

Файл file_manager.db

Полная схема таблиц базы данных file_manager.db (SQLite)

Локальная база данных модуля содержит 3 взаимосвязанные таблицы, которые полностью обеспечивают логику безопасности, распределения прав, учета дубликатов и ведения тегов.

Таблица 1: files_catalog (Главный реестр файлов)

Эта таблица хранит физические параметры каждого файла и жестко привязывает его к уровням приватности и пространству пользователя.

Имя поля (Column)	Тип данных	Флаги (Constraints)	Описание поля и его логическая роль в коде
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Уникальный сквозной номер файла внутри модуля.
user_id	INTEGER	NOT NULL	Владелец файла. Сюда записывается глобальный ID пользователя из таблицы core_users ядра.
orig_name	TEXT	NOT NULL	Оригинальное имя, которое видел юзер (например, чертеж_мурманск.dwg). Поддерживает кириллицу.
server_name	TEXT	NOT NULL	Безопасное имя файла на флешке (например, file_8834.dat). Исключает запуск хакерских скриптов.
file_type	TEXT	NOT NULL	Группа файла для быстрой фильтрации в интерфейсе ('photo', 'video', 'audio', 'document').
file_size	INTEGER	NOT NULL	Размер файла в байтах. Используется для вывода в таблице и контроля лимитов места.
file_hash	TEXT	NOT NULL	MD5-хэш содержимого. Нужен для защиты от дубликатов (дедупликации) и экономии памяти флешки.
virtual_folder	TEXT	DEFAULT '/'	Виртуальный путь в интерфейсе (например, '/мурманск/проекты/'). Избавляет от создания реальных папок на диске.
access_level	TEXT	DEFAULT 'private'	Контур приватности. Принимает строго 3 значения: 'private' (только владелец), 'group' (семья/отдел), 'public' (все).
download_count	INTEGER	DEFAULT 0	Счетчик скачиваний. Показывает Суперадмину популярность файла.

is_deleted	INTEGER	DEFAULT 0	Флаг Корзины. 0 — файл активен, 1 — файл удален пользователем в Корзину (скрыт из общего списка).
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Точная дата и время загрузки файла на сервер.

□ Таблица 2: files_tags (Индексатор тегов для быстрого поиска)

Позволяет привязывать к одному файлу бесконечное количество поисковых меток (например, к одной фотографии можно добавить теги: «2026», «Отпуск», «Мурманск»). Связана с главной таблицей отношением «один ко многим».

Имя поля (Column)	Тип данных	Флаги (Constraints)	Описание поля и его логическая роль в коде
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Уникальный номер тега.
file_id	INTEGER	NOT NULL	Связующий ключ. Указывает, к какому конкретно файлу из files_catalog привязан этот тег.
tag_name	TEXT	NOT NULL	Сам текст тега в нижнем регистре для удобства поиска (например, отпуск).

- **Правило каскадного удаления:** Для поля file_id прописывается связь FOREIGN KEY(file_id) REFERENCES files_catalog(id) ON DELETE CASCADE. Это значит, что если Игорь окончательно очистит файл из Корзины, ядро SQLite **автоматически и молча сотрет все теги**, привязанные к этому файлу, предотвращая появление мусора в базе.

□ Таблица 3: files_shares (Точечный доступ для конкретных людей)

Нужна для расширения логики приватности. Если уровень доступа файла равен private, но Игорь хочет показать этот конкретный файл только Алисе (не открывая его для всей группы или интернета), эта таблица зафиксирует исключение.

Имя поля (Column)	Тип данных	Флаги (Constraints)	Описание поля и его логическая роль в коде
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Уникальный номер записи.
file_id	INTEGER	NOT NULL	ID файла из таблицы files_catalog, к которому дается доступ.
shared_with_id	INTEGER	NOT NULL	ID пользователя из глобальной базы ядра (core_users), которому разрешено скачать этот приватный файл.

□ Готовый SQL-код для инсталлятора модуля (Для вашего руководства)

Когда Суперадмин будет впервые активировать модуль файлового менеджера, его controller.php выполнит этот чистый SQL-запрос для создания структуры:

SQL-- 1. Создаем таблицу каталога файлов

```
CREATE TABLE IF NOT EXISTS files_catalog (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER NOT NULL,  
    orig_name TEXT NOT NULL,  
    server_name TEXT NOT NULL,  
    file_type TEXT NOT NULL,  
    file_size INTEGER NOT NULL,  
    file_hash TEXT NOT NULL,  
    virtual_folder TEXT DEFAULT '/',  
    access_level TEXT DEFAULT 'private',  
    download_count INTEGER DEFAULT 0,  
    is_deleted INTEGER DEFAULT 0,  
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

-- 2. Создаем таблицу тегов с автоматической очисткой

```
CREATE TABLE IF NOT EXISTS files_tags (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    file_id INTEGER NOT NULL,  
    tag_name TEXT NOT NULL,  
    FOREIGN KEY(file_id) REFERENCES files_catalog(id) ON DELETE CASCADE  
);
```

-- 3. Создаем таблицу точечных доступов

```
CREATE TABLE IF NOT EXISTS files_shares (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    file_id INTEGER NOT NULL,  
    shared_with_id INTEGER NOT NULL,  
    FOREIGN KEY(file_id) REFERENCES files_catalog(id) ON DELETE CASCADE  
);
```

-- 4. Создаем системные индексы для ультрабыстрого поиска (чтобы флешка не тормозила)

```
CREATE INDEX IF NOT EXISTS idx_files_user ON files_catalog(user_id);  
CREATE INDEX IF NOT EXISTS idx_files_hash ON files_catalog(file_hash);  
CREATE INDEX IF NOT EXISTS idx_tags_file ON files_tags(file_id);
```

From:

<https://wwoss.ru/> - **worldwide open-source software**

Permanent link:

https://wwoss.ru/doku.php?id=manual:file_manager_db:test&rev=1783882453



Last update: **2026/07/12 21:54**