В тексте заданий используется диалект Microsoft SQL Server 2014, однако для решения можно

пользоваться другими версиями Microsoft SQL Server, а также другими RDBMS (PostgreSQL либо SQLite).

Даны таблицы

---------------------------------------------------------------------------------------------------

if object\_id('stack.OrderItems') is not null

drop table stack.OrderItems;

go

if object\_id('stack.Orders') is not null

drop table stack.Orders;

go

if object\_id('stack.Customers') is not null

drop table stack.Customers;

go

-- Заказчики

create table stack.Customers

(

row\_id int identity not null,

name nvarchar(max) not null, -- наименование заказчика

constraint PK\_Customers

primary key nonclustered(row\_id)

);

go

-- Заказы

create table stack.Orders

(

row\_id int identity not null,

parent\_id int, -- row\_id родительской группы

group\_name nvarchar(max), -- наименование группы заказов

customer\_id int, -- row\_id заказчика

registered\_at date -- дата регистрации заказа

constraint PK\_Orders

primary key nonclustered (row\_id),

constraint FK\_Orders\_Folder

foreign key (parent\_id)

references stack.Orders(row\_id)

on delete no action

on update no action,

constraint FK\_Customers

foreign key (customer\_id)

references stack.Customers(row\_id)

on delete cascade

on update cascade

);

go

-- Позиции заказов

create table stack.OrderItems

(

row\_id int identity not null,

order\_id int not null, -- row\_id заказа

name nvarchar(max) not null, -- наименование позиции

price int not null, -- стоимость позиции в рублях

constraint PK\_OrderItems

primary key nonclustered (row\_id),

constraint FK\_OrderItems\_Orders

foreign key (order\_id)

references stack.Orders(row\_id)

on delete cascade

on update cascade

);

go

---------------------------------------------------------------------------------------------------

Таблица Customers содержит информацию о покупателях интернет-магазина, таблица Orders - сведения о

заказах, а таблица OrderItems - позиции в заказах. Данные в таблице Orders имеют древовидную структуру.

Записи, у которых поле group\_name не равно null, предназначены для группировки заказов по категориям.

Такие записи не связаны с заказчиком, не имеют даты регистрации и связанных позиций в таблице OrderItems,

но могут содержать в себе заказы или другие группы. Если же поле group\_name равно null, то запись является

заказом. Она обязательно связана с покупателем, имеет позиции и дату регистрации, но не может иметь

вложенных заказов. Записи связываются с родительской группой с помощью поля parent\_id, которое хранит

row\_id этой группы. Записи с parent\_id равным null считаются корневыми. Заказ может иметь несколько позиций

с одинаковым наименованием.

В таблицах имеются следующие данные:

---------------------------------------------------------------------------------------------------

insert into stack.Customers -- 1

values(N'Иванов');

insert into stack.Customers -- 2

values(N'Петров');

insert into stack.Customers -- 3

values(N'Сидоров');

insert into stack.Customers -- 4

values(N'ИП Федоров');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 1

values (null, N'Все заказы', null, null);

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 2

values (1, N'Частные лица', null, null);

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 3

values (2, N'Оргтехника', null, null);

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 4

values (3, null, 1, '2019/10/02');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 5

values (3, null, 1, '2020/05/17');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 6

values (3, null, 1, '2020/04/28');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 7

values (3, null, 2, '2019/08/05');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 8

values (3, null, 2, '2020/05/17');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 9

values (3, null, 2, '2020/02/11');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 10

values (2, N'Канцелярия', null, null);

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 11

values (10, null, 3, '2020/04/09');

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 12

values (1, N'Юридические лица', null, null);

insert into stack.Orders(parent\_id, group\_name, customer\_id, registered\_at) -- 13

values (12, null, 4, '2020/06/25');

insert into stack.OrderItems(order\_id, name, price)

values (4, N'Принтер', 30);

insert into stack.OrderItems(order\_id, name, price)

values (4, N'Факс', 20);

insert into stack.OrderItems(order\_id, name, price)

values (5, N'Принтер', 50);

insert into stack.OrderItems(order\_id, name, price)

values (5, N'Кассовый аппарат', 40);

insert into stack.OrderItems(order\_id, name, price)

values (5, N'Факс', 30);

insert into stack.OrderItems(order\_id, name, price)

values (6, N'Кассовый аппарат', 30);

insert into stack.OrderItems(order\_id, name, price)

values (6, N'Кассовый аппарат', 40);

insert into stack.OrderItems(order\_id, name, price)

values (7, N'Копировальный аппарат', 50);

insert into stack.OrderItems(order\_id, name, price)

values (7, N'Калькулятор', 10);

insert into stack.OrderItems(order\_id, name, price)

values (7, N'Кассовый аппарат', 60);

insert into stack.OrderItems(order\_id, name, price)

values (8, N'Принтер', 50);

insert into stack.OrderItems(order\_id, name, price)

values (8, N'Калькулятор', 10);

insert into stack.OrderItems(order\_id, name, price)

values (9, N'Телефонный аппарат', 50);

insert into stack.OrderItems(order\_id, name, price)

values (9, N'Кассовый аппарат', 40);

insert into stack.OrderItems(order\_id, name, price)

values (11, N'Бумага', 2);

insert into stack.OrderItems(order\_id, name, price)

values (11, N'Ручки', 1);

insert into stack.OrderItems(order\_id, name, price)

values (13, N'Кулер', 100);

insert into stack.OrderItems(order\_id, name, price)

values (13, N'Стулья', 70);

insert into stack.OrderItems(order\_id, name, price)

values (13, N'Факс', 20);

go

---------------------------------------------------------------------------------------------------

Корневой группой является запись "Все заказы", в нее вложены группы "Частные лица" и "Юридические лица".

Примером заказа является запись "ИП Федоров", которая находится в группе "Юридические лица".

== Задание 1.

Написать функцию select\_orders\_by\_item\_name. Она получает один аргумент - наименование позиции (строка),

и должна найти все заказы, в которых имеется позиция с данным наименованием. Кроме того, она должна

подсчитать количество позиций с указанным наименованием в каждом отдельном заказе. Результатом вызова

функции должна быть таблица с тремя колонками:

- order\_id (row\_id заказа)

- customer (наименование заказчика)

- items\_count (количество позиций с данным наименованием в этом заказе)

Примеры вызова функции:

select \* from stack.select\_orders\_by\_item\_name(N'Факс')

-- 4 Иванов 1

-- 5 Иванов 1

-- 13 ИП Федоров 1

select \* from stack.select\_orders\_by\_item\_name(N'Кассовый аппарат')

-- 5 Иванов 1

-- 6 Иванов 2

-- 7 Петров 1

-- 9 Петров 1

select \* from stack.select\_orders\_by\_item\_name(N'Стулья')

-- 13 ИП Федоров 1

== Задание 2.

Написать функцию calculate\_total\_price\_for\_orders\_group. Она получает row\_id группы (либо заказа),

и возвращает суммарную стоимость всех позиций всех заказов в этой группе (заказе), причем

суммирование должно выполняться по всему поддереву заказов, начинающемуся с данной группы.

Функция должна возвращать число.

Примеры вызова функции:

select stack.calculate\_total\_price\_for\_orders\_group(1) as total\_price -- 703, все заказы

select stack.calculate\_total\_price\_for\_orders\_group(2) as total\_price -- 513, группа 'Частные лица'

select stack.calculate\_total\_price\_for\_orders\_group(3) as total\_price -- 510, группа 'Оргтехника'

select stack.calculate\_total\_price\_for\_orders\_group(12) as total\_price -- 190, группа 'Юридические лица'

select stack.calculate\_total\_price\_for\_orders\_group(13) as total\_price -- 190, заказ 'ИП Федоров'

== Задание 3.

Написать запрос, возвращающий наименования всех покупателей, у которых каждый заказ в 2020 году содержит

как минимум одну позициию с наименованием "Кассовый аппарат".

Результатом выполнения запроса на тестовых данных будет таблица с одной строкой "Иванов".