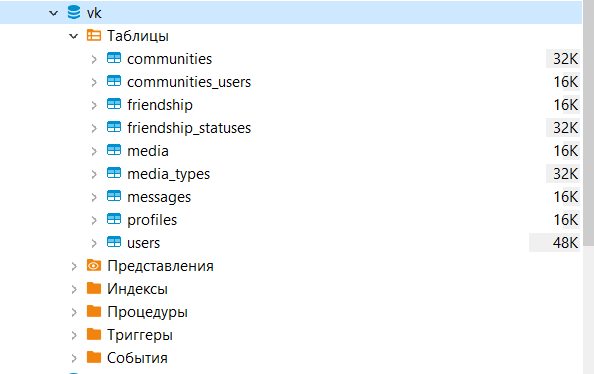
Практическое задание по теме “Введение в проектирование БД”

1. Создать структуру БД Вконтакте по скриптам, приложеным в файле примеров examples.sql

Создали таблицы через выполнение скриптов в окне sql-редактора DBeaver



1. Используя сервис [http://filldb.info](http://filldb.info/) или другой по вашему желанию, сгенерировать тестовые данные для всех таблиц, учитывая логику связей. Для всех таблиц, где это имеет смысл, создать не менее 100 строк. Загрузить тестовые данные в свою локальную базу данных ВК. Приложить к отчёту полученный дамп с данными.

Далее красным выделены настройки не по умолчанию, предлагаемые сервисом filldb:

Таблица users:

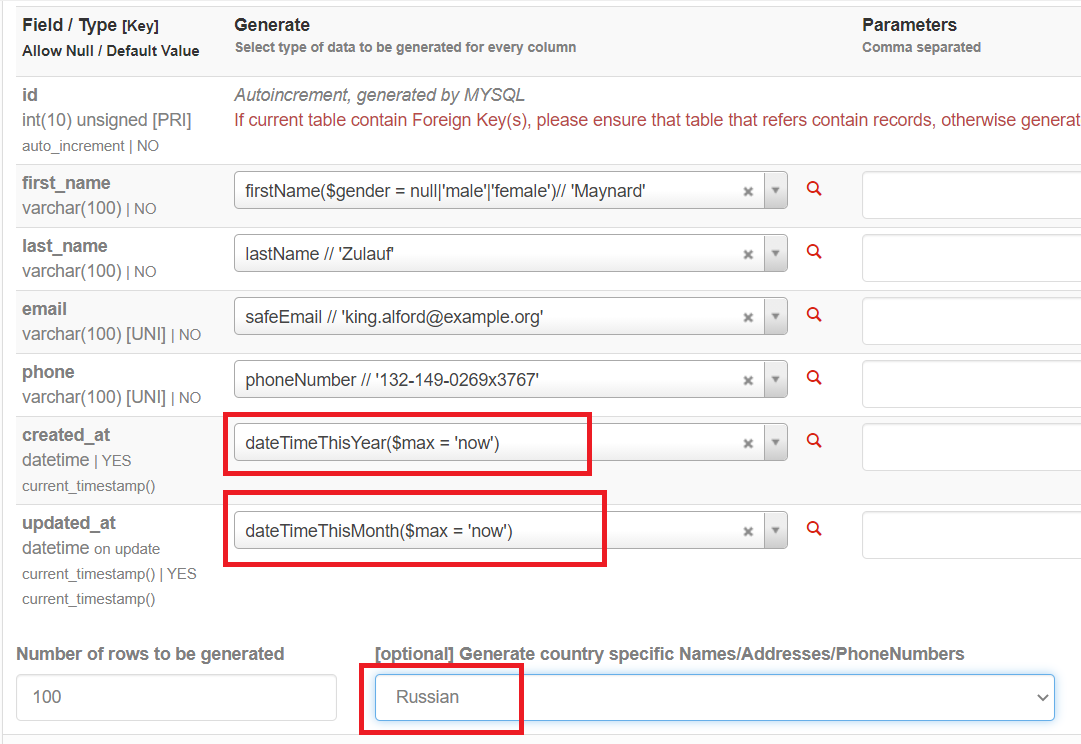


Таблица profiles

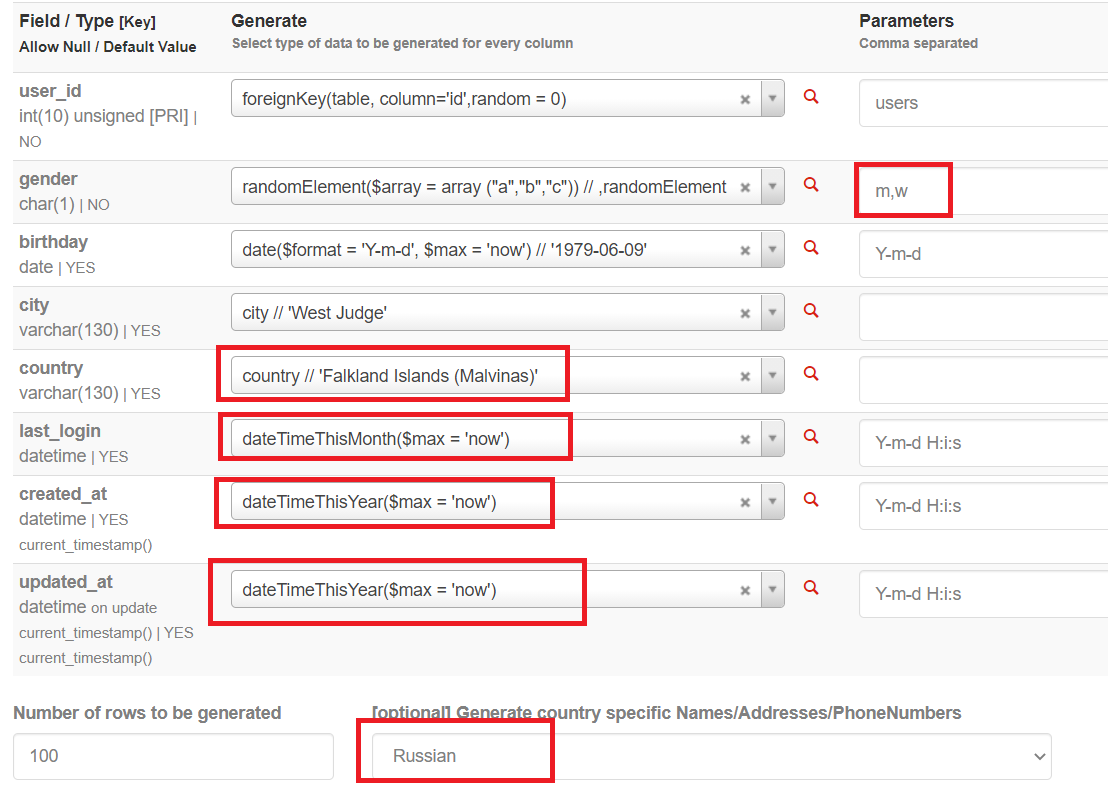


Таблица messages

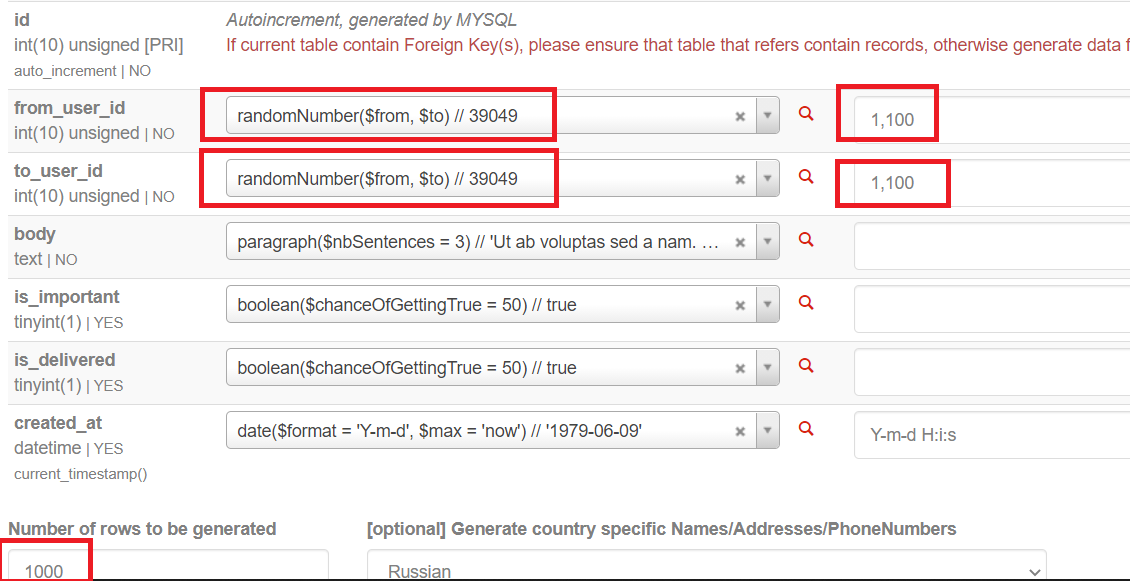


Таблица friendship\_statuses

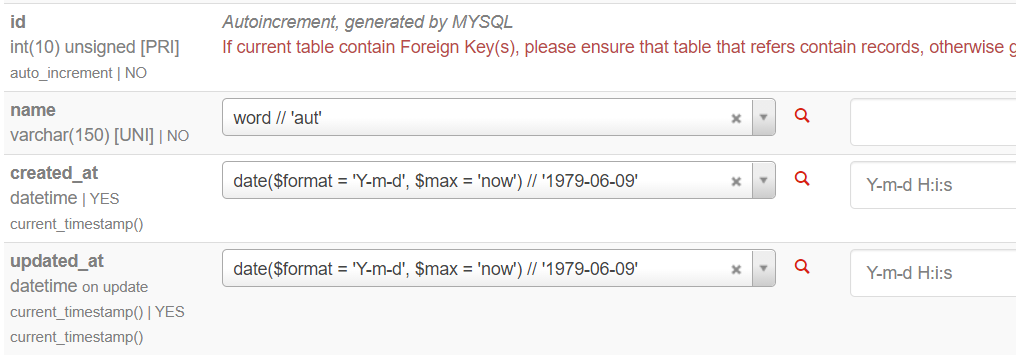


Таблица friendship

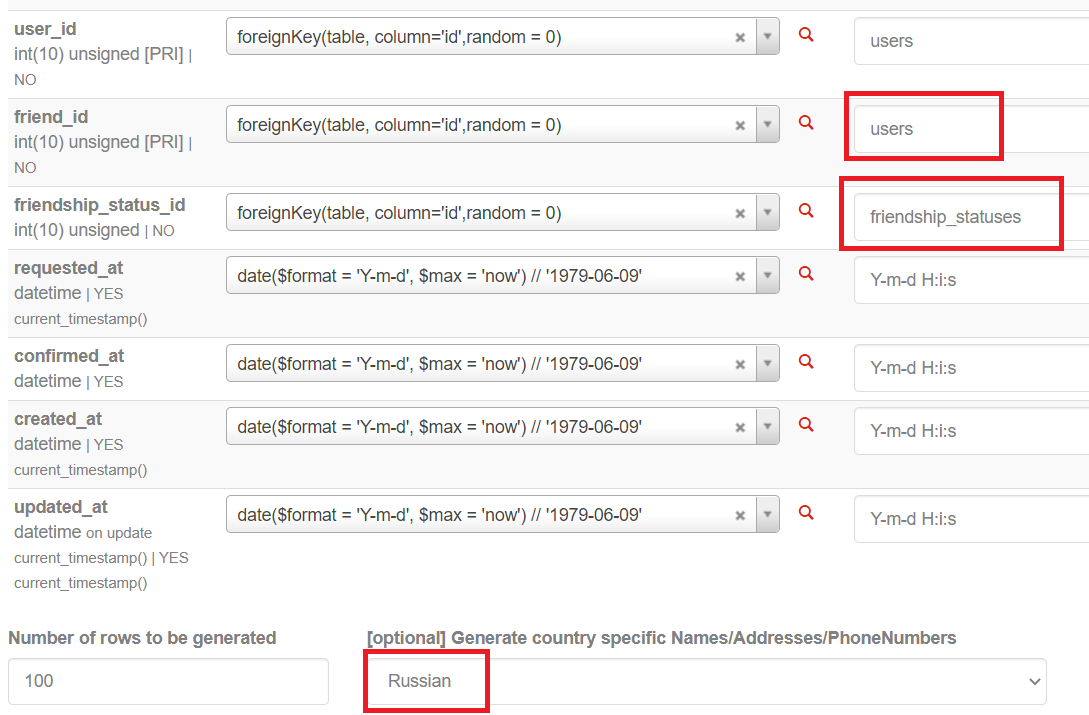
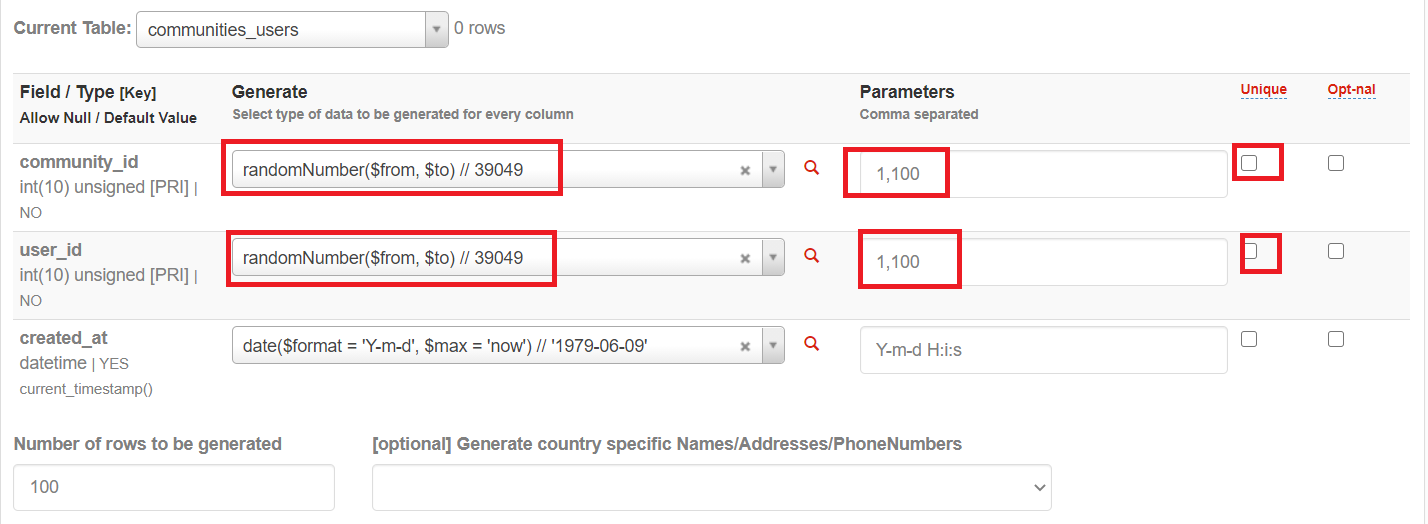


Таблица community\_users



**Таблицу media с помощью filldb не удалось выгрузить** при разных наборах правил формирования значений полей. Утилита написала SUCCESS, но фактически данных не сгенерировала.

Данные для таблицы media **сгенерировал вручную** с помощью SQL в postgresql:

**select**

'insert into media(id, user\_id, filename, size, metadata, media\_type\_id, created\_at, updated\_at)

values('|| **concat**(

t.val,', ',**floor**(**random**()\*(100-1+1))+1,', ''', 'c:\files\filname', t.val, ''', ',

**floor**(**random**()\*(100000000-1000+1))+1000, ', ''', **jsonb\_build\_object**('type', 'avi'), ''', ',

**floor**(**random**()\*(10-1+1))+1, ', ''',

**localtimestamp** - **make\_interval**(days=>t.val), ''', ''',

**localtimestamp** - **make\_interval**(days=>t.val), ''''

)

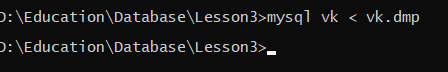
||');'

**from**

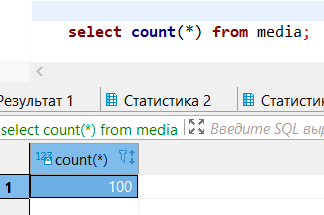
**generate\_series**(1,100,1) t(val)

;

Полученный дамп из fiildb объединили с данными для таблицы media и сделали файл vk.dmp для загрузки в целевую БД.



Выборочно проверяем, что файл загрузился.



1. (по желанию) Проанализировать структуру БД vk, которую мы создали на занятии, и внести предложения по усовершенствованию (если такие идеи есть). Напишите пожалуйста, всё-ли понятно по структуре.
   1. На всех автогенерируемых полях следует поставить ограничение not null, где его нет, например

**alter** **table** users **modify** **column** created\_at **datetime** **not** **null**;

**alter** **table** users **modify** **column** updated\_at **datetime** **not** **null**;

* 1. Для обеспечения правил ссылочной целостности следует создать внешние ключи там, где это предусмотрено логикой

**alter** **table** profiles **add** **constraint** fk\_profile\_users **foreign** **key** (user\_id) **references** users(id);

* 1. Поля таблицы users first\_name, last\_name, email, phone по смыслу больше подходят к таблице profiles, а поле таблицы profiles last\_login, наоборот, лучше перенести в таблицу users.

В таблице users не хватает поля username

**alter** **table** users **add** username **varchar**(30) **not** **null** comment "Имя пользователя для входа в сервис";

* 1. В таблицах messages и media требуется поменять тип для поля ID, т.к. существующий тип int не позволит хранить большие числа.

**alter** **table** messages **modify** **column** id **bigint**;

**alter** **table** media **modify** **column** id **bigint**;

* 1. Для полей created\_at, updated\_at в больших таблицах, наверное, лучше использовать тип TIMESTAMP, чем DATETIME для уменьшения объема таблиц.

**Вопрос: прошу прояснить как правильно работать в mysql с временными зонами. Какой тип использовать?**