

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ  
Факультет компьютерных систем и сетей  
Кафедра электронных вычислительных машин

### Лабораторная работа №6

Реализация SQL-запросов на выборку данных с использованием подзапросов,  
агрегатных функций, группировки и операций над множествами

Студент: А.В. Губаревич  
Преподаватель: С.С. Силич

МИНСК 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 COUNT .....	<b>Ошибка! Закладка не определена.</b>
2 COUNT (DISTINCT) .....	5
3 MAX.....	6
4 MAX + JOIN.....	7
5 SUM .....	8
6 SUM (DISTINCT) .....	9
7 GROUP BY.....	10
8 GROUP BY (HAVING) .....	11
9 HAVING (AVG).....	12
10 UNION (NOT) .....	13
11 UNION (ANY).....	15
12 INTERSECT .....	17
13 INTERSECT (WHERE) .....	18
14 EXCEPT (EXISTS).....	20
15 EXCEPT (ALL) .....	21
ЗАКЛЮЧЕНИЕ .....	22

## ВВЕДЕНИЕ

В лабораторной работе выполняется создание запросов на выборку данных на языке SQL с использованием подзапросов, агрегатных функций, а также группировки данных (предложение GROUP BY оператора SELECT) и операций над множествами (UNION, INTERSECT, MINUS).

Правила выполнения заданий:

- для каждого задания создать реализацию в виде одного оператора SQL **SELECT**, в котором можно использовать подзапросы и группировку данных;
- обратить внимание, что использование скалярных (особенно соотнесенных !) подзапросов в предложении **SELECT** следует ограничить, т.к. они ухудшают производительность и анализ запроса, поэтому, если запрос затрагивает несколько таблиц, то сначала надо собрать данные с помощью соединения данных таблиц, и только потом выполнять их обработку (например, группировать);
- перед запуском запроса на выполнение, изучить данные в используемых запросом таблицах, и если требуется добавить новые данные, чтобы результат выборки не был пустым;
- выполнить запрос и проанализировать его результат – если есть расхождения между изученными данными и результатом запроса, то есть повод задуматься о проверке правильности выполнения этого задания.

Оформить отчет.

## 1 COUNT

Вывести количество животных, за которыми закреплены волонтеры.

	animal_id bigint	lock	breed character varying (30)	lock	id_volunteer bigint	lock	volunteer_name character varying (50)	lock
1	1		Британская короткошерстная		1		Петров Алексей Иванович	
2	2		Немецкая овчарка		1		Петров Алексей Иванович	
3	3		Сиамская		2		Сидорова Мария Петровна	
4	4		Дворняга		2		Сидорова Мария Петровна	
5	5		Волнистый попугай		3		Кузнецов Дмитрий Сергеевич	
6	6		Хомяк сирийский		3		Кузнецов Дмитрий Сергеевич	
7	7		Мейн-кун		4		Николаева Елена Викторовна	
8	8		Лабрадор		4		Николаева Елена Викторовна	
9	9		Неразлучники		5		Васнецов Игорь Александрович	
10	10		Шиншилла		5		Васнецов Игорь Александрович	

Рисунок 1.1 – Исходная таблица

```
SELECT COUNT(*) AS total_animals_with_volunteers
FROM "Animal" a
JOIN "Animal_volunteer" av ON a.id = av.id_animal;
```

	total_animals_with_volunteers bigint	lock
1	10	10

Рисунок 1.2 – Результат выполнения

## 2 COUNT (DISTINCT)

Вывести количество уникальных пород животных, находящихся в вольерах статуса "Active".

	animal_id bigint	breed character varying (30)	aviary_status character varying (15)	aviary_type character varying (30)
1	1	Британская короткошерстная	Active	Outdoor
2	5	Волнистый попугай	Active	Mixed
3	7	Мейн-кун	Active	Outdoor
4	9	Неразлучники	Active	Outdoor
5	3	Сиамская	Active	Outdoor
6	10	Шиншилла	Active	Indoor
7	4	Дворняга	Cleaning	Indoor
8	2	Немецкая овчарка	Maintenance	Indoor
9	6	Хомяк сирийский	Quarantine	Isolation
10	8	Лабрадор	Renovation	Indoor

Рисунок 2.1 – Исходная таблица

```
SELECT COUNT(DISTINCT a.breed) AS  
unique_breeds_in_active_aviaries  
  
FROM "Animal" a  
  
JOIN "Aviary" av ON a.id_aviary = av.id  
  
WHERE av.status = 'Active';
```

	unique_breeds_in_active_aviaries bigint
1	6

Рисунок 2.2 – Результат выполнения

### 3 MAX

Вывести максимальную площадь среди всех вольеров типа "Outdoor".

	<b>id</b> [PK] bigint	<b>square</b> numeric	<b>status</b> character varying (15)	<b>type</b> character varying (30)	<b>location</b> character varying (30)
1	10	16.5	Active	Indoor	South Wing
2	2	15.0	Maintenance	Indoor	South Wing
3	8	14.0	Renovation	Indoor	North Wing
4	4	12.5	Cleaning	Indoor	West Wing
5	6	18.0	Quarantine	Isolation	Medical Wing
6	5	20.0	Active	Mixed	Central Area
7	3	30.0	Active	Outdoor	East Wing
8	9	28.0	Active	Outdoor	Playground
9	1	25.5	Active	Outdoor	North Wing
10	7	22.0	Active	Outdoor	Garden Area

Рисунок 3.1 – Исходная таблица

```
SELECT MAX(square) AS max_outdoor_square
FROM "Aviary"
WHERE type = 'Outdoor';
```

	<b>max_outdoor_square</b> numeric
1	30.0

Рисунок 3.2 – Результат выполнения

## 4 MAX + JOIN

Вывести самую высокую стоимость медицинской процедуры, проведенной животному, и вывести породу этого животного.

	<b>id</b> [PK] bigint	<b>cost</b> integer	<b>amount</b> integer	<b>reason</b> character varying (50)	<b>name</b> character varying (30)
1	6	5000	1	Хирургическая операц...	Перелом лапы
2	9	3500	1	Стоматология	Чистка зубов
3	2	3000	1	Вакцинация	Прививка от бешенства
4	4	2800	1	Стерилизация	Стерилизация
5	3	2500	1	Стерилизация	Кастрация
6	8	2000	1	Лечение	Лечение ушной инфекции
7	10	1800	1	Диагностика	УЗИ брюшной полости
8	1	1500	1	Ежегодный осмотр	Общий осмотр
9	5	1200	2	Обработка от паразитов	Обработка от блох
10	7	800	1	Диагностика	Анализ крови

Рисунок 4.1 – Исходная таблица

```
SELECT a.breed, mp.cost AS max_procedure_cost
FROM "Animal" a
JOIN "Animal_medical_procedure" amp ON a.id = amp.id_animal
JOIN "Medical_procedure" mp ON amp.id_medical = mp.id
WHERE mp.cost = (SELECT MAX(cost) FROM
"Medical_procedure");
```

	<b>breed</b> character varying (30)	<b>max_procedure_cost</b> integer
1	Хомяк сирийский	5000

Рисунок 4.2 – Результат выполнения

## 5 SUM

Вывести общее количество корма (в единицах), поставленного всеми поставщиками.

	<b>id</b> [PK] bigint	<b>quantity</b> integer	<b>delivery_date</b> date	<b>the_supplier</b> character varying (30)	<b>type_of_feed</b> character varying (30)
1	1	100	2024-01-15	ЗооМир	Сухой корм для собак
2	7	90	2024-01-21	ЗооМир	Сухой корм для кошек
3	3	75	2024-01-17	АгроКорм	Зерновой корм для птиц
4	5	60	2024-01-19	Фермер	Сено для кроликов
5	10	55	2024-01-24	ЗооЛюкс	Влажный корм для собак
6	2	50	2024-01-16	ПетШоп	Влажный корм для кошек
7	6	40	2024-01-20	АкваМир	Корм для рыб
8	9	35	2024-01-23	АгроКорм	Корм для рептилий
9	4	30	2024-01-18	ЗооЛюкс	Корм для грызунов
10	8	25	2024-01-22	ПетШоп	Лакомства для собак

Рисунок 5.1 – Исходная таблица

```
SELECT SUM(quantity) AS total_feed_supply  
FROM "Feed_supply";
```

	<b>total_feed_supply</b> bigint
1	560

Рисунок 5.2 – Результат выполнения

## 6 SUM (DISTINCT)

Вывести сумму уникальных значений стоимости медицинских процедур.

	<b>id</b> [PK] bigint	<b>cost</b> integer	<b>amount</b> integer	<b>reason</b> character varying (50)	<b>name</b> character varying (30)
1	7	800	1	Диагностика	Анализ крови
2	5	1200	2	Обработка от паразитов	Обработка от блох
3	1	1500	1	Ежегодный осмотр	Общий осмотр
4	10	1800	1	Диагностика	УЗИ брюшной полости
5	8	2000	1	Лечение	Лечение ушной инфекции
6	3	2500	1	Стерилизация	Кастрация
7	4	2800	1	Стерилизация	Стерилизация
8	2	3000	1	Вакцинация	Прививка от бешенства
9	9	3500	1	Стоматология	Чистка зубов
10	6	5000	1	Хирургическая операция	Перелом лапы

Рисунок 6.1 – Исходная таблица

```
SELECT SUM(DISTINCT cost) AS sum_unique_costs  
FROM "Medical_procedure";
```

	<b>sum_unique_costs</b> bigint
1	24100

Рисунок 6.2 – Результат выполнения

## 7 GROUP BY

Вывести животных по типу и для каждого типа посчитать количество особей.

	<b>id</b> [PK] bigint	<b>breed</b> character varying (30)	<b>date_of_receipt</b> date	<b>state_of_health</b> character varying (30)	<b>type</b> character varying (30)	<b>id_aviary</b> bigint	<b>id_employee</b> bigint
1	6	Хомяк сирийский	2024-01-20	Здоров	Грызун	6	6
2	10	Шиншилла	2024-01-24	Лечение	Грызун	10	10
3	1	Британская короткошерстная	2024-01-15	Здоров	Кошка	1	1
4	7	Мейн-кун	2024-01-21	Карантин	Кошка	7	7
5	3	Сиамская	2024-01-17	Лечение	Кошка	3	3
6	5	Волнистый попугай	2024-01-19	Здоров	Птица	5	5
7	9	Неразлучники	2024-01-23	Здоров	Птица	9	9
8	4	Дворняга	2024-01-18	Здоров	Собака	4	4
9	8	Лабрадор	2024-01-22	Здоров	Собака	8	8
10	2	Немецкая овчарка	2024-01-16	Здоров	Собака	2	2

Рисунок 7.1 – Исходная таблица

```
SELECT type, COUNT(*) AS animal_count
FROM "Animal"
GROUP BY type;
```

	<b>type</b> character varying (30)	<b>animal_count</b> bigint
1	Птица	2
2	Кошка	3
3	Собака	3
4	Грызун	2

Рисунок 7.2 – Результат выполнения

## 8 GROUP BY (HAVING)

Вывести типы вольеров, средняя площадь которых больше 20 м<sup>2</sup>.

	<b>id</b> [PK] bigint	<b>square</b> numeric	<b>status</b> character varying (15)	<b>type</b> character varying (30)	<b>location</b> character varying (30)
1	10	16.5	Active	Indoor	South Wing
2	2	15.0	Maintenance	Indoor	South Wing
3	8	14.0	Renovation	Indoor	North Wing
4	4	12.5	Cleaning	Indoor	West Wing
5	6	18.0	Quarantine	Isolation	Medical Wing
6	5	20.0	Active	Mixed	Central Area
7	3	30.0	Active	Outdoor	East Wing
8	9	28.0	Active	Outdoor	Playground
9	1	25.5	Active	Outdoor	North Wing
10	7	22.0	Active	Outdoor	Garden Area

Рисунок 8.1 – Исходные данные

```
SELECT type, ROUND(AVG(square), 3) AS avg_square
FROM "Aviary"
GROUP BY type
HAVING AVG(square) > 20;
```

	<b>type</b> character varying (30)	<b>avg_square</b> numeric
1	Outdoor	26.375

Рисунок 8.2 – Результат выполнения

## 9 HAVING (AVG)

Вывести сотрудников, средняя стоимость медицинских процедур для животных, за которыми они закреплены, превышает 2000 руб.

	<b>id</b> [PK] bigint	<b>cost</b> integer	<b>amount</b> integer	<b>reason</b> character varying (50)	<b>name</b> character varying (30)
1	6	5000	1	Хирургическая операц...	Перелом лапы
2	9	3500	1	Стоматология	Чистка зубов
3	2	3000	1	Вакцинация	Прививка от бешенства
4	4	2800	1	Стерилизация	Стерилизация
5	3	2500	1	Стерилизация	Кастрация
6	8	2000	1	Лечение	Лечение ушной инфекции
7	10	1800	1	Диагностика	УЗИ брюшной полости
8	1	1500	1	Ежегодный осмотр	Общий осмотр
9	5	1200	2	Обработка от паразитов	Обработка от блох
10	7	800	1	Диагностика	Анализ крови

Рисунок 9.1 – Исходные данные

```
SELECT e.snp, AVG(mp.cost)::integer AS avg_procedure_cost
FROM "Employee" e
JOIN "Animal" a ON e.id = a.id_employee
JOIN "Animal_medical_procedure" amp ON a.id = amp.id_animal
JOIN "Medical_procedure" mp ON amp.id_medical = mp.id
GROUP BY e.id, e.snp
HAVING AVG(mp.cost) > 2000;
```

	<b>snp</b> character varying (50)	<b>avg_procedure_cost</b> integer
1	Белов Денис Сергеевич	2800
2	Громова Екатерина Александровна	2250
3	Романова Наталья Игоревна	5000
4	Волков Андрей Николаевич	2250
5	Ершов Виктор Дмитриевич	3500
6	Соколов Павел Михайлович	2500

Рисунок 9.2 – Результат выполнения

## 10 UNION (NOT)

Вывести волонтеров, которые не занимаются выгулом или не занимаются уборкой

	<b>id</b> [PK] bigint	<b>snp</b> character varying (50)	<b>telephone</b> character varying (20)	<b>email</b> character varying (100)	<b>duty</b> character varying (50)
1	1	Петров Алексей Иванович	+79161234567	petrov@example.com	Выгул собак
2	3	Кузнецов Дмитрий Сергеевич	+79167778899	kuznetsov@example.com	Кормление животных
3	8	Зайцева Ольга Дмитриевна	+79168889900	zayceva@example.com	Обучение волонтеров
4	6	Орлова Анна Михайловна	+79163332211	orlova@example.com	Помощь ветеринару
5	10	Смирнова Татьяна Владимировна	+79161112233	smirnova@example.com	Работа с документами
6	5	Васнецов Игорь Александрович	+79162223344	vasnecov@example.com	Социализация животных
7	7	Лебедев Сергей Николаевич	+79164445566	lebedev@example.com	Транспортировка
8	4	Николаева Елена Викторовна	+79165554433	nikolaeva@example.com	Уборка вольеров
9	2	Сидорова Мария Петровна	+79169876543	sidorova@example.com	Уход за кошками
10	9	Федоров Максим Игоревич	+79169990011	fedorov@example.com	Фандрайзинг

Рисунок 10.1 – Исходная таблица

```
SELECT snp, duty, 'Не выгуливают собак' as reason
FROM "Volunteer"
WHERE duty != 'Выгул собак'
```

UNION

```
SELECT snp, duty, 'Не убирают вольеры' as reason
FROM "Volunteer"
WHERE duty != 'Уборка вольеров'
ORDER BY reason, duty, snp;
```

	snp character varying (50) 	duty character varying (50) 	reason text 
1	Кузнецов Дмитрий Сергеевич	Кормление животных	Не выгуливают собак
2	Зайцева Ольга Дмитриевна	Обучение волонтеров	Не выгуливают собак
3	Орлова Анна Михайловна	Помощь ветеринару	Не выгуливают собак
4	Смирнова Татьяна Владимировна	Работа с документами	Не выгуливают собак
5	Васнецов Игорь Александрович	Социализация животных	Не выгуливают собак
6	Лебедев Сергей Николаевич	Транспортировка	Не выгуливают собак
7	Николаева Елена Викторовна	Уборка вольеров	Не выгуливают собак
8	Сидорова Мария Петровна	Уход за кошками	Не выгуливают собак
9	Федоров Максим Игоревич	Фандрайзинг	Не выгуливают собак
10	Петров Алексей Иванович	Выгул собак	Не убирают вольеры
11	Кузнецов Дмитрий Сергеевич	Кормление животных	Не убирают вольеры
12	Зайцева Ольга Дмитриевна	Обучение волонтеров	Не убирают вольеры
13	Орлова Анна Михайловна	Помощь ветеринару	Не убирают вольеры
14	Смирнова Татьяна Владимировна	Работа с документами	Не убирают вольеры
15	Васнецов Игорь Александрович	Социализация животных	Не убирают вольеры
16	Лебедев Сергей Николаевич	Транспортировка	Не убирают вольеры
17	Сидорова Мария Петровна	Уход за кошками	Не убирают вольеры
18	Федоров Максим Игоревич	Фандрайзинг	Не убирают вольеры

Рисунок 10.2 – Результат выполнения

## 11 UNION (ANY)

Вывести животных, которые либо прошли процедуру "Общий осмотр", либо находятся в вольере типа "Outdoor".

	<b>id</b> bigint	<b>breed</b> character varying (30)	<b>aviary_type</b> character varying (30)
1	1	Британская короткошерстная	Outdoor
2	2	Немецкая овчарка	Indoor
3	3	Сиамская	Outdoor
4	4	Дворняга	Indoor
5	5	Волнистый попугай	Mixed
6	6	Хомяк сирийский	Isolation
7	7	Мейн-кун	Outdoor
8	8	Лабрадор	Indoor
9	9	Неразлучники	Outdoor
10	10	Шиншилла	Indoor

Рисунок 11.1 – Исходная таблица

```
SELECT a.id, a.breed, 'Прошел общий осмотр' as reason  
FROM "Animal" a  
JOIN "Animal_medical_procedure" amp ON a.id = amp.id_animal  
JOIN "Medical_procedure" mp ON amp.id_medical = mp.id  
WHERE mp.name = 'Общий осмотр'
```

UNION

```
SELECT a.id, a.breed, 'В Outdoor вольере' as reason  
FROM "Animal" a  
JOIN "Aviary" av ON a.id_aviary = av.id  
WHERE av.type = 'Outdoor'  
ORDER BY id;
```

	<b>id</b> bigint	<b>breed</b> character varying (30)	<b>reason</b> text
1	1	Британская короткошерстная	B Outdoor вольере
2	1	Британская короткошерстная	Прошел общий осмотр
3	2	Немецкая овчарка	Прошел общий осмотр
4	3	Сиамская	B Outdoor вольере
5	4	Дворняга	Прошел общий осмотр
6	7	Мейн-кун	B Outdoor вольере
7	9	Неразлучники	B Outdoor вольере

Рисунок 11.2 – Результат выполнения

## 12 INTERSECT

Показать животных, которые живут в активных вольерах и при этом здоровы.

	<b>id</b> bigint	<b>breed</b> character varying (30)	<b>state_of_health</b> character varying (30)	<b>status</b> character varying (15)
1	1	Британская короткошерстная	Здоров	Active
2	2	Немецкая овчарка	Здоров	Maintenance
3	3	Сиамская	Лечение	Active
4	4	Дворняга	Здоров	Cleaning
5	5	Волнистый попугай	Здоров	Active
6	6	Хомяк сирийский	Здоров	Quarantine
7	7	Мейн-кун	Карантин	Active
8	8	Лабрадор	Здоров	Renovation
9	9	Неразлучники	Здоров	Active
10	10	Шиншилла	Лечение	Active

Рисунок 12.1 – Исходные данные

```
SELECT a.id, a.breed, a.type, a.state_of_health, av.status  
AS aviary_status  
FROM "Animal" a  
JOIN "Aviary" av ON a.id_aviary = av.id  
WHERE av.status = 'Active'
```

INTERSECT

```
SELECT a.id, a.breed, a.type, a.state_of_health, av.status  
AS aviary_status  
FROM "Animal" a  
JOIN "Aviary" av ON a.id_aviary = av.id  
WHERE a.state_of_health = 'Здоров';
```

	<b>id</b> bigint	<b>breed</b> character varying (30)	<b>type</b> character varying (30)	<b>state_of_health</b> character varying (30)	<b>aviary_status</b> character varying (15)
1	5	Волнистый попугай	Птица	Здоров	Active
2	1	Британская короткошерстная	Кошка	Здоров	Active
3	9	Неразлучники	Птица	Здоров	Active

Рисунок 12.2 – Результат выполнения

## 13 INTERSECT (WHERE)

Вывести животных, которые прошли и дешевые (<2000 руб) и дорогие (>2000 руб) медицинские процедуры.

	<b>id</b> bigint	<b>breed</b> character varying (30)	<b>procedure_name</b> character varying (30)	<b>cost</b> integer
1	1	Британская короткошерстная	Общий осмотр	1500
2	1	Британская короткошерстная	Общий осмотр	2200
3	1	Британская короткошерстная	Прививка от бешенства	3000
4	2	Немецкая овчарка	Обработка от блох	1200
5	2	Немецкая овчарка	Общий осмотр	1500
6	3	Сиамская	Кастрация	2500
7	4	Дворняга	Общий осмотр	1500
8	4	Дворняга	Прививка от бешенства	3000
9	5	Волнистый попугай	Стерилизация	2800
10	6	Хомяк сирийский	Перелом лапы	5000
11	7	Мейн-кун	Анализ крови	800
12	8	Лабрадор	Лечение ушной инфекции	2000
13	9	Неразлучники	Чистка зубов	3500
14	10	Шиншилла	УЗИ брюшной полости	1800

Рисунок 13.1 – Исходная таблица

```
SELECT a.id, a.breed
FROM "Animal" a
JOIN "Animal_medical_procedure" amp ON a.id = amp.id_animal
JOIN "Medical_procedure" mp ON amp.id_medical = mp.id
WHERE mp.cost < 2000
```

INTERSECT

```
SELECT a.id, a.breed
FROM "Animal" a
JOIN "Animal_medical_procedure" amp ON a.id = amp.id_animal
JOIN "Medical_procedure" mp ON amp.id_medical = mp.id
WHERE mp.cost > 2000
ORDER BY id;
```

	<b>id</b> bigint 	<b>breed</b> character varying (30) 
1	1	Британская короткошерстная
2	4	Дворняга

Рисунок 13.2 – Результат выполнения

## 14 EXCEPT (EXISTS)

Вывести поставки корма, которые не используются ни одним животным.

	<b>id</b> [PK] bigint	<b>type_of_feed</b> character varying (30)	<b>quantity</b> integer	<b>the_supplier</b> character varying (30)
1	1	Сухой корм для собак	100	ЗооМир
2	2	Влажный корм для ко...	50	ПетШоп
3	3	Зерновой корм для птиц	75	АгроКорм
4	4	Корм для грызунов	30	ЗооЛюкс
5	5	Сено для кроликов	60	Фермер
6	6	Корм для рыб	40	АкваМир
7	7	Сухой корм для кошек	90	ЗооМир
8	8	Лакомства для собак	25	ПетШоп

Рисунок 14.1 – Исходная таблица

```
SELECT id, type_of_feed, quantity, the_supplier  
FROM "Feed_supply"
```

```
EXCEPT
```

```
-- Поставки корма, которые используются животными  
SELECT fs.id, fs.type_of_feed, fs.quantity, fs.the_supplier  
FROM "Feed_supply" fs  
WHERE EXISTS (  
    SELECT 1  
    FROM "Animal_feed_supply" afs  
    WHERE afs.id_feed_supply = fs.id  
) ;
```

	<b>id</b> bigint	<b>type_of_feed</b> character varying (30)	<b>quantity</b> integer	<b>the_supplier</b> character varying (30)
1	10	Влажный корм для собак	55	ЗооЛюкс
2	9	Корм для рептилий	35	АгроКорм

Рисунок 14.2 – Результат выполнения

## 15 EXCEPT (ALL)

Вывести найти вольеры, в которых нет животных.

	<b>id</b> [PK] bigint	<b>type</b> character varying (30)	<b>status</b> character varying (15)	<b>location</b> character varying (30)
1	1	Outdoor	Active	North Wing
2	2	Indoor	Maintenance	South Wing
3	3	Outdoor	Active	East Wing
4	4	Indoor	Cleaning	West Wing
5	5	Mixed	Active	Central Area
6	6	Isolation	Quarantine	Medical Wing
7	7	Outdoor	Active	Garden Area
8	8	Indoor	Renovation	North Wing
9	9	Outdoor	Active	Playground
10	10	Indoor	Active	South Wing

Рисунок 15.1 – Исходная таблица

```
SELECT id, type, status, location  
FROM "Aviary"
```

EXCEPT

```
SELECT av.id, av.type, av.status, av.location  
FROM "Aviary" av  
JOIN "Animal" a ON av.id = a.id_aviary;
```

	<b>id</b> bigint	<b>type</b> character varying (30)	<b>status</b> character varying (15)	<b>location</b> character varying (30)
1	11	Isolation	Active	Карантинная зона

Рисунок 15.2 – Результат выполнения

## **ЗАКЛЮЧЕНИЕ**

В ходе данной лабораторной работы были изучены дополнительные возможности SQL-запросов в Postgres на базе «PgAdmin 4».

Изучено формирование SQL-операторов для выборки данных из таблиц с использованием многотабличных запросов, подзапросов и секций WHERE, ORDER BY и других.