Tables (23)

Name	Type	Schema
admin_logs		CREATE TABLE admin_logs (id INTEGER PRIMARY KEY AUTOINCREMENT, admin_id INTEGER, action TEXT, target_user_id INTEGER, timestamp DATETIME DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (admin_id) REFERENCES users(id), FOREIGN KEY (target_user_id) REFERENCES users(id))
id	INTEGER	"id" INTEGER
admin_id	INTEGER	"admin_id" INTEGER
action	TEXT	"action" TEXT
target_user_id	INTEGER	"target_user_id" INTEGER
timestamp	DATETIME	"timestamp" DATETIME DEFAULT CURRENT_TIMESTAMP
age_categories		CREATE TABLE age_categories (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT UNIQUE NOT NULL, min_age INTEGER NOT NULL, max_age INTEGER NOT NULL , notes VARCHAR(100))
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL UNIQUE
min_age	INTEGER	"min_age" INTEGER NOT NULL
max_age	INTEGER	"max_age" INTEGER NOT NULL
notes	VARCHAR(100)	"notes" VARCHAR(100)
athlete_last_participation		CREATE TABLE athlete_last_participation (athlete_id INTEGER PRIMARY KEY, last_national_participation_date DATETIME, FOREIGN KEY (athlete_id) REFERENCES athletes(id))
athlete_id	INTEGER	"athlete_id" INTEGER
last_national_participation_date	DATETIME	"last_national_participation_date" DATETIME
athlete_merge_map		CREATE TABLE athlete_merge_map (old_id INTEGER, new_id INTEGER)
old_id	INTEGER	"old_id" INTEGER
new_id	INTEGER	"new_id" INTEGER
athletes		CREATE TABLE athletes (id INTEGER PRIMARY KEY AUTOINCREMENT, full_name TEXT NOT NULL, birth_date DATE NOT NULL, gender TEXT CHECK (gender IN ('male', 'female')) NOT NULL, age_category_id INTEGER, weight_category_id INTEGER, total_points INTEGER DEFAULT 0, club_id INTEGER, closing_points_2024 REAL DEFAULT 0, starting_points_2025 REAL DEFAULT 0, kata_points REAL DEFAULT 0, kumite_points REAL DEFAULT 0, last_reviewed_at DATETIME, review_status TEXT DEFAULT 'pending', FOREIGN KEY (age_category_id) REFERENCES age_categories(id), FOREIGN KEY (weight_category_id) REFERENCES weight_categories(id))
id	INTEGER	"id" INTEGER
full_name	TEXT	"full_name" TEXT NOT NULL
birth_date	DATE	"birth_date" DATE NOT NULL
gender	TEXT	"gender" TEXT NOT NULL CHECK("gender" IN ('male', 'female'))
age_category_id	INTEGER	"age_category_id" INTEGER
weight_category_id	INTEGER	"weight_category_id" INTEGER

Name	Type	Schema
total_points	INTEGER	"total_points" INTEGER DEFAULT 0
club_id	INTEGER	"club_id" INTEGER
closing_points_2024	REAL	"closing_points_2024" REAL DEFAULT 0
starting_points_2025	REAL	"starting_points_2025" REAL DEFAULT 0
kata_points	REAL	"kata_points" REAL DEFAULT 0
kumite_points	REAL	"kumite_points" REAL DEFAULT 0
last_reviewed_at	DATETIME	"last_reviewed_at" DATETIME
review_status	TEXT	"review_status" TEXT DEFAULT 'pending'
category_changes		CREATE TABLE category_changes (id INTEGER PRIMARY KEY AUTOINCREMENT, athlete_id INTEGER, old_age_category_id INTEGER, new_age_category_id INTEGER, old_weight_category_id INTEGER, new_weight_category_id INTEGER, change_date DATETIME DEFAULT CURRENT_TIMESTAMP, reason TEXT, FOREIGN KEY (athlete_id) REFERENCES athletes(id))
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER
old_age_category_id	INTEGER	"old_age_category_id" INTEGER
new_age_category_id	INTEGER	"new_age_category_id" INTEGER
old_weight_category_id	INTEGER	"old_weight_category_id" INTEGER
new_weight_category_id	INTEGER	"new_weight_category_id" INTEGER
change_date	DATETIME	"change_date" DATETIME DEFAULT CURRENT_TIMESTAMP
reason	TEXT	"reason" TEXT
clubs		CREATE TABLE clubs (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL UNIQUE)
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL UNIQUE
participation_points		CREATE TABLE participation_points (id INTEGER PRIMARY KEY, points INTEGER)
id	INTEGER	"id" INTEGER
points	INTEGER	"points" INTEGER
penalties		CREATE TABLE penalties (id INTEGER PRIMARY KEY AUTOINCREMENT, athlete_id INTEGER NOT NULL, result_id INTEGER, season_year INTEGER, penalty_points INTEGER NOT NULL, reason TEXT, FOREIGN KEY (athlete_id) REFERENCES athletes(id), FOREIGN KEY (result_id) REFERENCES results(id))
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER NOT NULL
result_id	INTEGER	"result_id" INTEGER
season_year	INTEGER	"season_year" INTEGER
penalty_points	INTEGER	"penalty_points" INTEGER NOT NULL
reason	TEXT	"reason" TEXT
placement_points		CREATE TABLE placement_points (placement INTEGER PRIMARY KEY, points INTEGER)
placement	INTEGER	"placement" INTEGER
points	INTEGER	"points" INTEGER
points_history		CREATE TABLE points_history (id INTEGER PRIMARY

Name	Type	Schema
		KEY AUTOINCREMENT, athlete_id INTEGER, date DATETIME DEFAULT CURRENT_TIMESTAMP, points_before REAL DEFAULT 0, points_after REAL DEFAULT 0, description TEXT, season_year INTEGER, delta REAL DEFAULT 0, total_after REAL DEFAULT 0, source TEXT DEFAULT 'manual', discipline TEXT CHECK (discipline IN ('KATA', 'KUMITE')) DEFAULT 'KUMITE', FOREIGN KEY (athlete_id) REFERENCES athletes(id))
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER
date	DATETIME	"date" DATETIME DEFAULT CURRENT_TIMESTAMP
points_before	REAL	"points_before" REAL DEFAULT 0
points_after	REAL	"points_after" REAL DEFAULT 0
description	TEXT	"description" TEXT
season_year	INTEGER	"season_year" INTEGER
delta	REAL	"delta" REAL DEFAULT 0
total_after	REAL	"total_after" REAL DEFAULT 0
source	TEXT	"source" TEXT DEFAULT 'manual'
discipline	TEXT	"discipline" TEXT DEFAULT 'KUMITE' CHECK("discipline" IN ('KATA', 'KUMITE'))
points_history_old		CREATE TABLE "points_history_old" (id INTEGER PRIMARY KEY AUTOINCREMENT, athlete_id INTEGER, date DATETIME NOT NULL, points INTEGER, reason TEXT, category_id INTEGER, weight_id INTEGER, delta REAL DEFAULT 0, source TEXT DEFAULT 'system', FOREIGN KEY (athlete_id) REFERENCES athletes(id))
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER
date	DATETIME	"date" DATETIME NOT NULL
points	INTEGER	"points" INTEGER
reason	TEXT	"reason" TEXT
category_id	INTEGER	"category_id" INTEGER
weight_id	INTEGER	"weight_id" INTEGER
delta	REAL	"delta" REAL DEFAULT 0
source	TEXT	"source" TEXT DEFAULT 'system'
ranking_logs		CREATE TABLE ranking_logs (id INTEGER PRIMARY KEY AUTOINCREMENT, admin_name TEXT DEFAULT 'system', recalculated INTEGER DEFAULT 0, timestamp DATETIME DEFAULT CURRENT_TIMESTAMP, notes TEXT)
id	INTEGER	"id" INTEGER
admin_name	TEXT	"admin_name" TEXT DEFAULT 'system'
recalculated	INTEGER	"recalculated" INTEGER DEFAULT 0
timestamp	DATETIME	"timestamp" DATETIME DEFAULT CURRENT_TIMESTAMP
notes	TEXT	"notes" TEXT
results		CREATE TABLE results (id INTEGER PRIMARY KEY AUTOINCREMENT, athlete_id INTEGER, tournament_id INTEGER, age_category_id INTEGER, weight_category_id INTEGER, placement INTEGER,

Name	Type	Schema
		wins INTEGER DEFAULT 0, participated INTEGER DEFAULT 0, approved INTEGER DEFAULT 0, countries_participated INTEGER DEFAULT 1, points_earned INTEGER DEFAULT 0, penalty_points INTEGER DEFAULT 0, approved_up_age INTEGER DEFAULT 0, raw_points REAL DEFAULT 0, season_year INTEGER, event_type TEXT DEFAULT 'kumite', FOREIGN KEY (athlete_id) REFERENCES athletes(id), FOREIGN KEY (tournament_id) REFERENCES tournaments(id), FOREIGN KEY (age_category_id) REFERENCES age_categories(id), FOREIGN KEY (weight_category_id) REFERENCES weight_categories(id))
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER
tournament_id	INTEGER	"tournament_id" INTEGER
age_category_id	INTEGER	"age_category_id" INTEGER
weight_category_id	INTEGER	"weight_category_id" INTEGER
placement	INTEGER	"placement" INTEGER
wins	INTEGER	"wins" INTEGER DEFAULT 0
participated	INTEGER	"participated" INTEGER DEFAULT 0
approved	INTEGER	"approved" INTEGER DEFAULT 0
countries_participated	INTEGER	"countries_participated" INTEGER DEFAULT 1
points_earned	INTEGER	"points_earned" INTEGER DEFAULT 0
penalty_points	INTEGER	"penalty_points" INTEGER DEFAULT 0
approved_up_age	INTEGER	"approved_up_age" INTEGER DEFAULT 0
raw_points	REAL	"raw_points" REAL DEFAULT 0
season_year	INTEGER	"season_year" INTEGER
event_type	TEXT	"event_type" TEXT DEFAULT 'kumite'
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
tournament_participants		CREATE TABLE tournament_participants (id INTEGER PRIMARY KEY AUTOINCREMENT, tournament_id INTEGER, athlete_id INTEGER, result TEXT)
id	INTEGER	"id" INTEGER
tournament_id	INTEGER	"tournament_id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER
result	TEXT	"result" TEXT
tournament_registrations		CREATE TABLE tournament_registrations (id INTEGER PRIMARY KEY AUTOINCREMENT, tournament_id INTEGER NOT NULL, athlete_id INTEGER NOT NULL, approved INTEGER DEFAULT 0, override_allowed INTEGER DEFAULT 0, approved_by TEXT, registered_at DATETIME DEFAULT CURRENT_TIMESTAMP, requires_approval INTEGER DEFAULT 0, UNIQUE (tournament_id, athlete_id))
id	INTEGER	"id" INTEGER
tournament_id	INTEGER	"tournament_id" INTEGER NOT NULL
athlete_id	INTEGER	"athlete_id" INTEGER NOT NULL

Name	Type	Schema
approved	INTEGER	"approved" INTEGER DEFAULT 0
override_allowed	INTEGER	"override_allowed" INTEGER DEFAULT 0
approved_by	TEXT	"approved_by" TEXT
registered_at	DATETIME	"registered_at" DATETIME DEFAULT CURRENT_TIMESTAMP
requires_approval	INTEGER	"requires_approval" INTEGER DEFAULT 0
tournaments		`CREATE TABLE tournaments (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, date DATETIME NOT NULL, location TEXT, type TEXT, difficulty_multiplier REAL DEFAULT 1, is_international INTEGER DEFAULT 0, requires_approval INTEGER DEFAULT 0 , exclude_from_scoring INTEGER DEFAULT 0, min_countries_participating INTEGER DEFAULT 3, min_age INTEGER, max_age INTEGER, tournament_type TEXT DEFAULT 'NATIONAL_CHAMPIONSHIP', season_year INTEGER DEFAULT 2025)`
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT NOT NULL
date	DATETIME	"date" DATETIME NOT NULL
location	TEXT	"location" TEXT
type	TEXT	"type" TEXT
difficulty_multiplier	REAL	"difficulty_multiplier" REAL DEFAULT 1
is_international	INTEGER	"is_international" INTEGER DEFAULT 0
requires_approval	INTEGER	"requires_approval" INTEGER DEFAULT 0
exclude_from_scoring	INTEGER	"exclude_from_scoring" INTEGER DEFAULT 0
min_countries_participating	INTEGER	"min_countries_participating" INTEGER DEFAULT 3
min_age	INTEGER	"min_age" INTEGER
max_age	INTEGER	"max_age" INTEGER
tournament_type	TEXT	"tournament_type" TEXT DEFAULT 'NATIONAL_CHAMPIONSHIP'
season_year	INTEGER	"season_year" INTEGER DEFAULT 2025
user_sessions		`CREATE TABLE user_sessions ( id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER, login_time DATETIME DEFAULT CURRENT_TIMESTAMP, logout_time DATETIME, ip_address TEXT, FOREIGN KEY (user_id) REFERENCES users(id) )`
id	INTEGER	"id" INTEGER
user_id	INTEGER	"user_id" INTEGER
login_time	DATETIME	"login_time" DATETIME DEFAULT CURRENT_TIMESTAMP
logout_time	DATETIME	"logout_time" DATETIME
ip_address	TEXT	"ip_address" TEXT
users		`CREATE TABLE users ( id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT UNIQUE NOT NULL, password TEXT NOT NULL, role TEXT NOT NULL CHECK (role IN ('admin', 'user', 'rules', 'superuser')), reset_token TEXT, reset_expires DATETIME , isActive INTEGER NOT NULL DEFAULT 0)`
id	INTEGER	"id" INTEGER
username	TEXT	"username" TEXT NOT NULL UNIQUE
password	TEXT	"password" TEXT NOT NULL
role	TEXT	"role" TEXT NOT NULL CHECK("role" IN ('admin', 'user', 'rules',

Name	Type	Schema
		'superuser'))
reset_token	TEXT	"reset_token" TEXT
reset_expires	DATETIME	"reset_expires" DATETIME
isActive	INTEGER	"isActive" INTEGER NOT NULL DEFAULT 0
weight_categories		`CREATE TABLE weight_categories ( id INTEGER PRIMARY KEY AUTOINCREMENT, age_category_id INTEGER, gender TEXT CHECK (gender IN ('male', 'female')) NOT NULL, name TEXT NOT NULL, FOREIGN KEY (age_category_id) REFERENCES age_categories(id) )`
id	INTEGER	"id" INTEGER
age_category_id	INTEGER	"age_category_id" INTEGER
gender	TEXT	"gender" TEXT NOT NULL CHECK("gender" IN ('male', 'female'))
name	TEXT	"name" TEXT NOT NULL
win_points		`CREATE TABLE win_points ( wins INTEGER PRIMARY KEY, points INTEGER )`
wins	INTEGER	"wins" INTEGER
points	INTEGER	"points" INTEGER
yearly_points		`CREATE TABLE yearly_points (id INTEGER PRIMARY KEY AUTOINCREMENT, athlete_id INTEGER NOT NULL, year INTEGER NOT NULL, closing_points REAL DEFAULT 0, starting_points REAL DEFAULT 0, created_at DATETIME DEFAULT CURRENT_TIMESTAMP, closing_raw_points REAL DEFAULT 0, UNIQUE (athlete_id, year), FOREIGN KEY (athlete_id) REFERENCES athletes(id))`
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER NOT NULL
year	INTEGER	"year" INTEGER NOT NULL
closing_points	REAL	"closing_points" REAL DEFAULT 0
starting_points	REAL	"starting_points" REAL DEFAULT 0
created_at	DATETIME	"created_at" DATETIME DEFAULT CURRENT_TIMESTAMP
closing_raw_points	REAL	"closing_raw_points" REAL DEFAULT 0

Indices (2)

Name	Type	Schema
idx_results_pair_temp		`CREATE INDEX idx_results_pair_temp ON results (athlete_id, tournament_id)`
athlete_id		"athlete_id"
tournament_id		"tournament_id"
idx_unique_athlete_tournament		`CREATE UNIQUE INDEX idx_unique_athlete_tournament ON results (athlete_id, tournament_id)`
athlete_id		"athlete_id"
tournament_id		"tournament_id"

Views (7)

Name	Type	Schema
athlete_details		CREATE VIEW athlete_details AS SELECT a.id, a.full_name, a.birth_date, a.gender, ac.name AS age_category, wc.name AS weight_category, a.total_points FROM athletes a LEFT JOIN age_categories ac ON a.age_category_id = ac.id LEFT JOIN weight_categories wc ON a.weight_category_id = wc.id
id	INTEGER	"id" INTEGER
full_name	TEXT	"full_name" TEXT
birth_date	DATE	"birth_date" DATE
gender	TEXT	"gender" TEXT
age_category	TEXT	"age_category" TEXT
weight_category	TEXT	"weight_category" TEXT
total_points	INTEGER	"total_points" INTEGER
points_history_details		CREATE VIEW points_history_details AS SELECT ph.id, ph.athlete_id, a.full_name, ph.date, ph.points, ph.reason FROM "points_history_old" ph JOIN athletes a ON ph.athlete_id = a.id
id	INTEGER	"id" INTEGER
athlete_id	INTEGER	"athlete_id" INTEGER
full_name	TEXT	"full_name" TEXT
date	DATETIME	"date" DATETIME
points	INTEGER	"points" INTEGER
reason	TEXT	"reason" TEXT
points_review		CREATE VIEW points_review AS SELECT a.id AS athlete_id, a.full_name, ROUND(IFNULL(a.total_points, 0), 3) AS athlete_total, ROUND(IFNULL(ph.total_after, 0), 3) AS history_total, ROUND(IFNULL(a.total_points, 0) - IFNULL(ph.total_after, 0), 3) AS difference, IFNULL(ph.season_year, 2025) AS season_year, IFNULL(ph.description, '') AS last_description FROM athletes a LEFT JOIN (SELECT athlete_id, total_after, season_year, description, MAX(date) AS max_date FROM points_history GROUP BY athlete_id) ph ON ph.athlete_id = a.id ORDER BY a.full_name COLLATE NOCASE
athlete_id	INTEGER	"athlete_id" INTEGER
full_name	TEXT	"full_name" TEXT
athlete_total		"athlete_total"
history_total		"history_total"
difference		"difference"
season_year		"season_year"
last_description		"last_description"
ranking_combined		CREATE VIEW ranking_combined AS SELECT a.id AS athlete_id, a.full_name, (SELECT SUM(points_earned) FROM results r1 WHERE r1.athlete_id = a.id AND r1.event_type = 'kumite') AS kumite_points, (SELECT SUM(points_earned) FROM results r2 WHERE r2.athlete_id = a.id AND r2.event_type = 'kata') AS kata_points, (COALESCE(kumite_points,0) + COALESCE(kata_points,0)) AS total_points FROM athletes a ORDER BY total_points DESC
athlete_id	INTEGER	"athlete_id" INTEGER
full_name	TEXT	"full_name" TEXT
kumite_points		"kumite_points"

Name	Type	Schema
kata_points		"kata_points"
total_points		"total_points"
ranking_kata		CREATE VIEW ranking_kata AS SELECT a.id AS athlete_id, a.full_name, a.gender, ac.name AS age_category, wc.name AS weight_category, SUM(r.points_earned) AS total_points FROM athletes a JOIN results r ON a.id = r.athlete_id LEFT JOIN age_categories ac ON r.age_category_id = ac.id LEFT JOIN weight_categories wc ON r.weight_category_id = wc.id WHERE r.event_type = 'kata' AND r.approved = 1 GROUP BY a.id, ac.name, wc.name ORDER BY total_points DESC
athlete_id	INTEGER	"athlete_id" INTEGER
full_name	TEXT	"full_name" TEXT
gender	TEXT	"gender" TEXT
age_category	TEXT	"age_category" TEXT
weight_category	TEXT	"weight_category" TEXT
total_points		"total_points"
ranking_kumite		CREATE VIEW ranking_kumite AS SELECT a.id AS athlete_id, a.full_name, a.gender, ac.name AS age_category, wc.name AS weight_category, SUM(r.points_earned) AS total_points FROM athletes a JOIN results r ON a.id = r.athlete_id LEFT JOIN age_categories ac ON r.age_category_id = ac.id LEFT JOIN weight_categories wc ON r.weight_category_id = wc.id WHERE r.event_type = 'kumite' AND r.approved = 1 GROUP BY a.id, ac.name, wc.name ORDER BY total_points DESC
athlete_id	INTEGER	"athlete_id" INTEGER
full_name	TEXT	"full_name" TEXT
gender	TEXT	"gender" TEXT
age_category	TEXT	"age_category" TEXT
weight_category	TEXT	"weight_category" TEXT
total_points		"total_points"
tournament_results		CREATE VIEW tournament_results AS SELECT r.id, r.tournament_id, t.name AS tournament_name, t.date, r.athlete_id, a.full_name AS athlete_name, r.age_category_id, ac.name AS age_category, r.weight_category_id, wc.name AS weight_category, r.placement, r.wins, r.participated, r.approved, r.points_earned FROM results r JOIN tournaments t ON r.tournament_id = t.id JOIN athletes a ON r.athlete_id = a.id LEFT JOIN age_categories ac ON r.age_category_id = ac.id LEFT JOIN weight_categories wc ON r.weight_category_id = wc.id
id	INTEGER	"id" INTEGER
tournament_id	INTEGER	"tournament_id" INTEGER
tournament_name	TEXT	"tournament_name" TEXT
date	DATETIME	"date" DATETIME
athlete_id	INTEGER	"athlete_id" INTEGER
athlete_name	TEXT	"athlete_name" TEXT
age_category_id	INTEGER	"age_category_id" INTEGER
age_category	TEXT	"age_category" TEXT
weight_category_id	INTEGER	"weight_category_id" INTEGER

Name	Type	Schema
weight_category	TEXT	"weight_category" TEXT
placement	INTEGER	"placement" INTEGER
wins	INTEGER	"wins" INTEGER
participated	INTEGER	"participated" INTEGER
approved	INTEGER	"approved" INTEGER
points_earned	INTEGER	"points_earned" INTEGER

Triggers (0)

Name	Type	Schema