**create** **table** public.tst\_contractor --Поставщик

( id **bigint** **not** **null**,

sName **varchar**(30) --Название

);

**alter** **table** public.tst\_contractor **add** **constraint** pk\_tst\_ctr **primary** **key** (id);

**create** **table** public.tst\_goods --Товар

( id **bigint** **not** **null**,

sName **varchar**(30) --Название

);

**alter** **table** public.tst\_goods **add** **constraint** pk\_tst\_goods **primary** **key** (id);

**create** **table** public.tst\_invoice --Накладная

( id **bigint** **not** **null**,

dDate **timestamp** **without** **time** **zone**, --Дата накладной

idCtr **bigint** --Поставщик

);

**alter** **table** public.tst\_invoice **add** **constraint** pk\_tst\_inv **primary** **key** (id);

**alter** **table** public.tst\_invoice **add** **constraint** fk\_tst\_ctr **foreign** **key** (idctr) **references** public.tst\_contractor (id);

**create** **table** public.tst\_invoice\_str --Строки накладной

( id **bigint** **not** **null**,

idInv **bigint**, --Накладная

idGds **bigint**, --Товар

fPrice **numeric**(6,2) **default** 0 **not** **null**, --Цена

nOrd **integer** **not** **null** -- Порядковый номер товарной позиции

);

**alter** **table** public.tst\_invoice\_str **add** **constraint** pk\_tst\_inv\_str **primary** **key** (id);

**alter** **table** public.tst\_invoice\_str **add** **constraint** fk\_tst\_Inv\_str **foreign** **key** (idInv) **references** public.tst\_invoice (id);

**alter** **table** public.tst\_invoice\_str **add** **constraint** fk\_tst\_gds\_str **foreign** **key** (idgds) **references** public.tst\_goods (id);

**insert** **into** public.tst\_contractor (id, sname)

**values** (1, 'Петя'),

(2, 'Вася'),

(3, 'Саня'),

(4, 'Федя'),

(5, 'Коля'),

(6, 'Вова');

**insert** **into** public.tst\_goods (id, sname)

**values** (1, 'Йод'),

(2, 'Зеленка'),

(3, 'Аспирин'),

(4, 'Бинт'),

(5, 'Пластырь');

**insert** **into** public.tst\_invoice (id, dDate, idCtr)

**values** (1, to\_date('01-03-2018', 'dd-mm-yyyy'), 1),

(2, to\_date('01-03-2018', 'dd-mm-yyyy'), 2),

(3, to\_date('01-03-2018', 'dd-mm-yyyy'), 3),

(4, to\_date('02-03-2018', 'dd-mm-yyyy'), 4),

(5, to\_date('03-03-2018', 'dd-mm-yyyy'), 5),

(6, to\_date('06-03-2018', 'dd-mm-yyyy'), 5),

(7, to\_date('06-03-2018', 'dd-mm-yyyy'), 6),

(8, to\_date('12-03-2018', 'dd-mm-yyyy'), 1),

(9, to\_date('17-03-2018', 'dd-mm-yyyy'), 2),

(10, to\_date('29-03-2018', 'dd-mm-yyyy'), 6);

**insert** **into** public.tst\_invoice\_str (id, idInv, idGds, fPrice,nOrd )

**values**

(1, 1, 1, 100, 1),

(2, 1, 2, 120, 2),

(3, 1, 3, 130, 3),

(4, 2, 1, 111, 1),

(5, 2, 2, 122, 2),

(6, 2, 3, 133, 3),

(7, 3, 1, 101, 1),

(8, 3, 3, 102, 2),

(9, 3, 2, 103, 3),

(10, 4, 1, 104, 1),

(11, 5, 5, 105, 1),

(12, 6, 4, 106, 1),

(13, 7, 5, 104, 1),

(14, 8, 1, 100, 1),

(15, 8, 2, 100, 2),

(16, 8, 3, 100, 3),

(17, 9, 1, 100, 1),

(18, 9, 2, 100, 2),

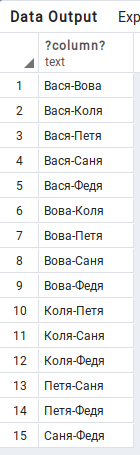
(19, 9, 4, 100, 3),

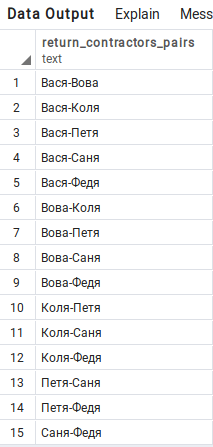
(20, 10, 4, 100, 1) ;

Задачи и ИХ РЕШЕНИЯ

1. Получить все пары поставщиков (из tst\_contractor) в виде 2-х колонок, например, (Петя- Вася). Учесть, что пары (Петя - Вася) и (Вася-Петя) одинаковые и показывать надо только одну из них.

|  |  |
| --- | --- |
| Реализация в виде хранимой функции | Реализация на чисто SQL |
| CREATE OR REPLACE FUNCTION cs\_pairs() RETURNS setof text AS $$  DECLARE  cs TEXT [];  i int;j int;iL int;  BEGIN  cs := ARRAY( SELECT sname FROM tst\_contractor order by sname );  i:=1;iL=array\_upper(cs, 1);  loop  if i > iL then  exit;  else  j := i+1;  loop  if j < iL+1 then  RETURN next cs[i] || '-' || cs[j];  else  exit;  end if ;  j:=j+1;  end loop;  i:=i+1;  end if;  end loop;  END  $$ LANGUAGE plpgsql;  select \* from cs\_pairs() | **SELECT**  **C.sname||' - '||CC.sname**  **as Пары\_поставщиков**  **FROM tst\_contractor AS C**  **Join tst\_contractor AS CC**  **ON CC.id > C.id**  **order by C.sname** |





2. Получить все пары поставщиков (из tst\_contractor), поставляющие (tst\_invoice,tst\_invoice\_str) одинаковое (непустое) множество товаров (tst\_goods).

**SELECT distinct 'Комплект поставщика '||c1.sname ||' = Комплекту поставщика '||c2.sname as ПоставщикиОбщегоКомпектаТоваров**

**FROM tst\_contractor c1**

**join tst\_contractor c2 on c1.id > c2.id**

**join tst\_invoice i1 on i1.idctr = c1.id**

**join tst\_invoice i2 on i2.idctr = c2.id**

**join tst\_invoice\_str is1 on i1.id = is1.idinv**

**join tst\_invoice\_str is2 on i2.id = is2.idinv**

**and not exists(**

**(**

**SELECT distinct is11.idgds FROM tst\_invoice\_str is11,tst\_invoice i1 WHERE i1.idctr=c1.id and i1.id = is11.idinv**

**EXCEPT**

**SELECT distinct is12.idgds FROM tst\_invoice\_str is12,tst\_invoice i2 WHERE i2.idctr=c2.id and i2.id = is12.idinv**

**)**

**union**

**(**

**SELECT distinct is11.idgds FROM tst\_invoice\_str is11,tst\_invoice i1 WHERE i1.idctr=c2.id and i1.id = is11.idinv**

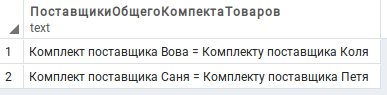
**EXCEPT**

**SELECT distinct is12.idgds FROM tst\_invoice\_str is12,tst\_invoice i2 WHERE i2.idctr=c1.id and i2.id = is12.idinv**

**)**

**)**

**order by ПоставщикиОбщегоКомпектаТоваров;**



3 Для каждого дня марта 2018 получить отчет в виде:

(Дата) (сумма на начало дня) (сумма за день) (сумма на конец дня)

**SELECT**

**generated\_day Дата,**

**coalesce(**

**(select sum(is11.fprice) from tst\_invoice\_str is11**

**join tst\_invoice i11 on is11.idinv=i11.id and i11.ddate<generated\_day**

**), 0.00**

**) СуммаНаНачалоДня,**

**coalesce(**

**(select sum(is11.fprice) from tst\_invoice\_str is11**

**join tst\_invoice i11 on is11.idinv=i11.id and i11.ddate=generated\_day**

**group by i11.ddate**

**), 0.00**

**) СуммаЗаДень,**

**coalesce(**

**(select sum(is11.fprice) from tst\_invoice\_str is11**

**join tst\_invoice i11 on is11.idinv=i11.id and i11.ddate<=generated\_day**

**), 0.00**

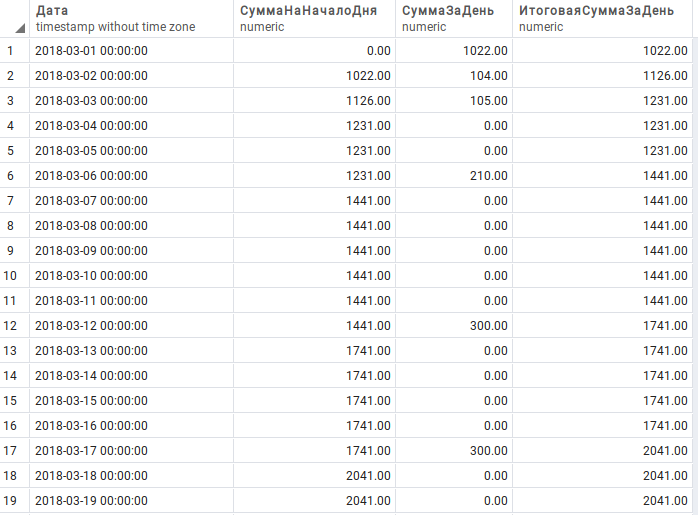
**) СуммаЗаДень**

**FROM Generate\_series(timestamp '2018-03-01', timestamp '2018-03-31', interval '1 day') generated\_day**

**left join tst\_invoice i1 on generated\_day=i1.DDate**

**Group BY generated\_day**

**order by generated\_day**



4. Для всех товаров (tst\_goods) получить отчет о цене поставки в разрезе поставщиков. Из всех поставщиков товара выбираются не более трех, поставляющих товар по минимальной цене. Отчет получить в виде:

(Товар) (Поставщик с минимальной ценой) (Минимальная цена) (Поставщик с ценой чуть больше) (Цена чуть больше) (Поставщик с ценой еще чуть больше) (Цена еще чуть больше)

Если товар поставляют менее 3-х поставщиков, заполнить соответствующие места **null**.

**select**

**g1.sname Товар,**

**(select 'min цена\_1 у '||c2.sname||'='||is2.fprice from tst\_contractor c2**

**join tst\_invoice as i2 on i2.idctr=c2.id**

**join tst\_invoice\_str as is2 on is2.idinv=i2.id**

**join tst\_goods as g2 on g2.id=is2.idgds and g1.id=g2.id**

**order by is2.fprice**

**LIMIT 1**

**Offset 1**

**) as Поставщик\_с\_min\_ценой\_1,**

**(select 'min цена\_2 у '||c2.sname||'='||is2.fprice from tst\_contractor c2**

**join tst\_invoice as i2 on i2.idctr=c2.id**

**join tst\_invoice\_str as is2 on is2.idinv=i2.id**

**join tst\_goods as g2 on g2.id=is2.idgds and g1.id=g2.id**

**order by is2.fprice**

**LIMIT 1**

**Offset 2**

**) as Поставщик\_с\_min\_ценой\_2,**

**(select 'min цена\_3 у '||c2.sname||'='||is2.fprice from tst\_contractor c2**

**join tst\_invoice as i2 on i2.idctr=c2.id**

**join tst\_invoice\_str as is2 on is2.idinv=i2.id**

**join tst\_goods as g2 on g2.id=is2.idgds and g1.id=g2.id**

**order by is2.fprice**

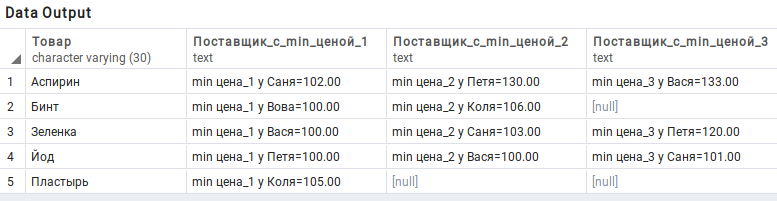
**LIMIT 1**

**Offset 3**

**) as Поставщик\_с\_min\_ценой\_3**

**from tst\_goods as g1**

**order by g1.sname**



5. Написать функцию, удаляющие все поставки (накладные) поставщика (id поставщика –параметр функции) кроме первой поставки (ее оставить). Порядок поставок определить по дате и id (для одинаковой даты поставки с меньшим id шли раньше).

CREATE OR REPLACE FUNCTION add\_last\_invoices(\_id integer) RETURNS text AS $$

DECLARE

BEGIN

if \_id=1 then

insert into tst\_invoice (id, dDate, idCtr) values (1, to\_date('01-03-2018', 'dd-mm-yyyy'), 1) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice (id, dDate, idCtr) values (8, to\_date('12-03-2018', 'dd-mm-yyyy'), 1) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (4, 2, 1, 111, 1) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (5, 2, 2, 122, 2) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (6, 2, 3, 133, 3) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (14, 8, 1, 100, 1) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (15, 8, 2, 100, 2) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (16, 8, 3, 100, 3) on CONFLICT (id) DO NOTHING;

end if;

if \_id=2 then

insert into tst\_invoice (id, dDate, idCtr) values (2, to\_date('01-03-2018', 'dd-mm-yyyy'), 2) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice (id, dDate, idCtr) values (9, to\_date('17-03-2018', 'dd-mm-yyyy'), 2) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (4, 2, 1, 111, 1) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (5, 2, 2, 122, 2) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (6, 2, 3, 133, 3) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (17, 9, 1, 100, 1) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (18, 9, 2, 100, 2) on CONFLICT (id) DO NOTHING;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (19, 9, 4, 100, 3) on CONFLICT (id) DO NOTHING;

end if;

if \_id=3 then

insert into tst\_invoice (id, dDate, idCtr) values (3, to\_date('01-03-2018', 'dd-mm-yyyy'), 3) on CONFLICT (id) do nothing;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (7, 3, 1, 101, 1),(8, 3, 3, 102, 2) on CONFLICT (id) do nothing;

insert into tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (9, 3, 2, 103, 3) on CONFLICT (id) do nothing;

end if;

if \_id=4 then

insert into public.tst\_invoice (id, dDate, idCtr) values (4, to\_date('02-03-2018', 'dd-mm-yyyy'), 4) on CONFLICT (id) do nothing;

insert into public.tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (10, 4, 1, 104, 1) on CONFLICT (id) do nothing;

end if;

if \_id=5 then

insert into public.tst\_invoice (id, dDate, idCtr) values (5, to\_date('03-03-2018', 'dd-mm-yyyy'), 5) on CONFLICT (id) do nothing;

insert into public.tst\_invoice (id, dDate, idCtr) values (6, to\_date('06-03-2018', 'dd-mm-yyyy'), 5) on CONFLICT (id) do nothing;

insert into public.tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (11, 5, 5, 105, 1) on CONFLICT (id) do nothing;

insert into public.tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (12, 6, 4, 106, 1) on CONFLICT (id) do nothing;

end if;

if \_id=6 then

insert into public.tst\_invoice (id, dDate, idCtr) values (7, to\_date('06-03-2018', 'dd-mm-yyyy'), 6) on CONFLICT (id) do nothing;

insert into public.tst\_invoice (id, dDate, idCtr) values (10, to\_date('29-03-2018', 'dd-mm-yyyy'), 6) on CONFLICT (id) do nothing;

insert into public.tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (13, 7, 5, 104, 1) on CONFLICT (id) do nothing;

insert into public.tst\_invoice\_str (id, idInv, idgds, fPrice,nOrd ) values (20, 10, 4, 100, 1) on CONFLICT (id) do nothing;

end if;

RETURN 'Добавили';

END;

$$

LANGUAGE plpgsql;

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

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

CREATE OR REPLACE FUNCTION delete\_last\_invoices(\_id integer) RETURNS text AS $$

DECLARE

BEGIN

DELETE

FROM tst\_invoice\_str is1

USING tst\_invoice i1

WHERE is1.idinv = i1.id

and

i1.id

in (

(select id FROM tst\_invoice WHERE idctr=\_id)

except

(SELECT id FROM tst\_invoice where idctr=\_id order by id limit 1)

);

DELETE

FROM tst\_invoice

WHERE

id

in (

(select id FROM tst\_invoice WHERE idctr=\_id)

except

(SELECT id FROM tst\_invoice where idctr=\_id order by id limit 1)

);

RETURN 'Удалил';

END;

$$

LANGUAGE plpgsql;

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

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

select from add\_last\_invoices(1);

select from add\_last\_invoices(2);

select from add\_last\_invoices(3);

select from add\_last\_invoices(4);

select from add\_last\_invoices(5);

select from add\_last\_invoices(6);

--select from delete\_last\_invoices(1);

select from delete\_last\_invoices(2);

select from delete\_last\_invoices(3);

select from delete\_last\_invoices(4);

select from delete\_last\_invoices(5);

select from delete\_last\_invoices(6);

select \* from tst\_invoice\_str order by id,idinv,nord;

--select \* from tst\_invoice order by id;

6. Написать **update**, который меняет цену (fPrice) в накладных (tst\_invoice, tst\_invoice\_str) по правилу: если в названии (sName) товара (tst\_goods) 4 буквы, то текущую цену удвоить, в остальных случаях цену утроить.

UPDATE tst\_invoice\_str is1

SET fprice = (

CASE (SELECT is1.id)

WHEN 1 THEN 100

WHEN 2 THEN 120

WHEN 3 THEN 130

WHEN 4 THEN 111

WHEN 5 THEN 122

WHEN 6 THEN 133

WHEN 7 THEN 101

WHEN 8 THEN 102

WHEN 9 THEN 103

WHEN 10 THEN 104

WHEN 11 THEN 105

WHEN 12 THEN 106

WHEN 13 THEN 104

WHEN 14 THEN 100

WHEN 15 THEN 100

WHEN 16 THEN 100

WHEN 17 THEN 100

WHEN 18 THEN 100

WHEN 19 THEN 100

WHEN 20 THEN 100

END)

FROM tst\_goods g1

WHERE is1.idgds = g1.id;

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

UPDATE tst\_invoice\_str is1

SET fprice = (

CASE (SELECT length(g1.sname) AS "Length of a String")

WHEN 4 THEN 2

ELSE 4

END)\*fprice

FROM

tst\_goods g1

WHERE

is1.idgds = g1.id;

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

Select fprice,g1.sname from tst\_invoice\_str is1

join tst\_goods g1 on is1.idgds = g1.id;

7. Необходимо найти пары накладных-дублей. Накладные являются дублями, если количество позиций (строк) у них совпадает и во всех позициях с одинаковым порядковым номером у них одинаковый товар.

SELECT distinct 'н/к '||i1.id||' - н/к '||i2.id as НакладныеДубликаты

FROM tst\_invoice i1

join tst\_invoice i2 on i2.id>i1.id

and not exists(

(

SELECT ' стр.'||is1.nord||'-т.'||is1.idgds ahead FROM tst\_invoice\_str is1 WHERE is1.idinv=i1.id

EXCEPT

SELECT ' стр.'||is2.nord||'-т.'||is2.idgds ahead1 FROM tst\_invoice\_str is2 WHERE is2.idinv=i2.id

)

UNION

(

SELECT ' стр.'||is2.nord||'-т.'||is2.idgds ahead FROM tst\_invoice\_str is2 WHERE is2.idinv=i2.id

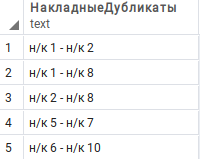
EXCEPT

SELECT ' стр.'||is1.nord||'-т.'||is1.idgds ahead1 FROM tst\_invoice\_str is1 WHERE is1.idinv=i1.id

)

)

order by НакладныеДубликаты



8. Написать функцию, проверяющую возможность удаления поставщика (id поставщика — параметр функции). Функция возвращает текст. Если удаление возможно, то вернуть слово 'Возможно', а если нет, то перехватить ошибку и вернуть текст ошибки. После выполнения скрипта поставщик не должен быть удален, т.е. функцией выполняется только проверка возможности удаления.

**CREATE or REPLACE FUNCTION delete\_possibility(\_id integer) RETURNS text AS $$**

**DECLARE**

**foreignkeyscount integer;**

**v\_stack text;**

**v\_error\_stack text;**

**contractor text;**

**BEGIN**

**WITH a AS (select 1 FROM tst\_invoice WHERE idctr = \_id)**

**SELECT count(\*) INTO foreignkeyscount FROM a;**

**Select sname into contractor from tst\_contractor where id=\_id;**

**if foreignkeyscount >0 then**

**DELETE FROM tst\_invoice WHERE id=\_id;**

**else**

**if char\_length(contractor)>0 then**

**SELECT FORMAT('Удаление поставщика %s ВОЗМОЖНО! !Внешние ссылки в таблице tst\_invoice отсутствуют.',contractor) into v\_stack;**

**else**

**SELECT FORMAT('Поставщик с ID=%s в базе отсутствует!',\_id) into v\_stack;**

**end if;**

**end if;**

**return v\_stack;**

**EXCEPTION WHEN OTHERS THEN**

**GET STACKED DIAGNOSTICS v\_error\_stack = PG\_EXCEPTION\_DETAIL;**

**SELECT FORMAT('Удаление поставщика %s НЕВОЗМОЖНО! %s',contractor,v\_error\_stack) into v\_stack;**

**return v\_stack;**

**END;**

**$$ LANGUAGE plpgsql;**

**select delete\_possibility(1) as Возможность\_удаления\_поставщика;**

**--select delete\_possibility(7) as Возможность\_удаления\_поставщика;**