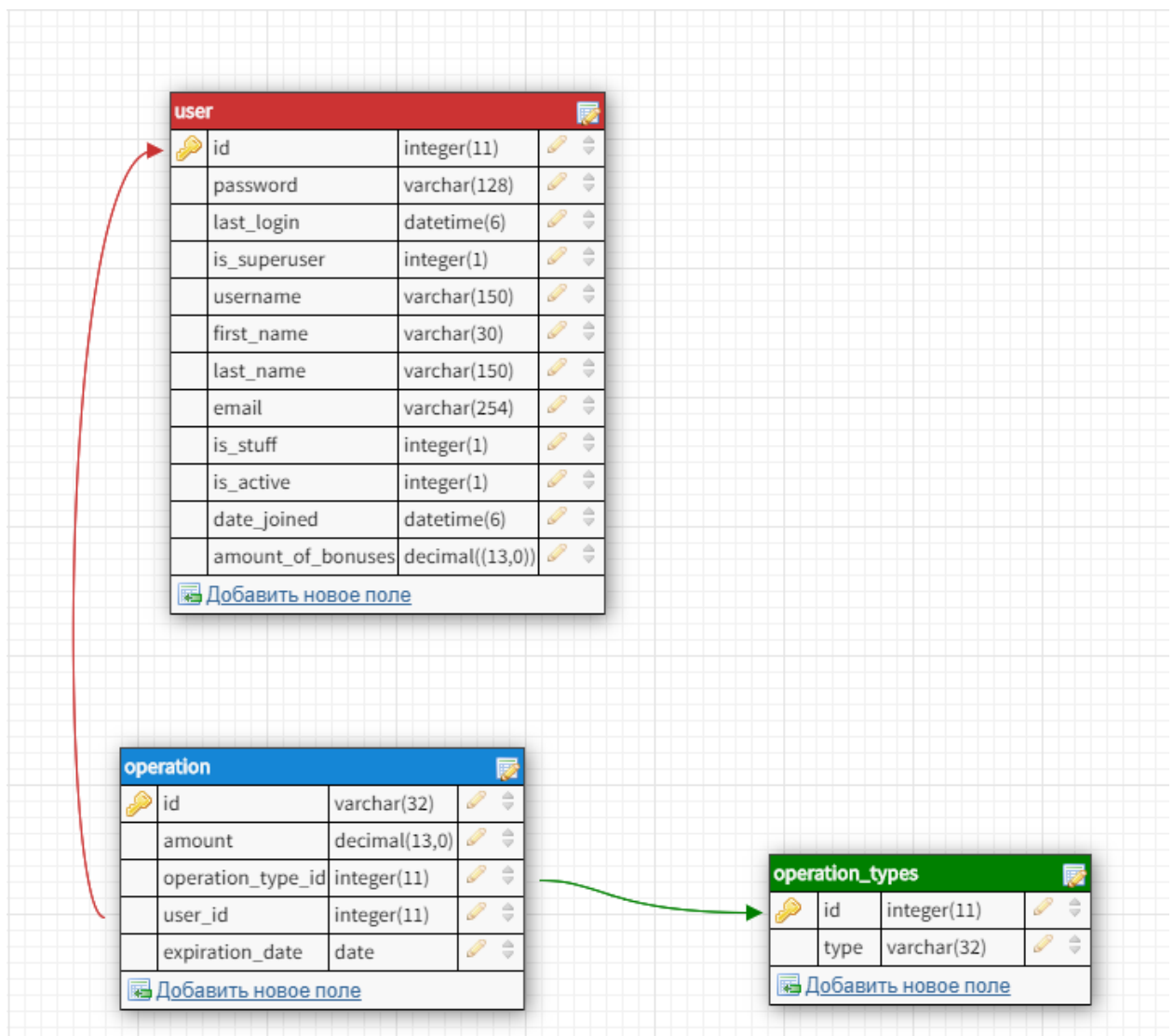


Схема БД и ее краткое описание



На схеме выше размещены только таблицы, которые являются необходимыми для реализации поставленной задачи. В реальной базе данных несколько больше таблиц, которые необходимы для работы Django. Полную структуру реальной базы данных можно увидеть в дампе, размещенном в папке с `bonus_system\sql`.

Все операции хранятся в отдельной таблице `operation`, тем самым появляется возможность откатывать любую операцию.

Для автоматического обновления поля `amount_of_bonuses` таблицы `user` были реализованы триггеры на добавление и удаление записей в таблицу `operations` (их код будет представлен ниже). Изменять записи в таблице `operation` нельзя.

Код триггеров:

Триггер на добавление операции:

```
DELIMITER;;
create trigger `insert_user_bonus_trigger` before insert on
`bonus_system_db`.`bonuses_operation`
  for each row
  begin
    if NEW.operation_type_id = 2 and NEW.expiration_date is not null
then
      set NEW.expiration_date = null;
    end if;
    if NEW.operation_type_id = 2 then
      update bonus_system_db.bonuses_user set amount_of_bonuses =
amount_of_bonuses - NEW.amount where bonus_system_db.bonuses_user.id =
NEW.user_id;
    else
      update bonus_system_db.bonuses_user set amount_of_bonuses =
amount_of_bonuses + NEW.amount where bonus_system_db.bonuses_user.id =
NEW.user_id;
    end if;
  end;;
DELIMITER;
```

Триггер на удаление операции:

```
DELIMITER;;
create trigger `delete_user_bonuses_trigger` before delete on
`bonus_system_db`.`bonuses_operation`
  for each row
  begin
    if OLD.operation_type_id = 2 then
      update bonus_system_db.bonuses_user set amount_of_bonuses =
amount_of_bonuses + OLD.amount where bonus_system_db.bonuses_user.id =
OLD.user_id;
    else
      update bonus_system_db.bonuses_user set amount_of_bonuses =
amount_of_bonuses - OLD.amount where bonus_system_db.bonuses_user.id =
OLD.user_id;
    end if;
  end;;
DELIMITER;
```

Описание эндпоинтов

Адрес	“Domain.com”/api/v1/users/detail/<int:pk>/
Метод	GET
Право доступа	Владелец записи или администратор
Описание	Возвращает данные о пользователе с id, указанным как параметр в url.
Тестовый запрос	api/v1/users/detail/2/
Пример работы	<pre> 1 { 2 "id": 2, 3 "username": "test", 4 "amount_of_bonuses": "60" 5 }</pre>

Адрес	“Domain.com”/api/v1/users/all/
Метод	GET
Право доступа	Администратор
Описание	Возвращает данные о всех пользователях
Тестовый запрос	api/v1/users/all/
Пример работы	<pre> 1 [2 { 3 "id": 1, 4 "username": "admin", 5 "amount_of_bonuses": "10" 6 }, 7 { 8 "id": 2, 9 "username": "test", 10 "amount_of_bonuses": "60" 11 } 12]</pre>

Адрес	“Domain.com”/api/v1/bonuses/increase/
Метод	POST
Право доступа	Администратор
Описание	Увеличивает количество бонусов пользователя посредством создания записи о новой операции в БД.
Тестовый запрос	api/v1/bonuses/increase/ Body: <pre> { "user_id": 1, "amount": 300 "expiration_date": "2019-12-30" //Не обязательно }</pre>

Пример работы	<pre>{ "id": "91eb2cb0-aa58-437e-949e-2c027e996a7c", "user": { "id": 1, "username": "admin", "amount_of_bonuses": "610" }, "amount": "300", "operation_type": { "id": 1, "type": "Debit" }, "expiration_date": "2019-12-30" }</pre>
---------------	---

Адрес	“Domain.com”/api/v1/bonuses/decrease/
Метод	POST
Право доступа	Администратор или владелец аккаунта
Описание	Уменьшает количество бонусов пользователя посредством создания записи о новой операции в БД.
Тестовый запрос	api/v1/bonuses/ decrease / Body: <pre>{ "user_id": 1, "amount": 300 }</pre>
Пример работы	<pre>{ "id": "914784ed-1d4f-4200-9520-b5d0f237d5e1", "user": { "id": 1, "username": "admin", "amount_of_bonuses": "310" }, "amount": "300", "operation_type": { "id": 2, "type": "Credit" }, "expiration_date": null }</pre>

Адрес	“Domain.com”/api/v1/bonuses/multiple-increase/
Метод	POST
Право доступа	Администратор
Описание	Увеличивает количество бонусов некоторого количества пользователей посредством создания записей о новых операциях в БД.
Тестовый запрос	api/v1/bonuses/ multiple-increase/

	Body: <pre> { "user_ids": [{"id": 1}, {"id": 2}], "amount": 100, "expiration_date": "2019-10-22" } </pre>
Пример работы	<pre> [{ "id": "9399982c-62cb-4941-a47a-eec5a74ab6cf", "user": { "id": 1, "username": "admin", "amount_of_bonuses": "410" }, "amount": "100", "operation_type": { "id": 1, "type": "Debit" }, "expiration_date": "2019-10-22" }, { "id": "3e672397-0b7c-48fb-8ca9-186931249d78", "user": { "id": 2, "username": "test", "amount_of_bonuses": "160" }, "amount": "100", "operation_type": { "id": 1, "type": "Debit" }, "expiration_date": "2019-10-22" }] </pre>

Адрес	“Domain.com”/api/v1/bonuses/operation/<uuid:pk>/
Метод	GET, DELETE
Право доступа	Администратор
Описание	GET: Возвращает детали операции с id, указанным в url DELETE: Удаляет операцию с id, указанным в url
Тестовый запрос	api/v1/bonuses/operation/9399982c-62cb-4941-a47a-eec5a74ab6cf/
Пример работы	GET:

	<pre>{ "id": "9399982c-62cb-4941-a47a-ee5a74ab6cf", "user": { "id": 1, "username": "admin", "amount_of_bonuses": "410" }, "amount": "100", "operation_type": { "id": 1, "type": "Debit" }, "expiration_date": "2019-10-22" }</pre>
--	--

Адрес	“Domain.com”/api/v1/bonuses/operation/all/
Метод	GET
Право доступа	Администратор
Описание	Возвращает все операции
Тестовый запрос	api/v1/bonuses/operation/all/
Пример работы	<pre>[{ "id": "f3bad30c-ecd3-4625-b2e6-c2e0c3abe53f", "user": { "id": 2, "username": "test", "amount_of_bonuses": "160" }, "amount": "10", "operation_type": { "id": 1, "type": "Debit" }, "expiration_date": "2019-07-12" }, { "id": "f3bad32c-ecd3-4625-b2e6-c2e0c3abe53f", "user": { "id": 2, "username": "test", "amount_of_bonuses": "160" }, "amount": "50", "operation_type": { "id": 1, "type": "Debit" }, "expiration_date": null }]</pre>

Для работы с API нужны зарегистрированные пользователи.

Администратор: (login:admin, password:1234)

Пользователь: (login:test, password:Test123456)

В приложении подключены стандартные системы авторизации Django, поддерживающие авторизацию по сессиям (для упрощённого тестирования через браузер) и токенам.

Список URL их эндпоинтов можно найти по адресу:

`“Domain.com”/api/v1/auth/`

Например, для получения токена необходимо отправить POST-запрос по адресу:

`“Domain.com”/api/v1/auth/api/v1/auth-token/token/login/`

С телом

```
{  
  "username": "test",  
  "password": "Test123456"  
}
```

В ответ вернется токен, который необходимо добавлять в headers запроса, под ключом Authorization и добавив слово token перед его значением.

Для удобства тестирования в приложении включена возможность доступа через браузер.