

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

**ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 4**

**по теме: Запросы на выборку данных к БД PostgreSQL. ПРЕДСТАВЛЕНИЯ В
PostgreSQL.**

по дисциплине: Проектирование и реализация баз данных

Специальность:

09.03.03 Мобильные и сетевые технологии

Проверил:

Говорова М.М. _____

Дата: « » Июня 2021г.

Оценка _____

Выполнил(и):

студент(ы)

группы К3241

Касаткин Д.А.

ЦЕЛЬ РАБОТЫ

Создание представлений и запросов на выборку данных к базе данных PostgreSQL и использование подзапросов при модификации данных.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов.
4. Просмотреть историю запросов.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант 8, БД «Аэропорт»

ВЫПОЛНЕНИЕ

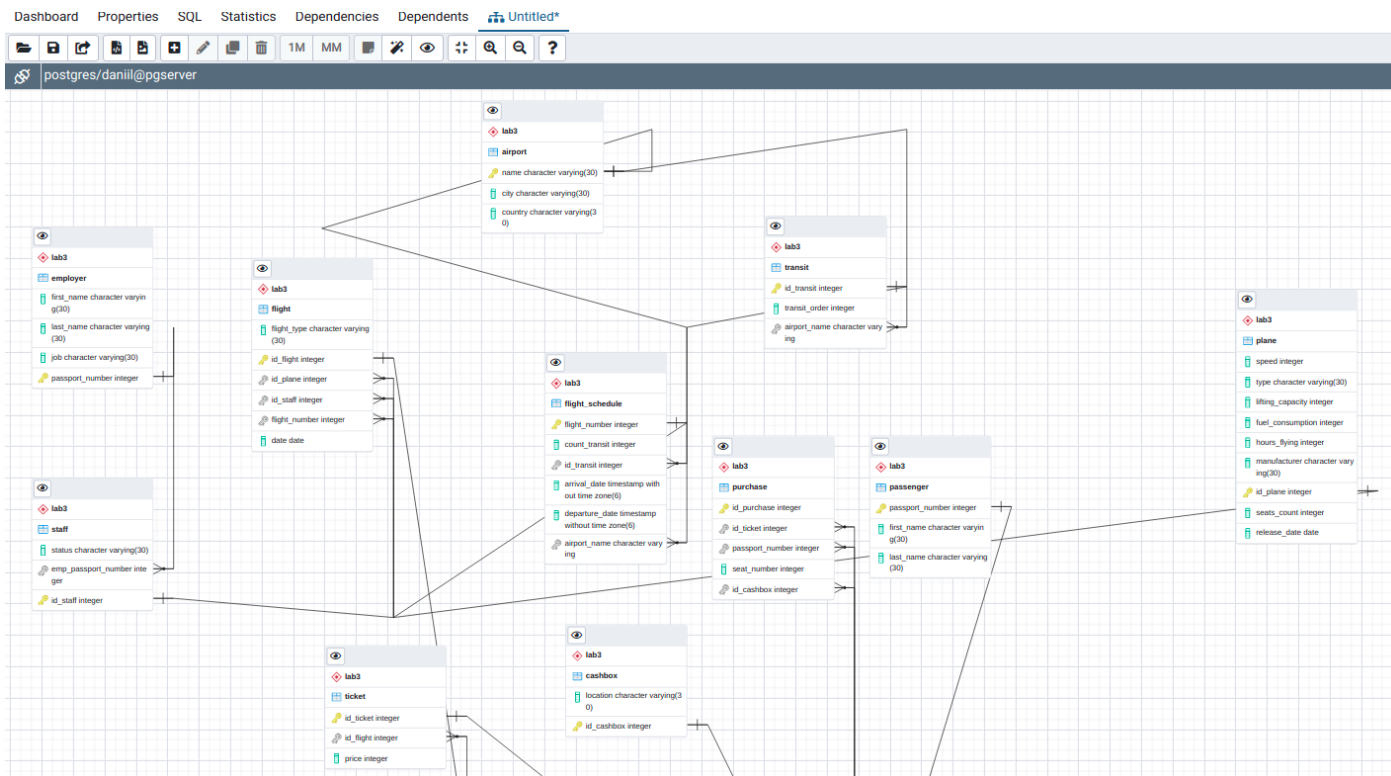
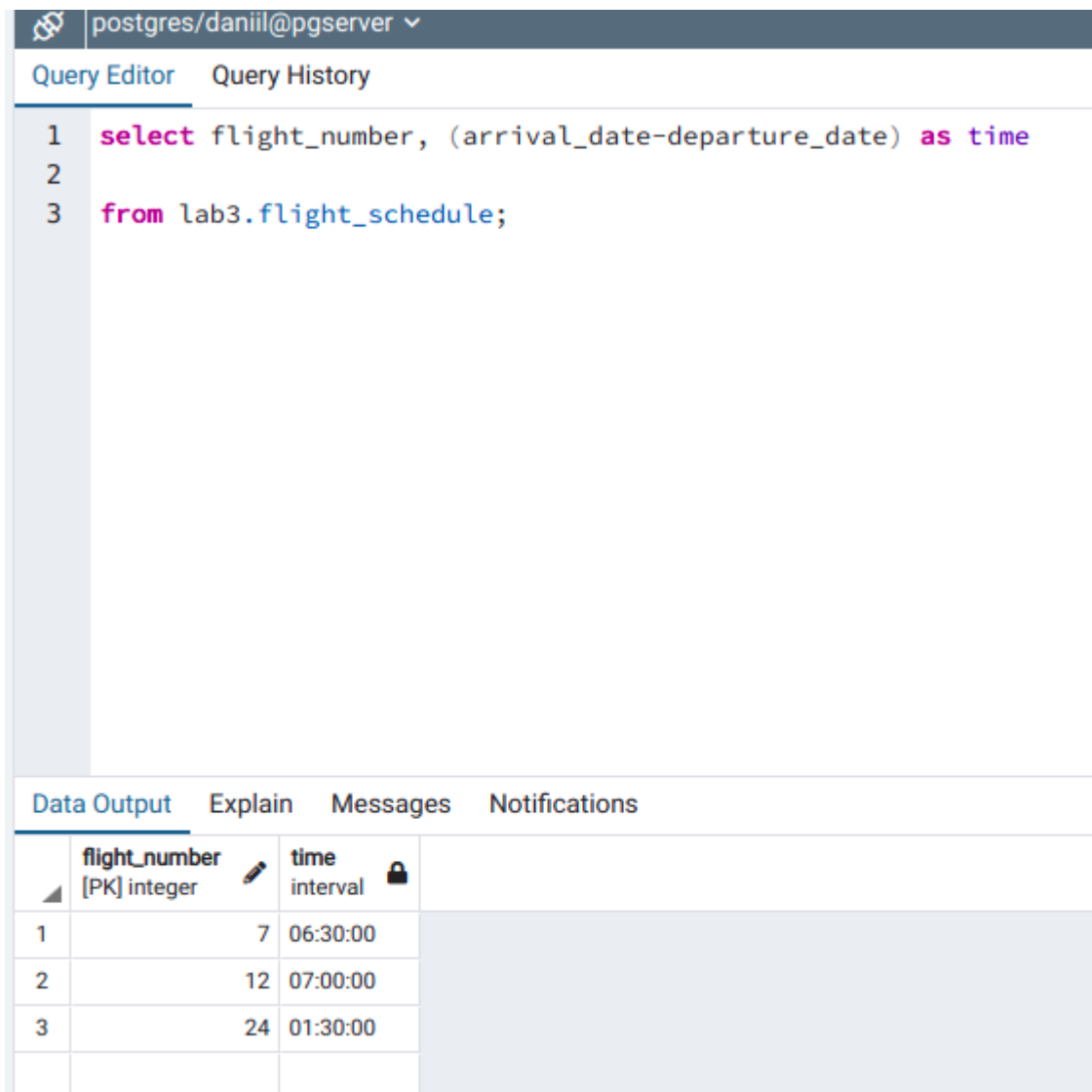


Рисунок 1 – Схема инфологической модели БД, сгенерированная в Generate ERD

Запросы:

1) Определить расчетное время полета по всем маршрутам.



The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is 'postgres/daniil@pgserver'. Below the connection bar, there are two tabs: 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query:

```
1 select flight_number, (arrival_date-departure_date) as time
2
3 from lab3.flight_schedule;
```

Below the query editor, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'flight_number' (integer, primary key) and 'time' (interval). The results are as follows:

	flight_number [PK] integer	time interval
1	7	06:30:00
2	12	07:00:00
3	24	01:30:00

2) Определить расход топлива по всем маршрутам.

postgres/daniil@pgserver

Query Editor Query History

```

1 select DISTINCT lab3.flight.flight_number, lab3.plane.id_plane,
2 round((EXTRACT(epoch from (arrival_date-departure_date))/3600 * fuel_consumption)::numeric,2)
3 as total_fuel_per_flight
4 from (lab3.flight_schedule join lab3.flight
5      on lab3.flight_schedule.flight_number = lab3.flight.flight_number)
6      join lab3.plane
7      on lab3.flight.id_plane=lab3.plane.id_plane;

```

Data Output Explain Messages Notifications

	flight_number integer	id_plane integer	total_fuel_per_flight numeric
1	7	102	585.00
2	12	103	1050.00
3	24	102	135.00

3) Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по одному из рейсов за вчерашний день.

```

1 SELECT current_date-1 as flight_date, min(seats_count)- count(id_ticket) as left_seats
2 from (lab3.ticket join lab3.flight
3      on lab3.ticket.id_flight=lab3.flight.id_flight)
4      join lab3.plane
5      on lab3.flight.id_plane =lab3.plane.id_plane
6      where lab3.flight.date=current_date-1;

```





Data Output Explain Messages Notifications

	flight_date date	left_seats bigint
1	2021-06-07	95

4) Рассчитать убытки компании за счет непроданных билетов за вчерашний день.

```
1 SELECT flight_number,date , price* (MIN(seats_count)- COUNT(id_ticket)) AS loss
2 FROM (lab3.ticket JOIN lab3.flight
3      ON lab3.ticket.id_flight=lab3.flight.id_flight)
4      JOIN lab3.plane
5      ON lab3.flight.id_plane =lab3.plane.id_plane
6 WHERE lab3.flight.date=CURRENT_DATE-1
7 GROUP BY flight_number, date, price
8
```

Data Output Explain Messages Notifications

	 flight_number integer	 date date	 loss bigint	
1	7	2021-06-07	227715	

5) Вывести список самолетов, которые не ремонтировались в течение более чем 3 лет.


```

1 SELECT id_plane, release_date FROM lab3.plane
2 WHERE lab3.plane.release_date < CURRENT_DATE-1095

```

Data Output Explain Messages Notifications

	id_plane [PK] integer	release_date date	
1	102	2016-04-09	
2	103	2011-03-23	

6) Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения.

Query Editor Query History

```

1 SELECT lab3.flight_schedule.airport_name, lab3.plane.type, COUNT (id_flight) from
2 (lab3.flight join lab3.plane on lab3.flight.id_plane =lab3.plane.id_plane )
3 join lab3.flight_schedule on lab3.flight.flight_number =lab3.flight_schedule.flight_number
4 group by lab3.plane.type, lab3.flight_schedule.airport_name having lab3.flight_schedule.airport_name='Plukovo' and count(id_flight)=
5 (select count(id_flight) as count_fi from (lab3.flight join lab3.plane on lab3.flight.id_plane=lab3.plane.id_plane) join lab3.flight_schedule on
6 lab3.flight.flight_number = lab3.flight_schedule.flight_number
7 group by lab3.plane.type, lab3.flight_schedule.airport_name
8 having lab3.flight_schedule.airport_name='Plukovo' order by count_fi desc limit 1);

```

Data Output Explain Messages Notifications

	airport_name character varying	type character varying (30)	count bigint
1	Plukovo	грузовой	1
2	Plukovo	пассажирский	1

- 7) Вывести список самолетов, “возраст” которых превышает средний “возраст” самолетов этого типа.

```
1 select id_plane, release_date from lab3.plane
2 where release_date < (SELECT to_timestamp(AVG(EXTRACT(epoch from release_date))::date
3                        from lab3.plane);
```

	Data Output	Explain	Messages	Notifications						
	<table><thead><tr><th></th><th>id_plane [PK] integer</th><th>release_date date</th></tr></thead><tbody><tr><td>1</td><td>103</td><td>2011-03-23</td></tr></tbody></table>		id_plane [PK] integer	release_date date	1	103	2011-03-23			
	id_plane [PK] integer	release_date date								
1	103	2011-03-23								

1. Представления

- 1) Для пассажиров авиакомпании о рейсах в Москву на ближайшую неделю.

postgres/daniil@pgserver

Query Editor Query History

```
1
2 create view lab3.moscow_flights as select lab3.flight.flight_number, lab3.flight.date, departure_date, arrival_date, city from(lab3.flight_schedule join lab3.flight
3 on lab3.flight_schedule.flight_number =lab3.flight.flight_number ) join lab3.airport on lab3.flight_schedule.airport_name= lab3.airport.name where
4 lab3.flight.date > current_date-1 and lab3.flight.date < current_date+7 and lab3.airport.city= 'Moscow'
5
6
```

	Data Output	Explain	Messages	Notifications										
	<table><thead><tr><th>flight_number</th><th>date</th><th>departure_date</th><th>arrival_date</th><th>city</th></tr><tr><td>integer</td><td>date</td><td>timestamp without time zone</td><td>timestamp without time zone</td><td>character varying (30)</td></tr></thead></table>	flight_number	date	departure_date	arrival_date	city	integer	date	timestamp without time zone	timestamp without time zone	character varying (30)			
flight_number	date	departure_date	arrival_date	city										
integer	date	timestamp without time zone	timestamp without time zone	character varying (30)										

Тоже самое для Питера

postgres/danill@pgserver

Query Editor Query History

```
1
2 create view lab3.saint_pet as select lab3.flight.flight_number, lab3.flight.date, departure_date, arrival_date, city from(lab3.flight_schedule join lab3.flight
3 on lab3.flight_schedule.flight_number =lab3.flight.flight_number ) join lab3.airport on lab3.flight_schedule.airport_name= lab3.airport.name where
4 lab3.flight.date > current_date-1 and lab3.flight.date < current_date+7 and lab3.airport.city= 'Saint-Petersburg'
5
6
```

Data Output Explain Messages Notifications

	flight_number integer	date date	departure_date timestamp without time zone	arrival_date timestamp without time zone	city character varying (30)
1	7	2021-06-12	2021-06-11 23:30:00	2021-06-12 02:00:00	Saint-Petersburg

2) Количество самолетов каждого типа, которыми владеет компания.

Query Editor Query History

```
1
2 create view lab3.number_of_planes as
3 select lab3.plane.type, count ( plane.id_plane) from lab3.plane
4 group by lab3.plane.type;
5
6
```

Data Output Explain Messages Notifications

	type character varying (30)	count bigint
1	пассажирский	2
2	грузовой	1

2. Запросы на модификацию данных

1) INSERT добавление новой записи о билете, стоимость которого меньше 5000



The screenshot shows a SQL Query Editor with a query that inserts a new record into the 'purchase' table. The query uses a subquery to find the next available 'id_purchase' by incrementing the maximum value in the table. The 'id_ticket' is taken from the 'ticket' table, and the 'price' is set to 308075673, which is less than 5000. The Messages pane shows 'INSERT 0 1' and 'Query returned successfully in 84 msec.'

```
1 insert into lab3.purchase
2 select (select max(id_purchase)+1 from lab3.purchase), id_ticket, 308075673, 32, 13
3 from lab3.ticket
4 where price <5000
```

Messages

INSERT 0 1

Query returned successfully in 84 msec.

До:

	id_purchase [PK] integer	id_ticket integer	passport_number integer	seat_number integer	id_cashbox integer
1	581	749	308043267	13	14
2	798	456	308075673	56	13

После:





Data Output							Explain	Messages	Notifications
	id_purchase [PK] integer	id_ticket integer	passport_number integer	seat_number integer	id_cashbox integer				
1	581	749	308043267	13	14				
2	798	456	308075673	56	13				
3	799	456	308075673	32	13				

2) UPDATE обновление цены на рейсы продолжительностью менее 3 часов





```

postgres/daniil@pgserver
Query Editor  Query History
1 update lab3.ticket
2 set price = 2497
3 where id_flight in (select id_flight from lab3.flight where flight_number in (select flight_number from lab3.flight_schedule where (arrival_date-departure_date)<'03:00:00'));
  
```

До:

Data Output					Explain	Messages	Notifications
	 id_ticket [PK] integer	 id_flight integer	 price integer				
1		456	123	2397			
2		749	124	5789			

После:

Data Output				Explain	Messages	Notifications
	 id_ticket [PK] integer	 id_flight integer	 price integer			
1		456	123	2497		
2		749	124	2497		

3) DELETE удаление записей о полетах, в которых дата выпуска самолета раньше чем 2011.01.01

postgres/daniil@pgserver

Query Editor Query History

```

1 delete from lab3.flight
2 where id_plane in (select id_plane from lab3.plane where lab3.plane.release_date<'2011-01-01');|

```

Data Output Explain Messages Notifications

DELETE 1

Query returned successfully in 92 msec.

До:

	flight_type character varying (30)	id_flight [PK] integer	id_plane integer	id_staff integer	flight_number integer	date date
1	Regular	123	102	98	7	2021-06-12
2	Regular	124	102	97	24	2021-06-21
3	Transfer	127	103	96	12	2021-03-19
4	Transfer	128	104	96	12	2021-03-19

После:

Data Output		Explain	Messages	Notifications			
	flight_type character varying (30)	id_flight [PK] integer	id_plane integer	id_staff integer	flight_number integer	date date	
1	Regular	123	102	98	7	2021-06-12	
2	Regular	124	102	97	24	2021-06-21	
3	Transfer	127	103	96	12	2021-03-19	

ВЫВОДЫ

В ходе выполнения лабораторной работы, согласно индивидуальному заданию, были созданы семь запросов к БД, два представления и три запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.