

# Домашняя работа №5

Попов Матвей, М8О-114СВ-24

## Задание 2

Этот запрос выбирает из таблицы «Билеты» (tickets) всех пассажиров с именами, состоящими из трех букв (в шаблоне присутствуют три символа «\_»):

```
SELECT passenger_name  
FROM tickets  
WHERE passenger_name LIKE '___ %';
```

Предложите шаблон поиска в операторе LIKE для выбора из этой таблицы всех пассажиров с фамилиями, состоящими из пяти букв.

## Запрос

```
SELECT passenger_name FROM tickets  
WHERE passenger_name LIKE '% _____';
```

## Результат

	 passenger_name  
1	ILYA POPOV
2	VLADIMIR POPOV
3	PAVEL GUSEV
4	LEONID ORLOV
5	EVGENIY GUSEV
6	NIKOLAY FOMIN
7	EKATERINA ILINA
8	ANTON POPOV
9	ARTEM BELOV
10	VLADIMIR POPOV
11	ALEKSEY ISAEV

С

## Задание 7

Самые крупные самолеты в нашей авиакомпании — это Boeing 777-300. Выяснить, между какими парами городов они летают, поможет запрос:

```
SELECT DISTINCT departure_city, arrival_city
  FROM routes r
 JOIN aircrafts a ON r.aircraft_code = a.aircraft_code
 WHERE a.model = 'Boeing 777-300'
 ORDER BY 1;
```

departure_city		arrival_city
Екатеринбург		Москва
Москва		Екатеринбург
Москва		Новосибирск
Москва		Пермь
Москва		Сочи
Новосибирск		Москва
Пермь		Москва
Сочи		Москва

(8 строк)

К сожалению, в этой выборке информация дублируется. Пары городов приведены по два раза: для рейса «туда» и для рейса «обратно». Модифицируйте запрос таким образом, чтобы каждая пара городов была выведена только один раз:

departure_city		arrival_city
Москва		Екатеринбург
Новосибирск		Москва
Пермь		Москва
Сочи		Москва

## Запрос

```
SELECT DISTINCT departure_city, arrival_city
  FROM routes r
 JOIN aircrafts a ON r.aircraft_code = a.aircraft_code
 WHERE a.aircraft_code = '773'
        AND departure_city > arrival_city
 ORDER BY 1;
```

## Результат

	departure_city ▾	arrival_city ▾
1	Москва	Екатеринбург
2	Новосибирск	Москва
3	Пермь	Москва
4	Сочи	Москва

## Задание 9

Для ответа на вопрос, сколько рейсов выполняется из Москвы в Санкт-Петербург, можно написать совсем простой запрос:

```
SELECT count( * )
  FROM routes
 WHERE departure_city = 'Москва'
        AND arrival_city = 'Санкт-Петербург';
```

```
count
-----
    12
(1 строка)
```

А с помощью какого запроса можно получить результат в таком виде?

```
departure_city | arrival_city | count
-----+-----+-----
Москва         | Санкт-Петербург |    12
(1 строка)
```

## Запрос

```
SELECT departure_city, arrival_city, COUNT(*) FROM routes
WHERE departure_city = 'Москва' AND arrival_city = 'Санкт-Петербург'
GROUP BY departure_city, arrival_city;
```

## Результат

	<input type="text" value="departure_city"/>	<input type="text" value="arrival_city"/>	<input type="text" value="count"/>
1	Москва	Санкт-Петербург	12

## Задание 13

Ответить на вопрос о том, каковы максимальные и минимальные цены билетов на все направления, может такой запрос:

```
SELECT f.departure_city, f.arrival_city,
       max( tf.amount ), min( tf.amount )
FROM flights_v f
JOIN ticket_flights tf ON f.flight_id = tf.flight_id
GROUP BY 1, 2
ORDER BY 1, 2;
```

departure_city	arrival_city	max	min
Абакан	Москва	101000.00	33700.00
Абакан	Новосибирск	5800.00	5800.00
Абакан	Томск	4900.00	4900.00
Анадырь	Москва	185300.00	61800.00
Анадырь	Хабаровск	92200.00	30700.00
...			
Якутск	Мирный	8900.00	8100.00
Якутск	Санкт-Петербург	145300.00	48400.00

(367 строк)

А как выявить те направления, на которые не было продано ни одного билета? Один из вариантов решения такой: если на рейсы, отправляющиеся по какому-то направлению, не было продано ни одного билета, то максимальная и минимальная цены будут равны NULL. Нужно получить выборку в таком виде:

departure_city	arrival_city	max	min
Абакан	Архангельск		
Абакан	Грозный		
Абакан	Кызыл		
Абакан	Москва	101000.00	33700.00
Абакан	Новосибирск	5800.00	5800.00
...			

Модифицируйте запрос, приведенный выше.

## Запрос

```
SELECT f.departure_city, f.arrival_city,
max( tf.amount ), min( tf.amount )
FROM flights_v f
LEFT JOIN ticket_flights tf ON f.flight_id = tf.flight_id
GROUP BY 1, 2
ORDER BY 1, 2;
```

## Результат

	departure_city	arrival_city	max	min
1	Абакан	Архангельск	<null>	<null>
2	Абакан	Грозный	<null>	<null>
3	Абакан	Кызыл	<null>	<null>
4	Абакан	Москва	101000	33700
5	Абакан	Новосибирск	5800	5800
6	Абакан	Томск	4900	4900
7	Анадырь	Москва	185300	61800
8	Анадырь	Хабаровск	92200	30700
9	Анапа	Белгород	18900	6300
10	Анапа	Москва	36600	12200
11	Анапа	Новокузнецк	<null>	<null>
12	Архангельск	Абакан	<null>	<null>
13	Архангельск	Иркутск	<null>	<null>
14	Архангельск	Москва	11100	10100

## Задание 19

**Задание 1.** Модифицируйте запрос, добавив в него столбец `level` (можно назвать его и `iteration`). Этот столбец должен содержать номер текущей итерации, поэтому нужно увеличивать его значение на единицу на каждом шаге. Не забудьте задать начальное значение для добавленного столбца в предложении `VALUES`.

**Задание 2.** Для завершения экспериментов замените `UNION ALL` на `UNION` и выполните запрос. Сравните этот результат с предыдущим, когда мы использовали `UNION ALL`.

### Запрос

```
WITH RECURSIVE ranges ( min_sum, max_sum, level )
AS (
VALUES( 0, 100000, 1 ),
( 100000, 200000, 2 ),
( 200000, 300000, 3 )
UNION ALL
SELECT min_sum + 100000, max_sum + 100000, level + 1
FROM ranges
WHERE max_sum < ( SELECT max( total_amount ) FROM bookings )
)
SELECT * FROM ranges;
```

```
WITH RECURSIVE ranges ( min_sum, max_sum, level )
AS (
VALUES( 0, 100000, 1 ),
( 100000, 200000, 2 ),
( 200000, 300000, 3 )
UNION
SELECT min_sum + 100000, max_sum + 100000, level + 1
FROM ranges
WHERE max_sum < ( SELECT max( total_amount ) FROM bookings )
)
SELECT * FROM ranges;
```

## Результат

	<input type="checkbox"/> min_sum ▾	↕	<input type="checkbox"/> max_sum ▾	↕	<input type="checkbox"/> level ▾	↕
1	0		100000		1	
2	100000		200000		2	
3	200000		300000		3	
4	100000		200000		2	
5	200000		300000		3	
6	300000		400000		4	
7	200000		300000		3	
8	300000		400000		4	
9	400000		500000		5	
10	300000		400000		4	
11	400000		500000		5	
12	500000		600000		6	

	<input type="checkbox"/> min_sum ▾	↕	<input type="checkbox"/> max_sum ▾	↕	<input type="checkbox"/> level ▾	↕
1	0		100000		1	
2	100000		200000		2	
3	200000		300000		3	
4	300000		400000		4	
5	400000		500000		5	
6	500000		600000		6	
7	600000		700000		7	
8	700000		800000		8	
9	800000		900000		9	
10	900000		1000000		10	
11	1000000		1100000		11	
12	1100000		1200000		12	
13	1200000		1300000		13	

## Задание 21

В тексте главы был приведен запрос, выводящий список городов, в которые нет рейсов из Москвы.

```
SELECT DISTINCT a.city  
FROM airports a  
WHERE NOT EXISTS (  
    SELECT * FROM routes r  
    WHERE r.departure_city = 'Москва'  
    AND r.arrival_city = a.city  
)  
AND a.city <> 'Москва'  
ORDER BY city;
```

Можно предложить другой вариант, в котором используется одна из операций над множествами строк: объединение, пересечение или разность.

Вместо знака «?» поставьте в приведенном ниже запросе нужное ключевое слово — UNION, INTERSECT или EXCEPT — и обоснуйте ваше решение.



```
SELECT city  
FROM airports  
WHERE city <> 'Москва'  
?  
SELECT arrival_city  
FROM routes  
WHERE departure_city = 'Москва'  
ORDER BY city;
```

## Запрос

```
SELECT city  
FROM airports  
WHERE city <> 'Москва'  
EXCEPT  
SELECT arrival_city  
FROM routes  
WHERE departure_city = 'Москва'  
ORDER BY city;
```



## Результат

	<input type="text" value="city"/> 	
1	Благовещенск	
2	Иваново	
3	Иркутск	
4	Калуга	
5	Когалым	
6	Комсомольск-на-Амуре	
7	Кызыл	
8	Магадан	
9	Нижнекамск	
10	Новокузнецк	
11	Стрежевой	
12	Сургут	
13	Удачный	
14	Усть-Илимск	
15	Усть-Кут	
16	Ухта	

## Задание 23

Предположим, что департамент развития нашей авиакомпании задался вопросом: каким будет общее число различных маршрутов, которые теоретически можно проложить между всеми городами?

Если в каком-то городе имеется более одного аэропорта, то это учитывать не будем, т. е. маршрутом будем считать путь между *городами*, а не между *аэропортами*. Здесь мы используем соединение таблицы с самой собой на основе неравенства значений атрибутов.

```
SELECT count( * )
  FROM ( SELECT DISTINCT city FROM airports ) AS a1
  JOIN ( SELECT DISTINCT city FROM airports ) AS a2
    ON a1.city <> a2.city;
```

```
count
-----
10100
(1 строка)
```

**Задание.** Перепишите этот запрос с общим табличным выражением.

## Запрос

```
WITH cities AS (
  SELECT DISTINCT city FROM airports
)
SELECT count(*)
FROM cities a1
JOIN cities a2 ON a1.city <> a2.city;
```

## Результат

	count
1	10100