

SQL. Семинар 5. Домашняя работа

Задание 1.

29 • **SELECT * FROM cars;**

<

Result Grid | Filter Rows:

	Id	Name	Cost
▶	1	Audi	52642
	2	Mercedes	57127
	3	Skoda	9000
	4	Volvo	29000
	5	Bentley	350000
	6	Citroen	21000
	7	Hummer	41400
	8	Volkswagen	41400

1.1 Создайте представление, в которое попадут автомобили стоимостью до 25000 долларов.

```
CREATE OR REPLACE VIEW cars_view_1 AS
SELECT Name, Cost FROM cars
WHERE Cost < 25000
ORDER BY Name;
SELECT * FROM cars_view_1;
```

35 • **CREATE OR REPLACE VIEW cars_view_1 AS**
36 **SELECT Name, Cost FROM cars**
37 **WHERE Cost < 25000**
38 **ORDER BY Name;**
39 • **SELECT * FROM cars_view_1;**

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Name	Cost
▶	Citroen	21000
	Skoda	9000

1.2 Изменить в существующем представлении порог для стоимости: пусть цена будет до 30 000 долларов (используя оператор ALTER VIEW).

```
ALTER VIEW cars_view_1 AS
SELECT Name, Cost FROM cars
WHERE Cost < 30000
ORDER BY Name;
SELECT * FROM cars_view_1;
```

46 • **ALTER VIEW cars_view_1 AS**
47 **SELECT Name, Cost FROM cars**
48 **WHERE Cost < 30000**
49 **ORDER BY Name;**
50 • **SELECT * FROM cars_view_1;**

<

Result Grid | Filter Rows: | Export:

	Name	Cost
▶	Citroen	21000
	Skoda	9000
	Volvo	29000

1.3 Создайте представление, в котором будут только автомобили марки “Шкода” и “Ауди”

```
CREATE OR REPLACE VIEW cars_view_2 AS
SELECT * FROM cars
WHERE Name = 'Skoda' OR Name = 'Audi'
ORDER BY Name;
SELECT * FROM cars_view_2;
```

56 • CREATE OR REPLACE VIEW cars_view_2 AS
57 SELECT * FROM cars
58 WHERE Name = 'Skoda' OR Name = 'Audi'
59 ORDER BY Name;
60 • SELECT * FROM cars_view_2;

Result Grid | Filter Rows: | Export: | Wrap Cell

	Id	Name	Cost
▶	1	Audi	52642
	3	Skoda	9000

Задание 2.

116 • SELECT * FROM analysis;

Result Grid | Filter Rows: | Export: | Wrap Cell

	an_id	an_name	an_cost	an_price	an_group
▶	101	Общий анализ крови	300	450	1
	102	Биохимический анализ крови	500	750	1
	103	Анализ крови на сахар	400	600	1
	104	Анализ крови на гормоны	600	900	1
	105	Иммунологический анализ крови	1000	1500	1
	201	Общий анализ мочи	300	450	2
	301	Исследования кала	400	600	3

122 • SELECT * FROM analysis_groups;

Result Grid | Filter Rows: | Export: |

	gr_id	gr_name	gr_temp
▶	1	Анализ крови	10
	2	Анализ мочи	10
	3	Анализ кала	10

140 • SELECT * FROM analysis_orders;

1 1 1
Result Grid | Filter Rows: | Edit: |

	ord_id	ord_datetime	ord_an
▶	1	2020-02-01 10:00:00	101
	2	2020-02-02 11:00:00	102
	3	2020-02-03 10:00:00	103
	4	2020-02-04 11:30:00	104
	5	2020-02-05 09:30:00	105
	6	2020-02-06 09:00:00	201
	7	2020-02-07 08:30:00	301
	8	2020-02-08 08:40:00	101
	9	2020-02-09 11:00:00	102
	10	2020-02-10 10:00:00	103
	11	2020-02-11 09:00:00	104
	12	2020-02-12 09:30:00	105
	13	2020-02-13 09:00:00	201
	14	2020-02-14 08:30:00	301
	15	2020-02-15 10:00:00	101

Вывести название и цену для всех анализов, которые продавались 5 февраля 2020 и всю следующую неделю.

```
SELECT analysis_orders.ord_datetime, analysis.an_name, analysis.an_price
FROM analysis
JOIN analysis_orders ON analysis.an_id = analysis_orders.ord_an
WHERE analysis_orders.ord_datetime BETWEEN '2020-02-05' AND '2020-02-13'
ORDER BY analysis_orders.ord_datetime;
```

```
144 • SELECT analysis_orders.ord_datetime, analysis.an_name, analysis.an_price
145 FROM analysis
146 JOIN analysis_orders ON analysis.an_id = analysis_orders.ord_an
147 WHERE analysis_orders.ord_datetime BETWEEN '2020-02-05' AND '2020-02-13'
148 ORDER BY analysis_orders.ord_datetime;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	ord_datetime	an_name	an_price
▶	2020-02-05 09:30:00	Иммунологический анализ крови	1500
	2020-02-06 09:00:00	Общий анализ мочи	450
	2020-02-07 08:30:00	Исследования кала	600
	2020-02-08 08:40:00	Общий анализ крови	450
	2020-02-09 11:00:00	Биохимический анализ крови	750
	2020-02-10 10:00:00	Анализ крови на сахар	600
	2020-02-11 09:00:00	Анализ крови на гормоны	900
	2020-02-12 09:30:00	Иммунологический анализ крови	1500

Задание 3.

Добавьте новый столбец под названием «время до следующей станции».

```
168 • SELECT * FROM train_tbl;
```

Result Grid | Filter Rows: |

	train_id	station	station_time
▶	110	San Francisco	10:00:00
	110	Redwood City	10:54:00
	110	Palo Alto	11:02:00
	110	San Jose	12:35:00
	120	San Francisco	11:00:00
	120	Palo Alto	12:49:00
	120	San Jose	13:30:00

```
SELECT *,
TIMEDIFF(LEAD(station_time,1) OVER(PARTITION BY train_id), station_time) AS
'time_to_next_station'
FROM train_tbl;
```

```
171 • SELECT *,
172 TIMEDIFF(LEAD(station_time,1) OVER(PARTITION BY train_id), station_time) AS 'time_to_next_station'
173 FROM train_tbl;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	train_id	station	station_time	time_to_next_station
▶	110	San Francisco	10:00:00	00:54:00
	110	Redwood City	10:54:00	00:08:00
	110	Palo Alto	11:02:00	01:33:00
	110	San Jose	12:35:00	NULL
	120	San Francisco	11:00:00	01:49:00
	120	Palo Alto	12:49:00	00:41:00
	120	San Jose	13:30:00	NULL