

Запросы и подзапросы

Как в общем случае работает команда SELECT (1)



1. Сначала вычисляются все элементы, приведенные в списке после ключевого слова FROM. Если таблиц больше одной, то формируется декартово произведение из множеств их строк. При этом в комбинированных строках сохраняются все атрибуты из каждой исходной таблицы.
2. Если в команде присутствует условие WHERE, то из полученного декартова произведения *исключаются* строки, которые этому условию не соответствуют.
3. Если присутствует предложение GROUP BY, то результирующие строки группируются на основе совпадения значений одного или нескольких атрибутов, а затем вычисляются значения агрегатных функций.
4. Если присутствует предложение HAVING, то оно отфильтровывает результирующие строки (группы), не удовлетворяющие критерию.

Как в общем случае работает команда SELECT (2)

5. Ключевое слово SELECT присутствует всегда.
- Но в списке выражений, идущих после него, могут быть не только простые имена атрибутов, но и их комбинации, созданные с использованием арифметических и других операций, а также вызовы функций.
 - Причем эти функции могут быть не только встроенные, но и созданные пользователем.
 - В списке выражений не обязаны присутствовать все атрибуты, представленные в строках используемых таблиц.
 - Например, атрибуты, на основе которых формируются условия в предложении WHERE, могут отсутствовать в списке выражений после ключевого слова SELECT.
 - Предложение SELECT DISTINCT удаляет дубликаты строк.

Как в общем случае работает команда SELECT (3)



6. Если присутствует предложение ORDER BY, то результирующие строки сортируются на основе значений одного или нескольких атрибутов. По умолчанию сортировка производится по возрастанию значений.
7. Если присутствует предложение LIMIT или OFFSET, то возвращается только подмножество строк из выборки.

Приведенная схема описывает работу команды SELECT *на логическом уровне*, а на уровне реализации запросов в дело вступает **планировщик**, который и формирует план выполнения запроса.

Задача: выбрать все самолеты компании Airbus.

← спецсимвол

```
SELECT * FROM aircrafts WHERE model LIKE 'Airbus%';
```

- Символ «%» имеет специальное значение. Он соответствует *любой последовательности* символов, т. е. вместо него могут быть подставлены:
 - любые символы в любом количестве,
 - а может и не быть подставлено ни одного символа.
- В результате будут выбраны строки, в которых значения атрибута `model` начинаются с символов «Airbus»:

aircraft_code	model	range
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700

(3 строки)

- Шаблон в операторе LIKE всегда покрывает всю анализируемую строку. Поэтому если требуется отыскать некоторую последовательность символов где-то внутри строки, то шаблон должен начинаться и завершаться символом «%».

```
SELECT * FROM aircrafts WHERE model LIKE '%A32%';
```

aircraft_code	model	range
320	Airbus A320-200	5700
321	Airbus A321-200	5600

(2 строки)

- Если по тому столбцу, к которому применяется оператор LIKE, создан индекс для ускорения доступа к данным, то при наличии символа «%» в начале шаблона этот индекс использоваться не будет. В результате может ухудшиться производительность.

- Кроме символа «%» в шаблоне может использоваться и символ «_», который соответствует в точности *одному любому символу*.
- В качестве примера найдем в таблице «Аэропорты» (airports) те из них, которые имеют названия длиной три символа (буквы).
- С этой целью зададим в качестве шаблона строку, состоящую из трех СИМВОЛОВ «_».

```
SELECT * FROM airports WHERE airport_name LIKE '___';
```

```
--[ RECORD 1 ]+-----
```

airport_code		UFA
airport_name		Уфа
city		Уфа
longitude		55.874417
latitude		54.557511
timezone		Asia/Yekaterinburg

3 символа

Если мы захотим узнать, какими самолетами, кроме машин компаний Airbus и Boeing, располагает наша авиакомпания, то придется усложнить условие:

```
SELECT * FROM aircrafts
WHERE model NOT LIKE 'Airbus%' AND
       model NOT LIKE 'Boeing%';
```

aircraft_code	model	range
SU9	Sukhoi SuperJet-100	3000
CN1	Cessna 208 Caravan	1200
CR2	Bombardier CRJ-200	2700

(3 строки)

- Эти операторы имеют больше возможностей, чем оператор LIKE.
- Для того чтобы выбрать, например, самолеты компаний Airbus и Boeing, можно сделать так:

```
SELECT * FROM aircrafts WHERE model ~ '^(A|Boe)';
```

aircraft_code	model	range
773	Boeing 777-300	11100
763	Boeing 767-300	7900
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700
733	Boeing 737-300	4200

(6 строк)

- Оператор ~ ищет совпадение с шаблоном с учетом регистра символов.
- Символ «^» означает, что поиск совпадения будет привязан к началу строки.
- Если же требуется проверить наличие такого символа в составе строки, то перед ним нужно поставить символ обратной косой черты: «\^».
- Выражение в круглых скобках означает альтернативный выбор между значениями, разделяемыми символом «|». Поэтому в выборку попадут значения, начинающиеся либо на «A», либо на «Boe».

альтернативный
выбор

- Для инвертирования смысла оператора ~ нужно перед ним добавить знак «!».
- Пример: найти модели самолетов, которые не завершаются числом 300.

```
SELECT * FROM aircrafts WHERE model !~ '300$';
```

В этом регулярном выражении символ «\$» означает привязку поискового шаблона к концу строки. Если же требуется проверить наличие такого символа в составе строки, то надо сделать так: «\».

aircraft_code	model	range
SU9	Sukhoi SuperJet-100	3000
320	Airbus A320-200	5700
321	Airbus A321-200	5600
319	Airbus A319-100	6700
CN1	Cessna 208 Caravan	1200
CR2	Bombardier CRJ-200	2700

(6 строк)

Задача: найти три самых восточных аэропорта.

Алгоритм: отсортировать строки в таблице по убыванию значений столбца «Долгота» (longitude) и включить в выборку только первые три строки.

```
SELECT airport_name, city, longitude
FROM airports
ORDER BY longitude DESC
LIMIT 3;
```

airport_name	city	longitude
Анадырь	Анадырь	177.741483
Елизово	Петропавловск-Камчатский	158.453669
Магадан	Магадан	150.720439

(3 строки)



Задача: найти еще три аэропорта, которые находятся немного западнее первой тройки, т. е. занимают места с четвертого по шестое.

Алгоритм, использованный в первой задаче, будет дополнен еще одним шагом: нужно *пропустить три первые строки*, прежде чем начать вывод.

```
SELECT airport_name, city, longitude
FROM airports
ORDER BY longitude DESC
LIMIT 3 OFFSET 3;
```

airport_name	city	longitude
Хомутово	Южно-Сахалинск	142.717531
Хурба	Комсомольск-на-Амуре	136.934
Хабаровск-Новый	Хабаровск	135.188361

(3 строки)



Выборка с группировкой

- Получить код и общее количество поставки каждой детали.

```
select SP.PID, sum(SP.Qty) as Total  
from SP  
group by SP.PID;
```

SP			Результат	
SID	PID	Qty		
S1	P5	10	PID	Total
S3	P2	20	P5	30
S2	P4	30	P2	50
S7	P1	40	P4	30
S5	P5	20	P1	120
S5	P3	50	P3	50
S8	P2	30		
S4	P1	80		

Выборка с группировкой

- Получить коды деталей, поставляемых более чем одним поставщиком.

```
select SP.PID  
from SP  
group by SP.PID  
having count(SP.SID)>1;
```

SP			Результат
SID	PID	Qty	
S1	P5	10	PID
S3	P2	20	P5
S2	P4	30	P2
S7	P1	40	P1
S5	P5	20	
S5	P3	50	
S8	P2	30	
S4	P1	80	

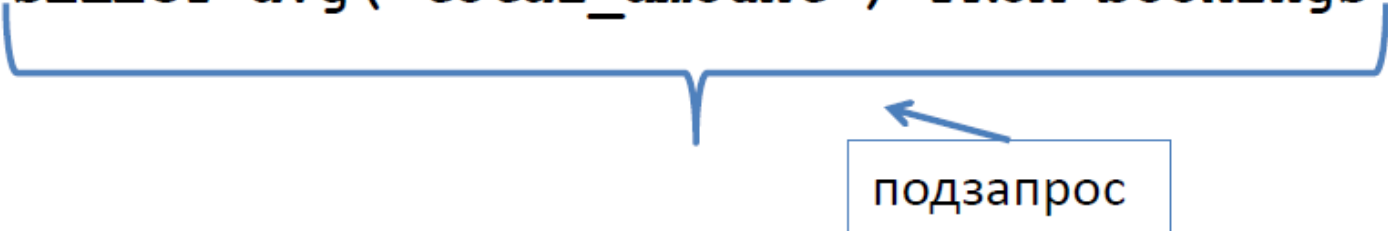
- Предположим, что сотрудникам аналитического отдела потребовалось провести статистическое исследование финансовых результатов работы авиакомпании. В качестве первого шага они решили подсчитать количество операций бронирования, в которых общая сумма превышает среднюю величину по всей выборке.

```
SELECT count( * ) FROM bookings
WHERE total_amount >
      ( SELECT avg( total_amount ) FROM bookings );
```

count

87224

(1 строка)



подзапрос

- Подзапрос является частью более общего запроса. Подзапросы могут присутствовать в предложениях SELECT, FROM, WHERE и HAVING, а также в предложении WITH.
- Здесь используется так называемый **скалярный подзапрос**. В результате его выполнения возвращается *только одно скалярное значение* (один столбец и одна строка).

Вопрос: какие маршруты существуют между городами часового пояса Asia/Krasnoyarsk.

```
SELECT flight_no, departure_city, arrival_city
FROM routes
WHERE departure_city IN ( SELECT city
                           FROM airports
                           WHERE timezone ~ 'Krasnoyarsk'
                         )
      AND arrival_city IN ( SELECT city
                            FROM airports
                            WHERE timezone ~ 'Krasnoyarsk'
                          ) ;
```

Такие подзапросы называются
некоррелированными

- Подзапрос будет выдавать список городов из этого часового пояса, а в предложении WHERE главного запроса с помощью предиката IN будет выполняться проверка на принадлежность города этому списку.
- При этом подзапрос выполняется *только один раз* для всего внешнего запроса, а не при обработке каждой строки из таблицы routes во внешнем запросе, т. к. результат подзапроса не зависит от значений, хранящихся в таблице routes.
- Такие подзапросы называются **некоррелированными**.

flight_no	departure_city	arrival_city
PG0070	Абакан	Томск
PG0071	Томск	Абакан
PG0313	Абакан	Кызыл
PG0314	Кызыл	Абакан
PG0653	Красноярск	Барнаул
PG0654	Барнаул	Красноярск

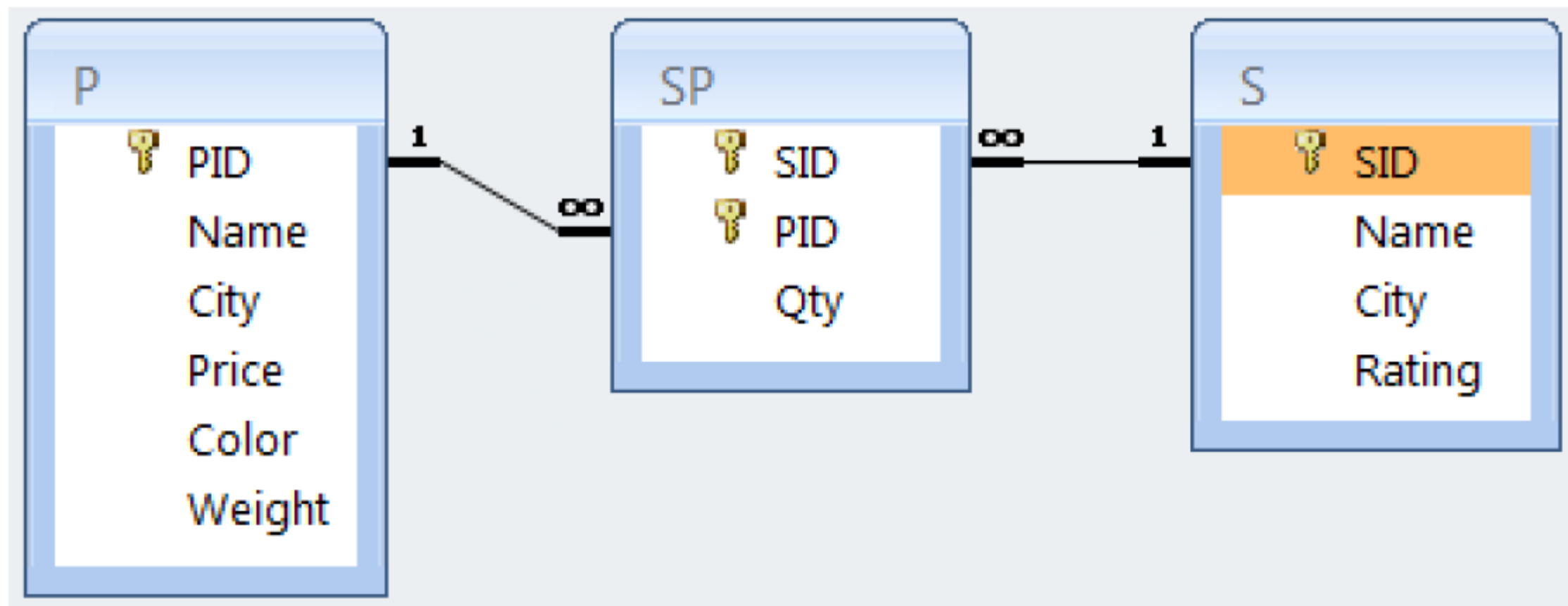
(6 строк)

Модельная база данных

Детали

Поставки

Поставщики



Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name  
from S  
where S.SID in  
  (select SP.SID  
   from SP  
   where SP.PID='P2');
```

S		SP		
SID	Name	SID	PID	Qty
S1	Саспенс +	S1	P5	10
S2	ИТ-монстр	S3	P2	20
S3	Разноимпорт	S2	P2	30
S5	МакроХард	S7	P1	40
S7	Овощторг	S5	P4	50
S8	Разноимпорт	S8	P2	30

*Результат подзапроса –
производное отношение:*

SP.SID
S3
S2
S8



```
select distinct S.Name  
from S  
where S.SID in  
  ({'S3'}, {'S2'}, {'S8'});
```



Результат

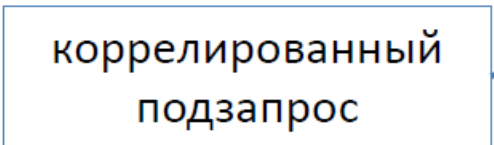
Name
ИТ-монстр
Разноимпорт

Иногда возникают ситуации, когда от подзапроса требуется лишь установить *сам факт наличия или отсутствия строк* в конкретной таблице, удовлетворяющих определённому условию, а непосредственные значения атрибутов в этих строках интереса не представляют. В подобных случаях используют предикат **EXISTS** (или **NOT EXISTS**).

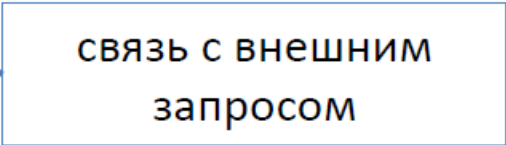
Вопрос: в какие города нет рейсов из Москвы.

```
SELECT DISTINCT a.city      -- требуется DISTINCT
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;
```

коррелированный
подзапрос



связь с внешним
запросом



Для каждой строки (каждого города) из таблицы airports выполняется поиск строки в представлении routes, в которой значение атрибута arrival_city такое же, как в *текущей строке* таблицы airports. Если такой строки не найдено, значит, в этот город маршрута из Москвы нет.

Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name  
from S  
where exists (  
    select *  
    from SP  
    where SP.SID=S.SID and  
        SP.PID='P2');
```

S	
SID	Name
S1	Саспенс +
S2	ИТ-монстр
S3	Разноимпорт
S5	МакроХард
S7	Овощторг
S8	Разноимпорт

SP		
SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P2	30
S7	P1	40
S5	P4	50
S8	P2	30

Результат

Name
ИТ-монстр
Разноимпорт

*Результат подзапроса
вычисляется для каждой
строки из S:*

Для 1-й строки:

SP.SID	SP.PID	SP.Qty
--------	--------	--------

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30

Для 3-й строки: ...



Для 1-й строки:

```
select distinct S.Name  
from S  
where exists (  
    {  
};
```



Результат

Name

*Результат подзапроса
вычисляется для каждой
строки из S:*

Для 1-й строки:

SP.SID	SP.PID	SP.Qty
--------	--------	--------

Для 2-й строки:

SP.SID	SP.PID	SP.Qty
S2	P2	30



Для 2-й строки:

```
select distinct S.Name  
from S  
where exists (  
    {'S2', 'P2', '30'}  
);
```



Результат

Name

ИТ-монстр

Выборка с подзапросом

- Получить имена поставщиков, НЕ поставляющих деталь P2.

```
select distinct S.Name  
from S  
where not exists (  
  select *  
  from SP  
  where SP.SID=S.SID and P.PID='P2');
```

```
select distinct S.Name  
from S  
where S.SID not in (  
  select SP.SID  
  from SP  
  where SP.PID='P2');
```

S

SID	Name
S1	Овощторг
S2	ИТ-монстр
S3	Разноимпорт
S5	МакроХард
S7	Овощторг
S8	Разноимпорт

SP

SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P2	30
S7	P1	40
S5	P4	50
S8	P2	30

Результат

Name
Овощторг
МакроХард

- Получить имена поставщиков, поставляющих деталь P2.

- select distinct S.Name
from S
where 'P2' = any
(
select SP.PID
from SP
where S.SID=SP.SID
);**

S	
SID	Name
S1	Саспенс +
S2	ИТ-монстр
S3	Разноимпорт
S5	МакроХард
S7	Овощторг
S8	Разноимпорт

SP		
SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P2	30
S7	P1	40
S5	P4	50
S8	P2	30

Результат

Name
ИТ-монстр
Разноимпорт

*Результат подзапроса
вычисляется для каждой
строки из S:*

Для 1-й строки:

SP.PID
P5

Для 2-й строки:

SP.PID
P2

Для 3-й строки: ...

*Результат подзапроса
вычисляется для каждой
строки из S:*

Для 1-й строки:

SP.PID
P5

Для 2-й строки:

SP.PID
P2

Для 3-й строки: ...



Для 1-й строки:

```
select distinct S.Name  
from S  
where 'P2' = any (  
    {'P5'}  
);
```



Результат

Name



Для 2-й строки:

```
select distinct S.Name  
from S  
where 'P2' = any (  
    {'P2'}  
);
```



Результат

Name

ИТ-монстр

Выборка с подзапросом

- Получить имена поставщиков, поставляющих деталь P2.

```
select distinct S.Name  
from S  
where 0 <  
(  
  select count(*) as cnt  
from SP  
where S.SID=SP.SID  
and SP.PID='P2'  
);
```

S

SID	Name
S1	Саспенс +
S2	ИТ-монстр
S3	Разноимпорт
S5	МакроХард
S7	Овощторг
S8	Разноимпорт

SP

SID	PID	Qty
S1	P5	10
S3	P2	20
S2	P2	30
S7	P1	40
S5	P4	50
S8	P2	30

Результат

Name
ИТ-монстр
Разноимпорт

*Результат подзапроса
вычисляется для каждой
строки из S:*

Для 1-й строки:

cnt
0

Для 2-й строки:

cnt
1

Для 3-й строки: ...



Для 1-й строки:

```
select distinct S.Name  
from S  
where 0 < ({0});
```



Результат

Name

*Результат подзапроса
вычисляется для каждой
строки из S:*

Для 1-й строки:

cnt
0

Для 2-й строки:

cnt
1

Для 3-й строки: ...



Для 2-й строки:

```
select distinct S.Name  
from S  
where 0 < ({1});
```



Результат

Name

ИТ-монстр

Выборка с подзапросом

- Получить названия деталей, цена которых больше цены любой детали из Челябинска.

```
select distinct P.Name  
from P  
where P.Price > all  
(select P.Price  
from P  
where P.City='Челябинск');
```

Результат подзапроса:

P.Price
30
50
23



```
select distinct P.Name  
from P  
where P.Price > all (  
    {30}, {50}, {23}  
)
```



Результат

Name
Брус
Гайка
Болт

P			
PID	Name	Price	City
P1	Болт	30	Челябинск
P2	Брус	99	Шепетовка
P3	Болт	50	Челябинск
P5	Шуруп	23	Челябинск
P6	Гайка	77	Москва
P7	Болт	88	Копейск

Подзапросы в предложении SELECT

Предположим, что для выработки ценовой политики авиакомпании необходимо знать, как распределяются места разных классов в самолетах всех типов.

```
SELECT a.model,  
      (  
        SELECT count( * )  
        FROM seats s  
        WHERE s.aircraft_code = a.aircraft_code  
              AND s.fare_conditions = 'Business'  
      ) AS business,  
      (  
        SELECT count( * )  
        FROM seats s  
        WHERE s.aircraft_code = a.aircraft_code  
              AND s.fare_conditions = 'Comfort'  
      ) AS comfort,  
      (  
        SELECT count( * )  
        FROM seats s  
        WHERE s.aircraft_code = a.aircraft_code  
              AND s.fare_conditions = 'Economy'  
      ) AS economy  
FROM aircrafts a  
ORDER BY 1;
```

Подзапросы коррелированные

model	business	comfort	economy
Airbus A319-100	20	0	96
Airbus A320-200	20	0	120
Airbus A321-200	28	0	142
Boeing 737-300	12	0	118
Boeing 767-300	30	0	192
Boeing 777-300	30	48	324
Bombardier CRJ-200	0	0	50
Cessna 208 Caravan	0	0	12
Sukhoi SuperJet-100	12	0	85

(9 строк)

Решим ту же **задачу**: для выработки ценовой политики авиакомпании необходимо знать, как распределяются места разных классов в самолетах всех типов.

```
SELECT s2.model,  
       string_agg( s2.fare_conditions || ' (' || s2.num ||  
                  ')', ', ' ,  
                  )
```

агрегатная функция

```
FROM ( SELECT a.model,  
             s.fare_conditions,  
             count( * ) AS num  
       FROM aircrafts a JOIN seats s  
         ON a.aircraft_code = s.aircraft_code  
       GROUP BY 1, 2  
       ORDER BY 1, 2  
     ) AS s2  
GROUP BY s2.model  
ORDER BY s2.model;
```

подзапрос

Агрегатная функция `string_agg` отличается от агрегатных функций `avg`, `min`, `max`, `sum` и `count` тем, что она возвращает не числовое значение, а строку символов, составленную из значений атрибутов, указанных в качестве ее параметров. Эти значения берутся из сгруппированных строк.

model	string_agg
Airbus A319-100	Business (20), Economy (96)
Airbus A320-200	Business (20), Economy (120)
Airbus A321-200	Business (28), Economy (142)
Boeing 737-300	Business (12), Economy (118)
Boeing 767-300	Business (30), Economy (192)
Boeing 777-300	Business (30), Comfort (48), Economy (324)
Bombardier CRJ-200	Economy (50)
Cessna 208 Caravan	Economy (12)
Sukhoi SuperJet-100	Business (12), Economy (85)

(9 строк)

Задача: определить число маршрутов, исходящих из тех аэропортов, которые расположены восточнее географической долготы 150°.

```
SELECT departure_airport, departure_city, count( * )
FROM routes
GROUP BY departure_airport, departure_city
HAVING departure_airport IN ( SELECT airport_code
                               FROM airports
                               WHERE longitude > 150
                             )
ORDER BY count DESC;
```

- Подзапрос формирует список аэропортов, которые и будут отобраны с помощью предложения **HAVING** после выполнения группировки.

departure_airport	departure_city	count
DYR	Анадырь	4
GDX	Магадан	3
PKC	Петропавловск-Камчатский	1

(3 строки)