Боцан Николай P-2023r

**Индивидуальная работа за 31.03.22**

**«Учёт продажи техники в компьютерном магазине»**

Оглавление

[Задание: 2](#_Toc99653456)

[Диаграмма базы данных: 2](#_Toc99653457)

[Домашняя работа: 3](#_Toc99653458)

[--5 сложных запросов с различными типами соединений-- 3](#_Toc99653459)

[--5 подзапросов-- 5](#_Toc99653460)

[--5 запросов на группировку и агрегацию информации-- 7](#_Toc99653461)

[Код базы данных: 8](#_Toc99653462)

# Задание:

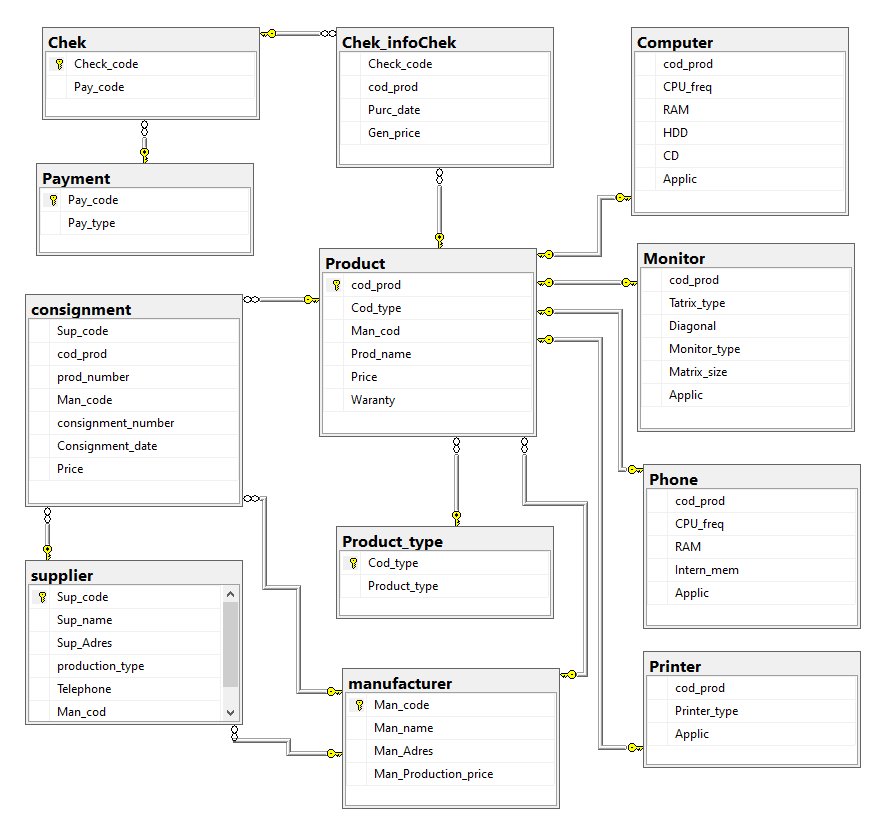
Используя базу данных собственного проекта создать:

1) 5 сложных запроса с применением различных типов соединений;

2) 5 подзапроса;

3) 5 запросов на группировку и агрегацию информации.

# Диаграмма базы данных:



# Домашняя работа:

## --5 сложных запросов с различными типами соединений--

/\*Это сложный запрос с функцией агрегации\*/

--посмотреть на какую сумму было поставлено каждого товара--

/\*Для этого создадим представление\*/

create view Product\_Type\_view *as*

select cod\_prod, Product\_type, Man\_cod, Prod\_name, Price, Waranty

from Product inner join Product\_type on Product.Cod\_type = Product\_type.Cod\_type

go

select Product\_type *as* Product, sum(consignment.Price\*prod\_number) *as* Total\_price

from Product\_Type\_view inner join consignment on Product\_Type\_view.cod\_prod = consignment.cod\_prod

group by Product\_type

go

--Вывести каким образом расчитался покупатель Болякин Андрей Никифорович за товар--

select Pay\_type *as* Paying\_by

from Payment inner join Pasport\_data on Payment.Pay\_code = Pasport\_data.Pay\_code

where C\_Name = 'Андрей' *and* C\_Surname = 'Болякин' *and* C\_FatherName = 'Никифорович'

go

--то же, но с подзапросами--

select Pay\_type *as* Paying\_by

from Payment

where Pay\_code = (select Pay\_code

              from Pasport\_data

              where C\_Name = 'Андрей' *and* C\_Surname = 'Болякин' *and* C\_FatherName = 'Никифорович'

             )

go

--вывести, какие товары были поставлены в феврале (02)--

select Prod\_name *as* Product

from Product inner join consignment on Product.cod\_prod = consignment.cod\_prod

where *month*(Consignment\_date) = 2

go

--то же, но с подзапросами--

select Prod\_name *as* Product

from Product

where cod\_prod *in* (select cod\_prod

               from consignment

               where *month*(Consignment\_date) = 2

              )

go

--вывести все продажи за 2021-12-31--

--естественное соединение c представлением Product\_Type\_view--

select distinct Product\_type

from Product\_Type\_view inner join Chek\_infoChek on Product\_Type\_view.cod\_prod = Chek\_infoChek.cod\_prod

where Purc\_date = '2021-12-31'

go

--то же, но с подзапросами--

select Product\_type

from Product\_Type

where Cod\_type *in* (select Cod\_type

               from Product

               where cod\_prod *in* (select cod\_prod

                             from Chek\_infoChek

                             where Purc\_date = '2021-12-31'

                             )

              )

go

/\*сложный запрос с группировкой и функцией агрегации\*/

--вывести количекство продаж каждого типа продукции за 2021-12-31--

select Product\_type, count(Product\_type) *as* amount\_of\_products

from Product\_Type\_view inner join Chek\_infoChek on Product\_Type\_view.cod\_prod = Chek\_infoChek.cod\_prod

where Purc\_date = '2021-12-31'

group by Product\_type

go

--+ Часть в подзапросах...--

## --5 подзапросов--

/\*Это сложный запрос с подзапросом\*/

--Вывести технические характеристики всех имеющихся компьютеров--

select Prod\_name *as* Product, CPU\_freq *as* CPU, RAM, HDD, CD, Applic *as* Aplication

from Computer inner join Product on Computer.cod\_prod = Product.cod\_prod

where Cod\_type = (select Cod\_type

              from Product\_type

              where Product\_type = 'PC'

             )

go

/\*Это сложный запрос с подзапросом\*/

--Вывести информацию из всех чеков, оплаченных наличкой--

select Prod\_name *as* Product, Purc\_date *as* Purcasing\_date, Gen\_price *as* General\_price

from Chek\_infoChek inner join Product on Chek\_infoChek.cod\_prod = Product.cod\_prod

where Check\_code *in* (select Check\_code --*in*, а не =, так как по итогу создастся отношение (Pay\_type = 'cash' > 2-х кортежей) => в Chek\_infoChek передастся уже не 1 значение, а > 1 (2)--

               from Chek

               where Pay\_code = (select Pay\_code

                             from Payment

                             where Pay\_type = 'cash'

                            )

               )

go

--посмотреть, какие товары поставляет поставщик XMoldova--

select Product\_type

from Product\_type

where Cod\_type *in* (select Cod\_type

               from Product

               where cod\_prod *in*(select cod\_prod

                            from consignment

                            where Sup\_code = (select Sup\_code

                                          from supplier

                                          where Sup\_name = 'XMoldova'

                                         )

                           )

             )

go

--вывести тип продукции, которую производит компания LG--

select Product\_type

from Product\_type

where Cod\_type *in* (select Cod\_type

               from Product

               where Man\_cod = (select Man\_code

                           from manufacturer

                           where Man\_name = 'LG'

                           )

             )

go

/\*Подзапрос со сложным запросом с функцией агрегации\*/

--вывести самый дорогой товар--

select Prod\_name *as* Product, Product\_type, Price

from Product inner join Product\_type on Product.Cod\_type = Product\_type.Cod\_type

where price = (select max(price)

            from Product)

go

--Вывести технические характеристики Redmi 5 pro--

select CPU\_freq *as* CPU, RAM, Intern\_mem *as* Memory, Applic *as* Aplication

from Phone

where cod\_prod = (select cod\_prod

              from Product

              where Prod\_name = 'Redmi 5 pro'

             )

go

--То же, но с естественным соединением--

select CPU\_freq *as* CPU, RAM, Intern\_mem *as* Memory, Applic *as* Aplication

from Product inner join Phone on Phone.cod\_prod = Product.cod\_prod

where Prod\_name = 'Redmi 5 pro'

go

## --5 запросов на группировку и агрегацию информации--

--посчитать количество товаров, поставленных в каждом месяце--

select *month*(Consignment\_date) *as* mounth, sum(prod\_number) *as* supply\_amount

from consignment

group by *month*(Consignment\_date)

go

--посчитать общую цену всех товаров, поставленных в феврале (02)--

select sum(price\*prod\_number) *as* Total\_suply\_price

from consignment

where *month*(Consignment\_date) = 2

go

--подстчитать число каждой продукции имеющейся в магазине--

select Product\_type *as* Production, count(\*) *as* Products\_amount

from Product\_type inner join Product on Product\_type.Cod\_type = Product.Cod\_type

group by Product\_type

go

--вывести количество продаж за 2021-12-31--

select count(cod\_prod) *as* amount\_of\_products

from Chek\_infoChek

where Purc\_date = '2021-12-31'

go

--подсчитать количество покупателей для каждого способа оплаты--

/\*Создадим представление\*/

create view client\_payment\_view *as*

select id, C\_Adres, C\_Surname, C\_Name, C\_FatherName, Pay\_type

from Payment inner join Pasport\_data on Payment.Pay\_code = Pasport\_data.Pay\_code

go

select \*

from client\_payment\_view

go --создаем представление так как Pay\_type не входит в Pasport\_data и в результате не произойдет агрегация--

select Pay\_type, count(\*) *as* amount\_of\_clients

from client\_payment\_view

group by Pay\_type

go

# Код базы данных:

--Создание контейнера базы данных--

create database Computer\_magazine

go

--Активизирование контейнера базы данных--

use Computer\_magazine

go

create type Adres from *varchar*(70);

go

create type applicationS from *varchar*(100);

go

create type TNumber from *char*(12);

go

create table *Pasport\_data*(Id int primary key,

                    C\_Adres Adres,

                    C\_Surname *varchar*(12),

                    C\_Name *varchar*(12),

                    C\_FatherName *varchar*(12),

                    Pay\_code int foreign key references *Payment*(Pay\_code) *not* null)

go

insert into *Pasport\_data*(Id, C\_Adres, C\_Surname, C\_Name, C\_FatherName, Pay\_code)

values (1001, 'ул.Пушкина 6/7', 'Балконский', 'Андрей', 'Николаевич', 102),

      (1002, 'ул. Николае Милеску Спэтарул 21/1', 'Безухов', 'Пьер', 'Анатольевичь', 101),

      (1003, 'ул. Садовяну 13/5', 'Ростов', 'Николай', 'Ильич', 103),

      (1004, 'ул. Петру Заднипру 28/8', 'Болякин', 'Андрей', 'Никифорович', 102),

      (1005, 'ул. Мунчешть 76/9', 'Шишкин', 'Алескандр', 'Игоревич', 103)

go

create table *Product\_type*(Cod\_type smallint primary key,

                    Product\_type *varchar*(20))

go

insert into *Product\_type*(Cod\_type, Product\_type)

values (2000, 'PC'),

      (2010, 'Laptop'),

      (2020, 'Monitor'),

      (2030, 'Phone'),

      (2040, 'Tablet PC'),

      (2050, 'Printer')

go

create table *supplier*(Sup\_code int primary key,

                 Sup\_name *varchar*(20),

                 Company *varchar*(20),

                 Sup\_Adres Adres,

                 production\_type *varchar*(40))

go

alter table supplier

add Telephone TNumber

go

alter table supplier

drop column Company

go

alter table supplier

add Man\_cod int foreign key references *manufacturer*(Man\_code) *not* null

go

insert into *supplier*(Sup\_code, Sup\_name, Man\_cod, Sup\_Adres, production\_type, Telephone)

values (3001, 'Comp\_Moldova', 4003, 'ул. Штефан чел маре 21/5', 'PCs, Laptops, Monitors, Phones', '+37367354672'),

      (3002, 'Hyper Supply', 4005, 'ул. Гоголя 34/4', 'PCs, Monitors, Printers, Laptops', '+37369984519'),

      (3003, 'GMSupply', 4006, 'ул. Колумна 5', 'PCs, Laptops', '+37378534765'),

      (3004, 'TechMD', 4002, 'ул. Узинелор 7', 'Monitors, Phones, Laptops', '+37379945543'),

      (3005, 'XMoldova', 4001, 'ул. Петру Мовилэ 7', 'Laptops, Phones, Printers, Monitors', '+37360606607'),

      (3006, 'Hyper Supply', 4004, 'ул. Михаил Садовяну 3', 'PCs, Monitors, Printers', '+37369984519')

go

update supplier

set Sup\_name = 'HPSupply'

where Sup\_code = 3006

go

create table *manufacturer*(Man\_code int primary key,

                    Man\_name *varchar*(20),

                    Man\_Adres Adres,

                    Man\_Production\_price smallint)

go

insert into *manufacturer*(Man\_code, Man\_name, Man\_Adres, Man\_Production\_price)

values (4001, 'Xiaomi', 'ул. Петру Мовилэ 8', 2000),

      (4002, 'Samsung', 'ул. Узинелор 10', 3000),

      (4003, 'LG', 'ул. Мунчешть 7', 3000),

      (4004, 'HP', 'ул. Гоголя 36', 2000),

      (4005, 'Hyper PC', 'ул. Михаил Садовяну 10', 4000),

      (4006, 'Game Max', 'ул. Отовыска 7', 4000)

go

create table *Payment*(Pay\_code int primary key,

                Pay\_type *varchar*(15) default 'cash')

go

insert into *Payment*(Pay\_code, Pay\_type)

values (101, 'credit'),

      (102, 'credit card'),

      (103, 'cash')

go

create table *Chek*(Check\_code int primary key,

              Pay\_code int foreign key references *Payment*(Pay\_code) *not* null)

go

insert into *Chek*(Check\_code, Pay\_code)

values (1013, 101),

      (1014, 102),

      (1017, 103),

      (1020, 103),

      (1022, 102)

go

create table *Product*(cod\_prod int primary key,

                Cod\_type smallint foreign key references *Product\_type*(Cod\_type) *not* null,

                Man\_cod int foreign key references *manufacturer*(Man\_code) *not* null,

                Prod\_name *varchar*(20),

                Price smallint,

                Waranty smallint *check*(Waranty <= 5) default 1)

go

insert into *Product*(cod\_prod, Cod\_type, Man\_cod, Prod\_name, Price, Waranty)

values (5000, 2000, 4005, 'Blye Yeti', 20000, 5),

      (5010, 2010, 4006, 'Nitro', 25000 ,3),

      (5020, 2020, 4003, 'Spin 5', 5000, 1),

      (5021, 2020, 4004, 'Hp one', 2000, 1),

      (5030, 2030, 4002, 'Redmi 5 pro', 4000, 1),

      (5040, 2040, 4001, 'Prestige Pro 3', 4700, 2),

      (5050, 2050, 4004, 'All Vew Pro', 2000, 3),

      (5051, 2050, 4001, 'Clear Sky', 3000, 4),

      (5052, 2050, 4001, 'Colors', 2500, 2)

go

create table *Computer*(cod\_prod int foreign key references *Product*(cod\_prod) unique *not* null, --связь 1 к 1--

                 CPU\_freq smallint default 2000,

                 RAM smallint default 1000,

                 HDD smallint default 250,

                 CD *varchar*(5) *check*((CD = 'true') *or* (CD = 'false')) default 'false',

                 Applic applicationS)

go

insert into *Computer*(cod\_prod, CPU\_freq, RAM, HDD, CD, Applic)

values (5000, 4200, 16000, 1000, 'false', 'Gaming'),

      (5010, 3900, 8000, 2500, 'true', 'Gaming')

go

create table *Monitor*(cod\_prod int foreign key references *Product*(cod\_prod) unique *not* null,

                Tatrix\_type *varchar*(5) default 'TN',

                Diagonal smallint default 21,

                Monitor\_type *varchar*(20) default 'without rotation',

                Matrix\_size *varchar*(10) default 'glossy',

                Applic applicationS)

go

insert into *Monitor*(cod\_prod, Tatrix\_type, Diagonal, Monitor\_type, Matrix\_size, Applic)

values (5020, 'IPS', 21, 'with rotation', 'mate', 'Gaming'),

      (5021, 'TN', 18, 'without rotation', 'glossy','Office')

go

create table *Phone*(cod\_prod int foreign key references *Product*(cod\_prod) unique *not* null,

               CPU\_freq smallint default 1000,

               RAM smallint default 1000,

               Intern\_mem smallint default 5000,

               Applic applicationS)

go

insert into *Phone*(cod\_prod, CPU\_freq, RAM, Intern\_mem, Applic)

values (5030, 2000, 3000, 32000, 'Dayly'),

      (5040, 3000, 4000, 32000, 'Graphic design')

go

create table *Printer*(cod\_prod int foreign key references *Product*(cod\_prod) unique *not* null,

                Printer\_type *varchar*(6) *check*((Printer\_type = 'color') *or* (Printer\_type = 'laser')) default 'laser',

                Applic applicationS)

go

insert into *Printer*(cod\_prod, Printer\_type, Applic)

values (5050, 'laser', 'Office'),

      (5051, 'color', 'Photos'),

      (5052, 'color', 'Photos')

go

create table *consignment*(Sup\_code int foreign key references *supplier*(Sup\_code) *not* null,

                   cod\_prod int foreign key references *Product*(cod\_prod) *not* null,

                   prod\_number smallint,

                   Man\_code int foreign key references *manufacturer*(Man\_code) *not* null,

                   consignment\_number int,

                   Consignment\_date date,

                   Price int)

go

insert into *consignment*(Sup\_code, cod\_prod, prod\_number, Man\_code, consignment\_number, Consignment\_date, Price)

values (3002, 5000, 5, 4005, 10456, '2021-12-21', 17000),

      (3003, 5010, 3, 4006, 10432, '2022-01-04', 14000),

      (3001, 5020, 4, 4003, 10487, '2021-10-10', 3000),

      (3006, 5021, 7, 4004, 10499, '2022-01-09', 1000),

      (3004, 5030, 10, 4002, 10444, '2022-02-27', 2500),

      (3005, 5040, 8, 4001, 10411, '2022-01-01', 3000),

      (3006, 5050, 11, 4004, 10410, '2022-02-04', 1500),

      (3005, 5051, 15, 4001, 10440, '2022-02-02', 2000),

      (3005, 5052, 12, 4001, 10404, '2021-12-13', 1700)

go

/\*

drop table consignment

go

\*/

create table *Chek\_infoChek*(Check\_code int foreign key references *Chek*(Check\_code) *not* null,

                     cod\_prod int foreign key references *Product*(cod\_prod) *not* null,

                     Purc\_date date,

                     Gen\_price smallint)

go

insert into *Chek\_infoChek*(Check\_code, cod\_prod, Purc\_date, Gen\_price)

values (1013, 5000, '2021-12-31', 20000),

      (1014, 5052, '2021-12-31', 2500),

      (1017, 5040, '2022-01-08', 4700),

      (1020, 5021, '2022-01-09', 2000),

      (1022, 5030, '2022-02-28', 4000)

go

/\*

drop table Chek\_infoChek

go

\*/

--Укажите, какая информация в созданной базе данных может изменяться и примените команды группы DML для её изменения.--

--Printer(Printer\_type)--

update Printer

set Printer\_type = 'laser'

where cod\_prod = 5052

go

update Printer

set Printer\_type = 'color'

where cod\_prod = 5052

go

--Computer(RAM, HDD, CD)--

update Computer

set RAM = 32000

where (cod\_prod = 5000)

go

update Computer

set RAM = 16000

where (cod\_prod = 5000)

go

--Product(price, name)--

update Product

set Price = 15000

where (Prod\_name = 'Blue Yeti') *and* (cod\_prod = 5000)

go

update Product

set Prod\_name = 'Blue Yeti'

where Prod\_name = 'Blye Yeti'

go

update Product

set Price = 20000

where (Prod\_name = 'Blue Yeti') *and* (cod\_prod = 5000)

go

--manufacterer(Man\_name, Man\_Production\_price)

update manufacturer

set Man\_name = 'Lenovo'

where Man\_name = 'HP'

go

update manufacturer

set Man\_name = 'HP'

where Man\_name = 'Lenovo'

go

update manufacturer

set Man\_Production\_price = 7000

where Man\_Production\_price = 2000

go

update manufacturer

set Man\_Production\_price = 2000

where Man\_Production\_price = 7000

go

select \*

from manufacturer

go

--supplier(production\_type, Sup\_name)

update supplier

set Sup\_name = 'SamsungS\_MD'

where Sup\_name = 'TechMD'

go

update supplier

set Sup\_name = 'TechMD'

where Sup\_name = 'SamsungS\_MD'

go

update supplier

set production\_type = 'Monitors, Phones, Laptops, PCs'

where Sup\_name = 'TechMD'

go

update supplier

set production\_type = 'Monitors, Phones, Laptops'

where Sup\_name = 'TechMD'

go

--Создайте необходимые индексы--

/\*Можно было бы создать индексы также и для отношений product\_type и Payment, но в данных отношениях только два столбца, один из которых первичный ключ \*/

--создание индексов для отношения manufacturer (редко вносим, часто берём => index)--

create index man\_name\_indx on *manufacturer*(Man\_name)

go

create index man\_adres\_indx on *manufacturer*(Man\_Adres)

go

--создание индексов для отношения supplyer (редко вносим, часто берём => index)--

create index sup\_name\_indx on *supplier*(Sup\_name)

go

create index sup\_adres\_indx on *supplier*(Sup\_Adres)

go

create index sup\_telephone\_indx on *supplier*(Telephone)

go

create index sup\_prodType\_indx on *supplier*(production\_type)

go

select \*

from Chek

go

select \*

from Chek\_infoChek

go

select \*

from Computer

go

select \*

from consignment

go

select \*

from manufacturer

go

select \*

from Monitor

go

select \*

from Pasport\_data

go

select \*

from Payment

go

select \*

from Phone

go

select \*

from Printer

go

select \*

from Product

go

select \*

from Product\_type

go

select \*

from supplier

go

--Individual work 31.03.22--

--5 сложных запросов с различными типами соединений--

/\*Это сложный запрос с функцией агрегации\*/

--посмотреть на какую сумму было поставлено каждого товара--

/\*Для этого создадим представление\*/

create view Product\_Type\_view *as*

select cod\_prod, Product\_type, Man\_cod, Prod\_name, Price, Waranty

from Product inner join Product\_type on Product.Cod\_type = Product\_type.Cod\_type

go

select Product\_type *as* Product, sum(consignment.Price\*prod\_number) *as* Total\_price

from Product\_Type\_view inner join consignment on Product\_Type\_view.cod\_prod = consignment.cod\_prod

group by Product\_type

go

--Вывести каким образом расчитался покупатель Болякин Андрей Никифорович за товар--

select Pay\_type *as* Paying\_by

from Payment inner join Pasport\_data on Payment.Pay\_code = Pasport\_data.Pay\_code

where C\_Name = 'Андрей' *and* C\_Surname = 'Болякин' *and* C\_FatherName = 'Никифорович'

go

--то же, но с подзапросами--

select Pay\_type *as* Paying\_by

from Payment

where Pay\_code = (select Pay\_code

              from Pasport\_data

              where C\_Name = 'Андрей' *and* C\_Surname = 'Болякин' *and* C\_FatherName = 'Никифорович'

             )

go

--вывести, какие товары были поставлены в феврале (02)--

select Prod\_name *as* Product

from Product inner join consignment on Product.cod\_prod = consignment.cod\_prod

where *month*(Consignment\_date) = 2

go

--то же, но с подзапросами--

select Prod\_name *as* Product

from Product

where cod\_prod *in* (select cod\_prod

               from consignment

               where *month*(Consignment\_date) = 2

              )

go

--вывести все продажи за 2021-12-31--

--естественное соединение c представлением Product\_Type\_view--

select distinct Product\_type

from Product\_Type\_view inner join Chek\_infoChek on Product\_Type\_view.cod\_prod = Chek\_infoChek.cod\_prod

where Purc\_date = '2021-12-31'

go

--то же, но с подзапросами--

select Product\_type

from Product\_Type

where Cod\_type *in* (select Cod\_type

               from Product

               where cod\_prod *in* (select cod\_prod

                             from Chek\_infoChek

                             where Purc\_date = '2021-12-31'

                             )

              )

go

/\*сложный запрос с группировкой и функцией агрегации\*/

--вывести количекство продаж каждого типа продукции за 2021-12-31--

select Product\_type, count(Product\_type) *as* amount\_of\_products

from Product\_Type\_view inner join Chek\_infoChek on Product\_Type\_view.cod\_prod = Chek\_infoChek.cod\_prod

where Purc\_date = '2021-12-31'

group by Product\_type

go

--+ Часть в подзапросах...--

--5 подзапросов--

/\*Это сложный запрос с подзапросом\*/

--Вывести технические характеристики всех имеющихся компьютеров--

select Prod\_name *as* Product, CPU\_freq *as* CPU, RAM, HDD, CD, Applic *as* Aplication

from Computer inner join Product on Computer.cod\_prod = Product.cod\_prod

where Cod\_type = (select Cod\_type

              from Product\_type

              where Product\_type = 'PC'

             )

go

/\*Это сложный запрос с подзапросом\*/

--Вывести информацию из всех чеков, оплаченных наличкой--

select Prod\_name *as* Product, Purc\_date *as* Purcasing\_date, Gen\_price *as* General\_price

from Chek\_infoChek inner join Product on Chek\_infoChek.cod\_prod = Product.cod\_prod

where Check\_code *in* (select Check\_code --*in*, а не =, так как по итогу создастся отношение (Pay\_type = 'cash' > 2-х кортежей) => в Chek\_infoChek передастся уже не 1 значение, а > 1 (2)--

               from Chek

               where Pay\_code = (select Pay\_code

                             from Payment

                             where Pay\_type = 'cash'

                            )

               )

go

--посмотреть, какие товары поставляет поставщик XMoldova--

select Product\_type

from Product\_type

where Cod\_type *in* (select Cod\_type

               from Product

               where cod\_prod *in*(select cod\_prod

                            from consignment

                            where Sup\_code = (select Sup\_code

                                          from supplier

                                          where Sup\_name = 'XMoldova'

                                         )

                           )

             )

go

--вывести тип продукции, которую производит компания LG--

select Product\_type

from Product\_type

where Cod\_type *in* (select Cod\_type

               from Product

               where Man\_cod = (select Man\_code

                           from manufacturer

                           where Man\_name = 'LG'

                           )

             )

go

/\*Подзапрос со сложным запросом с функцией агрегации\*/

--вывести самый дорогой товар--

select Prod\_name *as* Product, Product\_type, Price

from Product inner join Product\_type on Product.Cod\_type = Product\_type.Cod\_type

where price = (select max(price)

            from Product)

go

--Вывести технические характеристики Redmi 5 pro--

select CPU\_freq *as* CPU, RAM, Intern\_mem *as* Memory, Applic *as* Aplication

from Phone

where cod\_prod = (select cod\_prod

              from Product

              where Prod\_name = 'Redmi 5 pro'

             )

go

--То же, но с естественным соединением--

select CPU\_freq *as* CPU, RAM, Intern\_mem *as* Memory, Applic *as* Aplication

from Product inner join Phone on Phone.cod\_prod = Product.cod\_prod

where Prod\_name = 'Redmi 5 pro'

go

--5 запросов на группировку и агрегацию информации--

--посчитать количество товаров, поставленных в каждом месяце--

select *month*(Consignment\_date) *as* mounth, sum(prod\_number) *as* supply\_amount

from consignment

group by *month*(Consignment\_date)

go

--посчитать общую цену всех товаров, поставленных в феврале (02)--

select sum(price\*prod\_number) *as* Total\_suply\_price

from consignment

where *month*(Consignment\_date) = 2

go

--подстчитать число каждой продукции имеющейся в магазине--

select Product\_type *as* Production, count(\*) *as* Products\_amount

from Product\_type inner join Product on Product\_type.Cod\_type = Product.Cod\_type

group by Product\_type

go

--вывести количество продаж за 2021-12-31--

select count(cod\_prod) *as* amount\_of\_products

from Chek\_infoChek

where Purc\_date = '2021-12-31'

go

--подсчитать количество покупателей для каждого способа оплаты--

/\*Создадим представление\*/

create view client\_payment\_view *as*

select id, C\_Adres, C\_Surname, C\_Name, C\_FatherName, Pay\_type

from Payment inner join Pasport\_data on Payment.Pay\_code = Pasport\_data.Pay\_code

go

select \*

from client\_payment\_view

go --создаем представление так как Pay\_type не входит в Pasport\_data и в результате не произойдет агрегация--

select Pay\_type, count(\*) *as* amount\_of\_clients

from client\_payment\_view

group by Pay\_type

go