tlptaco Configuration Cheat-Sheet

Below is a **complete** YAML example showcasing every configuration key that tlptaco understands. Replace placeholder tokens (`<…>`), choose from the enumerated values (`value1 | value2`) and delete any block marked **OPTIONAL** if you don’t need it.

```yaml
# ---------------------------------------------------------------------------
# TOP-LEVEL METADATA
# ---------------------------------------------------------------------------
offer_code: <offer code>
campaign_planner: <planner name>                    # OPTIONAL
lead: <lead name>                                   # OPTIONAL

# ---------------------------------------------------------------------------
# LOGGING
# ---------------------------------------------------------------------------
logging:
  level: INFO | DEBUG | WARNING | ERROR
  file: path/to/run.log                             # OPTIONAL
  debug_file: path/to/run.debug.log                 # OPTIONAL
  sql_file: path/to/run.sql.log                     # OPTIONAL (captures rendered SQL)

# ---------------------------------------------------------------------------
# DATABASE CONNECTION (Teradata)
# ---------------------------------------------------------------------------
database:
  host: <td-host>
  user: <username>
  password: <password>                              # OPTIONAL
  logmech: KRB5 | LDAP                               # OPTIONAL (defaults KRB5)

# ---------------------------------------------------------------------------
# PRE-RUN SQL FILES (executed in order) — OPTIONAL
# ---------------------------------------------------------------------------
pre_sql:                                            # OPTIONAL
  - sql/setup_temp_tables.sql
  - sql/refresh_dimensions.sql

# ---------------------------------------------------------------------------
# ELIGIBILITY ENGINE
# ---------------------------------------------------------------------------
eligibility:
  eligibility_table: <schema.table>

  unique_identifiers:
    - c.customer_id
    - c.account_id                                   # as many as required

  tables:
    - name: <schema.customer>
      alias: c
      sql: |                                         # OPTIONAL inline view
        SELECT * FROM <schema.customer> WHERE load_dt = CURRENT_DATE
      join_type: ""                                  # e.g. LEFT JOIN – OPTIONAL
      join_conditions: ""                            # OPTIONAL
      where_conditions: ""                           # OPTIONAL (table-level)
      unique_index: customer_id                      # OPTIONAL
      collect_stats: [customer_id, status_cd]        # OPTIONAL list

    - name: <schema.account>
      alias: a
      join_type: LEFT JOIN                           # OPTIONAL (defaults INNER)
      join_conditions: c.customer_id = a.customer_id # REQUIRED for joined
tables

  conditions:
    main:
      BA:                                            # Base-eligibility checks
        - sql: c.age >= 18
          description: Must be adult                 # OPTIONAL
        - sql: c.country_cd = 'US'

    channels:
      email:                                         # <channel name>
        BA:
          - sql: c.email_opt_in = 1

        loyalty_gold:                                # <segment name>
          - sql: c.loyalty_tier = 'GOLD'

        loyalty_silver:
          - sql: c.loyalty_tier = 'SILVER'

      sms:
        BA:
          - sql: c.sms_opt_in = 1
        high_value:
          - sql: c.monthly_spend > 100
```

```
# ---------------------------------------------------------------------------
# WATERFALL ENGINE
# ---------------------------------------------------------------------------
waterfall:
  output_directory: path/to/waterfall/output

  count_columns:
    - c.customer_id                                 # single column group
    - [c.customer_id, c.account_id]                 # multi-column group

  history:                                          # OPTIONAL – run history
    track: true | false
    db_path: path/to/waterfall_history.sqlite        # OPTIONAL (default inside
out dir)
    recent_window_days: 30                           # alias: lookback_days –
OPTIONAL
    compare_offset_days: 90                          # alias:
days_ago_to_compare – OPTIONAL

# ---------------------------------------------------------------------------
# OUTPUT ENGINE
# ---------------------------------------------------------------------------
output:
  channels:
    email:
      columns:
        - c.customer_id
        - c.email_addr
        - c.template_id
      file_location: path/to/email                   # directory or DB schema
      file_base_name: email_list
      unique_on: [c.customer_id]                     # OPTIONAL – dedupe keys
      output_options:
        format: csv | excel | parquet | table
        additional_arguments: {}                     # OPTIONAL kwargs to writer
        custom_function: package.module.fn_name      # OPTIONAL post-process
hook

    sms:
      columns: [...]
      file_location: path/to/sms
      file_base_name: sms_list
      output_options:
        format: parquet

  failed_records:                                   # OPTIONAL – failure dump
    enabled: true | false
    first_reason_only: true | false             # OPTIONAL (default false)
    file_location: path/to/failed
    file_base_name: failed_list
    output_options:
      format: parquet | csv | excel | table
      additional_arguments: {}                      # OPTIONAL
```

Field-by-Field Explanation

Top-level

Key	Required	Description
offer_code	✓	Short label used by progress bar & filenames.
campaign_planner	✗	Free-text metadata surfaced in waterfall report header.
lead	✗	Another optional header field.
pre_sql	✗	Ordered list of `.sql` script paths to execute **before** any engine logic.

logging

Key	Required	Values / Notes
level	✓	Standard Python levels (`INFO`, `DEBUG`, …).
file	✗	Main log file; created if path supplied.
debug_file	✗	All messages at `DEBUG` level.

Key	Required	Values / Notes
sql_file	✗	Raw rendered SQL; extremely useful for copy-paste.

database

Key	Required	Notes
host	✓	TDPID / hostname or IP.
user	✓	Database user.
password	✗	Skip when using `logmech: KRB5`.
logmech	✗	`KRB5` (Kerberos) or `TD2` (password). Default `KRB5`.

eligibility

See inline comments in YAML. Important points: * `tables[0]` is treated as the **FROM** table; subsequent tables must specify `join_type` & `join_conditions`. * `conditions.main.BA` is mandatory; only BA checks are allowed under `main`. * Under each channel you can define `BA` plus any number of segments (any key that isn't `BA`).

waterfall

- `count_columns` – each entry is **either** a single column or a list that represents a composite group.
- History: set `track: true` to enable SQLite snapshots; choose either `recent_window_days` or `compare_offset_days` to pick the comparison run.

output

- Each **channel** key must also exist under `eligibility.conditions.channels`.
- `format` values:
  - `csv` – plain CSV via `pandas.to_csv()`.
  - `excel` or `xlsx` – Excel workbook (openpyxl / xlsxwriter backend).
  - `parquet` – Apache Parquet via `pandas.to_parquet()`.

    - `table` – create a Teradata table where `file_location` is the schema and `file_base_name` the table name.
- `unique_on` (OPTIONAL) – list of columns used for de-duplication inside the SQL (QUALIFY ROW_NUMBER…). Qualified identifiers allowed.

failed_records (OPTIONAL)

- `enabled` – master switch.
- `first_reason_only` – emit one row per ID (`true`) or one per failure reason (`false`).
- Remaining keys mirror the normal output channel pattern.

pre_sql (OPTIONAL)

Executed sequentially **before** the Eligibility Engine runs. Each file is split on `;` and run via the same DB connection.