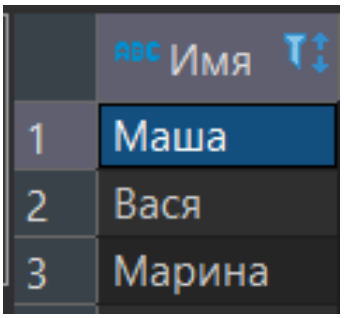
Задача 1

Листинг для первого способа:

```
select name as 'Имя' from test
order by age
limit 3
```

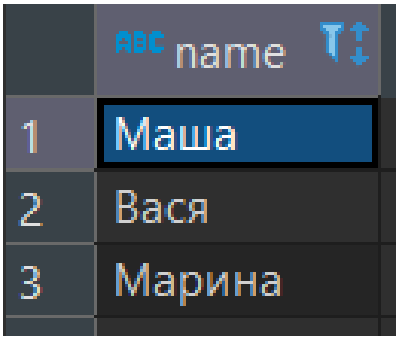


	Имя
1	Маша
2	Вася
3	Марина

Рисунок 1 – Результат работы кода

Листинг для 2 способа:

```
select name from test as v
left join (select name as name1 from test
order by age desc
limit 2) as v1
on v.name = v1.name1
where v1.name1 is null
order by v.age
```



	name
1	Маша
2	Вася
3	Марина

Рисунок 2 – Результат работы кода

Задача 2

Листинг:

```
select abonent, region_id , max(dttm) from zadanie2
GROUP BY abonent, date_format(dttm , '%Y%m%d' )
ORDER BY dttm
```

	abonent	region_id	max(dttm)
1	7072110988	32 722	2021-08-18 14:15:00
2	7072110988	32 722	2021-08-19 09:00:00
3	7071107101	12 533	2021-08-19 09:27:00

Рисунок 3 – Результат работы кода

Задача 3

Заполненность таблицы item_prices:

	item_id	item_name	item_price	created_dttm
1	1	Книга	150	2019-09-01 00:00:01
2	3	Кукла	300	2019-09-08 00:00:01
3	2	Мяч	200	2019-10-01 00:00:01
4	4	Лего	100	2019-11-17 00:00:01
5	3	Кукла	325	2019-11-24 00:00:01
6	1	Книга	100	2019-12-12 00:00:01
7	4	Лего	400	2020-01-09 00:00:01
8	2	Мяч	180	2020-05-13 00:00:01
9	4	Лего	305	2020-05-15 00:00:01
10	3	Кукла	415	2020-06-06 00:00:01
11	1	Книга	285	2021-01-03 00:00:01
12	4	Лего	300	2021-03-19 00:00:01
13	1	Книга	500	2022-02-02 00:00:01
14	4	Лего	305	2022-08-16 00:00:01

Рисунок 4 - Скриншот данных из таблицы item_prices

Листинг:

```
create table dict_item_prices (
	item_id INT,
	item_name VARCHAR(150),
	item_price INT,
	valid_from_dt DATE,
	valid_to_dt DATE
);
insert into dict_item_prices (item_id,item_name,item_price,valid_from_dt)
select item_id,item_name,item_price,DATE(created_dttm) from item_prices;
update dict_item_prices
set valid_to_dt = (select date(created_dttm)-1 from item_prices where
item_prices.item_id = dict_item_prices.item_id and
date(created_dttm) > dict_item_prices.valid_from_dt
order by created_dttm limit 1);
update dict_item_prices
set valid_to_dt = '9999-12-31'
```

```
where valid_to_dt IS null
```

	item_id	item_name	item_price	valid_from_dt	valid_to_dt
1	1	Книга	150	2019-09-01	2019-12-11
2	1	Книга	100	2019-12-12	2021-01-02
3	1	Книга	285	2021-01-03	2022-02-01
4	1	Книга	500	2022-02-02	9999-12-31
5	2	Мяч	200	2019-10-01	2020-05-12
6	2	Мяч	180	2020-05-13	9999-12-31
7	3	Кукла	300	2019-09-08	2019-11-23
8	3	Кукла	325	2019-11-24	2020-06-05
9	3	Кукла	415	2020-06-06	9999-12-31
10	4	Лего	100	2019-11-17	2020-01-08
11	4	Лего	400	2020-01-09	2020-05-14
12	4	Лего	305	2020-05-15	2021-03-18
13	4	Лего	300	2021-03-19	2022-08-15
14	4	Лего	305	2022-08-16	9999-12-31

Рисунок 5 – Результат работы кода

Задача 4

Заполненность таблиц:

	item_id	item_name	item_price	created_dttm
1	1	Книга	150	2019-09-01 00:00:01
2	3	Кукла	300	2019-09-08 00:00:01
3	2	Мяч	200	2019-10-01 00:00:01
4	4	Лего	100	2019-11-17 00:00:01
5	3	Кукла	325	2019-11-24 00:00:01
6	1	Книга	100	2019-12-12 00:00:01
7	4	Лего	400	2020-01-09 00:00:01
8	2	Мяч	180	2020-05-13 00:00:01
9	4	Лего	305	2020-05-15 00:00:01
10	3	Кукла	415	2020-06-06 00:00:01
11	1	Книга	285	2021-01-03 00:00:01
12	4	Лего	300	2021-03-19 00:00:01
13	1	Книга	500	2022-02-02 00:00:01
14	4	Лего	305	2022-08-16 00:00:01
15	1	Книга	331	2022-10-31 03:14:07
16	2	Мяч	331	2022-10-31 03:14:07
17	3	Кукла	431	2022-10-31 03:14:07
18	1	Книга	337	2022-11-05 03:14:07
19	2	Мяч	337	2022-11-06 03:14:07
20	3	Кукла	437	2022-11-07 03:14:07
21	4	Лего	663	2022-11-09 03:14:07
22	1	Книга	339	2022-11-10 03:14:07
23	2	Мяч	217	2022-11-11 03:14:07
24	3	Кукла	420	2022-11-11 03:14:07
25	2	Мяч	280	2022-11-24 03:14:07
26	3	Кукла	444	2022-11-24 03:14:07
27	1	Книга	377	2022-11-26 03:14:07
28	4	Лего	618	2022-11-28 03:14:07

Рисунок 6 -Скриншот данных из таблицы item_prices

	item_id	item_name	item_price	valid_from_dt	valid_to_dt
1	3	Кукла	300	2019-09-08	2019-11-23
2	1	Книга	150	2019-09-01	2019-12-11
3	4	Лего	100	2019-11-17	2020-01-08
4	2	Мяч	200	2019-10-01	2020-05-12
5	4	Лего	400	2020-01-09	2020-05-14
6	3	Кукла	325	2019-11-24	2020-06-05
7	1	Книга	100	2019-12-12	2021-01-02
8	4	Лего	305	2020-05-15	2021-03-18
9	1	Книга	285	2021-01-03	2022-02-01
10	4	Лего	300	2021-03-19	2022-08-15
11	2	Мяч	180	2020-05-13	2022-10-30
12	3	Кукла	415	2020-06-06	2022-10-30
13	1	Книга	500	2022-02-02	2022-10-30
14	1	Книга	331	2022-10-31	2022-11-04
15	2	Мяч	331	2022-10-31	2022-11-05
16	3	Кукла	431	2022-10-31	2022-11-06
17	4	Лего	305	2022-08-16	2022-11-08
18	1	Книга	337	2022-11-05	2022-11-09
19	2	Мяч	337	2022-11-06	2022-11-10
20	3	Кукла	437	2022-11-07	2022-11-10
21	2	Мяч	217	2022-11-11	2022-11-23
22	3	Кукла	420	2022-11-11	2022-11-23
23	1	Книга	339	2022-11-10	2022-11-25
24	4	Лего	663	2022-11-09	2022-11-27
25	1	Книга	377	2022-11-26	9999-12-31
26	2	Мяч	280	2022-11-24	9999-12-31
27	3	Кукла	444	2022-11-24	9999-12-31
28	4	Лего	618	2022-11-28	9999-12-31

Рисунок 7 - Скриншот данных из таблицы dict_item_prices

	transaction_id	customer_id	item_id	item_number	transaction_dttm
1	19	2	1	15	2020-06-06 03:14:07
2	20	2	2	20	2020-09-21 03:14:07
3	22	4	2	13	2020-12-09 03:14:07
4	23	4	1	1	2021-01-06 03:14:07
5	24	4	2	11	2021-03-09 03:14:07
6	26	2	1	14	2021-07-01 03:14:07
7	27	1	3	10	2021-07-01 03:14:07
8	28	1	3	11	2021-07-19 03:14:07
9	30	3	4	6	2021-09-30 03:14:07
10	31	1	4	8	2021-10-07 03:14:07
11	32	2	1	7	2022-01-05 03:14:07
12	33	5	2	10	2022-02-10 03:14:07
13	34	4	1	1	2022-04-06 03:14:07
14	35	1	3	2	2022-05-19 03:14:07
15	36	4	2	3	2022-05-21 03:14:07
16	37	4	2	6	2022-08-04 03:14:07
17	29	3	4	4	2022-08-04 03:14:07
18	38	1	1	9	2022-08-06 03:14:07
19	14	5	3	4	2022-09-04 03:14:07
20	39	2	4	10	2022-09-15 03:14:07
21	40	1	1	16	2022-11-01 03:14:07
22	21	2	1	17	2022-11-03 03:14:07
23	1	1	1	10	2022-11-04 03:14:07
24	2	2	1	1	2022-11-04 03:14:07
25	3	2	3	5	2022-11-04 03:14:07
26	4	1	2	11	2022-11-04 03:14:07
27	18	5	1	5	2022-11-04 03:14:07
28	25	1	1	2	2022-11-04 03:14:07
29	16	5	4	9	2022-11-04 03:14:07
30	17	5	4	7	2022-11-04 03:14:07
31	15	1	4	9	2022-11-04 03:14:07

Рисунок 8 - Скриншот данных из таблицы transaction_details

32	13	2	3	3	2022-11-04 03:14:07
33	12	3	3	8	2022-11-04 03:14:07
34	11	3	4	6	2022-11-04 03:14:07
35	10	1	2	6	2022-11-04 03:14:07
36	8	5	3	3	2022-11-04 03:14:07
37	6	1	2	2	2022-11-04 03:14:07
38	42	3	1	5	2022-11-05 03:14:07
39	7	2	1	1	2022-11-06 03:14:07
40	5	3	2	14	2022-11-10 03:14:07
41	9	5	1	5	2022-11-10 03:14:07
42	41	2	2	16	2022-11-14 03:14:07

Рисунок 9- Скриншот данных из таблицы transaction_details (Продолжение)

```
create table customer_aggr (select customer_id,sum(
(select dict_item_prices.item_price from dict_item_prices where
transaction_details.item_id = dict_item_prices.item_id and
```

```
(date(transaction_details.transaction_dttm) between dict_item_prices.valid_from_dt
and dict_item_prices.valid_to_dt) and
(date(transaction_details.transaction_dttm) between date_sub(date(now()),interval 1
month) and date(now())))
*transaction_details.item_number) as amount_spent_1m,substring((concat(max(
(select dict_item_prices.item_price from dict_item_prices where
transaction_details.item_id = dict_item_prices.item_id and
(date(transaction_details.transaction_dttm) between dict_item_prices.valid_from_dt
and dict_item_prices.valid_to_dt) and
(date(transaction_details.transaction_dttm) between date_sub(date(now()),interval 1
month) and date(now())))
*transaction_details.item_number),' ',(select distinct item_name from
dict_item_prices where dict_item_prices.item_id =
transaction_details.item_id))),INSTR((concat(max(
(select dict_item_prices.item_price from dict_item_prices where
transaction_details.item_id = dict_item_prices.item_id and
(date(transaction_details.transaction_dttm) between dict_item_prices.valid_from_dt
and dict_item_prices.valid_to_dt) and
(date(transaction_details.transaction_dttm) between date_sub(date(now()),interval 1
month) and date(now())))
*transaction_details.item_number),' ',(select distinct item_name from
dict_item_prices where dict_item_prices.item_id = transaction_details.item_id))),'
')+1) as top_item_1m
from transaction_details
group by customer_id
having amount_spent_1m is not null
order by customer_id
)
```

	customer_id	amount_spent_1m	top_item_1m
1	1	18 302	Книга
2	2	13 215	Книга
3	3	11 681	Мяч
4	5	9 523	Кукла

Рисунок 10 – Результат работы кода

Результат ручной проверки:

customer_id	item_id	transaction_id	transaction_dttm	item_price (из таблицы item_prices на дату transaction_dttm)	amount_spent_1m	Итого за последний месяц
3	4	6	22.09.30	300	1800	
	4	4	22.08.04	300	1200	
	4	6	22.11.04	305	1830	11681
	3	8	22.11.04	431	3448	
	1	5	22.11.05	337	1685	
	2 – Мяч	14	22.11.10	337	4718	
Итого:					14681	

Соответствие item_id и item_name:

1	Книга
2	Мяч
3	Кукла
4	Лего

Задача 5

Заполненность таблицы posts:

id	created_at	title
1	2022-01-17 08:50:58	Sberbank is the best bank
2	2022-01-17 18:36:41	Visa vs Mastercard
3	2022-01-17 16:16:17	Visa vs UnionPay
4	2022-01-17 18:01:00	Mastercard vs UnionPay
5	2022-01-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-01-16 14:57:32	NFC: wireless payment
1	2022-01-17 08:50:58	Sberbank is the best bank
2	2022-02-17 18:36:41	Visa vs Mastercard
3	2022-02-17 16:16:17	Visa vs UnionPay
4	2022-02-17 18:01:00	Mevastercard vs UnionPay
5	2022-02-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-02-16 14:57:32	NFC: wireless payment
4	2022-02-17 18:01:00	Mastveercard vs UnionPay
5	2022-02-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-02-16 14:57:32	NFC: wireless payment
1	2022-01-17 08:50:58	Sberbank is the best bank
2	2022-02-17 18:36:41	Visa vs Mastercard
3	2022-02-17 16:16:17	Visa vs UnionPay
4	2022-02-17 18:01:00	Mevastercard vs UnionPay
5	2022-02-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-02-16 14:57:32	NFC: wireless payment
4	2022-02-17 18:01:00	Mastveercard vs UnionPay
5	2022-03-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-03-16 14:57:32	NFC: wireless payment
5	2022-03-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-03-16 14:57:32	NFC: wireless payment
4	2022-03-17 18:01:00	Mastveercard vs UnionPay
5	2022-03-16 16:44:36	Hadoop or Greenplum: pros and cons
6	2022-02-16 14:57:32	NFC: wireless payment

```
with date_help as
(select min(date_format(created_at, '%Y-%m-%d')) as dt,count(*) as count_e from posts
group by (month(created_at))
)
select min(date_format(posts.created_at, '%Y-%m-%d')) as dt, count(*) as count,
concat(truncate(count(*)/(date_help.count_e)*100,2)-100,'%') as prcnt_growth from
posts
```

```
left join date_help on month(date_help.dt)  = month(posts.created_at) - 1
group by date_format(created_at, '%Y%m')
order by dt
```

	dt	count	prcnt_growth
1	2022-01-16	8	[NULL]
2	2022-02-16	15	87.50%
3	2022-03-16	6	-60.00%
4	2022-04-13	4	-33.34%

Рисунок 11– Результат работы кода