📖 Wrought Iron CLI Reference Manual

Version: 1.0.0 (Air-Gapped Edition) **Syntax Standard:** `wi [MODULE] [ACTION] [TARGET] [OPTIONS]`

Global Flags

Available on all commands.

- `--verbose / -v`: Enable debug logging to `stderr`.
- `--json`: Output result as raw JSON (for piping to other tools).
- `--yes / -y`: Auto-confirm all prompts (Non-interactive mode).
- `--engine [pandas|polars]`: Select the computation backend (Default: `pandas`).

---

1. Infrastructure (`wi connect`)

Management of SQLite files, connections, and storage security.

File Operations

1. **`wi connect file`** `[PATH]`
   - *Register an existing database as the active target.*
   - `--read-only`: Open in immutable mode.
   - `--check-wal`: Verify Write-Ahead-Log exists.
2. **`wi connect new`** `[PATH]`
   - *Initialize a fresh, empty Wrought Iron schema.*
   - `--force`: Overwrite if file exists.
   - `--page-size [4096|8192|16384]`: Set SQLite page size (Default: 4096).
3. **`wi connect list`**
   - *Show history of accessed databases.*
   - `--sort [access_time|size]`: Sort order.
   - `--limit [INT]`: Max rows to show (Default: 10).
4. **`wi connect alias`** `[NAME] [PATH]`
   - *Assign a short-name to a deep file path.*
   - `--overwrite`: Replace existing alias.
5. **`wi connect merge`** `[TARGET_DB] [SOURCE_DB]`
   - *ETL merge of two DB files.*
   - `--strategy [append|replace|ignore]`: Conflict resolution.
   - `--tables [LIST]`: Only merge specific tables (Comma-separated).
   - `--chunk-size [INT]`: Rows per commit (Default: 50000).

Storage Maintenance

6. `wi connect info`
   - *Display low-level metadata.*
   - `--extended`: Show WAL size, encoding, and user permissions.
7. `wi connect vacuum`
   - *Rebuild DB file to reclaim disk space.*
   - `--into [FILE]`: Vacuum into a new file instead of in-place.
8. `wi connect integrity-check`
   - *Run SQLite corruption scan.*
   - `--quick`: Skip index verification for speed.

Encryption (Air-Gapped Security)

9. `wi connect encrypt` `[PATH]`
   - *Encrypt database at rest (AES-256).*
   - `--key-file [PATH]`: Path to key (Interactive prompt if omitted).
   - `--output [PATH]`: Write encrypted file to new location.
10. `wi connect decrypt` `[PATH]`
    - *Decrypt a WI-locked file.*
    - `--key-file [PATH]`: Key to unlock.

---

2. Schema Management (`wi schema`)

Structure introspection, JSON "Backpack" management, and DDL ops.

Introspection

11. `wi schema list`
    - *List database objects.*
    - `--show-views / --no-views`: Toggle view visibility.
    - `--show-sys / --no-sys`: Toggle system tables (`sqlite_sequence`).
12. `wi schema describe` `[TABLE]`
    - *Show DDL structure.*
    - `--format [table|sql]`: Show as Grid or `CREATE TABLE` statement.
13. `wi schema inspect` `[TABLE]`
    - *Deep statistical profile.*
    - `--sample [FLOAT]`: % of data to scan (Default: 1.0).
    - `--histogram`: Include mini-histogram for numeric columns.
14. `wi schema diff` `[TABLE_A] [TABLE_B]`

- *Compare two schemas.*
- `--db-b [PATH]`: If Table B is in a different database.

Structure Evolution

15. **`wi schema detect-json`** `[TABLE]`
    - *Scan for hidden JSON structures.*
    - `--depth [INT]`: How deep to traverse nested JSON.
    - `--threshold [FLOAT]`: Min % of valid JSON rows to trigger detection (Default: 0.1).
16. **`wi schema flatten`** `[TABLE] [COL]`
    - *Explode JSON column into real columns.*
    - `--prefix [STR]`: Prefix for new columns (Default: `[COL]_`).
    - `--separator [STR]`: Nested key separator (Default: `_`).
    - `--drop-original`: Delete the source JSON column after flattening.
17. **`wi schema rename-col`** `[TABLE] [OLD] [NEW]`
    - *Rename column.*
    - `--dry-run`: Verify SQL generation without executing.
18. **`wi schema drop-col`** `[TABLE] [COL]`
    - *Drop column.*
    - `--vacuum`: Auto-vacuum after drop to reclaim space.
19. **`wi schema cast`** `[TABLE] [COL] [TYPE]`
    - *Change column type.*
    - `--type [INTEGER|TEXT|REAL|BLOB]`.
    - `--on-error [nullify|fail|ignore]`: Handling conversion failures.
20. **`wi schema graph`**
    - *Generate ERD.*
    - `--format [mermaid|dot]`: Output syntax.

---

3. Data Exploration (`wi query`)

Retrieval, filtering, and searching.

Basic Retrieval

21. **`wi query head`** `[TABLE]`
    - *Show first N rows.*
    - `-n [INT]`: Number of rows (Default: 10).
22. **`wi query tail`** `[TABLE]`
    - *Show last N rows.*
    - `-n [INT]`: Number of rows.
23. **`wi query sample`** `[TABLE]`

- *Get random subset.*
- `--frac [FLOAT]` (0.0-1.0) OR `-n [INT]`.
- `--seed [INT]`: Random seed for reproducibility.

Search & Filter

24. **`wi query filter`** `[TABLE]`
    - *Apply boolean logic.*
    - `--where "[STR]"`: Pandas query string (e.g., `age > 20`).
    - `--engine [numexpr|python]`: Backend speed optimization.
25. **`wi query sql`** `"[QUERY]"`
    - *Execute raw SQL.*
    - `--params [JSON]`: Safe parameter injection.
26. **`wi query search`** `[TABLE] [TERM]`
    - *Global full-text search.*
    - `--cols [LIST]`: Limit to specific columns.
    - `--case-sensitive / --ignore-case`.
    - `--regex`: Treat term as Regex.
27. **`wi query sort`** `[TABLE] [COL]`
    - *Order results.*
    - `--asc / --desc`.
    - `--alg [quicksort|mergesort]`.

Data Inspection

28. **`wi query distinct`** `[TABLE] [COL]`
    - *List unique values.*
    - `--counts`: Include frequency count per value.
29. **`wi query find-nulls`** `[TABLE]`
    - *Find incomplete rows.*
    - `--cols [LIST]`: Check only these columns.
    - `--mode [any|all]`: Row is null if *any* or *all* cols are null.
30. **`wi query dups`** `[TABLE] [COL]`
    - *Identify duplicates.*
    - `--keep [first|last|none]`: Which duplicates to mark.

---

4. Analytics (`wi aggregate`)

Statistical computing and transformation.

Aggregations

31. `wi aggregate groupby [TABLE] [GROUP_COLS]`
    - *Group and summarize.*
    - `--agg [DICT_STR]`: e.g., `"{'age': 'mean', 'salary': 'sum'}"`.
    - `--pivot`: Pivot result to wide format.
32. `wi aggregate pivot [TABLE] [INDEX] [COL] [VAL]`
    - *Create pivot table.*
    - `--func [mean|sum|count|max]`.
    - `--fill-value [VAL]`: Value for empty cells.
33. `wi aggregate crosstab [TABLE] [ROW] [COL]`
    - *Frequency matrix.*
    - `--normalize [index|columns|all]`: Show percentages.
    - `--margins`: Add row/column subtotals.

Statistical Measures

34. `wi aggregate describe [TABLE]`
    - *Summary statistics.*
    - `--percentiles [LIST]`: e.g., `0.1,0.5,0.9`.
    - `--include [all|number|object]`.
35. `wi aggregate corr [TABLE]`
    - *Correlation matrix.*
    - `--method [pearson|spearman|kendall]`.
    - `--min-periods [INT]`.
36. `wi aggregate skew [TABLE]`
    - *Calculate skewness.*
    - `--numeric-only`: Skip text columns.
37. `wi aggregate kurtosis [TABLE]`
    - *Calculate kurtosis.*
    - `--numeric-only`.

Window Functions

38. `wi aggregate moving-avg [TABLE] [COL]`
    - *Rolling window calculation.*
    - `--window [INT]`: Size of window.
    - `--center`: Center the window.
    - `--min-periods [INT]`.
39. `wi aggregate rank [TABLE] [COL]`
    - *Rank rows.*
    - `--method [average|min|max|first|dense]`.
    - `--pct`: Compute percentile rank.
40. `wi aggregate bin [TABLE] [COL]`

- *Discretize numbers.*
- `--bins [INT]`: Number of buckets.
- `--labels [LIST]`: Custom names for buckets.

---

5. Visualization (`wi plot`)

Plotext-based terminal charting.

Charts

41. **`wi plot bar`** `[TABLE] [CAT] [NUM]`
    - *Vertical Bar Chart.*
    - `--agg [sum|mean|count]`: Pre-aggregation function.
    - `--stack [COL]`: Stack bars by this column.
42. **`wi plot barh`** `[TABLE] [CAT] [NUM]`
    - *Horizontal Bar Chart.*
    - `--sort-by-val`: Sort bars by length.
43. **`wi plot hist`** `[TABLE] [NUM]`
    - *Histogram.*
    - `--bins [INT]`: Default 10.
    - `--orientation [vertical|horizontal]`.
44. **`wi plot scatter`** `[TABLE] [X] [Y]`
    - *Scatter Plot.*
    - `--color [COL]`: Color points by category.
    - `--marker [char]`: Default `•`.
45. **`wi plot line`** `[TABLE] [X] [Y]`
    - *Line Chart.*
    - `--group [COL]`: Draw multiple lines per group.
46. **`wi plot box`** `[TABLE] [NUM]`
    - *Box Plot.*
    - `--by [CAT]`: One box per category.
47. **`wi plot heatmap`** `[TABLE] [X] [Y]`
    - *Density Heatmap.*
    - `--bins [INT]`.
48. **`wi plot matrix`** `[TABLE]`
    - *Scatter Matrix.*
    - `--cols [LIST]`: Limit to specific numeric columns.

Configuration

49. **`wi plot save`** `[PATH]`

- *Export chart.*
- `--format [html|png|svg]`.

50. **`wi plot theme`** `[NAME]`
    - *Set aesthetics.*
    - `--name [dark|light|matrix|pro]`.

---

6. Data Wrangling (`wi clean`)

Repair, imputation, and harmonization.

Imputation

51. **`wi clean impute-mode`** `[TABLE] [COL]`
    - *Fill missing with mode.*
    - `--group-by [COL]`: Calculate mode per group.
52. **`wi clean impute-group`** `[TABLE] [TARGET]`
    - *Cohort Imputation (The "Source File" Fix).*
    - `--by [COL]`: Cohort ID (e.g., `source_file`).
    - `--std-max [FLOAT]`: Abort if group std-dev > X (Safety Valve).
    - `--min-samples [INT]`: Min records to calculate mode.
53. **`wi clean ml-impute`** `[TABLE] [COL]`
    - *KNN Imputation.*
    - `--neighbors [INT]`: K value (Default: 5).
    - `--weights [uniform|distance]`.
    - `--features [LIST]`: Columns to use for similarity.

Text Repair

54. **`wi clean harmonizes`** `[TABLE] [COL]`
    - *Cluster similar text.*
    - `--threshold [INT]`: Similarity % (Default: 90).
    - `--strategy [longest|shortest|most_frequent]`: Which alias to keep.
55. **`wi clean dedupe`** `[TABLE]`
    - *Find duplicates.*
    - `--col [COL]`: Field to check (e.g., `name`).
    - `--threshold [INT]`: Fuzzy match score.
    - `--block [COL]`: Block by column (e.g., `suburb`) for speed.
    - `--interactive`: Prompt user for every merge.
56. **`wi clean regex-replace`** `[TABLE] [COL] [PAT] [REPL]`
    - *Pattern substitution.*
    - `--count [INT]`: Max replacements per row.

57. `wi clean trim` `[TABLE] [COL]`
    - *Whitespace removal.*
    - `--side [left|right|both]`.

Sanitization

58. `wi clean drop-outliers` `[TABLE] [COL]`
    - *Statistical trimming.*
    - `--method [zscore|iqr]`.
    - `--threshold [FLOAT]`: e.g., 3.0 for Z-Score.
59. `wi clean map` `[TABLE] [COL] [FILE]`
    - *Apply dictionary mapping.*
    - `--ignore-case`.
    - `--default [VAL]`: Value if no match found.
60. `wi clean validate-schema` `[TABLE] [SCHEMA_FILE]`
    - *Enforce types/constraints.*
    - `--drop-invalid`: Remove rows that fail validation.

---

7. Geospatial (`wi geo`)

Offline Geocoding and Analysis.

Conversion

61. `wi geo geocode` `[TABLE] [ADDR_COL]`
    - *Address -> Lat/Lon.*
    - `--provider [local_db]`: **Must use bundled offline DB.**
    - `--fuzzy`: Allow minor misspellings in address.
62. `wi geo reverse` `[TABLE] [LAT] [LON]`
    - *Lat/Lon -> Suburb.*
    - `--provider [local_db]`.
63. `wi geo export-geojson` `[TABLE]`
    - *Export to GIS.*
    - `--lat [COL]`, `--lon [COL]`.
    - `--properties [LIST]`: Columns to include as metadata.

Analysis

64. `wi geo distance` `[TABLE] [LAT] [LON]`
    - *Point-to-Point.*
    - `--target [LAT] [LON]`: Reference point.

    - `--units [km|mi]`.
65. **`wi geo cluster`** `[TABLE]`
    - *DBSCAN Spatial Clustering.*
    - `--eps [FLOAT]`: Max distance (km).
    - `--min-samples [INT]`.
    - `--metric [haversine|euclidean]`.
66. **`wi geo centroid`** `[TABLE]`
    - *Geometric center.*
    - `--group-by [COL]`: Find center per group (e.g., per Source File).
67. **`wi geo bounds`** `[TABLE]`
    - *Bounding Box.*
    - `--buffer [FLOAT]`: Add padding (km).
68. **`wi geo nearest`** `[TABLE]`
    - *Nearest Neighbor.*
    - `--target-file [POI_DB]`: Secondary DB (e.g., Hospitals).
    - `--k [INT]`: Find nearest K neighbors.
69. **`wi geo heatmap`** `[TABLE]`
    - *ASCII Density Map.*
    - `--radius [INT]`: Blur radius.
70. **`wi geo validate`** `[TABLE]`
    - *Sanity check coordinates.*
    - `--drop-invalid`: Remove Lat > 90 / Lon > 180.

---

8. Machine Learning (`wi ml`)

Local Scikit-learn modeling.

Training

71. **`wi ml train-classifier`** `[TABLE] [TARGET] [FEATS]`
    - *Train Classifier (Categorical).*
    - `--algo [rf|svm|logistic]`.
    - `--estimators [INT]`: (RandomForest) Num trees.
    - `--split [FLOAT]`: Train/Test split size.
72. **`wi ml train-regressor`** `[TABLE] [TARGET] [FEATS]`
    - *Train Regressor (Continuous).*
    - `--algo [linear|ridge|lasso|rf]`.
73. **`wi ml save-model`** `[PATH]`
    - *Serialize model.*
    - `--format [pickle|joblib]`.
74. **`wi ml load-model`** `[PATH]`

- *Load model to active session.*

Inference & Metrics

75. `wi ml predict` `[TABLE] [MODEL]`
    - *Run Inference.*
    - `--out-col [NAME]`.
    - `--proba`: Output probability score instead of class.
76. `wi ml score` `[TABLE] [MODEL]`
    - *Evaluate Model.*
    - `--metric [accuracy|f1|r2|mse]`.
77. `wi ml feature-importance` `[MODEL]`
    - *Explainability.*
    - `--top-n [INT]`.

Unsupervised

78. `wi ml cluster-kmeans` `[TABLE]`
    - *K-Means Clustering.*
    - `--k [INT]`: Number of clusters.
    - `--features [LIST]`.
79. `wi ml detect-anomalies` `[TABLE]`
    - *Isolation Forest.*
    - `--contamination [FLOAT]`: Expected % of outliers.
    - `--features [LIST]`.
80. `wi ml split` `[TABLE]`
    - *Create Train/Test subsets.*
    - `--ratio [FLOAT]`.
    - `--stratify [COL]`.

---

9. Audit & Security (`wi audit`)

Forensics and Data Integrity.

Forensics

81. `wi audit log-view`
    - *Show Audit Log.*
    - `--limit [INT]`.
    - `--user [NAME]`.
    - `--action [CMD]`.

82. `wi audit hash-create` `[TABLE]`
    - *Generate Integrity Fingerprint.*
    - `--algo [sha256|sha512]`.
    - `--salt [STR]`.
83. `wi audit hash-verify` `[TABLE] [HASH]`
    - *Verify Integrity.*
    - `--salt [STR]`.
84. `wi audit export-cert`
    - *Generate Chain-of-Custody PDF.*
    - `--signer [NAME]`.
    - `--output [FILE]`.

Snapshotting

85. `wi audit snapshot` `[TABLE]`
    - *Backup table state.*
    - `--name [STR]`.
    - `--comment [STR]`.
86. `wi audit rollback` `[TABLE] [ID]`
    - *Restore table.*
    - `--dry-run`: Preview differences.

Protection

87. `wi audit scan-pii` `[TABLE]`
    - *Presidio Scan.*
    - `--entities [PHONE,EMAIL,CREDIT_CARD]`.
    - `--confidence [FLOAT]`.
88. `wi audit encrypt-col` `[TABLE] [COL]`
    - *Column Encryption (Fernet).*
    - `--key-file [PATH]`.
89. `wi audit decrypt-col` `[TABLE] [COL]`
    - *Column Decryption.*
    - `--key-file [PATH]`.
90. `wi audit anonymize` `[TABLE] [COL]`
    - *Masking.*
    - `--method [mask|hash|redact]`.
    - `--chars [INT]`: Number of chars to mask.

---

10. Operations (`wi ops`)

Automation and DataOps Pipelines.

Scheduling

91. **`wi ops schedule create`** `[CMD]`
    - *New Cron Job.*
    - `--cron [STR]`: e.g., `"0 3 * * *"`.
    - `--name [STR]`.
    - `--timeout [INT]`.
92. **`wi ops schedule list`**
    - *List Jobs.*
    - `--status [active|paused]`.
93. **`wi ops schedule delete`** `[ID]`
    - *Remove Job.*
    - `--force`.
94. **`wi ops trigger`** `[ID]`
    - *Manual run.*
    - `--wait`: Wait for completion.
95. **`wi ops logs`**
    - *View Job History.*
    - `--job-id [ID]`.
    - `--errors-only`.

Monitoring

96. **`wi ops drift-check`** `[TABLE]`
    - *Quality Control.*
    - `--baseline [SNAPSHOT]`.
    - `--threshold [FLOAT]`: Max P-value diff.
97. **`wi ops alert-config`**
    - *Alert Hooks.*
    - `--email [ADDR]`.
    - `--log-file [PATH]`.
98. **`wi ops pipeline run`** `[YAML]`
    - *Execute Multi-Step Workflow.*
    - `--continue-on-error`.
99. **`wi ops maintenance`**
    - *DB Optimization.*
    - `--reindex`.
    - `--analyze`.
100. **`wi ops export-status`** * *Health Check.* * `--format [json|xml]`.

---

11. Collaboration (`wi collab`)

Team sharing.

Views & Recipes

101. **`wi collab view save`** `[NAME]` * *Save Query View.* * `--query [SQL]`. * `--desc [STR]`.
102. **`wi collab view load`** `[NAME]` * *Load View.* * `--as-table [NAME]`: Materialize as table.
103. **`wi collab view list`** * *List Views.* * `--filter [STR]`.
104. **`wi collab config export`** `[CMD] [FILE]` * *Share Settings.* * `--include-secrets`: (Warning) Include keys.
105. **`wi collab config import`** `[FILE]` * *Load Settings.* * `--scope [user|project]`.
106. **`wi collab recipe bundle`** `[NAME]` * *Zip Project.* * `--out [FILE]`.
107. **`wi collab recipe install`** `[FILE]` * *Unzip Project.* * `--overwrite`.

Metadata

108. **`wi collab notes add`** `[TABLE]` * *Annotate.* * `--msg [STR]`. * `--author [STR]`.
109. **`wi collab notes show`** `[TABLE]` * *Read Notes.* * `--limit [INT]`.
110. **`wi collab workspace dump`** * *State Export.* * `--full`: Include data + config + logs.

---

12. Reporting (`wi report`)

Offline HTML/PDF generation.

Dashboards

111. **`wi report generate`** `[TABLE]` * *Sweetviz EDA.* * `--out [FILE]`. * `--layout [vertical|widescreen]`.
112. **`wi report diff`** `[TABLE]` * *Before/After Report.* * `--snapshot [ID]`.
113. **`wi report validation`** `[TABLE]` * *Quality Report.* * `--rules [FILE]`.
114. **`wi report profile`** `[TABLE]` * *JSON Stats Dump.* * `--minimal`: Exclude quantiles.

Documentation

115. **`wi report schema-doc`** * *Data Dictionary HTML.* * `--title [STR]`.
116. **`wi report audit-timeline`** * *Visual Log.* * `--range [START] [END]`.
117. **`wi report map`** `[TABLE]` * *Offline Leaflet Map.* * `--lat [COL]`, `--lon [COL]`.
118. **`wi report summary`** `[TABLE]` * *Executive PDF.* * `--include-charts`.
119. **`wi report dependencies`** * *Lineage DAG.* * `--format [png|svg]`.
120. **`wi report serve`** * *Localhost Preview.* * `--port [INT]`. * `--bind [IP]`.

End of Manual.