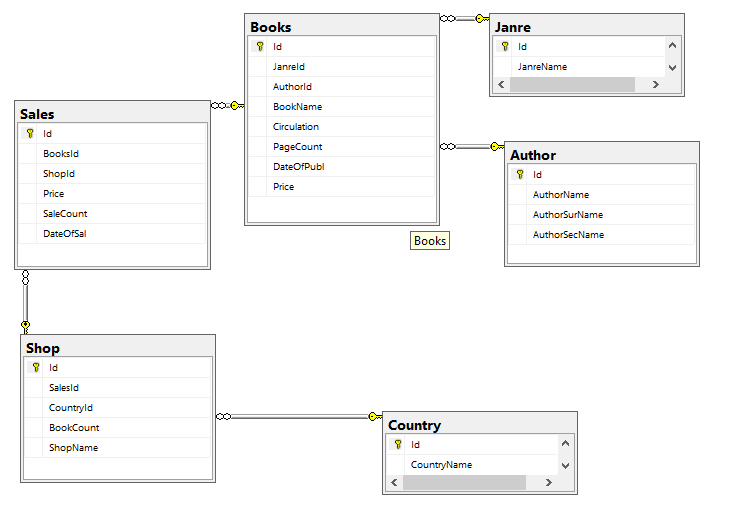
# Стурктура БД



Скрипт по созданию и заполнению («HT\_2.sql»)

use [master];

go

if db\_id('PublishingHouse') is not null

begin

drop database [PublishingHouse];

end

go

create database [PublishingHouse];

go

use [PublishingHouse];

go

create table [Janre]

(

[Id] int primary key identity not null,

[JanreName] nvarchar(100) not null UNIQUE check([JanreName]!=N'')

);

go

create table [Author]

(

[Id] int primary key identity not null,

[AuthorName] nvarchar(100) not null check([AuthorName]!=N''),

[AuthorSurName] nvarchar(100) not null check([AuthorSurName]!=N''),

[AuthorSecName] nvarchar(100) not null check([AuthorSecName]!=N'')

);

go

create table [Books]

(

[Id] int primary key identity not null,

[JanreId] int not null,

[AuthorId] int not null,

[BookName] nvarchar(100) not null ,

[Сirculation] int not null check([Сirculation]>0),

[PageCount] int not null check([PageCount]>0),

[DateOfPubl] date not null,

[Price] money not null

);

go

create table [Sales]

(

[Id] int primary key identity not null,

[BooksId] int not null,

[ShopId] int not null,

[Price] money not null check([Price]>=0),

[SaleCount] int not null check([SaleCount]>=0),

[DateOfSal] date not null

);

go

create table [Shop]

(

[Id] int primary key identity not null,

[SalesId] int not null,

[CountryId] int not null,

[BookCount] int not null check([BookCount]>=0),

[ShopName] nvarchar(100) not null check([ShopName]!=N'')

);

go

create table [Country]

(

[Id] int primary key identity not null,

[CountryName] nvarchar(100) not null UNIQUE check([CountryName]!=N'')

);

go

--connections

alter table [Books]

add foreign key ([JanreId]) references [Janre]([Id]);

go

alter table [Books]

add foreign key ([AuthorId]) references [Author]([Id]);

go

alter table [Sales]

add foreign key ([BooksId]) references [Books]([Id]);

go

alter table [Sales]

add foreign key ([ShopId]) references [Shop]([Id]);

go

alter table [Shop]

add foreign key ([CountryId]) references [Country]([Id]);

go

--Zapolnenie

INSERT into [Janre] VALUES ('Научная фантастика'),

('Боевая фантастика'),

('Альтернативная история'),

('Фэнтези');

INSERT into [Author] VALUES

('Светлана','Жданова','Ивановна'),

('Елена', 'Звездная','Борисовна'),

('Владимир', 'Поселягин','Петрович'),

('Сергей', 'Тармашев','Николаевич'),

('Еадежда','Кузьмина','Кирриловна');

INSERT into [Books] VALUES (4,1,'Невеста демона',150000,310,'2017.2.18',25),

(4,1,'Крылья феникса',132000,325,'2017.12.13',24),

(4,2,'Академия магии',14100,250,'2018.3.12',23),

(3,3,'Я танкист',152000,600,'2018.2.10',25),

(3,3,'Я летчик',93000,310,'2017.4.28',22),

(2,2,'Дочь воина, или Кадеты не сдаются',95000,450,'2010.8.22',20),

(2,2,'Хелл. Обучение наемницы',45000,280,'2013.4.24',21),

(1,4,'Корпорация',133000,610,'2017.4.19',30),

(3,4,'Протопип',85000,215,'2018.2.10',20.6),

(3,3,'Вечный',45000,310,'2018.7.11',21.9);

INSERT into [Country] VALUES ('Украина'),

('Россия'),

('Беларусь'),

('Казахстан'),

('Грузия');

INSERT into [Shop] VALUES (1, 1, 600,'Либрусек'),

(2, 2, 600,'Либрусек'),

(3, 3, 600,'Либрусек'),

(4, 4, 600,'Либрусек'),

(5, 5, 600,'Либрусек'),

(6, 2, 620,'Колумб'),

(7, 3, 630,'Колумб'),

(8, 4, 640,'Колумб'),

(9, 5, 610,'Колумб'),

(10, 1, 700,'Колумб'),

(11, 2, 600,'Книжный клуб семейного досуга'),

(12, 3, 600,'Книжный клуб семейного досуга'),

(13, 4, 600,'Книжный клуб семейного досуга'),

(14, 5, 600,'Книжный клуб семейного досуга'),

(15, 1, 600,'Книжный клуб семейного досуга'),

(16, 2, 620,'Плимут'),

(17, 3, 630,'Плимут'),

(18, 4, 640,'Плимут'),

(19, 5, 610,'Плимут'),

(20, 1, 700,'Плимут'),

(21, 1, 550,'Звезда'),

(22, 2, 570,'Звезда'),

(23, 3, 580,'Звезда'),

(24, 4, 660,'Звезда'),

(25, 5, 690,'Звезда'),

(26, 2, 630,'Либрусек'),

(27, 3, 640,'Либрусек'),

(28, 4, 690,'Либрусек'),

(29, 5, 700,'Либрусек'),

(30, 1, 750,'Либрусек'),

(31, 1, 600,'Колумб'),

(32, 2, 600,'Колумб'),

(33, 3, 600,'Колумб'),

(34, 4, 600,'Колумб'),

(35, 5, 600,'Колумб'),

(36, 2, 620,'Книжный клуб семейного досуга'),

(37, 3, 630,'Книжный клуб семейного досуга'),

(38, 4, 640,'Книжный клуб семейного досуга'),

(39, 5, 610,'Книжный клуб семейного досуга'),

(40, 1, 700,'Книжный клуб семейного досуга'),

(41, 2, 620,'Плимут'),

(42, 3, 630,'Плимут'),

(43, 4, 640,'Плимут'),

(44, 5, 610,'Плимут'),

(45, 1, 700,'Плимут'),

(46, 1, 550,'Звезда'),

(47, 2, 570,'Звезда'),

(48, 3, 580,'Звезда'),

(49, 4, 660,'Звезда'),

(50, 5, 690,'Звезда'),

(51, 1, 610,'Либрусек'),

(52, 2, 630,'Либрусек'),

(53, 3, 620,'Либрусек'),

(54, 4, 650,'Либрусек'),

(55, 5, 640,'Либрусек'),

(56, 2, 630,'Колумб'),

(57, 3, 660,'Колумб'),

(58, 4, 670,'Колумб'),

(59, 5, 680,'Колумб'),

(60, 1, 760,'Колумб'),

(61, 2, 670,'Книжный клуб семейного досуга'),

(62, 3, 680,'Книжный клуб семейного досуга'),

(63, 4, 680,'Книжный клуб семейного досуга'),

(64, 5, 620,'Книжный клуб семейного досуга'),

(65, 1, 640,'Книжный клуб семейного досуга'),

(66, 2, 660,'Плимут'),

(67, 3, 634,'Плимут'),

(68, 4, 623,'Плимут'),

(69, 5, 612,'Плимут'),

(70, 1, 720,'Плимут'),

(71, 1, 534,'Звезда'),

(72, 2, 576,'Звезда'),

(73, 3, 589,'Звезда'),

(74, 4, 664,'Звезда'),

(75, 5, 692,'Звезда'),

(76, 2, 730,'Либрусек'),

(77, 3, 640,'Либрусек'),

(78, 4, 690,'Либрусек'),

(79, 5, 605,'Либрусек'),

(80, 1, 650,'Либрусек'),

(81, 1, 600,'Колумб'),

(82, 2, 600,'Колумб'),

(83, 3, 710,'Колумб'),

(84, 4, 630,'Колумб'),

(85, 5, 640,'Колумб'),

(86, 2, 650,'Книжный клуб семейного досуга'),

(87, 3, 660,'Книжный клуб семейного досуга'),

(88, 4, 670,'Книжный клуб семейного досуга'),

(89, 5, 680,'Книжный клуб семейного досуга'),

(90, 1, 790,'Книжный клуб семейного досуга'),

(91, 2, 620,'Плимут'),

(92, 3, 610,'Плимут'),

(93, 4, 650,'Плимут'),

(94, 5, 660,'Плимут'),

(95, 1, 770,'Плимут'),

(96, 1, 540,'Звезда'),

(97, 2, 530,'Звезда'),

(98, 3, 520,'Звезда'),

(99, 4, 650,'Звезда'),

(100, 5, 660,'Звезда');

INSERT into [Sales] VALUES

(1,2,39,34,'2018.9.4'),

(1,3,38,71,'2018.11.15'),

(1,4,35,39,'2018.11.26'),

(1,5,35,25,'2018.10.18'),

(2,1,34,43,'2018.9.6'),

(2,2,36,66,'2018.11.18'),

(2,3,38,48,'2018.11.3'),

(2,4,37,75,'2018.10.25'),

(2,5,33,15,'2018.11.25'),

(3,1,42,42,'2018.11.30'),

(3,2,38,71,'2018.11.26'),

(3,3,36,29,'2018.10.29'),

(3,4,35,74,'2018.9.24'),

(3,5,40,50,'2018.11.15'),

(4,1,42,22,'2018.9.28'),

(4,2,39,66,'2018.11.24'),

(4,3,41,17,'2018.9.2'),

(4,4,39,50,'2018.11.10'),

(4,5,39,34,'2018.11.12'),

(5,1,36,37,'2018.11.25'),

(5,2,34,71,'2018.10.14'),

(5,3,39,51,'2018.10.18'),

(5,4,33,63,'2018.11.25'),

(5,5,38,74,'2018.10.16'),

(6,1,42,71,'2018.11.13'),

(6,2,38,49,'2018.9.19'),

(6,3,37,24,'2018.11.13'),

(6,4,39,71,'2018.11.26'),

(6,5,37,66,'2018.10.26'),

(7,1,36,72,'2018.9.8'),

(7,2,36,27,'2018.9.7'),

(7,3,38,40,'2018.11.8'),

(7,4,37,80,'2018.9.23'),

(7,5,41,51,'2018.10.19'),

(8,1,41,49,'2018.11.14'),

(8,2,38,25,'2018.9.7'),

(8,3,40,69,'2018.9.27'),

(8,4,33,36,'2018.11.2'),

(8,5,33,42,'2018.10.18'),

(9,1,39,66,'2018.10.11'),

(9,2,36,46,'2018.10.7'),

(9,3,40,63,'2018.9.28'),

(9,4,40,30,'2018.11.13'),

(9,5,35,67,'2018.10.24'),

(10,1,35,39,'2018.10.23'),

(10,2,42,30,'2018.9.21'),

(10,3,42,73,'2018.9.13'),

(10,4,34,56,'2018.10.29'),

(10,5,55,32,'2018.10.1');

# 

# Выполнение ДЗ

1. Показать все книги трех произвольных авторов. (Есть подозрение, что для больштх таблиц такой способ не годится)

SELECT b.BookName as 'Название книги', a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName 00as 'Автор'

FROM

Books b, Author a

WHERE b.AuthorId = a.Id and a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName in (SELECT Top 3

a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор'

FROM

Books b, Author a

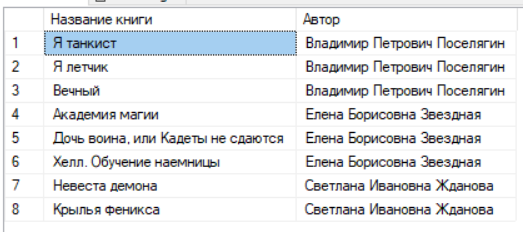
WHERE b.AuthorId = a.Id

Group by a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName

ORDER BY RAND()

)

order by 'Автор'



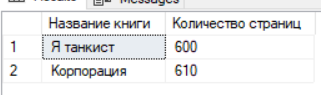
2. Показать все книги, в которых количество страниц больше 500, но меньше 650.

SELECT b.BookName as 'Название книги', b.PageCount as 'Количество страниц'

FROM Books b

where b.PageCount>500 and b.PageCount<650

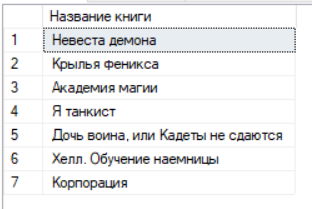
order by b.PageCount

  
3. Необходимо вывести все названия книг, в которых первая буква или А, или С.

SELECT b.BookName as 'Название книги'

FROM Books b

where b.BookName like '%[АС]%';

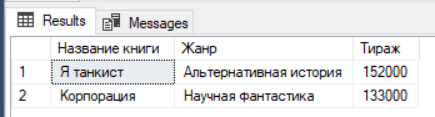


4. Показать названия книг, тематика которых не  
"Фэнтези" и тираж которых >=100 000 экземпляров.

SELECT B.BookName as 'Название книги', J.JanreName as 'Жанр', B.Сirculation as 'Тираж'

FROM BOOKS B, JANRE J

WHERE j.Id = b.JanreId and J.JanreName != 'Фэнтези' and b.Сirculation>1=00000

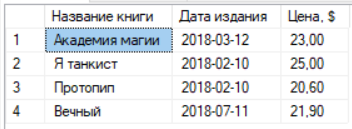


5. Показать все книги-новинки, цена которых ниже $30. (Новинкой будет считаться книга, которая была из дана на протяжении последнего года).

SELECT B.BookName as 'Название книги' , b.DateOfPubl as 'Дата издания', b.Price as 'Цена, $'

FROM Books B

where YEAR(GETDATE())-year(b.DateOfPubl)<=0 and b.Price <30



6. Показать книги, в названиях которых есть слово "Я", но нет слова "танкист".

SELECT B.BookName as 'Название книги'

FROM Books B

WHERE B.BookName LIKE 'Я [^танкист]%'

  
7. Вывести названия книг, тематику, автора (полное имя), цена одной страницы которых менее 6 центов.

SELECT B.BookName as 'Название книги', J.JanreName as 'Жанр', b.Price/b.PageCount as 'Цена',

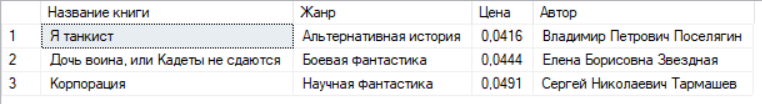
a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор'

FROM [Books] B

JOIN [Janre] J ON j.Id = b.JanreId

JOIN [Author] A ON a.Id = b.AuthorId

WHERE b.Price/b.PageCount<0.06



8. Вывести информацию обо всех книгах, в имени которых больше 4-х слов.

SELECT B.BookName as 'Название книги'

From Books b

WHERE b.bookname like '% % % %';

  
9. Вывести на экран все книги, их авторов и цены их продажи в у.е., дата продажи которых находится в диапазоне 01/01/2018 по сегодняшнюю дату.

SELECT B.BookName as 'Название книги', b.Price as 'Цена для диллеров, $',

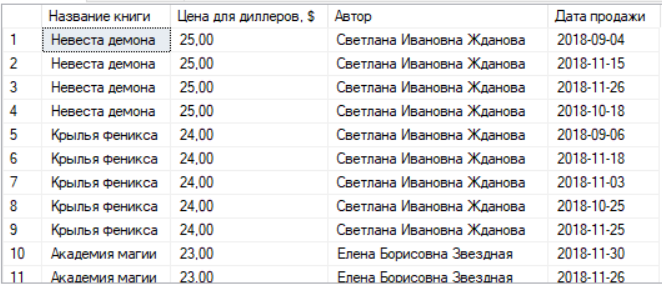
a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор' , s.DateOfsal as 'Дата продажи'

FROM [Sales] S

join [Books] B on b.Id = s.BooksId

join [Author] A on b.AuthorId=a.Id

Where s.DateOfSal>'01/01/2018'



10.Показать всю информацию по продажам книг в следующем виде:  
■ название книги;

■ тематик, которые касаются "Computer Science";  
■ автор книги (полное имя);  
■ цена продажи книги;  
■ имеющееся количество продаж данной книги;  
■ название магазина, который находится не в Украине  
и не в Канаде и продает эту книгу.

Select B.BookName as 'Название книги', j.JanreName as 'Жанр',

a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор' ,

b.Price as 'Цена для диллеров, $', sum (s.SaleCount) as 'Количество продаж',

sh.ShopName as 'Название магазина', c.CountryName as 'Страна'

From [Books] b

join [Janre] j on b.JanreId = j.Id

join [Author] a on a.Id = b.AuthorId

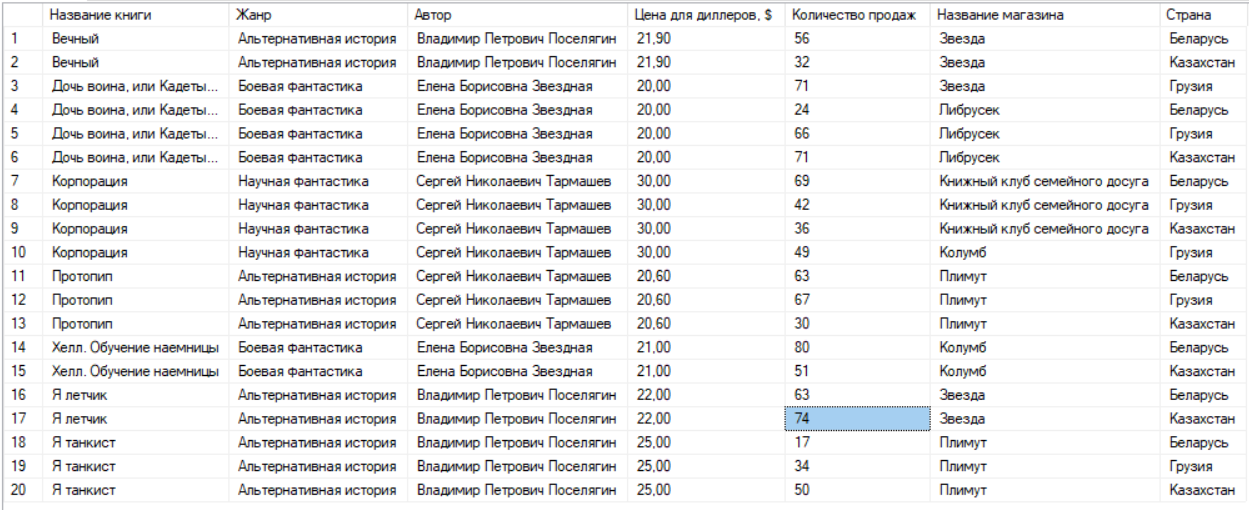
join [Sales] s on b.Id = s.BooksId

join [Shop] sh on sh.SalesId=s.Id

join [Country] c on c.Id=sh.CountryId

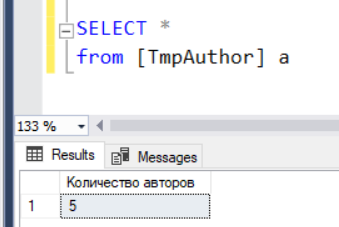
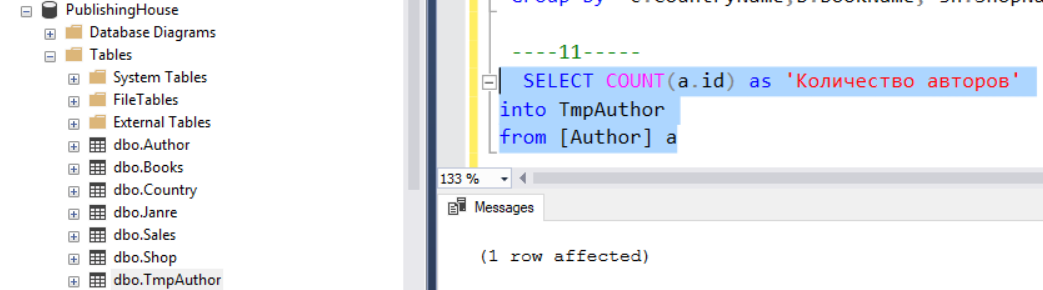
Where j.JanreName!='Фэнтези' and c.CountryName!='Украина' and c.CountryName!='Россия'

Group by B.BookName , j.JanreName , a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName ,

b.Price, sh.ShopName, c.CountryName

11.Показать количество авторов в базе данных. Резульат сохранить в другую таблицу.

SELECT COUNT(a.id) as 'Количество авторов'

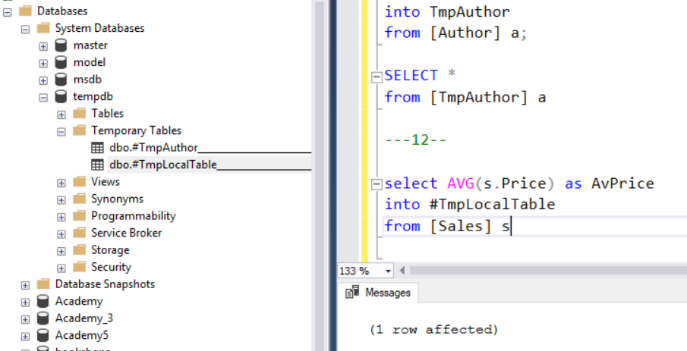
into TmpAuthor from [Author] a  


12.Показать среднеарифметическую цену продажи всех книг. Результат сохранить в локальную временную таблицу.

select AVG(s.Price) as AvPrice

into #TmpLocalTable

from [Sales] s

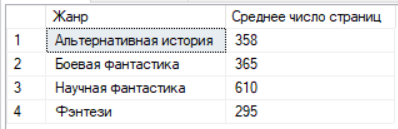
  
13.Показать тематики книг и сумму страниц по каждой из них.

SELECT j.JanreName as 'Жанр', avg(b.PageCount) as 'Среднее число страниц'

FROM [Janre] J

join [Books] b on b.JanreId= j.Id

Group by j.JanreName;



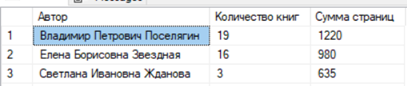
14.Вывести количество книг и сумму страниц этих книг по каждому из первых трех (!) авторов в базе данных.

Select top 3 a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор',

sum(b.Id) as 'Количество книг', sum(b.PageCount) 'Сумма страниц'

From [Books] b

join [Author] a on b.AuthorId= a.Id

Group by a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName

15.Вывести информацию о книгах по "Computer Science" с наибольшим количеством страниц.

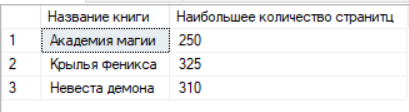
Select b.BookName as 'Название книги', max(b.PageCount) as 'Наибольшее количество странитц'

From [Janre]j

join [Books] b on b.JanreId = j.Id

where j.JanreName='Фэнтези'

Group by b.BookName



16.Показать авторов и самую старую книгу по каждому из них. Результат сохранить в глобальную временную таблицу.(не совсем то, что нужно выполнено)

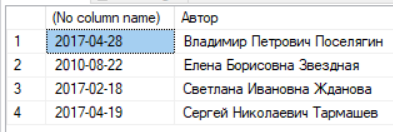
---тут получаем только мин дату публикации и авторов,без названия книги

select min(b.DateOfPubl) ,a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор'

From [Author] a

join [Books] b on b.AuthorId = a.Id

group by a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName

  
17.Показать на экран среднее количество страниц по каждой из тематик, при этом показать только тематики, в которых среднее количество более 400.

select avg(b.PageCount) as 'Среднее кол-во стр по тематике', j.JanreName as 'Жанр'

from [Janre] j

join [Books] b on b.JanreId= j.Id

group by j.JanreName

having avg(b.PageCount)>400;



18.Показать на экран сумму страниц по каждой из тематик, при этом учитывать только книги с количеством страниц более 300, но учитывать при этом только 3 тематики, например "Computer Science", "Science Fiction" и "Web Technologies".

select sum(b.PageCount) as 'Сумма страниц', j.JanreName as 'Жанр'

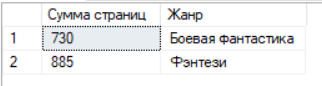
from [Janre] j

join [Books] b on b.JanreId= j.Id

where j.JanreName in('Фэнтези','Боевая фантастика')

group by j.JanreName

having sum(b.PageCount)>300;



19.Показать количество проданных книг по каждому магазину, в промежутке от 01/01/2007 до сегодняшней даты.

select sum (s.SaleCount) as BookSalCount , s.DateOfSal as DateSale, sh.ShopName as Shop

into #TMPTab\_19

from [Sales]s

join [Shop] sh on sh.SalesId = s.Id

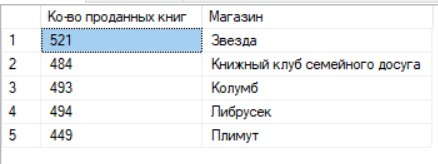
where s.DateOfSal>'01/10/2018'

group by s.DateOfSal, sh.ShopName;

select sum (tmp.BookSalCount) as 'Ко-во проданных книг', tmp.Shop as 'Магазин'

from [#TMPTab\_19]tmp

group by tmp.Shop;



20.Вывести всю информацию о работе магазинов: что, когда, сколько и кем было продано, а также указать страну, где находится магазин. Результат сохранить в другую таблицу.

Select sh.ShopName as 'Название магазина', B.BookName as 'Название книги',

j.JanreName as 'Жанр', a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Автор' ,

sum(s.Price) as 'Выручка, $', sum (s.SaleCount) as 'Количество продаж',

c.CountryName as 'Страна'

From [Books] b

join [Janre] j on b.JanreId = j.Id

join [Author] a on a.Id = b.AuthorId

join [Sales] s on b.Id = s.BooksId

join [Shop] sh on sh.SalesId=s.Id

join [Country] c on c.Id=sh.CountryId

Group by sh.ShopName,B.BookName , j.JanreName , a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName ,

c.CountryName

