Текстовые поля хранятся в виде массивов символов, поля для целых чисел обозначены как «цел.», для дробных чисел – «дроб.»

Необязательные данные отмечены *курсивом*.

# Регистрация пользователя:

**Проверка на то, зарегестрирован ли пользователь с таким именем**

SELECT COUNT (\*) FROM public."user" WHERE username = 'имя\_пользователя'

*(если нет – эта функция вернёт 0)*

## Внесение нового пользователя в БД

INSERT INTO public."user" (username, password) VALUES ('имя\_пользователя', 'пароль, пропущенный через алгоритм MD5');

**Создание таблицы данных для конкретного пользователя**

INSERT INTO public.cv\_data (user\_id) VALUES ((SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

**Создание таблицы стилей для конкретного пользователя**

INSERT INTO public.style (user\_id) VALUES ((SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

# Логин пользователя:

**Получение зашифрованного пароля и идентификатора от зарегестрированного пользователя**

SELECT user\_id, password FROM public."user" WHERE username = 'имя\_пользователя';

**Возвращает ошибку в случае, если пользователь с таким именем не зарегестрирован**

# Взаимодействие с учебными заведениями:

**Вставка учебного заведения по имени пользователя**

INSERT INTO public.education(cv\_id, educ\_title, educ\_desc, start\_year, *end\_year*) VALUES ((SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя')), 'название\_учебного\_заведения', 'описание\_учебного\_заведения', 'год\_начала (цел.)', *'год\_окончания (цел.)'*);

**Получение списка учебных заведений**

SELECT educ\_title, educ\_desc, start\_year, end\_year FROM public.education WHERE cv\_id = (SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

**Удаление учебного заведения**

DELETE FROM public.education WHERE education\_title = 'название\_учебного\_заведения' AND cv\_id = (SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

# Взаимодействие с навыками hard:

**Вставка навыка hard по имени пользователя**

INSERT INTO public.hard\_skill(cv\_id, hard\_skill\_name, hard\_skill\_eval) VALUES ((SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя')), 'название\_навыка\_hard', 'оценивание\_навыка\_hard (цел.)');

**Получение списка навыков hard по имени пользователя [названия навыков]**

SELECT hard\_skill\_name, hard\_skill\_eval FROM public.hard\_skill WHERE cv\_id = (SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

**Удаление навыка hard по названию навыка и имени пользователя**

DELETE FROM public.hard\_skill WHERE hard\_skill\_name = 'имя\_навыка\_hard' AND cv\_id = (SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

# Взаимодействие с навыками soft:

**Вставка навыка soft по имени пользователя**

INSERT INTO public.soft\_skill(cv\_id, soft\_skill\_name) VALUES ((SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя')), 'название\_навыка\_soft');

**Получение списка навыков soft по имени пользователя [названия навыков]**

SELECT soft\_skill\_name FROM public.soft\_skill WHERE cv\_id = (SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

**Удаление навыка soft по названию навыка и имени пользователя**

DELETE FROM public.soft\_skill WHERE soft\_skill\_name = 'имя\_навыка\_soft' AND cv\_id = (SELECT cv\_id FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя'));

# Обновление таблиц новыми данными и их чтение:

**Смена пароля у пользователя по имени пользователя**

UPDATE public."user" SET password = 'новый\_пароль\_через\_MD5' WHERE username = 'имя\_пользователя';

**Введение или изменение данных в CV по имени пользователя**

UPDATE public.cv\_data SET *first\_name = 'настоящее\_имя', last\_name = 'фамилия', about\_me = 'обо\_мне', short\_desc = 'главное\_описание\_пользователя'* WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя');

**Получение данных из таблицы данных**

SELECT *first\_name, last\_name, about\_me, short\_desc* FROM public.cv\_data WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя');

**Изменение стиля страницы CV по имени пользователя**

UPDATE public.style SET *left\_margin = 'левая\_граница (цел.)', right\_margin = 'правая\_граница (цел.)', top\_margin = 'верхняя\_граница (цел.)', text\_size = 'размер\_текста (цел.)', header\_size = 'размер\_заголовка (цел.)'* WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя');

*Подразумевается, что все размеры и отступы будут храниться в пикселях*

**Получение данных стиля страницы CV**

SELECT *left\_margin, right\_margin, top\_margin, text\_size, header\_size* FROM public.style WHERE user\_id = (SELECT user\_id FROM public."user" WHERE username = 'имя\_пользователя');

## Просмотр всех зарегестрированных пользователей

SELECT \* FROM public."user" ORDER BY user\_id ASC;