PostgreSQL Views Exercise: Managing Paris Tree Data

Scenario

You are a data analyst working for the Paris Parks and Gardens Department. Your team is responsible for maintaining and analyzing the extensive database of trees in Paris. To streamline common queries and improve data accessibility for various stakeholders, you've been tasked with creating and managing views based on the existing trees database.

Exercise Tasks

1. Create a Basic View

Create a view named `v_tree_info` that combines basic information about trees, including their ID, height, circumference, and location details.

```sql
CREATE VIEW v_tree_info AS
<your query>
```

Verify the view

```sql
SELECT * FROM v_tree_info LIMIT 5;
```

2. List All Views

List all views in the current database to confirm the creation of `v_tree_info`.

```sql
SELECT table_name
FROM information_schema.views
WHERE table_schema = 'public';
```

You can also simply use `\dv`

3. Create a Filtered View

Create a view named `v_remarkable_trees` that shows only remarkable trees with their taxonomy (name, genrre, species, variety) information. You can use a CTE to first get the taxonomy columns and then join on trees and locations.

```SQL
CREATE VIEW v_remarkable_trees AS
<your query>
```

Query the view

```SQL
SELECT * FROM v_remarkable_trees LIMIT 5;
```

4. Use a View in a SELECT Statement

Write a query that uses the `v_tree_info` view to find the average height of trees in each arrondissement.

```SQL
<your query>
```

5. Drop a View

Drop the `v_tree_info` view and verify that it has been removed.

```SQL
DROP VIEW IF EXISTS v_tree_info;

-- Verify that the view has been dropped
SELECT table_name
FROM information_schema.views
WHERE table_schema = 'public';
```