

```

CREATE TABLE Groups (
  gr_id INT PRIMARY KEY,
  gr_name VARCHAR(255),
  gr_temp VARCHAR(255)
);

CREATE TABLE Analysis (
  an_id INT PRIMARY KEY,
  an_name VARCHAR(255),
  an_cost DECIMAL(10,2),
  an_price DECIMAL(10,2),
  an_group INT,
  FOREIGN KEY (an_group) REFERENCES Groups(gr_id)
);

CREATE TABLE Orders (
  ord_id INT PRIMARY KEY,
  ord_datetime DATETIME,
  ord_an INT,
  FOREIGN KEY (ord_an) REFERENCES Analysis(an_id)
);

```

```

INSERT INTO Groups (gr_id, gr_name, gr_temp) VALUES
(1, 'Group 1', 'temp1'),
(2, 'Group 2', 'temp2'),
(3, 'Group 3', 'temp3');

INSERT INTO Analysis (an_id, an_name, an_cost, an_price, an_group) VALUES
(1, 'Analysis 1', 100.00, 200.00, 1),
(2, 'Analysis 2', 150.00, 300.00, 2),
(3, 'Analysis 3', 200.00, 400.00, 2),
(4, 'Analysis 4', 50.00, 100.00, 3);

INSERT INTO Orders (ord_id, ord_datetime, ord_an) VALUES
(1, '2020-02-06 10:00:00', 1),
(2, '2020-02-07 11:00:00', 2),
(3, '2020-02-08 12:00:00', 3),
(4, '2020-02-16 13:00:00', 4),
(5, '2022-01-04 13:00:00', 4);

INSERT INTO Orders (ord_id, ord_datetime, ord_an) VALUES
(6, '2022-02-01 10:00:00', 1),
(7, '2022-03-02 11:00:00', 2),
(8, '2022-04-03 12:00:00', 3),
(9, '2022-01-04 13:00:00', 4);

INSERT INTO Orders (ord_id, ord_datetime, ord_an) VALUES
(10, '2022-01-11 10:00:00', 1),
(11, '2022-01-12 11:00:00', 2),
(12, '2022-01-13 12:00:00', 3),
(13, '2022-01-14 13:00:00', 4);

```

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

```

SELECT an_name, an_price
FROM Analysis
INNER JOIN Orders ON Analysis.an_id = Orders.ord_an
WHERE ord_datetime >= '2020-02-05' AND ord_datetime <= '2020-02-12'

```

83 %

Results Messages

	an_name	an_price
1	Analysis 1	200.00
2	Analysis 2	300.00
3	Analysis 3	400.00

2. Нарастающим итогом рассчитать, как увеличивалось количество проданных тестов каждый месяц каждого года с разбивкой по группе.

SQLQuery2.sql D...adim_y9xms2s (76)*

```

SELECT
  an_group,
  FORMAT(ord_datetime, 'yyyy-MM-01') AS month,
  YEAR(ord_datetime) AS year,
  COUNT(ord_id) AS sold_tests
FROM
  Orders
JOIN Analysis ON Orders.ord_an = Analysis.an_id
GROUP BY
  an_group, FORMAT(ord_datetime, 'yyyy-MM-01'), YEAR(ord_datetime)
ORDER BY
  an_group, year, month;

```

83 %

Results Messages

	an_group	month	year	sold_tests
1	1	2020-02-01	2020	1
2	1	2022-01-01	2022	1
3	1	2022-02-01	2022	1
4	2	2020-02-01	2020	2
5	2	2022-01-01	2022	2
6	2	2022-03-01	2022	1
7	2	2022-04-01	2022	1
8	3	2020-02-01	2020	1
9	3	2022-01-01	2022	3