

Tables (49)

Name	Type	Schema
_sqlx_migrations		CREATE TABLE _sqlx_migrations (version BIGINT PRIMARY KEY, description TEXT NOT NULL, installed_on TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, success BOOLEAN NOT NULL, checksum BLOB NOT NULL, execution_time BIGINT NOT NULL)
version	BIGINT	"version" BIGINT
description	TEXT	"description" TEXT NOT NULL
installed_on	TIMESTAMP	"installed_on" TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
success	BOOLEAN	"success" BOOLEAN NOT NULL
checksum	BLOB	"checksum" BLOB NOT NULL
execution_time	BIGINT	"execution_time" BIGINT NOT NULL
audio_chunks		CREATE TABLE audio_chunks (id INTEGER PRIMARY KEY AUTOINCREMENT, file_path TEXT NOT NULL , timestamp TIMESTAMP)
id	INTEGER	"id" INTEGER
file_path	TEXT	"file_path" TEXT NOT NULL
timestamp	TIMESTAMP	"timestamp" TIMESTAMP
audio_tags		CREATE TABLE audio_tags (audio_chunk_id INTEGER NOT NULL, tag_id INTEGER NOT NULL, PRIMARY KEY (audio_chunk_id, tag_id), FOREIGN KEY (audio_chunk_id) REFERENCES audio_chunks(id) ON DELETE CASCADE, FOREIGN KEY (tag_id) REFERENCES tags(id) ON DELETE CASCADE)
audio_chunk_id	INTEGER	"audio_chunk_id" INTEGER NOT NULL
tag_id	INTEGER	"tag_id" INTEGER NOT NULL
audio_transcriptions		CREATE TABLE "audio_transcriptions" (id INTEGER PRIMARY KEY AUTOINCREMENT, audio_chunk_id INTEGER NOT NULL, offset_index INTEGER NOT NULL, timestamp TIMESTAMP NOT NULL, transcription TEXT NOT NULL, device TEXT NOT NULL DEFAULT '' , is_input_device BOOLEAN NOT NULL DEFAULT TRUE, speaker_id INTEGER, transcription_engine TEXT NOT NULL DEFAULT 'Whisper', start_time REAL, end_time REAL, text_length INTEGER, FOREIGN KEY (audio_chunk_id) REFERENCES audio_chunks(id))
id	INTEGER	"id" INTEGER
audio_chunk_id	INTEGER	"audio_chunk_id" INTEGER NOT NULL
offset_index	INTEGER	"offset_index" INTEGER NOT NULL
timestamp	TIMESTAMP	"timestamp" TIMESTAMP NOT NULL
transcription	TEXT	"transcription" TEXT NOT NULL
device	TEXT	"device" TEXT NOT NULL DEFAULT "
is_input_device	BOOLEAN	"is_input_device" BOOLEAN NOT NULL DEFAULT TRUE
speaker_id	INTEGER	"speaker_id" INTEGER
transcription_engine	TEXT	"transcription_engine" TEXT NOT NULL DEFAULT 'Whisper'
start_time	REAL	"start_time" REAL
end_time	REAL	"end_time" REAL
text_length	INTEGER	"text_length" INTEGER
audio_transcriptions_fts		CREATE VIRTUAL TABLE audio_transcriptions_fts USING fts5(transcription, device, audio_chunk_id UNINDEXED, speaker_id, start_time UNINDEXED, end_time UNINDEXED, tokenize='unicode61')

Name	Type	Schema
transcription		"transcription"
device		"device"
audio_chunk_id		"audio_chunk_id"
speaker_id		"speaker_id"
start_time		"start_time"
end_time		"end_time"
audio_transcriptions_fts_config		CREATE TABLE 'audio_transcriptions_fts_config' (k PRIMARY KEY, v) WITHOUT ROWID
k		"k"
v		"v"
audio_transcriptions_fts_content		CREATE TABLE 'audio_transcriptions_fts_content' (id INTEGER PRIMARY KEY, c0, c1, c2, c3, c4, c5)
id	INTEGER	"id" INTEGER
c0		"c0"
c1		"c1"
c2		"c2"
c3		"c3"
c4		"c4"
c5		"c5"
audio_transcriptions_fts_data		CREATE TABLE 'audio_transcriptions_fts_data' (id INTEGER PRIMARY KEY, block BLOB)
id	INTEGER	"id" INTEGER
block	BLOB	"block" BLOB
audio_transcriptions_fts_docsizes		CREATE TABLE 'audio_transcriptions_fts_docsizes' (id INTEGER PRIMARY KEY, sz BLOB)
id	INTEGER	"id" INTEGER
sz	BLOB	"sz" BLOB
audio_transcriptions_fts_idx		CREATE TABLE 'audio_transcriptions_fts_idx' (segid, term, pgno, PRIMARY KEY(segid, term)) WITHOUT ROWID
segid		"segid"
term		"term"
pgno		"pgno"
chunked_text_entries		CREATE TABLE chunked_text_entries (text_id INTEGER NOT NULL, frame_id INTEGER, audio_chunk_id INTEGER, timestamp DATETIME NOT NULL, engine TEXT NOT NULL, chunking_engine TEXT NOT NULL, source TEXT NOT NULL, FOREIGN KEY (text_id) REFERENCES chunked_text_index(text_id), FOREIGN KEY (frame_id) REFERENCES frames(id), FOREIGN KEY (audio_chunk_id) REFERENCES audio_chunks(id))
text_id	INTEGER	"text_id" INTEGER NOT NULL
frame_id	INTEGER	"frame_id" INTEGER
audio_chunk_id	INTEGER	"audio_chunk_id" INTEGER
timestamp	DATETIME	"timestamp" DATETIME NOT NULL

Name	Type	Schema
engine	TEXT	"engine" TEXT NOT NULL
chunking_engine	TEXT	"chunking_engine" TEXT NOT NULL
source	TEXT	"source" TEXT NOT NULL
chunked_text_index		CREATE TABLE chunked_text_index (text_id INTEGER PRIMARY KEY AUTOINCREMENT, text TEXT NOT NULL UNIQUE)
text_id	INTEGER	"text_id" INTEGER
text	TEXT	"text" TEXT NOT NULL UNIQUE
chunked_text_index_fts		CREATE VIRTUAL TABLE chunked_text_index_fts USING fts5(text_id UNINDEXED, text)
text_id		"text_id"
text		"text"
chunked_text_index_fts_config		CREATE TABLE 'chunked_text_index_fts_config'(k PRIMARY KEY, v) WITHOUT ROWID
k		"k"
v		"v"
chunked_text_index_fts_content		CREATE TABLE 'chunked_text_index_fts_content'(id INTEGER PRIMARY KEY, c0, c1)
id	INTEGER	"id" INTEGER
c0		"c0"
c1		"c1"
chunked_text_index_fts_data		CREATE TABLE 'chunked_text_index_fts_data'(id INTEGER PRIMARY KEY, block BLOB)
id	INTEGER	"id" INTEGER
block	BLOB	"block" BLOB
chunked_text_index_fts_docsizes		CREATE TABLE 'chunked_text_index_fts_docsizes'(id INTEGER PRIMARY KEY, sz BLOB)
id	INTEGER	"id" INTEGER
sz	BLOB	"sz" BLOB
chunked_text_index_fts_idx		CREATE TABLE 'chunked_text_index_fts_idx'(segid, term, pgno, PRIMARY KEY(segid, term)) WITHOUT ROWID
segid		"segid"
term		"term"
pgno		"pgno"
frames		CREATE TABLE frames (id INTEGER PRIMARY KEY AUTOINCREMENT, video_chunk_id INTEGER NOT NULL, offset_index INTEGER NOT NULL, timestamp TIMESTAMP NOT NULL, name TEXT, browser_url TEXT DEFAULT NULL, app_name TEXT DEFAULT NULL, window_name TEXT DEFAULT NULL, focused BOOLEAN DEFAULT NULL, FOREIGN KEY (video_chunk_id) REFERENCES video_chunks(id))
id	INTEGER	"id" INTEGER
video_chunk_id	INTEGER	"video_chunk_id" INTEGER NOT NULL
offset_index	INTEGER	"offset_index" INTEGER NOT NULL
timestamp	TIMESTAMP	"timestamp" TIMESTAMP NOT NULL
name	TEXT	"name" TEXT

Name	Type	Schema
browser_url	TEXT	"browser_url" TEXT DEFAULT NULL
app_name	TEXT	"app_name" TEXT DEFAULT NULL
window_name	TEXT	"window_name" TEXT DEFAULT NULL
focused	BOOLEAN	"focused" BOOLEAN DEFAULT NULL
frames_fts		CREATE VIRTUAL TABLE frames_fts USING fts5(name, browser_url, app_name, window_name, focused, id UNINDEXED, tokenize='unicode61')
name		"name"
browser_url		"browser_url"
app_name		"app_name"
window_name		"window_name"
focused		"focused"
id		"id"
frames_fts_config		CREATE TABLE 'frames_fts_config'(k PRIMARY KEY, v) WITHOUT ROWID
k		"k"
v		"v"
frames_fts_content		CREATE TABLE 'frames_fts_content'(id INTEGER PRIMARY KEY, c0, c1, c2, c3, c4, c5)
id	INTEGER	"id" INTEGER
c0		"c0"
c1		"c1"
c2		"c2"
c3		"c3"
c4		"c4"
c5		"c5"
frames_fts_data		CREATE TABLE 'frames_fts_data'(id INTEGER PRIMARY KEY, block BLOB)
id	INTEGER	"id" INTEGER
block	BLOB	"block" BLOB
frames_fts_docszie		CREATE TABLE 'frames_fts_docszie'(id INTEGER PRIMARY KEY, sz BLOB)
id	INTEGER	"id" INTEGER
sz	BLOB	"sz" BLOB
frames_fts_idx		CREATE TABLE 'frames_fts_idx'(segid, term, pgno, PRIMARY KEY(segid, term)) WITHOUT ROWID
segid		"segid"
term		"term"
pgno		"pgno"
friend_wearable_requests		CREATE TABLE friend_wearable_requests (id INTEGER PRIMARY KEY AUTOINCREMENT, request_id TEXT NOT NULL, memory_source TEXT NOT NULL, chunk_id_range TEXT NOT NULL, timestamp_range TEXT NOT NULL, friend_user_id TEXT NOT NULL, created_at DATETIME DEFAULT CURRENT_TIMESTAMP , filtered_text TEXT, structured_response TEXT, response_id TEXT, response_created_at DATETIME, is_successful BOOLEAN NOT NULL DEFAULT TRUE)
id	INTEGER	"id" INTEGER

Name	Type	Schema
request_id	TEXT	"request_id" TEXT NOT NULL
memory_source	TEXT	"memory_source" TEXT NOT NULL
chunk_id_range	TEXT	"chunk_id_range" TEXT NOT NULL
timestamp_range	TEXT	"timestamp_range" TEXT NOT NULL
friend_user_id	TEXT	"friend_user_id" TEXT NOT NULL
created_at	DATETIME	"created_at" DATETIME DEFAULT CURRENT_TIMESTAMP
filtered_text	TEXT	"filtered_text" TEXT
structured_response	TEXT	"structured_response" TEXT
response_id	TEXT	"response_id" TEXT
response_created_at	DATETIME	"response_created_at" DATETIME
is_successful	BOOLEAN	"is_successful" BOOLEAN NOT NULL DEFAULT TRUE
ne_table		CREATE TABLE ne_table (id SERIAL PRIMARY KEY, title TEXT NOT NULL, description TEXT, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP)
id	SERIAL	"id" SERIAL
title	TEXT	"title" TEXT NOT NULL
description	TEXT	"description" TEXT
created_at	TIMESTAMP	"created_at" TIMESTAMP DEFAULT CURRENT_TIMESTAMP
ocr_text		CREATE TABLE "ocr_text" (frame_id INTEGER NOT NULL, text TEXT NOT NULL, text_json TEXT, app_name TEXT NOT NULL DEFAULT '', ocr_engine TEXT NOT NULL DEFAULT 'unknown', window_name TEXT, focused BOOLEAN DEFAULT FALSE, text_length INTEGER)
frame_id	INTEGER	"frame_id" INTEGER NOT NULL
text	TEXT	"text" TEXT NOT NULL
text_json	TEXT	"text_json" TEXT
app_name	TEXT	"app_name" TEXT NOT NULL DEFAULT ''
ocr_engine	TEXT	"ocr_engine" TEXT NOT NULL DEFAULT 'unknown'
window_name	TEXT	"window_name" TEXT
focused	BOOLEAN	"focused" BOOLEAN DEFAULT FALSE
text_length	INTEGER	"text_length" INTEGER
ocr_text_embeddings		CREATE TABLE ocr_text_embeddings (id INTEGER PRIMARY KEY AUTOINCREMENT, frame_id INTEGER NOT NULL, embedding BLOB NOT NULL, -- Store embeddings as BLOB since they're float arrays created_at DATETIME DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (frame_id) REFERENCES frames(id) ON DELETE CASCADE)
id	INTEGER	"id" INTEGER
frame_id	INTEGER	"frame_id" INTEGER NOT NULL
embedding	BLOB	"embedding" BLOB NOT NULL
created_at	DATETIME	"created_at" DATETIME DEFAULT CURRENT_TIMESTAMP
ocr_text_fts		CREATE VIRTUAL TABLE ocr_text_fts USING fts5(text, app_name, window_name, frame_id UNINDEXED, tokenize='unicode61')
text		"text"
app_name		"app_name"
window_name		"window_name"
frame_id		"frame_id"

Name	Type	Schema
ocr_text_fts_config		CREATE TABLE 'ocr_text_fts_config'(k PRIMARY KEY, v) WITHOUT ROWID
k		"k"
v		"v"
ocr_text_fts_content		CREATE TABLE 'ocr_text_fts_content'(id INTEGER PRIMARY KEY, c0, c1, c2, c3)
id	INTEGER	"id" INTEGER
c0		"c0"
c1		"c1"
c2		"c2"
c3		"c3"
ocr_text_fts_data		CREATE TABLE 'ocr_text_fts_data'(id INTEGER PRIMARY KEY, block BLOB)
id	INTEGER	"id" INTEGER
block	BLOB	"block" BLOB
ocr_text_fts_docsizes		CREATE TABLE 'ocr_text_fts_docsizes'(id INTEGER PRIMARY KEY, sz BLOB)
id	INTEGER	"id" INTEGER
sz	BLOB	"sz" BLOB
ocr_text_fts_idx		CREATE TABLE 'ocr_text_fts_idx'(segid, term, pgno, PRIMARY KEY(segid, term)) WITHOUT ROWID
segid		"segid"
term		"term"
pgno		"pgno"
speaker_embeddings		CREATE TABLE speaker_embeddings (id INTEGER PRIMARY KEY AUTOINCREMENT, embedding FLOAT[512] NOT NULL check(typeof(embedding) == 'blob' and vec_length(embedding) == 512), speaker_id INTEGER REFERENCES speakers(id))
id	INTEGER	"id" INTEGER
embedding	FLOAT 512	"embedding" FLOAT 512 NOT NULL CHECK(typeof("embedding") == 'blob' AND vec_length("embedding") == 512)
speaker_id	INTEGER	"speaker_id" INTEGER
speakers		CREATE TABLE speakers (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, metadata JSON , hallucination BOOLEAN DEFAULT FALSE)
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT
metadata	JSON	"metadata" JSON
hallucination	BOOLEAN	"hallucination" BOOLEAN DEFAULT FALSE
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
tags		CREATE TABLE tags (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL UNIQUE, created_at DATETIME DEFAULT CURRENT_TIMESTAMP)
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL UNIQUE
created_at	DATETIME	"created_at" DATETIME DEFAULT CURRENT_TIMESTAMP

Name	Type	Schema
ui_monitoring		CREATE TABLE ui_monitoring (id INTEGER PRIMARY KEY AUTOINCREMENT, text_output TEXT NOT NULL, timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP, app TEXT NOT NULL, window TEXT NOT NULL, initial_traversal_at DATETIME , text_length INTEGER)
id	INTEGER	"id" INTEGER
text_output	TEXT	"text_output" TEXT NOT NULL
timestamp	DATETIME	"timestamp" DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
app	TEXT	"app" TEXT NOT NULL
window	TEXT	"window" TEXT NOT NULL
initial_traversal_at	DATETIME	"initial_traversal_at" DATETIME
text_length	INTEGER	"text_length" INTEGER
ui_monitoring_fts		CREATE VIRTUAL TABLE ui_monitoring_fts USING fts5(text_output, app, window, ui_id UNINDEXED, tokenize='unicode61')
text_output		"text_output"
app		"app"
window		"window"
ui_id		"ui_id"
ui_monitoring_fts_config		CREATE TABLE 'ui_monitoring_fts_config'(k PRIMARY KEY, v WITHOUT ROWID)
k		"k"
v		"v"
ui_monitoring_fts_content		CREATE TABLE 'ui_monitoring_fts_content'(id INTEGER PRIMARY KEY, c0, c1, c2, c3)
id	INTEGER	"id" INTEGER
c0		"c0"
c1		"c1"
c2		"c2"
c3		"c3"
ui_monitoring_fts_data		CREATE TABLE 'ui_monitoring_fts_data'(id INTEGER PRIMARY KEY, block BLOB)
id	INTEGER	"id" INTEGER
block	BLOB	"block" BLOB
ui_monitoring_fts_docszie		CREATE TABLE 'ui_monitoring_fts_docszie'(id INTEGER PRIMARY KEY, sz BLOB)
id	INTEGER	"id" INTEGER
sz	BLOB	"sz" BLOB
ui_monitoring_fts_idx		CREATE TABLE 'ui_monitoring_fts_idx'(segid, term, pgno, PRIMARY KEY(segid, term)) WITHOUT ROWID
segid		"segid"
term		"term"
pgno		"pgno"
ui_monitoring_tags		CREATE TABLE ui_monitoring_tags (ui_monitoring_id INTEGER NOT NULL, tag_id INTEGER NOT NULL, PRIMARY KEY (ui_monitoring_id, tag_id), FOREIGN KEY (ui_monitoring_id) REFERENCES ui_monitoring(id) ON

Name	Type	Schema
		DELETE CASCADE, FOREIGN KEY (tag_id) REFERENCES tags(id) ON DELETE CASCADE)
ui_monitoring_id	INTEGER	"ui_monitoring_id" INTEGER NOT NULL
tag_id	INTEGER	"tag_id" INTEGER NOT NULL
video_chunks		CREATE TABLE video_chunks (id INTEGER PRIMARY KEY AUTOINCREMENT, file_path TEXT NOT NULL , device_name TEXT NOT NULL DEFAULT '')
id	INTEGER	"id" INTEGER
file_path	TEXT	"file_path" TEXT NOT NULL
device_name	TEXT	"device_name" TEXT NOT NULL DEFAULT "
vision_tags		CREATE TABLE vision_tags (vision_id INTEGER NOT NULL, tag_id INTEGER NOT NULL, PRIMARY KEY (vision_id, tag_id), FOREIGN KEY (vision_id) REFERENCES frames(id) ON DELETE CASCADE, FOREIGN KEY (tag_id) REFERENCES tags(id) ON DELETE CASCADE)
vision_id	INTEGER	"vision_id" INTEGER NOT NULL
tag_id	INTEGER	"tag_id" INTEGER NOT NULL

Indices (34)

Name	Type	Schema
idx_audio_chunks_timestamp		CREATE INDEX idx_audio_chunks_timestamp ON audio_chunks(timestamp)
timestamp		"timestamp"
idx_audio_tags_audio_chunk_id		CREATE INDEX idx_audio_tags_audio_chunk_id ON audio_tags(audio_chunk_id)
audio_chunk_id		"audio_chunk_id"
idx_audio_tags_tag_id		CREATE INDEX idx_audio_tags_tag_id ON audio_tags(tag_id)
tag_id		"tag_id"
idx_audio_transcriptions_audio_chunk_id		CREATE INDEX idx_audio_transcriptions_audio_chunk_id ON audio_transcriptions(audio_chunk_id)
audio_chunk_id		"audio_chunk_id"
idx_audio_transcriptions_audio_chunk_id_timestamp		CREATE INDEX idx_audio_transcriptions_audio_chunk_id_timestamp ON audio_transcriptions(audio_chunk_id, timestamp)
audio_chunk_id		"audio_chunk_id"
timestamp		"timestamp"
idx_audio_transcriptions_length		CREATE INDEX idx_audio_transcriptions_length ON audio_transcriptions (text_length)
text_length		"text_length"
idx_audio_transcriptions_timestamp		CREATE INDEX idx_audio_transcriptions_timestamp ON audio_transcriptions(timestamp)
timestamp		"timestamp"
idx_audio_transcriptions_		CREATE INDEX idx_audio_transcriptions_transcription ON audio_transcriptions(transcription)

Name	Type	Schema
transcription		
transcription		"transcription"
idx_chunked_text_index_text		CREATE UNIQUE INDEX idx_chunked_text_index_text ON chunked_text_index(text)
text		"text"
idx_frames_app_name		CREATE INDEX idx_frames_app_name ON frames(app_name)
app_name		"app_name"
idx_frames_app_window		CREATE INDEX idx_frames_app_window ON frames(app_name, window_name)
app_name		"app_name"
window_name		"window_name"
idx_frames_browser_url		CREATE INDEX idx_frames_browser_url ON frames(browser_url)
browser_url		"browser_url"
idx_frames_focused		CREATE INDEX idx_frames_focused ON frames(focused)
focused		"focused"
idx_frames_timestamp		CREATE INDEX idx_frames_timestamp ON frames(timestamp)
timestamp		"timestamp"
idx_frames_timestamp_of_fset_index		CREATE INDEX idx_frames_timestamp_offset_index ON frames(timestamp, offset_index)
timestamp		"timestamp"
offset_index		"offset_index"
idx_frames_video_chunk_id		CREATE INDEX idx_frames_video_chunk_id ON frames(video_chunk_id)
video_chunk_id		"video_chunk_id"
idx_frames_video_chunk_id_timestamp		CREATE INDEX idx_frames_video_chunk_id_timestamp ON frames(video_chunk_id, timestamp)
video_chunk_id		"video_chunk_id"
timestamp		"timestamp"
idx_frames_window_name		CREATE INDEX idx_frames_window_name ON frames(window_name)
window_name		"window_name"
idx_friend_wearable_requests_friend_user_id		CREATE INDEX idx_friend_wearable_requests_friend_user_id ON friend_wearable_requests(friend_user_id)
friend_user_id		"friend_user_id"
idx_friend_wearable_requests_memory_source		CREATE INDEX idx_friend_wearable_requests_memory_source ON friend_wearable_requests(memory_source)
memory_source		"memory_source"
idx_ocr_text_frame_app_window		CREATE INDEX idx_ocr_text_frame_app_window ON ocr_text(frame_id, app_name, window_name)
frame_id		"frame_id"

Name	Type	Schema
app_name		"app_name"
window_name		"window_name"
idx_ocr_text_frame_id		CREATE INDEX idx_ocr_text_frame_id ON ocr_text(frame_id)
frame_id		"frame_id"
idx_ocr_text_length		CREATE INDEX idx_ocr_text_length ON ocr_text (text_length)
text_length		"text_length"
idx_ui_monitoring_app		CREATE INDEX idx_ui_monitoring_app ON ui_monitoring(app)
app		"app"
idx_ui_monitoring_app_window		CREATE INDEX idx_ui_monitoring_app_window ON ui_monitoring(app, window)
app		"app"
window		"window"
idx_ui_monitoring_length		CREATE INDEX idx_ui_monitoring_length ON ui_monitoring (text_length)
text_length		"text_length"
idx_ui_monitoring_tags_tag_id		CREATE INDEX idx_ui_monitoring_tags_tag_id ON ui_monitoring_tags(tag_id)
tag_id		"tag_id"
idx_ui_monitoring_tags_ui_id		CREATE INDEX idx_ui_monitoring_tags_ui_id ON ui_monitoring_tags(ui_monitoring_id)
ui_monitoring_id		"ui_monitoring_id"
idx_ui_monitoring_text_output		CREATE INDEX idx_ui_monitoring_text_output ON ui_monitoring(text_output)
text_output		"text_output"
idx_ui_monitoring_timestamp		CREATE INDEX idx_ui_monitoring_timestamp ON ui_monitoring(timestamp)
timestamp		"timestamp"
idx_ui_monitoring_timestamp_app		CREATE INDEX idx_ui_monitoring_timestamp_app ON ui_monitoring(timestamp, app)
timestamp		"timestamp"
app		"app"
idx_ui_monitoring_window		CREATE INDEX idx_ui_monitoring_window ON ui_monitoring(window)
window		"window"
idx_vision_tags_tag_id		CREATE INDEX idx_vision_tags_tag_id ON vision_tags(tag_id)
tag_id		"tag_id"
idx_vision_tags_vision_id		CREATE INDEX idx_vision_tags_vision_id ON vision_tags(vision_id)
vision_id		"vision_id"

Views (0)

Name	Type	Schema
------	------	--------

Triggers (12)

Name	Type	Schema
audio_transcriptions_ai		<pre>CREATE TRIGGER audio_transcriptions_ai AFTER INSERT ON audio_transcriptions WHEN NEW.transcription IS NOT NULL AND NEW.transcription != '' AND NEW.audio_chunk_id IS NOT NULL BEGIN INSERT OR IGNORE INTO audio_transcriptions_fts(transcription, device, audio_chunk_id, speaker_id, start_time, end_time) VALUES (NEW.transcription, COALESCE(NEW.device, ''), NEW.audio_chunk_id, NEW.speaker_id, NEW.start_time, NEW.end_time); END</pre>
audio_transcriptions_delete		<pre>CREATE TRIGGER audio_transcriptions_delete AFTER DELETE ON audio_transcriptions BEGIN DELETE FROM audio_transcriptions_fts WHERE audio_chunk_id = OLD.audio_chunk_id; END</pre>
audio_transcriptions_update		<pre>CREATE TRIGGER audio_transcriptions_update AFTER UPDATE ON audio_transcriptions WHEN NEW.transcription IS NOT NULL AND NEW.transcription != '' AND OLD.audio_chunk_id IS NOT NULL BEGIN UPDATE audio_transcriptions_fts SET transcription = NEW.transcription, device = COALESCE(NEW.device, ''), start_time = NEW.start_time, end_time = NEW.end_time WHERE audio_chunk_id = OLD.audio_chunk_id; END</pre>
frames_ad		<pre>CREATE TRIGGER frames_ad AFTER DELETE ON frames BEGIN DELETE FROM frames_fts WHERE id = OLD.id; END</pre>
frames_ai		<pre>CREATE TRIGGER frames_ai AFTER INSERT ON frames BEGIN INSERT INTO frames_fts(id, name, browser_url, app_name, window_name, focused) VALUES (NEW.id, COALESCE(NEW.name, ''), COALESCE(NEW.browser_url, ''), COALESCE(NEW.app_name, ''), COALESCE(NEW.window_name, ''), COALESCE(NEW.focused, 0)); END</pre>
frames_au		<pre>CREATE TRIGGER frames_au AFTER UPDATE ON frames WHEN (NEW.name IS NOT NULL AND NEW.name != '') OR (NEW.browser_url IS NOT NULL AND NEW.browser_url != '') OR (NEW.app_name IS NOT NULL AND NEW.app_name != '') OR (NEW.window_name IS NOT NULL AND NEW.window_name != '') OR (NEW.focused IS NOT NULL) BEGIN INSERT OR REPLACE INTO frames_fts(id, name, browser_url, app_name, window_name, focused) VALUES (NEW.id, COALESCE(NEW.name, ''), COALESCE(NEW.browser_url, ''), COALESCE(NEW.app_name, ''), COALESCE(NEW.window_name, ''), COALESCE(NEW.focused, 0)); END</pre>
ocr_text_ai		<pre>CREATE TRIGGER ocr_text_ai AFTER INSERT ON ocr_text WHEN NEW.text IS NOT NULL AND NEW.text != '' AND NEW.frame_id IS NOT NULL BEGIN INSERT OR IGNORE INTO ocr_text_fts(frame_id, text, app_name, window_name) VALUES (NEW.frame_id, NEW.text, COALESCE(NEW.app_name, ''), COALESCE(NEW.window_name, '')); END</pre>
ocr_text_delete		<pre>CREATE TRIGGER ocr_text_delete AFTER DELETE ON ocr_text BEGIN DELETE FROM ocr_text_fts WHERE frame_id = OLD.frame_id; END</pre>
ocr_text_update		<pre>CREATE TRIGGER ocr_text_update AFTER UPDATE ON ocr_text WHEN NEW.text IS NOT NULL AND NEW.text != '' AND OLD.frame_id IS NOT NULL BEGIN UPDATE ocr_text_fts SET text = NEW.text, app_name = COALESCE(NEW.app_name, ''), window_name = COALESCE(NEW.window_name, '') WHERE frame_id = OLD.frame_id; END</pre>
ui_monitoring_ai		<pre>CREATE TRIGGER ui_monitoring_ai AFTER INSERT ON</pre>

Name	Type	Schema
		ui_monitoring WHEN NEW.text_output IS NOT NULL AND NEW.text_output != '' AND NEW.id IS NOT NULL BEGIN INSERT OR IGNORE INTO ui_monitoring_fts(ui_id, text_output, app, window) VALUES (NEW.id, NEW.text_output, COALESCE(NEW.app, ''), COALESCE(NEW.window, '')); END
ui_monitoring_delete		CREATE TRIGGER ui_monitoring_delete AFTER DELETE ON ui_monitoring BEGIN DELETE FROM ui_monitoring_fts WHERE ui_id = OLD.id; END
ui_monitoring_update		CREATE TRIGGER ui_monitoring_update AFTER UPDATE ON ui_monitoring WHEN NEW.text_output IS NOT NULL AND NEW.text_output != '' AND OLD.id IS NOT NULL BEGIN UPDATE ui_monitoring_fts SET text_output = NEW.text_output, app = COALESCE(NEW.app, ''), window = COALESCE(NEW.window, '') WHERE ui_id = OLD.id; END