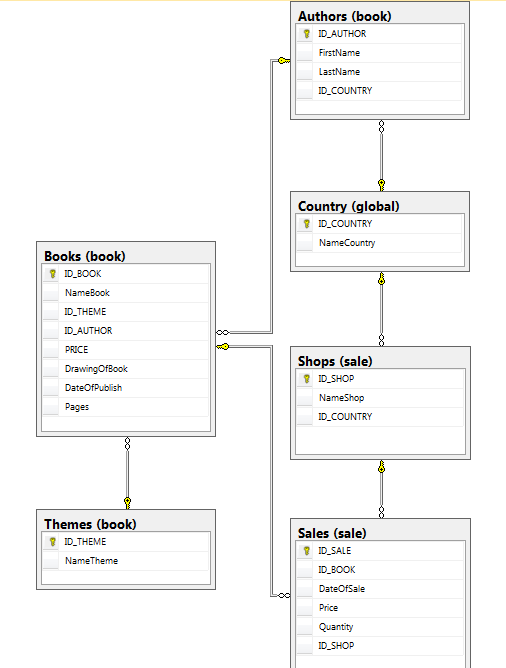
Содержание

[Структура БД 2](#_Toc534215725)

[Скрипт по созданию БД 2](#_Toc534215726)

[Решение ДЗ 6](#_Toc534215727)

# Структура БД



## Скрипт по созданию БД

CREATE DATABASE BookStore

uSE BookStore

GO

CREATE TABLE Country(

ID\_COUNTRY INT NOT NULL PRIMARY KEY IDENTITY,

NameCountry nvarchar(20) NOT NULL )

GO

CREATE SCHEMA global

GO

ALTER SCHEMA global Transfer Country

GO

SELECT \* FROM global.Country

GO

INSERT INTO global.Country

VALUES ('Украина'),

('Великобритания'),

('Бельгия'),

('Германия'),

('Голландия'),

('Испания'),

('Италия'),

('Латвия'),

('Польша'),

('США'),

('Финляндия'),

('Франция'),

('Швеция')

GO

CREATE TABLE Themes(

ID\_THEME INT NOT NULL PRIMARY KEY IDENTITY,

NameTheme nvarchar(30) NOT NULL)

GO

CREATE SCHEMA book

GO

ALTER SCHEMA book TRANSFER Themes

GO

GO

INSERT INTO book.Themes

VALUES ('Computer Science'),

('Science Fiction'),

('Web Technologies'),

('Programming'),

('Artiicial Intelligence'),

('Graphics and Design')

GO

CREATE TABLE Authors(

ID\_AUTHOR INT NOT NULL PRIMARY KEY IDENTITY,

FirstName nvarchar(20) NOT NULL,

LastName nvarchar(20) NOT NULL,

ID\_COUNTRY INT NOT NULL REFERENCES global.Country (ID\_COUNTRY)

ON DELETE NO ACTION ON UPDATE CASCADE )

GO

ALTER SCHEMA book TRANSFER Authors

GO

INSERT INTO [book].[Authors]

VALUES ('Nick', 'Bostrom', 2),

('Brian', 'Christian', 9),

('Ray', 'Kurzweil', 5),

('Nicolas', 'Carr', 4),

('Max', 'Tegmark', 7),

('Thomas', 'Cormen', 5),

('Joe', 'Baron', 3),

('David', 'Chandler', 6),

('Donella', 'Meadows', 8),

('Gareth', 'James', 7),

('Donald', 'Knuth', 2),

('Bjarne', 'Stroustrup', 13),

('Eric', 'Matthes', 10),

('Charlez', 'Petzold', 11),

('Robert', 'Nystrom', 12),

('Wallace', 'Wang', 12),

('Joaeph', 'Connor', 12),

('Jordan', 'Hudgens', 10),

('Aditya', 'Shankar', 12),

('Paul', 'Laurence', 12),

('Michael', 'Dawson', 2),

('Brian', 'Kerninghan', 2),

('Bill', 'Phillips', 4)

GO

CREATE TABLE Shops(

ID\_SHOP INT NOT NULL PRIMARY KEY IDENTITY,

NameShop nvarchar(20) NOT NULL,

ID\_COUNTRY INT NOT NULL REFERENCES global.Country (ID\_COUNTRY)

ON DELETE NO ACTION ON UPDATE CASCADE)

GO

CREATE SCHEMA sale

GO

ALTER SCHEMA sale TRANSFER Shops

GO

GO

INSERT INTO sale.Shops

VALUES ('Букинист', 1),

('Amazon', 13),

('Буква', 1),

('Waterstones', 2),

('Alfa Antikvariat', 13),

('Sterling Books', 3)

GO

GO

CREATE TABLE Books(

ID\_BOOK INT NOT NULL PRIMARY KEY IDENTITY,

NameBook nvarchar(30) NOT NULL,

ID\_THEME int NOT NULL REFERENCES book.Themes (ID\_THEME)

ON DELETE NO ACTION ON UPDATE CASCADE,

ID\_AUTHOR INT NOT NULL REFERENCES book.Authors(ID\_AUTHOR) ON DELETE NO ACTION ON UPDATE CASCADE,

PRICE money NOT NULL,

DrawingOfBook Binary,

DateOfPublish DATE NOT NULL DEFAULT GEtDATE(),

Pages INT NOT NULL)

GO

GO

ALTER SCHEMA book TRANSFER Books

GO

GO

INSERT INTO book.Books

(NameBook, ID\_THEME, ID\_AUTHOR, PRICE, DateOfPublish, Pages)

VALUES('Superintelligence 2', 5, 1, 170.0, '2017-06-29', 432),

('Algorithms to Live By', 4, 2, 130.0, '2017-02-12', 656),

('Introduction to Algorithms', 1, 9, 140.0, '2015-05-17', 632),

('AWS Certified Solutions', 1, 7, 90.0, '2017-10-01', 820),

('The Black Swan', 1, 8, 60.0, '2016-11-28', 408),

('Pro HTML5 Games', 3, 19, 70.0, '2016-04-22', 712),

('Beginning C++ Through Game', 4, 21, 99.9, '2014-05-13', 782),

('The Art of Programming', 4, 11, 90.0, '2015-01-25', 964),

('Python Crash Course', 4, 13, 70.0, '2017-08-30', 856),

('Code: The Hidden Language', 4, 14, 79.9, '2014-10-25', 795),

('Game Programming Patterns', 4, 15, 90.0, '2017-11-24', 800),

('Beginning Programming', 4, 16, 60.0, '2016-12-08', 540),

('Basics of Java, SQL & C++', 4, 17, 90.0, '2017-09-18', 590),

('Statistical Learning', 4, 10, 100.0, '2017-11-24', 790)

GO

CREATE TABLE Sales(

ID\_SALE INT NOT NULL PRIMARY KEY IDENTITY,

ID\_BOOK INT NOT NULL REFERENCES book.Books(ID\_BOOK),

DateOfSale DATE DEFAULT GetDate(),

Price Money NOT NULL,

Quantity int NOT NULL DEFAULT 1,

ID\_SHOP int NOT NULL REFERENCES sale.Shops(ID\_SHOP) ON DELETE NO ACTION ON UPDATE CASCADE)

GO

ALTER SCHEMA sale TRANSFER Sales

GO

SELECT \* FROM book.Authors

GO

SELECT \* FROM book.Themes

Go

SELECT \* FROM book.Books

GO

INSERT INTO sale.Sales

VALUES (1, DEFAULT, 160.0, 2, 1),

(2, DEFAULT, 130, 1, 1),

(3, DEFAULT, 175, 1, 2),

(4, DEFAULT, 135, 2, 2),

(5, DEFAULT, 147, 3, 2),

(6, DEFAULT, 110, 6, 2),

(6, DEFAULT, 105, 5, 3),

(6, DEFAULT, 107, 3, 1),

(7, DEFAULT, 70, 2, 1),

(7, DEFAULT, 75, 4, 1),

(8, DEFAULT, 68, 1, 1),

(8, DEFAULT, 76, 2, 3),

(9, DEFAULT, 110, 3, 3),

(10, DEFAULT, 115, 2, 1),

(10, DEFAULT, 120, 1, 3),

(11, DEFAULT, 80, 4, 2),

(12, DEFAULT, 88, 3, 4),

(13, DEFAULT, 110, 4, 2),

(14, DEFAULT, 68, 7, 1),

(14, DEFAULT, 77, 4, 2),

(14, DEFAULT, 73, 3, 4),

(13, DEFAULT, 93, 1, 3),

(13, DEFAULT, 115, 3, 1),

(12, '04-12-2017', 80, 3, 2),

(14, '11-20-2017', 69, 1, 1),

(13, '08-24-2017', 75, 4, 2),

(8, '04-12-2017', 112, 2, 3),

(10, '04-12-2016', 115, 5, 2)

# Решение ДЗ

1. Функцию, которая возвращает среднее арифметическое цен всех книг, проданных до указанной даты.

create function AvPrBookBeforDate(@day datetime)

returns money

as

begin

declare @avPrice money

set @avPrice =(

Select AVG(b.Price)

From sale.sales S join book.Books B on S.ID\_BOOK = B.ID\_BOOK

where s.DateOfSale<@day)

return @avPrice

end

go

select dbo.AvPrBookBeforDate('2019-01-02') as 'AVG Price'

go



2. Функцию, которая возвращает самую дорогую книгу издательства указанной тематики ( имеем две книги с максимальной ценой, сделано специально)

create function mostExpBookbyThem (@theme nvarchar(50))

returns table

as return(

Select b.NameBook, b.PRICE

from book.Books as b

where b.PRICE=(

select max(b.PRICE) as 'Price'

from book.Books as b join sale.Sales as s

on s.ID\_BOOK = b.ID\_BOOK

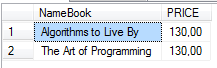
join book.Themes as t on b.ID\_THEME=T.ID\_THEME

where t.NameTheme=@theme));

go

select \* from mostExpBookbyThem('Programming');

go



3. Функцию, которая по ID магазина возвращает информацию о нем (ID, название, местоположение, средняя стоимость продаж за последний год книг вашего издательства) в табличном виде.

create function shopInfo (@Id int )

returns table

as return(

select sh.ID\_SHOP as'ID',sh.NameShop as 'Shop name' , avg(s.Price) as 'Average price',

c.NameCountry as 'Location'

from book.Books b join sale.Sales S on b.ID\_BOOK=s.ID\_BOOK

join sale.Shops sh on s.ID\_SHOP=sh.ID\_SHOP

join global.Country c on sh.ID\_COUNTRY=c.ID\_COUNTRY

where sh.ID\_SHOP=@Id

group by sh.ID\_SHOP,sh.NameShop, c.NameCountry

);

go

select \* from shopInfo(1);

go



4. Функцию, которая возвращает количество магазинов, которые не продали ни одной книги издательства.

create function nullShop()

returns int

as

begin

declare @count int

declare @tmpTable table (NameShop nvarchar(30), SalesCount int )

insert @tmpTable

select sh.NameShop as 'Shop name' , count (s.DateOfSale) as 'Salese count'

from book.Books b join sale.Sales S on b.ID\_BOOK=s.ID\_BOOK

join sale.Shops sh on s.ID\_SHOP=sh.ID\_SHOP

group by sh.NameShop

set @count =(

Select tt.SalesCount

From @tmpTable tt

where tt.SalesCount=0)

if @count=NULL-- тут бы должна была сработать закладка на присвоение 0, если @count=NULL, но не срабатывает от чего-то

set @count=0;

--print @count

return @count

end

go

select dbo.nullShop() as 'Count of NULLShops'

go



5. Функцию, которая возвращает минимальный из трех параметров.

create function Minimum(@a int, @b int, @c int )

returns int

as

begin

declare @min int

if @a<=@b and @a<=@c

set @min= @a;

else

if @b<=@a and @b<=@c

set @min= @b;

else

--if @c>=@a and @c>=@b

set @min= @c;

return @min;

end

go

select dbo.Minimum(1,2,3) as 'Minimum'

go



6. Многооператорную функцию, которая возвращает количество проданных книг по каждой из тематик и в разрезе каждого магазина.

create function BookCountJanreInShop()

returns @tmpTable table (ShopName varchar(30) not null, BookCount int not null, NameTheme varchar(30) not null)

as

begin

insert @tmpTable

select sh.NameShop as 'Shop name' , count(s.DateOfSale) as 'Count of sales',

t.NameTheme as 'Janre'

from book.Books b join sale.Sales S on b.ID\_BOOK=s.ID\_BOOK

join sale.Shops sh on s.ID\_SHOP=sh.ID\_SHOP

join book.Themes t on t.ID\_THEME=b.ID\_THEME

group by t.NameTheme, sh.NameShop

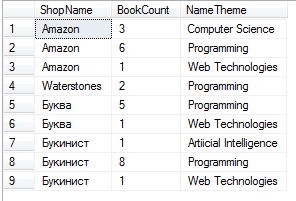
return

end

go

select \* from BookCountJanreInShop();

go



7. Функцию, которая возвращает список книг, которые соответствуют набору критериев (имя и фамилия автора, тематика), и отсортированы по фамилии автора в указанном в 4-м параметре направлении.

create function BooksBy (@Name nvarchar(30), @SurName nvarchar(30),

@janreName nvarchar(100) , @sort int)

returns @tmpTable table (FirstName varchar(30) not null,LastName varchar(30) not null,

NameBook varchar(50) not null, ThemeName varchar(50) not null)

as

begin

if @sort=0

begin

insert @tmpTable

select \* from ViewFor

where FirstName=@Name and LastName=@SurName and ThemeName=@janreName

order by FirstName

end

else

begin

insert @tmpTable

select \* from ViewFor

where FirstName=@Name and LastName=@SurName and ThemeName=@janreName

order by FirstName desc

end

return

end

go

select \* from BooksBy('Wallace','Wang','Programming',0);

go

