-- 1. Создание новой таблицы с данными из старой

CREATE TABLE neworders\_order AS SELECT \* FROM orders\_order;

-- 2. Добавление нового столбца unique\_key в новую таблицу

ALTER TABLE neworders\_order ADD COLUMN unique\_key UUIDField;

-- 3. Копирование данных из старой таблицы в новую с генерацией уникальных значений для unique\_key

-- Мы используем RANDOMBLOB для генерации уникальных значений для столбца unique\_key.

-- Обратите внимание, что мы исключаем новый столбец из списка столбцов при вставке данных.

INSERT INTO neworders\_order (id, created\_at, user\_id, address, comment, phone, recipient\_name, updated\_at, status, unique\_key)

SELECT id, created\_at, user\_id, address, comment, phone, recipient\_name, updated\_at, status, LOWER(HEX(RANDOMBLOB(8)))

FROM orders\_order;

-- 4. Замена значений NULL в столбце unique\_key на уникальные значения

UPDATE neworders\_order

SET unique\_key = LOWER(HEX(RANDOMBLOB(8)))

WHERE unique\_key IS NULL;

-- 5. Удаление таблицы

DROP TABLE IF EXISTS orders\_order;

-- 6. Измненение имени таблицы

ALTER TABLE neworders\_order RENAME TO orders\_order;

-- 1. Создание новой таблицы с данными из старой

CREATE TABLE neworders\_order AS

SELECT \* FROM orders\_order;

-- 2. Добавление нового столбца unique\_key в новую таблицу

ALTER TABLE neworders\_order ADD COLUMN unique\_key UUIDField;

-- 3. Копирование данных из старой таблицы в новую с генерацией уникальных значений для unique\_key

-- Мы используем RANDOMBLOB для генерации уникальных значений для столбца unique\_key.

-- Обратите внимание, что мы исключаем новый столбец из списка столбцов при вставке данных.

INSERT INTO neworders\_order (id, created\_at, user\_id, address, comment, phone, recipient\_name, updated\_at, status, unique\_key)

SELECT id, created\_at, user\_id, address, comment, phone, recipient\_name, updated\_at, status, LOWER(HEX(RANDOMBLOB(8)))

FROM orders\_order;

-- 4. Замена значений NULL в столбце unique\_key на уникальные значения

UPDATE neworders\_order

SET unique\_key = LOWER(HEX(RANDOMBLOB(8)))

WHERE unique\_key IS NULL;

-- \*\*Важно\*\*:

-- После выполнения SQL-запросов в DB Browser для SQLite нужно:

-- 1. Нажать на кнопку \*\*"Write Changes"\*\* в верхней панели, чтобы сохранить изменения в базе данных.

-- Это применяет изменения в текущей базе данных.

-- 2. Если вы работаете с транзакциями (например, `BEGIN TRANSACTION`), не забудьте выполнить команду `COMMIT`, чтобы сохранить все изменения.

-- Например:

-- COMMIT;

-- Вы можете использовать транзакции для объединения изменений, чтобы они были выполнены как единый блок.

-- Пример:

-- BEGIN TRANSACTION;

-- -- Все изменения (включая вышеуказанные) внутри транзакции

-- COMMIT;