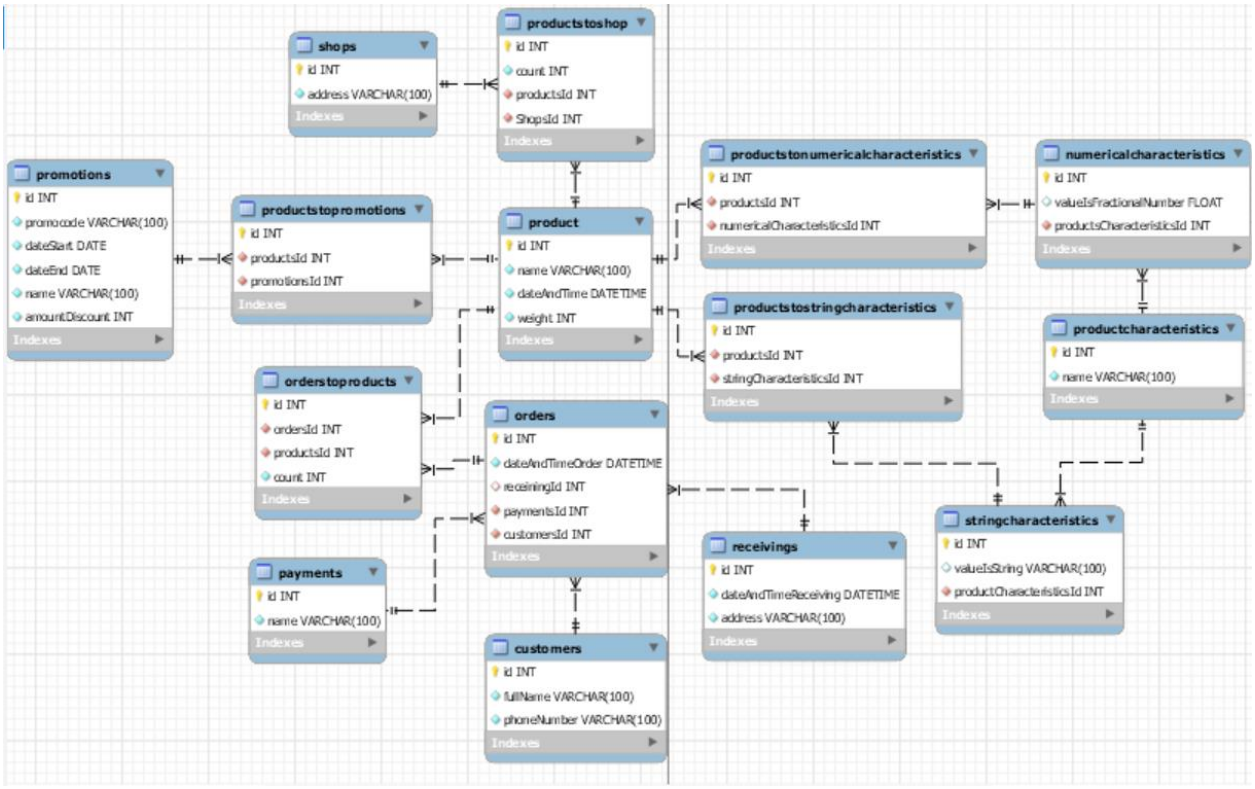
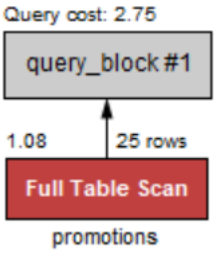
Лабораторная работа №5	Ф.И.О.	Якупова Д. А.
	Группа	ПрИн-266
	Преподаватель	Соколов Александр Александрович
	Дата сдачи	



1) Indexes

1.1

```
select promocode, datestart, dateend, name, amountdiscount FROM promotions
/*Дать информацию обо всех доступных акциях*/
where current_date() <= dateEnd;
```

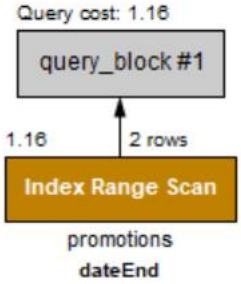



```
1 explain select promocode, dateStart, dateEnd, amountdiscount from promotions
2 /*Дать информацию обо всех доступных акциях*/
3 where current_date() <= dateEnd;
4
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	promotions	NULL	ALL	NULL	NULL	NULL	NULL	25	33.33	Using where

```
create index dateEnd on promotions(dateEnd);
```




```
1 explain select promocode, dateStart, dateEnd, amountdiscount from promotions
2 /*Дать информацию обо всех доступных акциях*/
3 where current_date() <= dateEnd;
4
```

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

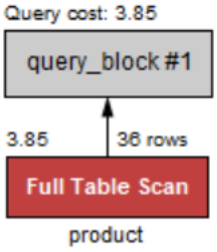
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	promotions	NULL	range	dateEnd	dateEnd	3	NULL	2	100.00	Using index condition

1.2)

```
select product.cost
/*Вывести стоимость определнного продукта*/
from product
where product.name='Медовик' and product.weight=1800;
```

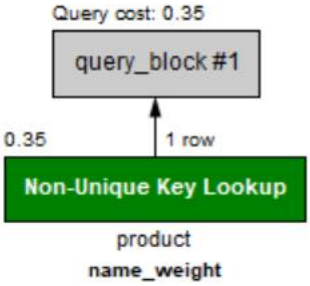
Result Grid | Filter Rows: | Export: | Wrap Cell Conter

cost
800



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	ALL	NULL	NULL	NULL	NULL	36	2.78	Using where

```
create index name_weight on product(name, weight);
```




```
explain select product.cost
/*Вывести стоимость определнного продукта*/
from product
where product.name='Медовик' and product.weight=1800;
```

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

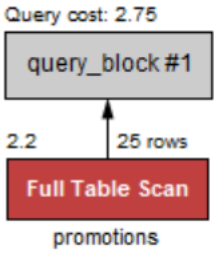
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	ref	name_weight	name_...	406	const,const	1	100.00	NULL

1.3)

```
select *
/*Получить весь список акций месяца*/
from promotions
where datestart>='2020-10-17 00:00:00' and datestart<=now();
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

id	promocode	dateStart	dateEnd	name	amountDiscount
4	HDKD	2020-11-10	2020-11-21	Осень	20
5	LK09	2020-10-18	2020-07-01	Осенние новинки	10
8	S100	2020-11-14	2020-11-29	Маффины	15
9	45AA	2020-11-10	2020-11-21	Скидки на весь ассотр...	10
11	FJ679	2020-11-12	2020-11-20	Скидки на весь ассорт...	5
NULL	NULL	NULL	NULL	NULL	NULL

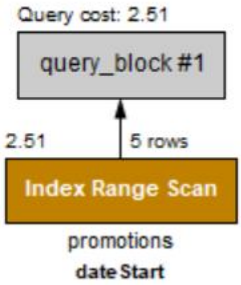



```
explain select *
/*Получить весь список акций месяца*/
from promotions
where datestart>='2020-10-17 00:00:00' and datestart<=now();
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	promotions	NULL	ALL	NULL	NULL	NULL	NULL	25	11.11	Using where

```
create index dateStart on promotions(dateStart);
```




```
explain select *
/*Получить весь список акций месяца*/
from promotions
where datestart>='2020-10-17 00:00:00' and datestart<=now();
```

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

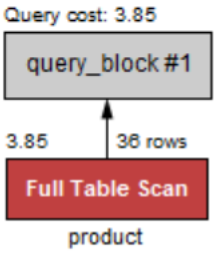
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	promotions	NULL	range	dateStart	dateStart	3	NULL	5	100.00	Using index condition

1.4)

```
select name from product
/*Дать информацию обо всех новинках*/
where isnew=1;
```

Result Grid | Filter Rows: | Export:

name
Шарлотка с яблоками
Хлеб баварский
Торт Птичье молоко
Слойка вишневая
Слойка вишневая
Прага
Пирожок ванильный
Печенье имбирное
Печенье имбирное
Маффин шоколадный

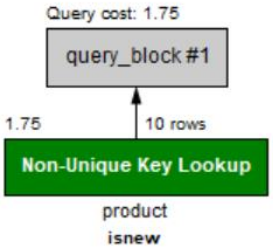



```
explain select name from product
/*Дать информацию обо всех новинках*/
where isnew=1;
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	ALL	NULL	NULL	NULL	NULL	36	10.00	Using where

```
create index isnew on product(isnew);
```




```
explain select name from product
/*Дать информацию обо всех новинках*/
where isnew=1;
```

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

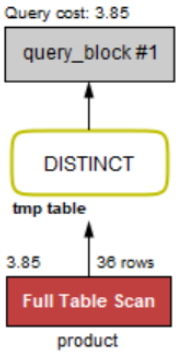
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	ref	isnew	isnew	1	const	10	100.00	NULL

1.5)

```
select DISTINCT name, weight, cost from product
/*Найти продукты, цены которых находятся в диапазоне 800-1500*/
where cost between 800 and 1500;
```

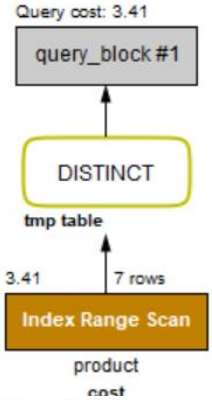
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	weight	cost
Медовик	1800	800
Тирамису	1950	1000
Маффин шоколадный	800	1500
Медовик	2000	800



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	ALL	NULL	NULL	NULL	NULL	36	11.11	Using where; Using temporary

```
create index cost on product(cost);
```




```
explain select DISTINCT name, weight, cost from product
/*Найти продукты, цены которых находятся в диапазоне 800-1500*/
where cost between 800 and 1500;
```

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

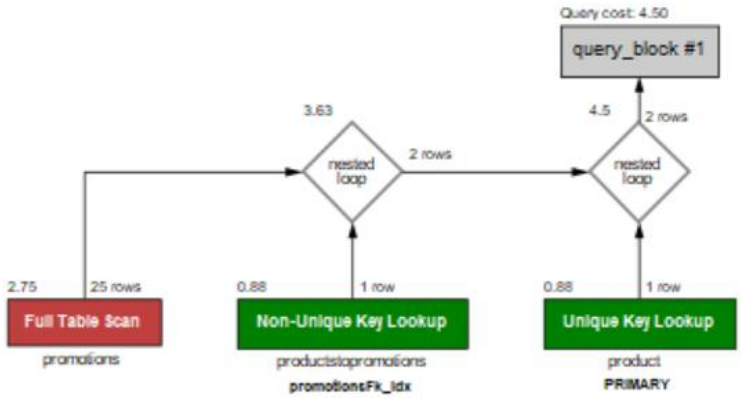
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	range	cost	cost	4	NULL	7	100.00	Using index condition; Using temporary

1.6)

```
select product.name, product.cost
from product
/*Вывести все акции, скидка которых ровняется 20*/
inner join productstopromotions on productstopromotions.productsId = product.id
inner join promotions on promotions.id = productstopromotions.promotionsId
where promotions.amountDiscount = 20;
```

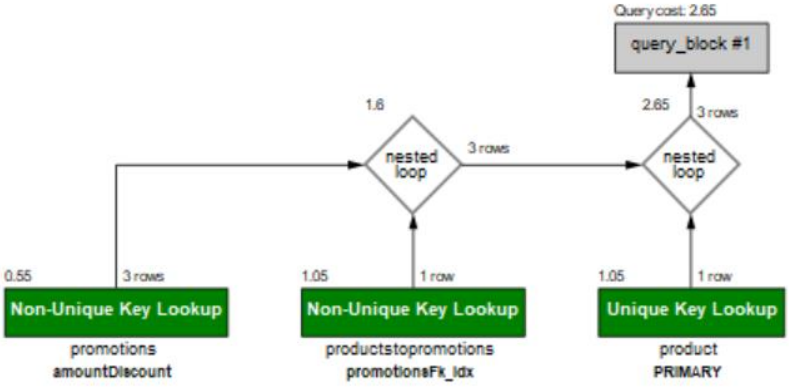
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	cost
Прага	2000



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	promotions	NULL	ALL	PRIMARY	NULL	NULL	NULL	25	10.00	Using whe
1	SIMPLE	productstopr...	NULL	ref	productsPromFk_idx,promotionsFk_idx	promot...	4	shop.promotions.id	1	100.00	NULL
1	SIMPLE	product	NULL	eq_ref	PRIMARY	PRIMARY	4	shop.productstopromotions.prod...	1	100.00	NULL

```
create index amountDiscount on promotions(amountDiscount);
```



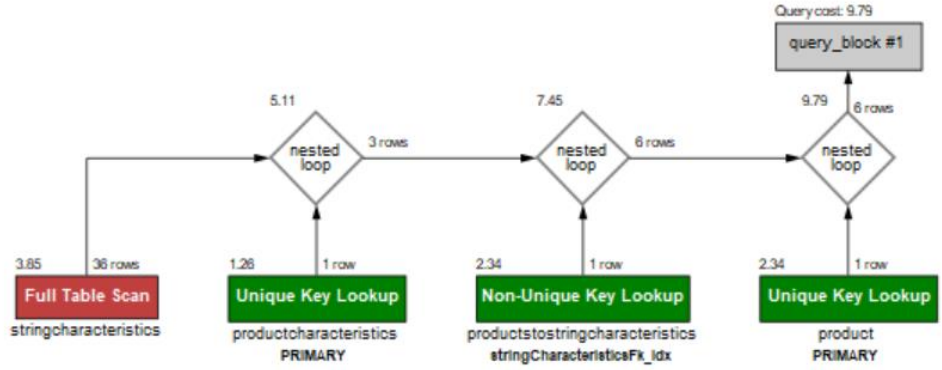
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	promotions	NULL	ref	PRIMARY,amountDiscount	amoun...	4	const	3	100.00	Using ind
1	SIMPLE	productstopr...	NULL	ref	productsPromFk_idx,promotionsFk_idx	promot...	4	shop.promotions.id	1	100.00	NULL
1	SIMPLE	product	NULL	eq_ref	PRIMARY	PRIMARY	4	shop.productstopromotions.prod...	1	100.00	NULL

1.7)

```
select product.name
/*Получить информацию обо всех продуктах данного типа*/
from product
inner join productstostringcharacteristics on productstostringcharacteristics.productsid=product.id
inner join stringcharacteristics on stringcharacteristics.id=productstostringcharacteristics.stringcharacteristicsid
inner join productcharacteristics on productcharacteristics.id=stringcharacteristics.productcharacteristicsid
where stringcharacteristics.valueisstring='Торты';
```

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

name
Прага
Медовик
Чизкейк лаймовый
Торт Птичье молоко
Чизкейк клубничный
Чизкейк лаймовый

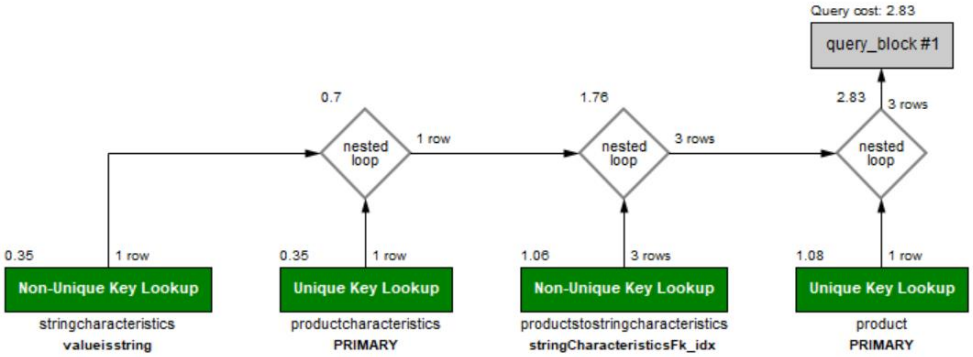



```
explain select product.name
/*Получить информацию обо всех продуктах данного типа*/
from product
inner join productstostringcharacteristics on productstostringcharacteristics.productsid=product.id
inner join stringcharacteristics on stringcharacteristics.id=productstostringcharacteristics.stringcharacteristicsid
inner join productcharacteristics on productcharacteristics.id=stringcharacteristics.productcharacteristicsid
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	stringcharact...	NULL	ALL	PRIMARY,productCharacteristicsFk_idx	NULL	NULL	NULL	36	10.00	Using where
1	SIMPLE	productchara...	NULL	eq_ref	PRIMARY	PRIMARY	4	shop.stringcharacteristics.produc...	1	100.00	Using index
1	SIMPLE	productstostri...	NULL	ref	productsFk_idx,stringCharacteristicsFk_idx	stringC...	4	shop.stringcharacteristics.id	1	100.00	NULL
1	SIMPLE	product	NULL	eq_ref	PRIMARY	PRIMARY	4	shop.productstostringcharacteris...	1	100.00	NULL

```
create index valueisstring on stringcharacteristics(valueisstring);
```



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	stringcharact...	NULL	ref	PRIMARY,productCharacteristicsFk_idx,valueiss...	valueis...	403	const	1	100.00	NULL
1	SIMPLE	productchara...	NULL	eq_ref	PRIMARY	PRIMARY	4	shop.stringcharacteristics.produc...	1	100.00	Using index
1	SIMPLE	productstostri...	NULL	ref	productsFk_idx,stringCharacteristicsFk_idx	stringC...	4	shop.stringcharacteristics.id	3	100.00	NULL
1	SIMPLE	product	NULL	eq_ref	PRIMARY	PRIMARY	4	shop.productstostringcharacteris...	1	100.00	NULL

2) Процедуры, Функции

2.1) Процедуры

2.1.1)

/*Вывести тип данного продукта*/

```
call GetTypeOfProduct (in name_of_product varchar(100))
begin
    select distinct stringcharacteristics.valueIsString
    from stringcharacteristics
    inner join productcharacteristics on stringcharacteristics.productCharacteristicsId = productcharacteristics.id
    inner join productstostringcharacteristics on productstostringcharacteristics.stringCharacteristicsId = stringCharacteristics.id
    inner join product on productstostringcharacteristics.productsId = product.id
    where product.name = name_of_product and productcharacteristics.name = 'Тип продукта';
end$$
```

```
1 •  call GetTypeOfProduct ('Медовик');
2
```

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

valueIsString
Торты

```
1 •  call GetTypeOfProduct ('Печенье творожное');
2
```

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

valueIsString
Печенье

2.1.2)

```
delimiter $$
create procedure GetAddrShop (in name_of_product varchar(100))
begin
select shops.address, product.dateandtime
/*Вывести адрес магазина и дату изготовления товара, срок которых не превышает 2-х дней*/
from shops
inner join productstoshop on productstoshop.shopsid=shops.id
inner join product on productstoshop.productsid=product.id
where product.dateandtime>='2020-11-16 00:00:00' and product.name=name_of_product;
end$$
delimiter ;
```

```
call GetAddrShop ('Маффин шоколадный');
```

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

address	dateandtime
Хорошева 156 80	2020-11-16 13:09:34
Балонина 32В	2020-11-17 09:09:32

```
call GetAddrShop ('Медовик');
```

Result Grid | Filter Rows: | Export:

address	dateandtime
Качинцев 32А 146	2020-11-20 17:15:04

2.1.3)

```
delimiter $$
create procedure GetProduct_MinSugar (in name_of_customer varchar(100))
begin
select product.name, numericalcharacteristics.valueisfractionalnumber
/*Вывести продукт, содержащий максимальное количество сахара в заказе у данного покупателя*/
from product
inner join orderstoproducts on orderstoproducts.productsid=product.id
inner join orders on orderstoproducts.ordersid=ordersid
inner join productstonumericalcharacteristics on productstonumericalcharacteristics.productsid=product.id
inner join numericalcharacteristics on productstonumericalcharacteristics.numericalcharacteristicsid=numericalcharacteristics.id
inner join productcharacteristics on numericalcharacteristics.productscharacteristicsid =productcharacteristics.id
inner join customers on orders.customersid=customers.id
where customers.fullname = name_of_customer and orders.customersid=customers.id and productcharacteristics.name='Количество сахара'
group by product.name
order by numericalcharacteristics.valueisfractionalnumber desc
limit 1;
end$$
delimiter ;
```

```
call GetProduct_MinSugar ('Иванов И. И.');
```

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

name	valueisfractionalnumber
Чизкейк клубничный	100

```
call GetProduct_MinSugar ('Иванов М. И.');
```

Result Grid | Filter Rows: | Export: | Wrap

name	valueisfractionalnumber
Прага	150

2.2) Функции

2.2.1)

```
delimiter $$
/*Вывести количество определенного продукта в заказе*/
create function GetCountOfProduct(name_product varchar(100), name_customer varchar(100))
returns int
deterministic
begin
declare count int default 0;
select count(orderstoproducts.count) into count
from orderstoproducts
inner join product on orderstoproducts.productsId = product.id
inner join orders on orderstoproducts.ordersId = orders.id
inner join customers on orders.customersId = customersId
where product.name = name_product and customers.fullName = name_customer;
return count;
end$$
```

```
select GetCountOfProduct('Прага', 'Иванов М. И.');
```

Result Grid | Filter Rows: | Export: | Wrap

GetCountOfProduct('Прага', 'Иванов М. И.')
2

```
select GetCountOfProduct('Слойка груша-сыр', 'Петров В. И.');
```

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

GetCountOfProduct('Слойка груша-сыр', 'Петров В. И.')
4

2.2.2)

```
delimiter $$
/*Вывести информацию о степени содержания сахара*/
create function GetInformationAboutSugar(name_product varchar(100))
returns varchar(100)
deterministic
begin
declare information varchar(100);
declare amount int;
select numericalcharacteristics.valueIsFractionalNumber into amount
from numericalcharacteristics
inner join productstonumericalcharacteristics on productstonumericalcharacteristics.numericalCharacteristicsId = numericalcharacteristics.id
inner join productcharacteristics on numericalcharacteristics.productsCharacteristicsId = productcharacteristics.id
inner join product on productstonumericalcharacteristics.productsId = product.id
where product.name = name_product and productcharacteristics.id=9;
case
when amount between 0 and 30
then set information='Низкое количество сахара';
when amount between 30 and 70
then set information='Среднее количество сахара';
else
set information='Высокое количество сахара';
end case;
return information;
end$$
delimiter ;
```

```
select GetInformationAboutSugar('Прага');
```

Result Grid | Filter Rows: | Export:

	GetInformationAboutSugar('Прага')
▶	Высокое количество сахара

```
select GetInformationAboutSugar('Печенье творожное');
```

esult Grid | Filter Rows: | Export: | Wrap C

	GetInformationAboutSugar('Печенье творожное')
	Среднее количество сахара

2.2.3)

```
delimiter $$
/*Количеств продукции в магазине*/
create function OrderCount (name_product varchar(100), address_shop varchar(100))
returns int
deterministic
begin
declare productcount int;
select sum(productstoshop.count) into productcount
from productstoshop
inner join product on productstoshop.productsId = product.id
inner join shops on productstoshop.ShopsId =shops.id
where product.name=name_product and shops.address=address_shop;
return productcount;
end$$
delimiter ;
```

```
select OrderCount ('Маффин шоколадный', 'Хорошева 156 80');
```

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

OrderCount ('Маффин шоколадный', 'Хорошева 156 80')
6

```
select OrderCount ('Слойка груша-сыр', 'Каспийская 120')
```

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

OrderCount ('Слойка груша-сыр', 'Каспийская 120')
10

3) Представления

3.1)

```
create view wiew_order as
select product.name, orderstoproducts.count
from product
inner join orderstoproducts on orderstoproducts.productsId = product.id
inner join orders on orderstoproducts.ordersId = orders.id
where product.name = 'Медовик' and orders.dateAndTimeOrder <= now() and orders.dateAndTimeOrder >= '2020-11-17 00:00:00';
```

```
select * from wiew_order;
```

	name	count
▶	Медовик	3

3.2)

```
create view wiew_popular_shop as
select shops.address
/*Получить информацию о самом популярном магазине*/
from shops
inner join productstoshop on productstoshop.shopsid=shops.id
group by shops.address
order by sum(productstoshop.count) desc
limit 1;
```

```
select * from wiew_popular_shop;
```

Result Grid | Filter Rows:

	address
▶	Хорошева 156 80