БД:

| EMP |  | DEPT |
| --- | --- | --- |
| e\_no  e\_name  date\_hire  salary  address  d\_no |  | d\_no  d\_name  location |

Первый пакет – обработка данных для отделов:

1. Создание нового отдела
2. Подсчет количества сотрудников в указанном отделе
3. Подсчет количества сотрудников в каждом отделе
4. Изменение названия указанного отдела га новое
5. Перевод указанного отдела в новый город.

**Спецификация**

create or replace package dept\_pkg as

v\_count number;

v\_msg varchar2(200);

e\_error exception;

procedure new\_dept(p\_dname in dept.d\_name%type, p\_location in dept.location%type);

function emp\_count(p\_dname in dept.d\_name%type) return varchar2;

procedure emp\_count\_all;

procedure rename\_dept(p\_old\_name in dept.d\_name%type, p\_new\_name in dept.d\_name%type);

procedure move\_dept(p\_dname in dept.d\_name%type, p\_new\_location in dept.location%type);

end dept\_pkg;

**Тело пакета**  
  
create or replace package body dept\_pkg as

procedure new\_dept(p\_dname in dept.d\_name%type, p\_location in dept.location%type) is

v\_new\_id dept.d\_no%type;

begin

select count(\*) into v\_count from dept where upper(d\_name) = upper(p\_dname);

if v\_count > 0 then

raise e\_error;

end if;

select max(d\_no) + 1 into v\_new\_id from dept;

insert into dept(d\_no, d\_name, location)

values (v\_new\_id, p\_dname, p\_location);

dbms\_output.put\_line('Отдел "' || p\_dname || '" создан успешно.');

exception

when e\_error then

raise\_application\_error(-20001, 'Такой отдел уже существует.');

end new\_dept;

function emp\_count(p\_dname in dept.d\_name%type) return varchar2 is

v\_result varchar2(100);

begin

select count(\*) into v\_count from dept where upper(d\_name) = upper(p\_dname);

if v\_count = 0 then

raise e\_error;

end if;

select count(\*) into v\_count

from emp

where d\_no = (select d\_no from dept where upper(d\_name) = upper(p\_dname));

v\_result := 'Количество сотрудников в отделе "' || p\_dname || '" — ' || v\_count;

return v\_result;

exception

when e\_error then

return 'Ошибка: отдела "' || p\_dname || '" не существует.';

end emp\_count;

procedure emp\_count\_all is

begin

for rec in (select d\_name, (select count(\*) from emp where emp.d\_no = dept.d\_no) as cnt from dept)

loop

dbms\_output.put\_line('Отдел "' || rec.d\_name || '": ' || rec.cnt || ' сотрудников');

end loop;

end emp\_count\_all;

procedure rename\_dept(p\_old\_name in dept.d\_name%type, p\_new\_name in dept.d\_name%type) is

begin

select count(\*) into v\_count from dept where upper(d\_name) = upper(p\_old\_name);

if v\_count = 0 then

raise e\_error;

end if;

update dept set d\_name = p\_new\_name where upper(d\_name) = upper(p\_old\_name);

dbms\_output.put\_line('Отдел "' || p\_old\_name || '" переименован в "' || p\_new\_name || '".');

exception

when e\_error then

raise\_application\_error(-20002, 'Ошибка: отдел "' || p\_old\_name || '" не найден.');

end rename\_dept;

procedure move\_dept(p\_dname in dept.d\_name%type, p\_new\_location in dept.location%type) is

begin

select count(\*) into v\_count from dept where upper(d\_name) = upper(p\_dname);

if v\_count = 0 then

raise e\_error;

end if;

update dept set location = p\_new\_location where upper(d\_name) = upper(p\_dname);

dbms\_output.put\_line('Отдел "' || p\_dname || '" переведён в город "' || p\_new\_location || '".');

exception

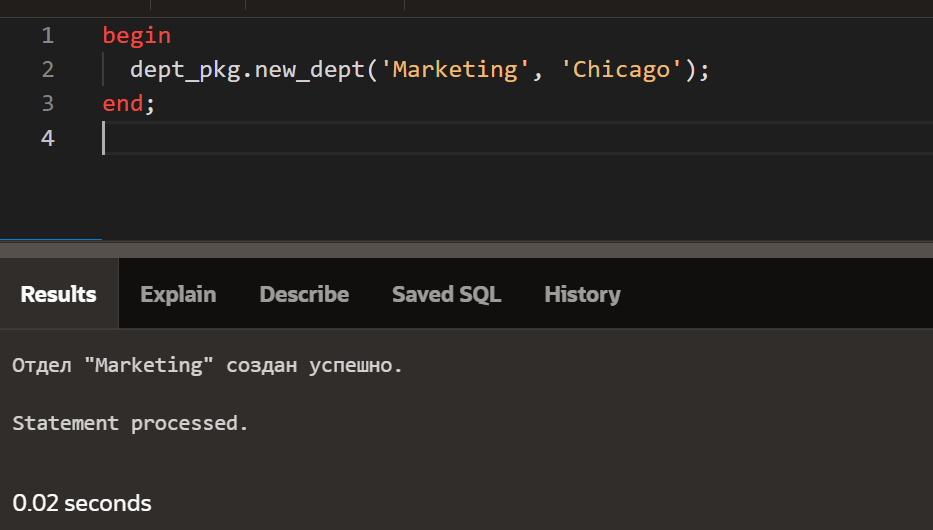
when e\_error then

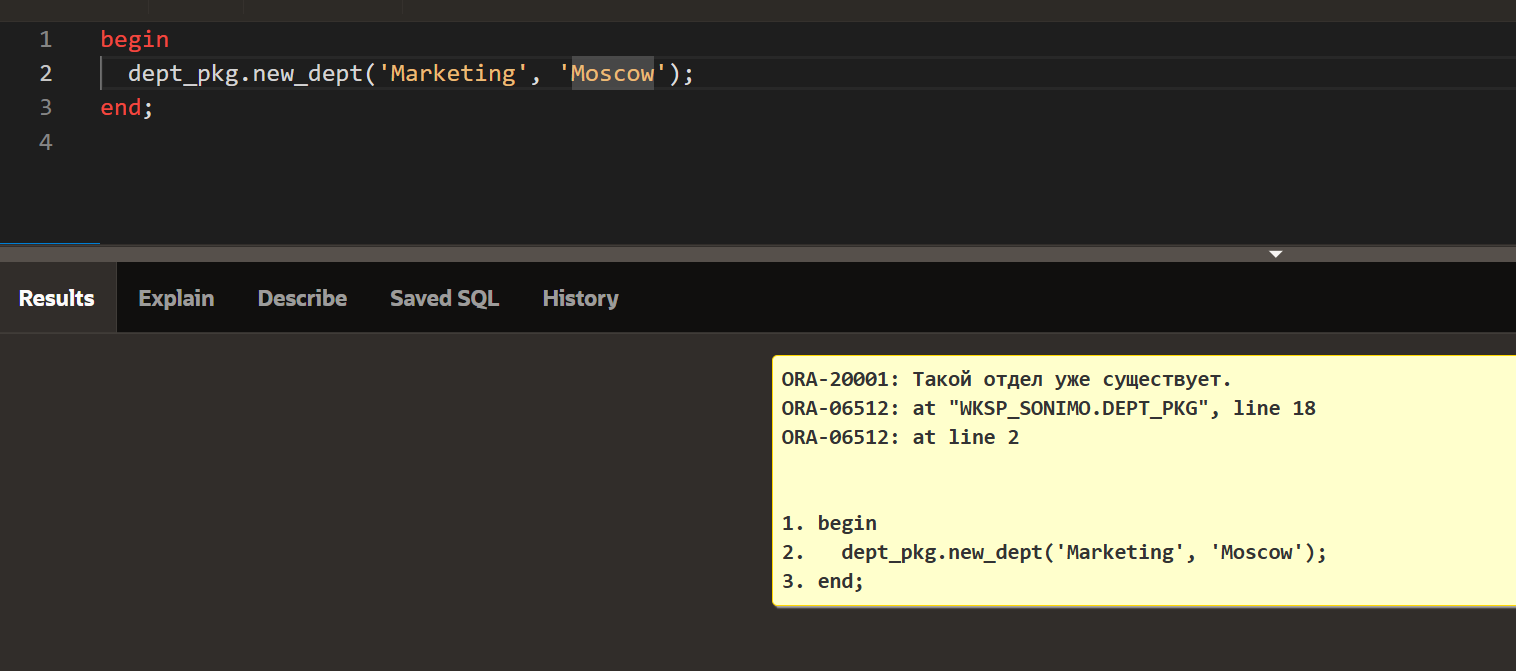
raise\_application\_error(-20003, 'Ошибка: отдел "' || p\_dname || '" не найден.');

end move\_dept;

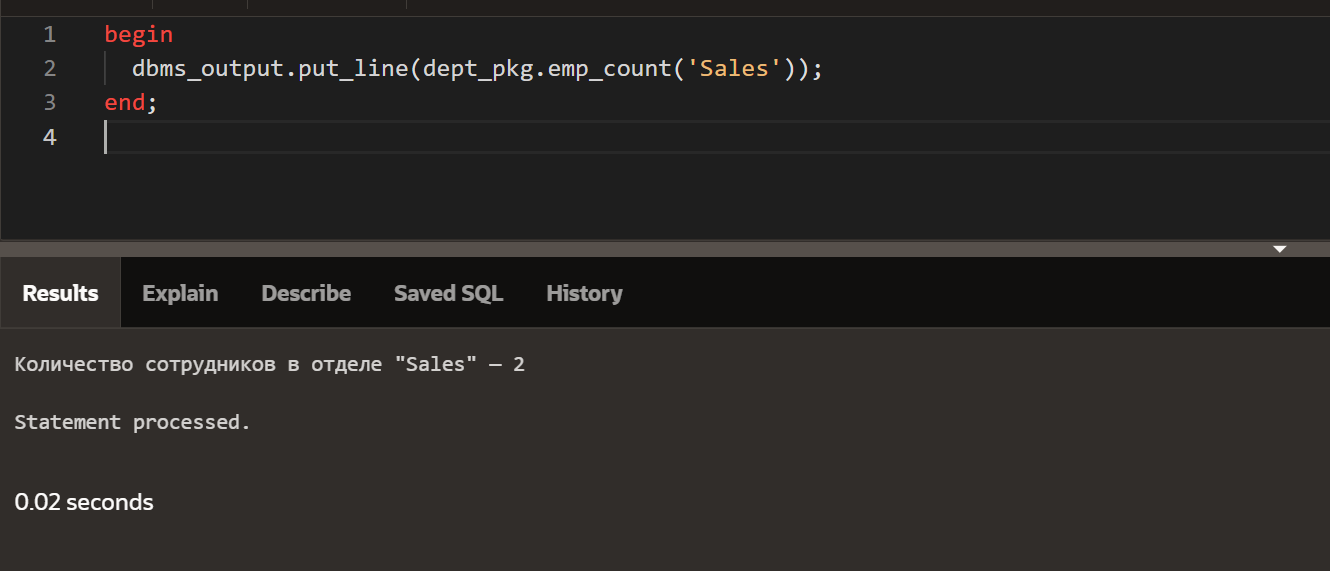
end dept\_pkg;

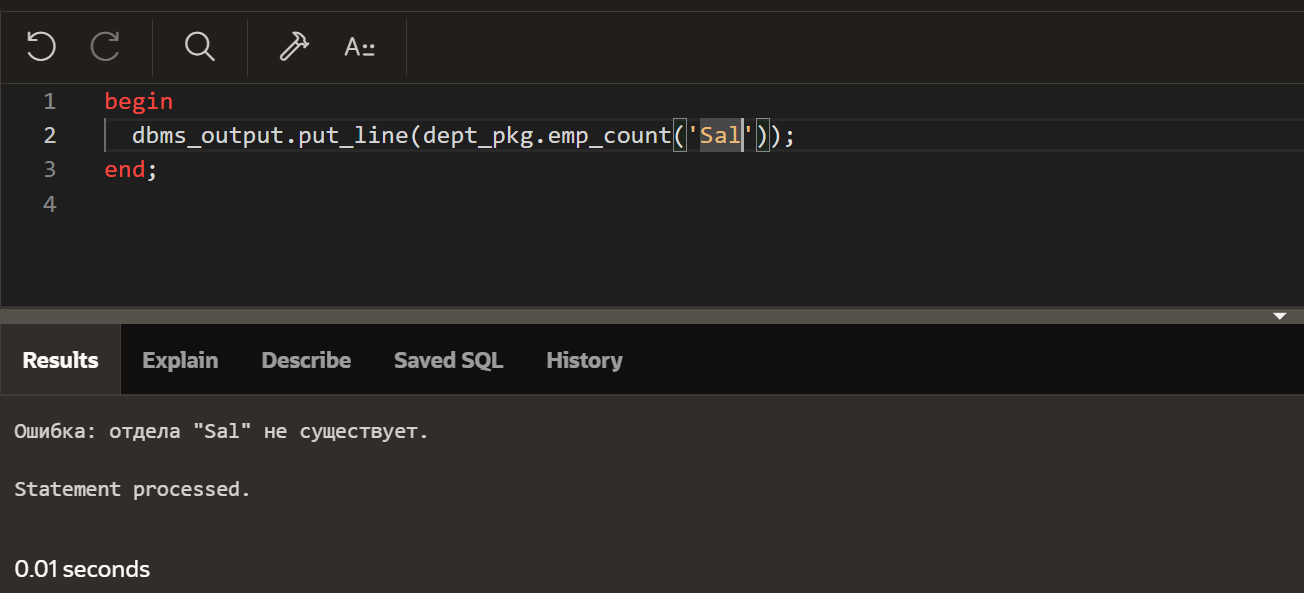
добавление нового отдела



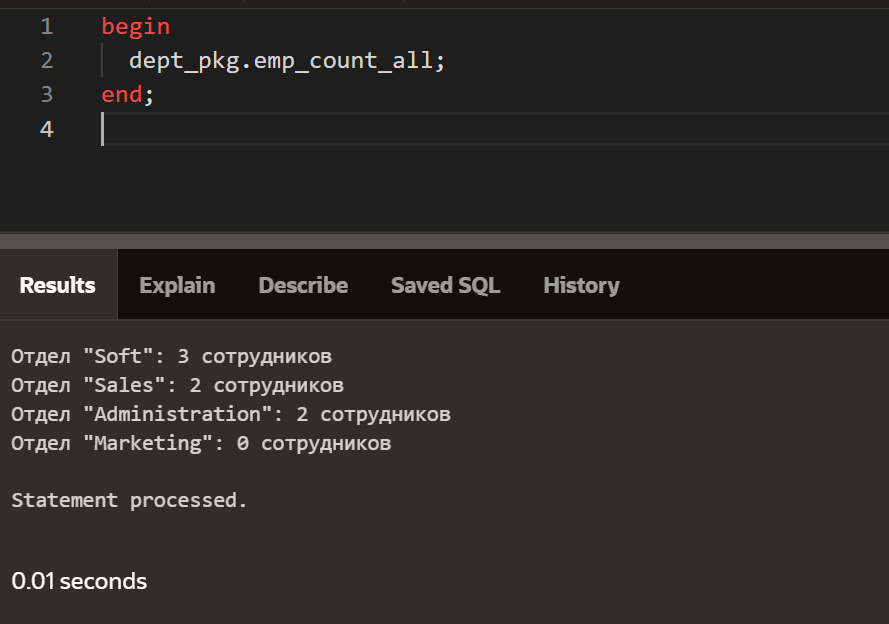


Количество сотрудников в отделе

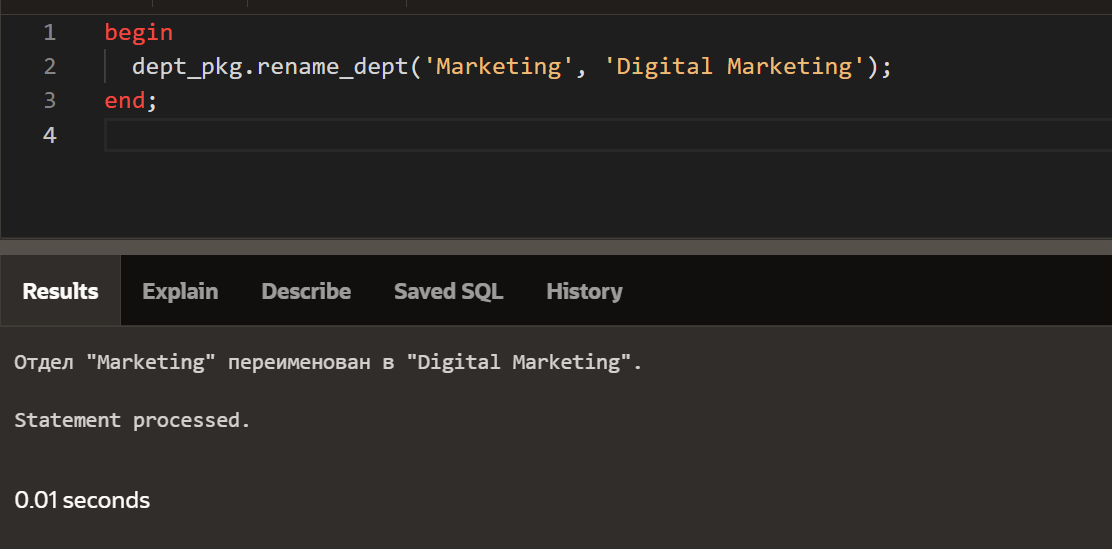


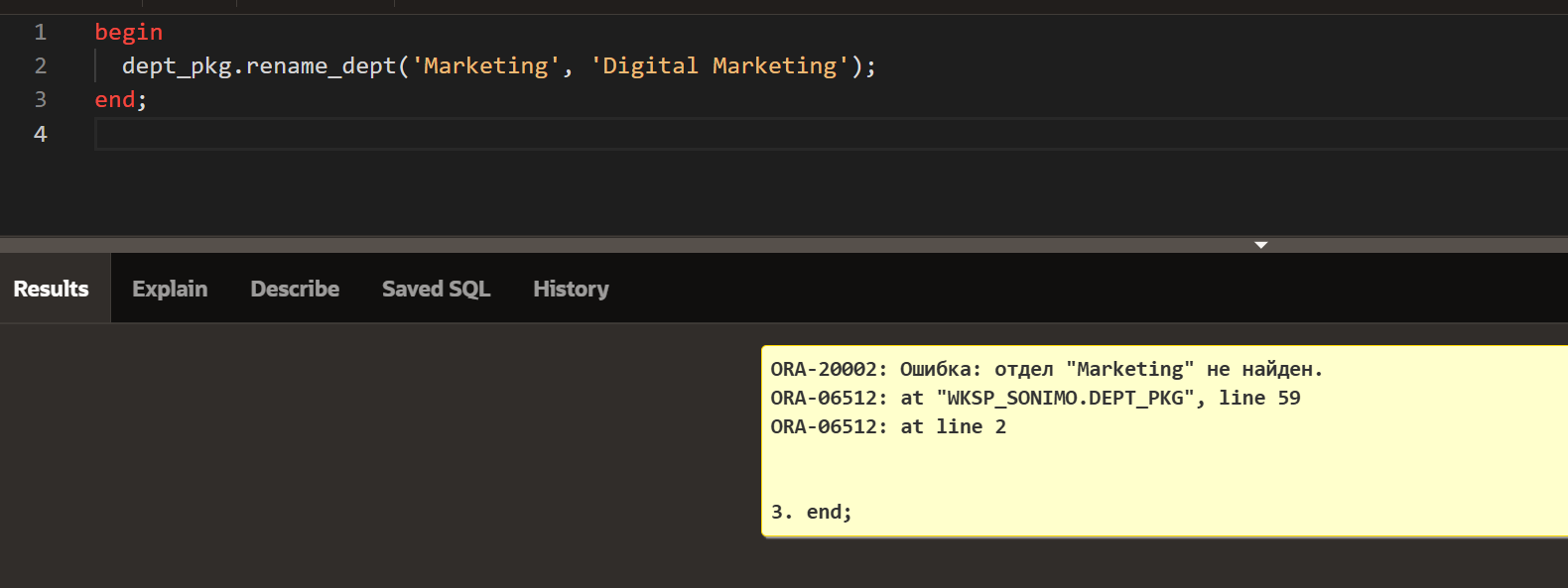


Количество сотрудников во всех отделах

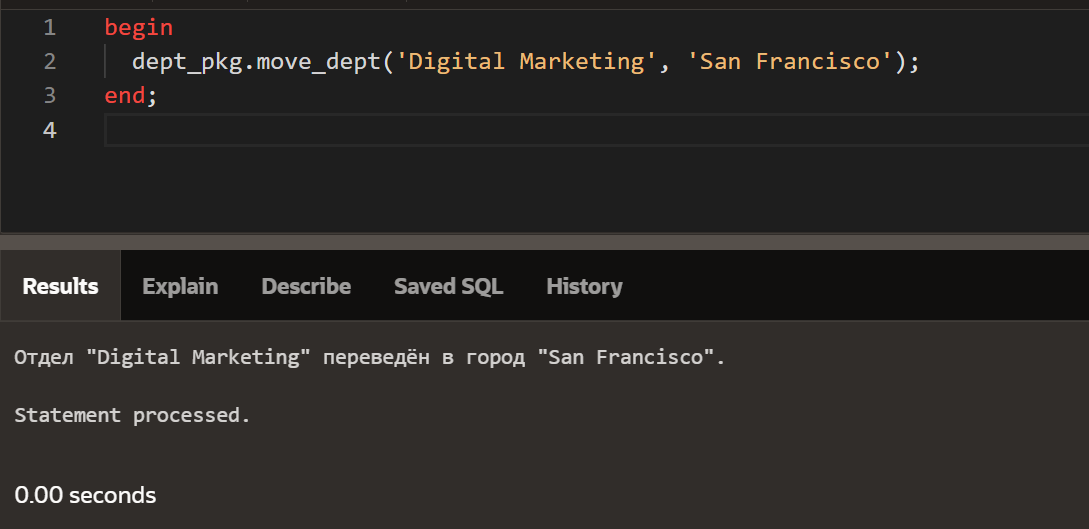


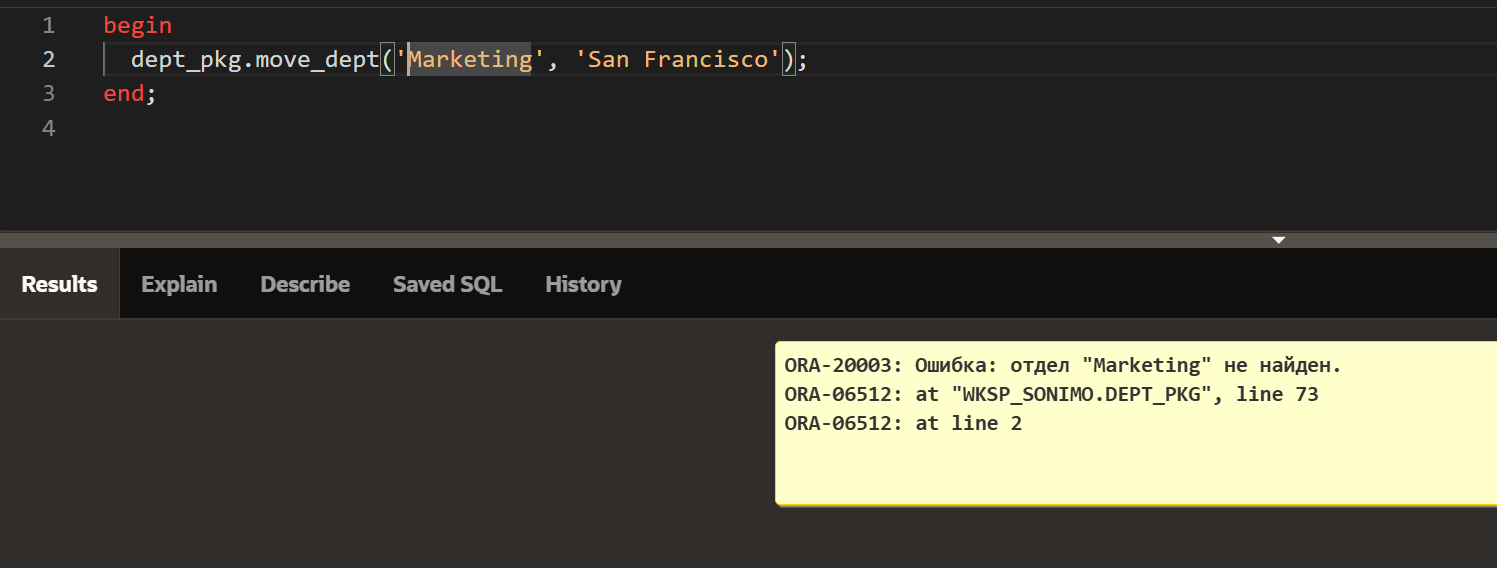
Переименование отдела





перевод отдела





Второй пакет – обработка данных для сотрудников фирмы:

1. Зачисление нового сотрудника
2. Изменение зарплаты указанного сотрудника на заданную величину
3. Расчет ежегодного объема зарплаты для указанного сотрудника
4. Расчет ежегодного объема зарплаты сотрудников в указанном отделе.
5. Перевод сотрудника в указанный отдел.

**Спецификация**

create or replace package emp\_pkg as

v\_count integer;

e\_error exception;

procedure hire\_employee(p\_name in emp.e\_name%type, p\_salary in emp.salary%type, p\_address in emp.address%type, p\_dept in dept.d\_name%type);

procedure update\_salary(p\_emp in emp.e\_name%type, p\_address in emp.address%type, p\_new\_salary in emp.salary%type);

function annual\_salary\_emp(p\_emp in emp.e\_name%type, p\_address in emp.address%type) return varchar2;

function annual\_salary\_dept(p\_dept in dept.d\_name%type) return varchar2;

procedure transfer\_employee(p\_emp in emp.e\_name%type, p\_address in emp.address%type, p\_new\_dept in dept.d\_name%type);

end emp\_pkg;

**Тело пакета**   
  
create or replace package body emp\_pkg as

-- Процедура для зачисления нового сотрудника

procedure hire\_employee(p\_name in emp.e\_name%type, p\_salary in emp.salary%type, p\_address in emp.address%type, p\_dept in dept.d\_name%type) is

v\_emp\_id emp.e\_no%type;

v\_dept\_id dept.d\_no%type;

v\_count integer;

begin

-- Проверка на существующего сотрудника с таким же именем и адресом

select count(\*) into v\_count from emp where e\_name = p\_name and address = p\_address;

if v\_count > 0 then

raise e\_error;

end if;

select d\_no into v\_dept\_id from dept where d\_name = p\_dept;

select nvl(max(e\_no), 0) + 1 into v\_emp\_id from emp;

insert into emp (e\_no, e\_name, salary, address, d\_no) values (v\_emp\_id, p\_name, p\_salary, p\_address, v\_dept\_id);

exception

when e\_error then

raise\_application\_error(-20001, 'Сотрудник с таким именем и адресом уже существует');

when no\_data\_found then

raise\_application\_error(-20002, 'Отдел с таким названием не найден');

end hire\_employee;

-- Процедура для изменения зарплаты сотрудника

procedure update\_salary(p\_emp in emp.e\_name%type, p\_address in emp.address%type, p\_new\_salary in emp.salary%type) is

v\_emp\_id emp.e\_no%type;

begin

select count(\*) into v\_count from emp where e\_name = p\_emp and address = p\_address;

if v\_count = 0 then

raise e\_error;

end if;

update emp set salary = p\_new\_salary where e\_name = p\_emp and address = p\_address;

exception

when e\_error then

raise\_application\_error(-20001, 'Сотрудник с таким именем и адресом не найден');

end update\_salary;

function annual\_salary\_emp(p\_emp in emp.e\_name%type, p\_address in emp.address%type) return varchar2 is

v\_salary emp.salary%type;

v\_result varchar2(100);

begin

select count(\*) into v\_count from emp where e\_name = p\_emp and address = p\_address;

if v\_count = 0 then

raise e\_error;

end if;

-- Расчет годовой зарплаты сотрудника

select salary \* 12 into v\_salary from emp where e\_name = p\_emp and address = p\_address;

v\_result := 'Годовая зарплата сотрудника ' || p\_emp || ' составляет ' || v\_salary || ' лей';

return v\_result;

exception

when e\_error then

raise\_application\_error(-20001, 'Такого сотрудника нет');

end;

-- Функция для расчета годовой зарплаты сотрудников в отделе

function annual\_salary\_dept(p\_dept in dept.d\_name%type) return varchar2 is

v\_salary number;

v\_result varchar2(1000);

begin

select count(\*) into v\_count from dept where d\_name = p\_dept;

if v\_count = 0 then

raise e\_error;

end if;

select sum(salary \* 12) into v\_salary from emp

where d\_no = (select d\_no from dept where d\_name = p\_dept);

v\_result := 'Годовая зарплата сотрудников отдела ' || p\_dept || ' составляет ' || v\_salary || ' лей';

return v\_result;

exception

when e\_error then

raise\_application\_error(-20001, 'Такого отдела нет');

end;

-- Процедура для перевода сотрудника в новый отдел

procedure transfer\_employee(p\_emp in emp.e\_name%type, p\_address in emp.address%type, p\_new\_dept in dept.d\_name%type) is

v\_emp\_id emp.e\_no%type;

v\_new\_dept\_id dept.d\_no%type;

begin

select count(\*) into v\_count from emp where e\_name = p\_emp and address = p\_address;

if v\_count = 0 then

raise e\_error;

end if;

-- Проверка существования нового отдела

select count(\*) into v\_count from dept where d\_name = p\_new\_dept;

if v\_count = 0 then

raise e\_error;

end if;

select d\_no into v\_new\_dept\_id from dept where d\_name = p\_new\_dept;

update emp set d\_no = v\_new\_dept\_id where e\_name = p\_emp and address = p\_address;

exception

when e\_error then

raise\_application\_error(-20001, 'Сотрудник или отдел не найден');

end;

end emp\_pkg;

