

## Data model: Phase 2

### A) **prompt\_backlog** (draft/staging)

Minimum columns:

- `prompt_id` (UUID)
- `addie_phase` (Analyze/Design/Develop/Implement/Evaluate)
- `title`
- `prompt_text`
- `intended_user`
- `model_target` (e.g., Gemini 1.5 Pro / 2.x)
- `inputs_schema` (JSON)
- `outputs_schema` (JSON)
- `acceptance_criteria` (JSON: rubric + constraints)
- `tags` (array)
- `status` (Draft → InReview → Testing → ApprovedForPublish → Rejected)
- `semantic_fingerprint` (hash of normalized prompt text)
- `embedding` (VECTOR or stored separately)
- `similarity_score_max` (float)
- `similar_to_prompt_id` (nullable)
- `created_by`, `created_at`, `updated_at`

## B) **prompt\_versions** (immutable history)

- `prompt_id`
- `version` (semantic: 1.0.0, 1.0.1, etc.)
- `change_type` (major/minor/patch)
- `prompt_text`
- `diff_summary`
- `created_at`, `created_by`
- `semantic_fingerprint`
- `embedding`

## C) **prompt\_tests** (evidence)

- `prompt_id`
- `version`
- `test_suite_id`
- `test_case_id`
- `input` (JSON)
- `expected` (JSON or rubric)
- `output` (JSON/text)
- `scores` (JSON: accuracy, format\_valid, policy\_safe, etc.)
- `pass_fail`

- `run_at`
- `run_metadata` (latency, cost estimates, model)

#### D) **prompt\_registry** (final data source for the Agent)

Only “published” prompts:

- `prompt_key` (stable human-friendly ID)
- `prompt_id`
- `current_version`
- `addie_phase`
- `title`
- `prompt_text`
- `inputs_schema, outputs_schema`
- `tags`
- `quality_grade` (A/B/C)
- `owner`
- `published_at`
- `deprecation_status`

This `prompt_registry` table is what your Agent queries.