

ANTBYTES GIS / LEVEL 2

From address matches to defensible GIS decisions.

A practical continuation of the Plant City + Hillsborough County 100-address exercise.

QGIS workflow · 36 review cases · Project delivery

Self-paced / intermediate / approximately 4–6 hours
Training snapshot retained from the previous exercise.

START HERE / 4–6 HOURS

Your next step: defensible GIS work

Level 1 showed how to turn addresses into points and compare them with a reference. Level 2 teaches you to decide what the evidence supports, preserve uncertainty, and deliver a project someone else can inspect.

Inspect → Review → Analyze → Deliver

Block	Time	Evidence you will produce
Set up & inspect	40 min	A QGIS project with reproducible paths and verified row counts
Adjudicate differences	90–120 min	A 36-record decision log with evidence and unresolved cases
Analyze & communicate	60 min	A projected analysis layer, summary, and branded map
Check & hand off	40 min	QA report, GeoPackage, and project README

Before you start

Use QGIS Desktop, a spreadsheet editor, and Python 3 if you want the included automated checks. Instructions use QGIS 3.44 terminology; labels can vary in other versions. You need basic layer loading and attribute-table skills from Level 1.

Work from the extracted Level 2 folder. Everything needed for the core exercises is included. Online source checks are an optional extension; write the date and source if you make one.

> **Completion standard**
>
> Finish with all 36 review records documented. Leaving a case Unresolved is acceptable when the evidence does not justify acceptance. The goal is supported decisions, not a forced 100% match rate.

DATA CONTRACT / FIXED BASELINE

Know exactly what you received

The exercise uses the completed comparison snapshot from Level 1. It is not a new address lookup. The original input contains 100 public-facility records across 46 input ZIP codes, including 9 records labeled Plant City.

File in data/	Expected content	Role
hillsborough-100-addresses.csv	100 unique record_id values	Original input; keep unchanged
hillsborough-100-e911-comparison.csv	100 rows: 64 exact, 27 review, 9 no match	Baseline outcomes
hillsborough-100-e911-points.geojson	64 Point features	Exact reference matches
hillsborough-100-e911-review-candidates.csv	79 candidate rows	Possible alternatives, not decisions
e911-review-candidates.geojson	79 Point features	Spatial view of the same candidates
hillsborough-100-e911-summary.json	Counts, timestamp, method, source URL	Provenance and reconciliation

Do not confuse the units of counting

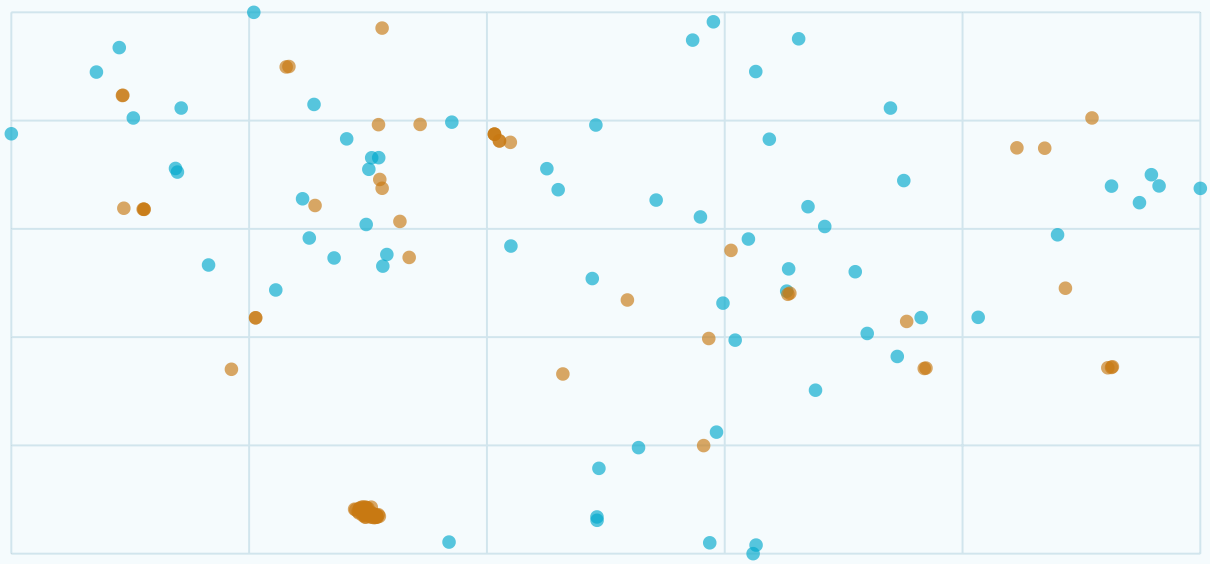
The 36 unresolved input records have 79 candidate rows between them; some have no candidate and some have many. Candidate count is not matched-address count. Six input addresses have multiple matching points, and HC-033 alone has 37 candidates.

The source is the public City of Tampa GIS county address-point service used after county feature requests failed. A Current point is a reference record; it does not independently verify the facility name, unit, entrance, or current occupancy.

> **Checkpoint A**
>
> Write these baseline totals in your notes: 100 inputs = 64 exact + 27 review + 9 no match. Within the 27: 17 address differences, 6 multiple-point cases, 3 ZIP differences, and 1 input needing cleanup.

Read the reference footprint

This diagram plots the actual coordinates supplied with the exercise. Use it to orient yourself before creating your own map.



Cyan: 64 exact reference points · Amber: 79 candidates (some overlap)

Longitude/latitude footprint only · No basemap or municipal boundaries · Not a routing map

What the diagram shows

Cyan marks represent the 64 exact reference matches. Amber marks represent 79 proposed candidates for review. Some points overlap; the visible symbol count is not a reliable count of records.

What it does not show

There is no basemap, municipal boundary, road network, or unit/entrance verification. The view uses longitude and latitude as a simple data footprint. Do not measure a distance or infer municipal membership from this diagram.

> **Try this in QGIS**
>
> Recreate the two categories from the supplied GeoJSON layers. Compare their attribute counts with the diagram legend. Then select HC-033 and inspect how many candidate rows occupy the same or nearby locations.

Build a reproducible project

1. Extract the pack to a working folder. Keep `data/` read-only in your workflow. Save your edits under `outputs/`; use the supplied `worksheets/` for decisions.
2. In QGIS, create a project and save it as `outputs/antbytes_level2.qgz`. In project properties, choose relative file paths where available. Save again after adding layers.
3. Open Data Source Manager → Delimited Text. Load the comparison CSV as **No geometry (attribute only table)**. Keep `record_id`, ZIP, and source IDs as text; inspect field types before adding.
4. Drag the two GeoJSON files into QGIS. Name the layers `reference_exact_64` and `review_candidates_79`. Their geographic coordinates are longitude/latitude in WGS 84.
5. Load `worksheets/review-decisions.csv` as an attribute-only table. It has 36 unique record IDs and starts with decision = Unresolved.
6. Open each attribute table and verify the counts. Save a screenshot of the layer panel and your count notes as `outputs/01_loaded_layers.png`.

> **Expected result**
>
> Comparison table: 100 rows. Reference points: 64. Candidate points: 79. Decision table: 36. A CSV table with zero geometry is correct; do not manufacture coordinates for records that have none.

Organize layers by purpose

Create groups named Source, Review, Analysis, and Presentation. Keep original geographic layers in Source. Later, put transformed copies in Analysis. A saved project points to files; it does not automatically embed all of them.

Reference: QGIS User Guide, Opening Data [1].

Join without losing the problem cases

A join can look successful while silently repeating or losing information. Decide which table is the target and whether the join key is unique on the side being attached.

1. Confirm `record_id` is unique in the comparison table and in the 36-row decision table. It is intentionally repeated in the candidate layer.
2. On `review_candidates_79`, open Properties → Joins and add the decision table. Use `record_id` for both fields and set the joined-field prefix to `review_`.
3. Inspect HC-033. All 37 candidate features may inherit that record's decision fields. The layer must still have 79 features.
4. Do not attach the candidate table as a normal one-to-one join to the 36-row review table and assume all alternatives were retained. Use the candidate layer as the target, a relation, or an explicit one-to-many workflow.
5. On `reference_exact_64`, join the 100-row comparison table on record_id only if you need more attributes. The point layer stays at 64; a join does not create the 36 missing geometries.

> **Checkpoint B**
>
> Explain why a 64-feature point layer can join correctly to a 100-row table yet still have 64 features. Then explain why HC-033 must not become 37 accepted input addresses.

Evidence to save

Capture the join settings and the HC-033 candidate filter. Write the before/after feature count. If a tool regenerates an internal feature ID, continue using the business key `record_id` for reconciliation.

Reference: QGIS User Guide, Joins and Relations [2].

Create an auditable decision log

Use `worksheets/review-decisions.csv` for QGIS and Python. The branded Excel workbook is a companion for organizing work and self-assessment. If you export Excel data, export only the table header and 36 data rows—not the branded title rows.

Decision	Use when	Required reasoning
Accepted	One location is supported by the evidence you recorded	Identify the point, proposed address/ZIP, coordinates, source, reviewer, and date
Rejected	You have evidence that the proposed/input association should not be used	Explain what was ruled out; a missing result alone is insufficient
Unresolved	Evidence is incomplete, conflicting, or ambiguous	State the open question and the next useful check

1. Preserve the input fields and baseline status. Make corrections in proposed fields so the original remains inspectable.
2. For a supplied candidate, record its `e911_objectid` in `candidate_objectid`. Record the proposed address and ZIP exactly as supported.
3. Keep accepted coordinates as longitude then latitude, in EPSG:4326. Leave the output-coordinate fields blank for Rejected or Unresolved decisions.
4. For a newly researched reference outside the supplied candidates, use `external:SOURCE_ID`, record a retrievable evidence URL and coordinates, and explain the additional verification.
5. For every final row, enter `decision_notes`, `evidence_source`, `reviewer`, and an ISO date such as `2026-10-23`.

> **What evidence should say**
>
> “The candidate name looks close” is weak evidence. A useful note identifies the difference, the source checked, whether the location/unit agrees, and why the remaining uncertainty is acceptable—or why it remains unresolved.

CASE STUDY A / HC-009

Investigate a street-name difference

Original: Edward Medard Park — 5726 Partner Lp, ZIP 33567

Candidate: 5726 Panther Loop, ZIP 33567 — object ID 359455

This looks like a small spelling error, but accepting a similar street blindly is a habit that fails on less obvious records.

1. Filter the comparison and candidate layers to `record_id = 'HC-009'`. Confirm that the proposed point came from this record's candidate list.
2. Inspect the candidate coordinate, full address, ZIP, point type, and the facility label in the original source record. Separate a text match from facility identity.
3. For stronger evidence, inspect the county park source or another official facility listing. If working offline, record that the candidate is plausible but independent facility confirmation is still missing.
4. Enter the supported proposed street in a proposed field. Preserve `5726 Partner Lp` in the input field.
5. Choose Accepted only if your recorded evidence supports the association. Otherwise leave Unresolved with the precise additional check needed.

> **Worked reasoning, not a preset answer**
>
> "Partner" and "Panther" are different strings. The same number and ZIP narrow the search but do not prove identity. A documented correction can be justified; silently overwriting the source input cannot.

Your deliverable

A before/after note, the candidate ID, a source citation, and a decision that follows your actual evidence. The supplied candidate is not an instructor-approved correction.

CASE STUDY B / HC-030

Resolve a ZIP discrepancy carefully

Original: 11020 Peppersong Dr — ZIP 33569

Reference candidate: 11020 Peppersong Dr — ZIP 33578

The normalized street text agrees, but the ZIP differs. A match that ignores that difference would overstate the original exact-match rate.

1. Compare the candidate’s full address and postal community with the input. Inspect the point’s location rather than accepting a city label as a boundary test.
2. Review the original source record and an appropriate current official reference if available. Record which source supports the proposed ZIP and when it was checked.
3. If justified, enter `33578` as the proposed ZIP while retaining `33569` as the original input ZIP.
4. Use the original input ZIP when measuring how well the original dataset matched. Use the proposed ZIP only in a clearly labeled corrected-address summary.
5. Explain why the result is a reviewed correction rather than an original exact match.

> **Checkpoint C**
>
> After accepting a ZIP correction, should the original baseline become 65 exact matches? No. The original snapshot remains 64 exact matches. A revised output can contain one additional reviewed acceptance, with its own denominator and label.

Keep geography precise

A postal city, ZIP, municipal boundary, planning area, and service area answer different questions. Do not infer Plant City municipal membership from a Plant City mailing label.

CASE STUDY C / HC-033 + HC-081

Handle multiple points and suites

HC-033: 2401 19th Ave NW has 37 candidate rows. **HC-081:** the input requests 7606 Paula Dr Suite 120; the supplied candidates include a base address and suites 110 and 139.

1. Filter each record separately. Inspect all alternatives, including unit text and point type. Do not assume the first row is the best row.
2. For HC-033, determine whether your intended mapping unit is a site, building, unit, or access point. The task is one defensible output per input record, not every matching candidate.
3. For HC-081, do not substitute suite 110 or 139 for suite 120. A base-address point also does not establish that suite 120 exists at that location.
4. If a justified candidate is accepted, record its object ID. After joining decisions to candidate points with prefix `review_`, filter to the selected object ID—not just the Accepted decision.
5. If no unit-level evidence is available, keep the suite case Unresolved and identify the missing evidence.

```
"review_decision" = 'Accepted'
AND to_string("e911_objectid") =
    to_string("review_candidate_objectid")
```

Common failure

If you filter only review_decision = Accepted after accepting HC-033, all 37 candidate rows may appear accepted. The object-ID condition prevents that duplication. External references need their own documented layer; they are not selected by this supplied-candidate filter.

Keep no-match and malformed records visible

HC-093 input: 10020 W Bridgeton Dr or 9791 Westch

Baseline: Input needs cleanup. No candidate was automatically accepted.

1. Recognize that the input appears to contain alternatives and truncated text. Do not guess a complete second address or invent a coordinate.
2. Inspect the source record and look for a complete facility address in an official listing. Preserve the original string and record the exact evidence for any proposed correction.
3. If you cannot resolve it, write an Unresolved decision that says which part is incomplete and where you would check next.
4. Apply the same discipline to the nine no-match records. A query failure, naming alias, stale address, missing unit, or limited search strategy can explain a missing result.
5. If a new official point is found, log its identifier as `external:ID` with evidence. The included checker can assess field completeness; it cannot verify that external source or your research.

> **No forced completion**
>
> Documenting an unresolved record is a valid outcome. Removing it from the spreadsheet would conceal the denominator and make your result look better without improving the data.

Your research queue

Rank unresolved cases by the consequence of a wrong location, the evidence already available, and the cost of the next check. State whether the next step is source correction, facility confirmation, unit verification, or a fresh reference lookup.

Use the right coordinates for analysis

1. Keep the original GeoJSON layers in EPSG:4326. Confirm this in each layer's source information; geographic longitude/latitude values should not be relabeled as State Plane coordinates.
2. Right-click the exact-reference layer → Export → Save Features As. Choose GeoPackage, save `outputs/antbytes_level2.gpkg`, name the layer `reference_exact_2237`, and select EPSG:2237 (NAD83 / Florida West, US survey feet).
3. Accept an appropriate available coordinate transformation if QGIS asks, and record your selection. Use the transformed layer for the exercise.
4. Select one Plant City-labeled input by record ID. Export the selected feature as a separate analysis layer, retaining its ID. Avoid buffering all 64 points when the task is to study one site.
5. Run Buffer from the Processing Toolbox on that selected-site layer. Confirm feet-based units and enter `5280` for an approximate one-mile planar radius. Save `site_buffer_1mi` in the GeoPackage.
6. Inspect the output and record the source point, output CRS, distance, and units. Check whether you intended a radius or a diameter.

> **Checkpoint D**
>
> A 5,280-foot buffer in this projected CRS approximates a one-mile straight-line radius. It is not a driving-time or walking-access area. Roads, barriers, entrances, speeds, and routing rules are not modeled by a circle.

Changing only the project display CRS can change how layers appear together; exporting a transformed analysis layer makes the chosen coordinate system explicit in the deliverable.

Reference: QGIS vector geometry tools and export workflow [1, 3].

Measure results without moving the denominator

Make a small audit table keyed by original input ZIP. Keep separate columns for baseline outcomes and your reviewed decisions. Retain all 100 original records in the overall denominator.

Metric	Calculation	What it does not establish
Baseline exact reference-match rate	64 / 100 = 64%	Independent positional or facility accuracy
Baseline review queue	27 review + 9 no match = 36	That all 36 are invalid
Reviewed reference coverage	(64 + newly accepted review IDs) / 100	A revised original exact-match rate
Original-ZIP match rate	Baseline exact matches for that input ZIP / input count for that ZIP	Corrected-ZIP totals or municipal membership

1. Count decisions by record ID, not by candidate rows. Check that Accepted + Rejected + Unresolved = 36.
2. For each original ZIP, report its input count and baseline exact count. Add your accepted-review count as a separate measure.
3. Calculate the remaining open queue from decision = Unresolved. Rejected records stay in the original denominator and need a reason.
4. Include a small-sample note where a ZIP contains only one or two inputs. This exercise is a selected facility sample, not a representative county accuracy study.
5. Write one paragraph explaining what improved and what remains uncertain. Attach the original snapshot timestamp and your review dates.

Example with hypothetical learner decisions

If you accept 10 review IDs, reject 2, and leave 24 unresolved, the decision totals are 36 and reviewed reference coverage is 74/100. The baseline exact-match rate remains 64%. These example decisions are not supplied results.

Design a map a reviewer can understand

Produce one overview map and, if useful, an inset for an unresolved cluster. Distinguish an exact reference point from a proposed candidate visually.

1. Create a presentation group. Use cyan circles for the original exact-reference points and a contrasting amber outline for candidates. Avoid presenting every candidate as a confirmed facility.
2. Label only a useful subset with record_id. Use filters or an inset to keep overlapping candidates readable. Treat shared coordinates as possible shared sites, not automatic duplicates.
3. Add the selected-site buffer with a light fill and a clear label: “Approximate one-mile straight-line buffer.”
4. Create a Print Layout. Add a map, title, legend, scale bar appropriate to the map CRS, source notes, author/reviewer, and a date. Add a north arrow if orientation would otherwise be unclear.
5. Use the AntBytes GIS logo and navy/cyan styling. Keep the map background subdued and preserve the amber review distinction. Supply any basemap attribution if you add a basemap.
6. Export a PDF to `outputs/antbytes_level2_map.pdf`. Open it and inspect text size, clipping, legend categories, and source notes before considering it finished.

> **Required limitation text**
>
> Reference matches are address-point associations from the supplied snapshot. Candidate points are for review. Site identity, unit, access point, and surveyed accuracy require separate evidence.

Reference: QGIS Print Layout and PDF output [4]. The included footprint diagram is a data overview, not a basemap, boundary, or completed student map.

Run checks you can explain

The included checker uses Python’s standard library and makes no network requests. It checks your CSV structure and decision evidence fields, then writes a JSON audit report.

```
python checks/check_submission.py worksheets/review-decisions.csv --output outputs/qa-report.json
```

1. Run it from the extracted tutorial folder. On some systems the command is `python3`. Read the terminal summary and the JSON output.
2. The untouched worksheet should report 36 Unresolved decisions and 36 incomplete review rows. That is an expected starting condition, not a completed submission.
3. Accepted rows require a point ID, proposed street/ZIP, usable longitude/latitude, evidence source, notes, reviewer, and date. Supplied candidate IDs must belong to the same record; their address, ZIP, and coordinates must agree.
4. New references use `external:ID`. They require a source URL and complete location fields. The script flags them for manual verification; it does not confirm the external source.
5. Before handoff, rerun with `--final`. It fails if required review fields are incomplete or structural errors remain. Properly documented Unresolved cases can pass.

> **What this check cannot prove**
>
> The checker cannot assess the quality of your research, inspect your QGIS transformations, confirm entrances or units, or grade cartography. Its output supports manual review; it is not a positional-accuracy certificate.

Keep the QA report with the submission. Do not edit the original dataset simply to make a structural check pass. Fix the decision log or explain a remaining limitation.

Deliver a compact, inspectable project

Submit a folder another analyst can open without guessing where the data came from or which records were accepted.

Deliverable under outputs/	Minimum content
antbytes_level2.qgz	Project with working relative references and clear layer groups
antbytes_level2.gpkg	Reference layer, selected accepted-review points where supported, and projected analysis/buffer layers
review-decisions-final.csv	All 36 review IDs; original input preserved; evidence and dates complete
qa-report.json	Final checker output with no structural errors or incomplete rows
antbytes_level2_map.pdf	Readable AntBytes-branded layout with sources and limitations
README.txt	Purpose, files, baseline timestamp, CRS/transformation, method, counts, unresolved issues

Assemble accepted points deliberately

If you create a combined reviewed point layer, merge the 64 baseline reference points with only the selected, supported review points. Preserve record_id and an origin/status field. Additional-reference points need an equivalent schema and provenance. Do not append all 79 candidates.

Each original input should contribute at most one accepted output feature for this capstone. Count distinct record IDs after merging; shared coordinates alone do not mean duplicate business records.

> **Handoff test**
>
> Copy your submission folder to a new location, open the project, inspect a known review case, and rerun the checker against the final CSV. If links break or the selected point cannot be traced to its decision, the handoff is not ready.

ASSESSMENT / 100 POINTS

Grade the evidence, not the match rate

Criterion	Points	Full-credit evidence
Data contract & IDs	20	Original 100 inputs preserved; 36 decision IDs unique; counts reconciled
Decision reasoning	30	All review rows documented; units, aliases, ZIPs, and competing points handled with cited evidence
CRS & analysis	15	Actual coordinate transformation; units and buffer parameters documented; correct interpretation
QA & statistics	15	Final structural checks pass; denominator and baseline/reviewed metrics clearly separated
Map communication	10	Readable symbols, sources, legend, dates, AntBytes styling, and limitations
Reproducible handoff	10	Project opens after moving; files and decisions can be traced from README

How to score yourself

Award up to half the points when a criterion is partly demonstrated but important evidence is missing. Give full credit only when the required artifact supports the claim. Use the workbook’s Rubric sheet to record the score and a brief justification.

Suggested learning target: **80/100 with no critical errors**. This is a self-assessment threshold for the tutorial, not an accredited certification or approval for emergency operations.

> **Critical errors to correct before passing**
>
> Missing or duplicated review IDs; accepting the wrong unit without evidence; presenting an unresolved candidate as confirmed; or measuring with a mislabeled CRS instead of transformed coordinates. An honest Unresolved decision is not a critical error.

Test your judgment

Answer in your own words and cite one of your project artifacts where possible. Each prompt is worth 5 points in this optional 30-point check; it is separate from the capstone's 100-point rubric.

1. You accept one candidate for HC-033. Why could filtering only on the joined Accepted status still show 37 points, and what additional condition prevents that?
2. HC-030 is reviewed and its proposed ZIP changes from 33569 to 33578. Which ZIP belongs in the original-data match-rate denominator?
3. You have a base-address point for 7606 Paula Dr but no evidence for suite 120. What decision is supported, and what is the next useful check?
4. A point layer displays correctly over a projected layer. Why is that insufficient proof that its stored coordinates are suitable for a feet-based buffer?
5. Your final decisions are 10 Accepted, 2 Rejected, and 24 Unresolved. What is the baseline exact-match rate, reviewed reference coverage, and unresolved queue?
6. Your automated QA passes, but the map labels every candidate as a verified facility. Is the capstone ready? Explain what the automated test did and did not establish.

> **Reflection**
>
> Choose the decision you found hardest. Explain what evidence changed your view, or why it did not justify changing the baseline. A useful analyst can explain uncertainty without hiding it.

Compare your reasoning

Prompt	Expected reasoning
1 / Join behavior	A record-level Accepted value is inherited by every candidate with that record_id. Filter by both Accepted and the selected candidate object ID; ensure one output per accepted review ID.
2 / ZIP denominator	Use original input ZIP 33569 for original-data performance. A corrected-ZIP distribution is a different, separately labeled output. Preserve the input value.
3 / Missing suite	Keep Unresolved unless independent evidence supports the requested unit. Check the facility/unit record or an appropriate official source; another suite is not a substitute.
4 / Coordinates	On-the-fly display transformation is not proof that stored coordinates were transformed. Export a projected analysis copy, verify its CRS/units, and record the transformation.
5 / Counts	Baseline exact matches: 64/100 = 64%. Reviewed reference coverage: (64+10)/100 = 74%. The open queue is 24; the two Rejected records remain in the original denominator.
6 / QA limits	Not ready. Structural field/ID checks do not validate facility identity or map claims. Revise symbology and language, then document remaining uncertainty.

Case studies have no universal acceptance key

The packet’s proposed candidates are evidence to investigate, not predetermined answers. An Accepted decision and an Unresolved decision can both be well reasoned when based on different documented evidence. Unsupported certainty should score worse than a justified unresolved outcome.

> **Your next professional habit**
>
> Treat the baseline as a versioned record of what was known, the decision log as the explanation of your changes, and the map as a communication product that must not claim more than those records support.

FINISH / KEEP THIS PAGE

Troubleshooting & reference desk

Symptom	Check first
CSV has the wrong feature count	Confirm delimiter, header row, and geometry setting; do not import workbook title rows as data.
Joined decisions are empty	Compare record_id text types, whitespace, matching fields, and join direction.
Many points appear accepted	Filter by selected object ID as well as decision. Review one-to-many behavior.
Points appear far away	Check coordinate order, signs, source CRS, and whether the CRS was assigned or actually transformed.
Buffer is much too large/small	Inspect input-layer units, selected features, and radius value; use the projected analysis layer.
Checker says rows are incomplete	Fill notes, evidence source, reviewer, and ISO date even when leaving a record Unresolved.
Project opens with missing layers	Recheck relative paths and include the referenced files with the handoff.

Primary references

1. QGIS 3.44 — Opening Data
2. QGIS — Joins and Relations
3. QGIS — Vector Geometry Tools
4. QGIS — Print Layout Output
5. Tampa GIS — public county address-point layer

AntBytes original instructional material. Data remains attributed to its public source publishers. Use the bundled comparison timestamp when reproducing the exercise; a fresh online query can return different records or IDs. No operational E911 system is part of this tutorial.