```
19        ('Volkswagen', 21600);
20
21   ● ⊖ CREATE VIEW V_CARS AS (
22            SELECT *
23            FROM Cars
24            WHERE cost < 25000
25      └ );
26
27   ●    SELECT * FROM V_CARS;
28
29        -- 2.   Изменить в существующем представлении
30        -- (используя оператор ALTER VIEW)
```

⇕ | 22:27

Result Grid | ▦ ↬ Filter Rows: 🔍 Search | Export:

id	Name	Cost
▶ 3	Skoda	9000
4	Volvo	29000
6	Citroen	21000
8	Volkswagen	21600

```
31
32   ● ⊖ ALTER VIEW V_CARS AS (
33            SELECT *
34            FROM Cars
35            WHERE cost < 30000
36      └ );
37   ●    SELECT * FROM V_CARS;
38
39        -- 3.   Создайте представление, в котором будут толь
40   ● ⊖ CREATE VIEW V_CARS_SK_AU AS (
41            SELECT *
42            FROM Cars
43            WHERE Name in ('Audi','Skoda')
44      └ );
45   ●    SELECT * FROM V_CARS_SK_AU;
46
```

⇕ | 22:37

Result Grid | ▦ ↬ Filter Rows: 🔍 Search | Export: 🖫

id	Name	Cost
▶ 3	Skoda	9000
4	Volvo	29000
6	Citroen	21000
8	Volkswagen	21600

```sql
39      -- 3.  Создайте представление, в котором будут только автомобили марки "Шкод
40      CREATE VIEW V_CARS_SK_AU AS (
41          SELECT *
42          FROM Cars
43          WHERE Name in ('Audi','Skoda')
44      );
45      SELECT * FROM V_CARS_SK_AU;
46
47      -- Задача 4 Вывести название и цену для всех анализов, которые продавались 5
48
49      CREATE TABLE Analysis (
50          an_id INT AUTO_INCREMENT PRIMARY KEY,
51          an_name VARCHAR(50)
```

`28:45`

id	Name	Cost
▶ 1	Audi	52642
3	Skoda	9000

```sql
100
101     SELECT a.an_name, a.an_cost, a.an_price,o.ord_datetime,WEEK(o.ord_datetime,1) as week_number,  DAYNAME(o.ord_datetime) as week_name
102     FROM    Analysis a
103     INNER JOIN Orders o ON a.an_id = o.ord_an
104     WHERE (YEAR(ord_datetime)=2020 and MONTH(ord_datetime)= 2 and DAY(ord_datetime) = 5)
105     OR  (WEEK(ord_datetime,1) = WEEK('2020-02-05',1)+1)
106     ORDER BY o.ord_datetime;
107
108     -- Задача 5
109     CREATE TABLE Station (
110         id INT AUTO_INCREMENT PRIMARY KEY,
111         train_id INT
```

`25:106`

an_name	an_cost	an_price	ord_datetime	week_number	week_name
▶ Кровь с вены	67	223	2020-02-05 12:00:00	6	Wednesday
УЗИ почек	72	203	2020-02-05 14:00:00	6	Wednesday
Mult gest-plac/sac NOS	51	202	2020-02-10 06:00:00	7	Monday
УЗИ почек	72	203	2020-02-11 11:00:00	7	Tuesday
КТ вн органов	81	162	2020-02-11 15:00:00	7	Tuesday

```sql
123
124     SELECT *,
125     CASE WHEN
126         TIMEDIFF(LEAD(station_time, 1, '') OVER (PARTITION
127     ELSE
128         TIME_FORMAT(TIMEDIFF(LEAD(station_time, 1, '') OVER
129     END as 'time_to_the_next_station'
130     FROM Station;
131
```

`18:108`

id	train_id	station	station_ti...	time_to_the_next_stat...
▶ 1	110	San Francisko	10:00:00	00:54:00
2	110	Redwood City	10:54:00	00:08:00
3	110	Palo Alto	11:02:00	01:33:00
4	110	San Jose	12:35:00	
5	120	San Francisko	11:00:00	01:49:00
6	120	Palo Alto	12:49:00	00:41:00
7	120	San Jose	13:30:00	