

1. Выборка из одной таблицы.

1.1 Выбрать из произвольной таблицы данные и отсортировать их по двум произвольным имеющимся в таблице признакам (разные направления сортировки).

1

```
SELECT
    c.client_name ,
    c.client_birth_date ,
    c.client_address
FROM clients c
ORDER BY c.client_birth_date ASC, c.client_name DESC
```

abc client name	client birth date	abc client address
Орлова А. Ф.	1984-06-06	Тутаев, Ленина, 10
Титова П. М.	1987-06-12	Углич, Ленина, 22
Софронова Е. Д.	1987-06-12	Ярославль, Ленина, 40
Филатов И. М.	1993-01-18	Ярославль, Стопани, 10
Колесников Я. П.	1994-03-08	Дубки, Гагарина, 3
Калинин А. Д.	1999-10-09	Углич, Февральская, 28
Крючкова В. А.	2001-11-03	Ярославль, Союзная, 144

Информация о клиентах отсортированных от старших к младшим, а затем в алфавитном порядке

1.2 Выбрать из произвольной таблицы те записи, которые удовлетворяют условию отбора (where). Привести 2-3 запроса.

2

```
SELECT
    c.client_name ,
    c.client_birth_date ,
    c.client_address
FROM clients c
WHERE DATEDIFF(YEAR, c.client_birth_date, GETDATE()) > 25
```

abc client name	client birth date	abc client address
Софронова Е. Д.	1987-06-12	Ярославль, Ленина, 40
Филатов И. М.	1993-01-18	Ярославль, Стопани, 10
Орлова А. Ф.	1984-06-06	Тутаев, Ленина, 10
Титова П. М.	1987-06-12	Углич, Ленина, 22
Колесников Я. П.	1994-03-08	Дубки, Гагарина, 3

Информация о клиентах старше 25

3

```
SELECT
    cl.client_name ,
    cl.client_birth_date ,
    c.contract_date_start ,
    c.contract_date_end
FROM contracts c
JOIN clients cl ON c.client_phone = cl.client_phone
WHERE DAY(cl.client_birth_date) >= DAY(c.contract_date_start)
AND MONTH(cl.client_birth_date) >= MONTH(c.contract_date_start)
AND DAY(cl.client_birth_date) <= DAY(c.contract_date_end)
AND MONTH(cl.client_birth_date) <= MONTH(c.contract_date_end)
```

abc client name	client birth date	contract date start	contract date end
Орлова А. Ф.	1984-06-06	2021-06-05	2021-06-15

Информация о клиентах, празднующих свой день рождения в период поездки

1.3 Привести примеры 2-3 запросов с использованием агрегатных функций (count, max, sum и др.) с группировкой и без группировки.

4

```
SELECT
    s.state_name ,
    COUNT(h.hotel_id)
FROM states s
JOIN hotels h ON s.state_id = h.state_id
GROUP BY s.state_name
```

abc state name	123
Астана	1
Москва	1
Самарканд	1
Санкт-Петербург	1
Фетхие	1
Ярославль	2

Количество отелей в городах, где есть хотя бы один отель

5

```
SELECT
    *
FROM carriers c
WHERE c.carrier_cost = (
    SELECT
        MAX(carrier_cost)
    FROM carriers c2
)
```

Перевозчик с самым дорогим тарифом

abc carrier name	abc carrier INN	123 carrier cost
Аэрофлот	7712040126	19 600

1.4 Привести примеры подведения подытога с использованием GROUP BY [ALL] [CUBE | ROLLUP](2-3 запроса). В ROLLUP и CUBE использовать не менее 2-х столбцов.

6

```
SELECT
    c.country_name ,
    s.state_name ,
    COUNT(h.hotel_id)
FROM countries c
JOIN states s ON c.country_id = s.country_id
JOIN hotels h ON h.state_id = s.state_id
GROUP BY ROLLUP(c.country_name, s.state_name)
```

Информация о количестве отелей по городам и странам

abc country name	abc state name	123
Казахстан	Астана	1
Казахстан	[NULL]	1
Россия	Москва	1
Россия	Санкт-Петербург	1
Россия	Ярославль	2
Россия	[NULL]	4
Турция	Фетхие	1
Турция	[NULL]	1
Узбекистан	Самарканд	1
Узбекистан	[NULL]	1
[NULL]	[NULL]	7

7

```

SELECT
    c.carrier_name ,
    t.transport_type_id ,
    SUM(c.carrier_cost)
FROM carriers c
JOIN transport t ON c.carrier_INN = t.carrier_INN
GROUP BY CUBE(c.carrier_name, t.transport_type_id )

```

Стоимость использования услуг у каждого перевозчика
на каждом предоставляемом им типе транспорта

abc carrier name	123 transport type id	123
Неоплан	[NULL]	15 700
РЖД	[NULL]	9 800
[NULL]	[NULL]	25 500
РЖД	1	9 800
[NULL]	1	9 800
РЖД	3	9 800
[NULL]	3	9 800
Аэрофлот	4	39 200
[NULL]	4	39 200
Аэрофлот	5	19 600
[NULL]	5	19 600
[NULL]	[NULL]	103 900
Аэрофлот	[NULL]	58 800
Неоплан	[NULL]	15 700
РЖД	[NULL]	29 400

1.5 Выбрать из таблиц информацию об объектах, в названиях которых нет заданной последовательности букв (LIKE).

8

```

SELECT
    c.client_name ,
    c.client_address
FROM clients c
WHERE c.client_address NOT LIKE 'Ярославль%'

```

Клиенты не проживающие в Ярославле

abc client name	abc client address
Соболева В. Д.	Данилов, Петербургская, 63
Кулешова П. А.	Тутаев, Герцена, 42
Соловьев В. А.	Ростов, Декабристов, 3
Кузнецова Е. С.	Данилов, Вятская, 17
Орлова А. Ф.	Тутаев, Ленина, 10
Калинин А. Д.	Углич, Февральская, 28
Титова П. М.	Углич, Ленина, 22
Колесников Я. П.	Дубки, Гагарина, 3
Орлова А. Е.	Нерехта, Седова, 5
Иванов Р. Р.	Мышкин, Павлова, 10
Архипова А. М.	Рыбинск, Луначарского, 99
Блинова П. А.	Гаврилов Ям, Клубная, 32
Третьякова М. Ф.	Ростов, Моравского, 18
Некрасова А. Д.	Некрасово, Центральная, 4
Покровская А. А.	Переславль Залесский, Светлая, 5

2. Выборка из нескольких таблиц.

2.1 Вывести информацию подчиненной (дочерней) таблицы, заменяя коды (значения внешних ключей) соответствующими символьными значениями из родительских таблиц. Привести 2-3 запроса с использованием классического подхода соединения таблиц (where).

9

```

SELECT
    c.contract_id ,
    c.contract_total_amount ,
    (SELECT
        t.tour_name
    FROM tours t
    WHERE c.tour_id = t.tour_id)
    AS 'Tour Name'
FROM contracts c

```

вывод не через запрос, а через прямое декартово произведение с фильтром

×

123 contract id	123 contract total amount	abc Tour Name
1	12 500	Каналы Санкт-Петербурга
2	13 000	Каналы Санкт-Петербурга

10

SELECT

```

cd.client_doc_number ,
cd.client_doc_issued_by ,
cd.client_doc_issue_date ,
(SELECT

```

вывод не через запрос, а через прямое декартово произведение с фильтром

✕

```

    c.client_name
FROM clients c
WHERE c.client_phone = cd.client_phone)
AS 'Client Name',
(SELECT
    dt.doc_type_name
FROM doc_types dt
WHERE dt.doc_type_id = cd.doc_type_id)
AS 'Document Type'
FROM client_docs cd

```

ABC client doc number	ABC client doc issued by	client doc issue date	ABC Client Name	ABC Document Type
123456000	[NULL]	[NULL]	Филатов И. М.	Свидетельство о рождении
123456001	[NULL]	[NULL]	Крючкова В. А.	Свидетельство о рождении
7810121314	УМВД РОССИИ	2010-10-05	Софронова Е. Д.	Паспорт
7812121314	УМВД РОССИИ	2012-06-23	Колесников Я. П.	Паспорт
7812121319	УМВД РОССИИ	2011-11-13	Орлова А. Ф.	Паспорт
9019125656	[NULL]	[NULL]	Титова П. М.	Водительские права

2.2. Реализовать запросы пункта 2.1 через внутреннее соединение inner join.

11

SELECT

```

c.contract_id ,
c.contract_total_amount ,
t.tour_name
AS 'Tour Name'

```

```

FROM contracts c
JOIN tours t ON t.tour_id = c.tour_id

```

12

SELECT

```

cd.client_doc_number ,
cd.client_doc_issued_by ,
cd.client_doc_issue_date ,
c.client_name
AS 'Client Name',
dt.doc_type_name
AS 'Document Type'

```

```

FROM client_docs cd
JOIN clients c ON c.client_phone = cd.client_phone
JOIN doc_types dt ON dt.doc_type_id = cd.doc_type_id

```

2.3. Правое внешнее соединение left join. Привести 2-3 запроса

13

```
SELECT
    c.client_name ,
    c.client_phone ,
    cd.client_doc_number
FROM clients c
LEFT JOIN client_docs cd ON c.client_phone = cd.client_phone
```

ABC client name	ABC client phone	ABC client doc number
Софронова Е. Д.	79151236460	7810121314
Филатов И. М.	79151236461	123456000
Крючкова В. А.	79151236462	123456001
Орлова А. Ф.	79151236463	7812121319
Калинин А. Д.	79151236464	[NULL]
Титова П. М.	79151236465	9019125656
Колесников Я. П.	79151236466	7812121314

Вывод всех клиентов, в том числе тех, которые не принесли свои документы

14

```
SELECT
    s.state_name ,
    t.tour_name
FROM states s
LEFT JOIN tours t ON t.state_id = s.state_id
```

	ABC state name	ABC tour name
1	Москва	[NULL]
2	Ярославль	Пешком по золотому кольцу
3	Санкт-Петербург	Каналы Санкт-Петербурга
4	Плес	Золотая осень
5	Астана	[NULL]
6	Караганда	[NULL]

В фирме есть города, в которые еще не организованы туры

2.4. Правое внешнее соединение right join. Привести 2-3 запроса

15

```
SELECT
    DISTINCT dt.doc_type_name
FROM doc_types dt
RIGHT JOIN client_docs cd ON dt.doc_type_id = cd.doc_type_id
```

тут нет смысла использовать правое внешнее соединение. У каждого документа есть тип, внешний ключ должен ссылаться на первичный.

	ABC doc type name
1	Водительские права
2	Паспорт

Вывод только тех типов документов, которые хранит в данный момент фирма.

```

SELECT
    t.transport_state_id ,
    tt.transport_type_name
FROM transport t
RIGHT JOIN transport_types tt ON tt.transport_type_id = t.transport_type_id

```

abc transport state id	abc transport type name
H785TP76	Автобус
[NULL]	Автомобиль
12345000	Поезд
12345001	Бизнес-джет
12345002	Бизнес-джет
12345003	Самолет
[NULL]	Такси

2.5. Привести примеры 2-3 запросов с использованием агрегатных функций и группировки.

```

SELECT
    tt.transport_type_name ,
    COUNT(t.transport_state_id)
FROM transport_types tt
LEFT JOIN transport t ON t.transport_type_id = tt.transport_type_id
GROUP BY tt.transport_type_name

```

	abc transport type name	123
1	Автобус	1
2	Автомобиль	0
3	Бизнес-джет	2
4	Поезд	1
5	Самолет	1
6	Такси	0

Выводит количество единиц каждого типа транспорта, используемого для сотрудничества с тур. фирмой

```

SELECT
    ia.insurance_agent_name,
    IIF(it.insurance_type_name IS NULL,
        'Контракты отсутствуют' ,
        it.insurance_type_name)
    AS 'Insurance type',
    COUNT(i.insurance_contract_id)
    AS 'Contracts count',
    IIF(COUNT(i.insurance_contract_id) = 0,
        0,
        AVG(i.insurance_coverage_amount))
    AS 'Average sum'
FROM insurance_agents ia
LEFT JOIN insurance i ON ia.insurance_agent_INN = i.insurance_agent_INN
LEFT JOIN insurance_types it ON i.insurance_type_id = it.insurance_type_id
GROUP BY ia.insurance_agent_name , it.insurance_type_name

```

abc insurance agent name	abc Insurance type	123 Contracts count	123 Average sum
Тинькофф-Страхование	Контракты отсутствуют	0	0
Альфа-Страхование	Медицинское страхование	2	75 000
Сбер-Страхование	Страхование багажа	2	86 550
Альфа-Страхование	Страхование билетов	1	25 000

Средняя сумма контрактов заключенных страховыми агентами по каждому типу страхования для каждого страхового агента с указанием числа заключенных контрактов.

2.6. Привести примеры 2-3 запросов с использованием группировки и условия отбора групп (Having).

19

```
SELECT
    tt.transport_type_name
FROM carriers c
JOIN transport t ON t.carrier_INN = c.carrier_INN
RIGHT JOIN transport_types tt ON t.transport_type_id = tt.transport_type_id
GROUP BY tt.transport_type_name
HAVING MIN(c.carrier_cost) > 10000
```

ABC	transport type name
	Бизнес-джет
	Самолет

Вывод только тех типов транспорта, минимальная стоимость услуг использования транспорта которых у разных перевозчиков не менее 10000 у.е.

20

```
SELECT
    ia.insurance_agent_name ,
    COUNT(i.insurance_contract_id)
    AS 'Insurances count'
FROM insurance_agents ia
LEFT JOIN insurance i ON ia.insurance_agent_INN = i.insurance_agent_INN
GROUP BY ia.insurance_agent_name
HAVING COUNT(i.insurance_contract_id) > 1
```

ABC	insurance agent name	123	Insurances count
	Альфа-Страхование		3

ABC	insurance agent name	123	Insurances count
	Альфа-Страхование		3
	Сбер-Страхование		1
	Тинькофф-Страхование		1

Вывод страховых агентств, количество оформленных страховых договоров у которых больше 1.

2.7. Привести примеры 3-4 вложенных (соотнесенных, с использованием IN, EXISTS) запросов.

21

```
SELECT
    t.transport_state_id
FROM transport t
WHERE t.transport_state_id NOT IN (
    SELECT
        t.transport_state_id
    FROM contracts c
    JOIN transport t ON c.transport_state_id = t.transport_state_id
)
```

ABC	Весь парк
	12345000
	12345001
	12345002
	12345003
	H785TP76
	C152KB76
	X432BX76

ABC	Запрос
	12345000
	12345001
	12345003
	C152KB76
	X432BX76

Вывод гос. номеров транспорта, который ни разу не использовался для заказа тур. фирмой по существующим контрактам

22

```

SELECT
    t.transport_state_id
FROM transport t
WHERE NOT EXISTS (
    SELECT
        t1.transport_state_id
    FROM contracts c
    JOIN transport t1 ON c.transport_state_id = t1.transport_state_id
    WHERE t.transport_state_id = t1.transport_state_id
)

```

Предыдущий запрос, только через EXISTS

23

```

SELECT
    s.state_name
FROM states s
WHERE NOT EXISTS (
    SELECT
        st.state_name
    FROM hotels h
    JOIN states st ON h.state_id = st.state_id
    WHERE s.state_id = st.state_id
)

```

abc state name
Плес
Караганда
Павлодар
Ташкент
Стамбул
Анкара
Анталья

Города для которых нет информации о расположенных на их территории отелях

3. Представления

3.1 На основе любых запросов из п. 2 создать два представления (VIEW).

24

```

CREATE VIEW HotelsCountInCountryStates AS
SELECT
    c.country_name ,
    s.state_name ,
    COUNT(h.hotel_id)
    AS 'hotels_count'
FROM states s
JOIN countries c ON s.country_id = c.country_id
LEFT JOIN hotels h ON h.state_id = s.state_id
GROUP BY c.country_name , s.state_name

```

	abc country name	abc state name	123 hotels count
4	Казахстан	Караганда	0
5	Россия	Москва	1
5	Казахстан	Павлодар	0
7	Россия	Плес	0
3	Узбекистан	Самарканд	1
9	Россия	Санкт-Петербург	1
10	Турция	Стамбул	0
11	Узбекистан	Ташкент	0
12	Турция	Фетхие	1
13	Россия	Ярославль	2

25

```

CREATE VIEW TransportCountInPark AS
SELECT
    tt.transport_type_name ,
    COUNT(t.transport_state_id)
    AS 'transport_count'
FROM transport_types tt
LEFT JOIN transport t ON t.transport_type_id = tt.transport_type_id
GROUP BY tt.transport_type_name

```

ABC transport type name	123 transport count
Автобус	1
Автомобиль	0
Бизнес-джет	2
Поезд	1
Самолет	1
Такси	0

Для view приведите пример использования.



3.2 Привести примеры использования общетабличных представлений (CTE) (2-3 запроса)

26

```

WITH CTE AS (
    SELECT
        cd.client_doc_number ,
        cd.client_doc_issued_by ,
        cd.client_doc_issue_date ,
        c.client_name,
        dt.doc_type_id
    FROM client_docs cd
    JOIN clients c ON c.client_phone = cd.client_phone
    JOIN doc_types dt ON dt.doc_type_id = cd.doc_type_id
)
SELECT
    CTE.client_name,
    CTE.client_doc_number
FROM CTE
WHERE CTE.doc_type_id = 3

```

ABC client name	ABC client doc number
Филатов И. М.	123456000
Титова П. М.	9019125656

Информация о документах клиентов
которые используют водительские права

27

```

WITH CST AS (
    SELECT
        c.country_name ,
        s.state_name ,
        t.tour_name ,
        t.tour_cost ,
        t.tour_food_type
    FROM states s
    JOIN countries c ON s.country_id = c.country_id
    JOIN tours t ON t.state_id = s.state_id
)
SELECT
    *
FROM CST
WHERE CST.tour_cost > 6000 AND CST.tour_food_type = 'AI'

```

ABC country name	ABC state name	ABC tour name	123 tour cost	ABC tour food type
Россия	Санкт-Петербург	Каналы Санкт-Петербурга	9 800	AI
Турция	Фетхие	7 дней в Фетхие	56 000	AI

Выбор информации о турах и их месте проведения, стоимостью выше 5000 у.е. и
типом питания «Все включено»

4. Функции ранжирования

4.1 Привести примеры 3-4 запросов с использованием ROW_NUMBER, RANK, DENSE_RANK (с PARTITION BY и без)

28

```
SELECT
  ROW_NUMBER()
  OVER (ORDER BY c.carrier_cost)
  AS 'N',
  c.*
FROM carriers c
```

123 N	abc carrier name	abc carrier INN	123 carrier cost
1	Победа	3243433434	5 700
2	Аврора	7712043444	8 000
3	РЖД	7708503727	9 800
4	Автобусы 76	123456789123	12 500
5	Неоплан	123456789124	15 700
6	S7	2342325345	18 900
7	Аэрофлот	7712040126	19 600

Все перевозчики пронумерованные по возрастанию стоимости услуг

29

```
SELECT
  t.tour_name ,
  RANK() OVER(
    PARTITION BY
      t.tour_food_type
    ORDER BY t.tour_cost)
  AS rnk,
  t.tour_food_type ,
  t.tour_cost
FROM tours t
```

abc tour name	123 rnk	abc tour food type	123 tour cost
Каналы Санкт-Петербурга	1	AI	9 800
7 дней в Фетхие	2	AI	56 000
Тур в Самарканд	1	FB	16 800
Новый год в Анапе	2	FB	18 200
Зимний Таганай	3	FB	31 000
Велопоход на Кольский	4	FB	37 000
Горнолыжный тур Германия	1	HB	95 110
Пешком по золотому кольцу	1	RO	5 200
Шавлинские озера	2	RO	11 000
Поход в Ергаки	3	RO	14 300
Неделя на Тагане	4	RO	16 000
На коньках по Байкалу	5	RO	18 100
Ликийская тропа	6	RO	27 000
Золотая осень	1	UAI	2 500
Рождество в Нюрнберге	2	UAI	45 000
Восхождение на Олимп	3	UAI	50 000

Разделение и ранжирование туров на группы по типу питания в порядке возрастания стоимости

30

```
SELECT
  t.tour_name ,
  RANK()
  OVER(ORDER BY
    t.tour_food_type)
  AS rnk,
  DENSE_RANK()
  OVER(ORDER BY
    t.tour_food_type)
  AS d_rnk,
  t.tour_food_type ,
  t.tour_cost
FROM tours t
```

abc tour name	123 rnk	123 d_rnk	abc tour food type	123 tour cost
Каналы Санкт-Петербурга	1	1	AI	9 800
7 дней в Фетхие	1	1	AI	56 000
Новый год в Анапе	3	2	FB	18 200
Тур в Самарканд	3	2	FB	16 800
Велопоход на Кольский	3	2	FB	37 000
Зимний Таганай	3	2	FB	31 000
Горнолыжный тур Германия	7	3	HB	95 110
На коньках по Байкалу	8	4	RO	18 100
Поход в Ергаки	8	4	RO	14 300
Неделя на Тагане	8	4	RO	16 000
Пешком по золотому кольцу	8	4	RO	5 200
Шавлинские озера	8	4	RO	11 000
Ликийская тропа	8	4	RO	27 000
Восхождение на Олимп	14	5	UAI	50 000
Золотая осень	14	5	UAI	2 500
Рождество в Нюрнберге	14	5	UAI	45 000

«Плотное» и «неплотное» ранжирование туров по типу питания.

5. Объединение, пересечение, разность

5.1 Привести примеры 3-4 запросов с использованием UNION / UNION ALL, EXCEPT, INTERSECT. Данные в одном из запросов отсортируйте по произвольному признаку.

31

```
SELECT
    s.state_name
FROM states s
EXCEPT
SELECT
    s2.state_name
FROM tours t
JOIN states s2 ON t.state_id = s2.state_id
ORDER BY s.state_name
```

Все города, в которые пока не существует туров

abc state name
Анкара
Астана
Бонн
Горно-Алтайск
Екатеринбург
Караганда
Лариса
Москва
Мурманск
Мюнхен
Павлодар
Салоники
Стамбул
Ташкент
Троисдорф
Шушенское

32

```
SELECT
    cl.client_name
FROM contracts c
JOIN clients cl ON c.client_phone = cl.client_phone
JOIN tours t ON c.tour_id = t.tour_id
WHERE t.tour_name = '7 дней в Фетхие'
INTERSECT
SELECT
    cl.client_name
FROM contracts c
JOIN clients cl ON c.client_phone = cl.client_phone
JOIN tours t ON c.tour_id = t.tour_id
WHERE t.tour_name = 'Рождество в Нюрнберге'
```

*Среди клиентов из 2ух разных туров
найти тех, кто едет в оба*

abc client name
Орлова А. Ф.

33

```
SELECT
    cl.client_name ,
    c.contract_date_start ,
    c.contract_date_end ,
    'Февраль' AS tour_month
FROM contracts c
JOIN clients cl ON c.client_phone = cl.client_phone
WHERE MONTH(c.contract_date_start) = 2 OR MONTH(c.contract_date_end) = 2
UNION
SELECT
    cl.client_name ,
    c.contract_date_start ,
    c.contract_date_end ,
    'Сентябрь' AS tour_month
FROM contracts c
JOIN clients cl ON c.client_phone = cl.client_phone
WHERE MONTH(c.contract_date_start) = 9 OR MONTH(c.contract_date_end) = 9
```

abc client name	contract date start	contract date end	abc tour month
Архипова А. М.	2022-09-22	2022-09-29	Сентябрь
Малышев Д. М.	2022-01-27	2022-02-04	Февраль
Орлова А. Е.	2022-09-19	2022-09-28	Сентябрь
Орлова А. Ф.	2021-02-03	2021-02-10	Февраль
Панов А. Т.	2022-01-17	2022-02-27	Февраль
Тарасова А. Т.	2022-08-26	2022-09-09	Сентябрь
Третьякова М. Ф.	2022-09-02	2022-09-11	Сентябрь

*Объединить клиентов, у которых
тур выпадает на февраль или
сентябрь*

6. Использование CASE, PIVOT и UNPIVOT.

6.1 Привести примеры получения сводных (итоговых) таблиц с использованием CASE

34

```
SELECT
  t.tour_name ,
  CASE
    WHEN t.tour_cost > 30000
    THEN 'Дорогой'
    WHEN t.tour_cost > 10000
    THEN 'Средний'
    WHEN t.tour_cost > 0
    THEN 'Бюджетный'
  END TourCostType,
  t.tour_cost
FROM tours t
```

Разделение туров на категории по стоимости

ABC	tour name	ABC	TourCostType	123	tour cost
	Пешком по золотому кольцу		Бюджетный		5 200
	Каналы Санкт-Петербурга		Бюджетный		9 800
	Золотая осень		Бюджетный		2 500
	Тур в Самарканд		Средний		16 800
	7 дней в Фетхие		Дорогой		56 000
	Ликийская тропа		Средний		27 000
	Новый год в Анапе		Средний		18 200
	Шавлинские озера		Средний		11 000
	Восхождение на Олимп		Дорогой		50 000
0	Поход в Ергаки		Средний		14 300
1	Неделя на Тагане		Средний		16 000
2	Велопоход на Кольский		Дорогой		37 000
3	Рождество в Нюрнберге		Дорогой		45 000
4	Горнолыжный тур Германия		Дорогой		95 110
5	Зимний Таганай		Дорогой		31 000
5	На коньках по Байкалу		Средний		18 100

35

```
WITH CTE AS (
  SELECT
    CASE
      WHEN t.tour_cost > 30000
      THEN 'Дорогой'
      WHEN t.tour_cost > 10000
      THEN 'Средний'
      WHEN t.tour_cost > 0
      THEN 'Бюджетный'
    END AS TourCostType,
    t.tour_food_type
  FROM tours t
)
SELECT
  CTE.TourCostType,
  SUM(CASE CTE.tour_food_type WHEN 'RO' THEN 1 ELSE 0 END) RO,
  SUM(CASE CTE.tour_food_type WHEN 'BB' THEN 1 ELSE 0 END) BB,
  SUM(CASE CTE.tour_food_type WHEN 'HB' THEN 1 ELSE 0 END) HB,
  SUM(CASE CTE.tour_food_type WHEN 'FB' THEN 1 ELSE 0 END) FB,
  SUM(CASE CTE.tour_food_type WHEN 'AI' THEN 1 ELSE 0 END) AI,
  SUM(CASE CTE.tour_food_type WHEN 'UAI' THEN 1 ELSE 0 END) UAI
FROM CTE
GROUP BY CTE.TourCostType
```

Вывод информации по категориям цен по стоимости с количеством туров по каждому типу питания в категории

Типы питания:
RO — без питания
BB — завтрак
HB — завтрак, ужин
FB — завтрак, обед, ужин
AI — все включено
UAI — ультра все включено

ABC	tour cost type	123	RO	123	BB	123	HB	123	FB	123	AI	123	UAI
	Бюджетный		1		0		0		0		1		1
	Дорогой		0		0		1		2		1		2
	Средний		5		0		0		2		0		0

36

```
SELECT
*
FROM (
  SELECT
    t.tour_food_type ,
    t.tour_id,
    CASE
      WHEN t.tour_cost > 30000
      THEN 'Дорогой'
      WHEN t.tour_cost > 10000
      THEN 'Средний'
      WHEN t.tour_cost > 0
      THEN 'Бюджетный'
    END AS tour_cost_type
  FROM tours t
) AS FoodTypesAndCosts
PIVOT (
  COUNT(tour_id)
  FOR tour_food_type
  IN ([RO], [BB], [HB], [FB], [AI], [UAI])
) AS pivot_table
```

Запрос 6.1.2, измененный через PIVOT

abc tour cost type	123 RO	123 BB	123 HB	123 FB	123 AI	123 UAI
Бюджетный	1	0	0	0	1	1
Дорогой	0	0	1	2	1	2
Средний	5	0	0	2	0	0

37

На основе предыдущего запроса создано VIEW

```
SELECT
  unpivot_table.tour_cost_type,
  unpivot_table.food_type,
  unpivot_table.tour_count
FROM (
  SELECT
    t.tour_cost_type,
    t.RO, t.BB, t.HB, t.FB, t.AI, t.UAI
  FROM TourCostTypeFoodType t
) AS TourCostTypeFoodType
UNPIVOT (
  tour_count
  FOR food_type
  IN ([RO], [BB], [HB], [FB], [AI], [UAI])
) AS unpivot_table
ORDER BY unpivot_table.tour_count DESC
```

Количество туров в каждой категории по стоимости с учетом типа питания, отсортированные по убыванию их количества.

abc tour cost	123 fo	123 to
Средний	RO	5
Средний	FB	2
Дорогой	FB	2
Дорогой	UAI	2
Дорогой	HB	1
Дорогой	AI	1
Бюджетный	RO	1
Бюджетный	AI	1
Бюджетный	UAI	1
Дорогой	RO	0
Дорогой	BB	0
Бюджетный	BB	0
Бюджетный	HB	0
Бюджетный	FB	0
Средний	AI	0
Средний	UAI	0
Средний	BB	0
Средний	HB	0