

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Базы данных»**  
**Тема: Реализация базы данных в СУБД PostgreSQL.**

Студент гр. 1304

Шаврин А.П.

Преподаватель

Заславский М.М

Санкт-Петербург

2023

## **Цель работы.**

Реализация базы данных в СУБД PostgreSQL.

## **Задание.**

Вариант 4.

Пусть требуется создать программную систему, предназначенную для организаторов выставки собак. Она должна обеспечивать хранение сведений о собаках - участниках выставки и экспертах. Для каждой собаки в БД должны храниться сведения, о том, к какому клубу она относится, кличка, порода и возраст, сведения о родословной (номер документа, клички родителей), дата последней прививки, фамилия, имя, отчество и паспортные данные хозяина. На каждый клуб отводится участок номеров, под которыми будут выступать участники выставки. Сведения об эксперте должны включать фамилию и имя, номер ринга, который он обслуживает; клуб, название клуба, в котором он состоит. Каждый ринг могут обслуживать несколько экспертов. Каждая порода собак выступает на своем ринге, но на одном и том же ринге в разное время могут выступать разные породы. Итогом выставки является определение медалистов по каждой породе. Организатор выставки должен иметь возможность добавить в базу нового участника или нового эксперта, снять эксперта с судейства, заменив его другим, отстранить собаку от участия в выставке. Организатору выставки могут потребоваться следующие сведения;

- На каком ринге выступает заданный хозяин со своей собакой?
- Какими породами представлен заданный клуб?
- Какие медали и сколько заслужены клубом?
- Какие эксперты обслуживают породу?
- Количество участников по каждой породе?

Необходимо развернуть PostgreSQL локально:

- Написать запросы для создания таблиц из предыдущей лабораторной работы

- Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
- Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
- Исходный код выложить на [www.db-fiddle.com](http://www.db-fiddle.com) для проверки работоспособности
- Исходный код в виде .sql файла загрузить в виде PR в репо
- В отчете описать:
  - Цель
  - Текст задания в соответствии с вариантом
  - Скриншоты работы с СУБД PostgreSQL (psql / DBeaver / Datagrip, ...)
  - Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
  - Исходный код в приложении
  - Ссылку на исходный код [www.db-fiddle.com](http://www.db-fiddle.com) в приложении
  - Ссылка на PR в приложении
  - Вывод

### **Выполнение работы.**

1. Сперва была установлена СУБД PostgreSQL на Linux и pgAdmin4 на Windows. Также для postgresql был настроен доступ из вне.

```
alex@alex-VirtualBox:~$ pg_config --version
PostgreSQL 15.4 (Ubuntu 15.4-0ubuntu0.23.04.1)
```

Рис 1. Версия PostgreSQL.

```

GNU nano 7.2 /etc/postgresql/15/main/postgresql.conf
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to listen on;

```

Рис 2. Настройка postgresql.conf

```

GNU nano 7.2 /etc/postgresql/15/main/pg_hba.conf
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer
host    all             all                0.0.0.0/0          scram-sha-256

```

Рис 3. Настройка pg\_hba.conf

2. Затем СУБД была запущена.

```

alex@alex-VirtualBox:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Mon 2023-10-02 23:01:18 MSK; 16h ago
   Process: 4214 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 4214 (code=exited, status=0/SUCCESS)
   CPU: 1ms

Oct 02 23:01:18 alex-VirtualBox systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Oct 02 23:01:18 alex-VirtualBox systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.

```

Рис 4. Проверка статуса СУБД.

3. После был сменен пользователь на postgres и была создана таблица dog\_show командой createdb dog\_show.

```

postgres=# \list

```

Name	Owner	Encoding	Collate	Ctype	ICU Locale	Locale Provider	Access privileges
dog_show	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		libc	

Рис 5. Создание БД.

4. Далее в командной строке psql были созданы таблицы посредством запуска группы sql запросов.

```
dog_show=# \dt
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | breed | table | postgres
public | club | table | postgres
public | club_numbers | table | postgres
public | dog | table | postgres
public | dog_expert_estimate | table | postgres
public | expert | table | postgres
public | owner | table | postgres
public | ring | table | postgres
(8 rows)
```

Рис 6. Создание таблиц.

```
dog_show=# \d ring
Table "public.ring"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
ring_number | integer | | not null | nextval('ring_ring_number_seq'::regclass)
Indexes:
    "ring_pkey" PRIMARY KEY, btree (ring_number)
Referenced by:
    TABLE "breed" CONSTRAINT "breed_ring_number_fkey" FOREIGN KEY (ring_number) REFERENCES ring(ring_number)
    TABLE "expert" CONSTRAINT "expert_ring_number_fkey" FOREIGN KEY (ring_number) REFERENCES ring(ring_number)
```

Рис 7. Структура таблицы ring.

```
dog_show=# \d club
Table "public.club"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
club_id | integer | | not null | nextval('club_club_id_seq'::regclass)
club_name | character varying(255) | | | 
participants_number | integer | | | 
Indexes:
    "club_pkey" PRIMARY KEY, btree (club_id)
Referenced by:
    TABLE "club_numbers" CONSTRAINT "club_numbers_club_id_fkey" FOREIGN KEY (club_id) REFERENCES club(club_id)
    TABLE "expert" CONSTRAINT "expert_club_id_fkey" FOREIGN KEY (club_id) REFERENCES club(club_id)
```

Рис 8. Структура таблицы club.

```
dog_show=# \d expert
Table "public.expert"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
expert_id | integer | | not null | nextval('expert_expert_id_seq'::regclass)
surname | character varying(255) | | | 
name | character varying(255) | | | 
club_id | integer | | not null | nextval('expert_club_id_seq'::regclass)
ring_number | integer | | not null | nextval('expert_ring_number_seq'::regclass)
Indexes:
    "expert_pkey" PRIMARY KEY, btree (expert_id)
Foreign-key constraints:
    "expert_club_id_fkey" FOREIGN KEY (club_id) REFERENCES club(club_id)
    "expert_ring_number_fkey" FOREIGN KEY (ring_number) REFERENCES ring(ring_number)
Referenced by:
    TABLE "dog_expert_estimate" CONSTRAINT "dog_expert_estimate_expert_id_fkey" FOREIGN KEY (expert_id) REFERENCES expert(expert_id)
```

Рис 9. Структура таблицы expert.



```

dog_show=# \d breed
               Table "public.breed"
   Column   |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
breed_name | character varying(255) |           | not null |
ring_number | integer          |           | not null | nextval('breed_ring_number_seq'::regclass)
Indexes:
    "breed_pkey" PRIMARY KEY, btree (breed_name)
Foreign-key constraints:
    "breed_ring_number_fkey" FOREIGN KEY (ring_number) REFERENCES ring(ring_number)
Referenced by:
    TABLE "dog" CONSTRAINT "dog_breed_name_fkey" FOREIGN KEY (breed_name) REFERENCES breed(breed_name)

```

Рис 10. Структура таблицы breed.

```

               Table "public.dog"
   Column   |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
dog_number  | integer          |           | not null | nextval('dog_dog_number_seq'::regclass)
owner_passport | character varying(11) |           |          |
breed_name  | character varying(255) |           |          |
pedigree_document_number | integer          |           | not null |
mother_nickname | character varying(255) |           |          |
father_nickname | character varying(255) |           |          |
nickname    | character varying(255) |           |          |
age         | integer          |           |          |
vaccination_date | date             |           |          |
Indexes:
    "dog_pkey" PRIMARY KEY, btree (dog_number)
Foreign-key constraints:
    "dog_breed_name_fkey" FOREIGN KEY (breed_name) REFERENCES breed(breed_name)
    "dog_owner_passport_fkey" FOREIGN KEY (owner_passport) REFERENCES owner(passport)
Referenced by:
    TABLE "club_numbers" CONSTRAINT "club_numbers_dog_number_fkey" FOREIGN KEY (dog_number) REFERENCES dog(dog_number)
    TABLE "dog_expert_estimate" CONSTRAINT "dog_expert_estimate_dog_number_fkey" FOREIGN KEY (dog_number) REFERENCES dog(dog_number)

```

Рис 11. Структура таблицы dog.

```

dog_show=# \d dog_expert_estimate
               Table "public.dog_expert_estimate"
   Column   |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
dog_number  | integer          |           | not null | nextval('dog_expert_estimate_dog_number_seq'::regclass)
expert_id   | integer          |           | not null | nextval('dog_expert_estimate_expert_id_seq'::regclass)
estimate    | integer          |           |          |
Indexes:
    "dog_expert_estimate_pkey" PRIMARY KEY, btree (dog_number, expert_id)
Foreign-key constraints:
    "dog_expert_estimate_dog_number_fkey" FOREIGN KEY (dog_number) REFERENCES dog(dog_number)
    "dog_expert_estimate_expert_id_fkey" FOREIGN KEY (expert_id) REFERENCES expert(expert_id)

```

Рис 12. Структура таблицы dog\_expert\_estimate.

```

dog_show=# \d club_numbers
               Table "public.club_numbers"
   Column   |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
dog_number  | integer          |           | not null | nextval('club_numbers_dog_number_seq'::regclass)
club_id     | integer          |           | not null | nextval('club_numbers_club_id_seq'::regclass)
Indexes:
    "club_numbers_pkey" PRIMARY KEY, btree (dog_number, club_id)
Foreign-key constraints:
    "club_numbers_club_id_fkey" FOREIGN KEY (club_id) REFERENCES club(club_id)
    "club_numbers_dog_number_fkey" FOREIGN KEY (dog_number) REFERENCES dog(dog_number)

```

Рис 13. Структура таблицы club\_numbers.

```
dog_show=# \d owner
Table "public.owner"
  Column      |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
passport      | character varying(11)  |           | not null |
surname       | character varying(255) |           |         |
name          | character varying(255) |           |         |
patronymic    | character varying(255) |           |         |
Indexes:
    "owner_pkey" PRIMARY KEY, btree (passport)
Referenced by:
    TABLE "dog" CONSTRAINT "dog_owner_passport_fkey" FOREIGN KEY (owner_passport) REFERENCES owner(passport)
```

Рис 14. Структура таблицы owner.

Дальнейшие действия с базой данных осуществлялись через pgAdmin4

5. Все таблицы были заполнены минимум 5 записями посредством группы SQL запросов.

```
1 INSERT INTO ring
2 VALUES (DEFAULT), (DEFAULT), (DEFAULT), (DEFAULT), (DEFAULT);
3
4 SELECT * FROM ring;
```

ring_number
1
2
3
4
5

Рис 15. Заполнение таблицы ring.

```
1 INSERT INTO breed (breed_name, ring_number)
2 VALUES ('Немецкая овчарка', 1),
3         ('Доберман', 1),
4         ('Померанский шпиц', 2),
5         ('Йоркширский терьер', 3),
6         ('Бультерьер', 4),
7         ('Пудель', 5);
8
9 SELECT * FROM breed;
```

breed_name	ring_number
Немецкая овчарка	1
Доберман	1
Померанский шпиц	2
Йоркширский терьер	3
Бультерьер	4
Пудель	5

Рис 16. Заполнение таблицы breed.

```

1 INSERT INTO club (club_name, participants_number)
2 VALUES ('Фаворит', 4),
3         ('Оскар', 4),
4         ('Статус', 4),
5         ('Звезда', 3),
6         ('Победа', 2);
7
8 SELECT * FROM club;

```

Data Output Messages Notifications			
	club_id [PK] integer	club_name character varying (255)	participants_number integer
1	1	Фаворит	4
2	2	Оскар	4
3	3	Статус	4
4	4	Звезда	3
5	5	Победа	2

Рис 17. Заполнение таблицы club.

```

1 INSERT INTO expert (club_id, ring_number, surname, name)
2 VALUES (1, 1, 'Кардаш', 'Ярослав'),
3         (2, 1, 'Шаврин', 'Алексей'),
4         (2, 1, 'Дешура', 'Дмитрий'),
5         (3, 3, 'Ефремов', 'Артем'),
6         (4, 4, 'Есенина', 'Анастасия'),
7         (5, 5, 'Насекина', 'Алена'),
8         (5, 2, 'Лепс', 'Григорий');
9
10 SELECT * FROM expert;

```

Data Output Messages Notifications					
	expert_id [PK] integer	surname character varying (255)	name character varying (255)	club_id integer	ring_number integer
1	1	Кардаш	Ярослав	1	1
2	2	Шаврин	Алексей	2	1
3	3	Дешура	Дмитрий	2	1
4	4	Ефремов	Артем	3	3
5	5	Есенина	Анастасия	4	4
6	6	Насекина	Алена	5	5
7	7	Лепс	Григорий	5	2

Рис 18. Заполнение таблицы expert.

```

1 INSERT INTO owner (passport, surname, name, patronymic)
2 VALUES ('1111 222222', 'Потолов', 'Олег', 'Алексеевич'),
3         ('2222 333333', 'Зубов', 'Александр', 'Андреевич'),
4         ('3333 444444', 'Башкирев', 'Алексей', 'Владимирович'),
5         ('4444 555555', 'Иванов', 'Иван', 'Иванович'),
6         ('5555 666666', 'Добрицкий', 'Ярослав', 'Валерьевич'),
7         ('6666 777777', 'Игнатьев', 'Владимир', 'Владимирович'),
8         ('7777 888888', 'Бочаров', 'Никита', 'Игоревич'),
9         ('8888 999999', 'Данилов', 'Дмитрий', 'Семенович');
10 SELECT * FROM owner;

```

Data Output Messages Notifications				
	passport [PK] character varying (11)	surname character varying (255)	name character varying (255)	patronymic character varying (255)
1	1111 222222	Потолов	Олег	Алексеевич
2	2222 333333	Зубов	Александр	Андреевич
3	3333 444444	Башкирев	Алексей	Владимирович
4	4444 555555	Иванов	Иван	Иванович
5	5555 666666	Добрицкий	Ярослав	Валерьевич
6	6666 777777	Игнатьев	Владимир	Владимирович
7	7777 888888	Бочаров	Никита	Игоревич
8	8888 999999	Данилов	Дмитрий	Семенович

Рис 19. Заполнение таблицы owner.



```

1 INSERT INTO dog
2 VALUES (DEFAULT, '1111 22222', 'Немецкая овчарка', 1111111, 'Элис', 'Арчи', 'Рекс', 5, '2023-10-01'),
3 (DEFAULT, '1111 22222', 'Доберман', 1111112, 'Меган', 'Дизель', 'Рокс', 7, '2023-09-27'),
4 (DEFAULT, '2222 33333', 'Немецкая овчарка', 2222221, 'Меган', 'Дизель', 'Рокс', 8, '2023-09-06'),
5 (DEFAULT, '2222 33333', 'Доберман', 2222222, 'Элис', 'Арчи', 'Рекс', 6, '2023-09-06'),
6 (DEFAULT, '3333 44444', 'Немецкая овчарка', 3333331, 'Элли', 'Оливер', 'Раймонд', 7, '2023-09-12'),
7 (DEFAULT, '3333 44444', 'Доберман', 3333332, 'Эмма', 'Руди', 'Алекс', 9, '2023-09-23'),
8 (DEFAULT, '4444 55555', 'Немецкая овчарка', 4444441, 'Герда', 'Пит', 'Балу', 5, '2023-09-29'),
9 (DEFAULT, '4444 55555', 'Пудель', 4444442, 'Витта', 'Мэверик', 'Гектор', 5, '2023-09-22'),
10 (DEFAULT, '5555 66666', 'Бультерьер', 5555551, 'Грейт', 'Веня', 'Барон', 8, '2023-09-15'),
11 (DEFAULT, '5555 66666', 'Пудель', 5555552, 'Инга', 'Ефим', 'Герой', 4, '2023-09-16'),
12 (DEFAULT, '6666 77777', 'Бультерьер', 6666661, 'Лейла', 'Джанго', 'Кекс', 4, '2023-09-03'),
13 (DEFAULT, '6666 77777', 'Пудель', 6666662, 'Скай', 'Зевс', 'Моджо', 7, '2023-09-08'),
14 (DEFAULT, '7777 88888', 'Померанский шпиц', 7777771, 'Руни', 'Мич', 'Зефирка', 4, '2023-10-01'),
15 (DEFAULT, '7777 88888', 'Йоркширский терьер', 7777772, 'Барби', 'Ник', 'Пряник', 4, '2023-09-28'),
16 (DEFAULT, '8888 99999', 'Померанский шпиц', 8888881, 'Джесси', 'Зик', 'Ерошка', 6, '2023-09-20'),
17 (DEFAULT, '8888 99999', 'Йоркширский терьер', 8888882, 'Кристи', 'Сириус', 'Цезарь', 6, '2023-09-17'),
18 (DEFAULT, '8888 99999', 'Йоркширский терьер', 8888883, 'Фокс', 'Арчи', 'Тимка', 7, '2023-09-17');
19 SELECT * FROM dog;

```

	dog_number [PK] integer	owner_passport character varying (11)	breed_name character varying (255)	pedigree_document_number integer	mother_nickname character varying (255)	father_nickname character varying (255)	nickname character varying (255)	age integer	vaccination_date date
1	1	1111 22222	Немецкая овчарка	1111111	Элис	Арчи	Рекс	5	2023-10-01
2	2	1111 22222	Доберман	1111112	Меган	Дизель	Рокс	7	2023-09-27
3	3	2222 33333	Немецкая овчарка	2222221	Меган	Дизель	Рокс	8	2023-09-06
4	4	2222 33333	Доберман	2222222	Элис	Арчи	Рекс	6	2023-09-06
5	5	3333 44444	Немецкая овчарка	3333331	Элли	Оливер	Раймонд	7	2023-09-12
6	6	3333 44444	Доберман	3333332	Эмма	Руди	Алекс	9	2023-09-23
7	7	4444 55555	Немецкая овчарка	4444441	Герда	Пит	Балу	5	2023-09-29
8	8	4444 55555	Пудель	4444442	Витта	Мэверик	Гектор	5	2023-09-22
9	9	5555 66666	Бультерьер	5555551	Грейт	Веня	Барон	8	2023-09-15
10	10	5555 66666	Пудель	5555552	Инга	Ефим	Герой	4	2023-09-16
11	11	6666 77777	Бультерьер	6666661	Лейла	Джанго	Кекс	4	2023-09-03
12	12	6666 77777	Пудель	6666662	Скай	Зевс	Моджо	7	2023-09-08
13	13	7777 88888	Померанский шпиц	7777771	Руни	Мич	Зефирка	4	2023-10-01
14	14	7777 88888	Йоркширский терьер	7777772	Барби	Ник	Пряник	4	2023-09-28
15	15	8888 99999	Померанский шпиц	8888881	Джесси	Зик	Ерошка	6	2023-09-20
16	16	8888 99999	Йоркширский терьер	8888882	Кристи	Сириус	Цезарь	6	2023-09-17
17	17	8888 99999	Йоркширский терьер	8888883	Фокс	Арчи	Тимка	7	2023-09-17

Рис 20. Заполнение dog.

```

1 INSERT INTO club_numbers
2 VALUES (1, 1), (2, 1),
3 (3, 1), (4, 1),
4 (5, 2), (6, 2),
5 (7, 2), (8, 2),
6 (9, 3), (10, 3),
7 (11, 3), (12, 3),
8 (13, 5), (14, 5),
9 (15, 4), (16, 4),
10 (17, 4);
11
12 SELECT * FROM club_numbers;

```

	dog_number [PK] integer	club_id [PK] integer
1	1	1
2	2	1
3	3	1
4	4	1
5	5	2
6	6	2
7	7	2
8	8	2
9	9	3
10	10	3
11	11	3
12	12	3
13	13	5
14	14	5
15	15	4
16	16	4
17	17	4

Рис 21. Заполнение club\_numbers.

```

1 INSERT INTO dog_expert_estimate (dog_number, expert_id, estimate)
2 VALUES (1, 1, 9), (3, 1, 6), (5, 1, 8), (7, 1, 7),
3          (1, 2, 10), (3, 2, 7), (5, 2, 9), (7, 2, 8),
4          (2, 3, 7), (4, 3, 10), (6, 3, 8),
5          (8, 6, 10), (10, 6, 10), (12, 6, 8),
6          (9, 5, 8), (11, 5, 9),
7          (14, 4, 5), (16, 4, 7), (17, 4, 8),
8          (13, 7, 10), (15, 7, 10);
9
10 SELECT * FROM dog_expert_estimate;

```

	dog_number [PK] integer	expert_id [PK] integer	estimate integer
1		1	9
2		3	1
3		5	1
4		7	1
5		1	2
6		3	2
7		5	2
8		7	2
9		2	3
10		4	3
11		6	3
12		8	6
13		10	6
14		12	6
15		9	5
16		11	5
17		14	4
18		16	4
19		17	4
20		13	7
21		15	7

Рис 21. Заполнение dog expert estimate.

6. Затем были реализованы запросы, отвечающие на вопросы.

- На каком ринге выступает заданный хозяин со своей собакой?

```
1 SELECT breed.ring_number, dog.dog_number, owner.surname, owner.name, owner.patronymic
2 FROM owner
3     INNER JOIN dog ON owner.passport = dog.owner_passport
4     INNER JOIN breed USING(breed_name)
5 ORDER BY breed.ring_number, dog.dog_number, owner.surname, owner.name, owner.patronymic;
```

	ring_number integer	dog_number integer	surname character varying (255)	name character varying (255)	patronymic character varying (255)
1	1	1	Потолов	Олег	Алексеевич
2	1	2	Потолов	Олег	Алексеевич
3	1	3	Зубов	Александр	Андреевич
4	1	4	Зубов	Александр	Андреевич
5	1	5	Башкирев	Алексей	Владимирович
6	1	6	Башкирев	Алексей	Владимирович
7	1	7	Иванов	Иван	Иванович
8	2	13	Бочаров	Никита	Игоревич
9	2	15	Данилов	Дмитрий	Семенович
10	3	14	Бочаров	Никита	Игоревич
11	3	16	Данилов	Дмитрий	Семенович
12	3	17	Данилов	Дмитрий	Семенович
13	4	9	Добрицкий	Ярослав	Валерьевич
14	4	11	Игнатьев	Владимир	Владимирович
15	5	8	Иванов	Иван	Иванович
16	5	10	Добрицкий	Ярослав	Валерьевич
17	5	12	Игнатьев	Владимир	Владимирович

Рис 22. Запрос №1.

- Какими породами представлен заданный клуб?

```

1  SELECT club.club_id, dog.breed_name
2  FROM club
3      INNER JOIN club_numbers USING(club_id)
4      INNER JOIN dog USING(dog_number)
5  GROUP BY club.club_id, dog.breed_name
6  ORDER BY club.club_id, dog.breed_name;
7

```

	club_id integer	breed_name character varying (255)
1	1	Доберман
2	1	Немецкая овчарка
3	2	Доберман
4	2	Немецкая овчарка
5	2	Пудель
6	3	Бультерьер
7	3	Пудель
8	4	Йоркширский терьер
9	4	Померанский шпиц
10	5	Йоркширский терьер
11	5	Померанский шпиц

Рис 22. Запрос №2.

- Какие медали и сколько заслужены клубом?

```

1  WITH dog_places AS (
2      SELECT dog.breed_name, dog.dog_number, ROUND(AVG(dog_expert_estimate.estimate), 2) AS Estimate,
3             DENSE_RANK() OVER (PARTITION BY dog.breed_name ORDER BY ROUND(AVG(dog_expert_estimate.estimate), 2) DESC) AS place
4      FROM dog
5      INNER JOIN dog_expert_estimate USING(dog_number)
6      GROUP BY dog.breed_name, dog.dog_number
7      ORDER BY dog.breed_name, Estimate DESC
8  )
9  SELECT club.club_id, club.club_name, dog_places.place, COUNT(dog_places.place)
10 FROM club
11     INNER JOIN club_numbers USING(club_id)
12     INNER JOIN dog_places USING(dog_number)
13 GROUP BY club.club_id, club.club_name, dog_places.place
14 ORDER BY club.club_id, dog_places.place;

```

	club_id [PK] integer	club_name character varying (255)	place bigint	count bigint
1	1	Фаворит	1	2
2	1	Фаворит	3	1
3	1	Фаворит	4	1
4	2	Оскар	1	1
5	2	Оскар	2	2
6	2	Оскар	3	1
7	3	Статус	1	2
8	3	Статус	2	2
9	4	Звезда	1	2
10	4	Звезда	2	1
11	5	Победа	1	1
12	5	Победа	3	1

Рис 23. Запрос №3.

- Какие эксперты обслуживают породу?

```

1  SELECT dog.breed_name, expert.expert_id, expert.name, expert.surname
2  FROM dog
3      INNER JOIN dog_expert_estimate USING(dog_number)
4      INNER JOIN expert USING(expert_id)
5  GROUP BY dog.breed_name, expert.expert_id
6  ORDER BY dog.breed_name;
7

```

	breed_name character varying (255)	expert_id integer	name character varying (255)	surname character varying (255)
1	Бультерьер	5	Анастасия	Есенина
2	Доберман	3	Дмитрий	Дешура
3	Йоркширский терьер	4	Артём	Ефремов
4	Немецкая овчарка	1	Ярослав	Кардаш
5	Немецкая овчарка	2	Алексей	Шаврин
6	Померанский шпиц	7	Григорий	Лепс
7	Пудель	6	Алена	Насекина

Рис 24. Запрос №4.

- Количество участников по каждой породе?

```

1  SELECT dog.breed_name, COUNT(*)
2  FROM dog
3  GROUP BY dog.breed_name
4  ORDER BY dog.breed_name;
5

```

	breed_name character varying (255)	count bigint
1	Бультерьер	2
2	Доберман	3
3	Йоркширский терьер	3
4	Немецкая овчарка	4
5	Померанский шпиц	2
6	Пудель	3

Рис 25. Запрос №5.

Разработанный программный код см. в приложении А.

Ссылки на PR и на исходный код [www.db-fiddle.com](http://www.db-fiddle.com) см. в приложении В.

## Выводы.

В ходе выполнения работы, реализована база данных в СУБД PostgreSQL, изучены основы работы с PostgreSQL и pgAdmin4.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab2.sql

```
CREATE TABLE owner (  
    passport VARCHAR(11) PRIMARY KEY,  
    surname VARCHAR(255),  
    name VARCHAR(255),  
    patronymic VARCHAR(255)  
);  
  
CREATE TABLE ring (  
    ring_number serial PRIMARY KEY  
);  
  
CREATE TABLE breed (  
    breed_name VARCHAR(255) PRIMARY KEY,  
    ring_number serial,  
    FOREIGN KEY (ring_number) REFERENCES ring (ring_number)  
);  
  
CREATE TABLE club (  
    club_id serial PRIMARY KEY,  
    club_name VARCHAR(255),  
    participants_number INT  
);  
  
CREATE TABLE expert (  
    expert_id serial PRIMARY KEY,  
    surname VARCHAR(255),  
    name VARCHAR(255),  
    club_id serial,  
    ring_number serial,  
    FOREIGN KEY (club_id) REFERENCES club (club_id),  
    FOREIGN KEY (ring_number) REFERENCES ring (ring_number)  
);  
  
CREATE TABLE dog (  
    dog_number serial PRIMARY KEY,  
    owner_passport VARCHAR(11),  
    breed_name VARCHAR(255),  
    FOREIGN KEY (owner_passport) REFERENCES owner (passport),  
    FOREIGN KEY (breed_name) REFERENCES breed (breed_name),  
    pedigree_document_number INT NOT NULL,  
    mother_nickname VARCHAR(255),  
    father_nickname VARCHAR(255),  
    nickname VARCHAR(255),  
    age INT,  
    vaccination_date DATE
```



```

);

CREATE TABLE dog_expert_estimate (
    dog_number serial,
    expert_id serial,
    FOREIGN KEY (dog_number) REFERENCES dog (dog_number),
    FOREIGN KEY (expert_id) REFERENCES expert (expert_id),
    estimate INT,
    PRIMARY KEY (dog_number, expert_id)
);

CREATE TABLE club_numbers (
    dog_number serial,
    club_id serial,
    FOREIGN KEY (dog_number) REFERENCES dog (dog_number),
    FOREIGN KEY (club_id) REFERENCES club (club_id),
    PRIMARY KEY (dog_number, club_id)
);

INSERT INTO ring
VALUES (DEFAULT), (DEFAULT), (DEFAULT), (DEFAULT), (DEFAULT);

INSERT INTO breed (breed_name, ring_number)
VALUES ('Немецкая овчарка', 1),
       ('Доберман', 1),
       ('Померанский шпиц', 2),
       ('Йоркширский терьер', 3),
       ('Бультерьер', 4),
       ('Пудель', 5);

INSERT INTO club (club_name, participants_number)
VALUES ('Фаворит', 4),
       ('Оскар', 4),
       ('Статус', 4),
       ('Звезда', 3),
       ('Победа', 2);

INSERT INTO expert (club_id, ring_number, surname, name)
VALUES (1, 1, 'Кардаш', 'Ярослав'),
       (2, 1, 'Шаврин', 'Алексей'),
       (2, 1, 'Дешура', 'Дмитрий'),
       (3, 3, 'Ефремов', 'Артем'),
       (4, 4, 'Есенина', 'Анастасия'),
       (5, 5, 'Насекина', 'Алена'),
       (5, 2, 'Лепс', 'Григорий');

```

```

INSERT INTO owner (passport, surname, name, patronymic)
VALUES ('1111 222222', 'Потолов', 'Олег', 'Алексеевич'),
      ('2222 333333', 'Зубов', 'Александр', 'Андреевич'),
      ('3333 444444', 'Башкирев', 'Алексей', 'Владимирович'),
      ('4444 555555', 'Иванов', 'Иван', 'Иванович'),
      ('5555 666666', 'Добрицкий', 'Ярослав', 'Валерьевич'),
      ('6666 777777', 'Игнатъев', 'Владимир', 'Владимирович'),
      ('7777 888888', 'Бочаров', 'Никита', 'Игоревич'),
      ('8888 999999', 'Данилов', 'Дмитрий', 'Семенович');

```

```

INSERT INTO dog
VALUES (DEFAULT, '1111 222222', 'Немецкая овчарка', 1111111, 'Элис',
      'Арчи', 'Рекс', 5, '2023-10-01'),
      (DEFAULT, '1111 222222', 'Доберман', 1111112, 'Меган',
      'Дизель', 'Рокс', 7, '2023-09-27'),
      (DEFAULT, '2222 333333', 'Немецкая овчарка', 2222221, 'Меган',
      'Дизель', 'Рокс', 8, '2023-09-06'),
      (DEFAULT, '2222 333333', 'Доберман', 2222222, 'Элис', 'Арчи',
      'Рекс', 6, '2023-09-06'),
      (DEFAULT, '3333 444444', 'Немецкая овчарка', 3333331, 'Элли',
      'Оливер', 'Раймонд', 7, '2023-09-12'),
      (DEFAULT, '3333 444444', 'Доберман', 3333332, 'Эмма', 'Руди',
      'Алекс', 9, '2023-09-23'),
      (DEFAULT, '4444 555555', 'Немецкая овчарка', 4444441, 'Герда',
      'Пит', 'Балу', 5, '2023-09-29'),
      (DEFAULT, '4444 555555', 'Пудель', 4444442, 'Витта', 'Мэверик',
      'Гектор', 5, '2023-09-22'),
      (DEFAULT, '5555 666666', 'Бультерьер', 5555551, 'Трейт',
      'Веня', 'Барон', 8, '2023-09-15'),
      (DEFAULT, '5555 666666', 'Пудель', 5555552, 'Инга', 'Ефим',
      'Герой', 4, '2023-09-16'),
      (DEFAULT, '6666 777777', 'Бультерьер', 6666661, 'Лейла',
      'Джанго', 'Кекс', 4, '2023-09-03'),
      (DEFAULT, '6666 777777', 'Пудель', 6666662, 'Скай', 'Зевс',
      'Моджо', 7, '2023-09-08'),
      (DEFAULT, '7777 888888', 'Померанский шпиц', 7777771, 'Руни',
      'Мич', 'Зефирка', 4, '2023-10-01'),
      (DEFAULT, '7777 888888', 'Йоркширский терьер', 7777772,
      'Барби', 'Ник', 'Пряник', 4, '2023-09-28'),
      (DEFAULT, '8888 999999', 'Померанский шпиц', 8888881, 'Джесси',
      'Зик', 'Ерошка', 6, '2023-09-20'),
      (DEFAULT, '8888 999999', 'Йоркширский терьер', 8888882,
      'Кристи', 'Сириус', 'Цезарь', 6, '2023-09-17'),
      (DEFAULT, '8888 999999', 'Йоркширский терьер', 8888883, 'Фокс',
      'Арчи', 'Тимка', 7, '2023-09-17');

```

```

INSERT INTO club_numbers
VALUES (1, 1), (2, 1),

```

```

(3, 1), (4, 1),
(5, 2), (6, 2),
(7, 2), (8, 2),
(9, 3), (10, 3),
(11, 3), (12, 3),
(13, 5), (14, 5),
(15, 4), (16, 4),
(17, 4);

```

```

INSERT INTO dog_expert_estimate (dog_number, expert_id, estimate)
VALUES (1, 1, 9), (3, 1, 6), (5, 1, 8), (7, 1, 7),
(1, 2, 10), (3, 2, 7), (5, 2, 9), (7, 2, 8),
(2, 3, 7), (4, 3, 10), (6, 3, 8),
(8, 6, 10), (10, 6, 10), (12, 6, 8),
(9, 5, 8), (11, 5, 9),
(14, 4, 5), (16, 4, 7), (17, 4, 8),
(13, 7, 10), (15, 7, 10);

```

```

SELECT breed.ring_number, dog.dog_number, owner.surname, owner.name,
owner.patronymic
FROM owner
INNER JOIN dog ON owner.passport = dog.owner_passport
INNER JOIN breed USING(breed_name)
ORDER BY breed.ring_number, dog.dog_number, owner.surname,
owner.name, owner.patronymic;

```

```

SELECT club.club_id, dog.breed_name
FROM club
INNER JOIN club_numbers USING(club_id)
INNER JOIN dog USING(dog_number)
GROUP BY club.club_id, dog.breed_name
ORDER BY club.club_id, dog.breed_name;

```

```

WITH dog_places AS (
SELECT dog.breed_name, dog.dog_number,
ROUND(AVG(dog_expert_estimate.estimate), 2) AS Estimate,
DENSE_RANK() OVER (PARTITION BY dog.breed_name ORDER BY
ROUND(AVG(dog_expert_estimate.estimate), 2) DESC) AS place
FROM dog
INNER JOIN dog_expert_estimate USING(dog_number)
GROUP BY dog.breed_name, dog.dog_number
ORDER BY dog.breed_name, Estimate DESC
)
SELECT club.club_id, club.club_name, dog_places.place,
COUNT(dog_places.place)
FROM club
INNER JOIN club_numbers USING(club_id)

```

```
        INNER JOIN dog_places USING(dog_number)
GROUP BY club.club_id, club.club_name, dog_places.place
ORDER BY club.club_id, dog_places.place;
```

```
        SELECT      dog.breed_name,      expert.expert_id,      expert.name,
expert.surname
        FROM dog
        INNER JOIN dog_expert_estimate USING(dog_number)
        INNER JOIN expert USING(expert_id)
GROUP BY dog.breed_name, expert.expert_id
ORDER BY dog.breed_name;
```

```
        SELECT dog.breed_name, COUNT(*)
        FROM dog
GROUP BY dog.breed_name
ORDER BY dog.breed_name;
```

## **ПРИЛОЖЕНИЕ Б**

### **ССЫЛКИ**

<https://www.db-fiddle.com/f/sV8f9F7Lg6R6HE89nXNhhV/9>