[Базы данных 2](#_Toc530566523)

[Стурктура БД PublishingHouse 2](#_Toc530566524)

[Скрипт по созданию и заполнению 2](#_Toc530566525)

[Структура БД BookShops 7](#_Toc530566526)

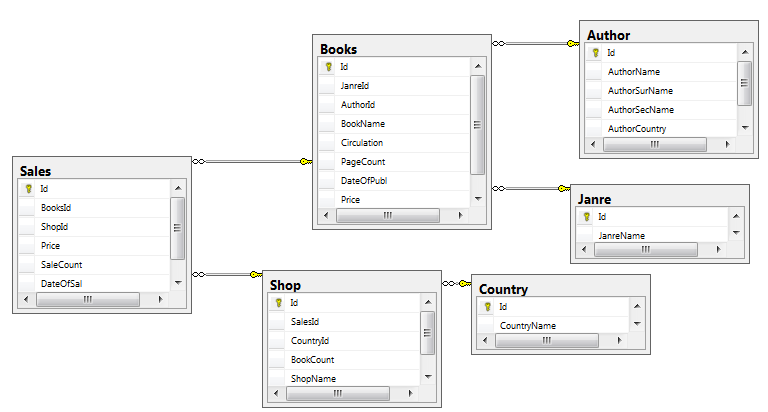
[Скрипт по созданию BookShops 7](#_Toc530566527)

[Выполнение ДЗ 11](#_Toc530566528)

# Базы данных

При выполнении ДЗ была произведена работа сразу с двумя БД : PublishingHouse и Bookshops. Так было сподручнее.

## Стурктура БД PublishingHouse



Скрипт по созданию и заполнению PublishingHouse(«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''),

[AuthorCountry] nvarchar(100) not null check([AuthorCountry]!=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,

[IsPublish] nvarchar(3) 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','Россия'),

('Елена', 'Звездная','Борисовна','США'),

('Владимир', 'Поселягин','Петрович','Россия'),

('Сергей', 'Тармашев','Николаевич','Беларусь'),

('Надежда','Кузьмина','Кирриловна','Украина');

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

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

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

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

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

(2,2,'Дочь воина, или Кадеты не сдаются',95000,450,'2010.8.22',20,'No'),

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

(1,4,'Корпорация',133000,610,'2017.4.19',30,'No'),

(3,4,'Протопип',85000,215,'2018.2.10',20.6,'No'),

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

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'),

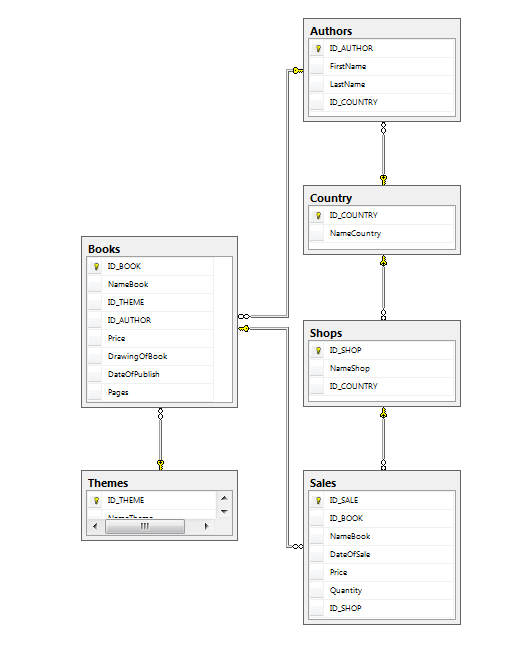
(10,2,42,30,'2018.9.21'),

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

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

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

## Структура БД BookShops



## Скрипт по созданию BookShops

--

create database bookshops

go

use bookshops

go

--

create default dch as 'no data'

go

create type myStr from nvarchar(50) not null

go

create type picS from varchar(256) not null

go

create table Themes(

ID\_THEME int primary key identity,

NameTheme myStr,

)

go

insert into themes values

('Базы данных'),

('Интернет'),

('Hardware'),

('Операционные системы') ,

('Сети') ,

('Программирование') ,

('Программное обеспечение') ,

('Прочее') ,

('Цифровая обработка сигналов')

go

create table Country

(

ID\_COUNTRY int primary key identity,

NameCountry myStr

)

go

insert into Country values('USA'),('Германия'),('СНГ'),('Финляндия'),('Великобритания')

go

create table Authors(

ID\_AUTHOR int primary key identity,

FirstName myStr,

LastName myStr,

ID\_COUNTRY int not null,

foreign key (ID\_COUNTRY) references Country(ID\_COUNTRY)

on delete cascade

on update cascade

)

go

insert into Authors values

( 'Бек', 'Кент', 1 ),

( 'Савин', 'Роман', 1 ),

( 'Шкритек' , 'Пауль', 2 ),

( 'Циммерман','Филипп', 3 ),

( 'А.И.','Кубин', 3 ),

( 'Торвальдс','Линус', 4 ),

( 'ДеМарко', 'Том', 1),

( 'Хэррис', 'Сара Л', 5),

( 'Паундстоун','Уильям', 1),

( 'Йордан', 'Эдвард', 1),

( 'Фаулер','Мартин', 1),

( 'Kaspersky','Kris', 1),

( 'Макконнелл','Стив', 1)

go

create table Shops

(

ID\_SHOP int primary key identity,

NameShop myStr,

ID\_COUNTRY int not null,

foreign key (ID\_COUNTRY) references Country(ID\_COUNTRY)

on delete cascade

on update cascade

)

go

insert into Shops values

( 'us books', 1 ),

( 'deutch books', 2 ),

( 'суничка', 3 ),

( 'you ', 4 ),

( 'great reading', 1 ),

( 'big great reading', 5 )

go

create table Books(

ID\_BOOK int primary key identity,

NameBook myStr,

ID\_THEME int not null,

foreign key (ID\_THEME) references Themes(ID\_THEME)

on delete cascade

on update cascade,

ID\_AUTHOR int not null,

foreign key (ID\_AUTHOR) references Authors(ID\_AUTHOR)

on delete cascade

on update cascade,

Price money not null,

DrawingOfBook pics,

DateOfPublish date not null,

Pages int not null

)

go

insert into Books values

( 'Экстрим программирование',6, 1 , 451.20, 'no pic', '2018-02-05', 6 ),

( 'tестирование dot com',6, 2 , 270.20, 'no pic', '2007-01-01', 2 ),

( 'Справочник по звуковой схемотехнике', 9, 3 , 150.65, 'no pic', '1991-01-01', 125 ),

( 'Введение в криптографию (ЛП)',8, 4 , 51.20, 'no pic', '1991-01-01', 18 ),

( '1000 и 1 секрет BIOS ',3, 5 , 120.65, 'no pic', '2007-01-01', 98 ),

( 'Just for Fun', 4, 6 , 320.65, 'no pic', '2001-01-01', 135 ),

( 'Deadline. Роман об управлении проектами', 6, 7 , 170.00, 'no pic', '2006-01-01', 61 ),

( 'Цифровая схемотехника и архитектура компьютера', 3, 8 , 150.00, 'no pic', '2012-01-01', 61 ),

( 'Как сдвинуть гору Фудзи?', 8, 9, 250.00, 'no pic', '2004-01-01', 206),

( 'Путь камикадзе [Смертельный марш]?', 6, 10, 236.00, 'no pic', '2003-01-01', 41),

( 'Рефакторинг', 6, 11, 232.00, 'no pic', '2003-01-01', 135),

( 'Тонкости дизассемблирования', 6, 12, 182.00, 'no pic', '2012-01-01', 195),

( 'Совершенный код', 6, 13, 182.00, 'no pic', '2005-01-01', 195)

go

create table Sales(

ID\_SALE int primary key identity,

ID\_BOOK int not null,

NameBook myStr,

foreign key (ID\_BOOK) references Books(ID\_BOOK)

on delete no action

on update no action,

DateOfSale date not null,

Price money not null,

Quantity int not null,

ID\_SHOP int not null,

foreign key (ID\_SHOP) references Shops(ID\_SHOP)

on delete no action

on update no action

)

go

insert into Sales values

( 11 , (select NameBook from books where id\_book = 11 ), '2018-07-10',(select price from books where id\_book = 11), 1 ,6),

( 2 , (select NameBook from books where id\_book = 2 ), '2018-07-10',(select price from books where id\_book = 2) , 1 ,1),

( 4 , (select NameBook from books where id\_book = 4 ), '2018-07-10',(select price from books where id\_book = 4), 1 ,2),

( 13 , (select NameBook from books where id\_book = 13 ), '2018-07-10',(select price from books where id\_book = 13),1 ,6),

( 1 , (select NameBook from books where id\_book = 1 ), '2018-07-10', (select price from books where id\_book = 1) , 1 ,1),

( 7 , (select NameBook from books where id\_book = 7 ), '2018-07-10',(select price from books where id\_book = 7), 1 ,4),

( 8 , (select NameBook from books where id\_book = 8 ), '2018-07-10',(select price from books where id\_book = 8), 1 ,4),

( 9 , (select NameBook from books where id\_book = 9 ), '2018-07-10',(select price from books where id\_book = 9), 1 ,5),

( 5 , (select NameBook from books where id\_book = 5 ), '2018-07-10',(select price from books where id\_book = 5), 1 ,3),

( 10 , (select NameBook from books where id\_book = 10 ), '2018-07-10',(select price from books where id\_book = 10),1 ,5),

( 6 , (select NameBook from books where id\_book = 6 ), '2018-07-10',(select price from books where id\_book = 6), 1 ,3),

( 3 , (select NameBook from books where id\_book = 3 ), '2018-07-10',(select price from books where id\_book = 3), 1 ,2),

( 12 , (select NameBook from books where id\_book = 12 ), '2018-07-10',(select price from books where id\_book = 12),1 ,6)

go

select \* from authors

select \* from books

select \* from country

select \* from sales

select \* from shops

select \* from themes

go

# Выполнение ДЗ

1. Вывести все книги, которые продаются более чем одним магазином

Запрос для БД BookHouse

select b.NameBook, count (sh.NameShop) as 'Shop count'

into #tmp\_1

from [books]b join [sales] s on b.ID\_BOOK = s.ID\_BOOK

join [Shops] sh on sh.ID\_SHOP = s.ID\_SHOP

group by b.NameBook ;

select b.NameBook, sh.NameShop

from [books]b join [sales] s on b.ID\_BOOK = s.ID\_BOOK

join [Shops] sh on sh.ID\_SHOP = s.ID\_SHOP

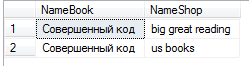
where b.NameBook in(

select tmp.NameBook

from [#tmp\_1]tmp

where tmp.[Shop count]>1

)



2. Вывести только тех авторов, чьи книги продаются больше, чем книги авторов США. Тут выполнен в БД PublishingHouse

Select sum(s.SaleCount) as 'SumSalesCount', b.BookName as 'Book name'

into #TMPTab\_1

from [Author] a

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

join [sales] s on b.Id=s.BooksId and a.AuthorCountry = 'США'

Group by b.bookname

select max (tmp.SumSalesCount)

from [#TMPTab\_1]tmp

Select sum(s.SaleCount) as 'CountSaleBooks', a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Author',

b.BookName as 'BookName', a.AuthorCountry as 'AuthorCountry'

into #TMPTab\_2

from [Author] a

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

join [sales] s on b.Id=s.BooksId and a.AuthorCountry!='США'

Group by a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName ,b.BookName, a.AuthorCountry;

select tmp2.CountSaleBooks as 'Количество проданых книг', tmp2.Author as 'Автор', tmp2.Bookname as 'Название книги',

tmp2.AuthorCountry as 'Страна автора'

from [#TMPtab\_2]tmp2

where tmp2.CountSaleBooks>(select max (tmp.SumSalesCount)from [#TMPTab\_1]tmp)



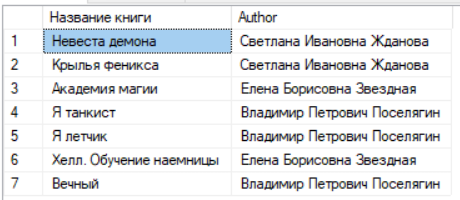
3. Вывести всех авторов, которые существуют в базе данных с указанием (при наличии) их книг, которые издаются издательством.

select b.BookName as 'Название книги', a.AuthorName+' '+a.AuthorSecName +' '+a.AuthorSurName as 'Author'

from [Author]a

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

where b.IsPublish='Yes';



4. С помощью подзапросов найдите всех авторов, которые живут в странах, где есть магазин, который продает их книги. Отсортировать выборку по фамилии  
автора.

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

a.AuthorCountry as 'Страна автора', sh.ShopName as 'Магазин', c.CountryName as 'Страна магазина'

From [Author] a

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

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

join [Shop]sh on s.ShopId=sh.CountryId

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

where a.AuthorCountry=c.CountryName and a.AuthorCountry in (

Select c.CountryName as 'Страна магазина'

From [Author] a

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

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

join [Shop]sh on s.ShopId=sh.CountryId

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

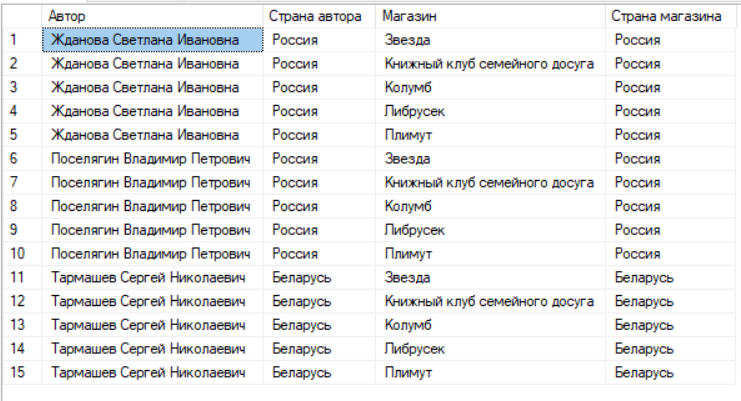
where a.AuthorCountry=c.CountryName

group by c.CountryName

)

group by a.AuthorSurName+' '+a.AuthorName+' '+a.AuthorSecName , a.AuthorCountry , sh.ShopName , c.CountryName

order by 1



5. Доказать, что книги тематики, например, "Computer Science" выпускаются самым большим тиражом. Примечание! Доказательством будет NULL значений, иначе – книги наиболее реализуемых тематик.

Условие задания - огонь. Интерпретировал его так: вывести лидера в категории темтики по тиражу . Также показать список всех темтик с общим тиражом по каждой, упорядочив их в порядке убывания

Для первой части условия

Select top 1 j.JanreName as 'Жанр', sum(b.Сirculation) as 'Суммарный тираж'

From [Author]a

join [Janre]j on a.Id=j.Id

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

Group by j.JanreName

order by 2 desc



Для второй части условия

Select j.JanreName as 'Жанр', sum(b.Сirculation) as 'Суммарный тираж'

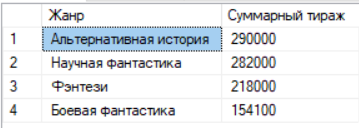
From [Author]a

join [Janre]j on a.Id=j.Id

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

Group by j.JanreName

order by 2 desc



6. Сформируйте объединения из трех запросов.

Первое будет выводить список авторов, тематики книг которых, например, 'Science Fiction';

второе – список авторов, которые издавали свои книги в 2014 году;  
третья выводит список самых дорогих авторов. В двух последних запросах сохраните дубликаты. Отсо ртировать выборку по фамилии автора по убыванию.

Выполненны первые 2 условия, третье в объединение не вошло. Также не ясно, что делать с условием, которое требует создания дубликата.

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

j.JanreName as 'Жанр', b.BookName as 'Книга', b.DateOfPubl 'Дата публикации'

From [Author]a

join [Janre]j on a.Id=j.Id

join [Books]b on a.Id=b.AuthorId and j.JanreName='Фэнтези'

Union

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

j.JanreName as 'Жанр', b.BookName as 'Книга', b.DateOfPubl 'Дата публикации'

From [Author]a

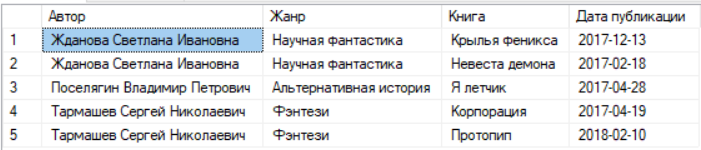
join [Janre]j on a.Id=j.Id

join [Books]b on a.Id=b.AuthorId and year(b.DateOfPubl)=2017

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

j.JanreName, b.BookName , b.DateOfPubl

order by 1



Результат 3-го подзадания текущего запроса, каким образом его нужно было заставить работать в объединении с первыми двумя – не понятно

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

sum(b.Сirculation\*b.Price) as 'Стоимость тиража'

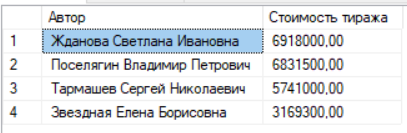
From [Author]a

join [Janre]j on a.Id=j.Id

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

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

order by 2 desc

  
7. Составить отчет о том, какие магазины реализовали наибольшее и наименьшее количество книг издательства (воспользоваться оператором UNION).

Select sh.ShopName , sum(sh.BookCount) as 'BookCount'

into #TMP\_tabl\_7\_3

From [Books]b

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

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

group by sh.ShopName

order by 2;

select tmp7.ShopName as 'Магазин', tmp7.BookCount as 'Максимум и минимум кол-ва книг'

from [#TMP\_tabl\_7\_3] tmp7

where tmp7.BookCount=(

select max(tmp7.BookCount) as 'Максимум и минимум кол-ва книг'

from [#TMP\_tabl\_7\_3] tmp7)

union

select tmp7.ShopName as 'Магазин',tmp7.BookCount as 'Максимум и минимум кол-ва книг'

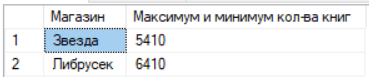
from [#TMP\_tabl\_7\_3] tmp7

where tmp7.BookCount=(

select min(tmp7.BookCount) as 'Максимум и минимум кол-ва книг'

from [#TMP\_tabl\_7\_3] tmp7)

order by 2;



8. Вывести имена всех авторов, книг которых не издавало еще издательство, то есть которые не присутствуют в таблице Books. (Написать два варианта запроса: один с использованием подзапросов, второй с использованием операторов объединения запросов).

1-й способ

Select a.AuthorSurName+' '+a.AuthorName+' '+a.AuthorSecName as 'Автор', b.IsPublish

from [Author]a

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

where a.Id not in (

select b.AuthorId from [Books]b

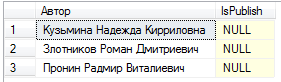
)

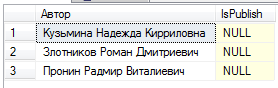
2й способ, но без UNION, т.к. без понятия, как это выполнить

Select a.AuthorSurName+' '+a.AuthorName+' '+a.AuthorSecName as 'Автор', b.IsPublish

from [Author]a left join [Books]b on a.Id=b.AuthorId

where b.Id is null





9. Создайте таблицу ShopAuthors, которая содержит следующие поля: имя и фамилия автора, название магазина и страна. Напишите команду, которая вставляла в нее всех авторов, книги которых реализуются более чем одним магазином издательства.

Не уверен, что это оптимальный сопособ решения. Выполнил через вспомогательную таблицу

Select a.AuthorName, a.AuthorSurName,count( sh.ShopName) as ShopCount, c.CountryName

into #TMP\_9

From [Author]a

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

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

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

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

Group by a.AuthorName, a.AuthorSurName, sh.ShopName, c.CountryName;

---эту таблицу пришлось создавать, т.к. в текущей БД ее не было

create table [ShopAuthors]

(

[Id] int primary key identity not null,

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

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

[ShopName] int not null ,

[CountryName] nvarchar(100) not null

);

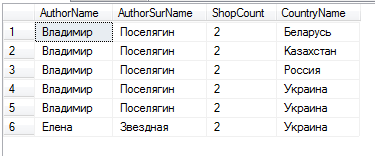
go

INSERT INTO ShopAuthors

select \*

from [#TMP\_9] tmp

where tmp.ShopCount>1;



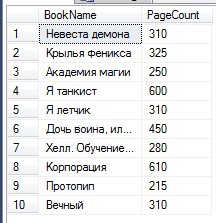
10.Удалить все книги, в которых количество страниц больше, чем среднее.

Сначала скопировали все нужные жанные во временную таблицу

select b.BookName, b.PageCount

into #tmp\_10

from [Books]b;



Удаляем из временной таблицы ненужные данные

delete

from #tmp\_10

where PageCount>(

select avg (tmp.PageCount)

from [#tmp\_10]tmp)

среднее кол-во стр в книгах, согласно

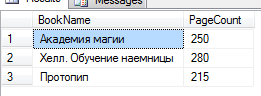
select avg (tmp.PageCount)

from [#tmp\_10]tmp

составляет 248 стр.

просматриваем временную таблицу

select \* from #tmp\_10



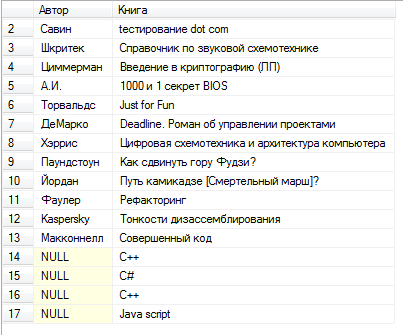
11.Удалить все книги, в которых не указан автор, при этом по результату исключить повторения.

Этот запрос выполнено в базе bookshops

смотрим на список всех книг

Select a.FirstName as 'Автор',b.NameBook as 'Книга'

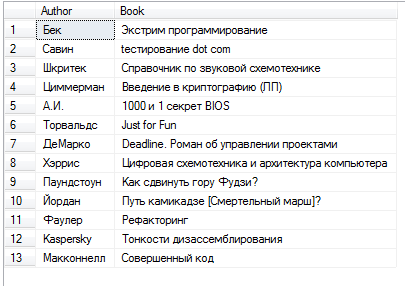
from [Books] b left join [Authors] a on a.ID\_AUTHOR= b.ID\_AUTHOR



delete

from Books

where ID\_AUTHOR is null;



12.В связи с закрытием магазинов в Англии, написать запрос, который уничтожает информацию обо всех магазинах, которые там размещались.

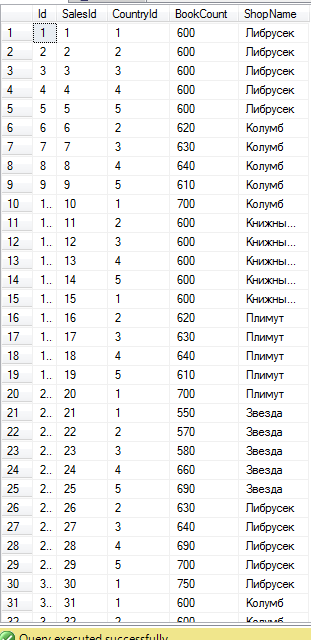
Удаляем не из основной БД , а из временной таблицы, куда предварительно скопировали данные из таблицы Shop

select \*

into #tmp\_12\_1

from [Shop]sh;

Содержимое таблицы Shop до удаления



Удаляем из временной таблицы все записи, связанные с магазинами , размещенными в Грузии (нет магазинов в текущей БД в Англии)

delete

from #tmp\_12\_1

where CountryId in(

Select c.Id

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

where c.CountryName= 'Грузия'

group by c.Id );

Согласно подзапросу

Select c.Id

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

where c.CountryName= 'Грузия'

group by c.Id

ID Грузии 5

Удаляем

delete

from #tmp\_12\_1

where CountryId in(

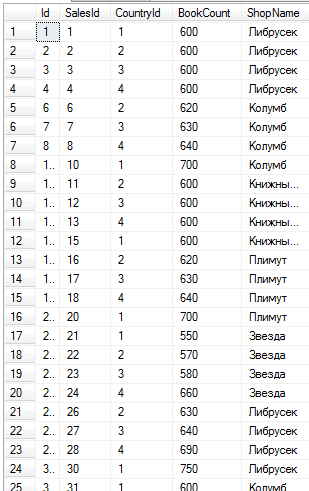
Select c.Id

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

where c.CountryName= 'Грузия'

group by c.Id );

После удаления



13.Изменить информацию о продаже книг, которые осуществлялись одним из магазинов в Украине, на пример, магазином "Либрусек" следующим образом:  
цены на книги, которые продавались 2018-09-08 года увеличить на 5%, а количество увеличить на 10 ед.

смотрим информацию до изменения

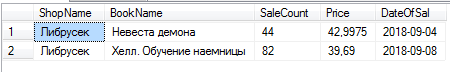
Select sh.ShopName, b.BookName, s.SaleCount, s.Price, s.DateOfSal

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

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

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

where c.CountryName = 'Украина' and sh.ShopName='Либрусек';



UPDATE sales

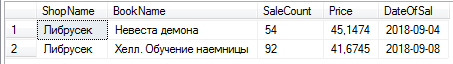
SET price = price \* 1.05, SaleCount=SaleCount+10

WHERE shopid IN

(SELECT s.ShopId FROM [Sales]s join [Shop]sh on sh.SalesId=s.Id

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

where c.CountryName = 'Украина' and sh.ShopName='Либрусек');



14.Уменьшить количество книг в магазинах, которые за последний год продали наименьшее количество книг, на 15%.

Показали 2 магазина , в которых было продано меньше всего книг, скопиров эти данные во временную таблицу

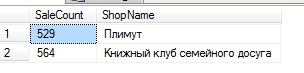
select top 2 sum(s.SaleCount) as SaleCount , sh.ShopName

into #tmp\_14

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

group by sh.ShopName

order by 1;



UPDATE Shop

SET BookCount=BookCount\*0.85

WHERE ShopName IN

(select tmp.ShopName

from [#tmp\_14\_1] tmp

)

Скрин не выполнял, т.к. ради этого над было бы создавать спец.запрос.