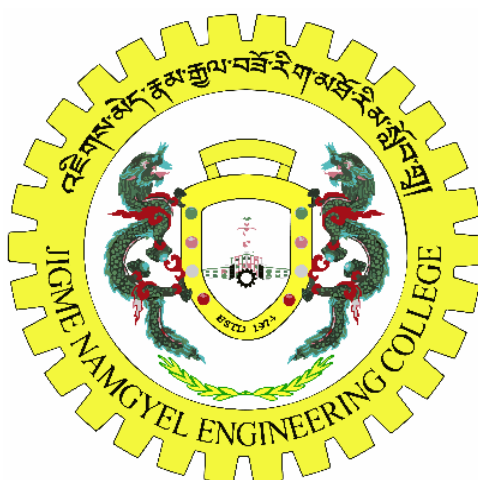




འཇམ་མགོན་མཆོད་སྐྱོན་རྒྱལ་ཁབ་རྒྱུ་རྒྱུ་རྒྱུ་
Royal University of Bhutan
Jigme Namgyal Engineering Collage, Dewathang



Database Setup Report – To Do List Tracker



Submitted by:

Norbu Wangmo (05240141)

DIPLOMA IN COMPUTER SYSTEM AND NETWORK

JIGME NAMGYEL ENGINEERING COLLEGE ROYAL UNIVERSITY OF BHUTAN

DEWATHANG

(20th August, 2025)

Contents

Introduction.....	3
Features:	4
Database Structure	4
Users Table.....	4
Data Type.....	5
Constraints	5
Tasks Table	7
Data Types	7
Constraints	8
Reference	9

Introduction

The To-Do List Tracker is a comprehensive personal task management system designed to help users organize, track, and manage their daily tasks and activities efficiently. This intuitive application allows users to create tasks, categorize them by status, set priorities, and monitor progress through a structured workflow. This system modernizes productivity by replacing manual lists with digital task management, reducing mental load through organized categorization. It enhances efficiency with visual progress tracking, deadline management, and reliable data persistence for all tasks.

Features:

- Allows users to create and manage tasks digitally instead of manually.
- Supports categorization of tasks by their status.
- Enables users to set task priorities (e.g., low, medium, high).
- Enhances efficiency with visual progress tracking.
- Includes deadline management for tasks.
- Ensures reliable data persistence for all tasks.

Database Structure

Users Table

The users table is created with the CREATE TABLE command. It defines seven columns: a unique auto-incrementing ID as the primary key, a mandatory and unique username with a minimum length, a mandatory and unique email with format validation, a mandatory password hash, automatic timestamps for creation and last login, and a status flag to indicate an active account, ensuring secure and organized user data management.

```
# Create users table
cursor.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL UNIQUE,
    email TEXT NOT NULL UNIQUE,
    password_hash TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_login TIMESTAMP,
    is_active INTEGER DEFAULT 1
);
""")
```

Sample:

	id	username	email	password_hash	created_at	last_login	is_active
1	1	Norbu	norbu@example.com	hashed_password_he...	2025-08-20 06:07:12	NULL	1
2	2	Wangmo	wangmo@gmail.com	hashed_password_he...	2025-08-20 07:10:12	NULL	1

Data Type

Column	Data Type	Purpose
id	INTEGER	Unique identifier
username	TEXT	User's unique name
email	TEXT	User's email address
Password_hash	TEXT	Encrypted password
created_at	TEXT	Account creation timestamp (YYYY-MM-DD format)
Last_login	TEXT	Last login timestamp (YYYY-MM-DD format)
Is_active	INTEGER	Account status (0/1)

Constraints

Primary Key (PK):

id: PRIMARY KEY AUTOINCREMENT

Uniquely identifies each user

Automatically increments (1, 2, 3, ...)

Prevents duplicate IDs

Not Null (NN):

username, email, password_hash: NOT NULL

These fields are mandatory

Cannot be empty or NULL

Unique (U):

username, email: UNIQUE

No duplicate usernames allowed

No duplicate email addresses allowed

Ensures each user is unique

Check Constraints:

username: CHECK(length(username) >= 3)

Username must be at least 3 characters long

Prevents short usernames like "ab"

email: CHECK(email LIKE '%@%.%')

Validates email format contains "@" and "."

is_active: CHECK(is_active IN (0, 1))

Only allows values 0 or 1

1 = Active account, 0 = Inactive account

Default Values:

created_at: DEFAULT CURRENT_TIMESTAMP

Automatically sets to current date/time

Format: "YYYY-MM-DD HH:MM:SS"

is_active: DEFAULT 1

New accounts are active by default

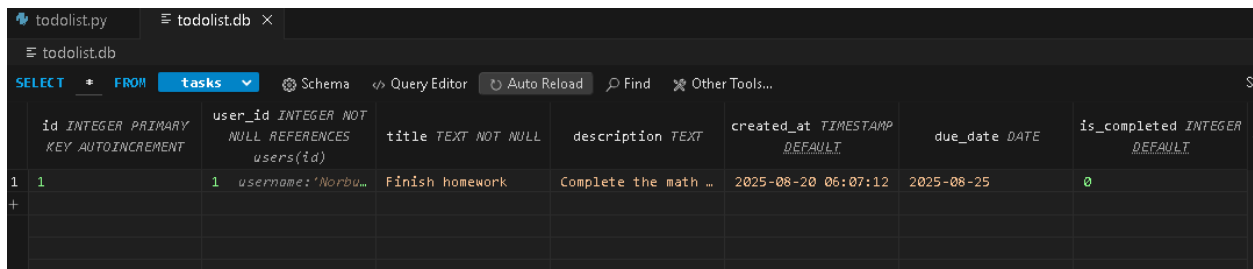
Sets to 1 if no value provided.

Tasks Table

The tasks table is created with the CREATE TABLE command. It defines seven columns: a unique auto-incrementing ID as the primary key, a mandatory title, an optional description, a status with predefined options, an automatic creation date, an optional due date, and a priority level between 1 and 3, ensuring organized and consistent tracking of task information.

```
# Create tasks table
cursor.execute("""
CREATE TABLE IF NOT EXISTS tasks (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL,
    title TEXT NOT NULL,
    description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    due_date DATE,
    is_completed INTEGER DEFAULT 0,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
""")
```

Sample:



	id INTEGER PRIMARY KEY AUTOINCREMENT	user_id INTEGER NOT NULL REFERENCES users(id)	title TEXT NOT NULL	description TEXT	created_at TIMESTAMP DEFAULT	due_date DATE	is_completed INTEGER DEFAULT
1	1	1 username: 'Norbu...	Finish homework	Complete the math ...	2025-08-20 06:07:12	2025-08-25	0
+							

Data Types

Column	Data Type	Purpose
id	INTEGER	Unique identifier
title	TEXT	Task name/description
description	TEXT	Detailed task notes
status	TEXT	Progress state
created_date	TEXT	Date created (YYYY-MM-DD format)
due_date	TEXT	Deadline date (YYYY-MM-DD format)
priority	INTEGER	Importance level (1, 2, or 3)

Constraints

Primary Key Constraints:

id: PRIMARY KEY AUTOINCREMENT

Ensures each task has unique ID

Automatically increments (1, 2, 3, ...)

Prevents duplicate IDs

Not Null Constraints (under NN):

title: NOT NULL - Task name is mandatory

status: NOT NULL - Status must always be specified

priority: NOT NULL - Priority level must be set

Check Constraints:

status: CHECK("status" IN ('Pending', 'To Do', 'Done'))

Only allows these 3 values

Prevents invalid statuses like "Started" or "Almost Done"

priority: CHECK("priority" BETWEEN 1 AND 3)

Only allows values 1, 2, or 3

1 = Low, 2 = Medium, 3 = High priority

Default Values:

created_date: DEFAULT CURRENT_DATE

Automatically sets to today's date

Format: YYYY-MM-DD (SQLite standard)

Reference

BoostMyTool. (2021, October 13). *Create and Read SQLite Databases using DB Browser: How to Use DB Browser 2021* [Video]. YouTube.

<https://www.youtube.com/watch?v=b0Dplx4M5zg>

Downloads - DB browser for SQLite. (n.d.). <https://sqlitebrowser.org/dl/>

TutorialBrain. (2022, April 20). *How to run SQLITE in Visual Studio Code* [Video]. YouTube.

<https://www.youtube.com/watch?v=JrAiefGNUq8>