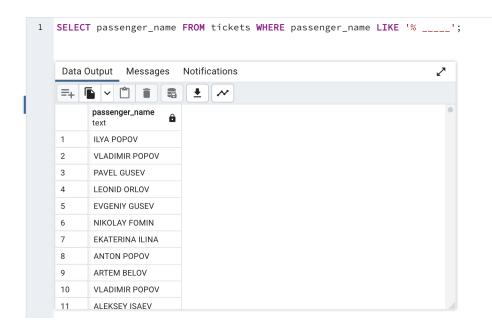
\mathfrak{A} ЗЫК $\mathrm{SQL},\ \mathfrak{Z}3-5$ Бурмашев Григорий, БПМИ-208 $_3$ ноября 2022 г.

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.



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;
```

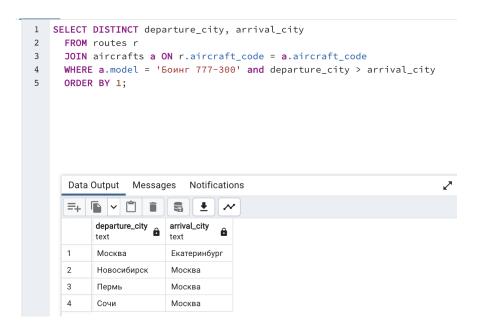
193

Глава 6. Запросы

departure_city	1	arrival_city
	-+-	
Екатеринбург	1	Москва
Москва	-	Екатеринбург
Москва	1	Новосибирск
Москва	1	Пермь
Москва	1	Сочи
Новосибирск	-	Москва
Пермь	1	Москва
Сочи	1	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 a	arrivat_city
Новосибирск М Пермь М	Екатеринбург Иосква Иосква Иосква



В моей бд почему-то aircrafts.model на русском, но это не влияет на решение

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 = 'MockBa'
AND arrival_city = 'CahkT-Петербург';
count
-----
12
(1 строка)
```

194

Контрольные вопросы и задания

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

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	- 1	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)					

198

Контрольные вопросы и задания

А как выявить те направления, на которые не было продано ни одного билета? Один из вариантов решения такой: если на рейсы, отправляющиеся по какомуто 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 Абакан Абакан Абакан Абакан Абакан	Архангельск Грозный Кызыл Москва Новосибирск	

Модифицируйте запрос, приведенный выше.

```
SELECT f.departure_city, f.arrival_city, max( tf.amount ), min( tf.amount )
2
   FROM flights_v f
3 LEFT JOIN ticket_flights tf ON f.flight_id = tf.flight_id
4 GROUP BY 1, 2
5 ORDER BY 1, 2;
    Data Output Messages Notifications
                                                                            ∠
    =+ 6 ~ 1 1 2 4
                                 arrival_city
text
         departure_city
                                                     max numeric
                                                                  min
numeric
         text
    1
          Абакан
                                 Архангельск
                                                             [null]
    2
          Абакан
                                 Грозный
                                                             [null]
    3
          Абакан
                                 Кызыл
                                                             [null]
    4
          Абакан
                                                         101000.00
                                                                    33700.00
                                 Москва
                                                           5800.00
                                                                     5800.00
    5
          Абакан
                                 Новосибирск
```

19.* В разделе 6.4 мы использовали рекурсивный алгоритм в общем табличном выражении. Изучите этот пример, чтобы лучше понять работу рекурсивного алгоритма:

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.

```
WITH RECURSIVE ranges ( min_sum, max_sum, iteration )
1
 2
         AS (
 3
               VALUES( 0,
                                 100000, 1),
                      ( 100000, 200000, 1 ),
 4
                      ( 200000, 300000, 1 )
 5
 6
               UNION ALL
               SELECT min_sum + 100000, max_sum + 100000, iteration + 1
 7
 8
 9
                 WHERE max_sum < ( SELECT max( total_amount ) FROM bookings )</pre>
10
     SELECT * FROM ranges;
11
                                                                                ~
     Data Output
                             Notifications
                 Messages
          min_sum
                                iteration
                     max_sum
          integer
                     integer
                                integer
     1
                  0
                        100000
                                        1
    2
             100000
                        200000
                                        1
                        300000
     3
             200000
                                        1
     4
             100000
                        200000
                                        2
     5
             200000
                        300000
                                        2
     6
             300000
                        400000
                                        2
    7
             200000
                        300000
                                        3
     8
             300000
                        400000
                                        3
     9
             400000
                        500000
                                        3
     10
             300000
                        400000
                                        4
             400000
                        500000
```

При замене на UNION удаляются дупликаты, результатом стало уменьшие rows с 36 до 13:

```
1
     WITH RECURSIVE ranges ( min_sum, max_sum )
 2
 3
               VALUES( 0, 100000 ),
                     ( 100000, 200000 ),
 4
 5
                     ( 200000, 300000 )
 6
 7
              SELECT min_sum + 100000, max_sum + 100000
 8
                 FROM ranges
 9
                 WHERE max_sum < ( SELECT max( total_amount ) FROM bookings )</pre>
10
11
    SELECT * FROM ranges;
                                                                                 ~
       Data Output Messages Notifications
                                      ~
             min_sum
                       max_sum
             integer
                       integer
       1
                    0
                           100000
       2
                100000
                           200000
       3
                200000
                           300000
       4
                300000
                           400000
                           500000
       5
                400000
                500000
                           600000
       6
       7
                600000
                           700000
                           800000
       8
                700000
                           900000
       9
                800000
       10
                900000
                          1000000
               1000000
                          1100000
       11
       12
               1100000
                          1200000
       13
               1200000
                          1300000
```

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 = 'Mockba'
AND r.arrival_city = a.city
)
AND a.city <> 'Mockba'
ORDER BY city;
```

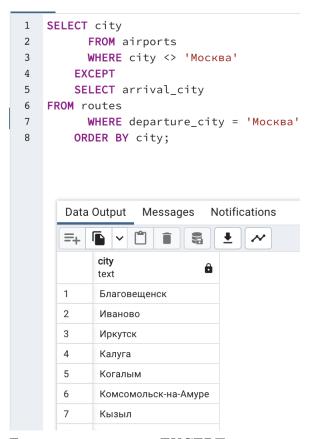
Можно предложить другой вариант, в котором используется одна из операций над множествами строк: объединение, пересечение или разность.

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

203

Глава 6. Запросы

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



Будем использовать EXCEPT, ведь мы хотим исключить оставить все, KPOME Москвы

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;
```

204

Контрольные вопросы и задания

```
count
-----
10100
(1 строка)
```

Задание. Перепишите этот запрос с общим табличным выражением.

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

Data Output Messages Notifications

Count bigint 1 10100
```