Geographic Information Systems 2022-2023

Exercise 4 - Organizing geographic data

Introduction

This exercise will use a dataset made available by the Lisbon City Council open data initiative. This corresponds to the with the location of trees in streets and public spaces in the city of Lisbon.

The goals of these exercise are:

> - Organise a dataset according to the rules of the Normal Forms (NF)
> - Create joins between geographic datasets and tables and visualize in your GIS

What do you need:

Two datasets:

- dataset with tree data (details bellow)
- the administrative map of Portugal, version 2012, downloaded at https://www.dgterritorio.gov.pt/sites/default/files/ficheiros-cartografia/Cont_AAd_CAOP20121.zip.

Dataset with tree data

The geographic dataset is available on the Lisboa Aberta platform.

In the platform, if you make a search for the keyword `arvoredo`, this will lead to a first result with metadata about the resource. To download as `csv`, you can click on the link http://geodados.cm-lisboa.pt/datasets/arvoredo.

For this exercise, we are making available two versions of the dataset in csv format, in portuguese and in english, to easier understanding of the data. These files were also cleaned for data inconsistencies. These are provided in Fenix for this exercise.

The **metadata** of the dataset is the following:

Item	Description
Dataset name	Arvoredo
source:	https://geodados-cml.hub.arcgis.com/search?q=arvoredo
description	Map service locating trees in Lisbon.
dataset schema	https://geodados-cml.hub.arcgis.com/datasets/CML::arvoredo/about

The downloaded table contains the following columns:

Column	Description

Column	Description
X	Longitude coordinate, in geographic format, in WGS84 reference system
Y	Latitude coordinate, in geographic format, in WGS84 reference system
OBJECTID	Unique identifier of the geographic feature
COD_SIG_NEW	Other unique identifier of the geographic feature
MORADA	Address of the location of the tree
ESPECIE_VA	Scientific name of the tree species
PAP	circumference at chest height
MANUTENCAO	Entiy responsible for the maintenance
OCUPACAO	type of occupation
LOCAL	type of place
TIPOLOGIA	type of implantation of the tree
FREG_2012	Name of the freguesia
NOME_VULGA	common name of the tree
GlobalID	global unique identifier

Submission of the exercise

You should submit your exercise report by email to `ruifigueira@isa.ulisboa.pt` until next class on 6th March 2023. The report should contain responses to parts of this exercise marked as **QUESTION**

Task 01 - Map the dataset in GIS

- Import and map the dataset in your GIS platform:

> **QGIS**
>
> - Create a new project named `Ex04_arvoredo`, and create **DataIn** and **DataOut** folders inside the project folder
> - Import the table with the menu **Layer > Add Layer > Add Delimited Text Layer**
> - File format: `CSV (comma separated values)`
> - Check `First record has field names`
> - Check: `Detect field types`
> - Geometry definition: `Point coordinates`, X field: `X`, Y field: `Y`
> - CRS: EPSG:4326 - WGS 84

> **ArcGIS**
>
> - Create a new project named `Ex04_arvoredo`. Using file explorer, create **DataIn** and **DataOut** folders inside the project folder, and place the csv file inside the **DataIn** folder
> - In ArcGIS, add the csv to the current map, with the tool **Map > Add Data > XY Point Data**

> - X field: X, Y field: Y
> - CRS: GCS_WGS_1984

- Open the attribute table schema:

> **QGIS**
>
> - Right-click on the layer and select Properties > Fields

> **ArcGIS**
>
> - Right-click on the layer and select Data Design > Fields

- Compare the table schema of the imported layer with schema at the source (see metadata above).
- Analyse the data columns and identify possible duplications and dependencies to be solved via data normalization

You can close your GIS application.

Task 02 - transform the dataset to a normalised format, according to the 3rd normal form

We will start to create a 3rd NF dataset from the provided csv file. We will use for this a spreadsheet application like Excel or Libreoffice. We suggest the following path to achieve that goal:

Step 1 - analyse you problem and create a diagram

You should study and plan for the data transformations required:

- analyze the original table to identify the different entities that might be present
- identify new tables that need to be created for each of the identified entities, in order to solve partial or transitive dependencies
- write down a text representation of the new table schemas, identifying the primary key. The text notation for the table schema representation is:

  ```
  <table name>(<attrib_01>, <attrib_02>, ...,  <attrib_n>)
  ```

  Underline with a solid line, or a dashed line, the primary and foreign keys, respectively.
- Rewrite the schema the original table, replacing, when applicable, the original column by the corresponding foreign key

QUESTION 1: provide the table schema in text notation for all the tables that you identified or created

Step 2 - perform data transformations

Once you're confident with the solution achieved, make data transformations:

- make a copy of the original file to make transformations. You should always keep the original data for reference or recovery.
- create a new table for each of these entities, removing duplicate rows

    - you can use pivot tables or data filters of your spreadsheet software. The specific method depends if your're using Excel, Libreoffice or OpenOffice, Google Sheets, MacOS Numbers or other.
    - you can also use OpenRefine for this purpose
- add unique identifiers to each of the new tables. This can be simple a sequential integer number
- add a new column in the original table to contain the IDs that you created for each of the new tables. These will be *foreign keys*.
- use `VLOOKUP` function of your spreadsheet software to fill in these columns with the unique identifiers. You have a tutorial on this function here. Do not forget to lock the table array cell references with `$`
- make IDs permanent with a **copy --> paste as values** action
- delete the column of the parameter that is replaced by its ID.
- **save as** each of the tables created as csv files.
- place all csv files in the **DataIn** folder of your project

At this point, you should have the transformed original table normalised in csv format (we will call it `arvoredo_norm`), and additional csv files for each of the new tables created for each entity.

QUESTION 2: Include in the report screenshots of the first ten rows of each table you created. The names of the columns should be visible and correspond to the text schema included in Question 1.

Task 03 - Add normalised tables to your GIS project

Open your GIS project and remove from the contents the layer previously imported. We will replace it by the newly created tables.

- add the normalised `arvoredo_norm` table, and mapped as done before
- add the remaining normalised data tables to your project. In this case, you do not need to define XY fields, as these tables do not contain coordinates
- create table relations between `arvoredo_norm` and the other tables

> **QGIS**
>
> - Right-click on the layer `arvoredo_norm` and select **Properties > Joins**
> - Click on the *green plus* button on the bottom-left corner of the panel, to add a new Vector Join
>   - Define:
>   - Join layer - the table you want to join
>   - Join field - the primary key of table to join
>   - target field - the foreign key on the `arvoredo_norm` layer
> - Repeat this operation for the other tables to be joined

> **ArcGIS**
>
> - Right-click on the layer `arvoredo_norm` and select **Joins and Relates > Add Join**
> - Define:
>   - Input Join Field - the foreign key on the `arvoredo_norm` layer
>   - Join table - the table you want to join
>   - Join Table Field - the primary key of table to join
>   - Optionally, click on *Validate Join* button and check the log report

> - Repeat this operation for the other tables to be joined

Task 04 - Perform analysis with related data

4.1. Create layer with freguesias of Lisboa

- Unzip and add the geographic dataset of the administrative map of Portugal to the GIS project
- Make a **Select by attribute** operation to select all freguesias that belong to the *Municipio* (*municipality*) `Lisboa`
- Export the selected features to a new geographic dataset named `Lisboa_freg`, and add it to the project

4.2. Join the attributes based on spatial location Although the table of `arvoredo_norm` has an attribute freguesia, it was not verified for error or consistency issues. For this reason, we will add the name of the freguesia to `arvoredo_norm` based on a **Spatial Join** operation.

> **QGIS**
>
> - Make a **Join by location** operation with the menu **Vector > Data Management Tools > Join Attributes by Location**
>   - Joint to features in: `arvoredo_norm`
>   - Features they: `intersect`
>   - By comparing to: `Lisboa_freg`
>   - Join type: one-to-many
>   - Joined layer: `arvoredo_freg_join`

> **ArcGIS**
>
> - Make a **Spatial Join** operation. Right-click on the layer and select **Joins and Relates > Spatial Join**
>   - Join features: `Lisboa_freg`
>   - Output feature class: `arvoredo_freg_join`
>   - Join Operation: one-to-many
>   - Match Option: `intersect`

4.3. Perform calculations

Now that we have the name of the tree species and the name of the freguesia added to the table of the trees, we can make calculations. We will use SQL to perform the queries that return the desired results

Determine the number of trees not identified by freguesia

First, make a Select by value operation, to select features that correspond to *not identified*, and then calculate the number of records by freguesia

> **QGIS**
>
> - open the attribute table of **arvoredo_freg_join**
> - use **Select by expression** to select all features which value in **ESPECIE_VA** is `Not identified`
> - Open **Processing > Vector Analysis > Select by categories**

    - input vector layer: `arvoredo_freg_join`
    - check Selected features only
    - Field(s) with categories: `Freguesia`
    - Statistics by category: `not_identified_count`

ArcGIS

- Open the attribute table of layer **arvoredo_freg_join**
- Do a Select by attributes to create a new selection with the field **ESPECIE_VA** equal to `Not identified`
- Change the view of the attribute table to show only selected features
- Right-click on the name of the column **Freguesia** and select `Summarize`
    - Field: `Freguesia`
    - Statistic Type: `Count`
- a new table will be added to the contents page with the results

QUESTION 3: How many trees with type of implantation `Canteiro` exists in freguesia `ALCÂNTARA`? Describe how did you determined it?

4.4. Use SQL queries (optional)

We can also make queries using SQL syntax. To calculate the not identified trees by freguesia (as above), but with SQL, you can do the following:

QGIS

- Open the menu **Database > DB Manager**
- in providers, select Virtual layers > Project layers > `arvoredo_freg_join`
- click on the menu of the window **Database > SQL Window**
- Add the following SQL statement and execute

```
SELECT Freguesia, count(*) as count
FROM arvoredo_Freg_Join
WHERE especie_va_ESPECIE_VA LIKE 'Não ident%'
GROUP BY Freguesia
ORDER BY count DESC
```

ArcGIS

The options to use native SQL in ArcGIS are more limited. It does not allow to define the columns of the SQL statement output, but only the **WHERE** clause.