Practical Assignment — 2021 Local Elections (Autárquicas)

Course: Programming and Databases
DB Requirement: SQLite (mandatory)
Groups: 3 students
Submit on Moodle: Jan 2 (23:59)
Viva: Oral presentations will be scheduled after submission
Reference display: a simple **clickable map of Portugal** similar to https://www.eleicoes.mai.gov.pt/autarquicas2021/resultados/territorio-nacional

---

Objective

Build an end-to-end mini-system that:

1. **ingests** official election data (CNE, 2021) from Excel into a **SQLite** database;
2. **displays** selected results in a **Python GUI** using **Tkinter** + **Matplotlib**, with a **clickable map** of Portugal.

Dataset: https://www.cne.pt/sites/default/files/dl/2021al_mapa_oficial.zip
(You may focus on a **subset**: e.g., only *Câmara Municipal* or only a few districts.)

Task A — Data understanding, schema, ETL (SQLite)

1. **Explore the Excel files** (organs: CM/AM/AF; geography: district/municipality/parish; parties/coalitions; votes, seats, participation, blanks/nulls).
2. **Design a relational schema (SQLite)** that fits your chosen scope. Keep it clear and normalized.
    - Example:
        - `district(id, name)`
        - `municipality(id, name, district_id)`
        - `party(id, sigla, name)`
        - `coalition(id, name)` + `coalition_member(coalition_id, party_id)`
        - `organ(id, name)` (e.g., CM)
        - `result(organ_id, geo_level, geo_id, holder_type, holder_id, votes, seats, total_valid, blanks, nulls, registered, turnout_pct)`
3. **ETL in Python** :
    - Clean headers, normalize names/codes, convert numbers, handle missing values.
    - Map coalitions and parties.
    - Scripts must be **re-runnable** (truncate/reload or upserts).

4. **(Optional but useful)**: create a **geometry helper table** with polygons (as **WKT TEXT**) for Portugal (e.g., districts, municipalities, parishes) to support the clickable map:
    - `district_shape(district_id, geom_wkt)`
    - You can precompute WKT outside SQLite (e.g., from PostGIS) or load a provided WKT file. **No spatial extensions are required**; treat WKT as plain text.
    - The site https://www.dgterritorio.gov.pt/cartografia/cartografia-tematica/caop as the required data for this representation.

Minimum acceptance (Task A): a working SQLite DB + ETL for at least the **Câmara Municipal** results for **Portugal or a clear geographic subset** (e.g., 2–3 districts).

Task B — GUI with Tkinter + Matplotlib (clickable map)

Build a **simple desktop GUI** inspired by the reference site:

Mandatory features

- **Main view = clickable map of Portugal** (districts or municipalities).
    - When the user clicks a region, **update** a result panel.
- **Results panel** (table + at least one chart via Matplotlib):
    - Show votes/seats by party/coalition for the selected region (CM at minimum).
- **Basic controls**: at least choose the **organ** (CM required; others optional) and allow navigating regions (e.g., district → list of municipalities).

Implementation notes

- Tkinter `Canvas` for the map; read **WKT TEXT** from SQLite, parse and draw polygons.
- Use **Matplotlib** (bar/pie) embedded in Tkinter (FigureCanvasTkAgg).
- Keep the interface **minimal and responsive**; prioritize correctness and clarity.

Deliverables (upload to Moodle by Jan 2, 23:59)

1. **Repository/Archive** containing:
    - `/db/` — **SQLite file** (e.g., `elections.db`) + **DDL** (`create_tables.sql`).
    - `/etl/` — ETL scripts and instructions.
    - `/app/` — Tkinter GUI app (`gui.py`) and assets.
    - `/docs/` — ER diagram, brief schema rationale, usage guide, screenshots.
    - `/README.md` — how to run ETL and app (one-command or clear steps).

2. **Short report (3–5 pages)**: scope chosen, schema design, ETL choices, GUI overview, known limitations.
3. **Slides (max 6–8)** for the **oral presentation**.

Marking (100%)

- **Schema & documentation (25%)** — normalization, keys, clarity, ER diagram, DDL.
- **ETL (25%)** — correctness, robustness, rerunnable, data cleaning choices.
- **Queries & correctness (10%)** — results match dataset; sensible indexes.
- **GUI (25%)** — clickable map works; table+chart accurate and update on click; UX clarity.
- **Code quality & reproducibility (15%)** — structure, readability, config separation, instructions that work.

Bonus (up to +10%)

- District→Municipality drill-down on the map.
- Comparison views (e.g., coalition vs parties, turnout analysis).
- Export current view (CSV/PNG).
- Tests for ETL helpers.

Practical tips

- **Start small**: load CM for 1–2 districts, validate, then scale.
- Prefer **stable IDs** over free text (when available). Document mapping assumptions.
- For the map, use **WKT stored as TEXT** (no SpatiaLite needed). Draw only exterior rings for speed.
- Keep your **GUI loop** simple: filters → query DB → update table + chart.

Academic integrity & teamwork

- Work in **groups of three**.
- Cite any external code snippets.
- Everyone should be able to **explain** the schema, ETL, and GUI in the oral.