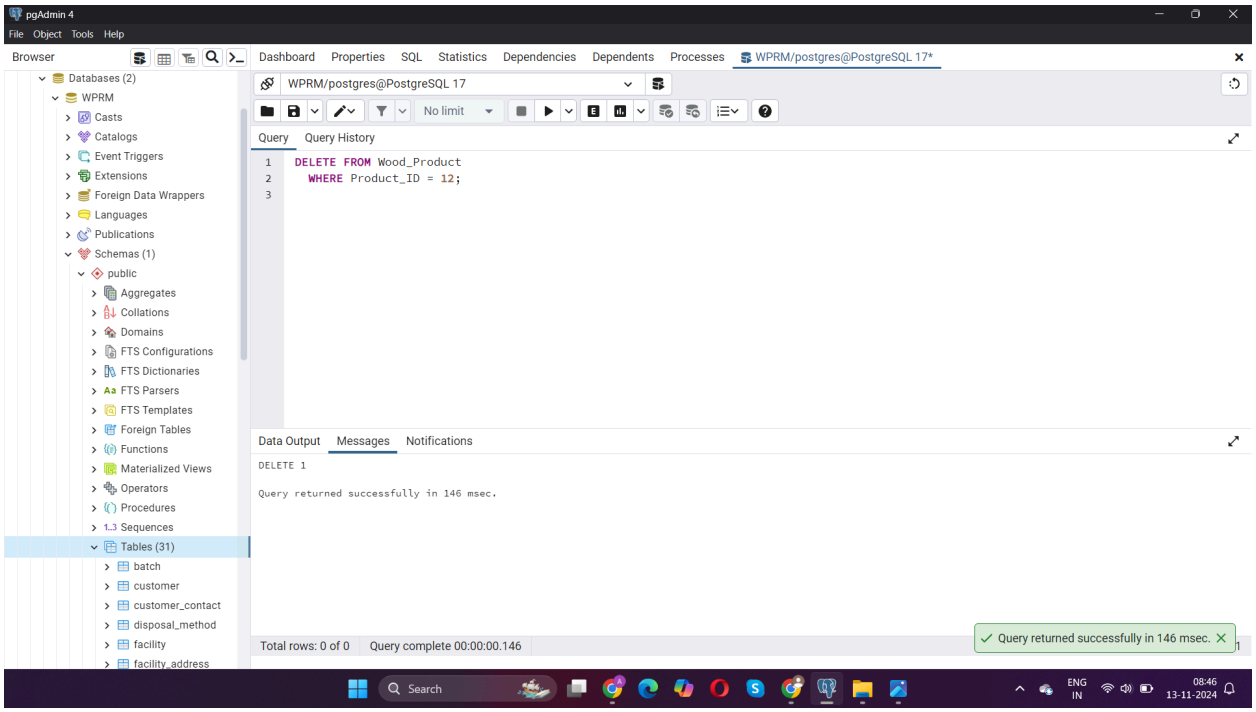
9. Update a new harvest record into the `Harvest` table:

```
UPDATE Harvest
SET Batch_ID = 6,
    Harvest_date = '2024-11-01',
    Volume = 800,
    Harvested_Tree_count = 1000,
    Replanted_Tree_count = 200
WHERE Harvest_ID = 21;
```



10. Delete a product record from the `Wood_Product` table:

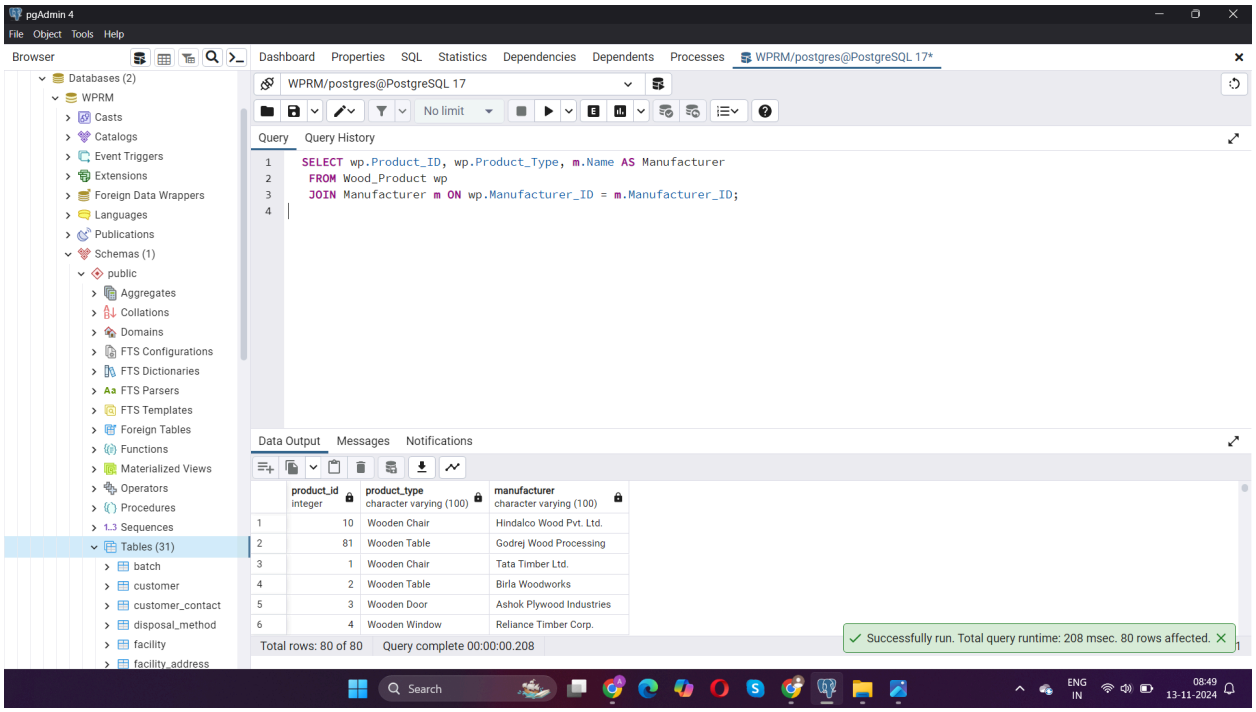
```
DELETE FROM Wood_Product
WHERE Product_ID = 12;
```



Intermediate Queries (With Joins)

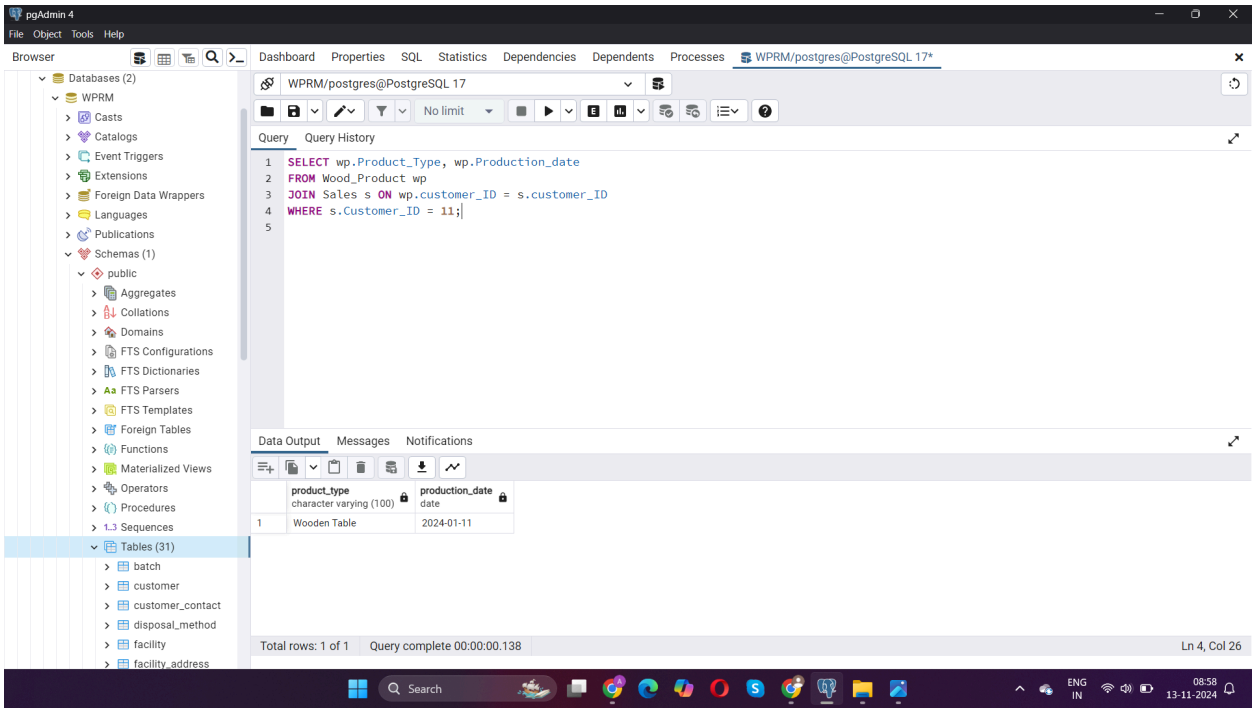
11.Select all products with their manufacturer name:

```
SELECT wp.Product_ID, wp.Product_Type, m.Name AS Manufacturer
FROM Wood_Product wp
JOIN Manufacturer m ON wp.Manufacturer_ID = m.Manufacturer_ID;
```



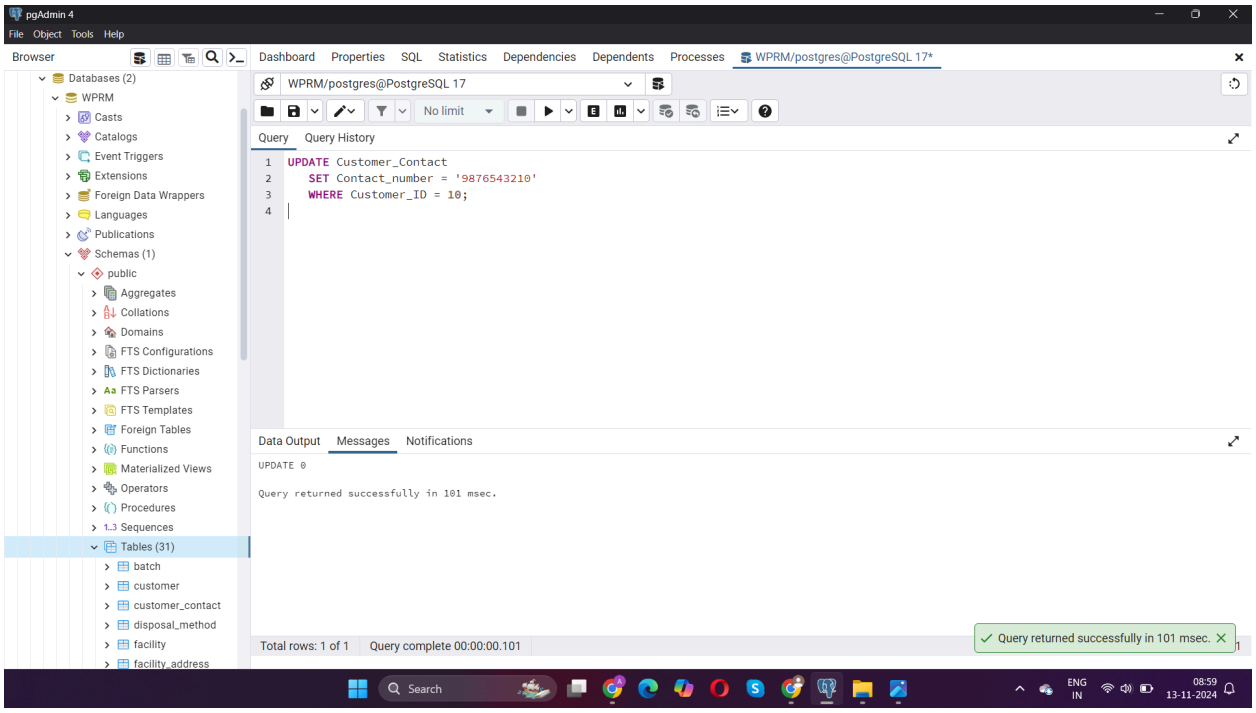
12. Get all products sold by a specific customer:

```
SELECT wp.Product_Type, wp.Production_date
FROM Wood_Product wp
JOIN Sales s ON wp.customer_ID = s.customer_ID
WHERE s.Customer_ID = 11;
```



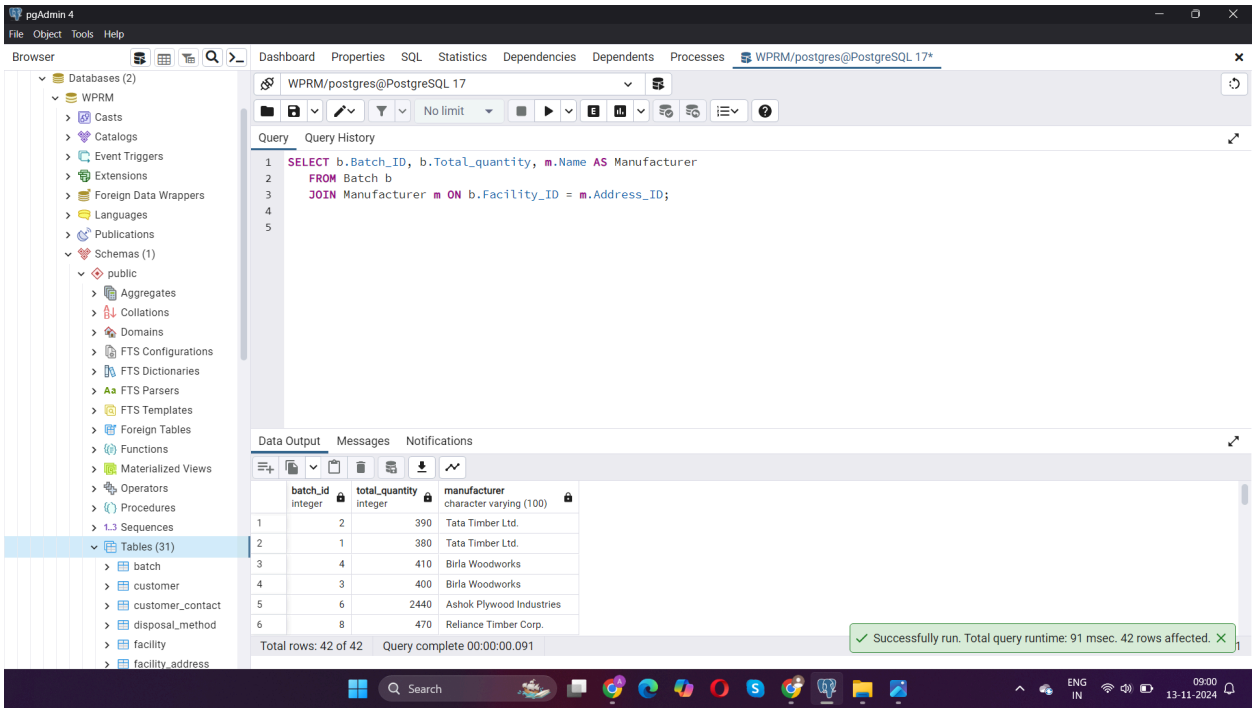
13. Update the customer contact number in the `Customer_Contact` table:

```
UPDATE Customer_Contact
SET Contact_number = '9876543210'
WHERE Customer_ID = 10;
```



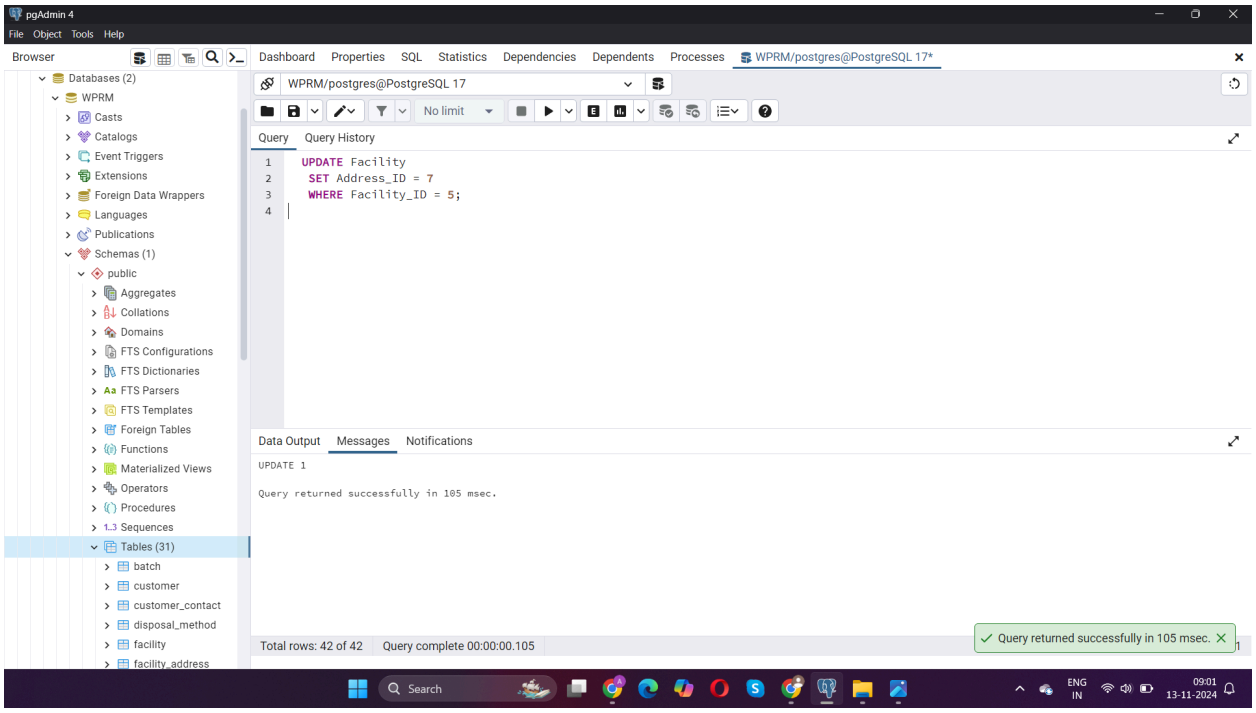
14. Get all batches and their corresponding manufacturers:

```
SELECT b.Batch_ID, b.Total_quantity, m.Name AS Manufacturer
FROM Batch b
JOIN Manufacturer m ON b.Facility_ID = m.Address_ID;
```



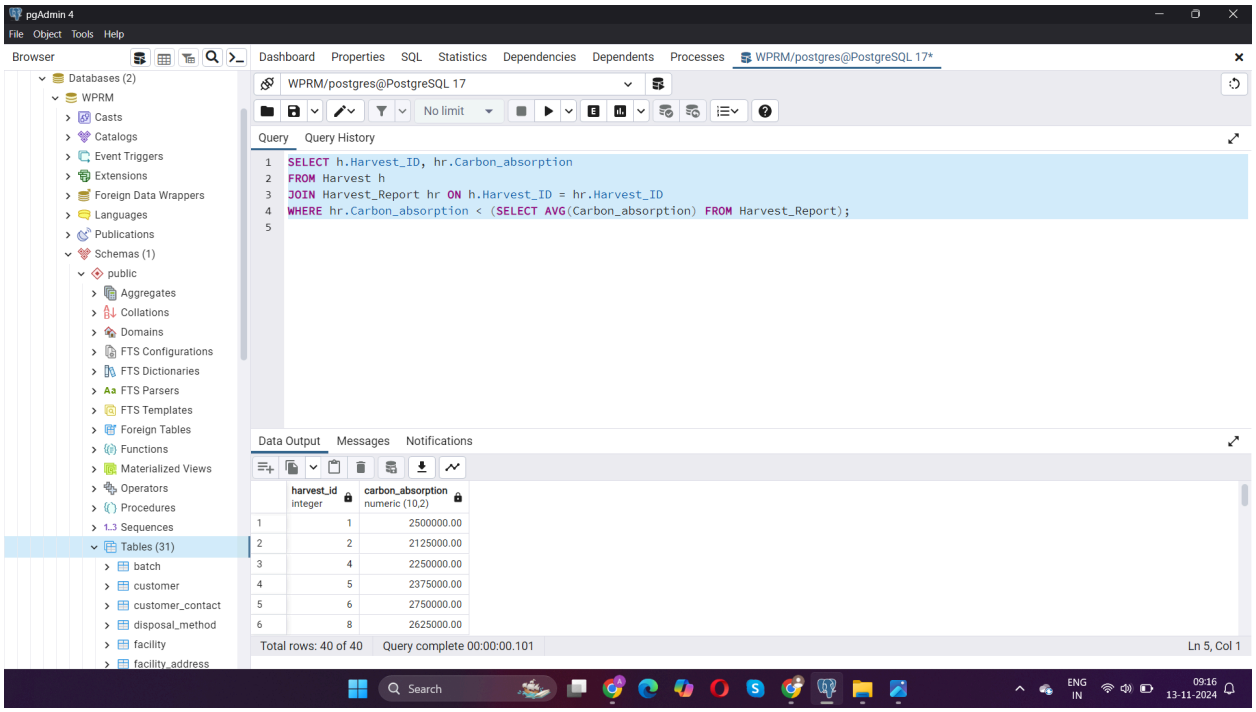
15.Update a facility address based on its ID:

```
UPDATE Facility
SET Address_ID = 7
WHERE Facility_ID = 5;
```



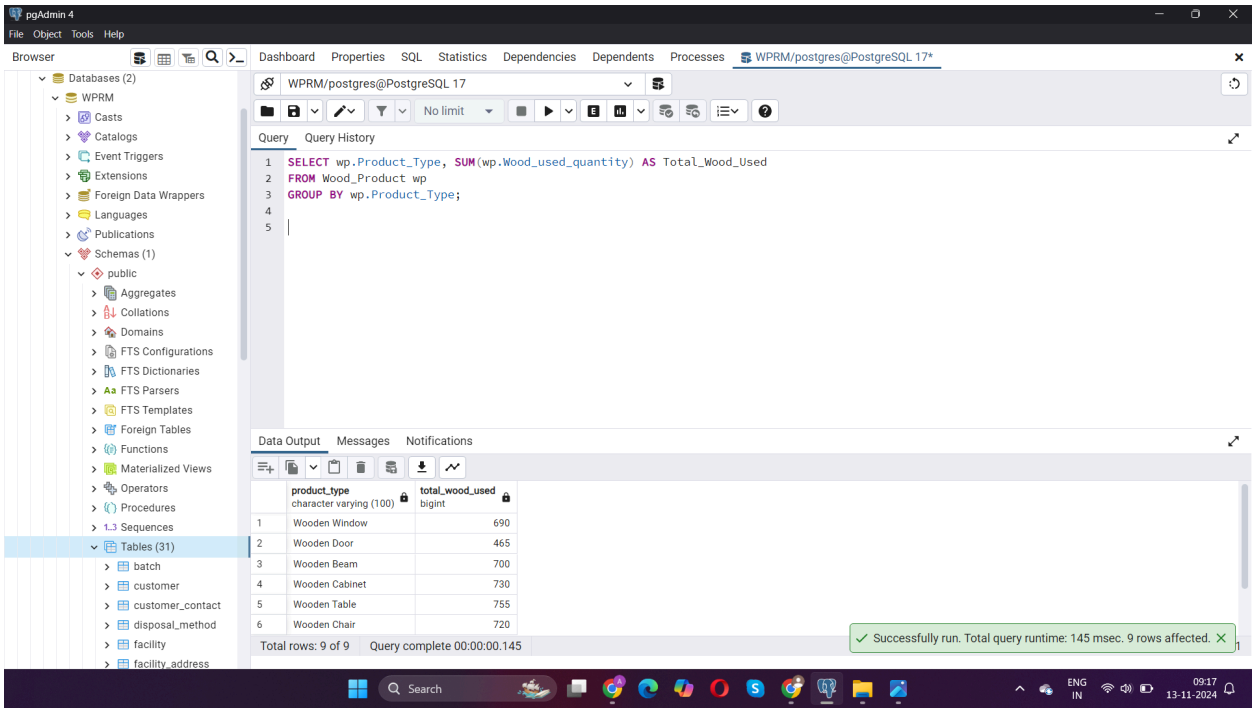
16. List all harvests where carbon absorption is below the average:

```
SELECT h.Harvest_ID, hr.Carbon_absorption
FROM Harvest h
JOIN Harvest_Report hr ON h.Harvest_ID = hr.Harvest_ID
WHERE hr.Carbon_absorption <
(SELECT AVG(Carbon_absorption) FROM Harvest_Report);
```



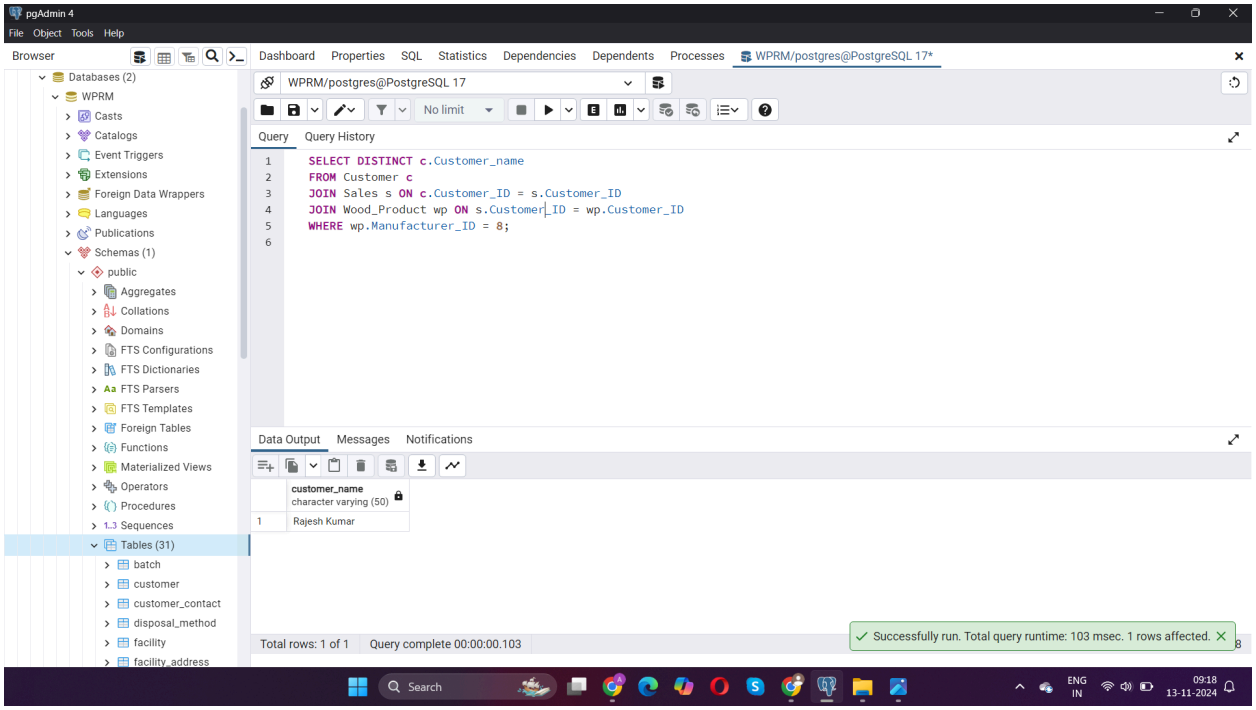
17. Get total quantity of wood used in the manufacturing of each product:

```
SELECT wp.Product_Type, SUM(wp.Wood_used_quantity) AS
Total_Wood_Used
FROM Wood_Product wp
GROUP BY wp.Product_Type;
```



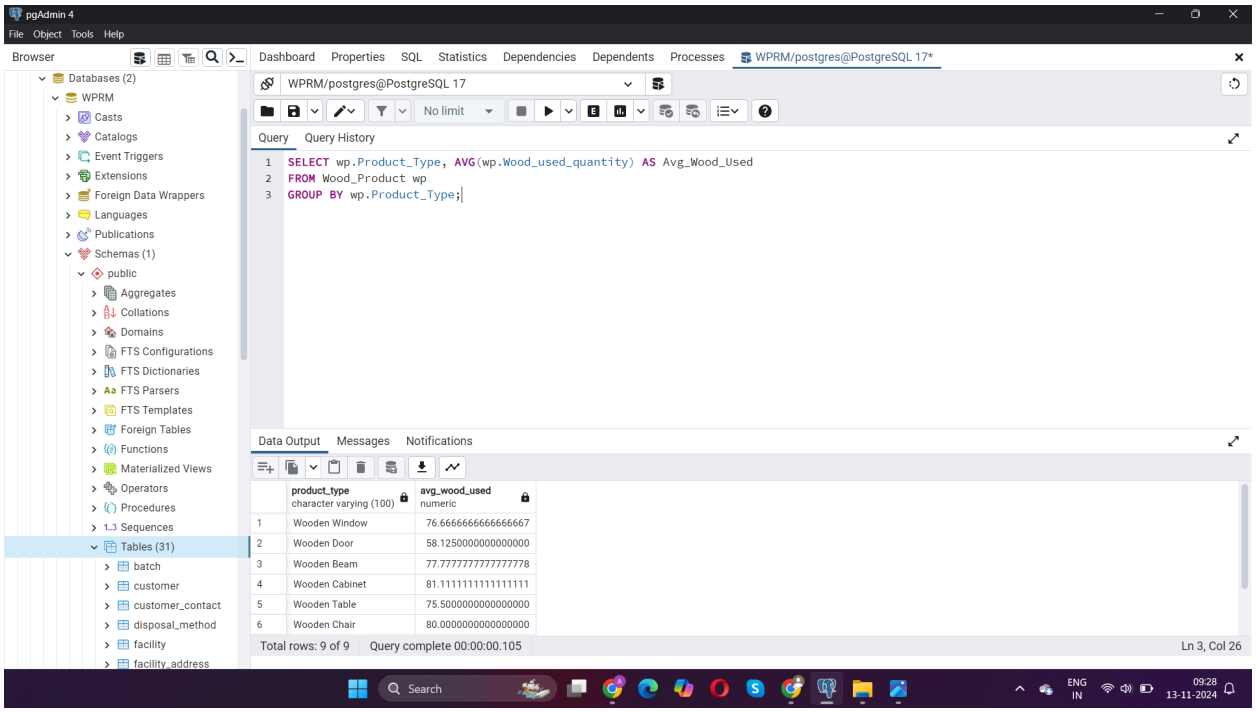
18.Select customers who have purchased products from a specific manufacturer:

```
SELECT DISTINCT c.Customer_name
FROM Customer c
JOIN Sales s ON c.Customer_ID = s.Customer_ID
JOIN Wood_Product wp ON s.Customer_ID = wp.Customer_ID
WHERE wp.Manufacturer_ID = 8;
```


pgAdmin 4
File Object Tools Help
Browser
Databases (2)
WPRM
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (31)
batch
customer
customer_contact
disposal_method
facility
facility_address
Dashboard Properties SQL Statistics Dependencies Dependents Processes WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
1 SELECT DISTINCT c.Customer_name
2 FROM Customer c
3 JOIN Sales s ON c.Customer_ID = s.Customer_ID
4 JOIN Wood_Product wp ON s.Customer_ID = wp.Customer_ID
5 WHERE wp.Manufacturer_ID = 8;
6
Data Output Messages Notifications
customer_name
character varying (50)
1 Rajesh Kumar
Total rows: 1 of 1
Query complete 00:00:00.103
Successfully run. Total query runtime: 103 msec. 1 rows affected.
Search
ENG
IN
09:18
13-11-2024

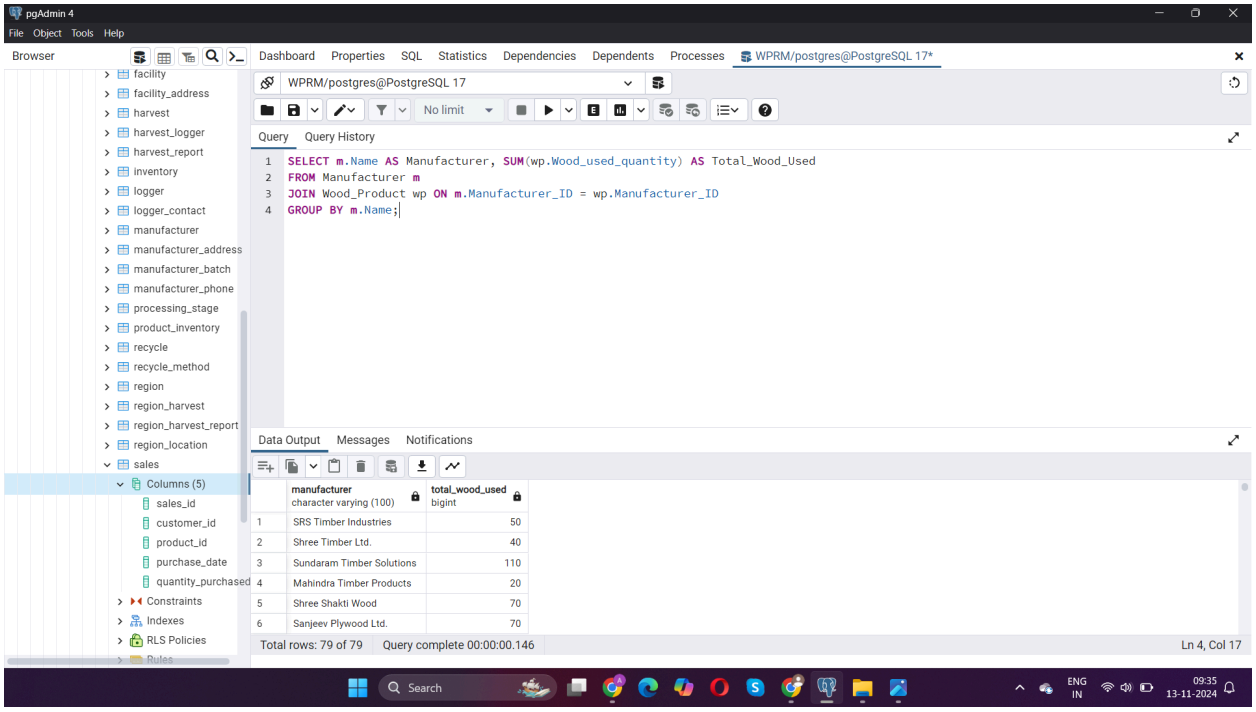
19.Get the average wood used per product type:

```
SELECT wp.Product_Type, AVG(wp.Wood_used_quantity) AS Avg_Wood_Used
FROM Wood_Product wp
GROUP BY wp.Product_Type;
```



20. Get the total quantity of wood used by each manufacturer for all products:

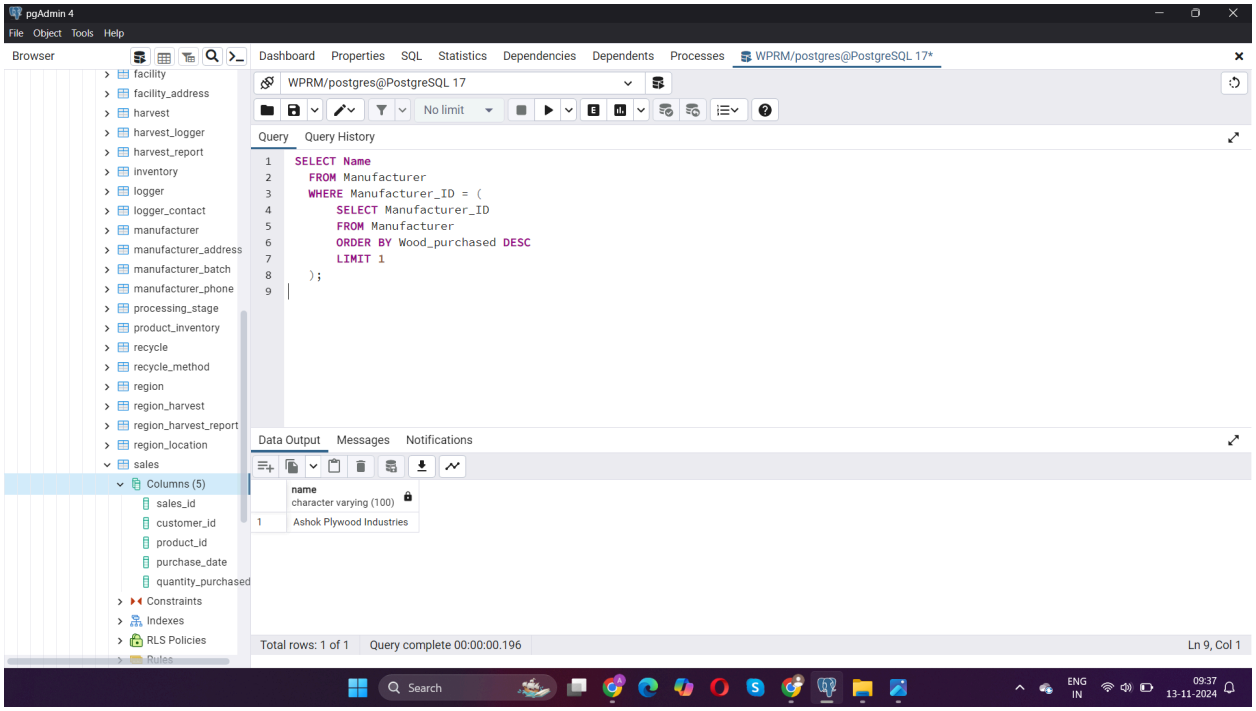
```
SELECT m.Name AS Manufacturer, SUM(wp.Wood_used_quantity) AS
Total_Wood_Used
FROM Manufacturer m
JOIN Wood_Product wp ON m.Manufacturer_ID = wp.Manufacturer_ID
GROUP BY m.Name;
```



Complex Queries (Subqueries, Multiple Joins)

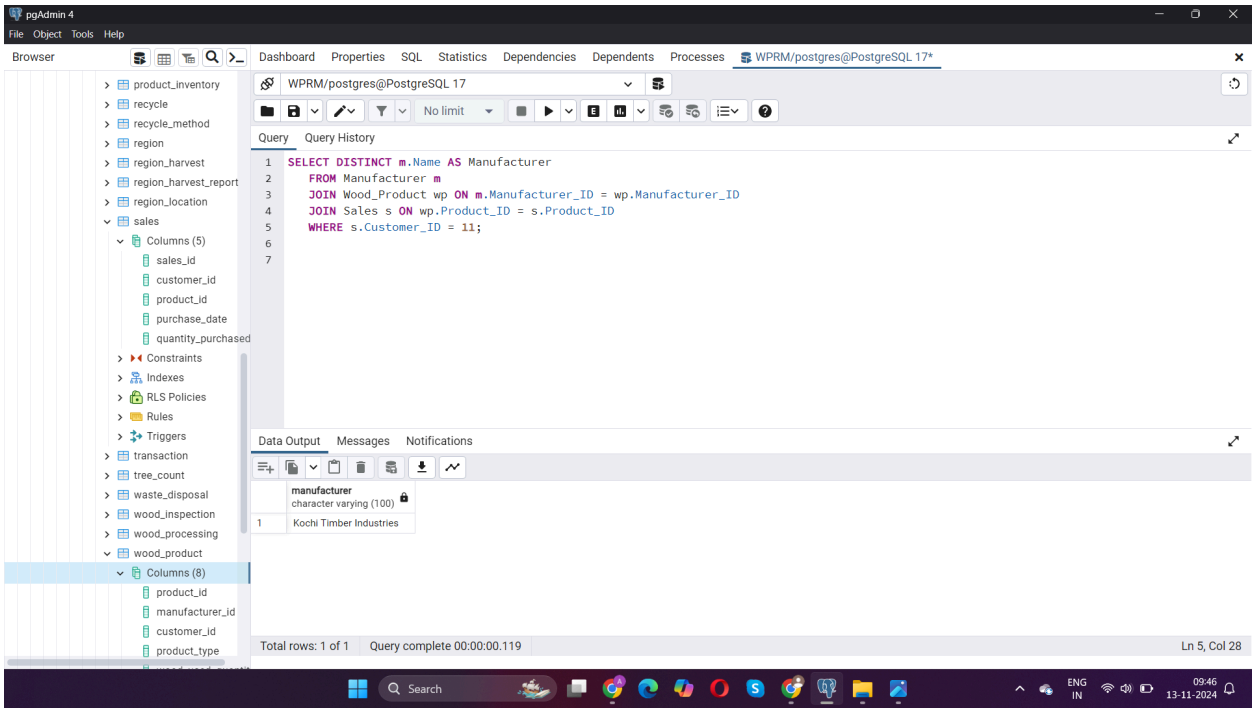
21.Get the manufacturer with the highest total wood purchased:

```
SELECT Name
FROM Manufacturer
WHERE Manufacturer_ID = (
    SELECT Manufacturer_ID
    FROM Manufacturer
    ORDER BY Wood_purchased DESC
    LIMIT 1
);
```



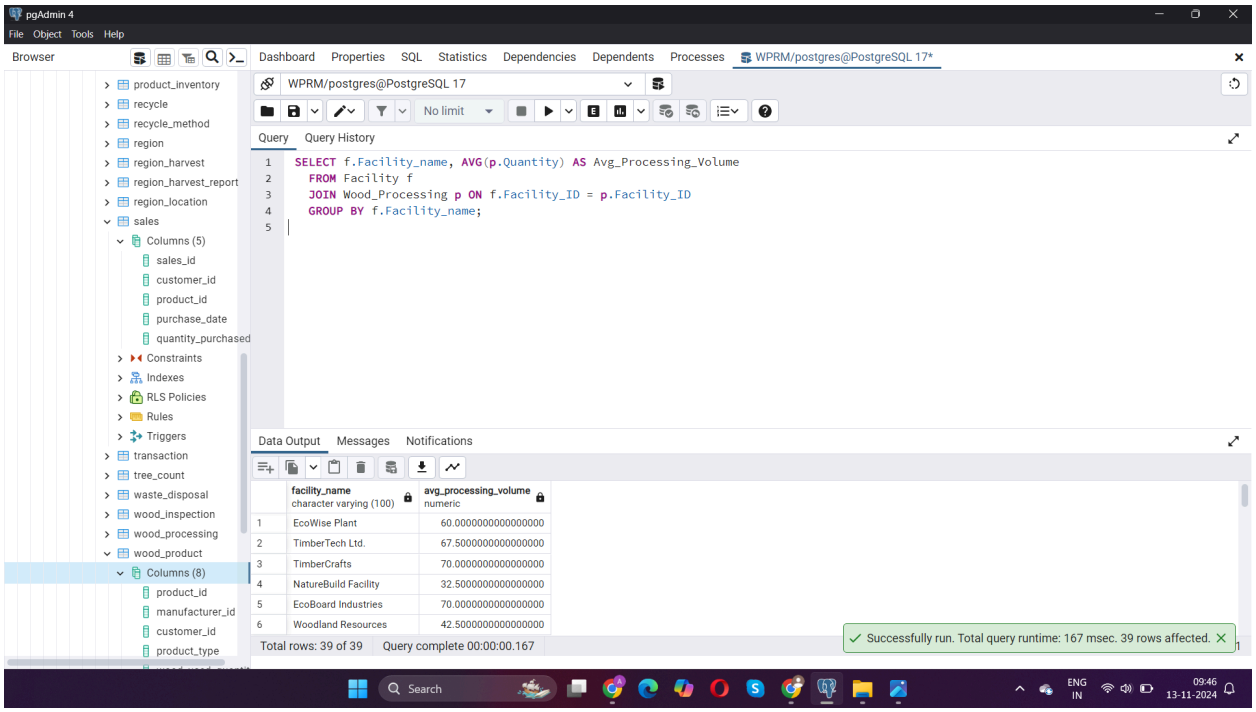
22. Get all manufacturers who have products that are sold to a specific customer:

```
SELECT DISTINCT m.Name AS Manufacturer
FROM Manufacturer m
JOIN Wood_Product wp ON m.Manufacturer_ID = wp.Manufacturer_ID
JOIN Sales s ON wp.Product_ID = s.Product_ID
WHERE s.Customer_ID = 10;
```



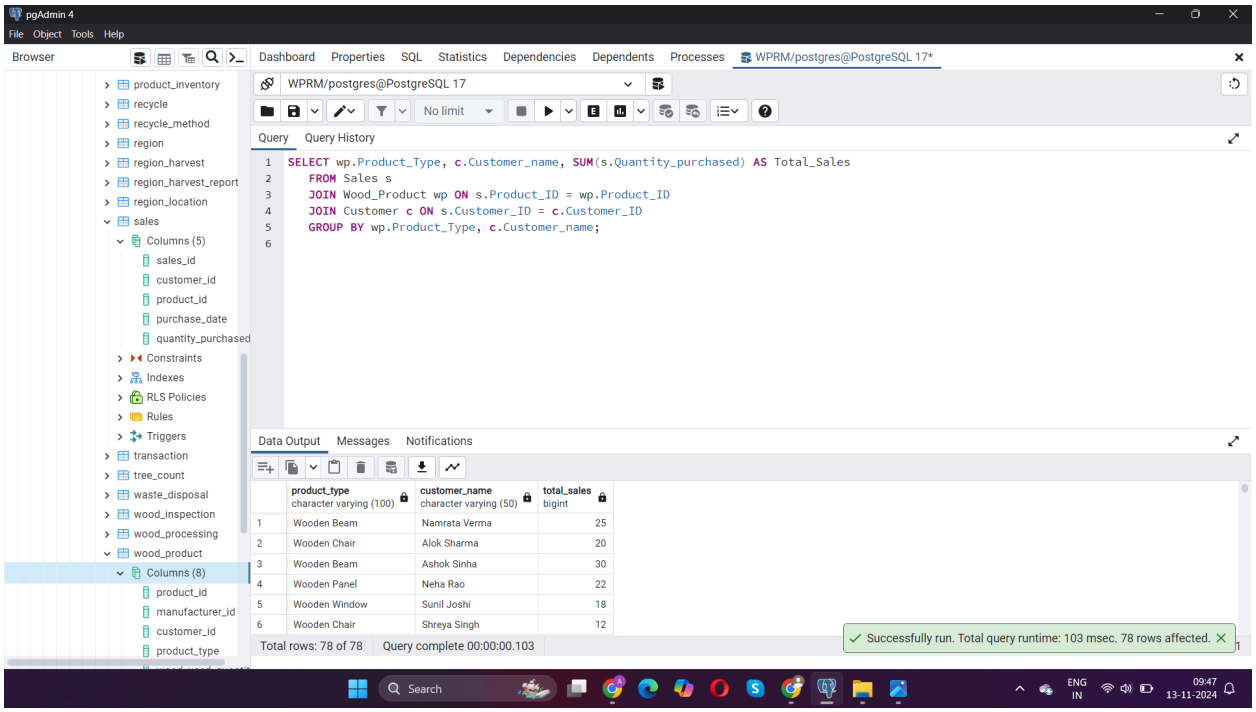
23. Get the average volume of wood processed by each facility:

```
SELECT f.Facility_name, AVG(p.Quantity) AS Avg_Processing_Volume
FROM Facility f
JOIN Wood_Processing p ON f.Facility_ID = p.Facility_ID
GROUP BY f.Facility_name;
```



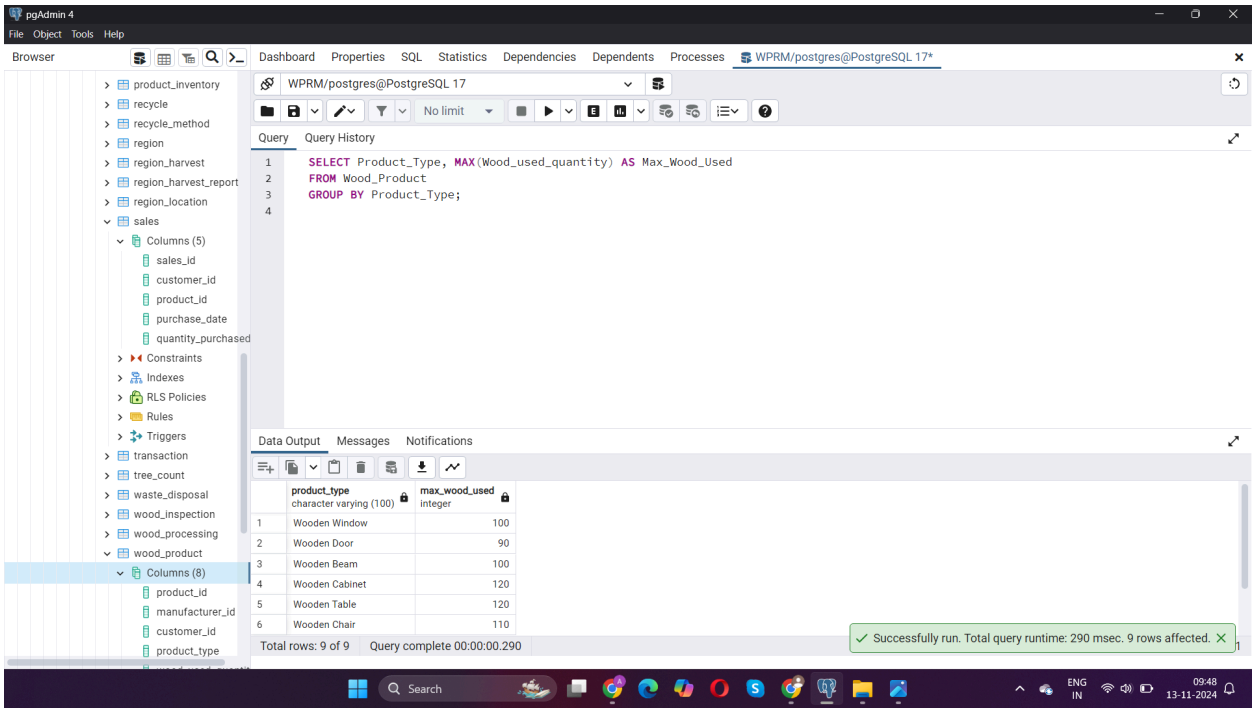
24.Get total sales amount for each product, including customer details:

```
SELECT wp.Product_Type, c.Customer_name, SUM(s.Quantity_purchased) AS
Total_Sales
  FROM Sales s
  JOIN Wood_Product wp ON s.Product_ID = wp.Product_ID
  JOIN Customer c ON s.Customer_ID = c.Customer_ID
  GROUP BY wp.Product_Type, c.Customer_name;
```



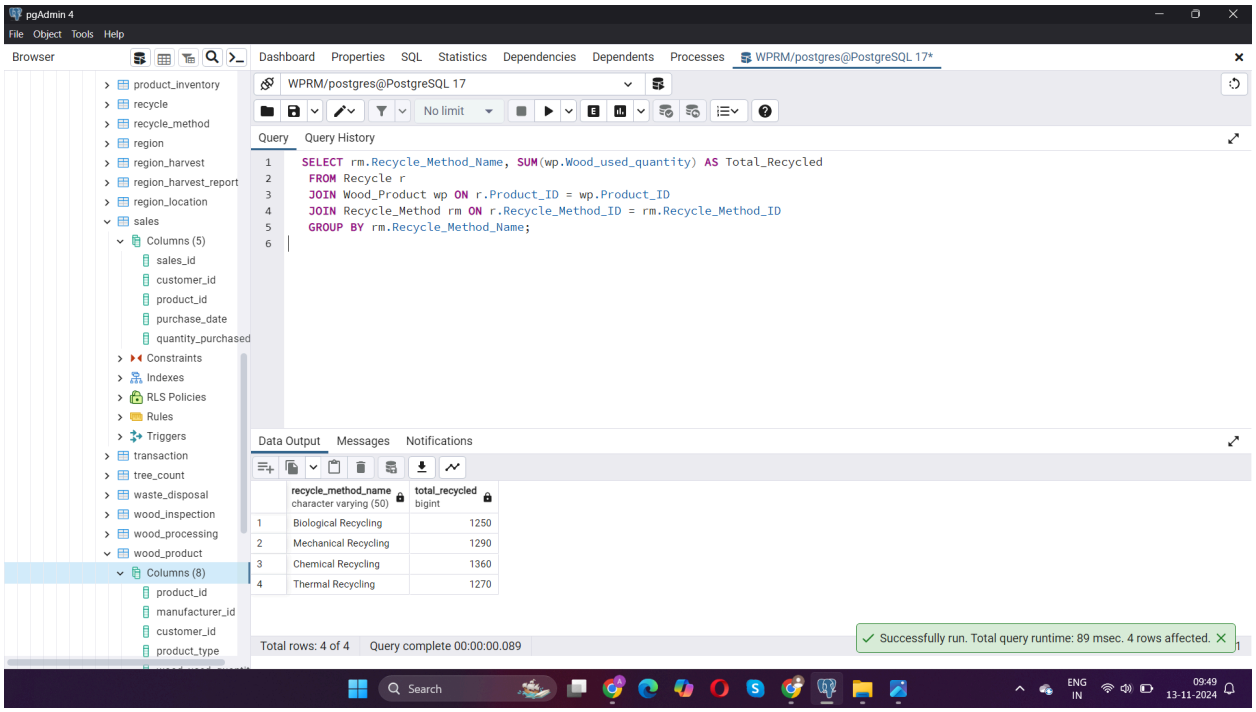
25.Get the product with the most wood used in production:

```
SELECT Product_Type, MAX(Wood_used_quantity) AS Max_Wood_Used
FROM Wood_Product
GROUP BY Product_Type;
```



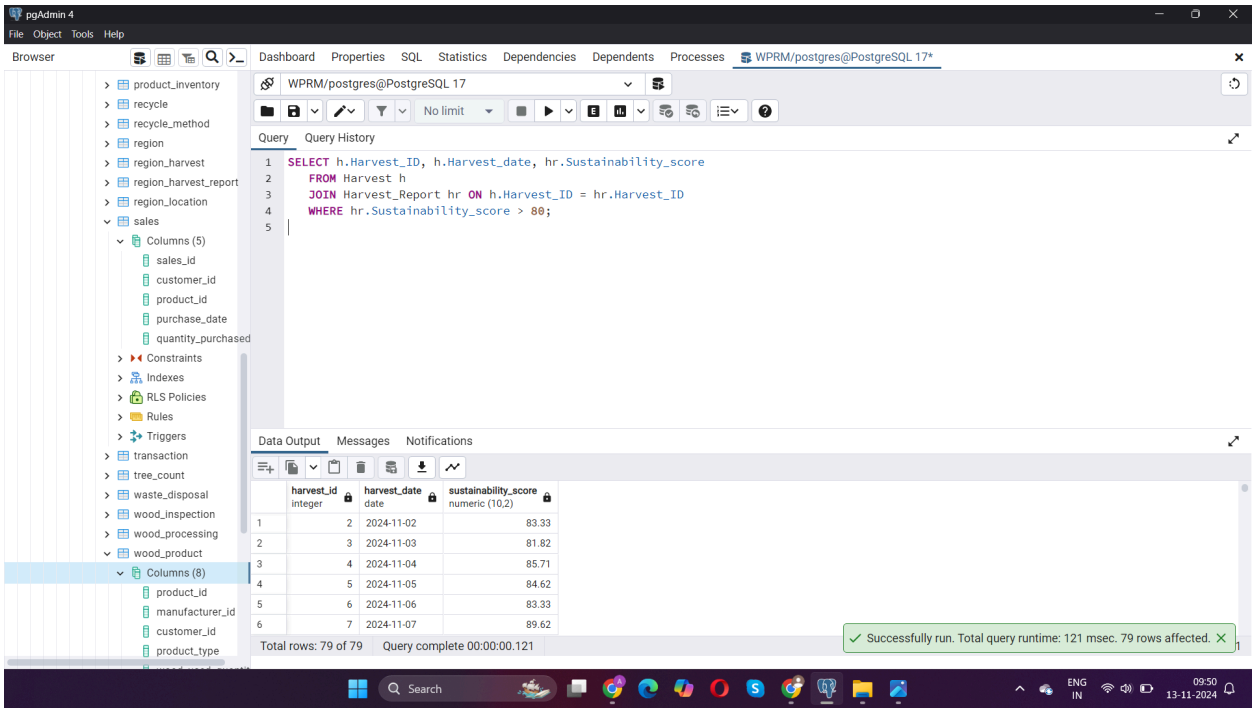
26. Get the total quantity of wood recycled by each recycling method:

```
SELECT rm.Recycle_Method_Name, SUM(wp.Wood_used_quantity) AS
Total_Recycled
FROM Recycle r
JOIN Wood_Product wp ON r.Product_ID = wp.Product_ID
JOIN Recycle_Method rm ON r.Recycle_Method_ID = rm.Recycle_Method_ID
GROUP BY rm.Recycle_Method_Name;
```



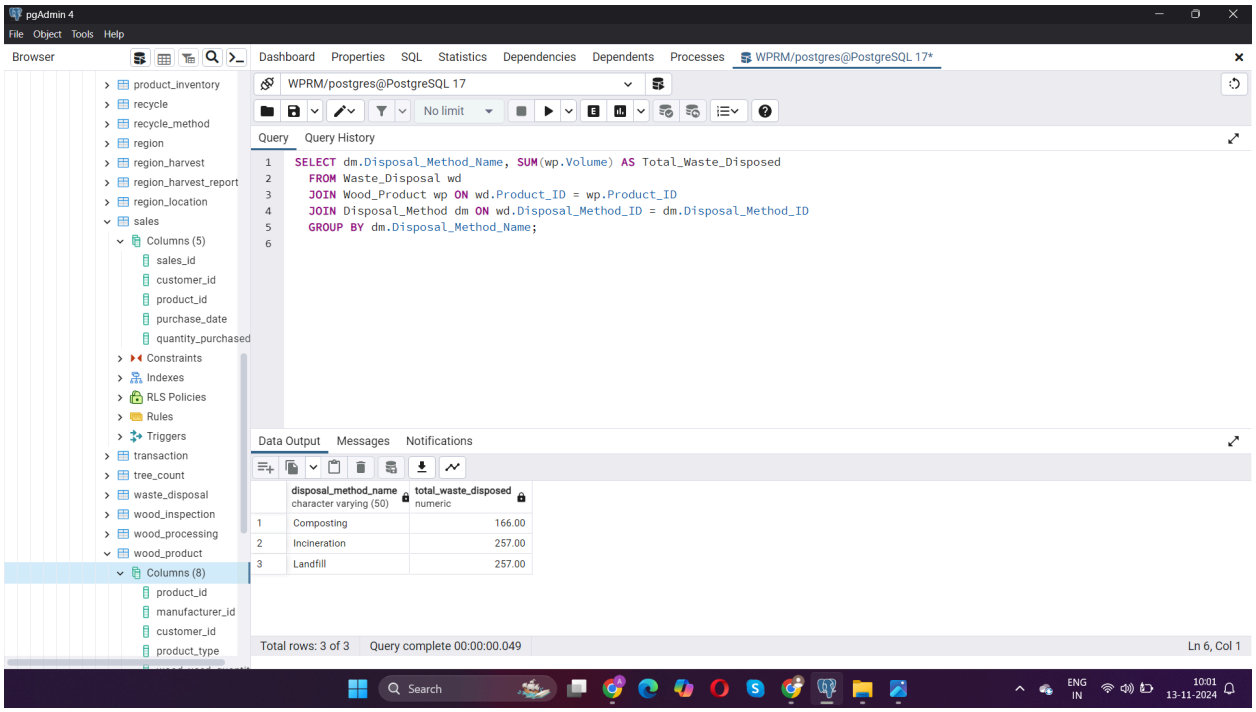
27. Get all harvests where the sustainability score is above 80:

```
SELECT h.Harvest_ID, h.Harvest_date, hr.Sustainability_score
FROM Harvest h
JOIN Harvest_Report hr ON h.Harvest_ID = hr.Harvest_ID
WHERE hr.Sustainability_score > 80;
```



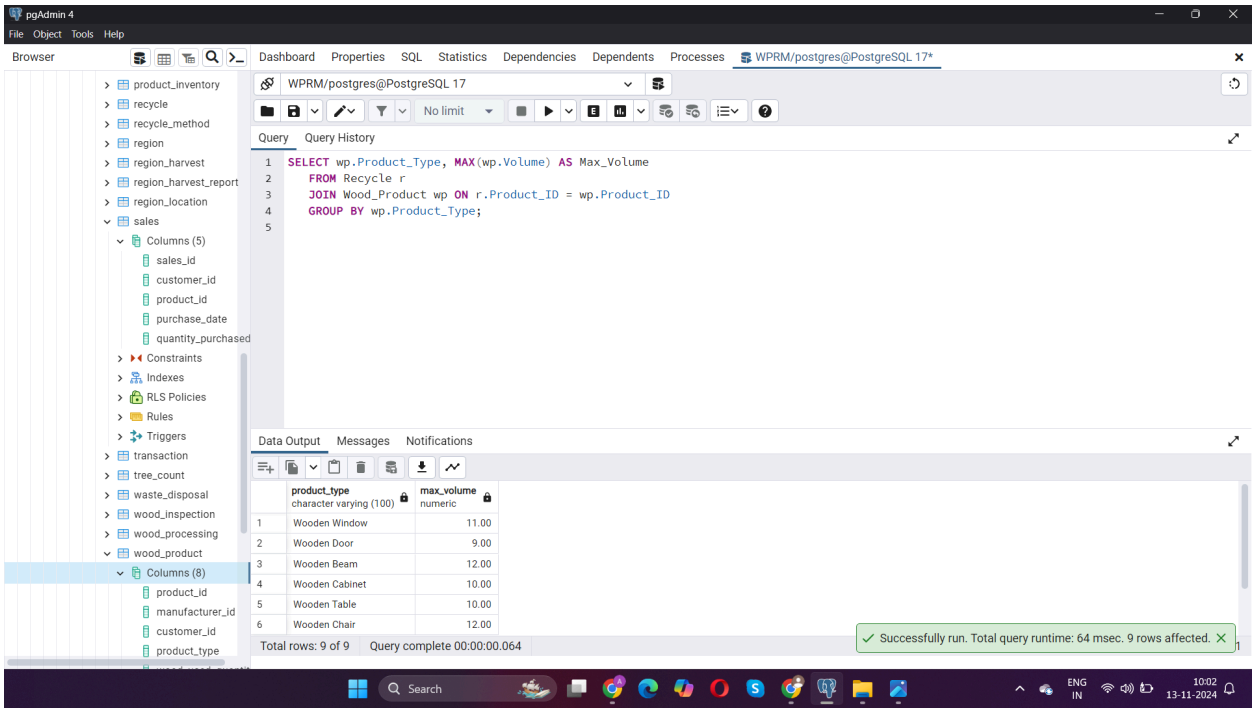
28. Get the total waste disposed of by each disposal method:

```
SELECT dm.Disposal_Method_Name, SUM(wp.Volume) AS
Total_Waste_Disposed
  FROM Waste_Disposal wd
  JOIN Wood_Product wp ON wd.Product_ID = wp.Product_ID
  JOIN Disposal_Method dm ON wd.Disposal_Method_ID =
dm.Disposal_Method_ID
  GROUP BY dm.Disposal_Method_Name;
```



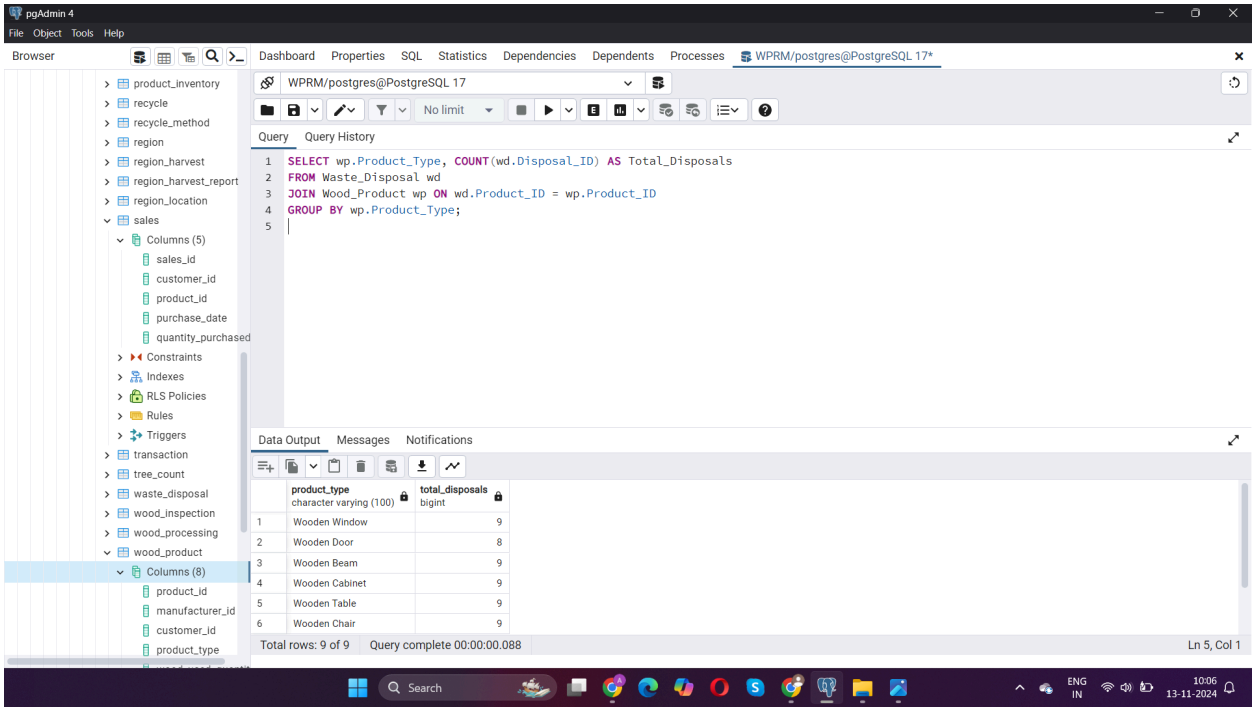
29. Get products with the highest volume that have been recycled:

```
SELECT wp.Product_Type, MAX(wp.Volume) AS Max_Volume
FROM Recycle r
JOIN Wood_Product wp ON r.Product_ID = wp.Product_ID
GROUP BY wp.Product_Type;
```



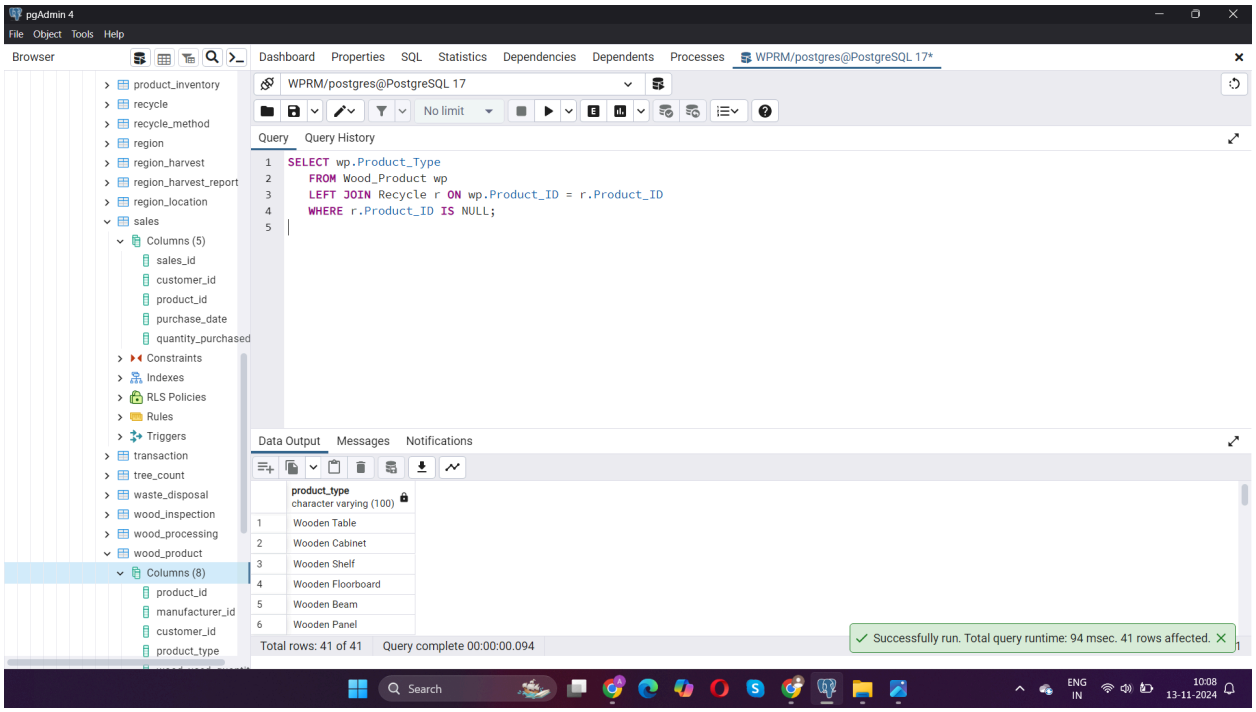
30.Find the total number of disposals for each product type:

```
SELECT wp.Product_Type, COUNT(wd.Disposal_ID) AS Total_Disposals
FROM Waste_Disposal wd
JOIN Wood_Product wp ON wd.Product_ID = wp.Product_ID
GROUP BY wp.Product_Type;
```



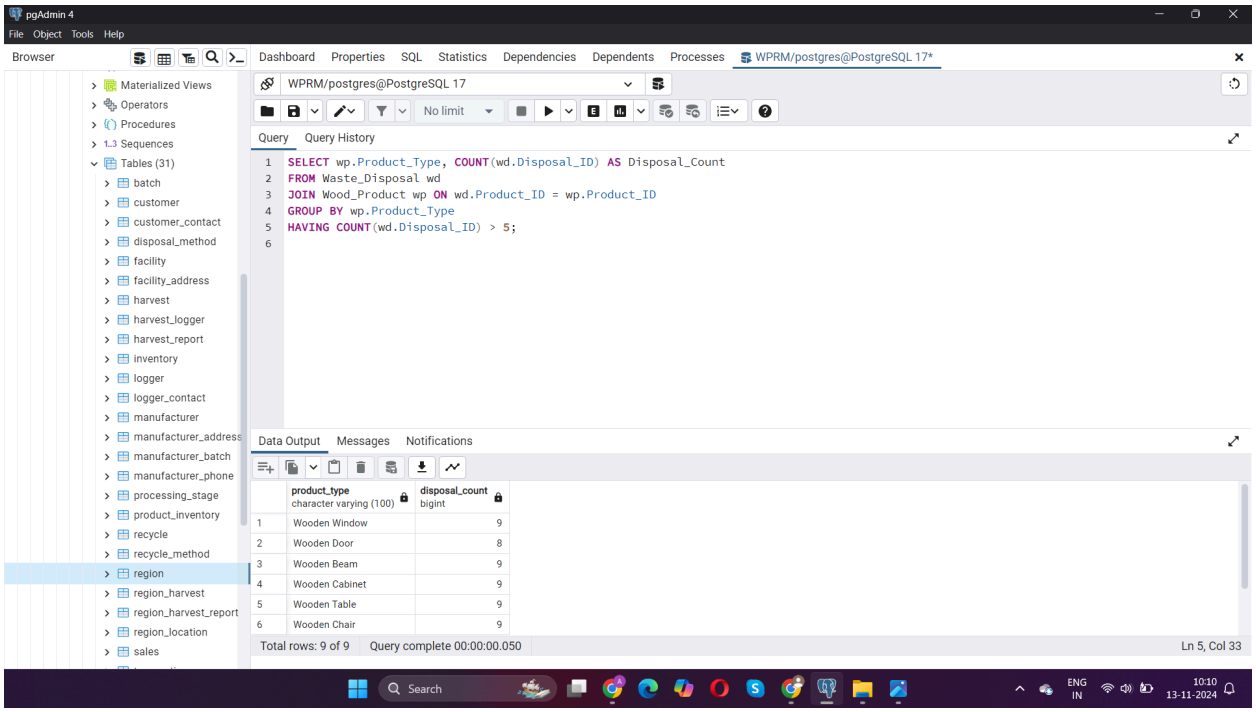
31. Get the list of products where no wood is recycled:

```
SELECT wp.Product_Type
FROM Wood_Product wp
LEFT JOIN Recycle r ON wp.Product_ID = r.Product_ID
WHERE r.Product_ID IS NULL;
```



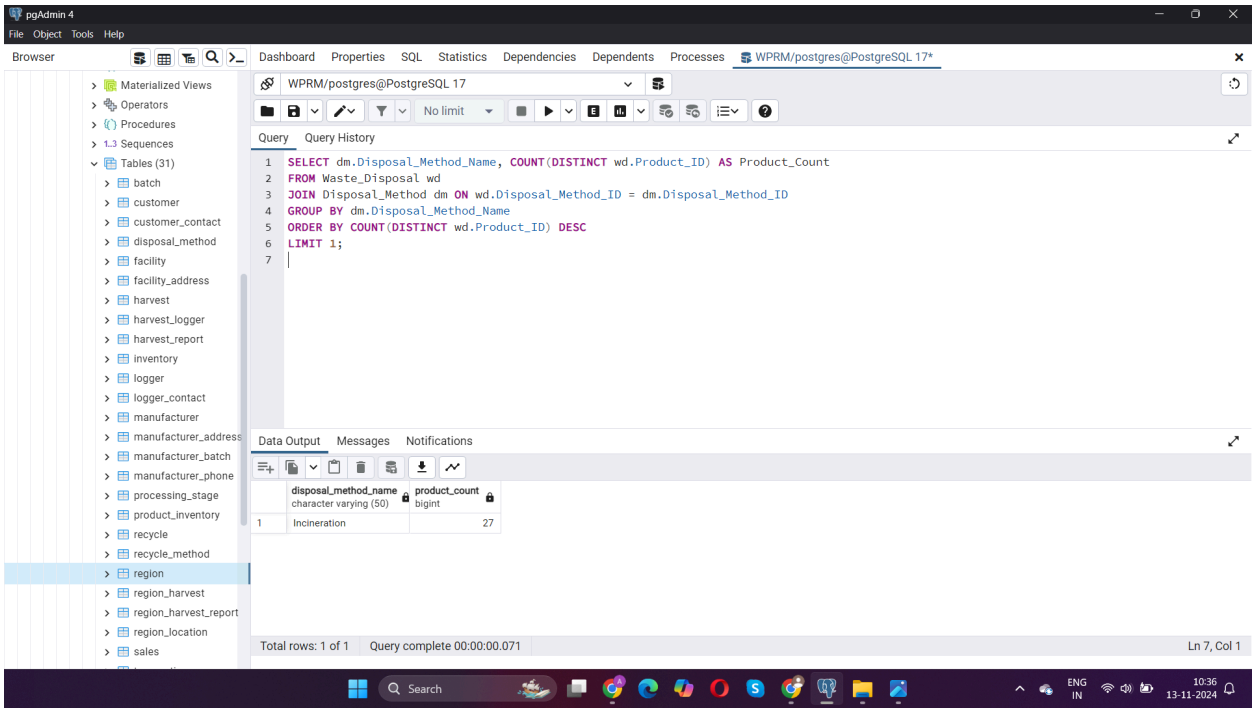
32. Get the product types that have more than 5 disposals:

```
SELECT wp.Product_Type, COUNT(wd.Disposal_ID) AS Disposal_Count
FROM Waste_Disposal wd
JOIN Wood_Product wp ON wd.Product_ID = wp.Product_ID
GROUP BY wp.Product_Type
HAVING COUNT(wd.Disposal_ID) > 5;
```



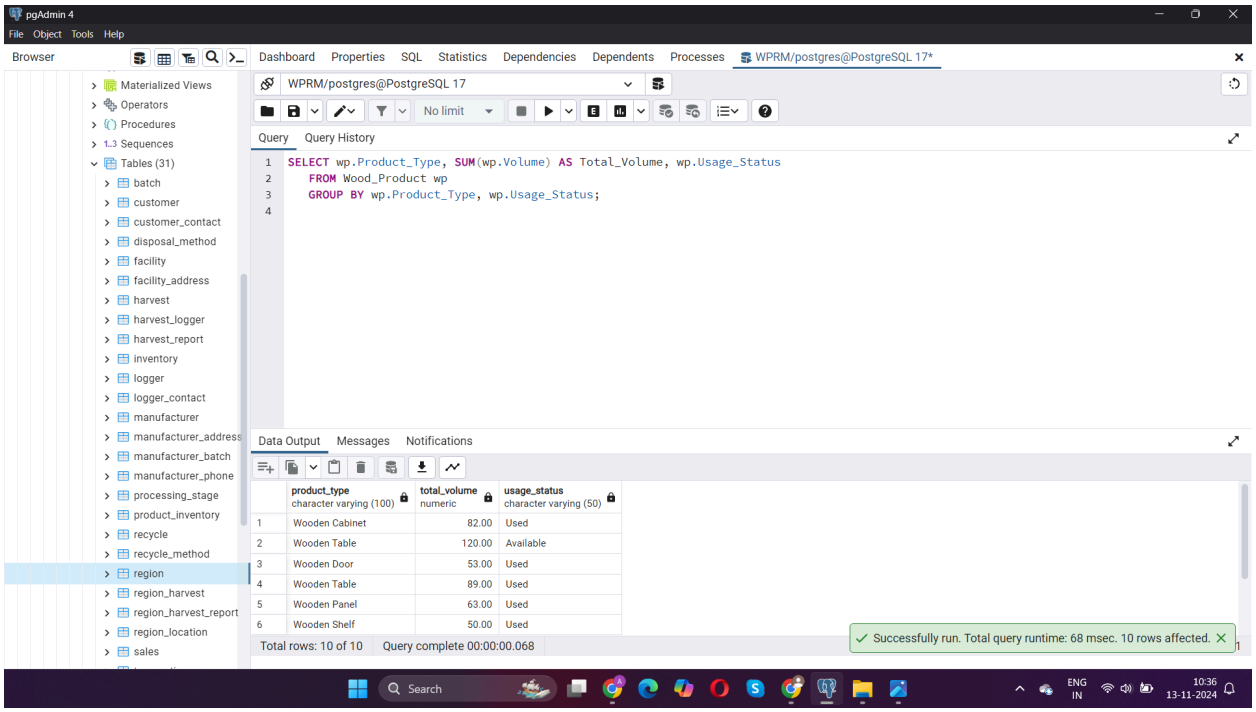
33. Find the disposal method used for the highest number of products:

```
SELECT dm.Disposal_Method_Name, COUNT(DISTINCT wd.Product_ID) AS
Product_Count
FROM Waste_Disposal wd
JOIN Disposal_Method dm ON wd.Disposal_Method_ID =
dm.Disposal_Method_ID
GROUP BY dm.Disposal_Method_Name
ORDER BY COUNT(DISTINCT wd.Product_ID) DESC
LIMIT 1;
```



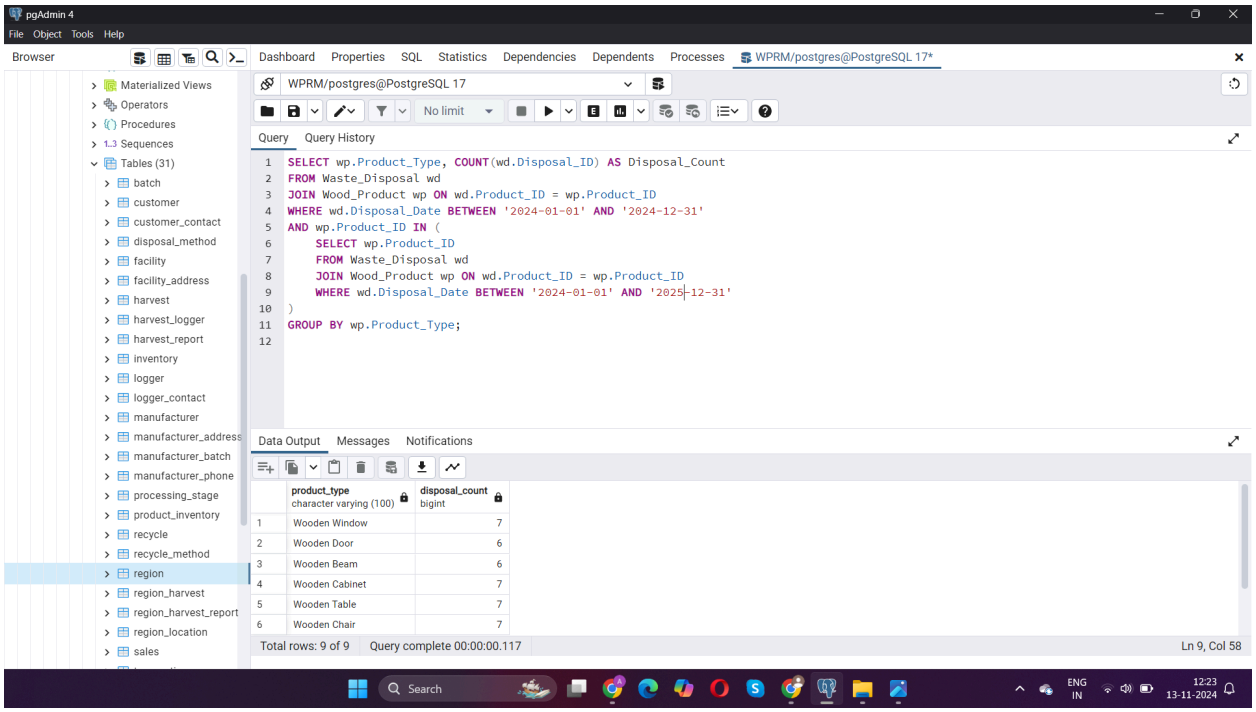
34. Get a list of products with their total volume and status:

```
SELECT wp.Product_Type, SUM(wp.Volume) AS Total_Volume,
wp.Usage_Status
  FROM Wood_Product wp
  GROUP BY wp.Product_Type, wp.Usage_Status;
```



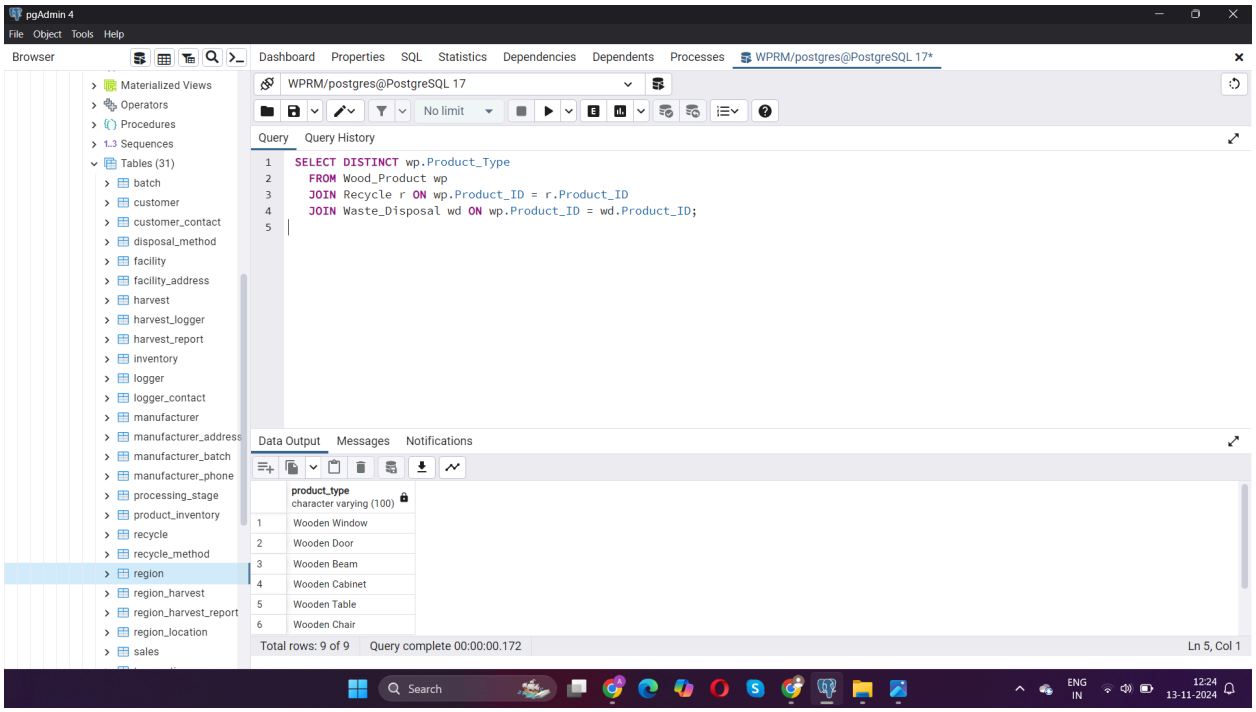
35.Retrieve products disposed of in both 2024 and 2025, with their disposal counts:

```
SELECT wp.Product_Name, COUNT(wd.Disposal_ID) AS Disposal_Count
FROM Waste_Disposal wd
JOIN Wood_Product wp ON wd.Product_ID = wp.Product_ID
WHERE wd.Disposal_Date BETWEEN '2024-01-01' AND '2024-12-31'
AND wp.Product_ID IN (
    SELECT wp.Product_ID
    FROM Waste_Disposal wd
    JOIN Wood_Product wp ON wd.Product_ID = wp.Product_ID
    WHERE wd.Disposal_Date BETWEEN '2025-01-01' AND '2025-12-31'
)
GROUP BY wp.Product_Name;
```



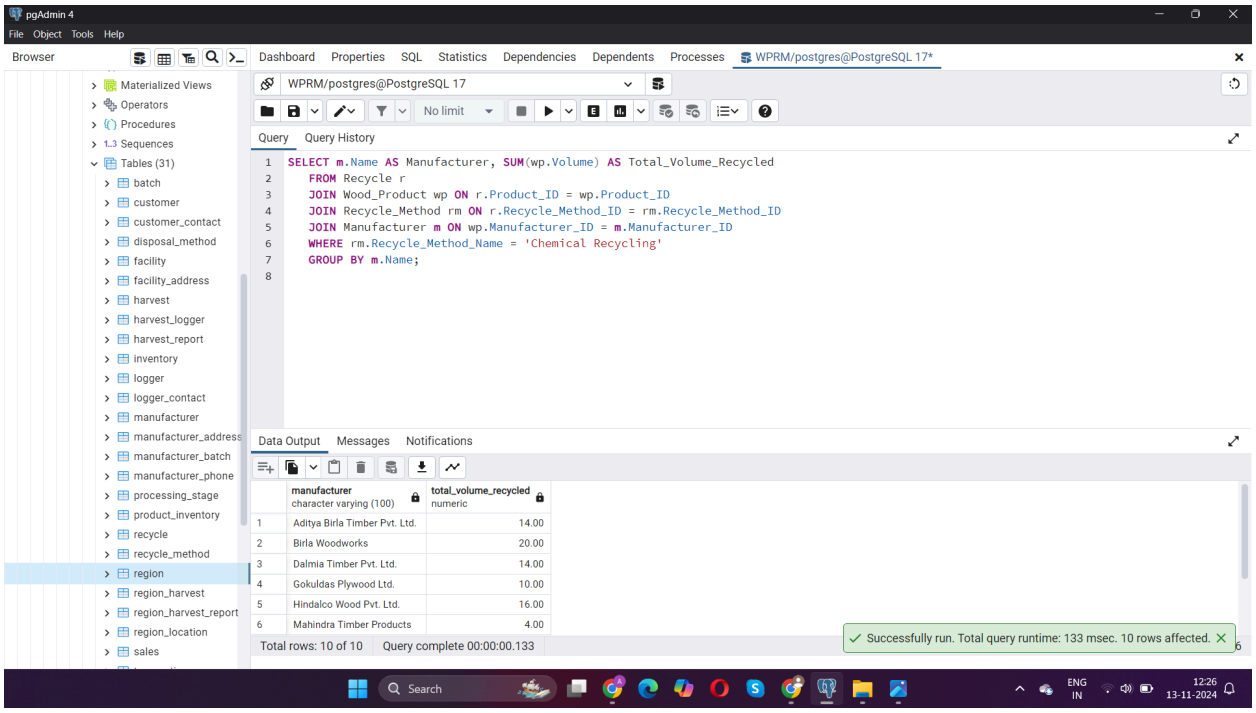
36. Get products that have been both recycled and disposed of:

```
SELECT DISTINCT wp.Product_Type
FROM Wood_Product wp
JOIN Recycle r ON wp.Product_ID = r.Product_ID
JOIN Waste_Disposal wd ON wp.Product_ID = wd.Product_ID;
```



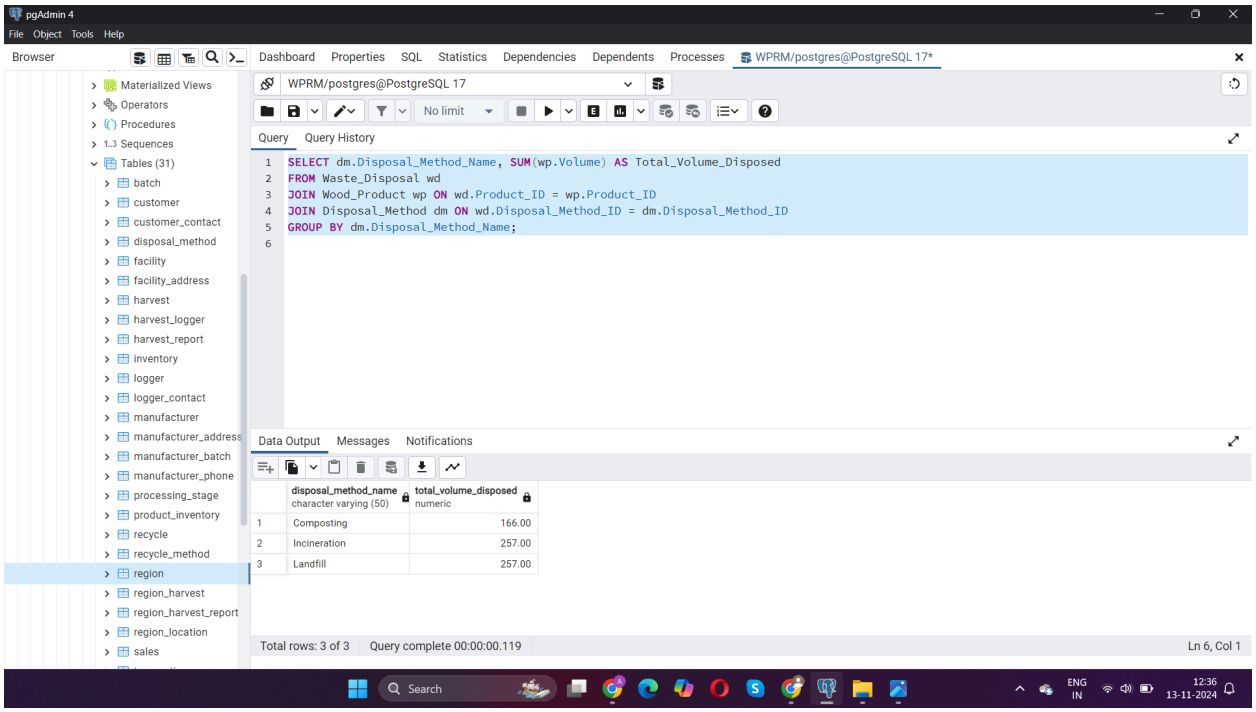
37. Get the sum of total volume by manufacturer for products recycled in a specific method:

```
SELECT m.Name AS Manufacturer, SUM(wp.Volume) AS
Total_Volume_Recycled
  FROM Recycle r
  JOIN Wood_Product wp ON r.Product_ID = wp.Product_ID
  JOIN Recycle_Method rm ON r.Recycle_Method_ID = rm.Recycle_Method_ID
  JOIN Manufacturer m ON wp.Manufacturer_ID = m.Manufacturer_ID
  WHERE rm.Recycle_Method_Name = 'Method A'
  GROUP BY m.Name;
```



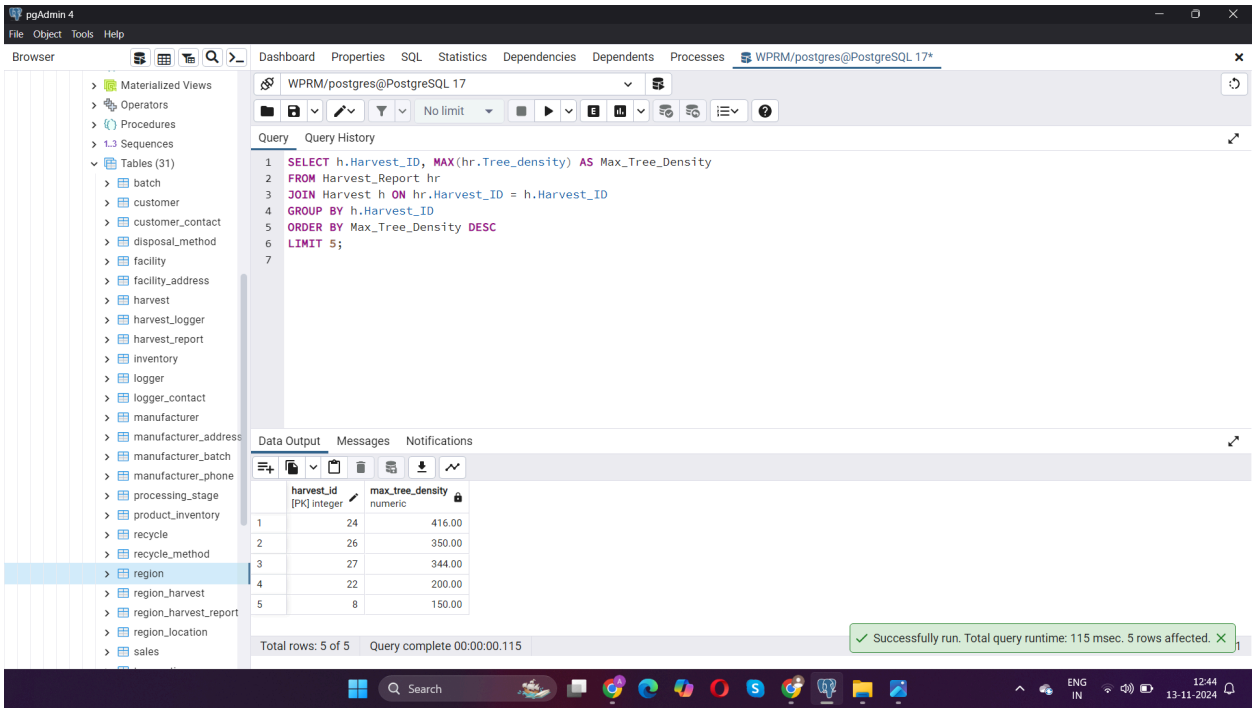
38.Retrieve the total volume disposed of by each disposal method:

```
SELECT dm.Disposal_Method_Name, SUM(wp.Volume) AS
Total_Volume_Disposed
FROM Waste_Disposal wd
JOIN Wood_Product wp ON wd.Product_ID = wp.Product_ID
JOIN Disposal_Method dm ON wd.Disposal_Method_ID =
dm.Disposal_Method_ID
GROUP BY dm.Disposal_Method_Name;
```



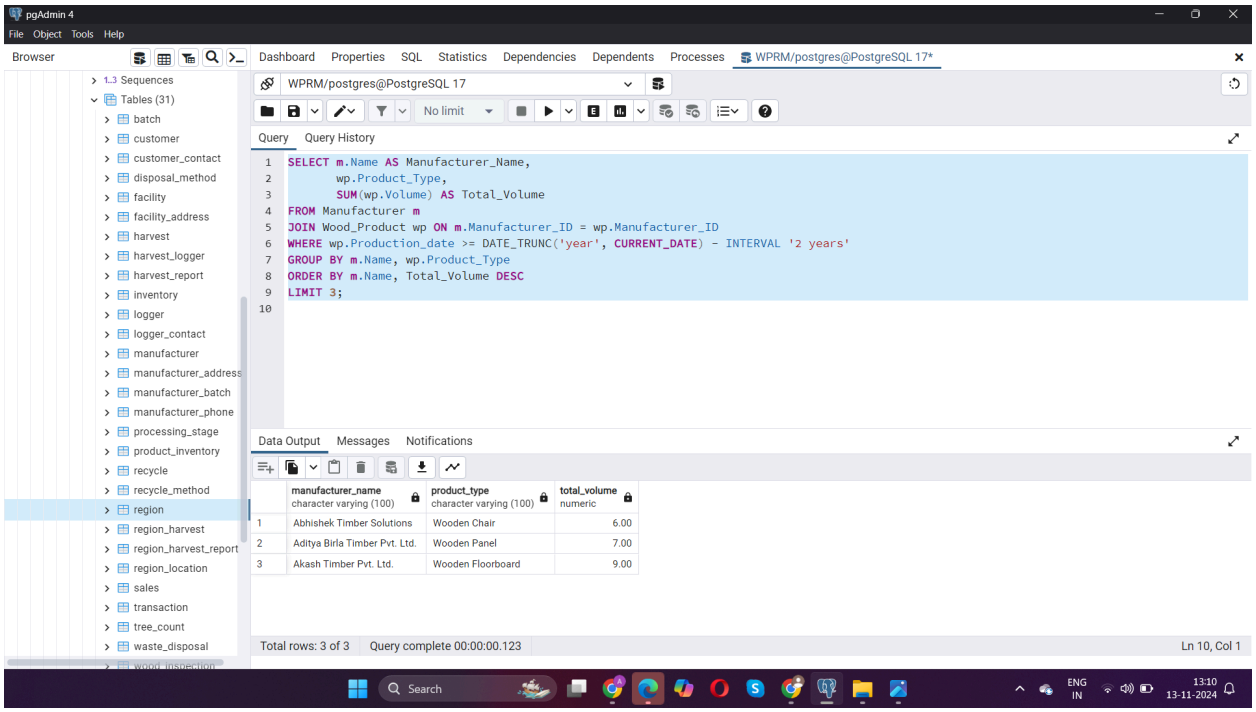
39. Retrieve the maximum tree density for each harvest and list only the top 5 highest densities:

```
SELECT h.Harvest_ID, MAX(hr.Tree_density) AS Max_Tree_Density
FROM Harvest_Report hr
JOIN Harvest h ON hr.Harvest_ID = h.Harvest_ID
GROUP BY h.Harvest_ID
ORDER BY Max_Tree_Density DESC
LIMIT 5;
```



40. Find the top 3 products (by volume) manufactured by each manufacturer over the last 2 years:

```
 SELECT m.Name AS Manufacturer_Name,
       wp.Product_Type,
       SUM(wp.Volume) AS Total_Volume
FROM Manufacturer m
JOIN Wood_Product wp ON m.Manufacturer_ID = wp.Manufacturer_ID
WHERE wp.Production_date >= DATE_TRUNC('year', CURRENT_DATE) -
INTERVAL '2 years'
GROUP BY m.Name, wp.Product_Type
ORDER BY m.Name, Total_Volume DESC
LIMIT 3;
```



Chapter - 5 Interface Implementation

❖ Setup JDBC and Basic GUI

Create a PostgreSQL Table and here we have that tables which are user and user_contact and user_Address for this particular GUI.We have done the prerequirements for this GUI like install JDK and install postgres and build path it for the project and all.Here i mention the code for the GUI.

Postgres

```
CREATE TABLE User_Address (
    Address_ID SERIAL PRIMARY KEY,
    User_ID INT REFERENCES "User"(User_ID) ON DELETE CASCADE,
    Street VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    Zip_Code VARCHAR(10)
);
```

```
CREATE TABLE User (
    User_ID SERIAL PRIMARY KEY,
    Customer_name VARCHAR(100)
);
```

```
CREATE TABLE User_Contact (
```

```
    User_ID INT REFERENCES User(User_ID) ON DELETE CASCADE,
    Contact_number VARCHAR(15),
    PRIMARY KEY (User_ID, Contact_number)
);
```

Java Code

UserGUI.java

```
package Package;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;

public class UserGUI extends JFrame {
    private JTextField userIdField, customerNameField, contactNumberField,
streetField, cityField, stateField, zipField;
    private JButton addButton, updateButton, deleteButton, loadButton;
    private JTable userTable;
    private DefaultTableModel tableModel;
    private DatabaseManager dbManager;

    public UserGUI() {
        dbManager = new DatabaseManager();
        setTitle("User Management");
        setLayout(new BorderLayout());

        // Input panel
        JPanel inputPanel = new JPanel(new GridLayout(8, 2));
```

```
inputPanel.add(new JLabel("User ID:"));
userIdField = new JTextField();
inputPanel.add(userIdField);
inputPanel.add(new JLabel("Customer Name:"));
customerNameField = new JTextField();
inputPanel.add(customerNameField);
inputPanel.add(new JLabel("Contact Number:"));
contactNumberField = new JTextField();
inputPanel.add(contactNumberField);
inputPanel.add(new JLabel("Street:"));
streetField = new JTextField();
inputPanel.add(streetField);
inputPanel.add(new JLabel("City:"));
cityField = new JTextField();
inputPanel.add(cityField);
inputPanel.add(new JLabel("State:"));
stateField = new JTextField();
inputPanel.add(stateField);
inputPanel.add(new JLabel("Zip Code:"));
zipField = new JTextField();
inputPanel.add(zipField);

// Button panel
JPanel buttonPanel = new JPanel();
addButton = new JButton("Add");
updateButton = new JButton("Update");
deleteButton = new JButton("Delete");
loadButton = new JButton("Load");

buttonPanel.add(addButton);
buttonPanel.add(updateButton);
buttonPanel.add(deleteButton);
buttonPanel.add(loadButton);

// Table for displaying users
```

```
        tableModel = new DefaultTableModel(new String[]{"User ID", "Customer
Name", "Contact Number", "Street", "City", "State", "Zip Code"}, 0);
        userTable = new JTable(tableModel);
        JScrollPane scrollPane = new JScrollPane(userTable);

        // Add components to the frame
        add(inputPanel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);
        add(buttonPanel, BorderLayout.SOUTH);

        // Action listeners for buttons
        addButton.addActionListener(e -> addUser());
        updateButton.addActionListener(e -> updateUser());
        deleteButton.addActionListener(e -> deleteUser());
        loadButton.addActionListener(e -> loadUsers());

        setSize(800, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    private void addUser() {
        int userId = Integer.parseInt(userIdField.getText());
        String customerName = customerNameField.getText();
        String contactNumber = contactNumberField.getText();
        String street = streetField.getText();
        String city = cityField.getText();
        String state = stateField.getText();
        String zip = zipField.getText();
        dbManager.insertUser(userId, customerName);
        dbManager.insertUserContact(userId, contactNumber);
        dbManager.insertUserAddress(userId, street, city, state, zip);
        loadUsers();
    }
```

```
private void updateUser() {
    int userId = Integer.parseInt(userIdField.getText());
    String customerName = customerNameField.getText();
    String contactNumber = contactNumberField.getText();
    String street = streetField.getText();
    String city = cityField.getText();
    String state = stateField.getText();
    String zip = zipField.getText();

    dbManager.updateUser(userId, customerName);
    dbManager.updateUserContact(userId, contactNumber);
    dbManager.updateUserAddress(userId, street, city, state, zip);
    loadUsers();
}

private void deleteUser() {
    int userId = Integer.parseInt(userIdField.getText());
    dbManager.deleteUser(userId);
    dbManager.deleteUserContact(userId);
    dbManager.deleteUserAddress(userId);
    loadUsers();
}

private void loadUsers() {
    tableModel.setRowCount(0);
    try {
        ResultSet userSet = dbManager.readUsers();
        while (userSet != null && userSet.next()) {
            int userId = userSet.getInt("User_ID");
            String customerName = userSet.getString("Customer_name");

            String contactNumber = dbManager.getUserContact(userId);

            ResultSet addressSet = dbManager.getUserAddress(userId);
            String street = "", city = "", state = "", zip = "";
```

```
            if (addressSet != null && addressSet.next()) {
                street = addressSet.getString("Street");
                city = addressSet.getString("City");
                state = addressSet.getString("State");
                zip = addressSet.getString("Zip_Code");
            }
            tableModel.addRow(new Object[]{userId, customerName,
contactNumber, street, city, state, zip});
        }
    } catch (SQLException e) {
        System.out.println("Error loading users: " + e.getMessage());
    }
  }

  public static void main(String[] args) {
    new UserGUI();
  }
}
```

DatabaseManager.java

```
package Package;
import java.sql.*;
public class DatabaseManager {
  private static final String URL = "jdbc:postgresql://localhost:5433/WPRM";
  private static final String USER = "postgres";
  private static final String PASSWORD = "Ayush12";
  public Connection connect() throws SQLException {
    return DriverManager.getConnection(URL, USER, PASSWORD);
  }
  public void insertUser(int userId, String customerName) {
    String insertSQL = "INSERT INTO \"User\" (User_ID, Customer_name)
VALUES (?, ?)";
    try (Connection connection = connect();
```

```
            PreparedStatement pstmt = connection.prepareStatement(insertSQL)) {
            pstmt.setInt(1, userId);
            pstmt.setString(2, customerName);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error inserting user: " + e.getMessage());
        }
    }
    public ResultSet readUsers() {
        String selectSQL = "SELECT * FROM \"User\"";
        try {
            Connection connection = connect();
            PreparedStatement pstmt = connection.prepareStatement(selectSQL);
            return pstmt.executeQuery();
        } catch (SQLException e) {
            System.out.println("Error reading users: " + e.getMessage());
        }
        return null;
    }
    public void updateUser(int userId, String customerName) {
        String updateSQL = "UPDATE \"User\" SET Customer_name = ? WHERE
User_ID = ?";
        try (Connection connection = connect();
             PreparedStatement pstmt = connection.prepareStatement(updateSQL)) {
            pstmt.setString(1, customerName);
            pstmt.setInt(2, userId);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error updating user: " + e.getMessage());
        }
    }
    public void deleteUser(int userId) {
        String deleteSQL = "DELETE FROM \"User\" WHERE User_ID = ?";
        try (Connection connection = connect();
             PreparedStatement pstmt = connection.prepareStatement(deleteSQL)) {
```

```
            pstmt.setInt(1, userId);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error deleting user: " + e.getMessage());
        }
    }
    public void insertUserContact(int userId, String contactNumber) {
        String insertSQL = "INSERT INTO user_contact (User_ID, Contact_number)
VALUES (?, ?)";
        try (Connection connection = connect();
             PreparedStatement pstmt = connection.prepareStatement(insertSQL)) {
            pstmt.setInt(1, userId);
            pstmt.setString(2, contactNumber);
            pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error inserting contact: " + e.getMessage());
        }
    }
    public String getUserContact(int userId) {
        String selectSQL = "SELECT Contact_number FROM user_contact WHERE
User_ID = ?";
        try (Connection connection = connect();
             PreparedStatement pstmt = connection.prepareStatement(selectSQL)) {
            pstmt.setInt(1, userId);
            ResultSet rs = pstmt.executeQuery();
            if (rs.next()) {
                return rs.getString("Contact_number");
            }
        } catch (SQLException e) {
            System.out.println("Error fetching contact: " + e.getMessage());
        }
        return null;
    }
    public void updateUserContact(int userId, String contactNumber) {
```

```
    String updateSQL = "UPDATE user_contact SET Contact_number = ?
WHERE User_ID = ?";
    try (Connection connection = connect();
         PreparedStatement pstmt = connection.prepareStatement(updateSQL)) {
        pstmt.setString(1, contactNumber);
        pstmt.setInt(2, userId);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Error updating contact: " + e.getMessage());
    }
  }
  public void deleteUserContact(int userId) {
    String deleteSQL = "DELETE FROM user_contact WHERE User_ID = ?";
    try (Connection connection = connect();
         PreparedStatement pstmt = connection.prepareStatement(deleteSQL)) {
        pstmt.setInt(1, userId);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Error deleting contact: " + e.getMessage());
    }
  }
  public void insertUserAddress(int userId, String street, String city, String state,
String zipCode) {
    String insertSQL = "INSERT INTO user_address (User_ID, Street, City, State,
Zip_Code) VALUES (?, ?, ?, ?, ?)";
    try (Connection connection = connect();
         PreparedStatement pstmt = connection.prepareStatement(insertSQL)) {
        pstmt.setInt(1, userId);
        pstmt.setString(2, street);
        pstmt.setString(3, city);
        pstmt.setString(4, state);
        pstmt.setString(5, zipCode);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Error inserting address: " + e.getMessage());
```

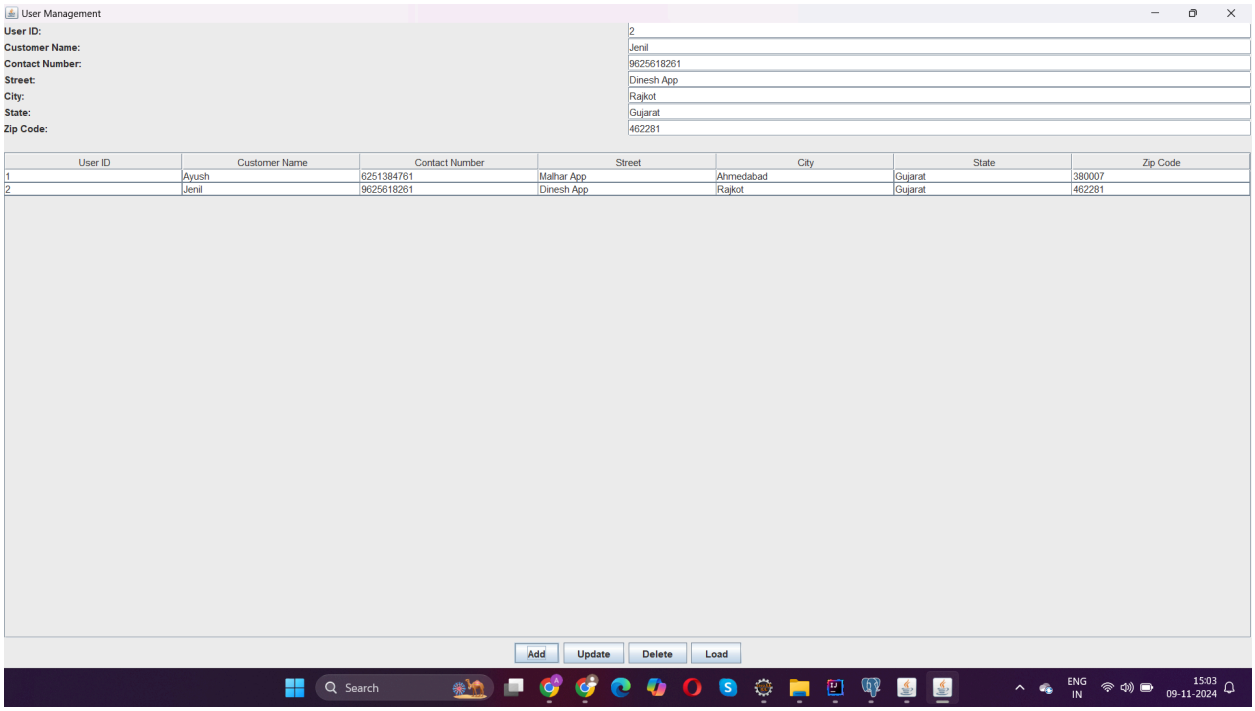
```
    }
  }
  public ResultSet getUserAddress(int userId) {
    String selectSQL = "SELECT * FROM user_address WHERE User_ID = ?";
    try {
      Connection connection = connect();
      PreparedStatement pstmt = connection.prepareStatement(selectSQL);
      pstmt.setInt(1, userId);
      return pstmt.executeQuery();
    } catch (SQLException e) {
      System.out.println("Error fetching address: " + e.getMessage());
    }
    return null;
  }
  public void updateUserAddress(int userId, String street, String city, String state,
String zipCode) {
    String updateSQL = "UPDATE user_address SET Street = ?, City = ?, State =
?, Zip_Code = ? WHERE User_ID = ?";
    try (Connection connection = connect();
       PreparedStatement pstmt = connection.prepareStatement(updateSQL)) {
      pstmt.setString(1, street);
      pstmt.setString(2, city);
      pstmt.setString(3, state);
      pstmt.setString(4, zipCode);
      pstmt.setInt(5, userId);
      pstmt.executeUpdate();
    } catch (SQLException e) {
      System.out.println("Error updating address: " + e.getMessage());
    }
  }
  public void deleteUserAddress(int userId) {
    String deleteSQL = "DELETE FROM user_address WHERE User_ID = ?";
    try (Connection connection = connect();
       PreparedStatement pstmt = connection.prepareStatement(deleteSQL)) {
      pstmt.setInt(1, userId);
```

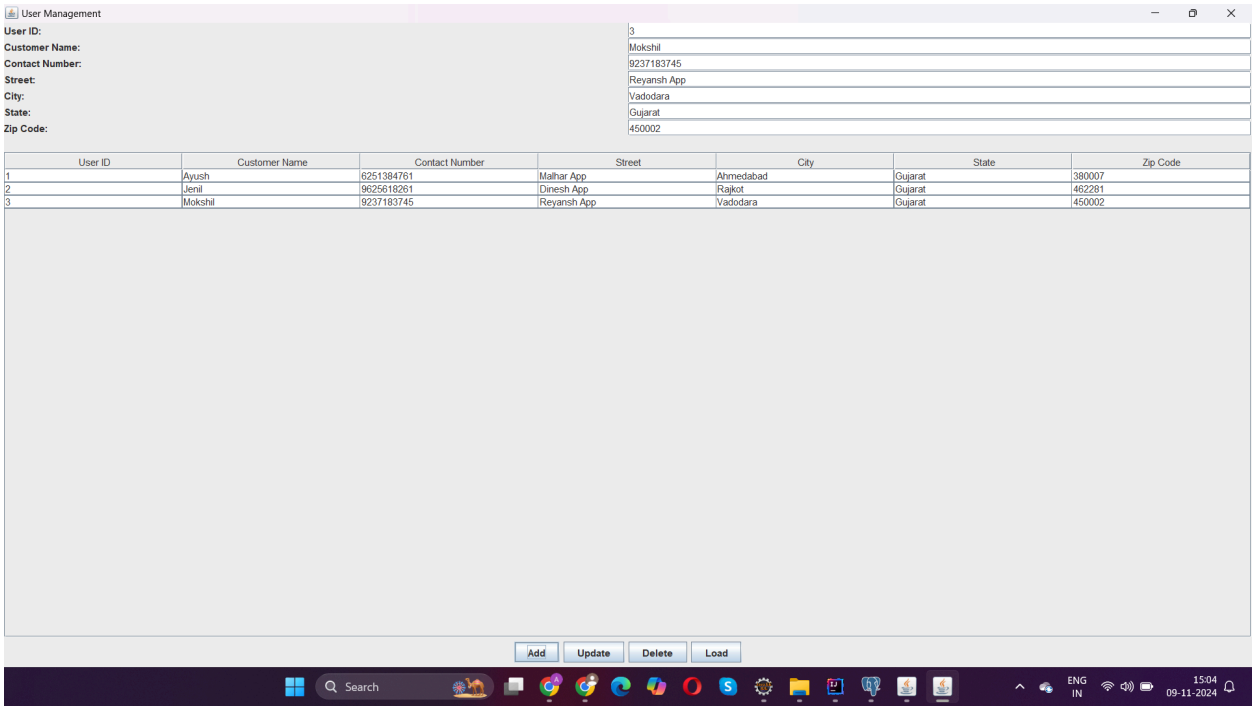
```
            pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error deleting address: " + e.getMessage());
        }
    }
}
```

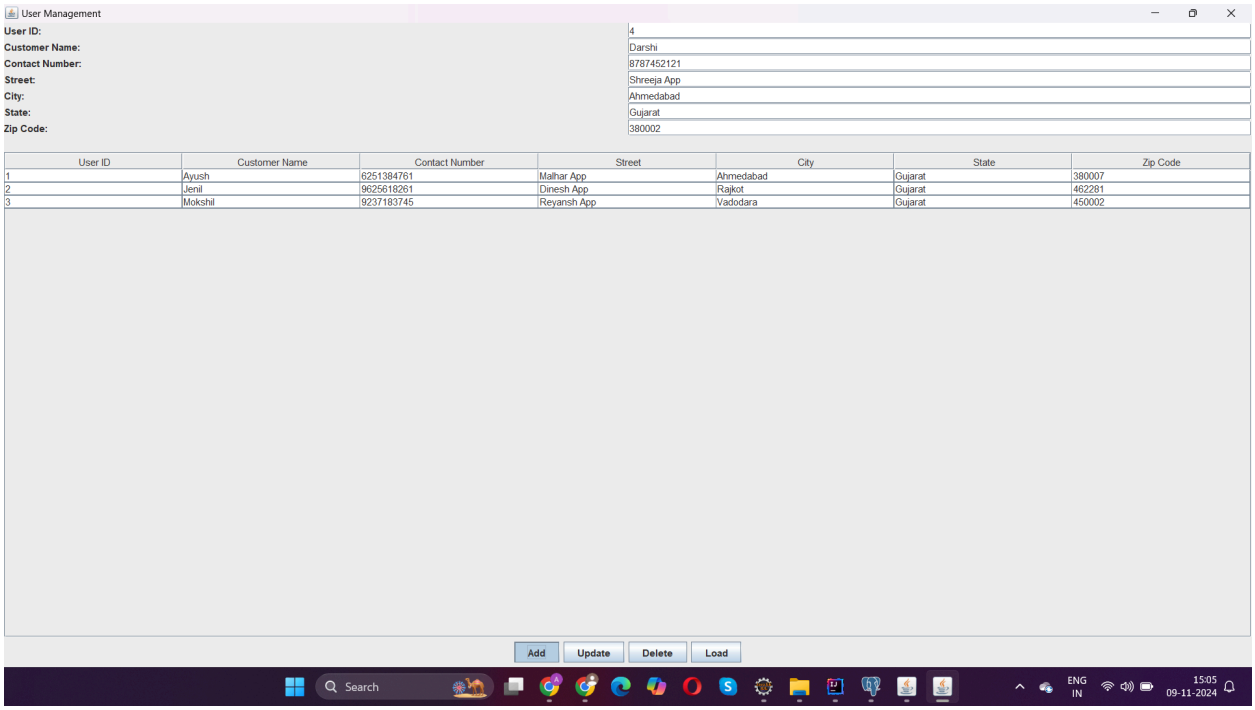
❖ CRUD Operations in GUI

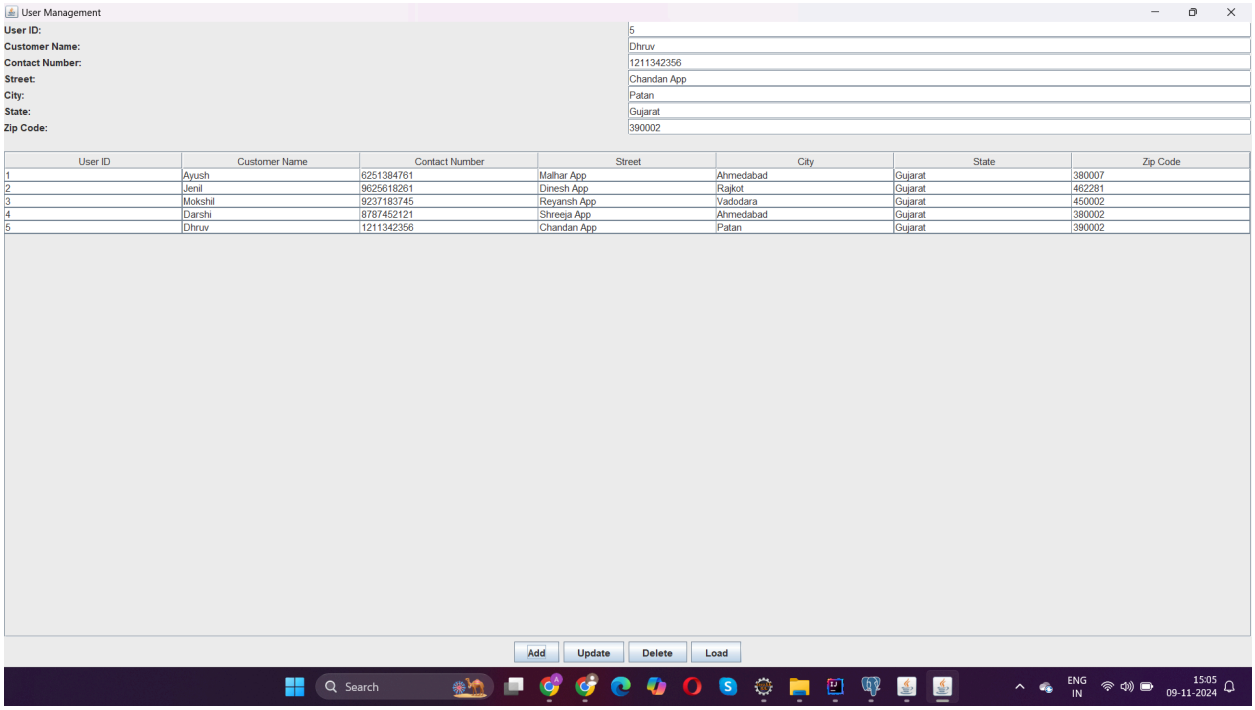
Above we have discussed about the code and all thing which can help us to make the GUI like Code and all so here we added the snapshots of our GUI so that you can easily understand that how the code works.

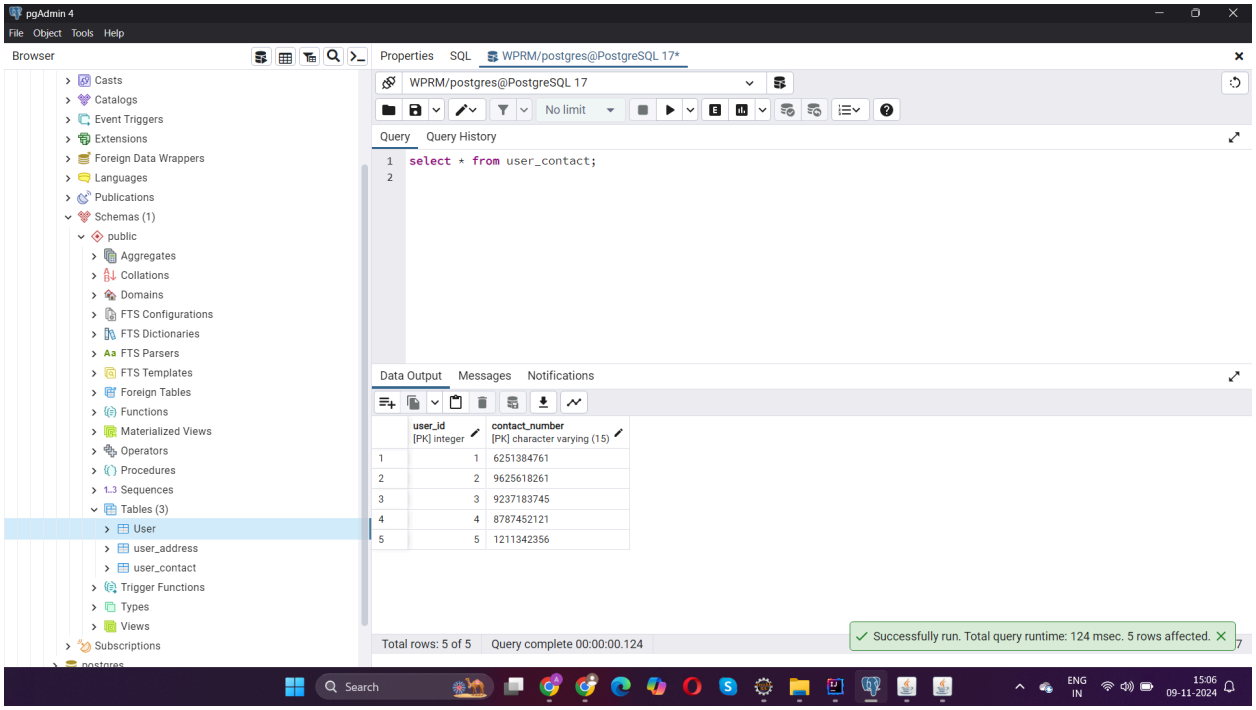
1) Insert entries in GUI

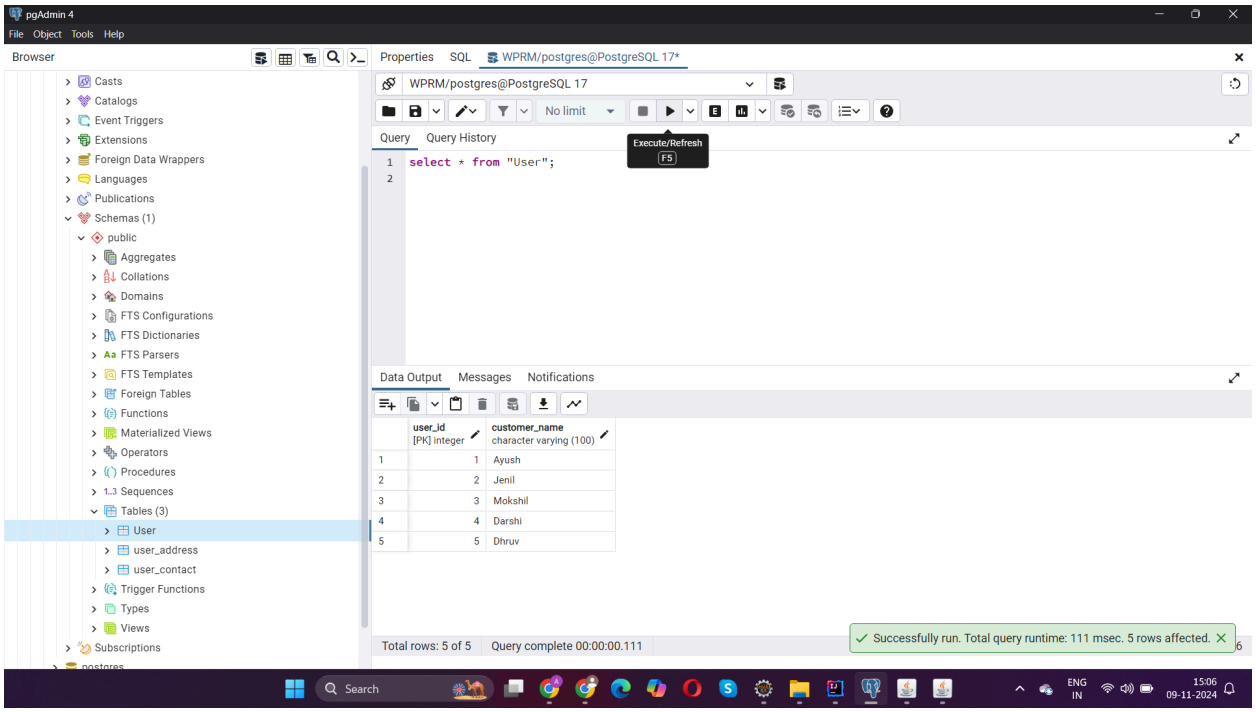


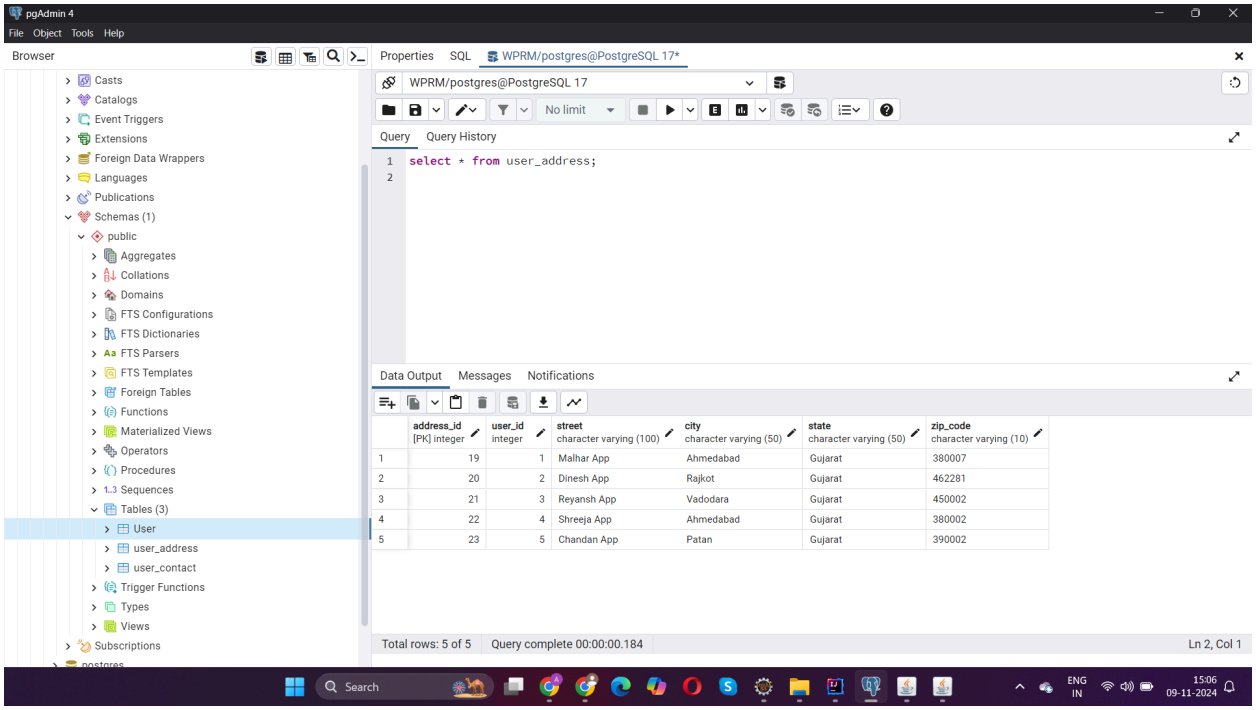

User Management
User ID:
3
Customer Name:
Mokshil
Contact Number:
9237183745
Street:
Reyansh App
City:
Vadodara
State:
Gujarat
Zip Code:
450002
User ID
Customer Name
Contact Number
Street
City
State
Zip Code
1
Ayush
6251384761
Malhar App
Ahmedabad
Gujarat
380007
2
Jenil
9625618261
Dinesh App
Rajkot
Gujarat
462281
3
Mokshil
9237183745
Reyansh App
Vadodara
Gujarat
450002
Add
Update
Delete
Load
Search
ENG
IN
15:04
09-11-2024


User Management
User ID:
4
Customer Name:
Darshi
Contact Number:
8787452121
Street:
Shreeja App
City:
Ahmedabad
State:
Gujarat
Zip Code:
380002
User ID | Customer Name | Contact Number | Street | City | State | Zip Code
1 | Ayush | 6251384761 | Malhar App | Ahmedabad | Gujarat | 380007
2 | Jenil | 9625618261 | Dinesh App | Rajkot | Gujarat | 462281
3 | Mokshil | 9237183745 | Reyansh App | Vadodara | Gujarat | 450002
Add
Update
Delete
Load
Search
ENG
IN
15:05
09-11-2024

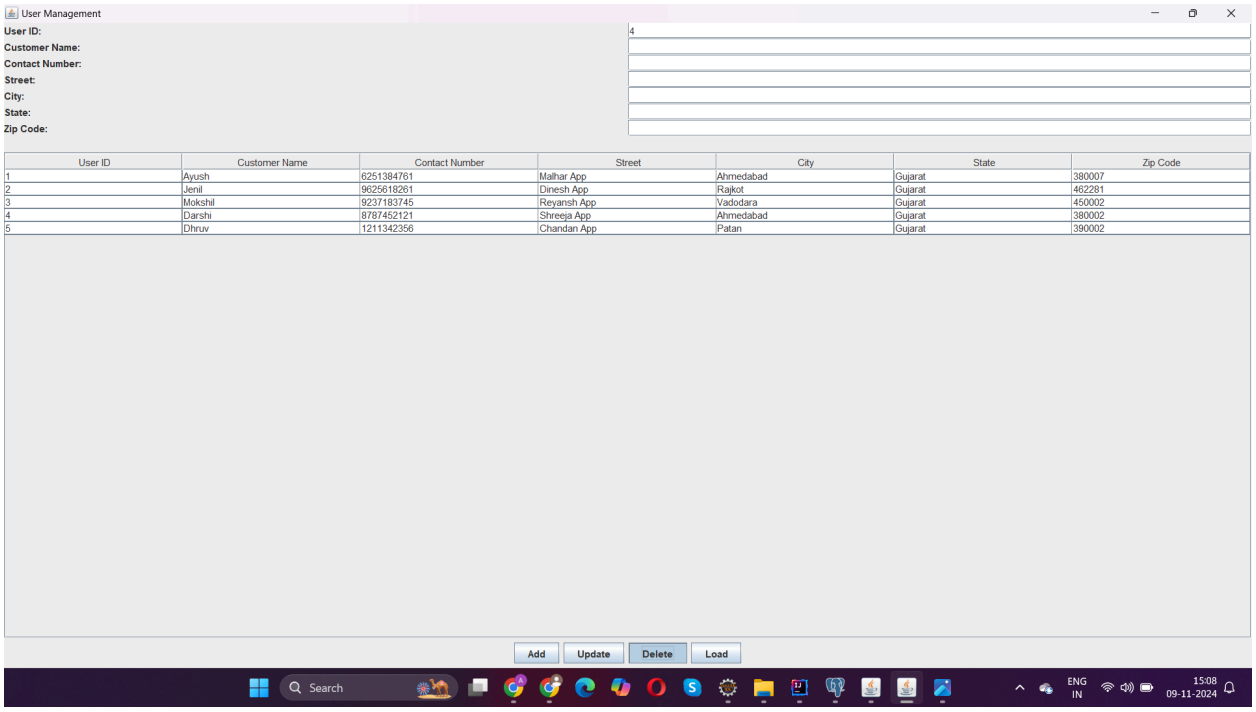

User Management
User ID:
5
Customer Name:
Dhruv
Contact Number:
1211342356
Street:
Chandan App
City:
Patan
State:
Gujarat
Zip Code:
390002
User ID | Customer Name | Contact Number | Street | City | State | Zip Code
1 | Ayush | 6251384761 | Malhar App | Ahmedabad | Gujarat | 380007
2 | Jenil | 9625618261 | Dinesh App | Rajkot | Gujarat | 462281
3 | Mokshil | 9237183745 | Reyansh App | Vadodara | Gujarat | 450002
4 | Darshi | 8787452121 | Shreeja App | Ahmedabad | Gujarat | 380002
5 | Dhruv | 1211342356 | Chandan App | Patan | Gujarat | 390002
Add
Update
Delete
Load
Search
ENG
IN
15:05
09-11-2024

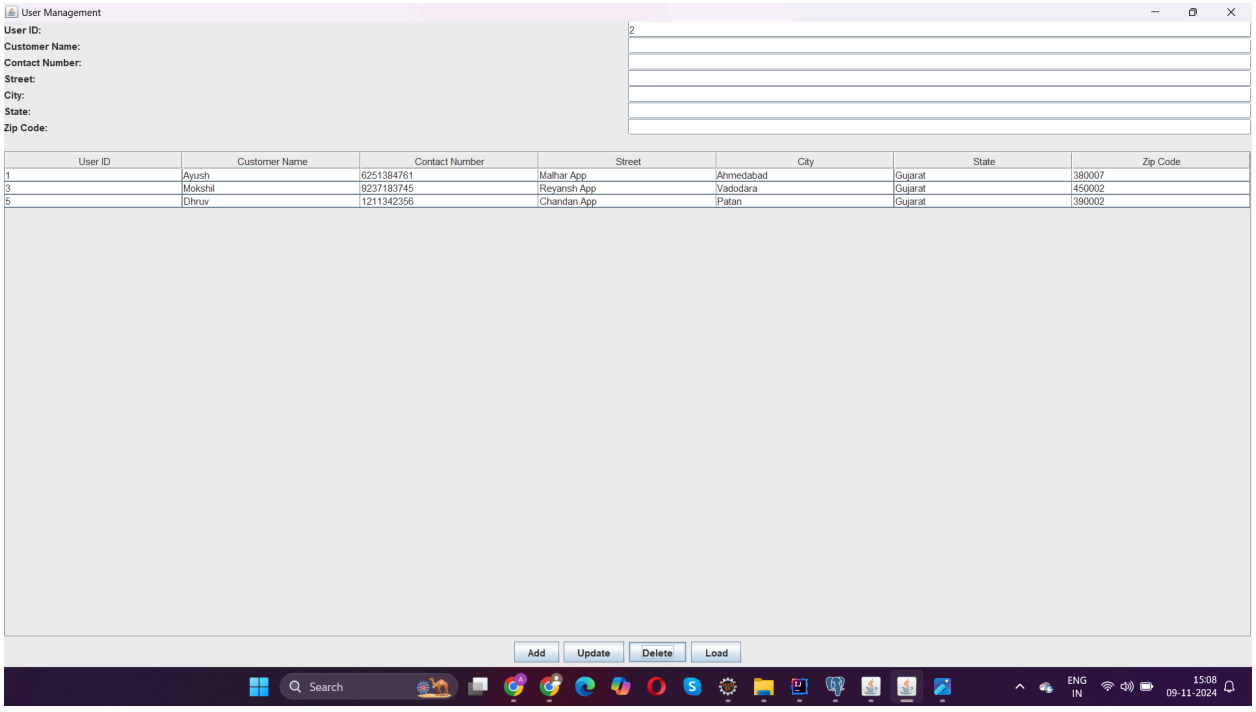

pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
select * from user_contact;
Data Output Messages Notifications
user_id [PK] integer
contact_number [PK] character varying (15)
1 1 6251384761
2 2 9625618261
3 3 9237183745
4 4 8787452121
5 5 1211342356
Total rows: 5 of 5
Query complete 00:00:00.124
Successfully run. Total query runtime: 124 msec. 5 rows affected.
Search
ENG IN
15:06 09-11-2024

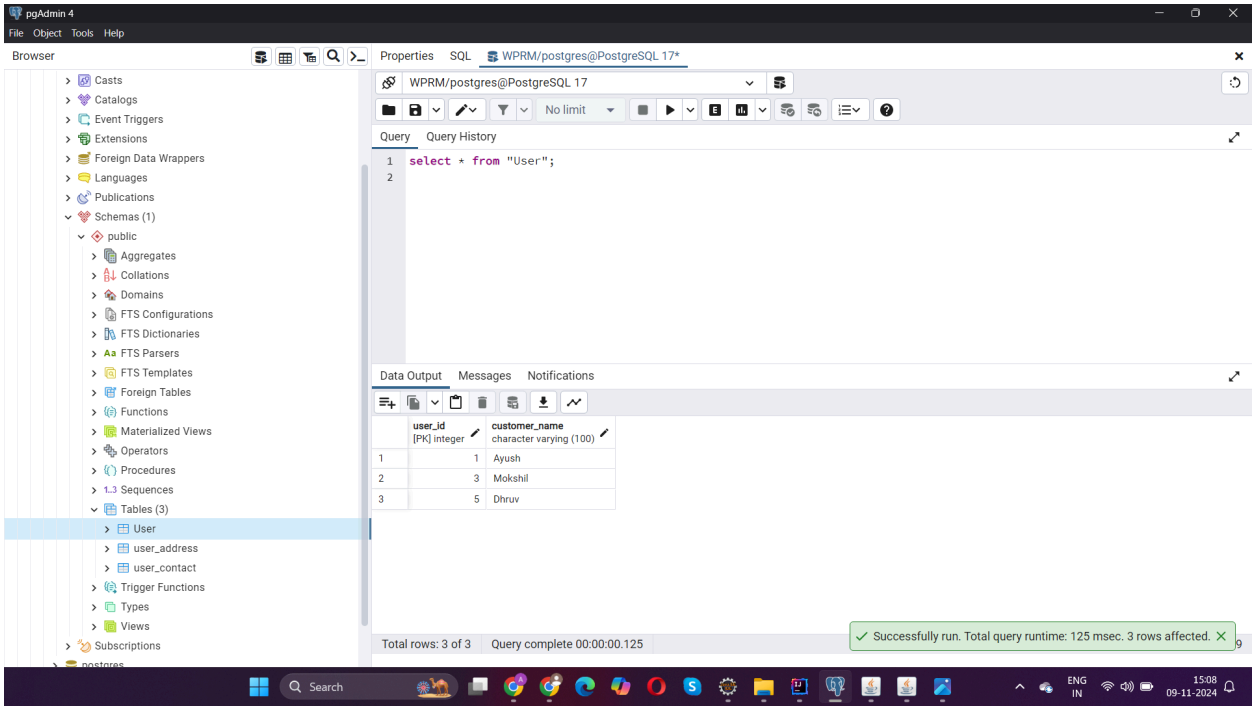

pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Execute/Refresh
F5
Query Query History
select * from "User";
Data Output Messages Notifications
user_id [PK] integer
customer_name character varying (100)
1 1 Ayush
2 2 Jenil
3 3 Mokshil
4 4 Darshi
5 5 Dhruv
Total rows: 5 of 5
Query complete 00:00:00.111
Successfully run. Total query runtime: 111 msec. 5 rows affected.
Search
ENG IN
15:06 09-11-2024

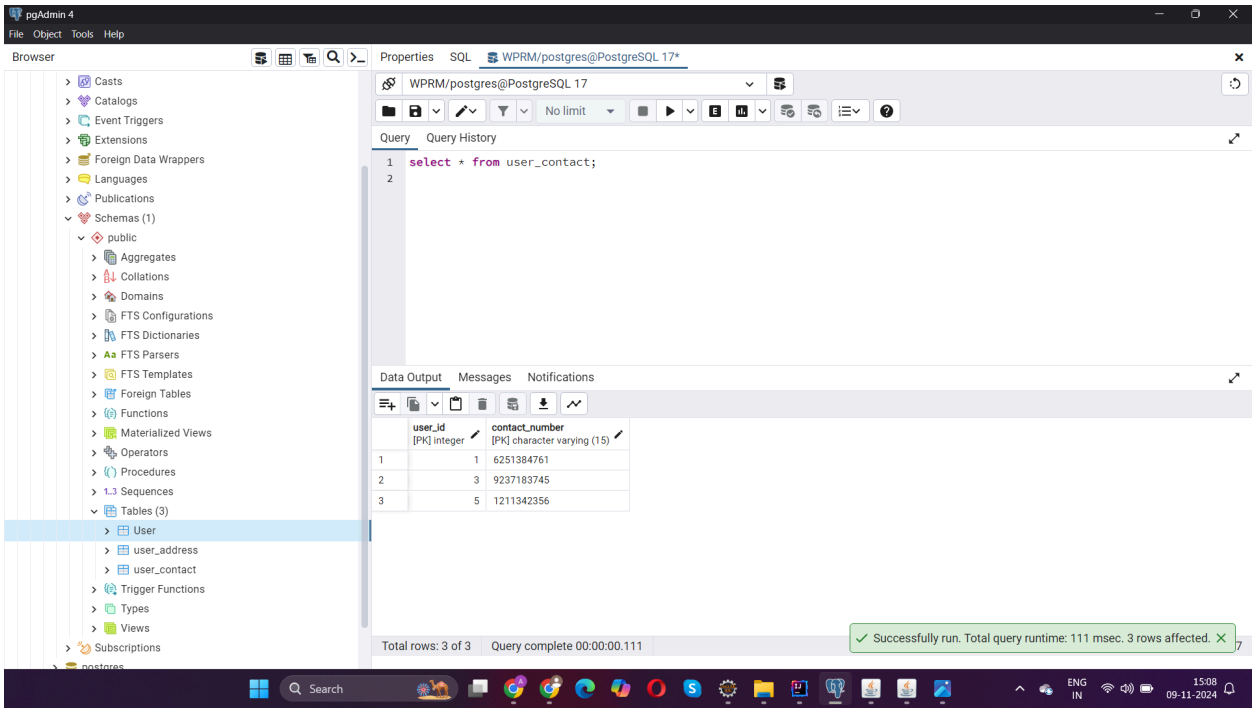

pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
1 select * from user_address;
2
Data Output Messages Notifications
address_id [PK] integer
user_id integer
street character varying (100)
city character varying (50)
state character varying (50)
zip_code character varying (10)
1 19 1 Malhar App Ahmedabad Gujarat 380007
2 20 2 Dinesh App Rajkot Gujarat 462281
3 21 3 Reyansh App Vadodara Gujarat 450002
4 22 4 Shreeja App Ahmedabad Gujarat 380002
5 23 5 Chandan App Patan Gujarat 390002
Total rows: 5 of 5
Query complete 00:00:00.184
Ln 2, Col 1
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (3)
User
user_address
user_contact
Trigger Functions
Types
Views
Subscriptions
Search
ENG IN
15:06 09-11-2024

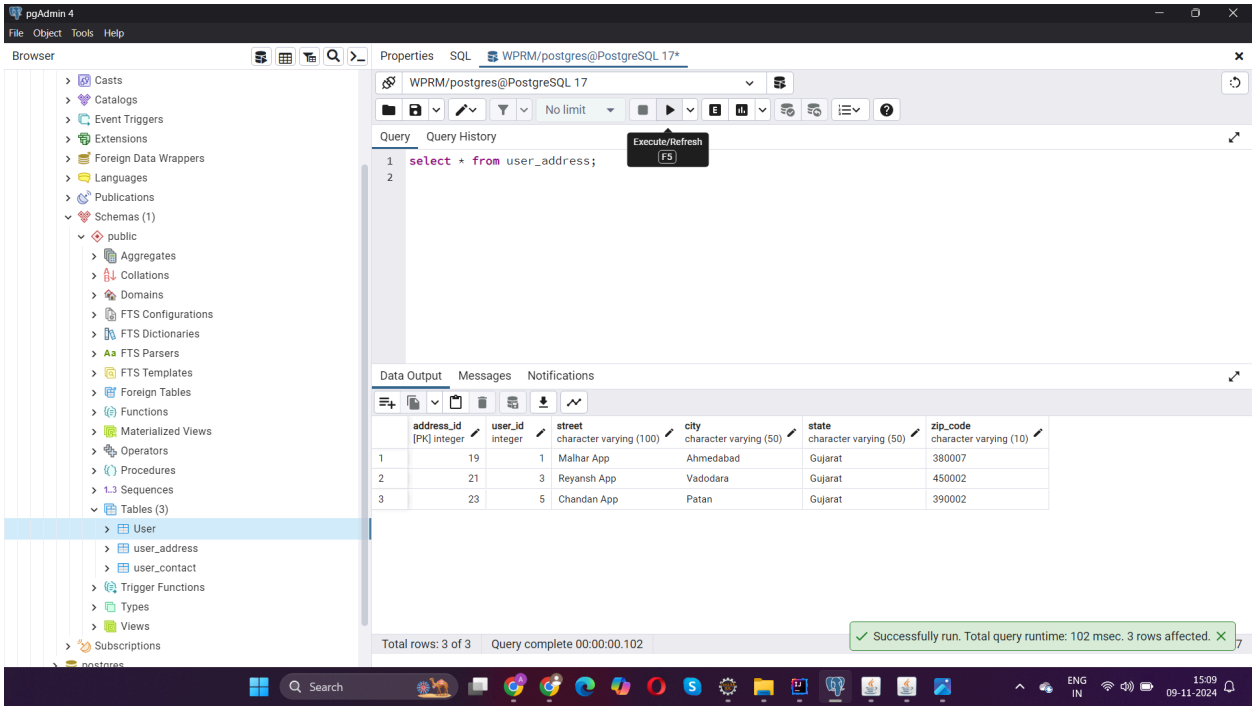
2) Delete entries in GUI



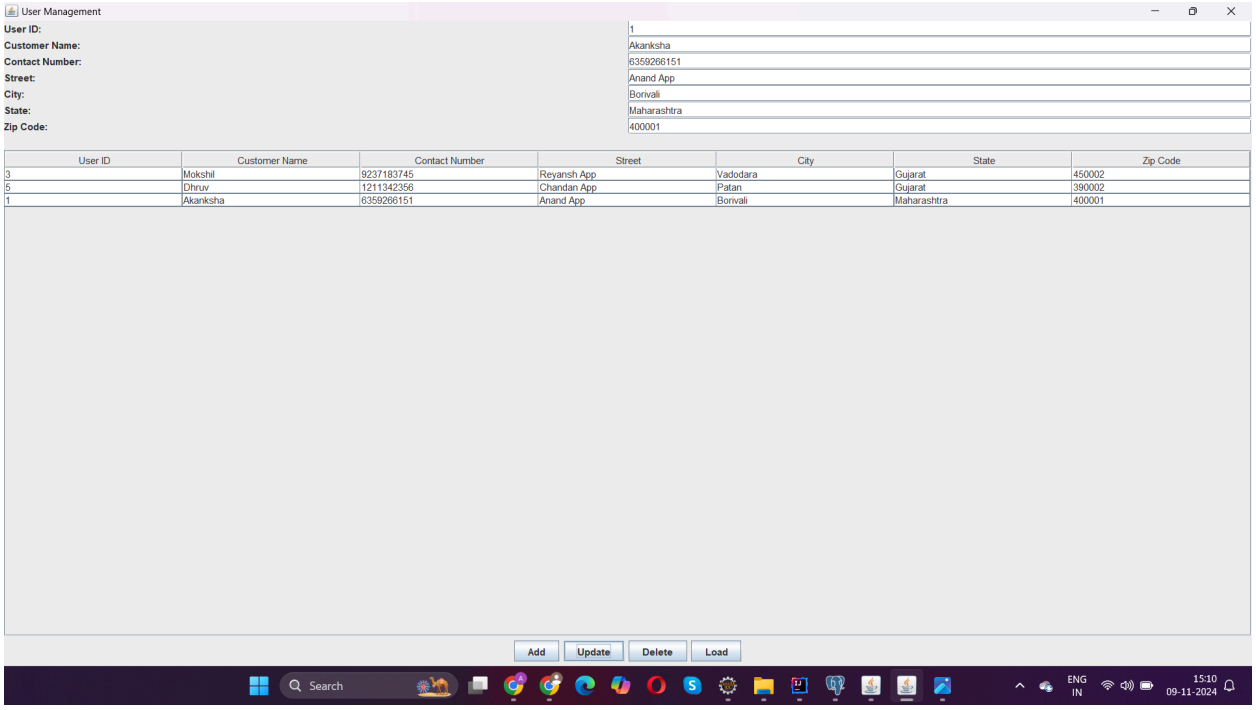


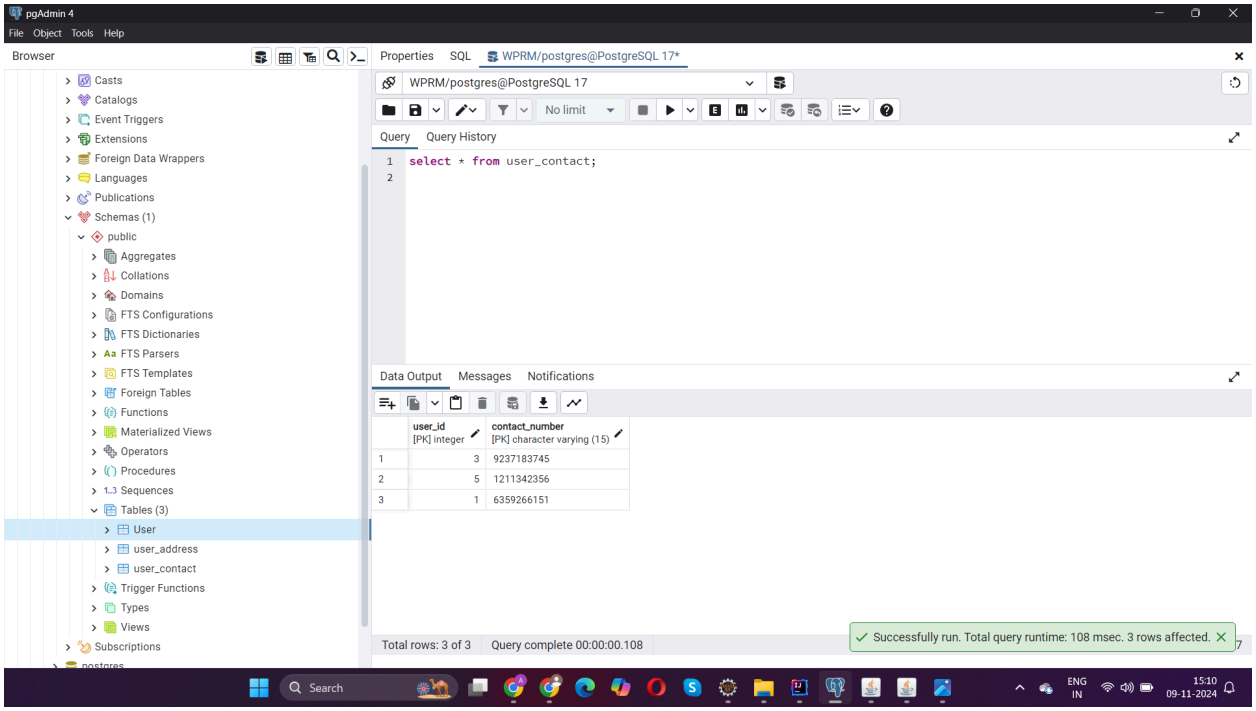

pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
select * from "User";
Data Output Messages Notifications
user_id [PK] integer
customer_name character varying (100)
1 1 Ayush
2 3 Mokshil
3 5 Dhruv
Total rows: 3 of 3
Query complete 00:00:00.125
Successfully run. Total query runtime: 125 msec. 3 rows affected.
Search
ENG IN
15:08 09-11-2024

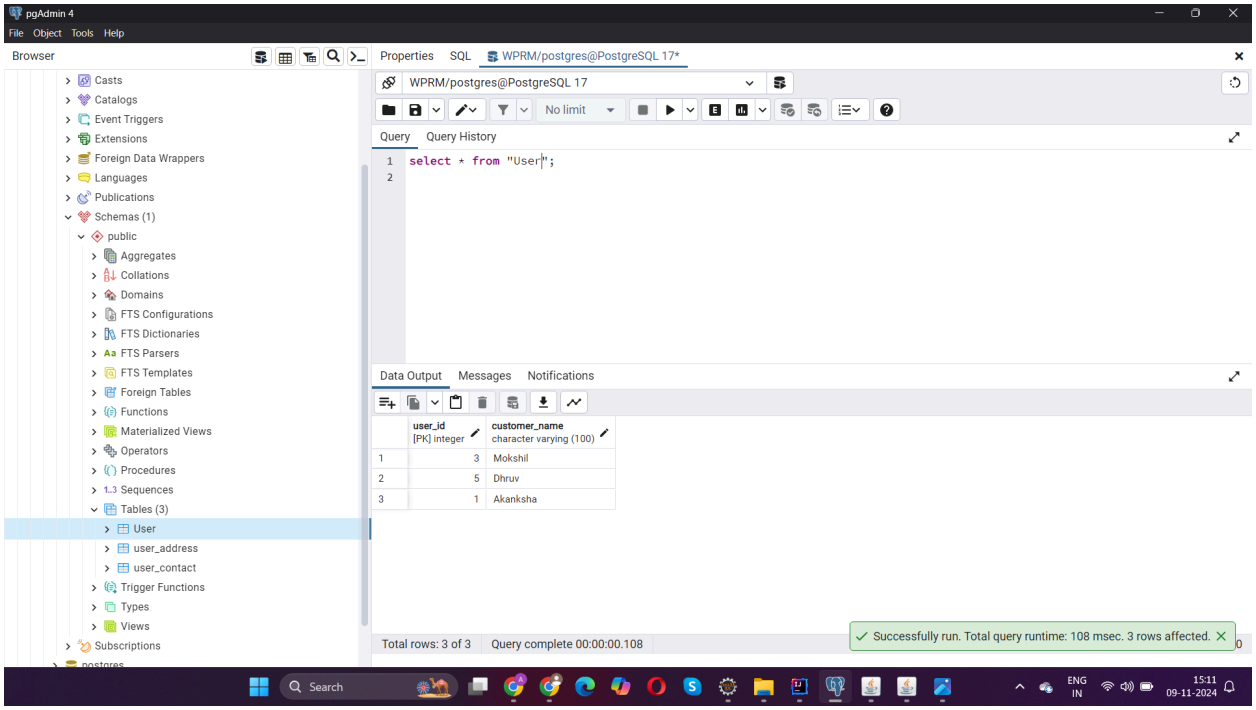

pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
select * from user_contact;
Data Output Messages Notifications
user_id [PK] integer
contact_number [PK] character varying (15)
1 1 6251384761
2 3 9237183745
3 5 1211342356
Total rows: 3 of 3
Query complete 00:00:00.111
Successfully run. Total query runtime: 111 msec. 3 rows affected.
Search
ENG IN
15:08 09-11-2024

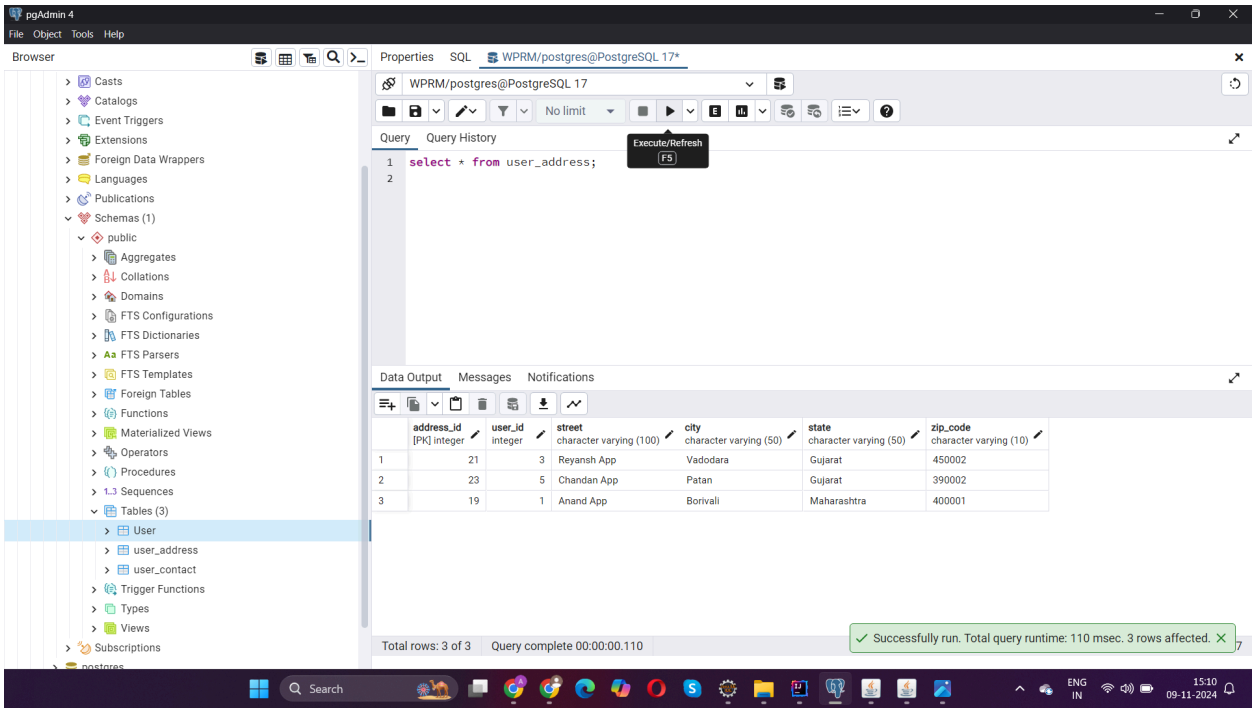

pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Execute/Refresh
F5
Query Query History
1 select * from user_address;
2
Data Output Messages Notifications
address_id [PK] integer
user_id integer
street character varying (100)
city character varying (50)
state character varying (50)
zip_code character varying (10)
1 19 1 Malhar App Ahmedabad Gujarat 380007
2 21 3 Reyansh App Vadodara Gujarat 450002
3 23 5 Chandan App Patan Gujarat 390002
Total rows: 3 of 3
Query complete 00:00:00.102
Successfully run. Total query runtime: 102 msec. 3 rows affected.
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (3)
User
user_address
user_contact
Trigger Functions
Types
Views
Subscriptions
Search
ENG
IN
15:09
09-11-2024

3) Update entries in GUI






pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
1 select * from "User";
2
Data Output Messages Notifications
user_id [PK] integer
customer_name character varying (100)
1 3 Mokshil
2 5 Dhruv
3 1 Akanksha
Total rows: 3 of 3
Query complete 00:00:00.108
Successfully run. Total query runtime: 108 msec. 3 rows affected.
Search
ENG IN
15:11 09-11-2024


pgAdmin 4
File Object Tools Help
Browser
Properties SQL WPRM/postgres@PostgreSQL 17*
WPRM/postgres@PostgreSQL 17
No limit
Query Query History
Execute/Refresh
F5
1 select * from user_address;
2
Data Output Messages Notifications
address_id [PK] integer
user_id integer
street character varying (100)
city character varying (50)
state character varying (50)
zip_code character varying (10)
1 21 3 Reyansh App Vadodara Gujarat 450002
2 23 5 Chandan App Patan Gujarat 390002
3 19 1 Anand App Borivali Maharashtra 400001
Total rows: 3 of 3
Query complete 00:00:00.110
Successfully run. Total query runtime: 110 msec. 3 rows affected.
Search
ENG IN
15:10 09-11-2024

Chapter-6 Technical Issue and Solution

1 Difficulty in Calculating and Maintaining Tree Count for a Region

- **Description**: In the system, the **Tree Count** in each **Region** is essential to assess the sustainability of the harvest and ensure proper tree density. Calculating the number of trees manually for each region posed significant challenges. For each tree to be tracked with an ID, it would require constant updates and maintenance to ensure data accuracy and consistency. This manual approach becomes more prone to human error and is not scalable, especially as the system grows to handle large regions with many trees.
- **Impact**: The challenge with manual tree counting and record-keeping is that it would lead to inefficiencies and errors, affecting data reliability. This could cause incorrect assessments of tree density, which is vital for sustainability reports and accurate harvesting practices. The project would face difficulties when scaling up, making the manual approach impractical as more regions and trees are added over time.
- **Challenge**: Ensuring accurate, consistent, and easily updatable tree counts across regions without manual intervention was a key difficulty.

Solution

Approach Taken to Resolve the Issue

- **Solution**: To resolve the issue, the **Tree Count** was treated as a separate entity within the system, recording the number of trees in a specific region. This approach allowed for a structured and automated way of tracking tree counts without manually updating each tree's record. The **Tree Count** entity was linked to the **Region** entity, and its data was updated regularly through system processes, allowing the **Region** to always reflect the current tree count and its density.
    - The **Tree Count** table is updated dynamically to reflect changes in tree numbers, whether due to harvesting, replanting, or other processes. This automated update reduces the risk of human error and ensures that tree data is accurate and consistent.
    - By separating the tree count into its own entity, the system also allows for historical tracking and reporting, as the tree count can be updated over time with timestamps.
- **Tools/Techniques Used**:
    - Relational database management system (RDBMS) to maintain the **Tree Count** entity as a part of the overall schema.
    - Triggers or batch processes to automatically update tree counts whenever a related change occurs in harvesting or planting.

Alternative Solutions Considered

- **Manual Tree Counting**: One alternative considered was manually counting the trees in a region and updating them manually within the system. This approach, while feasible for small-scale operations, would be increasingly difficult to maintain as the system scales. The complexity of counting and updating tree IDs for large regions with numerous trees would become unsustainable and prone to errors.
- **Reason for Choosing the Automated Tree Count Solution**: The manual counting approach was rejected due to its limitations in scalability, accuracy, and efficiency. The automated approach with a dedicated **Tree Count** entity

allows the system to scale, ensures consistent data updates, and reduces the chances of errors in tree counting as the system grows.

2 Difficulty in Processing and Tracking Wood Harvested Individually

- **Description**: The challenge in processing wood harvested one by one was that it became difficult to maintain accurate records for both processed and unprocessed wood. Monitoring each individual tree throughout the processing stages, from harvesting to final product creation, was an inefficient and labor-intensive process. Without grouping the wood, it would be hard to track which trees were processed, and the system would lack the ability to properly manage large volumes of harvested wood.
- **Impact**: The impact was that the process lacked organization and efficiency. It would be difficult to track the wood's journey through processing, including identifying which wood had been processed and which remained unprocessed. This would create challenges in monitoring wood usage, ensuring quality control, and maintaining sustainable harvesting practices.
- **Challenge**: Ensuring the correct tracking of wood from harvest to processing and being able to identify and manage individual wood batches and trees was a complex task.

Approach Taken to Resolve the Issue

- **Solution**: To address the issue, the concept of **wood batches** was introduced. After the harvest, the wood was grouped into **batches**, with each batch representing a collection of wood harvested together. This allowed for easier tracking and monitoring, as each batch could be assigned an ID and linked to multiple harvests. Each batch could then be further categorized for processing, and the system would track the processing stages of the wood as a whole rather than individual trees.
    - Each **batch** can consist of wood from different **harvests**, and tracking the batch allows for streamlined processing management, including the monitoring of processed and unprocessed wood. This also helped

improve visibility and control over the process, reducing the complexity of managing wood at the individual tree level.
    - Furthermore, by grouping wood into batches, it was easier to ensure that the wood was properly accounted for at each stage of the processing, improving consistency and efficiency.

Alternative Solutions Considered

- **Tracking Individual Trees**: One alternative was to continue processing and tracking wood on an individual tree basis. While this approach would provide the most granular data, it would require significant resources and would be difficult to scale. Manually tracking each tree's journey through the entire system was time-consuming and error-prone.
- **Reason for Choosing the Batch Solution**: The batch approach was selected because it simplified the process of managing large volumes of wood. It enabled efficient tracking of wood through the processing stages, reduced the complexity of managing individual trees, and provided a more scalable solution that could grow with the system.

3 Misclassification of Tree Count as a Strong Entity

- **Description**: Initially, the **Tree Count** entity was designed as a **strong entity** in the system, meaning it was considered independent and had its own primary key. However, as development progressed, it became clear that the **Tree Count** could not exist without a specific region, making it dependent on the **Region** entity for its identity. This realization highlighted a fundamental issue: the Tree Count was not independent but rather was associated with a particular **Region** and depended on the region's identity to maintain consistency.
- **Impact**: As a strong entity, **Tree Count** was treated as an independent record, which made it challenging to manage relationships with the **Region** entity. The system required a way to accurately represent the relationship between **Tree Count** and **Region**, ensuring that the tree count for each region was properly tracked and updated over time. The initial design led to

complications in linking the **Tree Count** with regions and managing updates across the system.

- **Challenge**: The main challenge was identifying that **Tree Count** should not be treated as a standalone entity but rather as a dependent entity that requires a foreign key to the **Region** entity, making it a **weak entity**.

Approach Taken to Resolve the Issue

- **Solution**: After recognizing the dependency, **Tree Count** was reclassified as a **weak entity**. It now depends on the **Region** entity for its identity and is identified by a combination of its own attributes and the **RegionID** from the **Region** entity. This relationship was formalized as a **"Has"** relationship between **Region** and **Tree Count**, where **Tree Count** cannot exist without being associated with a **Region**.
    - The **Tree Count** entity now uses **RegionID** (a foreign key from the **Region** entity) to form a composite primary key along with its own attributes, ensuring that it is always tied to a specific region. This change also made it easier to manage updates to the tree count as the data is directly linked to the region in which the trees are located.

Alternative Solutions Considered

- **Maintaining Tree Count as a Strong Entity**: One alternative was to keep **Tree Count** as a strong entity and attempt to manage its independence. However, this would lead to inconsistent data and difficulties when updating the tree count across multiple regions.
- **Reason for Choosing the Weak Entity Solution**: The weak entity solution was chosen because it accurately represented the real-world relationship between **Tree Count** and **Region**. The **Tree Count** cannot exist without a **Region**, so reclassifying it as a weak entity with a composite primary key solved the issue of dependency and ensured proper data integrity in the system.

4 Calculating Sustainability Score

- **Description**: The **Sustainability Score** in the **Harvest Report** is calculated using the formula:
  Sustainability Score={(Replanted TreesHarvested Trees)×100
  This formula works well and can be directly linked to the **Harvest Table**. However, automating the calculation process and linking it with the database posed a challenge in terms of real-time updates and consistency when new harvest data was added or modified.
    - **Impact**: Although the formula itself was straightforward, incorporating it into the system required creating a **trigger** that would automatically update the **Sustainability Score** in the **Harvest Report** whenever harvest data was modified. This added complexity in terms of system architecture and database performance.

5 Tree Density and Carbon Absorption Calculation

- **Description**: The values for **Tree Density** and **Carbon Absorption** in the **Harvest Report** presented a much more complex challenge because these metrics depend on the **Tree Count** for a given **Region**. This required an accurate way to track the number of trees in a region and integrate it with the **Harvest Report**. The real difficulty arose because of the **many-to-many relationship** between **Harvest Report** and **Region**, which is essential for associating each **Harvest Report** with the correct regions.
    - **Impact**: The **Tree Density** and **Carbon Absorption** calculations rely on **Tree Count** for each region, but since multiple harvests can occur in the same region, and each harvest might involve a different portion of that region, associating **Harvest Reports** with their respective regions was complex. This required creating a **join table** or an intermediary table to correctly link **Harvest Reports** to multiple regions, which could then be used to access the **Tree Count** for the region and calculate the metrics accordingly.

Solution

Approach Taken to Resolve the Issue

- **Sustainability Score Calculation**:
  To automate the calculation of the **Sustainability Score**, a **trigger** was implemented in the database to ensure that whenever new harvest data is entered or updated in the **Harvest Table**, the **Sustainability Score** is automatically calculated and updated in the **Harvest Report**. This trigger works by using the formula defined above and linking the harvested and replanted tree values directly from the **Harvest Table** to the **Harvest Report**.
- **Tree Density and Carbon Absorption Calculation**:
  The **Tree Density** and **Carbon Absorption** metrics depend on the **Tree Count** of a given **Region**. Since a **Harvest Report** can be associated with multiple regions, and each region has its own **Tree Count**, a **many-to-many relationship** table between **Harvest Report** and **Region** was created. This table stores the link between harvests and regions, allowing the system to fetch the correct **RegionID** for each harvest. From this, we can easily retrieve the **Tree Count** for each region.
    - **Join Table**: A new intermediary table, say **Region_Harvest_Report**, was created to link **Harvest Report** with multiple regions. This table associates a **Harvest Report** with its corresponding regions by using a composite key consisting of the **Harvest Report ID** and **Region ID**.
    - **Calculating Metrics**: Once the region is identified, the **Tree Count** for that region is fetched from the **Tree Count** table. Using the total **Tree Count** in the region, the **Tree Density** and **Carbon Absorption** can be calculated and included in the **Harvest Report**.

6 Identifying Processed Wood

- **Description**: A significant challenge was tracking the processed wood that had gone through all the stages of processing. Given that wood moves through multiple stages of processing (e.g., cutting, refining, treating, etc.), it was essential to accurately identify and monitor wood that had completed all the stages and was ready for further use or sale. Without a proper system in

place to track this, it could be difficult to ensure that only fully processed wood moves forward in the supply chain.

Solution :Creation of the Inventory Table: To effectively manage processed wood, we created an **Inventory Table** that keeps track of all processed wood. This table stores details such as the quantity of processed wood, the batch ID, and the current status of the wood (whether it's ready for manufacturing, in transit, or sold). The table ensures that only wood that has gone through all the processing stages is considered processed, while also linking the wood to the manufacturer’s transaction process.

7 Identifying the Fate of Used Wood (Recycle or Disposal)

- **Description**: Once wood products are used, it became a challenge to determine whether the product should be sent for recycling or disposal. The main difficulty was in identifying the condition of the used wood product and whether it met the criteria for recycling or if it should be disposed of. Without a proper classification system, there was a risk of incorrectly categorizing wood products, leading to inefficiencies in recycling efforts or improper disposal practices
- **Description**: Once the decision was made that a wood product should be recycled or disposed of, it was necessary to maintain an accurate record of the wood’s final disposition. The challenge was to ensure that the recycling process or disposal method was clearly documented, so that the company could track materials, monitor sustainability goals, and maintain compliance with environmental regulations.

Approach Taken to Resolve the Issue

- **Creation of the Wood Inspection Entity**: To resolve the challenge of determining whether used wood should be recycled or disposed of, we introduced a **Wood Inspection** entity. This entity records the condition and quality of the wood products that have been used, enabling a clear decision on whether the wood can be recycled or must be disposed of. The inspection entity stores critical information such as the product condition and an assessment decision (either "Recycle" or "Dispose").

- **Linking to Recycling and Disposal**: Based on the results from the **Wood Inspection**, a relationship was established with the **Recycle** and **Waste Disposal** tables. This ensures that once the inspection is completed, the used wood is either sent to recycling (if deemed reusable) or properly disposed of (if no longer suitable for reuse).