

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

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

Студент гр. 1303

\_\_\_\_\_

Депрейс. А.С.

Преподаватель

\_\_\_\_\_

Заславский М.М.

Санкт-Петербург

2023

### **Цель работы.**

Локально развернуть PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы в задании.

### **Задание.**

#### Вариант 7

Пусть требуется создать программную систему, ориентированную на администрирование птицефабрики и позволяющую работать с информацией о работниках фабрики и об имеющихся на ней курах. О каждой курице должна храниться следующая информация: вес, возраст, порода, количество ежемесячно получаемых от курицы яиц, а также информация о местонахождении курицы. Сведения о породе включают в себя: название породы, среднее количество яиц в месяц (производительность) и средний вес, номер рекомендованной диеты. Птицефабрика имеет несколько цехов, и за каждой курицей закреплена отдельная клетка. Код клетки, где находится курица, характеризуется номером цеха, номером ряда в цехе и номером клетки в ряду. О работниках птицефабрики в БД должна храниться следующая информация: паспортные данные, зарплата, закрепленные за работником клетки. Директор птицефабрики может принять или уволить работника, при этом не должно быть кур, не обслуживаемых ни одним работником. Количество кур может изменяться как в большую, так и в меньшую сторону, в отдельные моменты времени часть клеток может пустовать. Директору могут потребоваться следующие сведения:

- Какое количество яиц получают от каждой курицы данного веса, породы, возраста?
- В каком цехе наибольшее количество кур определенной породы?
- В каких клетках находятся куры указанного возраста с заданным номером диеты?
- Сколько яиц в день приносят куры указанного работника?

- Среднее количество яиц, которое получает в день каждый работник от обслуживаемых им кур?
- В каком цехе находится курица, от которой получают больше всего яиц.
- Сколько кур каждой породы в каждом цехе?
- Какое количество кур обслуживает каждый работник?

### **Выполнение работы.**

Созданы таблицы: «клетка», «курица», «работник», «порода», «наблюдает\_за\_курицей\_в\_клетке» в соответствии со структурой БД и написаны запросы на их заполнение.

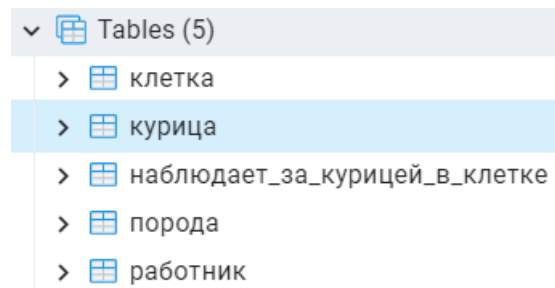


Рисунок 1. Созданные таблицы.

Написаны запросы на вывод нужных сведений для администрации птицефабрики:

Query Query History

```

1 SELECT курица_id, количество_яиц_в_месяц FROM курица
2 WHERE
3     вес < 1.95 + 0.0001 AND
4     вес > 1.95 - 0.0001 AND
5     возраст = 1 AND
6     название_породы = 'Русская белая';

```

Data Output Messages Notifications

	курица_id [PK] integer	количество_яиц_в_месяц integer
1	9	18
2	10	21

Рисунок 2. – Какое количество яиц получают от каждой курицы данного веса, породы, возраста.

Query Query History

```

1 SELECT номер_цеха FROM клетка
2     INNER JOIN наблюдает_за_курицей_в_клетке USING(клетка_id)
3     INNER JOIN курица USING(курица_id)
4 WHERE
5     название_породы = 'Орловские куры'
6 GROUP BY номер_цеха
7 ORDER BY COUNT(номер_цеха) DESC
8 LIMIT 1

```

Data Output Messages Notifications

	номер_цеха integer
1	2

Рисунок 3. – В каком цехе наибольшее количество кур определенной породы.

Query

Query History

1

SELECT
клетка\_id, номер\_цеха, номер\_ряда, номер\_клетки
FROM
клетка
INNER JOIN
наблюдает\_за\_курицей\_в\_клетке
USING(клетка\_id)
INNER JOIN
курица
USING(курица\_id)
INNER JOIN
порода
USING(название\_породы)
WHERE
номер\_рекомендованной\_диеты = 1
AND
возраст = 2

2

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗄

⬇

📈

	клетка_id [PK] integer	номер_цеха integer	номер_ряда integer	номер_клетки integer
1	8	2	3	2
2	7	2	3	1

Рисунок 4. - В каких клетках находятся куры указанного возраста с заданным номером диеты

Query	Query History
1	<b>SELECT</b> (SUM(количество_яиц_в_месяц) / 30.44) <b>AS</b> яиц_в_день <b>FROM</b> работник
2	<b>INNER JOIN</b> наблюдает_за_курицей_в_клетке <b>USING</b> (работник_id)
3	<b>INNER JOIN</b> курица <b>USING</b> (курица_id)
4	<b>WHERE</b>
5	работник_id = 2
6	

Data Output	Messages	Notifications
<div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	яиц_в_день numeric	🔒
1	1.5440210249671485	

Рисунок 5. – Сколько яиц в день приносят куры указанного работника

Query

Query History

1

SELECT работник\_id, (SUM(среднее\_количество\_яиц\_в\_месяц) / 30.44) AS среднее\_яиц\_в\_день FROM работник

2

INNER JOIN наблюдает\_за\_курицей\_в\_клетке USING(работник\_id)

3

INNER JOIN курица USING(курица\_id)

4

INNER JOIN порода USING(название\_породы)

5

GROUP BY работник\_id

6

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

	работник_id [PK] integer	среднее_яиц_в_день numeric
1	4	0.49277266754270696452
2	6	0.85413929040735873850
3	2	1.6425755584756899
4	3	0.49277266754270696452
5	5	0.42706964520367936925
6	7	0.59132720105124835742
7	1	0.59132720105124835742

Рисунок 6. – Среднее количество яиц, которое получает в день каждый работник от обслуживаемых им кур

Query

Query History

1

SELECT номер\_цеха FROM курица

2

INNER JOIN наблюдает\_за\_курицей\_в\_клетке USING(курица\_id)

3

INNER JOIN клетка USING(клетка\_id)

4

ORDER BY количество\_яиц\_в\_месяц DESC

5

LIMIT 1

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

номер\_цеха

integer

🔒

1

1

Рисунок 7. – В каком цехе находится курица, от которой получают больше всего яиц

Query

Query History

```

1 SELECT цехи.номер_цеха, порода.название_породы, COALESCE(количество,0) AS количество FROM (SELECT DISTINCT номер_цеха FROM клетка) AS цехи
2 CROSS JOIN порода
3 LEFT JOIN (SELECT номер_цеха, COUNT(номер_цеха) AS количество, название_породы FROM курица
4 INNER JOIN наблюдает_за_курицей_в_клетке USING(курица_id)
5 INNER JOIN клетка USING(клетка_id)
6 INNER JOIN порода USING(название_породы)
7 GROUP BY номер_цеха, название_породы
8 ORDER BY номер_цеха, название_породы) AS под ON цехи.номер_цеха = под.номер_цеха
9 AND под.название_породы = порода.название_породы
10 ORDER BY цехи.номер_цеха, порода.название_породы

```

Data Output

Messages

Notifications

номер_цеха integer	название_породы text	количество bigint
1	Адлерская серебристая курица	0
2	Араукан	0
3	Доминант	0
4	Легбар	0
5	Леггорн	0
6	Ломан Браун	1
7	Маран	2
8	Орловские куры	0
9	Русская белая	1
10	Хай Лайн	0
11	Адлерская серебристая курица	2
12	Араукан	0

Рисунок 8. – Сколько кур каждой породы в каждом цехе

Query	Query History																								
1	<code>SELECT</code> работник_id, <code>COUNT</code> (наблюдает_за_курицей_в_клетке_id) <code>AS</code> Количество <code>FROM</code> работник																								
2	<code>LEFT JOIN</code> наблюдает_за_курицей_в_клетке <code>USING</code> (работник_id)																								
3	<code>GROUP BY</code> работник_id																								
4	<code>ORDER BY</code> работник_id																								
5																									
6																									
Data Output	Messages Notifications																								
	<table> <tr> <th>работник_id</th><th>Количество</th></tr> <tr> <td>[PK] integer</td><td>bigint</td></tr> <tr><td>1</td><td>1</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>3</td><td>1</td></tr> <tr><td>4</td><td>1</td></tr> <tr><td>5</td><td>1</td></tr> <tr><td>6</td><td>2</td></tr> <tr><td>7</td><td>1</td></tr> <tr><td>8</td><td>0</td></tr> <tr><td>9</td><td>0</td></tr> <tr><td>10</td><td>0</td></tr> </table>	работник_id	Количество	[PK] integer	bigint	1	1	2	3	3	1	4	1	5	1	6	2	7	1	8	0	9	0	10	0
работник_id	Количество																								
[PK] integer	bigint																								
1	1																								
2	3																								
3	1																								
4	1																								
5	1																								
6	2																								
7	1																								
8	0																								
9	0																								
10	0																								

Рисунок 9. – Какое количество кур обслуживает каждый работник

### **Выводы.**

В ходе выполнения работы был локально развернут PostgreSQL, написаны запросы для создания и заполнения таблиц в соответствии со структурой БД, написаны запросы к БД указанные в задании.



## **ПРИЛОЖЕНИЕ А**

### **ССЫЛКИ**

Pull Request: <https://github.com/moevm/sql-2023-1303/pull/28>

DB Fiddle: <https://www.db-fiddle.com/f/9CACR6wyChUAXC1LZ9kTB4/4>

## ПРИЛОЖЕНИЕ В

### ИСХОДНЫЙ КОД

```
CREATE TABLE работник(  
    работник_id INTEGER PRIMARY KEY,  
    номер_паспорта INTEGER NOT NULL ,  
    серия_паспорта SMALLINT NOT NULL,  
    зарплата INTEGER NOT NULL,  
    фамилия TEXT NOT NULL,  
    имя TEXT NOT NULL,  
    отчество TEXT,  
    UNIQUE(номер_паспорта, серия_паспорта)  
);  
  
INSERT INTO работник(работник_id,  
    серия_паспорта,  
    номер_паспорта,  
    зарплата,  
    фамилия,  
    имя,  
    отчество) VALUES  
    (1, 1234, 345123, 12020, 'Депрейс', 'Александр', 'Сергеевич'),  
    (2, 1232, 740593, 12500, 'Безрукова', 'Софья', 'Ивановна'),  
    (3, 9365, 376945, 12300, 'Мельникова', 'Маргарита', 'Максимовна'),  
    (4, 2754, 745788, 13000, 'Волков', 'Александр', 'Андреевич'),  
    (5, 3437, 845375, 15200, 'Калашников', 'Никита', 'Николаевич'),  
    (6, 9374, 883654, 12023, 'Кондратьев', 'Никита', NULL),  
    (7, 6439, 234666, 12490, 'Спиридонова', 'София', NULL),  
    (8, 4386, 636485, 12850, 'Афанасьева', 'Милана', 'Данииловна'),  
    (9, 3567, 528473, 11500, 'Исаев', 'Георгий', 'Максимович'),  
    (10, 8756, 235747, 12050, 'Овсянников', 'Михаил', 'Серафимович');  
  
CREATE TABLE клетка(  
    клетка_id INTEGER PRIMARY KEY,  
    номер_цеха INTEGER NOT NULL,  
    номер_ряда INTEGER NOT NULL,  
    номер_клетки INTEGER NOT NULL,  
    UNIQUE(номер_цеха, номер_ряда, номер_клетки)  
);
```

```

INSERT INTO клетка(клетка_id,
                    номер_цеха,
                    номер_ряда,
                    номер_клетки) VALUES

(1,1,1,1),
(2,1,2,1),
(3,1,2,2),
(4,1,2,3),
(5,2,1,1),
(6,2,2,1),
(7,2,3,1),
(8,2,3,2),
(9,3,1,1),
(10,3,1,2);

```

```

CREATE TABLE порода(
    название_породы TEXT PRIMARY KEY,
    среднее_количество_яиц_в_месяц INTEGER NOT NULL,
    средний_вес REAL NOT NULL,
    номер_рекомендованной_диеты INTEGER NOT NULL
);

```

```

INSERT INTO порода(название_породы, среднее_количество_яиц_в_месяц,
средний_вес, номер_рекомендованной_диеты) VALUES

('Адлерская серебристая курица', 15, 2.65, 1),
('Леггорн', 16, 1.75, 2),
('Доминант', 25, 2.15, 2),
('Ломан Браун', 26, 1.8, 2),
('Орловские куры', 13, 2.5, 1),
('Хай Лайн', 21, 1.32, 1),
('Маран', 12, 2.9, 3),
('Легбар', 17, 2.35, 3),
('Араукан', 13, 1.6, 3),
('Русская белая', 18, 2.1, 1);

```

```

CREATE TABLE курица(
    курица_id INTEGER PRIMARY KEY,

```

```

        вес REAL NOT NULL,
        возраст INTEGER NOT NULL,
        количество_яиц_в_месяц INTEGER NOT NULL,
        название_породы TEXT,
        FOREIGN KEY(название_породы) REFERENCES порода(название_породы) ON
DELETE SET NULL
);

```

```

INSERT INTO курица(курица_id, вес, возраст, количество_яиц_в_месяц,
название_породы) VALUES
    (1, 2.65, 1, 13, 'Адлерская серебристая курица'),
    (2, 2.33, 3, 14, 'Адлерская серебристая курица'),
    (3, 1.65, 3, 23, 'Ломан Браун'),
    (4, 2.65, 2, 12, 'Орловские куры'),
    (5, 2.13, 5, 15, 'Орловские куры'),
    (6, 2.48, 2, 11, 'Орловские куры'),
    (7, 2.8, 2, 13, 'Маран'),
    (8, 2.9, 2, 11, 'Маран'),
    (9, 1.95, 1, 18, 'Русская белая'),
    (10, 1.95, 1, 21, 'Русская белая');

```

```

CREATE TABLE наблюдает_за_курицей_в_клетке(
    наблюдает_за_курицей_в_клетке_id INTEGER PRIMARY KEY,
    работник_id INTEGER NOT NULL,
    клетка_id INTEGER NOT NULL,
    курица_id INTEGER NOT NULL,
    UNIQUE(работник_id, клетка_id, курица_id),
    FOREIGN KEY(работник_id) REFERENCES работник(работник_id) ON DELETE
CASCADE,
    FOREIGN KEY(клетка_id) REFERENCES клетка(клетка_id) ON DELETE
CASCADE,
    FOREIGN KEY(курица_id) REFERENCES курица(курица_id) ON DELETE
CASCADE
);

```

```

INSERT INTO
наблюдает_за_курицей_в_клетке(наблюдает_за_курицей_в_клетке_id,
                                работник_id,

```

```

        клетка_id,
        курица_id) VALUES

(1,1,1,9),
(2,2,2,3),
(3,2,3,7),
(4,2,4,8),
(5,3,5,1),
(6,4,6,2),
(7,5,7,6),
(8,6,8,4),
(9,6,9,5),
(10,7,10,10);

```

```

SELECT курица_id, количество_яиц_в_месяц FROM курица
WHERE
    вес < 1.95 + 0.0001 AND
    вес > 1.95 - 0.0001 AND
    возраст = 1 AND
    название_породы = 'Русская белая';

```

```

SELECT номер_цеха FROM клетка
    INNER JOIN наблюдает_за_курицей_в_клетке USING(клетка_id)
    INNER JOIN курица USING(курица_id)
WHERE
    название_породы = 'Орловские куры'
GROUP BY номер_цеха
ORDER BY COUNT(номер_цеха) DESC
LIMIT 1;

```

```

SELECT клетка_id, номер_цеха, номер_ряда, номер_клетки FROM клетка
    INNER JOIN наблюдает_за_курицей_в_клетке USING(клетка_id)
    INNER JOIN курица USING(курица_id)
    INNER JOIN порода USING(название_породы)
WHERE

```

```

номер_рекомендованной_диеты = 1 AND
возраст = 2;

```

```

SELECT (SUM(количество_яиц_в_месяц) / 30.44) AS яиц_в_день FROM работник
INNER JOIN наблюдает_за_курицей_в_клетке USING(работник_id)
INNER JOIN курица USING(курица_id)
WHERE
    работник_id = 2;

```

```

SELECT работник_id, (SUM(среднее_количество_яиц_в_месяц) / 30.44) AS
среднее_яиц_в_день FROM работник
INNER JOIN наблюдает_за_курицей_в_клетке USING(работник_id)
INNER JOIN курица USING(курица_id)
INNER JOIN порода USING(название_породы)
GROUP BY работник_id;

```

```

SELECT номер_цеха FROM курица
INNER JOIN наблюдает_за_курицей_в_клетке USING(курица_id)
INNER JOIN клетка USING(клетка_id)
ORDER BY количество_яиц_в_месяц DESC
LIMIT 1;

```

```

SELECT цехи.номер_цеха, порода.название_породы, COALESCE(количество,0) AS
количество FROM (SELECT DISTINCT номер_цеха FROM клетка) AS цехи
CROSS JOIN порода
LEFT JOIN (SELECT номер_цеха, COUNT(номер_цеха) AS количество,
название_породы FROM курица
INNER JOIN наблюдает_за_курицей_в_клетке
USING(курица_id)
INNER JOIN клетка USING(клетка_id)
INNER JOIN порода USING(название_породы)
GROUP BY номер_цеха, название_породы
ORDER BY номер_цеха, название_породы) AS под ON
цехи.номер_цеха = под.номер_цеха

```

```
AND под.название_породы = порода.название_породы  
ORDER BY цехи.номер_цеха, порода.название_породы;
```

```
SELECT работник_id, COUNT(наблюдает_за_курицей_в_клетке_id) AS Количество  
FROM работник  
LEFT JOIN наблюдает_за_курицей_в_клетке USING(работник_id)  
GROUP BY работник_id  
ORDER BY работник_id;
```