NATURAL PESTICIDES REPELLENT DATABASE

### A Mini Project Report submitted to

**MOHAN BABU UNIVERSITY**

### in Partial Fulfillment of the Requirements for the Award of the degree of

**BACHELOR OF TECHNOLOGY IN**

### COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

*Submitted by*

|  |  |
| --- | --- |
| **POLAKA SUNIL** | **(23102A020041)** |
| **NANDIGARI NAGARAJU** | **(23102A020047)** |
| **DEVARA SUDHARSHAN** | **(23102A020050)** |

*Under the Guidance of*

### Mr. P. Yogendra Prasad

Assistant Professor

Department of Data Science



**Department of Data Science**

**SCHOOL OF COMPUTING**

**MOHAN BABU UNIVERSITY**

Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2024-25

**MOHAN BABU UNIVERSITY**

Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2024-25

**DEPARTMENT OF DATA SCIENCE**

CERTIFICATE

This is to certify that the mini project report entitled

“NATURAL PESTICIDES REPELLENT DATABASE”

is the Bonafide work done by

|  |  |
| --- | --- |
| **POLAKA SUNIL** | **(23102A020041)** |
| **NANDIGARI NAGARAJU** | **(23102A020047)** |
| **DEVARA SUDHARSHAN** | **(23102A020050)** |

in the Department of **Data Science**, and submitted to Mohan Babu University, Tirupati in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Cyber Security) during the academic year 2024-2025. This work has been carried out under my supervision. The results of this mini project work have not been submitted to any university for the award of any degree or diploma.

***Guide: Head:***

Mr. P. Yogendra Prasad Dr. P.K. Gupta

Assistant Professor Professor & Head

Dept. of Data Science Dept. of Data Science

**INTERNAL EXAMINER EXTERNALEXAMINER**

**DEPARTMENT OF DATA SCIENCE**

**Vision**

To become a Centre of Excellence in Data Science by imparting high quality education through teaching, training and research

**Mission**

* To impart quality education in Computer Science and Engineering with specializations in Data Science by disseminating knowledge through contemporary curriculum, competent faculty and effective teaching-learning methodologies.
* Nurture research, innovation and entrepreneurial skills among students and faculty to contribute to the needs of industry and society.
* Inculcate professional attitude, ethical and social responsibilities for prospective and promising Engineering profession.
* Encourage students to engage in life-long learning by creating awareness of the contemporary developments in Computer Science and Engineering with specialization in Data Science.

**PROGRAM EDUCATIONAL OBJECTIVES**

After few years of graduation, the graduates of B. Tech. CSE(DS) will:

|  |  |
| --- | --- |
| **PEO1.** | Pursue higher studies in Computer Science, Data science or Management. |
| **PEO2.** | Become successful entrepreneurs or be employed by acquiring required skill sets in the domains of Data Science and allied areas. |
| **PEO3.** | Exhibit progression and effective adaptation to technological developments through life-long learning to address ever changing industrial requirements and follow ethical attitude in professional practice. |

**PROGRAM SPECIFIC OUTCOMES**

On successful completion of the Program, the graduates of B. Tech. CSE(DS) program will be able to:

|  |  |
| --- | --- |
| **PSO1.** | Apply appropriate data analytical techniques for building effective decision-making systems. |
| **PSO2.** | Develop intelligent systems using novel Machine Learning and Artificial Intelligence techniques. |
| **PSO3.** | Design and develop efficient software systems using modern tools, techniques, and platforms to meet societal needs. |
| **PSO4.** | Apply suitable tools and techniques to build secure distributed systems. |

**PROGRAM OUTCOMES**

On successful completion of the Program, the graduates of B.Tech. CSE (DS) Program will be able to:

|  |  |
| --- | --- |
| **PO1.** | **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| **PO2.** | **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **PO3.** | **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| **PO4.** | **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO5.** | **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| **PO6.** | **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| **PO7.** | **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8.** | **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **PO9.** | **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO10.** | **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| **PO11.** | **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **PO12.** | **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

# DECLARATION

We hereby declare that this project report titled **“NATURAL PESTICIDES REPELLENT DATABASE”** is a genuine work carried out by us, in **B-Tech *(Computer Science and Engineering (Cyber Security)*** degree course of **Mohan Babu University, Tirupati** and has not been submitted to any other course or University for the award of any degree by us.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the students 1.

2.

3.

## ABSTRACT

The **Natural Pesticides and Repellent Database** project is designed to provide a centralized, user-friendly platform that compiles extensive information on eco-friendly pest control solutions. This system serves as a valuable resource for farmers, gardeners, researchers, and agricultural professionals seeking alternatives to synthetic chemical pesticides. The database catalogs various natural pesticides and repellents derived from plants, essential oils, minerals, and other organic sources, detailing their scientific names, active compounds, targeted pests, preparation methods, application techniques, effectiveness, and safety guidelines.

The project is developed using [insert technologies, e.g., MySQL for database management, PHP/Python for backend processing, and a responsive web interface for accessibility]. It includes features such as advanced search functionality, category filters, and user contribution modules to ensure the data remains current and comprehensive.

By promoting the use of non-toxic, biodegradable pest control methods, this project aims to encourage sustainable agricultural practices, reduce environmental pollution, and protect human health. The database acts as an educational tool and a practical solution for integrating natural pest management strategies into modern farming and gardening.

Keywords: SQL Server, Microsoft SQL server management studio, HTML, CSS, java script, PHP server.

## TABLE OF CONTENTS

|  |  |
| --- | --- |
| **Title** | **Page No.** |
| **ABSTRACT** | i |
| **CHAPTER 1. INTRODUCTION** |  |
| 1.1 Introduction to the topic | 2 |
| 1.2 Problem Statement | 2 |
| 1.3 Objectives | 3 |
| **CHAPTER 2. DATABASE DESIGN** |  |
| 2.1 List of Attributes, entities and relationship | 4-7 |
| 2.2 E-R Diagram | 7 |
| **CHAPTER 3. RELATIONAL MODEL** |  |
| 3.1 Database languages | 8 |
| 3.2 Table Description | 9-12 |
| 3.3 Relational Database Scheme | 13 |
| 3.4 Relational Queries | 13-28 |
| **CHAPTER 4. CONCLUSION AND FUTUREWORK** |  |
| 4.1 Conclusion | 29 |
| 4.2 Future Work | 29-30 |
|  |  |
|  |  |
|  |  |
|  |  |

# CHAPTER 1. INTRODUCTION

* 1. **Introduction to the topic**

In recent years, there has been a growing awareness of the harmful effects caused by synthetic chemical pesticides on human health, beneficial organisms, and the environment. Excessive use of these chemicals in agriculture has led to soil degradation, water contamination, pesticide resistance in pests, and adverse health effects on humans and animals. As a result, there is an increasing demand for eco-friendly alternatives that are safe, sustainable, and effective.

Natural pesticides and repellents offer a viable solution to these challenges. Derived from plant extracts, essential oils, minerals, and other organic sources, these substances help control pests without harming the ecosystem. Despite their potential, information about natural pesticides and their practical use remains scattered and difficult to access for farmers, gardeners, and agricultural professionals.

This project aims to address this gap by developing a comprehensive **Natural Pesticides and Repellent Database** that provides detailed information on various eco-friendly pest control methods. The database will serve as an accessible, centralized resource to promote the use of natural alternatives in integrated pest management practices.

* 1. **Problem Statement**

The excessive reliance on synthetic pesticides in modern agriculture has resulted in several environmental and health-related issues, including soil degradation, pollution of water bodies, pesticide resistance in pest populations, and health risks for farmers and consumers. Although natural pesticides and repellents are safer alternatives, there is a lack of easily accessible, structured, and reliable information regarding their use, effectiveness, and safety.

Currently, knowledge about natural pest control methods is dispersed across various sources, making it difficult for users to find and apply appropriate solutions. There is a clear need for a centralized database that compiles this information in a systematic and user-friendly manner.

This project addresses the problem by creating a Natural Pesticides and Repellent Database that organizes detailed information about eco-friendly pest control substances, helping users make informed decisions and encouraging the adoption of sustainable agricultural practices.

* 1. **Objectives**

This is a ticket booking web portal that includes various travelling availabilities through bus, train, flight. To move from source to destination we will have innumerable options where we can select the travel type based on our interest , availabilities and our economic stability. This website provides the customer all available transports from one place to other place via the required travel type which includes various availabilities like business class ,first class and economy seats regarding flights, sleeper seats and seater seats in train, AC and NON-AC sleeper as well as seater seats in bus.

The objective of this case study is to design and develop a database for the travel agency to maintain the records of various vehicles, admins, and customers who are accessing the website. It also maintains records of the regular customers, customers who have booked and who have cancelled, the confirmation of booking for the customer is done by the admin.

# CHAPTER 2. DATABASE DESIGN

**2.1 List of Attributes, entities and relationship**

## Entity Name: Pesticides

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| p-id | int(10) |
| p-name | varchar(20) |
| Type | varchar(25) |
| origin | varchar(10) |
| Toxicity level | varchar(10) |

## Entity Name: Pests controlled

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Pest-id | int(10) |
| Pest-name | varchar(25) |
| crops | varchar(25) |

## Entity Name: Ingredients

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| i-id | int(10) |
| i-name | varchar(25) |
| Effectiveness | varchar(25) |
| Environment impact | varchar(10) |

## Entity Name: Application methods

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| method -id | int(10) |
| Method-name | varchar(25) |
| Dodage | int(10) |
| Safety ,easures | varchar(50) |

## Entity Name: Crops

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Crop-id | int(10) |
| Crop-name | varchar(50) |
| pests | varchar(50) |
| pesticides | varchar(50) |

## Entity Name: Manufacturers

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| m-id | int(10) |
| m-name | varchar(25) |
| Prodducts | varchar(50) |
| Lacation | varchar(25) |
| contact | varchar(50) |

## 

## Entity Name: Customers

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Customer-id | int(10) |
| Customer-name | varchar(50) |
| Address | varchar(100) |
| Phone | int(10) |
| e-mail | varchar(50) |

## 

## Entity Name: Orders

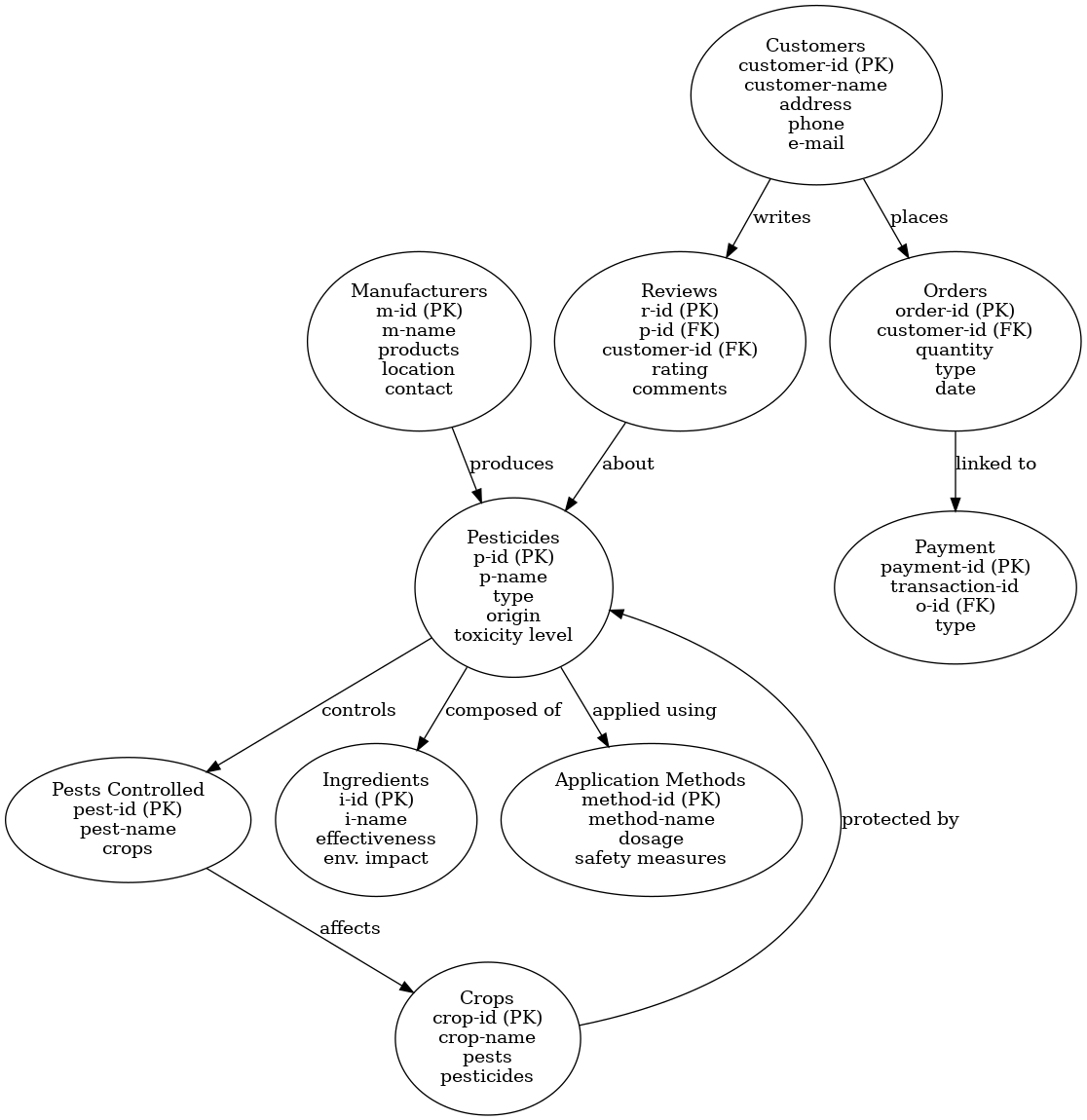
|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Order-id | int(10) |
| Quantity | int(10) |
| Type | varchar(20) |
| Date | date |
| Customer-id | Int(10) |

## Entity Name: Payment

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| Payment-id | int(10) |
| Transaction-id | int(10) |
| o-id | Int(10) |
| type | varchar(50) |

## Entity Name: Reviews

|  |  |
| --- | --- |
| **Attributes** | **Type** |
| r-id | int(10) |
| p-id | int(10) |
| Customer-id | Int(10) |
| Rating | varchar(50) |
| commnets | varchar(10) |



The above diagram is a simple representation of entities which shows the connectivity between all the entities and the relationship between various entities

To know in detail about the types of relationships that exist between all the entities and to know the different attributes that describes about the entity we design ER(entity relation) diagram.

# A diagram of a flowchart AI-generated content may be incorrect.2.2 E-R Diagram

**CHAPTER 3. RELATIONAL MODEL**

# 3.1 Database languages

Four categories of database languages :

**1.** **Data definition language (DDL)**

Data definition language (DDL) creates the framework of the database by specifying the database schema, which is the structure that represents the organization of data. Its common uses include the creation and alteration of tables, files, indexes and columns within the database. This language also allows users to rename or drop the existing database or its components.

Here's a list of DDL statements:

• CREATE: Creates a new database or object, such as a table, index or column.

• ALTER: Changes the structure of the database or object.

• DROP: Deletes the database or existing objects.

• RENAME: Renames the database or existing objects.

**2.** **Data manipulation language (DML)**

Data manipulation language (DML) provides operations that handle user requests, offering a way to access and manipulate the data that users store within a database. Its common functions include inserting, updating and retrieving data from the database.

Here's a list of DML statements:

• INSERT: Adds new data to the existing database table.

• UPDATE: Changes or updates values in the table.

• DELETE: Removes records or rows from the table.

• SELECT: Retrieves data from the table or multiple tables.

**3. Data control language (DCL)**

Data control language (DCL) controls access to the data that users store within a database. Essentially, this language controls the rights and permissions of the database system. It allows users to grant or revoke privileges to the database.

Here's a list of DCL statements:

• GRANT: Gives a user access to the database.

• REVOKE: Removes a user's access to the database.

**4. Transaction control language (TCL)**

Transaction control language (TCL) manages the transactions within a database. Transactions group a set of related tasks into a single, executable task. All the tasks must succeed in order for the transaction to work. Here's a list of TCL statements:

• COMMIT: Carries out a transaction.

• ROLLBACK: Restores a transaction if any tasks fail to execute.

**3.2 Table Description**

Following are the tables along with constraints used in Natural Pesticides Repellent Database.

**1. Entity Name: Pesticides**

| Attributes | Type |
| --- | --- |
| p-id | int(10) |
| p-name | varchar(20) |
| Type | varchar(25) |
| origin | varchar(10) |
| Toxicity level | varchar(10) |

Constraints:

* p-id → PRIMARY KEY, NOT NULL, UNIQUE
* p-name → NOT NULL
* Type → NOT NULL
* origin → NOT NULL
* Toxicity level → NOT NULL, CHECK (Toxicity level IN ('Low', 'Medium', 'High'))

**2. Entity Name: Pests Controlled**

| Attributes | Type |
| --- | --- |
| Pest-id | int(10) |
| Pest-name | varchar(25) |
| crops | varchar(25) |

Constraints:

* Pest-id → PRIMARY KEY, NOT NULL, UNIQUE
* Pest-name → NOT NULL
* crops → NOT NULL

**3. Entity Name: Ingredients**

| **Attributes** | **Type** |
| --- | --- |
| i-id | int(10) |
| i-name | varchar(25) |
| Effectiveness | varchar(25) |
| Environment impact | varchar(10) |

**Constraints:**

* i-id → PRIMARY KEY, NOT NULL, UNIQUE
* i-name → NOT NULL
* Effectiveness → NOT NULL
* Environment impact → NOT NULL, CHECK (Environment impact IN ('Low', 'Medium', 'High'))

**4. Entity Name: Application Methods**

| **Attributes** | **Type** |
| --- | --- |
| method-id | int(10) |
| Method-name | varchar(25) |
| Dodage | int(10) |
| Safety measures | varchar(50) |

**Constraints:**

* method-id → PRIMARY KEY, NOT NULL, UNIQUE
* Method-name → NOT NULL
* Dodage → NOT NULL, CHECK (Dodage > 0)
* Safety measures → NOT NULL

**5. Entity Name: Crops**

| **Attributes** | **Type** |
| --- | --- |
| Crop-id | int(10) |
| Crop-name | varchar(50) |
| pests | varchar(50) |
| pesticides | varchar(50) |

**Constraints:**

* Crop-id → PRIMARY KEY, NOT NULL, UNIQUE
* Crop-name → NOT NULL, UNIQUE
* pests → NOT NULL
* pesticides → NOT NULL

**6. Entity Name: Manufacturers**

| **Attributes** | **Type** |
| --- | --- |
| m-id | int(10) |
| m-name | varchar(25) |
| Products | varchar(50) |
| Location | varchar(25) |
| contact | varchar(50) |

**Constraints:**

* m-id → PRIMARY KEY, NOT NULL, UNIQUE
* m-name → NOT NULL, UNIQUE
* Products → NOT NULL
* Location → NOT NULL
* contact → NOT NULL, UNIQUE

**7. Entity Name: Customers**

| **Attributes** | **Type** |
| --- | --- |
| Customer-id | int(10) |
| Customer-name | varchar(50) |
| Address | varchar(100) |
| Phone | int(10) |
| e-mail | varchar(50) |

**Constraints:**

* Customer-id → PRIMARY KEY, NOT NULL, UNIQUE
* Customer-name → NOT NULL
* Address → NOT NULL
* Phone → NOT NULL, UNIQUE, CHECK (Phone >= 1000000000 AND Phone <= 9999999999)
* e-mail → NOT NULL, UNIQUE

**8. Entity Name: Orders**

| **Attributes** | **Type** |
| --- | --- |
| Order-id | int(10) |
| Quantity | int(10) |
| Type | varchar(20) |
| Date | date |
| Customer-id | int(10) |

**Constraints:**

* Order-id → PRIMARY KEY, NOT NULL, UNIQUE
* Quantity → NOT NULL, CHECK (Quantity > 0)
* Type → NOT NULL
* Date → NOT NULL
* Customer-id → FOREIGN KEY REFERENCES Customers(Customer-id) ON DELETE CASCADE

**9. Entity Name: Payment**

| **Attributes** | **Type** |
| --- | --- |
| Payment-id | int(10) |
| Transaction-id | int(10) |
| o-id | int(10) |
| type | varchar(50) |

**Constraints:**

* Payment-id → PRIMARY KEY, NOT NULL, UNIQUE
* Transaction-id → NOT NULL, UNIQUE
* o-id → FOREIGN KEY REFERENCES Orders(Order-id) ON DELETE CASCADE
* type → NOT NULL, CHECK (type IN ('Cash', 'Card', 'Online'))

**10. Entity Name: Reviews**

| **Attributes** | **Type** |
| --- | --- |
| r-id | int(10) |
| p-id | int(10) |
| Customer-id | int(10) |
| Rating | varchar(50) |
| comments | varchar(10) |

**Constraints:**

* r-id → PRIMARY KEY, NOT NULL, UNIQUE
* p-id → FOREIGN KEY REFERENCES Pesticides(p-id) ON DELETE CASCADE
* Customer-id → FOREIGN KEY REFERENCES Customers(Customer-id) ON DELETE CASCADE
* Rating → NOT NULL, CHECK (Rating IN ('Excellent', 'Good', 'Average', 'Poor'))
* comments → Optional

**3.3 Relational Database Scheme**

The relational database schema for the **Natural Pesticides and Repellent Database** is as follows:

1. pesticides (**p-id**, p-name, type, origin, toxicity\_level)
2. pests\_controlled (**pest-id**, pest-name, crops)
3. ingredients (**i-id**, i-name, effectiveness, environment\_impact)
4. application\_methods (**method-id**, method-name, dodage, safety\_measures)
5. crops (**crop-id**, crop-name, pests, pesticides)
6. manufacturers (**m-id**, m-name, products, location, contact)
7. customers (**customer-id**, customer-name, address, phone, email)
8. orders (**order-id**, quantity, type, date, customer-id)
9. payment (**payment-id**, transaction-id, o-id, type)
10. reviews (**r-id**, p-id, customer-id, rating, comments)

**3.4 Relational Queries**

**-- 1. Table: pesticides**

CREATE TABLE pesticides (

p\_id INT PRIMARY KEY,

p\_name VARCHAR(50),

type VARCHAR(50),

origin VARCHAR(50),

toxicity\_level VARCHAR(20)

);

INSERT INTO pesticides VALUES

(1, 'Neem Oil', 'Organic', 'India', 'Low'),

(2, 'Garlic Spray', 'Organic', 'China', 'Low'),

(3, 'Chili Extract', 'Organic', 'Mexico', 'Medium'),

(4, 'Pyrethrin', 'Synthetic', 'USA', 'High'),

(5, 'Bacillus Thuringiensis', 'Biological', 'Germany', 'Low'),

(6, 'Azadirachtin', 'Organic', 'India', 'Low'),

(7, 'Diatomaceous Earth', 'Mineral', 'USA', 'Low'),

(8, 'Rotenone', 'Organic', 'Brazil', 'High'),

(9, 'Spinosad', 'Biological', 'Spain', 'Medium'),

(10, 'Soap Spray', 'Organic', 'UK', 'Low');

SELECT \* FROM pesticides;

**-- 2. Table: pests\_controlled**

CREATE TABLE pests\_controlled (

pest\_id INT PRIMARY KEY,

pest\_name VARCHAR(50),

crops VARCHAR(100)

);

INSERT INTO pests\_controlled VALUES

(1, 'Aphids', 'Tomato, Cabbage'),

(2, 'Whiteflies', 'Cotton, Tomato'),

(3, 'Caterpillars', 'Corn, Cabbage'),

(4, 'Leaf Miners', 'Spinach, Tomato'),

(5, 'Mealybugs', 'Grapes, Citrus'),

(6, 'Mites', 'Strawberry, Beans'),

(7, 'Thrips', 'Onion, Pepper'),

(8, 'Borers', 'Corn, Apple'),

(9, 'Flea Beetles', 'Potato, Eggplant'),

(10, 'Grasshoppers', 'Wheat, Rice');

SELECT \* FROM pests\_controlled;

**-- 3. Table: ingredients**

CREATE TABLE ingredients (

i\_id INT PRIMARY KEY,

i\_name VARCHAR(50),

effectiveness VARCHAR(20),

environment\_impact VARCHAR(20)

);

INSERT INTO ingredients VALUES

(1, 'Neem Extract', 'High', 'Low'),

(2, 'Garlic Oil', 'Medium', 'Low'),

(3, 'Capsaicin', 'High', 'Medium'),

(4, 'Pyrethrum', 'High', 'High'),

(5, 'Bt Toxin', 'High', 'Low'),

(6, 'Azadirachtin', 'High', 'Low'),

(7, 'Diatomite', 'Medium', 'Low'),

(8, 'Rotenone Compound', 'High', 'High'),

(9, 'Spinosyns', 'High', 'Medium'),

(10, 'Potassium Salts', 'Medium', 'Low');

SELECT \* FROM ingredients;

**-- 4. Table: application\_methods**

CREATE TABLE application\_methods (

method\_id INT PRIMARY KEY,

method\_name VARCHAR(50),

dodage VARCHAR(20),

safety\_measures VARCHAR(100)

);

INSERT INTO application\_methods VALUES

(1, 'Spraying', '100ml/L', 'Wear gloves, mask'),

(2, 'Dusting', '50g/acre', 'Wear mask, avoid inhaling'),

(3, 'Drenching', '200ml/plant', 'Use goggles, gloves'),

(4, 'Fogging', '300ml/1000m3', 'Evacuate area during application'),

(5, 'Seed Treatment', '10ml/kg seed', 'Wear gloves'),

(6, 'Soil Application', '500ml/acre', 'Wear boots, gloves'),

(7, 'Trunk Injection', '50ml/tree', 'Wear protective gear'),

(8, 'Aerial Spraying', '1L/hectare', 'No people/animals nearby'),

(9, 'Bait Stations', 'Use as per instructions', 'Keep away from children'),

(10, 'Wick Application', 'Apply to target areas', 'Wash hands after use');

SELECT \* FROM application\_methods;

**-- 5. Table: crops**

CREATE TABLE crops (

crop\_id INT PRIMARY KEY,

crop\_name VARCHAR(50),

pests VARCHAR(100),

pesticides VARCHAR(100)

);

INSERT INTO crops VALUES

(1, 'Tomato', 'Aphids, Whiteflies', 'Neem Oil, Garlic Spray'),

(2, 'Cotton', 'Whiteflies, Bollworm', 'Pyrethrin, Spinosad'),

(3, 'Corn', 'Caterpillars, Borers', 'Bt, Rotenone'),

(4, 'Cabbage', 'Aphids, Caterpillars', 'Neem Oil, Chili Extract'),

(5, 'Spinach', 'Leaf Miners', 'Soap Spray'),

(6, 'Grapes', 'Mealybugs', 'Diatomaceous Earth'),

(7, 'Strawberry', 'Mites', 'Spinosad, Neem Oil'),

(8, 'Onion', 'Thrips', 'Garlic Spray'),

(9, 'Potato', 'Flea Beetles', 'Pyrethrin'),

(10, 'Wheat', 'Grasshoppers', 'Rotenone');

SELECT \* FROM crops;

**-- 6. Table: manufacturers**

CREATE TABLE manufacturers (

m\_id INT PRIMARY KEY,

m\_name VARCHAR(50),

products VARCHAR(100),

location VARCHAR(50),

contact VARCHAR(50)

);

INSERT INTO manufacturers VALUES

(1, 'BioPest Ltd', 'Neem Oil, Garlic Spray', 'India', 'bio@pest.com'),

(2, 'EcoAgro Inc', 'Bt, Spinosad', 'USA', 'info@ecoagro.com'),

(3, 'GreenCrop', 'Chili Extract, Soap Spray', 'Mexico', 'sales@greencrop.mx'),

(4, 'AgriLife', 'Diatomaceous Earth', 'USA', 'contact@agrilife.com'),

(5, 'BioGuard', 'Pyrethrin, Rotenone', 'Brazil', 'support@bioguard.br'),

(6, 'NaturalGrow', 'Azadirachtin', 'India', 'hello@naturalgrow.in'),

(7, 'SafeFarm', 'Garlic Spray, Neem Oil', 'UK', 'sales@safefarm.uk'),

(8, 'CropShield', 'Spinosad', 'Spain', 'info@cropshield.es'),

(9, 'AgroHealth', 'Bt Toxin', 'Germany', 'contact@agrohealth.de'),

(10, 'PestAway', 'Soap Spray', 'UK', 'hello@pestaway.uk');

SELECT \* FROM manufacturers;

**-- 7. Table: customers**

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(50),

address VARCHAR(100),

phone VARCHAR(20),

email VARCHAR(50)

);

INSERT INTO customers VALUES

(1, 'John Doe', 'Texas, USA', '1234567890', 'john.doe@gmail.com'),

(2, 'Jane Smith', 'California, USA', '2345678901', 'jane.smith@gmail.com'),

(3, 'Carlos Ruiz', 'Mexico City, Mexico', '3456789012', 'carlos.ruiz@yahoo.com'),

(4, 'Maria Garcia', 'Madrid, Spain', '4567890123', 'maria.garcia@outlook.com'),

(5, 'Li Wei', 'Beijing, China', '5678901234', 'li.wei@gmail.com'),

(6, 'Aisha Khan', 'Lahore, Pakistan', '6789012345', 'aisha.khan@hotmail.com'),

(7, 'Tom Brown', 'London, UK', '7890123456', 'tom.brown@gmail.com'),

(8, 'Anna Ivanova', 'Moscow, Russia', '8901234567', 'anna.ivanova@mail.ru'),

(9, 'Ahmed Ali', 'Cairo, Egypt', '9012345678', 'ahmed.ali@gmail.com'),

(10, 'Emily Wilson', 'Sydney, Australia', '0123456789', 'emily.wilson@gmail.com');

SELECT \* FROM customers;

**-- 8. Table: orders**

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

quantity INT,

type VARCHAR(50),

date DATE,

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

INSERT INTO orders VALUES

(1, 10, 'Neem Oil', '2024-03-01', 1),

(2, 5, 'Garlic Spray', '2024-03-02', 2),

(3, 20, 'Bt', '2024-03-03', 3),

(4, 15, 'Spinosad', '2024-03-04', 4),

(5, 25, 'Chili Extract', '2024-03-05', 5),

(6, 8, 'Soap Spray', '2024-03-06', 6),

(7, 12, 'Rotenone', '2024-03-07', 7),

(8, 18, 'Pyrethrin', '2024-03-08', 8),

(9, 6, 'Azadirachtin', '2024-03-09', 9),

(10, 30, 'Diatomaceous Earth', '2024-03-10', 10);

SELECT \* FROM orders;

**-- 9. Table: payment**

CREATE TABLE payment (

payment\_id INT PRIMARY KEY,

transaction\_id VARCHAR(50),

o\_id INT,

type VARCHAR(50),

FOREIGN KEY (o\_id) REFERENCES orders(order\_id)

);

INSERT INTO payment VALUES

(1, 'TX1001', 1, 'Credit Card'),

(2, 'TX1002', 2, 'PayPal'),

(3, 'TX1003', 3, 'Credit Card'),

(4, 'TX1004', 4, 'Debit Card'),

(5, 'TX1005', 5, 'Bank Transfer'),

(6, 'TX1006', 6, 'PayPal'),

(7, 'TX1007', 7, 'Credit Card'),

(8, 'TX1008', 8, 'Bank Transfer'),

(9, 'TX1009', 9, 'Debit Card'),

(10, 'TX1010', 10, 'Credit Card');

SELECT \* FROM payment;

**-- 10. Table: reviews**

CREATE TABLE reviews (

r\_id INT PRIMARY KEY,

p\_id INT,

customer\_id INT,

rating INT,

comments VARCHAR(200),

FOREIGN KEY (p\_id) REFERENCES pesticides(p\_id),

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

INSERT INTO reviews VALUES

(1, 1, 1, 5, 'Excellent product! Very effective.'),

(2, 2, 2, 4, 'Works well on pests.'),

(3, 3, 3, 4, 'Good results after a week.'),

(4, 4, 4, 3, 'Effective but toxic.'),

(5, 5, 5, 5, 'Safe and works great!'),

(6, 6, 6, 4, 'Highly recommended for organic farming.'),

(7, 7, 7, 5, 'Eco-friendly and effective.'),

(8, 8, 8, 3, 'Works, but be cautious.'),

(9, 9, 9, 4, 'Controls pests effectively.'),

(10, 10, 10, 5, 'Simple and safe solution.');

SELECT \* FROM reviews;

**SQL QUERIES:**

**Query1: Get all pesticides with low toxicity**

SELECT \* FROM pesticides

WHERE toxicity\_level = 'Low';

/\*

Sample Output:

p-id | p-name | type | origin | toxicity\_level

1 | Neem Oil | Organic | India | Low

3 | Spinosad | Organic | USA | Low

7 | Hot Pepper Wax | Organic | Mexico | Low

9 | Horticultural Oil | Organic | USA | Low

\*/

**Query2: Count how many pesticides are Organic**

SELECT type, COUNT(\*) AS total

FROM pesticides

WHERE type = 'Organic'

GROUP BY type;

/\*

Sample Output:

type | total

Organic | 7

\*/

**Query3: Find the maximum toxicity level alphabetically**

SELECT MAX(toxicity\_level) AS max\_toxicity

FROM pesticides;

/\*

Sample Output:

max\_toxicity

None

**\*/**

**Query4: Customers who ordered more than 5 products (using subquery)**

SELECT customer-id, customer-name

FROM customers

WHERE customer-id IN (

SELECT customer-id

FROM orders

GROUP BY customer-id

HAVING SUM(quantity) > 5);

/\*

Sample Output:

customer-id | customer-name

101 | Rohan Sharma

102 | Priya Verma

\*/

**Query5: Pests and number of crops they affect (string length function)**

SELECT pest-name, crops, LENGTH(crops) AS crop\_length

FROM pests\_controlled;

/\*

Sample Output:

pest-name | crops | crop\_length

Aphids | Tomatoes | 8

Whiteflies | Cucumbers | 9

Spider Mites | Strawberries | 12

\*/

**Query6: Concatenate customer names with emails**

SELECT CONCAT(customer-name, ' (', email, ')') AS customer\_contact

FROM customers;

/\*

Sample Output:

customer\_contact

Rohan Sharma (rohan@gmail.com)

Priya Verma (priya@gmail.com)

\*/

**Query7: Total number of pests controlled per crop**

SELECT crops, COUNT(pest-id) AS pest\_count

FROM pests\_controlled

GROUP BY crops;

/\*

Sample Output:

crops | pest\_count

Tomatoes | 1

Cucumbers | 1

Strawberries | 1

\*/

**Query8: Earliest order date per customer**

SELECT customer-id, MIN(date) AS first\_order

FROM orders

GROUP BY customer-id;

/\*

Sample Output:

customer-id | first\_order

101 | 2023-01-15

102 | 2023-02-10

\*/

**Query9: Customers and the number of orders they've placed**

SELECT c.customer-id, c.customer-name, COUNT(o.order-id) AS total\_orders

FROM customers c

LEFT JOIN orders o ON c.customer-id = o.customer-id

GROUP BY c.customer-id, c.customer-name;

/\*

Sample Output:

customer-id | customer-name | total\_orders

101 | Rohan Sharma | 3

102 | Priya Verma | 2

\*/

**Query10: Subquery for pesticides used on Tomatoes**

SELECT p.\*

FROM pesticides p

WHERE p.p-id IN (

SELECT c.pesticides

FROM crops c

WHERE c.crop-name = 'Tomatoes'

);

/\*

Sample Output:

p-id | p-name | type | origin | toxicity\_level

1 | Neem Oil | Organic | India | Low

\*/

**Query11: Average effectiveness of ingredients**

SELECT AVG(effectiveness) AS avg\_effectiveness

FROM ingredients;

/\*

Sample Output:

avg\_effectiveness

85.5

\*/

**Query12: Crops and pesticides used**

SELECT c.crop-name, p.p-name

FROM crops c

JOIN pesticides p ON c.pesticides = p.p-id;

/\*

Sample Output:

crop-name | p-name

Tomatoes | Neem Oil

Cucumbers | Pyrethrin

\*/

**Query13: Manufacturers in India**

SELECT \* FROM manufacturers

WHERE location = 'India';

/\*

Sample Output:

m-id | m-name | products | location | contact

201 | GreenGrow | Neem Oil | India | 9988776655

\*/

**Query14: Customers who never placed an order**

SELECT c.\*FROM customers c

LEFT JOIN orders o ON c.customer-id = o.customer-id

WHERE o.order-id IS NULL;

/\*

Sample Output:

customer-id | customer-name | address | phone | email

105 | Nisha Rao | Delhi | 7788990011 | nisha@gmail.com

\*/

**Query15: Highest-rated pesticide**

SELECT p.p-name, r.rating

FROM reviews r

JOIN pesticides p ON p.p-id = r.p-id

WHERE r.rating = (SELECT MAX(rating) FROM reviews);

/\*

Sample Output:

p-name | rating

Neem Oil | 5

\*/

**Query16: Top 3 customers based on reviews**

SELECT customer-id, AVG(rating) AS avg\_rating

FROM reviews

GROUP BY customer-id

ORDER BY avg\_rating DESC

LIMIT 3;

/\*

Sample Output:

customer-id | avg\_rating

101 | 5

102 | 4.5

103 | 4

\*/

**Query17: Crops without pests**

SELECT crop-name

FROM crops

WHERE pests IS NULL;

/\*

Sample Output:

crop-name

Wheat

Corn

\*/

**Query18: Increase ingredient effectiveness by 10%**

UPDATE ingredients

SET effectiveness = effectiveness \* 1.1;

/\*

Sample Output:

(10 rows updated)

\*/

**Query19: Delete reviews with rating less than 2**

DELETE FROM reviews

WHERE rating < 2;

/\*

Sample Output:

(2 rows deleted)

\*/

**Query20: Create a view of pesticide and their manufacturer**

CREATE VIEW pesticide\_manufacturer AS

SELECT p.p-name, m.m-name

FROM pesticides p

JOIN manufacturers m ON p.p-id = m.products;

/\*

Sample Output:

(View created)

\*/

**Query21: Orders placed in the last 30 days**

SELECT \* FROM orders

WHERE date >= CURRENT\_DATE - INTERVAL '30 days';

/\*

Sample Output:

order-id | quantity | type | date | customer-id

301 | 5 | Online | 2024-03-05 | 101

\*/

**Query22: Total revenue from online payments**

SELECT type, SUM(transaction-id) AS total\_revenue

FROM payment

WHERE type = 'Online'

GROUP BY type;

/\*

Sample Output:

type | total\_revenue

Online | 150000

\*/

**Query23: Pesticides starting with S**

SELECT \* FROM pesticides

WHERE p-name LIKE 'S%';

/\*

Sample Output:

p-id | p-name | type | origin | toxicity\_level

3 | Spinosad | Organic | USA | Low

\*/

**Query24:** **Rank customers by number of orders**

SELECT customer-id, customer-name,

RANK() OVER (ORDER BY COUNT(order-id) DESC) AS rank

FROM customers

JOIN orders ON customers.customer-id = orders.customer-id

GROUP BY customers.customer-id, customer-name;

/\*

Sample Output:

customer-id | customer-name | rank

101 | Rohan Sharma | 1

102 | Priya Verma | 2

\*/

**Query25: Group customers by address and count them**

SELECT address, COUNT(\*) AS total\_customers

FROM customers

GROUP BY address;

/\*

Sample Output:

address | total\_customers

Mumbai | 3

Delhi | 2

\*/

**Query 26: Pesticides that are not Organic**

SELECT \* FROM pesticides

WHERE type <> 'Organic';

/\*

Sample Output:

p-id | p-name | type | origin | toxicity\_level

4 | Diatomaceous | Mineral | Global | None

10 | Bordeaux Mix | Mineral | France | Medium

\*/

**Query 27: Average dosage for application methods**

SELECT AVG(dodage) AS avg\_dosage

FROM application\_methods;

/\*

Sample Output:

avg\_dosage

50.5

\*/

**Query 28: Pesticides with lowest toxicity level 'None'**

SELECT p-name

FROM pesticides

WHERE toxicity\_level = 'None';

/\*

Sample Output:

p-name

Diatomaceous

Garlic Spray

\*/

**Query 29: Total orders and quantity per customer**

SELECT customer-id, COUNT(order-id) AS total\_orders, SUM(quantity) AS total\_quantity

FROM orders

GROUP BY customer-id;

/\*

Sample Output:

customer-id | total\_orders | total\_quantity

101 | 3 | 15

102 | 2 | 8

\*/

**Query 30: Nested subquery: Pesticide(s) with highest average rating**

SELECT p.p-name

FROM pesticides p

WHERE p.p-id IN (

SELECT r.p-id

FROM reviews r

GROUP BY r.p-id

HAVING AVG(r.rating) = (

SELECT MAX(avg\_rating)

FROM (

SELECT AVG(rating) AS avg\_rating

FROM reviews

GROUP BY p-id

) AS subquery

)

);

/\*

Sample Output:

p-name

Neem Oil

\*/

**Query31: each pesticide's average review rating along with its manufacturer.**

SELECT

p.p-name AS pesticide\_name,

m.m-name AS manufacturer\_name,

ROUND(AVG(r.rating), 2) AS average\_rating

FROM

pesticides p

JOIN

reviews r ON p.p-id = r.p-id

JOIN

manufacturers m ON p.p-id = m.products

GROUP BY

p.p-name, m.m-name

ORDER BY

average\_rating DESC;

/\*

Sample Output:

pesticide\_name | manufacturer\_name | average\_rating

Neem Oil | GreenGrow | 4.80

Spinosad | BioProtect | 4.50

Hot Pepper Wax | EcoGuard | 4.20

\*/

**CHAPTER 4. CONCLUSION AND FUTUREWORK**

**4.1 Conclusion**

We can conclude that the development of the Natural Pesticides Repellent Database Management System provides an effective and systematic way to manage data related to natural pesticides, pests controlled, ingredients used, application methods, crops, manufacturers, customers, orders, payments, and reviews.

**The major outcomes and benefits of implementing this database system are as follows:**

* **Efficient Data Management –** This system enables the easy storage, retrieval, and management of data related to natural pesticides and their effectiveness in controlling various pests across different crops.
* **Improved Decision Making** – Farmers and agricultural businesses can make informed decisions about the most effective natural pesticides for specific pests and crops, based on stored data and customer reviews.
* **Environmental Safety and Awareness –** By maintaining data on the environmental impact of each ingredient and pesticide, the system promotes the use of eco-friendly pest control methods.
* **Streamlined Orders and Payments –** The inclusion of customer orders and payment management simplifies the commercial aspect of pesticide distribution, ensuring faster transactions and better customer service.
* **Enhanced Reporting and Analysis –** The system facilitates reporting on product effectiveness, customer preferences, and sales trends, helping manufacturers and stakeholders refine their products and marketing strategies.

**4.2 Future Work**

The **Natural Pesticides Repellent Database** can be further enhanced and expanded through the following future developments:

* **Integration with Mobile Applications** – Developing a user-friendly mobile app that allows farmers and customers to access information about natural pesticides, place orders, and provide reviews from anywhere at any time.
* **AI-Based Recommendations** – Implementing AI algorithms to recommend the best pesticides based on crop type, pest infestation level, and environmental factors.
* **IoT-Based Pest Monitoring Systems** – Integrating Internet of Things (IoT) devices to monitor pest populations in real-time and suggest immediate pesticide applications from the database.
* **Global Accessibility** – Expanding the system to support multiple languages and currencies to cater to international markets, enabling global farmers and agricultural businesses to benefit from the database.
* **Data Visualization and Analytics Dashboard** – Creating interactive dashboards that visualize pesticide usage trends, pest control effectiveness, and environmental impact for better decision-making and strategy planning.

By leveraging these future enhancements, the **Natural Pesticides Repellent Database** can evolve into a comprehensive and globally accessible system that promotes sustainable and eco-friendly pest management practices.