pravidha

Pramana

Reference

Governance evidence for Snowflake — classification, policy coverage, drift, audit packs

Document	Pramana — Reference
Version	v1.0
Date	17 November 2026
Author	Pravidha — Snowflake Select Service Partner
Status	Released
Classification	Public

© 2026 Pravidha. This document and the software it describes are Pravidha intellectual property.

Contents

- Application roles
- What the application is granted
- Procedures
- Views
- The policy pack tables
  - GOV_SETTING
  - GOV_SCOPE
  - GOV_CATEGORY
  - GOV_BINDING
  - GOV_TAG
  - GOV_TAG_VALUE
  - GOV_CONTROL
  - GOV_FRAMEWORK
  - GOV_EXEMPTION
  - GOV_EVIDENCE_SOURCE
- Controls
- Drift events
- Remediation statuses
- The evidence pack
- Log lines

© 2026 Pravidha · Public

For anyone who needs the exact name, column or rule. Everything here is in schema `CORE` (procedures, views, consoles) or schema `GOV` (tables) of the application, written as `PRAMANA.CORE.…` and `PRAMANA.GOV.…` when the application is installed as `PRAMANA`.

Application roles

Role	For	Can
`PRAMANA_ADMIN`	Platform or governance engineering	Everything: maintain the policy pack, run the engine, start and stop the schedule, load and clear the sample
`PRAMANA_STEWARD`	Data stewards	Read everything; decide proposals, acknowledge drift, add and close exemptions
`PRAMANA_AUDITOR`	Audit, risk, security	Read findings, drift, evidence packs and the policy pack
`PRAMANA_EVIDENCE_READER`	A GRC tool's service role	Read `CORE.PRAMANA_EVIDENCE_V` and nothing else

Each role is granted what it needs directly. An admin can do everything a steward can, and a steward everything an auditor can.

Console	Open to	Adds
`PRAMANA_VIEWER` (the default)	Admin, steward, auditor	Read-only pages: Overview, Controls, Review queue, Drift, Evidence, Policy pack
`PRAMANA_REVIEW`	Admin, steward	Approve and reject proposals, acknowledge drift, add exemptions
`PRAMANA_CONSOLE`	Admin	Setup, Remediation, Run and Sample

What the application is granted

Grant	Why	When
`EXECUTE TASK`, `EXECUTE MANAGED TASK`	One serverless nightly task	At install
`SNOWFLAKE.OBJECT_VIEWER`	`ACCOUNT_USAGE.TABLES`, `COLUMNS`	You grant it
`SNOWFLAKE.GOVERNANCE_VIEWER`	`TAG_REFERENCES`, `POLICY_REFERENCES`, `MASKING_POLICIES`, `ROW_ACCESS_POLICIES`, `DATA_CLASSIFICATION_LATEST`	You grant it
`SNOWFLAKE.SECURITY_VIEWER`	`ACCOUNT_USAGE.GRANTS_TO_ROLES`	You grant it
`READ SESSION`	The user name on review decisions	Optional

Procedures

All in schema `CORE`.

© 2026 Pravidha · Public

Procedure	Role	What it does
`GOV_RUN(action, options)`	Admin	The engine. `options` may be `NULL`
`GOV_NIGHTLY()`	Admin	The whole chain in order. What the task calls
`GOV_SCAN_AND_EVALUATE()`	Admin	Scan, classify and evaluate, without drift or evidence
`GOV_START()`, `GOV_STOP()`	Admin	Create and resume, or suspend, the nightly task
`GOV_STATUS()`	Admin, steward, auditor	Edition, version, task state, last run
`GOV_LOAD_CONFIG()`	Admin	Load the policy pack from CSV files on the application's stage
`GOV_SAMPLE_LOAD()`, `GOV_SAMPLE_BREAK()`, `GOV_SAMPLE_RESTORE()`, `GOV_SAMPLE_CLEAR()`	Admin	The built-in sample
`GOV_DECIDE(id, decision, note)`	Steward	Approve or reject one proposal: `APPROVE` or `REJECT`
`GOV_DECIDE_ALL(decision, min_confidence)`	Steward	Decide every open proposal at or above `HIGH`, `MEDIUM` or `LOW`
`GOV_ACKNOWLEDGE_DRIFT(id, note)`	Steward	Mark a drift event as reviewed
`GOV_ADD_EXEMPTION(target, control, reason, expires)`	Steward	Accept a finding until a date
`GOV_CLOSE_EXEMPTION(target, control)`	Steward	Withdraw an exemption

The actions of `GOV_RUN`:

Action	What it does
`ACCESS`	Reports what Pramana can read, with the `GRANT` statements still missing
`VALIDATE`	Checks the policy pack against itself and lists problems and warnings
`SCAN`	Takes a snapshot of the metadata of the databases in scope
`CLASSIFY`	Proposes a category for columns that look sensitive and are not tagged. Closes an approved proposal once its tag is on the column
`EVALUATE`	Evaluates every control against the latest snapshot
`DRIFT`	Compares the latest snapshot with the one before it
`EVIDENCE`	Writes one evidence pack per framework
`REMEDIATE`	Writes the statements that would close the open findings and the approved proposals still waiting for their tag
`NIGHTLY`	Scan, classify, evaluate, drift and evidence, in order
`PURGE`	Removes snapshots, and what was built on them, older than `EVIDENCE_RETENTION_DAYS`
`SAMPLE`	Used by the `GOV_SAMPLE_…` procedures

© 2026 Pravidha · Public

Views

View (schema CORE)	What it answers
GOV_CONTROL_STATUS_V	The thirteen controls: latest status, counts, note
GOV_FINDING_LATEST_V	Every failure, warning and exemption of the latest evaluation, with its action
GOV_COVERAGE_V	Columns, tagged and masked, per database and schema
GOV_SENSITIVE_COLUMN_V	Every tagged column with its category, the bound policy and what is attached
GOV_TAG_V	Every table against every governed tag: the value it carries, where the tag is set, the rank the pack gives that value and whether the pack lists it
GOV_REVIEW_QUEUE_V	Open proposals
GOV_DRIFT_OPEN_V	Unacknowledged drift, weakening first
GOV_EVIDENCE_LATEST_V	The newest pack per framework, as JSON and Markdown
GOV_REMEDIATION_OPEN_V	The latest batch, still to do, in order (STEP): statements ready to run, policy templates to complete, and what waits for them. No rows when the latest run found nothing left to do
GOV_SNAPSHOT_LATEST_V	The newest snapshot
PRAMANA_EVIDENCE_V	The contract for GRC tooling: one row per pack with coverage, weakening count, open proposals, controls and the clause map

The tables behind the views are in schema GOV and readable by every role except the evidence reader.

The policy pack tables

Ten tables in schema GOV. A holder of PRAMANA_ADMIN has INSERT, UPDATE and DELETE on them. They can also be loaded from CSV files, one file per table, named after the table in lower case.

GOV_SETTING

Key	Meaning	Default
CLASSIFICATION_TAG	Fully qualified tag whose value is the category	Empty
INCLUDE_VIEWS	Scan views as well as tables	FALSE
CLASSIFY_MIN_CONFIDENCE	HIGH, MEDIUM or LOW: the lowest classifier confidence that counts	MEDIUM

© 2026 Pravidha · Public

Key	Meaning	Default
CLASSIFICATION_STALE_DAYS	CLS-04: how old a classifier result may be, in days	30
NIGHTLY_SCHEDULE	The task's schedule. Call GOV_START again after changing it	USING CRON 0 2 * * * UTC
FRAMEWORKS	Frameworks a pack is built for on each run, comma separated	GDPR_UK,DORA
GOVERNED_OWNER_ROLES	ACC-02: a regular expression your table-owning roles match	See below
BROAD_GRANTEES	ACC-01: roles whose SELECT on a sensitive table is a finding, comma separated	PUBLIC
MAX_FAILURES_IN_PACK	Failures listed in full per control in a pack	200
EVIDENCE_RETENTION_DAYS	What PURGE keeps	1100

The default for GOVERNED_OWNER_ROLES matches roles whose name begins DATA_OWNER_ or ends _OWNER.

CATALOG_SOURCE, CLASSIFY_ON_SCAN and REMEDIATION_MODE belong to the standalone edition and are ignored inside the application. EDITION, VERSION, RUNNING, SAMPLE_ACTIVE and DEFAULT_PACK_SEEDED are Pramana's own bookkeeping.

GOV_SCOPE

DATABASE_NAME, SCHEMA_INCLUDE_REGEX, SCHEMA_EXCLUDE_REGEX, TABLE_EXCLUDE_REGEX, ENABLED. The three expressions are case-insensitive regular expressions; SCHEMA_INCLUDE_REGEX defaults to .*.

GOV_CATEGORY

The taxonomy.

Column	Meaning
CATEGORY	For example PII_EMAIL
PRIVACY_LEVEL	IDENTIFIER, QUASI_IDENTIFIER, SENSITIVE, or NONE (never counts as sensitive)
TAG_VALUE	The value of the classification tag that means this category. Usually the same name
CLASSIFIER_CATEGORIES	Snowflake semantic categories that map here, for example EMAIL
NAME_REGEX	Case-insensitive regular expression searched in the column name, for example E_?MAIL

It ships with PII_NAME, PII_EMAIL, PII_PHONE, PII_ADDRESS, PII_DOB, PII_NATIONAL_ID, FIN_ACCOUNT, HEALTH and INTERNAL. Rename, add or remove freely.

© 2026 Pravidha · Public

GOV_BINDING

`CATEGORY`, `MASKING_POLICY` (fully qualified; empty means none required), `ROW_ACCESS_POLICY`, `RAP_COLUMN` (the column the policy takes), `ENABLED`. A category without a binding is governed (the classification controls apply) but not protected (no policy is required); `VALIDATE` warns, so the gap is a decision and not an accident.

GOV_TAG

The table-level tags you govern by, one row each.

Column	Meaning
`TAG_CLASS`	What the tag is for, as one word of at most 24 characters: `RETENTION`, `RESIDENCY`, `OWNER`. It names the tag in findings, drift events (`residency_tag_removed`) and evidence
`TAG_NAME`	The tag, fully qualified. Empty: the tag is not evaluated and its controls are skipped. It cannot be the classification tag, and two rows cannot name the same tag
`REQUIRED_ON`	Which tables must carry it: `SENSITIVE` (tables holding sensitive columns; the default) or `ALL` (every table in scope)
`MISSING_CONTROL`	The control that fails a table that needs the tag and carries no value. Empty: presence is not checked
`VALUE_CONTROL`	The control that fails a value `GOV_TAG_VALUE` does not list. Empty: values are not checked
`WEAKER_WHEN`	`HIGHER` or `LOWER`: which direction of `VALUE_RANK` weakens your position, for drift. Empty: the values are not ranked
`RANK_UNIT`	What `VALUE_RANK` counts, as the evidence pack prints it (`days`)
`DESCRIPTION`, `ENABLED`	

Control ids are yours to choose, up to 20 characters, and cannot be one of the eleven built-in ids. Two tags may share a control, if you want one control to answer for both.

GOV_TAG_VALUE

`TAG_CLASS`, `TAG_VALUE` (a value the tag may carry), `VALUE_RANK` (optional: where the value stands), `DESCRIPTION`, `ENABLED`. A tag with no row here accepts any value. For retention the rank is the number of days the class keeps data, and `WEAKER_WHEN` is `HIGHER`: the evidence pack states the period of each class in use, and a change of class counts as weakening when the new class keeps the data longer. A value listed without a rank (a legal hold) counts as the weakest. For a tag ranked the other way, a confidentiality tier for instance, set `WEAKER_WHEN = LOWER`.

GOV_CONTROL

`CONTROL_ID`, `ENABLED`, `SEVERITY` (`FAIL` or `WARN`). The catalogue is Pramana's; you switch a control off or re-grade it. The controls your own governed tags name are added to it by `VALIDATE`, and re-graded the same way afterwards.

GOV_FRAMEWORK

`FRAMEWORK`, `CONTROL_ID`, `CLAUSE`, `CLAUSE_TITLE`. Add your own framework as rows and name it in `FRAMEWORKS`.

© 2026 Pravidha · Public

GOV_EXEMPTION

`TARGET` (`DB.SCHEMA.TABLE` or `DB.SCHEMA.TABLE.COLUMN`; a table-level row covers its columns), `CONTROL_ID`, `REASON`, `APPROVED_BY`, `EXPIRES`. An exemption whose `EXPIRES` date has passed no longer applies.

GOV_EVIDENCE_SOURCE

`SOURCE_NAME`, `SOURCE_TYPE` (`VIGIL` or `VIEW`), `OBJECT_FQN`, `ENABLED`. Type `VIEW` samples any view the application can read into the pack. A pack carries the row count of such a view and a sample of up to twenty of its rows, so name only views whose contents belong in an evidence pack.

Controls

Id	Family	Title	Default grade
CLS-01	Classification	Sensitive columns carry the classification tag	Fail
CLS-02	Classification	Tag values are from the taxonomy	Fail
CLS-03	Classification	Classifier agrees with the tag	Warn
CLS-04	Classification	Classification is current	Warn
POL-01	Policy	Tagged columns have the bound masking policy	Fail
POL-02	Policy	Tables with row-access categories carry the bound policy	Fail
POL-03	Policy	Masking policies sit only on tagged columns	Warn
POL-04	Policy	Bound policies exist and are referenced	Fail
ACC-01	Access	No broad grants on sensitive tables	Fail
ACC-02	Access	Sensitive tables are owned by a governed role	Warn
RET-01	Retention	Tables that need one carry a retention class	Fail
RET-02	Retention	Retention classes are from the pack	Fail
DQ-01	Quality	Reconciliation evidence is green for sensitive tables	Fail

What makes each one fail, and what closes it:

Id	Non-compliant when	What closes it
CLS-01	A column with no tag is detected as a sensitive category	Tag it, or reject the proposal
CLS-02	A tag value is not one of the taxonomy's	Decide: add the value as a category, or retag
CLS-03	A tagged column is detected as a different category (warning)	A steward's eye
CLS-04	A sensitive table has no classifier result within `CLASSIFICATION_STALE_DAYS` (warning)	Classify it

© 2026 Pravidha · Public

Id	Non-compliant when	What closes it
POL-01	A tagged column's masking policy is not the one bound to its category	Attach the bound policy
POL-02	A table holds a category that binds a row access policy, and lacks it	Attach it
POL-03	A masking policy sits on an untagged column (warning)	Tag the column, or exempt it
POL-04	A bound policy does not exist, or is attached nowhere in scope	Create it, or fix the binding
ACC-01	A broad grantee holds `SELECT`, `ALL` or `OWNERSHIP` on a sensitive table	Revoke
ACC-02	A sensitive table's owner is not a governed role (warning)	Transfer ownership
RET-01	A table holding sensitive data (or any table, when the `RETENTION` row of `GOV_TAG` says `REQUIRED_ON = ALL`) carries no retention class. Skipped until `GOV_TAG` names the retention tag	Give the table its class, where you govern your data
RET-02	A table carries a retention value that `GOV_TAG_VALUE` does not list. Skipped without a retention tag or without classes	Correct the tag, or add the class to the pack
Yours	The two controls of each governed tag you add: a table that needs the tag carries none; a value the pack does not list	Set or correct the tag, where you govern your data
DQ-01	Vigil's latest reconciliation check on a sensitive table fails. Skipped without Vigil	Reconcile

Statuses: `PASS`, `FAIL` or `WARN` per control; `EXEMPT` per finding (counted, listed in the pack, never fails); `SKIPPED` per control, with the reason in its note. Control ids never change, because auditors cite them.

Drift events

Event	Recorded when	Counts as weakening
`table_added`, `table_removed`	A table enters or leaves the scanned estate	No
`owner_changed`	A table's owner changes	When the table is sensitive and the new owner is not a governed role
`column_added`, `column_removed`	A column appears or disappears	No
`sensitive_column_added`	A new column is detected as sensitive and carries no tag	Yes
`tag_added`, `tag_changed`	A column gains a classification, or its value changes	No
`tag_removed`	A column loses its classification	Yes
`masking_added`	A masking policy is attached to a column	No

© 2026 Pravidha · Public

Event	Recorded when	Counts as weakening
`masking_changed`, `masking_removed`	A column's masking policy is replaced or detached	Yes
`rap_added`	A row access policy is attached to a table	No
`rap_removed`	A row access policy is detached	Yes
`grant_added`	A privilege is granted on a table	When the table is sensitive and the privilege is `SELECT`, `ALL` or `OWNERSHIP`
`grant_removed`	A privilege is revoked	No
`<class>_tag_added`	A table gains a governed tag, for example `retention_tag_added`	No
`<class>_tag_changed`	The value of a governed tag changes	When the new value ranks weaker, by the tag's `WEAKER_WHEN`
`<class>_tag_removed`	A table loses a governed tag	Yes

Policy attachments that appear only because Snowflake's metadata caught up with the account are not reported as drift.

Remediation statuses

Status	Meaning
`PROPOSED`	A statement that can run as written
`TEMPLATE`	A `CREATE … POLICY` for a policy the pack binds and the account does not have. The body is a placeholder for you to complete
`BLOCKED`	A statement that is correct and cannot run until the policy from a template exists

The evidence pack

Each pack is one row of `GOV.GOV_EVIDENCE_PACK`: `PACK_ID`, `GENERATED_AT`, `SNAPSHOT_ID`, `RUN_ID`, `FRAMEWORK`, `OVERALL`, `BASELINE_SNAPSHOT`, `PACK_SHA256`, `PACK` (JSON) and `PACK_MD` (the readable pack).

Key in `PACK`	Content
`pack_id`, `generated_at`, `product`, `framework`, `overall`, `snapshot`, `run_id`	The header. `overall` is `FAIL` if any control fails, `WARN` if any warns, otherwise `PASS`
`coverage`	Columns, tagged, sensitive detected, untagged detections, masked, coverage percentages, per database; and `governed_tags`, one entry per governed tag
`controls`	Per control: title, family, severity, status, counts, clauses, failures (target, status, message, action) and how many were left out
`framework_map`	Per clause: title, controls, and the worst status of those controls
`drift`	Baseline snapshot, counts, and each event, weakening first

© 2026 Pravidha · Public

Key in PACK	Content
`exemptions`	Every exemption in force: target, control, reason, approver, expiry
`review`	Open, approved, applied and rejected proposal counts
`evidence_sources`	Vigil's reconciliation summary where linked; for other views, a row count and a sample
`integrity`	`algorithm` (`sha256`), `value`, and what the value covers

The seal is the SHA-256 of the pack without its `integrity` block, with whole numbers written without a decimal point, serialised as JSON with sorted keys. `verify_pack.py`, on the documentation page, recomputes it; the Evidence Guide shows how.

Log lines

Written to your account's event table, if you have one.

Line	Level	When
`pramana.run`	`INFO`	Once per engine run
`pramana.weakening`	`WARNING`	When a run finds weakening drift, with the number of events and the two snapshots compared

© 2026 Pravidha · Public