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 лабораторный 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 для проверки работоспособности
 - Исходный код в виде .sql файла запушить в виде PR в репо
 - В отчете описать:
 - о Цель
 - о Текст задания в соответствии с вариантом
- о Скриншоты работы с СУБД PostgreSQL (psql / DBeaver / Datagrip, ...)
- о Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
 - о Исходный код в приложении
 - о Ссылку на исходный код www.db-fiddle.com в приложении
 - о Ссылка на PR в приложении
 - о 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-Oubuntu0.23.04.1)
```

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. Hастройка 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 0.0.0.0/0 scram-sha-256
```

Рис 3. Настройка pg_hba.conf

2. Затем СУБД была запущена.

```
ialex@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 база данных dog show командой createdb dog show.

```
postgres=# \list

List of databases

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 relati	ions
Schema Name	Type Owner
	+
public breed	table postgres
public club	table postgres
<pre>public club_numbers</pre>	table postgres
public dog	table postgres
<pre>public dog_expert_estimate</pre>	table postgres
public expert	table postgres
public owner	table postgres
public ring	table postgres
(8 rows)	

Рис 6. Создание таблиц.

```
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"
                                           | Collation | Nullable |
                                                                                           Default
   Column
                           Туре
expert_id
                                                          not null | nextval('expert_expert_id_seq'::regclass)
                integer
surname
                character varying(255)
                character varying(255)
name
                                                                       nextval('expert_club_id_seq'::regclass)
club_id
                integer
                                                          not null
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)
    TABLE "dog_expert_estimate" CONSTRAINT "dog_expert_estimate_expert_id_fkey" FOREIGN KEY (expert_id) REFER
ENCES expert(expert_id)
```

Рис 9. Структура таблицы expert.

```
dog show=# \d breed
                                            Table "public.breed"
  Column
                        Туре
                                       | 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
                                                    | Collation | Nullable |
          Column
                                     Type
                                                                                              Default
                                                                | not null | nextval('dog_dog_number_seq'::reg
dog number
                          | integer
class)
owner_passport
                            character varying(11)
breed_name
                            character varying(255)
pedigree_document_number
                                                                  not null
                            integer
mother_nickname
                            character varying(255)
father_nickname
                            character varying(255)
nickname
                            character varying(255)
                            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(do
g number)
   TABLE "dog_expert_estimate" CONSTRAINT "dog_expert_estimate_dog_number_fkey" FOREIGN KEY (dog_number) REF
ERENCES dog(dog_number)
```

Рис 11. Структура таблицы dog.

```
dog_show=# \d dog_expert_estimate
                                      Table "public.dog_expert_estimate"
                        | Collation | Nullable |
   Column
             | Type
                                                                               Default
                                                    nextval('dog_expert_estimate_dog_number_seq'::regclass)
nextval('dog_expert_estimate_expert_id_seq'::regclass)
 dog number | integer
                                        not null |
 expert_id
               integer
                                        not null
 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"
                                                                    Default
   Column
                      | Collation | Nullable |
               Type
                                   | not null | nextval('club_numbers_dog_number_seq'::regclass)
 dog_number | integer |
                                   | not null | nextval('club numbers club id seg'::regclass)
 club id
            | integer |
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)"
```

Puc 13. Структура таблицы club_numbers.

```
dog_show=# \d owner
                        Table "public.owner"
  Column
                                    | Collation | Nullable | Default
passport
            | character varying(11)
                                                   not null |
surname
            | character varying(255)
             character varying(255)
name
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.

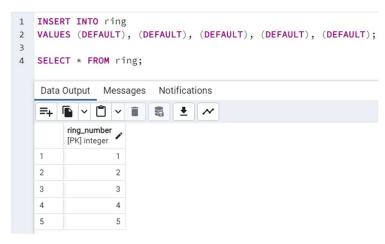


Рис 15. Заполнение таблицы ring.

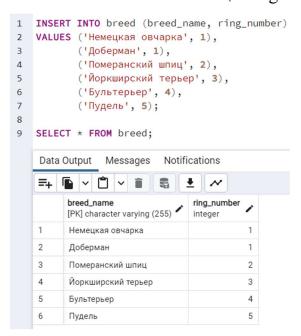


Рис 16. Заполнение таблицы breed.

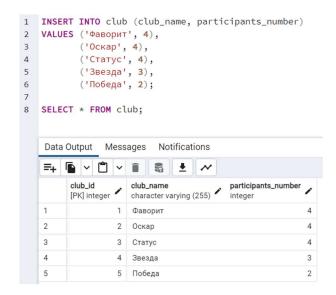


Рис 17. Заполнение таблицы club.

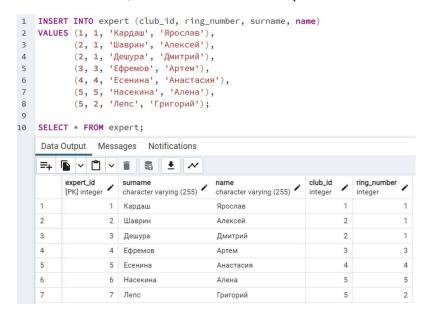


Рис 18. Заполнение таблицы expert.

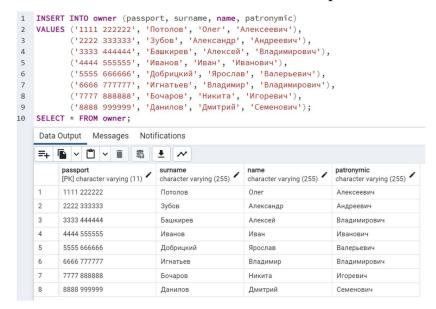


Рис 19. Заполнение таблицы owner.

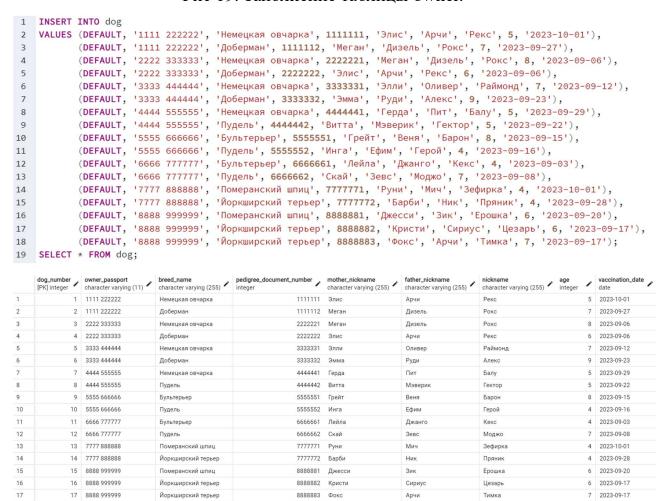


Рис 20. Заполнение dog.

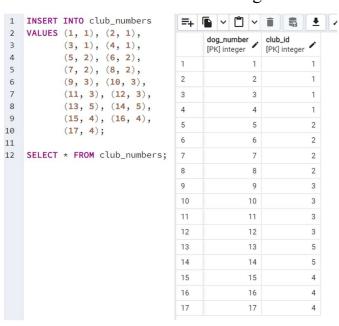


Рис 21. Заполнение club_numbers.

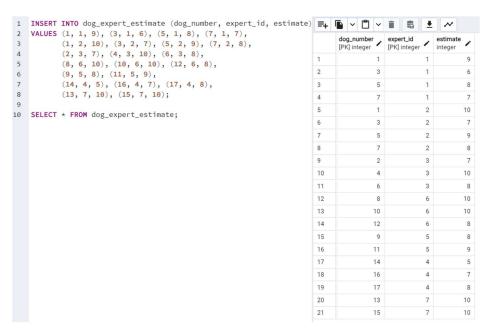


Рис 21. Заполнение dog expert estimate.

- 6. Затем были реализованы запросы, отвечающие на вопросы.
- На каком ринге выступает заданный хозяин со своей собакой?

```
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;
```

	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?

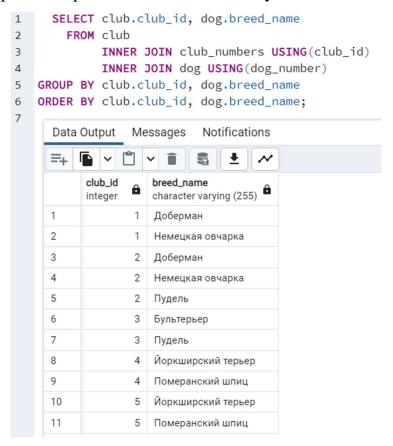


Рис 22. Запрос №2.

Какие медали и сколько заслужены клубом?

```
1 WITH dog_places AS (
       2
            DENSE_RANK() OVER (PARTITION BY dog.breed_name ORDER BY ROUND(AVG(dog_expert_estimate.estimate), 2) DESC) AS place
3
4
         FROM dog
            INNER JOIN dog_expert_estimate USING(dog_number)
5
 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)
           INNER JOIN dog_places USING(dog_number)
12
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	â
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?

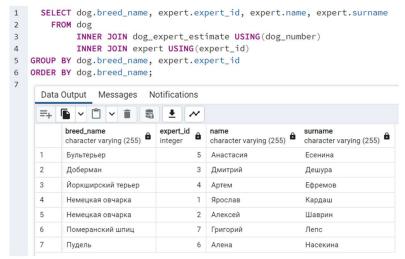


Рис 24. Запрос №4.

• Количество участников по каждой породе?

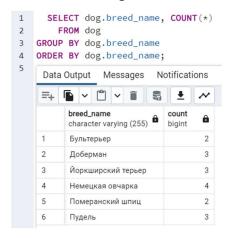


Рис 25. Запрос №5.

Разработанный программный код см. в приложении А.

Ссылки на PR и на исходный код 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);
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 ('ΦαΒορиτ', 4),
       ('Ockap', 4),
       ('CTaTyc', 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', 'Игнатьев', 'Владимир', 'Владимирович'),
            ('7777 888888', 'Бочаров', 'Никита', 'Игоревич'),
            ('8888 999999', 'Данилов', 'Дмитрий', 'Семенович');
     INSERT INTO dog
     VALUES (DEFAULT, '1111 222222', 'Немецкая овчарка', 11111111, 'Элис',
'Арчи', 'Рекс', 5, '2023-10-01'),
                      '1111 222222', 'Доберман', 1111112, 'Меган',
            (DEFAULT,
'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',
'Pekc', 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, 'Грейт',
'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)
                   breed.ring number, dog.dog number, owner.surname,
            BY
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)
```

```
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

ORDER BY dog.breed_name;
```

INNER JOIN dog places USING(dog number)

приложение б

ССЫЛКИ

https://www.db-fiddle.com/f/sV8f9F7Lg6R6HE89nXNhhV/9
https://github.com/moevm/sq1-2023-1304/pull/23