Tables (9)

Name	Type	Schema
role		CREATE TABLE role (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(100) NOT NULL UNIQUE, permission_create INTEGER NOT NULL, permission_read_all INTEGER NOT NULL, permission_move_backward INTEGER NOT NULL, permission_move_forward INTEGER NOT NULL, permission_edit INTEGER NOT NULL, permission_change_role INTEGER NOT NULL, permission_change_assignment INTEGER NOT NULL)
id	INTEGER	"id" INTEGER
name	VARCHAR(100)	"name" VARCHAR(100) NOT NULL UNIQUE
permission_create	INTEGER	"permission_create" INTEGER NOT NULL
permission_read_all	INTEGER	"permission_read_all" INTEGER NOT NULL
permission_move_backward	INTEGER	"permission_move_backward" INTEGER NOT NULL
permission_move_forward	INTEGER	"permission_move_forward" INTEGER NOT NULL
permission_edit	INTEGER	"permission_edit" INTEGER NOT NULL
permission_change_role	INTEGER	"permission_change_role" INTEGER NOT NULL
permission_change_assignment	INTEGER	"permission_change_assignment" INTEGER NOT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
task		CREATE TABLE task (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(150) NOT NULL, description VARCHAR(1000) NOT NULL, deadline DATE NULL DEFAULT NULL CHECK (deadline IS NULL OR strftime('%Y-%m-%d', deadline) > strftime('%Y-%m-%d', 'now')), priority INTEGER NOT NULL DEFAULT 0, created_at DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now')), updated_at DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now')), author_id INTEGER NOT NULL, task_status_id INTEGER NOT NULL, Foreign Key(author_id) References user(id), Foreign Key(task_status_id) References task_status(id))
id	INTEGER	"id" INTEGER
name	VARCHAR(150)	"name" VARCHAR(150) NOT NULL
description	VARCHAR(1000)	"description" VARCHAR(1000) NOT NULL
deadline	DATE	"deadline" DATE DEFAULT NULL CHECK("deadline" IS NULL OR strftime('%Y-%m-%d', "deadline") > strftime('%Y-%m-%d', 'now'))
priority	INTEGER	"priority" INTEGER NOT NULL DEFAULT 0
created_at	DATETIME	"created_at" DATETIME DEFAULT (strftime('%Y-

Name	Type	Schema
		%m-%d %H:%M:%S', 'now'))
updated_at	DATETIME	"updated_at" DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now'))
author_id	INTEGER	"author_id" INTEGER NOT NULL
task_status_id	INTEGER	"task_status_id" INTEGER NOT NULL
task_assignment		CREATE TABLE task_assignment (id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER NOT NULL, task_id INTEGER NOT NULL, Foreign Key(user_id) References user(id) Foreign Key(task_id) References task(id))
id	INTEGER	"id" INTEGER
user_id	INTEGER	"user_id" INTEGER NOT NULL
task_id	INTEGER	"task_id" INTEGER NOT NULL
task_label		CREATE TABLE task_label (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(500) NOT NULL, description VARCHAR(1000) NULL, rgb_color TEXT(6) NOT NULL)
id	INTEGER	"id" INTEGER
name	VARCHAR(500)	"name" VARCHAR(500) NOT NULL
description	VARCHAR(1000)	"description" VARCHAR(1000)
rgb_color	TEXT(6)	"rgb_color" TEXT(6) NOT NULL
task_status		CREATE TABLE task_status (id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(150) NOT NULL UNIQUE, description VARCHAR(1000) NULL, default_next_task_status_id INTEGER NULL, Foreign Key (default_next_task_status_id) References task_status(id))
id	INTEGER	"id" INTEGER
name	VARCHAR(150)	"name" VARCHAR(150) NOT NULL UNIQUE
description	VARCHAR(1000)	"description" VARCHAR(1000)
default_next_task_status_id	INTEGER	"default_next_task_status_id" INTEGER
task_task_label_pivot		CREATE TABLE task_task_label_pivot (id INTEGER PRIMARY KEY AUTOINCREMENT, task_id INTEGER NOT NULL, task_label_id INTEGER NOT NULL, Foreign Key(task_id) References task(id) Foreign Key(task_label_id) References task_label(id))
id	INTEGER	"id" INTEGER
task_id	INTEGER	"task_id" INTEGER NOT NULL
task_label_id	INTEGER	"task_label_id" INTEGER NOT NULL
todo_item		CREATE TABLE todo_item (id INTEGER PRIMARY KEY AUTOINCREMENT, description VARCHAR(1000) NOT NULL, deadline DATE NULL DEFAULT NULL CHECK (deadline IS NULL OR strftime('%Y-%m-%d', deadline)

Name	Type	Schema
		> strftime('%Y-%m-%d', 'now')), priority INTEGER NOT NULL DEFAULT 0, created_at DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now')), updated_at DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now')), done INTEGER NOT NULL DEFAULT 0, author_id INTEGER NOT NULL, task_id INTEGER NOT NULL, Foreign Key(author_id) References user(id), Foreign Key(task_id) References task(id))
id	INTEGER	"id" INTEGER
description	VARCHAR(1000)	"description" VARCHAR(1000) NOT NULL
deadline	DATE	"deadline" DATE DEFAULT NULL CHECK("deadline" IS NULL OR strftime('%Y-%m-%d', "deadline") > strftime('%Y-%m-%d', 'now'))
priority	INTEGER	"priority" INTEGER NOT NULL DEFAULT 0
created_at	DATETIME	"created_at" DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now'))
updated_at	DATETIME	"updated_at" DATETIME DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now'))
done	INTEGER	"done" INTEGER NOT NULL DEFAULT 0
author_id	INTEGER	"author_id" INTEGER NOT NULL
task_id	INTEGER	"task_id" INTEGER NOT NULL
user		CREATE TABLE user (id INTEGER PRIMARY KEY AUTOINCREMENT, username VARCHAR(100) NOT NULL UNIQUE, email VARCHAR(256) NOT NULL UNIQUE, password VARCHAR(256), role_id INTEGER NOT NULL, Foreign Key (role_id) References role(id))
id	INTEGER	"id" INTEGER
username	VARCHAR(100)	"username" VARCHAR(100) NOT NULL UNIQUE
email	VARCHAR(256)	"email" VARCHAR(256) NOT NULL UNIQUE
password	VARCHAR(256)	"password" VARCHAR(256)
role_id	INTEGER	"role_id" INTEGER NOT NULL

Indices (0)

Name	Type	Schema

Views (0)

Name	Type	Schema

Triggers (0)

Name	Type	Schema