

**Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики**

**Лабораторная работа №3  
Реализация SQL-запросов**

**Выполнил:**

**Хуторян А.**

**Гр. D41421**

**Проверил:**

**Говоров А.И.**

**Санкт-Петербург**

**2020**

**Цель работы:** Осуществить выдачу данных из модели БД на основе вариантов из 1 лабораторной работы.

#### Задание 1

Создать программную систему, предназначенную для администратора гостиницы.

Такая система должна обеспечивать хранение сведений об имеющихся в гостинице номерах, о проживающих в гостинице клиентах и о служащих, убирающихся в номерах. Количество номеров в гостинице известно, и имеются номера трех типов: одноместный, двухместный и трехместный, отличающиеся стоимостью проживания в сутки. В каждом номере есть телефон.

О каждом проживающем должна храниться следующая информация: номер паспорта, фамилия, имя, отчество, город, из которого он прибыл, дата поселения в гостинице, выделенный гостиничный номер.

О служащих гостиницы должна быть известна информация следующего содержания: фамилия, имя, отчество, где (этаж) и когда (день недели) он убирает. Служащий гостиницы убирает все номера на одном этаже в определенные дни недели, при этом в разные дни он может убирать разные этажи.

Работа с системой предполагает получение следующей информации:

- о клиентах, проживавших в заданном номере, в заданный период времени;
- о количестве клиентов, прибывших из заданного города,
- о том, кто из служащих убирал номер указанного клиента в заданный день недели,
- сколько в гостинице свободных номеров;
- список клиентов с указанием места жительства, которые проживали в те же дни, что и заданный клиент, в определенный период времени.

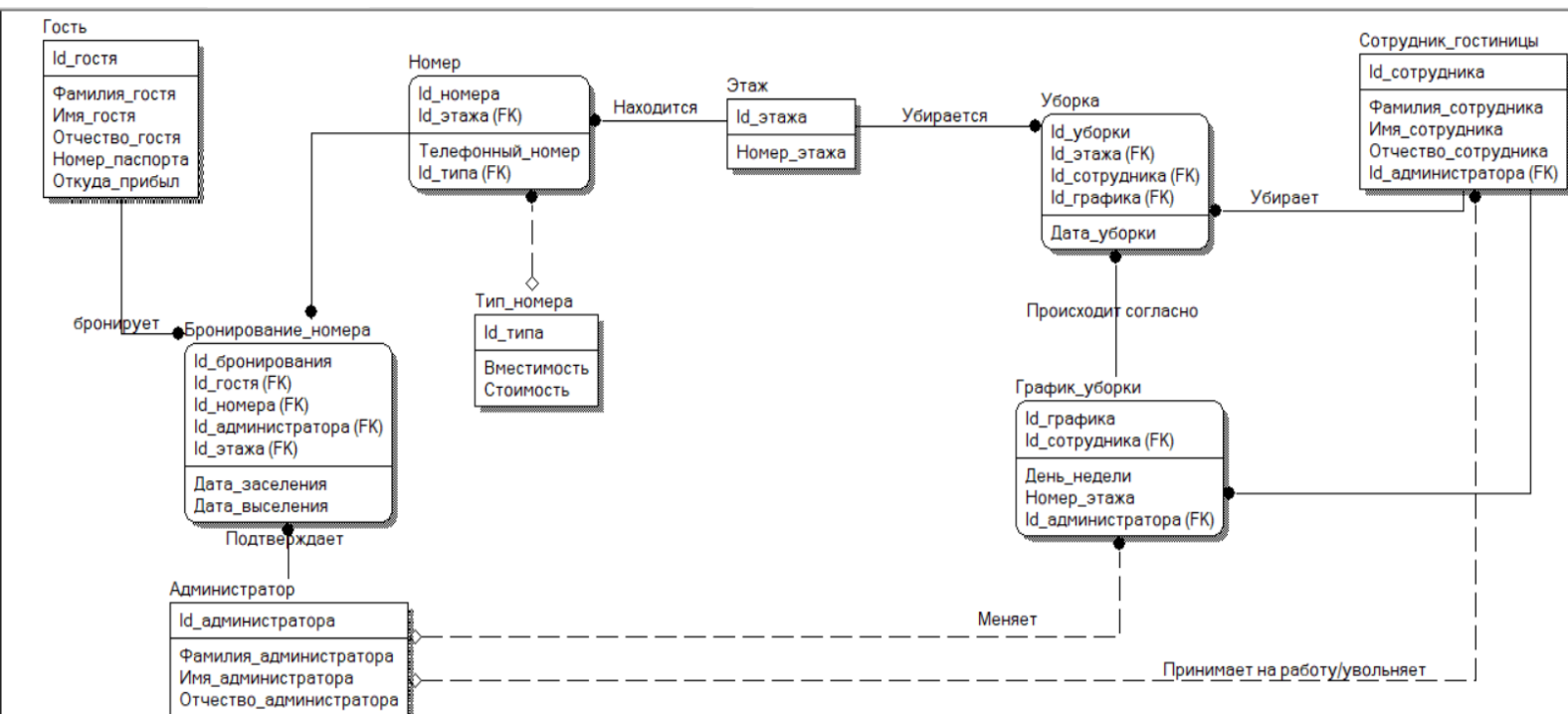
Администратор должен иметь возможность выполнить следующие операции:

- принять на работу или уволить служащего гостиницы;
- изменить расписание работы служащего;
- поселить или выселить клиента.

Необходимо предусмотреть также возможность автоматической выдачи отчета о работе гостиницы за указанный квартал текущего года. Такой отчет должен содержать следующие сведения:

- число клиентов за указанный период в каждом номере;
- количество номеров на каждом этаже;
- общая сумма дохода за каждый номер;
- суммарный доход по всей гостинице.

Схема инфологической модели данных БД, выполненная в среде CA ERwin Data Modeler:

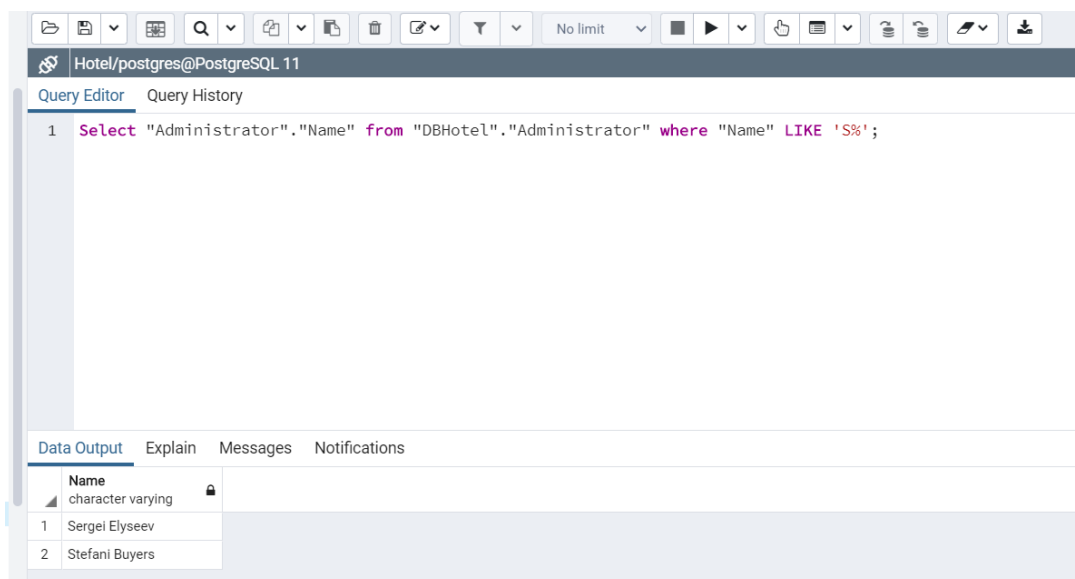


SQL-запросы к БД:

1. Вывести всех администраторов, чье имя начинается на «S»

*SELECT "Administrator"."Name" FROM "DBHotel"."Administrator" WHERE "NAME" LIKE 'S%';*

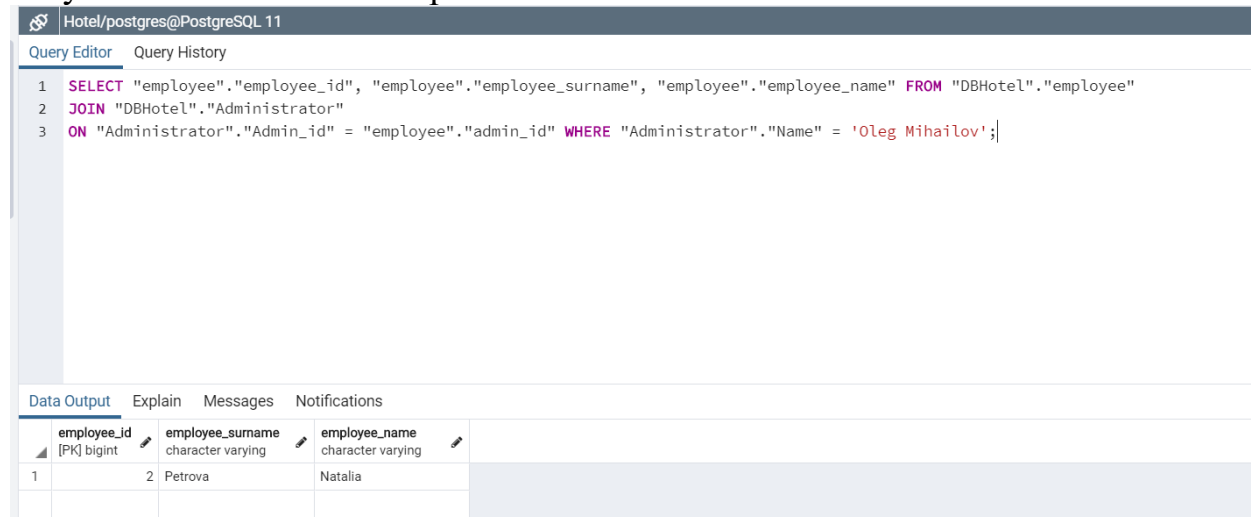
Результат выполнения запроса:



2. Вывести id, имя и фамилию уборщика, которого принял на работу администратор Oleg Mihailov

```
SELECT "employee"."employee_id", "employee"."employee_surname",  
"employee"."employee_name" FROM "DBHotel"."employee"  
JOIN "DBHotel"."Administrator"  
ON "Administrator"."Admin_id" = "employee"."admin_id" WHERE  
"Administrator"."Name" = 'Oleg Mihailov';
```

Результат выполнения запроса:



The screenshot shows the PostgreSQL Query Editor interface. The query editor contains the following SQL query:

```
1 SELECT "employee"."employee_id", "employee"."employee_surname", "employee"."employee_name" FROM "DBHotel"."employee"  
2 JOIN "DBHotel"."Administrator"  
3 ON "Administrator"."Admin_id" = "employee"."admin_id" WHERE "Administrator"."Name" = 'Oleg Mihailov';
```

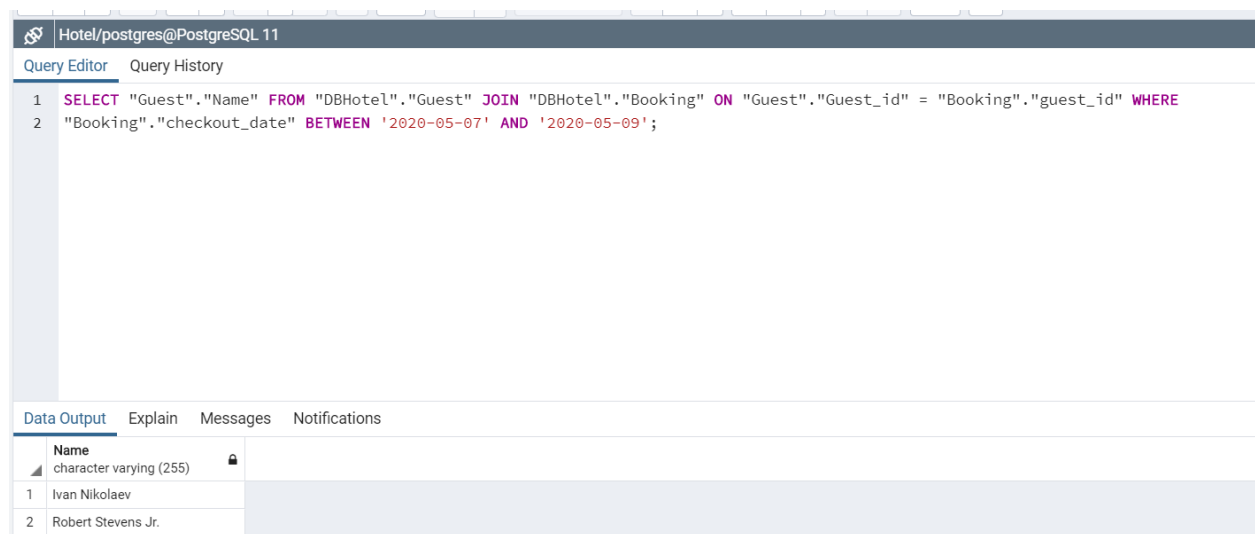
The "Data Output" tab is selected, showing the following result:

employee_id [PK] bigint	employee_surname character varying	employee_name character varying
1	Petrova	Natalia

3. Вывести имена гостей, выезжающих из отеля в промежутке между 7 и 9 мая

```
SELECT "Guest"."Name" FROM "DBHotel"."Guest" JOIN "DBHotel"."Booking" ON  
"Guest"."Guest_id" = "Booking"."guest_id" WHERE  
"Booking"."checkout_date" BETWEEN '2020-05-07' AND '2020-05-09';
```

Результат выполнения запроса:



The screenshot shows the PostgreSQL Query Editor interface. The query editor contains the following SQL query:

```
1 SELECT "Guest"."Name" FROM "DBHotel"."Guest" JOIN "DBHotel"."Booking" ON "Guest"."Guest_id" = "Booking"."guest_id" WHERE  
2 "Booking"."checkout_date" BETWEEN '2020-05-07' AND '2020-05-09';
```

The "Data Output" tab is selected, showing the following result:

Name character varying (255)
1 Ivan Nikolaev
2 Robert Stevens Jr.

4. Вывести даты заезда-выезда для гостей, имя которых начинается на N  
*SELECT "Booking"."checkin\_date", "Booking"."checkout\_date" FROM "DBHotel"."Booking" JOIN "DBHotel"."Guest" ON "Guest"."Guest\_id" = "Booking"."guest\_id" WHERE "Guest"."Name" LIKE 'N%';*

Результат выполнения запроса:

Hotel/postgres@PostgreSQL 11		
Query Editor   Query History		
<pre>1 SELECT "Booking"."checkin_date", "Booking"."checkout_date" FROM "DBHotel"."Booking" JOIN "DBHotel"."Guest" 2 ON "Guest"."Guest_id" = "Booking"."guest_id" WHERE "Guest"."Name" LIKE 'N%';</pre>		
Data Output   Explain   Messages   Notifications		
checkin_date date	checkout_date date	
1	2020-05-07	2020-05-14

5. Вывести имена гостей и администраторов их регистрировавших, которые приехали из городов на букву М

*Select "Administrator"."Name", "Guest"."Name", "Guest"."Where\_from" from "DBHotel"."Administrator" JOIN "DBHotel"."Booking" ON "Booking"."admin\_id" = "Administrator"."Admin\_id" JOIN "DBHotel"."Guest" ON "Guest"."Guest\_id" = "Booking"."guest\_id" WHERE "Guest"."Where\_from" LIKE 'M%';*

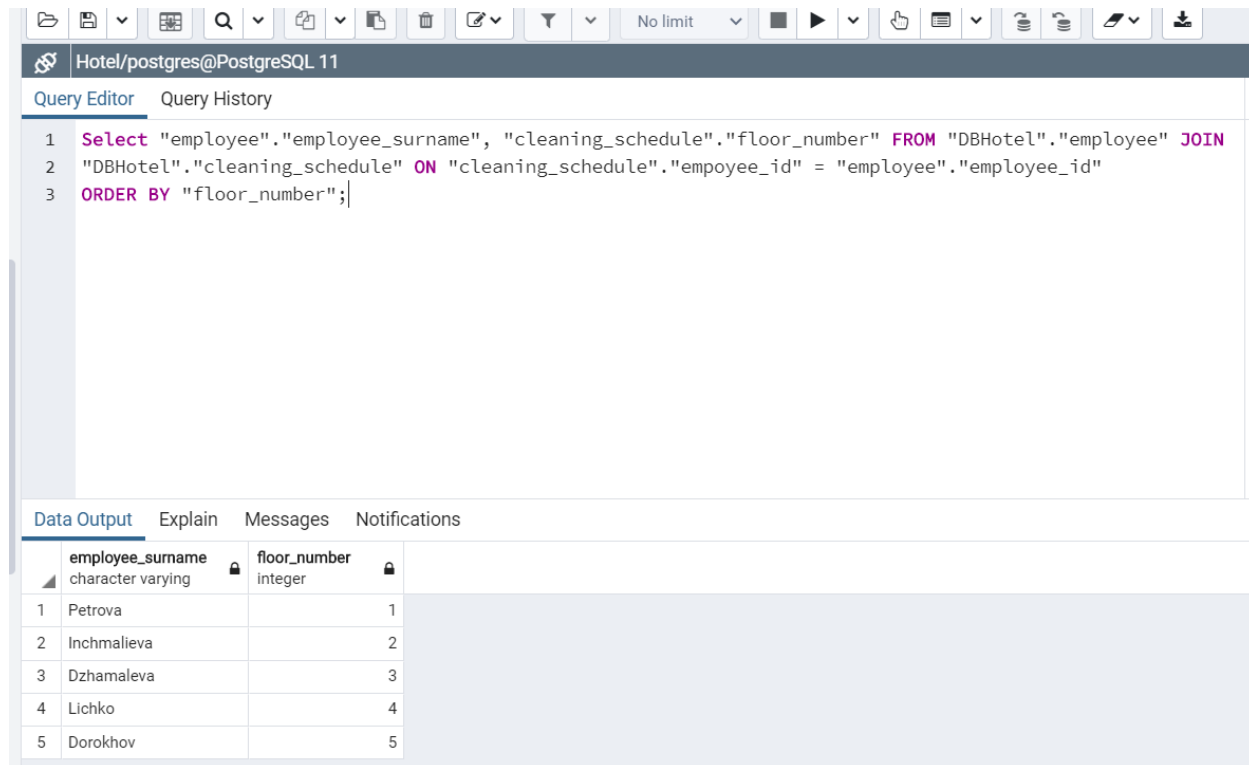
Результат выполнения запроса:

Hotel/postgres@PostgreSQL 11		
Query Editor   Query History   Scratch Pad		
<pre>1 Select "Administrator"."Name", "Guest"."Name", "Guest"."Where_from" from "DBHotel"."Administrator" 2 JOIN "DBHotel"."Booking" ON "Booking"."admin_id" = "Administrator"."Admin_id" JOIN "DBHotel"."Guest" 3 ON "Guest"."Guest_id" = "Booking"."guest_id" WHERE "Guest"."Where_from" LIKE 'M%';</pre>		
Data Output   Explain   Messages   Notifications		
Name character varying	Name character varying (255)	Where_from character varying (255)
1	Sergei Elyseev	Ivan Nikolaev
2	Luci Liu	Ngolo Wan-Bissaka
3	Oleg Mihailov	Sergei Nikolaev

6. Фамилии сотрудников-уборщиков и этаж, на котором они убираются согласно расписанию.

```
Select "employee"."employee_surname", "cleaning_schedule"."floor_number" FROM
"DBHotel"."employee" JOIN "DBHotel"."cleaning_schedule" ON
"cleaning_schedule"."employee_id" = "employee"."employee_id" ORDER BY
"floor_number";
```

Результат выполнения запроса:



The screenshot shows a PostgreSQL query editor interface. The query editor tab is active, displaying the following SQL query:

```
1 Select "employee"."employee_surname", "cleaning_schedule"."floor_number" FROM "DBHotel"."employee" JOIN
2 "DBHotel"."cleaning_schedule" ON "cleaning_schedule"."employee_id" = "employee"."employee_id"
3 ORDER BY "floor_number";
```

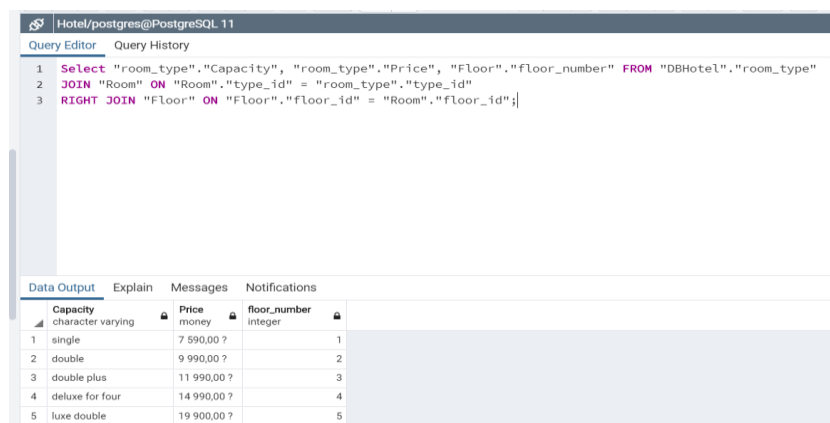
Below the query editor, the "Data Output" tab is active, showing the results of the query in a table format. The table has two columns: "employee\_surname" (character varying) and "floor\_number" (integer). The results are as follows:

employee_surname	floor_number
Petrova	1
Inchmalieva	2
Dzhamaleva	3
Lichko	4
Dorokhov	5

7. Тип номера, его стоимость и этаж на котором он расположен.

```
Select "room_type"."Capacity", "room_type"."Price", "Floor"."floor_number" FROM
"DBHotel"."room_type" JOIN "Room" ON "Room"."type_id" = "room_type"."type_id"
RIGHT JOIN "Floor" ON "Floor"."floor_id" = "Room"."floor_id";
```

Результат выполнения запроса:



The screenshot shows a PostgreSQL query editor interface. The query editor tab is active, displaying the following SQL query:

```
1 Select "room_type"."Capacity", "room_type"."Price", "Floor"."floor_number" FROM "DBHotel"."room_type"
2 JOIN "Room" ON "Room"."type_id" = "room_type"."type_id"
3 RIGHT JOIN "Floor" ON "Floor"."floor_id" = "Room"."floor_id";
```

Below the query editor, the "Data Output" tab is active, showing the results of the query in a table format. The table has three columns: "Capacity" (character varying), "Price" (money), and "floor\_number" (integer). The results are as follows:

Capacity	Price	floor_number
single	7 590,00 ?	1
double	9 990,00 ?	2
double plus	11 990,00 ?	3
deluxe for four	14 990,00 ?	4
lux double	19 900,00 ?	5

8. Посчитать, сколько людей проживало в отеле в указанный период (за месяц май)

```
SELECT COUNT("Booking"."guest_id") FROM "DBHotel"."Booking" JOIN  
"DBHotel"."Room" ON "Room"."room_id" = "Booking"."room_id" WHERE  
"Booking"."checkin_date" >= '2020-05-01' AND "Booking"."checkout_date" <= '2020-06-01';
```

Результат выполнения запроса:

Query Editor	Query History				
<pre>1 SELECT COUNT("Booking"."guest_id") FROM "DBHotel"."Booking" JOIN "DBHotel"."Room" 2 ON "Room"."room_id" = "Booking"."room_id" 3 WHERE "Booking"."checkin_date" &gt;= '2020-05-01' AND "Booking"."checkout_date" &lt;= '2020-06-01';</pre>					
Data Output	Explain Messages Notifications				
<table><thead><tr><th></th><th>count bigint</th></tr></thead><tbody><tr><td>1</td><td>5</td></tr></tbody></table>		count bigint	1	5	
	count bigint				
1	5				

9. Количество номеров на каждом этаже

```
SELECT COUNT("Room"."room_id"), "Floor"."floor_number" FROM "DBHotel"."Room"  
JOIN "DBHotel"."Floor" ON "Floor"."floor_id" = "Room"."floor_id" GROUP BY  
"Floor"."floor_number";
```

Результат выполнения запроса:

Query Editor

Query History

```
1 SELECT COUNT("Room"."room_id"), "Floor"."floor_number" FROM "DBHotel"."Room" JOIN "DBHotel"."Floor"
2 ON "Floor"."floor_id" = "Room"."floor_id" GROUP BY "Floor"."floor_number";
```

Data Output

Explain

Messages

Notifications

	<div>count</div> <div>bigint</div>	<div>floor_number</div> <div>integer</div>	
1	1	1	
2	1	2	
3	1	3	
4	1	4	
5	1	5	

10. Кто из уборщиков убирается на каком этаже в конкретный день недели

```
SELECT "employee"."employee_surname", "cleaning_schedule"."weekday",  
"cleaning_schedule"."floor_number" FROM "DBHotel"."employee" RIGHT JOIN  
"DBHotel"."cleaning_schedule" ON "cleaning_schedule"."employee_id" =  
"employee"."employee_id";
```

Результат выполнения запроса:

Query Editor		Query History
<pre>1 SELECT "employee"."employee_surname", "cleaning_schedule"."weekday", "cleaning_schedule"."floor_number" 2 FROM "DBHotel"."employee" RIGHT JOIN "DBHotel"."cleaning_schedule" ON "cleaning_schedule"."employee_id" = "employee"."employee_id" 3</pre>		
Data Output		Explain Messages Notifications
employee_surname	weekday	floor_number
character varying	character varying	integer
1 Dzhmaleva	monday	3
2 Petrova	tuesday	1
3 Lichko	wednesday	4
4 Inchmalieva	thursday	2
5 Dorokhov	friday	5

11. Вывести имена всех, кто заселился до 06.05

```
SELECT "Guest"."Name" FROM "DBHotel"."Guest" WHERE  
EXISTS (SELECT "Booking"."checkin_date" FROM "DBHotel"."Booking"  
WHERE "Booking"."guest_id" = "Guest"."Guest_id" AND  
"Booking"."checkin_date" < '2020-05-06')
```

Результат выполнения запроса:

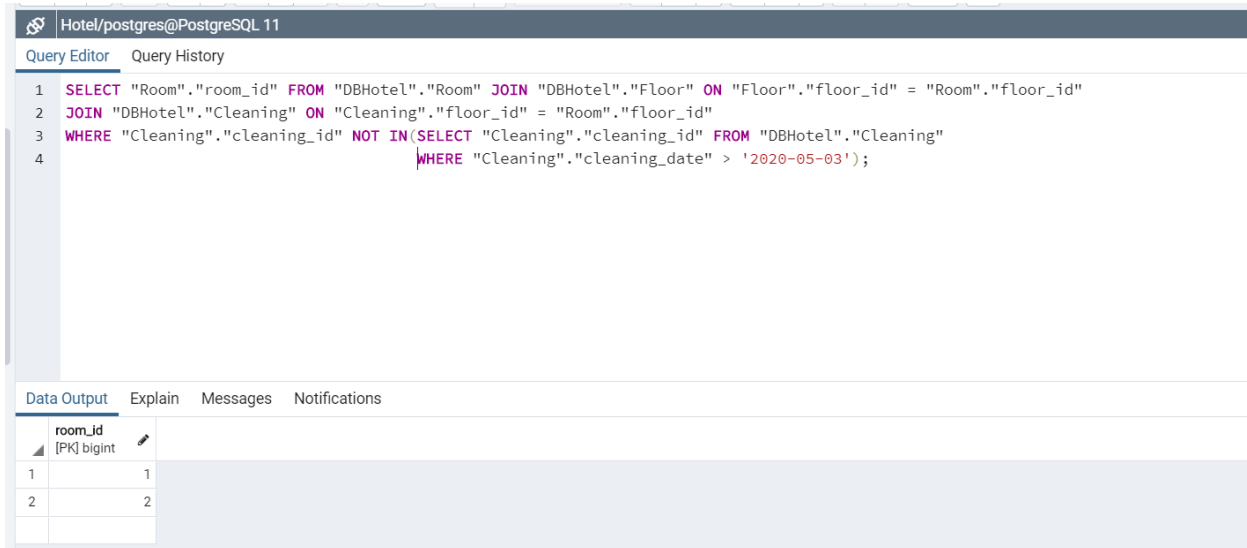
Hotel/postgres@PostgreSQL 11	
Query Editor	
Query History	
<pre>1 SELECT "Guest"."Name" FROM "DBHotel"."Guest" WHERE 2 EXISTS (SELECT "Booking"."checkin_date" FROM "DBHotel"."Booking" 3 WHERE "Booking"."guest_id" = "Guest"."Guest_id" AND "Booking"."checkin_date" &lt; '2020-05-06')</pre>	
Data Output	
Explain Messages Notifications	
Name	
character varying (255)	
1 Ivan Nikolaev	
2 Daniela Espinoza	
3 Robert Stevens Jr.	
4 Chon Guk	



12. Вывести id номеров, которые не убирались после 03.05

```
SELECT "Room"."room_id" FROM "DBHotel"."Room" JOIN "DBHotel"."Floor" ON  
"Floor"."floor_id" = "Room"."floor_id"  
JOIN "DBHotel"."Cleaning" ON "Cleaning"."floor_id" = "Room"."floor_id"  
WHERE "Cleaning"."cleaning_id" NOT IN(SELECT "Cleaning"."cleaning_id" FROM  
"DBHotel"."Cleaning" WHERE "Cleaning"."cleaning_date" > '2020-05-03');
```

Результат выполнения запроса:



The screenshot shows a PostgreSQL query editor with the following query:

```
1 SELECT "Room"."room_id" FROM "DBHotel"."Room" JOIN "DBHotel"."Floor" ON "Floor"."floor_id" = "Room"."floor_id"  
2 JOIN "DBHotel"."Cleaning" ON "Cleaning"."floor_id" = "Room"."floor_id"  
3 WHERE "Cleaning"."cleaning_id" NOT IN(SELECT "Cleaning"."cleaning_id" FROM "DBHotel"."Cleaning"  
4 WHERE "Cleaning"."cleaning_date" > '2020-05-03');
```

The results are displayed in a table with the following columns and data:

room_id [PK] bigint
1
2

13. Вывести номер, в котором не убирались с 1 по 6 мая, но в который кто-то заселился.

```
SELECT "Room"."room_id" FROM "DBHotel"."Room" JOIN "DBHotel"."Floor" ON  
"Floor"."floor_id" = "Room"."floor_id"  
JOIN "DBHotel"."Cleaning" ON "Cleaning"."floor_id" = "Room"."floor_id"  
WHERE EXISTS(SELECT "Room"."room_id" FROM "DBHotel"."Room"  
JOIN  
"DBHotel"."Booking" ON "Booking"."room_id" = "Room"."room_id"  
WHERE "Booking"."checkin_date" < '2020-05-05' AND "Cleaning"."cleaning_id"  
NOT IN(SELECT "Cleaning"."cleaning_id" FROM  
"DBHotel"."Cleaning"  
WHERE "Cleaning"."cleaning_date" < '2020-05-06'  
AND "Cleaning"."cleaning_date" > '2020-05-01'));
```

Результат выполнения запроса:

Query Editor	Query History						
<pre> 1 SELECT "Room"."room_id" FROM "DBHotel"."Room" JOIN "DBHotel"."Floor" ON "Floor"."floor_id" = "Room"."floor_id" 2 JOIN "DBHotel"."Cleaning" ON "Cleaning"."floor_id" = "Room"."floor_id" 3 WHERE EXISTS(SELECT "Room"."room_id" FROM "DBHotel"."Room" 4 JOIN "DBHotel"."Booking" ON "Booking"."room_id" = "Room"."room_id" 5 WHERE "Booking"."checkin_date" &lt; '2020-05-05' AND "Cleaning"."cleaning_id" 6 NOT IN(SELECT "Cleaning"."cleaning_id" FROM "DBHotel"."Cleaning" 7 WHERE "Cleaning"."cleaning_date" &lt; '2020-05-06' AND "Cleaning"."cleaning_date" &gt; '2020-05-01')); 8 </pre>							
Data Output	Explain Messages Notifications						
<table> <tr> <th>room_id</th><th></th></tr> <tr> <td>[PK] bigint</td><td></td></tr> <tr> <td>1</td><td>5</td></tr> </table>	room_id		[PK] bigint		1	5	
room_id							
[PK] bigint							
1	5						

14. Все свободные на момент 10 мая номера с датами выселения последних жильцов

```

SELECT "Room"."room_id", (MAX("Booking"."checkout_date")) AS checkout_date
FROM "DBHotel"."Room" LEFT JOIN "DBHotel"."Booking" ON "Booking"."room_id" =
"Room"."room_id"
GROUP BY "Room"."room_id"
HAVING (((MAX("Booking"."checkout_date"))< '2020-05-10')) OR
(((MAX("Booking"."checkout_date")) IS NULL));

```

Результат выполнения запроса:

Query Editor	Query History										
<pre> 1 SELECT "Room"."room_id", (MAX("Booking"."checkout_date")) AS checkout_date 2 FROM "DBHotel"."Room" LEFT JOIN "DBHotel"."Booking" ON "Booking"."room_id" = "Room"."room_id" 3 GROUP BY "Room"."room_id" 4 HAVING (((MAX("Booking"."checkout_date"))&lt; '2020-05-10')) OR (((MAX("Booking"."checkout_date")) IS NULL)) 5 </pre>											
Data Output	Explain Messages Notifications										
<table> <tr> <th>room_id</th><th>checkout_date</th></tr> <tr> <td>[PK] bigint</td><td>date</td></tr> <tr> <td>1</td><td>2020-05-09</td></tr> <tr> <td>2</td><td>2020-05-06</td></tr> <tr> <td>3</td><td>2020-05-07</td></tr> </table>	room_id	checkout_date	[PK] bigint	date	1	2020-05-09	2	2020-05-06	3	2020-05-07	
room_id	checkout_date										
[PK] bigint	date										
1	2020-05-09										
2	2020-05-06										
3	2020-05-07										

15. Сколько заплатил каждый постоялец за время своего пребывания в гостинице.

```
SELECT DISTINCT "room_type"."type_id" AS Room_type,
COUNT("Booking"."booking_id") AS Num_bookings,
SUM("room_type"."Price") AS living_price,
ABS(date("Booking"."checkout_date") - date("Booking"."checkin_date")) AS
Amount_of_days,
SUM("room_type"."Price" * ABS(date("Booking"."checkout_date") -
date("Booking"."checkin_date")) AS living_cost
FROM("DBHotel"."room_type" INNER JOIN ( "DBHotel"."Booking" INNER JOIN
"DBHotel"."Room" ON "Room"."room_id" = "Booking"."room_id")
ON "room_type"."type_id" = "Room"."type_id")
GROUP BY "room_type"."type_id", "Booking"."checkout_date", "Booking"."checkin_date";
```

Результат выполнения запроса:

Query EditorQuery History

```
1 SELECT DISTINCT "room_type"."type_id" AS Room_type, COUNT("Booking"."booking_id") AS Num_bookings,
2 SUM("room_type"."Price") AS living_price,
3 ABS(date("Booking"."checkout_date") - date("Booking"."checkin_date")) AS Amount_of_days,
4 SUM("room_type"."Price" * ABS(date("Booking"."checkout_date") - date("Booking"."checkin_date"))) AS living_cost
5 FROM("DBHotel"."room_type" INNER JOIN ( "DBHotel"."Booking" INNER JOIN "DBHotel"."Room" ON "Room"."room_id" = "Booking"."room_id"
6 ON "room_type"."type_id" = "Room"."type_id")
7 GROUP BY "room_type"."type_id", "Booking"."checkout_date", "Booking"."checkin_date";
8
```

Data OutputExplainMessagesNotifications

	room_type integer	num_bookings bigint	living_price money	amount_of_days integer	living_cost money
1		1	7 590,00 ?	5	37 950,00 ?
2		2	9 990,00 ?	13	129 870,00 ?
3		3	11 990,00 ?	10	119 900,00 ?
4		4	14 990,00 ?	4	59 960,00 ?
5		5	19 900,00 ?	7	139 300,00 ?