

PLANT GENETIC DATABASE MANAGEMENT

Database Management Project

Madhumitha A.

PES1UG21CS314

KHUSHI PATIL

PES1UG21CS279

ABSTRACT

The Plant Genetic Database is a comprehensive repository designed to catalogue and manage genetic information pertaining to diverse plant species. This project aims to facilitate research, analysis, and understanding of plant genomics by providing an organised and accessible platform for genetic data.

The database encompasses a structured schema consisting of tables representing various aspects of plant genetics, including **Plant Species, Genome, Chromosome, Gene, Allele, DNA Sequence, Genetic Variation, Genotype, and Protein entities**. These entities are interconnected through well-defined relationships, enabling the establishment of genetic lineage and hierarchical dependencies.

The **Plant Species table** serves as the **root entity**, anchoring information about different species, their common and scientific names, and family classification. Each species is associated with its genomic information stored in the Genome table, detailing genome size and chromosome count.

The Chromosome table captures chromosome-specific data linked to genomes, containing information about chromosome numbers, length, and the count of genes it holds. Genes are represented in the Gene table, describing gene names, descriptions, positions, and strand orientation within chromosomes.

Alleles denote variations within genes and are stored in the respective table along with their types and mutation details. DNA sequences

linked to genes are recorded in the DNA Sequence table, providing sequence data and lengths.

The Genetic Variation table tracks variations associated with alleles, storing information about variation types, positions, and altered sequences. Genotypes are defined in the Genotype table, establishing relationships between alleles.

A unique aspect of this database is the inclusion of the Protein table, representing a weak entity related to genes, detailing protein names and descriptions. Proteins are encoded by genes and are connected through the "Encoded_By" relationship.

This project aims to empower researchers, botanists, and geneticists by offering a centralised platform to explore, analyse, and extract insights into plant genomics, fostering advancements in agricultural sciences, biodiversity studies, and genetic research.

SOFTWARES USED:

- Software:

MySQL

MySQL Workbench

MySQL Client

Visual Studio Code

- Tools:

Streamlit

MySQL Client

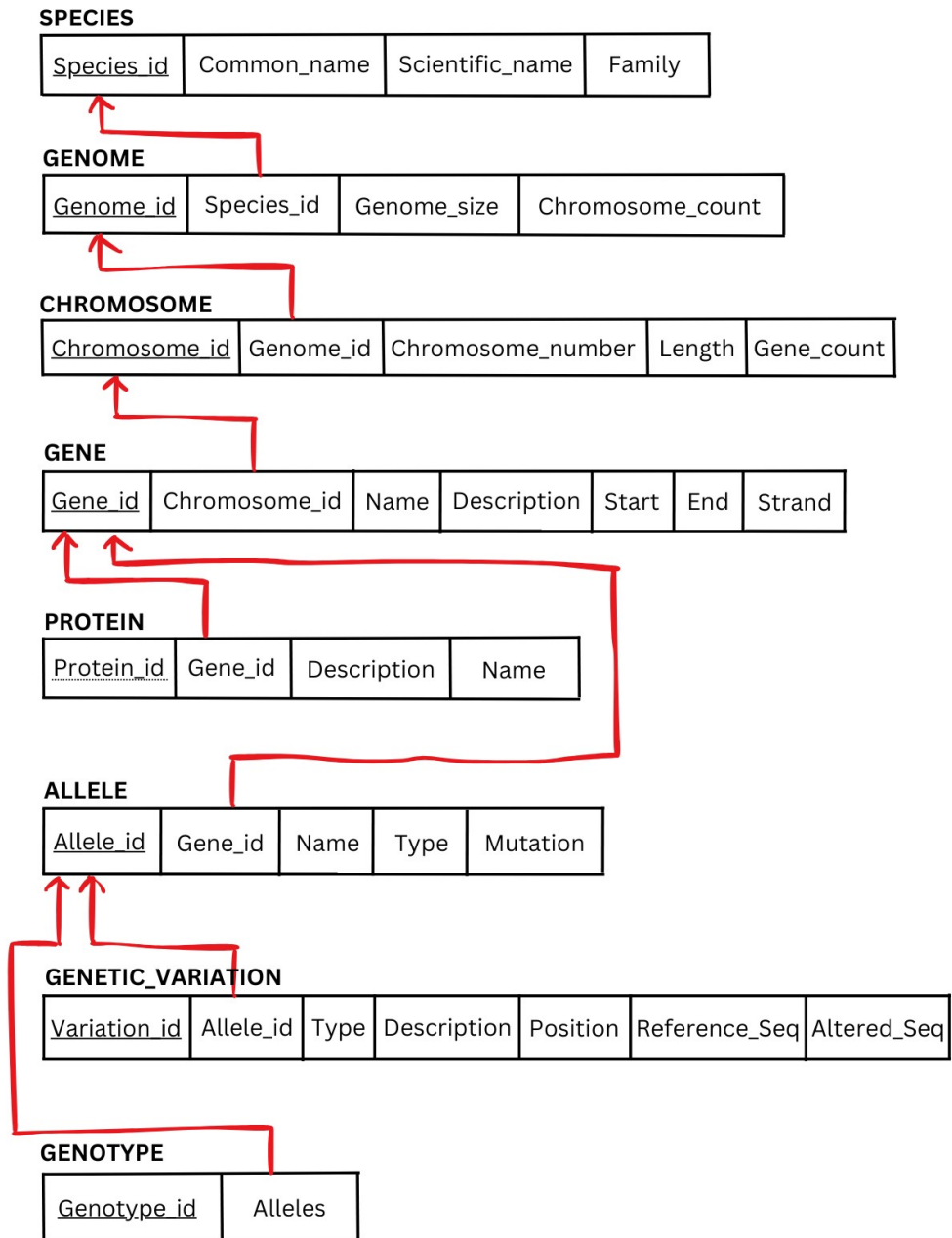
Visual Studio Code

- Programming Languages:

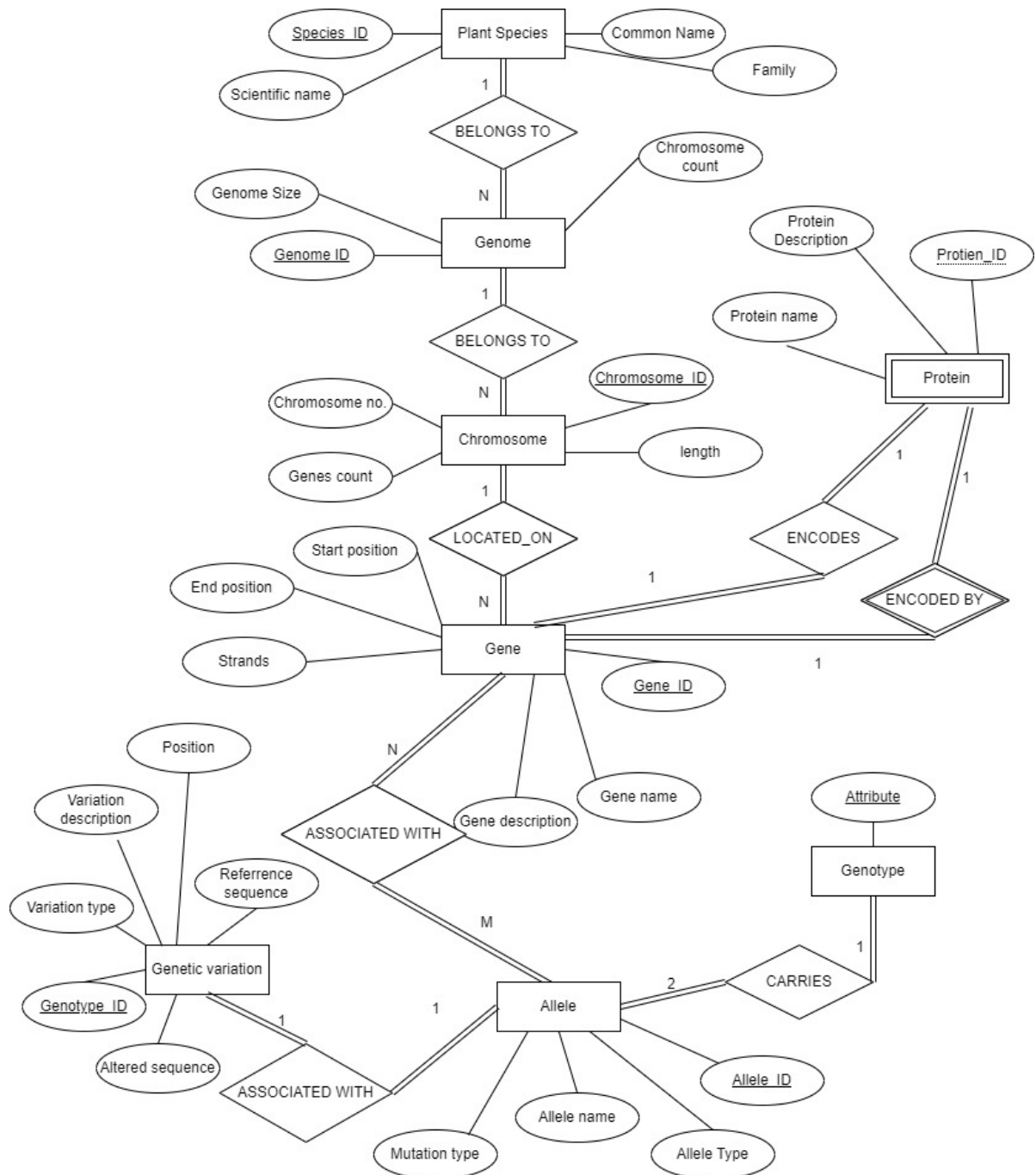
Python

HTML/CSS

Relational Schema



ER-Diagram



DDL SQL Commands:

```
-- Create Authentication database
CREATE DATABASE IF NOT EXISTS `plantdb3`;
USE `plantdb3`
`;

-- Create Users table
CREATE TABLE IF NOT EXISTS Users (
    Username VARCHAR(50) PRIMARY KEY,
    Password VARCHAR(255),
    Role VARCHAR(10)
);

-- Create Plant_Species table
CREATE TABLE IF NOT EXISTS Plant_Species (
    Species_ID INT AUTO_INCREMENT PRIMARY KEY,
    Common_Name VARCHAR(100),
    Scientific_Name VARCHAR(100),
    Family VARCHAR(100)
);

-- Create Genome table
CREATE TABLE IF NOT EXISTS Genome (
    Genome_ID INT AUTO_INCREMENT PRIMARY KEY,
    Species_ID INT,
    Genome_Size DECIMAL(10, 2),
    Chromosome_Count INT,
    FOREIGN KEY (Species_ID) REFERENCES Plant_Species(Species_ID)
);

-- Create Chromosome table
CREATE TABLE IF NOT EXISTS Chromosome (
    Chromosome_ID INT AUTO_INCREMENT PRIMARY KEY,
    Genome_ID INT,
    Chromosome_Number INT,
    Length DECIMAL(10, 2),
    Genes_Count INT,
    FOREIGN KEY (Genome_ID) REFERENCES Genome(Genome_ID)
);

-- Create Gene table
```

```

CREATE TABLE IF NOT EXISTS Gene (
    Gene_ID INT AUTO_INCREMENT PRIMARY KEY,
    Chromosome_ID INT,
    Gene_Name VARCHAR(100),
    Gene_Description TEXT,
    Start_Position INT,
    End_Position INT,
    Strand ENUM('+', '-'),
    FOREIGN KEY (Chromosome_ID) REFERENCES Chromosome(Chromosome_ID)
);

-- Create Allele table
CREATE TABLE IF NOT EXISTS Allele (
    Allele_ID INT AUTO_INCREMENT PRIMARY KEY,
    Gene_ID INT,
    Allele_Name VARCHAR(100),
    Allele_Type VARCHAR(100),
    Mutation_Type VARCHAR(100),
    FOREIGN KEY (Gene_ID) REFERENCES Gene(Gene_ID)
);

-- Create DNA_Sequence table
CREATE TABLE IF NOT EXISTS DNA_Sequence (
    Sequence_ID INT AUTO_INCREMENT PRIMARY KEY,
    Gene_ID INT,
    Sequence_Data TEXT,
    Sequence_Length INT,
    FOREIGN KEY (Gene_ID) REFERENCES Gene(Gene_ID)
);

-- Create Genetic_Variation table
CREATE TABLE IF NOT EXISTS Genetic_Variation (
    Variation_ID INT AUTO_INCREMENT PRIMARY KEY,
    Allele_ID INT,
    Variation_Type VARCHAR(100),
    Variation_Description TEXT,
    Position INT,
    Reference_Sequence TEXT,
    Altered_Sequence TEXT,
    FOREIGN KEY (Allele_ID) REFERENCES Allele(Allele_ID)
);

-- Create Genotype table

```



```
CREATE TABLE IF NOT EXISTS Genotype (  
    Genotype_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Allele_1 INT,  
    Allele_2 INT,  
    FOREIGN KEY (Allele_1) REFERENCES Allele(Allele_ID),  
    FOREIGN KEY (Allele_2) REFERENCES Allele(Allele_ID)  
);
```

```
-- Create Protein table
```

```
CREATE TABLE IF NOT EXISTS Protein (  
    Protein_ID INT AUTO_INCREMENT,  
    Protein_Name VARCHAR(100),  
    Protein_Description TEXT,  
    Gene_ID INT,  
    PRIMARY KEY (Protein_ID, Gene_ID),  
    FOREIGN KEY (Gene_ID) REFERENCES Gene(Gene_ID)  
);
```

CRUD:

The CRUD has been implemented for multiple tables but for sake of simplicity, only 1 table's screenshots have been added

CREATE:

Before:

| Plant_Species Table | | | | |
|---------------------|---|-----------|-------------------|------------|
| | 0 | 1 | 2 | 3 |
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |

Create New Plant Species:

Common Name:

Lotus

Scientific Name:

Nelumbo nucifera

Family:

Nelumbonaceae

Create

After:

| Plant_Species Table | | | | |
|---------------------|----|-----------|-------------------|---------------|
| | 0 | 1 | 2 | 3 |
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |
| 9 | 10 | Lotus | Nelumbo nucifera | Nelumbonaceae |

READ:

| Plant_Species Table | | | | |
|---------------------|----|-----------|-------------------|---------------|
| | 0 | 1 | 2 | 3 |
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |
| 9 | 10 | Lotus | Nelumbo nucifera | Nelumbonaceae |

UPDATE:

Before:

| Plant_Species Table | | | | |
|---------------------|----|-----------|-------------------|---------------|
| | 0 | 1 | 2 | 3 |
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |
| 9 | 10 | Lotus | Nelumbo nucifera | Nelumbonaceae |

Update Plant Species:

Enter Plant Species ID to Update:

10

New Common Name:

Lotus-1

New Scientific Name:

Nelumbo nucifera-1

New Family:

Nelumbonaceae-1

Update

After:

Plant_Species Table

| | 0 | 1 | 2 | 3 |
|---|----|-----------|--------------------|-----------------|
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |
| 9 | 10 | Lotus-1 | Nelumbo nucifera-1 | Nelumbonaceae-1 |

DELETE

Before:

Plant_Species Table

| | 0 | 1 | 2 | 3 |
|---|----|-----------|--------------------|-----------------|
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |
| 9 | 10 | Lotus-1 | Nelumbo nucifera-1 | Nelumbonaceae-1 |

Delete Plant Species:

Enter Plant Species ID to Delete:

10

Press Enter to apply

Delete

After:

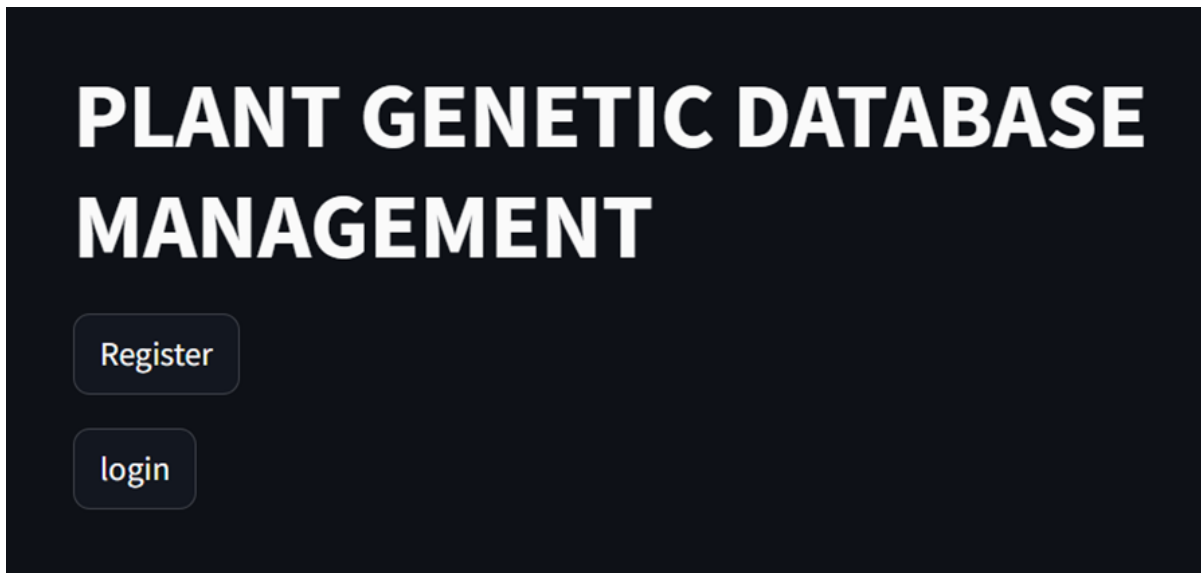
Plant_Species Table

| | 0 | 1 | 2 | 3 |
|---|---|-----------|-------------------|------------|
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |

List of functionalities

Home Page:

Users are must be registered to login to get access to the database



Registration Page:

Users are allowed 2 roles - Admin and Users.

Admins are allowed to perform CRUD operations on all tables

Users are allowed to view tables and perform queries to extract the data they want

Once the user registers, they get navigated to the login page.

Registration Page

username:

password:

Role:

 Press Enter to apply

register

Login page:

Users must login with correct credentials for authentication.

Base on the role they get different views.

Admins get redirected to the admin page while the user gets redirected to user page.

Login Page

Username:

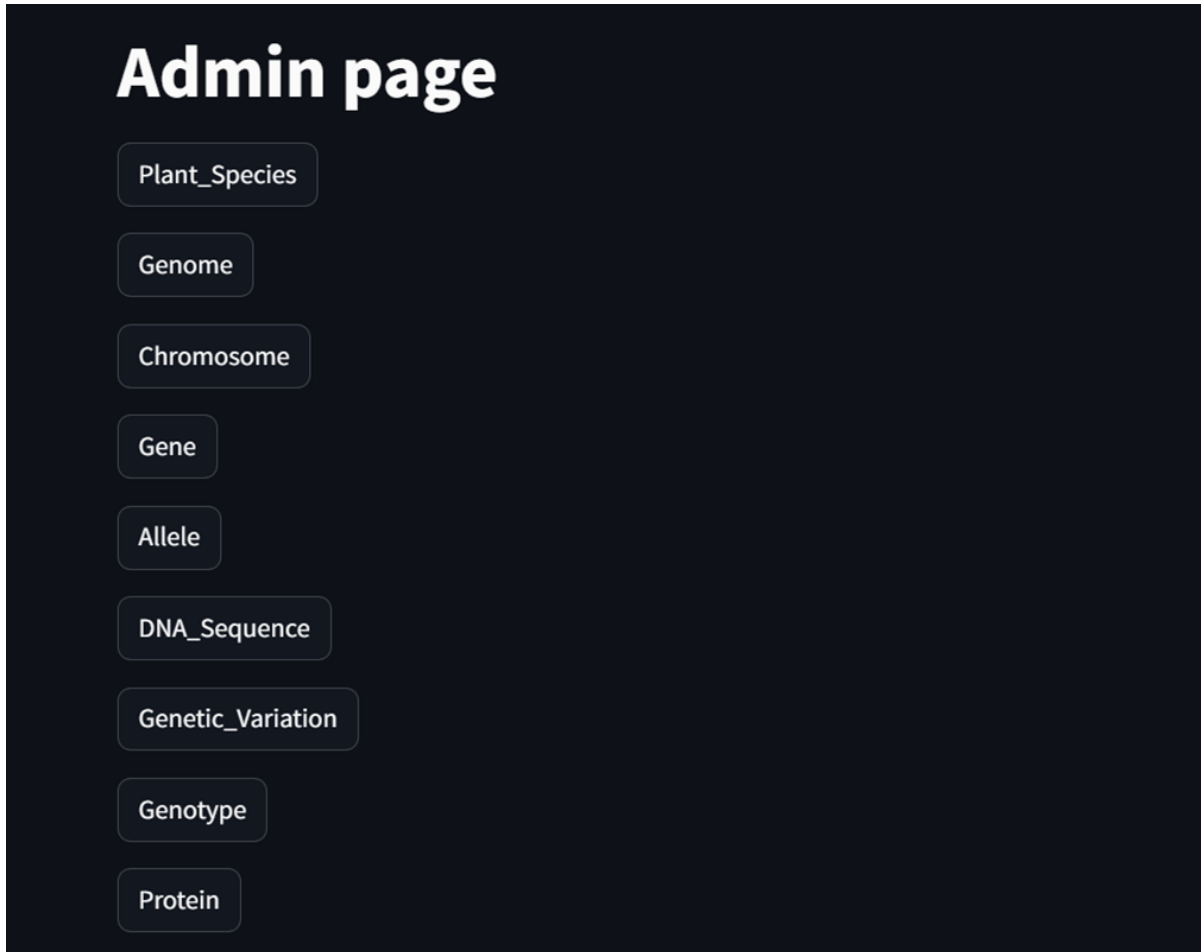
Password:

 Press Enter to apply

Login

Admin Page:

The admin is provided with buttons to alter the various tables of the database.



Entity Page:

Once the admin clicks on one of these entities, the table entries are listed and they get crud operations list they can choose from.

Admin page

Plant_Species

Plant_Species Table

| | 0 | 1 | 2 | 3 |
|---|---|-----------|-------------------|------------|
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |

CRUD Operations:

Select CRUD Operation:

- ☒ Create
- ☐ Read(Display)
- ☐ Update
- ☐ Delete

CRUD implementation has been provided for tables.

User page:

User is allowed to view and query any of these tables

User page

Plant_Species

Genome

Chromosome

Gene

Allele

DNA_Sequence

Genetic_Variation

Genotype

Protein

Entity page:

User is provided with read access and is allowed to query or custom query

Plant_Species Table

| | 0 | 1 | 2 | 3 |
|---|---|-----------|-------------------|------------|
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae |
| 3 | 4 | Red Rose | Rosa rubra | Rosaceae |
| 4 | 5 | Sunflower | Helianthus annuus | Asteraceae |
| 5 | 6 | Tulip | Tulipa gesneriana | Liliaceae |
| 6 | 7 | Red Rose | Rosa rubra | Rosaceae |
| 7 | 8 | Sunflower | Helianthus annuus | Asteraceae |
| 8 | 9 | Tulip | Tulipa gesneriana | Liliaceae |

Enter plant to query

Enter plant to query:

Custom SQL Query:

SQL query:

Run Query

Plant_query:

This functionality allows the user to extract all the genetic details of a given plant from the database.

Enter plant to query

1

Press Enter to apply

Enter plant to query:

All table data corresponding to given species id:

| | | | | |
|---|------------|-------------|-----------------|----------|
| | Species_ID | Common_Name | Scientific_Name | Family |
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |

| | | | | |
|---|-----------|------------|-------------|------------------|
| | Genome_ID | Species_ID | Genome_Size | Chromosome_Count |
| 0 | 1 | 1 | 120.50 | 12 |
| 1 | 4 | 1 | 120.50 | 12 |
| 2 | 7 | 1 | 120.50 | 12 |

| | | | | | |
|---|---------------|-----------|-------------------|--------|-------------|
| | Chromosome_ID | Genome_ID | Chromosome_Number | Length | Genes_Count |
| 0 | 1 | 1 | 1 | 20.30 | 4 |
| 1 | 2 | 1 | 2 | 18.50 | 120 |
| 2 | 7 | 1 | 1 | 20.30 | 150 |
| 3 | 8 | 1 | 2 | 18.50 | 120 |
| 4 | 13 | 1 | 1 | 20.30 | 150 |
| 5 | 14 | 1 | 2 | 18.50 | 120 |

| | | | | | | |
|---------|---------------|-------------|------------------|----------------|--------------|--------|
| Gene_ID | Chromosome_ID | Gene_Name | Gene_Description | Start_Position | End_Position | Strand |
| 1 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + |
| 7 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + |
| 13 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + |
| 19 | 1 | test | test | 1 | 10 | + |
| 20 | 1 | test1 | test1 | 3 | 4 | + |
| 2 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - |
| 8 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - |
| 14 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - |

| | | | | | |
|---|-----------|---------|---------------|-------------|---------------|
| | Allele_ID | Gene_ID | Allele_Name | Allele_Type | Mutation_Type |
| 0 | 1 | 1 | RUBRA_allele1 | TypeA | MutationA |
| 1 | 2 | 2 | RUBRA_allele2 | TypeB | MutationB |
| 2 | 7 | 1 | RUBRA_allele1 | TypeA | MutationA |
| 3 | 8 | 2 | RUBRA_allele2 | TypeB | MutationB |
| 4 | 13 | 1 | RUBRA_allele1 | TypeA | MutationA |
| 5 | 14 | 2 | RUBRA_allele2 | TypeB | MutationB |

| Variation_ID | Allele_ID | Variation_Type | Variation_Description | Position | Reference_Sequence | Alterec |
|--------------|-----------|----------------|--------------------------------|----------|--------------------|---------|
| 1 | 1 | SNP | Single Nucleotide Polymorphism | 150 | G | A |
| 7 | 1 | SNP | Single Nucleotide Polymorphism | 150 | G | A |
| 13 | 1 | SNP | Single Nucleotide Polymorphism | 150 | G | A |
| 2 | 2 | Insertion | Insertion Mutation | 300 | C | CGT |
| 8 | 2 | Insertion | Insertion Mutation | 300 | C | CGT |
| 14 | 2 | Insertion | Insertion Mutation | 300 | C | CGT |

| | Sequence_ID | Gene_ID | Sequence_Data | Sequence_Length |
|---|-------------|---------|---------------|-----------------|
| 0 | 1 | 1 | ATCGATCGATCG | 12 |
| 1 | 2 | 2 | GCTAGCTAGCTA | 12 |
| 2 | 7 | 1 | ATCGATCGATCG | 12 |
| 3 | 8 | 2 | GCTAGCTAGCTA | 12 |
| 4 | 13 | 1 | ATCGATCGATCG | 12 |
| 5 | 14 | 2 | GCTAGCTAGCTA | 12 |

| | Protein_ID | Protein_Name | Protein_Description | Gene_ID |
|---|------------|--------------|-------------------------|---------|
| 0 | 1 | Alpha-1 | Alpha Protein Variant 1 | 1 |
| 1 | 2 | Beta-1 | Beta Protein Variant 1 | 2 |

Custom Query Box:

This functionality allows users to enter any sql query to extract data of their requirement

Custom SQL Query:

SQL query:

Run Query

Trigger functionality:

When a new gene is created in Gene table, the corresponding gene_count in Chromosome table increases by 1 for the same chromosome_id

Before:

Chromosome Table

| | 0 | 1 | 2 | 3 | 4 |
|----|----|---|---|-------|-----|
| 0 | 1 | 1 | 1 | 20.30 | 4 |
| 1 | 2 | 1 | 2 | 18.50 | 120 |
| 2 | 3 | 2 | 1 | 15.70 | 100 |
| 3 | 4 | 2 | 2 | 14.20 | 80 |
| 4 | 5 | 3 | 1 | 12.80 | 110 |
| 5 | 6 | 3 | 2 | 11.50 | 90 |
| 6 | 7 | 1 | 1 | 20.30 | 150 |
| 7 | 8 | 1 | 2 | 18.50 | 120 |
| 8 | 9 | 2 | 1 | 15.70 | 100 |
| 9 | 10 | 2 | 2 | 14.20 | 80 |
| 10 | 11 | 3 | 1 | 12.80 | 110 |
| 11 | 12 | 3 | 2 | 11.50 | 90 |
| 12 | 13 | 1 | 1 | 20.30 | 150 |
| 13 | 14 | 1 | 2 | 18.50 | 120 |
| 14 | 15 | 2 | 1 | 15.70 | 100 |
| 15 | 16 | 2 | 2 | 14.20 | 80 |
| 16 | 17 | 3 | 1 | 12.80 | 110 |
| 17 | 18 | 3 | 2 | 11.50 | 90 |

Creating a gene of chromosome id 1

Chromosome ID:

1

Gene Name:

test2

Gene Description:

test2

Start Position:

1

End Position:

10

Strand:

+ Press Enter to apply

Create

After:

Chromosome Table

| | 0 | 1 | 2 | 3 | 4 |
|----|----|---|---|-------|-----|
| 0 | 1 | 1 | 1 | 20.30 | 5 |
| 1 | 2 | 1 | 2 | 18.50 | 120 |
| 2 | 3 | 2 | 1 | 15.70 | 100 |
| 3 | 4 | 2 | 2 | 14.20 | 80 |
| 4 | 5 | 3 | 1 | 12.80 | 110 |
| 5 | 6 | 3 | 2 | 11.50 | 90 |
| 6 | 7 | 1 | 1 | 20.30 | 150 |
| 7 | 8 | 1 | 2 | 18.50 | 120 |
| 8 | 9 | 2 | 1 | 15.70 | 100 |
| 9 | 10 | 2 | 2 | 14.20 | 80 |
| 10 | 11 | 3 | 1 | 12.80 | 110 |
| 11 | 12 | 3 | 2 | 11.50 | 90 |
| 12 | 13 | 1 | 1 | 20.30 | 150 |
| 13 | 14 | 1 | 2 | 18.50 | 120 |
| 14 | 15 | 2 | 1 | 15.70 | 100 |
| 15 | 16 | 2 | 2 | 14.20 | 80 |
| 16 | 17 | 3 | 1 | 12.80 | 110 |

Other Functionalities have been lited with Queries

PROCEDURES:

GetPlantsDetails()

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetPlantDetails`(IN
    plant_species_id INT)
BEGIN
    -- Query to retrieve plant species information
    SELECT * FROM Plant_Species WHERE Species_ID = plant_species_id;

    -- Query to retrieve Genome information based on the Species_ID
    SELECT * FROM Genome WHERE Species_ID = plant_species_id;

    -- Query to retrieve Chromosome information
    SELECT C.*
    FROM Chromosome C
    JOIN Genome G ON C.Genome_ID = G.Genome_ID
    WHERE G.Species_ID = plant_species_id;

    -- Query to retrieve Gene information
    SELECT G.*
    FROM Gene G
    JOIN Chromosome C ON G.Chromosome_ID = C.Chromosome_ID
    JOIN Genome G2 ON C.Genome_ID = G2.Genome_ID
    WHERE G2.Species_ID = plant_species_id;

    -- Query to retrieve Allele information
    SELECT A.*
    FROM Allele A
    JOIN Gene G ON A.Gene_ID = G.Gene_ID
    JOIN Chromosome C ON G.Chromosome_ID = C.Chromosome_ID
    JOIN Genome G2 ON C.Genome_ID = G2.Genome_ID
    WHERE G2.Species_ID = plant_species_id;

    -- Query to retrieve Genetic Variation information
    SELECT GV.*
    FROM Genetic_Variation GV
    JOIN Allele A ON GV.Allele_ID = A.Allele_ID
    JOIN Gene G ON A.Gene_ID = G.Gene_ID
    JOIN Chromosome C ON G.Chromosome_ID = C.Chromosome_ID
    JOIN Genome G2 ON C.Genome_ID = G2.Genome_ID
```

```

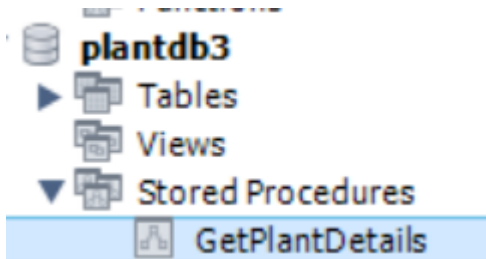
WHERE G2.Species_ID = plant_species_id;

-- Query to retrieve DNA Sequence information
SELECT DS.*
FROM DNA_Sequence DS
JOIN Gene G ON DS.Gene_ID = G.Gene_ID
JOIN Chromosome C ON G.Chromosome_ID = C.Chromosome_ID
JOIN Genome G2 ON C.Genome_ID = G2.Genome_ID
WHERE G2.Species_ID = plant_species_id;

SELECT P.*
FROM Protein P
JOIN Gene G ON P.Gene_ID = G.Gene_ID
JOIN Chromosome C ON G.Chromosome_ID = C.Chromosome_ID
JOIN Genome G2 ON C.Genome_ID = G2.Genome_ID
WHERE G2.Species_ID = plant_species_id;
END

```

GetPlantDetails():



Invoking procedure:

```
def plant_species_query(plant_species_id):

    cursor.callproc("GetPlantDetails", [plant_species_id])
    conn.commit()
    results_iterator = cursor.stored_results()

    for result_set in results_iterator:
        result_data = result_set.fetchall()
        #st.write(result_data)
        #st.write(result_set.description)

        if result_data:
            columns = [desc[0] for desc in result_set.description]
            result_df = pd.DataFrame(result_data, columns=columns)
            st.table(result_df)
        else:
            st.write("No data found for the specified argument.")
```

JOIN QUERIES

```
--Retrieve Plant Species with Genome Information
SELECT *
FROM Plant_Species
INNER JOIN Genome ON Plant_Species.Species_ID = Genome.Species_ID;
```

Custom SQL Query:

SQL query:

```
SELECT *
FROM Plant_Species
INNER JOIN Genome ON Plant_Species.Species_ID = Genome.Species_ID;
```

Run Query

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|-----------|-------------------|------------|---|---|--------|----|
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae | 1 | 1 | 120.50 | 12 |
| 1 | 2 | Sunflower | Helianthus annuus | Asteraceae | 2 | 2 | 90.20 | 8 |
| 2 | 3 | Tulip | Tulipa gesneriana | Liliaceae | 3 | 3 | 75.80 | 10 |
| 3 | 1 | Red Rose | Rosa rubra | Rosaceae | 4 | 1 | 120.50 | 12 |
| 4 | 2 | Sunflower | Helianthus annuus | Asteraceae | 5 | 2 | 90.20 | 8 |
| 5 | 3 | Tulip | Tulipa gesneriana | Liliaceae | 6 | 3 | 75.80 | 10 |
| 6 | 1 | Red Rose | Rosa rubra | Rosaceae | 7 | 1 | 120.50 | 12 |
| 7 | 2 | Sunflower | Helianthus annuus | Asteraceae | 8 | 2 | 90.20 | 8 |
| 8 | 3 | Tulip | Tulipa gesneriana | Liliaceae | 9 | 3 | 75.80 | 10 |

```
--Get Gene Details with Associated Alleles
SELECT *
FROM Gene
LEFT JOIN Allele ON Gene.Gene_ID = Allele.Gene_ID;
```

Custom SQL Query:

SQL query:

```
SELECT *
FROM Gene
LEFT JOIN Allele ON Gene.Gene_ID = Allele.Gene_ID;
```



Run Query

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|--------------|------------------|-----|-----|---|---------|--------|----------------|
| 0 | 1 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + | 1.0000 | 1.0000 | RUBRA_allele1 |
| 1 | 1 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + | 7.0000 | 1.0000 | RUBRA_allele1 |
| 2 | 1 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + | 13.0000 | 1.0000 | RUBRA_allele1 |
| 3 | 2 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - | 2.0000 | 2.0000 | RUBRA_allele2 |
| 4 | 2 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - | 8.0000 | 2.0000 | RUBRA_allele2 |
| 5 | 2 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - | 14.0000 | 2.0000 | RUBRA_allele2 |
| 6 | 3 | 3 | ANNUUS_gene1 | Sunflower Gene 1 | 50 | 250 | + | 3.0000 | 3.0000 | ANNUUS_allele1 |
| 7 | 3 | 3 | ANNUUS_gene1 | Sunflower Gene 1 | 50 | 250 | + | 9.0000 | 3.0000 | ANNUUS_allele1 |

```
--Fetch Genetic Variations with Corresponding Alleles
SELECT *
FROM Genetic_Variation
INNER JOIN Allele ON Genetic_Variation.Allele_ID = Allele.Allele_ID;
--Fetch Genetic Variations with Corresponding Alleles
SELECT *
FROM Genetic_Variation
INNER JOIN Allele ON Genetic_Variation.Allele_ID = Allele.Allele_ID;
```

Custom SQL Query:

SQL query:

```
SELECT *
FROM Genetic_Variation
INNER JOIN Allele ON Genetic_Variation.Allele_ID = Allele.Allele_ID;
```



Run Query

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|----|---|-----------|--------------------------------|-----|-----|-----|---|---|--------------------|-------|--------|
| 0 | 1 | 1 | SNP | Single Nucleotide Polymorphism | 150 | G | A | 1 | 1 | RUBRA_allele1 | TypeA | Mutati |
| 1 | 2 | 2 | Insertion | Insertion Mutation | 300 | C | CGT | 2 | 2 | RUBRA_allele2 | TypeB | Mutati |
| 2 | 3 | 3 | Deletion | Deletion Mutation | 80 | ATG | A | 3 | 3 | ANNUUS_allele1 | TypeC | Mutati |
| 3 | 4 | 4 | SNP | Single Nucleotide Polymorphism | 200 | T | G | 4 | 4 | ANNUUS_allele2 | TypeD | Mutati |
| 4 | 5 | 5 | Insertion | Insertion Mutation | 120 | A | ATG | 5 | 5 | GESNERIANA_allele1 | TypeE | Mutati |
| 5 | 6 | 6 | Deletion | Deletion Mutation | 250 | GA | G | 6 | 6 | GESNERIANA_allele2 | TypeF | Mutati |
| 6 | 7 | 1 | SNP | Single Nucleotide Polymorphism | 150 | G | A | 1 | 1 | RUBRA_allele1 | TypeA | Mutati |
| 7 | 8 | 2 | Insertion | Insertion Mutation | 300 | C | CGT | 2 | 2 | RUBRA_allele2 | TypeB | Mutati |
| 8 | 9 | 3 | Deletion | Deletion Mutation | 80 | ATG | A | 3 | 3 | ANNUUS_allele1 | TypeC | Mutati |
| 9 | 10 | 4 | SNP | Single Nucleotide Polymorphism | 200 | T | G | 4 | 4 | ANNUUS_allele2 | TypeD | Mutati |

```
-- Obtain Genotype Information and Linked Alleles
SELECT *
FROM Genotype
LEFT JOIN Allele AS A1 ON Genotype.Allele_1 = A1.Allele_ID
LEFT JOIN Allele AS A2 ON Genotype.Allele_2 = A2.Allele_ID;
```

Custom SQL Query:

SQL query:

```
SELECT *
FROM Genotype
LEFT JOIN Allele AS A1 ON Genotype.Allele_1 = A1.Allele_ID
LEFT JOIN Allele AS A2 ON Genotype.Allele_2 = A2.Allele_ID;
```

Run Query

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|--------------------|-------|-----------|---|---|--------------------|-------|----|
| 0 | 1 | 1 | 2 | 1 | 1 | RUBRA_allele1 | TypeA | MutationA | 2 | 2 | RUBRA_allele2 | TypeB | 1 |
| 1 | 2 | 3 | 4 | 3 | 3 | ANNUUS_allele1 | TypeC | MutationC | 4 | 4 | ANNUUS_allele2 | TypeD | 1 |
| 2 | 3 | 5 | 6 | 5 | 5 | GESNERIANA_allele1 | TypeE | MutationE | 6 | 6 | GESNERIANA_allele2 | TypeF | 1 |
| 3 | 4 | 1 | 2 | 1 | 1 | RUBRA_allele1 | TypeA | MutationA | 2 | 2 | RUBRA_allele2 | TypeB | 1 |
| 4 | 5 | 3 | 4 | 3 | 3 | ANNUUS_allele1 | TypeC | MutationC | 4 | 4 | ANNUUS_allele2 | TypeD | 1 |
| 5 | 6 | 5 | 6 | 5 | 5 | GESNERIANA_allele1 | TypeE | MutationE | 6 | 6 | GESNERIANA_allele2 | TypeF | 1 |
| 6 | 7 | 1 | 2 | 1 | 1 | RUBRA_allele1 | TypeA | MutationA | 2 | 2 | RUBRA_allele2 | TypeB | 1 |
| 7 | 8 | 3 | 4 | 3 | 3 | ANNUUS_allele1 | TypeC | MutationC | 4 | 4 | ANNUUS_allele2 | TypeD | 1 |
| 8 | 9 | 5 | 6 | 5 | 5 | GESNERIANA_allele1 | TypeE | MutationE | 6 | 6 | GESNERIANA_allele2 | TypeF | 1 |

```
--Retrieve Protein Details with Corresponding Genes
SELECT *
FROM Protein
INNER JOIN Gene ON Protein.Gene_ID = Gene.Gene_ID;
```


SQL query:

```
SELECT *
FROM Protein
INNER JOIN Gene ON Protein.Gene_ID = Gene.Gene_ID;
```



Run Query

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|-----------|---------------------------|---|---|---|------------------|------------------|-----|-----|----|
| 0 | 1 | Alpha-1 | Alpha Protein Variant 1 | 1 | 1 | 1 | RUBRA_gene1 | Red Rose Gene 1 | 100 | 300 | + |
| 1 | 2 | Beta-1 | Beta Protein Variant 1 | 2 | 2 | 2 | RUBRA_gene2 | Red Rose Gene 2 | 200 | 400 | - |
| 2 | 3 | Gamma-1 | Gamma Protein Variant 1 | 3 | 3 | 3 | ANNUUS_gene1 | Sunflower Gene 1 | 50 | 250 | + |
| 3 | 4 | Delta-1 | Delta Protein Variant 1 | 4 | 4 | 4 | ANNUUS_gene2 | Sunflower Gene 2 | 150 | 350 | - |
| 4 | 5 | Epsilon-1 | Epsilon Protein Variant 1 | 5 | 5 | 5 | GESNERIANA_gene1 | Tulip Gene 1 | 80 | 280 | + |
| 5 | 6 | Zeta-1 | Zeta Protein Variant 1 | 6 | 6 | 6 | GESNERIANA_gene2 | Tulip Gene 2 | 180 | 380 | - |

AGGREGATE QUERIES:

```
--Count the Number of Plant Species:  
SELECT COUNT(*) AS Total_Plant_Species  
FROM Plant_Species;
```

Custom SQL Query:

SQL query:

```
SELECT COUNT(*) AS Total_Plant_Species  
FROM Plant_Species;
```



Run Query

| | |
|---|---|
| | 0 |
| 0 | 9 |

```
--Calculate the Average Genome Size:  
SELECT AVG(Genome_Size) AS Average_Genome_Size  
FROM Genome;
```

SQL query:

```
SELECT AVG(Genome_Size) AS Average_Genome_Size  
FROM Genome;
```



Run Query

| | |
|---|-----------|
| | 0 |
| 0 | 95.500000 |

```
-- Find the Maximum Chromosome Count:  
SELECT MAX(Chromosome_Count) AS Max_Chromosome_Count  
FROM Genome;
```

Custom SQL Query:

SQL query:

```
SELECT MAX(Chromosome_Count) AS Max_Chromosome_Count
FROM Genome;
```

Run Query

| | |
|---|----|
| | 0 |
| 0 | 12 |

```
--Calculate the Total Genes Count:
SELECT SUM(Genes_Count) AS Total_Genes_Count
FROM Chromosome;
```

Custom SQL Query:

SQL query:

```
SELECT SUM(Genes_Count) AS Total_Genes_Count
FROM Chromosome;
```

Run Query

| | |
|---|-------|
| | 0 |
| 0 | 1715. |

```
-- Count the Number of Alleles per Gene:
SELECT Gene.Gene_ID, Gene.Gene_Name, COUNT(Allele.Allele_ID) AS
    Num_Of_Alleles
FROM Gene
LEFT JOIN Allele ON Gene.Gene_ID = Allele.Gene_ID
GROUP BY Gene.Gene_ID, Gene.Gene_Name;
```

Custom SQL Query:

SQL query:

```
SELECT Gene.Gene_ID, Gene.Gene_Name, COUNT(Allele.Allele_ID) AS Num_Of_Alleles
FROM Gene
LEFT JOIN Allele ON Gene.Gene_ID = Allele.Gene_ID
GROUP BY Gene.Gene_ID, Gene.Gene_Name;
```

Run Query

| | | | |
|----|----|------------------|---|
| | 0 | 1 | 2 |
| 0 | 1 | RUBRA_gene1 | 3 |
| 1 | 2 | RUBRA_gene2 | 3 |
| 2 | 3 | ANNUUS_gene1 | 3 |
| 3 | 4 | ANNUUS_gene2 | 3 |
| 4 | 5 | GESNERIANA_gene1 | 3 |
| 5 | 6 | GESNERIANA_gene2 | 3 |
| 6 | 7 | RUBRA_gene1 | 0 |
| 7 | 8 | RUBRA_gene2 | 0 |
| 8 | 9 | ANNUUS_gene1 | 0 |
| 9 | 10 | ANNUUS_gene2 | 0 |
| 10 | 11 | GESNERIANA_gene1 | 0 |
| 11 | 12 | GESNERIANA_gene2 | 0 |
| 12 | 13 | RUBRA_gene1 | 0 |

```
--Find the Number of Genetic Variations per Allele:
SELECT Allele.Allele_ID, Allele.Allele_Name,
COUNT(Genetic_Variation.Variation_ID) AS Num_Of_Variations
FROM Allele
LEFT JOIN Genetic_Variation ON Allele.Allele_ID = Genetic_Variation.Allele_ID
GROUP BY Allele.Allele_ID, Allele.Allele_Name;
```

Custom SQL Query:

SQL query:

```
SELECT Allele.Allele_ID, Allele.Allele_Name, COUNT(Genetic_Variation.Variation_ID) AS
Num_Of_Variations
FROM Allele
LEFT JOIN Genetic_Variation ON Allele.Allele_ID = Genetic_Variation.Allele_ID
```

Run Query

| | | | |
|----|----|--------------------|---|
| | 0 | 1 | 2 |
| 0 | 1 | RUBRA_allele1 | 3 |
| 1 | 2 | RUBRA_allele2 | 3 |
| 2 | 3 | ANNUUS_allele1 | 3 |
| 3 | 4 | ANNUUS_allele2 | 3 |
| 4 | 5 | GESNERIANA_allele1 | 3 |
| 5 | 6 | GESNERIANA_allele2 | 3 |
| 6 | 7 | RUBRA_allele1 | 0 |
| 7 | 8 | RUBRA_allele2 | 0 |
| 8 | 9 | ANNUUS_allele1 | 0 |
| 9 | 10 | ANNUUS_allele2 | 0 |
| 10 | 11 | GESNERIANA_allele1 | 0 |
| 11 | 12 | GESNERIANA_allele2 | 0 |
| 12 | 13 | RUBRA_allele1 | 0 |

```
--Get the Maximum and Minimum Sequence Length per Gene
SELECT Gene.Gene_ID, Gene.Gene_Name, MAX(DNA_Sequence.Sequence_Length) AS
Max_Length, MIN(DNA_Sequence.Sequence_Length) AS Min_Length
FROM Gene
LEFT JOIN DNA_Sequence ON Gene.Gene_ID = DNA_Sequence.Gene_ID
GROUP BY Gene.Gene_ID, Gene.Gene_Name;
```

Custom SQL Query:

SQL query:

```
SELECT Gene.Gene_ID, Gene.Gene_Name, MAX(DNA_Sequence.Sequence_Length) AS Max_Length,
MIN(DNA_Sequence.Sequence_Length) AS Min_Length
FROM Gene
LEFT JOIN DNA_Sequence ON Gene.Gene_ID = DNA_Sequence.Gene_ID
GROUP BY Gene.Gene_ID, Gene.Gene_Name;
```

Run Query

| | 0 | 1 | 2 | 3 |
|---|----|------------------|---------|---------|
| 0 | 1 | RUBRA_gene1 | 12.0000 | 12.0000 |
| 1 | 2 | RUBRA_gene2 | 12.0000 | 12.0000 |
| 2 | 3 | ANNUUS_gene1 | 12.0000 | 12.0000 |
| 3 | 4 | ANNUUS_gene2 | 12.0000 | 12.0000 |
| 4 | 5 | GESNERIANA_gene1 | 12.0000 | 12.0000 |
| 5 | 6 | GESNERIANA_gene2 | 12.0000 | 12.0000 |
| 6 | 7 | RUBRA_gene1 | <NA> | <NA> |
| 7 | 8 | RUBRA_gene2 | <NA> | <NA> |
| 8 | 9 | ANNUUS_gene1 | <NA> | <NA> |
| 9 | 10 | ANNUUS_gene2 | <NA> | <NA> |

NESTED QUERIES:

```
--for Filtering Data to retrieve plant species that have genomes larger than
100:
SELECT *
FROM Plant_Species
WHERE Species_ID IN (
    SELECT Species_ID
    FROM Genome
    WHERE Genome_Size > 100
);
```

Custom SQL Query:

SQL query:

```
FROM Plant_Species
WHERE Species_ID IN (
    SELECT Species_ID
    FROM Genome
    WHERE Genome_Size > 100
```

Run Query

| | | | | |
|---|---|----------|------------|----------|
| | 0 | 1 | 2 | 3 |
| 0 | 1 | Red Rose | Rosa rubra | Rosaceae |

```
--This finds plant species that have more than one genome entry.
SELECT Species_ID, Common_Name
FROM Plant_Species
WHERE Species_ID IN (
    SELECT Species_ID
    FROM Genome
    GROUP BY Species_ID
    HAVING COUNT(*) > 1
);
```

Custom SQL Query:

QL query:

```
SELECT Species_ID, Common_Name
FROM Plant_Species
WHERE Species_ID IN (
  SELECT Species_ID
FROM Genome
```

Run Query

| | | |
|---|---|-----------|
| | 0 | 1 |
| 0 | 1 | Red Rose |
| 1 | 2 | Sunflower |
| 2 | 3 | Tulip |

```
--This fetches plant species with genome sizes greater than the average
  genome size across all species.
SELECT Common_Name, Genome_Size
FROM Plant_Species
INNER JOIN Genome ON Plant_Species.Species_ID = Genome.Species_ID
WHERE Genome_Size > (
  SELECT AVG(Genome_Size)
FROM Genome
);
```

Custom SQL Query:

SQL query:

```
SELECT Common_Name, Genome_Size
FROM Plant_Species
INNER JOIN Genome ON Plant_Species.Species_ID = Genome.Species_ID
WHERE Genome_Size > (
  SELECT AVG(Genome Size)
```

Run Query

| | | |
|---|----------|--------|
| | 0 | 1 |
| 0 | Red Rose | 120.50 |

```
--This query retrieves plant species that have entries in the Genome table.
```



```
SELECT Common_Name
FROM Plant_Species AS P
WHERE EXISTS (
  SELECT 1
  FROM Genome AS G
  WHERE G.Species_ID = P.Species_ID
);
```

Custom SQL Query:

SQL query:

```
SELECT Common_Name
FROM Plant_Species AS P
WHERE EXISTS (
  SELECT 1
  FROM Genome AS G
```

Run Query

| | |
|---|-----------|
| | 0 |
| 0 | Red Rose |
| 1 | Sunflower |
| 2 | Tulip |

Triggers:

Screenshots of this have been inserted

```
CREATE DEFINER=`root`@`localhost` TRIGGER `gene_BEFORE_INSERT` BEFORE INSERT
  ON `gene` FOR EACH ROW BEGIN
  DECLARE chromosome_id_var INT;

  -- Get the chromosome_id of the chromosome associated with the new gene
  SET chromosome_id_var = NEW.Chromosome_ID;

  -- Update the Genes_Count in Chromosome table
  UPDATE Chromosome
  SET Genes_Count = (SELECT COUNT(*) FROM Gene WHERE Chromosome_ID =
    chromosome_id_var)
  WHERE Chromosome_ID = chromosome_id_var;
END
```