



Антон

CREATE OR REPLACE PROCEDURE PROC(COMPANY\_V CUSTOMERS.COMPANY%TYPE) IS

CURSOR cur IS

select order\_num, customers.company,

(select avg (amount \* price) from orders inner join products on products.mfr\_id = orders.mfr

and products.product\_id = orders.product where company = COMPANY\_V) average

from orders inner join customers

on orders.cust = customers.cust\_num where company = COMPANY\_V;

BEGIN

for o in cur

loop

dbms\_output.put\_line(cur%rowcount||' '||o.order\_num||' '||o.company || '' || o.average);

end loop;

EXCEPTION

WHEN OTHERS

THEN

dbms\_output.put\_line('error');

END;

BEGIN

PROC('JCP Inc.');

END;

drop function getCount

create or replace function getCount(userId number, my\_year integer) return number as

counter number;

begin

select count(\*) into counter from ORDERS where ORDERS.cust = userId and my\_year = extract(year from ORDERS.ORDER\_DATE);

return counter;

end;

set serveroutput on;

begin

dbms\_output.put\_line(getCount(2101, 2008));

end;

5 ВАРИАНТ

select \* from orders;

select \* from customers;

select \* from salesreps;

select \* from offices;

select \* from products;

CREATE OR REPLACE PROCEDURE addTenPercentsToProductPrice(pproduct\_id products.product\_id%type)

as

oldPrice products.price%type;

begin

SELECT price into oldPrice FROM products WHERE product\_id = pproduct\_id;

UPDATE products set price = price \* 1.1 WHERE product\_id = pproduct\_id;

dbms\_output.put\_line('old price: '||oldPrice||'; new price: '|| oldPrice \* 1.1);

exception

when no\_data\_found then

dbms\_output.put\_line('NO DATA FOUND: '||sqlerrm);

when others then

dbms\_output.put\_line(sqlerrm);

end;

begin

addTenPercentsToProductPrice('2A45C');

end;

CREATE or replace function getOrdersByCustomerAndYear(

pyear orders.order\_date%type,

pcust customers.cust\_num%type

)return number

as

counter number;

begin

SELECT count(\*) into counter FROM orders WHERE EXTRACT(Year FROM order\_date) = EXTRACT(Year FROM pyear);

return counter;

end;

begin

dbms\_output.put\_line(getOrdersByCustomerAndYear('2008-11-11', 2103));

end;

5 вариант

select count(\*) into counter from ORDERS where ORDERS.cust = userId and my\_year = extract(year from ORDERS.ORDER\_DATE);

return counter;

end;

set serveroutput on;

begin

dbms\_output.put\_line(getCount(2101, 2008));

end;

--2 proc

create or replace procedure showGoods(userId number) --2108

as

cursor corders is

SELECT

p.PRODUCT\_ID,

p."DESCRIPTION",

o.amount \* p.price as "COST"

FROM orders o inner join

products p on product\_id = product and o.cust = userId

order by "COST" desc;

cursorRow corders%rowtype;

begin

open corders;

loop

fetch corders into cursorRow;

exit when corders%notfound;

dbms\_output.put\_line(cursorRow.product\_id||': '||cursorRow.description||': '||cursorRow.cost);

end loop;

exception

when others then

dbms\_output.put\_line(sqlerrm);

close corders;

end;

begin

showGoods(2108);

end;

SELECT \* FROM orders

select \* from products

create or replace function period\_count()

as

begin

end;

select \* from orders where order\_date between DATE '2007-01-01' and DATE '2008-01-01';

8 ВАРИАНТ

select \* from orders;

select \* from customers;

select \* from salesreps;

select \* from offices;

select \* from products;

create or replace procedure ordNev(

date\_nev integer

)

as

cursor cproducts is SELECT \* FROM products;

cproductsRow cproducts%rowtype;

cursor corders is SELECT \* FROM orders WHERE date\_nev != extract(Year from order\_date);

cordersRow corders%rowtype;

flag boolean := false;

begin

open corders;

loop

fetch corders into cordersRow;

exit when corders%notfound;

open cproducts;

flag := false;

loop

fetch cproducts into cproductsRow;

exit when cproducts%notfound;

if cordersRow.product = cproductsRow.product\_id then

flag := true;

end if;

end loop;

close cproducts;

if flag then

dbms\_output.put\_line(cordersRow.product);

end if;

end loop;

close corders;

end;

begin

ordNev(2008);

end;

3 ВАРИАНТ

select \* from orders;

select \* from customers;

select \* from salesreps;

select \* from offices;

select \* from products;

create or replace procedure ordDesc(

date\_from orders.order\_date%type,

date\_to orders.order\_date%type

)

as

cursor ccusts is select \* from orders where order\_date between date\_from and date\_to order by amount desc;

ccust ccusts%rowtype;

begin

open ccusts;

loop

fetch ccusts into ccust;

exit when ccusts%notfound;

dbms\_output.put\_line(ccust.cust || ' ' || ccust.amount);

end loop;

close ccusts;

exception

when others then dbms\_output.put\_line(sqlerrm);

end;

begin

ordDesc(DATE '2007-01-01',DATE '2008-01-01');

end;

ВАРИАНТ6

select \* from orders;

select \* from customers;

select \* from salesreps;

select \* from offices;

select \* from products;

CREATE OR REPLACE PROCEDURE printOrdersPriceMoreThan(pamount orders.amount%type) as

value number;

cursor corders is SELECT \* FROM orders WHERE amount > pamount;

cursorRow corders%rowtype;

begin

open corders;

loop

fetch corders into cursorRow;

exit when corders%notfound;

dbms\_output.put\_line(cursorRow.order\_num||': '||cursorRow.amount);

end loop;

exception

when others then

dbms\_output.put\_line(sqlerrm);

close corders;

end;

set serveroutput on;

begin

printOrdersPriceMoreThan(22000);

end;

CREATE OR REPLACE FUNCTION getOrdersByUser(name customers.company%type) return number

as

counter number;

begin

select count(\*) into counter from orders ord

INNER JOIN customers cust on ord.cust = cust.cust\_num

WHERE cust.company = name;

return counter;

end;

begin

dbms\_output.put\_line(getOrdersByUser('Acme Mfg.'));

end;

ВАРИАНТ4

select \* from orders;

select \* from customers;

create or replace procedure itsjustaperiod(

date\_from orders.order\_date%type,

date\_to orders.order\_date%type

)

as

cursor ccusts is select \* from orders where order\_date between date\_from and date\_to;

ccust ccusts%rowtype;

begin

open ccusts;

loop

fetch ccusts into ccust;

exit when ccusts%notfound;

dbms\_output.put\_line(ccust.cust || ' ' || ccust.order\_date);

end loop;

close ccusts;

exception

when others then dbms\_output.put\_line(sqlerrm);

end;

begin

itsjustaperiod(DATE '2007-01-01',DATE '2008-01-01');

end;

select \* from orders where order\_date between DATE '2007-01-01' and DATE '2008-01-01';

ВАРИАНТ7

select \* from customers;

select \* from orders;

create or replace procedure deleteFrom (

cost orders.amount%type

)

as

cursor porders is SELECT \* FROM orders;

porder porders%rowtype;

begin

delete from orders where amount < cost;

open porders;

loop

fetch porders into porder;

exit when porders%notfound;

dbms\_output.put\_line(porder.order\_num||' '||porder.amount);

end loop;

close porders;

exception

when others then

dbms\_output.put\_line(sqlerrm);

end;

begin

deletefrom(4000);

end;

CREATE or replace function countProduct(

product\_name orders.product%type

)return number

as

count\_pr number;

begin

select count(\*) into count\_pr from orders where product = product\_name;

return count\_pr;

end;

begin

dbms\_output.put\_line(countproduct('2A44L'));

end;

ЯН

* CREATE OR REPLACE PROCEDURE PROC(COMPANY\_V [CUSTOMERS.COMPANY](https://vk.com/away.php?to=http%3A%2F%2FCUSTOMERS.COMPANY&cc_key=)%TYPE) IS  
  CURSOR cur IS  
  select order\_num, [customers.company](https://vk.com/away.php?to=http%3A%2F%2Fcustomers.company&cc_key=" \t "_blank),  
  (select avg (amount \* price) from orders inner join products on products.mfr\_id = orders.mfr  
  and products.product\_id = orders.product where company = COMPANY\_V) average  
  from orders inner join customers  
  on orders.cust = customers.cust\_num where company = COMPANY\_V;  
  BEGIN  
  for o in cur  
  loop  
  dbms\_output.put\_line(cur%rowcount||' '||o.order\_num||' '||[o.company](https://vk.com/away.php?to=http%3A%2F%2Fo.company&cc_key=" \t "_blank) || '' || o.average);  
  end loop;  
  EXCEPTION  
  WHEN OTHERS  
  THEN  
  dbms\_output.put\_line('error');  
  END;  
    
  BEGIN  
  PROC('JCP Inc.');  
  END;

ЯГОР

* create or replace procedure OfficeInfo(nameOffice number)  
  as  
  cursor pp is select mgr from offices;  
  sumNumber number;  
  buf number;  
  begin  
  Select sum(offices.sales) into sumNumber from SALESREPS join offices on salesreps.rep\_office = [offices.office](https://vk.com/away.php?to=http%3A%2F%2Foffices.office&cc_key=" \t "_blank) where office = nameOffice group by [offices.office](https://vk.com/away.php?to=http%3A%2F%2Foffices.office&cc_key=" \t "_blank);  
  dbms\_output.put\_line(sumNumber);  
    
  open pp;  
  loop  
  fetch pp into buf;  
  dbms\_output.put\_line(buf);  
  exit when pp%notfound;  
  end loop;  
  close pp;  
  end;  
    
    
  begin  
  OfficeInfo(12);  
  end;  
  select \* from orders;  
    
    
  select \* from orders;  
  select \* from offices;  
    
  create or replace function OfficeFunc(nameOffice number, startD date, endD date)  
  return number  
  as  
  sumNumber number;  
  begin  
  Select count(offices.sales) into sumNumber from orders join offices on offices.mgr = orders.rep where office = nameOffice and order\_date < endD and order\_date > startD;  
  return sumNumber;  
  end;  
    
  begin  
  dbms\_output.put\_line(OfficeFunc(11, to\_date('12/05/2003', 'dd/mm/yyyy'), to\_date('12/05/2023', 'dd/mm/yyyy')));  
  end;  
    
    
  Select count(offices.sales) from orders join offices on offices.mgr = orders.rep where office = 11 and order\_date > to\_date('12/05/2003', 'dd/mm/yyyy') and order\_date < to\_date('12/05/2023', 'dd/mm/yyyy');