



хочу написать приложение в котором будет такой функционал

я завожу задачу:

у задачи могут быть типы если тип = 1 то я указываю в дату окончания задачи
если тип = 2 то я указываю кол-во каких то повторений и в задаче я отмечаю сколько из необходимых повторений я сделал

ты можешь дополнить типы если у тебя есть идеи и опиши диаграмму для такой бд



Давайте начнем с базы данных для вашего приложения. Вот основные типы задач, которые вы описали:

- Срок выполнения (тип 1):** У задачи есть конечная дата, к которой она должна быть завершена.
- Повторения (тип 2):** У задачи есть требуемое количество повторений, и вы отмечаете прогресс (например, сделал 3 из 5).

Дополнительные типы задач:

- Ежедневная задача (тип 3):** Задача должна выполняться ежедневно (например, привычка — "читать 10 страниц каждый день").
- Целевая сумма (тип 4):** Нужно достичь определенной цели в числовом выражении (например, пробежать 100 км за месяц).
- Ожидающее событие (тип 5):** Задача завершится при наступлении определенного события (например, "дождаться ответа от клиента").

Структура базы данных

Для хранения данных предлагаем следующие таблицы:

1. Таблица `tasks` (Основная таблица задач)

Поле	Тип данных	Описание
<code>id</code>	INTEGER (PK)	Уникальный идентификатор задачи
<code>title</code>	TEXT	Название задачи
<code>description</code>	TEXT	Описание задачи
<code>type</code>	INTEGER	Тип задачи (1 — срок, 2 — повторения и т.д.)
<code>status</code>	TEXT	Статус задачи (active, completed, cancelled)
<code>created_at</code>	DATETIME	Дата создания задачи
<code>updated_at</code>	DATETIME	Дата последнего изменения

2. Таблица `task_deadlines` (Для задач с датой окончания)

Поле	Тип данных	Описание
<code>task_id</code>	INTEGER (FK)	ID задачи из таблицы <code>tasks</code>
<code>deadline</code>	DATETIME	Дата окончания задачи

3. Таблица `task_repetitions` (Для задач с повторениями)

Поле	Тип данных	Описание
<code>task_id</code>	INTEGER (FK)	ID задачи из таблицы <code>tasks</code>
<code>target_count</code>	INTEGER	Необходимое количество повторений
<code>completed_count</code>	INTEGER	Выполненное количество повторений

4. Таблица `task_daily` (Для ежедневных задач)

Поле	Тип данных	Описание
<code>task_id</code>	INTEGER (FK)	ID задачи из таблицы <code>tasks</code>
<code>last_completed</code>	DATETIME	Последнее выполнение

5. Таблица `task_targets` (Для целевых значений)

Поле	Тип данных	Описание
<code>task_id</code>	INTEGER (FK)	ID задачи из таблицы <code>tasks</code>
<code>target_value</code>	REAL	Целевая величина (например, 100 км)
<code>current_value</code>	REAL	Текущая достигнутая величина

Связи между таблицами:

- Таблица `tasks` связана с остальными таблицами через `task_id`, что позволяет хранить дополнительные данные, специфичные для типа задачи.

Диаграмма базы данных (ERD)



Эта структура гибкая и позволит легко добавлять новые типы задач. Могу также предложить код для создания этих таблиц в SQL.

лучше напиши код для визуализации этой бд в приложении mermaid



Вот код для визуализации структуры базы данных с помощью Mermaid:

mermaid

```
erDiagram
    tasks {
        int id PK
        string title
        string description
        int type
        string status
        datetime created_at
        datetime updated_at
    }
    task_deadlines {
        int task_id FK
        datetime deadline
    }
    task_repetitions {
        int task_id FK
        int target_count
        int completed_count
    }
    task_daily {
        int task_id FK
        datetime last_completed
    }
    task_targets {
        int task_id FK
        float target_value
        float current_value
    }

    tasks ||--o{ task_deadlines : "has"
    tasks ||--o{ task_repetitions : "has"
    tasks ||--o{ task_daily : "has"
    tasks ||--o{ task_targets : "has"
```

Как это использовать:

- Перейдите на [Mermaid Live Editor](#).
- Вставьте этот код в окно редактора.
- Нажмите "Generate Diagram", чтобы увидеть визуализацию.

Если что-то нужно уточнить или доработать, пишите!